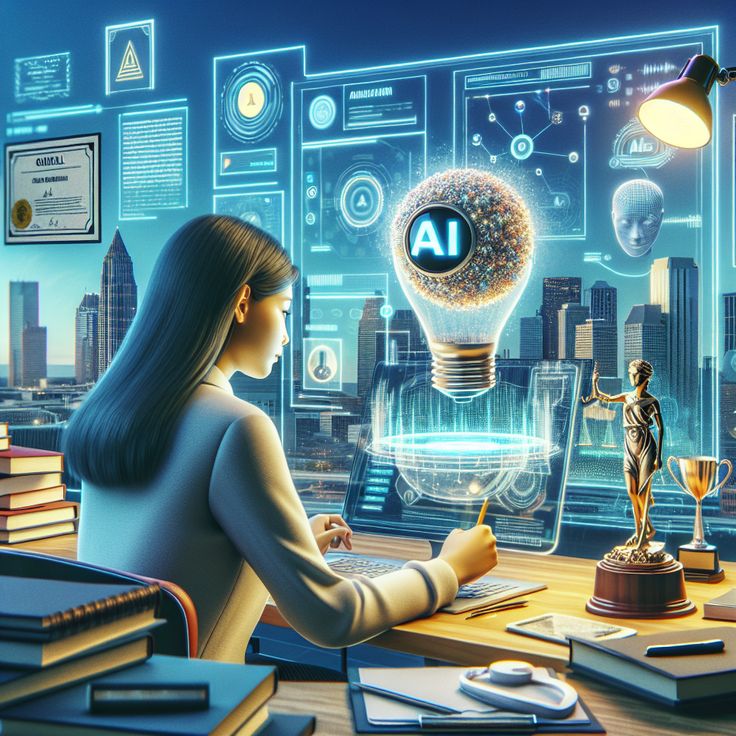
**EDU TUTOR AI: PERSONALIZED LEARNING**

**GENERATIVE AI WITH IBM**

****

**1. Introduction**

* **Project Title:** EduTutor AI : Personalized Learning
* **Team Members:J.Gopika**
* **Team Members:V.Gomathi**
* **Team Members:M.Kaviya**
* **Team Members:M.Anusharathi**
* **Team Members:M.Devadharshini**

**2. Project Overview**

**Purpose:**  
The purpose of EduTutor AI is to provide students with a **personalized virtual**

**tutoring system** powered by Artificial Intelligence. It acts as a **24/7 intelligent assistant** that supports doubt clarification, practice tests, progress tracking,and multilingual learning. By leveraging AI and LLMs, it helps bridge gaps in education, reduce dependence on private tuition, and provide equal access to quality learning.

**Features:**

* **Conversational Interface**  
  Key Point: Natural language Q&A  
  Functionality: Students can ask questions and get instant explanations.
* **Personalized Learning**  
  Key Point: Adaptive study plans  
  Functionality: Adjusts content difficulty based on student performance.
* **Multi-Subject Support**  
  Key Point: Covers academics & skill-based subjects  
  Functionality: Provides help in coding, math, science, and languages.
* **Assessment & Feedback**  
  Key Point: Continuous evaluation  
  Functionality: Quizzes, tests, and performance reports with feedback.
* **Learning Analytics**  
  Key Point: Track progress  
  Functionality: Monitors student performance and suggests improvements.
* **Multilingual Support**  
  Key Point: Regional + global language learning  
  Functionality: Explains concepts in student’s preferred language.

**3. Architecture**

* **Frontend (Gradio):** Provides an interactive web interface for users.
* **Backend (FastAPI/Python Modules in Colab):** Handles the logic and API requests from the frontend. Processes user input and communicates with the AI Model.
* **Generates:**Explanations,Quizzes,Learning content.
* **LLM Integration(Hugging face/IBM Granite/Open-source Models):**Core AI engine for natural language understanding, tutoring, and recommendations.

