Edu Tutor AI: Personalized Learning

Generative AI with IBM

Project Documentation

1.Introduction:

Project title : Edu Tutor AI: Personalized Learning

• Team member : Charulatha.S

• Team member : Harini.K

• Team member : Jeeivasudha.M.V

• Team member : Karunya adhirai.W

2.project overview:

• Purpose :

The purpose of EduTutor AI is to transform the learning experience by providing personalized, adaptive, and interactive educational support for students. By leveraging AI and real-time analytics, EduTutor AI identifies individual learning needs, strengths, and areas for improvement, delivering customized lessons, practice exercises, and feedback. For educators, it acts as an intelligent assistant—offering insights into student progress, suggesting targeted interventions, and simplifying administrative tasks. Ultimately, EduTutor AI bridges technology and education to create a more engaging, effective, and student-centered learning environment, fostering academic growth and lifelong learning.

• Features:

Concept Explanation

- **Key Point:** Detailed understanding
- **Functionality:** Explains concepts in detail with simple examples to help students understand difficult topics.

Quiz Generator

- **Key Point:** Practice and assessment
- **Functionality:** Generates quizzes with multiple types of questions (MCQ, True/False, Short Answer) along with an answers section.

Text Summarizer

- **Key Point:** Efficient learning
- **Functionality:** Summarizes long text into concise and easy-to-understand content.

Flashcard Creator

- **Key Point:** Memorization support
- **Functionality:** Creates flashcards in question-and-answer format to help students memorize key points.

Conversational Interface

- **Key Point:** Interactive learning
- **Functionality:** Allows students to input text naturally and receive intelligent responses from the AI assistant.

User-Friendly Gradio Interface

- **Key Point:** Easy access and interaction
- **Functionality:** Provides a simple and intuitive interface with separate tabs for each feature—Concept Explanation, Quiz Generator, Summary Generator, and Flashcard Creator.

• **User Authentication:**

1. **Signup System**
    - New users can register with a **username, password, and role** (Student or Faculty).
    - Ensures **unique usernames** for secure access.
2. **Login System**
    - Validates user credentials.
    - Displays **role-specific welcome messages**.

3. Architecture:

Frontend (Gradio UI):
The frontend is built using Gradio, providing an interactive web interface with multiple tabs for Concept Explanation, Quiz Generator, Text Summarizer, and Flashcard Creator. Each tab is designed to handle a specific learning feature, allowing students to interact with the AI assistant in a user-friendly and modular manner.

Backend (Transformers Model):
The backend leverages Hugging Face Transformers to load the IBM Granite LLM model (ibm-granite/granite-3.2-2b-instruct). The model processes natural language prompts, generates explanations, quizzes, summaries, and flashcards, and returns the results to the frontend.

Text Processing and Generation:
The system tokenizes input prompts, handles device placement (CPU/GPU), and generates responses with configurable parameters such as max_new_tokens, temperature, and top_p. Responses are cleaned and formatted before being displayed to the user.

Feature Modules:
Each feature—Concept Explanation, Quiz Generator, Text Summarizer, and Flashcard Creator—is implemented as a separate

function that constructs a tailored prompt, sends it to the LLM, and returns the generated output.

Device and Performance Optimization:
The model automatically uses GPU if available (torch.cuda.is_available()) to speed up response generation and uses mixed precision (float16) for efficient computation.

4. Setup Instructions:

Prerequisites:

- Python 3.9 or later
- pip and virtual environment tools
- Internet access to install packages and run the AI model

Installation Process:

- Clone the repository
- Install dependencies using:
  pip install transformers torch gradio
- pip install -r requirements.txt

(or pip install transformers torch gradio if requirements.txt is not used)

- Run the Jupyter Notebook (EduTutor_AI.ipynb) or Python script
- Launch the Gradio interface using the notebook or script
- Interact with the modules: Concept Explanation, Quiz Generator, Text Summarizer, and Flashcard Creator

5. Folder Structure:

notebooks/ – Contains the Jupyter Notebook (EduTutor_AI.ipynb) with all feature implementations and Gradio interface setup.

models/ – Stores any pretrained or downloaded model weights (e.g., IBM Granite LLM tokenizer and model).

features/ – Contains Python scripts for each learning module:

- concept_explanation.py – Handles generating detailed explanations of concepts.
- *quiz_generator.py* – Generates quizzes with multiple question types.
- *text_summarizer.py* – Summarizes long texts into concise content.
- *flashcard_creator.py* – Creates flashcards in question-and-answer format.

utils/ – Utility scripts for shared functions, such as text processing and tokenization helpers.

requirements.txt – Lists all Python dependencies required to run EduTutor AI.

