Alex Zorola

Advisor: Dr. Dongchul Kim

CSCI-4390-03-Fall 2019

12/11/2019

On The Luxury’s

**Motivation:**

I have always enjoyed music and I have an interest in web development; however, my knowledge and experience with it is at a beginner level. Therefore, it was more beneficial for me as a student to learn how to build something that already exists and learn some of the fundamentals of web development from it. So, I decided to use Spotify as a foundation to create my music streaming service from scratch using various tools and languages which will eventually help me build future projects, such as, a music website that producers can use to sell their beats/instrumentals to artists and perhaps vice versa.

**Tools and Languages Used (Sources for definitions can be found in the citations page):**

* **XAMPP (X: Cross-Platform + Apache + MySQL + PHP + Perl)**
  + “XAMPP is an all-around package that comes with Apache HTTP server, MariaDB (now replaced by MySQL) database, and interpreters for scripts written in the PHP and Perl programming languages.”
* **PHP:**
  + PHP: Hypertext Preprocessor is a scripting language that is widely used to create dynamic Web pages. Combining syntax from the C, Java and Perl languages, PHP code is embedded within HTML pages for server-side execution. It is commonly used to extract data out of a database on the Web server and present it on the Web page. Originally known as "Personal Home Page," PHP is supported by all Web servers and widely used with the MySQL database.
* **Javascript:**
  + A programming language commonly used in web development. A client-side scripting language, which means the source code is processed by the client’s web browser rather than on the web server. Like server-side scripting languages, such as [PHP](https://techterms.com/definition/php) and [ASP](https://techterms.com/definition/asp), JavaScript code can be inserted anywhere within the [HTML](https://techterms.com/definition/html) of a [webpage](https://techterms.com/definition/webpage). However, only the [output](https://techterms.com/definition/output) of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate [.JS](https://fileinfo.com/extension/js) file, which may also be viewed in a browser.
* **jQuery:**
  + jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.
* **MySQL:**
  + Is an Oracle-backed open source relational database management system based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX, and Windows. Most associated with web applications and online publishing.

**Features:**

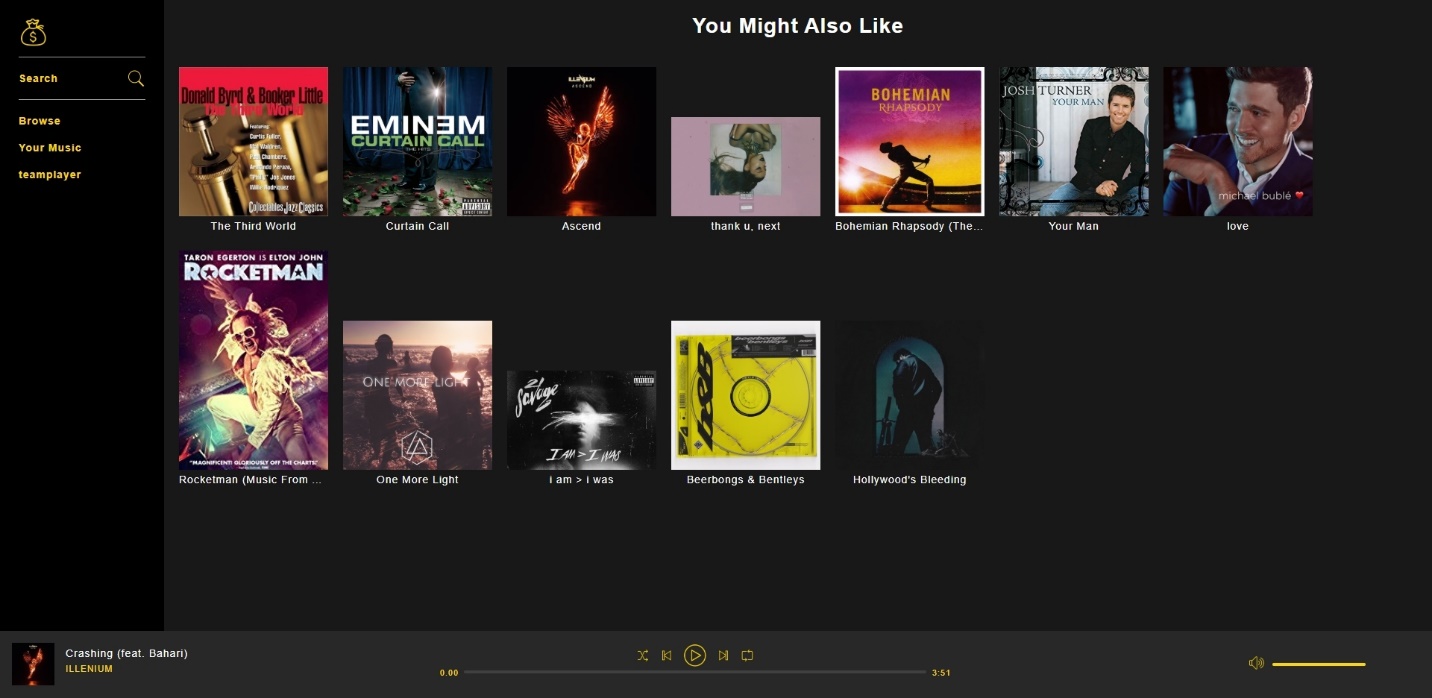
Features found within the application include; Login & Register functionality, Password Encryption, a Music Player, a Navigation Section, the main Browse Page, the ability to create and delete customizable playlists. In the individual songs there is an options menu that allows the user to add the desired song to their custom playlist. In their profile the user can update their email or password stored in the database.

