Fronted Development with React.js

MUSIC STREAMING APP

Team Members

MUKESH S

MOHAN KUMAR K

MANOJ KRISHNAN S

KANNAN J

Project Overview

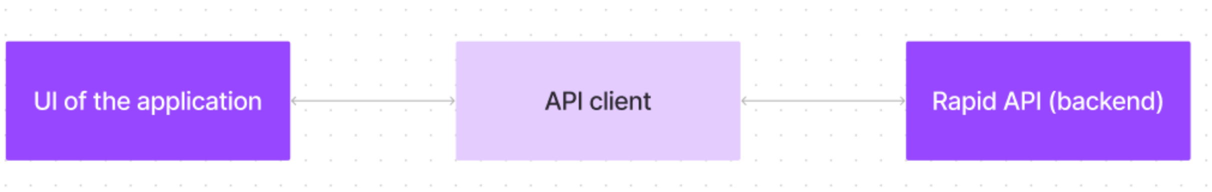
Purpose:

The Music Streaming App allows users to browse, play, and manage a collection of songs. It provides a seamless experience with a user-friendly interface, playlist management, and song recommendations.

Features:

- User authentication and profile management
- Browse and play music from the local library
- Create and manage playlists
- Search functionality for songs and artists
- Responsive UI with Bootstrap & Tailwind CSS
- Mock backend using JSON server
- Light and dark mode support
- Future enhancement: Streaming from an external API

Architecture



Component Structure:

- **Auth Components**: Handles user login, registration, and authentication.
- **Music Player**: Controls song playback.
- **Playlist Manager**: Allows users to create and manage playlists.
- **Search Bar**: Filters songs and artists.

Theme Toggle: Switches between light and dark mode

State Management:

- **React Context API** is used for global state management.
- Local state for UI interactions such as play/pause and theme toggling.

Routing:

- **React Router** is used for page navigation:
    - / - Home
    - /login - User authentication
    - /browse - Browse songs
    - /playlist - User-created playlists

Setup Instructions

Setup Instructions:

Prerequisites:

- Node.js (latest LTS version)
- npm or yarn
- Git

Node.js:

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.

Node.js & npm:

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.

- Install npm in terminal:

```
npm install
```

npm will be installed.

- **Navigate to the project directory:**

```
cd code
```

This will navigate your project directory.

- **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:5173 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

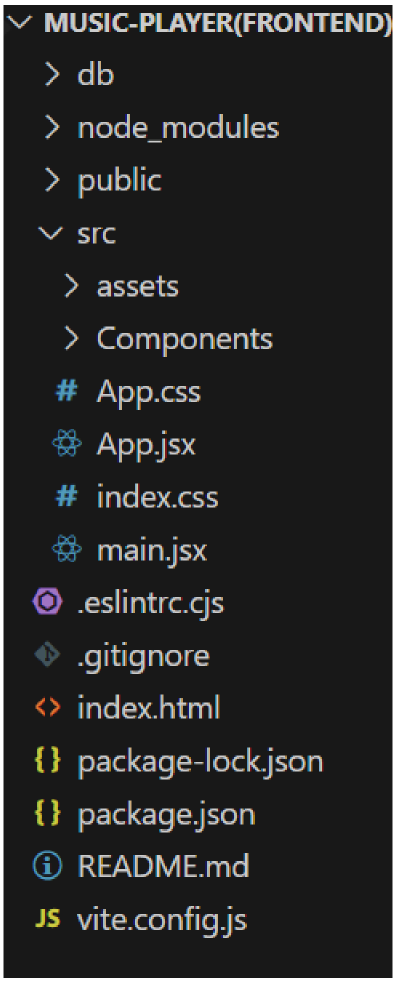
Installation:

1. Clone the repository:
2. git clone https://github.com/MUKESH763-S/MUSIC-PLAYER.git
3. Navigate to the project directory:
4. cd FITNESS-HUB-APP
5. Install dependencies:
6. npm install
7. Start the development server:
8. npm start

Folder Structure

```
MUSIC-STREAMING-APP/
|-- src/
|   |-- components/   # Reusable React components
|   |-- pages/        # Main application pages
|   |-- assets/       # Images and static assets
|   |-- context/      # Global state management using Context API
|   |-- utils/        # Helper functions and custom hooks
|   |-- App.js        # Main application component
|   |-- index.js      # Entry point of the React app
```

