Project Documentation

# Introduction

Project Title: RHYTHMIC TUNES: YOUR MELODIC COMPANION

Team Leader: UMA MAGESHWARI R

Team Members:

• AMRITHA K

• SRISHA T R

•INDUMATHI P

# Project Overview

Purpose: The purpose of this project is to create an engaging music companion app that provides rhythmic tunes and melodies to users, enhancing their listening experience.

Features:

• Personalized playlists based on user preferences

• Rhythm-based tune recommendations

• Easy-to-use interface for seamless navigation

# Architecture

Component Structure: The project will follow a modular structure, with separate components for music playback, playlist management, and user interaction.

State Management: Local state and context API for managing user preferences and playlists.

Routing: Simple navigation across different sections like Home, Playlist, and Settings.

# Setup Instructions

Prerequisites: Node.js, npm

Installation: Clone the repository, run `npm install`, configure environment variables, and then start the project.

# Folder Structure

• components: Contains reusable UI elements  
• pages: Contains main views like Home, Playlist, Settings  
• assets: Stores audio files, images, and icons  
• utils: Helper functions and custom hooks

# Running the Application

Use the command `npm start` to run the application locally.

# Component Documentation

Key Components:  
• MusicPlayer: Handles audio playback  
• PlaylistManager: Manages playlist creation and modifications  
• TuneRecommender: Suggests tunes based on rhythm analysis

# State Management

Global State: Managed using Context API for user preferences and playlists.  
Local State: Used for temporary UI states such as modal visibility.

# User Interface

The application will have an intuitive and modern UI featuring rhythmic animations, interactive buttons, and easy navigation.

# Styling

CSS Frameworks/Libraries: Styled Components for modular CSS-in-JS styling.  
Theming: Support for light and dark modes.

# Testing

Testing Strategy: Unit testing with Jest and integration testing with React Testing Library.  
Code Coverage: Ensured by Jest coverage tools.

# Screenshots or Demo

Screenshots or demo links can be added here once the application is developed.

# Known Issues

Currently no known issues.

# Future Enhancements

• Integration with third-party music APIs  
• Advanced AI-based tune recommendation system  
• Offline listening support