EduTutor-AI Personalized-Learning-with-GenerativeAI-and-LMS-Integratio

1. Introduction

• **Project Title**: EduTutor AI
• **Team Members**: K.DEEPIKA
K.MOULIKA
I.MOUNIKA
K.CHANDANA

2. Project Overview

• **Purpose:** This project aims to develop EduTutor AI, a personalized learning platform powered by AI that provides dynamic content, quizzes, and assessments for students.
• **Features:** AI-based Concept Clarification
Mood-to-Method Learning
Voice-based Concept Generator
Real-time Quiz Generation
Student Progress Tracker
Feedback and Recommendation System

3. Architecture

• **Frontend:** Built using React for a responsive, interactive UI.
• **Backend:** Developed using Node.js and Express.js to handle API requests.
• **Database:** Uses MongoDB to store user data, quiz records, and progress details.

4. Setup Instructions

• **Prerequisites:** Node.js, MongoDB.
• **Installation:** 1. Clone the repository.
2. Run npm install in both client and server directories.
3. Set environment variables as per .env.example file

5. Folder Structure

• **Client:** Contains React frontend code.
• **Server:** Contains Node.js backend code and API logic.

6. Running the Application

• **Frontend:** Run "npm start" inside the client directory.
• **Backend:** Run "npm start" inside the server directory.

7. API Documentation

• **Example Endpoints:** GET /api/concepts/:topic
POST /api/quiz
GET /api/progress/:userId

8. Authentication

• Uses JWT tokens for secure authentication.
• User sessions are maintained via tokens.

9. User Interface

• The UI is designed to be user-friendly with interactive quizzes, feedback popups, and progress tracking dashboards.

10. Testing

• Jest and Postman are used for unit and API testing.

11. Screenshots or Demo

• [Screenshots or demo link to be added]

12. Known Issues

• No critical bugs at present. Minor UI optimizations pending.

13. Future Enhancements

• Google Classroom integration.
• Mobile app version.
• Advanced analytics dashboard.
• AI-powered voice interaction.