**Registration:**



This feature can come with various checks and restrictions pertaining to the email and the password. My application does not contain major restrictions, only checks to see if the fields are not being used by another user, the validation of the email, and require the password to be between 5 and 30 characters. Password Encryption is a must have attribute when implementing a Login and Registration feature to a wesite or web application. For this project MD5 was used, which stands for message-digest algorithm, it is a widely used hash function producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It is a fast and memory-conserving algorithm, meaning that a hacker can compute the hash of a large number of passwords per second. A string will always map to a certain hash, for example, alex123 will always map to **b75bd008d5fecb1f50cf026532e8ae67, it will never be different, where as other password encryption methods are. This method of “encryption” is enough for educational development purposes but should not be used in actual public projects, an encryption method like BCRYPT would be better because it is slower. Therefore, the system needs more CPU cycles to authenticate users, forcing an attacker to have a lot more processing power to try and brute-force passwords should they obtain your database.**

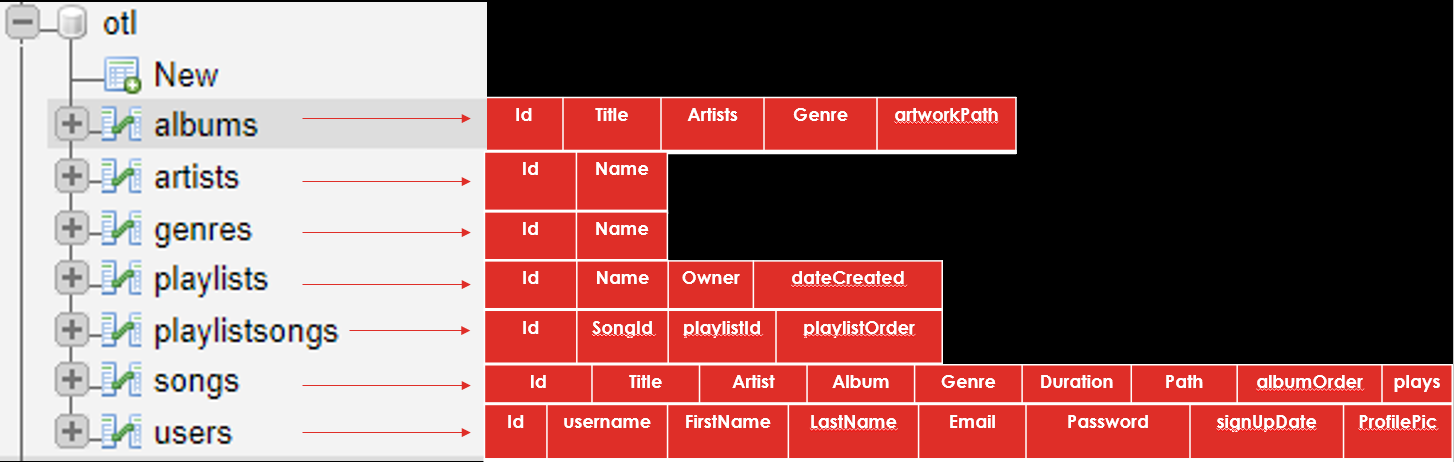


After successfully registering and logging into the page the user is met by a Browser Page, Navigation Bar and a Music Player. 

