PROJECT DOCUMENTATION

Project Title

RhythmicTunes: Your Melodic Companion

1.Introduction

Team ID: NM2025TMID47831

Team Leader:

NIRMALA G (nirmalagvan@gmail.com)

Team Members:

OVIYA S (oviyao117@gmail.com)

PRIYADHARSHINI M (pd260202@gmail.com)

KOWSALYA N (nkowsalya43@gmail.com)

Roles and Responsibilities

Team Leader – Nirmala G

- Oversees project progress and task allocation.
- Coordinates team communication and final integration.
- Ensures timely submission and quality of deliverables.

Team Member – Oviya S

- Designs and develops UI/UX.
- Implements music dashboard and playlist features.
- Integrates frontend with backend APIs.

Team Member – Priyadharshini M

- Develops server-side logic and
- Creates APIs for user, music, and playlists.
- Manages authentication and data flow.

Team Member – Kowsalya N

- Conducts testing and debugging.
- Ensures application quality and performance.
- Prepares research inputs and documentation.

2. Project Overview

RhythmiCTune – Your Melody Companion is a modern application designed to enhance the way users interact with music. It serves as both a music player and a melody creation tool, allowing users to explore, compose, and customize tunes according to their personal preferences. The app combines an intuitive interface with intelligent features to provide a seamless and engaging user experience.

The main goals of the project are:

1. User-Friendly Interface: The app is designed to be simple and easy to navigate, ensuring that users of all ages can interact with it without difficulty.

2. Melody Creation and Customization: Users can create their own melodies, modify existing ones, and save them for future use. This feature encourages creativity and experimentation with music.

3. Smart Recommendations: The app analyzes user behavior and preferences to suggest songs, tunes, or melodies that match their taste, making music discovery more personalized.

4. Performance and Reliability: A strong backend system ensures smooth operation, fast response times, and secure storage of user data.

5. Testing and Quality Assurance: Rigorous testing is conducted to identify and fix bugs, ensuring the app functions correctly across different devices and platforms.

3.Architecture

The architecture of RhythmiCTune – Your Melody Companion is designed to ensure smooth interaction between the user, application, and database. It follows a three-tier architecture, which separates the system into Presentation Layer, Application Layer, and Data Layer for better performance, scalability, and maintainability.

Presentation Layer (Frontend / User Interface)

Provides an interactive and user-friendly interface.
Allows users to play, create, and customize melodies.
Developed using modern frontend technologies to ensure smooth navigation and attractive design.

Application Layer (Backend / Business Logic)

Acts as the bridge between the user interface and the database.
Handles music generation logic, recommendation algorithms, and melody customization.
Ensures secure communication between frontend and database.

Data Layer (Database Management)

Stores user information, preferences, and saved melodies.
Maintains data integrity and provides fast retrieval.
Supports scalability to handle large numbers of users and music files.

4.Setup Instructions

Prerequisites

Before setting up RhythmiCTune – Your Melody Companion, ensure the following software and tools are installed on your system:

Node.js – Runtime environment for executing JavaScript on the server side.

MongoDB – NoSQL database for storing user data and melodies.

Git – Version control system for managing project code.

React.js – Frontend library for building the user interface.

Express.js – Backend framework for handling server-side logic.

Mongoose – ODM library for interacting with MongoDB.

Visual Studio Code (VS Code) – Recommended code editor for development.

Installation Steps

1. Clone the Project Repository:
   git clone <repository-url>

2. Navigate to the Project Directory:
   cd rhythmiCTune

3. Install Backend Dependencies:
   npm install

4. Install Frontend Dependencies (if in separate folder):
   cd client
   npm install

5. Start the Backend Server:
   npm start

6. Start the Frontend Application:
   npm start

7. Access the Application:
   Open your browser and go to:
   http://localhost:3000

5.Folder Structure

The project RhythmiCTune – Your Melody Companion follows a structured layout to separate the frontend and backend for better organization and maintainability:

```
SB-Works/
├── client/           # React frontend
│   ├── components/    # Reusable UI components (buttons, forms, etc.)
│   └── pages/         # Application pages (Home, Create Melody, Profile, etc.)
├── server/           # Node.js backend
│   ├── routes/        # API routes (endpoints for frontend requests)
│   ├── models/        # Database models (User, Melody, etc.)
│   └── controllers/   # Business logic for handling requests
```

Explanation

client/: Contains all frontend code built using React.js. Components are reusable UI elements, and pages represent different views in the application.
server/: Contains all backend code built using Node.js and Express.js. Routes handle API requests, models define database structure using Mongoose, and controllers contain the logic for processing requests.

6. Running the Application

Once the setup is complete, follow these steps to run the RhythmiCTune – Your Melody Companion application:

Frontend (React)

1. Navigate to the frontend directory:
   cd client

2. Start the React application:
   npm start

Backend (Node .js / Express)

1. Navigate to the backend directory:
   cd server

2. Start the backend server:
   npm start

Accessing the Application

Open a web browser and go to:
http://localhost:3000

7. API Documentation

The backend of RhythmiCTune – Your Melody Companion provides several RESTful APIs to handle user interactions, project management, applications, and chat functionality.

User APIs

Register User
POST /api/user/register
Registers a new user with the system. Requires user details such as name, email, and password.

Login User
POST /api/user/login
Authenticates a user and provides an access token for secured routes.

Projects APIs

Create Project
POST /api/projects/create
Allows a user to create a new project with details like title, description, and team members.
Get Project by ID
GET /api/projects/:id
Retrieves project details for a given project ID

Applications API

Apply to Project
POST /api/apply
Enables users to apply for participation in a project.

