

NORTHWEST NAZARENE UNIVERSITY

Full Stack Application Development

THESIS


Submitted to the Department of Mathematics and Computer Science  
in partial fulfillment of the requirements  
for the degree of  
BACHELOR OF ARTS


Kellen Fairchild  
2020


THESIS  
Submitted to the Department of Mathematics and Computer Science  
in partial fulfillment of the requirements  
for the degree of  
BACHELOR OF ARTS


Kellen Fairchild  
2020

Full Stack Mission Application Development

Author:   
Kellen Fairchild

Approved:   
Dale Hamilton, Ph.D., Associate Professor of Computer Science,  
Department of Mathematics & Computer Science, Faculty Advisor

Approved:   
Joe Gorman, D.Min., Professor of Practical Theology, Department  
of Philosophy and Religion, Second Reader

Approved:   
Barry L. Myers, Ph.D., Chair, Department of Mathematics &  
Computer Science

## **Abstract**

Full-Stack Mission Application Development.

FAIRCHILD, KELLEN (Department of Math and Computer Science).

Missions is a very significant part of the life of many practicing Christians worldwide. However, like any endeavor aiming to accomplish a substantial goal, mission projects require significant personnel and logistics management, and if this capacity is lacking, projects can become remarkably difficult to manage. To help begin to remedy this problem, the Mission2Go system was designed. Using server-side JavaScript and Microsoft SQL Server, this application allows for the creation of teams, projects, and the registration of members. The program allows the user to view all projects scheduled to occur in the next 15 days and gives the option to edit project and team details, assign a different team to a project, change budget details, change team members and project dates, and other such pertinent information. By having this application hosted on a server, this app could make life drastically easier for regional coordinators, pastors, church boards, and more. When this application is complete, it will be able to create and control login sessions into the application, and facilitate more efficient administration of mission activities through registrations and team reassignments.

## **ACKNOWLEDGMENTS**

Without the support of the Mission2Go team and Josh Williams from the Church of the Nazarene's Global Mission IT Team, this project would never have become an opportunity to learn and to serve the kingdom of God. I would also like to express my gratitude to Northwest Nazarene University's computer science, math, and theology faculty for their help in giving this project the tools necessary to move forward.

## **TABLE OF CONTENTS**

Title Page.....	i
Signature Page.....	ii
Abstract.....	iii
Acknowledgments.....	iv
Table of Contents.....	v
List of Figures.....	vi
Introduction	
Project Overview.....	1
Body	
JavaScript Frameworks and Database Integration.....	2
Application Security Measures.....	3
Salts and Hashes.....	3
User Sessions.....	5
Project Planning.....	8
Development.....	9
Database Schema.....	11
User Interface (UI) Creation Via Pug.js.....	13
Conclusion	
Conclusion.....	22
Future Work.....	23
References.....	25
Appendices	
A. UI Source.....	27
B. Server.js Source.....	127
C. Schema.....	261

## List of Figures

Figure 1. Terminal Running Redis.....	7
Figure 2. Project Stack (Vasquez, 2019).....	8
Figure 3. Database Schema.....	12
Figure 4. Example Array Creation.....	15
Figure 5. Application Home Screen.....	17
Figure 6. Example Data Entry Form.....	18
Figure 7. Client Side View.....	20
Figure 8. View Handler- Back End.....	21

# Project Overview

Managing projects, teams, supplies, and general logistics for any sort of work project is absolutely essential for efficiency, productivity, and accountability. As the Church of the Nazarene seeks to make Christ known, it must stay organized in order to ensure that teams and supplies are where they are needed, when they are needed. A web-based application allowing administrators and coordinators to keep track of their teams, projects, and funds is a substantial need for many mission partnerships and Christian denominations. This application, to be effective after implementation, would need to be fast, lightweight, and easy to use. In pursuit of growth in the global church, it is fortunate that there are a variety of methods that have appeared over the last several years which are well-suited for this type of work, making the present era a great time to develop such an application. After the project and its objectives were explained by Mr. Roger Kellogg, a Work and Witness coordinator for the Church of the Nazarene, it was decided that the application would need to be capable of registering or editing teams and users, allowing users to join teams, provide a means to register new missions projects with relevant details, and provide the capability to assign a team to a project or to change a team's registration status. These functionalities were implemented with a Microsoft SQL Server database as a back-end, and with a JavaScript runtime called Node.js with front-end-focused libraries Express.js and Pug.js to allow users to view, delete, and edit records within the database from a web browser. These technologies and their uses and purposes in the final project will be explained in greater detail when they are reached.

# JavaScript Frameworks and Database Integration

JavaScript is a remarkably versatile language. While this scripting language has long been used for client-side programming, several frameworks which facilitate the writing of server-side JavaScript exist. One particularly well-known and popular example of such a framework is the open-source framework Node.js (“Node.js”). The internet is full of documentation for installing a variety of packages which can facilitate diverse tasks. Such tasks include managing browser cookies or configuring database connections with parallel threads, for example. One of the more widely-used and discussed packages for this framework is Express.js, which facilitates app development with intuitive syntax and consistent parameter names for request and response objects (“Node.js Web”). This framework is compatible with Pug.js, a templating engine that can render modified HTML (“Getting Started”). Thus, all of the basic components for the structure of this application were in place.

When dealing with significant amounts of structured data, a database is an ideal solution for working with said data. This is because databases are designed to manipulate data in an efficient manner. A database prototype, developed in Microsoft Access, was received from Roger Kellogg, the project stakeholder. Fortunately, tools exist in Microsoft’s SQL Server Management Studio (SSMS) application which can migrate data and schemata from Access to Microsoft SQL Server (Microsoft 2020). Once the schema and data were migrated, connecting to this SQL database from a Node.js application was a very straightforward process due to the MSSQL package available from NPM, the Node Package Manager (“Mssql -npm”). This package provides the ability to pass values



back and forth from the client's user interface to the database and vice versa as necessary. This means that end users can view, edit, and delete records from the database without ever being aware of the database's inner workings.

# Application Security Measures

Security is an important consideration when developing applications. Without some form of security strategy, potentially sensitive data could be exposed, or the application could be misused or abused, whether purposefully or on accident. In order to make the Mission2Go application more secure, two strategies were implemented. Firstly, salts and hashes were applied to user passwords to obfuscate them within the database, making it more difficult for attackers to utilize them. Secondly, managing user sessions within the application ensures that users are logged out automatically after a given period to prevent others from gaining unauthorized access.

## Salts and Hashes

When considering how and where to store sensitive information, such as user passwords, social security numbers, addresses, and other such personally identifiable information, it is universally agreed that storing such information as plain text in a database is not advisable. Storing information in this way leaves it remarkably vulnerable to interception by attackers to use for their nefarious purposes. This means that developers must apply strategies to ensure that private information is as close to secure and indecipherable as possible, while still being able to decrypt these values when needed for use. Even as encryption algorithms grow increasingly complex, (and theoretically

more computationally expensive and time-consuming to crack), modern hashing and encryption algorithms can still be deciphered, leaving information security and privacy compromised. However, the addition of a randomly generated string called a salt can make sensitive data more difficult to decode, as detailed by Robert Morris and Ken Thompson of Bell Laboratories (1978). By adding a relatively long, randomly generated string to the password before putting it through the chosen hashing algorithm, the password becomes much more complex and therefore more difficult to crack, in theory.

For this project, I chose the SHA-512 hashing algorithm due to its ease of use in Microsoft SQL with the HASHBYTES function (“Hashbytes,” 2016). I also chose this method due to learning that this is perhaps the most relevant algorithm supported natively in MSSQL, according to Microsoft’s documentation on the HASHBYTES function (“Hashbytes,” 2016). A salt for the user’s password was generated using the uniqueidentifier datatype paired with the *NewID* function, as detailed in Microsoft’s documentation on these SQL keywords (“Uniqueidentifier” , 2017). The salt is stored along with the passwords. Authentication for login to the app works by utilizing the following procedure:

- 1 Send inputted username and password securely over HTTPS to the database.
- 2 Find the username and corresponding hashed password and salt.
- 3 Append the salt to the password, and hash the result.
- 4 If the result of the hash matches the value stored in the database, login is successful.

The advantages of this system are plentiful, and while there are potentially more secure strategies, I found spending excessive time on cryptography to be beyond the scope of this project. As a result, the best reasonable effort was made to ensure security in good faith. Additionally, it is likely this application will have relatively few users, at least initially, and attackers likely would not be able to find very much actionable intelligence anyway.

## User Sessions

The Node.js and Express.js frameworks have robust and secure session-managing middleware which uses encrypted cookies and a randomly-generated session ID to ensure users have permission to perform certain actions. In addition, if a user is forced to leave their workspace or forgets to log out, the session can be set to close on its own and log the user out after a set period of inactivity to prevent unauthorized persons from gaining access to sensitive information. Various packages are available for Node.js which allow the storage and access of sessions. Notably, keeping track of sessions with a file store system or in the database is not an ideal solution. Database and file accesses take time, and could slow down the application drastically, especially with multiple simultaneous users. Fortunately, there is an effectively perfect solution to this problem. This solution consists of two major parts. Firstly, the Linux Mint operating system was used with the MATE desktop environment (Linux Mark Institute, 2020). Secondly, a memory-based session store that is exclusive to Linux platforms, called Redis, was installed and implemented (Redis Labs, 2020). These pieces of software helped to circumvent the problem of the database bottleneck.

Once the system was migrated to its new operating system, the user session management solution could be implemented with Redis (Redis Labs, 2020). While there is a native memory cache for user sessions in Node.js, the authors of the package make it abundantly clear that it is prone to memory leaks and is suitable only for testing, not for production (“Express-session”). Redis states that companies such as StackOverflow and Twitter use their user session solution, and this suggests that it has a reputation for security and reliability (“Who’s Using Redis?”). Redis is, for all intents and purposes, an open-source database or data structure that is based entirely in memory, which has several advantages over conventional database and file session storage (“Redis”). A Linux terminal running a Redis instance can be seen in Figure 1. The documentation for Redis lists some of its advantages, including speed and volatility settings (Redis’s storage options are highly configurable), and optimization of memory usage for speed and efficiency (“Session Management”). Due to these advantages and recommendations from the overwhelming majority of sources I consulted on this matter, Redis was by far the best choice, even though it created a delay in development due to migration difficulties.

```
kfairchild@kfairchild-ThinkPad-P50: ~  
File Edit View Search Terminal Help  
kfairchild@kfairchild-ThinkPad-P50:~$ redis-server  
12882:C 28 Nov 2020 16:47:22.457 # oOoOoOoOoOoOo Redis is starting oOoOoOoOoOoOo  
12882:C 28 Nov 2020 16:47:22.457 # Redis version=6.0.8, bits=64, commit=00000000, m  
odified=0, pid=12882, just started  
12882:C 28 Nov 2020 16:47:22.457 # Warning: no config file specified, using the def  
ault config. In order to specify a config file use redis-server /path/to/redis.conf  
12882:M 28 Nov 2020 16:47:22.459 * Increased maximum number of open files to 10032  
(it was originally set to 1024).  
  
Redis 6.0.8 (00000000/0) 64 bit  
  
Running in standalone mode  
Port: 6379  
PID: 12882  
  
http://redis.io  
  
12882:M 28 Nov 2020 16:47:22.461 # Server initialized  
12882:M 28 Nov 2020 16:47:22.461 # WARNING overcommit_memory is set to 0! Backgroun  
d save may fail under low memory condition. To fix this issue add 'vm.overcommit_me  
memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcom  
mit_memory=1' for this to take effect.  
12882:M 28 Nov 2020 16:47:22.462 * Loading RDB produced by version 6.0.8  
12882:M 28 Nov 2020 16:47:22.462 * RDB age 760806 seconds  
12882:M 28 Nov 2020 16:47:22.462 * RDB memory usage when created 0.77 Mb  
12882:M 28 Nov 2020 16:47:22.462 * DB loaded from disk: 0.000 seconds  
12882:M 28 Nov 2020 16:47:22.462 * Ready to accept connections
```

Figure 1. Terminal Running Redis

# Project Planning

The Mission2Go Project was born when Mr. Roger Kellogg visited an Intro to Senior Project class at Northwest Nazarene University in early 2020. Shortly afterward, a team was assembled, the members met to discuss the basic requirements and goals for the project, and a system architecture was provided by freelance software developer Mr. Jorge Vasquez. Vasquez's project stack is diagrammed in Figure 2.

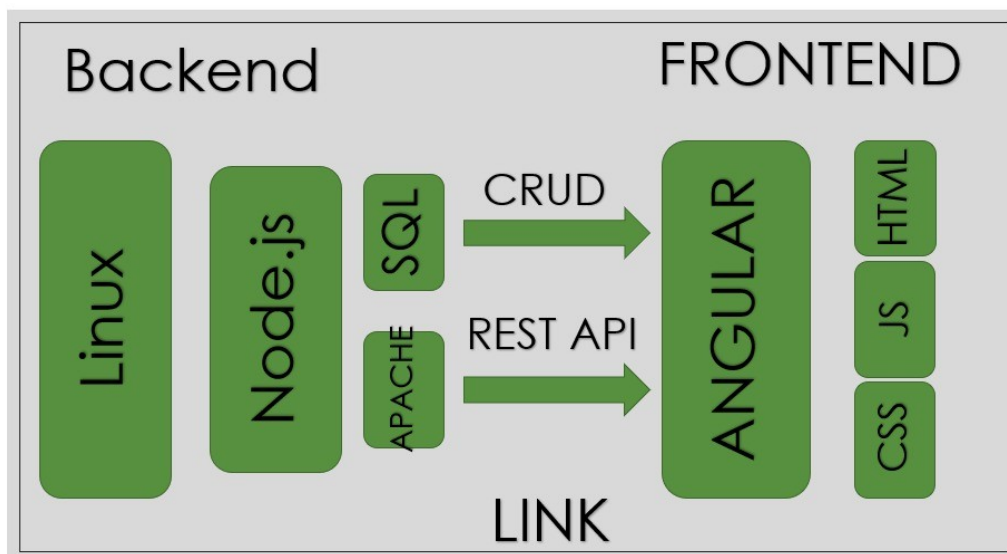


Figure 2. Project Stack (J. Vasquez, personal communication, 2020)

This portion of the project is focused on the server and database components. However, cascading style sheets written by Ms. Telah Dickerson, a senior computer science student at Northwest Nazarene University, will be implemented to improve the look of the front-end in the near future. Due to its high flexibility, speed, popularity, wealth of

documentation, and open-source nature, Node.js was chosen immediately as the back-end scripting language. While MySQL was initially chosen, due to the team's greater familiarity with Microsoft SQL, MSSQL was later chosen instead. PhpMyAdmin was used early on in the development process, but was later rejected due to issues with database migration and inflexibility with navigation and other such functions within the final application. The speed and flexibility of page and form design with the Pug.js templating engine and Node.js made it immediately apparent that this combination of technologies was by far the principal option. While discussing the project's agenda, the following requirements for the application were established:

- 1 Allow a person to join a team
- 2 Assign a team to a project
- 3 View (and edit, if necessary) flight information
- 4 View and edit project funds
- 5 Create a secure method of logging into the application

Once these requirements were selected and the application stack was decided, development could proceed.

## Development

This project began with significant amounts of research. With a remarkable amount of choices of libraries to choose from, research needed to be conducted to find the most scalable and expandable libraries and frameworks on which to build the application's basic logic and frame. Express.js was chosen for this due to the ease with which routing

and common web features, such as GET and POST requests, could be implemented. This package could also be easily implemented with Node.js' standard HTTP and HTTPS packages, which can be used to host a server on a specified port number. After this server was operational, a login system to access the application needed to be created and implemented on the server. This system was implemented as a SQL stored procedure, which allows the execution of procedural database operations. This procedure was written in SSMS, as described in the "Salts and Hashes" subsection of the "Application Security Measures" section. If a user is successfully authenticated, they are directed to the home page of the application.

After this basic login functionality was shown to work reliably, the home page was developed. This homepage was based on a form from the Access prototype. Many of the queries necessary for these forms were accessible from within Access' design view, but some modifications were needed due to slight syntax differences. The Pug.js templating engine has a functionality which allows the use of a *forEach loop*, which iterates over an array until the end of its data is reached, and thus facilitating the printing of SQL result sets by utilizing an array of objects. The items in this array can then be printed as data in HTML objects, which for the majority of this project were exclusively tables. By printing buttons at the end of each table row, the user can choose to edit a given record. A variety of buttons and links were also included on this page to allow logging out, signing up a user for a team, editing records, and other such actions. However, much of this functionality required substantial changes to the database's structure and stored procedures.



## Database Schema

Referential integrity and the enforcement of foreign keys are very important concepts in designing a database that retains accuracy and usability. When the Access prototype's schema was migrated to Microsoft SQL, many of these foreign key constraints and references were not maintained. This is due to the existence of a different method of reference creation in Access, in which primary and foreign keys can be different data types, such as tinyint and int. As a result, the table structures needed to be fixed throughout much of the database. Additionally, it is common practice when developing with Node.js to use stored procedures, rather than using query strings to manipulate data, as this is known to improve security and performance. As a result, stored procedures were written in order to implement the updating, insertion, and deletion of records, and to ensure such operations cascaded when necessary to avoid records pointing to nonexistent data. Microsoft's SQL Server Management Studio has several tools which allow easy visualization of table and column dependencies, as well as making the creation or deletion of such dependencies very straightforward (Microsoft, 2020). Figure 3 shows an image of the main area of the schema of the Mission2Go database within one such tool, the Database Diagrams utility.

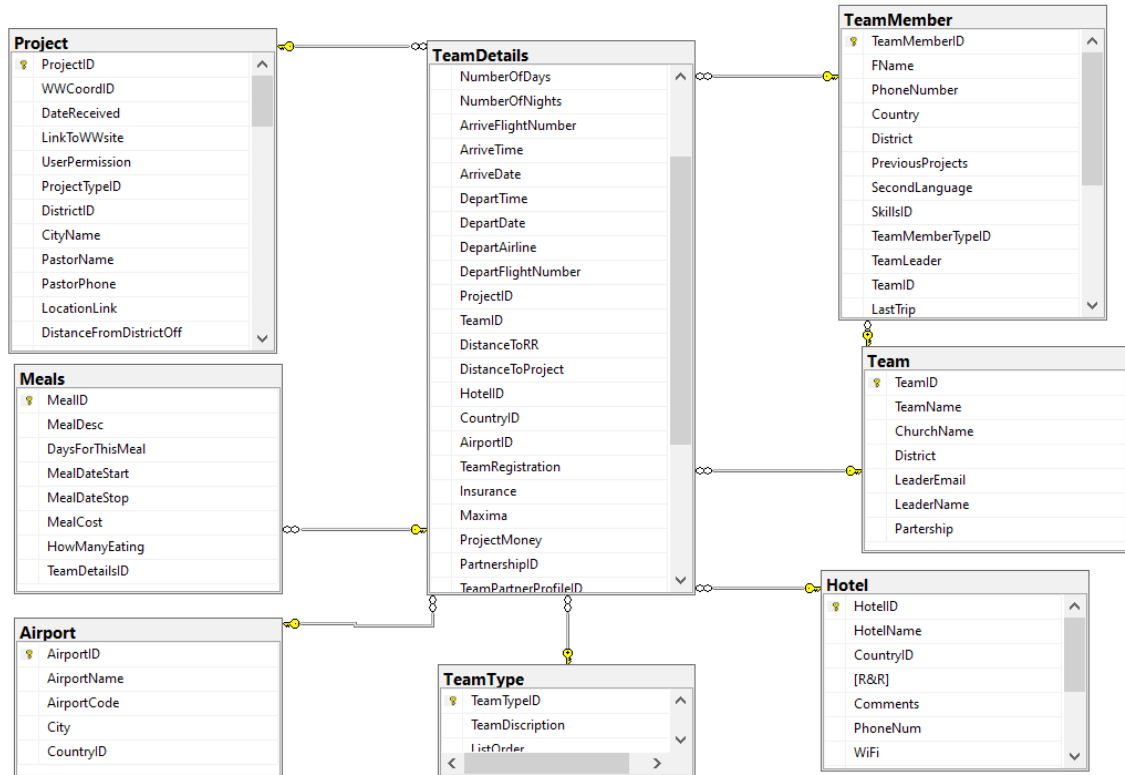


Figure 3. Database Schema

By utilizing such tools, the database could be structured to be fast, accurate, and ensure changes in one table would be reflected in tables which depended on these tables. Figure 3 details how the “TeamDetails” table is related to other tables. The table acts as a bridge table between the “Project” table and the “Team” table, and has relations to other tables that add additional information about a project/team assignment, such as the hotel the team is staying in or what airports will be used for arrival and departure.

While the tools SSMS provides were very helpful, there were still difficulties that arose while attempting to finalize the database’s structure. For example, the bit data type

is used in SQL as the equivalent of a Boolean variable in most other programming languages. However, updating the bit data type presented some problems, notably that deciding to cancel a team did not update the team's record properly. Additionally, in SSMS, values cannot update from the NULL state unless this state is explicitly handled. As a result, novel ways of circumventing these troublesome issues needed to be found. For the first issue, the bit type values were changed to integers, but still only set to the values 0 or 1. For the second, all null values were updated to have a value of 0, or for strings the value 'NONE PROVIDED'. These changes allowed updates, inserts, and deletions work as they needed to in order for the application to function properly. However, upon testing this structure and stored procedures in the database, the task remained of implementing them to cooperate with the client side of the application.

## User Interface (UI) Creation via Pug.js

Perhaps one of the most significant reasons Node.js is well suited for this particular project is its extensive support of various database languages. Node's Microsoft SQL (MSSQL) package allows the execution of stored procedures, as well as the printing of SQL error messages directly to the application's console. This capability also makes debugging and tracing the movement of data significantly easier than this would be otherwise. This is due to the fact that the package can print all SQL resultsets and recordsets in their raw form to the console, making monitoring data incredibly straightforward and intuitive. However, Node.js also provided some difficulties during the development of this application. Most notably, some documentation for certain packages was outdated, requiring me to email its authors or issue a GitHub error report or

pull request. Additionally, due to the existence of several extensions for connecting to SQL Server, some packages used objects known as connection pools, whereas others used a global connection variable for all SQL actions and requests, making referencing documentation an almost daily occurrence. However, over time, these issues became less prominent. With the available database packages, stored procedure results could easily be accessed in an array and have their values assigned to corresponding variables in an array of objects. Figure 4 demonstrates how this can be done.

```

app.get('/ScheduledTeamsView&Edit', function (req, res){

var UsNm = req.session.user;
console.log(req.sessionID);
if(UsNm == null && sessionID == null){res.redirect('/')};
console.log(`logged in as: ${UsNm}`);
const promise2 = new Promise(function (resolve, reject) {
var check2 = new sql.Request();
var tableArray = [];

check2.execute('ProjectTeamTables').then(function(result){
var len = result.recordset.length;
var today = new Date();
for(var i = 0; i < len; i++) {
tableArray.push({
"Maxima": result.recordset[i].Maxima,
"TeamName": result.recordset[i].TeamName,
"CountryName": result.recordset[i].CountryName,
"Days": Math.round((result.recordset[i].ArrivalDate - today)/(1000*60*60*24)),
"ArrivalDate": moment(result.recordset[i].ArrivalDate).format('YYYY-MM-DD'),
"DepartDate": moment(result.recordset[i].DepartDate).format('YYYY-MM-DD'),
"NumTeamMemb": result.recordset[i].NumTeamMemb,
"ProjectID": result.recordset[i].ProjectID,
"FName": result.recordset[i].FName,
"ProjectMoney": result.recordset[i].ProjectMoney,
"SumOfUSD": result.recordset[i].SumOfUSD,

});

};

//The above loop creates an array of objects...
//...which is easily rendered on the client side upon redirect and rendering
resolve(tableArray);
//console.log("redirect success");
res.render('ScheduledTeamsView&Edit', {tableArray : tableArray});
delete tableArray;

```

```
});  
});  
});
```

Figure 4. Example Array Creation

In Figure 4, results from the “ProjectTeamTables” procedure are loaded into the array “tableArray” and each item in the object is given a label. On the second to last line, the page showing this data is rendered, and this new object is passed with it, allowing the page to display the results of the query as the data changes over time. In a similar fashion, data that is inputted by the client into a form can be pulled with POST requests, and passed to the SQL object before running the procedure and returning results. The resulting client view is shown in Figure 5.

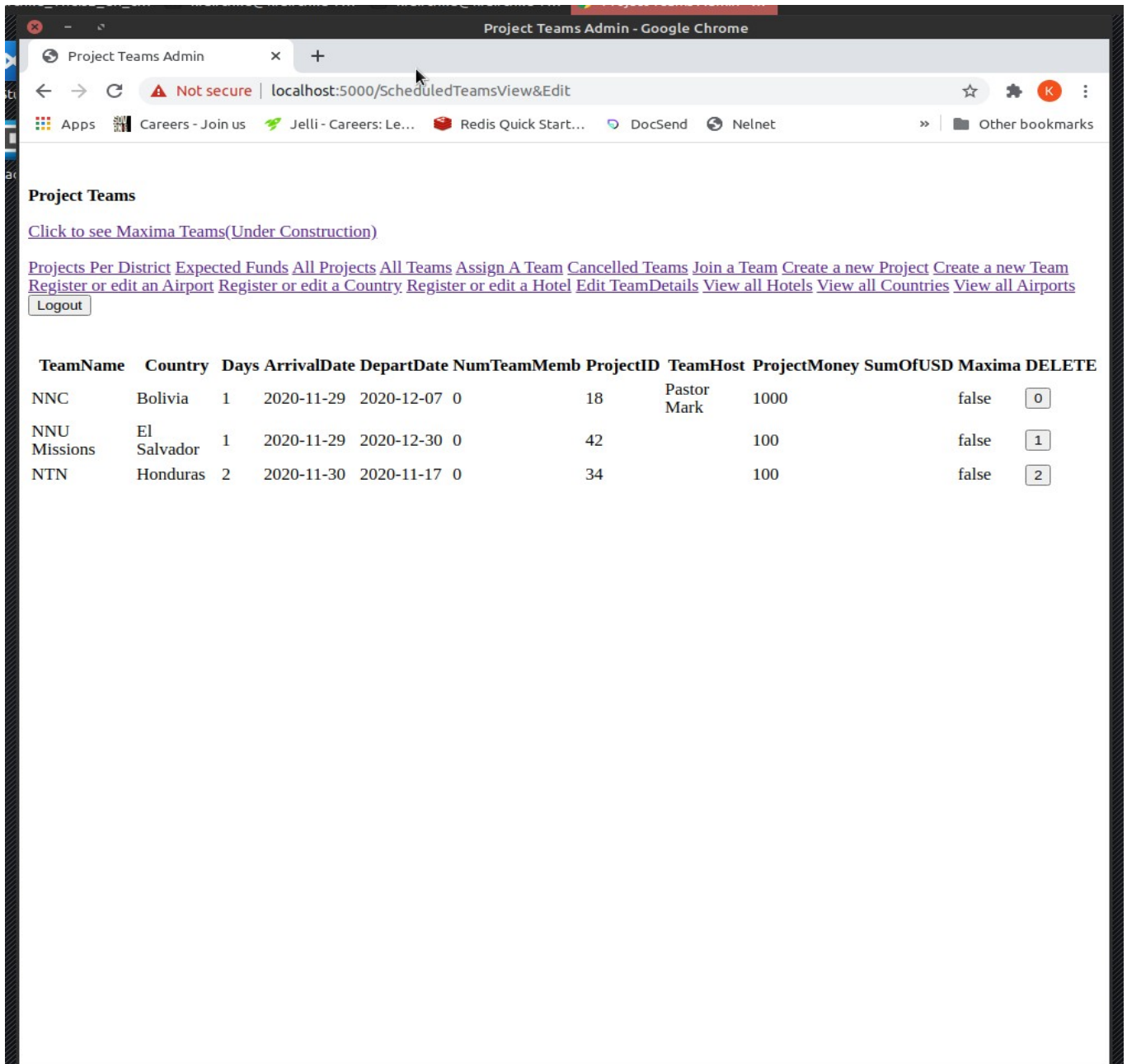


Figure 5. Application Home Screen

Once a procedure was established for communicating between the client and server-side components of the application, the creation of the basic necessary forms was a remarkably straightforward process, though with some exceptions. These forms will later be improved with cascading style sheets (CSS) written by Ms. Telah Dickerson. An example of a data entry form can be seen in Figure 6.

The image shows a web browser window titled "Project Teams Record Editor - Google Chrome". The address bar shows "localhost:5000/createT...". The page content includes a header "Edit Records [AUTHORIZED USERS ONLY]" and a sub-header "Currently Creating a New Team:". Below this are several input fields: "New Team Name:", "New Church Name:", "New District:", "New Leader Email:", and "New Leader Name:". The "New Partnership:" field is a dropdown menu with "YES" selected. A "Save Changes" button is located at the bottom of the form. The browser's address bar also shows "Not secure" and "Careers - Join us" in the bookmarks bar.

Figure 6. Example Data Entry Form



Some forms created difficulties due to the project stack's inability to pass more than one object (e.g. an array) to a given page prior to rendering. Because of this limitation, values needed to be held while all necessary variables were printed or gathered on separate forms before finally operating on the database with these variables as parameters. While this was initially difficult to plan, it became progressively easier as these situations continued to arise throughout the application's development.

The layout of the user interface for the Mission2Go application was initially very simple, containing only the necessary buttons and text fields for essential functioning, with little to no regard to user experience, intuitiveness, or aesthetics. These forms were created with the Pug.js templating engine, which has syntax practically identical to HTML, with the exception of some interpolation features and the complete absence of closing tags, although these tags can be included to suit the user's preference ("Tags"). Values are entered into forms, and forms are rendered with current data for users, via the use of GET and POST requests, respectively. An example of the use of these requests is shown in figures 7 and 8.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Registration</title>
    <meta charset="utf-8"/>
    style
    include ./CSS/registration.css
  </head>
  <body>
    <nav>
      <a href ="/">Log In</a>
    </nav>

    <div>

      <h1> Mission 2 Go Registration</h1>

      <form action="register" method="POST">
        <label for="newUser">New User Username: </label>
        <input id="newUser" type="text" name="newUser" required><br><br>

        <label for="newPass">New User Password: </label>
        <input id="newPass" type="text" name="newPass" required><br><br>

        <label for="newEmail">New User Email: </label>
        <input id="newEmail" type="text" name="newEmail" required><br><br>

        <input type="Submit" value="Register">
      </form>
    </div>
  </body>
</html>

```

Figure 7. Client Side View

Figure 7 shows the Pug.js code for a simple form that takes inputs for a username, password, and email address. This code creates an HTML form with an action called “register” and implements the POST method. Figure 8 shows the handling of this form once submitted.

```

app.get('/registration', function(req,res){

res.render('registration');

});

app.post('/register', function(req, res){
var newUser = req.body.newUser;
var newPass = req.body.newPass;
var newEmail = req.body.newEmail;
const promiseReg = new Promise(function (resolve, reject) { //promise awaits results
before proceeding
var R = new sql.Request();
R.input('newUser', sql.NVarChar(64), newUser); //pass user and pass to stored
procedure
R.input('newPass', sql.NVarChar(64), newPass);
R.input('newEmail', sql.NVarChar(64), newEmail);
R.execute('registerUser').then(function (result, recordset){
var flag = result.returnValue;
resolve(flag);
}).catch(function(error){
console.log(error);
sql.close();
reject(error);
});
});
promiseReg.then((verdict) => {
if (verdict === 0) {
res.send("THIS USER ALREADY EXISTS");
res.redirect('/login');
}
else{
//console.log("...ACCOUNT CREATION SUCCESSFUL...");
res.redirect('/');
}
});
});
});

```

Figure 8. View Handler- Back End

Figure 8 shows the server-side handling of the form in figure 7. The “app.get” block at the top uses a GET request to render the view to the client. Once the “Submit” button is clicked to submit the form, the “app.post” block is called. This block pulls data from the form using calls to the request (req) object and specific variables from the client side code in figure 7, such as the “newUser” variable. After this is complete, these values are passed to a SQL stored procedure to store them in the appropriate table, and a value is returned from this procedure specifying if registration was successful or if an account already exists. In this manner, GET and POST requests are used to move data from the client to the server where it is to be processed and stored.

## Conclusion

While this project certainly was difficult and had a remarkably steep learning curve, it was incredibly rewarding and challenging. Perhaps the best way to learn software development skills is to work on projects in skills that are foreign in order to learn and become proficient, if not a master in these skills. With the rapid advancement of computing technologies, this need to take on new challenges is a mainstay in the field of computer science. This project exposed its creators to database design and server-side programming, and while I have performed both of these tasks for courses throughout my time at NNU, this project certainly increased my appreciation and enjoyment of these aspects of system design.

Although the Mission2Go application does still have some minor bugs, the basic functionality works sufficiently to be used in the field. Bugs will be fixed as quickly as

possible and as they appear during use to ensure the most stable and effective application possible. Teams can be assigned to projects with ease, and new records can be created. Many of the bugs in the application are due to difficulties with moving parameters between functions. This is because the web framework used is designed to only take two parameters, making it impossible to pass additional information effectively between functions. However, this application works relatively well and accomplishes its objectives overall.

This project serves as a lesson to its creators that just because the rules of a framework cannot seem to be broken, does not mean they cannot be bent to work toward a desired end. The project forced the team to think about problems and necessary algorithms in new ways in order to work with the strengths and weaknesses of a given technology, or to know when to switch methods or languages if necessary, to find a more advantageous methodology of approaching a solution. While this project still needs more development work, user feedback, and testing, this project could have a substantial impact on the administration of missions projects, which could certainly increase opportunities to serve for people everywhere, including the people of Northwest Nazarene University.

## Future Work

While the work completed on the Mission2Go project is certainly a significant step towards streamlining the administration of mission projects and outreach all over the world, there is significant potential for this project to be improved with a more diverse

selection of operations, and perhaps the addition of new tables to the database to reflect other data that are of use and interest to a missions coordinator. Additionally, this application will certainly be able to improve in terms of its speed and efficiency, particularly for clients, as new and faster Node.js packages are developed to meet new needs, or old needs in new ways. Because the process of systems analysis and design is an iterative process, it is the team's hope that work on this application can continue into the future, perhaps as another NNU student's senior project.

## References

Express-session. (n.d.). Retrieved October 28, 2020, from

<http://expressjs.com/en/resources/middleware/session.html>

Getting Started. [pugjs.org/api/getting-started.html](http://pugjs.org/api/getting-started.html)

Hashbytes (Transact-SQL). (2016, July 29).

<https://docs.microsoft.com/en-us/sql/t-sql/functions/hashbytes-transact-sql?redirectedfrom=MSDN&view=sql-server-ver15>

Linux Mark Institute. (2020). Linux Mint 20 Ulyana (20) [Operating System]. Retrieved

From <https://linuxmint.com/download.php>

Microsoft. (2020). SQL Server Management Studio (18.7.1) [Database Management

Software]. Retrieved From <https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15>

Morris, Robert and Ken Thompson (1978). *Password Security: a Case Study*. Bell

Laboratories. <https://web.archive.org/web/20130821093338/http://cm.bell-labs.com/cm/cs/who/dmr/passwd.ps>

Mssql -npm. (n.d.). Retrieved November 30, 2020, from

<https://www.npmjs.com/package/mssql>

Node.js. (n.d.). *Node.js*. OpenJS Foundation, [nodejs.org/en/](https://nodejs.org/en/)

Node.js Web Application Framework. (n.d.) *Express*. [expressjs.com/](https://expressjs.com/)

Redis. (n.d.). Retrieved November 10, 2020, from <https://redis.io/>

Redis Labs. (2020). Redis (6.0.9) [Memory-Based Session Store]

Retrieved From <https://redis.io/download>

Session Management. (n.d.). Retrieved October 28, 2020, from

<https://redislabs.com/solutions/use-cases/session-management/>

Tags. (n.d.). Retrieved October 28, 2020, from <https://pugjs.org/language/tags.html>

Uniqueidentifier (Transact-SQL). (2017, December 1).

[https://docs.microsoft.com/en-us/sql/t-sql/data-types/uniqueidentifier-transact-sql?  
redirectedfrom=MSDN&view=sql-server-ver15](https://docs.microsoft.com/en-us/sql/t-sql/data-types/uniqueidentifier-transact-sql?redirectedfrom=MSDN&view=sql-server-ver15)

Who's using Redis? (n.d.). Retrieved November 28, 2020, from

<https://redis.io/topics/whos-using-redis>



# Appendices

## A. UI Source

```
<!DOCTYPE html>

<html lang="en">

<head>
  <title>Project Records</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Viewing all Projects [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing all project records: </b>
<br>
<form id="getProjectID" action="getProjectID" method="POST"
name="getProjectID">
  <table>
    <tr>
      <th>ProjectID
      <th>TeamID
      <th>LinkToWwsite
      <th>ProjectTypeID
      <th>DistrictID
      <th>CityName
      <th>ChurchName
      <th>PastorName
      <th>PastorPhone
      <th>ProjectDesc
      <th>ProjectPlans
      <th>ProjectCompleted
      <th>CountryID
      <th>EDIT
      <th>DELETE
      <tbody>
        <tr>
          <td colspan="8">
            -ProjectsViewArray.forEach(function(element, index){
          <tr>
            <td> #{element.ProjectID}
            <td> #{element.TeamID}
            <td> #{element.LinkToWwsite}
            <td> #{element.ProjectTypeID}
            <td> #{element.DistrictID}
```

```
 #{element.CityName} | #{element.ChurchName} | #{element.PastorName} | #{element.PastorPhone} | #{element.ProjectDesc} | #{element.ProjectPlans} | #{element.ProjectCompleted} | #{element.CountryID} | <input id= #{index} type="Submit" name="RowForEditProject" value= #{index}> | <input id= #{index} type="Submit" name="RowForDeleteProject" value= #{index}> | -}); </table> <br> <br> </form> <br> <br> <br> <br> <a href = '/ScheduledTeamsView&Edit'> HOME </a> <br> <br> <br> </body> </html> |
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Viewing all TeamDetails [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing all TeamDetails records: </b>
<br>
<form id="getTeamDetailsID" action="getTeamDetailsID" method="POST"
name="getTeamDetailsID">
  <table>
    <tr>
      <th> TeamDetailsID
      <th> NumTeamMemb
      <th> ArriveFlightNumber
      <th> ArriveDate
      <th> DepartDate
      <th> DepartAirline
      <th> DepartFlightNumber
      <th> ProjectID
      <th> ProjectName
      <th> TeamID
      <th> TeamName
      <th> CountryID
      <th> CountryName
      <th> Maxima
      <th> EDIT
      <th> DELETE
      <tbody>
        <tr>
          <td colspan="13"> -AllTeamDetails.forEach(function(element, index){
          <tr>
            <td> #{element.TeamDetailsID}
            <td> #{element.NumTeamMemb}
            <td> #{element.ArriveFlightNumber}
            <td> #{element.ArriveDate}
            <td> #{element.DepartDate}
            <td> #{element.DepartAirline}
            <td> #{element.DepartFlightNumber}
            <td> #{element.ProjectID}
            <td> #{element.ProjectName}
            <td> #{element.TeamID}
            <td> #{element.TeamName}
            <td> #{element.CountryID}

```

```
        td #{element.CountryName}
        td #{element.Maxima}
        td <input id= #{index} type="Submit"
name="RowForEditTeamDetails" value= #{index}>
        td <input id= #{index} type="Submit"
name="RowForDeleteTeamDetails" value= #{index}>
        td
        -});
    </table>
    <br>
    <br>
</form>
<br>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> View Records (TEAM)[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing all teams records: </b>
<br>
<form id="allTeams" action="allTeams" method="POST" name="allTeams">
  <table>
    <tr>
      <th TeamID>
      <th TeamName>
      <th ChurchName>
      <th District>
      <th LeaderEmail>
      <th LeaderName>
      <th EDIT>
      <th DELETE>
      <th View Members>
        -teamViewArray.forEach(function(element, index){
          <tr>
            <td #{element.TeamID}>
            <td #{element.TeamName}>
            <td #{element.ChurchName}>
            <td #{element.District}>
            <td #{element.LeaderEmail}>
            <td #{element.LeaderName}>
            <td <input id= #{index} type="Submit" name="RowForEditTeam"
value= #{index}>
            <td <input id= #{index} type="Submit"
name="RowForDeleteTeam" value= #{index}>
            <td <input id= #{index} type="Submit"
name="RowForViewMembers" value = #{element.TeamID}>
            -});
          </td>
        </tr>
      </th>
    </tr>
  </table>
<br>
<br>
</form>
<br>
<br>
<br>

```

```
<a href = '/ScheduledTeamsView&Edit'> HOME </a>  
<br>  
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Admin</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Project Assign: Choose a Project. If this project is currently
assigned to another team, the existing team will be cancelled. </b>
<br>
<br>
<form action = "GetProjectIndex" method = "POST">
  <table>
    <tr>
      <th>ProjectID
      <th>ChurchName
      <th>PastorName
      <th>Country
      <th>DateRecieved
      <th>ASSIGN
        -tableArrayAssignProject.forEach(function(element){
          <tr>
            <td> #{element.ProjectID}
            <td> #{element.ChurchName}
            <td> #{element.PastorName}
            <td> #{element.Country}
            <td> #{element.DateRecieved}
            <td> <input id= #{element.ProjectID} type="Submit"
name="RowForProjectToAssign" value= #{element.ProjectID}>
            -});
          </tr>
        </table>
    </form>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Assign Team(Select TeamMemberID) [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Assigning the following Team: </b>
<b> [A TEAM WITH A TEAM ID OF 0 IS UNASSIGNED]
<br>
  <table>
    <tr>
      <th> TeamID
      <th> TeamDetailsID
      <th> TeamName
      <th> ChurchName
      <th> District
      <th> LeaderEmail
      <th> LeaderName
      <th> Partership
      -teamViewAssignOpts.forEach(function(element){
        <tr>
          <td> #{element.TeamID}
          <td> #{element.TeamDetailsID}
          <td> #{element.TeamName}
          <td> #{element.ChurchName}
          <td> #{element.District}
          <td> #{element.LeaderEmail}
          <td> #{element.LeaderName}
          <td> #{element.Partership}
        -});
    </tr>
  </table>
<br>
<br>
  <form id="assign-Team" action="assign-Team" method="POST"
name="assign-Team">
    <input type = "hidden" id="newTeamID" name="newTeamID" required
readonly>
      -teamViewAssignOpts.forEach(function(element){
        document.getElementById('TeamID').value=element.TeamID
      -});
    <select id = 'SpecifyTeamDetail' name = 'SpecifyTeamDetail'>
      -teamViewAssignOpts.forEach(function(element){

```



```
document.getElementById('TeamDetailsID').value=element.TeamDetailsID
        document.write("<option value =
#{TeamDetailsID}>#{teamDetailsID}</option>")
        -});
        </select>
        <br>
        <br>
        <input type="Submit" value="Proceed and Choose Project">
    </form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Choose Team [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Choose a team to assign : </b>
<br>
<form id="GetTeamIndex" action="GetTeamIndex" method="POST"
name="GetTeamIndex">
  <table>
    <tr>
      <th>TeamID
      <th>TeamName
      <th>ChurchName
      <th>District
      <th>LeaderEmail
      <th>LeaderName
      <th>Partership
      <th>ASSIGN
      -teamViewAssign.forEach(function(element, index){
    <tr>
      <td>#{element.TeamID}
      <td>#{element.TeamName}
      <td>#{element.ChurchName}
      <td>#{element.District}
      <td>#{element.LeaderEmail}
      <td>#{element.LeaderName}
      <td>#{element.Partership}
      <td><input type="Submit" id= #{index}
name="RowForTeamToAssign" value= #{element.TeamID}>
      -});
    </tr>
  </table>
<br>
<br>
<br>
<br>
</form>
<br>
<br>
<br>
<br>

```