When the page loads for the first time the database is queried to create a playlist of ten random songs that the user can begin listening to before they decide to explore the application further. They can listen to this play list by interacting with the fully functioning music player found at the bottom of the page. The Navigation Bar can be found on the left side of the page, this element contains hyperlinks to other pages within the application; The search page, the Browse page (which is the current page shown above), the users music page which allows them to create, delete and customize their playlist, and the users profile page where they can change their email and password, as well as log out of the page.

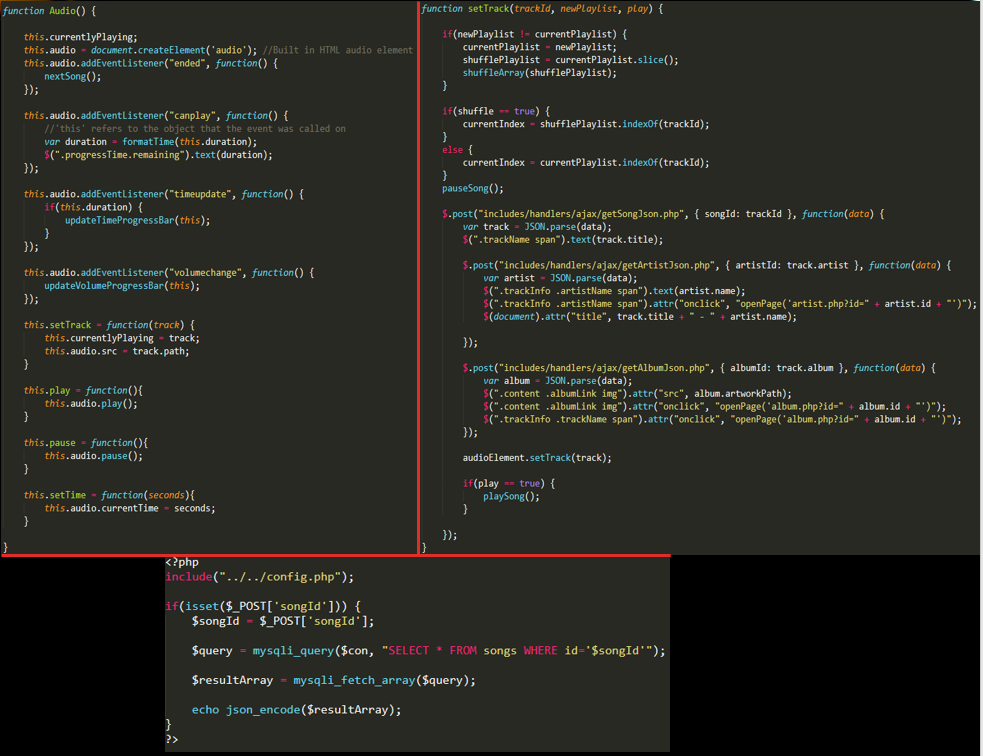
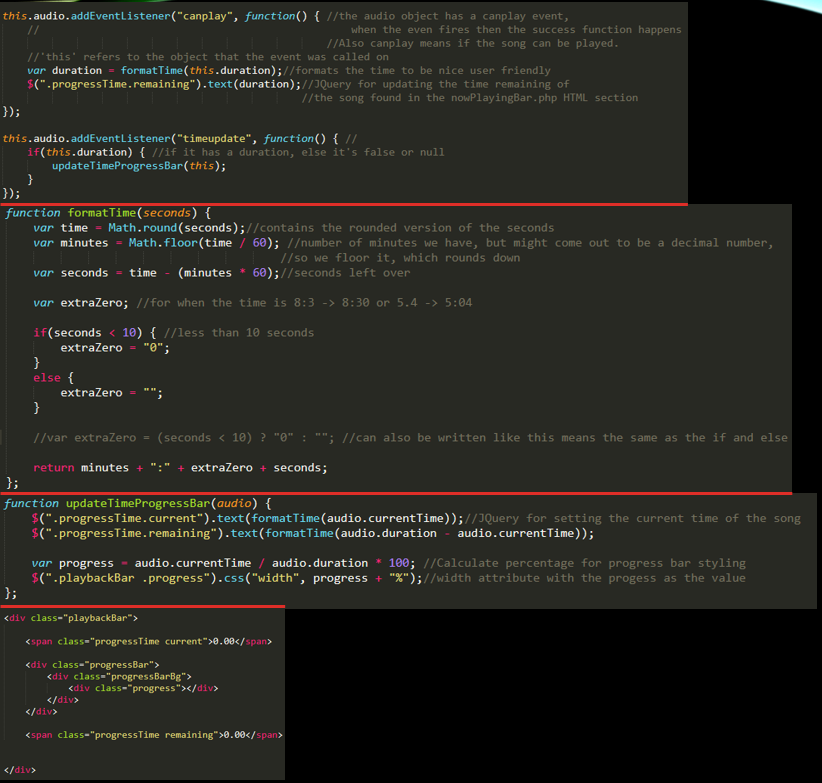
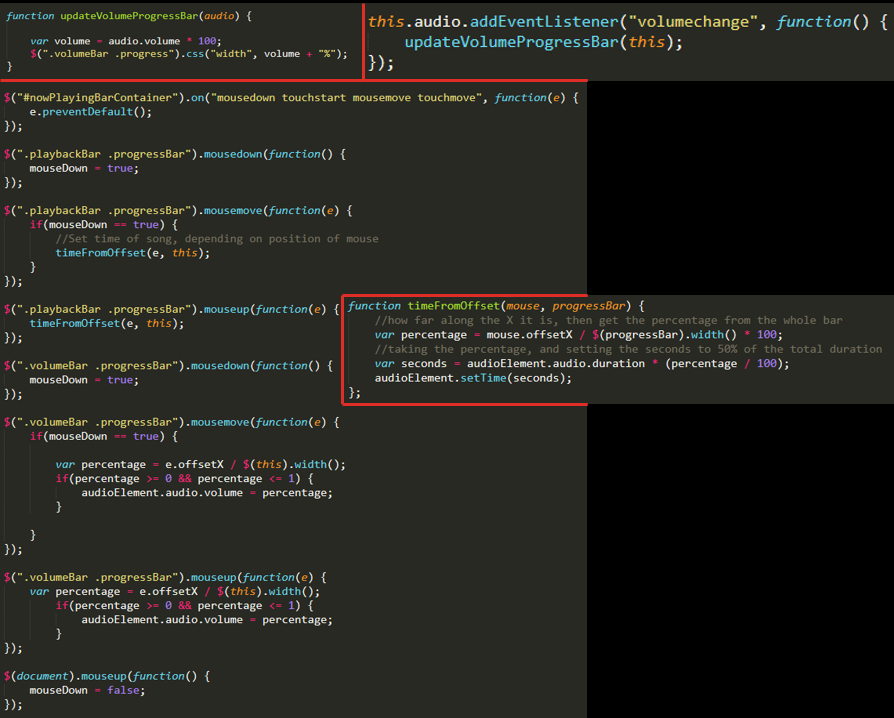
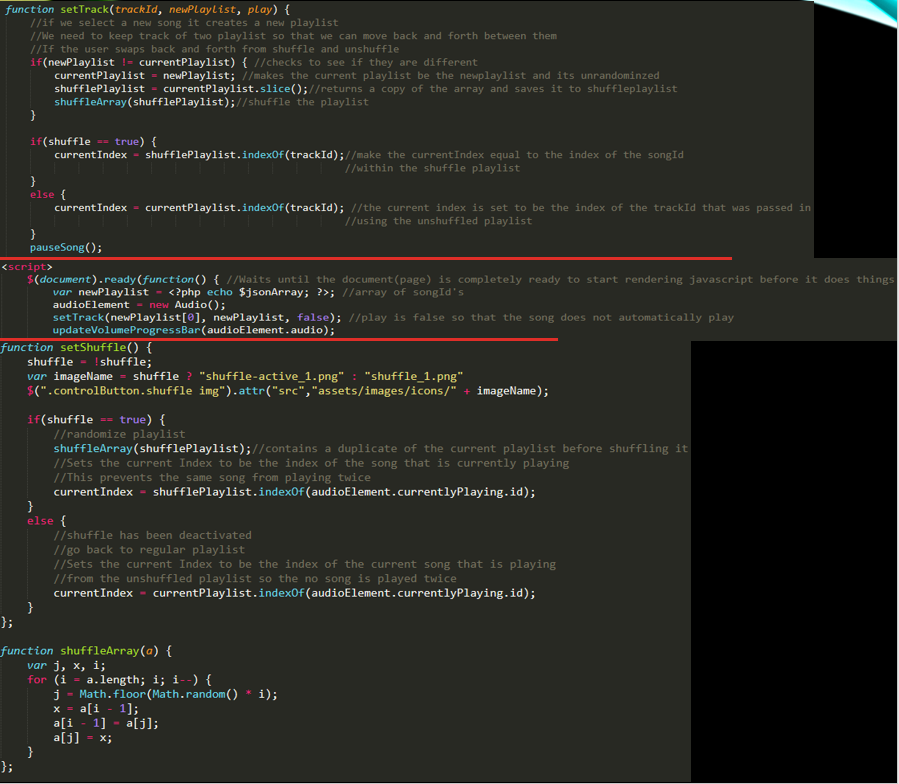
**Database Structure:**

