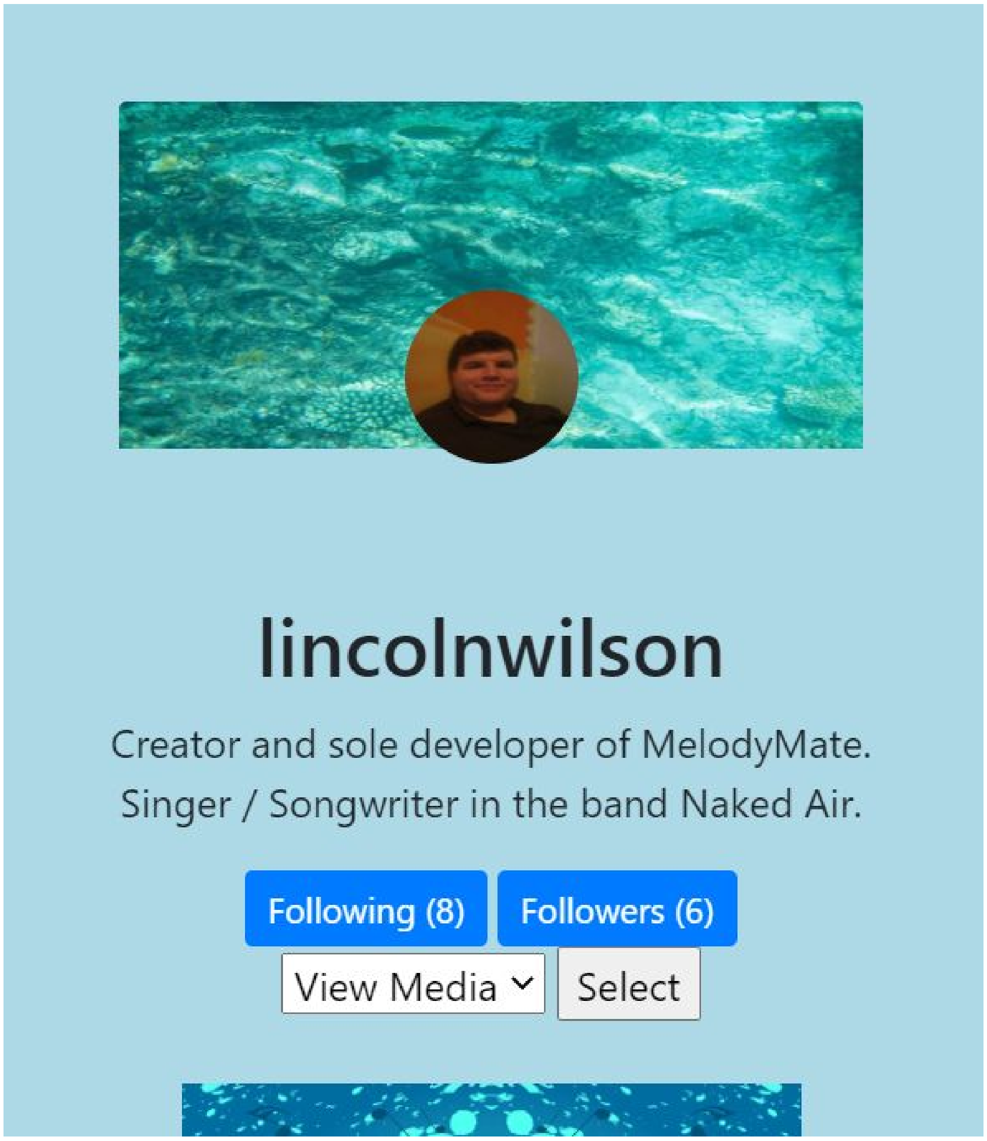
MelodyMate


lincolnwilson

Creator and sole developer of MelodyMate. Singer / Songwriter in the band Naked Air.

Social Media / Music Sharing Platform

http://34.72.223.127/melodymate/

Created By: Lincoln Wilson

Initial Background and Project Conception

Local musicians will be given the ability to share their music and make connections via the internet on one single platform. Currently, musicians and the music industry struggle as music becomes ever so more accessible and live shows become more of a scarcity. We are living through a global pandemic and for now, musicians cannot do live performances, severely cutting income. Many musicians have no place to go or share their music and connect with other musicians and fans. Ultimately, this website will be a music sharing service for local musicians that also functions as social media, but it will also give people a place to go and delve into their local music scene in a time where that is not possible.

This website will have a user and music database which could hold user information and data relating to musicians and their media. It will contain a social media service that will allow users to create their own personalized home page where they can connect their music and share their content. Their homepage should contain a feed which displays content and posts of users they added. Currently, the music industry has a focus on singles rather than full album releases. This has been a major upset to the industry and bad for musicians. Pages will have an album format focus regarding that, highlighting user albums/playlists at the forefront. Users should be able to switch between what content they want highlighted, such as singles and videos, and give an ability to toggle between and view without impacting the highlighted media as desired. This website could also contain a private messaging system, allowing users to chat with each-other.

Potential users include musicians, bands, music curators, and music professionals. Musicians can post content and media, view content and media, and share content and media. Bands can do the same, but through an initial musician login. Music curators can post content, view content and media, and share content and media (of musicians/bands). Music professionals can post content, view content and media, and share content and media; they can also make the media of musicians they are connected to appear on their profile. For example: A music professional could be a promoter for a local record company or just the page for that company. The media of the musicians and bands signed to them could be highlighted like musician and band pages.

User Stories

1. As a musician, I want to be able to search through music professionals and other musicians to make connections to better promote and/or sell my music.
2. As a band member, I would like to create a page for my band and be able to post as my band using my musician account.
3. As a concertgoer, although at this time there are no shows because of the pandemic, I would like to in the future be able to view upcoming shows of musicians and bands I like.
4. As a music professional, I would like to be able to search through and message musicians and bands in my area to scout potential business endeavors.
5. As a social media influencer, I would like to utilize this website to create a fanbase and communicate with that fanbase.

Use Cases

Make a comment on a user's post:

Use Case –

1. User searches for other users using the search bar and selects users they want to comment on.
2. User scrolls down other users' profiles and selects a post they wish to comment on.
3. User types in comment and click reply.
4. Users receive notification indicating a response to their comment.
5. Users can select notification and decide if they wish to continue communications.