Project structure



```
MUSIC-PLAYER(FRONTEND)
  db
  node_modules
  public
  src
    assets
    Components
    App.css
    App.jsx
    index.css
    main.jsx
  .eslintrc.cjs
  .gitignore
  index.html
  package-lock.json
  package.json
  README.md
  vite.config.js
```

Running the Application

Run the code locally:

```
npm run dev
```

Component Documentation

Key Components:

- **MusicPlayer**: Controls song playback.
- **PlaylistManager**: Manages user playlists.
- **SongCard**: Displays song details.
- **ThemeToggle**: Allows switching between light and dark mode.

Reusable Components:

- **Button**: Custom button component with variant support.
- **Modal**: Reusable modal dialog for playlist creation.
- **Card**: Generic component for displaying songs.

State Management

Global State:

- React Context API is used for managing authentication, music player state, and theme toggling.

Local State:

- Component-level states are used for play/pause, search functionality, and UI interactions.

User Interface

Include screenshots or GIFs showcasing:

- Home Page
- Music Player
- Playlist Management
- Search Functionality

Styling

CSS Frameworks/Libraries:

- **Bootstrap** for UI components.
- **Tailwind CSS** for additional styling.

Testing

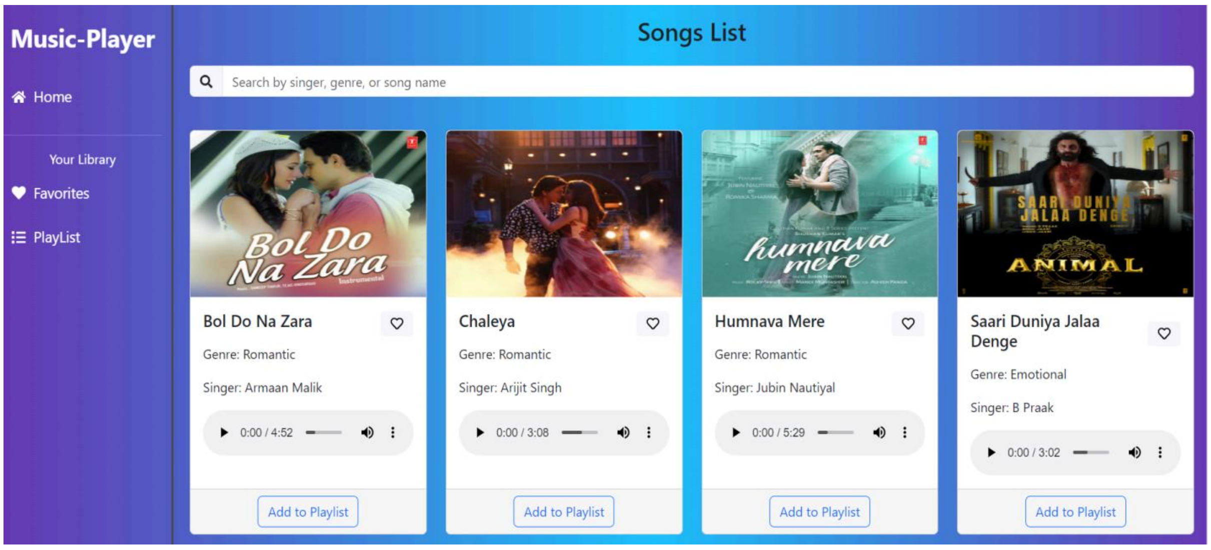
Testing Strategy:

- **Jest** and **React Testing Library** for unit testing.
- **Manual UI Testing** for UX and interaction testing.

Code Coverage:

- Ensuring key functionalities have test coverage above 80%.

