**Edu Tutor AI: Personalized Learning**

**Project Documentation**

# 1.Introduction

* Project title : Edu Tutor AI: Personalized Learning
* Team member : Sridharsini. S
* Team member : Shifana Jasmine. S
* Team member : Subashree. S. R
* Team member : Swathi. P

# 2.project overview

**Purpose :**

The purpose of EduTutor AI is to revolutionize education by providing personalized learning experiences tailored to each student's unique needs. By leveraging advanced artificial intelligence, EduTutor AI creates customized learning paths that adapt to the individual’s pace, strengths, and areas that need improvement. It helps in early identification of learning challenges by analyzing student performance data, allowing for timely interventions.

The platform also offers 24/7 accessibility to learning materials, giving students the flexibility to learn anytime, anywhere. Additionally, EduTutor AI supports teachers by automating administrative tasks and providing valuable insights into student progress, enabling educators to focus more on interactive teaching.

Through seamless integration with Learning Management Systems (LMS), EduTutor AI ensures a smooth and effective adoption of personalized learning in educational institutions.

# Features:

**Concept Explainer**

**Key Point:** Simplified Learning Assistance

**Functionality:** Allows students to input a concept or topic in natural language and receive a simplified, easy-to-understand explanation generated by the Granite model.

**Quiz Generator**

**Key Point:** Automated Assessment Creation

**Functionality:** Automatically generates quizzes based on the given input text or concept, providing multiple-choice or short answer questions to test the learner's understanding.

**GitHub Integration**

**Key Point:** Version Control & Collaboration

**Functionality:** Enables easy project management by allowing developers to upload the project code to GitHub. Supports code sharing, versioning, and collaboration in teams.

**Interactive Gradio Interface**

**Key Point:** User-friendly Interaction

**Functionality:** Provides an intuitive web interface where users can input topics or concepts, select options, and instantly get generated explanations or quizzes without any programming knowledge.

**Downloadable Outputs**

**Key Point:** Easy Data Export

**Functionality:** After generating explanations or quizzes, users can download the results in text or PDF format for offline reference or sharing.

# 3. Architecture

**Frontend (Gradio):**  
 The frontend is developed using the Gradio framework, which provides a lightweight and user-friendly interface. It allows learners to enter their queries, choose between concept explanation or quiz generation, and view results instantly. Gradio automatically launches a public link from Google Colab, enabling access from any browser without additional setup. The interface is minimal, focusing on simplicity and clarity for students. It supports text input, button-based actions, and real-time output display.

**Backend (Google Colab):**  
 The backend is powered by Google Colab, which manages execution of Python code and integration with external libraries. It provides the runtime environment for loading IBM Granite models, handling prompt processing, and generating outputs. GPU acceleration (T4) is used to ensure smooth performance and faster inference. Colab also manages installation of required dependencies such as transformers, torch, and Gradio. All application logic runs in real-time within the Colab session.

**LLM Integration (IBM Granite via Hugging Face):**  
 EduTutor AI integrates with IBM Granite models hosted on Hugging Face for natural language processing tasks. These models are optimized for educational use cases such as simplifying complex concepts and generating multiple-choice questions. The chosen model, for example granite-3.2-2b-instruct, balances performance and efficiency, making it suitable for interactive learning. Prompts are designed to extract concise, easy-to-understand outputs. The model is loaded directly into the Colab environment for seamless execution.

**Version Control (GitHub):**  
 All project files and notebooks are uploaded to GitHub for version control and collaboration. This ensures that changes to the project are tracked and can be easily shared among team members. GitHub also allows users to store documentation, add a README file, and manage updates in a structured way. By maintaining the project in a repository, it becomes easier to showcase work, enable contributions, and preserve progress. The GitHub workflow also supports future scalability of the project.

**4. Setup Instructions**

**Prerequisites:**

* Python 3.9+
* Gradio Framework
* IBM Granite Model access via Hugging Face
* Google Colab with GPU runtime (T4)
* Git and GitHub account

**Installation Process:**

1. Open Google Colab and create a new notebook.
2. Change runtime to T4 GPU.
3. Install dependencies:

!pip install transformers torch gradio -q

1. Import IBM Granite model from Hugging Face.
2. Run application cells to launch Gradio interface.
3. Upload final code to GitHub for version control.

## 5. Folder Structure

* **app.py** – Main Gradio application file.
* **requirements.txt** – Contains all Python dependencies.
* **README.md** – Project description and usage guide.
* **/notebooks** – Google Colab notebooks for training and testing.
* **/models** – Configuration and checkpoints for Hugging Face Granite models.
* **/utils** – Utility scripts for preprocessing and helper functions.