**4. Setup Instructions**

**Prerequisites:**

* Google Account(for Google Colab access)
* Python 3.9+(Colab comes pre-installed)
* Hugging Face account (optional, for API/Model access)
* Basic knowledge of Python and Internet access

**Installation Steps:**

1. Open Google Colab
2. Upload the files EduTutor\_AI.ipynb
3. Runtime🡪Change runtime type🡪Select GPU
4. Run commands to install libraries
5. Load the model from Hugging face/Watsonx
6. Run all cells in the notebook. At the end you will get a Gradio link to open the Edu Tutor AI.

**5. Folder Structure**

* **app/** – Backend logic (APIs, routes, models)
* **model\_loader.py** – loads the AI Model and tokenizer
* **quiz\_generator.py--** Frontend functions
* **interface.py-** shows the input boxes
* **report\_generator.py** – Creates progress reports

**6. Running the Application**

* Start backend server (FastAP)
* Launch frontend (Gradio)
* Students login → Ask questions → Get AI tutoring → Take tests → Receive feedback

**7. API Documentation (Examples)**

* POST /chat/ask – Ask a question, get AI explanation
* POST /upload-test – Upload test results for analysis
* GET /progress-report – Fetch performance reports
* POST /feedback – Submit student feedback

**8. Authentication**

* Token-based authentication (JWT)
* Role-based access (Student, Teacher, Admin)

