**Frontend Development with React.js Project Documentation format.**

**RhythmicTunes: Your Melodic Companion**

1. **Introduction**

Project Title: **RhythmicTunes: Your Melodic Companion**

Team Members:

Team Leader Name: KANNIYAPPAN.S

Team Member Name: JOHN JOSEPH.I

Team Member Name: KAMESH.

Team Member Name: KAMALA KANNAN.V

1. **Project Overview**

Purpose:

RhythmicTunes is a modern music streaming application designed to offer users an immersive listening experience. It provides a vast collection of songs, playlists, and personalized recommendations. The app ensures a seamless interface for browsing, discovering, and enjoying music.

Features:

* Browse Songs by Genre, Artist, or Album
* Personalized Playlist Creation
* Search Functionality with Real-Time Suggestions
* Media Player with Play/Pause, Next/Previous, and Volume Control
* Bookmark Feature for Saving Favorite Tracks
* Light/Dark Mode for Improved User Experience
* Responsive Design for Desktop and Mobile Devices

1. **Architecture**

**Component Structure:**

**App.js -** Root component managing layout and routing

**Header.js** - Navigation bar with search functionality

**SongList.js** - Displays a list of available songs

**SongCard.js** - Individual card showing song details and controls

**Playlist.js** - Manages user-created playlists

**MediaPlayer.js** - Provides audio playback controls

**Footer.js** - Displays copyright and app links

State Management

Context API – Used for managing global states such as theme preferences, playlist data, and bookmarked songs

Routing:

**React Router** - Implemented for seamless navigation between music categories, playlist

Views, and song details

1. **Setup Instructions**

**Prerequisites:**

- Node.js (v18 or higher)

