

Project Documentation

Rythemic Tunes Your Melodic Companion

1. Introduction

- **Project Title :** Rythemic Tunes Your Melodic Companion
- **Team ID :** NM2025TMID41659
- **Team leader : Name -** Gopi Krishnan G
Mail id - gopi102006@gmail.com
- **Team Members :**
 - Member 1 : Name –** Gokul Anand C A
Mail id - gokulanand0223344@gmail.com
 - Member 2 : Name -** Dharshan k
Mail id - dhارشangymnast007@gmail.com
 - Member 3 : Name -** Dhatchanamoorthy S
Mail id - sdhatchanamoorthy22@gmail.com

2. Project Overview

Purpose: Rythemic Tunes is a personalized music application designed to provide users with seamless access to their favorite songs, curated playlists, and an engaging music discovery experience. It serves as a user's melodic companion, enhancing everyday life with rhythm and tunes.

Features:

- Music streaming with play, pause, next, and previous control
- Personalized playlists & user-created playlists
- Favourite songs feature
- Search functionality for songs, artists, and albums

3. Architecture

- **Frontend:** React.js with Bootstrap and Material UI
- **Backend:** Node.js and Express.js managing server logic and API endpoints
- **Database:** MongoDB for storing user details, playlists, and music metadata

4. Setup Instructions

- **Prerequisites:**

- Node.js
- MongoDB
- Git
- React.js (vite)
- Express.js – Mongoose
- Visual Studio Code

- **Installation Steps:**

```
# Clone the repository git clone  
  
# Install client dependencies cd client  
npm install  
  
# Install server dependencies cd  
../server npm install
```

5. Folder Structure

```
| |-- 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: Visit <http://localhost:5173>

7. API Documentation

User:

- /api/user/register
- /api/user/login

Songs APIs:

- /api/songs/upload
- /api/songs/:id

Playlist APIs:

- /api/playlist/create
- /api/playlist/:id

Favourites APIs:

- /api/favourite/add
- /api/favourite/remove/:id

8. Authentication

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

9. User Interface

- Landing Page with trending songs & recommendations
- User Dashboard with playlists and favorites
- Song Details Page
- Admin Panel for managing content

10. Testing

- Manual testing during milestones
- Tools: Postman, Chrome Dev Tools

11. Demo Video

[https://drive.google.com/file/d/1V5Tgn4UHI00bevDFawLfw-W9081-D2Mu/view?usp=drive link](https://drive.google.com/file/d/1V5Tgn4UHI00bevDFawLfw-W9081-D2Mu/view?usp=drive_link)

12. Known Issues

- Limited music storage
- Occasional API response delay

13. Future Enhancements

- AI-based personalized music recommendations
- Offline music playback
- Social features (share playlists, follow friends)
- Integration with wearable devices