Project Title

Project Documentation

1 . Introduction

- Project title: EduTutor AI: Personalized Learning
  - Team member
  - Megasri R (Team leader)
  - Jerlin Jeno V
  - Padma Sree M
  - Pavithra T

2 Project Overview •

purpose:

The purpose of EduTutor AI is to provide personalized learning support using Generative AI with IBM Granite models.
The project helps students with concept explanations, quiz generation, and interactive learning through AI-driven solutions.
It leverages Google Colab for execution with low setup and GitHub for version control.

- Features:

Concept Explainer

Key Point: Simplified learning

Functionality: Generates easy-to-understand explanations for complex topics.

Quiz Generator

Key Point: Interactive practice

Functionality: Creates quizzes based on chosen topics for self-assessment.

AI Integration with IBM Granite

Key Point: Generative AI support

Functionality: Uses Hugging Face Granite models to enhance learning outcomes.

Cloud Deployment

Key Point: Easy accessibility

Functionality: Runs smoothly on Google Colab with GPI-J support.

GitHub Integration

GitHub Integration

Key Point: Version control and collaboration

Functionality: Stores and shares project code.

3. Architecture

Frontend (Gradio UI):

The frontend is built with Gradio, offering an interactive interface for students to access explainers and quizzes.

Backend (Python & IBM Granite):

Python scripts interact with IBM Granite models on Hugging Face to generate learning content and responses.

Deployment (Google Colab):

The project is deployed on Google Colab with GPU acceleration for efficiency and low-cost execution.

Version Control (GitHub):

Project code and files are maintained on GitHub for collaboration and progress tracking.

4. Setup Instructions

Prerequisites:

o Python 3.8 or later o Google Colab with T4 GPI-J enabled o Hugging Face account for IBM Granite models o GitHub account for version control

Installation Process:

o Access the Naan Mudhalvan Smart Internz Portal and enroll in EduTutor AI.

o Select IBM Granite model from Hugging Face (e.g., granite-3.2-2b-instruct).

o Run the notebook in Google Colab, install dependencies, and execute code.

o Upload the project to GitHub with documentation and source code.

5. Folder Structure app/ - Contains main project scripts for quiz generation and explainers.
ui/ - Contains Gradio interface code.

notebooks/ - Google Colab notebooks for running the project.
README.md - Documentation of the project.

6. Running the Application

To start the project:

➢ Open Google Colab and load the EduTutor AI notebook.

➢ Enable GPU runtime and install dependencies.

➢ Run the provided Python code cells to generate explainers and quizzes.

➢ Open the Gradio link to interact with the application.

➢ Upload your project files to GitHub for submission.

7. API Documentation

The project relies on IBM Granite Hugging Face API models.

APIs used include:

POST /generate-explainer — Generates concept explanations.

POST /generate-quiz — Creates quizzes based on topics.

GET /project-progress — Tracks project progress on Naan Mudhalvan portal.

8. Authentication

The project runs in an open environment for demonstration.

For secure deployment, integrate:

- Token-based authentication for API access
- GitHub user authentication for collaboration

9. User Interface

The interface is simple and student-friendly:

Sidebar with navigation

Tabs for quizzes and explainers

Real-time AI-generated content

Output preview in Gradio

10. Testing

Testing was done In multiple phases: Unit Testing:
For Al model prompt responses.

Manual Testing: For quiz and explainer generation.

Integration Testing: Google Colab and Hugging Face API connectivity.

Edge Case Handling: Invalid topics, large quiz sizes, missing API keys.

1 1 . Screenshots

(Add screenshots of Colab notebook, Gradio app, and GitHub repo)

12. Known Issues

- Dependent on internet connectivity for API access.
- Limited to IBM Granite model capacity.

13. Future Enhancements

- Add support for more subjects and multimodal learning (PDFs, images).
- Enhance analytics and personalized student feedback.
- Deploy as a standalone web app beyond Google Colab.

