**Project Documentation**

RythmicTunes:Your Melodic Companion

# Introduction

* **Project Title:** **RythmicTunes:Your Melodic Companion**
* **Team ID**:NM2025MID42254
* **Team Leader:** PUSHPALATHA S & [pushpalathasubash@gmail.com](mailto:pushpalathasubash@gmail.com)
* **Team Members**
* RANJINI E & [ranjinielumalai177@gmail.com](mailto:ranjinielumalai177@gmail.com)

- RINSHA & [rinishas390@gmail.com](mailto:rinishas390@gmail.com)

- PUGALARASI S & [pugal0531@gmail.com](mailto:pugal0531@gmail.com)

**2. Project Overview:**

RythmicTune is a music streaming and discovery platform that provides personalized music recommendations, curated playlists, and social features for music enthusiasts.

**· Features:**

· Personalized music recommendations based on listening history

· Curated playlists for different moods and activities

· Social features to share music with friends

· Offline listening capability

· Lyrics synchronization

· Artist radio stations

# Architecture

·**Frontend**: React.js with Tailwind CSS and Framer Motion for animations

· **Backend**: Node.js and Express.js handling music streaming and user management

· **Database**: MongoDB stores user profiles, playlists, listening history, and music metadata

· **Audio Processing**: Custom audio engine for seamless streaming

# 4. Setup Instructions

* **Prerequisites:**

Node.js (v16 or higher)

· MongoDB (v5 or higher)

· Git

· React.js

· Express.js

· Mongoose

· Visual Studio Code (recommended)

* **Installation Steps:**

# Clone the repository

Git clone <https://github.com/rythmictune/app.git>

# Install client dependencies

Cd client

Npm install

# Install server dependencies

Cd ../server

Npm install

# 5. Folder Structure

RythmicTune/

|-- client/ # React frontend

|\_\_ components/ # Reusable UI components

|\_\_ pages/ # Application pages

|\_\_ hooks/ # Custom React hooks

|\_\_ utils/ # Helper functions

|-- server/ # Node.js backend

|\_\_ routes/ # API routes

|\_\_ models/ # Database models

|\_\_ controllers/ # Route controllers

|\_\_ middleware/ # Custom middleware

|\_\_ utils/ # Server utilities

|-- audio-engine/ # Custom audio processing

|-- docs/ # Project documentation

# 6. Running the Application

* **Frontend:**

cd client

Npm start

* · Backend:

Cd server

Npm run dev

* · Access: Visit <http://localhost:5173/>

# 7. API Documentation

· Authentication:

· POST /api/auth/register

· POST /api/auth/login

· POST /api/auth/logout

· GET /api/auth/refresh

· Users:

· GET /api/users/profile

· PUT /api/users/profile

· GET /api/users/friends

· POST /api/users/friends

· Music:

· GET /api/music/songs

· GET /api/music/songs/:id

· GET /api/music/artists

· GET /api/music/artists/:id

· GET /api/music/albums

· GET /api/music/albums/:id

· Playlists:

· GET /api/playlists

· POST /api/playlists

· GET /api/playlists/:id

· PUT /api/playlists/:id

· DELETE /api/playlists/:id

· Recommendations:

· GET /api/recommendations

· GET /api/recommendations/:type

# 8. Authentication

· JWT-based authentication for secure login

· Refresh token implementation for persistent sessions

· Middleware protects private routes and resources

· OAuth integration for social media login options

# 9. User Interface

· Landing Page with featured music and genres

· User Dashboard with recent activity and recommendations

· Music Player with full controls and visualizations

· Search functionality with filters

· User Profile with listening statistics

· Social Feed showing friends’ activity

# 10. Teating

· Unit testing with Jest and React Testing Library

· Integration testing for API endpoints

· End-to-end testing with Cypress

· Manual testing during development milestones

· Tools: Postman, Chrome Dev Tools, Lighthouse

# 11. Screenshots or Demo

<https://drive.google.com/file/d/1KtYI8PemtmJ-RzjPwlziMGJXC6F6VpMJ/view?usp=drivesdk>

# 12. Known Issues

Mobile Safari occasionally has audio playback issues

· High memory usage when managing large playlists

· Social notifications sometimes delayed

· Lyrics synchronization can be off by 1-2 seconds for some songs

# 13. Future Enhancements

· Podcast integration

· Collaborative playlist editing

· Music identification feature (Shazam-like)

· Crossfade between tracks

· Smart alarms with gradual music wake-up

· Integration with smart home devices

· Virtual concerts and live events

· Advanced audio equalizer and sound profiles