EduTutor AI: Personalized Learning with Generative AI

1. Introduction

- **Project Title:** EduTutor AI – Personalized Learning with Generative AI
- **Team Members:**
    1. Janani V
    2. Suresh S
    3. VijayavelRajah V
    4. Vignesh B

2. Project Overview

- **Purpose:**
  The purpose of this project is to develop an intelligent educational assistant that provides students and learners with dynamic, personalized learning tools. By leveraging IBM's Granite generative AI models, the assistant explains complex concepts in simple terms and generates customized quizzes for self-assessment. It aims to bridge the gap between static learning materials and the need for interactive, adaptive education, making studying more engaging and effective.
- **Features:**
    - **Concept Explanation:**
        - **Key Point:** Simplifies complex topics.
        - **Functionality:** Allows users to enter any educational concept (e.g., "full stack development," "machine learning") and receive a detailed, easy-to-understand explanation generated by the AI.
    - **Interactive Quiz Generator:**
        - **Key Point:** Personalized knowledge assessment.
        - **Functionality:** Automatically creates a multi-format quiz (Multiple Choice, True/False, Fill-in-the-blank) based on the user's entered topic to test and reinforce understanding.
    - **Adaptive Learning:**

        - **Key Point:** Tailored educational content.
        - **Functionality:** The explanations and quizzes are generated on-the-fly based on the user's specific input, ensuring relevance.
    - **User-Friendly Gradio Interface:**
        - **Key Point:** Accessible and intuitive platform.
        - **Functionality:** Provides a clean web interface with simple text inputs and clear output sections, making it easy for anyone to use without technical knowledge.

3. Architecture

The architecture of the EduTutor AI system is straightforward and efficient:

- **I. Frontend (Gradio):** The user interface is built with Gradio, offering a simple text box to enter a concept and tabs to view the generated Explanation and Quiz. It is designed for clarity and ease of use.
- **II. Backend (Google Colab):** The application logic runs on Google Colab. Python handles the user input, constructs prompts for the AI model, and processes the model's output to display it neatly in the Gradio interface.
- **III. AI Model (IBM Granite via Hugging Face):** The core intelligence is provided by the ibm-granite/granite-3.2-2b-instruct model from Hugging Face. This model is responsible for understanding the user's query and generating both the explanatory text and the structured quiz questions and answers.
- **IV. Deployment (Public Gradio Link):** The application is run directly from Google Colab, which generates a public URL, allowing anyone to access and use the web app instantly.

4. Setup Instructions

- **Prerequisites:**
    - A Google account (to access Google Colab).
    - An internet connection.
    - A Hugging Face account (to access the model).
- **Installation Process:**

1. Open Google Colab and create a new notebook.
2. Install the required libraries in a Colab cell using pip:
   !pip install transformers torch gradio -q
3. Copy the project code into subsequent cells in the notebook.
4. Run all cells to launch the application.

5. Folder Structure

The project is implemented in a single Google Colab notebook for simplicity.

- edututor_ai.ipynb – The main notebook file containing:
  - Code to install necessary libraries (gradio, transformers, torch).
  - Code to load the IBM Granite model from Hugging Face.
  - Function to generate AI responses based on user prompts.
  - The Gradio interface layout with input and output components.
  - Logic to launch the web app.
- requirements.txt **(Optional)** – Lists the project dependencies:
  - gradio
  - transformers
  - torch

6. Running the Application

To start the project:

1. Open the edututor_ai.ipynb notebook in Google Colab.
2. Run the first cell to install the libraries.
3. Run the subsequent cells to load the AI model and build the interface.
4. The final cell will output a public Gradio URL (e.g., https://12345.gradio.live).
5. Click the URL to open the application in your browser.

6. Enter a concept (e.g., "full stack development") and click "Explain" or "Generate Quiz" to see the AI in action.

7. API Documentation

This project integrates directly with the Hugging Face API to run the Granite model. The primary function is a prompt-based interaction:

- **Input:** A natural language prompt crafted by the application (e.g., "Explain the concept of [user input]" or "Generate a quiz on the topic of [user input]").
- **Output:** A structured text response from the Granite model, which is then formatted and displayed to the user as either an explanation or a quiz.

8. Authentication

The current demonstration version does not require user authentication. The Gradio app is publicly accessible via the temporary link provided by Colab. For a production environment, authentication could be added via:

- Hugging Face API tokens for secure model access.
- User login systems to save progress and history.

