## RhythmicTunes: Your Melodic Companion

**1. Introduction**

* **Project Title:** RhythmicTunes: Your Melodic Companion
* **Team Members:** VishvaKumar K **(Team Leader)**

Vijaya Kumar V **(Team Member)**

Mathews Shine Joshi **(Team Member)**

Bharathi E **(Team Member)**

Aswin Raja K **(Team Member)**

* **GitHub Link:** [**vishv2k/music-streaming-app**](https://github.com/vishv2k/music-streaming-app)

**2. Project Overview**

* **Purpose:**
  + A web-based music streaming application that allows users to browse, play, and manage music.
  + Provides an interactive and smooth UI for users to experience seamless music playback.
* **Features:**
  + Browse and search for music tracks.
  + Play, pause, and skip tracks with an interactive music player.
  + Create and manage playlists.
  + Responsive design for mobile and desktop users.

**3. Architecture**

* **Component Structure:**
  + **App Component:** Main entry point of the application.
  + **Navbar:** Contains navigation links.
  + **Music Player:** Handles playback controls and track display.
  + **Playlist Manager:** Allows users to create and manage playlists.
  + **Search Component:** Enables searching for music tracks.
* **State Management:**
  + **Context API** used for managing global state across the application.
* **Routing:**
  + Implemented using **React Router** for navigation between home and playlists pages.

**4. Setup Instructions**

* **Prerequisites:**
  + Node.js (Latest LTS version recommended)
  + npm or yarn package manager
* **Installation:**
* git clone https://github.com/vishv2k/music-streaming-app.git
* cd music-streaming-app
* npm install
* **Environment Variables:**
  + Create a .env file and add required API keys and configurations.

**5. Folder Structure**

/music-streaming-app

│── public/ # Static assets

│── src/

│ │── components/ # Reusable UI components

│ │── pages/ # Page components

│ │── assets/ # Images, icons, etc.

│ │── utils/ # Helper functions

│ │── context/ # Context API state management

│── package.json

│── README.md

**6. Running the Application**

npm install

npm start

**7. Component Documentation**

* **MusicPlayer**
  + Props: track, isPlaying, onPlay, onPause, onNext
* **SearchComponent**
  + Props: onSearch, query
* **PlaylistManager**
  + Props: playlists, onCreatePlaylist, onDeletePlaylist

**8. State Management**

* **Global State:**
  + Managed using Context API for user authentication and music playback.
* **Local State:**
  + Controlled using useState hooks for UI interactions like toggling modals and managing search queries.

**9. User Interface**

* **Main Pages:**
  + Home Page
  + Music Player Interface
  + Playlist Management Page

**10. Styling**

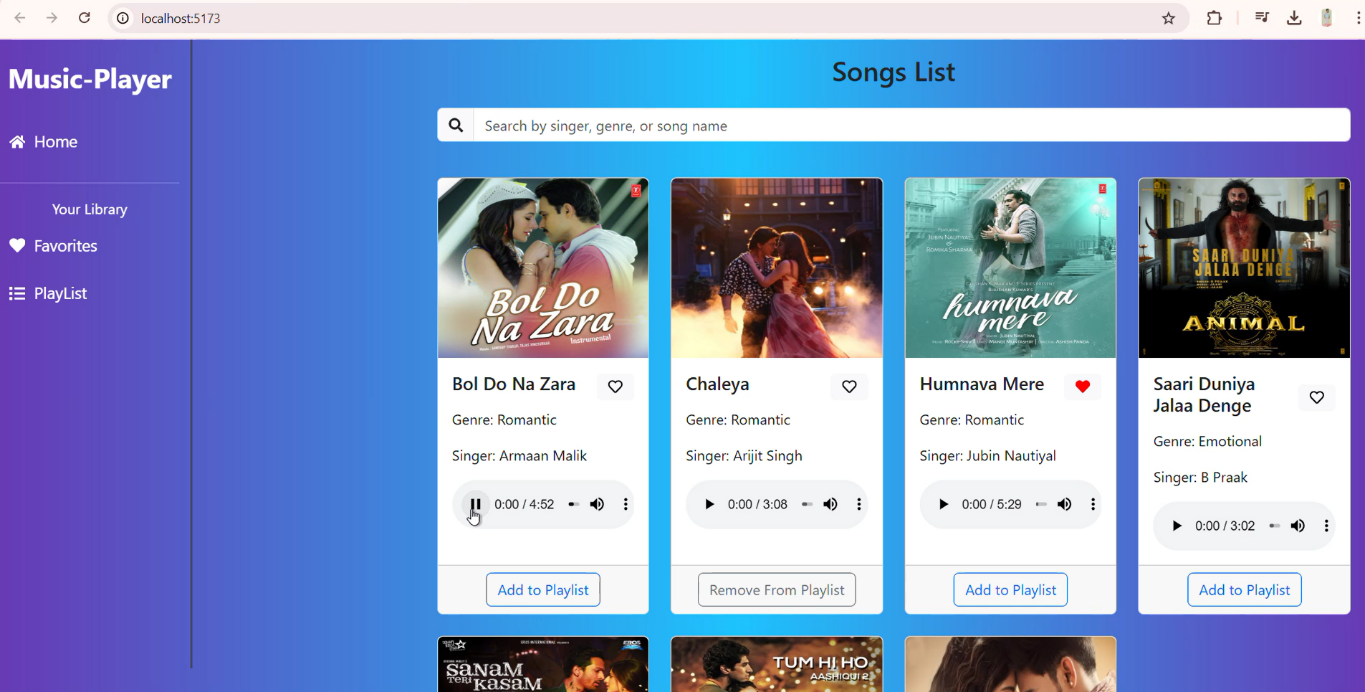
* **CSS Frameworks/Libraries:**
  + Tailwind CSS used for styling.
  + Custom CSS components for responsive UI.
* **Theming:**
  + Light/Dark mode support for better user experience.
  + **Gradient Background:** The app features a visually appealing gradient background (blue to purple) for a modern and dynamic look.
  + **Sidebar Styling:** The sidebar uses a dark purple theme, providing a contrast with the main content area, improving readability and user navigation.

