EduTutor AI: Personalized Learning with Generative AI and LMS Integration

Project Documentation

1. Introduction

• **Project Title:** EduTutor AI: Personalized Learning with Generative AI and LMS Integration

• Team Members: Shifa Sabreen A

• Team Members: Thilagavathy M

• Team Members: Yuva Priya S

• Team Members: Susmitha N

2. Project Overview

Purpose: The purpose of EduTutor AI is to provide a personalized, AI-powered learning assistant integrated with Learning Management Systems (LMS). By leveraging generative AI, it adapts to students' individual learning styles, offers real-time explanations, generates practice questions, and provides feedback. For educators, it simplifies content creation, progress tracking, and engagement with learners.

Features:

- Conversational Tutor - AI-powered Q&A in natural language.
- Content Generation - Generates quizzes, summaries, and assignments tailored to the syllabus.
- Personalized Learning Path - Suggests next topics, difficulty levels, and resources based on learner performance.
- LMS Integration - Syncs with LMS platforms (Moodle, Google Classroom, etc.) for progress tracking.
- Analytics Dashboard - Provides performance analytics, engagement metrics, and skill-gap reports.
- Feedback & Assessment - Grades quizzes and assignments with personalized feedback.

3. Architecture

• Frontend (React / Streamlit): Provides interactive dashboards for students and teachers.

• Backend (FastAPI / Django): Handles API endpoints for AI interactions and LMS integration.

• LLM Integration: Generative AI powers Q&A, content creation, and personalization.

• Database (PostgreSQL/MySQL): Stores student profiles, progress data, and generated content.

• LMS API Integration: Connects with LMS systems for seamless syncing.

4. Setup Instructions

Prerequisites: Python 3.9+, Node.js (if using React), virtual environment tools, API keys, internet access.

Installation Process:

1. Clone the repository.
2. Install backend dependencies from requirements.txt.
3. Install frontend dependencies (npm install).
4. Configure .env with API keys and LMS credentials.
5. Run backend server (uvicorn main:app).
6. Launch frontend (npm start or streamlit run).
7. Access web interface and test modules.

5. Folder Structure

```
app/ – Backend logic
 └── api/ – Routes for chat, quiz, progress
ui/ – Frontend components
models/ – AI model integration
database/ – Schema and data storage
dashboard.py – Analytics dashboard
ai_tutor.py – Chat and content generation
quiz_generator.py – Quiz/assignment generation
lms_connector.py – LMS API integration
```

6. Running the Application

• Start backend API server
• Launch frontend dashboard
• Login as student/teacher

• Ask queries, generate quizzes, track performance
• Sync results with LMS

7. API Documentation

POST /chat/ask – Ask AI tutor a question

POST /quiz/generate – Generate quiz questions

GET /progress/{student_id} – Get student’s progress data

POST /lms/sync – Sync results with LMS

8. Authentication

• JWT token-based authentication
• Role-based access (Student, Teacher, Admin)
• Secure API keys for external integrations

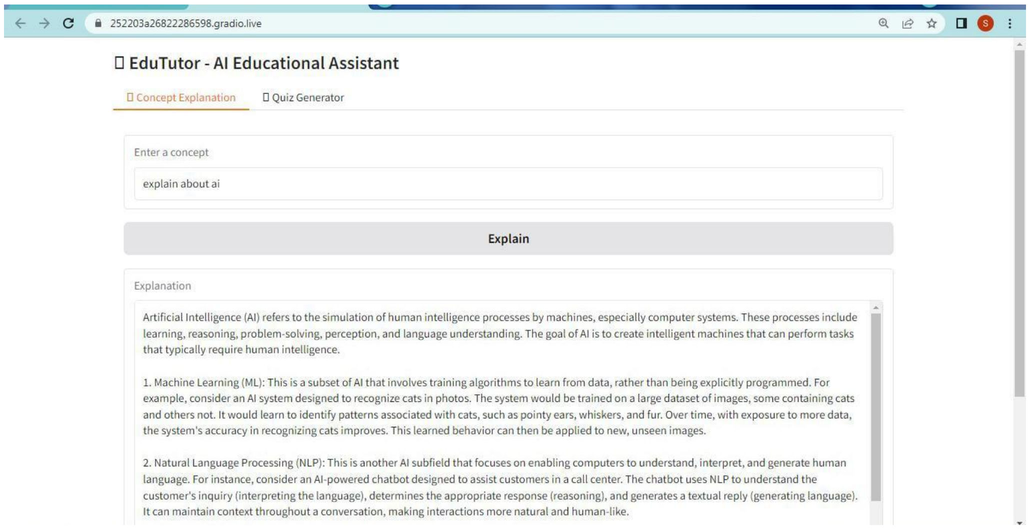
9. User Interface

• Student Dashboard: Chat, quizzes, recommendations
• Teacher Dashboard: Analytics, quiz creation, insights
• LMS Sync: Automatic gradebook updates

10. Testing

• Unit Testing – AI functions and APIs
• API Testing – Postman/Swagger UI
• User Testing – Student and teacher interaction trials
• Edge Cases – Invalid queries, missing LMS data

11. Screenshots



explain about ai

Artificial Intelligence (AI) refers to the simulation of human intelligence processes by machines, especially computer systems. These processes include learning, reasoning, problem-solving, perception, and language understanding. The goal of AI is to create intelligent machines that can perform tasks that typically require human intelligence.

1. Machine Learning (ML): This is a subset of AI that involves training algorithms to learn from data, rather than being explicitly programmed. For example, consider an AI system designed to recognize cats in photos. The system would be trained on a large dataset of images, some containing cats and others not. It would learn to identify patterns associated with cats, such as pointy ears, whiskers, and fur. Over time, with exposure to more data, the system's accuracy in recognizing cats improves. This learned behavior can then be applied to new, unseen images.

2. Natural Language Processing (NLP): This is another AI subfield that focuses on enabling computers to understand, interpret, and generate human language. For instance, consider an AI-powered chatbot designed to assist customers in a call center. The chatbot uses NLP to understand the customer's inquiry (interpreting the language), determines the appropriate response (reasoning), and generates a textual reply (generating language). It can maintain context throughout a conversation, making interactions more natural and human-like.

12. Known Issues

- Dependency on internet for AI queries
- LMS compatibility limited to supported platforms

13. Future Enhancements

- Support for multiple languages
- Offline AI assistant
- AR/VR integration
- Gamification features