```
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing all cancelled teams' records: </b>
<br>
<form id="cancelledTeams" action="cancelledTeams" method="POST"
name="cancelledTeams">
  <table>
    <tr>
      <th TeamID>
      <th TeamName>
      <th ChurchName>
      <th District>
      <th LeaderEmail>
      <th LeaderName>
      <th Partnership>
      -tableArrayCancelledTeams.forEach(function(element, index){
    <tr>
      <td #{element.TeamID}>
      <td #{element.TeamName}>
      <td #{element.ChurchName}>
      <td #{element.District}>
      <td #{element.LeaderEmail}>
      <td #{element.LeaderName}>
      <td #{element.Partnership}>
      <td <input id= #{index} type="Submit"
name="RowForEditCancelledTeam" value= #{index}>
      <td <input id= #{index} type="Submit"
name="RowForDeleteCancelledTeam" value= #{index}>
      <td
      -});
    </td>
  </tr>
  </table>
  <br>
  <br>
</form>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>

```

```
<br>  
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Airport Creator/Editor[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Choosing TD airport: </b>
<br>
<br>
<a href="/newAirport" >Or Make A New Airport here</a>
<br>
<br>
  <table>
    <tr>
      <th AirportID>
      <th AirportName>
      <th AirportCode>
      <th City>
      <th Country>
      -tableArrayChooseAirport.forEach(function(element){
    <tr>
      <td #{element.AirportID}>
      <td #{element.AirportName}>
      <td #{element.AirportCode}>
      <td #{element.City}>
      <td #{element.Country}>
      -});
    </tr>
  </table>
<br>
<br>
  <form id="edit-td-Airport" action="edit-td-Airport" method="POST"
name="edit-td-Airport">
    <label for ="tda">Select An Airport: </label>
    <select name = 'tda' id = 'tda'>
      -tableArrayChooseAirport.forEach(function(element,index){
        <option value =
#{element.AirportID}>#{element.AirportName}</option>
      -});
    </select>
    <input type="Submit" value="Choose Hotel">
  </form>
<br>

```

```
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Choose TeamDetails Country[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently choosing the TD Country: </b>
<br>
<br>
<a href="/newCountry" >Or Make A New Country Record here</a>
<br>
<br>
  <table>
    <tr>
      <th>CountryID
      <th>CountryName
      <th>FieldID
      -tableArrayChooseCountry.forEach(function(element){
        <tr>
          <td> #{element.CountryID}
          <td> #{element.CountryName}
          <td> #{element.FieldID}
        -});
    </table>
<br>
<br>
    <form id="edit-td-Country" action="edit-td-Country" method="POST"
name="edit-td-Country">
      <label for ="ctd">Select A Country: </label>
      <select name = 'ctd' id = 'ctd'>
        -tableArrayChooseCountry.forEach(function(element,index){
          <option value =
#{element.CountryID}>#{element.CountryName}</option>
        -});
      </select>
      <input type="Submit" value="Choose Airport">
    </form>
<br>
<br>
<br>
<br>
<br>

```



```
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Airport Creator/Editor[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Choosing TD Airport: </b>
<br>
<br>
<a href="/newHotel" >Or Make A New Hotel here</a>
<br>
<br>
  <table>
    <tr>
      <th>HotelID
      <th>HotelName
      <th>City
      <th>PhoneNum
      <tbody>
        <tr>
          <td>#{element.HotelID}
          <td>#{element.HotelName}
          <td>#{element.City}
          <td>#{element.PhoneNum}
        </tr>
      </tbody>
    </table>
<br>
<br>
    <form id="edit-td-Hotel" action="edit-td-Hotel" method="POST"
name="edit-td-Hotel">
      <label for ="tdh">Select A Hotel: </label>
      <select name = 'tdh' id = 'tdh'>
        <tbody>
          <tr>
            <td>#{element.HotelID}>#{element.HotelName}</td>
          </tr>
        </tbody>
      </select>
      <input type="Submit" value="Meal Info">
    </form>
<br>
<br>
<br>

```

```
<br>  
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Meals [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently creating a Meal: </b>
<br>
  <table>
    <tr>
      <th>TeamDetailsID
      <th>ArriveDate
      <th>DepartDate
      <th>ProjectID
      <th>TeamID
      <th>CountryID
      <th>ProjectMoney
      <th>CancelTeam
      <th>ProjectFundsReceived
      -AllTeamDetailsForMeal.forEach(function(element){
        <tr>
          <td>#{element.TeamDetailsID}
          <td>#{element.ArriveDate}
          <td>#{element.DepartDate}
          <td>#{element.ProjectID}
          <td>#{element.TeamID}
          <td>#{element.CountryID}
          <td>#{element.ProjectMoney}
          <td>#{element.CancelTeam}
          <td>#{element.ProjectFundsReceived}
        -});
    </table>
    <form id="create-Meals" action="create-Meals" method="POST"
name="create-Meals">
      <br>
      <label for="newMealDesc">Meal Description:
</label>
      <br>
      <input id="newMealDesc" type = "text" name = "newMealDesc">
      <br>
      <label for="newMealDateStart">MealDateStart:
</label>

```

```

        <br>
        <input id="newMealDateStart" type = "date" name =
"newMealDateStart">
        <br>
        <label for="newMealDateStop">Meal Date Stop:
</label>
        <br>
        <input id="newMealDateStop" type = "date" name =
"newMealDateStop">
        <br>
        <label for="newMealCost">Meal Cost:                </label>
        <br>
        <input id="newMealCost" type = "number" name = "newMealCost">
        <br>
        <label for="newHowManyEating">How Many Eating:
</label>
        <br>
        <input id="newHowManyEating" type = "text" name =
"newHowManyEating">
        <br>
        <label for ="tdm">Select TeamDetails for Assign: </label>
        <select name = 'tdm' id = 'tdm'>
            -AllTeamDetailsForMeal.forEach(function(element,index){
                <option value =
#{element.TeamDetailsID}>#{element.TeamDetailsID}</option>
                -});
        </select>
        <br>
        <input type="Submit" value="Save Changes">
    </form>
<br>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Creating a New Project: </b>
<br>
  <br>
  <br>
  <form id="create-Project" action="create-Project" method="POST"
name="create-Project">
    <br>
    <br>
    <br>
    <label for ="tdh">Select A WWCoordinator: </label>
    <select name = 'newWWCoorID' id = 'newWWCoorID'>
      -WWcoordArray.forEach(function(element,index){
        <option value =
#{element.WWcoordID}>#{element.FullName}</option>
      -});
    </select>
    <br>
    <label for="newDateReceived">New DateRecieved:
</label>
    <br>
    <input id="newDateReceived" type = "date" name =
"newDateReceived" required>
    <br>
    <label for="newLinkToWwsite">New Website Link:
</label>
    <br>
    <input id="newLinkToWwsite" type = "text" name =
"newLinkToWwsite" required>
    <br>
    <label for="newUserPermission">New User Permission:
</label>
    <br>
    <input id="newUserPermission" type = "text" name =
"newUserPermission" required>
    <br>

```

```

    <label for="newProjectTypeID">New Project Type:
</label>
    <br>
    <select id="newProjectTypeID" name = "newProjectTypeID">
      <option value = 0>None Provided</option>
      <option value = 1>Templo</option>
      <option value = 2>Casa Pastoral</option>
      <option value = 3>Aulas De ED</option>
      <option value = 4>Casa Districal</option>
      <option value = 5>Centro Districal</option>
      <option value = 6>Techo</option>
      <option value = 7>NazaGol</option>
      <option value = 10>Extreme Team</option>
      <option value = 11>Medical Team</option>
      <option value = 12>Leadership Training</option>
      <option value = 13>Seminary Maintenance</option>
      <option value = 14>Water Filters</option>
    </select>
    <br>
    <label for="newDistrictID">New District ID:
</label>
    <br>
    <select id="newDistrictID" name = "newDistrictID">
      <option value = 0>None Provided</option>
      <option value = 1>Central (Guatemala)</option>
      <option value = 38>Franja Occidental Guatemala
(Guatemala)</option>
      <option value = 39>Nor Occidente (Honduras)</option>
      <option value = 40>Centro Sur-Oriente (Honduras)</option>
      <option value = 41>Sureste De Nicaragua</option>
      <option value = 42>Franja Central (Guatemala)</option>
      <option value = 43>Sur Occidental de Peten
(Guatemala)</option>
      <option value = 44>Central (El Salvador)</option>
      <option value = 45>Norte De Nicaragua</option>
      <option value = 46>Noroccidente (Guatemala)</option>
      <option value = 47>Verapaz del Norte (Guatemala)</option>
      <option value = 48>Norte (Guatemala)</option>
      <option value = 49>Verapaz Oreiental (Guatemala)</option>
      <option value = 50>Suroriente (Guatemala)</option>
      <option value = 51>Atlantico (Guatemala)</option>
      <option value = 52>Franja Oriental (Guatemala)</option>
      <option value = 53>Norte de Coban (Guatemala)</option>
      <option value = 54>Occidente (Guatemala)</option>
      <option value = 56>Occidente (El Salvador)</option>
      <option value = 57>Baja Verapaz (Guatemala)</option>
      <option value = 58>Sur (Nicaragua)</option>
      <option value = 59>Oriente (El Salvadoe)</option>
      <option value = 60>Verpaz del Sur (Guatemala)</option>
      <option value = 61>Central (Nicaragua)</option>

```

```

        </select>
        <br>
        <label for="newCityName">New City:                </label>
        <br>
        <input id="newCityName" type = "text" name = "newCityName"
required>
        <br>
        <label for="newChurchName">New Church Name:
</label>
        <br>
        <input id="newChurchName" type = "text" name = "newChurchName"
required>
        <br>
        <label for="newPastorName">New Pastor Name:
</label>
        <br>
        <input id="newPastorName" type = "text" name = "newPastorName"
required>
        <br>
        <label for="newPastorPhone">New Pastor Phone:
</label>
        <br>
        <input id="newPastorPhone" type = "text" name = "newPastorPhone"
required>
        <br>
        <label for="newCountry">New Country:                </label>
        <br>
        <input id="newCountry" type = "text" name = "newCountry"
required>
        <br>
        <label for="newLocationLink">New Location Link:                </
label>
        <br>
        <input id="newLocationLink" type = "text" name =
"newLocationLink" required>
        <br>
        <label for="newDistanceFromDistrictOff">New Distance From
District Office:                </label>
        <br>
        <input id="newDistanceFromDistrictOff" type = "number" name =
"newDistanceFromDistrictOff" required>
        <br>
        <label for="newDistrictBudgetCurrent">New District Budget
Current:                </label>
        <br>
        <input id="newDistrictBudgetCurrent" type = "number" name =
"newDistrictBudgetCurrent" required>
        <br>
        <label for="newDistrictBudgetLast                ">New District Budget
Last:                </label>

```



```

        <br>
        <input id="newDistrictBudgetLast" type = "number" name =
"newDistrictBudgetLast" required>
        <br>
        <label for="newFEMCurrent">New FEM Current:
</label>
        <br>
        <input id="NewFEMCurrent" type = "number" name = "NewFEMCurrent"
required>
        <br>
        <label for="newFEMLast">New FEM Last: </label>
        <br>
        <input id="newFEMLast" type = "number" name = "newFEMLast"
required>
        <br>
        <label for="newDateOrganized">New Date Organized:
</label>
        <br>
        <input id="newDateOrganized" type = "date" name =
"newDateOrganized" required>
        <br>
        <label for="newMembersLast">New Members Last:
</label>
        <br>
        <input id="newMembersLast" type = "number" name =
"newMembersLast" required>
        <br>
        <label for="newMembersCurrent">New Members Current:
</label>
        <br>
        <input id="newMembersCurrent" type = "number" name =
"newMembersCurrent" required>
        <br>
        <label for="newAvgAttendanceLast" ">New Avg Attendance
Last: </label>
        <br>
        <input id="newAvgAttendanceLast " type = "number" name =
"newAvgAttendanceLast" required>
        <br>
        <label for="newAvgAttendanceCurrent" ">New AvgAttendance
Current </label>
        <br>
        <input id="newAvgAttendanceCurrent" type = "number" name =
"newAvgAttendanceCurrent" required>
        <br>
        <label for="newProjectDesc">New Project Desc:
</label>
        <br>
        <input id="newProjectDesc" type = "text" name = "newProjectDesc"
required>

```

```

        <br>
        <label for="newTempleCapacity">New Temple Capacity:
</label>
        <br>
        <input id="newTempleCapacity" type = "number" name =
"newTempleCapacity" required>
        <br>
        <label for="newRoofWidth">New Roof Width:
</label>
        <br>
        <input id="newRoofWidth" type = "number" name = "newRoofWidth"
required>
        <br>
        <label for="newRoofLength">New Roof Length:
</label>
        <br>
        <input id="newRoofLength" type = "number" name = "newRoofLength"
required>
        <br>
        <label for="newBriefDescriptionConst">New Brief Desc Const:
</label>
        <br>
        <input id="newBriefDescriptionConst" type = "text" name =
"newBriefDescriptionConst" required>
        <br>
        <label for="newPropertyWidth">New Property Width:
</label>
        <br>
        <input id="newPropertyWidth" type = "number" name =
"newPropertyWidth" required>
        <br>
        <label for="newPropertyLength">New Property Length:
</label>
        <br>
        <input id="newPropertyLength" type = "number" name =
"newPropertyLength" required>
        <br>
        <label for="newPhotos">New Photos:                                </label>
        <br>
        <select id = "newPhotos" name = "newPhotos">
          <option value = 1>YES</option>
          <option value = 0>NO</option>
        </select>
        <br>
        <label for="newDigital">New Digital:                                </label>
        <br>
        <select id = "newDigital" name = "newDigital">
          <option value = 1>YES</option>
          <option value = 0>NO</option>
        </select>

```

```

        <br>
        <label for="newLotLayout">New Lot Layout:
</label>
        <br>
        <select id = "newLotLayout" name = "newLotLayout">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newProjectPlans">New Project Plans:
</label>
        <br>
        <select id = "newProjectPlans" name = "newProjectPlans">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newMoneyRaised">New Money Raised:
</label>
        <br>
        <input id="newMoneyRaised" type = "number" name =
"newMoneyRaised" required>
        <br>
        <label for="newMoneyInvested">New Money Invested:
</label>
        <br>
        <input id="newMoneyInvested" type = "number" name =
"newMoneyInvested" required>
        <br>
        <label for="newMoney12Months">New Money 12 Months:
</label>
        <br>
        <input id="newMoney12Months" type = "number" name =
"newMoney12Months" required>
        <br>
        <label for="newProjectNeed">New Project Need:
</label>
        <br>
        <input id="newProjectNeed" type = "text" name = "newProjectNeed"
required>
        <br>
        <label for="newTitleToProperty">New Title to Property:
</label>
        <br>
        <select id = "newTitleToProperty" name = "newTitleToProperty">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>

```

```

Church:      <label for="newPropertyDeededToChurch">New Property Deeded to
              </label>
              <br>
              <select id = "newPropertyDeededToChurch" name =
"newPropertyDeededToChurch">
                <option value = 1>YES</option>
                <option value = 0>NO</option>
              </select>
              <br>
              <label for="newBuildingPermitNeeded">New Building Permit Needed:
</label>
              <br>
              <select id = "newBuildingPermitNeeded" name =
"newBuildingPermitNeeded">
                <option value = 1>YES</option>
                <option value = 0>NO</option>
              </select>
              <br>
              <label for="newBuildingPermitApproved">New Building Permit
Approved:      </label>
              <br>
              <select id = "newBuildingPermitApproved" name =
"newBuildingPermitApproved">
                <option value = 1>YES</option>
                <option value = 0>NO</option>
              </select>
              <br>
              <label for="newPastorApproved">New Pastor Approved:
</label>
              <br>
              <select id = "newPastorApproved" name = "newPastorApproved">
                <option value = 1>YES</option>
                <option value = 0>NO</option>
              </select>
              <br>
              <label for="newChurchBoardApproved">New Church Board Approved:
</label>
              <br>
              <select id = "newChurchBoardApproved" name =
"newChurchBoardApproved">
                <option value = 1>YES</option>
                <option value = 0>NO</option>
              </select>
              <br>
              <label for="newTreasureApproved">New Treasure Approved:
</label>
              <br>
              <select id = "newTreasureApproved" name = "newTreasureApproved">
                <option value = 1>YES</option>
                <option value = 0>NO</option>

```

```

        </select>
        <br>
        <label for="newTreasureApprovedDate">New Treasure Approved Date:
</label>
        <br>
        <input id="newTreasureApprovedDate" type = "date" name =
"newTreasureApprovedDate" required>
        <br>
        <label for="newSuperApproval">New Super Approval:
</label>
        <br>
        <select id = "newSuperApproval" name = "newSuperApproval">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
        </select>
        <br>
        <label for="newSuperApprovalDate">New Super Approval Date:
</label>
        <br>
        <input id="newSuperApprovalDate" type = "date" name =
"newSuperApprovalDate" required>
        <br>
        <label for="newDistrictSecretaryApproval">New District Secretary
Approval: </label>
        <br>
        <select id = "newDistrictSecretaryApproval" name =
"newDistrictSecretaryApproval" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
        </select>
        <br>
        <label for="newDistrictTreasureApproval">New District Treasure
Approval: </label>
        <br>
        <select id = "newDistrictTreasureApproval" name =
"newDistrictTreasureApproval" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
        </select>
        <br>
        <label for="newDistrictTreasureApprovalDate">New District
Treasure Approval Date: </label>
        <br>
        <input id="newDistrictTreasureApprovalDate" type = "date" name =
"newDistrictTreasureApprovalDate" required>
        <label for="newDistrictPriority">New District Priority:
</label>
        <br>
        <select id = "newDistrictPriority" name = "newDistrictPriority">
        <option value = 1>YES</option>

```

```

        <option value = 0>NO</option>
    </select>
    <br>
    <label for="newTyTapproved">New TyTapproved:
</label>
    <br>
    <select id = "newTyTapproved" name = "newTyTapproved">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="newTyTPriority">New TyTPriority:
</label>
    <br>
    <select id = "newTyTPriority" name = "newTyTPriority">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="newProjectCompleted">New Project Completed:
</label>
    <br>
    <select id = "newProjectCompleted" name = "newProjectCompleted">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="newWWLink">New WW Link:                </label>
    <br>
    <input id="newWWLink" type = "text" name = "newWWLink">
    <br>
    <label for="newPhotoLocation">New Photo Location:
</label>
    <br>
    <input id="newPhotoLocation" type = "text" name =
"newPhotoLocation">
    <br>
    <label for="newPermitStart">New Permit Start:
</label>
    <br>
    <input id="newPermitStart" type = "date" name = "newPermitStart"
required>
    <br>
    <label for="newPermitEnd">New Permit End:
</label>
    <br>
    <input id="newPermitEnd" type = "date" name = "newPermitEnd"
required>
    <br>
    <label for="newCountryID">New CountryID:                </label>

```

```
<br>
<select id="newCountryID" name = "newCountryID">
  <option value = 1>Guatemala</option>
  <option value = 2>El Salvador</option>
  <option value = 3>Honduras</option>
  <option value = 4>Nicaragua</option>
</select>
<input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Creating a New Team: </b>
<br>
  <form id="create-Team" action="create-Team" method="POST"
name="create-Team">
    <br>
    <label for="newTeamName">New Team Name: </label>
    <br>
    <input id="newTeamName" type = "text" name = "newTeamName"
required>
    <br>
    <label for="newChurchName">New Church Name:
</label>
    <br>
    <input id="newChurchName" type = "text" name = "newChurchName"
required>
    <br>
    <label for="newDistrict">New District: </label>
    <br>
    <input id="newDistrict" type = "text" name = "newDistrict"
required>
    <br>
    <label for="newLeaderEmail">New Leader Email:
</label>
    <br>
    <input id="newLeaderEmail" type = "text" name = "newLeaderEmail"
required>
    <br>
    <label for="newLeaderName">New Leader Name:
</label>
    <br>
    <input id="newLeaderName" type = "text" name = "newLeaderName"
required>
    <br>
    <label for="newPartership">New Partership:
</label>
    <select id = "newPartership" name = "newPartership" required>

```



```
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <br>
    <br>
    <br>
    <br>
    <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```
<!DOCTYPE html>

<html lang="en">
<head>
  <title>Log In</title>
  <meta charset="utf-8"/>
</head>
<body>
  <form action="login" method="POST">
    <label for="user">Username: </label>
    <input id="user" type="text" name="user" required/><br><br>
    <label for="pass">Password: </label>
    <input id="pass" type="password" name="pass" required/><br><br>
    <input type="Submit" value="Sign In">
  </form>
  <a href="/registration" >Register Here!!!</a>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently editing the following Cancelled Team: </b>
<br>
  <table>
    <tr>
      <th TeamID>
      <th TeamName>
      <th ChurchName>
      <th District>
      <th LeaderEmail>
      <th LeaderName>
      <th Partership>
      -teamCTdelete.forEach(function(element){
        <tr>
          <td #{element.TeamID}>
          <td #{element.TeamName}>
          <td #{element.ChurchName}>
          <td #{element.District}>
          <td #{element.LeaderEmail}>
          <td #{element.LeaderName}>
          <td #{element.Partership}>
        -});
      </tr>
    </table>
<br>
<br>
    <form id="deleteCTteam" action="deleteCTteam" method="POST"
name="deleteCTteam">
      <input type = "hidden" id="newTeamID" name="newTeamID" required
readonly>
      -teamCTdelete.forEach(function(element){
        document.getElementById('TeamID').value=element.TeamID
      -});
      <input type="Submit" value="Delete Record">
    </form>
<br>
<br>
<br>

```

```
<br>  
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Delete a Project</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Delete Project </b>
<br>
<br>
<b> Currently deleting the following Project: </b>
<br>
  <table>
    <tr>
      <th> ProjectID
      <th> TeamID
      <th> ChurchName
      <th> BriefDescriptionConst
        -projectViewArrayDelete.forEach(function(element){
          <tr>
            <td> #{element.ProjectID}
            <td> #{element.TeamID}
            <td> #{element.ChurchName}
            <td> #{element.BriefDescriptionConst}
          </td>
        -});
    </tr>
  </table>
<br>
<br>
  <form id="project-delete" action="project-delete" method="POST"
name="project-delete">
    <input type = "hidden" id="newProjectID" name="newProjectID"
required readonly>
      -teamViewArrayDelete.forEach(function(element){
document.getElementById('ProjectID').value=element.ProjectID
      -});
    <br>
    <br>
    <br>
    <br>
    <input type="Submit" value="Save Changes">
  </form>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Delete Records [AUTHORIZED USERS ONLY. PLEASE ENSURE DATA IS CORRECT
BEFORE SUBMITTING!] </b>
<br>
<br>
<b> Currently deleting the following record: </b>
<br>
  <table>
    <tr>
      <th>TeamDetailsID
      <th>TeamID
      <th>ProjectID
      <th>TeamName
      <tbody>
        <tr>
          <td>
            -tableArrayDelete.forEach(function(element){
              <tr>
                <td> #{element.TeamDetailsID}
                <td> #{element.TeamID}
                <td> #{element.ProjectID}
                <td> #{element.TeamName}
              </tr>
            -});
          </td>
        </tr>
      </tbody>
    </table>
<br>
<br>
<form id="deleteRecord" action="deleteRecord" method="POST"
name="deleteRecord">
  <input type = "hidden" id="tdfd" name="tdfd" required readonly>
    -tableArrayDelete.forEach(function(element){
      document.getElementById('TeamDetailsID').value=element.TeamDetailsID
    -});
  <br>
  <br>
  <br>
  <br>
  <input type="Submit" value="DELETE RECORD">
</form>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY. PLEASE ENSURE DATA IS CORRECT
BEFORE SUBMITTING!] </b>
<br>
<br>
<b> Currently deleting the following Team: </b>
<br>
  <table>
    tr
    th TeamID
    th TeamName
    th ChurchName
    th District
    th LeaderEmail
    th LeaderName
    -teamViewArrayDelete.forEach(function(element){
      tr
      td #{element.TeamID}
      td #{element.TeamName}
      td #{element.ChurchName}
      td #{element.District}
      td #{element.LeaderEmail}
      td #{element.LeaderName}
    -});
  </table>
<br>
<br>
  <form id="team-delete" action="Team-delete" method="POST"
name="team-delete">
    <input type = "hidden" id="newTeamID" name="newTeamID" required
readonly>
    -teamViewArrayDelete.forEach(function(element){
      document.getElementById('TeamID').value=element.TeamID
    -});
    <br>
    <br>
    <br>
    <br>
    <input type="Submit" value="Save Changes">
  </form>

```

```
<br>  
<br>  
</body>  
</html>
```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Member Viewer</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Remove TeamMember (TEAMMEMBERS)[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Select a Team Member to Remove: </b>
<br>
<form id="delete-Member" action="delete-Member" method="POST"
name="delete-Member">
  <table>
    <tr>
      <th>TeamID
      <th>TeamMemberID
      <th>FName
      <th>PhoneNumber
      <th>District
      <th>Email
      <th>FoodAllergies
      <th>MedicalAllergies
      <th>REMOVE MEMBER
      <td colspan="7">
        -tableArrayDeleteMember.forEach(function(element, index){
          <tr>
            <td>#{element.TeamID}
            <td>#{element.TeamMemberID}
            <td>#{element.FName}
            <td>#{element.PhoneNumber}
            <td>#{element.District}
            <td>#{element.Email}
            <td>#{element.FoodAllergies}
            <td>#{element.MedicalAllergies}
            <td><input id= #{index} type="Submit"
name="RowForMemberToDelete" value= #{element.TeamMemberID}>
          -});
        </td>
      </tr>
    </table>
    <br>
    <br>
  </form>
  <br>
  <br>
  <br>
  <a href = '/ScheduledTeamsView&Edit'> HOME </a>

```

```
<br>  
<br>  
<br>  
</body>  
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently editing the following Cancelled Team: </b>
<br>
  <table>
    <tr>
      <th TeamID>
      <th TeamName>
      <th ChurchName>
      <th District>
      <th LeaderEmail>
      <th LeaderName>
      <th Partership>
      -teamCTedit.forEach(function(element){
        <tr>
          <td #{element.TeamID}>
          <td #{element.TeamName}>
          <td #{element.ChurchName}>
          <td #{element.District}>
          <td #{element.LeaderEmail}>
          <td #{element.LeaderName}>
        -});
      </table>
<br>
<br>
  <form id="editCTteam" action="editCTteam" method="POST"
name="editCTteam">
    <input type = "hidden" id="newTeamID" name="newTeamID" required
readonly>
      -teamCTedit.forEach(function(element){
        document.getElementById('TeamID').value=element.TeamID
      -});
    <br>
    <label for="newTeamName">New Team Name: </label>
    <br>
    <input id="newTeamName" type = "text" name = "newTeamName">
    <br>

```

```

        <label for="newChurchName">New Church Name:
</label>
        <br>
        <input id="newChurchName" type = "text" name = "newChurchName">
        <br>
        <label for="newDistrict">New District:                </label>
        <br>
        <input id="newDistrict" type = "text" name = "newDistrict">
        <br>
        <label for="newLeaderEmail">New Leader Email:
</label>
        <br>
        <input id="newLeaderEmail" type = "text" name =
"newLeaderEmail">
        <br>
        <label for="newLeaderName">New Leader Name:
</label>
        <br>
        <input id="newLeaderName" type = "text" name = "newLeaderName">
        <br>
        <label for="newPartership">New Partership: </label>
        <select id = "newPartership" name = "newPartership">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
        </select>
        <br>
        <br>
        <br>
        <br>
        <br>
        <input type="Submit" value="Save Changes">
    </form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project ExpectedFunds Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit ExpectedFunds Records [AUTHORIZED USERS ONLY. PLEASE ENSURE
DATA IS CORRECT BEFORE SUBMITTING!] </b>
<br>
<br>
<b> Currently editing the following record: </b>
<br>
  <table>
    <tr>
      <th TeamDetailsID
      <th ProjectFundsRecieved
      <th CancelTeam
      <th ProjectMoney
      <th ArriveDate
      <th ChurchName
      <th TeamName
      <th TeamID
      <tbody>
        <tr>
          <td <!--tableArrayFundsEditor.forEach(function(element){
          <td <!--element.TeamDetailsID}
          <td <!--element.ProjectFundsRecieved}
          <td <!--element.CancelTeam}
          <td <!--element.ProjectMoney}
          <td <!--element.ArriveDate}
          <td <!--element.ChurchName}
          <td <!--element.TeamName}
          <td <!--element.TeamID}
          </tr>
        </tbody>
      </table>
<br>
<br>
  <form id="editEFs" action="editEFs" method="POST" name="editEFs">
    <input type = "hidden" id="newTeamDetailsID"
name="newTeamDetailsID" required readonly>
      <tbody>
        <tr>
          <td <!--tableArrayFundsEditor.forEach(function(element){
          <td <!--element.TeamDetailsID}
          <td <!--element.ProjectFundsRecieved}
          <td <!--element.CancelTeam}
          <td <!--element.ProjectMoney}
          <td <!--element.ArriveDate}
          <td <!--element.ChurchName}
          <td <!--element.TeamName}
          <td <!--element.TeamID}
          </tr>
        </tbody>
      </table>
    <input type = "hidden" id="newTeamID" name = "newTeamID"
required readonly>

```

```

        -tableArrayFundsEditor.forEach(function(element){
        document.getElementById("newTeamID").value=element.TeamID
        -});
    <br>
    <label for="newProjectFundsRecieved">Funds Recieved?:
</label>
    <br>
    <select id = "newProjectFundsRecieved" name =
"newProjectFundsRecieved">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="newCancelTeam">Team Cancelled?:           </label>
    <br>
    <select id = "newCancelTeam" name = "newCancelTeam">
        <option value = 1>TRUE</option>
        <option value = 0>FALSE</option>
    </select>
    <br>
    <label for="newProjectMoney">New Project Money:       </label>
    <br>
    <input id="newProjectMoney" type = "number" name =
"newProjectMoney">
    <br>
    <label for="newArriveDate">New Arrival Date: </label>
    <br>
    <input id="newArriveDate" type = "date" name = "newArrivalDate">
    <br>
    <label for="newChurchName">New Church Name:           </label>
    <br>
    <input id="newChurchName" type = "text" name = "newChurchName">
    <br>
    <label for="newTeamName">New Team Name:               </label>
    <br>
    <input id="newTeamName" type = "text" name = "newTeamName">
    <br>
    <br>
    <br>
    <br>
    <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Meals [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently creating a Meal: </b>
<br>
  <table>
    <tr>
      <th>MealID
      <th>MealDesc
      <th>DaysForThisMeal
      <th>MealDateStart
      <th>MealDateStop
      <th>MealCost
      <th>HowManyEating
      <th>TeamDetailsID
      -tableArrayEditMeals.forEach(function(element){
        <tr>
          <td>#{element.MealID}
          <td>#{element.MealDesc}
          <td>#{element.DaysForThisMeal}
          <td>#{element.MealDateStart}
          <td>#{element.MealDateStop}
          <td>#{element.MealCost}
          <td>#{element.HowManyEating}
          <td>#{element.TeamDetailsID}
        -});
    </table>
    <br>
    <br>
    <form id="edit-Meals" action="edit-Meals" method="POST" name="edit-
Meals">
      <input type = "hidden" id="newMeal" name="newMeal" required
readonly>
      -tableArrayEditMeals.forEach(function(element){
        document.getElementById('MealID').value=element.MealID
      -});
      <br>
      <label for="newMealDesc">Meal Description:
</label>

```

```

        <br>
        <input id="newMealDesc" type = "text" name = "newMealDesc">
        <br>
        <label for="newMealDateStart">MealDateStart:
</label>
        <br>
        <input id="newMealDateStart" type = "date" name =
"newMealDateStart">
        <br>
        <label for="newMealDateStop">Meal Date Stop:
</label>
        <br>
        <input id="newMealDateStop" type = "date" name =
"newMealDateStop">
        <br>
        <label for="newMealCost">Meal Cost:                </label>
        <br>
        <input id="newMealCost" type = "number" name = "newMealCost">
        <br>
        <label for="newHowManyEating">How Many Eating:
</label>
        <br>
        <input id="newHowManyEating" type = "text" name =
"newHowManyEating">
        <br>
        <br>
        <input type="Submit" value="Save Changes">
    </form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Member(assign team) (TEAM)[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing all teams records: </b>
<br>
<form id="editMemberAssign" action="editMemberAssign" method="POST"
name="editMemberAssign">
<label for ="ttj">Select A Team: </label>
<select name = 'ttj' id = 'ttj'>
  -teamViewArray.forEach(function(element){
    <option value = #{element.TeamID}>#{element.TeamName}</option>
  -});
</select>
  </table>
  <br>
  <br>
  <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently editing the following Project: </b>
<br>
  <table>
    tr
      th ProjectID
      th NewWWCoor
      th DateRec
      th LinkToWwsite
      th UserPermission
      th ProjectTypeID
      th DistrictID
      th City
      th Pastor
      th PastorPhone
      th LocationLink
      th DistanceFromDistrictOff
      th DistrictBudgetCurrent
      th DistrictBudgetLast
      th FEMcurrent
      th FEMlast
      th DateOrganized
      th MembersLast
      th MembersCurrent
      th AvgAttendanceLast
      th AvgAttendanceCurr
      th ProjectDesc
      th TempleCapacity
      th RoofWidth
      th RoofLength
      th BriefDesc
      th PropertyWidth
      th PropertyLength
      th Photos
      th Digital
      th LotLayout
      th ProjectPlans
  </table>

```

```

th MoneyRaised
th MoneyInvested
th Money12Months
th ProjectNeed
th TitleToProperty
th PropertyDeededToChurch
th BuildingPermitNeeded
th BuildingPermitApproved
th PastorApproved
th ChurchBoardApproved
th TreasureApproved
th TreasureApprovedDate
th SuperApproved
th SuperApprovedDate
th DistrictSecretaryApproval
th DistrictSecretaryApprovalDate
th DistrictPriority
th TyTApproved
th TyTPriority
th ProjectCompleted
th NewWwLink
th PhotoLocation
th PermitStart
th PermitEnd
th CountryID
  -tableArrayEditProject.forEach(function(element){
    tr
      td #{element.ProjectID}
      td #{element.WWCoorID}
      td #{element.newDateReceived}
      td #{element.LinkToWwsite}
      td #{element.UserPermission}
      td #{element.ProjectTypeID}
      td #{element.DistrictID}
      td #{element.CityName}
      td #{element.PastorName}
      td #{element.PastorPhone}
      td #{element.LocationLink}
      td #{element.DistanceFromDistrictOff}
      td #{element.DistrictBudgetCurrent}
      td #{element.DistrictBudgetLast}
      td #{element.FEMcurrent}
      td #{element.FEMlast}
      td #{element.DateOrganized}
      td #{element.MembersLast}
      td #{element.MembersCurrent}
      td #{element.AvgAttendanceLast}
      td #{element.AvgAttendanceCurrent}
      td #{element.ProjectDesc}
      td #{element.TempleCapacity}

```

```

        td #{element.RoofWidth}
        td #{element.RoofLength}
        td #{element.BriefDescriptionConst}
        td #{element.PropertyWidth}
        td #{element.PropertyLength}
        td #{element.Photos}
        td #{element.Digital}
        td #{element.LotLayout}
        td #{element.ProjectPlans}
        td #{element.MoneyRaised}
        td #{element.MoneyInvested}
        td #{element.Money12Months}
        td #{element.ProjectNeed}
        td #{element.TitleToProperty}
        td #{element.PropertyDeededToChurch}
        td #{element.BuildingPermitNeeded}
        td #{element.BuildingPermitApproved}
        td #{element.PastorApproved}
        td #{element.ChurchBoardApproved}
        td #{element.TreasureApproved}
        td #{element.TreasureApprovedDate}
        td #{element.SuperApproval}
        td #{element.SuperApprovalDate}
        td #{element.DistrictSecretaryApproval}
        td #{element.DistrictTreasureApprovalDate}
        td #{element.DistrictPriority}
        td #{element.TyTapproved}
        td #{element.TyTPriority}
        td #{element.ProjectCompleted}
        td #{element.WwLink}
        td #{element.PhotoLocation}
        td #{element.PermitStart}
        td #{element.newPermitEnd}
        td #{element.CountryID}
    -});
</table>
<br>
<br>
<form id="edit-Project" action="edit-Project" method="POST"
name="edit-Project">
    <input type = "hidden" id="newProjectID" name="newProjectID"
required readonly>
        -tableArrayEditProject.forEach(function(element){
            <input type = "hidden" name="newProjectID" value =
#{element.ProjectID} required readonly>
            -});
        <br>
        <label for="newWWCoorID">New WW Coordinator:
</label>
        <br>

```

```

        <select id="newWWCoorID" name = "newWWCoorID">
            <option value = 0>None Provided</option>
        </select>
        <br>
        <label for="newDateReceived">New Date Recieved:                </
label>
        <br>
        <input id="newDateReceived" type = "date" name =
"newDateReceived">
        <br>
        <label for="newLinkToWwsite">New Website Link:
</label>
        <br>
        <input id="newLinkToWwsite" type = "text" name =
"newLinkToWwsite">
        <br>
        <label for="newUserPermission">New User Permission:
</label>
        <br>
        <input id="newUserPermission" type = "text" name =
"newUserPermission">
        <br>
        <label for="newProjectTypeID">New Project Type ID?:
</label>
        <br>
        <select id="newProjectTypeID" name = "newProjectTypeID">
            <option value = 0>None Provided</option>
            <option value = 1>Templo</option>
            <option value = 2>Casa Pastoral</option>
            <option value = 3>Aulas De ED</option>
            <option value = 4>Casa Districal</option>
            <option value = 5>Centro Districal</option>
            <option value = 6>Techo</option>
            <option value = 7>NazaGol</option>
            <option value = 10>Extreme Team</option>
            <option value = 11>Medical Team</option>
            <option value = 12>Leadership Training</option>
            <option value = 13>Seminary Maintenance</option>
            <option value = 14>Water Filters</option>
        </select>
        <br>
        <label for="newDistrictID">New District ID:
</label>
        <br>
        <select id="newDistrictID" name = "newDistrictID">
            <option value = 0>None Provided</option>
            <option value = 1>Central (Guatemala)</option>
            <option value = 38>Franja Occidental Guatemala
(Guatemala)</option>
            <option value = 39>Nor Occidente (Honduras)</option>

```

```

        <option value = 40>Centro Sur-Oriente (Honduras)</option>
        <option value = 41>Sureste De Nicaragua</option>
        <option value = 42>Franja Central (Guatemala)</option>
        <option value = 43>Sur Occidental de Peten
(Guatemala)</option>
        <option value = 44>Central (El Salvador)</option>
        <option value = 45>Norte De Nicaragua</option>
        <option value = 46>Noroccidente (Guatemala)</option>
        <option value = 47>Verapaz del Norte (Guatemala)</option>
        <option value = 48>Norte (Guatemala)</option>
        <option value = 49>Verapaz Oreiental (Guatemala)</option>
        <option value = 50>Suroriente (Guatemala)</option>
        <option value = 51>Atlantico (Guatemala)</option>
        <option value = 52>Franja Oriental (Guatemala)</option>
        <option value = 53>Norte de Coban (Guatemala)</option>
        <option value = 54>Occidente (Guatemala)</option>
        <option value = 56>Occidente (El Salvador)</option>
        <option value = 57>Baja Verapaz (Guatemala)</option>
        <option value = 58>Sur (Nicaragua)</option>
        <option value = 59>Oriente (El Salvadoe)</option>
        <option value = 60>Verpaz del Sur (Guatemala)</option>
        <option value = 61>Central (Nicaragua)</option>
    </select>
    <br>
    <label for="newCityName">New City:                </label>
    <br>
    <input id="newCityName" type = "text" name = "newCityName">
    <br>
    <label for="newPastorName">New Pastor Name:
</label>
    <br>
    <input id="newPastorName" type = "text" name = "newPastorName">
    <br>
    <label for="newPastorPhone">New Pastor Phone:
</label>
    <br>
    <input id="newPastorPhone" type = "text" name =
"newPastorPhone">
    <br>
    <label for="newLocationLink">New Location Link:                </
label>
    <br>
    <input id="newLocationLink" type = "text" name =
"newLocationLink">
    <br>
    <label for="newDistanceFromDistrictOff">New Distance From
District Office:                </label>
    <br>
    <input id="newDistanceFromDistrictOff" type = "number" name =
"newDistanceFromDistrictOff">

```

```

        <br>
        <label for="newDistrictBudgetCurrent">New District Budget
Current:      </label>
        <br>
        <input id="newDistrictBudgetCurrent" type = "number" name =
"newDistrictBudgetCurrent">
        <br>
        <label for="newDistrictBudgetLast      ">New District Budget
Last:        </label>
        <br>
        <input id="newDistrictBudgetLast      " type = "number" name =
"newDistrictBudgetLast ">
        <br>
        <label for="newFEMCurrent">New FEM Current:
</label>
        <br>
        <input id="NewFEMCurrent" type = "number" name =
"NewFEMCurrent">
        <br>
        <label for="newFEMLast">New FEM Last:          </label>
        <br>
        <input id="newFEMLast" type = "number" name = "newFEMLast">
        <br>
        <label for="newDateOrganized">New Date Organized:
</label>
        <br>
        <input id="newDateOrganized" type = "date" name =
"newDateOrganized">
        <br>
        <label for="newMembersLast">New Members Last:
</label>
        <br>
        <input id="newMembersLast" type = "number" name =
"newMembersLast">
        <br>
        <label for="newMembersCurrent">New Members Current:
</label>
        <br>
        <input id="newMembersCurrent" type = "number" name =
"newMembersCurrent">
        <br>
        <label for="newAvgAttendanceLast      ">New Avg Attendance
Last:        </label>
        <br>
        <input id="newAvgAttendanceLast " type = "number" name =
"newAvgAttendanceLast">
        <br>
        <label for="newAvgAttendanceCurrent      ">New AvgAttendance
Current      </label>
        <br>

```

```

        <input id="newAvgAttendanceCurrent" type = "number" name =
"newAvgAttendanceCurrent">
        <br>
        <label for="newProjectDesc">New Project Desc:
</label>
        <br>
        <input id="newProjectDesc" type = "text" name = "newProjectDesc
">
        <br>
        <label for="newTempleCapacity">New Temple Capacity:
</label>
        <br>
        <input id="newTempleCapacity" type = "number" name =
"newTempleCapacity">
        <br>
        <label for="newRoofWidth">New Roof Width:
</label>
        <br>
        <input id="newRoofWidth" type = "number" name = "newRoofWidth">
        <br>
        <label for="newRoofLength">New Roof Length:
</label>
        <br>
        <input id="newRoofLength" type = "number" name =
"newRoofLength">
        <br>
        <label for="newBriefDescriptionConst">New Brief Desc Const:
</label>
        <br>
        <input id="newBriefDescriptionConst" type = "text" name =
"newBriefDescriptionConst">
        <br>
        <label for="newPropertyWidth">New Property Width:
</label>
        <br>
        <input id="newPropertyWidth" type = "number" name =
"newPropertyWidth">
        <br>
        <label for="newPropertyLength">New Property Length:
</label>
        <br>
        <input id="newPropertyLength" type = "number" name =
"newPropertyLength">
        <br>
        <label for="newPhotos">New Photos:                </label>
        <br>
        <select id = "newPhotos" name = "newPhotos">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
        </select>

```



```

        <br>
        <label for="newDigital">New Digital:                                </label>
        <br>
        <select id = "newDigital" name = "newDigital">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newLotLayout">New Lot Layout:
</label>
        <br>
        <select id = "newLotLayout" name = "newLotLayout">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newProjectPlans">New Project Plans:
</label>
        <br>
        <select id = "newProjectPlans" name = "newProjectPlans">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newMoneyRaised">New Money Raised:
</label>
        <br>
        <input id="newMoneyRaised" type = "number" name =
"newMoneyRaised">
        <br>
        <label for="newMoneyInvested">New Money Invested:
</label>
        <br>
        <input id="newMoneyInvested" type = "number" name =
"newMoneyInvested">
        <br>
        <label for="newMoney12Months">New Money 12 Months:
</label>
        <br>
        <input id="newMoney12Months" type = "number" name =
"newMoney12Months">
        <br>
        <label for="newProjectNeed">New Project Need:
</label>
        <br>
        <input id="newProjectNeed" type = "text" name =
"newProjectNeed">
        <br>
        <label for="newTitleToProperty">New Title to Property:
</label>

