Music Library React (Frontend) Django

Introduction

The primary focus of the Music Library React (Frontend) project is to further familiarize yourself with React.js and grow in confidence regarding the understanding of making API calls with Postman as well as in our React applications utilizing the Axios HTTP library.

Technologies

React.js, React Hooks, HTML/CSS, Axios, Postman, Adobe XD (or wireframing software of choice)

Learning Objective

The learning objective of this project is to learn how to interact with a third-party API and consume its response data. First, you will learn and practice testing API calls in Postman. By doing this you better prepare yourself of making the actual request in a code implementation. After it has been successfully tested you can implement the call using the Axios HTTP library and consume the response data within the React application. This is a great initial learning experience on how most applications in the modern age work and interact with third party resources to receive and consume data.

The second large part of the Music Library React (Frontend) project is further familiarization with React syntax as well as how to consume and display JSON data within the application. Specifically in this case the need to use the “map” higher order array method to generate table rows for the music table.

Lastly, implementing a form component with the ability to filter and reduce the data based off the user search will provide good exercise in forms and passing data between component using function references and props.

*💡 Make sure to be debugging and console logging API request methods and responses!*

Resources

**PowerPoints**

* Web Services
* Request & Axios
* Promises
* Postman
* Intro To React

**Documents**

* User stories
* Steps To Create React App (Day 39 Portal)

**Relevant Projects**

Weight Tracker – (Video tutorial series project)

Social Feed

**Other Resources**

Music Library final result demo video

Tasks

* **Review all videos and documentation.** Make sure to have a clear direction regarding the end result of the project as well as all supporting documentation that is available to you.
* **Design the React application using a wireframing/prototyping tool like Adobe XD or Figma.** Remember that this is a single page application.
* **Build and test the React Frontend** using React best practices (components organized into their own folders with external CSS stylesheets, lifting state up as high as possible in component hierarchy, passing data from parents to children as props, etc)

Setup Steps

*For Step 3 from Tasks*

* Start by setting up the project using CREATE REACT APP. Be sure to test running the development server and verifying that your application runs in the browser. You will need these components at a minimum:
* App.jsx
* Index.js
* DisplayMusic.jsx
* SearchBar.jsx
* Install the Axios package to your react project by running the command npm install axios within your React frontend application.
* Then, set up your App component to make axios requests to the Music Library API for all songs. Verify that the data is appearing in your console upon component mounting using a console log. (your API request should be in a useEffect hook function with an empty dependency array).
* Once you have confirmed you are getting the data back you can proceed to pass it down to your “DisplayMusic” child component.
* Within the DisplayMusic component you will create a HTML table and map over the songs passed down through props to generate the table data rows.
* In a separate “SearchBar” component create a form with one input. This is where the user will type term, they would like to the filter the table by. Once the form is submitted the value the user entered should be passed to a function on the App component that then filters the songs by the term depending on if that term matches any of the song’s properties.

End Result

The final result of the project will be a styled table consisting of all songs retrieved from the Django REST API. The application will also provide the functionality to filter the table based on a string match to any of the song’s properties.