

Project Overview

- Repo ownership of project on github (github.com)
 - User: TwinsMusic
 - Password: 8h3TpGq9@EA^HSj
- Email credentials (login.yahoo.com)
 - User: musme_test@yahoo.com
 - Password: fY8urgF4vCWASy7
- Github is registered with above email address

Frontend Technologies

- [ReactJS framework](#)
- JavaScript
- [Bulma CSS framework](#) + Sass
- [React Bootstrap](#)
- [React Routers](#)
- Node/NPM

Backend Technologies

- [PostgreSQL setup](#)
- [Java 11 with Spring Framework](#)
- Maven
- [Node](#)/NPM

Frontend - Major Notes

- Playlist
 - Each mood needs matching tracks uploaded – can either do this manually or link to S3 files in database by querying by tag/mood
- Special files
 - `index.html` and `index.js` load on every web page. `Index.js` is what should usually be directly edited if you want to edit something that appears on all pages
 - `mystyles.css` should not be edited directly, it is where the Bulma document `mystyles.scss` gets compiled
 - To edit Bulma variables across the entire website, edit the `mystyles.scss` file according to the Bulma documentation, and then compile it by running the following command while in the `src` folder:

```
sass sass/mystyles.scss:public/mystyles.css
```
 - To edit a webpage, go to the corresponding `.js` file
- Sass
 - In order to use Sass, you will need to **install [Ruby](#) and install [Sass](#)**.
 - You will have to do this if you want to edit any of the variables in `mystyles.scss` to change the appearance of the site
- Moods
 - How to add and generate tags
 - Link to S3 bucket tracks

- Admin
 - Tracks are stored in database
 - Tracks eventually need to link in with home page tracks
 - URL field should be the link to the file in the S3 bucket
 - The home page should then pull the tracks from the database, fetch from the URL to the S3 bucket, load the music files, and display the waveform with the other details of the song from the database
- Login/Signup
 - Passwords are not hashed yet
 - All user information is stored in the database
 - Forget password uses an email account and auto-generated tokens to allow users to reset passwords
 - Sign Up is located behind the login screen so that access is restricted and only authorized users can add new accounts
 - Make sure that when database is initialized there is a master login for the admin backend, and tell users to make their own logins (similar to root/user in Linux)
- Request Music
 - Sends all information from the form to an email address
 - Email address to send to needs to be changed in the backend

Backend - Major Notes

- Database
 - Structure of database
 - User.java controls backend user creation
 - Track.java controls track metadata values
 - PostgreSQL should auto generate the fields in the database when run
 - AWS
 - Link S3 bucket on the back end, specifically in the maven files
 - Double check the localhost port numbers are the same as the ones in the frontend
 - Should be able to drag and drop files onto S3 bucket on AWS dashboard and call any file you need on the front end
- SQL queries to make moods work (if queries)
 - See above
- Password Reset info
 - Passwords are not hashed, simple hashing will do fine
 - Make sure the reset function hashes them when uploading to the database and when checking current password
 - Password reset email setup
 - Password reset emails are currently sending from musme_test@yahoo.com
 - If this needs to be changed, make sure SMTP server link matches email service

- For email, generate a secure app password to use in credentials if changed
- Once URL is changed, make sure link in password reset email template matches URL for musme website

Tasks Still to Do:

- Deploy to AWS
- Link home page tracks to tracks in AWS S3 bucket
- Link Browse page (Moods, Ringtones, etc.) tracks to tracks in AWS S3 bucket
- Link admin page to home page and allow for track manipulation
- Make passwords hashed for security
- Email address to send to for request music needs to be changed in the backend
- After deploying remove github users from this project

AWS Account:

- Create a new AWS Root Account with MusMe information - Choose Personal and Free-Tier
- Create an S3 Bucket - Instructions in resources and on AWS site
- Backend should be setup already for S3 bucket, just change the url and authentication for new MusMe bucket

Resources:

- <https://mmafrar.medium.com/integrating-your-spring-boot-project-with-amazon-s3-ea3453af7fa0>
- <https://aws.amazon.com/getting-started/hands-on/build-react-app-amplify-graphql/>
- <https://blog.cloudthat.com/step-by-step-guide-to-deploy-reactjs-app-on-aws-s3/>

If any questions or code are not covered here, contact us:

Contact Emails:

Jimmy Rosser - jimmyrosser2000@gmail.com

Gracen Hoyle (contact for email issues) - gracen.hoyle@pm.me