Project Documentation

Team Information

Team ID: NM2025TMID00373
Team Size: 5
Team Leader: Prithivi Rajan S
Team Member: Kuberan N
Team Member: Vetrivel G
Team Member: Prithish P S
Team Member: Deepak S

1. Introduction

Project Title: Educational AI Assistant (EDU-TUTOR_AI)

2. Project Overview

Purpose

The Educational AI Assistant is designed to enhance the learning experience for students, teachers, and self-learners. By leveraging IBM Granite LLM models, it provides clear explanations of academic concepts and generates quizzes automatically. This project bridges AI and education, making learning more interactive, accessible, and personalized.

Features

- **Concept Explanation:** Explains user-specified topics in detail with examples.
- **Quiz Generator:** Generates 5 mixed-format quiz questions with answers.
- **Conversational AI:** Accepts natural language inputs and provides instant responses.
- **User-Friendly Interface:** Built with Gradio for simple interaction.

3. Architecture

Frontend (Gradio): Provides a tabbed dashboard with Concept Explanation and Quiz Generator.
Backend (Python & Transformers): Uses IBM Granite LLM for generating responses. Core functions include generate_response, concept_explanation, and quiz_generator.
Deployment: Runs as a Gradio application, accessible locally or via shareable link.

4. Setup Instructions

Prerequisites

- Python 3.9 or later
- pip (package manager)
- Internet access to download IBM Granite model

Installation Process

- Clone or download the project source code.
- Install dependencies: pip install transformers torch gradio.

- Run the application: python educational_ai.py
- Open the Gradio-provided URL in a browser.

5. Folder Structure

project_root/ ■ ■■■ educational_ai.py # Main application file ■■■ requirements.txt # Python dependencies ■■■ outputs/ # (Optional) Store generated outputs

6. Running the Application

- Launch the app with: python educational_ai.py
- Concept Explanation Tab: Enter a concept (e.g., machine learning) and click Explain.
- Quiz Generator Tab: Enter a topic (e.g., physics) and click Generate Quiz to receive 5 questions with answers.

7. API Documentation

Core Functions:

- generate_response(prompt, max_length) → Returns AI-generated text.
- concept_explanation(concept) → Returns detailed explanation.
- quiz_generator(concept) → Returns quiz questions and answers.

8. Authentication

Currently open-access for demo. Future enhancements may include API keys, role-based access, and user accounts.

