**PROJECT DOCUMENTATION**

Rhythmic Tunes – Melodic Companion

Introduction

1. Project Title: Rhythmic Tunes – Melodic Companion

2. Team ID: NM2025TMID30016

3. Team Leader: K.Boomika

Leader Mail ID: 202400694@sigc.edu

4. Team Members:

* A.ABINAYA- [202400332@sigc.edu](mailto:202400332@sigc.edu)
* S.DHARSHINI- [202400155@sigc.edu](mailto:202400155@sigc.edu)
* G.JEMIMA- [202400557@sigc.edu](mailto:202400557@sigc.edu)

2. Project Overview:

* Rhythmic Tunes – Melodic Companion is a music-based application designed to provide users with a seamless listening experience. The primary goal of the app is to curate personalized playlists, deliver mood-based song recommendations, and ensure smooth music streaming. It serves as a companion for users who want music that adapts to their preferences, moods, and daily activitie.
* Features:
* Personalized Playlists: Automatically generated playlists tailored to user tastes.
* Mood-Based Song Recommendations: Suggests songs depending on mood (happy, calm, energetic, etc.).
* Seamless Music Streaming: High-quality streaming with minimal buffering and lag.

3. Architecture

* Component Structure:
* Player Component (handles playback controls)
* Playlist Component (manages and displays user playlists)
* Search Component (enables users to find songs/artists)
* Recommendation Component (provides mood-based suggestions)
* Management: Implemented using Redux (or Context API) to manage authentication, active track details, and playlists globally.
* Routing: Navigation handled using React Router, connecting Home, Playlist, Player, and Profile pages.

4. Setup Instructions

* Prerequisites: Node.js, npm, React, API keys for music data.
* Installation Steps:

1. Clone the repository: <https://github.com/Kboomika/Rhythmic-tunes>
2. Run npm install to install dependencies.
3. Configure environment variables with API keys.
4. Start the app using npm start.

5. Folder Structure

* Client:components/ (Player, Playlist, Search, Recommendation)pages/ (Home, Login, Profile)assets/ (images, icons, audio files)
* Utilities:Helper functions for API calls Audio control utilities (play, pause, skip, volume)

6. Running the Application

* Run the following command: npm start. The application will open in your browser on

7. Component Documentation

* Player Component: Provides playback functionality (play, pause, next, previous, volume control).
* Playlist Component: Displays playlists fetched from the backend or generated for the user.
* Search Component: Enables search by track, artist, or album name.
* Recommendation Component: Suggests songs dynamically based on moods and listening history.

8. State Management

* Global State: User authentication, Active track details, User playlists and preferences.
* Local State: Volume control, Play/pause toggle, UI state (theme, modals)

9. User Interface

* Player Screen: Minimalistic music player with essential controls.
* Playlist View: List of songs with options to add, remove, or play.
* Recommendation Screen: Personalized mood-based song suggestions.
* (Screenshots can be added once UI is designed.)

10. Styling

* Libraries Used: Styled-components or Tailwind CSS for fast UI development.
* Theming: Light and dark mode toggle for better user experience.

11. Testing

* Strategy:Unit testing with Jest for components, UI testing with React Testing Library,End-to-end testing for login and streaming flows
* Coverage: Focus on playback controls, search functionality, and playlist generation.

12. Screenshots or Demo

* Add UI screenshots of player, playlist, and recommendation screens.
* Provide demo link when deployed.[(https://drive.google.com/file/d/1ZOBXYLMrCgU0W3AR8RbXP\_9CRtSlopbw/view?usp=drive\_link)](https://drive.google.com/file/d/1ZOBXYLMrCgU0W3AR8RbXP_9CRtSlopbw/view?usp=drive_link)

13. Known Issues

* API rate limits may restrict frequent searches.
* Occasional playback lag on slower networks.

14. Future Enhancements

* AI-powered music recommendations based on activity or environment.
* Social sharing of playlists with friends and community features.
* Offline music download support.