EDUTUTOR AI

Project Documentation

1.Introduction

- Project Title: EduTutor AI
- Team Leader : RAGAVAN D
- Team member : MULLAI NATHAN S
- Team member : PREM K
- Team member : PRINSON JOSE J
- Team member : SAKTHI R

2. Project Overview

• Purpose:
The purpose of EduTutor AI is to transform the way students learn and teachers evaluate by combining the power of Generative AI and interactive learning platforms. Unlike traditional learning tools, EduTutor AI can explain any concept (like ChatGPT, Gemini-style responses) in simple and personalized ways, making knowledge accessible to every learner.

For students, it provides instant explanations, interactive guidance, and quizzes that adapt to their learning pace. For educators, it acts as a teaching partner—helping them create practice material, assess student performance, and gain insights into classroom progress.

Ultimately, EduTutor AI bridges education, AI technology, and student engagement to build a smarter, more personalized learning experience that is efficient, inclusive, and future-ready.

• Features:

1. Conversational Learning Assistant

Key Point: AI-powered Q&A system

Functionality: Students can ask questions on any subject/topic, and EduTutor AI explains concepts in simple, clear language—similar to ChatGPT or Gemini.

2. Quiz Generator

Key Point: Automated question creation

Functionality: Based on any concept, EduTutor AI generates 5 MCQ questions along with the correct answers, helping students test themselves instantly.

3. Concept Simplification

Key Point: Easy-to-understand summaries

Functionality: Converts complex study material into simplified explanations for better learning.

4. Personalized Study Tips

Key Point: AI-driven guidance

Functionality: Provides study strategies and resource suggestions tailored to a student's performance and learning style.

5. Performance Tracking

Key Point: Smart evaluation

Functionality: Tracks quiz results and learning progress, giving insights to both students and educators.

6. Document & Content Analysis

Key Point: Multi-format learning support

Functionality: Accepts text, PDFs, and notes for generating quizzes, summaries, and explanations.

7. Interactive UI (Streamlit/Gradio)

Key Point: User-friendly platform

Functionality: Provides a clean, dashboard-style interface for both students and teachers to interact with EduTutor AI.

3. Architecture

Frontend (Gradio):
The frontend is developed using Gradio, which provides an interactive web-based user interface. It supports multiple functionalities including concept explanation, quiz generation, file uploads, and chat-style interactions. Gradio's components (textboxes, buttons, file uploaders) make it easy for both students and educators to interact with EduTutor AI in a simple and user-friendly way.

Backend (Hugging Face + FastAPI Layer):
The backend runs primarily on Hugging Face integrations, where API endpoints are used to process student queries, generate quizzes, and analyze content. The IBM Granite 3.2.2-b Instruct model is leveraged for natural language understanding and generation tasks. The backend can be further extended using FastAPI for modular routing, scalability, and API management if required.

LLM Integration (IBM Granite 3.2.2-b Instruct):
The core of EduTutor AI is powered by Granite LLM from IBM , accessed via Hugging Face. The model is used for concept explanation, summarization, and quiz generation. Prompts are carefully engineered to provide accurate, student-friendly responses.

Development Environment (Google Colab):
The project is developed and tested on Google Colab, which provides GPU support and an easy-to-use environment for rapid prototyping. All dependencies are managed via pip installs, ensuring smooth execution.

Key Dependencies:

Transformers → For handling pre-trained models and text generation.

Torch → Core deep learning framework for model inference.

Gradio → For building the frontend web interface.

4. Setup Instructions

Prerequisites:

Python 3.9 or later

pip package manager

Google account (for Colab)

IBM API key

Steps:

1. Open Google Colab and create a new Python file.

2. Install required libraries:

pip install transformers torch gradio

3. Configure your IBM Watsonx API key inside .env or directly in Colab.

4. Import libraries, load the Granite 3.2.2-b Instruct model, and connect with Gradio.

5. Run the notebook cell to launch the Gradio interface.

6. Interact with EduTutor AI by asking concepts or generating quizzes.

5. Folder Structure

EduTutor AI is a single-file project built and executed in Google Colab. All logic for frontend (Gradio UI) and backend (model + processing) is combined into one script for simplicity.

EduTutor.py

Gradio Frontend

Chat interface for concept explanation

Quiz generator (5 MCQs + answers)

Backend Logic

IBM Granite 3.2.2-b Instruct model integration

Functions for text generation and quiz creation

Configuration

API key setup (IBM)

Library imports (transformers, torch, gradio)

Utilities

Handles input prompts

Returns AI-generated responses

Manages interactive session

6. Running the Application

To start the project:
➢ Open Google Colab and create a new Python file (EduTutor.py).
➢ Install required libraries (transformers, torch, gradio).
➢ Add your IBM Watsonx API key in the script or via environment variables.
➢ Run the notebook cells to start the application.
➢ A Gradio interface link will appear – click to open the EduTutor AI app.
➢ Interact with the assistant by:

Asking any concept (example: “Explain OOP in Python”).

