**EDUTUTOR AI — PERSONALIZED LEARNING WITH GENERATIVE AI AND LMS INTEGRATION**

**PROJECT DOCUMENTATION**

**INTRODUCTION**

**Project Title:**

EduTutor AI — Personalized Learning with Generative AI and LMS Integration

**Team Members:**

- Jothika E

- Muthuselvi D

- Raafiya R

- Reyana H

**PROJECT OVERVIEW**

**Purpose:**

EduTutor AI is designed to offer students a personalized and interactive learning companion that breaks down difficult academic concepts into simpler explanations and provides practice quizzes to support self-assessment. Using the power of Generative AI combined with a straightforward Gradio interface, the system ensures that learning is both accessible and engaging for a wide range of users.

For students, EduTutor AI functions as a supportive guide by delivering detailed clarifications, practical real-world examples, and instant quizzes tailored to their needs. For teachers, it serves as a valuable aid by generating practice questions quickly and encouraging students to stay actively involved in the learning process.

### ****Features:****

* **Concept Explanation  
  Key Point:** AI-powered academic support  
  **Functionality:** Provides detailed explanations of any concept entered by the user, with examples for better understanding.
* **Quiz Generator  
  Key Point:** Interactive practice tool  
  **Functionality:** Generates 5 quiz questions (MCQs, True/False, Short Answer) for the given topic, along with an **Answers section.**
* **User-Friendly Interface  
  Key Point:** Simple navigation  
  **Functionality:** Gradio-based tab layout with two sections — Concept Explanation and Quiz Generator, for easy use by students.
* **Cloud-Based Execution  
  Key Point:** No installation needed  
  **Functionality:** Runs on **Google Colab**, accessible with just an internet connection.
* **Shareable Access  
  Key Point:** Easy collaboration  
  **Functionality:** app.launch(share=True) generates a public .gradio.live link so others can use the app without running the notebook.
* **Adaptability across Subjects  
  Key Point:** Multi-domain learning  
  **Functionality:** Works for science, mathematics, computer science, and other academic subjects.
* **Scalability for LMS Integration  
  Key Point:** Future-ready system  
  **Functionality:** Can be extended to track student progress, recommend learning materials, and integrate with Learning Management Systems (LMS).

## ARCHITECTURE

**Frontend (Gradio):**   
The frontend is developed using Gradio, offering an interactive and straightforward interface. It is organized into two main tabs — Concept Explanation and Quiz Generator. Users can enter any concept or topic, and the system instantly provides AI-generated explanations or quizzes. Gradio ensures a neat layout, user-friendliness, and accessibility, all without the need for additional installations.

**Backend (Google Colab):**   
Google Colab is used as the execution environment for the project. All required dependencies such as Transformers, Torch, and Gradio are installed within the notebook itself. With GPU support (T4), Colab accelerates model inference and enables cloud-based execution, allowing the project to run completely online without requiring a local setup.

**LLM Integration (IBM Granite):**   
The project incorporates the IBM Granite Instruct Model through Hugging Face Transformers. This language model generates detailed explanations and quizzes. Carefully crafted prompts are used to obtain structured responses, complete with examples and quiz-style formatting.

**Response Generation (Custom Functions):**   
Two main functions, concept\_explanation() and quiz\_generator(), are implemented to handle user queries. These functions prepare prompts based on the input, send them to the Granite model, and return the generated output, which is then presented in the interface.

**Link Sharing (Gradio Live):**   
The application makes use of Gradio’s app.launch(share=True) feature to generate a .gradio.live link. This allows the demo to be shared publicly, enabling anyone to access the application through a web link without needing to run the notebook directly. It also supports easy collaboration and remote access

## SETUP INSTRUCTIONS

### ****Prerequisites****

* Python 3.9 or later (handled automatically in Google Colab)
* Google Account (for accessing Colab)
* Internet connection (for model download and execution)
* Hugging Face access (optional — for authentication if private models are used)

### ****Installation Process****

1. **Open the Colab Notebook**
   * Download or access the file EduTutorAI.ipynb from the GitHub repository.
   * Open it in **Google Colab.**
2. **Install Dependencies**
   * Run the first code cell to install all required libraries:
   * !pip install transformers torch gradio -q
3. **Run Model Setup**
   * Execute the cell that loads the **IBM Granite model** and tokenizer.
   * The model will be downloaded (first run may take time).
4. **Run the Application**
   * Execute the Gradio interface cell.
   * The app will launch with a **public** .gradio.live **link**.
5. **Interact with the Application**
   * Use the **Concept Explanation tab** to enter any topic (e.g., Photosynthesis).
   * Use the **Quiz Generator tab** to generate 5 quiz questions with answers.