EduTutor_AI.ipynb – Main entry notebook to launch the Gradio interface and interact with all modules.

6. Running the Application:

To start the project:

1. **Launch the Notebook or Script**

- Open and run the Jupyter Notebook EduTutor_AI_with_Login.ipynb or the Python script containing the Gradio interface.
- The Gradio web interface will launch automatically.

2. **Login / Signup**

- New users can Sign Up by providing a username, password, and selecting a role (Student or Faculty).
- Existing users can Log In with their credentials.
- Successful login grants access to the educational modules and displays a role-specific welcome message.

3. **Navigating the Educational Modules**

- After login, use the Gradio tabs to access the following features:
- Concept Explanation – Enter a concept to receive detailed explanations with examples.
- Quiz Generator – Enter a topic to generate quizzes with MCQs, True/False, and Short Answer questions, along with answers.
- Text Summarizer – Paste long text to receive concise summaries in simple language.
- Flashcard Creator – Enter a topic to generate Q&A flashcards for memorization.

4. **Real-Time Interaction**

- All inputs are processed by the IBM Granite LLM backend, generating responses dynamically.
- Outputs are displayed instantly in the respective tabs for easy access and interaction.

Frontend (Gradio UI)

- Provides a modular tabbed layout for login/signup and each learning feature.
- Includes input text boxes, buttons, and output areas for user-friendly interaction.

Backend (Hugging Face Transformers Model)

- Loads and runs the IBM Granite LLM (ibm-granite/granite-3.2-2b-instruct) for natural language understanding and generation.
- Handles tokenization, response generation, and device optimization (CPU/GPU).
- Processes user inputs from the frontend and returns educational content in real-time.

7.API Documentation:

The backend of EduTutor AI exposes the following APIs for educational functionalities:

- **POST /concept-explanation** – Accepts a concept as input and returns a detailed explanation with simple examples.
- **POST /quiz-generator** – Accepts a topic and generates a set of quiz questions (MCQ, True/False, Short Answer) along with answers.
- **POST /text-summarizer** – Accepts long text and returns a concise summary in simple language.
- **POST /flashcard-creator** – Accepts a topic and generates flashcards in question-and-answer format for memorization.

Each endpoint is designed to interact with the backend LLM model and return responses in real-time.

8. Authentication:

All endpoints in EduTutor AI are currently accessible in an open environment for demonstration and testing purposes.

- Each feature (Concept Explanation, Quiz Generator, Text Summarizer, Flashcard Creator) can be tested directly via the Gradio interface or backend API calls.

- No authentication mechanisms (such as API keys or user login) are implemented in this version.
- Planned enhancements may include:
    - Token-based authentication (JWT or API keys) for secure access
    - Role-based access control for different types of users (students, educators, administrators)
    - User session management and history tracking for personalized learning

9.User Interface:

The interface of EduTutor AI is minimalist, intuitive, and designed for ease of use by students and educators. Key elements include:

- **Tabbed Layouts** – Separate tabs for Concept Explanation, Quiz Generator, Text Summarizer, and Flashcard Creator for organized navigation.
- **Input Textboxes and Buttons** – Clear input fields for entering concepts, topics, or text, and buttons to trigger AI-generated outputs.
- **Output Display Areas** – Text areas to display explanations, quizzes, summaries, and flashcards in real-time.
- **Real-Time Interaction** – All modules process inputs and return outputs instantly via the Gradio interface.
- **User-Friendly Design** – Focuses on clarity and accessibility, with simple layouts and straightforward interaction flows.