Generating quizzes (example: “Create 5 MCQs on Data Structures”).

Frontend (Gradio):
The frontend is built using Gradio, providing a simple web-based UI. It includes:

Chat interface for concept explanation.

Quiz generator with questions & answers.

File input support for text/PDF-based content.

Backend (IBM Granite on Hugging Face):
The backend logic is integrated directly inside the same script. It handles:

Concept explanation using IBM Granite 3.2.2-b Instruct model.

Quiz generation with MCQs and answers.

AI-driven text processing and response generation.

All interactions happen in real-time within Google Colab and Gradio, ensuring an easy and interactive learning experience

7. API Documentation

EduTutor AI is built as a single-file Gradio application, so backend endpoints are not exposed separately like FastAPI. Instead, all functionalities are available through the Gradio interface.

Core Functions:

Concept Explanation

Input: Any subject/topic (e.g., “Explain Machine Learning”)

Output: AI-generated explanation in simple language.

Quiz Generator

Input: Topic or concept (e.g., “Generate 5 MCQs on Python Functions”)

Output: 5 multiple-choice questions with correct answers.

Testing:

All functions can be tested directly within Gradio UI.

No Swagger documentation required since the project runs inside Colab with interactive interface.

8. User Interface

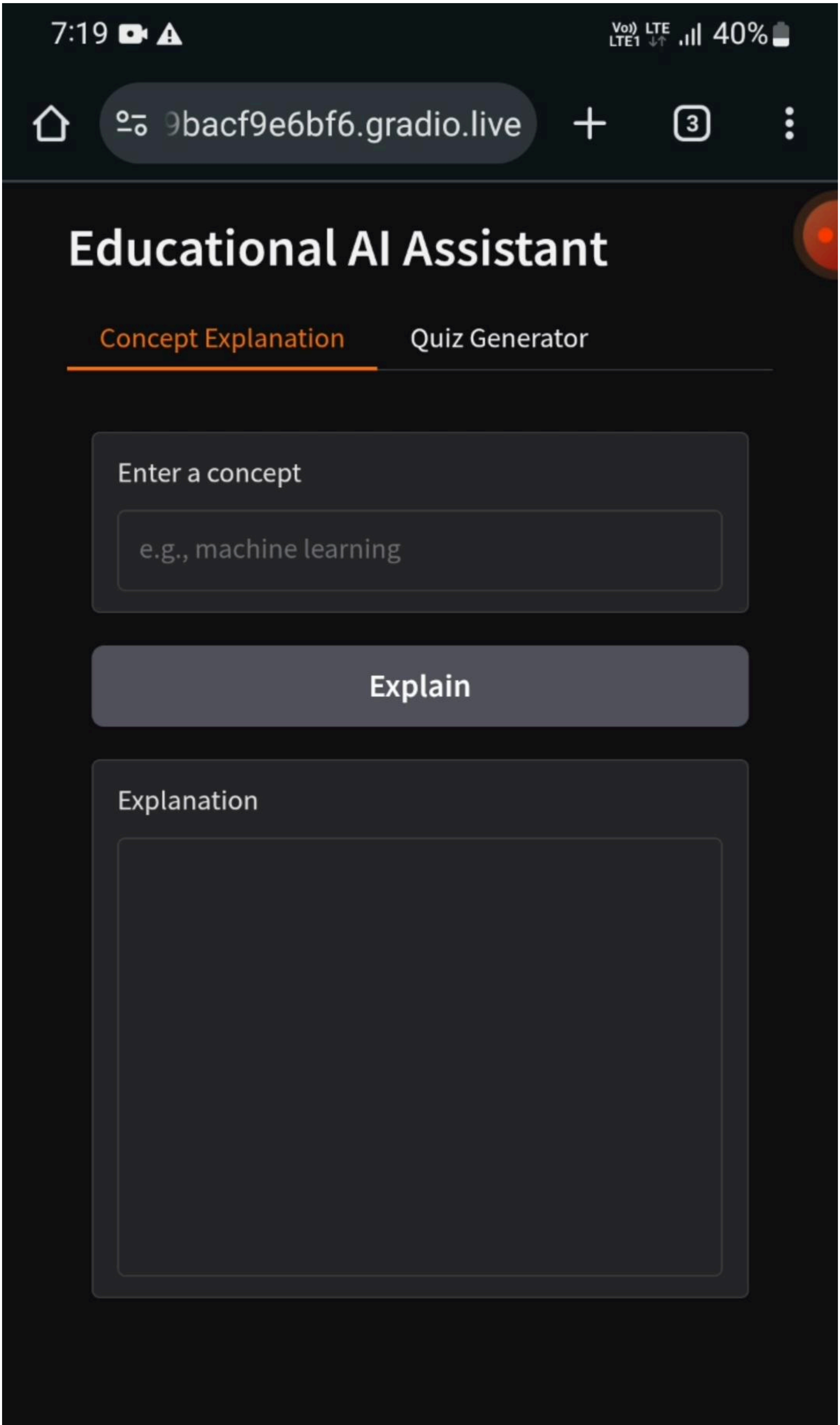
EduTutor AI provides a simple and user-friendly interface built with Gradio. The application has two main modules:

1. Concept Explanation

Input: Any concept or topic (e.g., Machine Learning, OOP in Python).