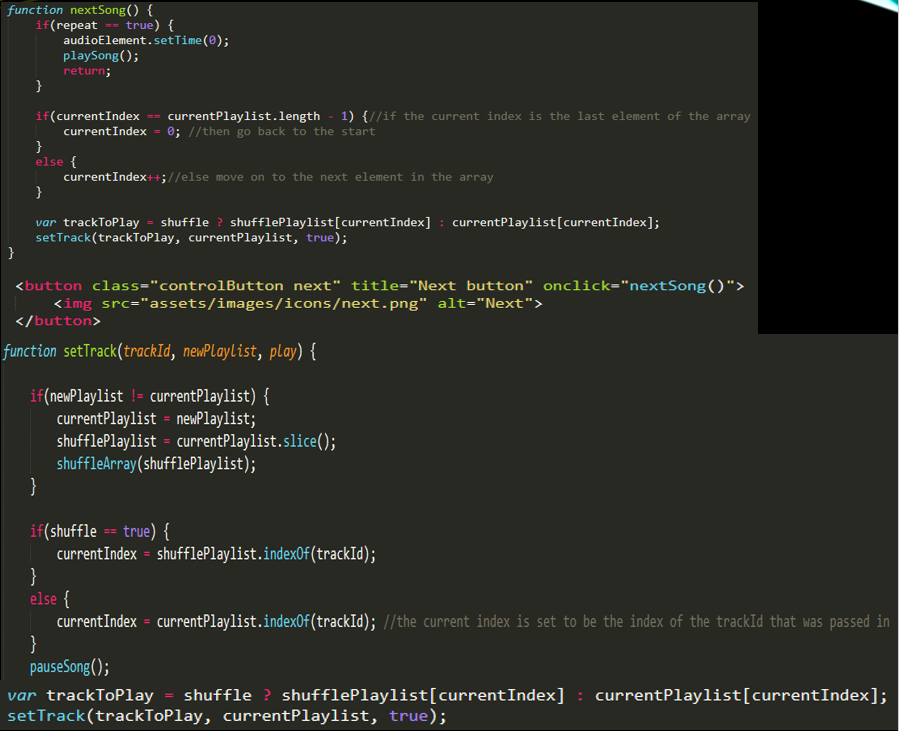
Databases consist of the actual database name which is used within your code to set up a connection to the database and be able to access the data within it. Now, within the database there are the tables which will hold specific data that is entered it. For this project the database name was “otl”, an abbreviation of the web application name “On The Luxury’s”, it consists of seven tables; albums, artists, genres, playlists, playlistsongs, songs and users. Each table has its respective columns which represent what the data is (shown below). Some tables hold foreign keys, meaning they hold the ID of the respective information so that it can be accessed through that table. In the songs table there are artist, album and genre columns, accessing the songs table and retrieving these column allows us to access the information in other tables by using the ID that was returned (i.e.. we can get the artist name, album name ect…). XAMMP allows to have this MySQL database accessible through PHPMYADMIN, another feature found within the package, and insert, update and delete data from the database.