Change password:

Use Case 1 –

1. User logs in and selects settings.
2. User types in their old password and new password and selects submit.
3. User receives a confirmation email requesting to select a link to confirm.
4. User confirms, changing their password.

Use Case 2 –

1. User selects "Forgot My Password" on login page.
2. User is given a security question they answered when creating their account, which they fill out and submit.
3. User is sent an email with a code to confirm their password reset.
4. User enters code on the page and enters a new password, and submits, changing their password.

Use Case 3 (Fraudulent User) –

1. Fraudulent User selects "Forgot My Password" on login page.
2. Fraudulent User is given a security question they do not know the answer to.
3. Fraudulent User enters a wrong answer and the real user is sent an email notifying them someone tried accessing their account.
4. Fraudulent User is given a notification telling them they are locked out of the account for an allotted amount of time depending on the number of tries.

Post album:

Use Case 1 –

1. Users select Add Media on their homepage.
2. Users are brought to a dialog where they enter the type of media they want posted, album.
3. Users are brought to a new dialog where they add album and track information for the songs on the album, as well as upload the files in proper format for those songs.
4. User uploads an image in square format to be the album art.
5. Users submit an album, appear on homepage and are posted to appear on their follower's feeds.
6. Users receive notification confirming their album has been posted.

Use Case 2 –

1. Users select Add Media on their homepage.
2. Users are brought to a dialog where they enter the type of media they want posted, album.

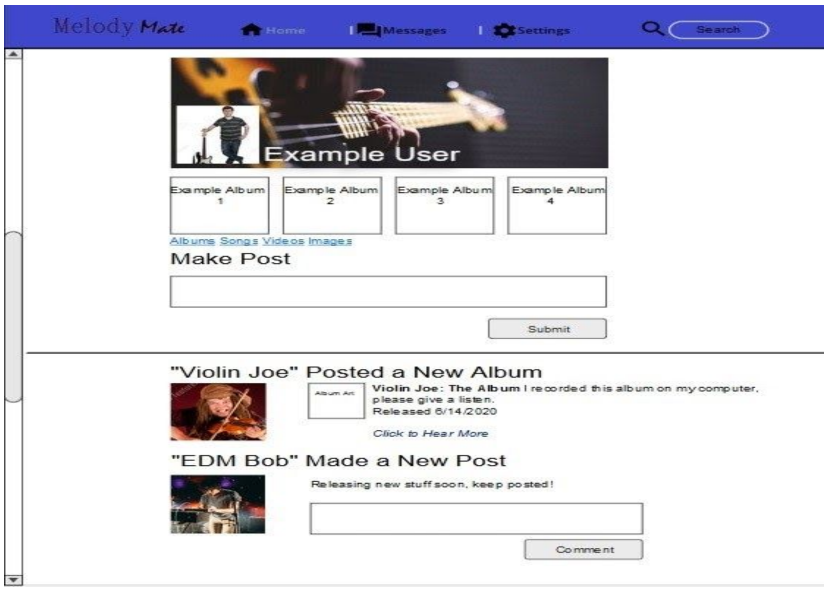
3. Users are brought to a new dialog where they add album and track information for the songs on the album, they upload the files for those songs as FLAC, which is unsupported, ignoring a warning that the file should be mp3 or wav..
4. User uploads an image in square format to be the album art.
5. User submits and is given a notification telling them they used the wrong file format.

Initial User Interface Design

This initial design features a home page, a messages page, and a settings page, as well as an example user page that is searched. There was to be a landing page for account creation and logging in and other features such as uploading and notifications which was to be further designed and implanted in the future.

This website was to be navigated using the options on the top bar as well as the search bar. Users can scroll through their feed to view posts of users they follow sorted by post date/time. The design was intended to be simple but purposeful. Things that were to be built upon beyond this design included the ability to add media to their page. There was to be a SQL database for users, media, posts, and messages. The goal was to create the code with PHP, HTML, CSS, and some JavaScript.

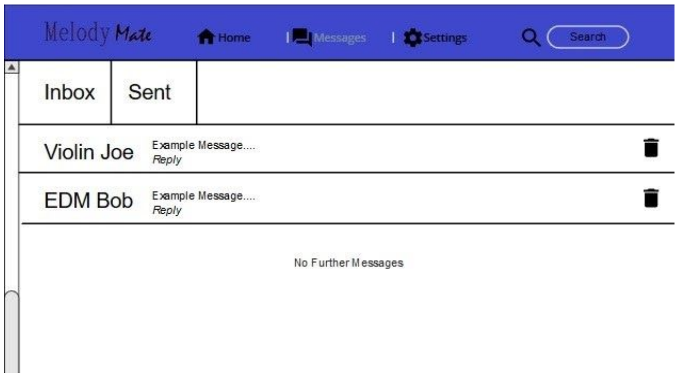
Home Page:



Violin Joe: The Album I recorded this album on my computer, please give a listen.
Released 6/14/2020

Releasing new stuff soon, keep posted!

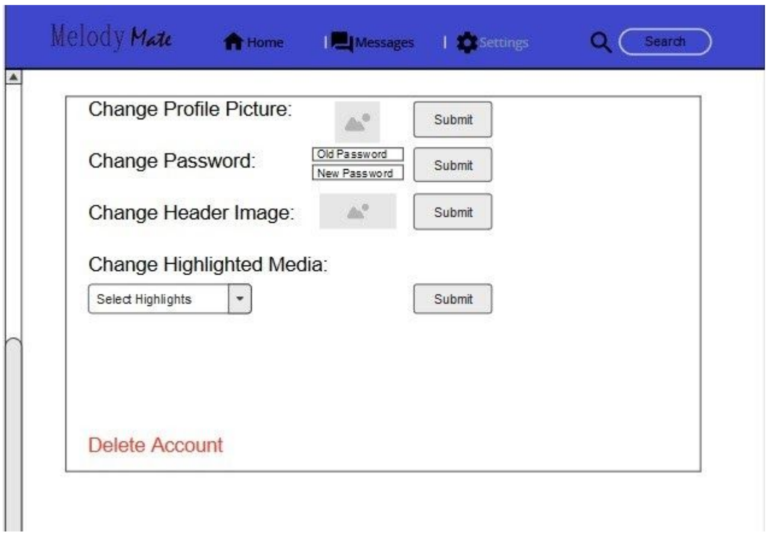
Messages Page:



