Project documentation

RYTHMIC TUNES:

Your melodic companion

TEAM ID: NM2025TMID40234

PROJECT TITLE: RYTHMIC TUNES: YOUR MELODIC COMPANION.

TEAM ID: NM2025TMID40234

TEAM LEADER: Keerthana S

NM ID: 594359CAF62F242B63B4AC3DD8D2DB05

MAIL ID: rajansoundar364@gmail.com

TEAM MEMBER: Nandhini A

NM ID: 99A3B7865A7005CAE4743934E0E81FAA

TEAM MEMBER:  SHAMINI S

NM ID: 64656147A2D71B08FDB4B09E98754E88

TEAM MEMBER: SAMINATHAN S

NM ID: 46DDF6FB68A0E64A1364959E526702CA

PROJECT OBJECTIVES: RYTHMIC TUNES

The primary goal of Music Streaming is to provide a seamless platform for music enthusiasts, enjoying, and sharing diverse musical experiences. Our objectives include:

**User-Friendly Interface:** Develop an intuitive interface that allows users to effortlessly explore, save, and share their favorite music tracks and playlists.

**Comprehensive Music Streaming:** Provide robust features for organizing and managing music content, including advanced search options for easy discovery.

**Modern Tech Stack:** Harness cutting-edge web development technologies, such as React.js, to ensure an efficient and enjoyable user experience while navigating and interacting with the music streaming application.

Architecture:

* FRONTEND: ReactJS (Web), React Native (Mobile)
* BACKEND: Node.js with RESTful APIs
* DATABASE: Json web server

Setup instructions:

* Prerequisites:

Here are the key prerequisites for developing a frontend application using React.js:

* Node.js and npm: Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.
  + Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.
  + Download: <https://nodejs.org/en/download/>
  + Installation instructions: <https://nodejs.org/en/download/package-manager/>
* React.js: React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

* Create a new React app:

npx create-react-app my-react-app

Replace my-react-app with your preferred project name.

* Navigate to the project directory:

**cd my-react-app**

* Running the React App:

With the React app created, you can now start the development server and see your React application in action.

* Start the development server:

**npm start**

This command launches the development server, and you can access your React app at http://localhost:3000 in your web browser.

* HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.
* Version Control: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.
  + Git: Download and installation instructions can be found at: <https://git-scm.com/downloads>
* Development Environment: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.
  + Visual Studio Code: Download from <https://code.visualstudio.com/download>
  + Sublime Text: Download from <https://www.sublimetext.com/download>
  + WebStorm: Download from [https://www.jetbrains.com/webstorm/download](https://www.sublimetext.com/download)

Follow below steps:

* Install Dependencies:
  + Navigate into the cloned repository directory and install libraries:

**cd tunes**

**npm install**

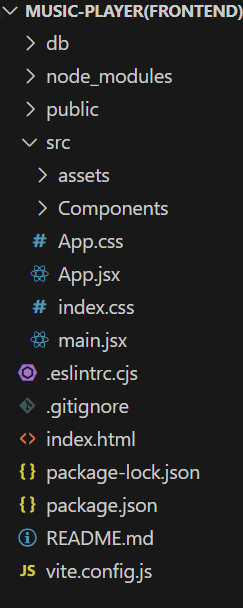
* Start the Development Server:
  + To start the development server, execute the following command:

**npm start**

* Access the App: Open your web browser and navigate to http://localhost:3000.
* You should see the rythmic tunes app's homepage, indicating that the installation and setup were successful.

You have successfully installed and set up the application on your local machine. You can now proceed with further customization, development, and testing as needed.

**Project structure:**

****

The project structure may vary depending on the specific library, framework, programming language, or development approach used. It's essential to organize the files and directories in a logical and consistent manner to improve code maintainability and collaboration among developers.

app/app.component.css, src/app/app.component: These files are part of the main AppComponent, which serves as the root component for the React app. The component handles the overall layout and includes the router outlet for loading different components based on the current route.