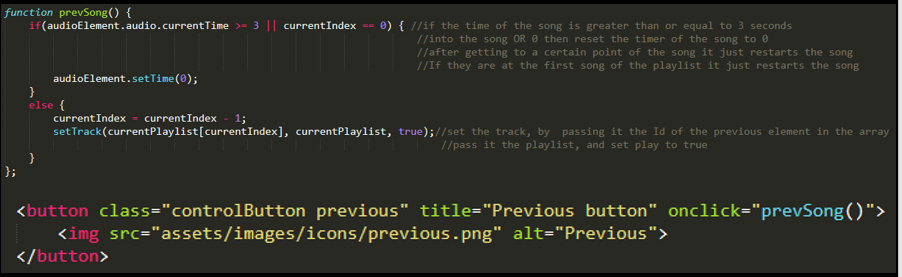


**Music Player (NowPlayingBar):**

The most important feature of a music streaming service is the Music Player, because it handles all the music playing functionality and without it the application is essentially just a music library that the user can interact with but never actually being able to listen to any of the music.

* **SET TRACK:** Various steps go towards being able to play a song. JavaScript is used to handle all the functions going into the music player. So, to begin make a Script.js file which will be used by another file containing the music player, NowPlayingBar.php. Within the Js file a class called “Audio” is made that will hold the “this.audio” variable. This Variable is a built in HTML audio element that comes with a multitude of attributes, such as, some being the play and pause attributes. Moving to the NowPlayingBar.php file create a script tag in the html that will contain JavaScript functions it can be used in the HTML. The function setTrack will handle the retrieval of the songs and any other information that may be needed through AJAX call from another file. The problem is that PHP is used to query the database and retrieve the information, but we can’t use PHP within the functions because our functions are written in JavaScript and PHP renders as soon as the page loads. A workaround is to encode the data returned into JSON (JavaScript Object Notation) and then within the setTrack function parse the data and retrieve an object containing all the data needed. Finally, the song is set by calling the setTrack function within the audio class that takes a track object.
* **SONG PROGRESSION:** This feature is what handles the calculation of the time remaining in the song, how much time of the song has passed and formatting the time to be user friendly. Create an Event Listener within the audio class to check to see if the current track can be played. Then call the “formatTime” function by passing it the duration of the song, the track object contains this information. Within the “formatTime” function round the time in seconds, convert it to minutes and calculate the remaining seconds. Afterwards, just make it look aesthetically pleasing by adding an extra zero if needed, for example, before the song would be “8:3” instead of “8:30” or “5:4” instead of “5:04” this function fixes that. However, the actual updating of the time within the HTML is handled by another function called “UpdateTimeProgressBar” using jQuery change the text attribute of any element with the class name “progressTime.current”, “progressTime.remaining”. 
* **SONG BAR & VOLUME CONTROL:** Another Event Listener is needed to check for a change in volume within the browser then call the “updateVolumeProgressBar” function found within the NowPlayingBar.php file. This function checks how the user interacts with either the song bar or the volume bar from clicking on it. On that instance calculate the percent at which the user let go regarding the total width of the bar, then calculate the number of seconds that percentage is in relation to the duration of the song. Do the same with the volume bar, however in the jQuery functions for ”mousemove” and “mouseup” we fix a bug where the user moves the volume outside of the range. To fix this only set the volume with the percentage if its within the 0 to 1 range nothing less and nothing more. 
* **SHUFFLE PLAYLIST:** This functionality takes in a lot more elements and steps to be able to shuffle the playlist. First, create a variable in the Script.js file called “shuffle” and set it to false as a default. This will handle the “on” and “off” feature of the button in the music player and determine whether to shuffle the playlist or not. Create a “shufflePlaylist” array to keep track of the shuffled array and a currentIndex variable to keep track of the index within the array of a songs ID. As stated, before when the page first loads, we get an array of 10 song ID’s that are used to play the songs. In a “document.ready” function set that array to be equal to a new array called “newPlaylist”, which is sent to the setTrack function to be used to give the “currentPlaylist” and “shufflePlaylist” values. These two arrays are essential because one contains the unshuffled version of the array and one contains the shuffled version of the array, because it is expected for the user to bounce back and forth from them. There is a function called “shuffleArray” which handles the logic of shuffling an array, and the other function is called “setShuffle” which makes the current playlist into the shuffled playlist and vice versa due to the users’ interaction with the button. The current Index is important because we don’t want the same song to play twice when the shuffle functionality is being turned off and on. 
* **NEXT SONG:** This function essentially just updates the current Index variable we have that keeps track of the index within the array of a song that is currently playing.
  + **0 (index) :** “2” (song ID)
  + **1 (index) :** “54” (song ID)
  + **2 (index) :** “15” (song ID)