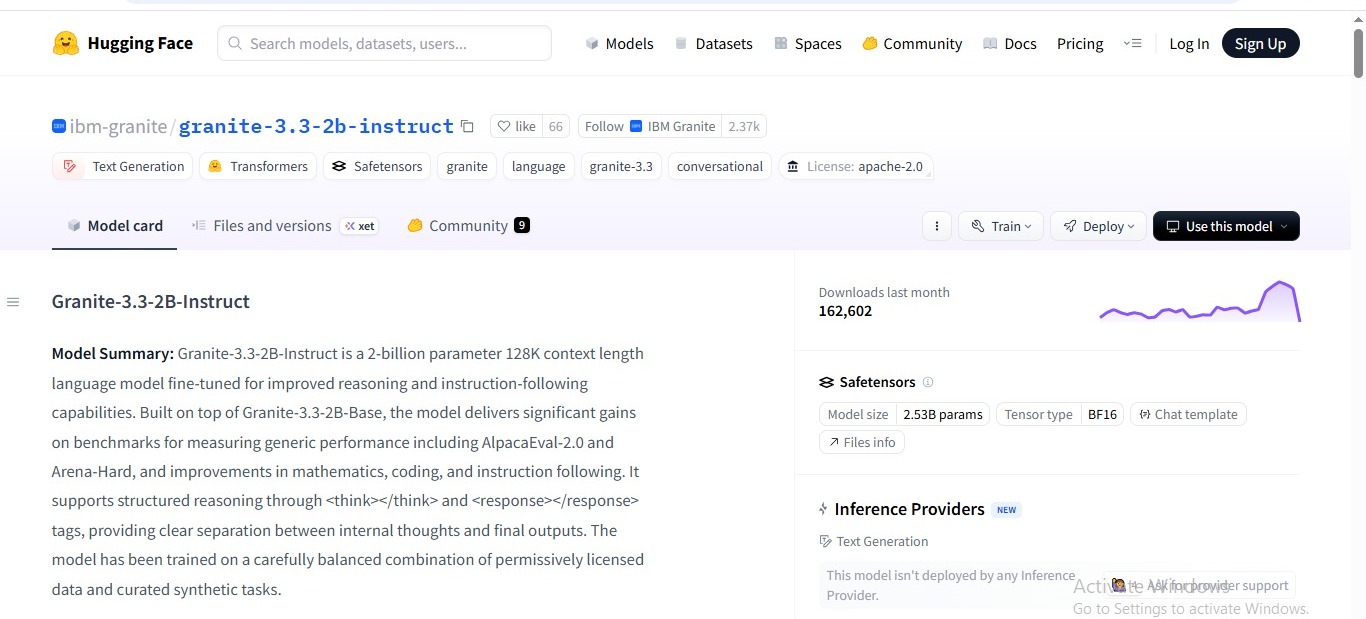
**9. User Interface**

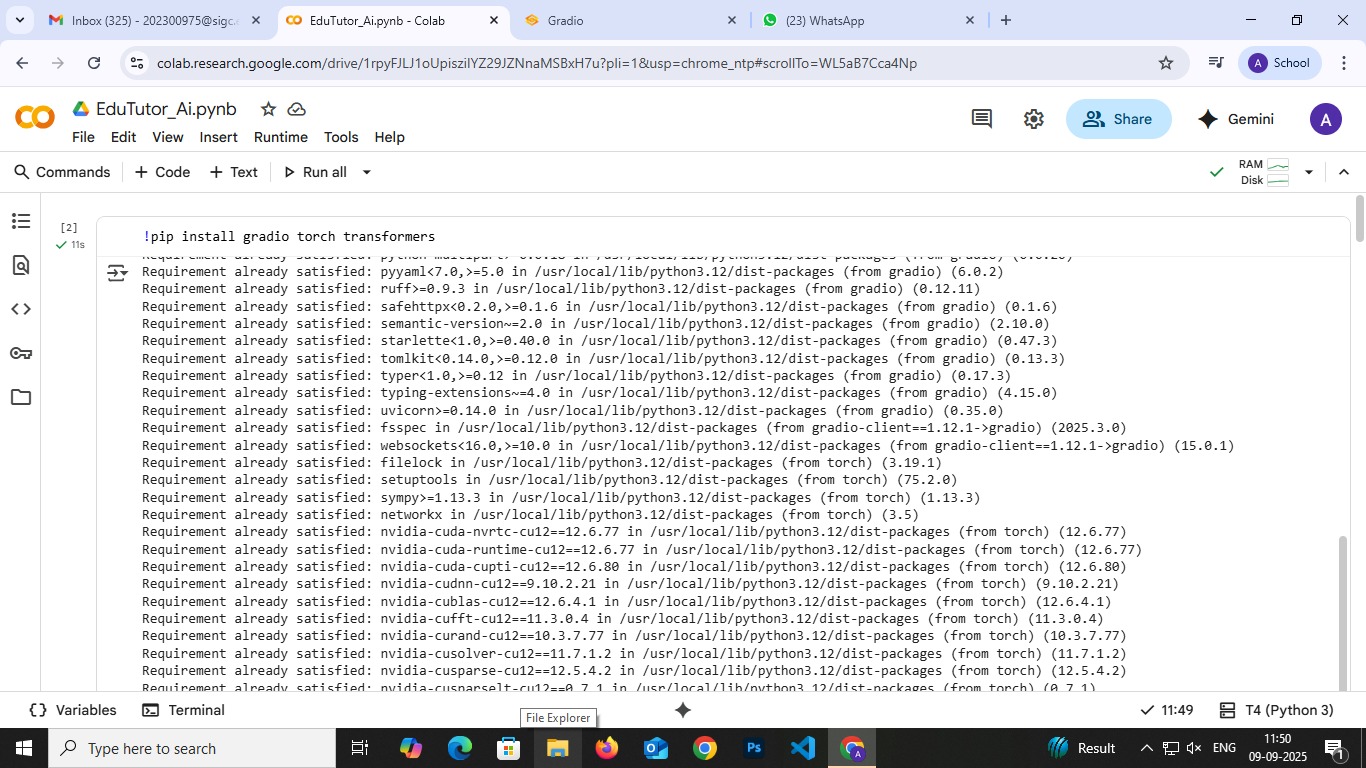
* Clean, student-friendly dashboard
* Chatbox for instant Q&A
* Test/quiz panel
* Progress visualization charts
* Downloadable reports