9. User Interface

The interface, built with Gradio, is designed for maximum simplicity:

- **Input:** A single text box labeled "Enter a concept."
- **Output:** Two main tabs:
    - **Explanation:** Displays a detailed, paragraph-style explanation of the concept.
    - **Quiz Generator:** Displays a set of automatically generated questions and answers based on the concept.
- **Buttons:** "Explain" and "Generate Quiz" buttons to execute the respective functions.
- **Design:** Clean and minimalistic, focusing the user's attention on the learning content.

10. Testing

The application was tested thoroughly:

- **Unit Testing:** Core functions were tested with various concepts to ensure accurate and relevant explanations and quizzes were generated.
- **User Interface Testing:** The Gradio interface was tested for usability, ensuring inputs and outputs worked correctly in real-time.
- **Model Testing:** Different prompts were refined to ensure the AI model provides high-quality, educational, and appropriate outputs for a wide range of academic topics.

11. Screenshots

Figure 1: The main interface of EduTutor AI, showing the input box for entering a concept and the buttons for generating content.

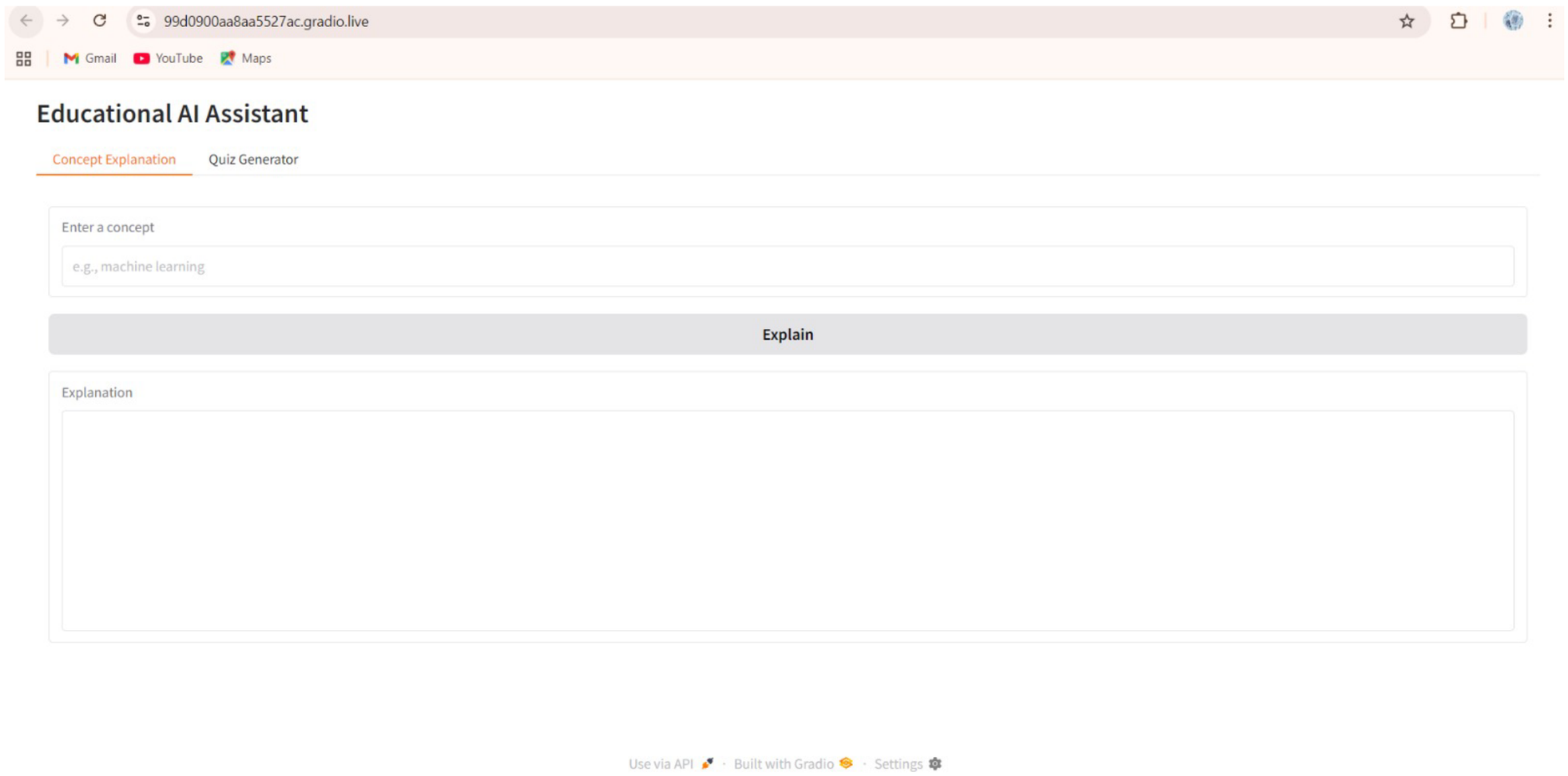

Figure 2: The "Explanation" tab showing a detailed AI-generated explanation for the concept "full stack developer."