Installation Steps:

# Clone the repository git clone

# Install client dependencies cd client npm install

# Install server dependencies cd ../server npm install

Folder Structure

SB-Works/ |-- client/

|\_\_components/

L\_\_ pages/

|\_\_ server/

|\_\_routes/

|\_\_ models/

|\_\_ controllers/

Running the Application

• Frontend: cd client

npm start

• Backend: cd server npm

Start

API Documentation

• User: – /api/user/register

– /api/user/login

• Projects:

– /api/projects/create

– /api/projects/:id • Applications: /api/apply

* Chats:
* – /api/chat/send
* – /api/chat/:userId

Authentication

• JWT-based authentication for secure login

• Middleware protects private routes

Testing

• Manual testing during milestones

• Tools: Postman, Chrome Dev Tools

**Project Implementation & Execution (output):**

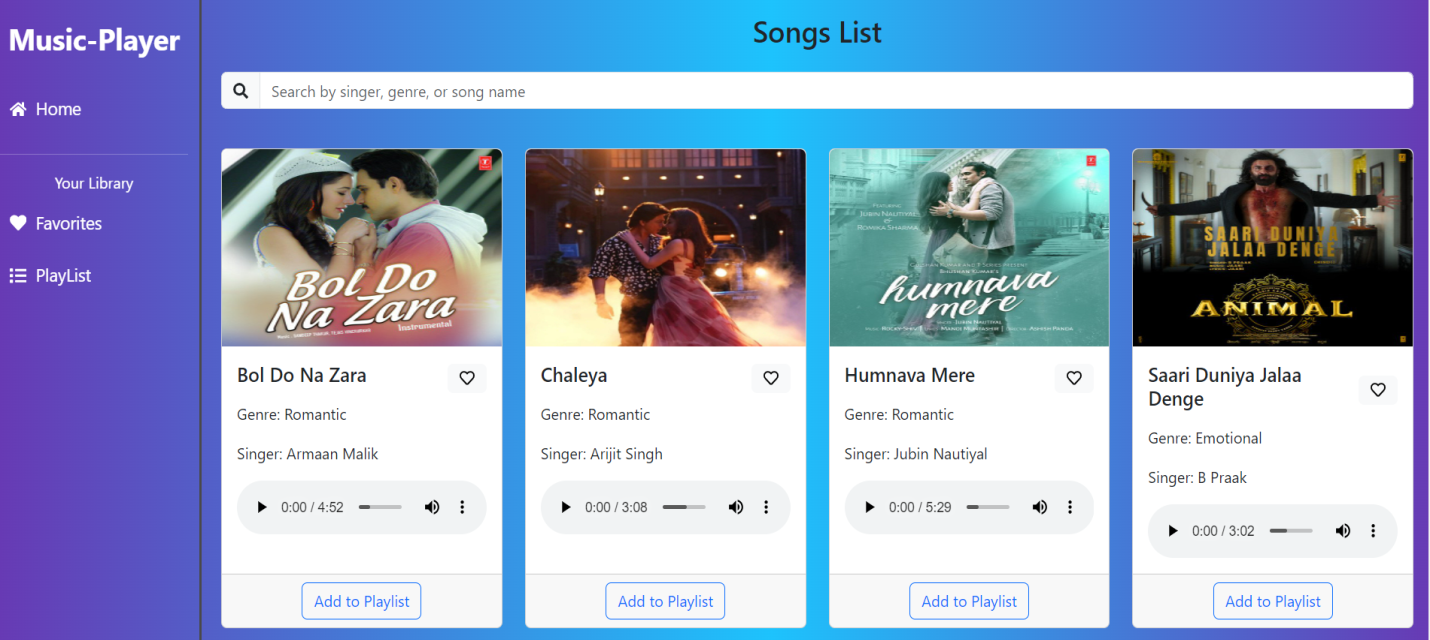
After completing the code, run the react application by using the command “npm start” or “npm run dev” if you are using vite.js

