Edu Tutor AI: Personalized Learning

Project Documentation

1.Introduction

- Project title: EduTutor AI
- Team member: Bharathi
- Team member: Anushree
- Team member: Akshaya
- Team member: Anandhi

2.Project overview

- Purpose:

  The purpose of the Educational AI Assistant is to enhance learning and teaching by providing detailed concept explanations and interactive quiz generation. Powered by IBM Watsonx Granite LLM and deployed through Gradio, this assistant serves as a personal tutor for students and a teaching aid for educators. It delivers clear explanations, diverse question sets, and intelligent responses to improve knowledge retention.
- Features:

1. Concept Explanation

Key Point: Detailed learning support

Functionality: Explains any concept in simple, structured, and example-driven language.

2. Quiz Generator

Key Point: Interactive assessment tool

Functionality: Creates multiple types of quiz questions (MCQ, True/False, Short Answer) along with an answer key.

3. Conversational Interface

Key Point: Natural language interaction

Functionality: Enables users to type queries and receive AI-powered responses in real time.

4. User-Friendly Interface (Gradio UI)

Key Point: Minimalist design

Functionality: Provides tabs for easy navigation between Concept Explanation and Quiz Generation.

5. Extensible Design

Key Point: Future-ready system

Functionality: Can be enhanced to include flashcards, voice interaction, policy summarization, and integration with external knowledge sources.

3. Architecture

Frontend (Gradio):

- Provides an interactive web-based UI with tabbed sections for concept explanations and quizzes.
- Handles text input/output with a clean interface.

Backend (Python):

- Uses Fast, lightweight Python scripts for processing inputs and generating AI-driven outputs.
- Built with Transformers library for model handling.

4. LLM Integration (IBM Watsonx Granite):

Uses Granite-3.2-2B-Instruct model for natural language understanding and generation. Generates coherent explanations and quizzes with prompt engineering.

Model Inference:

- Uses Hugging Face pipeline with PyTorch (torch.float16 on GPU or torch.float32 on CPU).

Flow:

- User enters concept/topic.
- Gradio forwards input → Tokenizer → Granite Model.
- Model generates structured response.
- Response decoded → displayed in UI.

5. Setup Instructions

Prerequisites:

Python 3.9+

pip and venv tools

Internet connection (to download model)

GPU (optional, for faster inference)

Installation Process:

- Clone the repository.

Install dependencies using:

```
pip install -r requirements.txt
```

Run the app:

```
python app.py
```

Launches Gradio app locally with option share=True to generate a public link.

6. Folder Structure

```
educational_ai/

│
├── app.py              # Main Gradio app
├── requirements.txt      # Dependencies
├── utils/
│   ├── model_loader.py    # Loads tokenizer and model
│   ├── response_gen.py    # Handles response generation
│   ├── quiz_module.py     # Creates quizzes
│
├── docs/
│   └── project_doc.md     # Documentation file
│
└── tests/
    └── test_app.py        # Unit tests
```

7. Running the Application

- Start the Python script (app.py).
- Open Gradio interface in browser or use the public share link.
- Select Concept Explanation tab → enter concept → get explanation.
- Select Quiz Generator tab → enter topic → generate quiz + answers.

8. API / Function Documentation

- generate_response(prompt, max_length=512)
- Takes a prompt, tokenizes, sends to model, and generates response.

concept_explanation(concept)

Creates a detailed explanation with examples.

quiz_generator(concept)

Generates 5 quiz questions in varied formats + answers.

9. Authentication

Current demo runs open access.

Future deployments can include:

- API Key protection
- JWT Authentication
- Role-based access (Student, Teacher, Admin)

10. User Interface

Tabs:

- Concept Explanation
- Quiz Generator
- Inputs: Textboxes for concept/topics.
- Outputs: Large text boxes with results.
- Buttons: Explain / Generate Quiz.

11. Testing

Unit Testing:

- Response generation tested for consistenc
- Quiz format validated for completeness.

