**Frontend Development with React.js**

**Project Documentation format**

1. **Introduction**
   * **Project Title**: Rhythemic tunes[music streaming app]

* **Team Members**: **Team Members**:

|  |  |
| --- | --- |
| **NAME** | **EMAIL ID** |
| Rindhiya M | lakshmirindhi@gmail.com |
| Poornima S | padmapoorni11@gmail.com |
| Shanthini D | shanthinidesappan@gmail.com |
| Pooja S | poojasrani12@gmail.com |

1. **Project Overview**
   * **Objectives**: The primary goal of Music Streaming is to provide a seamless platform for music enthusiasts, enjoying, and sharing diverse musical experiences.
   * **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.
   * **Features**:
   * **Song Listings:** Display a comprehensive list of available songs with details such as title, artist, genre, and release date.
   * **Playlist Creation**: Empower users to create personalized playlists, adding and organizing songs based on their preferences.
   * **Playback Control**: Implement seamless playback control features, allowing users to play, pause, skip, and adjust volume during music playback.
   * **Offline Listening:** Allow users to download songs for offline listening, enhancing the app's accessibility and convenience.
   * **Search Functionality**: Implement a robust search feature for users to easily find specific songs, artists, or albums within the
2. **Architecture**

**Component Structure**

1. **Frontend Components (User Interface)**

These components handle user interactions and display information

**Music Discovery & Library:**

* **Home Screen** – Displays recommended songs, playlists, and new releases.
* **Search Component** – Allows users to search for songs, albums, artists, and playlists.
* **Browse & Categories** – Shows trending music, genres, moods, and top charts.

#### **Playlist & Library Management:**

* **Playlist Creator** – Allows users to create and manage playlists.
* **Favorites & Liked Songs** – Users can save songs and albums they like.
* **History & Recently Played** – Displays the user's listening history.

**routing**:using react router, Routing allows the app to fetch and display the new page with help of URL.User can directly access the application efficiently.

**Setup Instructions**

* + **Prerequisites**:
  + Node.js and npm
  + React.js
  + HTML,CSS,Javascript
  + Visual Studio Code(IDE)
  + Git(Version control)
  + **Installation**:

**1**.**clone the repository:**

* Git clone: <https://github.com/dowmiya/music-streaming-app>
* Demolink: https://drive.google.com/file/d/1-Atvln55GpOtUTMKKRCB\_fjnqfGd-Opa/view?usp=drivesdk.

**2.navigate to the project directory:**

* Cd code

**3.install dependencies:**

* npm install

**4.Start the development server**

* npm run dev

1. **Folder Structure**
   * **Client**:
2. Src/-source code Directory

This the core directory containing all the code for the application.

2.components/-reusable components:

* Favorities.jsx
* Playlist.jsx
* Search.jsx
* Sidebar,jsx
* Uhome.jsx
* Uitem.jsx
* Wishlist.jsx
* Index.jsx
* App .jsx
* Main.jsx

3.assets/-static media files:

* This folder contains images, icons, and other static assets used in the application.

4.Styles/-CSS Styleshee

* Uhome.csss
* Index.css
* App.css
* Sidebar.css

**Running the Application**

* Open the terminal.
* Navigate to the client directory and type: cd db
* Start the frontend server using npm:
* npm dev run

**Styling**

This project uses Bootstrap as the primary CSS framework for styling and

layout. Bootstrap provides a predefined design system, including a

responsive grid system for layout management.

