**Project Title**: Rhythmic Tunes : Your Melodic Companion

## ****1. Introduction****

* **Project Title**: Rhythmic Tunes : Your Melodic Companion
* **Team Members**:
  + **ONAIZUNNISSA A** : Coding
  + **SASIKALA G** : Coding Development
  + **ROSHINI S : Document**
  + **RAJESHWARI B :** Demo Video

## ****2. Project Overview****

* **Purpose**:  
  Provide a brief description of what the project aims to achieve.  
  Example: "This project is a task management app that allows users to create, organize, and track tasks."
* **Features**:  
  List key features of the project, such as:
  + User authentication
  + Task creation and management
  + Real-time updates
  + Responsive design

## ****3. Architecture****

* **Component Structure**:  
  Outline the major components of your app and how they interact with each other.  
  Example: "The main components are the TaskList, TaskItem, and UserProfile, all of which are linked via shared state in the App component."
* **State Management**:  
  Describe how state is managed in the project.  
  Example: "We are using React's Context API for global state management and local state for individual component interactions."
* **Routing**:  
  Explain the routing strategy used (e.g., React Router).  
  Example: "We use React Router to manage routes like /login, /dashboard, and /profile."

## ****4. Setup Instructions****

* **Prerequisites**:  
  List software dependencies like Node.js, npm, or others that are needed to run the project.  
  Example: "Node.js version 14+ and npm."
* **Installation**:  
  Provide step-by-step instructions on how to clone the repo, install dependencies, and configure environment variables.
* git clone <repo-url>
* cd <project-folder>
* npm install
  + If there are environment variables, mention how to set them.  
    Example: "Create a *.env* file and set the *REACT\_APP\_API\_URL* variable."

## ****5. Folder Structure****

* **Client**:  
  Describe the organization of your React application.  
  Example: "The *client* folder is structured as follows:
* /src
* /components
* /pages
* /assets
* /utils
  + **components/**: Contains reusable components like Buttons, Form Inputs, etc.
  + **pages/**: Contains the various page-level components, e.g., Home, Dashboard, etc.
  + **assets/**: Contains images, fonts, or other static files.
* **Utilities**:  
  Explain any utility classes, functions, or custom hooks used in the project.  
  Example: "We use a custom *useAuth* hook to manage user authentication and session."

## ****6. Running the Application****

* **Frontend**:  
  To start the frontend server locally, run:
* npm start

This will run the app on http://localhost:3000 by default.

## ****7. Component Documentation****

* **Key Components**:  
  Document each important component, its purpose, and any props it receives.  
  Example: "The *TaskList* component displays a list of tasks and accepts an array of tasks as a prop."
* **Reusable Components**:  
  List any components that can be reused across different parts of the application, e.g., Button, Input.  
  Example: "The *Button* component is a reusable UI element that accepts *label*, *onClick*, and *style* props."

## ****8. State Management****

* **Global State**:  
  Describe how global state is managed across the app.  
  Example: "We use the Context API to store user information globally, such as authentication status and user profile."
* **Local State**:  
  Explain how local states are handled within individual components.  
  Example: "Local states are stored using the *useState* hook for handling form input values or toggling visibility of elements."

## ****9. User Interface****

* **UI Showcase**:  
  Provide screenshots or GIFs showcasing the UI features, such as:
  + Login page
  + Dashboard view
  + Task management UI  
    Example: A GIF of user interacting with the task creation form.

## ****10. Styling****

* **CSS Frameworks/Libraries**:  
  Mention any CSS tools or frameworks used (e.g., Bootstrap, Tailwind, Styled Components).  
  Example: "We use TailwindCSS for utility-first styling and Styled Components for more complex, dynamic styles."
* **Theming**:  
  If the app has a custom design system or theming, explain it here.  
  Example: "The app supports light and dark mode using a custom theme context to toggle styles globally."

## ****11. Testing****

* **Testing Strategy**:  
  Describe your approach for testing components and interactions (e.g., unit tests, integration tests).  
  Example: "We use Jest for unit tests and React Testing Library for DOM testing."
* **Code Coverage**:  
  Mention any tools for measuring code coverage, e.g., Istanbul, Jest coverage reports.  
  Example: "Code coverage is ensured with Jest, and we aim for at least 90% coverage."

## ****12. Screenshots or Demo****

* blob:https://web.whatsapp.com/03ff73b8-3ca9-41ec-be64-ef9e063b3d26

## ****13. Known Issues****

* **Bugs and Limitations**:  
  List any known issues, bugs, or limitations.  
  Example: "Currently, the app does not support mobile landscape mode due to responsive design issues."

## ****14. Future Enhancements****

* **Planned Features**:  
  Outline any future enhancements, features, or improvements.  
  Example: "We plan to integrate real-time updates via WebSockets and add notifications."