```

```

    <br>
    <select id = "newTitleToProperty" name = "newTitleToProperty">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>
    <label for="newPropertyDeededToChurch">New Property Deeded to
Church:
      </label>
    <br>
    <select id = "newPropertyDeededToChurch" name =
"newPropertyDeededToChurch">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>
    <label for="newBuildingPermitNeeded">New Building Permit Needed:
</label>
    <br>
    <select id = "newBuildingPermitNeeded" name =
"newBuildingPermitNeeded">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>
    <label for="newBuildingPermitApproved">New Building Permit
Approved:
      </label>
    <br>
    <select id = "newBuildingPermitApproved" name =
"newBuildingPermitApproved">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>
    <label for="newPastorApproved">New Pastor Approved:
</label>
    <br>
    <select id = "newPastorApproved" name = "newPastorApproved">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>
    <label for="newChurchBoardApproved">New Church Board Approved:
</label>
    <br>
    <select id = "newChurchBoardApproved" name =
"newChurchBoardApproved">
      <option value = 1>YES</option>
      <option value = 0>NO</option>
    </select>
    <br>

```

```

        <label for="newTreasureApproved">New Treasure Approved:
</label>
        <br>
        <select id = "newTreasureApproved" name = "newTreasureApproved">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newTreasureApprovedDate">New Treasure Approved Date:
</label>
        <br>
        <input id="newTreasureApprovedDate" type = "date" name =
"newTreasureApprovedDate">
        <br>
        <label for="newSuperApproval">New Super Approval:
</label>
        <br>
        <select id = "newSuperApproval" name = "newSuperApproval">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newSuperApprovalDate">New Super Approval Date:
</label>
        <br>
        <input id="newSuperApprovalDate" type = "date" name =
"newSuperApprovalDate">
        <br>
        <label for="newDistrictSecretaryApproval">New District Secretary
Approval:                </label>
        <br>
        <select id = "newDistrictSecretaryApproval" name =
"newDistrictSecretaryApproval">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newDistrictTreasureApproval">New District Treasure
Approval:                </label>
        <br>
        <select id = "newDistrictTreasureApproval" name =
"newDistrictTreasureApproval">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newDistrictTreasureApprovalDate">New District
Treasure Approval Date:                </label>
        <br>

```

```

        <input id="newDistrictTreasureApprovalDate" type = "date" name =
"newDistrictTresureApprovalDate">
        <br>
        <label for="newDistrictPriority">New District Priority:
</label>
        <br>
        <select id = "newDistrictPriority" name = "newDistrictPriority">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newTyTapproved">New TyTapproved:
</label>
        <br>
        <select id = "newTyTapproved" name = "newTyTapproved">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newTyTPriority">New TyTPriority:
</label>
        <br>
        <select id = "newTyTPriority" name = "newTyTPriority">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newProjectCompleted">New Project Completed:
</label>
        <br>
        <select id = "newProjectCompleted" name = "newProjectCompleted">
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newWWLink">New WW Link:                                </label>
        <br>
        <input id="newWWLink" type = "text" name = "newWWLink">
        <br>
        <label for="newPhotoLocation">New Photo Location:
</label>
        <br>
        <input id="newPhotoLocation" type = "text" name =
"newPhotoLocation">
        <br>
        <label for="newPermitStart">New Permit Start:
</label>
        <br>
        <input id="newPermitStart" type = "date" name =
"newPermitStart">

```

```
<br>
<label for="newPermitEnd">New Permit End:
</label>
<br>
<input id="newPermitEnd" type = "date" name = "newPermitEnd">
<br>
<label for="newCountryID">New CountryID:           </label>
<br>
<select id="newCountryID" name = "newCountryID">
  <option value = 1>Guatemala</option>
  <option value = 2>El Salvador</option>
  <option value = 3>Honduras</option>
  <option value = 4>Nicaragua</option>
</select>
<br>
<br>
<input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY. PLEASE ENSURE DATA IS CORRECT
BEFORE SUBMITTING!] </b>
<br>
<br>
<b> Currently editing the following record: </b>
<br>
  <table>
    <tr>
      <th TeamName
      <th Country
      <th Days
      <th ArrivalDate
      <th DepartDate
      <th NumTeamMemb
      <th ProjectID
      <th FName
      <th ProjectMoney
      <th SumOfUSD
      <th Maxima
      -tableArrayEditor.forEach(function(element){
        <tr>
          <td #{element.TeamName}
          <td #{element.CountryName}
          <td #{element.Days}
          <td #{element.ArrivalDate}
          <td #{element.DepartDate}
          <td #{element.NumTeamMemb}
          <td #{element.ProjectID}
          <td #{element.FName}
          <td #{element.ProjectMoney}
          <td #{element.SumOfUSD}
          <td #{element.Maxima}
        -});
    </table>
<br>
<br>
    <form id="edit" action="Edit" method="POST" name="Reditor">
      <input type = "hidden" id="newTeamDetailsID"
name="newTeamDetailsID" required readonly>

```

```

        -tableArrayEditor.forEach(function(element){
document.getElementById('newTeamDetailsID').value=element.TeamDetailsID
        -});
        <input type = "hidden" id = "newProjectID" name = "newProjectID"
required readonly>
        -tableArrayEditor.forEach(function(element){

document.getElementById("newProjectID").value=element.ProjectID
        -});
        <input type = "hidden" id="newTeamID" name = "newTeamID"
required readonly>
        -tableArrayEditor.forEach(function(element){
document.getElementById("newTeamID").value=element.TeamID
        -});
        <br>
        <label for="newTeamName">New Team Name:                </label>
        <br>
        <input id="newTeamName" type = "text" name = "newTeamName">
        <br>
        <label for="newCountryID">New Country:                </label>
        <br>
        <select id="newCountryID" name = "newCountryID">
        <option value = 1>Guatemala</option>
        <option value = 2>El Salvador</option>
        <option value = 3>Honduras</option>
        <option value = 4>Nicaragua</option>
        </select>
        <br>
        <label for="newArriveDate">New Arrival Date: </label>
        <br>
        <input id="newArriveDate" type = "date" name = "newArrivalDate">
        <br>
        <label for="newDepartDate">New Departure Date: </label>
        <br>
        <input id="newDepartDate" type = "date" name = "newDepartDate">
        <br>
        <label for="newNumTeamMemb">New Num Team Members:      </label>
        <br>
        <input id="newNumTeamMemb" type = "number" name =
"newNumTeamMemb">
        <br>
        <label for="newFName">New HostName:
</label>
        <br>
        <input id="newFName" type = "text" name = "newFName">
        <br>
        <label for="newProjectMoney">New Project Money:        </label>
        <br>

```

```

        <input id="newProjectMoney" type = "number" name =
"newProjectMoney">
        <br>
        <label for="newMaxima">New Maxima:                </label>
        <br>
        <select id = "newMaxima" name = "newMaxima">
            <option value = 1>TRUE</option>
            <option value = 0>FALSE</option>
        </select>
        <br>
        <label for="newChurchName">New Church Name:        </label>
        <br>
        <input id="newChurchName" type = "text" name = "newChurchName">
        <br>
        <label for="newArriveFlightNumber">New Arrive Flight #: </label>
        <br>
        <input id="newArriveFlightNumber" type = "number" name =
"newArriveFlightNumber">
        <br>
        <label for="newDepartFlightNumber">New Depart Flight #: </label>
        <br>
        <input id="newDepartFlightNumber" type = "number" name =
"newDepartFlightNumber">
        <br>
        <label for="newCancelTeam">Team Cancelled?:        </label>
        <br>
        <select id = "newCancelTeam" name = "newCancelTeam">
            <option value = 1>TRUE</option>
            <option value = 0>FALSE</option>
        </select>
        <br>
        <br>
        <br>
        <br>
        <input type="Submit" value="Save Changes">
    </form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently editing the following Team: </b>
<br>
  <table>
    <tr>
      <th> TeamID
      <th> TeamName
      <th> ChurchName
      <th> District
      <th> LeaderEmail
      <th> LeaderName
      <tbody>
        -teamViewArrayEdit.forEach(function(element){
          <tr>
            <td> #{element.TeamID}
            <td> #{element.TeamName}
            <td> #{element.ChurchName}
            <td> #{element.District}
            <td> #{element.LeaderEmail}
            <td> #{element.LeaderName}
          </td>
        -});
      </tbody>
    </table>
<br>
<br>
    <form id="edit-Team" action="edit-Team" method="POST" name="edit-
Team">
      <input type = "hidden" id="newTeamID" name="newTeamID" required
readonly>
      <tbody>
        -teamViewArrayEdit.forEach(function(element){
          document.getElementById('TeamID').value=element.TeamID
        -});
      </tbody>
      <br>
      <label for="newTeamName">New Team Name: </label>
      <br>
      <input id="newTeamName" type = "text" name = "newTeamName">
      <br>
      <label for="newChurchName">New Church Name:
</label>

```

```

<br>
<input id="newChurchName" type = "text" name = "newChurchName">
<br>
<label for="newDistrict">New District:                </label>
<br>
<select id="newDistrict" name = "newDistrictID">
  <option value = 0>None Provided</option>
  <option value = 1>Central (Guatemala)</option>
  <option value = 38>Franja Occidental Guatemala
(Guatemala)</option>
  <option value = 39>Nor Occidente (Honduras)</option>
  <option value = 40>Centro Sur-Oriente (Honduras)</option>
  <option value = 41>Sureste De Nicaragua</option>
  <option value = 42>Franja Central (Guatemala)</option>
  <option value = 43>Sur Occidental de Peten
(Guatemala)</option>
  <option value = 44>Central (El Salvador)</option>
  <option value = 45>Norte De Nicaragua</option>
  <option value = 46>Noroccidente (Guatemala)</option>
  <option value = 47>Verapaz del Norte (Guatemala)</option>
  <option value = 48>Norte (Guatemala)</option>
  <option value = 49>Verapaz Oreiental (Guatemala)</option>
  <option value = 50>Suroriente (Guatemala)</option>
  <option value = 51>Atlantico (Guatemala)</option>
  <option value = 52>Franja Oriental (Guatemala)</option>
  <option value = 53>Norte de Coban (Guatemala)</option>
  <option value = 54>Occidente (Guatemala)</option>
  <option value = 56>Occidente (El Salvador)</option>
  <option value = 57>Baja Verapaz (Guatemala)</option>
  <option value = 58>Sur (Nicaragua)</option>
  <option value = 59>Oriente (El Salvadoe)</option>
  <option value = 60>Verpaz del Sur (Guatemala)</option>
  <option value = 61>Central (Nicaragua)</option>
</select>
<br>
<label for="newLeaderEmail">New Leader Email:
</label>
<br>
<input id="newLeaderEmail" type = "text" name =
"newLeaderEmail">
<br>
<label for="newLeaderName">New Leader Name:
</label>
<br>
<input id="newLeaderName" type = "text" name = "newLeaderName">
<br>
<select id = "newCancelled" name = "newCancelled">
  <option value = 1>YES</option>
  <option value = 0>NO</option>
</select>

```

```
        <br>
        <br>
        <br>
        <br>
        <br>
        <input type="Submit" value="Save Changes">
    </form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit TeamDetails [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently editing the following Project/Team Details: </b>
<br>
  <table>
    <tr>
      <th TeamDetailsID
      <th TotalDailyFunds
      <th ArriveFlightNumber
      <th ArriveDate
      <th DepartDate
      <th DepartAirline
      <th DepartFlightNumber
      <th ProjectID
      <th TeamID
      <th DistanceToRR
      <th DistanceToProject
      <th HotelID
      <th CountryID
      <th AirportID
      <th TeamRegistration
      <th Insurance
      <th Maxima
      <th ProjectMoney
      <th PartnershipID
      <th TeamPartnerProfileID
      <th TeamHost
      <th CulBroker
      <th TruckDriver
      <th BusDriver
      <th Translator
      <th StructureDelivery
      <th CancelTeam
      <th ProjectFundsRecieved
      -tableArrayEditTeamDetails.forEach(function(element){
    <tr>
      <td #{element.TeamDetailsID}
      <td #{element.TotalDailyFunds}

```

```

        td #{element.ArriveFlightNumber}
        td #{element.ArriveDate}
        td #{element.DepartDate}
        td #{element.DepartAirline}
        td #{element.DepartFlightNumber}
        td #{element.ProjectID}
        td #{element.TeamID}
        td #{element.DistanceRR}
        td #{element.DistanceToProject}
        td #{element.HotelID}
        td #{element.CountryID}
        td #{element.AirportID}
        td #{element.TeamRegistration}
        td #{element.Insurace}
        td #{element.Maxima}
        td #{element.ProjectMoney}
        td #{element.PartnershipID}
        td #{element.TeamPartnerProfileID}
        td #{element.TeamHost}
        td #{element.CulBroker}
        td #{element.TruckDriver}
        td #{element.BusDriver}
        td #{element.Translator}
        td #{element.StructureDelivery}
        td #{element.CancelTeam}
        td #{element.ProjectFundsRecieved}
    -});
</table>
<br>
<br>
<form id="edit-TeamDetails" action="edit-TeamDetails" method="POST"
name="edit-TeamDetails">
    -tableArrayEditTeamDetails.forEach(function(element){
        <input type = "hidden" name="newTeamDetailsID" value =
#{element.TeamDetailsID} required readonly>
        -});
    <label for="newTotalDailyFunds">New Total Daily Funds:
</label>
    <br>
    <input id="newTotalDailyFunds" type = "number" name =
"newTotalDailyFunds">
    <br>
    <label for="newArriveFlightNumber">New Arrive Flight Number:
</label>
    <br>
    <input id="newArriveFlightNumber" type = "text" name =
"newArriveFlightNumber">
    <br>
    <label for="newArriveDate">New Arrive Date:
</label>

```

```

        <br>
        <input id="newArriveDate" type = "date" name = "newArriveDate">
        <br>
        <label for="newDepartDate">New Depart Date:
</label>
        <br>
        <input id="newDepartDate" type = "date" name = "newDepartDate">
        <br>
        <label for="newDepartAirline">New Depart Airline:
</label>
        <br>
        <input id="newDepartAirline" type = "text" name =
"newDepartAirline">
        <br>
        <label for="newDepartFlightNumber">New Depart Flight Number:
</label>
        <br>
        <input id="newDepartFlightNumber" type = "text" name =
"newDepartFlightNumber">
        <br>
        <label for="newDistanceToRR">New Distance to RR:
</label>
        <br>
        <input id="newDistanceToRR" type = "number" name =
"newDistanceToRR">
        <br>
        <label for="newDistanceToProject">New Distance to Project:
</label>
        <br>
        <input id="newDistanceToProject" type = "number" name =
"newDistanceToProject">
        <br>
        <label for="newTeamRegistration">Team Registration:
</label>
        <br>
        <select id = "newTeamRegistration" name = "newTeamRegistration"
required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newInsurance">New Insurance:                </label>
        <br>
        <select id = "newInsurance" name = "newInsurance" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newMaxima">Maxima:                </label>
        <br>

```

```

        <select id = "newMaxima" name = "newMaxima" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <label for="newProjectMoney">New Project Money:
</label>
        <br>
        <input id="newProjectMoney" type = "number" name =
"newProjectMoney">
        <br>
        <label for="newTeamHost">New Team Host:                </label>
        <br>
        <select id = "newTeamHost" name = "newTeamHost" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="newCulBroker">New CulBroker:
</label>
        <br>
        <select id = "newCulBroker" name = "newCulBroker" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <label for="newTruckDriver">New Truck Driver
</label>
        <br>
        <select id = "newTruckDriver" name = "newTruckDriver" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <label for="newBusDriver">New Bus Driver:
</label>
        <br>
        <select id = "newBusDriver" name = "newBusDriver" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <label for="newTranslator">New Translator:
</label>
        <br>
        <select id = "newTranslator" name = "newTranslator" required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <label for="newStructureDelivery">New Structure Delivery:
</label>
        <br>
        <select id = "newStructureDelivery" name =
"newStructureDelivery" required>

```

```

        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <label for="newCancelTeam">New CancelTeam:
</label>
    <br>
    <select id = "newCancelTeam" name = "newCancelTeam" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <label for="newProjectFundsReceived">New Project Funds Received:
</label>
    <br>
    <select id = "newProjectFundsReceived" name =
"newProjectFundsReceived" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <input type="Submit" value="Proceed To Choose Country">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Member Viewer</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> EDIT TeamMember (TEAMMEMBERS)[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Select a Team Member to EDIT: </b>
<br>
<form id="edit-Member" action="edit-Member" method="POST" name="edit-
Member">
  <table>
    <tr>
      <th TeamID
      <th TeamMemberID
      <th FName
      <th PhoneNumber
      <th District
      <th Email
      <th FoodAllergies
      <th MedicalAllergies
      -tableArrayEditMember.forEach(function(element){
        <tr>
          <td #{element.TeamID}
          <td #{element.TeamMemberID}
          <td #{element.FName}
          <td #{element.PhoneNumber}
          <td #{element.District}
          <td #{element.Email}
          <td #{element.FoodAllergies}
          <td #{element.MedicalAllergies}
        -});
      </table>
      <form id="edit-Member" action="edit-Member" method="POST"
name="edit-Member">
        <input type = "hidden" id="EMTID" name="EMTID" required
readonly>
          -tableArrayEditMember.forEach(function(element){

document.getElementById('TeamMemberID').value=element.TeamMemberID
          -});
        <br>
        <label for="EMFName">Full Name:           </label>

```

```

<br>
<input id="EMFName" type = "text" name = "EMFName">
<br>
<label for="EMPhoneNumber">Phone (format: (#####)):
</label>
<br>
<input id="EMPhoneNumber" type = "text" name = "EMPhoneNumber">
<br>
<select id="EMCountry" name = "EMCountry">
  <option value = 'United States'>United States</option>
  <option value = 'Canada'>Canada</option>
  <option value = 'Honduras'>Mexico</option>
  <option value = 'Nicaragua'>Nicaragua</option>
  <option value = 'Guatemala'>Guatemala</option>
  <option value = 'El Salvador'>El Salvador</option>
  <option value = 'Honduras'>Honduras</option>
  <option value = 'Nicaragua'>Argentina</option>
</select>
<label for="EMDistrict">District Name: </label>
<br>
<input id="EMDistrict" type = "text" name = "EMDistrict">
<br>
<label for="EMPreviousProjects">Have you done a Project for this
organization before?:</label>
<select id = "EMPreviousProjects" name = "EMPreviousProjects"
required>
  <option value = 1>YES</option>
  <option value = 0>NO</option>
</select>
<br>
<label for="EMLastTrip">If so, provide an approximate date,
otherwise, leave this blank: </label>
<br>
<input id="EMLastTrip" type = "date" name = "EMLastTrip">
<br>
<label for="EMSecondLanguage">Do you speak a second language
well enough for basic conversation? if not, select [none]:</label>
<select id = "EMSecondLanguage" name = "EMSecondLanguage"
required>
  <option value = 1>NONE</option>
  <option value = 2>Spanish</option>
  <option value = 3>Portuguese</option>
</select>
<br>
<label for="EMSkillsID">Skills (If more than one applies, Choose
one at your discretion): </label>
<select id = "EMSkillsID" name = "EMSkillsID">
  <option value = 0>NONE</option>
  <option value = 1>Team Host</option>
  <option value = 2>Truck Driver</option>

```

```

        <option value = 3>Bus Driver</option>
        <option value = 4>Translator</option>
        <option value = 5>Electrician</option>
        <option value = 6>Welder</option>
        <option value = 7>Block Layer</option>
        <option value = 8>Tool Driver</option>
    </select>
    <br>
    <label for="EMTeamMemberTypeID">Team Leader Type?</label>
    <select id = "EMTeamMemberTypeID" name = "EMTeamMemberTypeID">
        <option value = 0>NONE</option>
        <option value = 1>USA</option>
        <option value = 2>MAXIMA</option>
    </select>
    <br>
    <label for="EMTeamLeader">Team Leader?</label>
    <select id = "EMTeamLeader" name = "EMTeamLeader">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="EMMale">Are you Male?:</label>
    <select id = "EMMale" name = "EMMale" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="EMFemale">Are you Female?:</label>
    <select id = "EMFemale" name = "EMFemale" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="EMSpouseOnTeam">Is your spouse on the same team as
you? (if N/A, select no):</label>
    <select id = "EMSpouseOnTeam" name = "EMSpouseOnTeam" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="EMPassPortExp">Passport Expiration Date:
</label>
    <br>
    <input id="EMPassPortExp" type = "date" name = "EMPassPortExp"
required>
    <br>
    <b> Please leave the next 2 fields blank if you have no
allergies </b>
    <label for="EMFoodAllergies">Please list all food allergies you
have in a comma separated list: </label>

```

```
        <br>
        <input id="EMFoodAllergies" type = "text" name =
"EMFoodAllergies">
        <br>
        <label for="EMMedicalAllergies">Please list any medical
allergies you have (e.g. penicillin), in a comma seperated list:
</label>
        <br>
        <input id="EMMedicalAllergies" type = "text" name =
"EMMedicalAllergies">
        <br>
        <label for="EMDPI">DPI:                </label>
        <br>
        <input id="EMDPI" type = "text" name = "EMDPI">
        <br>
        <label for="EMEmail">Email Address:                </label>
        <br>
        <input id="EMEmail" type = "text" name = "EMEmail" required>
        <br>
        <br>
        <br>
        <br>
        <br>
        <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing ExpectedFunds: </b>
<br>
<form id="EF" action="EF" method="POST" name="EF">
  <table>
    <tr>
      <th TeamDetailsID
      <th CancelTeam
      <th ProjectMoney
      <th ArriveDate
      <th TeamName
      -tableArrayFunds.forEach(function(element, index){
    <tr>
      <td #{element.TeamDetailsID}
      <td #{element.CancelTeam}
      <td #{element.ProjectMoney}
      <td #{element.ArriveDate}
      <td #{element.TeamName}
      <td <input id= #{index} type="Submit" name="RowForEditEF"
value= #{index}>
      <td
      -});
    </table>
  <br>
  <br>
</form>
<br>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
<br>
</body>
</html>

```

```
<!DOCTYPE html>

<html lang="en">
<head>
  <title>Log In</title>
  <meta charset="utf-8"/>
</head>
<body>
  <form action="login" method="POST">
    <label for="user">User: </label>
    <input id="user" type="text" name="user" required/><br><br>
    <label for="pass">Pass: </label>
    <input id="pass" type="text" name="pass" required/><br><br>
    <input type="Submit" value="Sign In">
  </form>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Team Member Registration [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Adding A New Member. Available Teams Are Listed Below:
</b>
<br>
  <table>
    <tr>
      <th> TeamID
      <th> TeamName
      <th> ChurchName
      <th> District
      <th> LeaderEmail
      <th> LeaderName
      <th> Partership
      <th> index
      <tbody>
        -teamJoinArray.forEach(function(element, index){
          <tr>
            <td> #{element.TeamID}
            <td> #{element.TeamName}
            <td> #{element.ChurchName}
            <td> #{element.District}
            <td> #{element.LeaderEmail}
            <td> #{element.LeaderName}
            <td> #{element.Partership}
          </td>
        -});
      </tbody>
    </table>
<br>
<br>
    <form id="joinATeam" action="joinATeam" method="POST"
name="joinATeam">
      <label for ="ttj">Select A Team: </label>
      <select name = 'ttj' id = 'ttj'>
        -teamJoinArray.forEach(function(element,index){
          <option value =
#{element.TeamID}>#{element.TeamName}</option>
        -});
      </select>
<br>

```

```

        <label for="NTMName">Full Name:                </label>
        <br>
        <input id="NTMName" type = "text" name = "NTMName" required>
        <br>
        <label for="NTMPhoneNumber">Phone (format: (#####)):
</label>
        <br>
        <input id="NTMPhoneNumber" type = "text" name = "NTMPhoneNumber"
required>
        <br>
        <select id="NTMCountry" name = "NTMCountry" required>
            <option value = 'United States'>United States</option>
            <option value = 'Canada'>Canada</option>
            <option value = 'Honduras'>Mexico</option>
            <option value = 'Nicaragua'>Nicaragua</option>
            <option value = 'Guatemala'>Guatemala</option>
            <option value = 'El Salvador'>El Salvador</option>
            <option value = 'Honduras'>Honduras</option>
            <option value = 'Nicaragua'>Argentina</option>
        </select>
        <label for="NTMDistrict">District Name:                </label>
        <br>
        <input id="NTMDistrict" type = "text" name = "NTMDistrict">
        <br>
        <input type = "hidden" id="NTMRankingID" name="NTMRankingID"
value = 0 required readonly>
        <br>
        <label for="NTMPreviousProjects">Have you done a Project for
this organization before?:</label>
        <select id = "NTMPreviousProjects" name = "NTMPreviousProjects"
required>
            <option value = 1>YES</option>
            <option value = 0>NO</option>
        </select>
        <br>
        <label for="NTMLastTrip">If so, provide an approximate date,
otherwise, leave this blank:                </label>
        <br>
        <input id="NTMLastTrip" type = "date" name = "NTMLastTrip">
        <br>
        <label for="NTMSecondLanguage">Do you speak a second language
well enough for basic conversation? if not, select [none]:</label>
        <select id = "NTMSecondLanguage" name = "NTMSecondLanguage"
required>
            <option value = 1>NONE</option>
            <option value = 2>Spanish</option>
            <option value = 3>Portuguese</option>
        </select>
        <br>

```



```

    <label for="NTMSkillsID">Skills (If more than one applies,
Choose one at your discretion):          </label>
    <select id = "NTMSkillsID" name = "NTMSkillsID">
        <option value = 0>NONE</option>
        <option value = 1>Team Host</option>
        <option value = 2>Truck Driver</option>
        <option value = 3>Bus Driver</option>
        <option value = 4>Translator</option>
        <option value = 5>Electrician</option>
        <option value = 6>Welder</option>
        <option value = 7>Block Layer</option>
        <option value = 8>Tool Driver</option>
    </select>
    <br>
    <label for="NTMTeamMemberTypeID">Team Leader Type?</label>
    <select id = "NTMTeamMemberTypeID" name = "NTMTeamMemberTypeID">
        <option value = 0>NONE</option>
        <option value = 1>USA</option>
        <option value = 2>MAXIMA</option>
    </select>
    <br>
    <label for="NTMTeamLeader">Team Leader?</label>
    <select id = "NTMTeamLeader" name = "NTMTeamLeader">
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <input type = "hidden" id="TeamDetailsID" name="TeamDetailsID"
value = 0 required readonly>
    <input type = "hidden" id="TeamID" name="TeamID" value = 0
required readonly>
    <label for="NTMMale">Are you Male?:</label>
    <select id = "NTMMale" name = "NTMMale" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="NTMFemale">Are you Female?:</label>
    <select id = "NTMFemale" name = "NTMFemale" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="NTMSpouseOnTeam">Is your spouse on the same team as
you? (if N/A, select no):</label>
    <select id = "NTMSpouseOnTeam" name = "NTMSpouseOnTeam"
required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>

```

```

        <br>
        <label for="NTMPassPortExp">Passport Expiration Date:
</label>
        <br>
        <input id="NTMPassPortExp" type = "date" name = "NTMPassPortExp"
required>
        <br>
        <b> Please leave the next 2 fields blank if you have no
allergies </b>
        <label for="NTMFoodAllergies">Please list all food allergies you
have in a comma seperated list:          </label>
        <br>
        <input id="NTMFoodAllergies" type = "text" name =
"NTMFoodAllergies">
        <br>
        <label for="NTMMedicalAllergies">Please list any medical
allergies you have (e.g. penicillin), in a comma seperated list:
</label>
        <br>
        <input id="NTMMedicalAllergies" type = "text" name =
"NTMMedicalAllergies">
        <br>
        <label for="NTMDPI">DPI:          </label>
        <br>
        <input id="NTMDPI" type = "text" name = "NTMDPI">
        <br>
        <label for="NTMEmail">Email Address:          </label>
        <br>
        <input id="NTMEmail" type = "text" name = "NTMEmail" required>
        <br>
        <br>
        <br>
        <br>
        <br>
        <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Admin</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Project Teams(MAXIMA ONLY) </b>
<br>
<br>
<a href="/ScheduledTeamsView&edit" >Click to see all teams(Under
Construction)</a>
<br>
<br>
<input type="Button" value="Projects Per District">
<input type="reset" value="Refresh" onClick ="window.location.reload()">
<input type="Button" value="Expected Funds">
<button onClick="window.location=window.location">Email List</button>
<input type="Button" value="Cancelled Teams">
<input type="Button" value="Logout">
<br>
<br>
  table
    tr
    th TeamName
    th Country
    th Days
    th ArrivalDate
    th DepartDate
    th NumTeamMemb
    th ProjectID
    th FName
    th ProjectMoney
    th SumOfUSD
    th Maxima
    th EDIT
    th DELETE
    -tableArray.forEach(function(element){
    tr
    td #{element.TeamName}
    td #{element.CountryName}
    td #{element.Days}
    td #{element.ArrivalDate}
    td #{element.DepartDate}
    td #{element.NumTeamMemb}
    td #{element.ProjectID}

```

```
td   #{element.FName}
td   #{element.ProjectMoney}
td   #{element.SumOfUSD}
td   #{element.Maxima}
td   <input type="button" value="...">
td   <input type="button" value="X">
-});

<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> View Meals [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently Viewing All Meals : </b>
<br>
  <form id="get-MealsID" action="get-MealsID" method="POST" name="get-
MealsID">
    <table>
      <tr>
        <th>MealID
        <th>MealDesc
        <th>DaysForThisMeal
        <th>MealDateStart
        <th>MealDateStop
        <th>MealCost
        <th>HowManyEating
        <th>TeamDetailsID
        <th>DELETE
          -MealsViewArray.forEach(function(element){
            <tr>
              <td>#{element.MealID}
              <td>#{element.MealDesc}
              <td>#{element.DaysForThisMeal}
              <td>#{element.MealDateStart}
              <td>#{element.MealDateStop}
              <td>#{element.MealCost}
              <td>#{element.HowManyEating}
              <td>#{element.TeamDetailsID}
              <td><input id= #{index} type="Submit"
name="RowForMealToDelete" value= #{index}>
              <td><input type = "hidden" name = "MID" value =
#{element.MealID}>
              -});
            </tr>
          </table>
          <br>
          <br>
          <input type="Submit" value="Save Changes">
        </form>
      <br>

```

```
<br>
<a href = '/createMeals'>Create A New Meal</a>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Airport Creator/Editor[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently registering a new airport: </b>
<br>
  <br>
  <br>
  <form id="new-Airport" action="new-Airport" method="POST" name="new-
Airport">
    <br>
    <label for="newAirportName">Airport Name: </label>
    <br>
    <input id="newAirportName" type = "text" name = "newAirportName"
required>
    <br>
    <label for="newAirportCode">Airport Code: </label>
    <br>
    <input id="newAirportCode" type = "text" name = "newAirportCode"
required>
    <br>
    <label for="newAirportCity">Airport City: </label>
    <br>
    <input id="newAirportCity" type = "text" name = "newAirportCity"
required>
    <br>
    <label for="newAirportCountry">Airport Country: </
label>
    <br>
    <label for ="nac">Select A Country: </label>
    <select name = 'nac' id = 'nac' required>
      -tableArrayChooseCountry.forEach(function(element,index){
        <option value =
#{element.CountryID}>#{element.CountryName}</option>
      -});
    </select>
    <input type="Submit" value="Save Changes">
  </form>
<br>
<br>

```

```
<br>  
<br>  
<br>  
<br>  
</body>  
</html>
```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Country [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently registering a new country: </b>
<br>
  <br>
  <br>
  <form id="new-Country" action="new-Country" method="POST" name="new-
Country">
    <br>
    <label for="newCountryName">New Country Name:
</label>
    <br>
    <input id="newCountryName" type = "text" name =
"newCountryName">
    <br>
    <label for ="ncf">Select A Field: </label>
    <select name = 'ncf' id = 'ncf'>
      -tableArrayField.forEach(function(element,index){
        <option value =
#{element.FieldID}>#{element.FieldName}</option>
      -});
    </select>
    <br>
    <br>
    <input type="Submit" value="Save Changes">
  </form>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Edit Hotel [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently registering a new Hotel: </b>
<br>
  <br>
  <br>
  <form id="new-Hotel" action="new-Hotel" method="POST" name="new-
Hotel">
    <br>
    <label for="newHotelName">Hotel Name: </label>
    <br>
    <input id="newHotelName" type = "text" name = "newHotelName">
    <br>
    <label for="newHotelCountry">Hotel Country:
</label>
    <br>
    <select name = 'hc' id = 'hc'>
      -tableArrayChooseCountry.forEach(function(element,index){
        <option value =
#{element.CountryID}>#{element.CountryName}</option>
      -});
    </select>
    <br>
    <label for="RandR">R and R: </label>
    <br>
    <input id="RandR" type = "number" name = "RandR">
    <br>
    <label for="Comments">Hotel Comments: </label>
    <br>
    <input id="Comments" type = "text" name = "Comments">
    <br>
    <label for="PhoneNum">Phone Number: </label>
    <br>
    <input id="PhoneNum" type = "text" name = "PhoneNum">
    <br>
    <label for="WiFi">WiFo: </label>
    <br>
    <select id = "WiFi" name = "WiFi" required>

```

```

        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="AirCond">Air Conditioning:                </label>
    <br>
    <select id = "AirCond" name = "AirCond" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="Pool">Pool:                                </label>
    <br>
    <select id = "Pool" name = "Pool" required>
        <option value = 1>YES</option>
        <option value = 0>NO</option>
    </select>
    <br>
    <label for="MapLoc">MapLoc:                            </label>
    <br>
    <input id="MapLoc" type = "text" name = "MapLoc">
    <br>
    <label for="city">City:                                </label>
    <br>
    <input id="City" type = "text" name = "City">
    <br>
    <label for="LastUsed">Last Used (if n/a leave blank):
</label>
    <br>
    <input id="LastUsed" type = "date" name = "LastUsed">
    <br>
    <br>
    <input type="Submit" value="Save Changes">
</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> View Records [AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Currently viewing Projects By District: (View Only) </b>
<br>
<form id="PBD" action="PBD" method="GET" name="PBD">
  <table>
    <tr>
      <th> DistrictName
      <th> ArriveDate
      <th> CountryName
      -tableArrayPBD.forEach(function(element, index){
        <tr>
          <td> #{element.DistrictName}
          <td> #{element.ArriveDate}
          <td> #{element.CountryName}
        -});
    </table>
  <br>
  <br>
</form>
<br>
<br>
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
<br>
</body>
</html>

```

```
<!DOCTYPE html>

<html lang="en">
<head>
  <title>Log In</title>
  <meta charset="utf-8"/>
</head>
<body>
  <form action="register" method="POST">
    <label for="newUser">New User Username: </label>
    <input id="newUser" type="text" name="newUser" required><br><br>
    <label for="newPass">New User Password: </label>
    <input id="newPass" type="text" name="newPass" required><br><br>
    <label for="newEmail">New User Email: </label>
    <input id="newEmail" type="text" name="newEmail" required><br><br>
    <input type="Submit" value="Register">
  </form>
</body>
</html>
```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Teams Admin</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Project Teams </b>
<br>
<br>
<a href="/MaximaTeams" >Click to see Maxima Teams(Under
Construction)</a>
<br>
<br>
<a href="/ppd" >Projects Per District</a>
<input type="Button" value="Refresh" onClick
="window.location.reload()">
<a href="/funds" >Expected Funds</a>
<a href="/allProjects" >All Projects</a>
<a href="/allTeams" >All Teams</a>
<a href="/assignTeams" >Assign A Team</a>
<a href="/cancelled" >Cancelled Teams</a>
<a href="/joinTeam" >Join a Team</a>
<a href="/createProject" >Create a new Project</a>
<a href="/createTeam" >Create a new Team</a>
<a href="/newAirport" >Register or edit an Airport</a>
<a href="/newCountry" >Register or edit a Country</a>
<a href="/newHotel" >Register or edit a Hotel</a>
<a href="/allTeamDetails">Edit TeamDetails</a>
<a href="/viewHotels">View all Hotels</a>
<a href="/viewCountries">View all Countries</a>
<a href="/viewAirports">View all Airports</a>

<form method = "GET" action = "logout">
  <input type="Submit" value="Logout">
</form>
<br>
<br>
<form action = "GetIndex" method = "POST">
  table
    tr
    th TeamName
    th Country
    th Days
    th ArrivalDate

```

```

th DepartDate
th NumTeamMemb
th ProjectID
th TeamHost
th ProjectMoney
th SumOfUSD
th Maxima
th EDIT
th DELETE
    -tableArray.forEach(function(element, index){
      tr
      td #{element.TeamName}
      td #{element.CountryName}
      td #{element.Days}
      td #{element.ArrivalDate}
      td #{element.DepartDate}
      td #{element.NumTeamMemb}
      td #{element.ProjectID}
      td #{element.FName}
      td #{element.ProjectMoney}
      td #{element.SumOfUSD}
      td #{element.Maxima}
      td <input id= #{index} type="Submit" name="RowForEdit"
value= #{index}>
      td <input id= #{index} type="Submit" name="RowForDelete"
value= #{index}>
      -});
</form>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>All Countries</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Viewing all currently registered airports</b>
<br>
<br>
<br>
<a href="/newCountry" >Make A New Airport Record here</a>
<br>
<br>
  <table>
    <tr>
      <th> AirportID
      <th> AirportName
      <th> AirportCode
      <th> City
      <th> CountryID
      <tbody>
        <tr>
          <td> <td> <td> <td> <td>
            -tableArrayAllAirports.forEach(function(element){
            -});
        </tr>
      </tbody>
    </table>
  <br>
  <br>
  <a href="/ScheduledTeamsView&Edit" >HOME</a>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```



```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>All Countries</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Viewing all currently registered countries</b>
<br>
<br>
<br>
<a href="/newCountry" >Make A New Country Record here</a>
<br>
<br>
  <table>
    <tr>
      <th>CountryID
      <th>CountryName
      <th>FieldID
      -tableArrayAllCountries.forEach(function(element){
      <tr>
      <td> #{element.CountryID}
      <td> #{element.CountryName}
      <td> #{element.FieldID}
      -});
    </table>
  <br>
  <br>
  <a href="/ScheduledTeamsView&Edit" >HOME</a>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Project Record Editor</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> Viewing all currently registered hotels</b>
<br>
<br>
<br>
<br>
<a href="/newHotel" >Or Make A New Hotel here</a>
<br>
<br>
  <table>
    <tr>
      <th>HotelID
      <th>HotelName
      <th>City
      <th>PhoneNum
      <tbody>
        <tr>
          <td>
            -tableArrayAllHotels.forEach(function(element){
              <tr>
                <td> #{element.HotelID}
                <td> #{element.HotelName}
                <td> #{element.City}
                <td> #{element.PhoneNum}
              </tr>
            </tbody>
          </tr>
        </table>
      <br>
      <br>
<br>
<br>
<a href="/ScheduledTeamsView&Edit" >HOME</a>
<br>
<br>
<br>
<br>
</body>
</html>

```

```

<!DOCTYPE html>

<html lang="en">
<head>
  <title>Teams Member Viewer</title>
  <meta charset="utf-8"/>
</head>
<body>
<br>
<br>
<b> View/Remove TeamMembers (TEAMMEMBERS)[AUTHORIZED USERS ONLY] </b>
<br>
<br>
<b> Select a Team Member to Remove if desired: </b>
<br>
<form id="getMemberID" action="getMemberID" method="POST"
name="getMemberID">
  <table>
    <tr>
      <th TeamMemberID
      <th FName
      <th PhoneNumber
      <th District
      <th Email
      <th FoodAllergies
      <th MedicalAllergies
      <th EDIT MEMBER
      <th REMOVE MEMBER
      -MembersArray.forEach(function(element, index){
    <tr>
      <td #{element.TeamMemberID}
      <td #{element.FName}
      <td #{element.PhoneNumber}
      <td #{element.District}
      <td #{element.Email}
      <td #{element.FoodAllergies}
      <td #{element.MedicalAllergies}
      <td <input id= #{index} type="Submit"
name="RowForMemberToEdit" value= #{index}>
      <td <input id= #{index} type="Submit"
name="RowForMemberToDelete" value= #{index}>
        <input type = "hidden" name = "TMID" value =
#{element.TeamMemberID}>
      -});
    </table>
<br>
<br>
</form>
<br>

```

```
<br>
<br>
<a href = '/ScheduledTeamsView&Edit'> HOME </a>
<br>
<br>
<br>
<br>
</body>
</html>
```

## Server.js Code

```
////////////////////////////////////  
////////////////////////////////////  
  
//  
//  
  
//                (C)MISSION2GO  
//  
  
//                2020  
//  
  
//  LINES MARKED /*** ARE NOT MY ORIGINAL WORK BUT WERE USED UPON  
//  OBTAINING    //  
  
//                GRACIOUS WRITTEN PERMISSION FROM THE AUTHOR  
//  
  
//  
//  
  
//  
//  
  
////////////////////////////////////  
////////////////////////////////////  
  
const express = require('express'); //prereq  
  
const uuid = require('uuid/v4');  
  
const session = require('express-session'); //manages session cookies  
  
const helmet = require('helmet');  
  
const https = require('https');  
  
const fs = require('fs');  
  
const app = express(); //prereq
```

```

const sql = require('mssql');

const path = require('path');

var cookieParser = require('cookie-parser');

app.use(helmet());

var bodyParser = require('body-parser'); //for finding user and pass
data

var moment = require('moment');

app.use(cookieParser());

//configure session for cookies and uuid

//DB Credentials

const config = {

    user:      'sa',

    password: 'M2Gadmin',

    server:    'localhost',

    database: 'mission2Go'

};

var options = {

    ttl: 40000

};

app.use(bodyParser.urlencoded({extended : true}));

app.use(bodyParser.json());

//Includes for Redis (memory-based session store)

const redis = require('redis');

const redisStore = require('connect-redis')(session);

```

```

const redisCli = redis.createClient();

redisCli.on('error', (err) => {

    console.log('ERROR: REDIS FAILED: ');

    console.log(err);

});

app.use(

    session({

        id: (req) => {

            console.log(req.sessionID);

            return uuid(); // uuid random string is session id (encrypted by
middleware)

        },

        store: new redisStore({host: 'localhost', port: 6379, client:
redisCli, ttl: 90000}),

        secret: 'adminSecret',

        name: 'RedisConn',

        resave: false,

        saveUninitialized: true,

        cookie: {secure: false},

    )))

var indexV;

var indexE;

var indexD;

```

```

var indexA;

var indexP;

var TIDref;

var PIDref;

var indexTIDE;

var indexTIDD;

var ids = [];

var defaults = [];

var TDFD;

////////////////////////////////////
////////

//X   set up DB credentials, load login page, and ensure DB is
connected   X//

////////////////////////////////////
////////

app.get('/', async function(req,res){

    app.set('view engine', 'pug'); // added for pug

    app.set("views", path.join(__dirname, "views"));

    res.render('default'); //added for pug

    var sql = require('mssql'); //official mssql package

    //credentials

    var connection = sql.connect(config, function(err){

        if (err) { console.log(err); } //print what's wrong

        else { console.log('[DATABASE CONNECTED]'); } //good! move
on....

```



```

});

console.log(req.sessionID);

console.log("<- SessionID");

});

////////////////////////////////////
////////////////////////////////////

//X          use POST data from login and process it
X//

////////////////////////////////////
////////////////////////////////////

app.post('/login', function(request, response, next){

    var user = request.body.user; //pull credentials that were
entered and save them

    var pass = request.body.pass;

    const promise1 = new Promise(function (resolve, reject) { //promise
awaits results before proceeding

        var check = new sql.Request();

        check.input('user', sql.NVarChar(64), user); //pass user and
pass to stored procedure

        check.input('pass', sql.NVarChar(64), pass);

        check.execute('AuthenticateUser').then(function (result,
recordset){

            var flag = result.returnValue;

            resolve(flag);

```

```

    }).catch(function(error){
        console.log(error);
        sql.close();
        reject(error);
    });
});

promise1.then((verdict) => {
    if (verdict === 1) {
        console.log("...[LOGIN SUCCESS!]...");
        var SessInfo = request.session;
        SessInfo.user= user;
        response.redirect('ScheduledTeamsView&Edit');
        //response.end();    //maybe?????
    }
    else{
        console.log("...[LOGIN FAILURE!]...");
        response.send('Login Failed! Please return to the
previous page and try again!');
    }
});