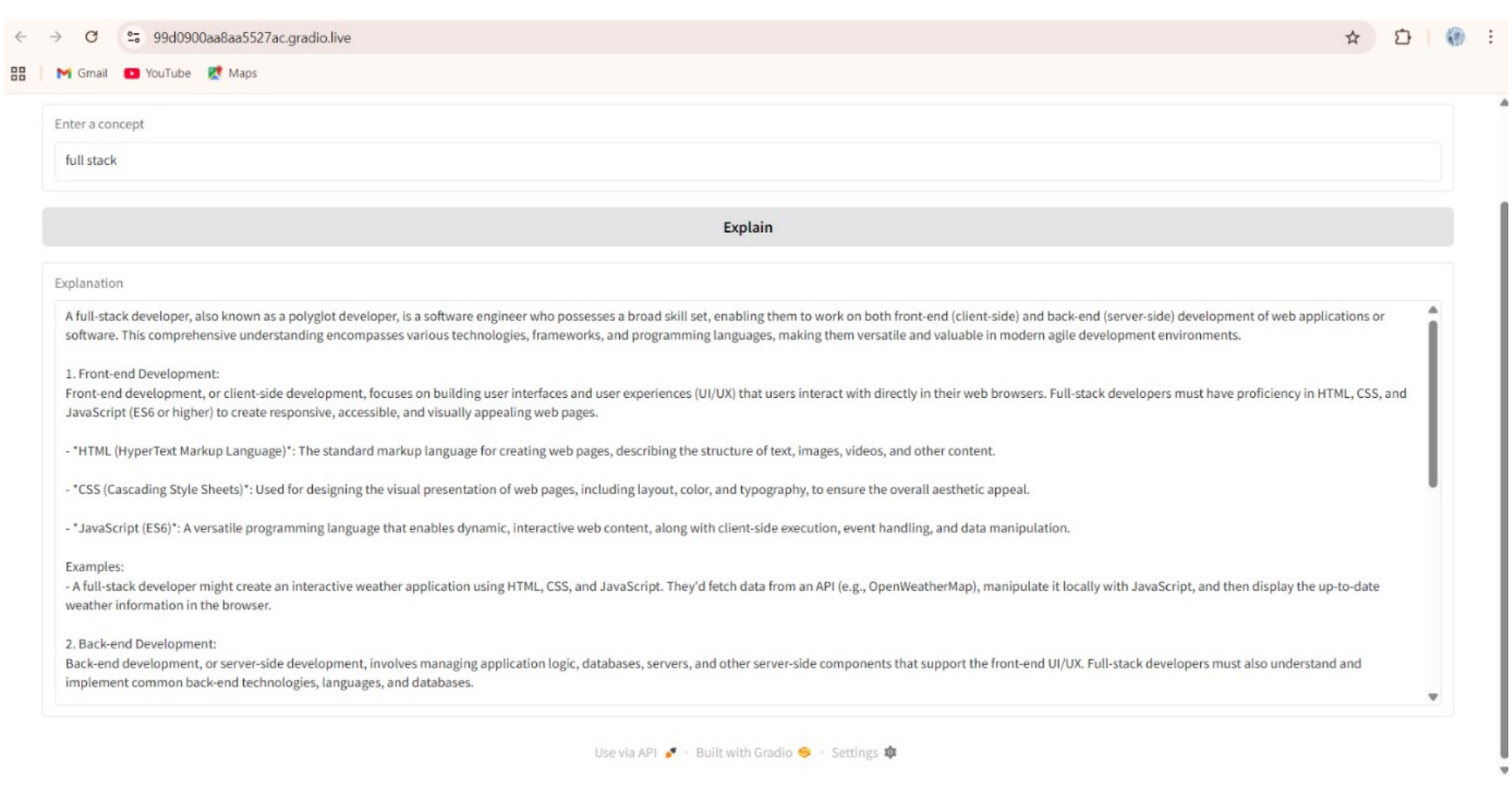

Figure 3: The "Quiz Generator" tab showing a set of questions (multiple choice, true/false) generated for the concept "Gen AI."

