Project Documentation

1. Introduction

• Project Title: Rhythmic Tune – Your Melodic Companion

• Team ID: NM2025TMID44270

• Team Leader: KARTHIKA P ( karthikaprabu427@gmail.com )

• Team Members:

o BHARATHIPRIYA N (bharathinkrcs@gmail.com)

o ANANTHI S (ananthinkr98@gmail.com)

o VAISHNAVI DEVI S (vaishnavinkrcs@gmail.com)

2. Project Overview

• Purpose:

Rhythmic Tune is a web and mobile-based application that allows users to create, customize, and play rhythmic beats and melodies. It aims to help students, beginners, and music enthusiasts experiment with rhythm patterns and tune generation without needing professional instruments.

• Features:

o Beat and tune generator with adjustable tempo.

o Predefined rhythm templates (classical, western, folk, fusion).

o Option to save and share created tunes.

o Visualization of rhythms using waveforms.

o User-friendly interface for beginners.

3. Architecture

• Frontend: React.js with Tailwind CSS for responsive UI.

• Backend: Node.js and Express.js for server logic and APIs.

• Database: MongoDB to store user profiles, tunes, and preferences.

• Audio Processing: Web Audio API / Tone.js for real-time sound generation.

4. Setup Instructions

• Prerequisites:

o Node.js

o MongoDB

o Git

o React.js

o 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-Tune/

|-- client/ # React frontend

| |-- components/

| |-- pages/

|-- server/ # Node.js backend

| |-- routes/

| |-- models/

| |-- controllers/

|-- assets/ # Music samples & icons

6. Running the Application

• Frontend:

• cd client

• npm start

• Backend:

• cd server

• npm start

• Access: Visit http://localhost:3000

7. API Documentation

• User:

o /api/user/register

o /api/user/login

• Tunes:

o /api/tunes/create

o /api/tunes/:id

• Rhythm:

o /api/rhythm/play

o /api/rhythm/save

8. Authentication

• JWT-based authentication for secure login.

• Middleware protects routes for saved tunes.

9. User Interface

• Landing Page

• Rhythm/Tune Generator Page

• Saved Tunes Dashboard

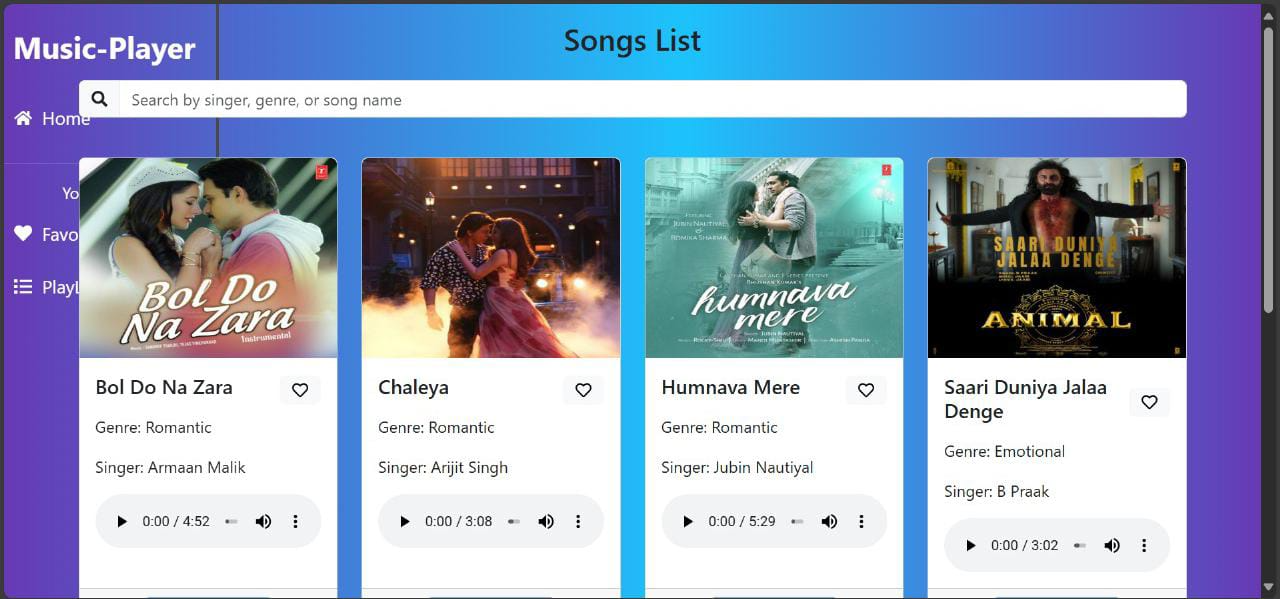
• Admin Panel

10. Testing

• Manual testing for beat generation and playback.

• Tools: Postman, Chrome Dev Tools.

11. Screenshots or Demo



12. Known Issues

• Limited offline functionality.

• Audio lag on low-performance devices.

13. Future Enhancements

• AI-based automatic tune generation.

• Collaboration feature for multiple users to compose together.

• Mobile app deployment with React Native.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_