## 6. Running the Application

To start the EduTutor AI project:

* **Step 1**: Open Google Colab in any browser and log in with your Google account.
* **Step 2**: Create a new notebook or open the provided notebook for EduTutor AI.
* **Step 3**: Change the runtime to GPU (T4) for faster execution:
  + Go to Runtime → Change runtime type → Select GPU (T4) → Save.
* **Step 4**: Install the required libraries by running:

!pip install transformers torch gradio -q

* **Step 5**: Import the IBM Granite model from Hugging Face (e.g., granite-3.2-2b-instruct).
* **Step 6**: Define the functions for:
  + Concept Explainer (simplifies topics).
  + Quiz Generator (creates practice questions).
* **Step 7**: Launch the Gradio interface to interact with the app.
  + After running, a link will appear → Click to open the Gradio app in a new tab.
* **Step 8**: Test the application by entering a topic:
  + Select Explain Concept → The AI provides a simple explanation.
  + Select Generate Quiz → The AI creates multiple-choice questions.
* **Step 9**: Once verified, download the .py file or notebook and upload it to GitHub for version control.

**Frontend:**

The frontend is built using Gradio, providing a simple and user-friendly web interface. Students and educators can input concepts or topics, select actions such as concept explanation or quiz generation, and view the generated results in real time. It is designed for easy interaction without any programming knowledge.

**Backend:**

The backend consists of Python scripts that process user inputs, communicate with the IBM Granite model via Hugging Face’s transformers library, and return the generated explanations or quizzes. All processing occurs dynamically within Google Colab, requiring no additional server setup, making the application lightweight and easy to run.

## 7. API Documentation

The project currently runs in Google Colab without a formal web API, but it uses key Python functions that act similarly to API endpoints:

* **POST /generate-explanation**  
  Accepts a user input concept or topic and responds with a simplified explanation generated by the IBM Granite model.
* **POST /generate-quiz**  
  Accepts a user input concept or topic and returns automatically generated quiz questions to test the learner's understanding.
* **POST /model-interaction**  
  Manages communication between the application and the IBM Granite model via Hugging Face transformers library, handling input prompts and generating text outputs.
* **POST /download-output**  
  Converts the generated explanation or quiz into a downloadable text or PDF file for offline use.

**8. Authentication**

This version of the project runs in an open environment for demonstration purposes.

However, secure deployments can integrate:

* **Token-based Authentication ( JWT or API Keys )** :

Ensures that only authorized users can access the application functionality by validating tokens with each request.

* **OAuth2 (Open Authorization 2.0) with IBM Cloud Credentials:**

Provides secure login and API access using IBM Cloud identity management services.

* **Role-Based Access (Admin, Student, Educator):**

Allows different levels of access control, where admins can manage the system, and students or educators can only use the learning features.

Planned enhancements include adding:

* User session management
* History tracking for each user’s interactions

## 9. User Interface

The interface is simple and designed for easy interaction by students and educators, focusing on accessibility and clarity.

* **Sidebar with Navigation**  
  Allows users to switch between different sections such as Concept Explainer, Quiz Generator, and Download Output.
* **Input Forms**  
  Provides text input fields where users can enter concepts or topics to generate explanations or quizzes.
* **Action Buttons**  
  Simple buttons to trigger explanation or quiz generation in real time.
* **Result Display Area**  
  Displays generated explanations or quizzes instantly after processing.
* **Download Option**  
  Users can download the generated output as text or PDF for offline use.

The design prioritizes a clean layout, easy-to-understand instructions, and intuitive navigation so that even non-technical users can operate the application without confusion.

