

EDUCATIONAL AI ASSISTANT USING IBM GRANITE LLM

PROJECT DOCUMENTATION

1.Introduction

Project Title: **Educational AI Assistant Using IBM Granite LLM**

1.Team Member: Pavithra S

2.Team Member: Poongavanam P

3.Team Member: Pradhima S

4.Team Member: Revathi M

2. Project Overview

Purpose:

---> To explain difficult academic concepts in simple terms with examples.

---> To generate quiz questions for practice and self-assessment.

Features:

---> Concept Explanation: Users can enter any concept (e.g., “machine learning”),and the system provides a detailed explanation with real-world examples.

---> Quiz Generator: Users can enter a topic (e.g., “physics”), and the system generates 5quiz questions in multiple formats (MCQ, True/False, short answer) with an ANSWERS section.

---> Simple Interface: Organized into two tabs for easy navigation.

---> Interactive Learning: Provides instant responses, making the learning process engaging.

3. Architecture

---> Frontend (Gradio): A two-tab interface created with Gradio for “Concept Explanation”

and “Quiz Generator.”

---> Backend (Python): Handles model loading, input processing, and response generation.

---> LLM Integration (IBM Granite): Uses IBM Granite 3.2-2B-Instruct for natural language understanding and generation.

---> Hosting: Runs in Google Colab or locally, with Gradio providing a shareable web link.

4. Setup Instructions

Prerequisites:

- Python environment or Google Colab.
- Libraries: torch, transformers, gradio.
- Hugging Face account (recommended for model access).

Installation:

1. Open Google Colab or local IDE.
2. Install required libraries using pip.
3. Load the IBM Granite model from Hugging Face.
4. Run the provided code.

Running the Application

1. Execute the notebook or Python file.
 2. Launch the Gradio app using `app.launch(share=True)`.
 3. A public link will be generated.
 4. Open the link in a browser.
- In Concept Explanation Tab, enter a concept → click Explain → detailed explanation appears.
 - In Quiz Generator Tab, enter a topic → click Generate Quiz → quiz with answers is displayed.

5. User Interface

---> Tabs: Concept Explanation, Quiz Generator.

---> Inputs: Textboxes for entering concept or topic.

---> Outputs: Textboxes displaying explanations or quizzes.

---> Buttons: “Explain” and “Generate Quiz” to trigger responses.

---> Accessibility: Public Gradio link makes the app accessible for browsers

Doc app.py → Your main script.

requirements.txt → Lists all Python libraries to install.

models/ → (Optional) Local model storage if you don’t want to download from Hugging Face every time.

docs/ → Holds all documentation (program explanation, folder structure, user guide).

tests/ → Space for adding test cases.

assets/ → Images/screenshots used in docs.

README.md → Quick start guide for user

6. Test Plan

Objective:

Ensure that the Educational AI Assistant works correctly, generates valid responses, and the UI is functional.

Model loading

Gradio interface (UI tabs, input/output fields)

Concept explanation generation

Quiz generation with answers

7.Tools Used:

pytest (for automated testing)

gradio_client (optional for UI testing)

8.Web Interface

1. *Concept Explanation Tab* → Type a concept (e.g., machine learning) → Click Explain → AI gives explanation.

2. *Quiz Generator Tab* → Type a topic (e.g., physics) → Click Generate Quiz → AI creates quiz + answers.

9.Known issues

Model & Performance

Large Model Size – The ibm-granite-3.2-2b-instruct model is heavy; it requires good hardware (GPU recommended). On CPU, responses may be slow.

Memory Usage – Running on low-RAM systems may cause out-of-memory errors.

Response Time – Quiz generation and long explanations may take several seconds to complete.

2. Functionality

Repetitive Outputs – Sometimes the model repeats parts of the input in the output.

Quiz Format Issues – Generated quizzes may not always follow the requested format (e.g., fewer than 5 questions or missing answer section).

Empty Input Handling – If the user submits an empty concept/topic, the app may return unclear or blank responses instead of a friendly error message.

Prompt Sensitivity – Output quality depends heavily on prompt phrasing. Some vague prompts may return generic or irrelevant answers.