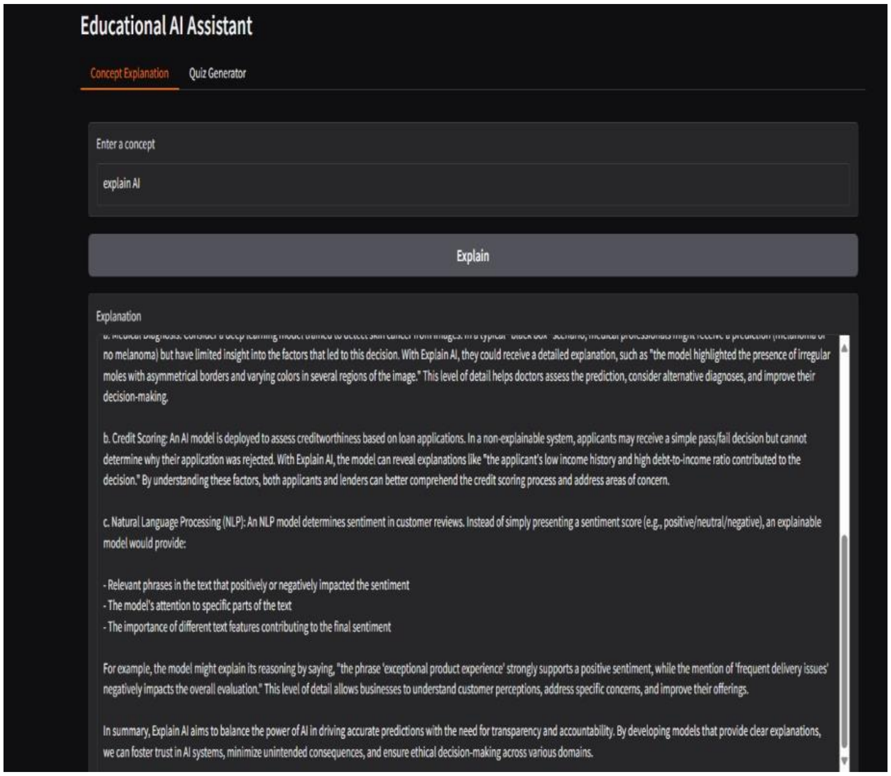
Manual Testing:

- Verified input/output flow via Gradio.
- Checked handling of unknown/complex topics.

Edge Case Handling:

- Long inputs truncated to 512 tokens.
- Model fallbacks to EOS token for padding.

12. Screenshots



Enter a concept

explain AI

Explanation

a. Medical Diagnosis: Consider a deep learning model trained to detect skin cancer from images. In a typical "black box" scenario, medical professionals might receive a prediction (melanoma or no melanoma) but have limited insight into the factors that led to this decision. With Explain AI, they could receive a detailed explanation, such as "the model highlighted the presence of irregular moles with asymmetrical borders and varying colors in several regions of the image." This level of detail helps doctors assess the prediction, consider alternative diagnoses, and improve their decision-making.