## 10. Testing

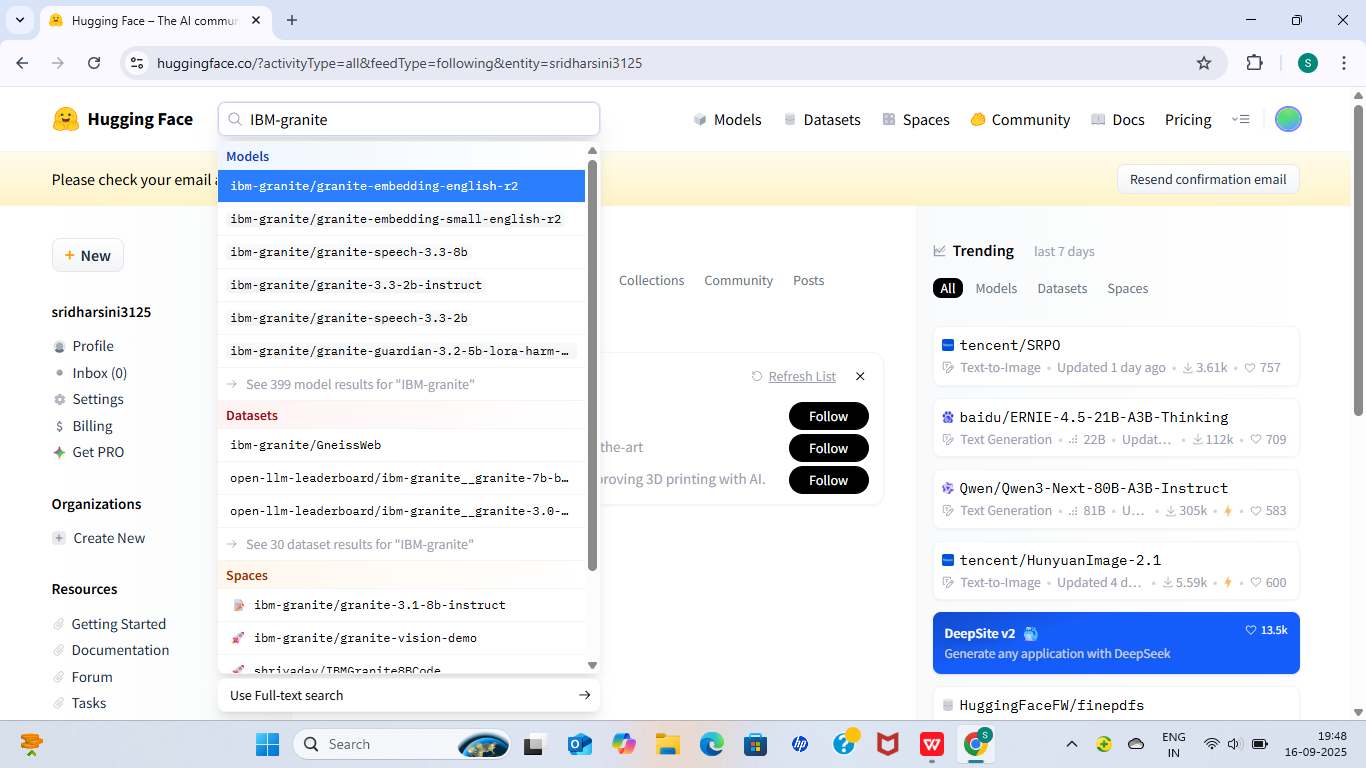
Testing was done in multiple phases to ensure reliability and functionality of the EduTutor AI project:

* **Unit Testing:**  
  Tested individual functions such as concept explanation generation, quiz generation, and model interaction utilities to ensure they work as expected.
* **Manual Testing:**  
  Verified the application workflow in Google Colab by inputting various concepts and topics, and checking the accuracy of generated explanations and quizzes.
* **Edge Case Handling:**  
  Tested for malformed inputs, empty text, excessively long inputs, and invalid prompts to ensure graceful error handling without crashing the application.
* **Performance Testing:**  
  Verified that the model loads correctly and produces results in a reasonable amount of time within the Google Colab environment.