});

app.get('/registration', function(req,res){
    res.render('registration');

```

```

});

app.post('/register', function(req, res){

    var newUser = req.body.newUser;

    var newPass = req.body.newPass;

    var newEmail = req.body.newEmail;

    const promiseReg = new Promise(function (resolve, reject) { //promise
    awaits results before proceeding

        var R = new sql.Request();

        R.input('newUser', sql.NVarChar(64), newUser); //pass user and
    pass to stored procedure

        R.input('newPass', sql.NVarChar(64), newPass);

        R.input('newEmail', sql.NVarChar(64), newEmail);

        R.execute('registerUser').then(function (result, recordset){

            var flag = result.returnValue;

            resolve(flag);

        }).catch(function(error){

            console.log(error);

            sql.close();

            reject(error);

        });

    });

    promiseReg.then((verdict) => {

        if (verdict === 0) {

```

```

        console.log("...USER ALREADY EXISTS...");

        res.send("THIS USER ALREADY EXISTS");

        res.redirect('/login');

        //maybe?????
    }

    else{

        console.log("...ACCOUNT CREATION SUCCESSFUL...");

        res.redirect('/');

    }

});

});

////////////////////////////////////
////////////////////////////////////

//--                Maxima Teams Page handler
--//

////////////////////////////////////
////////////////////////////////////

app.get('/MaximaTeams', function (req, res){

    const promise3 = new Promise(async function r (resolve,
reject) {

        var check3 = new sql.Request();

        var tableArray = [];

        check3.execute('ProjectTeamTables').then(function(result){

            var len = result.recordset.length;

```

```

var today = new Date();

for(var i = 0; i < len; i++) {

    tableArray.push({

        "Maxima":          result.recordset[i].Maxima,

        "TeamName":
result.recordset[i].TeamName,

        "CountryName":
result.recordset[i].CountryName,

        "Days":
Math.round((result.recordset[i].ArrivalDate - today)/(1000*60*60*24)),

        "ArrivalDate":
moment(result.recordset[i].ArrivalDate).format('YYYY-MM-DD'),

        "DepartDate":
moment(result.recordset[i].DepartDate).format('YYYY-MM-DD'),

        "NumTeamMemb":
result.recordset[i].NumTeamMemb,

        "ProjectID":
result.recordset[i].ProjectID,

        "FName":          result.recordset[i].FName,

        "ProjectMoney":
result.recordset[i].ProjectMoney,

        "SumOfUSD":
result.recordset[i].SumOfUSD,

    });

};

tableArray=tableArray.filter(tableArray => tableArray.Maxima
== 1);

console.log("redirect success");

res.render('MaximaTeams', {tableArray : tableArray});

```

```

    });
});
});

////////////////////////////////////
////////////////////////////////////

//X                The default visible page upon login
X//

////////////////////////////////////
////////////////////////////////////

app.get('/ScheduledTeamsView&Edit', function (req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    const promise2 = new Promise(function (resolve, reject) {

    var check2 = new sql.Request();

    var tableArray = [];

    check2.execute('ProjectTeamTables').then(function(result){

        var len = result.recordset.length;

        var today = new Date();

        for(var i = 0; i < len; i++) {

            tableArray.push({

                "Maxima":        result.recordset[i].Maxima,

                "TeamName":      result.recordset[i].TeamName,

```

```

        "CountryName":
result.recordset[i].CountryName,

        "Days":
Math.round((result.recordset[i].ArrivalDate - today)/(1000*60*60*24)),

        "ArrivalDate":
moment(result.recordset[i].ArrivalDate).format('YYYY-MM-DD'),

        "DepartDate":
moment(result.recordset[i].DepartDate).format('YYYY-MM-DD'),

        "NumTeamMemb":
result.recordset[i].NumTeamMemb,

        "ProjectID":      result.recordset[i].ProjectID,

        "FName":          result.recordset[i].FName,

        "ProjectMoney":
result.recordset[i].ProjectMoney,

        "SumOfUSD":      result.recordset[i].SumOfUSD,

    });

};

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

    resolve(tableArray);

    console.log("redirect success");

    res.render('ScheduledTeamsView&Edit', {tableArray :
tableArray});

    delete tableArray;

```

```

        });
    });
});

////////////////////////////////////
////////////////////////////////////

//      Get the row of the record and send that records info to editor
//

////////////////////////////////////
////////////////////////////////////

app.post('/GetIndex', function(req,res){

    indexE = req.body.RowForEdit;

    indexD = req.body.RowForDelete;

    console.log(indexE);

    console.log(indexD);

    if(indexD == undefined){

        res.redirect('editRecord');

    }

    else {

        res.redirect('deleteRecord');

    }

});

//res.end();

app.get('/editRecord', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

```



```

if(UsNm == null && sessionID == null){res.redirect('/')};

console.log(`logged in as: ${UsNm}`);

var tableArrayEditor = [];

// const promise4 = new Promise(function(resolve, reject) {

var check4 = new sql.Request();

check4.execute('ProjectTeamTables').then(function(result){

    var today = new Date();

    var ie = indexE;

        tableArrayEditor.push({

            "TeamDetailsID":
result.recordset[ie].TeamDetailsID, //PK, MAKE NOT EDITABLE

            "TeamID":                result.recordset[ie].TeamID,
//PK, MAKE NOT EDITABLE

            "ChruchName":            result.recordset[ie].ChurchName,
//EDITABLE, BUT NOT IN INITAL view

            "CancelTeam":            result.recordset[ie].CancelTeam,

            "ArriveFlightNumber":
result.recordset[ie].ArriveFlightNumber,

            "DepartFlightNumber":
result.recordset[ie].DepartFlightNumber,

            "Maxima":                result.recordset[ie].Maxima,

            "TeamName":              result.recordset[ie].TeamName,

            "CountryName":
result.recordset[ie].CountryName,

            "Days":
Math.round((result.recordset[ie].ArrivalDate - today)/(1000*60*60*24)),

            "ArrivalDate":
moment(result.recordset[ie].ArrivalDate).format('YYYY-MM-DD'),

```

```

                "DepartDate":
moment(result.recordset[ie].DepartDate).format('YYYY-MM-DD'),

                "NumTeamMemb":
result.recordset[ie].NumTeamMemb,

                "ProjectID":           result.recordset[ie].ProjectID,
//PK, MAKE NOT EDITABLE

                "FName":               result.recordset[ie].FName,

                "ProjectMoney":
result.recordset[ie].ProjectMoney,

                "TID":                 result.recordset[ie].TID

            });

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

console.log(tableArrayEditor);

ids[0] = result.recordset[ie].TeamDetailsID;
ids[1] = result.recordset[ie].TeamID;
ids[2] = result.recordset[ie].ProjectID;
ids[3] = result.recordset[ie].TID;

console.log(ids[0]);
console.log(ids[1]);
console.log(ids[2]);

res.render('editRecord', {tableArrayEditor : tableArrayEditor});

res.end();

});

});

```

```

app.post('/Edit', function(req, res){

    var NewTeamDetailsID = ids[0]; //PK

    var NewTeamName = req.body.newTeamName;

        if(NewTeamName == null || NewTeamName == undefined ||
NewTeamName.length === 0){

            NewTeamName = 'None Provided';

        }

    var NewNumTeamMemb = req.body.newNumTeamMemb;

        if(NewNumTeamMemb == null || NewNumTeamMemb == undefined ||
NewNumTeamMemb.length === 0){

            NewNumTeamMemb = 0;

        }

    var NewProjectID = ids[2];          //PK

    var NewArrivalDate = req.body.newArrivalDate;

        if(NewArrivalDate == null || NewArrivalDate == undefined ||
NewArrivalDate.length === 0){

            NewArrivalDate = new Date();

            moment(NewArrivalDate).format('YYYY-MM-DD');

        }

    var NewDepartDate = req.body.newDepartDate;

        if(NewDepartDate == null || NewDepartDate == undefined ||
NewDepartDate.length === 0){

            NewDepartDate = new Date();

            moment(NewDepartDate).format('YYYY-MM-DD');

        }

}

```

```

    }

    var NewCountryID = req.body.newCountryID;

    var NewTeamID = ids[1];                //PK

    var NewChurchName = req.body.newChurchName;

    if(NewChurchName == null || NewChurchName == undefined ||
NewChurchName.length === 0){

        NewChurchName = 'None Provided';

    }

    var NewFName = req.body.newFName;

    if(NewFName == null || NewFName == undefined || NewFName.length
=== 0){

        NewFName = 'None Provided';

    }

    var NewMaxima = req.body.newMaxima;

    var NewArriveFlightNumber = req.body.newArriveFlightNumber;

    if(NewArriveFlightNumber == null || NewArriveFlightNumber ==
undefined || NewArriveFlightNumber.length === 0){

        NewArriveFlightNumber = 'None Provided';

    }

    var NewDepartFlightNumber = req.body.newDepartFlightNumber;

    if(NewDepartFlightNumber == null || NewDepartFlightNumber ==
undefined || NewDepartFlightNumber.length === 0){

        NewDepartFlightNumber = 'None Provided';

    }

```

```
var NewCancelTeam = req.body.newCancelTeam;

var NewTID = ids[3];

var NewProjectMoney = req.body.newProjectMoney;

    if(NewProjectMoney == null || NewProjectMoney == undefined ||
NewProjectMoney.length === 0){

        NewProjectMoney = 0.0;

    }

console.log(NewTeamDetailsID);
console.log(NewTeamID);
console.log(NewChurchName);
console.log(NewCancelTeam);
console.log(NewArriveFlightNumber);
console.log(NewDepartFlightNumber);
console.log(NewMaxima);
console.log(NewTeamName);
console.log(NewCountryID);
console.log(NewArrivalDate);
console.log(NewDepartDate);
console.log(NewNumTeamMemb)
console.log(NewProjectID);
console.log(NewFName);
console.log(NewProjectMoney);
```

```

console.log(NewTID);

var record = new sql.Request();

record.input('NewTeamDetailsID', sql.INT, NewTeamDetailsID); //PK

record.input('NewTeamName', sql.NVARCHAR(255), NewTeamName);

record.input('NewNumTeamMemb', sql.INT, NewNumTeamMemb);

record.input('NewProjectID', sql.INT, NewProjectID);

record.input('NewArrivalDate', sql.DATE, NewArrivalDate);

record.input('NewDepartDate', sql.DATE, NewDepartDate);

record.input('NewCountryID', sql.INT, NewCountryID);

record.input('NewTeamID', sql.INT, NewTeamID); //PK

record.input('NewChurchName', sql.NVARCHAR(255), NewChurchName);

record.input('NewFName', sql.NVARCHAR(255), NewFName);

record.input('NewMaxima', sql.INT, NewMaxima);

record.input('NewArriveFlightNumber', sql.NVARCHAR(255),
NewArriveFlightNumber);

record.input('NewDepartFlightNumber', sql.NVARCHAR(255),
NewDepartFlightNumber);

record.input('NewCancelTeam', sql.INT, NewCancelTeam);

record.input('NewProjectMoney', sql.MONEY, NewProjectMoney);

record.input('NewTID', sql.INT, NewTID);

record.execute('EditRecord').then(function(result, recordset){

    console.log(result, recordset);

    delete ids;

});

```

```

        res.redirect('ScheduledTeamsView&Edit');

    });

app.get('/deleteRecord', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayDelete = [];

    // const promise4 = new Promise(function(resolve, reject) {

    var check5 = new sql.Request();

    check5.execute('ProjectTeamTables').then(function(result){

        var today = new Date();

        var ie = indexD

        TDFD = result.recordset[ie].TeamDetailsID;

            tableArrayDelete.push({

                "TeamDetailsID":
result.recordset[ie].TeamDetailsID, //PK, MAKE NOT EDITABLE

                "TeamID":                result.recordset[ie].TeamID,
//PK, MAKE NOT EDITABLE

                "ChruchName":            result.recordset[ie].ChurchName,
//EDITABLE, BUT NOT IN INITAL view

                "CancelTeam":            result.recordset[ie].CancelTeam,

```

```

        "ArriveFlightNumber":
result.recordset[ie].ArriveFlightNumber,

        "DepartFlightNumber":
result.recordset[ie].DepartFlightNumber,

        "Maxima":                result.recordset[ie].Maxima,

        "TeamName":              result.recordset[ie].TeamName,

        "CountryName":
result.recordset[ie].CountryName,

        "Days":
Math.round((result.recordset[ie].ArrivalDate - today)/(1000*60*60*24)),

        "ArrivalDate":
moment(result.recordset[ie].ArrivalDate).format('YYYY-MM-DD'),

        "DepartDate":
moment(result.recordset[ie].DepartDate).format('YYYY-MM-DD'),

        "NumTeamMemb":
result.recordset[ie].NumTeamMemb,

        "ProjectID":              result.recordset[ie].ProjectID,
//PK, MAKE NOT EDITABLE

        "FName":                  result.recordset[ie].FName,

        "ProjectMoney":
result.recordset[ie].ProjectMoney,

        "TID":                    result.recordset[ie].TID

    });

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    console.log(tableArrayDelete);

    res.render('deleteRecord', {tableArrayDelete : tableArrayDelete});

    res.end();

```



```

});

});

app.post('/deleteRecord', function(req,res){

    var deleteRec = new sql.Request();

    deleteRec.input('TeamDetailsID', sql.INT, TDFD);

    deleteRec.execute('DeleteTeamDetails').then(function(result,
recordset){

        console.log(result, recordset);

    });

    res.redirect('ScheduledTeamsView&Edit');

});

//view all teams

app.get('/allTeams', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamViewArray = [];

    var check5 = new sql.Request();

    check5.execute('ViewTeams').then(function(result){

        var len = result.recordset.length;

        var c;

        for(var i = 0; i < len; i++){

            c=i;

```

```

        teamViewArray.push({
            "TeamID":      result.recordset[i].TeamID,
            "TeamName":    result.recordset[i].TeamName,
            "ChurchName":  result.recordset[i].ChurchName,
            "District":    result.recordset[i].District,
            "LeaderEmail": result.recordset[i].LeaderEmail,
            "LeaderName":  result.recordset[i].LeaderName,
        });
    };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

console.log(teamViewArray);

defaults[0] = result.recordset[c].TeamName;
defaults[1] = result.recordset[c].ChurchName;
defaults[2] = result.recordset[c].District;
defaults[3] = result.recordset[c].LeaderEmail;
defaults[4] = result.recordset[c].LeaderName;

ids[0] = result.recordset[c].TeamID;

console.log(ids[0]);

res.render('allTeams', {teamViewArray: teamViewArray});

res.end();

});

});

```

```

app.post('/allTeams', function(req,res){
    indexE = req.body.RowForEditTeam;
    indexD = req.body.RowForDeleteTeam;
    indexV = req.body.RowForViewMembers;
    console.log(indexE);
    console.log(indexD);
    if(indexD == undefined && indexE == undefined){
        res.redirect('viewTeamMembers');
    }
    else if(indexD == undefined && indexV == undefined) {
        res.redirect('editTeam');
    }
    else{
        res.redirect('deleteTeam');
    }
});

app.get('/editTeam', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
}

```

```

var teamViewArrayEdit = [];

var check4 = new sql.Request();

check4.execute('ViewTeams').then(function(result){

    var ie = indexE;

        teamViewArrayEdit.push({

            "TeamID":      result.recordset[ie].TeamID,

            "TeamName":    result.recordset[ie].TeamName,

            "ChurchName":  result.recordset[ie].ChurchName,

            "District":    result.recordset[ie].District,

            "LeaderEmail": result.recordset[ie].LeaderEmail,

            "LeaderName":  result.recordset[ie].LeaderName

        });

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

console.log(teamViewArrayEdit);

ids[0] = result.recordset[ie].TeamID;

console.log(ids[0]);

res.render('editTeam', {teamViewArrayEdit : teamViewArrayEdit});

res.end();

});

});

//

```

```

app.post('/edit-Team', function(req,res){

    var NewTeamID = ids[0]; //PK

    var NewTeamName = req.body.newTeamName;

        if(NewTeamName == null || NewTeamName == undefined ||
NewTeamName.length === 0){

            NewTeamName = defaults[0];

        }

    var NewChurchName = req.body.newChurchName;

        if(NewChurchName == null || NewChurchName == undefined ||
NewChurchName.length === 0){

            NewChurchName = defaults[1];

        }

    var NewDistrict = req.body.newDistrict;

        if(NewDistrict == null || NewDistrict == undefined ||
NewDistrict.length === 0){

            NewDistrict = defaults[2];

        }

    var NewLeaderEmail = req.body.newLeaderEmail;

        if(NewLeaderEmail == null || NewLeaderEmail == undefined ||
NewLeaderEmail.length === 0){

            NewLeaderEmail = defaults[3];

        }

    var NewLeaderName = req.body.newLeaderName;

        if(NewLeaderName == null || NewLeaderName == undefined ||
NewLeaderName.length === 0){

            NewLeaderName = defaults[4];

```

```

    }

    var NewCancelTeam = req.body.newCancelTeam

    if(NewCancelTeam == null || NewCancelTeam == undefined ||
NewCancelTeam.length === 0){

        CancelTeam = 0;

    }

    var recordTeams = new sql.Request();

    recordTeams.input('NewTeamID', sql.INT, NewTeamID);

    recordTeams.input('NewTeamName', sql.NVARCHAR(255), NewTeamName);

    recordTeams.input('NewChurchName', sql.NVARCHAR(255),
NewChurchName);

    recordTeams.input('NewDistrict', sql.NVARCHAR(255), NewDistrict);

    recordTeams.input('NewLeaderEmail', sql.NVARCHAR(255),
NewLeaderEmail);

    recordTeams.input('NewLeaderName', sql.NVARCHAR(255),
NewLeaderName);

    recordTeams.input('NewCancelled', sql.INT, NewCancelTeam);

    recordTeams.execute('EditTeam').then(function(err){

        console.log(err);

    });

    res.redirect('ScheduledTeamsView&Edit');

});

```

```

app.get('/deleteTeam', function(req, res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
    var teamViewArrayDelete = [];
    var check4 = new sql.Request();
    check4.execute('ViewTeams').then(function(result){
        var ie = indexD;
        teamViewArrayDelete.push({
            "TeamID":      result.recordset[ie].TeamID,
            "TeamName":    result.recordset[ie].TeamName,
            "ChurchName":  result.recordset[ie].ChurchName,
            "District":    result.recordset[ie].District,
            "LeaderEmail": result.recordset[ie].LeaderEmail,
            "LeaderName":  result.recordset[ie].LeaderName,
        });
    });
    //The above loop creates an array of objects...
    //...which is easily rendered on the client side upon redirect and
    rendering
    console.log(teamViewArrayDelete);
    ids[0] = result.recordset[ie].TeamID;
    console.log(ids[0]);
}

```

```

res.render('deleteTeam', {teamViewArrayDelete : teamViewArrayDelete});

res.end();

});

});

//

app.post('/Team-Delete', function(req,res){

    var NewTeamID = ids[0];

    var check = new sql.Request();

    check.input('NewTeamID', sql.INT, NewTeamID);

    check.execute('DeleteTeam');

    res.redirect('ScheduledTeamsView&Edit');

});

//FUNDS

app.get('/funds', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayFunds = [];

    var check5 = new sql.Request();

    check5.execute('ExpectedFunds').then(function(result){

```



```

    var len = result.recordset.length;

    for(var i = 0; i < len; i++){

        tableArrayFunds.push({

            "TeamDetailsID":
result.recordset[i].TeamDetailsID,

            "ProjectFundsRecieved":
result.recordset[i].ProjectFundsRecieved,

            "CancelTeam":
result.recordset[i].CancelTeam,

            "ProjectMoney":
result.recordset[i].ProjectMoney,

        });

    };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

console.log(tableArrayFunds);

res.render('funds', {tableArrayFunds: tableArrayFunds});

res.end();

});

});

app.post('/EF', function(req,res){

indexE = req.body.RowForEditEF;

console.log(indexE);

res.redirect('editEF');

```

```

});

app.get('/editEF', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayFundsEditor = [];

    // const promise4 = new Promise(function(resolve, reject) {

    var check7 = new sql.Request();

    check7.execute('ExpectedFunds').then(function(result){

        var i = indexE;

            tableArrayFundsEditor.push({

                "TeamDetailsID":
result.recordset[i].TeamDetailsID,

                "ProjectFundsReceived":
result.recordset[i].ProjectFundsReceived,

                "CancelTeam":
result.recordset[i].CancelTeam,

                "ProjectMoney":
result.recordset[i].ProjectMoney,

            });

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering

        console.log(tableArrayFundsEditor);

```

```

ids[0] = result.recordset[i].TeamDetailsID;
ids[1] = result.recordset[i].TeamID;
defaults[0] = result.recordset[i].ProjectFundsRecieved;
defaults[1] = result.recordset[i].CancelTeam;
defaults[2] = result.recordset[i].ProjectMoney;
defaults[3] = result.recordset[i].ArriveDate;
res.render('editEF', {tableArrayFundsEditor : tableArrayFundsEditor});
res.end();
});
});

app.post('/editEFs', function(req,res){
    var NewTeamDetailsID = ids[0]; //PK
    var NewProjectFundsRecieved = req.body.newProjectFundsRecieved;
    if(NewProjectFundsRecieved == null || NewProjectFundsRecieved ==
undefined || NewProjectFundsRecieved.length === 0){
        NewProjectFundsRecieved = defaults[0];
    }
    var NewCancelTeam = req.body.newCancelTeam;
    if(NewCancelTeam == null || NewCancelTeam == undefined ||
NewCancelTeam.length === 0){
        NewCancelTeam = defaults[1];
    }
    var NewProjectMoney = req.body.newProjectMoney;

```

```

        if(NewProjectMoney == null || NewProjectMoney == undefined ||
NewProjectMoney.length === 0){

            NewProjectMoney = defaults[2];

        }

        var NewArrivalDate = req.body.newArrivaltDate;

        if(NewArrivalDate == null || NewArrivalDate == undefined ||
NewArrivalDate.length === 0){

            NewArrivalDate = defaults[3];

        }

        console.log(NewTeamDetailsID);

        console.log(NewProjectFundsRecieved);

        console.log(NewCancelTeam);

        console.log(NewProjectMoney);

        var recordFunds = new sql.Request();

        recordFunds.input('NewTeamDetailsID', sql.INT,
NewTeamDetailsID); //PK

        recordFunds.input('NewProjectFundsRecieved', sql.INT,
NewProjectFundsRecieved);

        recordFunds.input('NewCancelTeam', sql.INT, NewCancelTeam);

        recordFunds.input('NewProjectMoney', sql.MONEY, NewProjectMoney);

        recordFunds.execute('EditFunds').then(function(result, recordset){

            console.log(result, recordset);

        });

        res.redirect('ScheduledTeamsView&Edit');

```

```

});

app.get('/cancelled', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayCancelledTeams = [];

    var check5 = new sql.Request();

    check5.execute('CancelledTeams').then(function(result){

        var len = result.recordset.length;

        var c;

        for(var i = 0; i < len; i++){

            tableArrayCancelledTeams.push({

                "TeamID":
result.recordset[i].TeamID,

                "TeamName":
result.recordset[i].TeamName,

                "ChurchName":
result.recordset[i].ChurchName,

                "District":
result.recordset[i].District,

                "LeaderEmail":
result.recordset[i].LeaderEmail,

```

```

                "LeaderName":
result.recordset[i].LeaderName,

                "CancelTeam":
result.recordset[i].CancelTeam,

            });

        };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

console.log(tableArrayCancelledTeams);

res.render('cancelled', {tableArrayCancelledTeams:
tableArrayCancelledTeams});

res.end();

});

});

app.post('/cancelledTeams', function(req,res){

    indexE = req.body.RowForEditCancelledTeam;

    indexD = req.body.RowForDeleteCancelledTeam;

    console.log(indexE);

    console.log(indexD);

    if(indexD == undefined){

        res.redirect('editCT');

    }

    else {

```

```

        res.redirect('deleteCT');
    }
});

app.get('/editCT', function(req,res){
    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamCTedit = [];

    var check4 = new sql.Request();

    check4.execute('CancelledTeams').then(function(result){
        var ie = indexE;

        teamCTedit.push({
            "TeamID":      result.recordset[ie].TeamID,
            "TeamName":    result.recordset[ie].TeamName,
            "ChurchName":  result.recordset[ie].ChurchName,
            "District":    result.recordset[ie].District,
            "LeaderEmail": result.recordset[ie].LeaderEmail,
            "LeaderName":  result.recordset[ie].LeaderName,
        });

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering
    });
});

```

```

console.log(teamCTedit);

ids[0] = result.recordset[ie].TeamID;

defaults[0] = result.recordset[ie].TeamName;

defaults[1] = result.recordset[ie].ChurchName;

defaults[2] = result.recordset[ie].District;

defaults[3] = result.recordset[ie].LeaderEmail;

defaults[4] = result.recordset[ie].LeaderName;

console.log(ids[0]);

res.render('editCT', {teamCTedit : teamCTedit});

res.end();

});

});

app.get('/deleteCT', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamCTdelete = [];

    var check4 = new sql.Request();

    check4.execute('CancelledTeams').then(function(result){

        var ie = indexD;

        teamCTdelete.push({

```



```

        "TeamID":      result.recordset[ie].TeamID,
        "TeamName":   result.recordset[ie].TeamName,
        "ChurchName": result.recordset[ie].ChurchName,
        "District":   result.recordset[ie].District,
        "LeaderEmail": result.recordset[ie].LeaderEmail,
        "LeaderName": result.recordset[ie].LeaderName,
    });

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    console.log(teamCTdelete);

    ids[0] = result.recordset[ie].TeamID;

    console.log(ids[0]);

    res.render('deleteCT', {teamCTdelete : teamCTdelete});

    res.end();

    });

});

app.post('/editCTteam', function(req,res){

    var NewTeamID = ids[0]; //PK

    var NewTeamName = req.body.newTeamName;

    if(NewTeamName == null || NewTeamName == undefined ||
    NewTeamName.length === 0){

        NewTeamName = defaults[0];
    }
}

```

```

    }

    var NewChurchName = req.body.newChurchName;

    if(NewChurchName == null || NewChurchName == undefined ||
NewChurchName.length === 0){

        NewChurchName = defaults[1];

    }

    var NewDistrict = req.body.newDistrict;

    if(NewDistrict == null || NewDistrict == undefined ||
NewDistrict.length === 0){

        NewDistrict = defaults[2];

    }

    var NewLeaderEmail = req.body.newLeaderEmail;

    if(NewLeaderEmail == null || NewLeaderEmail == undefined ||
NewLeaderEmail.length === 0){

        NewLeaderEmail = defaults[3];

    }

    var NewLeaderName = req.body.newLeaderName;

    if(NewLeaderName == null || NewLeaderName == undefined ||
NewLeaderName.length === 0){

        NewLeaderName = defaults[4];

    }

    var NewCancelTeam = req.body.newCancelTeam

    if(NewCancelTeam == null || NewCancelTeam == undefined ||
NewCancelTeam.length === 0){

        NewCancelTeam = 1;

    }

```

```

var recordTeams = new sql.Request();

recordTeams.input('NewTeamID', sql.INT, NewTeamID);

recordTeams.input('NewTeamName', sql.NVARCHAR(255), NewTeamName);

recordTeams.input('NewChurchName', sql.NVARCHAR(255),
NewChurchName);

recordTeams.input('NewDistrict', sql.NVARCHAR(255), NewDistrict);

recordTeams.input('NewLeaderEmail', sql.NVARCHAR(255),
NewLeaderEmail);

recordTeams.input('NewLeaderName', sql.NVARCHAR(255),
NewLeaderName);

recordTeams.input('NewCancelled', sql.INT, NewCancelTeam);

recordTeams.execute('editCTteam').then(function(err){

    console.log(err);

    delete ids;

    delete defaults;

});

delete ids;

delete defaults;

res.redirect('ScheduledTeamsView&Edit');

});

```

```

app.post('/deleteCTteam', function(req, res){

```

```

var NewTeamID = ids[0];

var check9 = new sql.Request();

check9.input('NewTeamID', sql.INT, NewTeamID);

check9.execute('DeleteTeam');

delete ids;

delete defaults;

res.redirect('ScheduledTeamsView&Edit');

});

app.get('/ppd', function(req, res){

var UsNm = req.session.user;

console.log(req.sessionID);

if(UsNm == null && sessionID == null){res.redirect('/')};

console.log(`logged in as: ${UsNm}`);

var tableArrayPBD = [];

var check7 = new sql.Request();

check7.execute('ProjectsByDistrict').then(function(result){

var len = result.recordset.length;

for(var i = 0; i < len; i++){

tableArrayPBD.push({

"DistrictName": result.recordset[i].DistrictName,

"ArriveDate": result.recordset[i].ArriveDate,

"CountryName": result.recordset[i].CountryName

```

```

        });
    };

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    res.render('ppd', {tableArrayPBD: tableArrayPBD});

    res.end();

});

});

////////////////////////////////////
/////

// NOW FOR THE REAL MEAT OF THE PROJECT

////////////////////////////////////
/////

app.get('/allProjects', function(req,res)
{////////////////////////////////////
////////////////////////////////////

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var ProjectsViewArray = [];

    var check = new sql.Request();

    check.execute('ViewProjects').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

```

```

        ProjectsViewArray.push({
            "ProjectID":
result.recordset[i].ProjectID,
            "TeamLeaderID":
result.recordset[i].TeamLeaderID,
            "WWCoorID":
result.recordset[i].WWCoorID,
            "TeamID":
result.recordset[i].TeamID,
            "LinkToWwsite":
result.recordset[i].LinkToWwsite,
            "ProjectTypeID":
result.recordset[i].ProjectTypeID,
            "DistrictID":
result.recordset[i].DistrictID,
            "CityName":
result.recordset[i].CityName,
            "ChurchName":
result.recordset[i].ChurchName,
            "PastorName":
result.recordset[i].PastorName,
            "PastorPhone":
result.recordset[i].PastorPhone,
            "ProjectDesc":
result.recordset[i].ProjectDesc,
            "ProjectPlans":
result.recordset[i].ProjectPlans,
            "TeamDetailsID":
result.recordset[i].TeamDetailsID,
            "ProjectCompleted":
result.recordset[i].ProjectCompleted,

```

```

                "CountryID":
result.recordset[i].CountryID,

                });

        };

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
rendering

        res.render('allProjects', {ProjectsViewArray: ProjectsViewArray});

        res.end();

    });

});

app.get('/joinTeam', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamJoinArray = [];

    var checkTJ = new sql.Request();

    checkTJ.execute('ViewTeams').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

            teamJoinArray.push({

                "TeamID":
result.recordset[i].TeamID,

```

```

                "TeamName":
result.recordset[i].TeamName,

                "ChurchName":
result.recordset[i].ChurchName,

                "District":
result.recordset[i].District,

                "LeaderEmail":
result.recordset[i].LeaderEmail,

                "LeaderName":
result.recordset[i].LeaderName

            });

        };

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering

        res.render('joinTeam', {teamJoinArray: teamJoinArray});

        res.end();

    });

});

app.post('/joinAteam', function(req,res){

var NTMFName = req.body.NTMFName;

if(NTMFName == null || NTMFName == undefined || NTMFName.length === 0){

    NTMFName = 'None Provided';

}

var NTMPhoneNumber = req.body.NTMPhoneNumber;

```



```

if(NTMPhoneNumber == null || NTMPhoneNumber == undefined ||
NTMPhoneNumber.length === 0){

    NTMPhoneNumber = 'None Provided';

}

var NTMCountry = req.body.NTMCountry;

if(NTMCountry == null || NTMCountry == undefined || NTMCountry.length
=== 0){

    NTMCountry = 'None Provided';

}

var NTMDistrict = req.body.NTMDistrict;

if(NTMDistrict == null || NTMDistrict == undefined || NTMDistrict.length
=== 0){

    NTMDistrict = 'None Provided';

}

var NTMPreviousProjects = req.body.NTMPreviousProjects;

if(NTMPreviousProjects == null || NTMPreviousProjects == undefined ||
NTMPreviousProjects.length === 0){

    NTMPreviousProjects = 0;

}

var NTMSecondLanguage = req.body.NTMSecondLanguage;

if(NTMSecondLanguage == null || NTMSecondLanguage == undefined ||
NTMSecondLanguage.length === 0){

    NTMSecondLanguage = 1;

}

var NTMSkillsID =req.body.NTMSkillsID;

```

```

if(NTMSkillsID == null || NTMSkillsID == undefined || NTMSkillsID.length
=== 0){

    NTMSkillsID = 0;

}

var NTMTeamMemberTypeID = req.body.NTMTeamMemberTypeID;

if(NTMTeamMemberTypeID == null || NTMTeamMemberTypeID == undefined ||
NTMTeamMemberTypeID.length === 0){

    NTMTeamMemberTypeID = 0

}

var NTMTeamLeader = req.body.NTMTeamLeader;

if(NTMTeamLeader == null || NTMTeamLeader == undefined ||
NTMTeamLeader.length === 0){

    NTMTeamLeader = 0;

}

var TeamID = req.body.ttj;

if(TeamID == null || TeamID == undefined || TeamID.length === 0){

    TeamID = 0;

}

var NTMLastTrip = req.body.NTMLastTrip;

if(NTMLastTrip == null || NTMLastTrip == undefined || NTMLastTrip.length
=== 0){

    NTMLastTrip = '09/09/9999';

}

var NTMMale = req.body.NTMMale;

if(NTMMale == null || NTMMale == undefined || NTMMale.length === 0){

```

```

        NTMMale = 0;
    }

    var NTMFemale = req.body.NTMFemale;

    if(NTMFemale == null || NTMFemale == undefined || NTMFemale.length ===
    0){

        NTMFemale = 0;

    }

    var NTMSpouseOnTeam = req.body.NTMSpouseOnTeam;

    if(NTMSpouseOnTeam == null || NTMSpouseOnTeam == undefined ||
    NTMSpouseOnTeam.length === 0){

        NTMSpouseOnTeam = 0;

    }

    var NTMPassPortExp = req.body.NTMPassPortExp;

    if(NTMPassPortExp == null || NTMPassPortExp == undefined ||
    NTMPassPortExp.length === 0){

        NTMPassPortExp = '09/09/9999';

    }

    var NTMFoodAllergies = req.body.NTMFoodAllergies;

    if(NTMFoodAllergies == null || NTMFoodAllergies == undefined ||
    NTMFoodAllergies.length === 0){

        NTMFoodAllergies = 'None Provided';

    }

    var NTMMedicalAllergies = req.body.NTMMedicalAllergies;

    if(NTMMedicalAllergies == null || NTMMedicalAllergies == undefined ||
    NTMMedicalAllergies.length === 0){

        NTMMedicalAllergies = 'None Provided';
    }

```

```

}

var NTMDPI = req.body.NTMDPI;

if(NTMDPI== null || NTMDPI == undefined || NTMDPI.length === 0){

    NTMDPI = 'None Provided';

}

var NTMEmail = req.body.NTMEmail;

if(NTMEmail == null || NTMEmail == undefined || NTMEmail.length === 0){

    NTMEmail = 'None Provided';

}

var jat = new sql.Request();

const promise = new Promise(function (resolve, reject) {

    jat.input('NTMFName', sql.NVARCHAR(255), NTMFName);

    jat.input('NTMPhoneNumber', sql.NVARCHAR(255), NTMPhoneNumber);

    jat.input('NTMCountry', sql.NVARCHAR(255), NTMCountry);

    jat.input('NTMDistrict', sql.NVARCHAR(255), NTMDistrict);

    jat.input('NTMPreviousProjects', sql.INT, NTMPreviousProjects);

    jat.input('NTMSecondLanguage', sql.NVARCHAR(50), NTMSecondLanguage);

    jat.input('NTMSkillsID', sql.INT, NTMSkillsID);

    jat.input('NTMTeamMemberTypeID', sql.INT, NTMTeamMemberTypeID);

    jat.input('NTMTeamLeader', sql.INT, NTMTeamLeader);

    jat.input('NTMTeamID', sql.INT, TeamID);

    jat.input('NTMLastTrip', sql.DATE, NTMLastTrip);

    jat.input('NTMMale', sql.INT, NTMMale);

```

```

jat.input('NTMFemale',sql.INT, NTMFemale);

jat.input('NTMSpouseOnTeam', sql.INT, NTMSpouseOnTeam);

jat.input('NTMPassPortExp', sql.DATE, NTMPassPortExp);

jat.input('NTMFoodAllergies', sql.NVARCHAR(255), NTMFoodAllergies);

jat.input('NTMMedicalAllergies', sql.NVARCHAR(255),
NTMMedicalAllergies);

jat.input('NTMDPI', sql.NVARCHAR(255), NTMDPI);

jat.input('NTMEmail', sql.NVARCHAR(255), NTMEmail);

jat.execute('AddTeamMember').then(function (result, recordset){

var flag = result.returnValue;

resolve(flag);

}).catch(function(error){

    console.log(error);

    sql.close();

    reject(error);

});

});

promise.then((verdict) => {

if (verdict === 1) {

    console.log("...[ASSIGNMENT SUCCESS]...");

    res.redirect('ScheduledTeamsView&Edit');

}

else{

    console.log("...[DUPLICATE FOUND. REDIRECTING]...");

```

```

        ids[0]=TeamID;

        ids[1]=NTMFName;

        ids[2]=NTMEmail;

        res.redirect('joinOnTDDuplicate');
    }

});

});

app.get('/joinOnTDDuplicate', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamJoinArrayDup = [];

    var check = new sql.Request();

    check.input('NTMTeamID', sql.INT, ids[0]);

    check.execute('AddTeamMemberOnMultiple').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

            teamJoinArrayDup.push({

                "TeamDetailsID":
result.recordset[i].TeamDetailsID,

                "ProjectId":          result.recordset[i].ProjectId,

```

```

                "NumTeamMemb":
result.recordset[i].NumTeamMemb

                });

        };

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering

        console.log(teamJoinArrayDup);

        res.render('joinOnTDDuplicate', {teamJoinArrayDup :
teamJoinArrayDup});

        res.end();

    });

});

app.post('/TIDAssign', function(req,res){

    var TID = req.body.TeamDetailsIDToAssign;

    var teamJoinArrayDup = [];

    var check = new sql.Request();

    check.input('NTMTeamID', sql.INT, ids[0]);

    check.execute('AddTeamMemberOnMultiple').then(function(result){

        var ie = TID;

            teamJoinArrayDup.push({

                "TeamDetailsID":
result.recordset[ie].TeamDetailsID,

                "ProjectId":
result.recordset[ie].ProjectId,

```

```

        "NumTeamMemb":
result.recordset[ie].NumTeamMemb

        });

var check20 = new sql.Request();

var TeID = result.recordset[ie].TeamDetailsID;

var PrID = result.recordset[ie].ProjectId;

check20.input('NTMName', sql.NVARCHAR(255), ids[1]);

check20.input('NTMEmail', sql.NVARCHAR(255), ids[2]);

check20.input('NTMTeamDetailsID', sql.INT, TeID);

check20.input('NTMProjectID', sql.INT, PrID);

check20.execute('AddTeamMemberOnMultipleWork').then(function(err){

    console.log(err);

});

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.redirect('/ScheduledTeamsView&Edit');

res.end();

});

});

app.get('/assignTeams', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

```



```

console.log(`logged in as: ${UsNm}`);

var teamViewAssign = [];

var check5 = new sql.Request();

check5.execute('ViewTeams').then(function(result){

    var len = result.recordset.length;

    var c;

    for(var i = 0; i < len; i++){

        c=i;

        teamViewAssign.push({

            "TeamID":      result.recordset[i].TeamID,

            "TeamName":    result.recordset[i].TeamName,

            "ChurchName":  result.recordset[i].ChurchName,

            "District":    result.recordset[i].District,

            "LeaderEmail": result.recordset[i].LeaderEmail,

            "LeaderName":  result.recordset[i].LeaderName

        });

    };

});

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.render('assignTeams', {teamViewAssign : teamViewAssign});

res.end();

});

});

```

```

app.post('/GetTeamIndex', function(req,res){
    indexA = req.body.RowForTeamToAssign;
    console.log(indexA);
    res.redirect('assignProject');
    res.end();
});

```

```

app.get('/assignProject', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
    var tableArrayAssignProject = [];
    var check5 = new sql.Request();
    check5.execute('ViewProjects').then(function(result){
        var len = result.recordset.length;

        for(var i = 0; i < len; i++){
            tableArrayAssignProject.push({
                "ProjectID":      result.recordset[i].ProjectID,
                "TeamID":         result.recordset[i].TeamID,
                "ChurchName":     result.recordset[i].ChurchName,

```

```

                "Country":         result.recordset[i].Country,
                "PastorName":      result.recordset[i].PastorName,
            });
        };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.render('assignProject', {tableArrayAssignProject:
tableArrayAssignProject});

res.end();

});

});

app.post('/GetProjectIndex', function(req, res){

    const promise1 = new Promise(function (resolve, reject) {

        indexP = req.body.RowForProjectToAssign;

        console.log(indexA);

        console.log(indexP);

        var teamID = indexA;

        var projectID = indexP;

        console.log("A AND P INDEX TEST BOI");

        var nsr = new sql.Request()

        nsr.input('teamID',teamID);

        nsr.input('projectID',projectID);

        console.log("EXEC AssignTeam" + " " + teamID + ", " + projectID +
        ""))
    });
}

```

```

        sql.query("EXEC AssignTeam" + " " + teamID + ", " + projectID +
        "").then(function(result){

            var flag = result.returnValue;

            resolve(flag);

        }).catch(function(error){

            console.log(error);

            sql.close();

            reject(error);

        });

    });

    promise1.then((flag) => {

        if (flag === 0) {

            res.redirect('AssignTeamOnDuplicate');

            //maybe?????