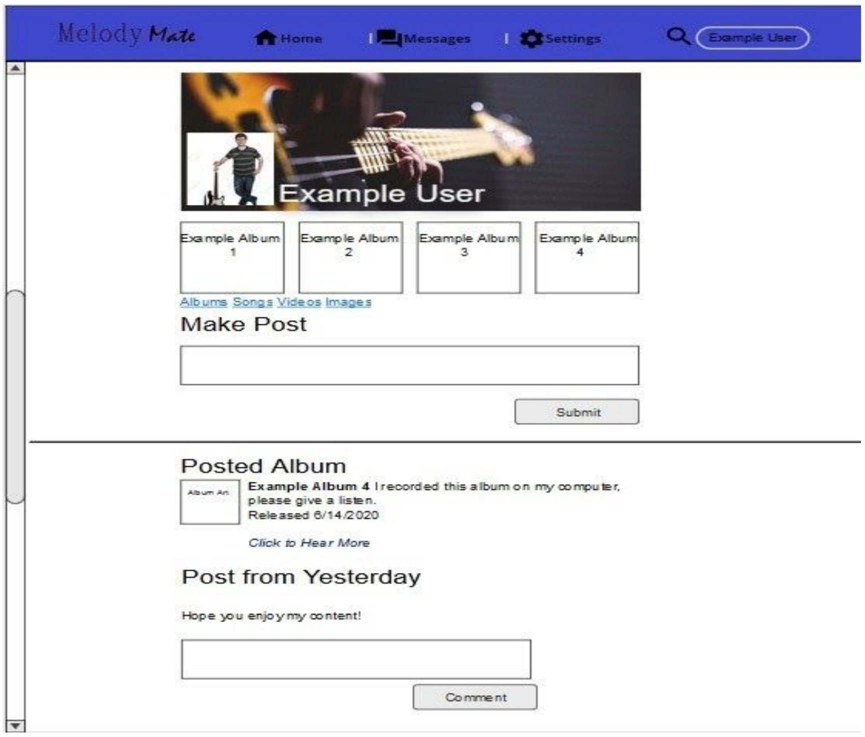
Example Message....

Example Message....

Settings Page:



User Page:



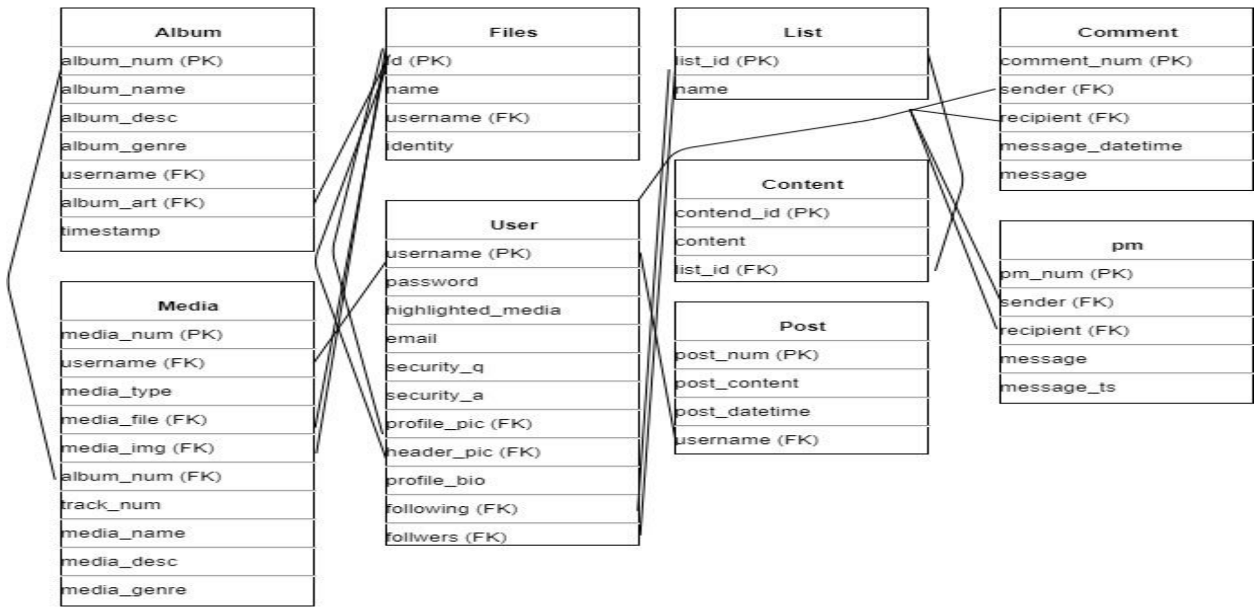
Example Album 4 I recorded this album on my computer, please give a listen.
Released 6/14/2020

Hope you enjoy my content!

The web application Moqups was used in browsers to create this design.

Prior to implementation, but after initial design, I created an improved database schema to represent my website.

Database Schema



Implementation

The appearance and functionality of the site changed drastically with the implementation of it. This is mainly due to ideas adapting over time, as well as the use of bootstrap and JQuery libraries. As this site is large scale for a school project, it contains many queries and algorithms. The core ones include: The news feed and profile feed queries, the private messaging query, media uploading queries, and media display / audio playing queries / algorithms.

- **News Feed / Profile Feed** - The first step in running this news feed query / algorithm, was to query a list of users that you follow, first by getting the list number of the users following list by querying the list table. Then querying and pushing to a list every value in the content table that has that list number. That is simple enough a query, But, after this a query to export the posts of those that the user follows as well as their own had to be made. The following is that:

```
<?php while ($znum != $numfollowing){
                $paramstring = $paramstring."username =  '".$following[$znum]."' OR ";
                $znum = $znum+1;
            }
            $paramstring =  $paramstring."username = '".$username."'";
            $followingpostfinder = "SELECT username, post_content, post_datetime
FROM post WHERE ".$paramstring." ORDER BY post_datetime desc;";
            $stmt = $connection->prepare($followingpostfinder);
            $stmt->execute();
            $result = $stmt->get_result(); // get the mysqli result
            while ($row= $result->fetch_assoc()) {
                array_push($following_posts,
$row['username']."=>".$row['post_content']."=>".$row['post_datetime']);
                $num = $num+1;
```

```
        }
```

