# Project Documation: Rhythmic Tunes

1. ***Frontend Development with React.js — Project Documentation***

## 1.Introduction

* **Project Title:** Rhythmic Tunes
* **Team ID:**NM2025TMI46857
* **Team Leader:**Thirisha.S v46359412@gmail.com

|  |  |  |
| --- | --- | --- |
| S.NO | NAME | E-MAIL ID |
| 1 | KAMALINI.T | [kamalini654@gmail.com](mailto:kamalini654@gmail.com) |
| 2 | BHUVANA.P | [pooji0416@gmail.com](mailto:pooji0416@gmail.com) |
| 3 | DEEPIKA SREE.V | [dd7019600@gmail.com](mailto:dd7019600@gmail.com) |
| 4 | DURGA.M | [sankarisankari5418@gmail.com](mailto:sankarisankari5418@gmail.com) |

**2. Project Overview**

* **Purpose:**  
  Rhythmic Tunes is a music streaming web application where users can search, play, and manage their favorite songs or playlists. It aims to deliver a smooth and engaging music-listening experience using a modern React.js frontend.
* **Features:**
  + Search and play songs
  + Create and manage playlists
  + Like and save favorite tracks
  + Responsive UI for mobile and desktop
  + Smooth audio player integration
  + Dark/light theme toggle

**3. Architecture**

* **Component Structure:**  
  Major components include:
  + Navbar: Navigation and theme toggle
  + SearchBar: For searching songs
  + SongList: Displays list of tracks
  + Player: Controls audio playback
  + Playlist: Manage user playlists
* **State Management:**  
  Using **Context API** for global state (e.g., current track, player state), and local state for form controls and UI toggles.
* **Routing:**  
  Implemented with **React Router v6**:
  + / – Home
  + /search – Search Results
  + /playlist/:id – Playlist View
  + /favorites – Liked Songs

**4. Setup Instructions**

* **Prerequisites:**
  + Node.js ≥ 16.x
  + npm or yarn
* **Installation:**
* git clone https://github.com/yourusername/rhythmic-tunes.git
* cd rhythmic-tunes
* npm install
* npm start

**5. Folder Structure**

* **Client:**
* /src
* ├── assets/
* ├── components/
* ├── context/
* ├── pages/
* ├── utils/
* ├── App.js
* └── index.js
* **Utilities:**
  + audioUtils.js – for formatting duration, managing playback
  + api.js – handles song search and playlist APIs

**6. Running the Application**

npm start

Run this command inside the root project folder to start the development server.

**7. Component Documentation**

* **Key Components:**
  + SearchBar: Takes user input and triggers search
  + Player: Accepts current song as prop and controls playback
* **Reusable Components:**
  + Button, Card, Modal, Loader, etc.

**8. State Management**

* **Global State:**  
  Managed via AppContext:
  + currentSong, playlist, isPlaying, theme
* **Local State:**  
  Inside components for:
  + Input fields
  + Toggle states
  + Filter options

**9. User Interface**

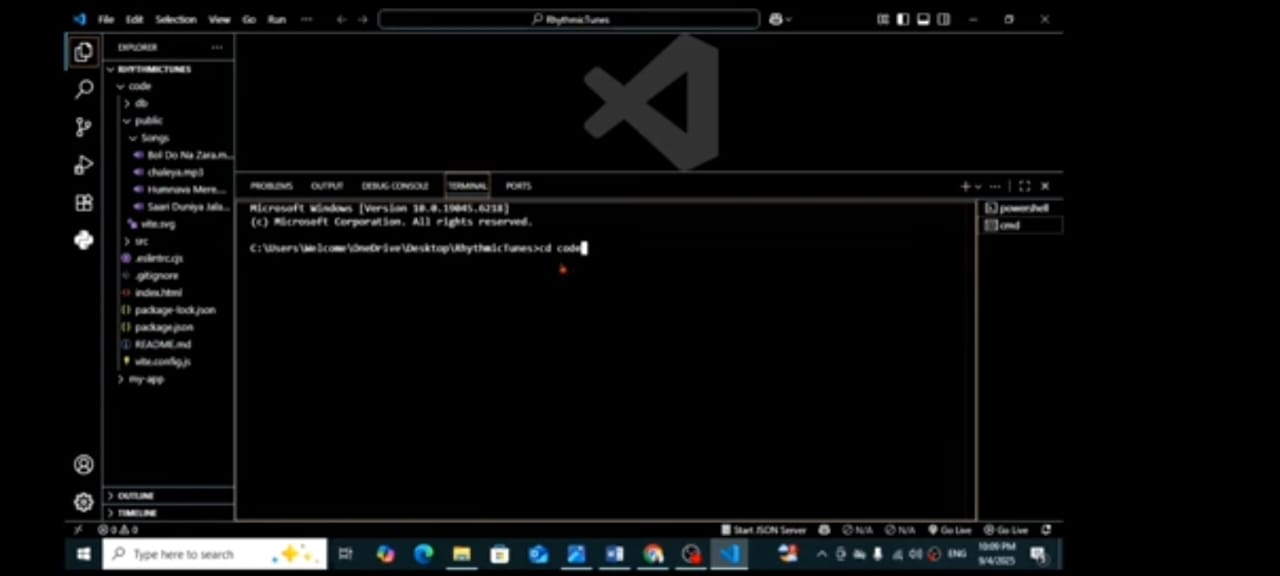
* Dark and light theme toggle
* Mobile-responsive layout
* Screenshots:
  + Home page
  + Song player interface
  + Playlist management