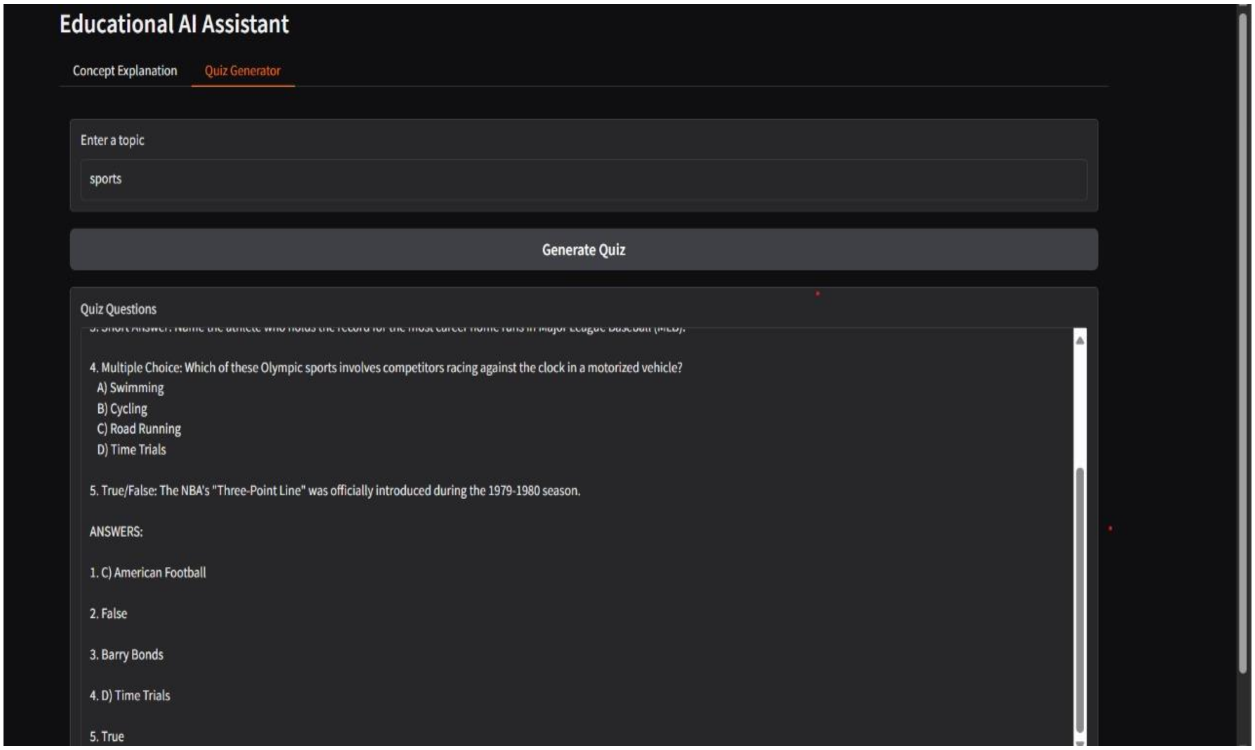
b. Credit Scoring: An AI model is deployed to assess creditworthiness based on loan applications. In a non-explainable system, applicants may receive a simple pass/fail decision but cannot determine why their application was rejected. With Explain AI, the model can reveal explanations like "the applicant's low income history and high debt-to-income ratio contributed to the decision." By understanding these factors, both applicants and lenders can better comprehend the credit scoring process and address areas of concern.

c. Natural Language Processing (NLP): An NLP model determines sentiment in customer reviews. Instead of simply presenting a sentiment score (e.g., positive/neutral/negative), an explainable model would provide:

- Relevant phrases in the text that positively or negatively impacted the sentiment
- The model's attention to specific parts of the text
- The importance of different text features contributing to the final sentiment

For example, the model might explain its reasoning by saying, "the phrase 'exceptional product experience' strongly supports a positive sentiment, while the mention of 'frequent delivery issues' negatively impacts the overall evaluation." This level of detail allows businesses to understand customer perceptions, address specific concerns, and improve their offerings.

In summary, Explain AI aims to balance the power of AI in driving accurate predictions with the need for transparency and accountability. By developing models that provide clear explanations, we can foster trust in AI systems, minimize unintended consequences, and ensure ethical decision-making across various domains.



Enter a topic

sports

Quiz Questions

```
3. Short Answer: Name the athlete who holds the record for the most career home runs in Major League Baseball (MLB).

4. Multiple Choice: Which of these Olympic sports involves competitors racing against the clock in a motorized vehicle?
  A) Swimming
  B) Cycling
  C) Road Running
  D) Time Trials

5. True/False: The NBA's "Three-Point Line" was officially introduced during the 1979-1980 season.

ANSWERS:

1. C) American Football

2. False

3. Barry Bonds

4. D) Time Trials

5. True
```

13. Known Issues

- Long responses may get cut off due to max length.
- Model responses may vary slightly for same input (stochastic sampling).
- Limited question variety in some subjects.

14. Future Enhancement

- Add voice input/output for accessibility.
- Generate flashcards for quick revision.
- Export quizzes to PDF/CSV for teachers.
- Add multi-language support.
- Integration with school LMS platforms.