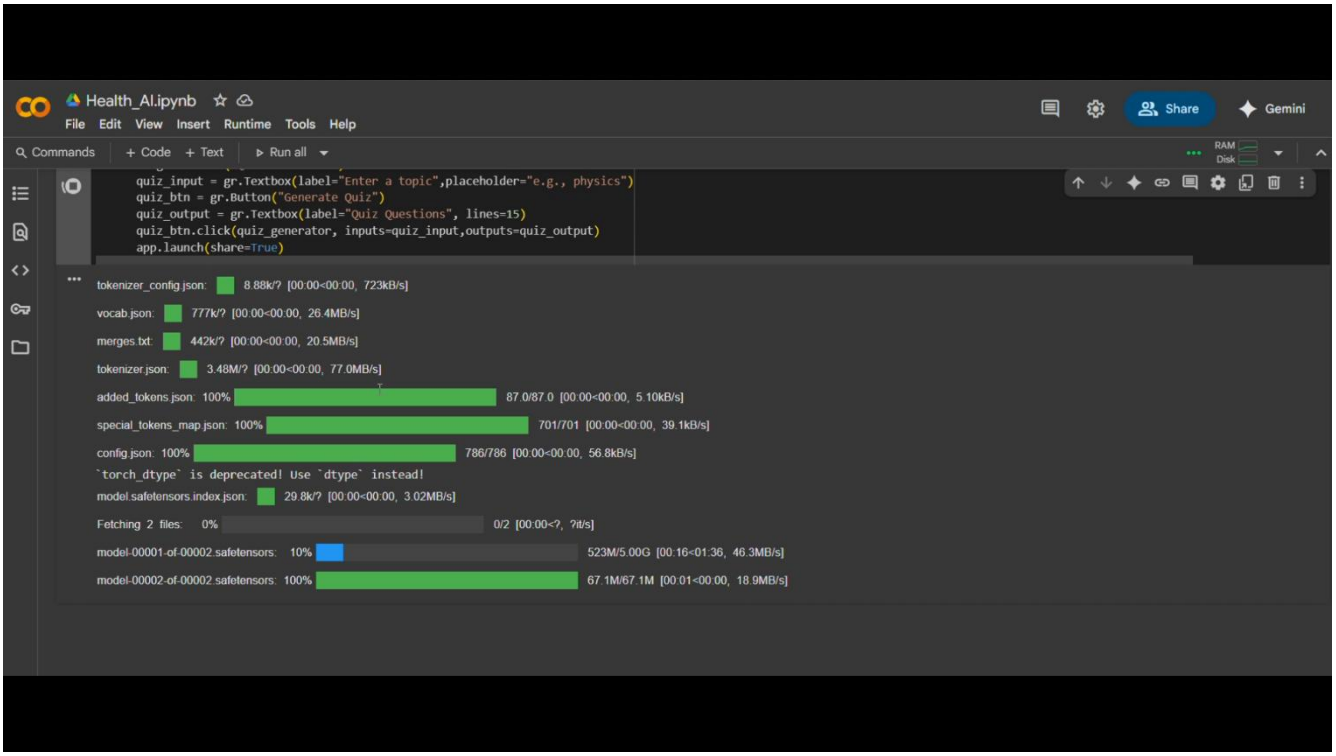
9. User Interface

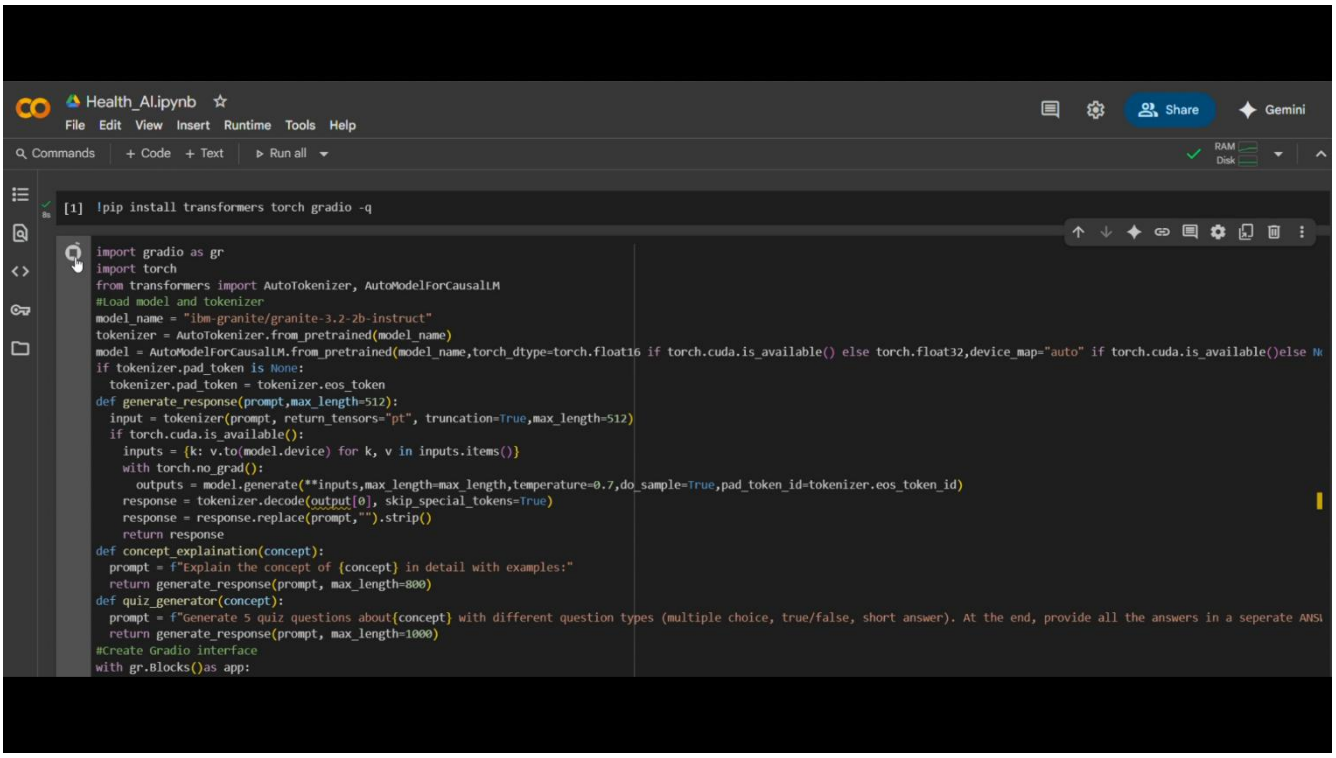
Tab 1: Concept Explanation → Input concept, get detailed explanation.
Tab 2: Quiz Generator → Input topic, get 5 quiz questions and answers.