**FOLDER STRUCTURE**

EduTutor-AI/

│

├── EduTutorAI.ipynb # Main Google Colab notebook containing all code (dependencies, model loading, functions, UI)

│

├── README.md # Project description, setup instructions, and usage guide

│

├── requirements.txt # List of required Python libraries (transformers, torch, and gradio)

│

└── assets/ # (Optional) Folder for screenshots, architecture diagram, or demo images

## Running the Application

To start the project:   
➢ Open the Colab notebook EduTutorAI.ipynb in **Google Colab.**➢ Run the first cell to install required dependencies.  
➢ Run the model loading and function definition cells.  
➢ Launch the Gradio interface using app.launch(share=True).  
➢ A **public .gradio.live link** will be generated.  
➢ Open the link in a browser to interact with the application.  
➢ Use the Concept Explanation tab to type any topic and get a detailed response.  
➢ Use the Quiz Generator tab to generate 5 quiz questions with answers.  
➢ All interactions are real-time and powered by the IBM Granite model.

### Frontend (Gradio)

The frontend is built with **Gradio**, providing a simple two-tab interface for users. Navigation is straightforward with:

* Concept Explanation Tab — for detailed academic explanations.
* Quiz Generator Tab — for interactive quiz creation.

The interface is lightweight, clean, and requires no installation.

### Backend (Google Colab)

Google Colab acts as the backend environment for this project. It provides:

* Cloud-based execution with **GPU acceleration (T4).**
* Automatic library installation.
* Access to Hugging Face models.

## API Documentation

While this project runs primarily through a notebook interface, its **functional modules** can be imagined as APIs for scalability.

* **Concept Explanation API (concept\_explanation)**
  + **Input:** Topic name (e.g., “Water Cycle”)
  + **Output:** Detailed explanation with examples.
* **Quiz Generator API (quiz\_generator)**
  + **Input:** Subject/topic (e.g., “Periodic Table”)
  + **Output:** 5 quiz questions in mixed formats + answer section.
* **Gradio Live API (app.launch)**
  + **Output:** Generates a .gradio.live link for sharing with others.

All these components are modular, meaning they could be extended into **real API endpoints** if integrated with frameworks like **FastAPI** in the future.

## AUTHENTICATION

This version of **EduTutor AI** runs in an **open Google Colab environment** for demonstration purposes. The Gradio interface with share=True generates a public link that can be accessed by anyone without login requirements.

For secure deployments beyond the demo stage, the following authentication mechanisms can be integrated:

* **Token-Based Authentication (API Keys / JWT):**   
  Restricts access so only authorized users can query the AI.
* **OAuth2 with Google or IBM Cloud Credentials:**   
  Enables verified sign-in for students and teachers.
* **Role-Based Access Control:**   
  Different access levels for learners, faculty, and administrators.
* **User Sessions and History Tracking (Planned):**   
  Saving each student’s learning history, attempted quizzes, and progress for future personalization.

**USER INTERFACE**

The application features a **minimalist and functional interface,** designed for **accessibility and ease of use,** especially for non-technical users. Key elements include:

* **Sidebar Navigation** – Provides quick access to all major modules and features.
* **KPI Visualizations** – Summary cards display key performance indicators at a glance.
* **Tabbed Layouts** – Separate tabs for chat, eco tips, and forecasting ensure organized access to different functionalities.
* **Real-time Form Handling** – Interactive forms update and validate inputs instantly for smoother user interaction.

The design prioritizes **clarity, speed, and user guidance**, with help texts and intuitive flows.

