**Project Documentation**

**Rhythmic Tunes- Your Melodic Companion**

**1. Introduction**

**Project Title:** Rhythmic tunes -Your melodic Companion

**Team ID:** [NM2025TMID42833]

**Team Leader**: [AISHWARYA.S & [aishwaryascsc35@gmail.com](mailto:aishwaryascsc35@gmail.com)]

**Team Members:**

[DIVYA.M & [divyalakkamma@gmail.com](mailto:divyalakkamma@gmail.com)]

[HANISHA.G & [ghanisha27@gmail.com](mailto:ghanisha27@gmail.com)]

[INDHUMATHI.B [&indhuindhumathi033@gmail.com](mailto:&indhuindhumathi033@gmail.com) ]

**2. Project Overview**

**Purpose:**

Rhythmic Tunes is the personalized music companion designed to enhance the listening experience. It allows user to explore organize and enjoy their favorite tracks with intelligent recommendation and an interactive interface.

**Features:**

Song search and playlist creation

Personalized music recommendations

Audio player with playback controls

Favorites and history management

User authentication and profile management

Admin dashboard for managing content

**3. Architecture:**

**Frontend:**React.js with Tailwind CSS / Material UI

**Backend:** Node.js with Express.js for server logic and APIs

**Database:** MongoDB for storing users, songs, playlists, and preferences

**Music Integration:** APIs for streaming and metadata retrieval

**4. Setup Instructions:**

**Prerequisites:**

Node.js

MongoDB

Git

React.js

Express.js

Visual Studio Code

**Installation Steps**:

# Clone the repository

git clone <repository-link>

# Install client dependencies

cd client

npm install

# Install server dependencies

cd ../server

npm install

**5. Folder Structure:**

RhythmicTunes/

|-- client/ # React frontend

| |-- components/

| |-- pages/

|

|-- server/ # Node.js backend

| |-- routes/

| |-- models/

| |-- controllers/

|

|-- database/ # MongoDB collections

**6. Running the Application:**

**Frontend:**

cd client

npm start

**Backend:**

cd server

npm start

Access: Visit http://localhost:3000

**7. API Documentation:**

**User:**

POST /api/user/register

POST /api/user/login

**Music:**

GET /api/music/search

POST /api/music/playlist

GET /api/music/favourites

**Admin:**

POST /api/admin/add-song

DELETE /api/admin/remove-song

**8. Authentication:**

JWT-based authentication for secure login

Middleware-protected private routes

**9. User Interface:**

Landing Page with music highlights

Search and Browse Page

Music Player (play, pause, skip, repeat, shuffle)

Playlist & Favorites Section

User Profile Page

Admin Dashboard

**10. Testing:**

**Methods:** Manual testing at development milestones

**Tools:** Postman, Chrome DevTools

**11. Screenshots or Demo link:**

https://drive.google.com/file/d/1LQ34mJR3Ny5wyCQERa3NdTooGZ7ZZGHP/view?usp=drive\_link

**12. Known Issue:**

API rate limits for free music APIs

Offline mode not fully supported

**13. Future Enhancements:**

AI-powered mood-based playlist generation

Social sharing of playlists

Mobile app version