- npm or yarn

**Installation:**

1. Clone the Repository:

\*\*\*Bash

Git clone <https://github.com/darunraj0071/rhythmic-tunes.git> cd rhythmic-tunes

1. Install Dependencies:

\*\*\*Bash

npm install

**3. Run the Application:**

\*\*\*bash

Npm run dev

**Instatating json-server:**

1. **Install dependencies:**

Npm install json-server

1. **Running json-server globally:**

Json-server db/db.json 3000

1. **Folder Structure**

/src

|--/components

----Favorites.js

----PlayList.js

----Search.js

----Sidebar.js

----Sidebar.css

----Song.js

----uhome.css

----uhome.js

----Uitem.js

----Unavbar.js

----Wishlist.js

|--/assets

----react.svg

|--App.js

|--App.css

|--Index.html

1. **Running the Application**

**Frontend Server Command:**

\*\*\*Bash

Npm run dev

1. **Component Documentation**

**Key Components:**

**Header:** Provides navigation, search functionality, and theme toggling

**SongList**: Displays categorized or searched songs in a structured format

**SongCard:** Displays individual song details with play/pause control

**MediaPlayer**: Controls audio playback, volume, and track navigation

**Reusable Components**:

**Button Component**: Used for actions like “Play,” “Pause,” or “Add to Playlist”

**Modal Component**: Displays song details or playlist creation forms.

1. **State Management**

**Global State:** Managed via Context API for bookmarks, playlists, and theme control

**Local State**: Managed using React’s ‘useState() for track progress, user interactions, and dynamic updates.

1. **User Interface**

- Clean, minimalistic UI designed for seamless navigation

- Fully responsive layout with smooth transitions

- Light/Dark mode toggle for personalized visual comfort

1. **Styling**

**CSS Frameworks/Libraries:**

**Tailwind CSS** - Used for fast and modern UI design

**Theming:**

* Integrated theme switcher for enhanced user experience

1. **Testing**

**Testing Strategy:**

- Jest for unit testing

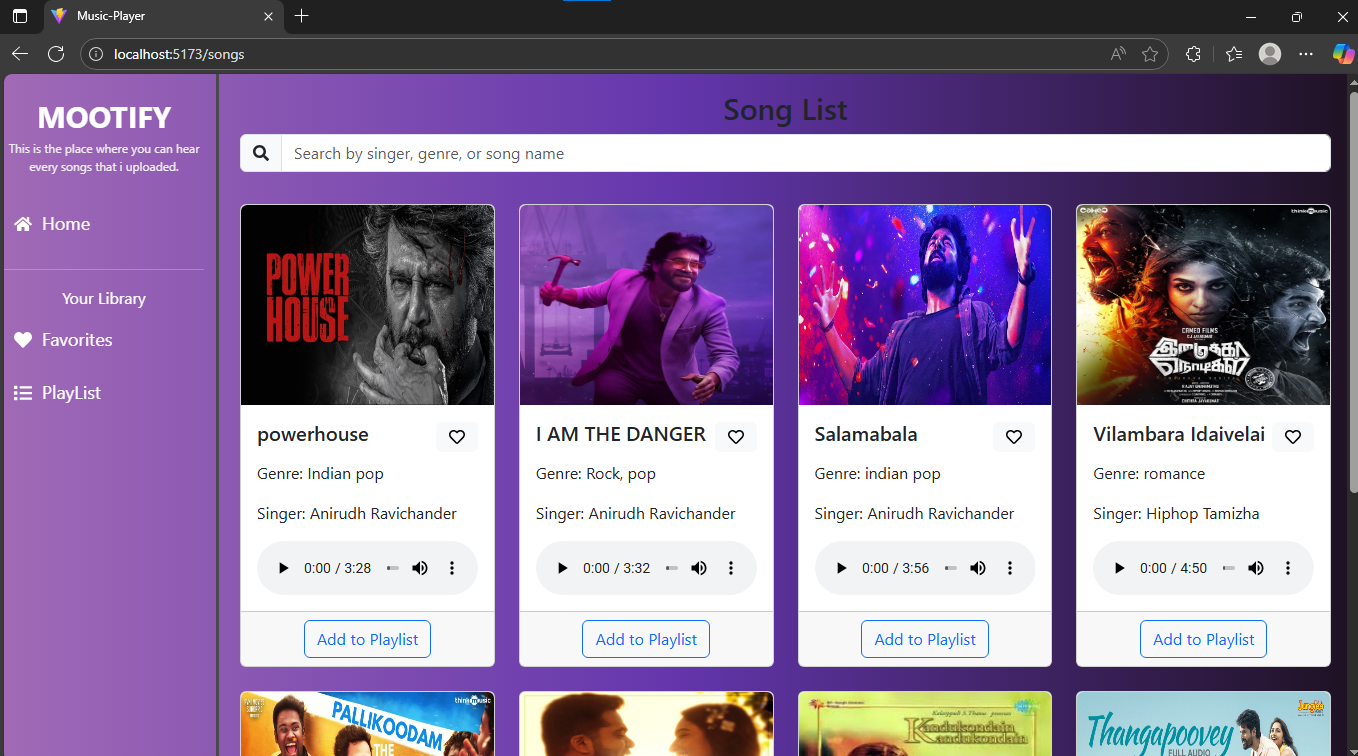
-React Testing Library for comprehensive component testing

**Code Coverage:**

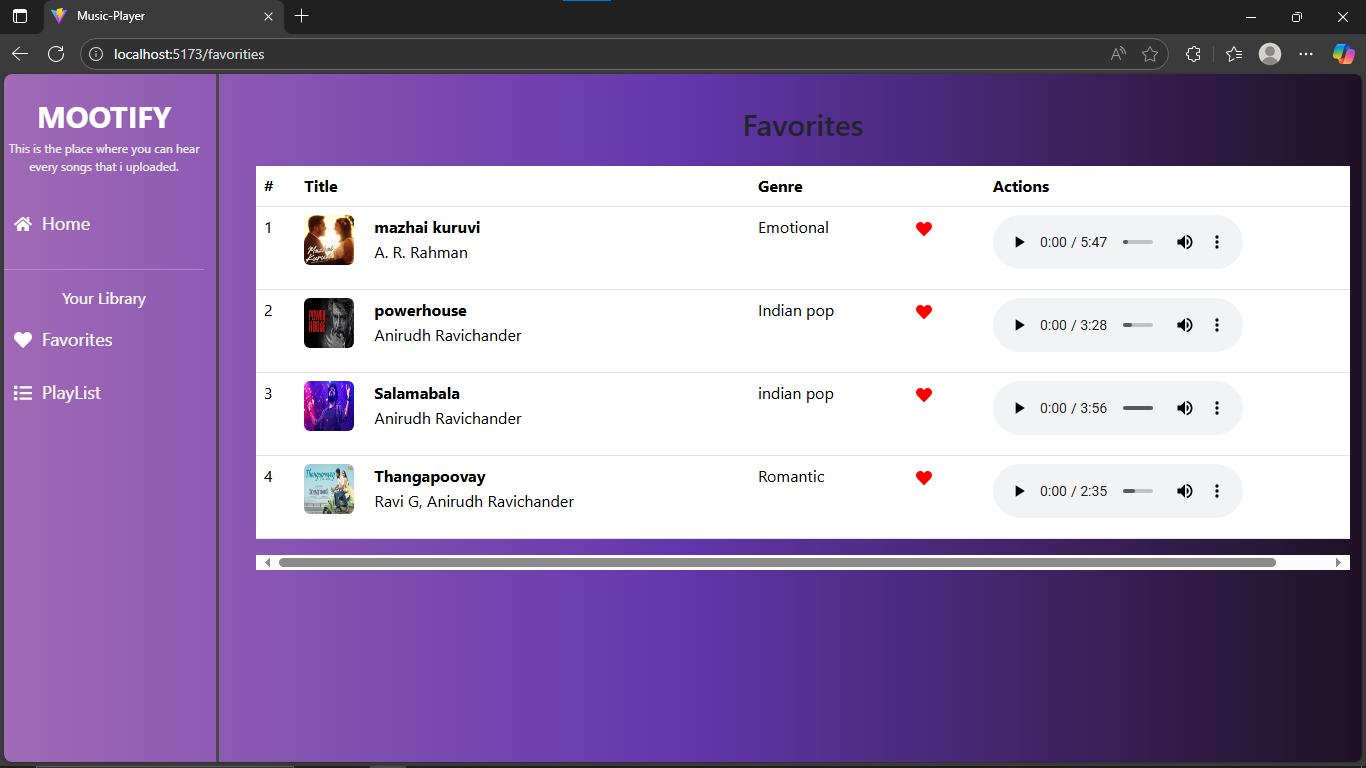
- Ensured high code coverage with Jest reports