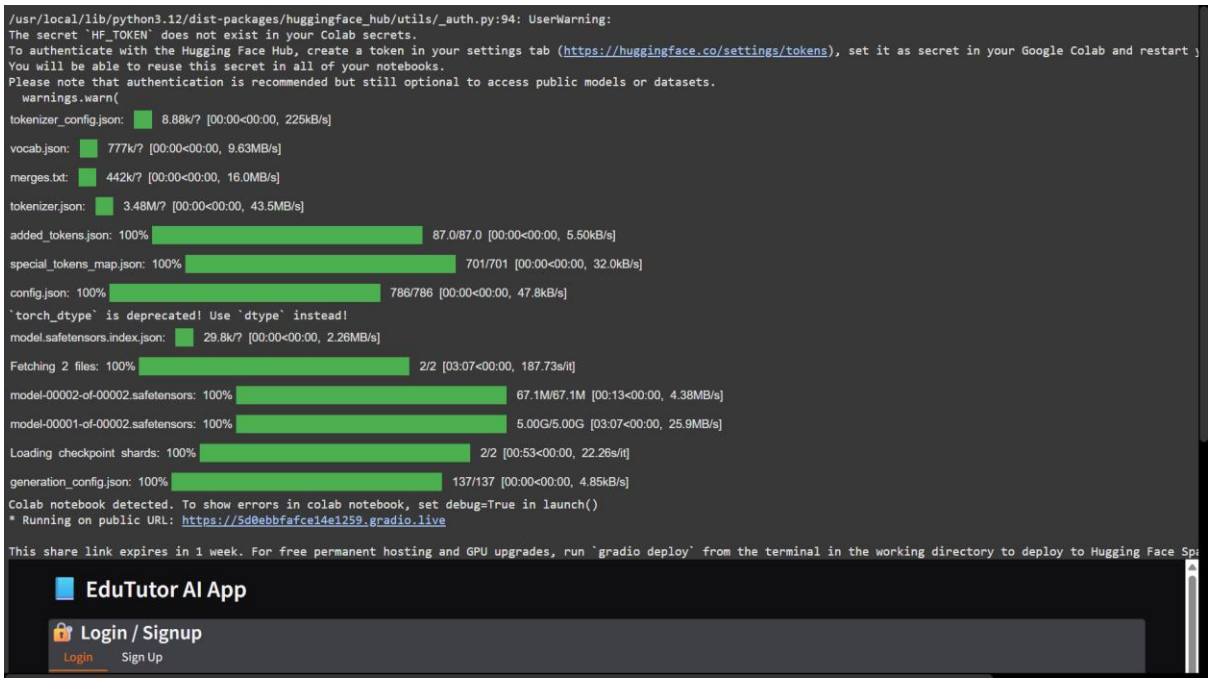
10.Testing:

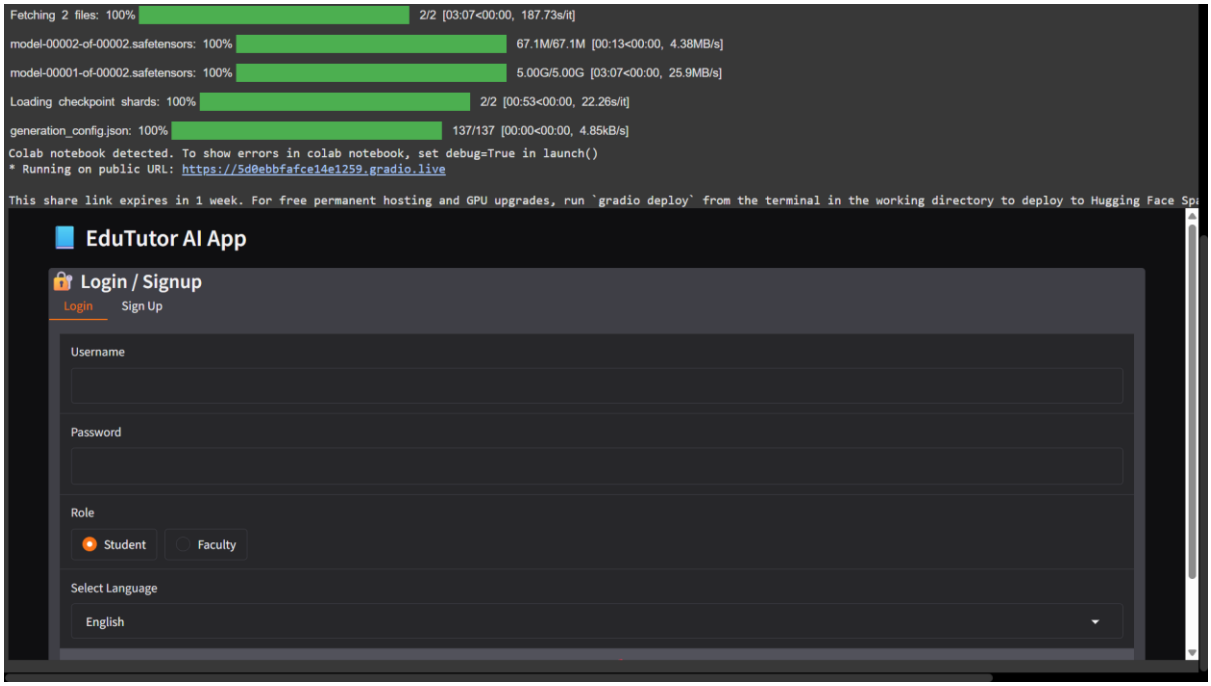
Testing of EduTutor AI was conducted in multiple phases:

- **Unit Testing:** For each feature module (Concept Explanation, Quiz Generator, Text Summarizer, Flashcard Creator) to validate prompt construction and response generation.
- **API Testing:** Using direct function calls in the notebook and scripts to ensure each module returns correct outputs.
- **Manual Testing:** For interactions via the Gradio interface, including input submission and real-time display of explanations, quizzes, summaries, and flashcards.
- **Edge Case Handling:** Tested for unusual or large inputs, empty fields, and invalid text to ensure consistent and reliable outputs.

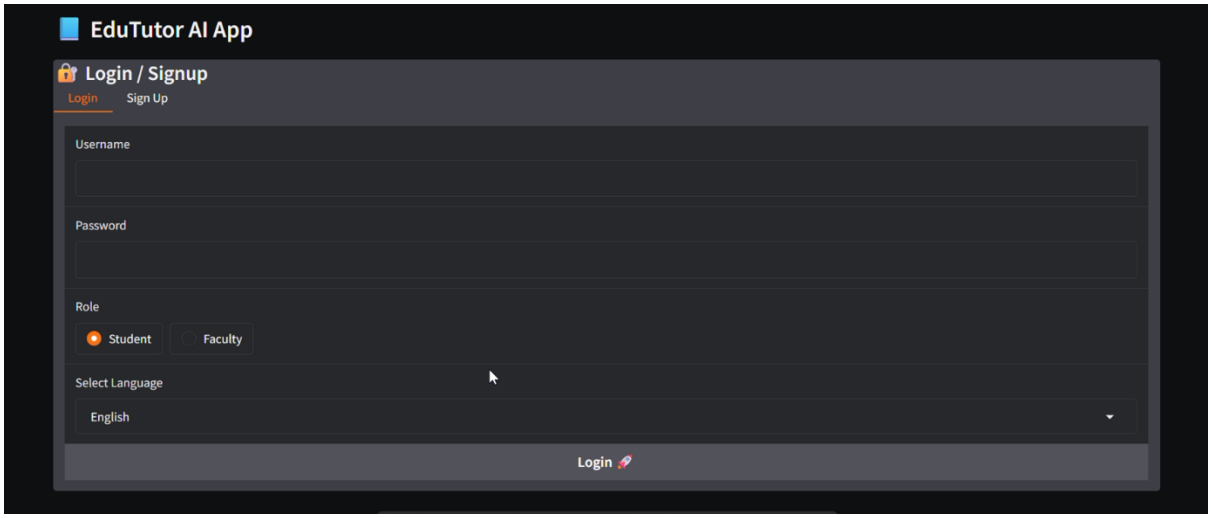
Each module was validated to ensure reliability and accuracy during both interactive and automated usage.