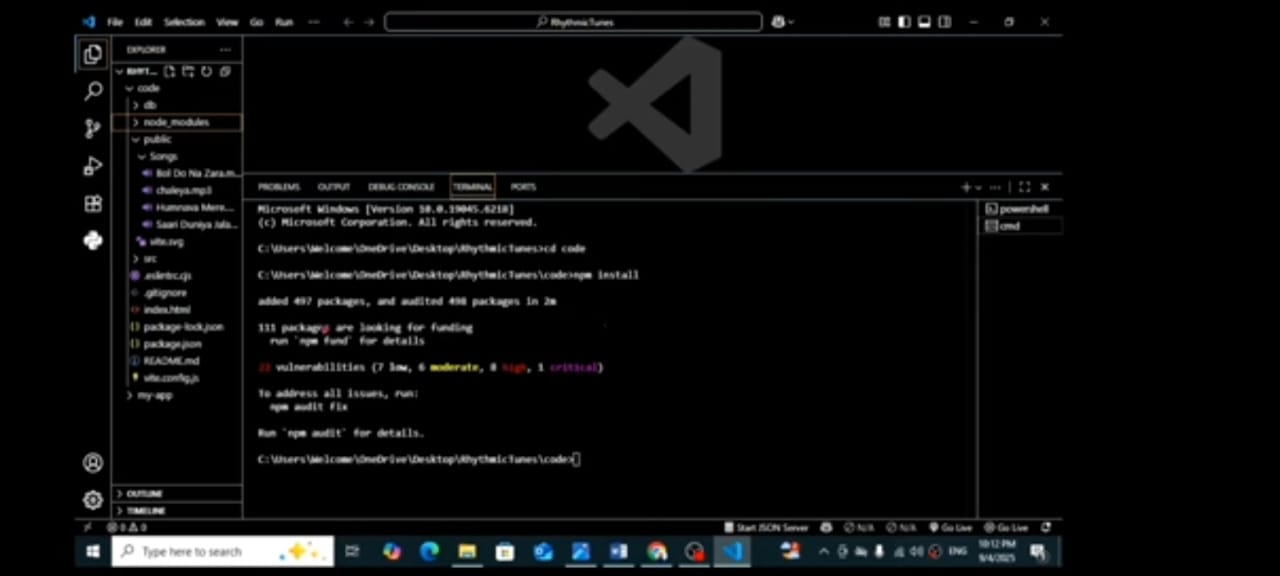
*(You can insert screenshots or link to a demo video)*