1. **Screenshots or Demo**

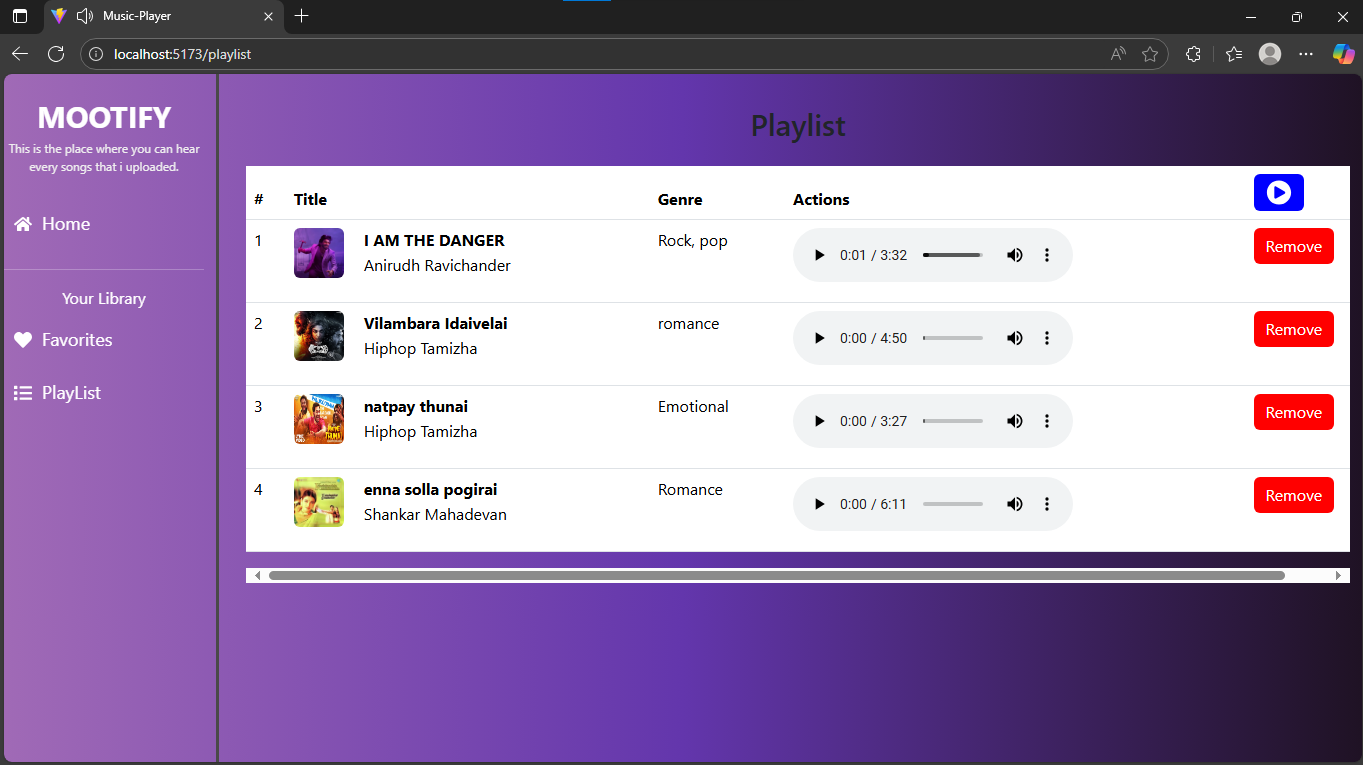
Screenshots:



**Screen shot of Home page:** Where you can search the songs based on genre, artist, or by using song name.



Favorites



Playlist

Live Demo Link:

1. **Known Issues**

- Occasional delays in song loading due to API limits

- Playlist feature may require additional optimization for large data sets

1. **Future Enhancements**

- Implement User Authentication for personalized playlists

- Introduce Podcast and Radio Streaming Integration

- Social Sharing Feature for tracks and playlists

- Offline Mode for saved playlists without an internet connection