A while loop is used to push every username to a parameter string which is called in the SQL string. This parameter is then queried and used to pull all this information to an array. In PHP, to format an array with 3 values, I had to separate each value for posts with "=>", later separating them when they are echoed to the user. Here is the code I used to export this information:

```
        while ($num != $numposts){
                $usernamea = substr($following_posts[$num], 0,
    strpos($following_posts[$num], '=>', 0));
                $stringx = str_replace($usernamea."=>",
"",$following_posts[$num]);
                $thepost = substr($stringx, 0, strpos($stringx, '=>', 0));
                $timestamp = str_replace($thepost."=>", "",$stringx);
                $sql = "SELECT name FROM files WHERE username = ? AND
identity = ?";
                $stmt = $connection->prepare($sql);
                $stmt->bind_param("ss", $usernamea, $identity);
                $stmt->execute();
                $result = $stmt->get_result(); // get the mysqli result
                while ($row= $result->fetch_assoc()) {
                        $profilepica = $row['name'];
                }
```

This code splits the values from the array to respective variables and uses this to get profile information.

```
                $comments = [];
                $postnosql = "SELECT post_num FROM post WHERE username =
? AND post_datetime = ?";
                $stmt = $connection->prepare($postnosql);
                $stmt->bind_param("ss", $usernamea, $timestamp);
                $stmt->execute();
                $result = $stmt->get_result(); // get the mysqli result
                while ($row= $result->fetch_assoc()) {
                        $postnumber  = $row['post_num'];
                }
                $sql = "SELECT message, sender, message_datetime FROM
comment WHERE recipient = ? AND post_num = ?";
                $stmt = $connection->prepare($sql);
                $stmt->bind_param("ss", $usernamea, $postnumber);
                $stmt->execute();
                $result = $stmt->get_result(); // get the mysqli result
                while ($row= $result->fetch_assoc()) {
                        $msgdatetime = $row['message_datetime'];
                        $msg = $row['message'];
                        $sndr = $row['sender'];
                        //Get profilepic
```

```
                                    $picfindingmodule = "SELECT name FROM files WHERE
username = ? AND identity = ?";
                                    $stmt9000 = $connection->prepare($picfindingmodule);
                                    $stmt9000->bind_param("ss", $sndr, $identity);
                                    $stmt9000->execute();
                                    $result3 = $stmt9000->get_result(); // get the mysqli result
                                    while ($row3= $result3->fetch_assoc()) {
                                           $profilepicb = $row3['name'];
                                    }
                                    $string = "
                                    <div class='media mt-3'>
                                      <div class='mr-3'>
                                      <img class='rounded-circle ml-2'
src='/melodymate/profilepics/".$sndr."/".$profilepicb."' alt='Profilepic' style='width:50px;
height:50px;'>
                                      </div>
                                      <div class='media-body mr-5'>
                                           <h5 class='mt-0'>".$sndr."</h5>

                                           <p>".$msg."</p>
                                        <small>".$msgdatetime."</small>
                                    </div>
                                    </div>
                                    ";
                                    array_push($comments, $string);
                             }
                             echo "
```

You can see here, the code queries comments related to the post and pushes them to a comments array.

```
                               <img class='rounded-circle'
src='/melodymate/profilepics/".$usernamea."/".$profilepica."' alt='Profilepic'
style='width:60px; height:60px;'>
                                    <div class=''><h4 class='mt-2'>".$usernamea."</h4>
                                    <br><p>".$thepost."</p>
                                    <small>".$timestamp."</small><br><br>
                                    <button type='button' class='btn btn-primary'
data-toggle='collapse' data-html='true' data-target='#replies".$num."'>View
Replies</button>
                                    ";
                                  if($usernamea == $username){
                                           echo "<br><br><a href='#'><button type='button'
class='btn btn-danger'><small>X</small></button></a>";
                                    }
```

The post is displayed here using html through PHP, and then uses a while loop to export all the comments to the post.

```
					echo "<div id='replies".$num."' class='collapse'>";
					if (isset($comments)){
						$numofcomments = count($comments);
						$commentnum = 0;
						while ($commentnum != $numofcomments){
							echo $comments[$commentnum];
							$commentnum = $commentnum+1;
						}
					}?>
```

- **Private Messaging** - The private messaging algorithm / queries runs a query function repeatedly over and over again using a meta tag to refresh in a code set in an iframe. This is a barebones chat set up, but in my time frame and with my limited resources, it was what was possible. The following is a snippet of this code:

```
   <?php	$messagessql = "SELECT message, message_ts, sender, recipient FROM
pm WHERE (sender = ? AND recipient = ?) OR (sender = ? AND recipient = ?) ORDER
BY message_ts DESC";
	$stmt = $connection->prepare($messagessql);
	$stmt->bind_param("ssss", $username, $chatrecipient, $chatrecipient, $username);
	$stmt->execute();
	$result = $stmt->get_result(); // get the mysqli result
	while ($row= $result->fetch_assoc()) {
		$messages[$num] = $row['message'];
		$message_ts[$num] = $row['message_ts'];
		$sender[$num] = $row['sender'];
		$recipient[$num] = $row['recipient'];
		$num = $num +1;
	}
	$nummessages = $num;
	$numberofpages = ceil($nummessages / 5);
	if ($numberofpages > 5){
		$numberofpages =5;
	}
	$num = 0;
	$identity = 'profilepic';
	$profilepicfinder = 'SELECT name FROM files WHERE username = ? AND
identity = ?;';
	if (!empty($messages)){
		if ($numberofpages == 1){
			$num =0;
			$endnum = $nummessages;
		}
		if ($numberofpages ==2 AND $page==1){
			$num =0;
			$endnum = 5;
```

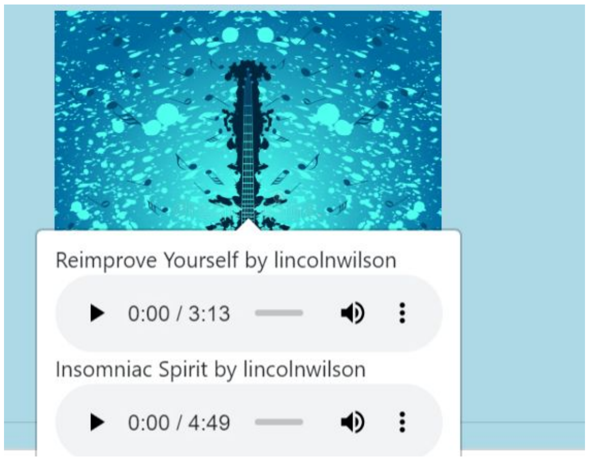
```
            }?>
```

These if statements go up to $numberofpages == 5 and $page == 5 respectively. This allows up to five pages of recent messages. To get the number of pages, this algorithm takes the number of messages, divides them by five and rounds them up. For example, if there were 6 messages, that would be divided by five and be bigger than 1, so two pages would be used. The number of pages would be limited to five with an if statement, preventing too much from being displayed. All the messages related to their respective number of pages and page numbers are then echoed. This echo, from messaging.php appears in the iframe in chat.php.