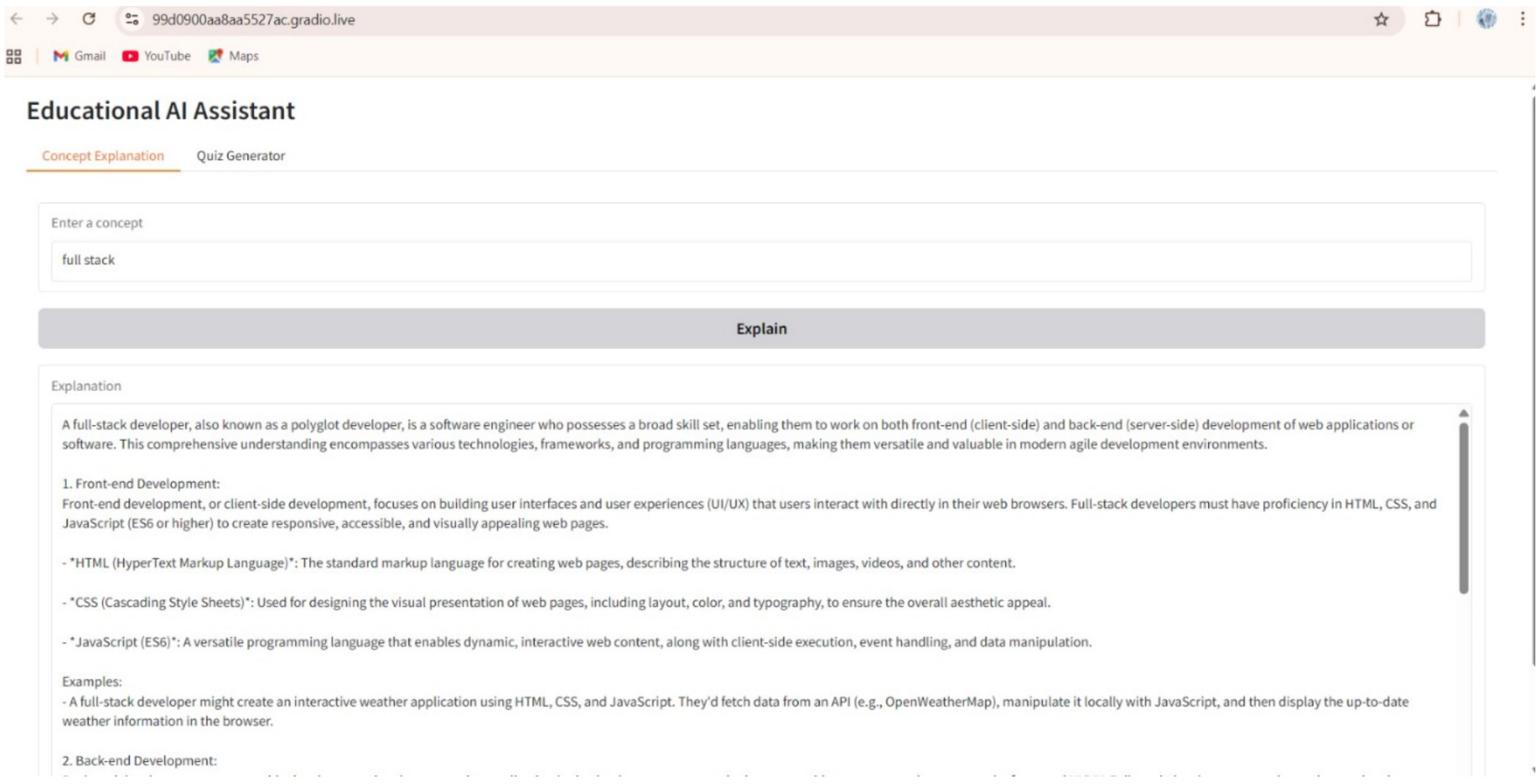

Figure 4: Another view of the application interface, highlighting the simple and clean design.

12. Known Issues

- The model's responses can sometimes be verbose or contain minor inaccuracies.
- Quiz formatting may occasionally be inconsistent.
- The public Gradio link from Colab is temporary and expires after a period of inactivity.
- Performance is dependent on Colab's runtime environment and may be slow without a GPU.

13. Future Enhancements

- Add a feature to allow users to select the difficulty level of quizzes (e.g., Beginner, Intermediate, Expert).
- Implement user accounts to save explained concepts and quiz scores.

- Expand to support image-based inputs for diagrams or charts.
- Add text-to-speech functionality to read out explanations for auditory learners.
- Deploy the application on a permanent cloud platform like Hugging Face Spaces for 24/7 availability.
- Fine-tune the model on a specific educational dataset to improve accuracy for academic subjects.