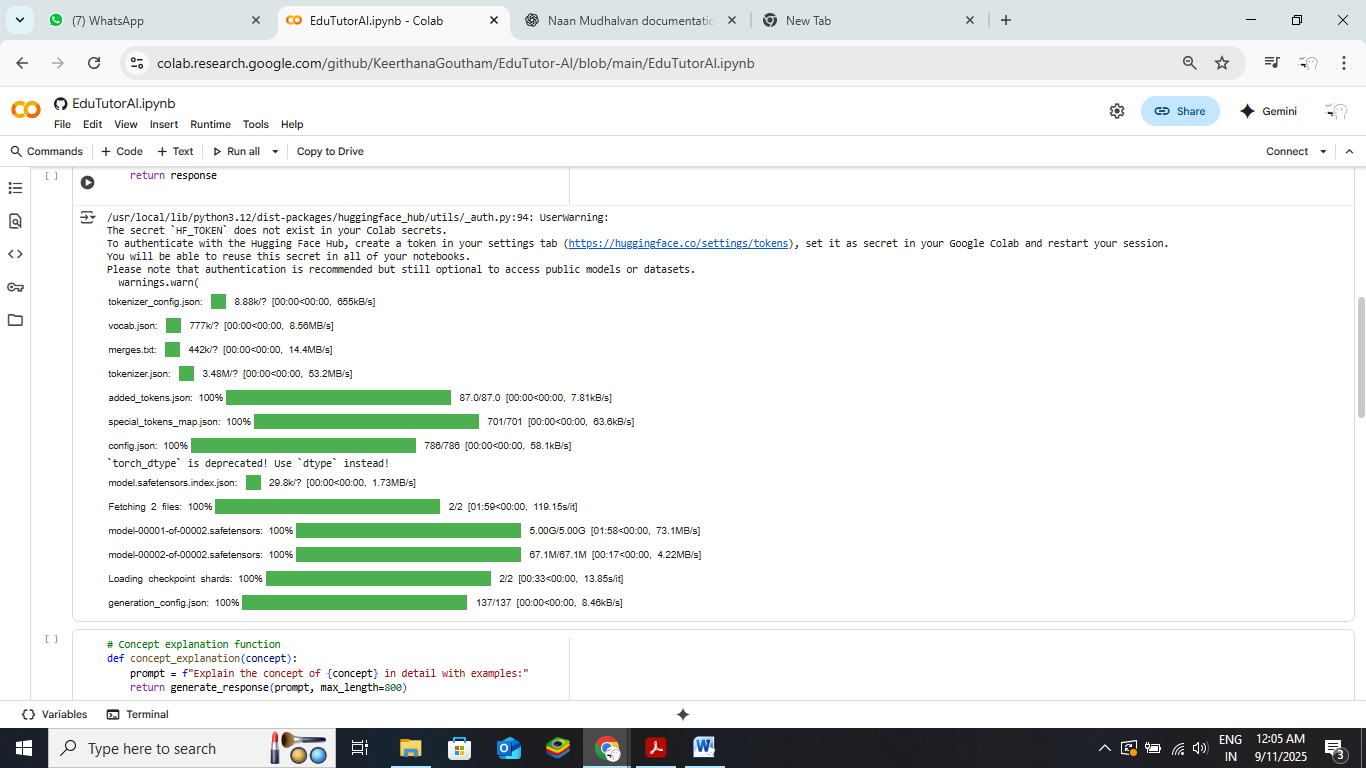
**TESTING**

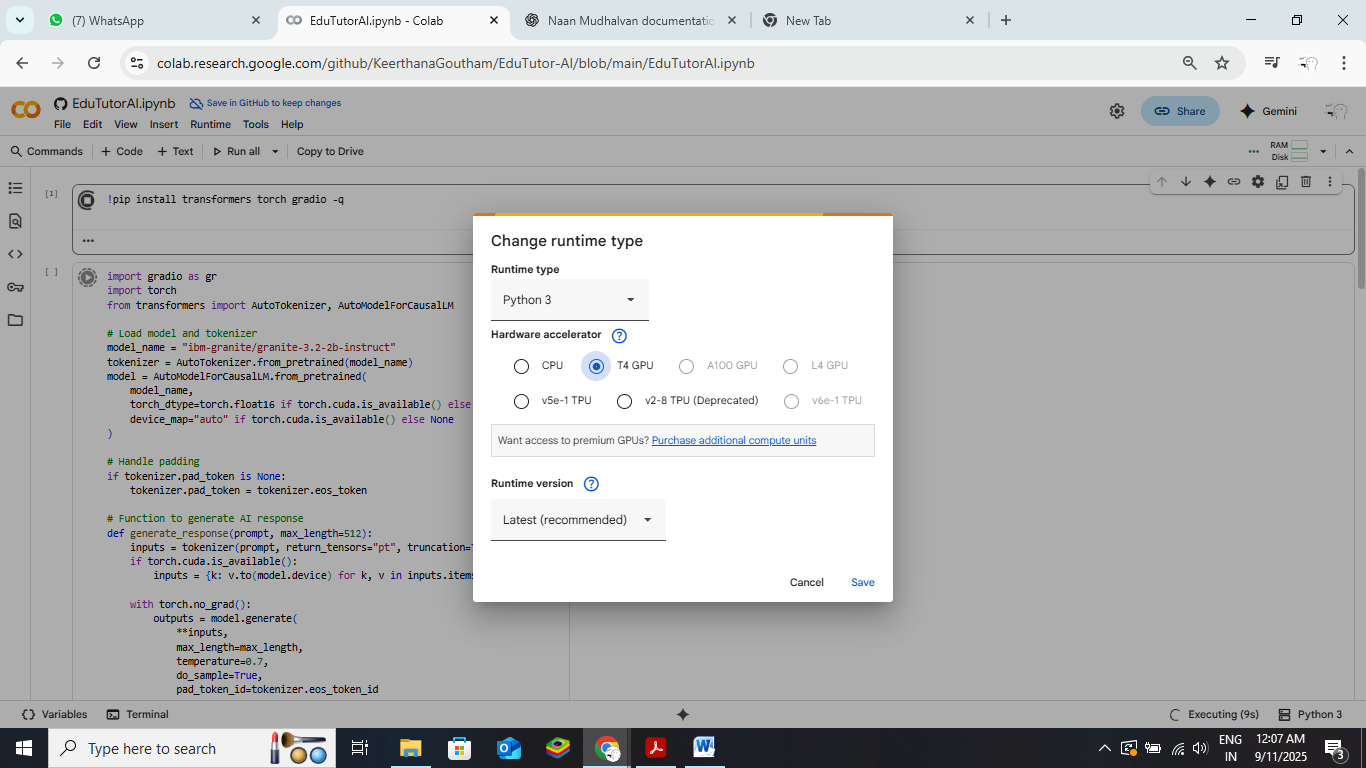
Testing was carried out in **multiple phases** to ensure the application’s reliability and correctness:

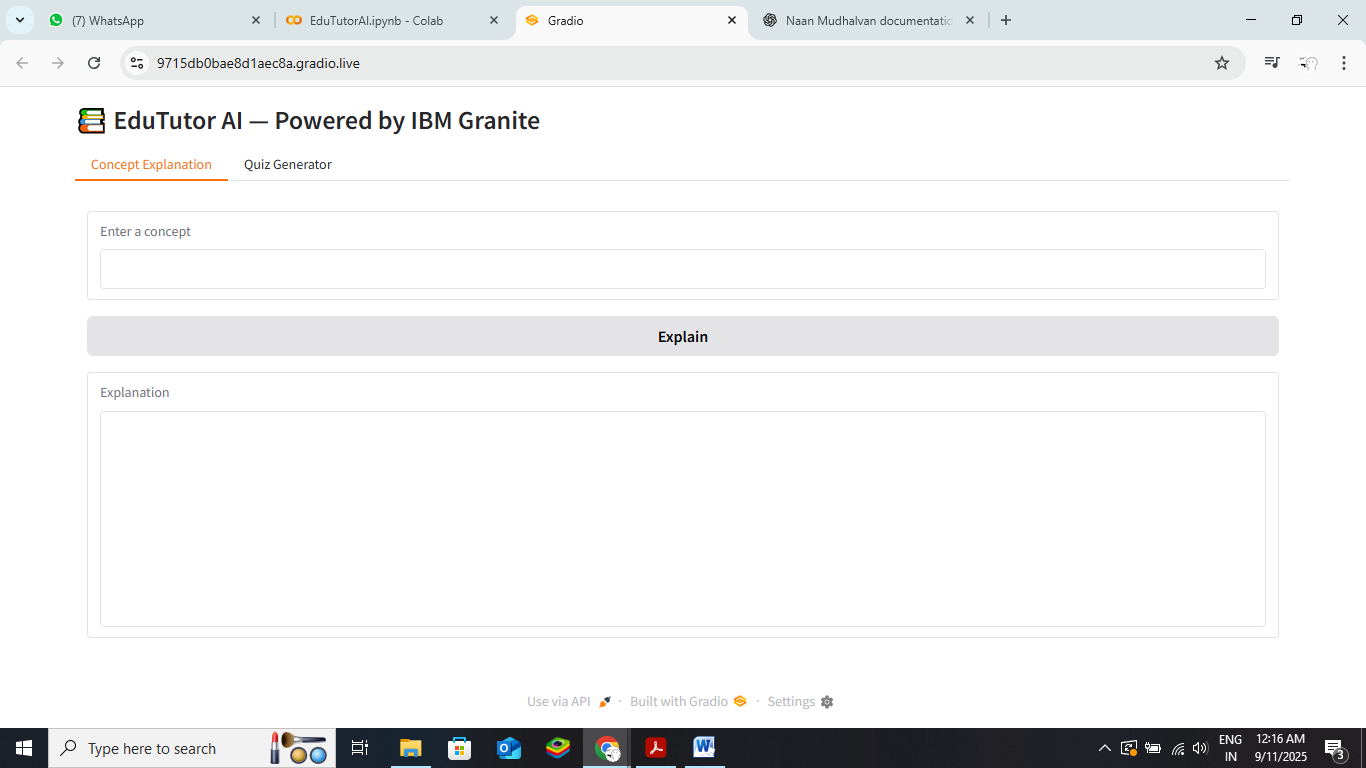
* **Unit Testing** – Focused on prompt engineering functions and utility scripts.
* **API Testing** – Conducted via Swagger UI, Postman, and custom test scripts.
* **Manual Testing** – Checked file uploads, chat responses, and output consistency.
* **Edge Case Handling** – Tested malformed inputs, large files, and invalid API keys.