Screenshots or Demo


Music-Player

Home

Your Library

Favorites

PlayList

Songs List

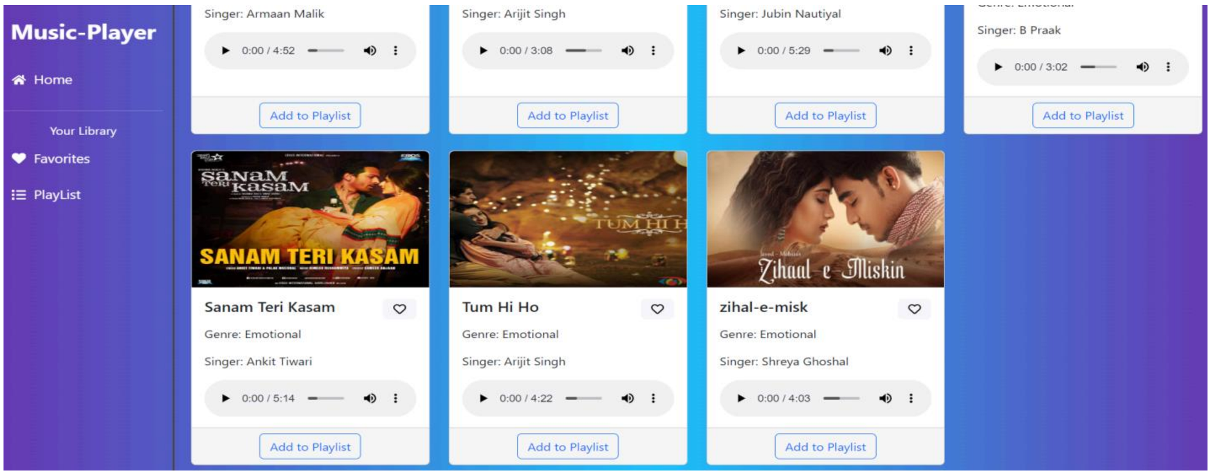
Search by singer, genre, or song name

Bol Do Na Zara
Genre: Romantic
Singer: Armaan Malik
0:00 / 4:52
Add to Playlist

Chaleya
Genre: Romantic
Singer: Arijit Singh
0:00 / 3:08
Add to Playlist

Humnava Mere
Genre: Romantic
Singer: Jubin Nautiyal
0:00 / 5:29
Add to Playlist

Saari Duniya Jalaa Denge
Genre: Emotional
Singer: B Praak
0:00 / 3:02
Add to Playlist


Music-Player

Home

Your Library

Favorites

PlayList

Singer: Armaan Malik
0:00 / 4:52
Add to Playlist

Singer: Arijit Singh
0:00 / 3:08
Add to Playlist

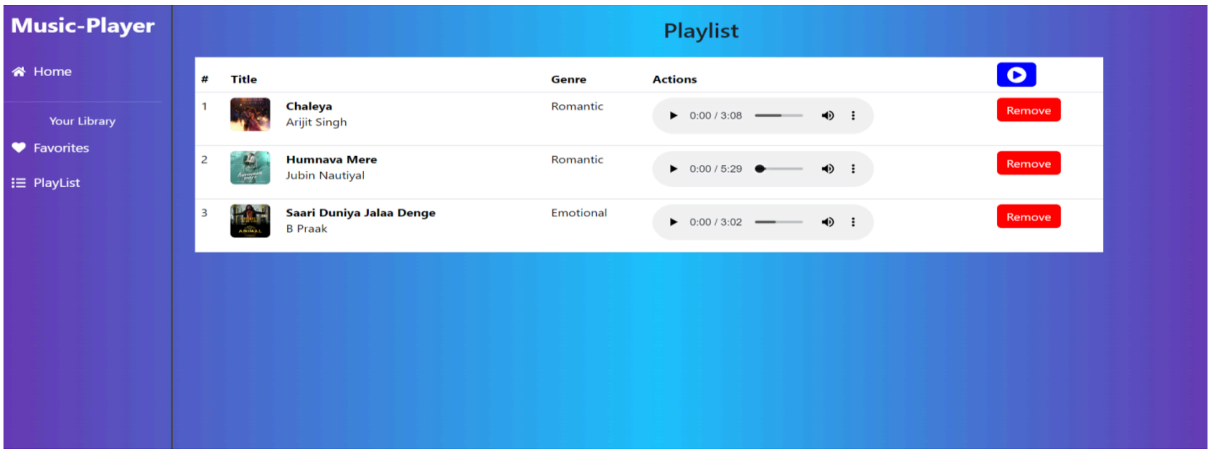
Singer: Jubin Nautiyal
0:00 / 5:29
Add to Playlist

