

Project Documentation

Rhythmic tune your melodic companion

1. Introduction

- **Project Title:** Rhythmic tune your melodic companion

- **Team ID :** NM2025TMID39933

- **Team Leader:** Janani Sri K

sujithkoteeswari@gmail.com

- **Team Members:**

- Vaishnavi S

vaishunavivaishnavi3@gmail.com

- Dhatchayinee S

dhatchayineedhatchayinee1920@Gmail.com

- Lokeshwari V

lokeshwarilokeshwari520@Gmail.com

2. Project Overview

Purpose: Rhythmic Tunes 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

3. Architecture:

Component Structure:

App. Js — Root component managing layout and routing

Header:.js — Navigation bar with search functionality

SongListjs-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 copy right 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

4. Setup Instructions

Prerequisites:

- Node.js (v18 or higher)
- npm or yarn

Installation Steps:

1.Clone the Repository:

```
"bash
```

```
Git clone https://github.com/darunraj0071/hythmic-tunes. Git
```

```
Cd rhythmic-tunes
```

2.Install Dependencies;

```
"bash
```

```
Npm install
```

- 3. Create env file (for API keys or environment variables):

```
"env
```

```
REACT_APP_MUSIC_APIKEY=your_apikey.here
```

- 4.Run the Application:

```
'bash
```

```
Npm start
```

5. Folder Structure :

```
|_db/  
|_Node_modules/  
|_public/  
|_src/  
  |_assets/  
  |_components/  
  |_App.css/  
  |_App.jsx/  
  |_index.css/  
  |_main.jsx/  
|_eslinttrc.cjs  
|_gitignore.  
|_index.html  
|_package.json  
|_package-lock.json  
|_README.md  
|_vite.co
```

6. Running the Application :

- Frontend:

```
cd client npm
```

```
start
```

•**Backend:**

```
cd server npm  
start
```

- **Access:** Visit <http://localhost:3000>

7. Component Documentation :

Key Components:

Header: Provides navigation, search functionality, and theme toggling
Song List: Displays categorized or searched songs in a structured format
Song Card: Displays individual song details with play/pause control
Media Player: 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

8. 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.

