

USER SETUP/ SET UP STEPS	Notes	completion
Setup Steps		
1. Start by setting up the project using CREATE REACT APP. Be sure to test-run the development server and verify that your application runs in the browser. You will need these components at a minimum:	Created react app using tutorial	Done
App.js		
MusicTable.jsx		
SearchBar.jsx		Done
2. Install the Axios package to your react project by running the command npm install axios within your React frontend application.	React front end has been installed	
3. 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.	Went through tutorial on react set up again to get through this section	Done
4. Once you have confirmed you are getting the data back you can proceed to pass it down to your “MusicTable” child component.	Data is being passed down into MusicTable component	Done
5. Within the MusicTable component, you will create an HTML table and map over the songs passed down through props to generate the table data rows.	MusicTable has been set up	Done
In a separate “SearchBar” component create a form with one input. This is where the user will type a search term that they would like to 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	Search Bar has been set up and is functional	Done

TASKS/ USER STORIES	Notes	
Main Stories		
(5 points): As a developer, I want to make good, consistent commits.	Commits have been as sections have been completed as well as certain milestones.	Done
(5 points): As a developer, I want to use the Create-React-App command to create my React project.	Create-React_app has been used to create this project	Done
(10 points) As a developer, I want to use AXIOS to consume the Flask REST API on the frontend.	Axios has been set up to consume the Flask Rest API....	Done
(10 points) As a developer, I want to display the data (song title, album, artist, genre, release date, and running_time) from the API within a table on the frontend.	Data is coming through the as atable	Done
(5 points): As a developer, I want to create a minimum of three React components and use them within my React application. (EX: NavigationBar, SearchBar, MusicTable)	Search Bar, Filter By, Add Song, Music Table	Done
(5 points) As a developer, I want to have an aesthetically pleasing user interface to ensure a great user experience.	Basic Set up to mimic a SQL report with dashboards., wil work on more CSS after bonuses	Done
(10 points) As a music enthusiast, I want to be able to filter the table of music by album, artist, genre, release date, or title.	Filter function is function	Done
(10 points) As a music enthusiast, I want to be able to add a song to my database by using Axios to make a POST request to the Flask REST API	Songs Blue, Yesterday and Sweet Child of Mine were added in using the APP, once after some basic CSS	Done
Bonus		

(10 points) As a music enthusiast, I want to be able to update the song data in my table using a modal popup window that initiates a PUT request to the Flask REST API	thoughts on how to handle this...1. on music table there needs to be a 'on click' function that would display the song information of the selected song. 2. Selected song is what will be passed through, 3. An update function needs to be created to edit the selected song. 4. Upon submission or closing the pop up, the song and the song list is updated after refreshing.....not quite as originally planned, but functional	Done
(5 points) As a music enthusiast, I want to be able to delete a song from my table, initiating a DELETE request to my Flask REST API	Deleted Wonderwall using the App when tested	Done
(5 points) As a developer, I want to display the combined length of the running_time of all songs in the database (Note: This does not need to change depending on which songs meet the current filter)	This has been added. Decided to go with minutes displayed as opposed to seconds just to try	Done
PUNCHLIST ITEMS		
clear filter button	I want to be able to clear the filter after completed instead of having to refresh to bring back to the table	Done
Clear Search or Cancel Search button and also clear add	I want to be able to cancel a search or reset, as well as cancel an add of a song	Done
prompt for deletion of song	I want to confirm that the user wants to delete selected song	Done
prompt for updates to a song	I want to be able to confirm that whatever was changed is being confirmed	Done
Pop up window for edit song	I want the pop up to come above the table instead of below the table when the edit button is hit	Done
Need a cancel button for the edit song window	I want to be able to cancel editing if I choose to not edit any of the songs	Done
END RESULT	Notes	completion