        }

        else{

            res.redirect('ScheduledTeamsView&Edit');

        }

    });

});

});

app.get('/AssignTeamOnDuplicate', function(req, res){

    var UsNm = req.session.user;

```

```

console.log(req.sessionID);

if(UsNm == null && sessionID == null){res.redirect('/')};

console.log(`logged in as: ${UsNm}`);

var teamViewAssignOpts = [];

var check5 = new sql.Request();

check5.execute('AssignTeam').then(function(result){

    var len = result.recordset.length;

    for(var i = 0; i < len; i++){

        teamViewAssignOpts.push({

            "TeamID":      result.recordset[i].TeamID,

            "TeamName":    result.recordset[i].TeamName,

            "ChurchName":  result.recordset[i].ChurchName,

            "District":    result.recordset[i].District,

            "LeaderEmail": result.recordset[i].LeaderEmail,

            "LeaderName":  result.recordset[i].LeaderName

        });

    };

});

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

teamViewAssignOpts=teamViewAssignOpts.filter(teamViewAssignOpts =>
teamViewAssignOpts.TeamID == indexA);

res.render('AssignTeamOnDuplicate', {teamViewAssignOpts:
teamViewAssignOpts});

res.end();

```

```

});

});

app.post('AssignTeamOnDuplicate', function(req, res){
    var NewTeamID = req.body.newTeamID;
    var TeamDetailsID = req.body.TeamDetailsID;
    var pa = new sql.Request();
    pa.input('TeamID', sql.Int, NewTeamID);
    pa.input('ProjectID', sql.Int, indexP);
    pa.input('TeamDetailsID', sql.Int, TeamDetailsID);
    pa.execute('AssignTeamOnDuplicate');
    res.redirect('/');
});

app.get('/createTeam', function(req, res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/');}
    console.log(`logged in as: ${UsNm}`);
    res.render('createTeam');
    res.end();
});

```

```

app.post('/create-Team', function(req,res){

    var NewTeamName = req.body.newTeamName;

    var NewChurchName = req.body.newChurchName;

    var NewDistrict = req.body.newDistrict;

    var NewLeaderEmail = req.body.newLeaderEmail;

    var NewLeaderName = req.body.newLeaderName;

    var ntq = new sql.Request();

    ntq.input('NewTeamName', sql.NVarChar(255), NewTeamName);

    ntq.input('NewChurchName', sql.NVarChar(255), NewChurchName);

    ntq.input('NewDistrict', sql.NVarChar(255), NewDistrict);

    ntq.input('NewLeaderEmail', sql.NVarChar(255), NewLeaderEmail);

    ntq.input('NewLeaderName', sql.NVarChar(255), NewLeaderName);

    ntq.execute('createNewTeam');

    res.redirect('/ScheduledTeamsView&Edit');

});

```

```

app.get('/createProject', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var check = new sql.Request();

    var WWcoordArray = [];

```

```

    check.query("SELECT * FROM WWCoordinator").then(function(result){
        var len = result.recordset.length;
        for(var i = 0; i < len; i++){
            WWcoordArray.push({
                "WWcoordID":    result.recordset[i].WWCoordID,
                "FullName":    result.recordset[i].FullName
            });
        }
        res.render('createProject', {WWcoordArray : WWcoordArray});
        res.end();
    });

});

app.post('/create-Project', function(req, res){
    var NewWWCoorID = req.body.newWWCoorID;
    var NewDateReceived = req.body.newDateReceived;
    var NewLinkToWWSite = req.body.newLinkToWWSite;
    var NewUserPermission = req.body.newUserPermission;
    var NewProjectTypeID = req.body.newProjectTypeID;
    var NewDistrictID = req.body.newDistrictID;
    var NewCityName = req.body.newCityName;
    var NewPastorName = req.body.newPastorName;
    var NewPastorPhone = req.body.newPastorPhone;

```



```
var NewLocationLink = req.body.newLocationLink;

var NewDistanceFromDistrictOff =
req.body.newDistanceFromDistrictOff;

var NewDistrictBudgetCurrent = req.body.newDistrictBudgetCurrent;

var NewDistrictBudgetLast = req.body.newDistrictBudgetLast;

var NewFEMCurrent = req.body.newFEMCurrent;

var NewFEMLast = req.body.newFEMLast;

var NewDateOrganized = req.body.newDateOrganized;

var NewMembersLast = req.body.newMembersLast;

var NewMembersCurrent = req.body.newMembersCurrent;

var NewAvgAttendanceLast = req.body.newAvgAttendanceLast;

var NewAvgAttendanceCurent = req.body.newAvgAttendanceCurent;

var NewProjectDesc = req.body.newProjectDesc;

var NewTempleCapacity = req.body.newTempleCapacity;

var NewRoofWidth = req.body.newRoofWidth;

var NewRoofLength = req.body.newRoofLength;

var NewBriefDescriptionConst = req.body.newBriefDescriptionConst;

var NewPropertyWidth = req.body.newPropertyWidth;

var NewPropertyLength = req.body.newPropertyLength;

var NewPhotos = req.body.newPhotos;

var NewDigital = req.body.newDigital;

var NewLotLayout = req.body.newLotLayout;

var NewProjectPlans = req.body.newProjectPlans;

var NewMoneyRaised = req.body.NewMoneyRaised;
```

```
var NewMoneyInvested = req.body.newMoneyInvested;

var NewMoney12Months = req.body.newMoney12Months;

var NewProjectNeed = req.body.newProjectNeed;

var NewTitleToProperty = req.body.newTitleToProperty;

var NewPropertyDeededToChurch = req.body.newPropertyDeededToChurch;

var NewBuildingPermitNeeded = req.body.newBuildingPermitNeeded;

var NewBuildingPermitApproved = req.body.newBuildingPermitApproved;

var NewPastorApproved = req.body.newPastorApproved;

var NewChurchBoardApproved = req.body.newChurchBoardApproved;

var NewTreasureApproved = req.body.newTreasureApproved;

var NewTreasureApprovedDate = req.body.NewTreasureApprovedDate;

var NewSuperApproval = req.body.newSuperApproval;

var NewSuperApprovalDate = req.body.newSuperApprovalDate;

var NewDistrictSecretaryApproval =
req.body.newDistrictSecretaryApproval;

var NewDistrictTreasureApproval =
req.body.newDistrictTreasureApproval;

var NewDistrictTreasureApprovalDate =
req.body.newDistrictTreasureApprovalDate;

var NewDistrictPriority = req.body.newDistrictPriority;

var NewTyTApproved = req.body.newTyTApproved;

var NewTyTPriority = req.body.newTyTPriority;

var NewProjectCompleted = req.body.newProjectCompleted;

var NewWWLink = req.body.newWWLink;

var NewPhotoLocation = req.body.newPhotoLocation;
```

```

var NewPermitStart = req.body.newPermitStart;

var NewPermitEnd = req.body.newPermitEnd;

var NewCountryID = req.body.newCountryID;

var npq = new sql.Request();

npq.input('NewWWCoorID', sql.Int, NewWWCoorID);

npq.input('NewDateRecieved', sql.Date, NewDateReceived);

npq.input('NewLinkToWwsite', sql.NVarChar(255), NewLinkToWwsite);

npq.input('NewUserPermission', sql.NVarChar(255),
NewUserPermission);

npq.input('NewProjectTypeID', sql.Int, NewProjectTypeID);

npq.input('NewDistrictID', sql.Int, NewDistrictID);

npq.input('NewCityName', sql.NVarChar(255), NewCityName);

npq.input('NewPastorName', sql.NVarChar(255), NewPastorName);

npq.input('NewPastorPhone', sql.NVarChar(255), NewPastorPhone);

npq.input('NewLocationLink', sql.NVarChar(255), NewLocationLink);

npq.input('NewDistanceFromDistrictOff', sql.NVarChar(255),
NewDistanceFromDistrictOff);

npq.input('NewDistrictBudgetCurrent', sql.Money,
NewDistrictBudgetCurrent);

npq.input('NewDistrictBudgetLast', sql.Money,
NewDistrictBudgetLast);

npq.input('NewFEMCurrent', sql.Money, NewFEMCurrent);

npq.input('NewFEMLast', sql.Money, NewFEMLast);

npq.input('NewDateOrganized', sql.Date, NewDateOrganized);

npq.input('NewMembersLast', sql.Int, NewMembersLast);

```

```

npq.input('NewMembersCurrent', sql.Int, NewMembersCurrent);

npq.input('NewAvgAttendanceLast', sql.Int, NewAvgAttendanceLast);

npq.input('NewAvgAttendanceCurent', sql.Int,
NewAvgAttendanceCurent);

npq.input('NewProjectDesc', sql.NVarChar(255), NewProjectDesc);

npq.input('NewTempleCapacity', sql.Int, NewTempleCapacity);

npq.input('NewRoofWidth', sql.Int, NewRoofWidth);

npq.input('NewRoofLength', sql.Int, NewRoofLength);

npq.input('NewBriefDescriptionConst', sql.NVarChar(255),
NewBriefDescriptionConst);

npq.input('NewPropertyWidth', sql.Int, NewPropertyWidth);

npq.input('NewPropertyLength', sql.Int, NewPropertyLength);

npq.input('NewPhotos', sql.Int, NewPhotos);

npq.input('NewDigital', sql.Int, NewDigital);

npq.input('NewLotLayout', sql.Int, NewLotLayout);

npq.input('NewProjectPlans', sql.Int, NewProjectPlans);

npq.input('NewMoneyRaised', sql.Money, NewMoneyRaised);

npq.input('NewMoneyInvested', sql.Money, NewMoneyInvested);

npq.input('NewMoney12Months', sql.Money, NewMoney12Months);

npq.input('NewProjectNeed', sql.NVarChar(255), NewProjectNeed);

npq.input('NewTitleToProperty', sql.Int, NewTitleToProperty);

npq.input('NewPropertyDeededToChurch', sql.Int,
NewPropertyDeededToChurch);

npq.input('NewBuildingPermitNeeded', sql.Int,
NewBuildingPermitNeeded);

```

```

    npq.input('NewBuildingPermitApproved', sql.Int,
NewBuildingPermitApproved);

    npq.input('NewPastorApproved', sql.Int, NewPastorApproved);

    npq.input('NewChurchBoardApproved', sql.Int,
NewChurchBoardApproved);

    npq.input('NewTreasureApproved', sql.Int, NewTreasureApproved);

    npq.input('NewTreasureApprovedDate', sql.Date,
NewTreasureApprovedDate);

    npq.input('NewSuperApproval', sql.Int, NewSuperApproval);

    npq.input('NewSuperApprovalDate', sql.Date, NewSuperApprovalDate);

    npq.input('NewDistrictSecretaryApproval', sql.Int,
NewDistrictSecretaryApproval);

    npq.input('NewDistrictTreasureApproval', sql.Int,
NewDistrictTreasureApproval);

    npq.input('NewDistrictTreasureApprovalDate', sql.Date,
NewDistrictTreasureApprovalDate);

    npq.input('NewDistrictPriority', sql.Int, NewDistrictPriority);

    npq.input('NewTyTApproved', sql.Int, NewTyTApproved);

    npq.input('NewTyTPriority', sql.Int, NewTyTPriority);

    npq.input('NewProjectCompleted', sql.Int, NewProjectCompleted);

    npq.input('NewWwLink', sql.NVarChar(255), NewWWLink);

    npq.input('NewPhotoLocation', sql.NVarChar(255), NewPhotoLocation);

    npq.input('NewPermitStart', sql.Date, NewPermitStart);

    npq.input('NewPermitEnd', sql.Date, NewPermitEnd);

    npq.input('NewCountryID', sql.Int, NewCountryID);

    npq.execute('createNewProject');

```

```

        res.redirect('/ScheduledTeamsView&Edit');

    });

    app.get('/viewTeamMembers', function(req, res){

        var viewMem = new sql.Request();

        var MembersArray = [];

        var TeamID = indexV;

        viewMem.input('TeamID', sql.Int, TeamID);

        viewMem.execute('ViewTeamMembers').then(function(result){

            var len = result.recordset.length;

            for(var i = 0; i < len; i++){

                if(result.recordset[i].TeamID != indexV){

                    console.log("searching...");

                }

                else{

                    MembersArray.push({

                        "TeamMemberID":    result.recordset[i].TeamMemberID,

                        "FName":    result.recordset[i].FName,

                        "PhoneNumber":    result.recordset[i].PhoneNumber,

                        "District":    result.recordset[i].District,

                        "Email":    result.recordset[i].Email,

                        "FoodAllergies":    result.recordset[i].FoodAllergies,

```

```

                "MedicalAllergies":
result.recordset[i].MedicalAllergies

                });

            };

        };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.render('viewTeamMembers', {MembersArray : MembersArray});

res.end();

});

});

app.post('/getMemberID', function(req,res){

    indexE = req.body.RowForMemberToEdit;

    indexD = req.body.RowForMemberToDelete;

    if(indexD == undefined){

        res.redirect('editTeamMember');

    }

    else{

        res.redirect('deleteTeamMember');

    }

});

```

```

app.get('/editTeamMember', function(req, res){
    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayEditMember = [];

    //const promise4 = new Promise(function(resolve, reject) {

    var check7 = new sql.Request();

    check7.input('TeamID', sql.Int, indexV);

    check7.execute('ViewTeamMembers').then(function(result){

        var ie = indexE;

            tableArrayEditMember.push({

                "TeamMemberID":
result.recordset[ie].TeamMemberID,

                "FName":
result.recordset[ie].FName,

                "PhoneNumber":
result.recordset[ie].PhoneNumber,

                "Country":
result.recordset[ie].Country,

                "District":
result.recordset[ie].District,

                "PreviousProjects":
result.recordset[ie].PreviousProjects,

                "SecondLanguage":
result.recordset[ie].SecondLanguage,

```



```

        "SkillsID":
result.recordset[ie].SkillsID,

        "TeamMemberTypeID":
result.recordset[ie].TeamMemberTypeID,

        "TeamLeader":
result.recordset[ie].TeamLeader,

        "TeamID":
result.recordset[ie].TeamID,

        "LastTrip":
result.recordset[ie].LastTrip,

        "Male":
result.recordset[ie].Male,

        "Female":
result.recordset[ie].Female,

        "SpouseOnTeam":
result.recordset[ie].SpouseOnTeam,

        "PassPortExp":
result.recordset[ie].PassPortExp,

        "FoodAllergies":
result.recordset[ie].FoodAllergies,

        "MedicalAllergies":
result.recordset[ie].MedicalAllergies,

        "DPI":
result.recordset[ie].DPI,

        "Email":
result.recordset[ie].Email
    });

    console.log(tableArrayEditMember);

    ids[0] = result.recordset[ie].TeamMemberID;
    ids[1] = result.recordset[ie].TeamID;
    defaults[0] = result.recordset[ie].FName;
    defaults[1] = result.recordset[ie].PhoneNumber;
    defaults[2] = result.recordset[ie].Country;

```

```

        defaults[3] = result.recordset[ie].District;
        defaults[5] = result.recordset[ie].PreviousProjects;
        defaults[6] = result.recordset[ie].SecondLanguage;
        defaults[7] = result.recordset[ie].SkillsID;
        defaults[8] = result.recordset[ie].TeamMemberTypeID;
        defaults[9] = result.recordset[ie].TeamLeader;
        defaults[10] = result.recordset[ie].TeamID;
        defaults[11] = result.recordset[ie].LastTrip;
        defaults[12] = result.recordset[ie].Male;
        defaults[13] = result.recordset[ie].Female;
        defaults[14] = result.recordset[ie].SpouseOnTeam;
        defaults[15] = result.recordset[ie].PassPortExp;
        defaults[16] = result.recordset[ie].FoodAllergies;
        defaults[17] = result.recordset[ie].MedicalAllergies;
        defaults[18] = result.recordset[ie].DPI;
        defaults[19] = result.recordset[ie].Email;

        res.render('editTeamMember', {tableArrayEditMember :
tableArrayEditMember});

    })

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
rendering

    //res.end();

```

```
});
```

```
app.post('/edit-Member', function(req,res){  
    var EMtid = ids[0];  
  
    TIFref = EMtid;  
  
    console.log(EMtid);  
  
    var EMFName = req.body.EMFName;  
  
    var EMPhoneNumber = req.body.EMPhoneNumber;  
  
    var EMCountry = req.body.EMCountry;  
  
    var EMDistrict = req.body.EMDistrict;  
  
    var EMPreviousProjects = req.body.EMPreviousProjects;  
  
    var EMSecondLanguage = req.body.EMSecondLanguage;  
  
    var EMSkillsID = req.body.EMSkillsID;  
  
    var EMTeamMemberTypeID = req.body.EMTeamMemberTypeID;  
  
    var EMTeamLeader = req.body.EMTeamLeader;  
  
    var EMTeamID = req.body.EMTeamID;  
  
    var EMLastTrip = req.body.EMLastTrip;  
  
    var EMMale = req.body.EMMale;  
  
    var EMFemale = req.body.EMFemale;  
  
    var EMSpouseOnTeam = req.body.EMSpouseOnTeam;  
  
    var EMPassPortExp = req.body.EMPassPortExp;
```

```

var EMFoodAllergies = req.body.EMFoodAllergies;

var EMMedicalAllergies = req.body.EMMedicalAllergies;

var EMdpi = req.body.EMDPI;

var EMEmail = req.body.EMEmail;

if(EMFName == null || EMFName == undefined || EMFName.length === 0){
    EMFName = defaults[0];
}

if(EMPhoneNumber == null || EMPhoneNumber == undefined ||
EMPhoneNumber.length === 0){

    EMPhoneNumber = defaults[1];
}

if(EMCountry == null || EMCountry == undefined || EMCountry.length
=== 0){

    EMCountry = defaults[2];
}

if(EMDistrict == null || EMDistrict == undefined ||
EMDistrict.length === 0){

    EMDistrict = defaults[3];
}

if(EMPreviousProjects == null || EMPreviousProjects == undefined ||
EMPreviousProjects.length === 0){

    EMPreviousProjects = defaults[5];
}

if(EMSecondLanguage == null || EMSecondLanguage == undefined ||
EMSecondLanguage.length === 0){

    EMSecondLanguage = defaults[6];
}

```

```

    }

    if(EMSkillsID == null || EMSkillsID == undefined ||
    EMSkillsID.length === 0){

        EMSkillsID = defaults[7];

    }

    if(EMTeamMemberTypeID == null || EMTeamMemberTypeID == undefined ||
    EMTeamMemberTypeID.length === 0){

        EMTeamMemberTypeID = defaults[8];

    }

    if(EMTeamLeader == null || EMTeamLeader == undefined ||
    EMTeamLeader.length === 0){

        EMTeamLeader = defaults[9];

    }

    if(EMTeamID == null || EMTeamID == undefined || EMTeamID.length ===
    0){

        EMTeamID = defaults[10];

    }

    if(EMLastTrip == null || EMLastTrip == undefined ||
    EMLastTrip.length === 0){

        EMLastTrip = defaults[11];

    }

    if(EMMale == null || EMMale == undefined || EMMale.length === 0){

        EMMale = defaults[12];

    }

    if(EMFemale == null || EMFemale == undefined || EMFemale.length ===
    0){

```

```

        EMFemale = defaults[13];
    }

    if(EMSpouseOnTeam == null || EMSpouseOnTeam == undefined ||
    EMSpouseOnTeam.length === 0){

        EMSpouseOnTeam = defaults[14];
    }

    if(EMPassPortExp == null || EMPassPortExp == undefined ||
    EMPassPortExp.length === 0){

        EMPassPortExp = defaults[15];
    }

    if(EMFoodAllergies == null || EMFoodAllergies == undefined ||
    EMFoodAllergies.length === 0){

        EMFoodAllergies = defaults[16];
    }

    if(EMMedicalAllergies == null || EMMedicalAllergies == undefined ||
    EMMedicalAllergies.length === 0){

        EMMedicalAllergies = defaults[17];
    }

    if(EMdpi == null || EMdpi == undefined || EMdpi.length === 0){

        EMdpi = defaults[18];
    }

    if(EMEmail == null || EMail == undefined || EMail.length === 0){

        EMail = defaults[19];
    }

```

```

var em = new sql.Request();

em.input('EMTID', sql.Int, EMtid);

em.input('EMFName', sql.NVarChar(255), EMFName);

em.input('EMPhoneNumber', sql.NVarChar(255), EMPhoneNumber);

em.input('EMCountry', sql.NVarChar(255), EMCountry);

em.input('EMDistrict', sql.NVarChar(255), EMDistrict);

em.input('EMPreviousProjects', sql.Int, EMPreviousProjects);

em.input('EMSecondLanguage', sql.NVarChar(50), EMSecondLanguage);

em.input('EMSkillsID', sql.Int, EMSkillsID);

em.input('EMTeamMemberTypeID', sql.Int, EMTeamMemberTypeID);

em.input('EMTeamLeader', sql.Int, EMTeamLeader);

em.input('EMTeamID', sql.Int, EMTeamID);

em.input('EMLastTrip', sql.Date, EMLastTrip);

em.input('EMMale', sql.Int, EMMale);

em.input('EMFemale', sql.Int, EMFemale);

em.input('EMSpouseOnTeam', sql.Int, EMSpouseOnTeam);

em.input('EMPassPortExp', sql.Date, EMPassPortExp);

em.input('EMFoodAllergies', sql.NVarChar(255), EMFoodAllergies);

em.input('EMMedicalAllergies', sql.NVarChar(255),
EMMedicalAllergies);

em.input('EMDPI', sql.NVarChar(255), EMdpi);

em.input('EMEmail', sql.NVarChar(255), EMEmail);

em.execute('EditTeamMember').then(function(err){

    console.log(err);

```

```

    });

    res.redirect('editMemberTeam');
});

app.get('/editMemberTeam', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var teamViewArray = [];

    var check5 = new sql.Request();

    check5.execute('ViewTeams').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

            c=i;

            teamViewArray.push({

                "TeamID":      result.recordset[i].TeamID,

                "TeamName":    result.recordset[i].TeamName,

                "ChurchName":  result.recordset[i].ChurchName,

                "District":    result.recordset[i].District,

                "LeaderEmail": result.recordset[i].LeaderEmail,

                "LeaderName":  result.recordset[i].LeaderName,

            });
        }
    });

```



```

        };

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.render('editMemberTeam', {teamViewArray: teamViewArray});

res.end();

});

});

app.post('/editMemberAssign', function(req,res){

    var TMid = TIFref;

    var TeamID = req.body.ttj;

    var Atm = new sql.Request();

    Atm.input('TMID', sql.Int, TMid);

    Atm.input('TeamID', sql.Int, TeamID);

    Atm.execute('EditMemberTeam').then(function(err){

        console.log(err);

    })

    res.redirect('ScheduledTeamsView&Edit');

});

app.get('/deleteTeamMember', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

```

```

if(UsNm == null && sessionID == null){res.redirect('/')};

console.log(`logged in as: ${UsNm}`);

var tableArrayDeleteMember = [];

//const promise4 = new Promise(function(resolve, reject) {

var check7 = new sql.Request();

check7.input('TeamID', sql.Int, indexV);

check7.execute('ViewTeamMembers').then(function(result){

    var ie = indexD;

        tableArrayDeleteMember.push({

            "TeamMemberID":
result.recordset[ie].TeamMemberID,

            "FName":
                result.recordset[ie].FName,

            "PhoneNumber":
result.recordset[ie].PhoneNumber,

            "Country":
result.recordset[ie].Country,

            "District":
result.recordset[ie].District,

            "PreviousProjects":
result.recordset[ie].PreviousProjects,

            "SecondLanguage":
result.recordset[ie].SecondLanguage,

            "SkillsID":
result.recordset[ie].SkillsID,

            "TeamMemberTypeID":
result.recordset[ie].TeamMemberTypeID,

            "TeamLeader":
result.recordset[ie].TeamLeader,

```

```

        "TeamID":                result.recordset[ie].TeamID,
        "LastTrip":              result.recordset[ie].LastTrip,
        "Male":                  result.recordset[ie].Male,
        "Female":                result.recordset[ie].Female,
        "SpouseOnTeam":         result.recordset[ie].SpouseOnTeam,
        "PassPortExp":          result.recordset[ie].PassPortExp,
        "FoodAllergies":         result.recordset[ie].FoodAllergies,
        "MedicalAllergies":      result.recordset[ie].MedicalAllergies,
        "DPI":                   result.recordset[ie].DPI,
        "Email":                 result.recordset[ie].Email
    });

    res.render('deleteTeamMember', {tableArrayDeleteMember :
tableArrayDeleteMember});

});

});

app.post('/delete-Member', function(req,res){

    var TMID = req.body.RowForMemberToDelete;

    var check = new sql.Request();

    check.input('NewTeamMemberID', sql.Int, TMID);

    check.execute('DeleteTeamMember').then(function(err){

```

```

        if(err){console.log(err);}
    });
    res.redirect('ScheduledTeamsView&Edit');
});

app.post('/getProjectID', function(req,res){
    indexE = req.body.RowForEditProject;
    indexD = req.body.RowForDeleteProject;
    if(indexD == undefined){
        res.redirect('editProject');
    }
    else{
        res.redirect('deleteProject');
    }
});

app.get('/editProject', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
    var tableArrayEditProject = [];

```

```

//const promise4 = new Promise(function(resolve, reject) {
var check = new sql.Request();
check.execute('ViewProjects').then(function(result){
    var ie = indexE;

        PIDref = result.recordset[ie].ProjectID;

        tableArrayEditProject.push({

            "ProjectID":          result.recordset[ie].ProjectID,

            "WWCoorID":
result.recordset[ie].WWCoorID,

            "DateRecieved":
result.recordset[ie].DateRecieved,

            "LinkToWwsite":
result.recordset[ie].LinkToWwsite,

            "UserPermission":
result.recordset[ie].UserPermission,

            "ProjectTypeID":
result.recordset[ie].ProjectTypeID,

            "DistrictID":        result.recordset[ie].DistrictID,

            "CityName":          result.recordset[ie].CityName,

            "PastorName":        result.recordset[ie].PastorName,

            "PastorPhone":
result.recordset[ie].PastorPhone,

            "LocationLink":
result.recordset[ie].LocationLink,

            "DistanceFromDistrictOff":
result.recordset[ie].DistanceFromDistrictOff,

            "DistrictBudgetCurrent":
result.recordset[ie].DistrictBudgetCurrent,

```

```
        "DistrictBudgetLast":
result.recordset[ie].DistrictBudgetLast,

        "FEMCurrent":
result.recordset[ie].FEMCurrent,

        "FEMLast":           result.recordset[ie].FEMLast,

        "DateOrganized":
result.recordset[ie].DateOrganized,

        "MembersLast":
result.recordset[ie].MembersLast,

        "MembersCurrent":
result.recordset[ie].MembersCurrent,

        "AvgAttendanceLast":
result.recordset[ie].AvgAttendanceLast,

        "AvgAttendanceCurent":
result.recordset[ie].AvgAttendanceCurent,

        "ProjectDesc":
result.recordset[ie].ProjectDesc,

        "TempleCapacity":
result.recordset[ie].TempleCapacity,

        "RoofWidth":
result.recordset[ie].RoofWidth,

        "RoofLength":
result.recordset[ie].RoofLength,

        "BriefDescriptionConst":
result.recordset[ie].BriefDescriptionConst,

        "PropertyWidth":
result.recordset[ie].PropertyWidth,

        "PropertyLength":
result.recordset[ie].PropertyLength,

        "Photos":           result.recordset[ie].Photos,

        "Digital":         result.recordset[ie].Digital,
```

```
        "LotLayout":                result.recordset[ie].LotLayout,

        "ProjectPlans":
result.recordset[ie].ProjectPlans,

        "MoneyRaised":
result.recordset[ie].MoneyRaised,

        "MoneyInvested":
result.recordset[ie].MoneyInvested,

        "Money12Months":
result.recordset[ie].Money12Months,

        "ProjectNeed":
result.recordset[ie].ProjectNeed,

        "TitleToProperty":
result.recordset[ie].TitleToProperty,

        "PropertyDeededToChurch":
result.recordset[ie].PropertyDeededToChurch,

        "BuildingPermitNeeded":
result.recordset[ie].BuildingPermitNeeded,

        "BuildingPermitApproved":
result.recordset[ie].BuildingPermitApproved,

        "PastorApproved":
result.recordset[ie].PastorApproved,

        "ChurchBoardApproved":
result.recordset[ie].ChurchBoardApproved,

        "TreasureApproved":
result.recordset[ie].TreasureApproved,

        "TreasureApprovedDate":
result.recordset[ie].TreasureApprovedDate,

        "SuperApproval":
result.recordset[ie].SuperApproval,

        "SuperApprovalDate":
result.recordset[ie].SuperApprovalDate,
```

```

        "DistrictSecretaryApproval":
result.recordset[ie].DistrictSecretaryApproval,

        "DistrictTreasureApproval":
result.recordset[ie].DistrictTreasureApproval,

        "DistrictTreasureApprovalDate":
result.recordset[ie].DistrictTreasureApprovalDate,

        "DistrictPriority":
result.recordset[ie].DistrictPriority,

        "TyApproved":
result.recordset[ie].TyApproved,

        "TyTPriority":      result.recordset[ie].TyTPriority,

        "ProjectCompleted":
result.recordset[ie].ProjectCompleted,

        "Wwlink":          result.recordset[ie].Wwlink,

        "PhotoLocation":
result.recordset[ie].PhotoLocation,

        "PermitStart":
result.recordset[ie].PermitStart,

        "PermitEnd":       result.recordset[ie].PermitEnd,

        "CountryID":
result.recordset[ie].CountryID,

        "ChurchID":
result.recordset[ie].ChurchID

    });

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    console.log(tableArrayEditProject);

    TIDref = result.recordset[ie].ProjectID;

```



```
ids[0] = result.recordset[ie].ProjectID;
defaults[0] = result.recordset[ie].WWCoorID;
defaults[1] = result.recordset[ie].DateRecieved;
defaults[2] = result.recordset[ie].LinkToWwsite;
defaults[3] = result.recordset[ie].UserPermission;
defaults[4] = result.recordset[ie].ProjectTypeID;
defaults[5] = result.recordset[ie].DistrictID;
defaults[6] = result.recordset[ie].CityName;
defaults[8] = result.recordset[ie].PastorName;
defaults[9] = result.recordset[ie].PastorPhone;
defaults[11] = result.recordset[ie].LocationLink;
defaults[12] = result.recordset[ie].DistanceFromDistrictOff;
defaults[13] = result.recordset[ie].DistrictBudgetCurrent;
defaults[14] = result.recordset[ie].DistrictBudgetLast;
defaults[15] = result.recordset[ie].FEMCurrent;
defaults[16] = result.recordset[ie].FEMLast;
defaults[17] = result.recordset[ie].DateOrganized;
defaults[18] = result.recordset[ie].MembersLast;
defaults[19] = result.recordset[ie].MembersCurrent;
defaults[20] = result.recordset[ie].AvgAttendanceLast;
defaults[21] = result.recordset[ie].AvgAttendanceCurent;
defaults[22] = result.recordset[ie].ProjectDesc;
defaults[23] = result.recordset[ie].TempleCapacity;
```

```
defaults[24] = result.recordset[ie].RoofWidth;
defaults[25] = result.recordset[ie].RoofLength;
defaults[26] = result.recordset[ie].BriefDescriptionConst;
defaults[27] = result.recordset[ie].PropertyWidth;
defaults[28] = result.recordset[ie].PropertyLength;
defaults[29] = result.recordset[ie].Photos;
defaults[30] = result.recordset[ie].Digital;
defaults[31] = result.recordset[ie].LotLayout;
defaults[32] = result.recordset[ie].ProjectPlans;
defaults[33] = result.recordset[ie].MoneyRaised;
defaults[34] = result.recordset[ie].MoneyInvested;
defaults[35] = result.recordset[ie].Money12Months;
defaults[36] = result.recordset[ie].ProjectNeed;
defaults[37] = result.recordset[ie].TitleToProperty;
defaults[38] = result.recordset[ie].PropertyDeededToChurch;
defaults[39] = result.recordset[ie].BuildingPermitNeeded;
defaults[40] = result.recordset[ie].BuildingPermitApproved;
defaults[41] = result.recordset[ie].PastorApproved;
defaults[42] = result.recordset[ie].ChurchBoardApproved;
defaults[43] = result.recordset[ie].TreasureApproved;
defaults[44] = result.recordset[ie].TreasureApprovedDate;
defaults[45] = result.recordset[ie].SuperApproval;
defaults[46] = result.recordset[ie].SuperApprovalDate;
```

```

defaults[47] = result.recordset[ie].DistrictSecretaryApproval;
defaults[48] = result.recordset[ie].DistrictTreasureApproval;
defaults[49] = result.recordset[ie].DistrictTreasureApprovalDate;
defaults[50] = result.recordset[ie].DistrictPriority;
defaults[51] = result.recordset[ie].TyTApproved;
defaults[52] = result.recordset[ie].TyTPriority;
defaults[53] = result.recordset[ie].ProjectCompleted;
defaults[54] = result.recordset[ie].Wwlink;
defaults[55] = result.recordset[ie].PhotoLocation;
defaults[56] = result.recordset[ie].PermitStart;
defaults[57] = result.recordset[ie].PermitEnd;
defaults[58] = result.recordset[ie].CountryID;
defaults[59] = result.recordset[ie].ChurchID;

res.render('editProject', {tableArrayEditProject :
tableArrayEditProject});

res.end();

});

});

```

```

app.post('/edit-Project', function(req,res){

var NewProjectID = TIDref;

var NewWWCoorID = req.body.newWWCoorID;

var NewDateRecieved = req.body.newDateReceived;

```

```
var NewLinkToWwsite = req.body.newLinkToWwsite;
var NewUserPermission = req.body.newUserPermission;
var NewProjectTypeID = req.body.newProjectTypeID;
var NewDistrictID = req.body.newDistrictID;
var NewCityName = req.body.newCityName;
var NewPastorName = req.body.newPastorName;
var NewPastorPhone = req.body.newPastorPhone;
var NewLocationLink = req.body.newLocationLink;
var NewDistanceFromDistrictOff = req.body.newDistanceFromDistrictOff;
var NewDistrictBudgetCurrent = req.body.newDistrictBudgetCurrent;
var NewDistrictBudgetLast = req.body.newDistrictBudgetLast;
var NewFEMCurrent = req.body.newFEMCurrent;
var NewFEMLast = req.body.newFEMLast;
var NewDateOrganized = req.body.newDateOrganized;
var NewMembersLast = req.body.newMembersLast;
var NewMembersCurrent = req.body.newMembersCurrent;
var NewAvgAttendanceLast = req.body.newAvgAttendanceLast;
var NewAvgAttendanceCurent = req.body.newAvgAttendanceCurrent;
var NewProjectDesc = req.body.newProjectDesc;
var NewTempleCapacity = req.body.newTempleCapacity;
var NewRoofWidth = req.body.newRoofWidth;
var NewRoofLength = req.body.newRoofLength;
var NewBriefDescriptionConst = req.body.newBriefDescriptionConst;
```

```
var NewPropertyWidth = req.body.newPropertyWidth;
var NewPropertyLength = req.body.newPropertyLength;
var NewPhotos = req.body.newPhotos;
var NewDigital = req.body.newDigital;
var NewLotLayout = req.body.newLotLayout;
var NewProjectPlans = req.body.newProjectPlans;
var NewMoneyRaised = req.body.NewMoneyRaised;
var NewMoneyInvested = req.body.newMoneyInvested;
var NewMoney12Months = req.body.newMoney12Months;
var NewProjectNeed = req.body.newProjectNeed;
var NewTitleToProperty = req.body.newTitleToProperty;
var NewPropertyDeededToChurch = req.body.newPropertyDeededToChurch;
var NewBuildingPermitNeeded = req.body.newBuildingPermitNeeded;
var NewBuildingPermitApproved = req.body.newBuildingPermitApproved;
var NewPastorApproved = req.body.newPastorApproved;
var NewChurchBoardApproved = req.body.newChurchBoardApproved;
var NewTreasureApproved = req.body.newTreasureApproved;
var NewTreasureApprovedDate = req.body.NewTreasureApprovedDate;
var NewSuperApproval = req.body.newSuperApproval;
var NewSuperApprovalDate = req.body.newSuperApprovalDate;

var NewDistrictSecretaryApproval =
req.body.newDistrictSecretaryApproval;

var NewDistrictTreasureApproval =
req.body.newDistrictTreasureApproval;
```

```

    var NewDistrictTreasureApprovalDate =
req.body.newDistrictTreasureApprovalDate;

    var NewDistrictPriority = req.body.newDistrictPriority;

    var NewTyTapproved = req.body.newTyTapproved;

    var NewTyTPriority = req.body.newTyTPriority;

    var NewProjectCompleted = req.body.newProjectCompleted;

    var NewWWLink = req.body.newWWLink;

    var NewPhotoLocation = req.body.newPhotoLocation;

    var NewPermitStart = req.body.newPermitStart;

    var NewPermitEnd = req.body.newPermitEnd;

    var NewCountryID = req.body.newCountryID;

    var NewChurchID = req.body.newChurchID;

    if(NewProjectID == null || NewProjectID == undefined ||
NewProjectID.length === 0){

        NewProjectID = defaults[0];

    }

    if(NewWWCoorID == null || NewWWCoorID == undefined ||
NewWWCoorID.length === 0){

        NewWWCoorID = defaults[1];

    }

    if(NewDateRecieved == null || NewDateRecieved == undefined ||
NewDateRecieved.length === 0){

        NewDateRecieved = defaults[2];

    }

    if(NewLinkToWwsite == null || NewLinkToWwsite == undefined ||
NewLinkToWwsite.length === 0){

```

```

        NewLinkToWwsite = defaults[3];
    }

    if(NewUserPermission == null || NewUserPermission == undefined ||
NewUserPermission.length === 0){

        NewUserPermission = defaults[4];
    }

    if(NewProjectTypeID == null || NewProjectTypeID == undefined ||
NewProjectTypeID.length === 0){

        NewProjectTypeID = defaults[5];
    }

    if(NewDistrictID == null || NewDistrictID == undefined ||
NewDistrictID.length === 0){

        NewDistrictID = defaults[6];
    }

    if(NewCityName == null || NewCityName == undefined ||
NewCityName.length === 0){

        NewCityName = defaults[7];
    }

    if(NewPastorName == null || NewPastorName == undefined ||
NewPastorName.length === 0){

        NewPastorName = defaults[9];
    }

    if(NewPastorPhone == null || NewPastorPhone == undefined ||
NewPastorPhone.length === 0){

        NewPastorPhone = defaults[10];
    }

```

```

    if(NewLocationLink == null || NewLocationLink == undefined ||
NewLocationLink.length === 0){

        NewLocationLink = defaults[12];

    }

    if(NewDistanceFromDistrictOff == null || NewDistanceFromDistrictOff
== undefined || NewDistanceFromDistrictOff.length === 0){

        NewDistanceFromDistrictOff = defaults[13];

    }

    if(NewDistrictBudgetCurrent == null || NewDistrictBudgetCurrent ==
undefined || NewDistrictBudgetCurrent.length === 0){

        NewDistrictBudgetCurrent = defaults[14];

    }

    if(NewDistrictBudgetLast == null || NewDistrictBudgetLast == undefined
|| NewDistrictBudgetLast.length === 0){

        NewDistrictBudgetLast = defaults[15];

    }

    if(NewFEMCurrent == null || NewFEMCurrent == undefined ||
NewFEMCurrent.length === 0){

        NewFEMCurrent = defaults[16];

    }

    if(NewFEMLast == null || NewFEMLast == undefined || NewFEMLast.length
=== 0){

        NewFEMLast = defaults[17];

    }

    if(NewDateOrganized == null || NewDateOrganized == undefined ||
NewDateOrganized.length === 0){

        NewDateOrganized = defaults[18];

```



```

}

if(NewMembersLast == null || NewMembersLast == undefined ||
NewMembersLast.length === 0){

    NewMembersLast = defaults[19];

}

if(NewMembersCurrent == null || NewMembersCurrent == undefined ||
NewMembersCurrent.length === 0){

    NewMembersCurrent = defaults[20];

}

if(NewAvgAttendanceLast == null || NewAvgAttendanceLast == undefined
|| NewAvgAttendanceLast.length === 0){

    NewAvgAttendanceLast = defaults[21];

}

if(NewAvgAttendanceCurent == null || NewAvgAttendanceCurent ==
undefined || NewAvgAttendanceCurent.length === 0){

    NewAvgAttendanceCurent = defaults[22];

}

if(NewProjectDesc == null || NewProjectDesc == undefined ||
NewProjectDesc.length === 0){

    NewProjectDesc = defaults[23];

}

if(NewTempleCapacity == null || NewTempleCapacity == undefined ||
NewTempleCapacity.length === 0){

    NewTempleCapacity = defaults[24];

}

if(NewRoofWidth == null || NewRoofWidth == undefined ||
NewRoofWidth.length === 0){

```

```

        NewRoofWidth = defaults[25];
    }

    if(NewRoofLength == null || NewRoofLength == undefined ||
NewRoofLength.length === 0){

        NewRoofLength = defaults[26];

    }

    if(NewBriefDescriptionConst == null || NewBriefDescriptionConst ==
undefined || NewBriefDescriptionConst.length === 0){

        NewBriefDescriptionConst = defaults[27];

    }

    if(NewPropertyWidth == null || NewPropertyWidth == undefined ||
NewPropertyWidth.length === 0){

        NewPropertyWidth = defaults[28];

    }

    if(NewPropertyLength == null || NewPropertyLength == undefined ||
NewPropertyLength.length === 0){

        NewPropertyLength = defaults[29];

    }

    if(NewPhotos == null || NewPhotos == undefined || NewPhotos.length ===
0){

        NewPhotos = defaults[30];

    }

    if(NewDigital == null || NewDigital == undefined || NewDigital.length
=== 0){

        NewDigital = defaults[31];

    }

```

```

    if(NewLotLayout == null || NewLotLayout == undefined ||
NewLotLayout.length === 0){

        NewLotLayout = defaults[32];

    }

    if(NewProjectPlans == null || NewProjectPlans == undefined ||
NewProjectPlans.length === 0){

        NewProjectPlans = defaults[33];

    }

    if(NewMoneyRaised == null || NewMoneyRaised == undefined ||
NewMoneyRaised.length === 0){

        NewMoneyRaised = defaults[34];

    }

    if(NewMoneyInvested == null || NewMoneyInvested == undefined ||
NewMoneyInvested.length === 0){

        NewMoneyInvested = defaults[35];

    }

    if(NewMoney12Months == null || NewMoney12Months == undefined ||
NewMoney12Months.length === 0){

        NewMoney12Months = defaults[36];

    }

    if(NewProjectNeed == null || NewProjectNeed == undefined ||
NewProjectNeed.length === 0){

        NewProjectNeed = defaults[37];

    }

    if(NewTitleToProperty == null || NewTitleToProperty == undefined ||
NewTitleToProperty.length === 0){

        NewTitleToProperty = defaults[38];

```

```

}

if(NewPropertyDeededToChurch == null || NewPropertyDeededToChurch ==
undefined || NewPropertyDeededToChurch.length === 0){

    NewPropertyDeededToChurch = defaults[39];

}

if(NewBuildingPermitNeeded == null || NewBuildingPermitNeeded ==
undefined || NewBuildingPermitNeeded.length === 0){

    NewBuildingPermitNeeded = defaults[40];

}

if(NewBuildingPermitApproved == null || NewBuildingPermitApproved ==
undefined || NewBuildingPermitApproved.length === 0){

    NewBuildingPermitApproved = defaults[41];

}

if(NewPastorApproved == null || NewPastorApproved == undefined ||
NewPastorApproved.length === 0){

    NewPastorApproved = defaults[42];

}

if(NewChurchBoardApproved == null || NewChurchBoardApproved ==
undefined || NewChurchBoardApproved.length === 0){

    NewChurchBoardApproved = defaults[43];

}

if(NewTreasureApproved == null || NewTreasureApproved == undefined ||
NewTreasureApproved.length === 0){

    NewTreasureApproved = defaults[44];

}

if(NewTreasureApprovedDate == null || NewTreasureApprovedDate ==
undefined || NewTreasureApprovedDate.length === 0){

```

```

        NewTreasureApprovedDate = defaults[45];
    }

    if(NewSuperApproval == null || NewSuperApproval == undefined ||
NewSuperApproval.length === 0){

        NewSuperApproval = defaults[46];
    }

    if(NewSuperApprovalDate == null || NewSuperApprovalDate == undefined
||NewSuperApprovalDate.length === 0){

        NewSuperApprovalDate = defaults[47];
    }

    if(NewDistrictSecretaryApproval == null ||
NewDistrictSecretaryApproval == undefined ||
NewDistrictSecretaryApproval.length === 0){

        NewDistrictSecretaryApproval = defaults[48];
    }

    if(NewDistrictTreasureApproval == null || NewDistrictTreasureApproval
== undefined ||NewDistrictTreasureApproval.length === 0){

        NewDistrictTreasureApproval = defaults[49];
    }

    if(NewDistrictTreasureApprovalDate == null ||
NewDistrictTreasureApprovalDate == undefined ||
NewDistrictTreasureApprovalDate.length === 0){

        NewDistrictTreasureApprovalDate = defaults[50];
    }

    if(NewDistrictPriority == null || NewDistrictPriority == undefined ||
NewDistrictPriority.length === 0){

        NewDistrictPriority = defaults[51];
    }

```

```

    if(NewTyTapproved == null || NewTyTapproved == undefined ||
NewTyTapproved .length === 0){

        NewTyTapproved = defaults[52];

    }

    if(NewTyTPriority == null || NewTyTPriority == undefined ||
NewTyTPriority.length === 0){

        NewTyTPriority = defaults[53];

    }

    if(NewProjectCompleted == null || NewProjectCompleted == undefined ||
NewProjectCompleted.length === 0){

        NewProjectCompleted = defaults[54];

    }

    if(NewWWLink == null || NewWWLink == undefined ||NewWWLink.length ===
0){

        NewWWLink = defaults[55];

    }

    if(NewPhotoLocation == null || NewPhotoLocation == undefined ||
NewPhotoLocation.length === 0){

        NewPhotoLocation = defaults[56];

    }

    if(NewPermitStart == null || NewPermitStart == undefined ||
NewPermitStart.length === 0){

        NewPermitStart = defaults[57];

    }

    if(NewPermitEnd == null || NewPermitEnd == undefined ||
NewPermitEnd.length === 0){

        NewPermitEnd = defaults[58];

    }

```

```

    }

    if(NewCountryID == null || NewCountryID == undefined ||
NewCountryID.length === 0){

        NewCountryID = defaults[59];

    }

    if(NewChurchID == null || NewChurchID == undefined ||
NewChurchID.length === 0){

        NewChurchID = defaults[60];