11.Screenshots:

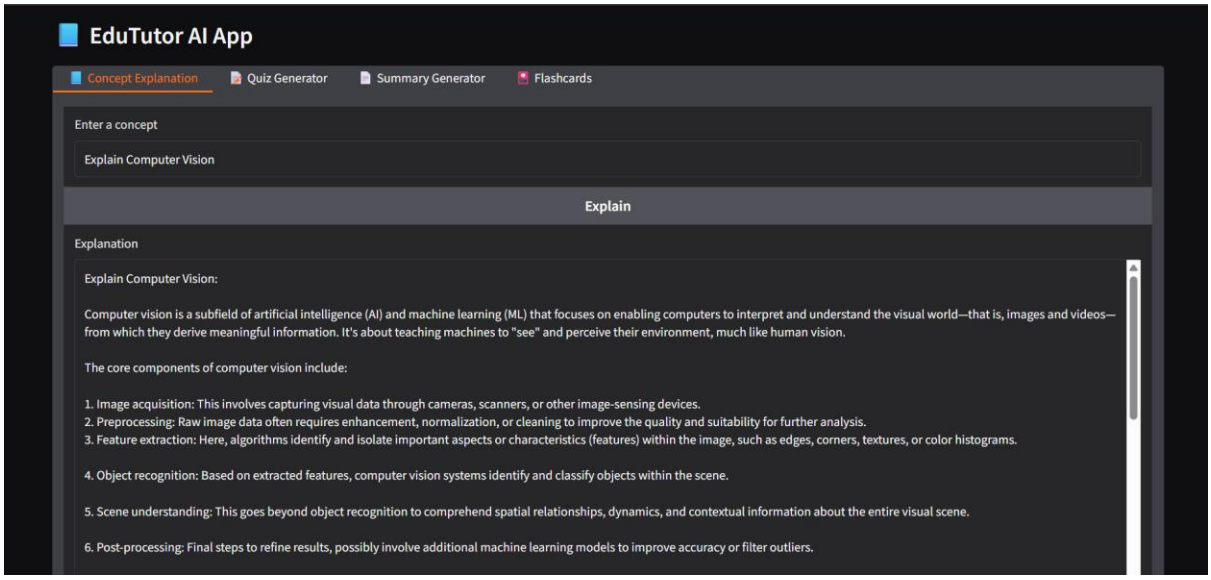




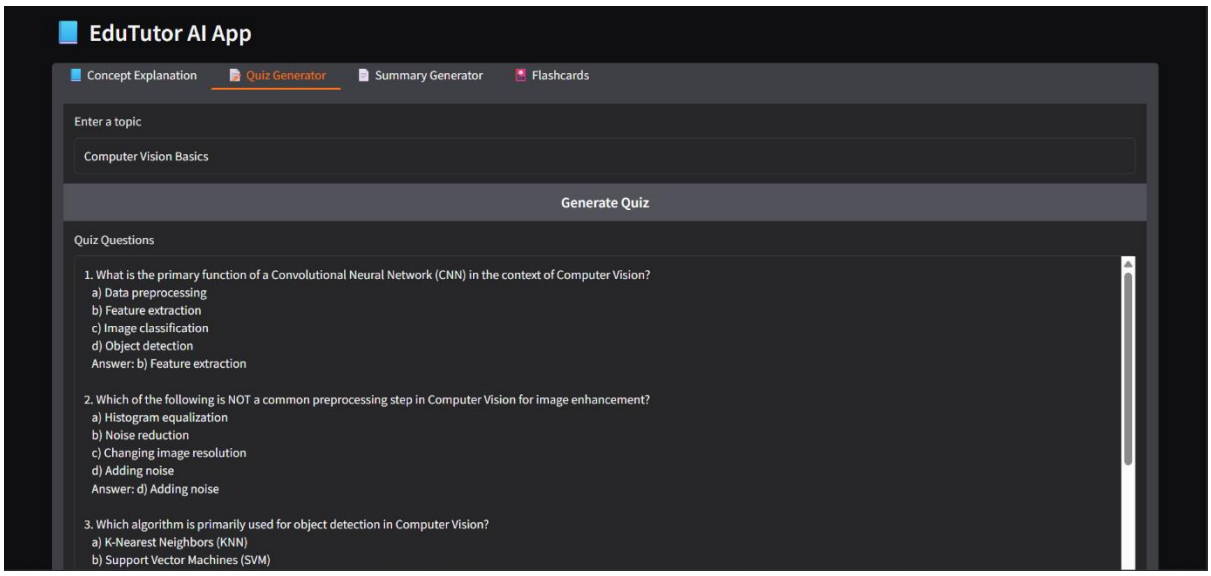
EduTutor AI landing page showing the Gradio interface with tabs for Concept Explanation, Quiz Generator, Summary Generator, and Flashcard Creator.