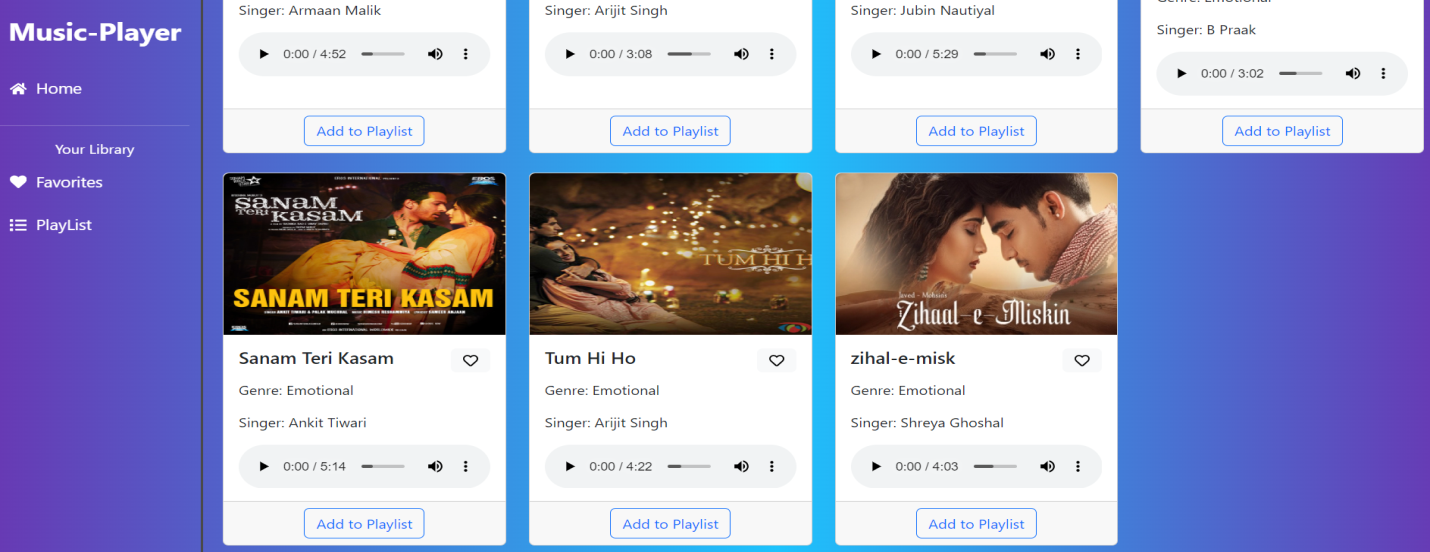
And the Open new Terminal type this command    “json-server --watch  ./db/db.json” to start the json server too.

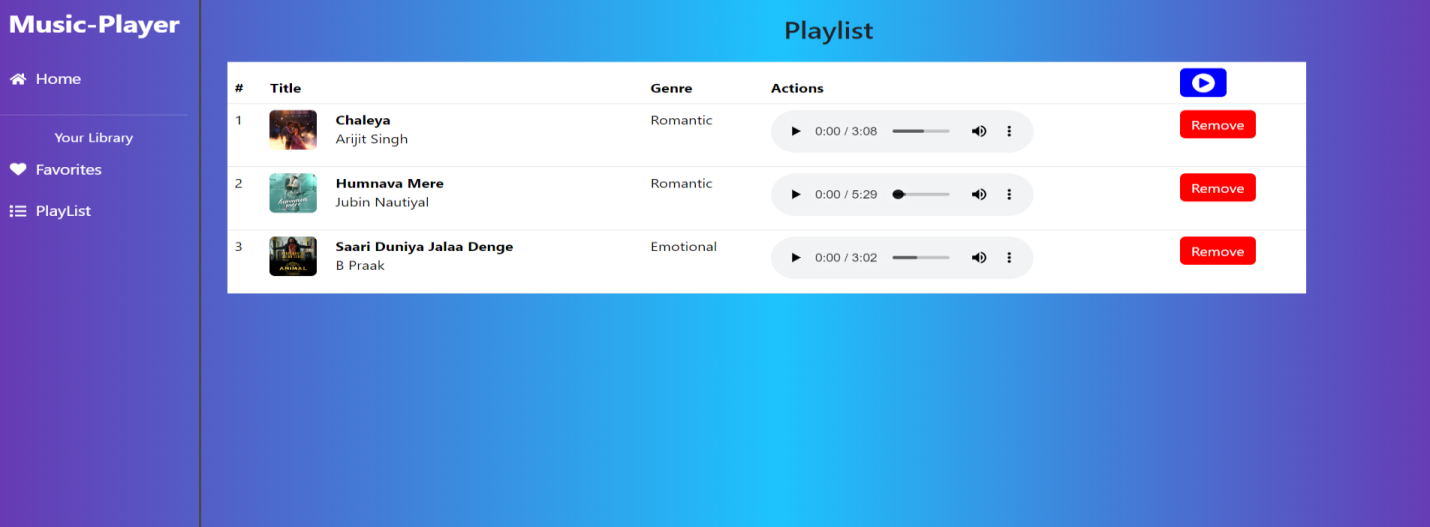
After that launch the Rythmic Tunes.