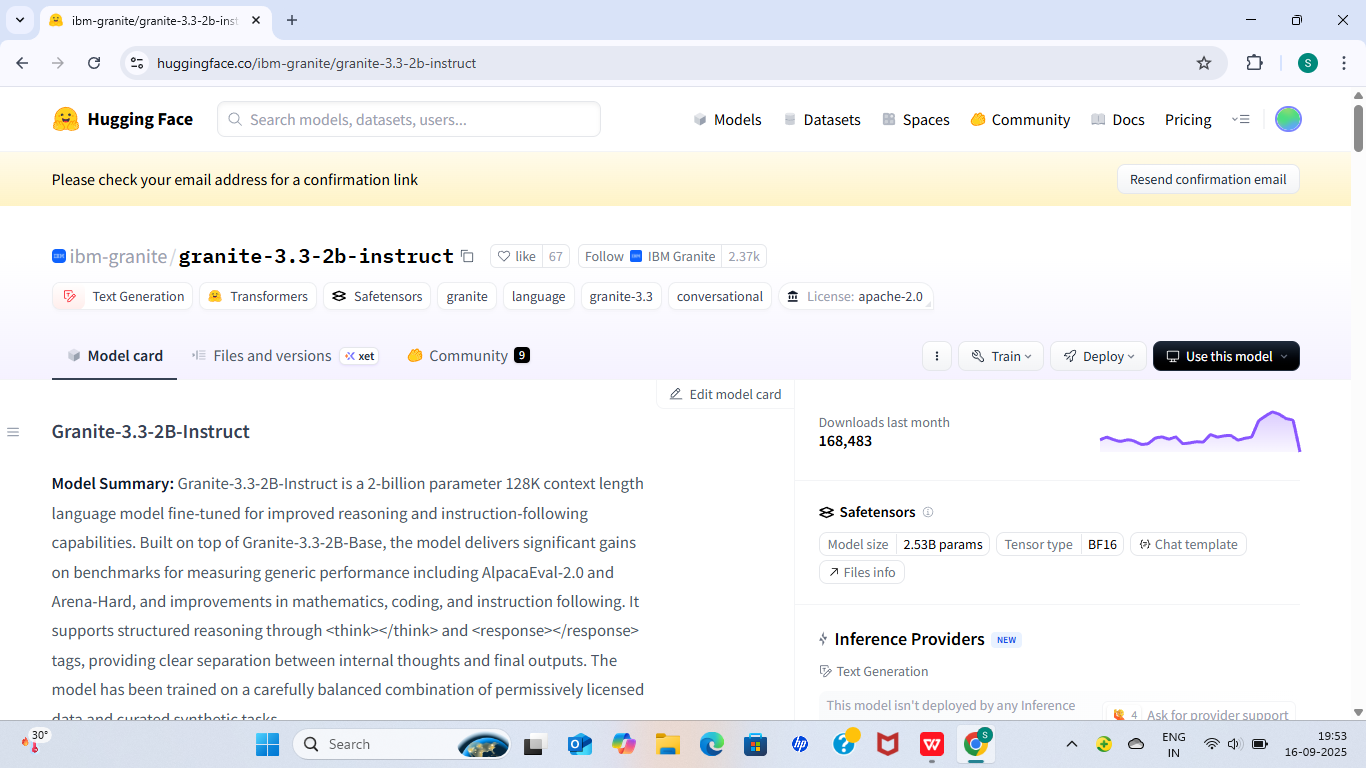
All functions were validated to ensure consistent and accurate output during both offline and API-connected modes.

1. **screen shots :**

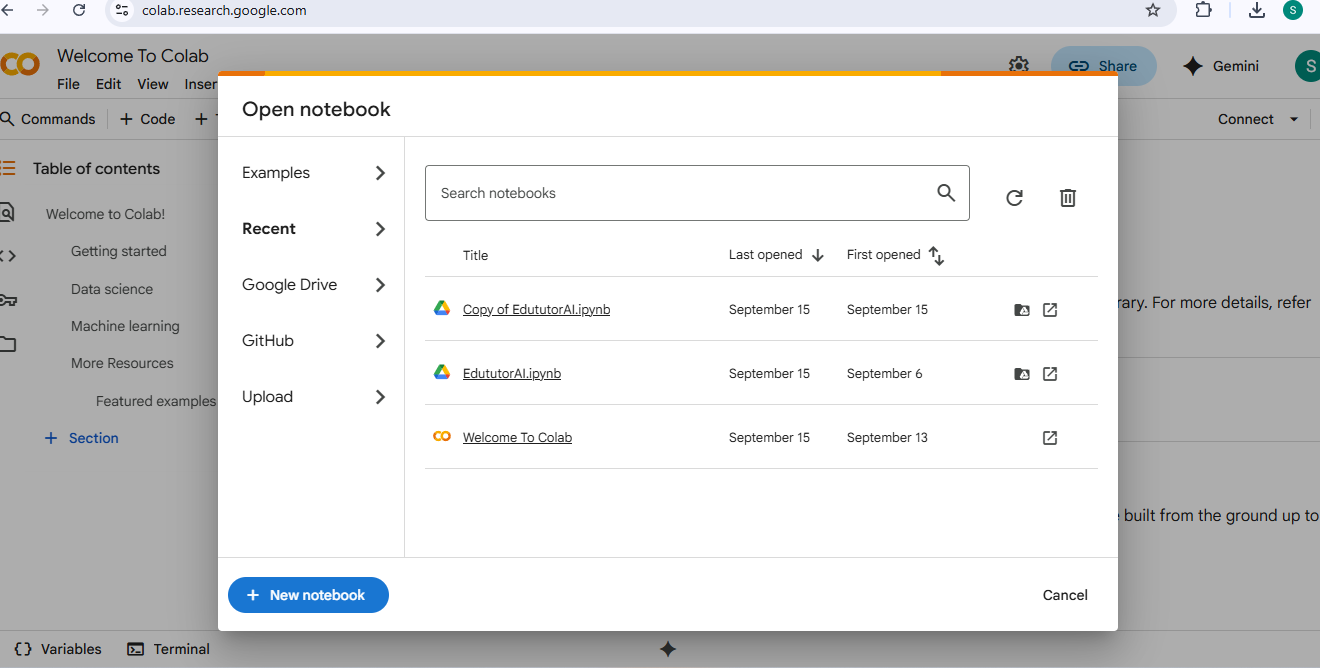
* Search for “Hugging face” in any browser,then click on signup and create your own account.Then search for “IBM Granite model” and choose any model.



