DOCUMENTATION FOR EDU_TUTOR

1. Introduction

Project Title: [EDU_TUTOR]
Team Members: M Bhuvaneshwaran, A Abishek, Branham Moses, J Lakshamanan

2. Project Overview

Purpose: EduTutor is an AI-powered educational assistant designed to explain concepts in detail and generate quizzes for learners. It provides interactive learning support using natural language models.

Features:

1. Concept explanation with detailed examples
2. Quiz generation with multiple question types
3. Interactive Gradio interface for learners
4. AI-powered responses using IBM Granite model

3. Architecture

Component Structure:

1. Concept Explanation Tab – Generates detailed concept explanations
2. Quiz Generator Tab – Creates quizzes with answers
3. Gradio Interface – Provides user-friendly front-end

State Management:

State is handled internally using Python functions and Gradio callbacks.

Routing:

Tabs within Gradio are used for navigation between concept explanation and quiz generation.

4. Setup Instructions

Prerequisites:

1. Python 3.8 or higher
2. pip package manager
3. Transformers library
4. Torch library
5. Gradio library

Installation:

1. Install Python and pip
2. Install required packages: pip install transformers torch gradio
3. Run the EduTutor script
4. Access the app in the browser: http://localhost:7860

5. Folder Structure

```
EduTutor/ ■■■ EduTutor.ipynb ■■■ main.py ■■■ requirements.txt ■■■ README.md
```

6. Running the Application

1. Start the EduTutor application: python main.py
2. The Gradio interface will launch in the browser
3. Use the Concept Explanation or Quiz Generator tabs

7. Component Documentation

1. Concept Explanation – Generates a detailed explanation of user-provided concepts
2. Quiz Generator – Produces quiz questions and answers based on a topic
3. App – Manages Gradio interface and tab navigation

8. State Management

State is managed through Python functions and Gradio interactions. Model inference is handled using the Transformers library.

9. User Interface

The application uses Gradio with tabs for Concept Explanation and Quiz Generation. Each tab provides input fields and buttons with output text areas for responses.

10. Styling

The interface uses Gradio's default styling with clean layouts. Future improvements may include custom themes.

11. Testing

Testing Strategy:

1. Manual testing of concept explanation and quiz generation
2. Validation of Gradio interface functionality
3. Ensuring proper integration with IBM Granite model

12. Screenshots or Demo

Provide screenshots of the Concept Explanation and Quiz Generator tabs, or link to a live demo.

13. Known Issues

1. Response generation may take longer on CPU compared to GPU
2. Occasional truncation of responses if max length is exceeded

14. Future Enhancements

1. Add support for more interactive quizzes (MCQ with options)
2. Improve explanation clarity with diagrams or visuals
3. Enable multi-language support
4. Allow saving and exporting quizzes