**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 member: THRISHA S
* Team member: MANIMEGALAI S
* Team member: SANTHIYA S
* Team member: GIRIJA S

**2.Project overview**

* **Purpose:**

The Edututor AI project is designed as an advanced educational assistant that integrates generative AI capabilities with traditional Learning Management Systems (LMS) to create a more effective and personalized learning environment. In today’s education system, one of the biggest challenges is addressingthe different learning styles, speeds, and knowledge levels of individual students. Traditional classrooms and even online courses often follow a “one-size-fits-all” approach, where the same material is delivered to everyone, regardless of whether a student is struggling or excelling. Edututor AI aims to solve this issue by acting as a virtual AI tutorthat is always available to explain concepts, generate study material, and create personalized quizzes tailored to each learner.

The platform uses Large Language Models (LLMs) to generate explanations, examples, practice questions, and feedback. For example, a student who finds physics difficult can type in a topic such as “Newton’s Laws of Motiona clear” and immediately receiverexplanation, complete with examples and illustrations. Similarly, the AI can generate quizzes with multiple-choice, true/false, or short-answer questions, along with a separate answer key for self-assessment. This dynamicapproach not only keeps learners engagedbut also allows them to strengthen their weak areas with instant guidance.

Key Features of Edututor AI:

1. Personalized Learning

* Adapts to each student’s learning pace, style, and performance.
* Provides customized explanations and examples.

2. AI-Powered Concept Explanation

* Explains any concept in simple, detailed language with examples.
* Can adjust explanations for beginner, intermediate, or advanced learners.

3. Automatic Quiz Generation

* Creates quizzes instantly on any topic.
* Supports multiple formats: multiple choice, true/false, short answer.
* Provides an answer key for self-checking.

4. Instant Feedback

* Gives real-time responses to student answers.
* Explains why an answer is right or wrong.

5. LMS Integration

* Works with platforms like Moodle, Blackboard, or Google Classroom.
* Syncs assignments, grades, and learning progress automatically.

6. Learning Progress Tracking

* Keeps records of student performance over time.
* Identifies strengths and weaknesses for both students and teachers.

7. Teacher Support Tools

* Helps teachers by generating lesson summaries, quizzes, and study materials.
* Saves time on repetitive work so teachers can focus on mentoring.

8. Study Material Generation

* Creates summaries, flashcards, and key points from lessons.
* Suggests personalized study paths for improvement.

9. Interactive Web Interface

* Simple interface built with Gradio.
* Students can ask questions and get instant AI answers.
* Easy for teachers to create and manage quizzes.

10. Scalability & Accessibility

* Can be used by individuals, schools, or large institutions.
* Works as a 24/7 virtual tutor, accessible anytime and anywhere.

Architecture

1. User Interface (Frontend) – A web or mobile app where students ask questions, request explanations, and take quizzes, while teachers track progress and manage content.

2. Application Layer (Backend Services) – Acts as the middle layer that processes user requests, calls AI functions (explanation, quiz generation), and manages communication between all components.

3. AI Engine (Generative AI Models) – Uses large language models (like GPT or Granite) to generate detailed explanations, quizzes, feedback, and study materials tailored to each learner.

4. Database Layer – Stores student data, quiz results, progress history, and performance analytics for long-term tracking and personalization.

5. LMS Integration Layer – Connects Edututor AI with existing Learning Management Systems (like Moodle or Google Classroom) through APIs, syncing assignments, grades, and reports automatically.

Setup instructions

**Software Requirements:**

* Programming language: Python / Node.js
* Database: MySQL or PostgreSQL
* Frameworks: Django/Flask (for backend) and React/Angular (for frontend)
* AI Integration: Generative AI APIs (e.g., OpenAI, Hugging Face)
* LMS Integration: APIs from platforms like Moodle or Canvas

**Installation Steps:**

* Install all required dependencies and libraries.
* Configure the database and apply migrations.
* Set up environment variables for API keys, database credentials, and security keys.

Folder structure

main.py → Runs the app (gr. Blocks with Concept Explanation + Quiz Generator).

models/ → All code for loading AI models and tokenizers.

services/ → Business logic (AI response, quiz, explanations).

interface/ → Gradio UI tabs and layouts.

data/ → Prompts, sample quizzes, test data.

tests/ → Unit tests for each service.

config/ → Store API keys, LMS integration settings.

Running the application

Local Environment:

* Start the backend server.
* Launch the frontend application.
* Access the application through a web browser using localhost.

Cloud Deployment:

* Deploy on cloud platforms like AWS, Azure, or Heroku.
* Configure domain and SSL for secure access.
* Use containerization tools like Docker for portability.

User Access:

* The application can be accessed through a web interface where students and teachers log in with credentials.

API Documentation

The project provides a set of APIs for communication between the frontend, backend, AI model, and LMS:

User API: Handles registration, login, and profile management.