9. User Interface

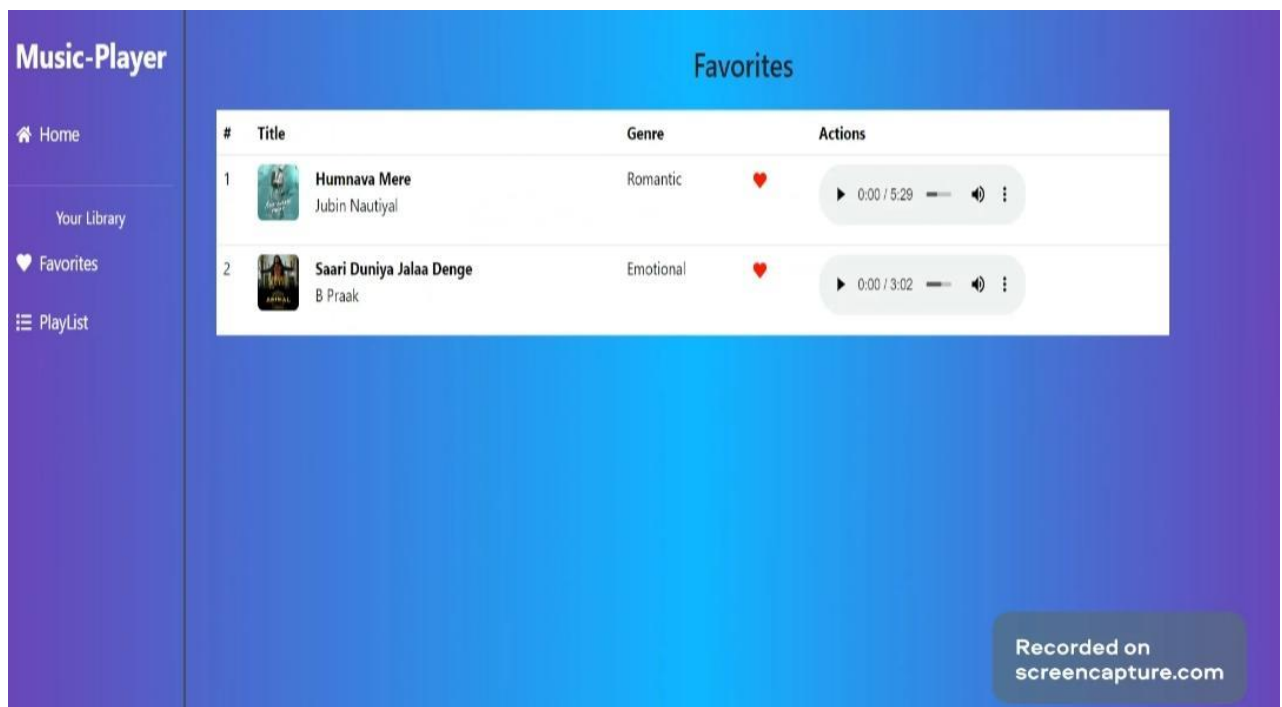
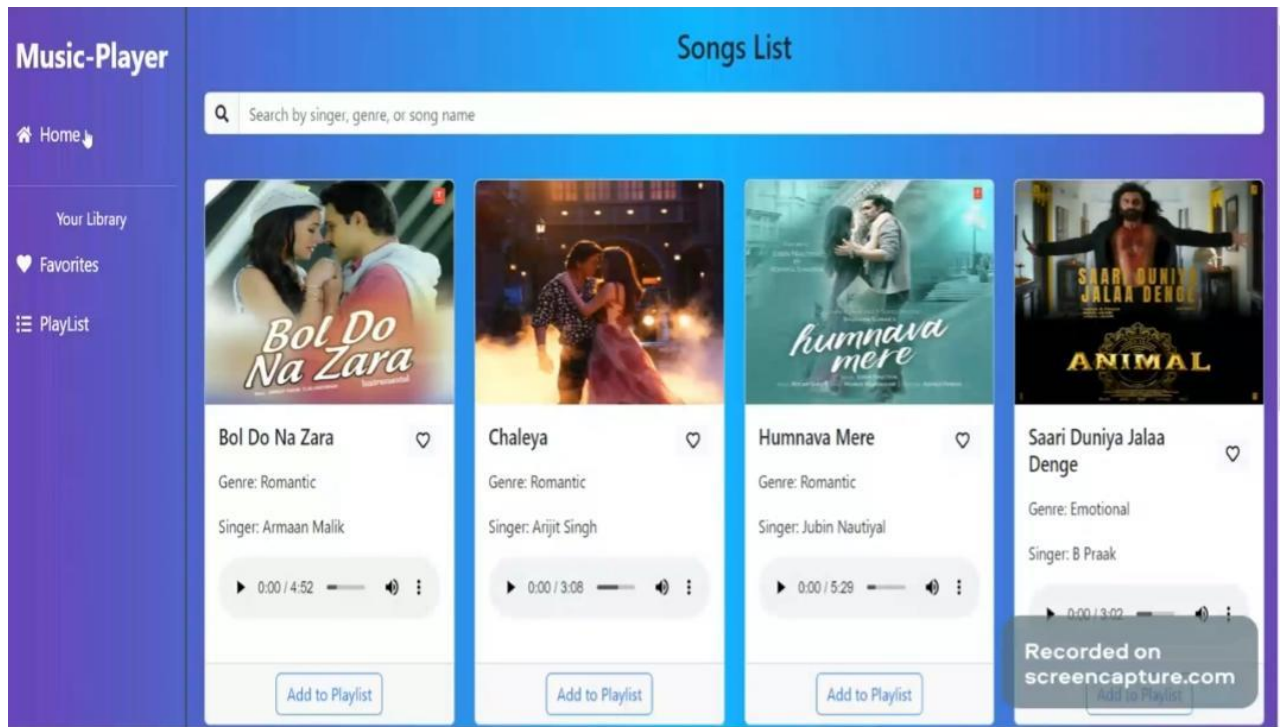
- Clean, minimalistic UI designed for seamless navigation
- Fully responsive layout with smooth transitions
- Light/Dark mode toggle for personalized visual comfort

