

Anh Dang, Lam Dang, Vuong Khuat, Aleks Sergeev

CS 345 Software Engineering

Music Recommender Webpage

I. Project Description

Our project is a web application with both frontend and backend components written fully and exclusively by members of the group. We wrote all of the code in Javascript, with the backend utilizing the React library to pass all HTTP requests to the public Spotify API.

Our web application was designed with the idea that a user may open up the page, and without signing in provide a selection of artists, songs, and genres to generate recommendations. It is designed as quick-use application, and therefore does not have the intention of storing data about the user. Specifically referring to our proposal, we have implemented the following features that meet the main goal of our project:

- Searching for song/artist/genre to use for an input,
- Keeping track of inputs and displaying them to the user,
- Removal of inputs to allow multiple different searches,
- Returning artist and song recommendations and previewing them directly from the app.

We also were able to implement the extra implementation of recommending playlists for users, which comes directly from a genre input. All the recommendations come named and with images, the same for searches, which makes it clear and friendly for the user to search/look through recommendations.

II. Install Instructions

Your computer must have node installed. More information here: <https://nodejs.org/en/>

- In the folder `./backend`:

1. Install the necessary packages:

npm install

2. Run the file `App.js`, which starts the backend server (note that this must not be interrupted during use of the app):

node App.js

- In another terminal, in the folder `./frontend`:

1. Install the necessary packages:

npm install

2. Run the application

npm start

III. Unit testing

For unit testing, since our backend calls use a recognized library for HTTP requests and pass the calls directly to Spotify's stable public API, we believe the backend does not require extensive unit testing. We wrote unit tests for functions which parses the data from the API responses to formats more appropriate in our app. To run the test, in the folder `./frontend`, run the command

npm run test

The unit tests are written using the MochaJS library.