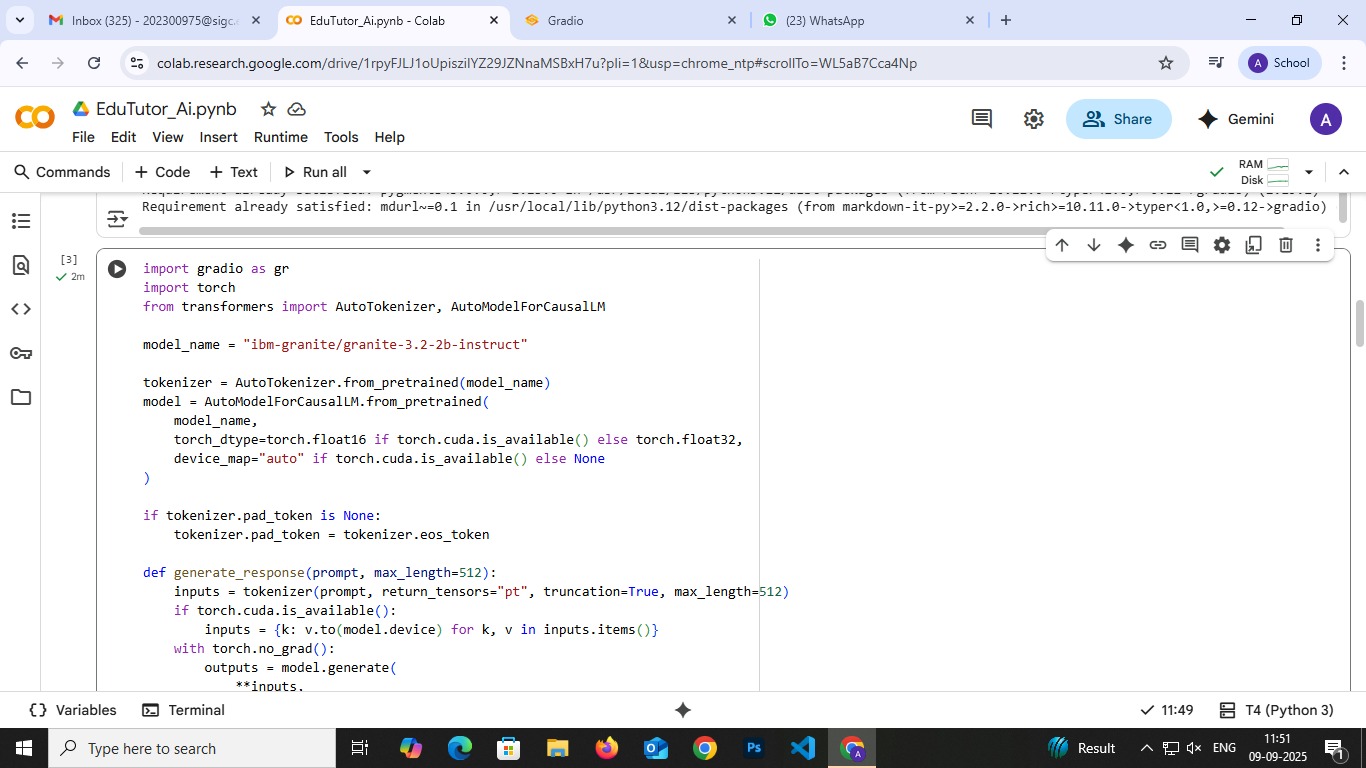
**10. Testing**

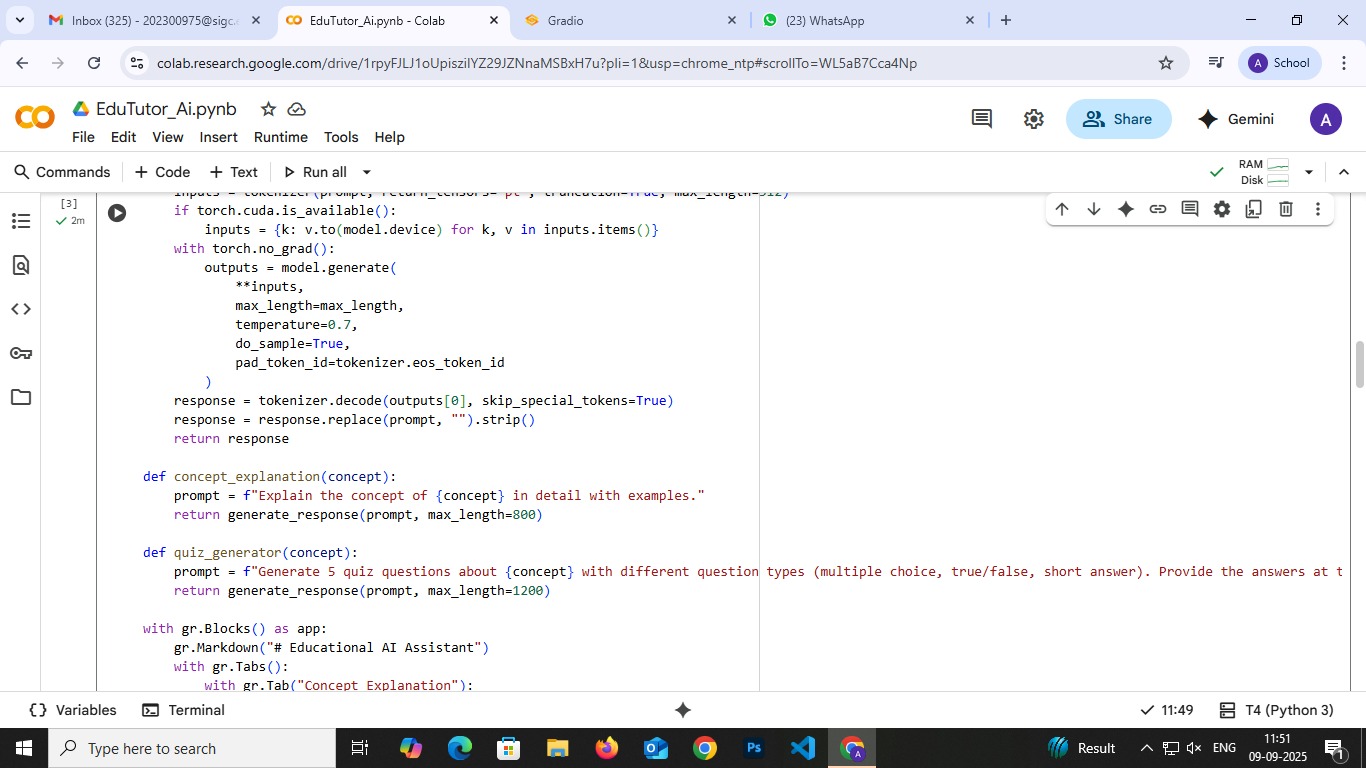
* Unit testing for AI responses
* API testing (Swagger/Postman)
* Manual testing for assessments & chat flow
* Edge case handling (invalid input, large queries)