Chats APIs

Send Message
POST /api/chat/send
Sends a message from one user to another in the chat system.
Get Chat with User
GET /api/chat/:userId
Retrieves the chat history between the logged-in user and another user by their user ID.

8. Authentication

RhythmiCTune – Your Melody Companion uses JWT (JSON Web Token) based authentication to ensure secure access for users.

Secure Login:

Users are authenticated via email and password. Upon successful login, a JWT token is generated and sent to the client. This token is required for accessing protected routes.

Protected Routes:

Backend routes that require user authentication are secured using middleware. The middleware verifies the JWT token before granting access, ensuring that only authorized users can perform sensitive operations like creating projects, applying, or sending messages.

9. User Interface

The RhythmiCTune – Your Melody Companion application provides a clean and intuitive user interface, designed to enhance user experience and simplify navigation. The main UI components include:

Landing Page

The first page users see when accessing the application.
Provides information about the app, its features, and options to register or log in.

Freelancer Dashboard

Personalized dashboard for users to view and manage their projects.
Allows users to create melodies, apply for projects, and track their activity.

Admin Panel

Administrative interface for managing users, projects, and overall application settings.
Enables monitoring of activities, approvals, and analytics.

Project Details Page

Displays detailed information about individual projects.
Shows project description, team members, application status, and allows users to apply or interact with the project.

10.Testing

Testing is a critical part of the development process for RhythmiCTune – Your Melody Companion to ensure reliability, performance, and user satisfaction.

Manual Testing:

Conducted during each project milestone to verify functionality, usability, and overall performance.

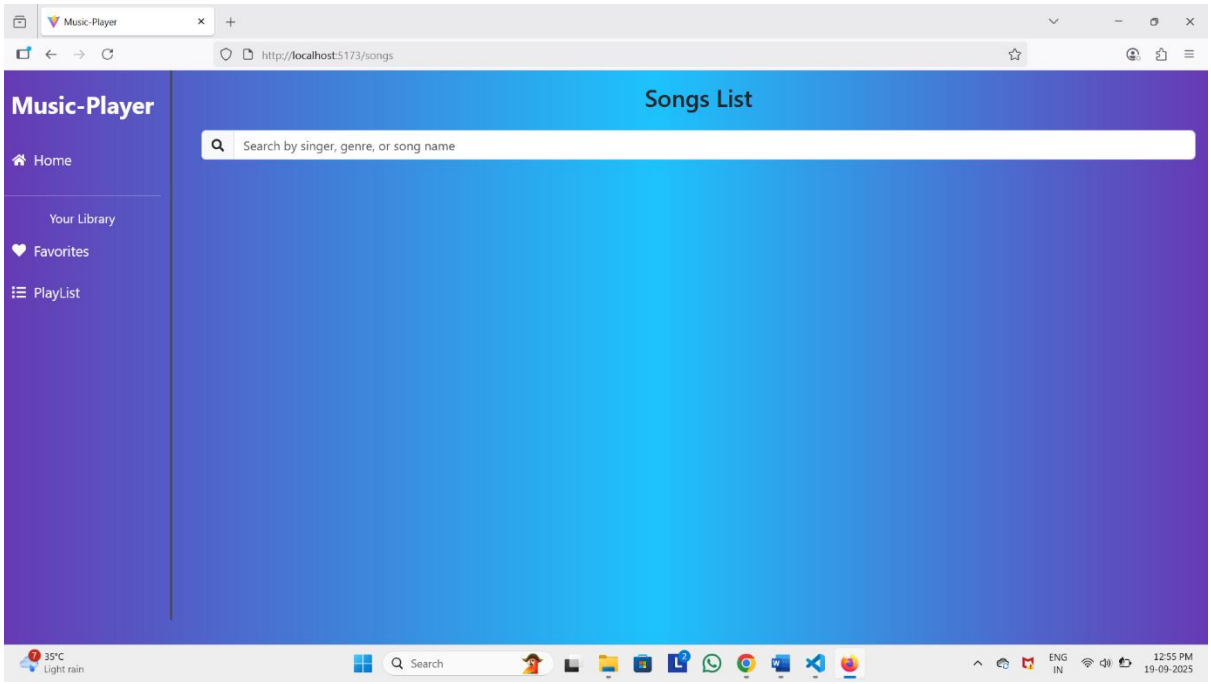
Ensures that all features, including melody creation, project management, and chat, work as intended.

Testing Tools:

Postman: Used for testing and validating backend API endpoints.

Chrome Dev Tools: Used for debugging, performance monitoring, and inspecting frontend behavior.

11.Screenshots or Demo



12.Known Issues

While RhythmiCTune – Your Melody Companion is functional and stable, the following issues have been identified and are planned for future improvement:

Performance Lag on Large Playlists:

The app may experience slight delays when handling very large melody collections.

Limited Offline Functionality:

Currently, most features require an active internet connection; offline support is limited.

Cross-Browser UI Inconsistencies:

Minor differences in layout and styling may appear on some browsers.

Notification System:

Real-time notifications for chats or project updates are basic and could be enhanced.

Error Handling:

Some backend errors may return generic messages instead of detailed feedback for users.

13.Future Enhancements

To further improve RhythmiCTune – Your Melody Companion, the following enhancements are planned for future development:

1. Offline Mode:

Allow users to access and play saved melodies without an internet connection.

2. Advanced Recommendation System:

Implement AI-based algorithms to provide smarter and more personalized melody suggestions.

3. Real-Time Notifications:

Add push notifications for chat messages, project updates, and important alerts.

4. Cross-Platform Support:

Extend support for mobile devices and tablets for a seamless experience across all platforms.

5. Enhanced User Interface:

Introduce more interactive features, themes, and customization options for users.

6. Collaboration Features:

Enable multiple users to collaborate on melody creation and project development in real-time.