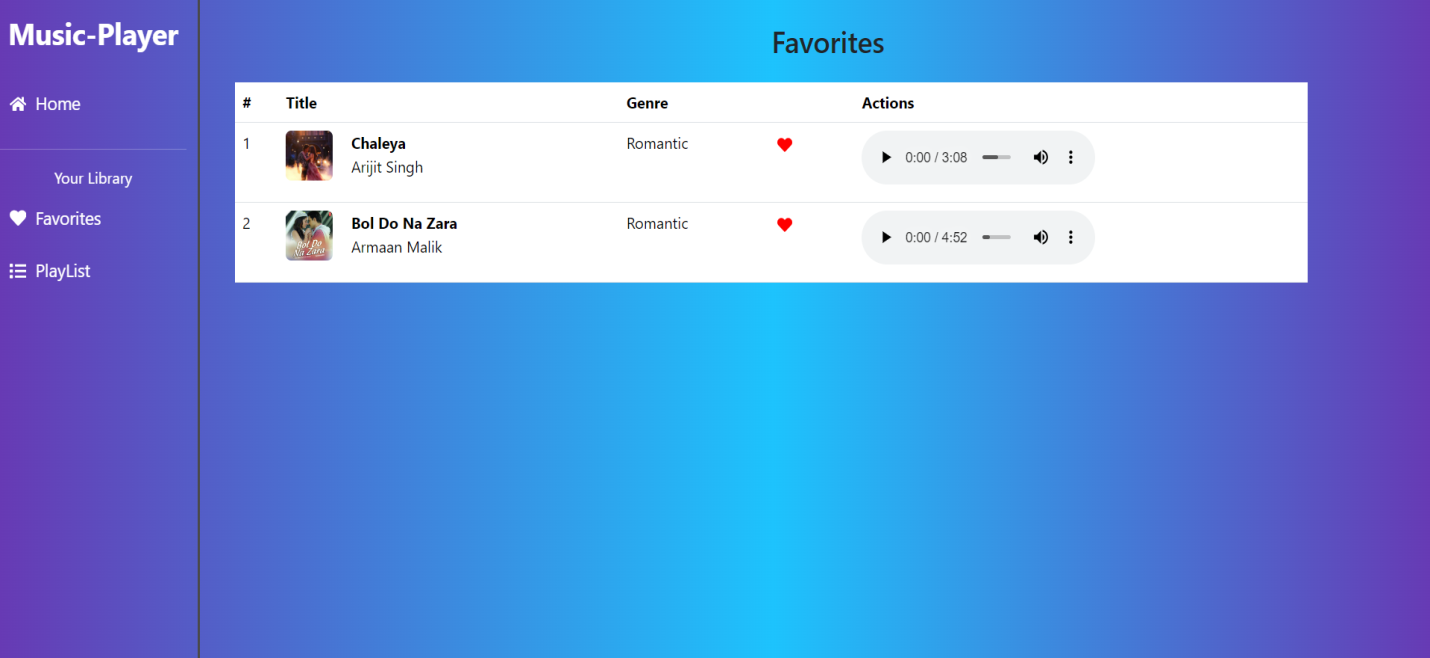
Here are some of the screenshots of the application.

* Homepage:







****