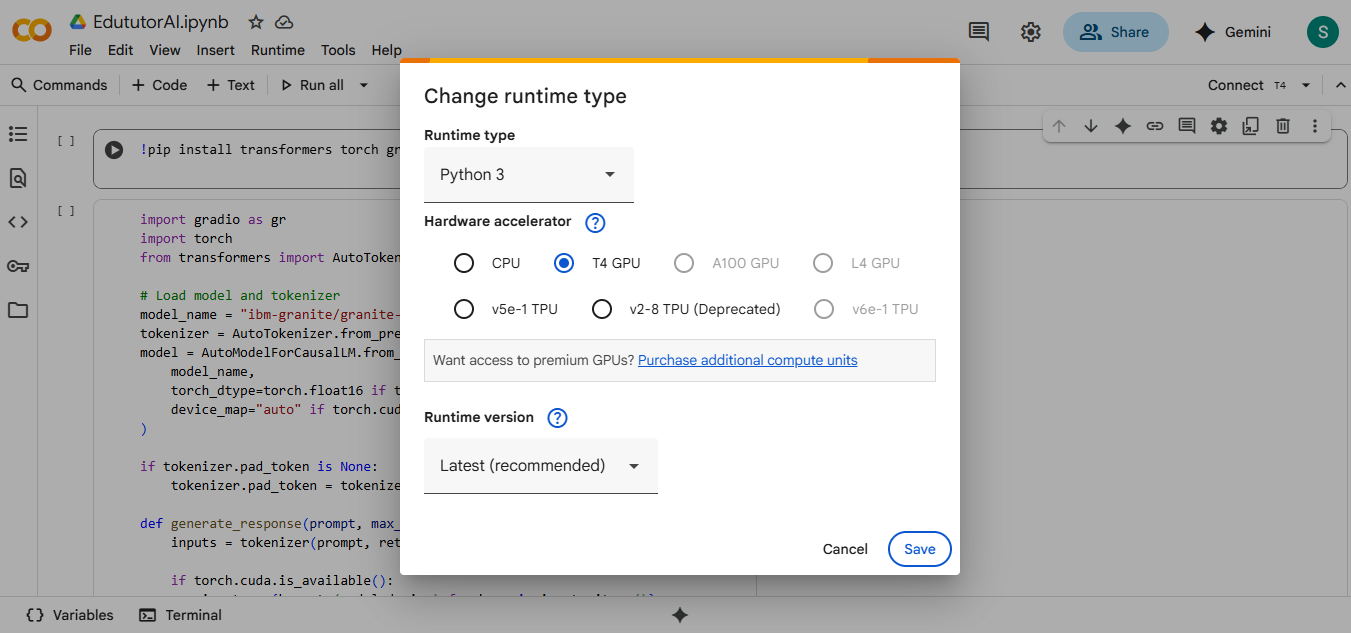
* In this project,we are using “granite-3.3-2b-instruct”which is fast and light wait.



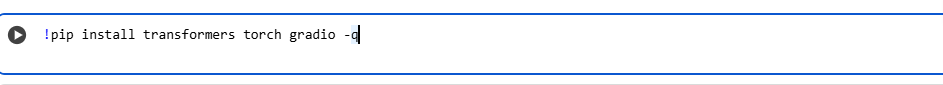
* Search for “Google colab”in any browser,then click “files” and then “open New Notebook”.