- **Media Display** - Queries are used to export to a list all the albums of the current user. All this is used to set album and track info for each album, and sent to audio players with this code:

```
 <?php while ($wnum != $numberoftracks) {
                        //Get Track File
                        $trackfilesql = "SELECT name FROM files WHERE username = ?
AND id = ?;";
                        $stmt = $connection->prepare($trackfilesql);
                        $file = $trackfiles[$wnum];
                        $stmt->bind_param("si", $username, $file);
                        $stmt->execute();
                        $result = $stmt->get_result(); // get the mysqli result
                        while ($row= $result->fetch_assoc()) {
                            $trackfile = $row['name'];
                        }
                        //Get Track Name
                        $namefinder = "SELECT media_name FROM media WHERE
username = ? AND album_num = ? AND track_num = ?;";
                        $stmt = $connection->prepare($namefinder);
                        $b = $wnum+1;
                        $stmt->bind_param("sii", $username, $albumnums[$num], $b);
                        $stmt->execute();
                        $result = $stmt->get_result(); // get the mysqli result
                        while ($row= $result->fetch_assoc()) {
                            $trackname = $row['media_name'];
                        }
                        //Append String
                        $trackstring = $trackstring.$trackname.' by '.$username.'
                        <br><audio controls style="width:250px; height:50px;">
                          <source src="'.$albumlocation.$trackfile.'" type="audio/mpeg">
                        Your browser does not support the audio element.
                        </audio><br>';
                        $wnum = $wnum+1;
                    }
                    $audioPlayer[$num] =
                '
                        <div class="text-left" style="width:700px;">
                                '.$trackstring.'
                        </div>
                ';
                $num = $num+1;?>
```

This makes the audio appear in a player that appears like this on the album once it is clicked:



Reimprove Yourself by lincolnwilson

0:00 / 3:13

Insomniac Spirit by lincolnwilson

0:00 / 4:49

The album art and audio popups are shown with the following code, utilizing bootstrap carousel and popovers.

```
<?php if(isset($albumartfilelist[0])){
        echo"<div class='mx-auto text-center'><div id='firstrowcarousel' class='carousel
slide' data-ride='carousel'>
                <div class='carousel-inner'>
                  <div class='carousel-item active'>
                      <a  data-toggle='popover' role='button'  data-placement='bottom'
data-content='".$audioPlayer[0]."'>
                      <img src='".$albumlocationslist[0].$albumartfilelist[0]."'
style='width:250px;height:250px;'>
                        </a>
                   </div>
                    ";
}
if(isset($albumartfilelist[1])){
        echo "
                                <div class='carousel-item'>
                                            <a  data-toggle='popover' role='button'
data-placement='bottom' data-content='".$audioPlayer[1]."'>
                        <img src='".$albumlocationslist[1].$albumartfilelist[1]."'
style='width:250px;height:250px;'>
                        </a>
                  </div>
                    ";}?>
```

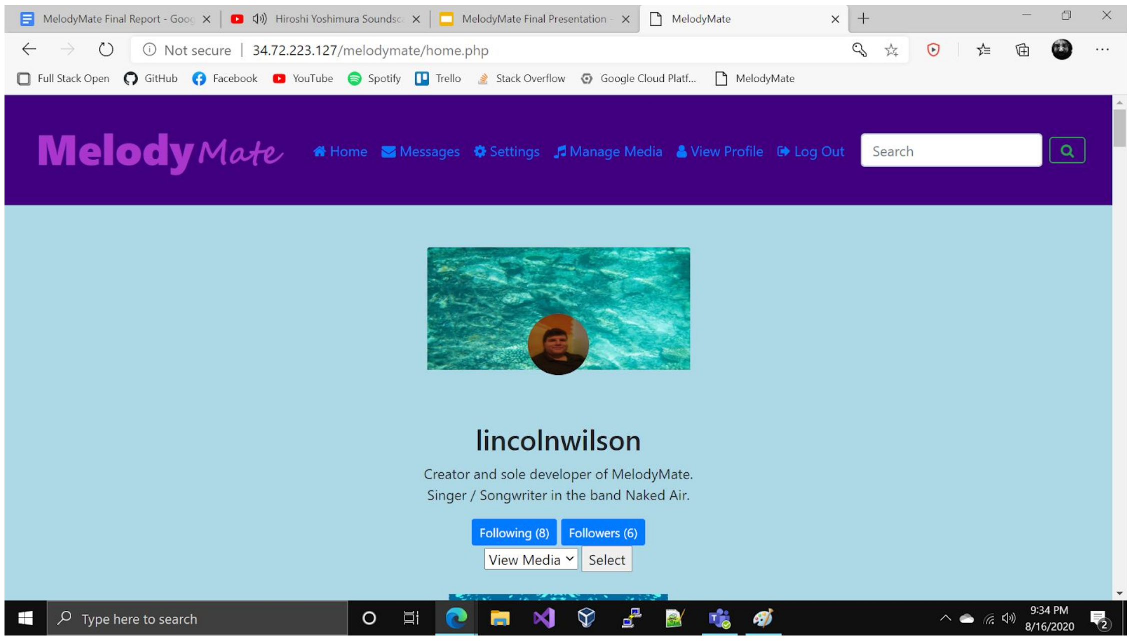
This code is repeated for the rest of the values in the $albumartfilelist array. As you can see, the popover is a part of the Jquery library, using javascript to refer to its settings.

There are many more algorithms and queries that could be gone over, but these are the essentials. The appearance of the site relied heavily on Bootstrap and Jquery, making its appearance much more professional, and as you can see had a big part in displaying content from algorithms.