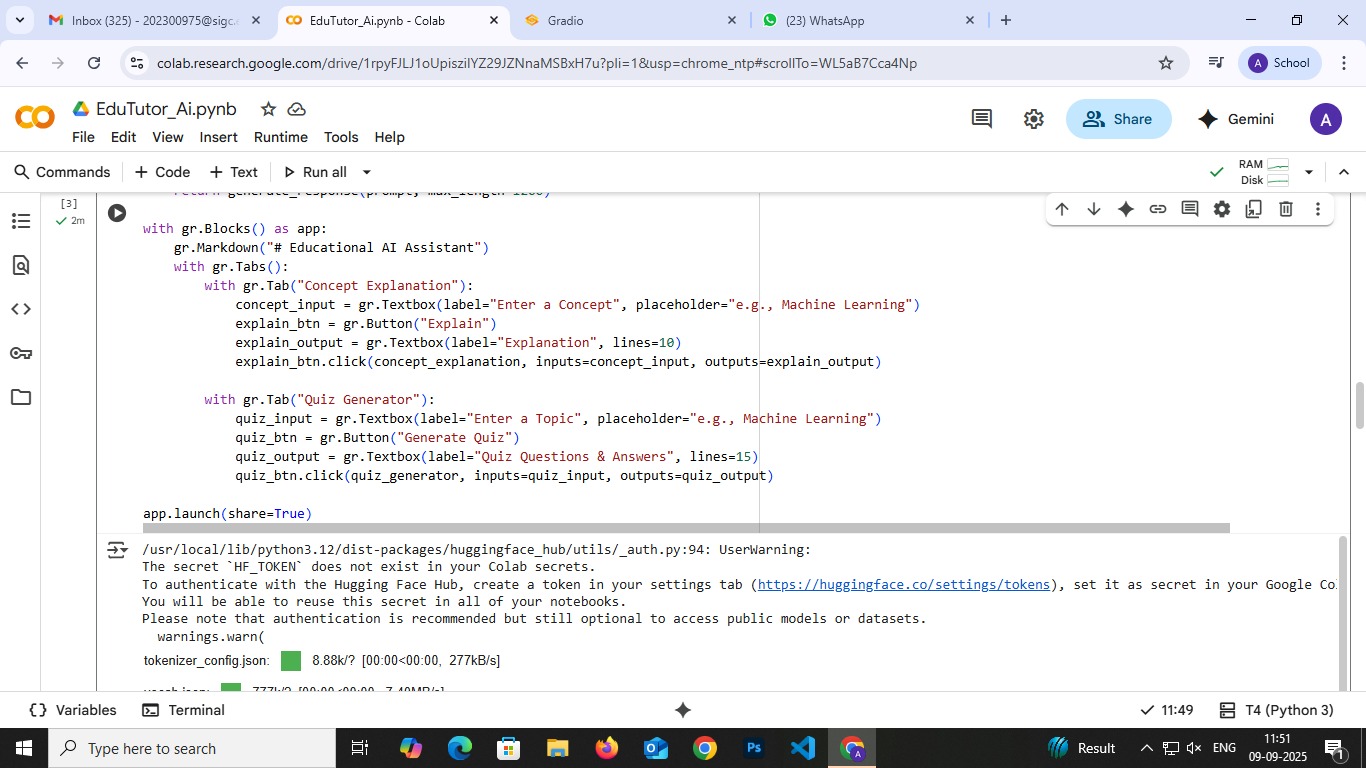
**11.Screenshots**

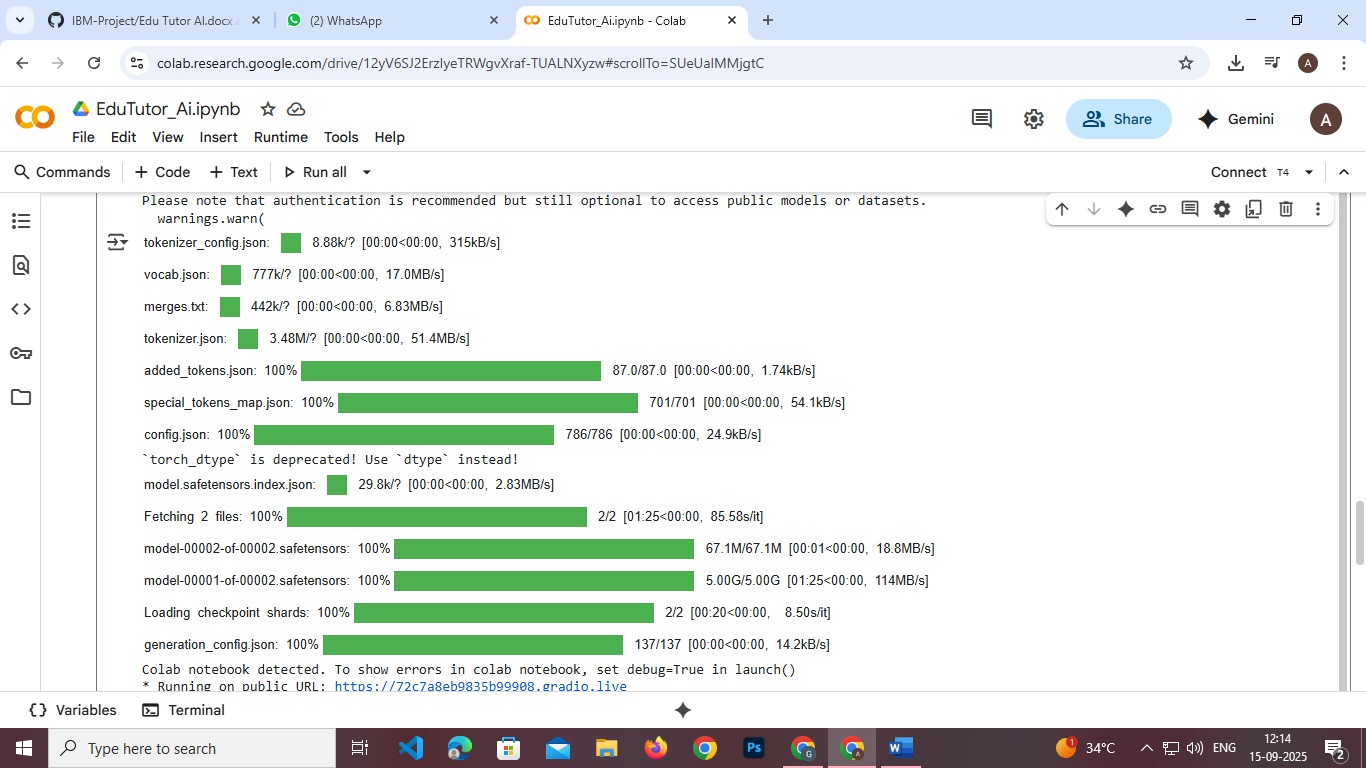
****

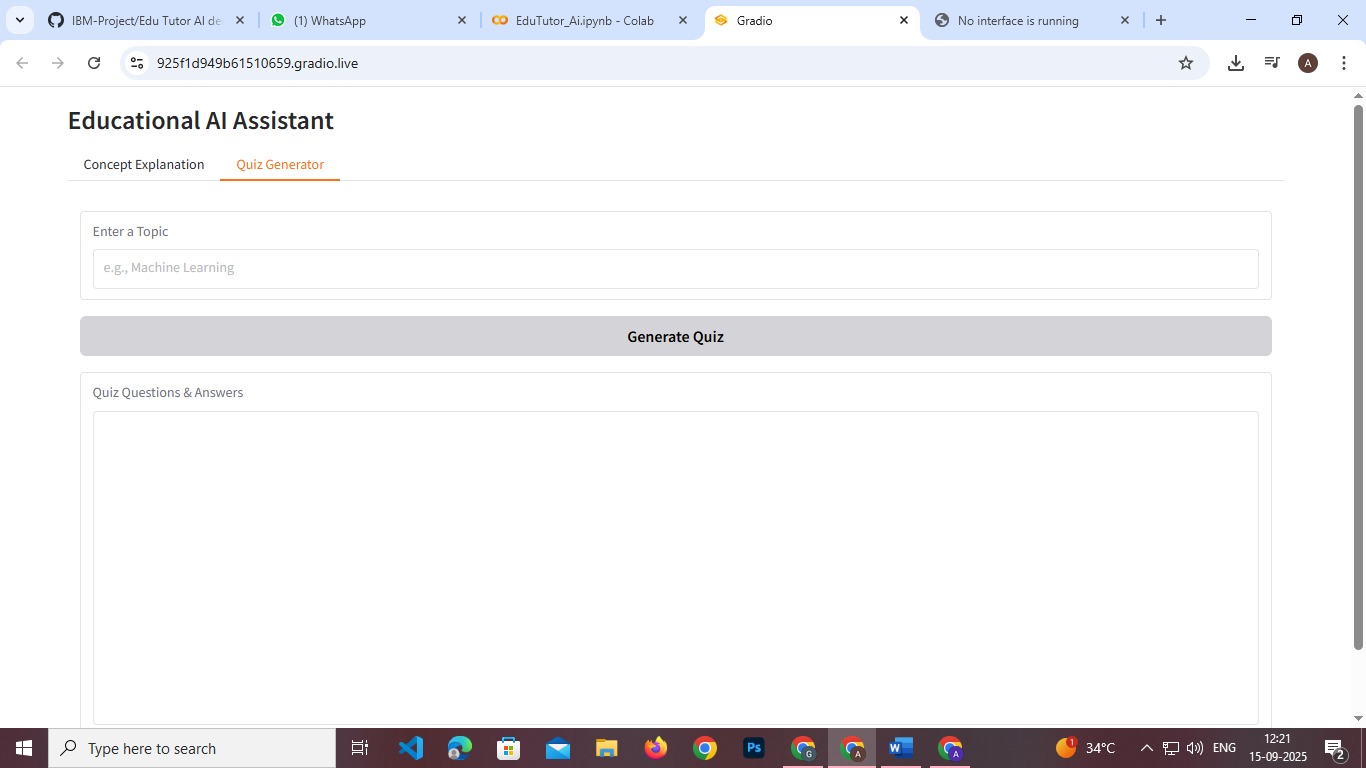