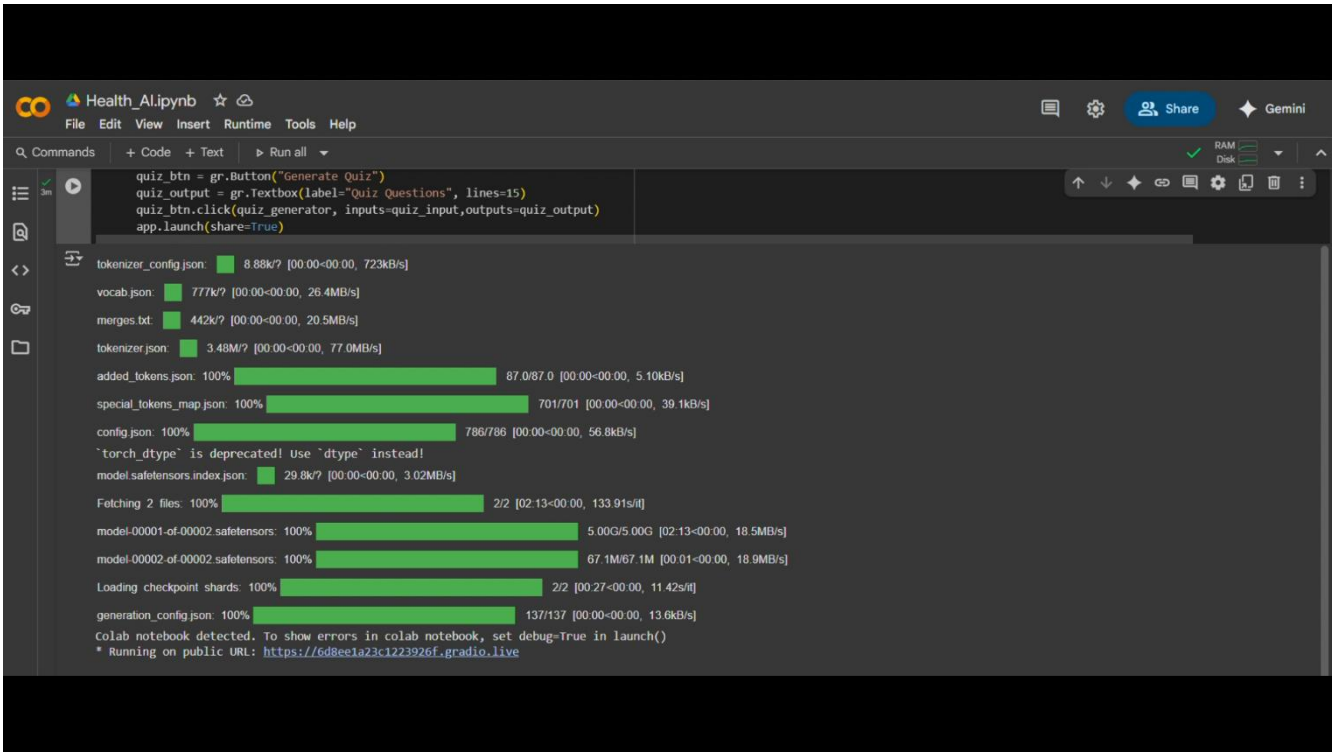
10. Testing

- Unit Testing: Verified correctness of core functions.
- Manual Testing: Tested with multiple topics.
- Edge Cases: Empty inputs handled gracefully.

11.Screenshot







12. Known Issues

- Occasional repetition in quiz question formats.
- Answers may sometimes be verbose.
- Requires stable internet connection for model.

13. Future Enhancements

- Add MCQ format with options.
- Provide explanations for quiz answers.
- Enable PDF export of quizzes.
- Add voice-based input/output.
- Integrate with LMS systems.