Screenshots

This will include screenshots of the site in both desktop and mobile format:

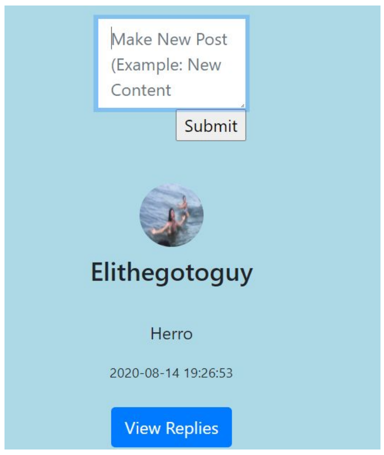
- **Navbar and Homescreen**



lincolnwilson

Creator and sole developer of MelodyMate.
Singer / Songwriter in the band Naked Air.

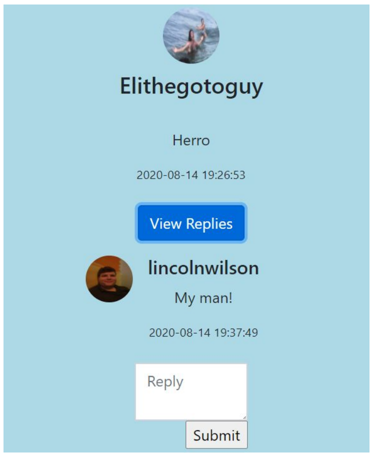
- **News Feed and Replies**



Elithegotoguy

Herro

2020-08-14 19:26:53



Elithegotoguy

Herro

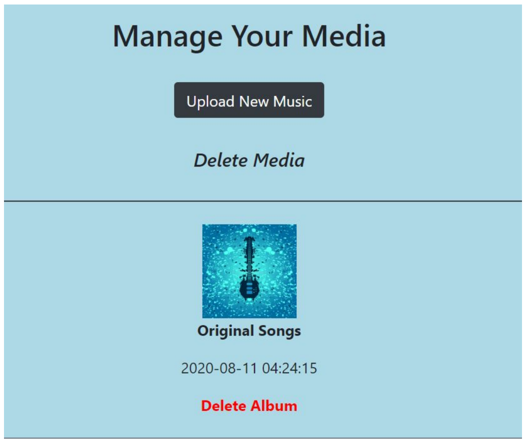
2020-08-14 19:26:53

lincolnwilson

My man!

2020-08-14 19:37:49

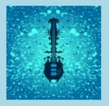
- **Manage and Upload Media**



Manage Your Media

Upload New Music

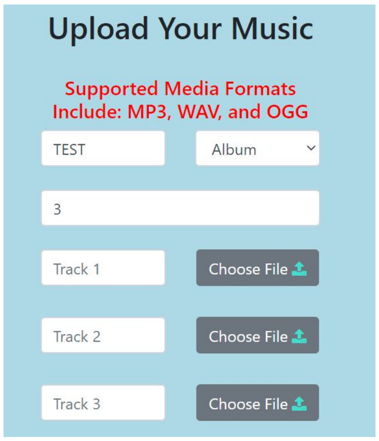
Delete Media



Original Songs

2020-08-11 04:24:15

Delete Album



Upload Your Music

Supported Media Formats Include: MP3, WAV, and OGG

TEST | Album

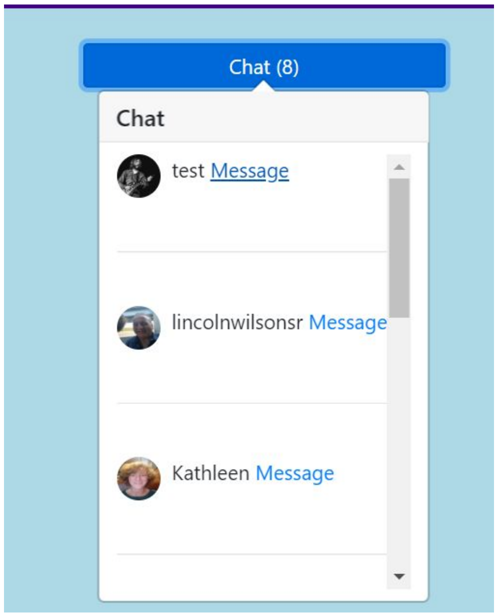
3

Track 1 | Choose File

Track 2 | Choose File

Track 3 | Choose File

- **Messaging**



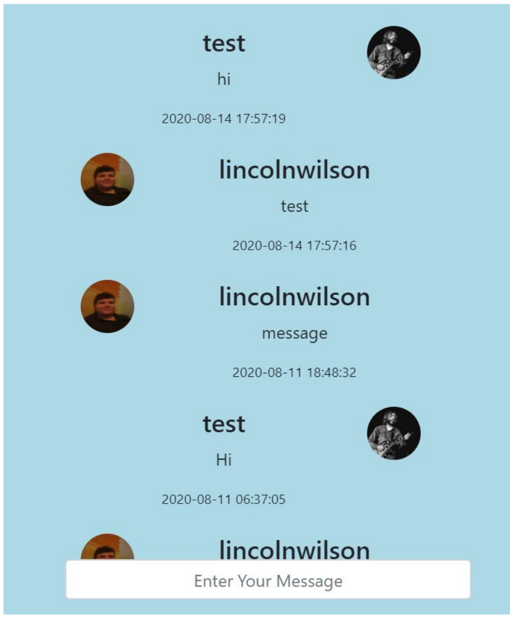
Chat (8)

Chat

test Message

lincolnwilsonsr Message

Kathleen Message



test
hi
2020-08-14 17:57:19

lincolnwilson
test
2020-08-14 17:57:16

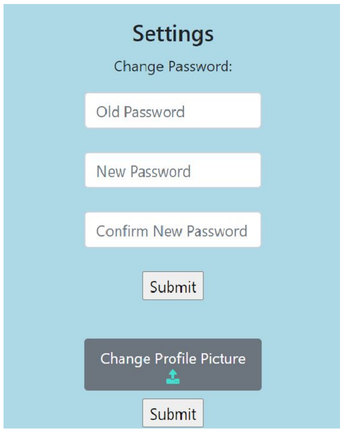
lincolnwilson
message
2020-08-11 18:48:32

test
Hi
2020-08-11 06:37:05

lincolnwilson

Enter Your Message

- **Settings**



Settings

Change Password:

Old Password

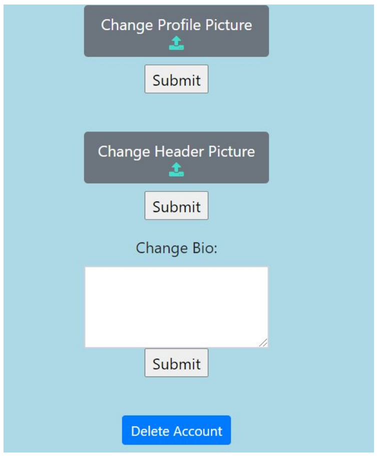
New Password

Confirm New Password

Submit

Change Profile Picture

Submit



Change Profile Picture

Submit

Change Header Picture

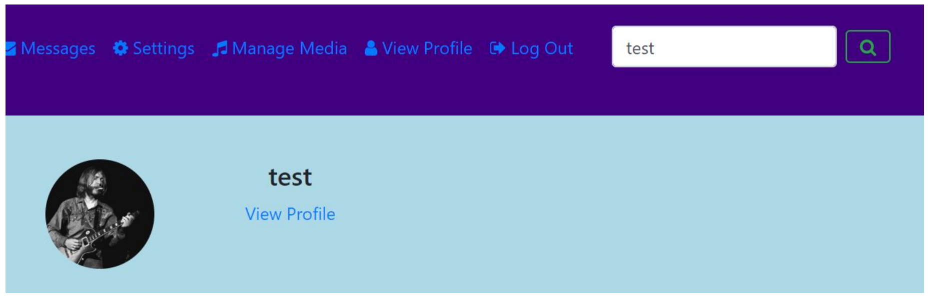
Submit

Change Bio:

Submit

Delete Account

- **Functioning User Search**



Currently, this website is being hosted using a google web server so if one wanted to access this site, they would do so by visiting the following link: MelodyMate.

However, if one wanted to go about installing this server themselves on their own personal server, they would go about doing that by copying the source code into the public-html folder in a lamp stack or whatever apache/php server they use and import the database .sql file through PHP My Admin.

State of Implementation

Currently, I have reached most of my goals. The only major feature I didn't implement to save time was a non-user group page prospect. The things that a user can do include:

- Create an account, use a security question and answer to recover their password, and login to their own personal profile.
- Create and view theirs and others posts on their own profile feed and respond to other users
- Manage their media, given the ability to upload new media and delete old ones.
- View that media on their own profile and view other's media.
- Search for other users using the search bar and being able to follow them, adding them to their following list.
- Messaging other users through a private messaging service.
- Adjust their settings, able to change password, bio, profile and header picture, as well as delete their account.
- View and scroll through their following and followers list.

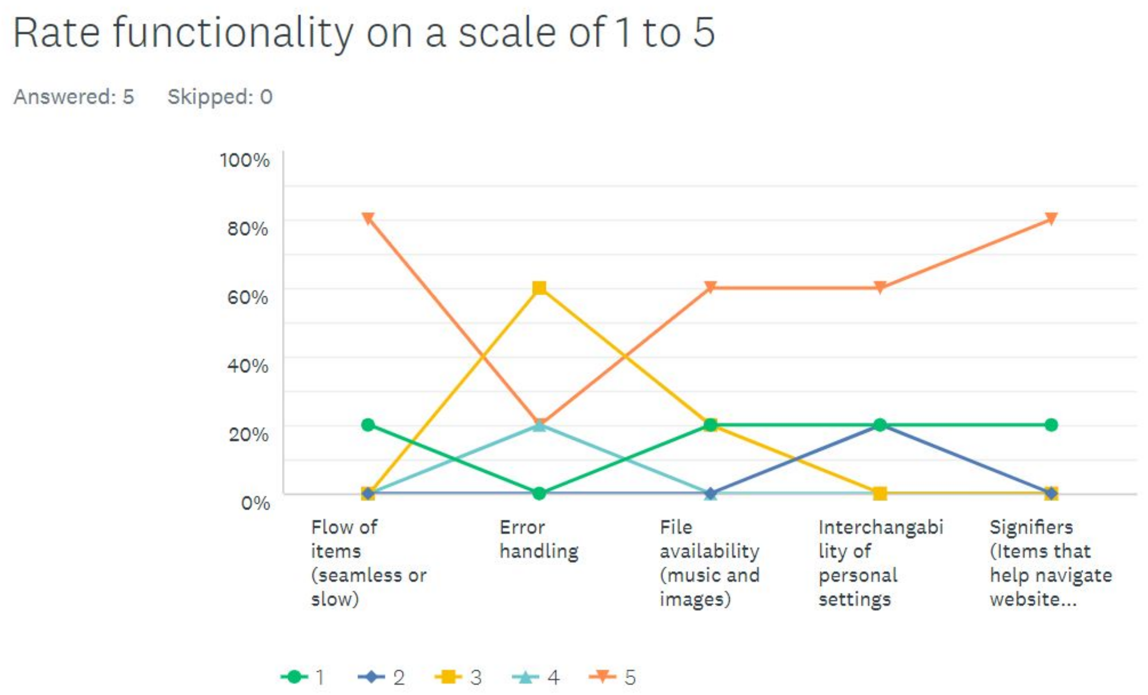
The only features I would add with more time for this first version, would be, like I mentioned before, group pages, the ability to view single track media, and the ability to delete personal posts. These will all most likely be implemented after I finish this course as I plan to continue working on this site.