Output: Clear and concise explanation generated by the IBM Granite model.

Use Case: Helps students quickly understand new topics in simple terms.



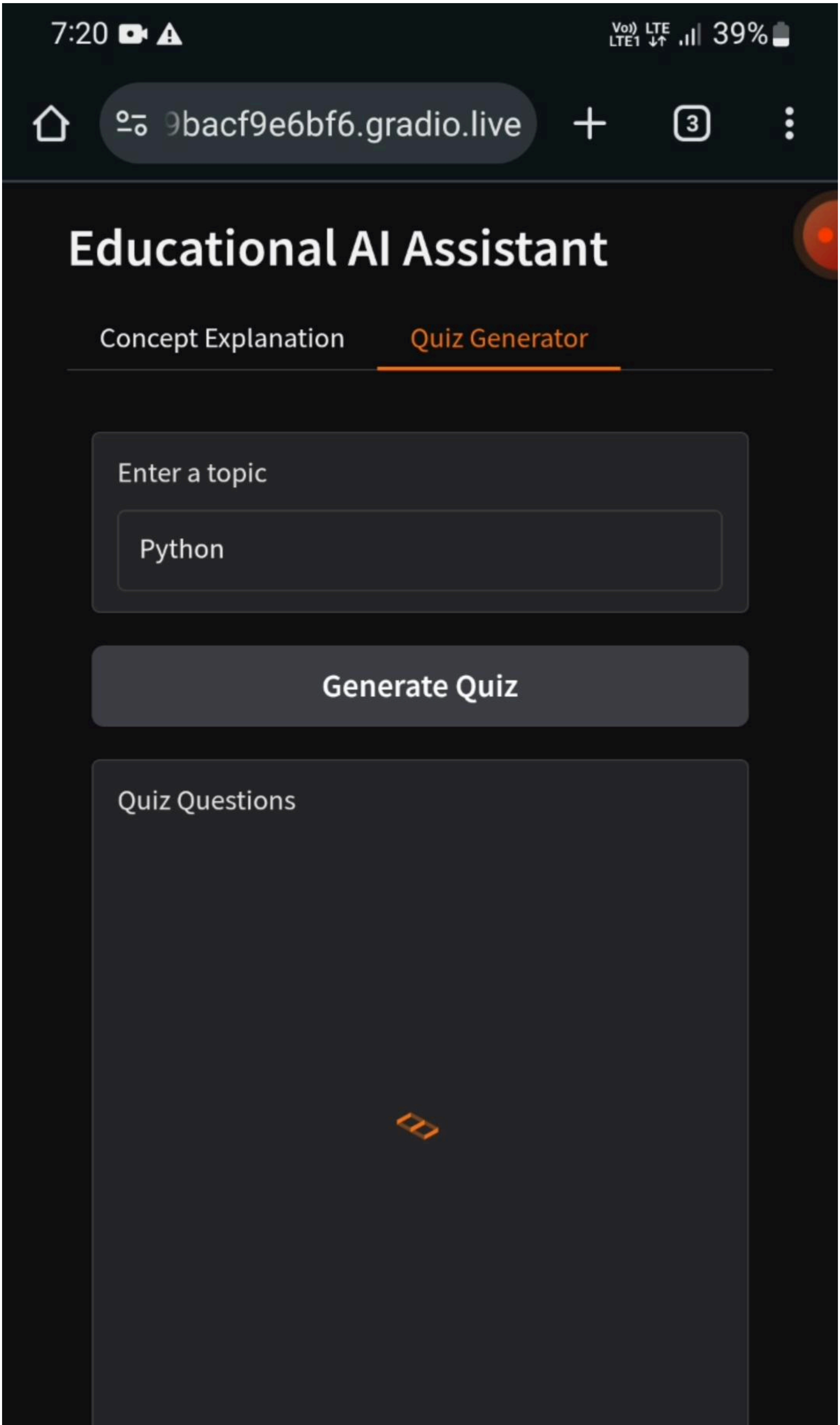
Educational AI Assistant

2. Quiz Generator

Input: A topic or subject name (e.g., Python).

Output: 5 Multiple Choice Questions (MCQs) along with correct answers.

Use Case: Useful for self-assessment, practice, and teacher-led assignments.



Educational AI Assistant

Enter a topic

Python

Quiz Questions