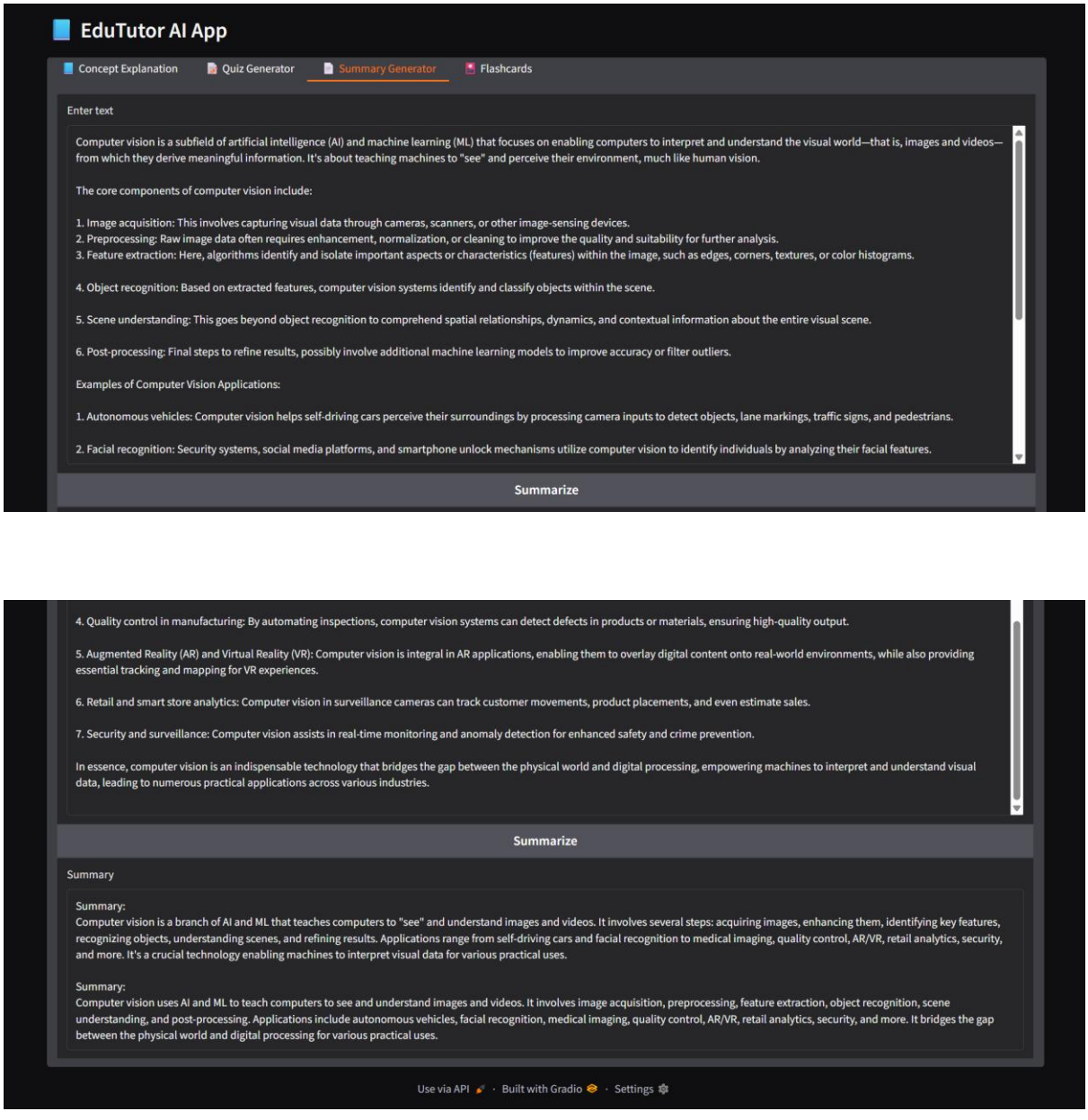
Login and Signup interface allowing users to create an account or log in with existing credentials, including role selection (Student/Faculty).