    }

    console.log("PID REF ->>>>");

    console.log(PIDref);

    var pe = new sql.Request();

    pe.input('NewProjectID', sql.INT, NewProjectID);

    pe.input('NewWWCoorID', sql.INT, NewWWCoorID);

    pe.input('NewDateRecieved', sql.DATE, NewDateRecieved);

    pe.input('NewLinkToWwsite', sql.NVARCHAR(255), NewLinkToWwsite);

    pe.input('NewUserPermission', sql.NVARCHAR(255), NewUserPermission);

    pe.input('NewProjectTypeID', sql.INT, NewProjectTypeID);

    pe.input('NewDistrictID', sql.INT, NewDistrictID);

    pe.input('NewCityName', sql.NVARCHAR(255), NewCityName);

    pe.input('NewPastorName', sql.NVARCHAR(255), NewPastorName);

    pe.input('NewPastorPhone', sql.NVARCHAR(255), NewPastorPhone);

    pe.input('NewLocationLink', sql.NVARCHAR(255), NewLocationLink);

    pe.input('NewDistanceFromDistrictOff', sql.NVARCHAR(255),
NewDistanceFromDistrictOff);

```

```

    pe.input('NewDistrictBudgetCurrent', sql.MONEY,
NewDistrictBudgetCurrent);

    pe.input('NewDistrictBudgetLast', sql.MONEY, NewDistrictBudgetLast);

    pe.input('NewFEMCurrent', sql.MONEY, NewFEMCurrent);

    pe.input('NewFEMLast', sql.MONEY, NewFEMLast);

    pe.input('NewDateOrganized', sql.DATE, NewDateOrganized);

    pe.input('NewMembersLast', sql.INT, NewMembersLast);

    pe.input('NewMembersCurrent', sql.INT, NewMembersCurrent);

    pe.input('NewAvgAttendanceLast', sql.INT, NewAvgAttendanceLast);

    pe.input('NewAvgAttendanceCurent', sql.INT, NewAvgAttendanceCurent);

    pe.input('NewProjectDesc', sql.INT, NewProjectDesc);

    pe.input('NewTempleCapacity', sql.INT, NewTempleCapacity);

    pe.input('NewRoofWidth', sql.INT, NewRoofWidth);

    pe.input('NewRoofLength', sql.INT, NewRoofLength);

    pe.input('NewBriefDescriptionConst', sql.NVarChar(255),
NewBriefDescriptionConst);

    pe.input('NewPropertyWidth', sql.INT, NewPropertyWidth);

    pe.input('NewPropertyLength', sql.INT, NewPropertyLength);

    pe.input('NewPhotos', sql.INT, NewPhotos);

    pe.input('NewDigital', sql.INT, NewDigital);

    pe.input('NewLotLayout', sql.INT, NewLotLayout);

    pe.input('NewProjectPlans', sql.INT, NewProjectPlans);

    pe.input('NewMoneyRaised', sql.MONEY, NewMoneyRaised);

    pe.input('NewMoneyInvested', sql.MONEY, NewMoneyInvested);

```



```

pe.input('NewMoney12Months', sql.MONEY, NewMoney12Months);

pe.input('NewProjectNeed', sql.NVARCHAR(255), NewProjectNeed);

pe.input('NewTitleToProperty', sql.INT, NewTitleToProperty);

pe.input('NewPropertyDeededToChurch', sql.INT,
NewPropertyDeededToChurch);

pe.input('NewBuildingPermitNeeded', sql.INT, NewBuildingPermitNeeded);

pe.input('NewBuildingPermitApproved', sql.INT,
NewBuildingPermitApproved);

pe.input('NewPastorApproved', sql.INT, NewPastorApproved);

pe.input('NewChurchBoardApproved', sql.INT, NewChurchBoardApproved);

pe.input('NewTreasureApproved', sql.INT, NewTreasureApproved);

pe.input('NewTreasureApprovedDate', sql.DATE,
NewTreasureApprovedDate);

pe.input('NewSuperApproval', sql.INT, NewSuperApproval);

pe.input('NewSuperApprovalDate', sql.DATE, NewSuperApprovalDate);

pe.input('NewDistrictSecretaryApproval', sql.INT,
NewDistrictSecretaryApproval);

pe.input('NewDistrictTreasureApproval', sql.INT,
NewDistrictTreasureApproval);

pe.input('NewDistrictTreasureApprovalDate', sql.DATE,
NewDistrictTreasureApprovalDate);

pe.input('NewDistrictPriority', sql.INT, NewDistrictPriority);

pe.input('NewTyTapproved', sql.INT, NewTyTapproved);

pe.input('NewTyTPriority', sql.INT, NewTyTPriority);

pe.input('NewProjectCompleted', sql.INT, NewProjectCompleted);

pe.input('NewWwLink', sql.NVARCHAR(255), NewWWLink);

```

```

pe.input('NewPhotoLocation', sql.NVARCHAR(255), NewPhotoLocation);
pe.input('NewPermitStart', sql.DATE, NewPermitStart);
pe.input('NewPermitEnd', sql.DATE, NewPermitEnd);
pe.input('NewCountryID', sql.INT, NewCountryID);
pe.execute('EditProject').then(function(err){
    if(err){console.log(err);}
});
res.redirect('/ScheduledTeamsView&Edit');
});

```

```

app.get('/deleteProject', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    var check = new sql.Request();
    var projectViewArrayDelete = [];

    check.execute('ViewTeams').then(function(result){
        var ie = indexD;
        projectViewArrayDelete.push({
            "ProjectID":    result.recordset[ie].ProjectID,
            "TeamID":      result.recordset[ie].TeamID,

```

```

        "ChurchName":    result.recordset[ie].ChurchName,
        "BriefDescriptionConst":
result.recordset[ie].BriefDescriptionConst,
    });
});

console.log("redirect success");

res.render('deleteProject', {projectViewArrayDelete :
projectViewArrayDelete});
});

app.post('/delete-Project', function(req,res){
    var ProID = req.body.ProjectID;
    var PD = new sql.Request();
    PD.input('NewProjectID', sql.INT, ProID);
    PD.execute('DeleteProject').then(function(err){
        if(err){console.log(err)}
    });
    res.redirect('ScheduledTeamsView&EditTeam');
});

app.get('/allTeamDetails', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};

```

```

var check = new sql.Request();

var AllTeamDetails= [];

check.execute('ViewTeamDetails').then(function(result){

    var len = result.recordset.length;

    for(var i = 0; i < len; i++){

        AllTeamDetails.push({

            "TeamDetailsID":
result.recordset[i].TeamDetailsID,

            "TotalDailyFunds":
result.recordset[i].TotalDailyFunds,

            "ArriveFlightNumber":
result.recordset[i].ArriveFlightNumber,

            "ArriveDate":      result.recordset[i].ArriveDate,

            "DepartDate":      result.recordset[i].DepartDate,

            "DepartAirline":
result.recordset[i].DepartAirline,

            "DepartFlightNumber":
result.recordset[i].DepartFlightNumber,

            "ProjectID":      result.recordset[i].ProjectID,

            "TeamID":      result.recordset[i].TeamID,

            "DistanceToRR":
result.recordset[i].DistanceToRR,

            "DistanceToProject":
result.recordset[i].DistanceToProject,

            "HotelID":      result.recordset[i].HotelID,

            "CountryID":      result.recordset[i].CountryID,

            "AirportID":      result.recordset[i].AirportID,

```

```

        "TeamRegistration":
result.recordset[i].TeamRegistration,

        "Insurance":      result.recordset[i].Insurance,

        "Maxima":        result.recordset[i].Maxima,

        "ProjectMoney":
result.recordset[i].ProjectMoney,

        "PartnershipID":  result.recordset[i].PartnershipID,

        "TeamPartnerProfileID":
result.recordset[i].TeamPartnerProfileID,

        "TeamHost":      result.recordset[i].TeamHost,

        "CulBroker":     result.recordset[i].CulBroker,

        "TruckDriver":   result.recordset[i].TruckDriver,

        "BusDriver":     result.recordset[i].BusDriver,

        "Translator":    result.recordset[i].Translator,

        "StructureDelivery":
result.recordset[i].StructureDelivery,

        "CancelTeam":    result.recordset[i].CancelTeam,

        "ProjectFundsReceived":
result.recordset[i].ProjectFundsReceived

    });

};

console.log("redirect success");

res.render('allTeamDetails', {AllTeamDetails : AllTeamDetails});

});

app.post('/getTeamDetailsID', function(req, res){

    indexTIDE = req.body.RowForEditTeamDetails;

```

```

    indexTIDD = req.body.RowForDeleteTeamDetails;

    if(indexTIDD == undefined){
        res.redirect('editTeamDetails');
    }
    else{
        res.redirect('deleteTeamDetails');
    }

});

app.get('/editTeamDetails', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    var check = new sql.Request();

    var tableArrayEditTeamDetails = [];

    check.execute('ViewTeamDetails').then(function(result){

        var ie = indexTIDE;

        TIDref = result.recordset[ie].TeamDetailsID;

        tableArrayEditTeamDetails.push({

            "TeamDetailsID":
result.recordset[ie].TeamDetailsID,

            "TotalDailyFunds":
result.recordset[ie].TotalDailyFunds,

```

```

        "ArriveFlightNumber":
result.recordset[ie].ArriveFlightNumber,

        "ArriveDate":
result.recordset[ie].ArriveDate,

        "DepartDate":
result.recordset[ie].DepartDate,

        "DepartAirline":
result.recordset[ie].DepartAirline,

        "DepartFlightNumber":
result.recordset[ie].DepartFlightNumber,

        "ProjectID":      result.recordset[ie].ProjectID,

        "TeamID":        result.recordset[ie].TeamID,

        "DistanceToRR":
result.recordset[ie].DistanceToRR,

        "DistanceToProject":
result.recordset[ie].DistanceToProject,

        "HotelID":      result.recordset[ie].HotelID,

        "CountryID":    result.recordset[ie].CountryID,

        "AirportID":
result.recordset[ie].AirportID,

        "TeamRegistration":
result.recordset[ie].TeamRegistration,

        "Insurance":    result.recordset[ie].Insurance,

        "Maxima":      result.recordset[ie].Maxima,

        "ProjectMoney":
result.recordset[ie].ProjectMoney,

        "PartnershipID":
result.recordset[ie].PartnershipID,

        "TeamPartnerProfileID":
result.recordset[ie].TeamPartnerProfileID,

```

```

        "TeamHost":      result.recordset[ie].TeamHost,
        "CulBroker":    result.recordset[ie].CulBroker,
        "TruckDriver":  result.recordset[ie].TruckDriver,
        "BusDriver":    result.recordset[ie].BusDriver,
        "Translator":   result.recordset[ie].Translator,
        "StructureDelivery": result.recordset[ie].StructureDelivery,
        "CancelTeam":   result.recordset[ie].CancelTeam,
        "ProjectFundsReceived": result.recordset[ie].ProjectFundsReceived
    });
});

console.log("redirect success");

res.render('editTeamDetails', {tableArrayEditTeamDetails :
tableArrayEditTeamDetails});

});

});

app.post('/edit-TeamDetails', function(req,res){

    console.log(TIDref);

    console.log("TID REF CHECK");

    ids[0] = req.body.newTeamDetailsID;

    defaults[0] = req.body.newTotalDailyFunds;

    defaults[1] = req.body.newArriveFlightNumber;

    defaults[2] = req.body.newArriveDate;

```



```
defaults[3] = req.body.newDepartDate;
defaults[4] = req.body.newDepartAirline;
defaults[5] = req.body.newDepartFlightNumber;
defaults[6] = req.body.newDistanceToRR;
defaults[7] = req.body.newDistanceToProject;
defaults[8] = req.body.newTeamRegistration;
defaults[9] = req.body.newInsurance;
defaults[10] = req.body.newMaxima;
defaults[11] = req.body.newProjectMoney;
defaults[12] = req.body.newTeamHost;
defaults[13] = req.body.newCulBroker;
defaults[14] = req.body.newTruckDriver;
defaults[15] = req.body.newBusDriver;
defaults[16] = req.body.newTranslator;
defaults[17] = req.body.StructureDelivery;
defaults[18] = req.body.newCancelTeam;
defaults[19] = req.body.newProjectFundsReceived;

res.redirect('/chooseCountry');

res.end();

});

app.get('/deleteTeamDetails', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);
```

```

if(UsNm == null && sessionID == null){res.redirect('/')};

var check = new sql.Request();

var deleteTeamDetails = [];

check.execute('ViewTeamDetails').then(function(result){

    var ie = indexTIDD;

        deleteTeamDetails.push({

            "TeamDetailsID":
result.recordset[ie].TeamDetailsID,

            "TotalDailyFunds":
result.recordset[ie].TotalDailyFunds,

            "ArriveFlightNumber":
result.recordset[ie].ArriveFlightNumber,

            "ArriveDate":
result.recordset[ie].ArriveDate,

            "DepartDate":
result.recordset[ie].DepartDate,

            "DepartAirline":
result.recordset[ie].DepartAirline,

            "DepartFlightNumber":
result.recordset[ie].DepartFlightNumber,

            "ProjectID":      result.recordset[ie].ProjectID,

            "TeamID":        result.recordset[ie].TeamID,

            "DistanceToRR":
result.recordset[ie].DistanceToRR,

            "DistanceToProject":
result.recordset[ie].DistanceToProject,

            "HotelID":       result.recordset[ie].HotelID,

            "CountryID":     result.recordset[ie].CountryID,

```

```

        "AirportID":
result.recordset[ie].AirportID,

        "TeamRegistration":
result.recordset[ie].TeamRegistration,

        "Insurance":      result.recordset[ie].Insurance,

        "Maxima":      result.recordset[ie].Maxima,

        "ProjectMoney":
result.recordset[ie].ProjectMoney,

        "PartnershipID":
result.recordset[ie].PartnershipID,

        "TeamPartnerProfileID":
result.recordset[ie].TeamPartnerProfileID,

        "TeamHost":      result.recordset[ie].TeamHost,

        "CulBroker":
result.recordset[ie].CulBroker,

        "TruckDriver":   result.recordset[ie].TruckDriver,

        "BusDriver":     result.recordset[ie].BusDriver,

        "Translator":
result.recordset[ie].Translator,

        "StructureDelivery":
result.recordset[ie].StructureDelivery,

        "CancelTeam":   result.recordset[ie].CancelTeam,

        "ProjectFundsReceived":
result.recordset[ie].ProjectFundsReceived

    });

    console.log("redirect success");

    res.render('deleteTeamDetails', {deleteTeamDetails :
deleteTeamDetails});

});

```

```

});

app.post('/delete-TeamDetails', function(req,res){

    var TeamDetailsID = req.body.TDdelete;

    var check = new sql.Request();

    check.input('TeamDetailsID', sql.Int, TeamDetailsID)

    check.execute('DeleteTeamDetails').then(function(err){

        console.log(err);

    })

    res.redirect('ScheduledTeamsView&Edit');

})

app.get('/chooseCountry', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayChooseCountry = [];

    var check = new sql.Request();

    check.execute('ViewCountries').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

            tableArrayChooseCountry.push({

                "CountryID":      result.recordset[i].CountryID,

                "CountryName":

result.recordset[i].CountryName,

```

```

                "FieldID":        result.recordset[i].FieldID
            });
        };

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering

        res.render('chooseCountry', {tableArrayChooseCountry:
tableArrayChooseCountry});

        res.end();

    });
});

app.post('/edit-td-Country', function(req,res){

    var tdCountry = req.body.ctd;

    defaults[20] = tdCountry;

    res.redirect('chooseAirport');

});

app.get('/chooseAirport', function(req,res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayChooseAirport = [];

    var check5 = new sql.Request();

    check5.execute('ViewAirports').then(function(result){

```

```

    var len = result.recordset.length;

    for(var i = 0; i < len; i++){

        tableArrayChooseAirport.push({

            "AirportID":      result.recordset[i].AirportID,

            "AirportName":
result.recordset[i].AirportName,

            "AirportCode":   result.recordset[i].AirportCode,

            "City":          result.recordset[i].City,

            "Country":       result.recordset[i].Country

        });

    };

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    res.render('chooseAirport', {tableArrayChooseAirport:
tableArrayChooseAirport});

    res.end();

});

});

app.post('/edit-td-Airport', function(req,res){

    var airportID = req.body.tda

    defaults[21] = airportID;

    res.redirect('chooseHotel');

});

```

```

app.get('/meals', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var MealsViewArray = [];

    var check = new sql.Request();

    check.execute('ViewMeals').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

            MealsViewArray.push({

                "MealID":        result.recordset[i].MealID,

                "MealDesc":      result.recordset[i].MealDesc,

                "MealDateStart":
moment(result.recordset[i].MealDateStart).format('YYYY-MM-DD'),

                "MealDateStop":
moment(result.recordset[i].MealDateStop).format('YYYY-MM-DD'),

                "DaysForThisMeal":
Math.round((result.recordset[i].MealDateStart -
result.recordset.MealDateStop)/(1000*60*60*24)),

                "MealCost":      result.recordset[i].MealCost,

                "HowManyEating":
result.recordset[i].HowManyEating,

                "TeamDetailsID":
result.recordset[i].TeamDetailsID

            });
        }
    });

```

```

        };

        //The above loop creates an array of objects...

        //...which is easily rendered on the client side upon redirect and
        rendering

        res.render('meals', {MealsViewArray: MealsViewArray});

        res.end();

    });

});

app.post('/get-MealsID', function(req,res){

    indexD = req.body.MID;

    console.log(indexD);

    if(indexD == undefined){

        res.redirect('ScheduledTeamsView&Edit');

    }

    else{

        res.redirect('deleteMeals');

    }

});

app.get('/deleteMeals', function(req,res){

    var ie = indexD;

    var tableArrayDeleteMeal = [];

    var check = new sql.Reques();

    check.execute('ViewMeals').then(function(result){

        tableArrayDeleteMeal.push({

```



```

        "MealID":      result.recordset[ie].MealID,
        "MealDesc":   result.recordset[ie].MealDesc,
        "MealDateStart":
moment(result.recordset[ie].MealDateStart).format('YYYY-MM-DD'),
        "MealDateStop":
moment(result.recordset[ie].MealDateStop).format('YYYY-MM-DD'),
        "DaysForThisMeal":
(moment(result.recordset[ie].MealDateStart).format('YYYY-MM-
DD')).diff(moment(result.recordset[ie].MealDateStop).format('YYYY-MM-
DD'), 'days'),
        "MealCost":      result.recordset[ie].MealCost,
        "HowManyEating":  result.recordset[ie].HowManyEating,
        "TeamDetailsID":  result.recordset[ie].TeamDetailsID
    });
});

    res.render('deleteMeals', {tableArrayDeleteMeal:
tableArrayDeleteMeal});

    res.end();
});

app.post('/delete-Meals', function(req, res){

    var mealID = indexD;

    var check = new sql.Request();

    check.input('MealID', sql.Int, mealID).

    check.execute('DeleteMeal').then(function(err){

        console.log(err);

    });
});

```

```

        res.redirect('ScheduledTeamsView&Edit');
    });

app.get('/createMeals', function(req, res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
    var AllTeamDetailsForMeal = [];
    var check = new sql.Request();
    check.execute('ViewTeamDetails').then(function(result){
        var len = result.recordset.length;
        for(var i = 0; i < len; i++){
            AllTeamDetailsForMeal.push({
                "TeamDetailsID":
                result.recordset[i].TeamDetailsID,
                "ArriveDate":
                result.recordset[i].ArriveDate,
                "DepartDate":
                result.recordset[i].DepartDate,
                "ProjectID":
                result.recordset[i].ProjectID,
                "TeamID":      result.recordset[i].TeamID,
                "CountryID":   result.recordset[i].CountryID,
                "ProjectMoney":
                result.recordset[i].ProjectMoney,
            });
        }
    });
});

```

```

                "CancelTeam":    result.recordset[i].CancelTeam,
                "ProjectFundsReceived":
result.recordset[i].ProjectFundsReceived
            });
        };
        console.log("redirect success");
        res.render('createMeals', {AllTeamDetailsForMeal :
AllTeamDetailsForMeal});
    });
    });
    app.post('/create-Meals', function(req,res){
        var MealDesc = req.body.newMealDesc;
        var MealDateStart = req.body.newMealDateStart;
            var MealDateStart = moment(MealDateStart).format('YYYY-MM-
DD');
        var MealDateStop = req.body.newMealDateStop;
            var MealDateStop= moment(MealDateStop).format('YYYY-MM-DD');
        var MealCost = req.body.newMealCost;
        var HowManyEating = req.body.HowManyEating;
        var TeamDetailsID= req.body.tdm;
        console.log("SEE TID FOR MEAL INSERT BELOW:");
        console.log(TeamDetailsID);
        var check = new sql.Request();
        check.input('MealDesc', sql.NVarChar(255), MealDesc)
        check.input('MealDateStart', sql.Date, MealDateStart)

```

```

    check.input('MealDateStop', sql.Date, MealDateStop)
    check.input('MealCost', sql.Money, MealCost)
    check.input('HowManyEating', sql.Int, HowManyEating)
    check.input('TeamDetailsID', sql.Int, TeamDetailsID)
    check.execute('UpsertMeals').then(function(err){
        console.log(err)
    });
    res.redirect('ScheduledTeamsView&Edit');
});

app.get('/chooseHotel', function(req,res){
    var UsNm = req.session.user;
    console.log(req.sessionID);
    if(UsNm == null && sessionID == null){res.redirect('/')};
    console.log(`logged in as: ${UsNm}`);
    var tableArrayChooseHotel = [];
    var check5 = new sql.Request();
    check5.execute('ViewHotels').then(function(result){
        var len = result.recordset.length;
        for(var i = 0; i < len; i++){
            tableArrayChooseHotel.push({
                "HotelID":      result.recordset[i].HotelID,
                "HotelName":    result.recordset[i].HotelName,

```

```

        "City":    result.recordset[i].City,
        "PhoneNum":    result.recordset[i].PhoneNum,
    });
};

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

res.render('chooseHotel', {tableArrayChooseHotel:
                           tableArrayChooseHotel});

res.end();

});

});

app.post('/edit-td-Hotel', function(req,res){

    var HotelID = req.body.tdh

    defaults[22] = HotelID;

    if(defaults[0] == null || defaults[0] == undefined || defaults[0].length
    === 0){

        defaults[0] = 0.00

    }

    if(defaults[1] == null || defaults[1] == undefined || defaults[1].length
    === 0){

        defaults[1] = 'NONE PROVIDED'

    }

    if(defaults[2] == null || defaults[2] == undefined || defaults[2].length
    === 0){

```

```
        defaults[2] = '09/09/9999'
    }

    if(defaults[3] == null || defaults[3] == undefined || defaults[3].length
    === 0){

        defaults[3] = '09/09/9999'

    }

    if(defaults[4] == null || defaults[4] == undefined || defaults[4].length
    === 0){

        defaults[4] = 'NONE PROVIDED'

    }

    if(defaults[5] == null || defaults[5] == undefined || defaults[5].length
    === 0){

        defaults[5] = 'NONE PROVIDED'

    }

    if(defaults[6] == null || defaults[6] == undefined || defaults[6].length
    === 0){

        defaults[6] = 'NONE PROVIDED'

    }

    if(defaults[7] == null || defaults[7] == undefined || defaults[7].length
    === 0){

        defaults[7] = 'NONE PROVIDED'

    }

    if(defaults[8] == null || defaults[8] == undefined || defaults[8].length
    === 0){

        defaults[8] = 0

    }
}
```

```
if(defaults[9] == null || defaults[9] == undefined || defaults[9].length
=== 0){

    defaults[9] = 0

}

if(defaults[10] == null || defaults[10] == undefined ||
defaults[10].length === 0){

    defaults[10] = 0

}

if(defaults[11] == null || defaults[11] == undefined ||
defaults[11].length === 0){

    defaults[11] = 0

}

if(defaults[12] == null || defaults[12] == undefined ||
defaults[12].length === 0){

    defaults[12] = 0

}

if(defaults[13] == null || defaults[13] == undefined ||
defaults[13].length === 0){

    defaults[13] = 0

}

if(defaults[14] == null || defaults[14] == undefined ||
defaults[14].length === 0){

    defaults[14] = 0.00

}

if(defaults[15] == null || defaults[15] == undefined ||
defaults[15].length === 0){

    defaults[15] = 0

}
```

```

}

if(defaults[16] == null || defaults[16] == undefined ||
defaults[16].length === 0){

    defaults[16] = 0

}

if(defaults[17] == null || defaults[17] == undefined ||
defaults[17].length === 0){

    defaults[17] = 0

}

if(defaults[18] == null || defaults[18] == undefined ||
defaults[18].length === 0){

    defaults[18] = 0

}

if(defaults[19] == null || defaults[19] == undefined ||
defaults[19].length === 0){

    NewProjectID = 0

}

if(defaults[20] == null || defaults[20] == undefined ||
defaults[20].length === 0){

    defaults[20] = 0

}

if(defaults[21] == null || defaults[21] == undefined ||
defaults[21].length === 0){

    defaults[21] = 0

}

var check = new sql.Request();

```



```
check.input('TeamDetailsID', sql.Int, TIDref)
check.input('TotalDailyFunds', sql.Money, defaults[0])
check.input('ArriveFlightNumber', sql.NVarChar(255), defaults[1])
check.input('ArriveDate', sql.Date, defaults[2])
check.input('DepartDate', sql.Date, defaults[3])
check.input('DepartAirline', sql.NVarChar(255), defaults[4])
check.input('DepartFlightNumber', sql.NVarChar(255), defaults[5])
check.input('DistanceToRR', sql.Int, defaults[6])
check.input('DistanceToProject', sql.Int, defaults[7])
check.input('HotelID', sql.Int, defaults[22])
check.input('CountryID', sql.Int, defaults[20])
check.input('AirportID', sql.Int, defaults[21])
check.input('TeamRegistration', sql.Int, defaults[8])
check.input('Insurance', sql.Int, defaults[9])
check.input('Maxima', sql.Int, defaults[10])
check.input('ProjectMoney', sql.Money, defaults[11])
check.input('TeamHost', sql.Int, defaults[12])
check.input('CulBroker', sql.Int, defaults[13])
check.input('TruckDriver', sql.Int, defaults[14])
check.input('BusDriver', sql.Int, defaults[15])
check.input('Translator', sql.Int, defaults[16])
check.input('StructureDelivery', sql.Int, defaults[17])
check.input('CancelTeam', sql.Int, defaults[18])
```

```

check.input('ProjectFundsReceived', sql.Int, defaults[19])

check.execute('EditTeamDetails').then(function(err){
    console.log(err);
});

res.redirect('meals');
});

```

```

app.get('/newAirport', function(req,res){
    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var check = new sql.Request();

    var tableArrayChooseCountry = [];

    check.execute('ViewCountries').then(function(result){
        var len = result.recordset.length;

        for(var i = 0; i < len; i++){
            tableArrayChooseCountry.push({
                "CountryID":      result.recordset[i].CountryID,
                "CountryName":    result.recordset[i].CountryName,
                "FieldID":        result.recordset[i].FieldID
            });
        }
    });
}

```

```

        });
    });

//The above loop creates an array of objects...

//...which is easily rendered on the client side upon redirect and
rendering

    res.render('newAirport', {tableArrayChooseCountry :
tableArrayChooseCountry});

    res.end();

});

});

app.post('/new-Airport', function(req,res){

    var newAirportName = req.body.newAirportName;

    var newAirportCode = req.body.newAirportCode;

    var newAirportCity = req.body.newAirportCity;

    var newAirportCountryID = req.body.nac;

    var check = new sql.Request();

    check.input('AirportName', sql.NVarChar(255), newAirportName)

    check.input('AirportCode', sql.NVarChar(255), newAirportCode)

    check.input('City', sql.NVarChar(255), newAirportCity)

    check.input('CountryID', sql.Int, newAirportCountryID)

    check.execute('UpsertAirport').then(function(err){

        console.log(err);

    })

    res.redirect('ScheduledTeamsView&Edit');
}

```

```

});

app.get('/newCountry', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var tableArrayField = [];

    var check = new sql.Request();

    check.execute('GetAllField').then(function(result){

        var len = result.recordset.length

        for(var i = 0; i < len; i++){

            tableArrayField.push({

                "FieldID":      result.recordset[i].FieldID,

                "FieldName":    result.recordset[i].FieldName,

            });

        };

    });

    //The above loop creates an array of objects...

    //...which is easily rendered on the client side upon redirect and
    rendering

    res.render('newCountry', {tableArrayField: tableArrayField});

    res.end();

});

```

```

});

app.post('/new-Country', function(req, res){

    var countryName = req.body.newCountryName;

    var fieldID = req.body.ncf;

    var check = new sql.Request();

    check.input('CountryName', sql.NVarChar(255), countryName)

    check.input('FieldID', sql.Int, fieldID)

    check.execute('UpsertCountry').then(function(err){

        console.log(err);

    })

    res.redirect('ScheduledTeamsView&Edit');

});

```

```

app.get('/newHotel', function(req, res){

    var UsNm = req.session.user;

    console.log(req.sessionID);

    if(UsNm == null && sessionID == null){res.redirect('/')};

    console.log(`logged in as: ${UsNm}`);

    var check = new sql.Request();

    var tableArrayChooseCountry = [];

    check.execute('ViewCountries').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i < len; i++){

```

```

        tableArrayChooseCountry.push({
            "Country":      result.recordset[i].CountryID,
            "CountryName":  result.recordset[i].CountryName,
            "FieldID":      result.recordset[i].FieldID
        });
    };

    res.render('newHotel', {tableArrayChooseCountry :
        tableArrayChooseCountry});

    res.end();
});

});

app.post('/new-Hotel', function(req,res){

    var HotelName = req.body.newHotelName;

    var CountryID = req.body.hc;

    var RandR = req.body.RandR;

    var Comments = req.body.Comments;

    var PhoneNum = req.body.PhoneNum;

    var WiFi = req.body.WiFi;

    var AirCond = req.body.AirCond;

    var Pool = req.body.Pool;

    var MapLoc = req.body.MapLoc;

    var City = req.body.City;

    var LastUsed = req.body.LastUsed;

    var check = new sql.Request;

```

```

    check.input('HotelName', sql.NVarChar(255), HotelName)
    check.input('CountryID', sql.Int, CountryID)
    check.input('RandR', sql.Bit, RandR)
    check.input('Comments', sql.NVarChar(255), Comments)
    check.input('PhoneNum', sql.NVarChar(255), PhoneNum)
    check.input('WiFi', sql.Bit, WiFi)
    check.input('AirCond', sql.Bit, AirCond)
    check.input('Pool', sql.Bit, Pool)
    check.input('MapLoc', sql.NVarChar(255), MapLoc)
    check.input('City', sql.NVarChar(255), City)
    check.input('LastUsed', sql.Date, LastUsed)
    check.execute('UpsertHotel').then(function(err){
        console.log(err)
    })
    res.redirect('ScheduledTeamsView&Edit');
});

app.get('/viewCountries', function(req,res){
    var tableArrayAllCountries = [];
    var check = new sql.Request();
    check.query('SELECT * FROM Country').then(function(result){
        var len = result.recordset.length;
        for(var i = 0; i< len; i++){

```

```

        tableArrayAllCountries.push({
            "CountryID":    result.recordset[i].CountryID,
            "CountryName":  result.recordset[i].CountryName,
            "FieldID":      result.recordset[i].FieldID
        });
    };

    res.render('viewCountries',{tableArrayAllCountries :
tableArrayAllCountries});

    res.end();
})
});

app.get('/viewAirports', function(req,res){

    var tableArrayAllAirports = [];

    var check = new sql.Request();

    check.query('SELECT * FROM Airport').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i< len; i++){

            tableArrayAllAirports.push({

                "AirportID":    result.recordset[i].AirportID,

                "AirportName":  result.recordset[i].AirportName,

                "AirportCode":  result.recordset[i].AirportCode,

                "City":          result.recordset[i].City,
            });
        }
    });
});

```



```

                "Country":        result.recordset[i].Country
            });
        };

        res.render('viewAirports',{tableArrayAllAirports :
tableArrayAllAirports});

        res.end();

    })
});

app.get('/viewHotels', function(req,res){

    var tableArrayAllHotels = [];

    var check = new sql.Request();

    check.query('SELECT * FROM Hotel').then(function(result){

        var len = result.recordset.length;

        for(var i = 0; i< len; i++){

            tableArrayAllHotels.push({

                "HotelID":        result.recordset[i].HotelID,

                "HotelName":      result.recordset[i].HotelName,

                "City":           result.recordset[i].City,

                "PhoneNum":       result.recordset[i].PhoneNum

            });

        };

        res.render('viewHotels',{tableArrayAllHotels :
tableArrayAllHotels});

        res.end();
    });
});

```

```

    });
});

////////////////////////////////////
// NOW IT'S TIME TO MOP UP
////////////////////////////////////

app.get('/logout', function(req, res){
    req.session.destroy();
    req.session = null;
    res.redirect('/');
});

https.createServer({
    key: fs.readFileSync(__dirname + '/privateKey.key'),
    cert: fs.readFileSync(__dirname + '/certificate.crt')
}, app).listen(5000, function() { //***
    console.log("RUNNING!!! SECURED!!!"); //***
});

// server code from https://www.tutorialsteacher.com/nodejs/access-sql-
server-in-nodejs

// I contacted this site and recieved written permission to use server
script. Will show email upon request.

```

## C. Schema

```
USE [master]

GO

/***** Object: Database [mission2Go]    Script Date: 11/9/2020 6:28:00
PM *****/

CREATE DATABASE [mission2Go]

    CONTAINMENT = NONE

    ON PRIMARY

    ( NAME = N'mission2Go', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.TEAMTRACKING\MSSQL\DATA\mission2Go.mdf' , SIZE =
73728KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

    LOG ON

    ( NAME = N'mission2Go_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.TEAMTRACKING\MSSQL\DATA\mission2Go_log.ldf' , SIZE =
73728KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

    WITH CATALOG_COLLATION = DATABASE_DEFAULT

GO

ALTER DATABASE [mission2Go] SET COMPATIBILITY_LEVEL = 150

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [mission2Go].[dbo].[sp_fulltext_database] @action = 'enable'
end

GO

ALTER DATABASE [mission2Go] SET ANSI_NULL_DEFAULT OFF

GO
```

```
ALTER DATABASE [mission2Go] SET ANSI_NULLS OFF
GO
ALTER DATABASE [mission2Go] SET ANSI_PADDING OFF
GO
ALTER DATABASE [mission2Go] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [mission2Go] SET ARITHABORT OFF
GO
ALTER DATABASE [mission2Go] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [mission2Go] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [mission2Go] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [mission2Go] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [mission2Go] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [mission2Go] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [mission2Go] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [mission2Go] SET QUOTED_IDENTIFIER OFF
GO
```

```
ALTER DATABASE [mission2Go] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [mission2Go] SET DISABLE_BROKER
GO
ALTER DATABASE [mission2Go] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [mission2Go] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [mission2Go] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [mission2Go] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [mission2Go] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [mission2Go] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [mission2Go] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [mission2Go] SET RECOVERY SIMPLE
GO
ALTER DATABASE [mission2Go] SET MULTI_USER
GO
ALTER DATABASE [mission2Go] SET PAGE_VERIFY CHECKSUM
GO
```

```

ALTER DATABASE [mission2Go] SET DB_CHAINING OFF

GO

ALTER DATABASE [mission2Go] SET FILESTREAM( NON_TRANSACTED_ACCESS =
OFF )

GO

ALTER DATABASE [mission2Go] SET TARGET_RECOVERY_TIME = 60 SECONDS

GO

ALTER DATABASE [mission2Go] SET DELAYED_DURABILITY = DISABLED

GO

ALTER DATABASE [mission2Go] SET QUERY_STORE = OFF

GO

USE [mission2Go]

GO

/***** Object: UserDefinedFunction [dbo].[Authenticate]      Script
Date: 11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE FUNCTION [dbo].[Authenticate](@user nvarchar(40), @pass
nvarchar(40))

RETURNS INT

AS

          BEGIN

DECLARE @result INT

IF NOT EXISTS(

```

```

        (SELECT TOP(1)
        [userID],
        [userName],
        [userPassword],
        [userEmail],
        [userClearanceLevel]
        FROM userInfo
        WHERE
        userName = @user
        AND
        userPassword = @pass))
        SET @result = 0 --@FailString
ELSE
        SET @result = 1 --@SuccessString

        RETURN @result

END

GO

/***** Object: Table [dbo].[Airport]      Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

```

CREATE TABLE [dbo].[Airport](
    [AirportID] [int] IDENTITY(1,1) NOT NULL,
    [AirportName] [nvarchar](255) NULL,
    [AirportCode] [nvarchar](255) NULL,
    [City] [nvarchar](255) NULL,
    [CountryID] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [AirportID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[CheckSolicitud]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[CheckSolicitud](
    [CheckSolicitudID] [int] IDENTITY(1,1) NOT NULL,
    [SolicitudDate] [datetime2](7) NULL,
    [CheckName] [nvarchar](255) NULL,
    [CheckAmount] [money] NULL,
    [TextDollarAmount] [nvarchar](255) NULL,

```



```

[PurchaseDescription] [nvarchar](255) NULL,
[AccountingCode] [nvarchar](255) NULL,
[ShortDesc] [nvarchar](255) NULL,
[USD] [money] NULL,
[LocalCurrency] [money] NULL,
[ExchangeRate] [money] NULL,
[RequestedBy] [nvarchar](255) NULL,
[ReviewedBy] [nvarchar](255) NULL,
[ApprovedBy] [nvarchar](255) NULL,
[ProjectID] [int] NULL,

PRIMARY KEY CLUSTERED
(
    [CheckSolicitudID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO

/***** Object: Table [dbo].[ChurchName]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ChurchName](
    [ChurchID] [int] IDENTITY(1,1) NOT NULL,

```

```

        [ChurchName] [nvarchar](255) NULL,
        [DistrictID] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [ChurchID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[Country]    Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Country](
        [CountryID] [int] IDENTITY(1,1) NOT NULL,
        [CountryName] [nvarchar](255) NULL,
        [FieldID] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [CountryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO

/***** Object: Table [dbo].[CurrencyType]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[CurrencyType](
        [CurrencyTypeID] [int] IDENTITY(1,1) NOT NULL,
        [CurrencyIndicator] [nvarchar](255) NULL,
        [CurrencyName] [nvarchar](255) NULL,
        [CountryID] [int] NULL,

PRIMARY KEY CLUSTERED
(
        [CurrencyTypeID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[ExchangeRate]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ExchangeRate](

```

```

        [ExchangeRateID] [int] IDENTITY(1,1) NOT NULL,
        [LastUpdate] [datetime2](7) NULL,
        [ExchangeRate] [money] NULL,
        [CurrencyTypeID] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [ExchangeRateID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[Field]    Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Field](
        [FieldID] [int] IDENTITY(1,1) NOT NULL,
        [FieldName] [nvarchar](255) NULL,
        [RegionID] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [FieldID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Table [dbo].[Hotel]    Script Date: 11/9/2020 6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Hotel](
```

```
    [HotelID] [int] IDENTITY(1,1) NOT NULL,
```

```
    [HotelName] [nvarchar](255) NULL,
```

```
    [CountryID] [int] NULL,
```

```
    [R&R] [bit] NULL,
```

```
    [Comments] [nvarchar](max) NULL,
```

```
    [PhoneNum] [nvarchar](255) NULL,
```

```
    [WiFi] [bit] NULL,
```

```
    [AirCond] [bit] NULL,
```

```
    [Pool] [bit] NULL,
```

```
    [MapLoc] [nvarchar](255) NULL,
```

```
    [City] [nvarchar](255) NULL,
```

```
    [LastUsed] [datetime2](7) NULL,
```

```
PRIMARY KEY CLUSTERED
```

```
(
```

```

        [HotelID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

/***** Object: Table [dbo].[Language]      Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Language](
        [LanguageID] [int] IDENTITY(1,1) NOT NULL,
        [LanguageSpoken] [nvarchar](255) NULL,

PRIMARY KEY CLUSTERED
(
        [LanguageID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[LocalCoordinator]      Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

```

GO

```
CREATE TABLE [dbo].[LocalCoordinator](
    [LocalCoordID] [int] IDENTITY(1,1) NOT NULL,
    [CoordName] [nvarchar](255) NULL,
    [DistrictID] [int] NULL,
    [CoordPhone] [nvarchar](255) NULL,
    [CoordEmail] [nvarchar](255) NULL,
    PRIMARY KEY CLUSTERED
(
    [LocalCoordID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
/***** Object: Table [dbo].[Meals]    Script Date: 11/9/2020 6:28:00
PM *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[Meals](
    [MealID] [int] IDENTITY(1,1) NOT NULL,
    [MealDesc] [nvarchar](255) NULL,
    [DaysForThisMeal] [int] NULL,
    [MealDateStart] [datetime2](7) NULL,
```

```

        [MealDateStop] [datetime2](7) NULL,
        [MealCost] [money] NULL,
        [HowManyEating] [int] NULL,
        [TeamDetailsID] [int] NOT NULL,
CONSTRAINT [PK__Meals__ACF6A65DEB5000B6] PRIMARY KEY CLUSTERED
(
        [MealID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[Partnerships]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Partnerships](
        [PartnershipID] [int] IDENTITY(1,1) NOT NULL,
        [Vision] [nvarchar](255) NULL,
        [Scope] [nvarchar](255) NULL,
        [TeamID] [int] NOT NULL,
        [SelectedPartnerID] [int] NOT NULL,
CONSTRAINT [PK__Partners__AD185E5966985140] PRIMARY KEY CLUSTERED
(

```



```

[PartnershipID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Project]    Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Project](
    [ProjectID] [int] IDENTITY(1,1) NOT NULL,
    [WWCoordID] [int] NULL,
    [DateReceived] [date] NULL,
    [LinkToWwsite] [nvarchar](max) NULL,
    [UserPermission] [nvarchar](255) NULL,
    [ProjectTypeID] [int] NULL,
    [DistrictID] [int] NULL,
    [CityName] [nvarchar](255) NULL,
    [PastorName] [nvarchar](max) NULL,
    [PastorPhone] [nvarchar](max) NULL,
    [LocationLink] [nvarchar](255) NULL,
    [DistanceFromDistrictOff] [nvarchar](255) NULL,
    [DistrictBudgetCurrent] [money] NULL,

```

[DistrictBudgetLast] [money] NULL,  
[FEMcurrent] [money] NULL,  
[FEMlast] [money] NULL,  
[DateOrganized] [datetime2](7) NULL,  
[MembersLast] [int] NULL,  
[MembersCurrent] [int] NULL,  
[AvgAttendanceLast] [int] NULL,  
[AvgAttendanceCurr] [int] NULL,  
[ProjectDesc] [nvarchar](max) NULL,  
[TempleCapacity] [int] NULL,  
[RoofWidth] [int] NULL,  
[RoofLength] [int] NULL,  
[BriefDescriptionConst] [nvarchar](max) NULL,  
[PropertyWidth] [int] NULL,  
[PropertyLength] [int] NULL,  
[Photos] [bit] NULL,  
[Digital] [bit] NULL,  
[LotLayout] [bit] NULL,  
[ProjectPlans] [bit] NULL,  
[MoneyRaised] [money] NULL,  
[MoneyInvested] [money] NULL,  
[Money12Months] [money] NULL,  
[ProjectNeed] [nvarchar](max) NULL,  
[TitleToProperty] [int] NULL,

[PropertyDeededToChurch] [int] NULL,  
[BuildingPermitNeeded] [int] NULL,  
[BuildingPermitApproved] [int] NULL,  
[PastorApproved] [int] NULL,  
[ChurchBoardApproved] [int] NULL,  
[TreasureApproved] [int] NULL,  
[TreasureApprovalDate] [date] NULL,  
[SuperApproval] [int] NULL,  
[SuperApprovalDate] [date] NULL,  
[DistrictSecretaryApproval] [int] NULL,  
[DistrictTreasureApproval] [int] NULL,  
[DistrictTreasureApprovalDate] [date] NULL,  
[DistrictPriority] [int] NULL,  
[TyTApproved] [int] NULL,  
[TyTPriority] [int] NULL,  
[ProjectCompleted] [int] NULL,  
[Wwlink] [nvarchar](255) NULL,  
[PhotoLocation] [nvarchar](255) NULL,  
[PermitStart] [date] NULL,  
[PermitEnd] [date] NULL,  
[CountryID] [int] NULL,  
[ChurchID] [int] NULL,