3. Technical & Deployment

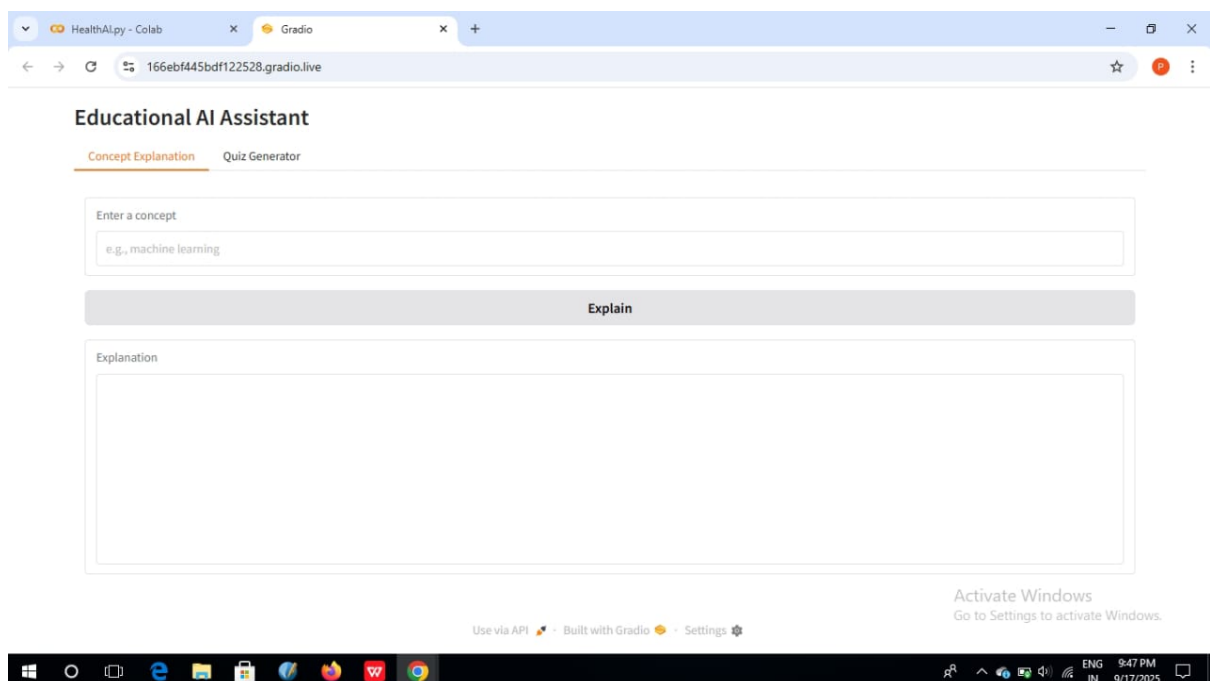
Dependency Versions – Newer versions of transformers or torch may cause compatibility issues.

Slow First Run – The first run downloads the model from Hugging Face, which may take several minutes depending on internet speed.

Gradio Share Links – The share=True feature works but may sometimes fail due to network/firewall restrictions.

10.Screenshots

Outputs:



HealthAI.py - Colab x Gradio

166ebf445bdf122528.gradio.live

Educational AI Assistant

Concept Explanation **Quiz Generator**

Enter a topic

cloud computing

Generate Quiz

Quiz Questions

1. **Multiple Choice:** What type of cloud computing deployment model allows for the greatest degree of customization and control over infrastructure?
A) Infrastructure as a Service (IaaS)
B) Platform as a Service (PaaS)
C) Software as a Service (SaaS)
D) Serverless
2. **True or False:** The primary advantage of using cloud computing is the elimination of upfront hardware costs.
3. **Short Answer:** Briefly explain the concept of **multi-tenancy** in the context of cloud computing.
4. **Multiple Choice:** Which of the following is NOT a typical service offered by a cloud provider?
A) Virtual machine instances
B) Container orchestration (e.g., Kubernetes)
C) Serverless computing
D) **Data center facilities**

Activate Windows
Go to Settings to activate Windows.

Windows taskbar: 9:50 PM 9/17/2025

HealthAI.py - Colab x Gradio

166ebf445bdf122528.gradio.live

Educational AI Assistant

Concept Explanation Quiz Generator

Enter a concept

cloud computing

Explain

Explanation

3. **Software as a Service (SaaS):** At the topmost level, SaaS delivers fully functional applications to end-users via web browsers, eliminating the need for localized software installations. Users access these applications through a subscription model, with the CSP handling all maintenance, updates, and security. Examples range from Office 365 (productivity tools) to Salesforce (CRM) and Slack (team collaboration).

Example:
- An accountant, C, subscribes to Microsoft 365. Now, C can access Word, Excel, PowerPoint, and other Office applications directly from any web browser, without the need to install and maintain the software on their personal computers. This setup also ensures that C always uses the latest version of these tools, updated by Microsoft.

4. **Serverless Computing:** A subset of cloud computing, serverless architectures allow developers to run code without managing servers. The cloud provider automatically manages the underlying infrastructure, including provisioning, scaling, and capacity planning. Examples include AWS Lambda, Azure Functions, and Google Cloud Functions.

Example:
- Web developer D creates a simple API to fetch weather data for various locations. Instead of maintaining a server for this API, D writes the code using JavaScript in an AWS Lambda function. AWS Lambda automatically provisions the necessary resources to execute the code whenever the API receives a request, scales it according to the need, and manages the costs.

5. **Multi-cloud and Hybrid Cloud:** These strategies involve using multiple CSPs' services (multi-cloud) or combining on-premises infrastructure with cloud services (hybrid cloud).

Windows taskbar: 9:49 PM 9/17/2025

11. Future Enhancements

- > Add support for subject-specific quizzes (math, science, coding, etc.).
- > Provide explanations with diagrams or flowcharts for better understanding.
- > Enable downloading quizzes in PDF format.
- > Add speech-based input/output for accessibility.
- > Deploy the system on a dedicated sever for continuous availability.