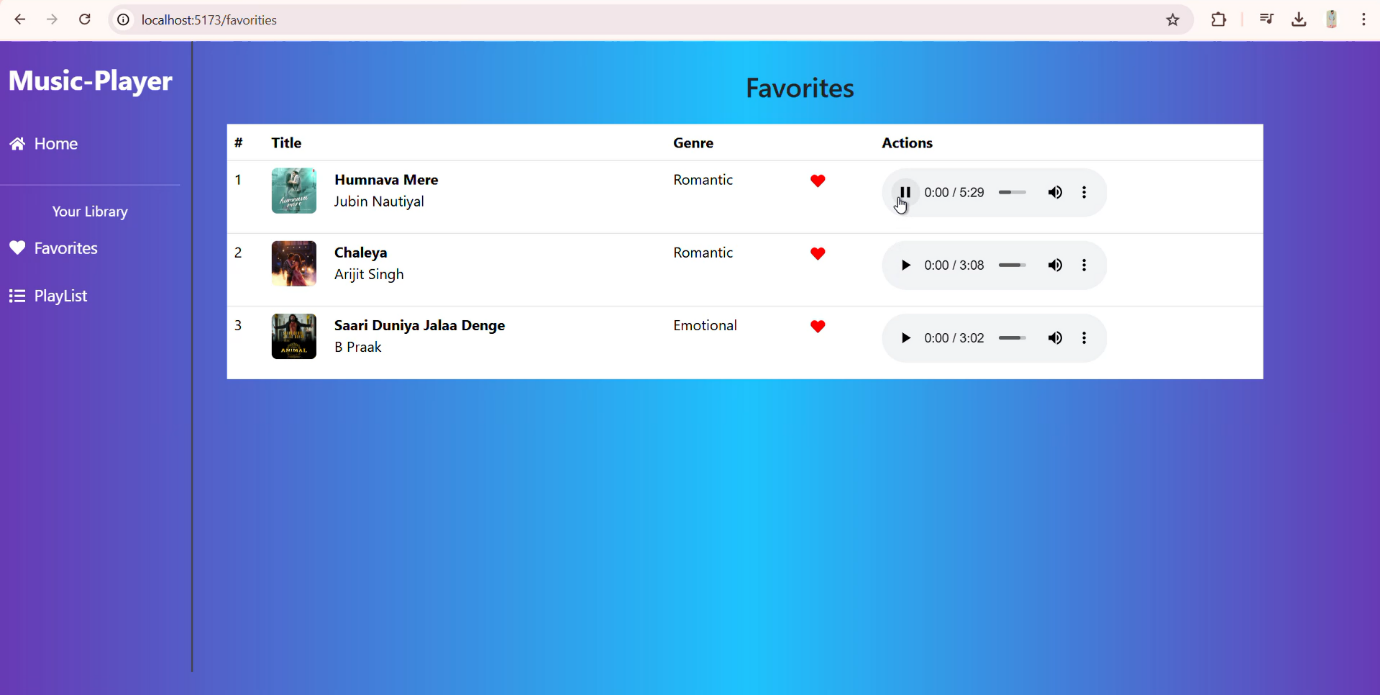
**11. Testing**

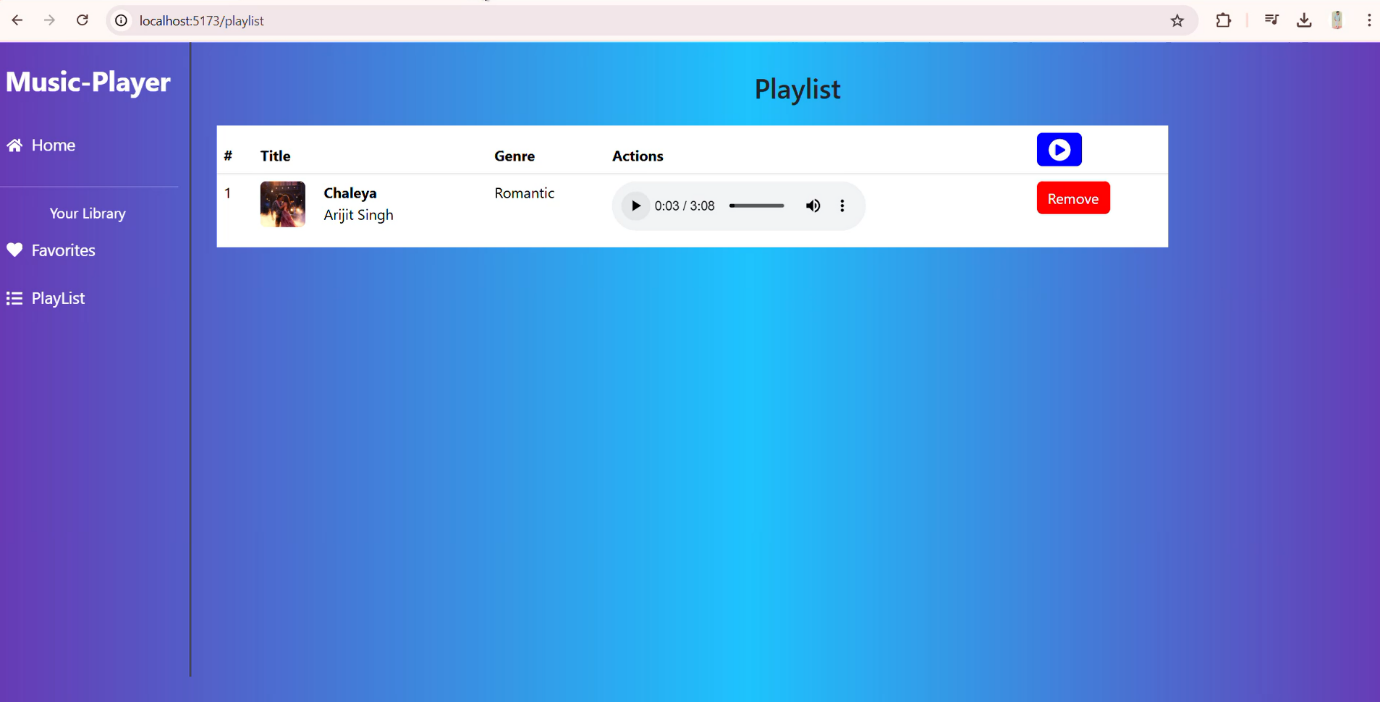
* **Testing Strategy:**
  + Unit testing with Jest and React Testing Library.
* **Code Coverage:**
  + Ensured critical components have test cases.

**12. Screenshots or Demo**

* Demo Link: <https://drive.google.com/file/d/1GttqFOQoRUgA1N8033HW2TcB3q_420Pk/view?usp=sharing>







**13. Known Issues**

* Some API responses may be slow due to third-party dependencies.
* Mobile UI needs further optimization for smaller screens.

**14. Future Enhancements**

* Integration with Spotify API for real-time music recommendations.
* Improved search functionality with filters.
* Adding offline music playback capability.