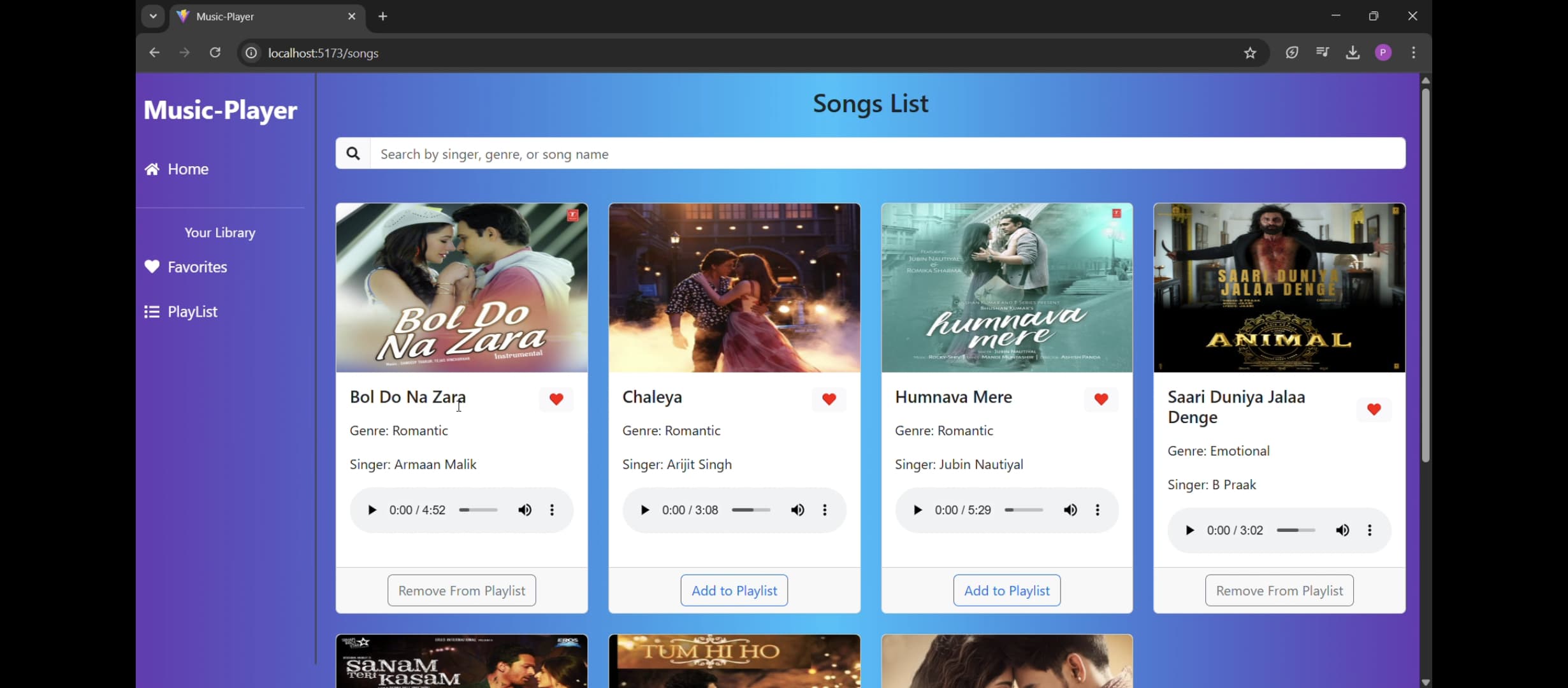
**Theming**:

* Implemented light mode using CSS variables or design system for consistent colors, fonts, and spacing.
* **Testing**