Using this check to see if the current index is the last element in the array or not, if it is then set it to be the first element in the array, otherwise go to the next element and set the track. 

* **REPEAT SONG:** Create a variable named “repeat” and default it to false in the Script.js file. In the “nextSong” function if repeat is true (the user pressed the button) then set the time of the currently playing song to 0 and play the song. The setRepeat function handles the change of the image based on the user interaction with the button, if it is “on” the button will change to green, if it is “off” the button will go back to the default gold color. 
* **PREVIOUS SONG:** This feature works the same as the next song one, but in the opposite direction. The only difference is to check to see how much time has passed since the song first started playing, because in most music streaming services depending on how much time has passed the music player will either restart the song or go to the previous song in the array. Restart the song by setting the time to zero if the song is equal to or greater than 3 seconds or if it is the first song in the playlist. Otherwise, set the current index to be the previous element in the playlist and pass it into the set track function to play the song. The rest is just hooking it onto the button like the other functions, so that it can be used.
* **PLAY AND PAUSE SONG:** There is not much to this functionality since the audio variable we created comes with these attributes which we can just call and use them with their respective buttons. Hooking them onto the buttons we just add jQuery for aesthetic reasons, if the play button is pressed, then hide the pause button and vice versa.
* **DIFFICULTIES:**
  + After working on this project, I tried to transfer it from localhost to an actual wed server using the Google Cloud Platform and Bitnami. I was able to create an account for both services, link them, create a virtual machine that would host the application and upload the files to the server. However, I was unable to get the server database to link and talk to my local database through PHPMYADMIN, thus making my effort futile.
  + Entering data into a database manually and all at once is not a good practice or a good idea. I overestimated my abilities by trying to enter 181 songs, which caused me to make some mistakes along the way. When I tried to press the save button on the table, due to the errors, the data was not entered. I clicked something I should not have and ended up clearing all the data I entered forcing me to do it again, but around 20 songs at a time this time around.
  + Entering information containing accented letters, like Michael Bublé, would display the wrong song and not return the artists name. I later found out it was because of the accent in his name and the way to fix that is by changing the “char set” within the MySQL database you are using, however I didn’t personally do this since it is a small detail.

**Conclusion:**

Working on this project I had the opportunity of learning a multitude of things; PHP, JavaScript, jQuery, AJAX and using a MySQL database. Foundamental elements of the languages can come in handy, such as, the fact that PHP renders as soon as the page loads, so workarounds for this are created, like using JSON. As I stated before I have an interest in web development and would like to pursue a career in it or something related to it, like Game Development. Having both logical elements and creative elements is why this field is so interesting to me and this project is a good gateway into learning the fundamentals of working on a project like this one. For example, the process of splitting up the project into individual parts and working on them seperately, while maintaining your files well organized. For projects like On The Luxury’s that are streaming services that require database queries, it’s mainly about knowing where to put the information you are getting and how to link it together with other elements in the project.

Citiation Page

“Apache Friends.” *Apache Friends RSS,* [www.apachefriends.org/index.html](http://www.apachefriends.org/index.html).

“JavaScript.” *JavaScript Definition*, techterms.com/definition/javascript.

JS Foundation. *JQuery*, jquery.com/.

“PHP.” *PHP Dictionary Definition | PHP Defined*, [www.yourdictionary.com/php](http://www.yourdictionary.com/php).

Rouse, Margaret, and Lindsay Moore. “What Is MySQL? - Definition from WhatIs.com.” *SearchOracle*, searchoracle.techtarget.com/definition/MySQL.

<https://www.youtube.com/watch?v=xr6q_UdTzsQ&feature=youtu.be> (Link to video I created for the project)