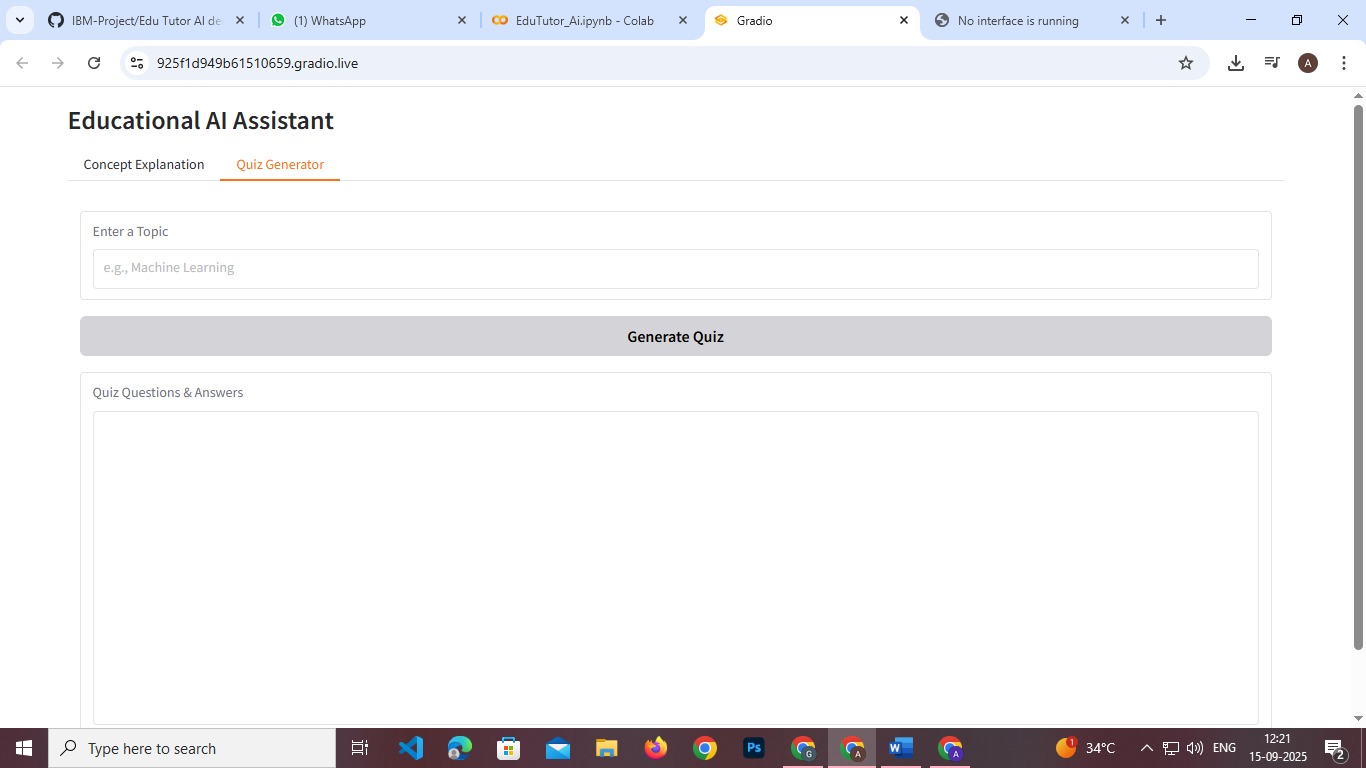


****

****

****

****

****

**12. Known Issues**

* Dependency on internet connectivity
* May require fine-tuning for complex academic queries

**13. Future Enhancements**

* Voice-based tutoring
* Integration with AR/VR for immersive learning
* Teacher dashboard for monitoring multiple students