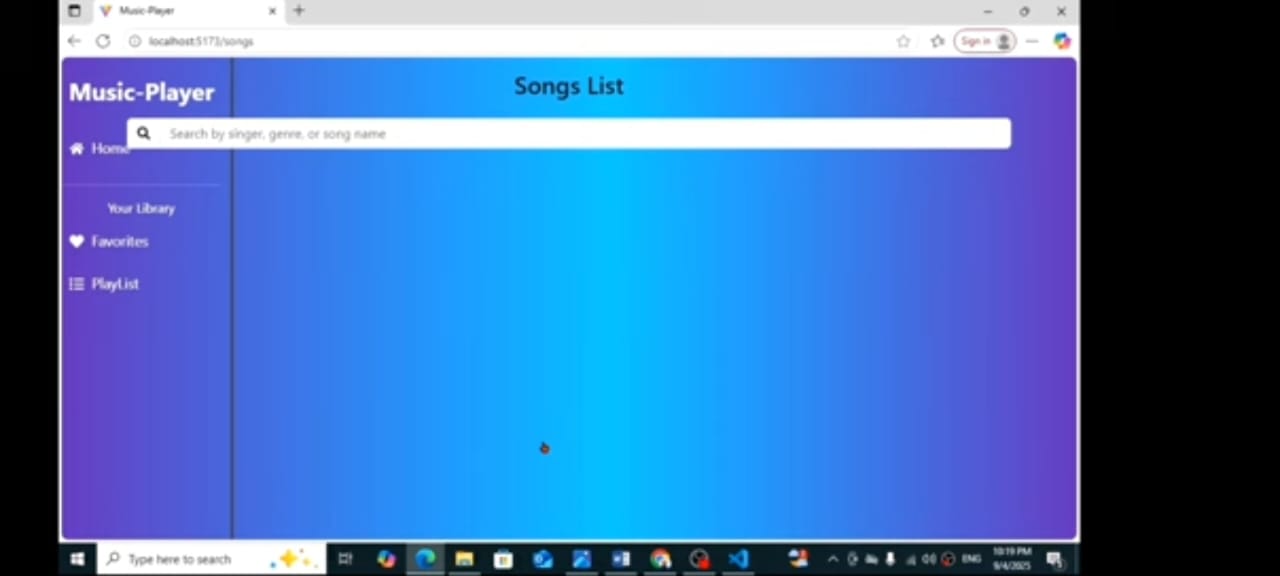
**10. Styling**

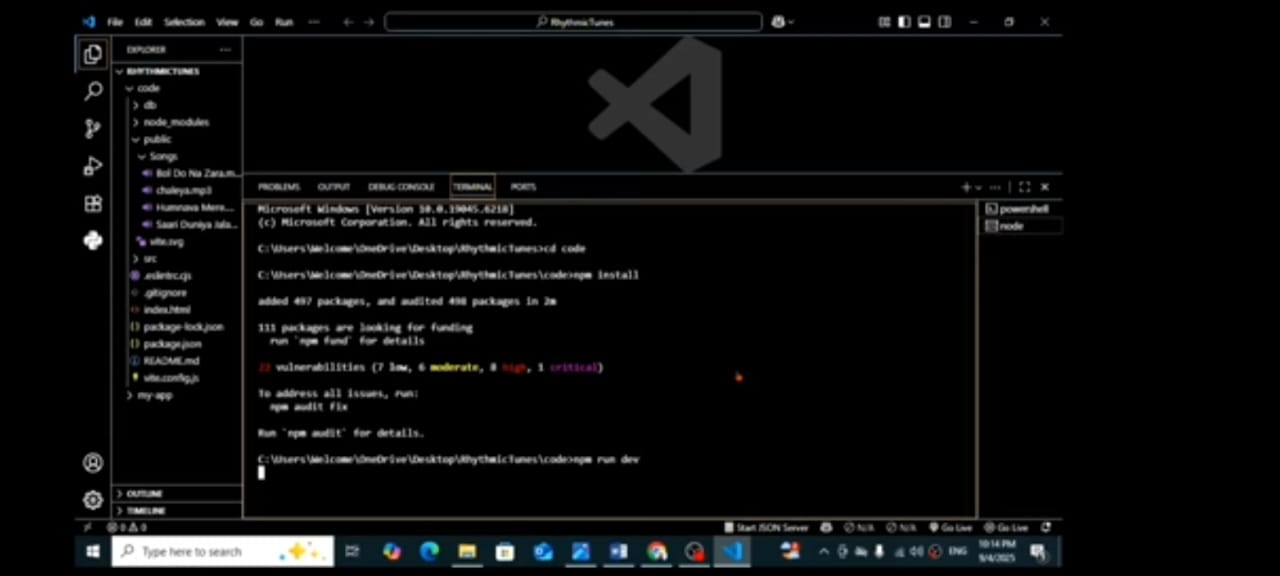
* **CSS Frameworks/Libraries:**
  + **Tailwind CSS** for utility-first styling
  + **Framer Motion** for animations
* **Theming:**  
  Light and dark themes are implemented via context and CSS variables.

**11. Testing**

* **Testing Strategy:**
  + **Unit tests** using **Jest** and **React Testing Library**
  + **Integration tests** for context + component interaction
* **Code Coverage:**  
  Jest coverage reports with minimum 85% target.



**12. Screenshots or Demo **

****

**13. Known Issues**

* Some mobile browsers autoplay restrictions affect initial playback
* Audio might briefly stutter on slow networks
* Playlist names are not validated for duplicates

**14. Future Enhancements**

* Add user login and account sync
* Integrate Spotify/Apple Music APIs
* Song lyrics and artist bios
* Offline caching for favorite tracks
* AI-based playlist recommendat