CONSTRAINT [PK\_\_Project\_\_761ABED000DB60F3] PRIMARY KEY CLUSTERED

(

```

        [ProjectID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO

```

```

/***** Object: Table [dbo].[ProjectNotes]    Script Date: 11/9/2020
6:28:00 PM *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

CREATE TABLE [dbo].[ProjectNotes](
        [ProjectNoteID] [int] IDENTITY(1,1) NOT NULL,
        [Pdate] [datetime2](7) NULL,
        [Pnotes] [nvarchar](max) NULL,
        [ProjectID] [int] NULL,

```

```

PRIMARY KEY CLUSTERED

```

```

(

```

```

        [ProjectNoteID] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

```

```

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

```

```

GO

```

```

/***** Object: Table [dbo].[ProjectPhotos]    Script Date: 11/9/2020
6:28:00 PM *****/

```

```

SET ANSI_NULLS ON

```

```

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProjectPhotos](
        [ProjectPhotosID] [int] IDENTITY(1,1) NOT NULL,
        [ProjectPhotos] [nvarchar](255) NULL,
        [ProjectID] [int] NULL,

PRIMARY KEY CLUSTERED
(
        [ProjectPhotosID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[ProjectType]      Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[ProjectType](
        [ProjectTypeId] [int] IDENTITY(1,1) NOT NULL,
        [ProjectType] [nvarchar](255) NULL,
        [ListOrder] [int] NULL,

PRIMARY KEY CLUSTERED

```

```

(
    [ProjectTypeId] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Region]    Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Region](
    [RegionID] [int] IDENTITY(1,1) NOT NULL,
    [RegionName] [nvarchar](255) NULL,

PRIMARY KEY CLUSTERED
(
    [RegionID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[SkillLevel]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[SkillLevel](  
    [SkillLevelId] [int] NOT NULL,  
    [SkillLevel] [nvarchar](255) NULL,  
    [SkillsID] [int] NULL  
  
    ) ON [PRIMARY]
```

```
GO
```

```
/***** Object: Table [dbo].[Skills]    Script Date: 11/9/2020 6:28:00  
PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[Skills](  
    [SkillsID] [int] IDENTITY(1,1) NOT NULL,  
    [Skill] [nvarchar](255) NULL,  
  
    PRIMARY KEY CLUSTERED  
  
    (  
        [SkillsID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =  
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,  
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/***** Object: Table [dbo].[SSMA$District$local]    Script Date:  
11/9/2020 6:28:00 PM *****/
```

```

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[SSMA$District$local](
        [DistrictID] [int] IDENTITY(1,1) NOT NULL,
        [DistrictName] [nvarchar](255) NULL,
        [CountryID] [int] NULL,

PRIMARY KEY CLUSTERED
(
        [DistrictID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/***** Object: Table [dbo].[Supers]    Script Date: 11/9/2020 6:28:00
PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Supers](
        [SuperID] [int] IDENTITY(1,1) NOT NULL,
        [DistrictID] [int] NULL,
        [SuperName] [nvarchar](255) NULL,

```



```

        [SuperPhone] [nvarchar](255) NULL,
        [SuperEmail] [nvarchar](255) NULL,
        [SuperMobile] [nvarchar](255) NULL,
        [SuperWorkEmail] [nvarchar](255) NULL,
PRIMARY KEY CLUSTERED
(
        [SuperID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[Team]    Script Date: 11/9/2020 6:28:00 PM
*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Team](
        [TeamID] [int] IDENTITY(1,1) NOT NULL,
        [TeamName] [nvarchar](255) NULL,
        [ChurchName] [nvarchar](255) NULL,
        [District] [nvarchar](255) NULL,
        [LeaderEmail] [nvarchar](255) NULL,
        [LeaderName] [nvarchar](255) NULL,
        [Partership] [bit] NULL,

```

PRIMARY KEY CLUSTERED

(

[TeamID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Table [dbo].[TeamDetails] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TeamDetails](

[TeamDetailsID] [int] IDENTITY(1,1) NOT NULL,

[TotalDailyFunds] [money] NULL,

[NumTeamMemb] [int] NULL,

[TeamMemberID] [int] NULL,

[TeamTypeID] [int] NULL,

[NumberOfDays] [int] NULL,

[NumberOfNights] [int] NULL,

[ArriveFlightNumber] [nvarchar](255) NULL,

[ArriveTime] [datetime2](7) NULL,

[ArriveDate] [date] NULL,

[DepartTime] [datetime2](7) NULL,

[DepartDate] [date] NULL,  
[DepartAirline] [nvarchar](255) NULL,  
[DepartFlightNumber] [nvarchar](255) NULL,  
[ProjectID] [int] NULL,  
[TeamID] [int] NULL,  
[DistanceToRR] [int] NULL,  
[DistanceToProject] [int] NULL,  
[HotelID] [int] NULL,  
[CountryID] [int] NULL,  
[AirportID] [int] NULL,  
[TeamRegistration] [bit] NULL,  
[Insurance] [bit] NULL,  
[Maxima] [bit] NULL,  
[ProjectMoney] [money] NULL,  
[PartnershipID] [int] NULL,  
[TeamPartnerProfileID] [int] NULL,  
[TeamHost] [int] NULL,  
[CulBroker] [int] NULL,  
[TruckDriver] [int] NULL,  
[BusDriver] [int] NULL,  
[Translator] [int] NULL,  
[StructureDelivery] [int] NULL,  
[CancelTeam] [int] NULL,  
[ProjectFundsReceived] [bit] NULL,

```
CONSTRAINT [PK__TeamDeta__7A68EB530C969A05] PRIMARY KEY CLUSTERED
(
    [TeamDetailsID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/***** Object: Table [dbo].[TeamMember]    Script Date: 11/9/2020
6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[TeamMember](
    [TeamMemberID] [int] IDENTITY(1,1) NOT NULL,
    [FName] [nvarchar](255) NULL,
    [PhoneNumber] [nvarchar](255) NULL,
    [Country] [nvarchar](255) NULL,
    [District] [nvarchar](255) NULL,
    [PreviousProjects] [bit] NULL,
    [SecondLanguage] [int] NULL,
    [SkillsID] [int] NULL,
    [TeamMemberTypeID] [int] NULL,
    [TeamLeader] [int] NULL,
    [TeamID] [int] NULL,
```

```

        [LastTrip] [date] NULL,
        [Male] [int] NULL,
        [Female] [int] NULL,
        [SpouseOnTeam] [int] NULL,
        [PassPortExp] [date] NULL,
        [FoodAllergies] [nvarchar](255) NULL,
        [MedicalAllergies] [nvarchar](255) NULL,
        [DPI] [nvarchar](255) NULL,
        [Email] [nvarchar](255) NULL,

CONSTRAINT [PK__TeamMemb__C7C0928568918DC2] PRIMARY KEY CLUSTERED
(
        [TeamMemberID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO

/***** Object: Table [dbo].[TeamMemberSkills]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TeamMemberSkills](
        [TMSid] [int] IDENTITY(1,1) NOT NULL,
        [TeamMemberID] [int] NULL,

```

```

        [SkillsID] [int] NULL,
        [SkillLevelID] [int] NULL,
PRIMARY KEY CLUSTERED
(
        [TMSid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: Table [dbo].[TeamMemberType]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TeamMemberType](
        [TeamMemberTypeID] [int] IDENTITY(1,1) NOT NULL,
        [MemberType] [nvarchar](255) NULL,
PRIMARY KEY CLUSTERED
(
        [TeamMemberTypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO

```

/\*\*\*\*\* Object: Table [dbo].[TeamPartnerProfile] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[TeamPartnerProfile](
    [TeamPartnerProfileID] [int] IDENTITY(1,1) NOT NULL,
    [TeamID] [int] NULL,
    [Scope] [nvarchar](255) NULL,
    [StartDate] [datetime2](7) NULL,
    [EndDate] [datetime2](7) NULL,
    [Finished] [int] NULL,
    [ProjectedTerm] [int] NULL,
    [ScopeTitle] [nvarchar](255) NULL,
    CONSTRAINT [PK__TeamPart__BA857138728A944D] PRIMARY KEY CLUSTERED
(
    [TeamPartnerProfileID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Table [dbo].[TeamTrackInfo] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

```

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TeamTrackInfo](
            [ID] [int] IDENTITY(1,1) NOT NULL,
            [EmailPath] [nvarchar](255) NULL,
            [TeamID] [int] NULL,
            [TeamMemberTypeID] [int] NULL,
            [TeamHost] [int] NULL,
            [TruckDriver] [int] NULL,
            [BusDriver] [int] NULL,
            [Translator] [int] NULL,
            [ToolDriver] [int] NULL,
            [StructureDelivery] [int] NULL,

PRIMARY KEY CLUSTERED
(
            [ID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO

/***** Object: Table [dbo].[TeamType]      Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

```



```

GO

CREATE TABLE [dbo].[TeamType](
    [TeamTypeID] [int] IDENTITY(1,1) NOT NULL,
    [TeamDiscription] [nvarchar](255) NULL,
    [ListOrder] [int] NULL,
    PRIMARY KEY CLUSTERED
    (
        [TeamTypeID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
    OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]

GO

/***** Object: Table [dbo].[userInfo]    Script Date: 11/9/2020
6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE TABLE [dbo].[userInfo](
    [userID] [int] IDENTITY(1,1) NOT NULL,
    [userName] [nvarchar](64) NULL,
    [userPassword] [nvarchar](64) NULL,
    [userEmail] [nvarchar](64) NULL,
    [userClearanceLevel] [int] NULL,
    [userSalt] [nvarchar](64) NULL

```

```
) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Table [dbo].[WWCoordinator]      Script Date: 11/9/2020  
6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[WWCoordinator](
```

```
    [WWcoordID] [int] IDENTITY(1,1) NOT NULL,
```

```
    [FullName] [nvarchar](255) NULL,
```

```
    [StartDate] [datetime2](7) NULL,
```

```
    [NumberOfTeams] [int] NULL,
```

```
    [RegionID] [int] NULL,
```

```
    [MaxHostingTeams] [int] NULL,
```

```
    [CoordStatusID] [int] NULL,
```

```
PRIMARY KEY CLUSTERED
```

```
(
```

```
    [WWcoordID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =  
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,  
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

/\*\*\*\*\* Object: Index [AirportCode] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [AirportCode] ON [dbo].[Airport]
(
    [AirportCode] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

/\*\*\*\*\* Object: Index [AirportCountryID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [AirportCountryID] ON [dbo].[Airport]
(
    [CountryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

SET ANSI\_PADDING ON

GO

/\*\*\*\*\* Object: Index [AccountingCode] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [AccountingCode] ON [dbo].[CheckSolicitud]
(
    [AccountingCode] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

/\*\*\*\*\* Object: Index [ProjectID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [ProjectID] ON [dbo].[CheckSolicitud]
(
    [ProjectID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [DistrictID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [DistrictID] ON [dbo].[ChurchName]
(
    [DistrictID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [CountryID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[Country]
(
    [CountryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [RegionID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [RegionID] ON [dbo].[Country]
```

```

(
    [FieldID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

/***** Object: Index [CountryID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[CurrencyType]

(
    [CountryID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

/***** Object: Index [CurrencyTypeID]    Script Date: 11/9/2020
6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [CurrencyTypeID] ON [dbo].[ExchangeRate]

(
    [CurrencyTypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO

/***** Object: Index [DistrictID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [DistrictID] ON [dbo].[Field]

(
    [RegionID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/****** Object: Index [FieldID] Script Date: 11/9/2020 6:28:00 PM
*****/
```

```
CREATE NONCLUSTERED INDEX [FieldID] ON [dbo].[Field]
```

```
(
```

```
[FieldID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/****** Object: Index [CountryID] Script Date: 11/9/2020 6:28:00 PM
*****/
```

```
CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[Hotel]
```

```
(
```

```
[CountryID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/****** Object: Index [HotelID] Script Date: 11/9/2020 6:28:00 PM
*****/
```

```
CREATE NONCLUSTERED INDEX [HotelID] ON [dbo].[Hotel]
```

```
(
```

```
[HotelID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```

GO

SET ANSI_PADDING ON

GO

/***** Object: Index [PhoneNum]      Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [PhoneNum] ON [dbo].[Hotel]

(

        [PhoneNum] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [LanguageID]     Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [LanguageID] ON [dbo].[Language]

(

        [LanguageID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [LocalCoordinatorDistrictID]  Script Date:
11/9/2020 6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [LocalCoordinatorDistrictID] ON [dbo].
[LocalCoordinator]

(

        [DistrictID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

```

GO

/\*\*\*\*\* Object: Index [MealID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [MealID] ON [dbo].[Meals]

(

[MealID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [SelectedPartnerID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [SelectedPartnerID] ON [dbo].[Partnerships]

(

[SelectedPartnerID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamID] ON [dbo].[Partnerships]

(

[TeamID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [CountryID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/



```

CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[Project]

(

    [CountryID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [DistrictID] Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [DistrictID] ON [dbo].[Project]

(

    [DistrictID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [ProjectID] Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [ProjectID] ON [dbo].[Project]

(

    [ProjectID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [ProjectProjectTypeId] Script Date: 11/9/2020
6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [ProjectProjectTypeId] ON [dbo].[Project]

(

```

```

        [ProjectTypeID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [ProjectWwcoordID]    Script Date: 11/9/2020
6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [ProjectWwcoordID] ON [dbo].[Project]

(

        [WwCoordID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [ProjectNotesProjectID]    Script Date: 11/9/2020
6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [ProjectNotesProjectID] ON [dbo].
[ProjectNotes]

(

        [ProjectID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [ProjectID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [ProjectID] ON [dbo].[ProjectPhotos]

(

        [ProjectID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [ProjectPhotosID] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [ProjectPhotosID] ON [dbo].[ProjectPhotos]
```

```
(
```

```
[ProjectPhotosID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [ProjectTypeId] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [ProjectTypeId] ON [dbo].[ProjectType]
```

```
(
```

```
[ProjectTypeId] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [RegionID] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [RegionID] ON [dbo].[Region]
```

```
(
```

```
[RegionID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [SkillLevelId] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [SkillLevelId] ON [dbo].[SkillLevel]

(

[SkillLevelId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [SkillsID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [SkillsID] ON [dbo].[SkillLevel]

(

[SkillsID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [SkillID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [SkillID] ON [dbo].[Skills]

(

[SkillsID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [CountryID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[SSMA$District$local]
```

```
(
```

```
    [CountryID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [DistrictID]    Script Date: 11/9/2020 6:28:00 PM  
*****/
```

```
CREATE NONCLUSTERED INDEX [DistrictID] ON [dbo].[SSMA$District$local]
```

```
(
```

```
    [DistrictID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [SupersDistrictID]    Script Date: 11/9/2020  
6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [SupersDistrictID] ON [dbo].[Supers]
```

```
(
```

```
    [DistrictID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [USAteamID]    Script Date: 11/9/2020 6:28:00 PM  
*****/
```

```
CREATE NONCLUSTERED INDEX [USAteamID] ON [dbo].[Team]
```

```
(
```

```

[TeamID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [AirportID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [AirportID] ON [dbo].[TeamDetails]

(

[AirportID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [CountryID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[TeamDetails]

(

[CountryID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [HotelID]    Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [HotelID] ON [dbo].[TeamDetails]

(

[HotelID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [NumberOfDays] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [NumberOfDays] ON [dbo].[TeamDetails]
```

```
(
```

```
[NumberOfDays] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [NumberOfNights] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [NumberOfNights] ON [dbo].[TeamDetails]
```

```
(
```

```
[NumberOfNights] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [NumTeamMemb] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [NumTeamMemb] ON [dbo].[TeamDetails]
```

```
(
```

```
[NumTeamMemb] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [PartnershipID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [PartnershipID] ON [dbo].[TeamDetails]

(

[PartnershipID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [ProjectID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [ProjectID] ON [dbo].[TeamDetails]

(

[ProjectID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamDetailsTeamDetailsID] Script Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamDetailsTeamDetailsID] ON [dbo].[TeamDetails]

(

[TeamDetailsID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO



/\*\*\*\*\* Object: Index [TeamDetailsTeamID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamDetailsTeamID] ON [dbo].[TeamDetails]

(

[TeamID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamDetailsTeamMemberID] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamDetailsTeamMemberID] ON [dbo].  
[TeamDetails]

(

[TeamMemberID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamDetailsTeamTypeID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamDetailsTeamTypeID] ON [dbo].[TeamDetails]

(

[TeamTypeID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamPartnerProfileID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

```
CREATE NONCLUSTERED INDEX [TeamPartnerProfileID] ON [dbo].[TeamDetails]
```

```
(
```

```
    [TeamPartnerProfileID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [LanguageID]    Script Date: 11/9/2020 6:28:00 PM  
*****/
```

```
CREATE NONCLUSTERED INDEX [LanguageID] ON [dbo].[TeamMember]
```

```
(
```

```
    [SecondLanguage] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [LocalTeamMemberID]    Script Date: 11/9/2020  
6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [LocalTeamMemberID] ON [dbo].[TeamMember]
```

```
(
```

```
    [TeamMemberID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Index [PreviousProjectsID]    Script Date: 11/9/2020  
6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [PreviousProjectsID] ON [dbo].[TeamMember]
```

```
(
```

```

[PreviousProjects] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [SkillsID] Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [SkillsID] ON [dbo].[TeamMember]

(

[SkillsID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [TeamID] Script Date: 11/9/2020 6:28:00 PM
*****/

CREATE NONCLUSTERED INDEX [TeamID] ON [dbo].[TeamMember]

(

[TeamID] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =
OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]

GO

/***** Object: Index [TeamMemberTypeID] Script Date: 11/9/2020
6:28:00 PM *****/

CREATE NONCLUSTERED INDEX [TeamMemberTypeID] ON [dbo].[TeamMember]

(

[TeamMemberTypeID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [SkillLevelID] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [SkillLevelID] ON [dbo].[TeamMemberSkills]
```

```
(
```

```
[SkillLevelID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [TeamMemberSkillsSkillsID] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [TeamMemberSkillsSkillsID] ON [dbo].[TeamMemberSkills]
```

```
(
```

```
[SkillsID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [TeamMemberSkillsTeamMemberID] Script Date: 11/9/2020 6:28:00 PM *****/
```

```
CREATE NONCLUSTERED INDEX [TeamMemberSkillsTeamMemberID] ON [dbo].[TeamMemberSkills]
```

```
(
```

```
[TeamMemberID] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

/\*\*\*\*\* Object: Index [TeamMemberTypeID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamMemberTypeID] ON [dbo].[TeamMemberType]

(

[TeamMemberTypeID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamPartnerProfileTeamID] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamPartnerProfileTeamID] ON [dbo].  
[TeamPartnerProfile]

(

[TeamID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamID] ON [dbo].[TeamTrackInfo]

(

[TeamID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamMemberTypeID] Script Date: 11/9/2020  
6:28:00 PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamMemberTypeID] ON [dbo].[TeamTrackInfo]

(

[TeamMemberTypeID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [TeamTypeID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [TeamTypeID] ON [dbo].[TeamType]

(

[TeamTypeID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [CountryID] Script Date: 11/9/2020 6:28:00 PM  
\*\*\*\*\*/

CREATE NONCLUSTERED INDEX [CountryID] ON [dbo].[WWCoordinator]

(

[RegionID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB =  
OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON,  
ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

GO

/\*\*\*\*\* Object: Index [NumberOfTeams] Script Date: 11/9/2020 6:28:00  
PM \*\*\*\*\*/

CREATE NONCLUSTERED INDEX [NumberOfTeams] ON [dbo].[WWCoordinator]

```
(  
  
        [NumberOfTeams] ASC  
  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =  
    OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
/***** Object: Index [StatusID]    Script Date: 11/9/2020 6:28:00 PM  
*****/
```

```
CREATE NONCLUSTERED INDEX [StatusID] ON [dbo].[WWCoordinator]
```

```
(  
  
        [CoordStatusID] ASC  
  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB =  
    OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[Airport] ADD DEFAULT ((0)) FOR [CountryID]
```

GO

```
ALTER TABLE [dbo].[CheckSolicitud] ADD DEFAULT ((0)) FOR [CheckAmount]
```

GO

```
ALTER TABLE [dbo].[CheckSolicitud] ADD DEFAULT ((0)) FOR [USD]
```

GO

```
ALTER TABLE [dbo].[CheckSolicitud] ADD DEFAULT ((0)) FOR  
[LocalCurrency]
```

GO

```
ALTER TABLE [dbo].[CheckSolicitud] ADD DEFAULT ((0)) FOR [ExchangeRate]
```

GO

```
ALTER TABLE [dbo].[CheckSolicitud] ADD DEFAULT ((0)) FOR [ProjectID]
```

GO

```

ALTER TABLE [dbo].[ChurchName] ADD DEFAULT ((0)) FOR [DistrictID]
GO

ALTER TABLE [dbo].[Country] ADD DEFAULT ((0)) FOR [FieldID]
GO

ALTER TABLE [dbo].[CurrencyType] ADD DEFAULT ((0)) FOR [CountryID]
GO

ALTER TABLE [dbo].[ExchangeRate] ADD DEFAULT ((0)) FOR [ExchangeRate]
GO

ALTER TABLE [dbo].[ExchangeRate] ADD DEFAULT ((0)) FOR [CurrencyTypeID]
GO

ALTER TABLE [dbo].[Field] ADD DEFAULT ((0)) FOR [RegionID]
GO

ALTER TABLE [dbo].[Hotel] ADD DEFAULT ((0)) FOR [CountryID]
GO

ALTER TABLE [dbo].[Hotel] ADD DEFAULT ((0)) FOR [R&R]
GO

ALTER TABLE [dbo].[LocalCoordinator] ADD DEFAULT ((0)) FOR [DistrictID]
GO

ALTER TABLE [dbo].[Meals] ADD CONSTRAINT
[DF__Meals__DaysForTh__1AD3FDA4] DEFAULT ((0)) FOR [DaysForThisMeal]
GO

ALTER TABLE [dbo].[Meals] ADD CONSTRAINT
[DF__Meals__MealCost__1BC821DD] DEFAULT ((0)) FOR [MealCost]
GO

ALTER TABLE [dbo].[Meals] ADD CONSTRAINT
[DF__Meals__HowManyEa__1CBC4616] DEFAULT ((0)) FOR [HowManyEating]

```



```

GO

ALTER TABLE [dbo].[Partnerships] ADD CONSTRAINT
[DF__Partnersh__TeamI__76969D2E] DEFAULT ((0)) FOR [TeamID]

GO

ALTER TABLE [dbo].[Partnerships] ADD CONSTRAINT
[DF__Partnersh__Selec__787EE5A0] DEFAULT ((0)) FOR [SelectedPartnerID]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__WWCoord__3E1D39E1] DEFAULT ((0)) FOR [WWCoordID]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Project__40058253] DEFAULT ((0)) FOR [ProjectTypeID]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__40F9A68C] DEFAULT ((0)) FOR [DistrictID]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__41EDCAC5] DEFAULT ((0)) FOR
[DistrictBudgetCurrent]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__42E1EEFE] DEFAULT ((0)) FOR [DistrictBudgetLast]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__FEMcurr__43D61337] DEFAULT ((0)) FOR [FEMcurrent]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__FEMlast__44CA3770] DEFAULT ((0)) FOR [FEMlast]

GO

```

```

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Members__45BE5BA9] DEFAULT ((0)) FOR [MembersLast]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Members__46B27FE2] DEFAULT ((0)) FOR [MembersCurrent]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__AvgAtte__47A6A41B] DEFAULT ((0)) FOR [AvgAttendanceLast]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__AvgAtte__489AC854] DEFAULT ((0)) FOR [AvgAttendanceCurr]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__TempleC__498EEC8D] DEFAULT ((0)) FOR [TempleCapacity]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__RoofWid__4A8310C6] DEFAULT ((0)) FOR [RoofWidth]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__RoofLen__4B7734FF] DEFAULT ((0)) FOR [RoofLength]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Propert__4C6B5938] DEFAULT ((0)) FOR [PropertyWidth]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Propert__4D5F7D71] DEFAULT ((0)) FOR [PropertyLength]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Photos__4E53A1AA] DEFAULT ((0)) FOR [Photos]

GO

```

```

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Digital__4F47C5E3] DEFAULT ((0)) FOR [Digital]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__LotLayo__503BEA1C] DEFAULT ((0)) FOR [LotLayout]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Project__51300E55] DEFAULT ((0)) FOR [ProjectPlans]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__MoneyRa__5224328E] DEFAULT ((0)) FOR [MoneyRaised]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__MoneyIn__531856C7] DEFAULT ((0)) FOR [MoneyInvested]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Money12__540C7B00] DEFAULT ((0)) FOR [Money12Months]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__TitleTo__55009F39] DEFAULT ((0)) FOR [TitleToProperty]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Propert__55F4C372] DEFAULT ((0)) FOR
[PropertyDeededToChurch]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Buildin__56E8E7AB] DEFAULT ((0)) FOR
[BuildingPermitNeeded]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Buildin__57DD0BE4] DEFAULT ((0)) FOR
[BuildingPermitApproved]

```

```

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__PastorA__58D1301D] DEFAULT ((0)) FOR [PastorApproved]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__ChurchB__59C55456] DEFAULT ((0)) FOR
[ChurchBoardApproved]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Treasur__5AB9788F] DEFAULT ((0)) FOR [TreasureApproved]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__SuperAp__5BAD9CC8] DEFAULT ((0)) FOR [SuperApproval]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__5CA1C101] DEFAULT ((0)) FOR
[DistrictSecretaryApproval]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__5D95E53A] DEFAULT ((0)) FOR
[DistrictTreasureApproval]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Distric__5E8A0973] DEFAULT ((0)) FOR [DistrictPriority]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__TyTPrio__607251E5] DEFAULT ((0)) FOR [TyTPriority]

GO

ALTER TABLE [dbo].[Project] ADD CONSTRAINT
[DF__Project__Country__6166761E] DEFAULT ((0)) FOR [CountryID]

GO

```

```

ALTER TABLE [dbo].[ProjectNotes] ADD DEFAULT ((0)) FOR [ProjectID]

GO

ALTER TABLE [dbo].[ProjectPhotos] ADD DEFAULT ((0)) FOR [ProjectID]

GO

ALTER TABLE [dbo].[ProjectType] ADD DEFAULT ((0)) FOR [ListOrder]

GO

ALTER TABLE [dbo].[SkillLevel] ADD CONSTRAINT
[DF__SkillLeve__Skill__7B264821] DEFAULT ((0)) FOR [SkillsID]

GO

ALTER TABLE [dbo].[SSMA$District$local] ADD DEFAULT ((0)) FOR
[CountryID]

GO

ALTER TABLE [dbo].[Supers] ADD DEFAULT ((0)) FOR [DistrictID]

GO

ALTER TABLE [dbo].[Supers] ADD DEFAULT (N'0') FOR [SuperMobile]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Total__13F1F5EB] DEFAULT ((0)) FOR [TotalDailyFunds]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__NumTe__14E61A24] DEFAULT ((0)) FOR [NumTeamMemb]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__TeamM__15DA3E5D] DEFAULT ((0)) FOR [TeamMemberID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__TeamT__16CE6296] DEFAULT ((0)) FOR [TeamTypeID]

GO

```

```

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Numbe__17C286CF] DEFAULT ((0)) FOR [NumberOfDays]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Numbe__18B6AB08] DEFAULT ((0)) FOR [NumberOfNights]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Proje__19AACF41] DEFAULT ((0)) FOR [ProjectID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__TeamI__1A9EF37A] DEFAULT ((0)) FOR [TeamID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Dista__1B9317B3] DEFAULT ((0)) FOR [DistanceToRR]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Dista__1C873BEC] DEFAULT ((0)) FOR [DistanceToProject]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Hotel__1D7B6025] DEFAULT ((0)) FOR [HotelID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Count__1F63A897] DEFAULT ((0)) FOR [CountryID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Airpo__2057CCD0] DEFAULT ((0)) FOR [AirportID]

GO

ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Proje__2334397B] DEFAULT ((0)) FOR [ProjectMoney]

GO

```

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Partn__24285DB4] DEFAULT ((0)) FOR [PartnershipID]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__TeamP__251C81ED] DEFAULT ((0)) FOR
[TeamPartnerProfileID]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__TeamH__2610A626] DEFAULT ((0)) FOR [TeamHost]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__CulBr__2704CA5F] DEFAULT ((0)) FOR [CulBroker]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Truck__27F8EE98] DEFAULT ((0)) FOR [TruckDriver]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__BusDr__28ED12D1] DEFAULT ((0)) FOR [BusDriver]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Trans__29E1370A] DEFAULT ((0)) FOR [Translator]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] ADD CONSTRAINT
[DF__TeamDetai__Struc__2AD55B43] DEFAULT ((0)) FOR [StructureDelivery]
```

GO

```
ALTER TABLE [dbo].[TeamMember] ADD CONSTRAINT
[DF__TeamMembe__Langu__3EDC53F0] DEFAULT ((0)) FOR [SecondLanguage]
```

GO

```
ALTER TABLE [dbo].[TeamMember] ADD CONSTRAINT
[DF__TeamMembe__Skill__3FD07829] DEFAULT ((0)) FOR [SkillsID]
```

```

GO

ALTER TABLE [dbo].[TeamMember] ADD CONSTRAINT
[DF__TeamMembe__TeamM__40C49C62] DEFAULT ((0)) FOR [TeamMemberTypeID]

GO

ALTER TABLE [dbo].[TeamMember] ADD CONSTRAINT
[DF__TeamMembe__TeamL__41B8C09B] DEFAULT ((0)) FOR [TeamLeader]

GO

ALTER TABLE [dbo].[TeamMember] ADD CONSTRAINT
[DF__TeamMembe__TeamI__43A1090D] DEFAULT ((0)) FOR [TeamID]

GO

ALTER TABLE [dbo].[TeamMemberSkills] ADD DEFAULT ((0)) FOR
[TeamMemberID]

GO

ALTER TABLE [dbo].[TeamMemberSkills] ADD DEFAULT ((0)) FOR [SkillsID]

GO

ALTER TABLE [dbo].[TeamMemberSkills] ADD DEFAULT ((0)) FOR
[SkillLevelID]

GO

ALTER TABLE [dbo].[TeamPartnerProfile] ADD CONSTRAINT
[DF__TeamPartn__TeamI__3F115E1A] DEFAULT ((0)) FOR [TeamID]

GO

ALTER TABLE [dbo].[TeamPartnerProfile] ADD CONSTRAINT
[DF__TeamPartn__Proje__40058253] DEFAULT ((0)) FOR [ProjectedTerm]

GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [TeamID]

GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [TeamHost]

GO

```



```

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [TruckDriver]
GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [BusDriver]
GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [Translator]
GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR [ToolDriver]
GO

ALTER TABLE [dbo].[TeamTrackInfo] ADD DEFAULT ((0)) FOR
[StructureDelivery]
GO

ALTER TABLE [dbo].[TeamType] ADD DEFAULT ((0)) FOR [ListOrder]
GO

ALTER TABLE [dbo].[WWCoordinator] ADD DEFAULT ((0)) FOR [NumberOfTeams]
GO

ALTER TABLE [dbo].[WWCoordinator] ADD DEFAULT ((0)) FOR [RegionID]
GO

ALTER TABLE [dbo].[WWCoordinator] ADD DEFAULT ((0)) FOR
[MaxHostingTeams]
GO

ALTER TABLE [dbo].[WWCoordinator] ADD DEFAULT ((0)) FOR [CoordStatusID]
GO

ALTER TABLE [dbo].[CheckSolicitud] WITH CHECK ADD CONSTRAINT
[FK_CheckSolicitud_Project] FOREIGN KEY([ProjectID])
REFERENCES [dbo].[Project] ([ProjectID])
GO

```

```
ALTER TABLE [dbo].[CheckSolicitud] CHECK CONSTRAINT  
[FK_CheckSolicitud_Project]
```

GO

```
ALTER TABLE [dbo].[ChurchName] WITH CHECK ADD CONSTRAINT  
[FK_ChurchName_SSMA$District$local] FOREIGN KEY([DistrictID])
```

```
REFERENCES [dbo].[SSMA$District$local] ([DistrictID])
```

GO

```
ALTER TABLE [dbo].[ChurchName] CHECK CONSTRAINT  
[FK_ChurchName_SSMA$District$local]
```

GO

```
ALTER TABLE [dbo].[Country] WITH CHECK ADD CONSTRAINT  
[FK_Country_Field] FOREIGN KEY([FieldID])
```

```
REFERENCES [dbo].[Field] ([FieldID])
```

GO

```
ALTER TABLE [dbo].[Country] CHECK CONSTRAINT [FK_Country_Field]
```

GO

```
ALTER TABLE [dbo].[CurrencyType] WITH CHECK ADD CONSTRAINT  
[FK_CurrencyType_Country] FOREIGN KEY([CountryID])
```

```
REFERENCES [dbo].[Country] ([CountryID])
```

GO

```
ALTER TABLE [dbo].[CurrencyType] CHECK CONSTRAINT  
[FK_CurrencyType_Country]
```

GO

```
ALTER TABLE [dbo].[ExchangeRate] WITH CHECK ADD CONSTRAINT  
[FK_ExchangeRate_CurrencyType] FOREIGN KEY([CurrencyTypeID])
```

```
REFERENCES [dbo].[CurrencyType] ([CurrencyTypeID])
```

GO

```
ALTER TABLE [dbo].[ExchangeRate] CHECK CONSTRAINT  
[FK_ExchangeRate_CurrencyType]
```

```

GO

ALTER TABLE [dbo].[Field] WITH CHECK ADD CONSTRAINT [FK_Field_Region]
FOREIGN KEY([RegionID])

REFERENCES [dbo].[Region] ([RegionID])

GO

ALTER TABLE [dbo].[Field] CHECK CONSTRAINT [FK_Field_Region]

GO

ALTER TABLE [dbo].[LocalCoordinator] WITH CHECK ADD CONSTRAINT
[FK_LocalCoordinator_SSMA$District$local] FOREIGN KEY([DistrictID])

REFERENCES [dbo].[SSMA$District$local] ([DistrictID])

GO

ALTER TABLE [dbo].[LocalCoordinator] CHECK CONSTRAINT
[FK_LocalCoordinator_SSMA$District$local]

GO

ALTER TABLE [dbo].[Meals] WITH CHECK ADD CONSTRAINT
[FK_Meals_TeamDetails] FOREIGN KEY([TeamDetailsID])

REFERENCES [dbo].[TeamDetails] ([TeamDetailsID])

GO

ALTER TABLE [dbo].[Meals] CHECK CONSTRAINT [FK_Meals_TeamDetails]

GO

ALTER TABLE [dbo].[Partnerships] WITH CHECK ADD CONSTRAINT
[FK_Partnerships_Team] FOREIGN KEY([TeamID])

REFERENCES [dbo].[Team] ([TeamID])

GO

ALTER TABLE [dbo].[Partnerships] CHECK CONSTRAINT [FK_Partnerships_Team]

GO

ALTER TABLE [dbo].[Project] WITH CHECK ADD CONSTRAINT
[FK_Project_ChurchName] FOREIGN KEY([ChurchID])

```

```

REFERENCES [dbo].[ChurchName] ([ChurchID])

GO

ALTER TABLE [dbo].[Project] CHECK CONSTRAINT [FK_Project_ChurchName]

GO

ALTER TABLE [dbo].[Project] WITH CHECK ADD CONSTRAINT
[FK_Project_Country] FOREIGN KEY([CountryID])

REFERENCES [dbo].[Country] ([CountryID])

GO

ALTER TABLE [dbo].[Project] CHECK CONSTRAINT [FK_Project_Country]

GO

ALTER TABLE [dbo].[Project] WITH CHECK ADD CONSTRAINT
[FK_Project_ProjectType] FOREIGN KEY([ProjectTypeID])

REFERENCES [dbo].[ProjectType] ([ProjectTypeID])

GO

ALTER TABLE [dbo].[Project] CHECK CONSTRAINT [FK_Project_ProjectType]

GO

ALTER TABLE [dbo].[Project] WITH CHECK ADD CONSTRAINT
[FK_Project_SSMA$District$local] FOREIGN KEY([DistrictID])

REFERENCES [dbo].[SSMA$District$local] ([DistrictID])

GO

ALTER TABLE [dbo].[Project] CHECK CONSTRAINT
[FK_Project_SSMA$District$local]

GO

ALTER TABLE [dbo].[Project] WITH CHECK ADD CONSTRAINT
[FK_Project_WWCoordinator] FOREIGN KEY([WWCoordID])

REFERENCES [dbo].[WWCoordinator] ([WWcoordID])

GO

```

```
ALTER TABLE [dbo].[Project] CHECK CONSTRAINT [FK_Project_WWCoordinator]
```

```
GO
```

```
ALTER TABLE [dbo].[ProjectNotes] WITH CHECK ADD CONSTRAINT  
[FK_ProjectNotes_Project] FOREIGN KEY([ProjectID])
```

```
REFERENCES [dbo].[Project] ([ProjectID])
```

```
GO
```

```
ALTER TABLE [dbo].[ProjectNotes] CHECK CONSTRAINT  
[FK_ProjectNotes_Project]
```

```
GO
```

```
ALTER TABLE [dbo].[ProjectPhotos] WITH CHECK ADD CONSTRAINT  
[FK_ProjectPhotos_Project1] FOREIGN KEY([ProjectID])
```

```
REFERENCES [dbo].[Project] ([ProjectID])
```

```
GO
```

```
ALTER TABLE [dbo].[ProjectPhotos] CHECK CONSTRAINT  
[FK_ProjectPhotos_Project1]
```

```
GO
```

```
ALTER TABLE [dbo].[SkillLevel] WITH CHECK ADD CONSTRAINT  
[FK_SkillLevel_Skills] FOREIGN KEY([SkillsID])
```

```
REFERENCES [dbo].[Skills] ([SkillsID])
```

```
GO
```

```
ALTER TABLE [dbo].[SkillLevel] CHECK CONSTRAINT [FK_SkillLevel_Skills]
```

```
GO
```

```
ALTER TABLE [dbo].[SSMA$District$local] WITH CHECK ADD CONSTRAINT  
[FK_SSMA$District$local_Country] FOREIGN KEY([CountryID])
```

```
REFERENCES [dbo].[Country] ([CountryID])
```

```
GO
```

```
ALTER TABLE [dbo].[SSMA$District$local] CHECK CONSTRAINT  
[FK_SSMA$District$local_Country]
```

GO

```
ALTER TABLE [dbo].[Supers] WITH CHECK ADD CONSTRAINT  
[FK_Supers_SSMA$District$local] FOREIGN KEY([DistrictID])
```

```
REFERENCES [dbo].[SSMA$District$local] ([DistrictID])
```

GO

```
ALTER TABLE [dbo].[Supers] CHECK CONSTRAINT  
[FK_Supers_SSMA$District$local]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT  
[FK_TeamDetails_Airport] FOREIGN KEY([AirportID])
```

```
REFERENCES [dbo].[Airport] ([AirportID])
```

GO

```
ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT  
[FK_TeamDetails_Airport]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT  
[FK_TeamDetails_Hotel] FOREIGN KEY([HotelID])
```

```
REFERENCES [dbo].[Hotel] ([HotelID])
```

GO

```
ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT [FK_TeamDetails_Hotel]
```

GO

```
ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT  
[FK_TeamDetails_Project] FOREIGN KEY([ProjectID])
```

```
REFERENCES [dbo].[Project] ([ProjectID])
```

GO

```
ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT  
[FK_TeamDetails_Project]
```

GO

```

ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT
[FK_TeamDetails_Team] FOREIGN KEY([TeamID])

REFERENCES [dbo].[Team] ([TeamID])

GO

ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT [FK_TeamDetails_Team]

GO

ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT
[FK_TeamDetails_TeamMember] FOREIGN KEY([TeamMemberID])

REFERENCES [dbo].[TeamMember] ([TeamMemberID])

GO

ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT
[FK_TeamDetails_TeamMember]

GO

ALTER TABLE [dbo].[TeamDetails] WITH CHECK ADD CONSTRAINT
[FK_TeamDetails_TeamType1] FOREIGN KEY([TeamTypeID])

REFERENCES [dbo].[TeamType] ([TeamTypeID])

GO

ALTER TABLE [dbo].[TeamDetails] CHECK CONSTRAINT
[FK_TeamDetails_TeamType1]

GO

ALTER TABLE [dbo].[TeamMember] WITH CHECK ADD CONSTRAINT
[FK_TeamMember_Language] FOREIGN KEY([SecondLanguage])

REFERENCES [dbo].[Language] ([LanguageID])

GO

ALTER TABLE [dbo].[TeamMember] CHECK CONSTRAINT [FK_TeamMember_Language]

GO

ALTER TABLE [dbo].[TeamMember] WITH CHECK ADD CONSTRAINT
[FK_TeamMember_Team] FOREIGN KEY([TeamID])

```

```

REFERENCES [dbo].[Team] ([TeamID])

GO

ALTER TABLE [dbo].[TeamMember] CHECK CONSTRAINT [FK_TeamMember_Team]

GO

ALTER TABLE [dbo].[TeamMember] WITH CHECK ADD CONSTRAINT
[FK_TeamMember_TeamMemberType] FOREIGN KEY([TeamMemberTypeID])

REFERENCES [dbo].[TeamMemberType] ([TeamMemberTypeID])

GO

ALTER TABLE [dbo].[TeamMember] CHECK CONSTRAINT
[FK_TeamMember_TeamMemberType]

GO

ALTER TABLE [dbo].[TeamMemberSkills] WITH CHECK ADD CONSTRAINT
[FK_TeamMemberSkills_Skills] FOREIGN KEY([SkillsID])

REFERENCES [dbo].[Skills] ([SkillsID])

GO

ALTER TABLE [dbo].[TeamMemberSkills] CHECK CONSTRAINT
[FK_TeamMemberSkills_Skills]

GO

ALTER TABLE [dbo].[TeamMemberSkills] WITH CHECK ADD CONSTRAINT
[FK_TeamMemberSkills_TeamMember] FOREIGN KEY([TeamMemberID])

REFERENCES [dbo].[TeamMember] ([TeamMemberID])

GO

ALTER TABLE [dbo].[TeamMemberSkills] CHECK CONSTRAINT
[FK_TeamMemberSkills_TeamMember]

GO

ALTER TABLE [dbo].[TeamPartnerProfile] WITH CHECK ADD CONSTRAINT
[FK_TeamPartnerProfile_Team] FOREIGN KEY([TeamID])

REFERENCES [dbo].[Team] ([TeamID])

```



```

GO

ALTER TABLE [dbo].[TeamPartnerProfile] CHECK CONSTRAINT
[FK_TeamPartnerProfile_Team]

GO

ALTER TABLE [dbo].[TeamTrackInfo] WITH CHECK ADD CONSTRAINT
[FK_TeamTrackInfo_Team] FOREIGN KEY([TeamID])

REFERENCES [dbo].[Team] ([TeamID])

GO

ALTER TABLE [dbo].[TeamTrackInfo] CHECK CONSTRAINT
[FK_TeamTrackInfo_Team]

GO

ALTER TABLE [dbo].[WWCoordinator] WITH CHECK ADD CONSTRAINT
[FK_WWCoordinator_Region] FOREIGN KEY([RegionID])

REFERENCES [dbo].[Region] ([RegionID])

GO

ALTER TABLE [dbo].[WWCoordinator] CHECK CONSTRAINT
[FK_WWCoordinator_Region]

GO

/***** Object: StoredProcedure [dbo].[AddTeamMember]      Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROC [dbo].[AddTeamMember] (
        @NTMName NVARCHAR(255),
        @NTMPhoneNumber NVARCHAR(255),
        @NTMCountry NVARCHAR(255),

```