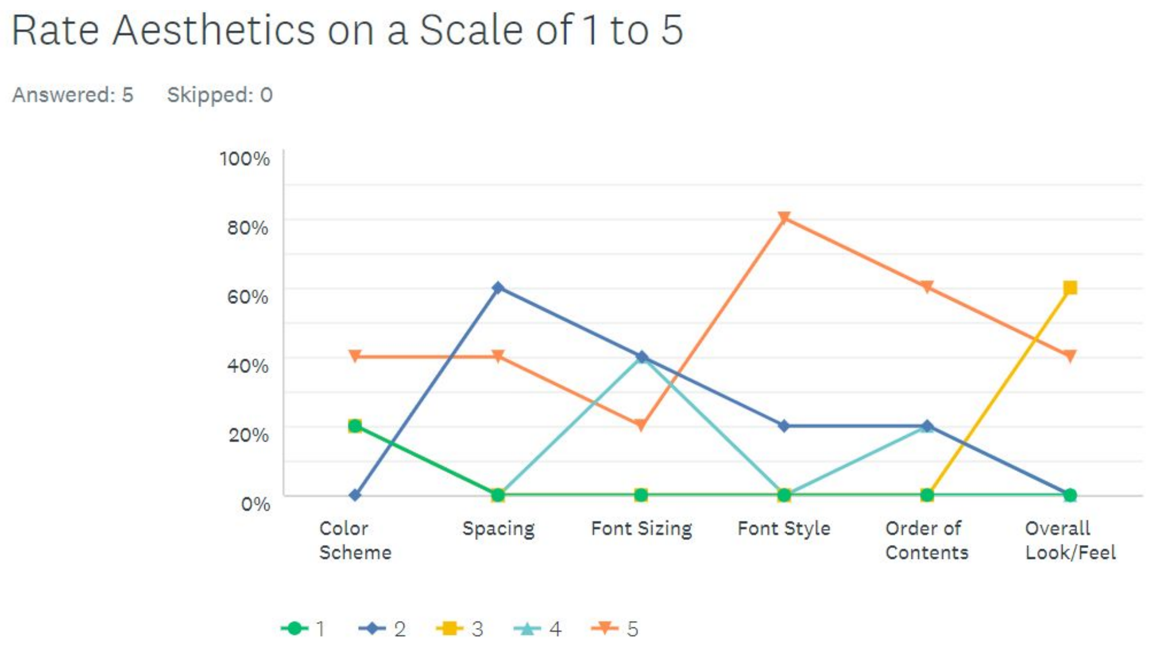
Testing and Evaluation

For testing, due to pandemic restraints, I was limited to about 5 to 7 testees who were all friends and family. I still had a fruitful testing experience. I delivered my website to my testees through the google server ip link. I used a survey monkey to create and distribute a survey.

The survey had 9 questions, 5 open word questions, 2 multiple choice, and two grid. These results were overall positive, as most problems were due to bugs I later fixed. This testing

helped me find a lot of bugs, for example, one that caused incompatibility with the iPhone Safari browser, and another causing bios to not contain line breaks.
Here are the results of this survey:
https://www.surveymonkey.com/results/SM-CGXLHZWB7/





These results would be more accurate with more testers, especially with a couple that were more tech savvy, as most of my testers were not, still it helped yield great results.

Lessons Learned and Reflections

There were many changes to this site over time, and it looks nothing like it did initially. I changed courses many times, but one thing that stood true most of the time was my database schema. After creating my schema once finishing my initial design, I only added one table and very little columns. I had just what I needed to store my data. Initially, I intended to include a feature that allowed users to great groups/band pages for their music. I decided to put this on hold to focus on the rest of implementation and never got to it. Instead, I expanded on the profile feed idea, adding the idea to have dropdown replies and other features. One major conflict I had was making my website responsive on a mobile platform. As a social media site, mobile view is important to consider. I was at one point frustrated at how bootstrap items such as grids and media objects would be a pain to adapt to mobile, that I considered switching to an adaptive design, pushing it off till later. Eventually, however, I returned to a responsive design and was able to implement it successfully. If I was to do this again, I would make my code more organized. I would separate my queries from my PHP and have them prepared before implementation. This taught me on the limits of full stack development. If I focused on either one single thing, backend or frontend of the code, with another person doing the other, a lot more could've been done. Still, I do not regret this and rather enjoyed the process, and may consider full stack development if offered.

Further Work

I plan on continuing the development of this site in the future for my portfolio. I will implement the features I wanted to get around to but never did, including group pages, viewing single tracks, and deleting posts. I would also like to implement a better chat option, as the current one flashes with the refreshing of the query. I also would like to make suggestions possible, where users could be given a list of users in their area or those that like or play similar things. I would also like to add the ability to upload images and video, as well as the ability to share images and video and albums and tracks. One major thing I was not able to implement in time was notifications, I hope to implement that soon as well.
Here is a link to my trello board which I keep updated:
https://trello.com/b/qk26301n/melodymate

Professional Development and Lifelong Learning

For this project, I used the Bootstrap and JQuery javascript libraries. I had little experience with Javascript and had never set up a LAMP Stack server before. Although I had plenty of knowledge about PHP, I had to learn a lot on my own. I feel I have greatly improved my PHP skills as well as Javascript. To learn about these things I relied heavily on online research. I used the W3schools website, Bootstraps website, and Stackoverflow the most. As I knew next to nothing about bootstrap starting off, I decided the best way to go about using it was to read every page of its W3schools and Bootstrap articles. I didn't understand everything, but it gave me a general idea on how to utilize bootstraps classes along with JQuery. Stackoverflow was particularly useful as it provided answers to specific questions and often counted for multiple different languages and libraries. An example of a situation where I had to utilize Stackoverflow was when I had a bug causing popovers to not display on iPhone Safari browsers. Having looked at the W3schools and Bootstrap page on popovers many times, I knew the solution would not be there. I looked it up on Stackoverflow and the solution was one of the first results, relating to the data-trigger='focus' setting in the javascript/html. There were a couple times where I was unable to use Stackoverflow to find the solution to an issue. I had to review the entirety of my code to find discrepancies. Having done this several times, I know if I coded a site with the same framework again, I would do it in such a way that I would not have to worry about the same bugs.

A great challenge I faced later on in the development of my site was transferring my files from a local XAMPP server to a google web services LAMP stack. I expected a quick transfer of files would be easy and I would face little issue. However, I faced bug after bug attempting to get the site fully functional. It took about 2-3 days to remove the bugs created by the transfer. In the process, I learned a lot about PHP and Apache configuration, as well as a MySQL. I have used google web services for a server before, but never a LAMP Stack, so I learned even more about the possibilities with google. I feel like I could apply this knowledge to other types of servers/stacks. Now that I know more about this along with PHP, SQL, Bootstrap, Javascript, and JQuery, I feel I could much easilier create a web server if needed, and I could apply this in the workplace.

Resources Used

- *Libraries*
  - Bootstrap
    https://getbootstrap.com/
  - JQuery
    https://jquery.com/
  - Popper.js
    https://popper.js.org/
- *Languages*
  - PHP
  - Bash
  - Javascript
  - SQL
  - CSS
- *Services*
  - Google Cloud Platform
  - LAMP Stack w/ Apache
- *Database Services*
  - MySQL
  - PHPMyAdmin