

# Edu Tutor AI – Personalized Learning Assistant

Generative AI with IBM

## Project Documentation

### 1. Introduction

- **Project Title:** Edu Tutor AI – Personalized Learning Assistant
  - **Team Members:**
    - ◆ SATHIYA G
    - ◆ MOHANAPRIYA K
    - ◆ NIKKITHA BERNA F
    - ◆ PRABAVATHI R
    - ◆ SHERINA M
- 

### 2. Project Overview

- **Purpose:**

Edu Tutor AI is designed to act as an intelligent assistant for students, teachers, and institutions. By leveraging AI, adaptive learning, and real-time analytics, it personalizes study paths, provides instant feedback, summarizes educational materials, and supports collaborative learning. The system empowers students to learn at their own pace while giving teachers insights into progress, weak areas, and performance trends.
- **Features:**
  - **Conversational Interface**

Natural language interaction for asking questions, explanations, and tutoring support.

- **Content Summarization**  
Simplifies textbooks, notes, and research articles into concise, student-friendly explanations.
- **Adaptive Learning Path**  
Tracks student performance and dynamically recommends lessons, quizzes, and resources.
- **Quiz & Test Generator**  
Creates personalized quizzes based on difficulty level, previous mistakes, and subject focus.
- **Student Feedback Loop**  
Collects responses and self-assessments to tailor recommendations.
- **Progress & KPI Tracking**  
Monitors grades, time spent, mastery level, and suggests study improvements.
- **Anomaly Detection**  
Identifies sudden drops in performance, inconsistent learning patterns, or disengagement.
- **Multimodal Input Support**  
Accepts text, PDFs, lecture slides, and videos for knowledge extraction.
- **User-Friendly Dashboard (Streamlit/Gradio)**  
Intuitive UI for students and teachers with reports, analytics, and study materials.

---

### 3. Architecture

- **Frontend (Streamlit/Gradio):** Interactive dashboards, chat interface, file uploads, progress visualization.

- **Backend (FastAPI):** Handles content processing, chat responses, test generation, and learning path updates.
  - **LLM Integration (IBM Watsonx/OpenAI/Other):** Used for question answering, summarization, and personalized tutoring.
  - **Vector Search (Pinecone/Faiss):** Stores embedded textbooks, notes, and lecture content for semantic search.
  - **ML Modules:** Adaptive learning analytics, difficulty prediction, and anomaly detection using scikit-learn.
- 

## 4. Setup Instructions

(Similar steps: Python  $\geq 3.9$ , dependencies, API keys, run backend & frontend, upload study material, interact with tutor.)

---

## 5. Folder Structure

- app/ – FastAPI backend (chat, content processing, student analytics)
  - ui/ – Streamlit/Gradio frontend (student dashboard, teacher reports)
  - quiz\_generator.py – Generates quizzes/tests dynamically
  - progress\_tracker.py – Monitors learning KPIs and engagement
  - content\_summarizer.py – Converts notes/books into summaries
  - report\_generator.py – Creates performance and learning reports
- 

## 6. Running the Application

Launch backend, run dashboard, upload study material, interact with tutor, view quizzes, summaries, and progress reports.

---

## 7. API Documentation

- POST /chat/ask – Student asks a question
  - POST /upload-material – Upload notes, textbooks, lectures
  - GET /get-summary – Returns simplified explanations
  - GET /generate-quiz – Creates quizzes/tests
  - POST /submit-feedback – Student feedback on learning
- 

## 8. Authentication

Token-based login (student, teacher, admin), with role-based access.

---

## 9. User Interface

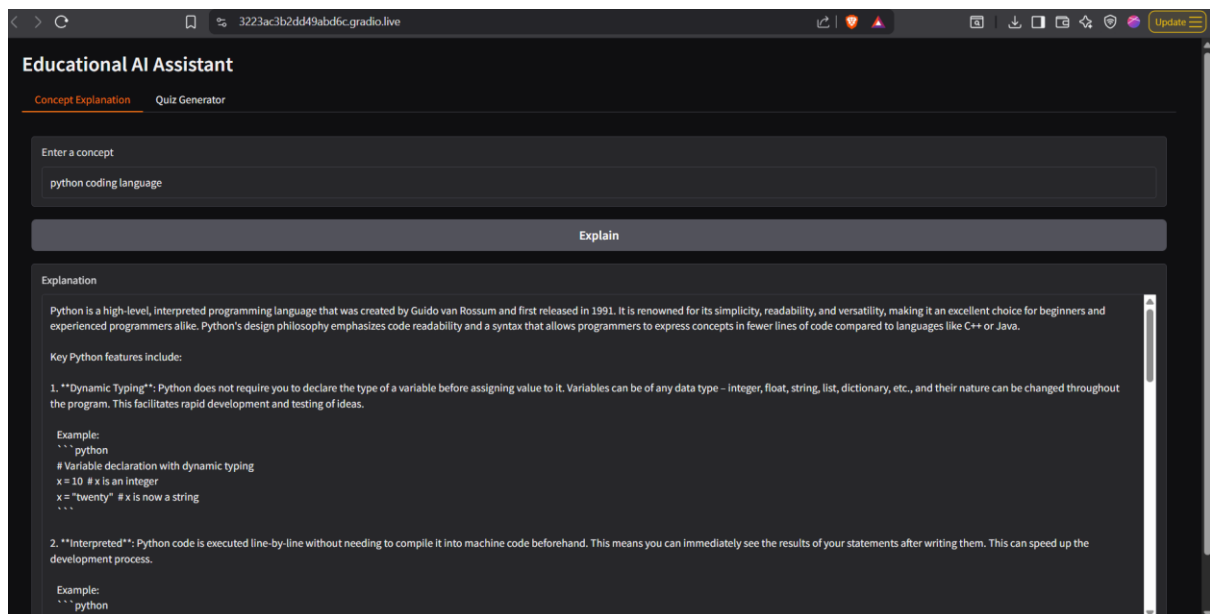
Simple, accessible dashboards for:

- **Students:** Chat tutor, summaries, quizzes, progress tracker.
  - **Teachers:** Student analytics, weak-area insights, performance reports.
- 

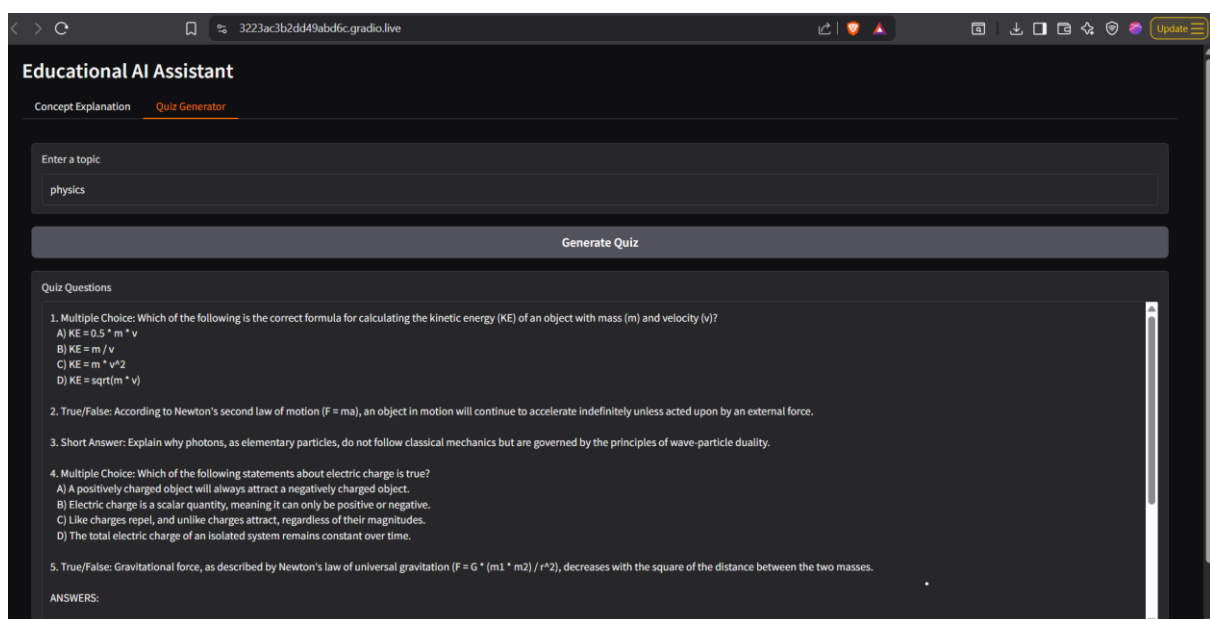
## 10. Testing

- Unit Testing (quiz generation, summarization)
- API Testing (Postman/Swagger)
- Manual Testing (student interactions, file uploads)
- Edge Case Handling (low-quality inputs, large files).

## 11. Screenshots



We can give a topic and it will explain the concept of the topic in short and clear explanation.



We can generate quiz by a topic given from the user and it will give the answers too with the quiz.

## 12. Known Issues

1. **AI Misinterpretation of Queries** – Sometimes the assistant may provide answers slightly off-topic or too generic.
2. **Quiz Question Quality** – AI-generated quizzes may occasionally include repetitive or ambiguous questions.
3. **Limited Subject Coverage** – Current model may not fully support highly specialized subjects or advanced-level topics.

## 14. Future Enhancements

1. **Multilingual Support** – Expanding to regional and global languages to support diverse learners.
2. **Integration with LMS** – Connecting with platforms like Google Classroom, Moodle, or MS Teams for seamless usage.
3. **Gamification Features** – Adding badges, leaderboards, and rewards to increase student engagement.
4. **Voice-Based Learning** – Enabling speech-to-text and text-to-speech for accessibility and hands-free interaction.