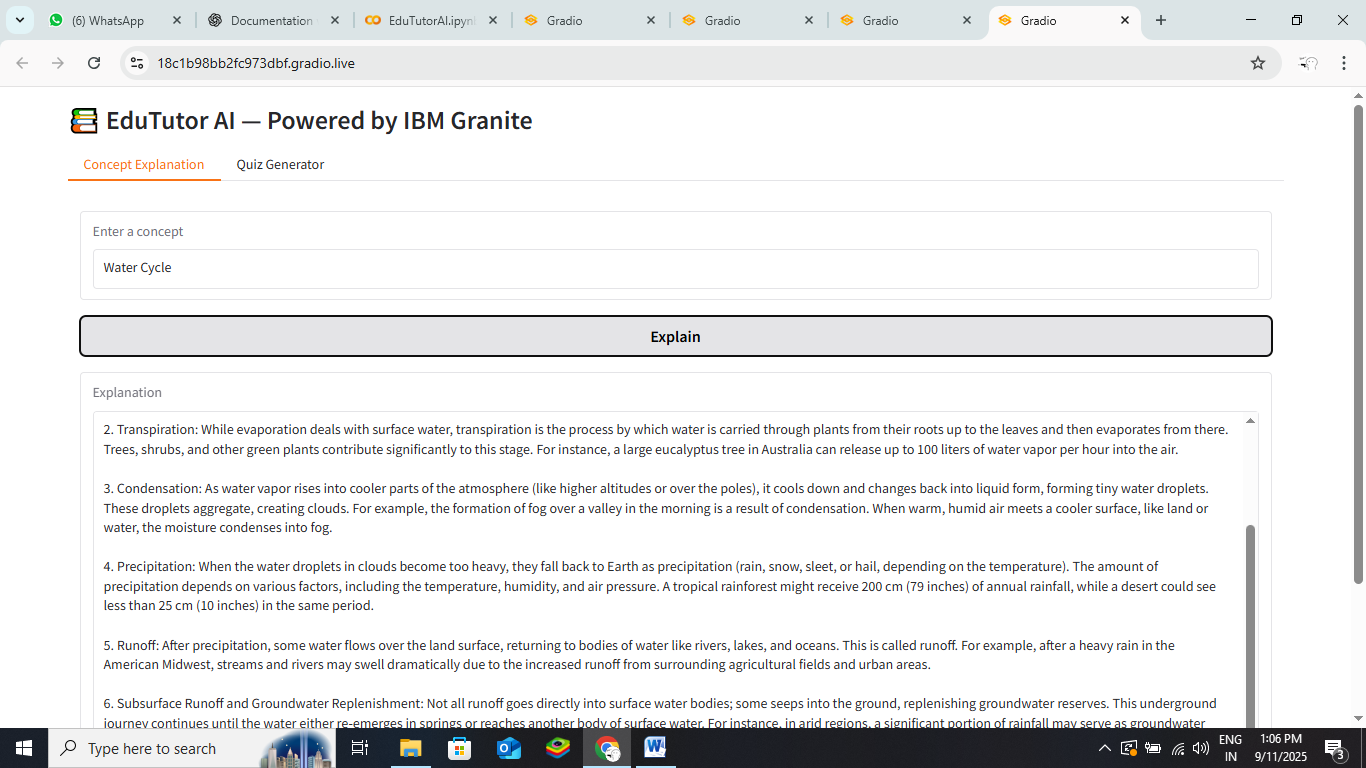
Each function was **validated** to ensure reliable performance in both **offline** and **API-connected** modes.