Recommendation API: Provides personalized course recommendations based on student data and AI analysis.

LMS API Integration: Synchronizes grades, assignments, and course progress with the LMS.

Administration API: Allows administrators and teachers to manage content, monitor student performance, and upload learning materials.

Each API follows standardized request and response structures, including success and error messages.

Authentication

The system uses secure authentication methods to ensure safe access:

Login Process: Users (students, teachers, admins) log in using credentials.

Authorization: Role-based access control ensures that each user only accesses features assigned to their role.

Security Measures:

Passwords are stored in encrypted form.

Tokens are issued for secure communication between client and server.

Automatic session expiry and refresh mechanisms are implemented.

User Interface

The project includes a clean and user-friendly interface designed for different roles:

Login/Signup Screen: Allows users to securely enter the system.

Student Dashboard: Displays personalized learning recommendations, course progress, and AI-generated practice quizzes.

Teacher dashboard: Enables teachers to upload materials, track student performance, and view AI insights.

Admin Panel: Manages the entire system, including user management, LMS connectivity, and monitoring.

Testing

Testing ensures the quality and reliability of the application:

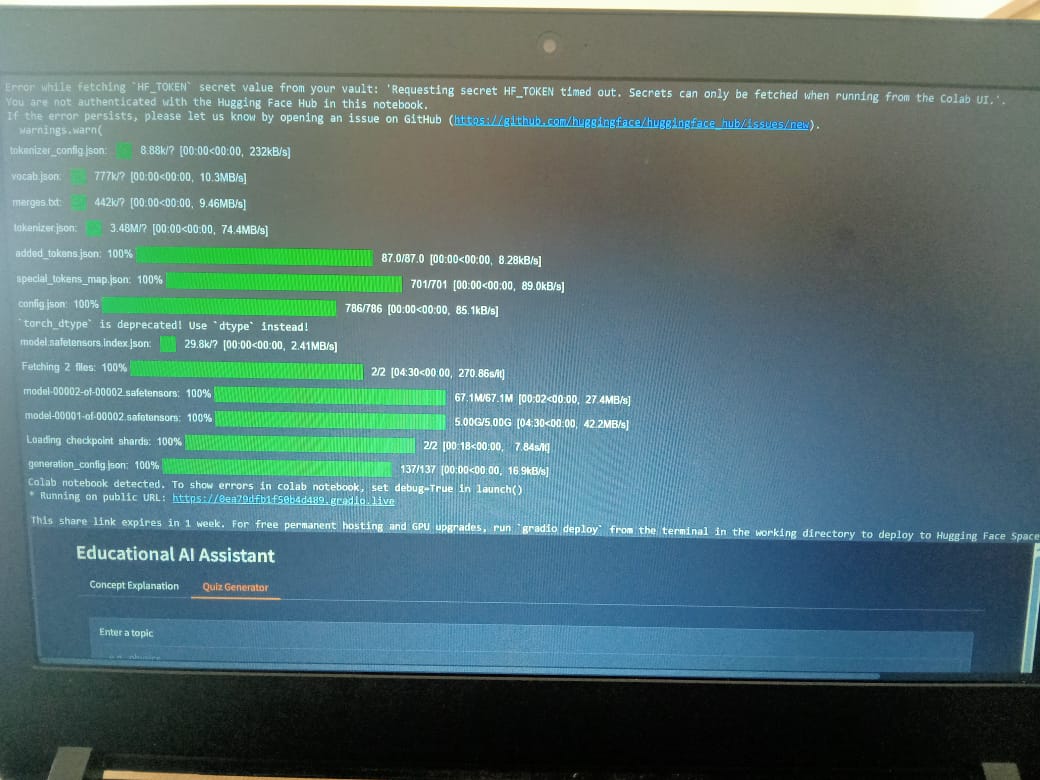
Unit Testing: Verifies that individual components such as recommendation generation, login, and LMS sync work correctly.