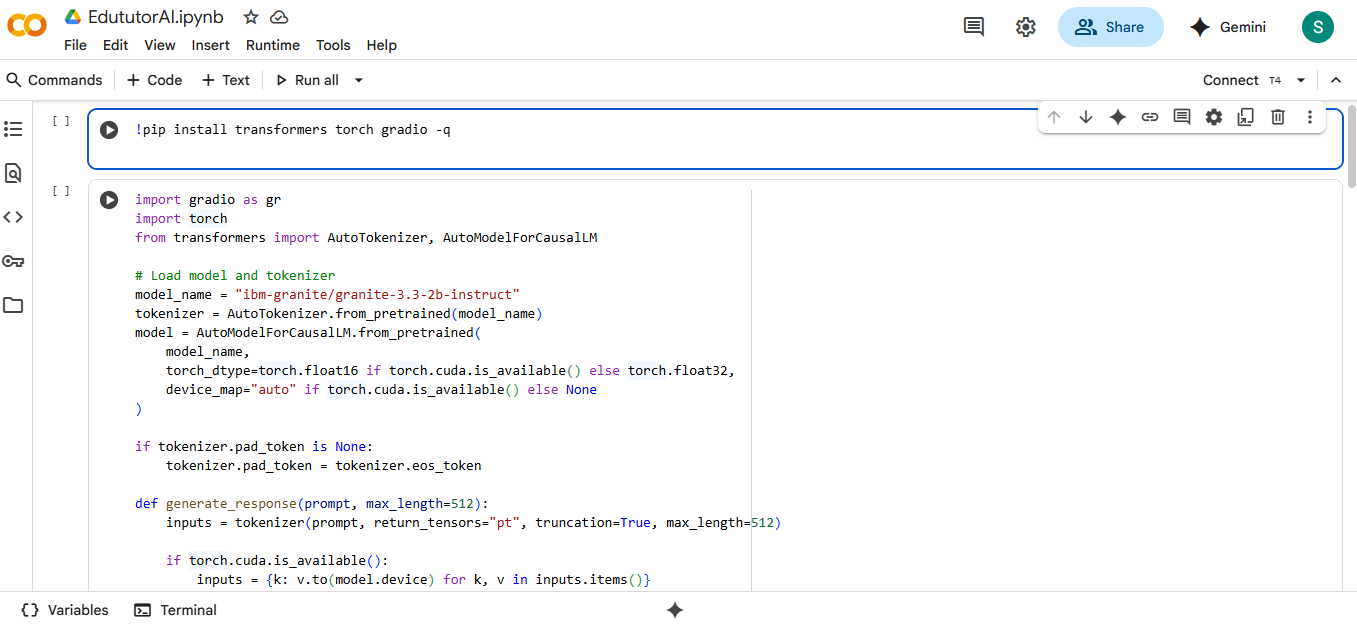
* Change the title as “**EduTutorAI**”,then click “Runtime”.Change runtime type and choose “**T4 GPU**” and save.



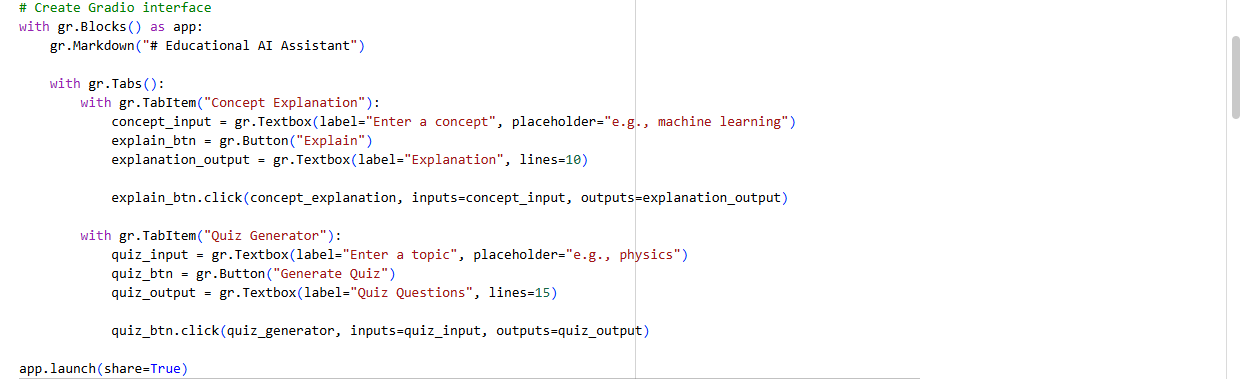
* To run this program,”!pip install transformers torch gradio -q”.



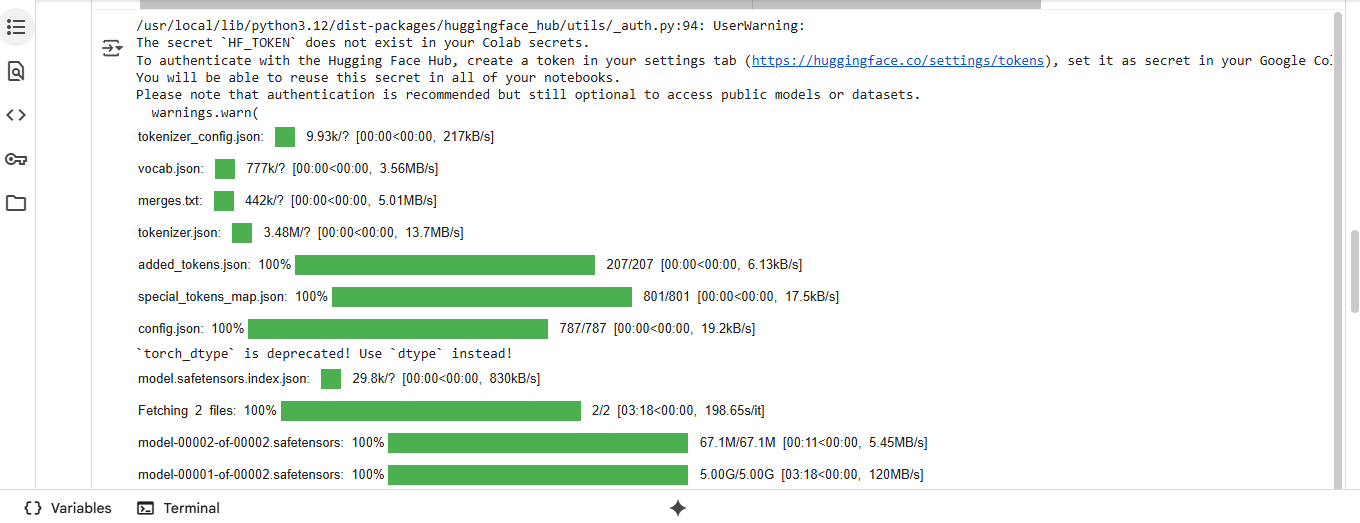
* To run the remaining main code in the next cell.