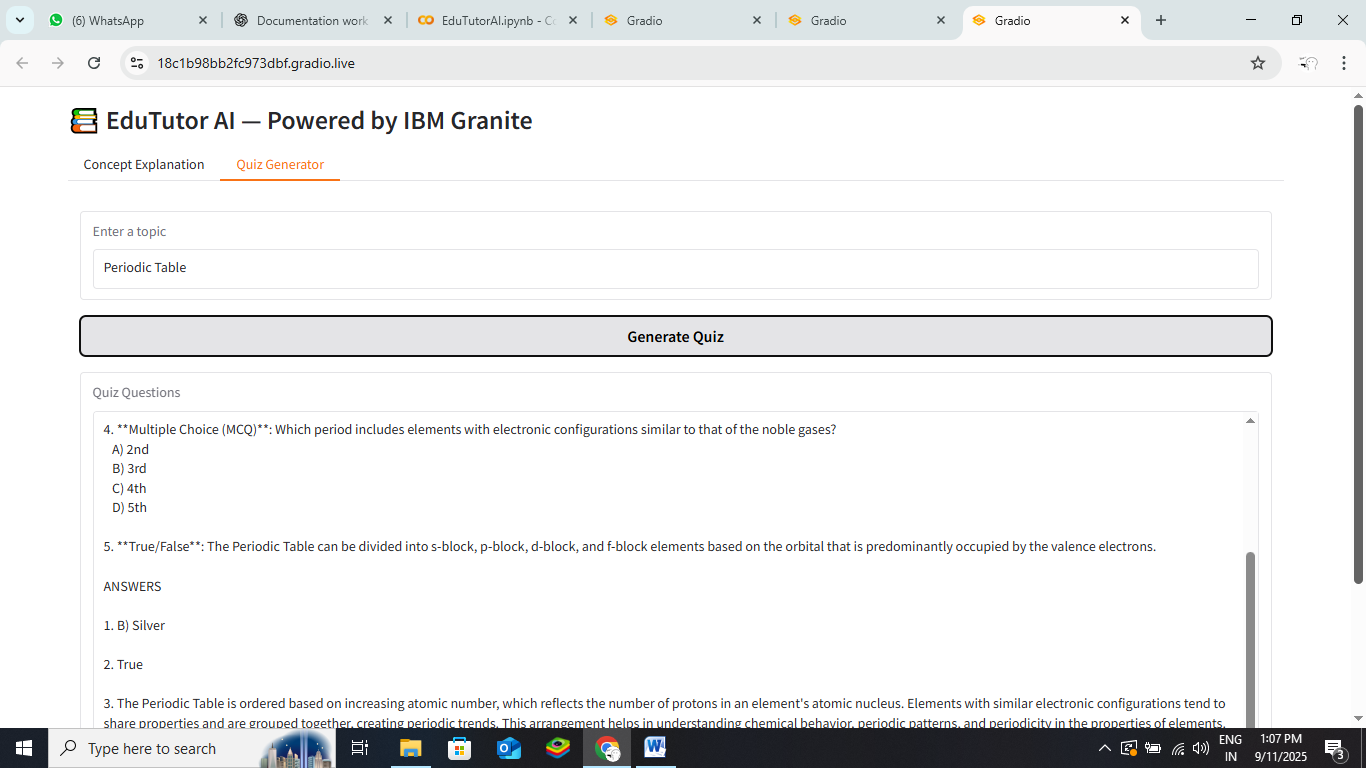
**SCREENSHOTS**











**KNOWN ISSUES**

During testing, a few minor issues were identified:

* Uploading large files can sometimes result in slower processing times.
* Chat responses may occasionally experience slight delays due to API connectivity.
* Certain malformed inputs may still need improved handling in future updates.

Overall, no critical functionality problems were detected, and the application performs reliably across its core features

**FUTURE ENHANCEMENTS**

Potential improvements for future versions include:

* **PDF or Report Generation** – Enable users to download chat histories or analytics summaries.
* **Expanded Eco Tips Library** – Include more dynamic suggestions based on user preferences.
* **Enhanced Error Handling** – Improve handling of edge cases and invalid inputs.
* **User Customization Options** – Allow theme changes, tab arrangements, and interface personalization.