**Project Documentation**

**Rhythmic tunes: your melodic companion**

# 1. Introduction

* **Project Title:** Rhythmic tunes: your melodic companion

# Team ID : NM2025TMID37410

* **Team Leader:** KEERTHANA B & [kamithakeerthana@gmail.com](mailto:kamithakeerthana@gmail.com)
* **Team Members:**
  + VINITHA R & [vini1708006@gmail.com](mailto:vini1708006@gmail.com)
  + SHARMIKA D & [sharmisharmika23@gmail.com](mailto:sharmisharmika23@gmail.com)
  + NITHIYA SRI NK & [sreenithya752@gmail.com](mailto:sreenithya752@gmail.com)

# 2. Project Overview :

# Purpose: Rhythmic Tones is designed to provide clear, engaging, and expressive voiceovers for projects, enhancing communication with natural flow and impact.

# Features:

* + Natural and expressive voice modulation
  + Clear pronunciation with smooth flow
  + Adjustable pitch, pace, and tone
  + Consistent rhythm for engaging delivery
  + Suitable for narration, presentations, and creative content

# 3. Architecture

* **Frontend:** React.js with Bootstrap and Material UI for user interface
* **Backend:** Node.js and Express.js managing server logic and API handling
* **Database:** MongoDB stores user data, project information, applications, and chat messages

# 4. Setup Instructions

* **Prerequisites:**
  + Node.js
  + MongoDB
  + Git
  + React.js
  + Express.js **–** Mongoose **–** Visual Studio Code

**Installation Steps:**

#Clone the project repository using git clone <repository-url>.

#Navigate to the project folder: cd rhythmic-tunes.

#Install dependencies: npm install (or pip install -r requirements.txt if it's a Python project).

#Set up environment variables if required (e.g., .env file).

#Start the application using npm start or the relevant run command.

Top of Form

Bottom of Form

# 5. Folder Structure :

Rhythmic-Tunes/

|-- src/ # React frontend code

| |--components/

| |-- pages/

|-- db/ #JSON Server data

| |-- db.json

|-- package .json #Project dependencies

# 6. Running the Application

* **Frontend:**

cd client npm start

• **Backend:**

cd server npm start

* **Access:** Visit http://localhost:3000

# 7. API Documentation

* **User:**
  + /api/user/register
  + /api/user/login
* **Projects:**
  + /api/projects/create
  + /api/projects/:id • **Applications:** /api/apply
* **Chats:**
  + /api/chat/send **–** /api/chat/:userId

# 8. Authentication

* Authentication
* JWT-based authentication ensures secure login for users and trainers
* Role-based access (Admin, Trainer, User) is implemented for better security.

# 9. User Interface

* Landing Page: This is the first page visitors see, likely providing an overview of the application's features.
* User Dashboard: A personalized page for standard users to manage their profiles and interact with the application.
* Trainer Dashboard: A specialized dashboard for trainers with features tailored to their role, such as managing clients and creating workout plans.

# 10. Testing

* Manual Testing: The application is tested manually during each development phase to ensure it works correctly.
* Postman: This tool is used specifically for API testing to check if the server's endpoints are functioning as expected.
* Chrome DevTools: This browser-based tool is used for debugging and testing the frontend of the application.
* **Screenshots or Demo**

**Link:** [https://drive.google.com/file/d/1lbwdGrzmJYJ\_ADOg- 7X073a3Jd7GJ9DN/view?usp=sharing](https://drive.google.com/file/d/1lbwdGrzmJYJ_ADOg-%20%20%207X073a3Jd7GJ9DN/view?usp=sharing)

**12.Known Issues :**

* Real-time chat delays: There may be occasional delays in the real-time chat updates.
* API failures under load: Some API requests might fail when the system is under heavy user load.
* Limited client-side error handling: There is limited error handling on the client-side, which can cause issues for the user.

# 13. Future Enhancements :

* Wearable Device Integration: The plan is to integrate with wearable fitness devices like smartwatches.
* Personalized Recommendations: The app will be enhanced with AI to provide personalized workout recommendations.
* Push Notifications: Users will receive push notifications for workout reminders and messages from trainers.
* Mobile App Development: A native mobile app will be developed for both iOS and Android platforms.