Singer: B Praak
0:00 / 3:02
Add to Playlist

Sanam Teri Kasam
Genre: Emotional
Singer: Ankit Tiwari
0:00 / 5:14
Add to Playlist

Tum Hi Ho
Genre: Emotional
Singer: Arijit Singh
0:00 / 4:22
Add to Playlist

zihal-e-misk
Genre: Emotional
Singer: Shreya Ghoshal
0:00 / 4:03
Add to Playlist


Music-Player

Home

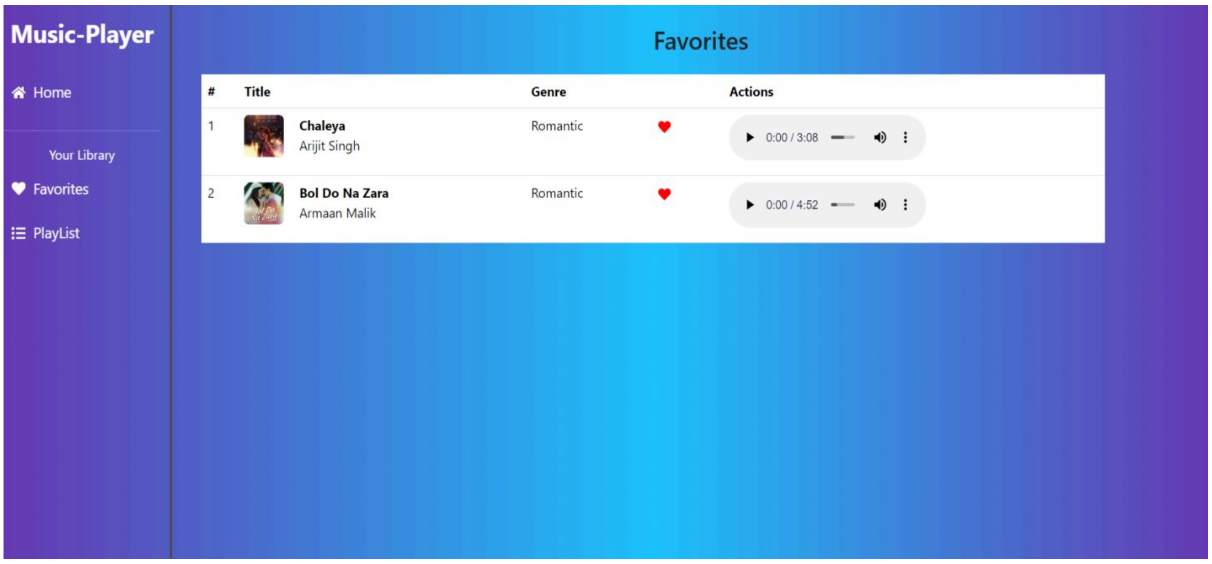
Your Library

Favorites

PlayList

Playlist

#	Title	Genre	Actions	
1	**Chaleya** Arijit Singh	Romantic	0:00 / 3:08	Remove
2	**Humnava Mere** Jubin Nautiyal	Romantic	0:00 / 5:29	Remove
3	**Saari Duniya Jalaa Denge** B Praak	Emotional	0:00 / 3:02	Remove



Favorites

#	Title	Genre		Actions
1	Chaleya Arijit Singh	Romantic	❤	0:00 / 3:08
2	Bol Do Na Zara Armaan Malik	Romantic	❤	0:00 / 4:52

Known Issues

- **Mock Backend**: Uses json-server, which may not work well in production.
- **No TypeScript Support**: Lacks static type checking.
- **Strict ESLint Rules**: May cause warnings or errors if not followed.
- **Limited Song Sources**: Currently only supports local MP3 files.

Future Enhancements

- Integration with external music streaming APIs (Spotify, Apple Music).
- AI-based song recommendations.
- Offline playback support.
- Social sharing of playlists.
- Improved UI animations.

Demo link:
https://drive.google.com/file/d/1iPDWO0jngx53tUMgFdSr1gPskx9gq_sk/view?usp=sharing

GitHub Code link:

https://github.com/MUKESH763-S/MUSIC-PLAYER.git