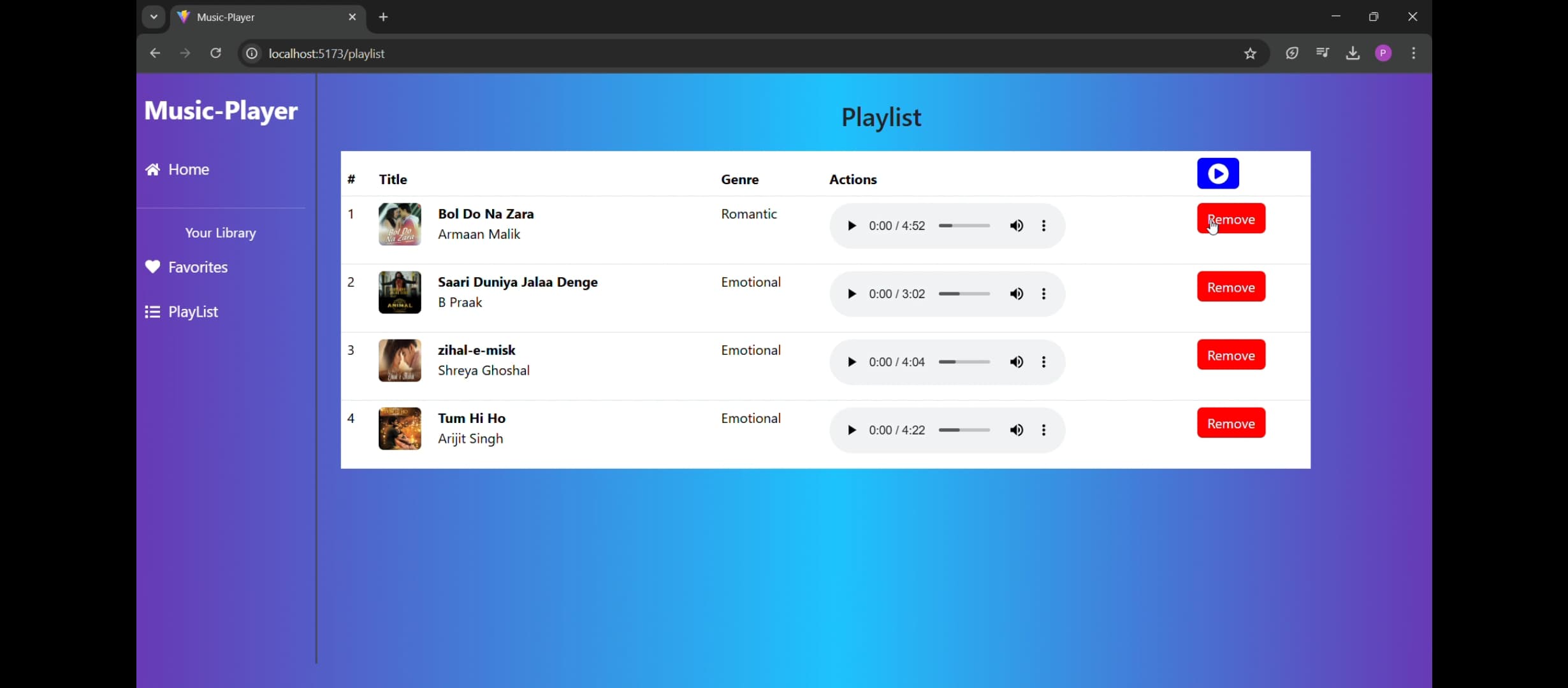
**Testing Strategy**:

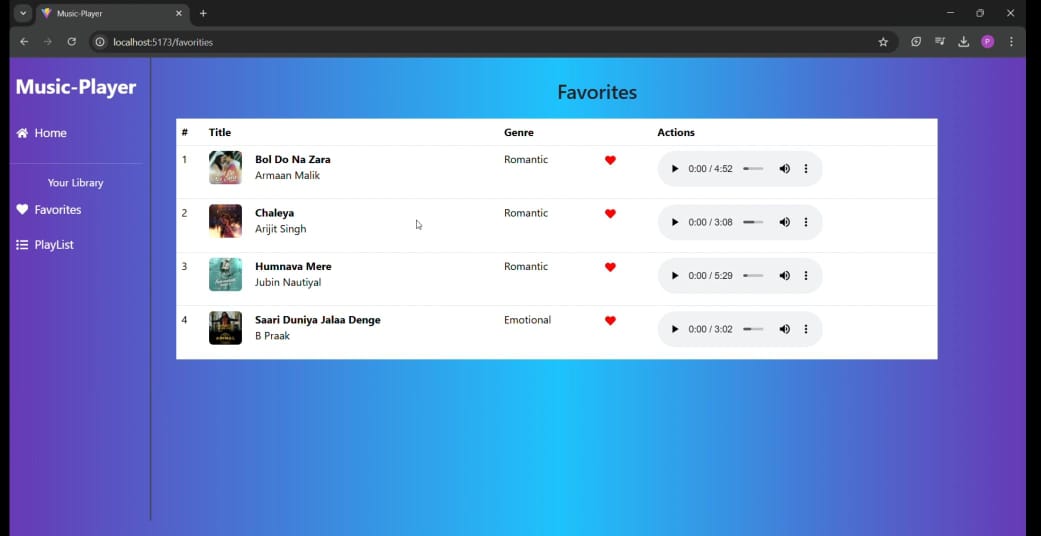
* Unit Testing: Used Jest and React Testing Library for testing components

**User interface:**



* **Screenshots or Demo**





**Known Issues**

* Some UI components may require optimization for mobile devices.

**Future Enhancements**

* **Live Listening Parties** – Users can sync playback and chat in real-time.
* **Artist-Fan Interactions** – Exclusive Q&A sessions, behind-the-scenes content, and live updates from artists.
* **Collaborative Playlists 2.0** – Playlists that update dynamically based on group preferences.