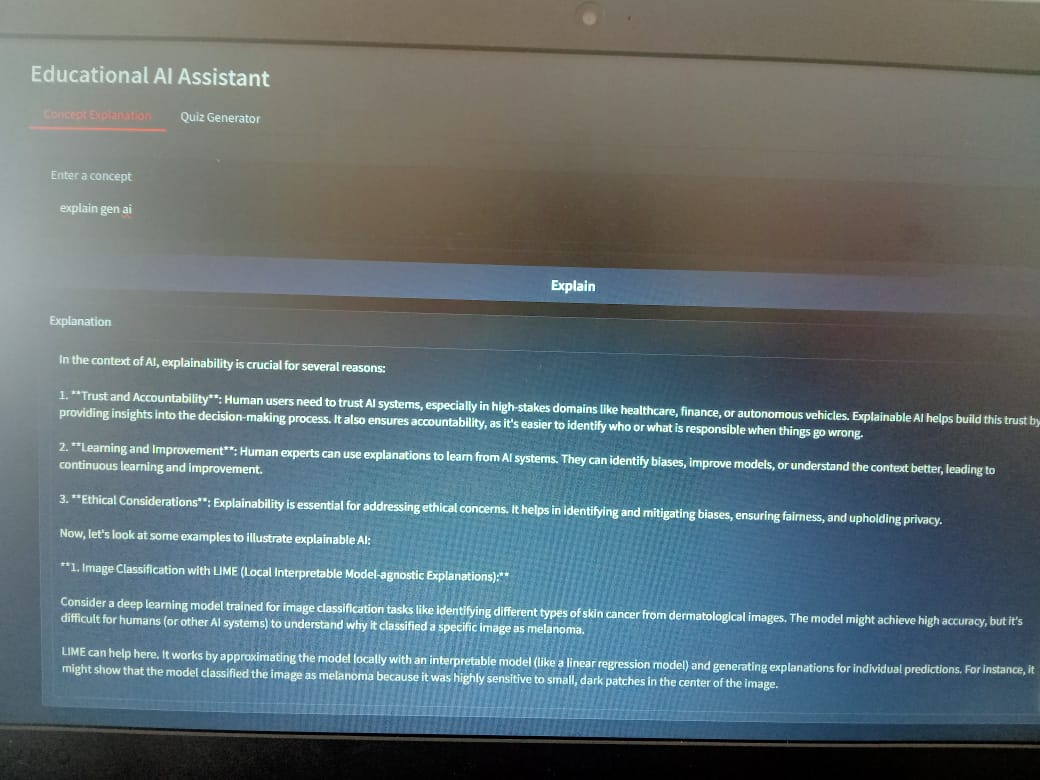
Integration Testing: Ensures smooth interaction between the AI model, LMS, and application backend.

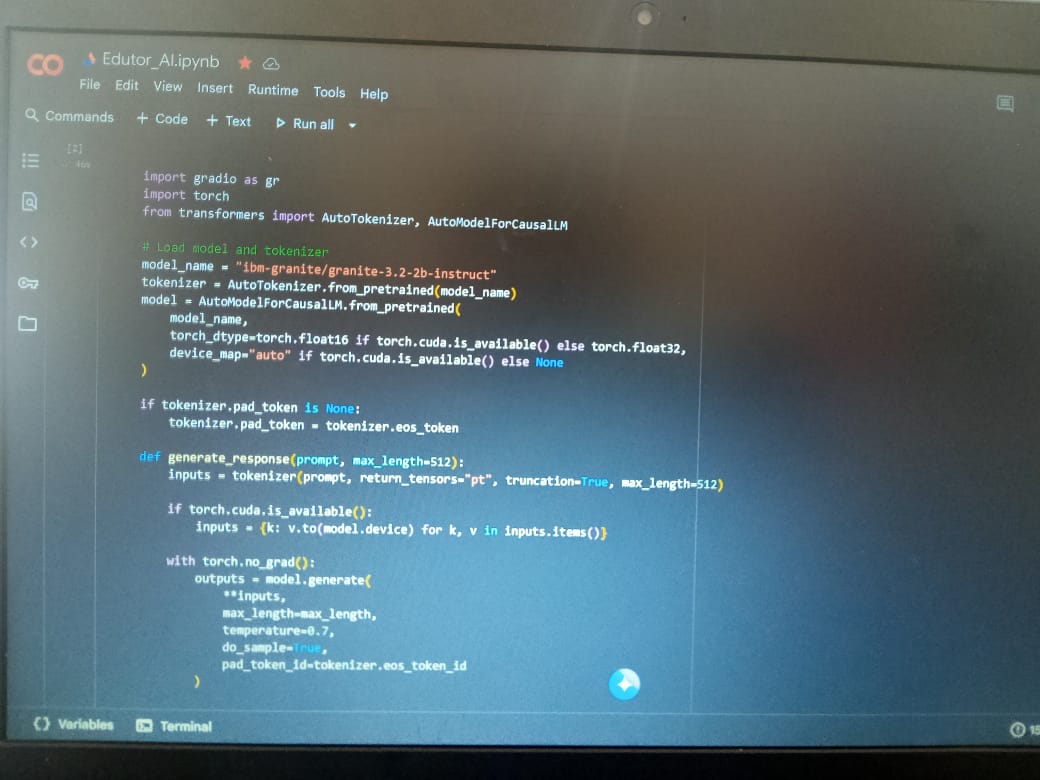
User Interface Testing: Confirms that all buttons, forms, and dashboards function properly for different user roles.

Performance Testing: Tests the application under heavy load to ensure it can handle multiple users at the same time.

Security Testing: Identifies vulnerabilities and ensures data protection.

**Output** **:**



**Coding:**

