

Frontend development with react. js

RHYTHMIC TUNES: YOUR MELODIC COMPANION

1.Introduction

- Project Title:Rhythmic Tunes:Youe melodic companion
- Team Id:NM2025TMID41429
- Team leader:ELANTHAMIL.B/elanthamilsathya@gmail. com
- Team members:
 - KARTHIKA. M/karthikasanthosam50@gmail. com
 - NISHANTHINI. S/snishanthini380@gmail. com

2.Project Overview

- Purpose:

Rhythmic Tunes ia 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
 - Light/Dark mode for improved user experience
 - Responsive design for Desktop and Mobile Devices

3.Architecture

Component Structure:

App. js—Root component managing layout and routing
Header. js—Navigation bar with search functionality
SongList. js—Displays a list of availabke 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

4 Setup Instruction

Prerequisites:

- Node. js
- npm

Installation:

1.Clone the Repository:

```
```bash
git clone https://github.
com/elanthamilsathya/vysya-college_C24UG131CSC025_Title-RHYTHMIC-TUNES-. git
cd rhythmic-tunes
```
```

2.Install Dependencies:

```
```bash
npm install
```
```

3.Create ` . env` file(for API keys):

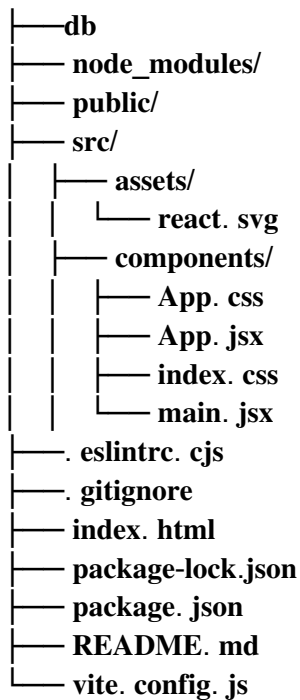
```
```env
REACT_APP_MUSIC_API_KEY=your_api_key_here
```
```

4.Run the Application:

```
```bash
npm run dev
```
```

5.Folder Structure

/src



6. Running the Application

Frontend Server Command:

```
```bash
npm run dev
```
```

- Access: visit <http://localhost:3173>

7. 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 navigations

8. State Management

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

Local State: Managed using React's `useState()` for progress, user interactions, and dynamic updates

9. User interface

-Clean, minimalistic UI designed for seamless navigation

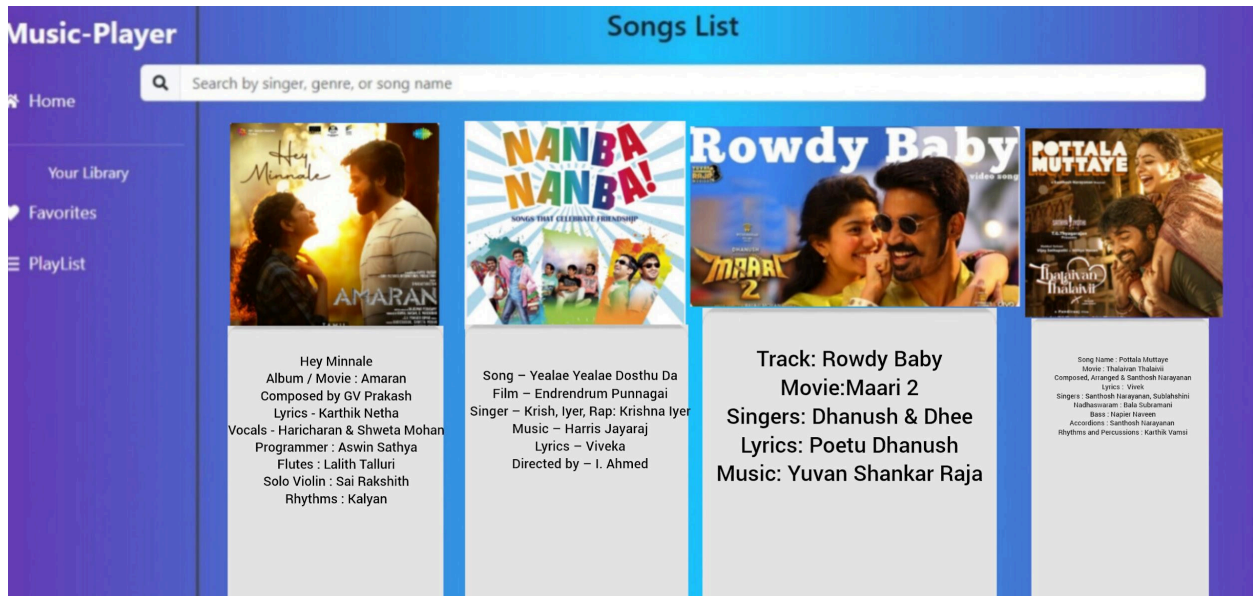
- Fully responsive layout with smooth transitions
- Light/Dark mode toggle for personalized visual comfort

10. Testing

- **Testing strategy:**
 - Test for unit testing
 - React Testing Library for comprehensive component testing
- **Tools:**
 - Chrome dev tools

11. Screenshot or Demo

Screenshot:



12. Known issues

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

13. Future Enhancement

- 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

1. **Team Leader:Elanthamil. B/She worked on the frontend design of our project.**
2. **Team Members:**
 - a. **Karthika. M/She worked on the backend and Database integration.**
 - b. **Nishanthini. S/She focused on testing and documentation.**