**Rhythmic tune** – **Your** **Melody** **Companion**

Team Member 1: J.Safiya Asrin

Team Member 2: S.Sabira Fathima

Team Member 3: B.Asma

Team Member 4: N.Sumaiya Fathima

Team Member 5:A.Sabitha

Project Title: Rhythmic – Your Melody Companion

2. 📝 Overview

🍽️ Purpose

Rhythm8c is an AI-powered platform built using Node.js + React, designed to help users generate, analyze, recommend, and refine music and melodies. Whether you're a casual music creator or a pro, Rhythm8c becomes your AI-enhanced melody assistant.

🎯 Key Features

Feature Description

🎼 Melody Generator AI composes original melodies based on mood, tempo, or theme

🧠 Mood Detection Analyzes uploaded music files to detect emotion or genre

🎧 Smart Recommender Suggests similar songs or scales using AI & audio fingerprinting

🛠️ Melody Refinement Offers suggestions to improve melody based on music theory and AI insights

🗣️ Chat-based Assistant Interact with an AI bot to ask music theory questions or generate ideas

📊 Audio Analyzer Visualize waveforms, tempo, pitch, and more

---

3. 🏗️ System Architecture (Node.js + AI Stack)

🔹 Frontend

Tech: React + TailwindCSS or Material UI

Features:

Upload & play music

Interactive charts (e.g., waveform, pitch contour)

Melody editor (keyboard grid or piano roll)

Chat interface with AI Assistant

🔹 Backend

Tech: Node.js + Express.js

Features:

RESTful APIs

File upload with Multer

Socket.io for real-time feedback (e.g., waveform rendering)

API integration with AI models (OpenAI, Hugging Face, etc.)

🔹 AI Integration

Task Tool / Model Used

Melody Generation Music Transformer, Magenta.js (Python svc)

Mood Detection CNN-based model (via Python svc)

Recommendations Spotify API + Vector Similarity Search

Melody Refinement Rules-based + GPT-based suggestions

Assistant Chat OpenAI ChatGPT API (Node.js SDK)

Audio Analysis Librosa (Python), converted via microservice

---

4. ⚙️ Setup Instructions

🛠️ Prerequisites

Node.js v18+

Python 3.8+ (for AI microservices)

MongoDB or PostgreSQL (optional for session/history)

FFmpeg (for audio file processing)

OpenAI API key

Docker (optional)

🚀 Installation Steps

# Clone repo

git clone https://github.com/your-org/rhythm8c.git

cd rhythm8c

# Backend setup

npm install

cp .env.example .env

# Add API keys to .env

npm run dev

# Frontend setup

cd client

npm install

npm start

---

5. 📁 Folder Structure

rhythm8c/

├── backend/

│ ├── routes/

│ │ ├── melody.js

│ │ ├── mood.js

│ │ ├── recommend.js

│ │ └── chat.js

│ ├── services/

│ │ └── aiModels.js

│ ├── controllers/

│ ├── utils/

│ └── server.js

├── client/

│ ├── src/

│ │ ├── pages/

│ │ │ ├── MelodyGen.jsx

│ │ │ ├── MoodDetect.jsx

│ │ │ ├── Recommender.jsx

│ │ │ ├── Analyzer.jsx

│ │ │ └── Assistant.jsx

│ └── App.jsx

├── python-services/

│ ├── melody\_generator.py

│ ├── mood\_detector.py

│ └── analyzer.py

├── .env

└── package.json

---

6. 🔌 Running the Application

Step-by-Step

1. Start Backend

npm run dev

2. Start Frontend

cd client

npm start

3. Access App

Open http://localhost:3000

---

7. 🔍 API Endpoints (Swagger UI)

Endpoint Method Description

/api/melody/generate POST Generates a melody based on user input

/api/mood/detect POST Detects emotion from uploaded audio

/api/recommend POST Recommends similar tracks or scales

/api/analyze POST Analyzes tempo, pitch, etc.

/api/chat POST AI assistant for music questions

Swagger: http://localhost:8000/docs

---

8. 🔐 Authentication

JWT-based login (optional)

OAuth2 with Spotify or Google

API key restriction (for AI endpoints)

Rate limiting on melody generation

---

9. 🧭 UI Highlights

🎹 Piano roll for melody edi…