```

        @NTMDistrict                NVARCHAR(255),
        @NTMPreviousProjects        INT,
        @NTMSecondLanguage          NVARCHAR(50),
        @NTMSkillsID                INT,
        @NTMTeamMemberTypeID        INT,
        @NTMTeamLeader              INT,
        @NTMTeamID                  INT,
        @NTMLastTrip                DATE,
        @NTMMale                    INT,
        @NTMFemale                  INT,
        @NTMSpouseOnTeam            INT,
        @NTMPassPortExp              DATE,
        @NTMFoodAllergies           NVARCHAR(255),
        @NTMMedicalAllergies        NVARCHAR(255),
        @NTMDPI                      NVARCHAR(255),
        @NTMEmail                    NVARCHAR(255)
    )
AS
BEGIN
    IF NOT EXISTS (SELECT * FROM TeamMember WHERE FName = @NTMFName AND
    Email = @NTMEmail)
    BEGIN
        INSERT INTO TeamMember(FName, PhoneNumber, Country,
        District, PreviousProjects, SecondLanguage, SkillsID, TeamMemberTypeID,
        TeamLeader

```

```

,TeamID, LastTrip, Male, Female, SpouseOnTeam,
PassPortExp, FoodAllergies, MedicalAllergies, DPI, Email)

VALUES (@NTMFName, @NTMPhoneNumber, @NTMCountry,
@NTMDistrict, @NTMPreviousProjects, @NTMSecondLanguage, @NTMSkillsID,
@NTMTeamMemberTypeID,

@NTMTeamLeader, @NTMTeamID, @NTMLastTrip, @NTMMale,
@NTMFemale, @NTMSpouseOnTeam, @NTMPassPortExp, @NTMFoodAllergies,

@NTMMedicalAllergies, @NTMDPI, @NTMEmail)

UPDATE TeamDetails

SET

NumTeamMemb = NumTeamMemb + 1

WHERE TeamID = @NTMTeamID

INSERT INTO TeamMemberSkills(TeamMemberID, SkillsID)

VALUES (SCOPE_IDENTITY(), @NTMSkillsID)

END

ELSE

BEGIN --This record exists, update it

UPDATE TeamMember

SET

FName = @NTMFName,

PhoneNumber = @NTMPhoneNumber,

Country = @NTMCountry,

District = @NTMDistrict,

PreviousProjects = @NTMPreviousProjects,

SecondLanguage = @NTMSecondLanguage,

SkillsID = @NTMSkillsID,

```

```

TeamMemberTypeID      = @NTMTeamMemberTypeID,
TeamLeader            = @NTMTeamLeader,
TeamID                = @NTMTeamID,
LastTrip              = @NTMLastTrip,
Male                  = @NTMMale,
Female                = @NTMFemale,
SpouseOnTeam         = @NTMSpouseOnTeam,
PassPortExp          = @NTMPassPortExp,
FoodAllergies        = @NTMFoodAllergies,
MedicalAllergies     = @NTMMedicalAllergies,
DPI                   = @NTMDPI,
Email                 = @NTMEmail

```

```

WHERE Email = @NTMEmail AND FName = @NTMFName AND PhoneNumber =
@NTMPhoneNumber

```

```

RETURN 1

```

```

END

```

```

END

```

```

GO

```

```

/***** Object: StoredProcedure [dbo].[AssignTeam]    Script Date:
11/9/2020 6:28:00 PM *****/

```

```

SET ANSI_NULLS ON

```

```

GO

```

```

SET QUOTED_IDENTIFIER ON

```

```

GO

```

```

CREATE          PROC [dbo].[AssignTeam] (
                @TeamID          INT,

```

```

        @ProjectID INT
    )
AS
BEGIN
    Declare @LeaderID INT;
    Declare @TeamProjCount INT;
    SET @LeaderID = (SELECT TeamMemberID
    FROM TeamMember
    WHERE TeamID = @TeamID AND TeamLeader = 1)
    SET @TeamProjCount = (SELECT COUNT(*) FROM TeamDetails WHERE TeamID =
    @TeamID)
    IF (@TeamProjCount > 1)
    BEGIN
        SELECT * FROM TeamDetails WHERE TeamID = @TeamID
    RETURN 0
    END
    ELSE
    BEGIN
    UPDATE TeamDetails
    SET
        CancelTeam = 1,
        ProjectID = 0
    WHERE ProjectID = @ProjectID
    UPDATE TeamDetails          --update teamdetails ID
    SET                          --
    choose which TeamDetailsID to Update

```

```

ProjectID = @ProjectID

WHERE TeamID = @TeamID

IF NOT EXISTS(SELECT * FROM TeamDetails WHERE TeamID = @TeamID or
ProjectID = @ProjectID)

BEGIN

INSERT INTO TeamDetails(ProjectID, TeamID) VALUES(@ProjectID, @TeamID)

END

ELSE

BEGIN

INSERT INTO TeamDetails(CountryID)

SELECT

[CountryID]

FROM Project WHERE ProjectID = @ProjectID;

RETURN 1

END

END

END

GO

/***** Object: StoredProcedure [dbo].[AuthenticateUser] Script
Date: 11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROC [dbo].[AuthenticateUser] (
@user nvarchar(64),

```

```

        @pass nvarchar(64))

AS

BEGIN

        DECLARE @TESTSALT                NVARCHAR(64);
        DECLARE @TESTPASS                NVARCHAR(64);
        DECLARE @RESULT                    NVARCHAR(64);
        DECLARE @RESCONVERTER            NVARCHAR(64);

SELECT @TESTSALT = userSalt, @TESTPASS = userPassword
FROM userInfo
WHERE userName = @user

SELECT @TESTSALT

SET @RESULT = @pass + @TESTSALT;

SELECT @RESULT;

SET @RESCONVERTER = (HASHBYTES('SHA2_512',@RESULT));

SELECT @RESCONVERTER

IF NOT EXISTS(

        (SELECT TOP(1)

                [userID],

                [userName],

                [userPassword],

                [userEmail],

                [userClearanceLevel]

        FROM userInfo

```

```

WHERE
    userName = @user
    AND
    userPassword = @RESCONVERTER))
    BEGIN
        SELECT @user
        SELECT @RESCONVERTER
        SELECT ('FAIL')
        RETURN 0 --Fail
    END
ELSE
    BEGIN
        SELECT @user
        SELECT @RESCONVERTER
        SELECT ('SUCCESS')
        RETURN 1 --Success
    END
END

GO

/***** Object: StoredProcedure [dbo].[CancelledTeams]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

```



```

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[CancelledTeams]

AS

BEGIN

SELECT DISTINCT

TeamDetails.[TeamID],

[TeamName],

[ChurchName],

[District],

[LeaderEmail],

[LeaderName],

[Partership],

[CancelTeam]

FROM Team INNER JOIN TeamDetails ON (Team.TeamID = TeamDetails.TeamID)

WHERE CancelTeam = 1;

END

GO

/***** Object: StoredProcedure [dbo].[createNewProject]      Script
Date: 11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

```

```

CREATE PROC [dbo].[createNewProject](
    @NewWWCoorID
    INT,
    @NewDateRecieved
    DATE,
    @NewLinkToWwsite
    NVARCHAR(255),
    @NewUserPermission
    NVARCHAR(255),
    @NewProjectTypeID
    INT,
    @NewDistrictID
    INT,
    @NewCityName
    NVARCHAR(255),
    @NewPastorName
    NVARCHAR(MAX),
    @NewPastorPhone
    NVARCHAR(MAX),
    @NewLocationLink
    NVARCHAR(255),
    @NewDistanceFromDistrictOff
    NVARCHAR(255),
    @NewDistrictBudgetCurrent
    MONEY,
    @NewDistrictBudgetLast
    MONEY,
    @NewFEMCurrent
    MONEY,
    @NewFEMLast
    MONEY,
    @NewDateOrganized
    DATE,
    @NewMembersLast
    INT,

```

@NewMembersCurrent  
 INT,  
 @NewAvgAttendanceLast INT,  
 @NewAvgAttendanceCurent INT,  
 @NewProjectDesc  
 NVARCHAR(MAX),  
 @NewTempleCapacity  
 INT,  
 @NewRoofWidth  
 INT,  
 @NewRoofLength  
 INT,  
 @NewBriefDescriptionConst  
 NVARCHAR(MAX),  
 @NewPropertyWidth INT,  
 @NewPropertyLength  
 INT,  
 @NewPhotos  
 INT,  
 @NewDigital  
 INT,  
 @NewLotLayout  
 INT,  
 @NewProjectPlans INT,  
 @NewMoneyRaised  
 MONEY,  
 @NewMoneyInvested MONEY,  
 @NewMoney12Months MONEY,  
 @NewProjectNeed  
 NVARCHAR(MAX),  
 @NewTitleToProperty  
 INT,

```

        @NewPropertyDeededToChurch          INT,
        @NewBuildingPermitNeeded            INT,
        @NewBuildingPermitApproved          INT,
        @NewPastorApproved
INT,
        @NewChurchBoardApproved            INT,
        @NewTreasureApproved                INT,
        @NewTreasureApprovedDate            DATE,
        @NewSuperApproval                   INT,
        @NewSuperApprovalDate               DATE,
        @NewDistrictSecretaryApproval       INT,
        @NewDistrictTreasureApproval        INT,
        @NewDistrictTreasureApprovalDate    Date,
        @NewDistrictPriority                 INT,
        @NewTyTApproved
INT,
        @NewTyTPriority
INT,
        @NewProjectCompleted                INT,
        @NewWwLink
NVARCHAR(255),
        @NewPhotoLocation
NVARCHAR(255),
        @NewPermitStart
DATE,
        @NewPermitEnd
DATE,
        @NewCountryID
INT

```

```
)  
AS  
BEGIN  
INSERT INTO Project(  
    WWCoordID,  
    DateReceived,  
    LinkToWbsite,  
    UserPermission,  
    ProjectTypeID,  
    DistrictID,  
    CityName,  
    PastorName,  
    PastorPhone,  
    LocationLink,  
    DistanceFromDistrictOff,  
    DistrictBudgetCurrent,  
    DistrictBudgetLast,  
    FEMcurrent,  
    FEMlast,  
    DateOrganized,  
    MembersLast,  
    MembersCurrent,  
    AvgAttendanceLast,  
    AvgAttendanceCurr,
```

ProjectDesc,  
TempleCapacity,  
RoofWidth,  
RoofLength,  
BriefDescriptionConst,  
PropertyWidth,  
PropertyLength,  
Photos,  
Digital,  
LotLayout,  
ProjectPlans,  
MoneyRaised,  
MoneyInvested,  
Money12Months,  
ProjectNeed,  
TitleToProperty,  
PropertyDeededToChurch,  
BuildingPermitNeeded,  
BuildingPermitApproved,  
PastorApproved,  
ChurchBoardApproved,  
TreasureApproved,  
TreasureApprovalDate,  
SuperApproval,

```
SuperApprovalDate,  
DistrictSecretaryApproval,  
DistrictTreasureApproval,  
DistrictTreasureApprovalDate,  
DistrictPriority,  
TyTapproved,  
TyTPriority,  
ProjectCompleted,  
Wwlink,  
PhotoLocation,  
PermitStart,  
PermitEnd,  
CountryID  
)  
VALUES(  
@NewWWCoorID,  
@NewDateRecieved,  
@NewLinkToWWSite,  
@NewUserPermission,  
@NewProjectTypeID,  
@NewDistrictID,  
@NewCityName,  
@NewPastorName,  
@NewPastorPhone,
```

@NewLocationLink,  
@NewDistanceFromDistrictOff,  
@NewDistrictBudgetCurrent,  
@NewDistrictBudgetLast,  
@NewFEMcurrent,  
@NewFEMlast,  
@NewDateOrganized,  
@NewMembersLast,  
@NewMembersCurrent,  
@NewAvgAttendanceLast,  
@NewAvgAttendanceCurent,  
@NewProjectDesc,  
@NewTempleCapacity,  
@NewRoofWidth,  
@NewRoofLength,  
@NewBriefDescriptionConst,  
@NewPropertyWidth,  
@NewPropertyLength,  
@NewPhotos,  
@NewDigital,  
@NewLotLayout,  
@NewProjectPlans,  
@NewMoneyRaised,  
@NewMoneyInvested,



@NewMoney12Months,  
@NewProjectNeed,  
@NewTitleToProperty,  
@NewPropertyDeededToChurch,  
@NewBuildingPermitNeeded,  
@NewBuildingPermitApproved,  
@NewPastorApproved,  
@NewChurchBoardApproved,  
@NewTreasureApproved,  
@NewTreasureApprovedDate,  
@NewSuperApproval,  
@NewSuperApprovalDate,  
@NewDistrictSecretaryApproval,  
@NewDistrictTreasureApproval,  
@NewDistrictTreasureApprovalDate,  
@NewDistrictPriority,  
@NewTyTApproved,  
@NewTyTPriority,  
@NewProjectCompleted,  
@NewWwlink,  
@NewPhotoLocation,  
@NewPermitStart,  
@NewPermitEnd,  
@NewCountryID)

END

GO

/\*\*\*\*\* Object: StoredProcedure [dbo].[createNewTeam] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE          PROC [dbo].[createNewTeam](  
                @NewTeamName          NVARCHAR(255),  
                @NewChurchName        NVARCHAR(255),  
                @NewDistrict          NVARCHAR(255),  
                @NewLeaderEmail       NVARCHAR(255),  
                @NewLeaderName        NVARCHAR(255),  
                @NewPartership        INT
```

```
)
```

AS

BEGIN

```
SET @NewTeamName = CONVERT(NVARCHAR(255), @NewTeamName)
```

```
SET @NewChurchName = CONVERT(NVARCHAR(255), @NewChurchName)
```

```
SET @NewDistrict = CONVERT(NVARCHAR(255), @NewDistrict)
```

```
SET @NewLeaderEmail = CONVERT(NVARCHAR(255), @NewLeaderEmail)
```

```
SET @NewLeaderName = CONVERT(NVARCHAR(255), @NewLeaderName)
```

```
SET @NewPartership = CONVERT(INT, @NewPartership)
```

```

INSERT INTO Team (TeamName, ChurchName, District, LeaderEmail,
LeaderName, Partership)
                VALUES(@NewTeamName, @NewChurchName, @NewDistrict,
@NewLeaderEmail, @NewLeaderName, @NewPartership)

END

GO

/***** Object: StoredProcedure [dbo].[DeleteCTteam]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[DeleteCTteam](
                @NewTeamID                                INT --pk, not editable
                )
AS
BEGIN
SET              @NewTeamID = CONVERT(INT, @NewTeamID)

DELETE TOP(1) FROM Team
WHERE TeamID = @NewTeamID

SELECT '[' + ...TEAM DELETED...'

END

GO

```

/\*\*\*\*\* Object: StoredProcedure [dbo].[DeleteProject] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE          PROC [dbo].[DeleteProject](  
                @NewProjectID          INT --pk, not editable  
            )
```

AS

BEGIN

UPDATE TeamDetails

SET

ProjectID = 0

WHERE ProjectID = @NewProjectID

DELETE FROM Project WHERE ProjectID = @NewProjectID

END

GO

/\*\*\*\*\* Object: StoredProcedure [dbo].[DeleteRecord] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```

CREATE          PROC [dbo].[DeleteRecord](
                @NewTeamDetailsID      INT,--pk, not editab
                @NewProjectID          INT,          -- --
                @NewTeamID              INT, --pk, not editable
                @NewTID                  INT
                --
)
AS
BEGIN
SET @NewTeamDetailsID = CONVERT(INT, @NewTeamDetailsID)
SET @NewProjectID = CONVERT(INT, @NewProjectID)
SET          @NewTeamID = CONVERT(INT, @NewTeamID)
SET @NewTID = CONVERT(INT, @NewTID)

DELETE TOP(1) FROM TeamMember
WHERE @NewTID = TeamMemberID AND @NewTID IS NOT NULL

DELETE TOP(1) FROM TeamDetails
WHERE TeamDetailsID = @NewTeamDetailsID

DELETE TOP(1) FROM Team
WHERE TeamID = @NewTeamID

```

```
DELETE TOP(1) FROM Project
WHERE ProjectID = @NewProjectID
```

```
SELECT ' [...RECORD DELETED...]'
```

```
END
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[DeleteTeam]      Script Date:
11/9/2020 6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE          PROC [dbo].[DeleteTeam](
                @NewTeamID                INT --pk, not editable
            )
AS
BEGIN
SET              @NewTeamID = CONVERT(INT, @NewTeamID)
SELECT ' [...TEAM DELETED...]'
```

```
DELETE FROM TeamMemberSkills WHERE (TeamMemberID = (SELECT TeamMemberID
FROM TeamMember WHERE TeamID = @NewTeamID))
```

```
DELETE FROM TeamMember WHERE TeamID = @NewTeamID;
```

```
DELETE FROM TeamDetails WHERE TeamID = @NewTeamID;
```

```
DELETE FROM TeamPartnerProfile WHERE TeamID = @NewTeamID;
```

```
END
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[DeleteTeamMember]    Script
Date: 11/9/2020 6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE          PROC [dbo].[DeleteTeamMember](
                @NewTeamMemberID          INT --pk, not editable
            )
```

```
AS
```

```
BEGIN
```

```
DECLARE @TEAMID INT
```

```
SET @TEAMID = (SELECT TeamID FROM TeamMember Where TeamMemberID =
@NewTeamMemberID)
```

```
DELETE FROM TeamMemberSkills WHERE TeamMemberID = @NewTeamMemberID;
```

```
DELETE FROM TeamMember WHERE TeamMemberID = @NewTeamMemberID;
```

```
UPDATE TeamDetails
```

```
SET NumTeamMemb = NumTeamMemb - 1
```

```

WHERE TeamID = @TeamID;

DECLARE @VALID INT

SET @VALID = (SELECT NumTeamMemb FROM TeamDetails Where TeamID =
@TeamID)

IF @VALID < 1

BEGIN

DELETE FROM TeamDetails WHERE TeamID = @TeamID

END

END

GO

/***** Object: StoredProcedure [dbo].[EditCTteam]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditCTteam](

editable      @NewTeamID                INT, --pk, not

               @NewTeamName            NVARCHAR(255),

               @NewChurchName          NVARCHAR(255),

               @NewDistrict            NVARCHAR(255),

               @NewLeaderEmail         NVARCHAR(255),

               @NewLeaderName          NVARCHAR(255),

```



```

        @NewPartership                INT,
        @NewCancelled                 INT
    )
AS
BEGIN
    SET @NewTeamID =          CONVERT(INT, @NewTeamID)
    SET @NewTeamName =       CONVERT(NVARCHAR(255), @NewTeamName)
    SET @NewChurchName =     CONVERT(NVARCHAR(255), @NewChurchName)
    SET @NewDistrict =       CONVERT(NVARCHAR(255), @NewDistrict)
    SET @NewLeaderEmail =    CONVERT(NVARCHAR(255), @NewLeaderEmail)
    SET @NewLeaderName =     CONVERT(NVARCHAR(255), @NewLeaderName)
    SET @NewPartership =     CONVERT(INT, @NewPartership)
    SET @NewCancelled =      CONVERT(INT, @NewCancelled)

    UPDATE Team
    SET
        TeamName      = @NewTeamName,
        ChurchName    = @NewChurchName,
        District      = @NewDistrict,
        LeaderEmail   = @NewLeaderEmail,
        LeaderName    = @NewLeaderName,
        Partership    = @NewPartership
    WHERE TeamID      = @NewTeamID;

    UPDATE TeamDetails
    SET

```

```

CancelTeam = @NewCancelled

WHERE TeamID = @NewTeamID

END

GO

/***** Object: StoredProcedure [dbo].[EditFunds]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditFunds](
                @NewTeamDetailsID          INT,--pk, not editable
                @NewProjectFundsRecieved   BIT,      --
                @NewCancelTeam              INT,  --
                --
                @NewProjectMoney            MONEY --      --
                --
            )
AS
BEGIN
SET @NewTeamDetailsID = CONVERT(INT, @NewTeamDetailsID)
SET @NewProjectFundsRecieved = CONVERT(BIT, @NewProjectFundsRecieved)
SET @NewCancelTeam = CONVERT(INT, @NewCancelTeam)
SET @NewProjectMoney = CONVERT(MONEY, @NewProjectMoney)

```

```

UPDATE TeamDetails

SET ProjectFundsReceived = @NewProjectFundsRecieved,

           CancelTeam =                @NewCancelTeam,

           ProjectMoney =                @NewProjectMoney

WHERE TeamDetailsID = @NewTeamDetailsID;

END

GO

/***** Object: StoredProcedure [dbo].[EditMemberTeam]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditMemberTeam] (

           @TMID          INT,

           @TeamID        INT

)

AS

BEGIN

UPDATE TeamMember

SET

           TeamID = @TeamID

WHERE TeamMemberID = @TMID

```

UPDATE TeamDetails

SET

NumTeamMemb = NumTeamMemb + 1

WHERE TeamID = @TeamID

END

GO

/\*\*\*\*\* Object: StoredProcedure [dbo].[EditProject] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE          PROC [dbo].[EditProject](
                @NewProjectID
                INT,
                @NewWWCoorID
                INT,
                @NewDateRecieved                DATE,
                @NewLinkToWwsite
                NVARCHAR(255),
                @NewUserPermission
                NVARCHAR(255),
                @NewProjectTypeID                INT,
                @NewDistrictID
                INT,
                @NewCityName
                NVARCHAR(255),
```

@NewPastorName  
 NVARCHAR(MAX),

@NewPastorPhone  
 NVARCHAR(MAX),

@NewLocationLink  
 NVARCHAR(255),

@NewDistanceFromDistrictOff  
 NVARCHAR(255),

@NewDistrictBudgetCurrent MONEY,

@NewDistrictBudgetLast MONEY,

@NewFEMCurrent  
 MONEY,

@NewFEMLast  
 MONEY,

@NewDateOrganized DATE,

@NewMembersLast  
 INT,

@NewMembersCurrent  
 INT,

@NewAvgAttendanceLast INT,

@NewAvgAttendanceCurent INT,

@NewProjectDesc  
 NVARCHAR(MAX),

@NewTempleCapacity  
 INT,

@NewRoofWidth  
 INT,

@NewRoofLength  
 INT,

@NewBriefDescriptionConst  
 NVARCHAR(MAX),

@NewPropertyWidth INT,

	@NewPropertyLength	
INT,		
	@NewPhotos	
INT,		
	@NewDigital	
INT,		
	@NewLotLayout	
INT,		
	@NewProjectPlans	INT,
	@NewMoneyRaised	
MONEY,		
	@NewMoneyInvested	MONEY,
	@NewMoney12Months	MONEY,
	@NewProjectNeed	
NVARCHAR(MAX),		
	@NewTitleToProperty	
INT,		
	@NewPropertyDeededToChurch	INT,
	@NewBuildingPermitNeeded	INT,
	@NewBuildingPermitApproved	INT,
	@NewPastorApproved	
INT,		
	@NewChurchBoardApproved	INT,
	@NewTreasureApproved	INT,
	@NewTreasureApprovedDate	DATE,
	@NewSuperApproval	INT,
	@NewSuperApprovalDate	DATE,
	@NewDistrictSecretaryApproval	INT,
	@NewDistrictTreasureApproval	INT,

```

        @NewDistrictTreasureApprovalDate    DATE,
        @NewDistrictPriority                INT,
        @NewTyApproved                      INT,
        @NewTyTPriority                     INT,
        @NewProjectCompleted                INT,
        @NewWwLink                          NVARCHAR(255),
        @NewPhotoLocation                   NVARCHAR(255),
        @NewPermitStart                     DATE,
        @NewPermitEnd                       DATE,
        @NewCountryID                       INT
    )
AS
BEGIN
UPDATE Project
SET
        WwCoordID                          =
        @NewWwCoordID,
        DateReceived                        =
        @NewDateRecieved,
        LinkToWwsite                        =
        @NewLinkToWwsite,
        UserPermission                       =
        @NewUserPermission,

```

@NewProjectTypeID,	ProjectTypeID	=
	DistrictID	=
	@NewDistrictID,	
	CityName	=
	@NewCityName,	
	PastorName	=
	@NewPastorName,	
	PastorPhone	=
	@NewPastorPhone,	
	LocationLink	=
	@NewLocationLink,	
@NewDistanceFromDistrictOff,	DistanceFromDistrictOff	=
	DistrictBudgetCurrent	=
	@NewDistrictBudgetCurrent,	
	DistrictBudgetLast	=
	@NewDistrictBudgetLast,	
	FEMcurrent	=
	@NewFEMCurrent,	
	FEMlast	=
	= @NewFEMLast,	
	DateOrganized	=
	@NewDateOrganized,	
	MembersLast	=
	@NewMembersLast,	
	MembersCurrent	=
	@NewMembersCurrent,	
	AvgAttendanceLast	=
	@NewAvgAttendanceLast,	
	AvgAttendanceCurr	=
	@NewAvgAttendanceCurent,	
	ProjectDesc	=
	@NewProjectDesc,	



TempleCapacity	=
@NewTempleCapacity,	
RoofWidth	=
@NewRoofWidth,	
RoofLength	=
@NewRoofLength,	
BriefDescriptionConst	=
@NewBriefDescriptionConst,	
PropertyWidth	=
@NewPropertyWidth,	
PropertyLength	=
@NewPropertyLength,	
Photos	
= @NewPhotos,	
Digital	
= @NewDigital,	
LotLayout	=
@NewLotLayout,	
ProjectPlans	=
@NewProjectPlans,	
MoneyRaised	=
@NewMoneyRaised,	
MoneyInvested	=
@NewMoneyInvested,	
Money12Months	=
@NewMoney12Months,	
ProjectNeed	=
@NewProjectNeed,	
TitleToProperty	=
@NewTitleToProperty,	
PropertyDeededToChurch	=
@NewPropertyDeededToChurch,	
BuildingPermitNeeded	=
@NewBuildingPermitNeeded,	

BuildingPermitApproved @NewBuildingPermitApproved,	=	=
PastorApproved @NewPastorApproved,	=	=
ChurchBoardApproved @NewChurchBoardApproved,	=	=
TreasureApproved @NewTreasureApproved,	=	=
TreasureApprovalDate @NewTreasureApprovalDate,	=	=
SuperApproval @NewSuperApproval,	=	=
SuperApprovalDate @NewSuperApprovalDate,	=	=
DistrictSecretaryApproval @NewDistrictSecretaryApproval,	=	=
DistrictTreasureApproval @NewDistrictTreasureApproval,	=	=
DistrictTreasureApprovalDate @NewDistrictTreasureApprovalDate,	=	=
DistrictPriority @NewDistrictPriority,	=	=
TyTApproved @NewTyTApproved,	=	=
TyTPriority @NewTyTPriority,	=	=
ProjectCompleted @NewProjectCompleted,	=	=
Wwlink = @NewWwLink,		
PhotoLocation @NewPhotoLocation,		=
PermitStart @NewPermitStart,		=

```

                PermitEnd                =
                @NewPermitEnd,

                CountryID                 =
                @NewCountryID

WHERE ProjectID = @NewProjectID

END

GO

/***** Object: StoredProcedure [dbo].[EditRecord]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditRecord](

                @NewTeamDetailsID        INT,--pk, not editable

                @NewTeamName              NVARCHAR (255),    --

                @NewNumTeamMemb           INT,                --

                @NewProjectID             INT,                --

                @NewArrivalDate           DATE,               --

                @NewDepartDate            DATE,

                --

                @NewCountryID             INT,                --

                @NewTeamID                 INT, --pk, not editable

                @NewChurchID              INT,                --

                @NewFName                  NVARCHAR (255),    --

```

```

        @NewMaxima                INT,
        --

        @NewArriveFlightNumber NVARCHAR (255), --
        @NewDepartFlightNumber NVARCHAR (255), --
        @NewCancelTeam          INT, --
        @NewProjectMoney        MONEY, --
        @NewTID                  INT
        --
    )
AS
BEGIN
    SET @NewTeamDetailsID = CONVERT(INT, @NewTeamDetailsID)
    SET @NewTeamName = CONVERT(NVARCHAR (255), @NewTeamName)
    SET @NewNumTeamMemb = CONVERT(INT, @NewNumTeamMemb)
    SET @NewProjectID = CONVERT(INT, @NewProjectID)
    SET @NewArrivalDate = CONVERT(DATE, @NewArrivalDate)
    SET @NewDepartDate = CONVERT(DATE, @NewDepartDate)
    SET @NewCountryID = CONVERT(INT , @NewCountryID)
    SET @NewTeamID = CONVERT(INT, @NewTeamID)
    SET @NewChurchID = CONVERT(INT, @NewChurchID)
    SET @NewFName = CONVERT(NVARCHAR (255), @NewFName)
    SET @NewMaxima = CONVERT(INT, @NewMaxima)
    SET @NewArriveFlightNumber = CONVERT(NVARCHAR (255),
    @NewArriveFlightNumber)
    SET @NewDepartFlightNumber = CONVERT(NVARCHAR (255),
    @NewDepartFlightNumber)
    SET @NewCancelTeam = CONVERT(INT, @NewCancelTeam)

```

```

SET @NewProjectMoney = CONVERT(MONEY, @NewProjectMoney)

IF @NewTID = NULL

        SET @NewFName = NULL

ELSE

        SET @NewFName = @NewFName

UPDATE TeamMember

SET FName = @NewFName

WHERE @NewTID = TeamMemberID AND @NewTID IS NOT NULL

UPDATE TeamDetails

SET

        NumTeamMemb = @NewNumTeamMemb,

        CountryID = @NewCountryID,

        ProjectID = @NewProjectID,

        ArriveDate = @NewArrivalDate,

        DepartDate = @NewDepartDate,

        CancelTeam = @NewCancelTeam,

        ArriveFlightNumber = @NewArriveFlightNumber,

        DepartFlightNumber = @NewDepartFlightNumber,

        Maxima = @NewMaxima,

        ProjectMoney = @NewProjectMoney

WHERE TeamDetailsID = @NewTeamDetailsID

```

```
UPDATE Team

SET TeamName = @NewTeamName

WHERE TeamID = @NewTeamID
```

```
UPDATE Project

SET ChurchID = @NewChurchID

WHERE ProjectID = @NewProjectID
```

```
Return 0
```

```
END
```

```
GO
```

```
/***** Object: StoredProcedure [dbo].[EditTeam]      Script Date:
11/9/2020 6:28:00 PM *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE          PROC [dbo].[EditTeam](
editable      @NewTeamID                INT, --pk, not
              @NewTeamName              NVARCHAR(255),
              @NewChurchName             NVARCHAR(255),
              @NewDistrict               NVARCHAR(255),
              @NewLeaderEmail            NVARCHAR(255),
              @NewLeaderName             NVARCHAR(255),
```

```

        @NewPartership                INT,
        @NewCancelled                  INT
    )
AS
BEGIN
    SET @NewTeamID =          CONVERT(INT, @NewTeamID)
    SET @NewTeamName =       CONVERT(NVARCHAR(255), @NewTeamName)
    SET @NewChurchName =     CONVERT(NVARCHAR(255), @NewChurchName)
    SET @NewDistrict =      CONVERT(NVARCHAR(255), @NewDistrict)
    SET @NewLeaderEmail =   CONVERT(NVARCHAR(255), @NewLeaderEmail)
    SET @NewLeaderName =    CONVERT(NVARCHAR(255), @NewLeaderName)
    SET @NewPartership =    CONVERT(INT, @NewPartership)
    SET @NewCancelled =      CONVERT(INT, @NewCancelled)

    UPDATE Team
    SET
        TeamName      = @NewTeamName,
        ChurchName    = @NewChurchName,
        District      = @NewDistrict,
        LeaderEmail   = @NewLeaderEmail,
        LeaderName    = @NewLeaderName,
        Partership    = @NewPartership
    WHERE TeamID      = @NewTeamID;

    UPDATE TeamDetails
    SET

```

```

                CancelTeam = @NewCancelled

WHERE TeamID = @NewTeamID

END

GO

/***** Object: StoredProcedure [dbo].[EditTeamDetails]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditTeamDetails](
                @NewTeamDetailsID      INT,--pk, not editable
                @NewTotalDailyFunds     MONEY,                --
                @NewNumTemMemb          INT,--
                @NewTeamMemberID        INT,
                @NewTeamTypeID          INT,
                @NewNumberOfDays        INT,
                @NewNumberOfNights     INT,
                @NewArriveFlightNumber  NVARCHAR(255),--
                @NewArriveTime          DATETIME2(7),
                @NewArriveDate          DATE,--
                @NewDepartTime          DATETIME2(7),
                @NewDepartDate          DATE,
                @NewDepartAirline       NVARCHAR(255),

```



```

        @NewDepartFlightNumber  NVARCHAR(255),
        @NewDistanceToRR        INT,
        @NewDistanceToProject    INT,
        @NewHotelID              INT,
        @NewCountryID            INT,
        @NewAirportID            INT,
        @NewTeamRegistration      INT, --bool
        @NewInsurance             INT, -- bool
        @NewMaxima                INT, --bool
        @NewProjectMoney         MONEY,
        @NewPartnershipID        INT,
        @NewPartnerProfileID     INT,
        @NewTeamHost              INT, --bool
        @NewCulBroker             INT, --bool?
        @NewTruckDriver           INT, --bool
        @NewBusDriver             INT, --bool
        @NewTranslator            INT, --bool
        @NewStructureDelivery     INT, --bool
        @NewCancelTeam           INT, --bool
        @NewProjectFundsReceived INT --bool
    )
AS
BEGIN
UPDATE TeamDetails

```

SET

TotalDailyFunds = @NewTotalDailyFunds,  
NumTeamMemb = @NewNumTemMemb,  
TeamMemberID = @NewTeamMemberID,  
TeamTypeID = @NewTeamTypeID,  
NumberOfDays = @NewNumberOfDays,  
NumberOfNights = @NewNumberOfNights,  
ArriveFlightNumber = @NewArriveFlightNumber,  
ArriveTime = @NewArriveTime,  
ArriveDate = @NewArriveDate,  
DepartTime = @NewDepartTime,  
DepartDate = @NewDepartDate,  
DepartAirline = @NewDepartAirline,  
DepartFlightNumber = @NewDepartFlightNumber,  
DistanceToRR = @NewDistanceToRR,  
DistanceToProject = @NewDistanceToProject,  
HotelID = @NewHotelID,  
CountryID = @NewCountryID,  
AirportID = @NewAirportID,  
TeamRegistration = @NewTeamRegistration,  
Insurance = @NewInsurance,  
Maxima = @NewMaxima,  
ProjectMoney = @NewProjectMoney,  
PartnershipID = @NewPartnershipID,

```

TeamPartnerProfileID = @NewPartnerProfileID,

TeamHost = @NewTeamHost,

CulBroker = @NewCulBroker,

Translator = @NewTranslator,

StructureDelivery = @NewStructureDelivery,

CancelTeam = @NewCancelTeam,

ProjectFundsReceived = @NewProjectFundsReceived

WHERE TeamDetailsID = @NewTeamDetailsID

END

GO

/***** Object: StoredProcedure [dbo].[EditTeamMember]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[EditTeamMember](

                @EMTID                                INT,

                @EMFName                                NVARCHAR(255),

                @EMPhoneNumber                            NVARCHAR(255),

                @EMCountry                                NVARCHAR(255),

                @EMDistrict                              NVARCHAR(255),

                @EMPreviousProjects                      INT,

```

```

        @EMSecondLanguage          NVARCHAR(50),
        @EMSkillsID                INT,
        @EMTeamMemberTypeID        INT,
        @EMTeamLeader              INT,
        @EMTeamID                  INT,
        @EMLastTrip                DATE,
        @EMMale                    INT,
        @EMFemale                  INT,
        @EMSpouseOnTeam            INT,

        @EMPassPortExp             DATE,
        @EMFoodAllergies           NVARCHAR(255),
        @EMMedicalAllergies        NVARCHAR(255),

        @EMDPI                      NVARCHAR(255),
        @EMEmail                    NVARCHAR(255)
    )
AS
BEGIN --This record exists, update it
    UPDATE TeamMember
    SET
        FName                      = @EMFName,

        PhoneNumber                 = @EMPhoneNumber,

        Country                     = @EMCountry,

        District                    = @EMDistrict,

```

```

PreviousProjects      = @EMPreviousProjects,
SecondLanguage        = @EMSecondLanguage,
SkillsID              = @EMSkillsID,
TeamLeader            = @EMTeamLeader,
LastTrip              = @EMLastTrip,
Male                  = @EMMale,
Female                = @EMFemale,
SpouseOnTeam          = @EMSpouseOnTeam,
PassPortExp           = @EMPassPortExp,
FoodAllergies         = @EMFoodAllergies,
MedicalAllergies     = @EMMedicalAllergies,
DPI                   = @EMDPI,
Email                 = @EMEmail

WHERE TeamMemberID = @EMTID

IF ((SELECT TeamID FROM TeamMember WHERE TeamMemberID = @EMTID) !=
@EMTeamID)

BEGIN

UPDATE TeamDetails

        SET NumTeamMemb = NumTeamMemb - 1

WHERE TeamID = (SELECT TeamID FROM TeamMember WHERE TeamMemberID =
@EMTID)

UPDATE TeamDetails

        SET NumTeamMemb = NumTeamMemb + 1

WHERE TeamID = @EMTeamID

UPDATE TeamMember

        SET TeamID = @EMTeamID

```

```

WHERE TeamMemberID = @EMTID

RETURN 0

END

END

GO

/***** Object: StoredProcedure [dbo].[ExpectedFunds]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[ExpectedFunds]

AS

BEGIN

SELECT DISTINCT TeamDetails.TeamDetailsID,

                TeamDetails.ProjectFundsReceived,

                TeamDetails.CancelTeam,

                TeamDetails.ProjectMoney,

                TeamDetails.ArriveDate,

                Team.TeamName,

                Team.TeamID

FROM (Team

      INNER JOIN TeamDetails ON Team.TeamID =

TeamDetails.TeamID) WHERE

                (((TeamDetails.ProjectFundsReceived)=0) AND

((TeamDetails.CancelTeam)=0));

```

```

END

GO

/***** Object: StoredProcedure [dbo].[ProjectsByDistrict]    Script
Date: 11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE      PROC [dbo].[ProjectsByDistrict]
AS
BEGIN
SELECT
                [SSMA$District$local].[DistrictName],
                [TeamDetails].[ArriveDate],
                [Country].[CountryName]

FROM Country

INNER JOIN (TeamDetails RIGHT JOIN (SSMA$District$local INNER JOIN
Project ON SSMA$District$local.DistrictID = Project.DistrictID)
                ON TeamDetails.ProjectID = Project.ProjectID)
                ON Country.CountryID = SSMA$District$local.CountryID

GROUP BY SSMA$District$local.DistrictName, TeamDetails.ArriveDate,
Country.CountryName

HAVING (TeamDetails.ArriveDate)>(DATEADD(day, -15, GETDATE()))

END

GO

```

/\*\*\*\*\* Object: StoredProcedure [dbo].[ProjectTeamTables] Script  
Date: 11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROC [dbo].[ProjectTeamTables]

AS

BEGIN

DROP TABLE IF EXISTS ##ProjectMoneyRequested;

--projectMoneyRequested query text

SELECT DISTINCT Project.ProjectID AS ProjectID, Sum(CheckSolicitud.USD)  
AS SumOfUSD, CheckSolicitud.SolicitudDate AS SD

INTO ##ProjectMoneyRequested

FROM TeamDetails INNER JOIN (CheckSolicitud INNER JOIN Project ON  
CheckSolicitud.ProjectID = Project.ProjectID)

ON TeamDetails.ProjectID = CheckSolicitud.ProjectID

GROUP BY Project.ProjectID, CheckSolicitud.SolicitudDate

--SELECT \* FROM #Project

SELECT

TeamDetails.TeamDetailsID AS TeamDetailsID,

Team.TeamName AS TeamName,

TeamDetails.NumTeamMemb AS NumTeamMemb,

TeamDetails.ProjectID AS ProjectID,

TeamDetails.ArriveDate AS ArrivalDate,

TeamDetails.DepartDate AS DepartDate,



```

Country.CountryName AS CountryName,
Team.TeamID AS TeamID,
Project.ChurchID AS ChurchID,
TeamMember.FName AS FName,
TeamMember.TeamMemberID AS TID,
TeamDetails.Maxima AS Maxima,
TeamDetails.ArriveFlightNumber AS ArriveFlightNumber,
TeamDetails.DepartFlightNumber AS DepartFlightNumber,
TeamDetails.CancelTeam AS CancelTeam,
##ProjectMoneyRequested.SumOfUSD AS SumOfUSD,
TeamDetails.ProjectMoney AS ProjectMoney

FROM (TeamMember RIGHT JOIN (((Team INNER JOIN
TeamDetails ON Team.TeamID = TeamDetails.TeamID)
LEFT JOIN Country ON TeamDetails.CountryID = Country.CountryID)
LEFT JOIN Project ON TeamDetails.ProjectID = Project.ProjectID)
ON TeamMember.TeamMemberID = TeamDetails.TeamHost)
LEFT JOIN ##ProjectMoneyRequested ON Project.ProjectID =
##ProjectMoneyRequested.ProjectID

WHERE (((TeamDetails.ArriveDate)>(DATEADD(day, -15, GETDATE()))) And
((TeamDetails.Maxima) = 0))

OR

(((TeamDetails.ArriveDate)>(DATEADD(day, -15,
GETDATE()))) And (TeamDetails.CancelTeam = 0) And ((TeamDetails.Maxima) =
1)))

ORDER BY TeamDetails.ArriveDate;

END

```

```

GO

/***** Object: StoredProcedure [dbo].[registerUser]      Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE      PROC [dbo].[registerUser] (

            @newUser  NVARCHAR(64),

            @newPass  NVARCHAR(64),

            @newEmail NVARCHAR(64))

AS

BEGIN

IF EXISTS(

            (SELECT TOP(1)

            [userID],

            [userName],

            [userPassword],

            [userEmail],

            [userClearanceLevel],

            [userSalt]

            FROM userInfo

            WHERE

            userName = @newUser

            ))

```

```

        BEGIN
            SELECT('This Username is in use. Please
try a new username')

            RETURN 0 --Fail

        END

ELSE

        BEGIN --create salt

            -- salt and resulting pass variables
            DECLARE @SALT          NVARCHAR(64);
            DECLARE @RESULT        NVARCHAR(64);
            DECLARE @RESCONVERTER  VARBINARY(64);
            DECLARE @RESCONVERTERPASS  VARCHAR(64);

            --variables for the salt

            DECLARE @SEED      INT;
            DECLARE @REP      INT;
            DECLARE @NOW      DATETIME;

            SET @NOW =
CONVERT(DATETIME, DATEADD(DAY, 4, GETDATE()));

            SET @SEED = (DATEPART(hh, @NOW) *1000) +
(DATEPART(mi, @NOW) *10) + (DATEPART(s, @NOW) *1000000) + (DATEPART(ms,
@NOW) * 10000)

            SET @SALT = CHAR(ROUND((RAND(@SEED) *
87.0)+ 64, 3));

            SET @REP = 1;

            WHILE (@REP < 50)

```

```

BEGIN
    SET @SALT = @SALT +
CHAR(ROUND((RAND() * 87.0) , 64, 3));
    SET @REP = @REP + 1;
END;

SET @RESULT = @newPass + @SALT;
SELECT @RESULT;

SET @RESCONVERTER =
(HASHBYTES('SHA2_512',@RESULT));

SELECT @RESCONVERTER;

INSERT INTO userInfo(userSalt, userName,
userPassword, userEmail, userClearanceLevel)
VALUES(@SALT, @newUser, @RESCONVERTER,
@newEmail, 0)

RETURN 1 --Success
END

```

END

GO

/\*\*\*\*\* Object: StoredProcedure [dbo].[ViewAllMembers] Script Date:  
11/9/2020 6:28:00 PM \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

```

GO

CREATE          PROC [dbo].[ViewAllMembers]

AS

BEGIN

SELECT * FROM TeamMember WHERE TeamMemberID != 0

END

GO

/***** Object: StoredProcedure [dbo].[ViewProjects]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[ViewProjects]

AS

BEGIN

SELECT * FROM Project WHERE ProjectID != 0

END

GO

/***** Object: StoredProcedure [dbo].[ViewTeamMembers]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[ViewTeamMembers](

```

```

@TeamID                INT
)
AS
BEGIN
SELECT * FROM TeamMember WHERE TeamID = @TeamID
END
GO

/***** Object: StoredProcedure [dbo].[ViewTeams]    Script Date:
11/9/2020 6:28:00 PM *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE          PROC [dbo].[ViewTeams]

AS

BEGIN

SELECT * FROM Team WHERE TeamID != 0

END

GO

USE [master]

GO

ALTER DATABASE [mission2Go] SET  READ_WRITE

GO

```