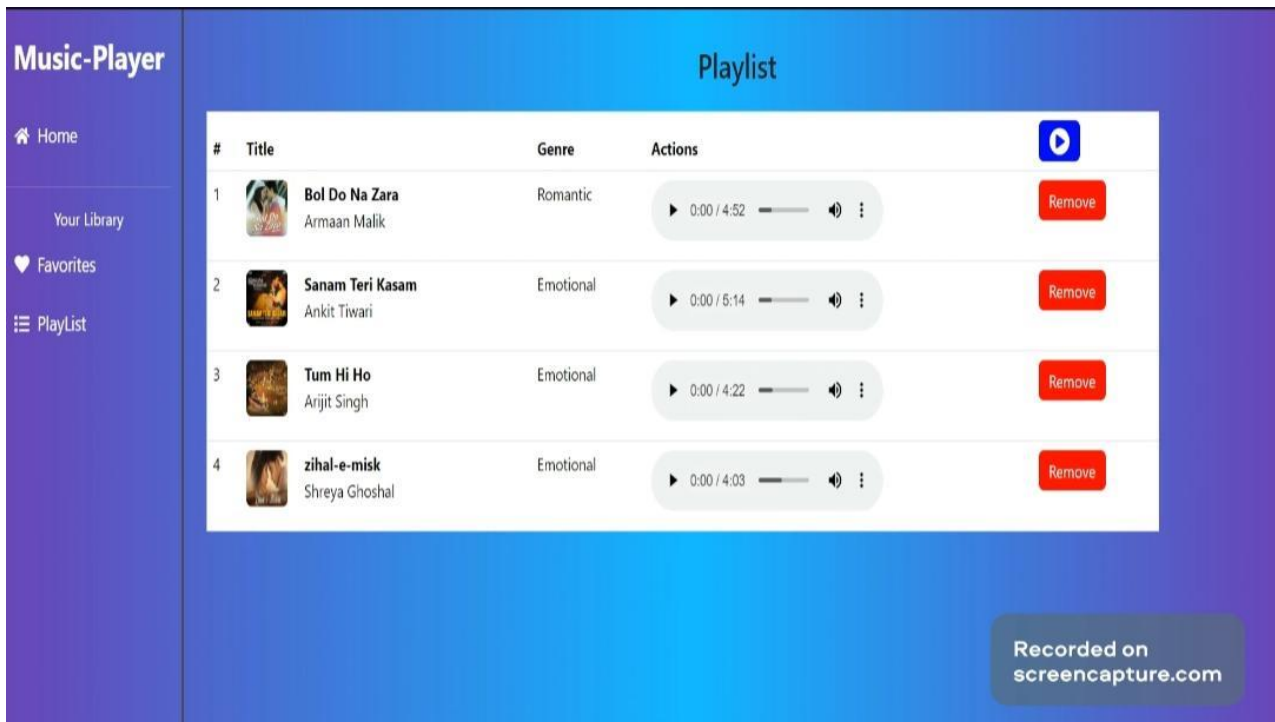
10. Testing

Testing Strategy:

- Jest for unit testing
- React Testing Library for comprehensive component testing

11.Screenshots:





11. Known Issues:

- Occasional delays in song loading due to API Limits
- Playlist feature may require additional optimization for large data sets

13. 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