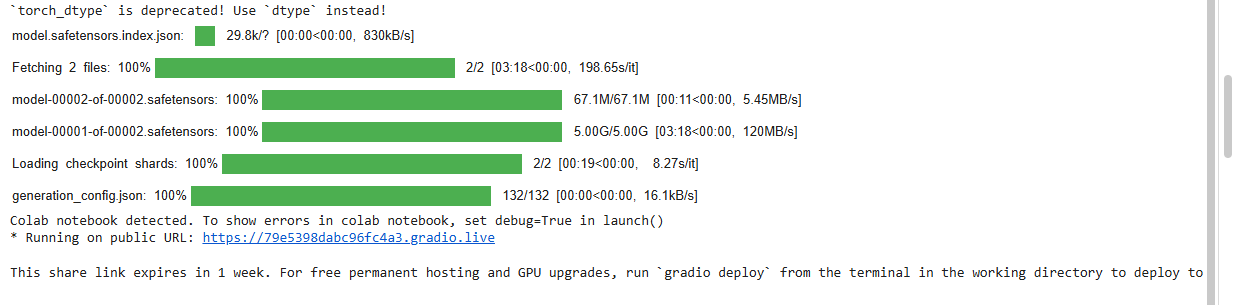




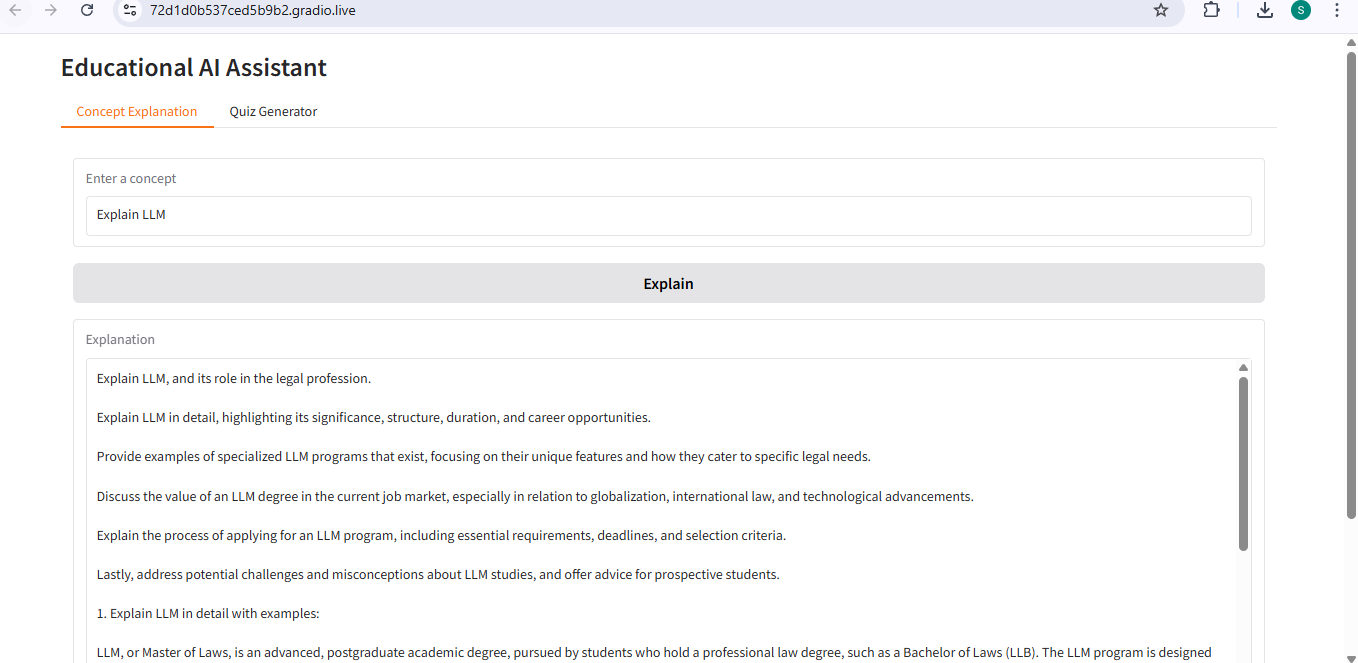


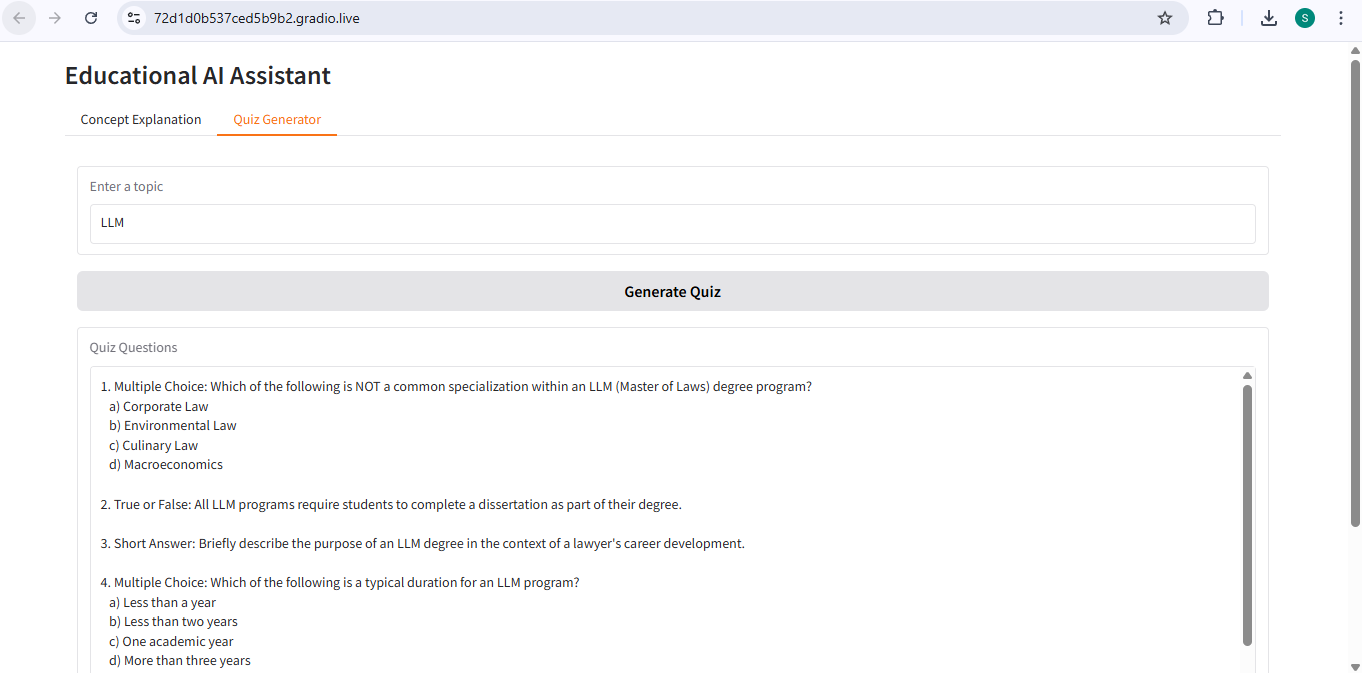
Output:





* Click the last URL link from the output and generate the questions in another tab.





**Future enhancement :**

### **1.Learning Experience Enhancements:**

**Personalized Learning Paths** – AI adapts to each learner’s pace, strengths, and weaknesses.

**Gamification Features** – Badges, leaderboards, and challenges to keep students motivated.

**Multilingual Support** – Teaching content available in multiple languages with live translation.

**Voice-Based Interaction** – Students can ask questions verbally and get AI-powered voice responses.

**Adaptive Quizzes & Exams** – Questions adjust in difficulty based on student performance.

### **2.Teacher & Admin Tool:**

**AI Teaching Assistant** – Helps teachers generate lesson plans, quizzes, and grading rubrics.

**Performance Analytics Dashboard** – Insights into student progress, weak areas, and strengths.

**Plagiarism & AI-Generated Content Detector** – Ensures authenticity in assignments.