**✦✦✦✦✦✦✦✦✦✦✦✦✦✦**

**PROJECT DOCUMENTATION**

**RHYTHMIC TUNES : YOUR MELODIC COMPANION**

**✦✦✦✦✦✦✦✦✦✦✦✦✦✦**

**1. INTRODUCTION**

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

**Team ID** : NM2025TMID29962

**TEAM LEADER** : Kaviya K (**Mail ID**: [202400085@sigc.edu])

**TEAM LEADER** :

- Prabha K (**Mail ID**: [202400805@SIGC.edu])

- Dharshini R (**Mail ID**: [202400695@sigc.edu])

- Janapriya S (**Mail ID**: [202400071@sigc.edu])

**2. PROJECT OVERVIEW**

**PURPOSE**:Rhythmic Tunes is a melodic companion that helps users explore, play, and enjoy music in a personalized way. The platform allows users to listen to rhythmic tunes, discover new songs, and experience an engaging musical environment.

**FEATURES**:

* Music playlist creation
* Song recommendations based on user choice
* Simple and interactive user interface
* User authentication and profile management
* Admin control for content and user monitoring

**3. ARCHITECTURE**

**FRONTEND**: React.js with Bootstrap and Material UI

**BACKEND**: Node.js and Express.js for server logic and APIs

**DATABASE**: MongoDB for storing user profiles, playlists, and music data

**4. SETUP INFORMATION**

**Prerequisites**:

- Node.js  
- MongoDB  
- Git  
- React.js  
- Express.js  
- Mongoose  
- Visual Studio Code

**Installation Steps**:

# Clone the repository git clone <repo-link>  
# Install client dependencies cd client npm install  
# Install server dependencies cd ../server npm install

**5. FOLDER STRUCTURE**

Rhythmic-Tunes/  
│-- client/ # React frontend  
│ └── components/  
│ └── pages/  
│-- server/ # Node.js backend  
│ └── routes/  
│ └── models/  
│ └── controllers/

**6. RUNNING THE APPLICATION**

**Frontend**:  
 cd client  
 npm start

**Backend**:  
 cd server  
 npm start

**Access**: **Open browser** → [http://localhost:3000](http://localhost:3000/)

**7. API DOCUMENTATION**

* **User APIs**:
* /api/user/register
* /api/user/login
* **Music APIs**:
* /api/music/add
* /api/music/:id
* **Playlist APIs**:
* /api/playlist/create
* /api/playlist/:id
* **Chats / Interaction**:
* /api/chat/send
* /api/chat/:userId

**8. AUTHENTICATION**

**🎶** JWT - based authentication for secure login  
**🎶** Middleware protects private routes

**9. USER INTERFACE**

* Landing Page
* Music Dashboard
* Playlist Manager
* Admin Panel
* User Profile Page

**10. TESTING**

**♫** Manual testing during each milestone  
**♫** Tools: Postman for API testing, Chrome Dev Tools for frontend debugging

**11. SCREENSHOTS / DEMO**

**Demo link :** [**https://drive.google.com/drive/folders/1-XOWhbWuifbjNJp9bIkQM93yXb9FVuat?usp=drive\_link**](https://drive.google.com/drive/folders/1-XOWhbWuifbjNJp9bIkQM93yXb9FVuat?usp=drive_link)

**12. KNOWN ISSUES**

**🎧** Limited music dataset (only sample tunes available)  
**🎶** Some UI elements need better responsiveness on mobile

**🎧** Advanced Recommendations – **AI-driven** song and playlist suggestions.

**🎶** **Dark Mode** – Improved UI personalization.

🎧 **Cross-Platform Support** – Mobile apps for iOS and Android.

**🎶 Collaborative Playlists** – Multiple users creating and editing playlists together.

**🎧 Voice Commands** – Hands-free music control.

**🎶 Enhanced Offline Mode** – Full offline access with better storage management.

**🎧 Social Features** – Follow friends, share playlists, and discover through community trends.

**13. FUTURE ENHANCEMENTS**

**♫** **Advanced Recommendations** – AI-driven song and playlist suggestions.

**🎼** **Dark Mode** – Improved UI personalization.

**♫** **Cross-Platform Support** – Mobile apps for iOS and Android.

**🎼** **Collaborative Playlists** – Multiple users creating and editing playlists together.

**♫** **Voice Commands** – Hands-free music control.

**🎼** **Enhanced Offline Mode** – Full offline access with better storage management.

**♫** **Social Features** – Follow friends, share playlists, and discover through community trends.

**🎼**

**♫ ♫ THANK YOU ♫ ♫**

**🎼**