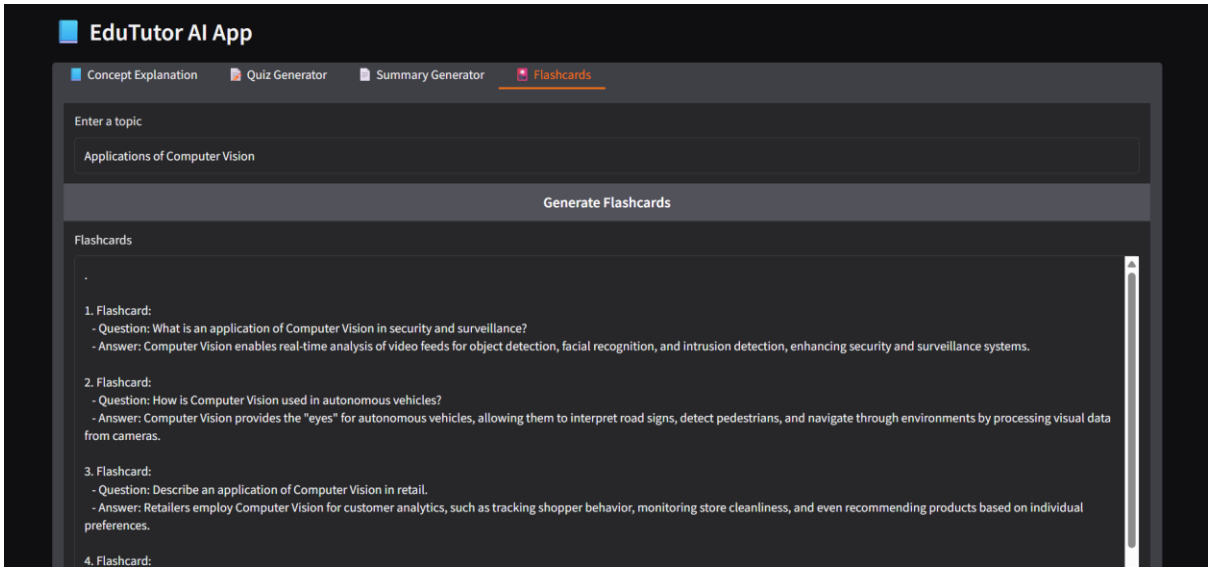
Concept Explanation tab where users can enter a topic to receive detailed explanations with simple examples in real-time.



Quiz Generator tab displaying automatically generated quiz questions (MCQ, True/False, Short Answer) along with answers for a given topic.



Text Summarizer tab showing concise summaries of long text input, simplifying content for easier understanding.



Flashcard Creator tab generating question-and-answer flashcards for effective memorization of key concepts.

12.Known Issues:

1. **Model Limitations**
    - The AI may produce incorrect, misleading, or repetitive answers. Fact-checking is often required.
2. **Token Limit Constraints**
    - Very long inputs may be truncated due to the 1024-token limit, leading to incomplete responses or summaries.
3. **Performance and Resource Usage**
    - Large models consume significant RAM/VRAM. Response generation can be slow, especially on CPU, and may crash on low-memory systems.
4. **Content Formatting Issues**
    - Generated quizzes, flashcards, or summaries may not always follow the requested format, requiring manual proofreading.

5. **Lack of Contextual Memory**
    - The AI does not remember previous interactions, so follow-up questions need explicit context for accurate answers.

13.Future Enhancements:

1. **Improved Model Accuracy and Updates**
    - Upgrade or fine-tune the AI model to provide more accurate explanations and reduce errors.
2. **Contextual Memory and Multi-turn Conversations**
    - Enable the AI to remember previous queries for coherent follow-up answers.
3. **Enhanced Quiz and Flashcard Features**
    - Support multiple difficulty levels, interactive elements, and visual aids for better learning.
4. **User Personalization**
    - Track user progress, allow bookmarking, and provide adaptive suggestions based on performance.
5. **Web & Mobile Deployment Enhancements**
    - Move from temporary Gradio sharing to a full web app and mobile-friendly version for easy access.