EduTutor AI.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

ommands + Code + Text ▷ Run all RAM Disk

[2]

```
app.launch(share=True)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and re
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 8.88k/? [00:00<00:00, 800kB/s]
vocab.json: 777k/? [00:00<00:00, 10.2MB/s]
merges.txt: 442k/? [00:00<00:00, 26.6MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 88.2MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 7.74kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 57.2kB/s]
config.json: 100% 786/786 [00:00<00:00, 98.9kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.49MB/s]
Fetching 2 files: 0% 0/2 [00:00<?, ?it/s]
model-00001-of-00002.safetensors: 66% 3.30G/5.00G [01:57<01:16, 22.2MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:08<00:00, 7.42MB/s]
```

Executing (2m 41s) T4 (Python 3)

EduTutor AI.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all RAM Disk

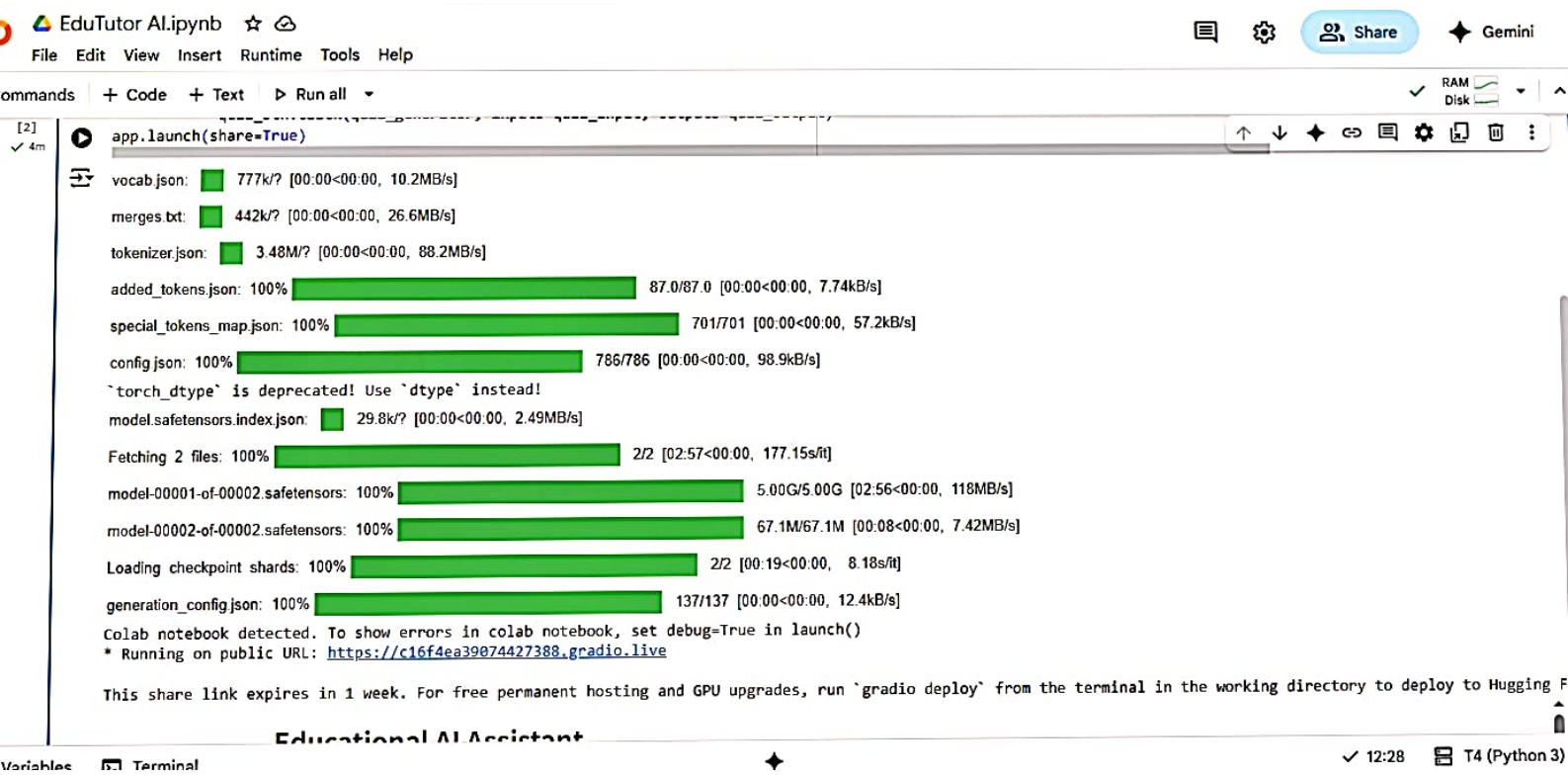
[2] 4m

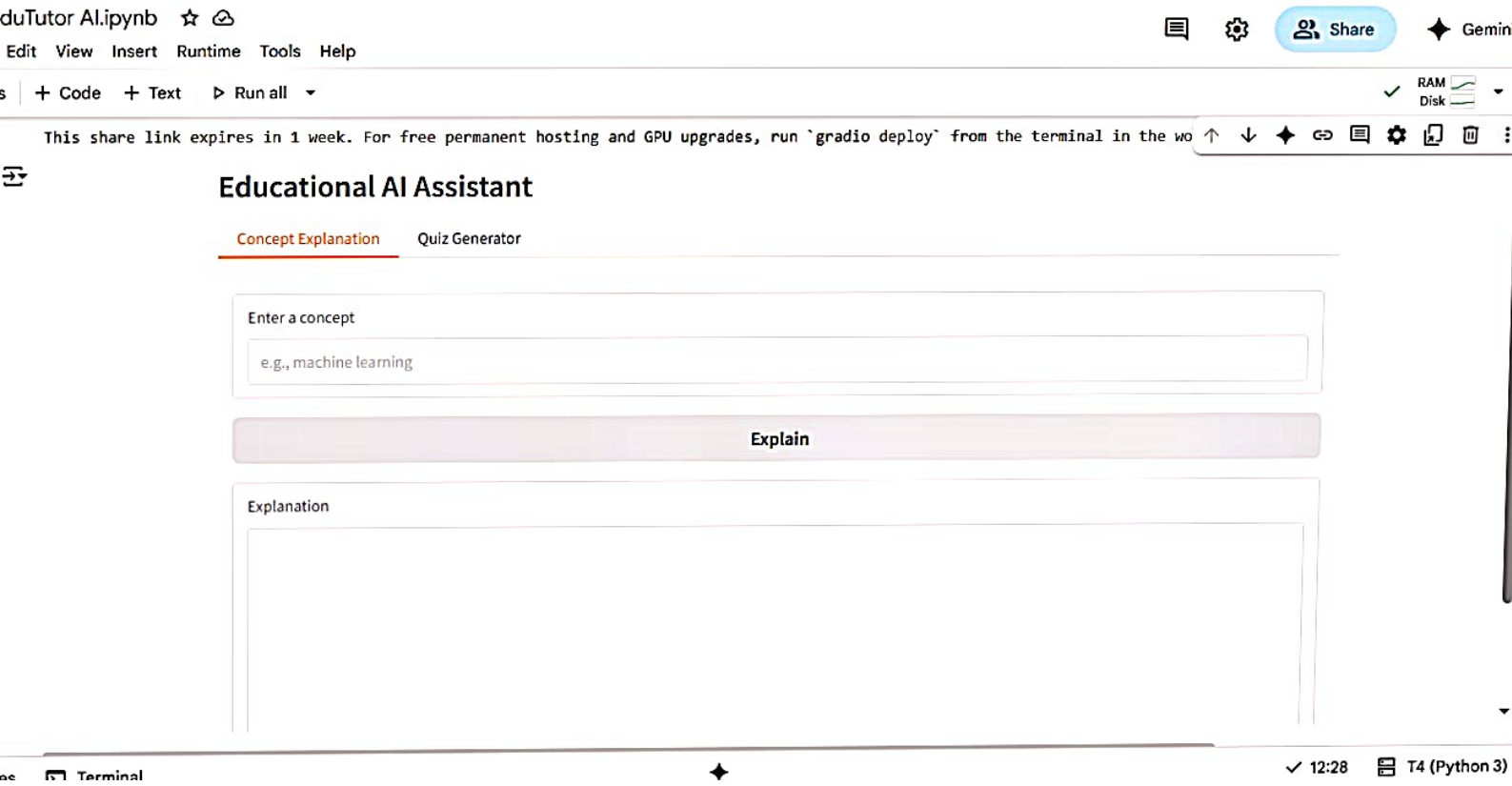
```
app.launch(share=True)
```

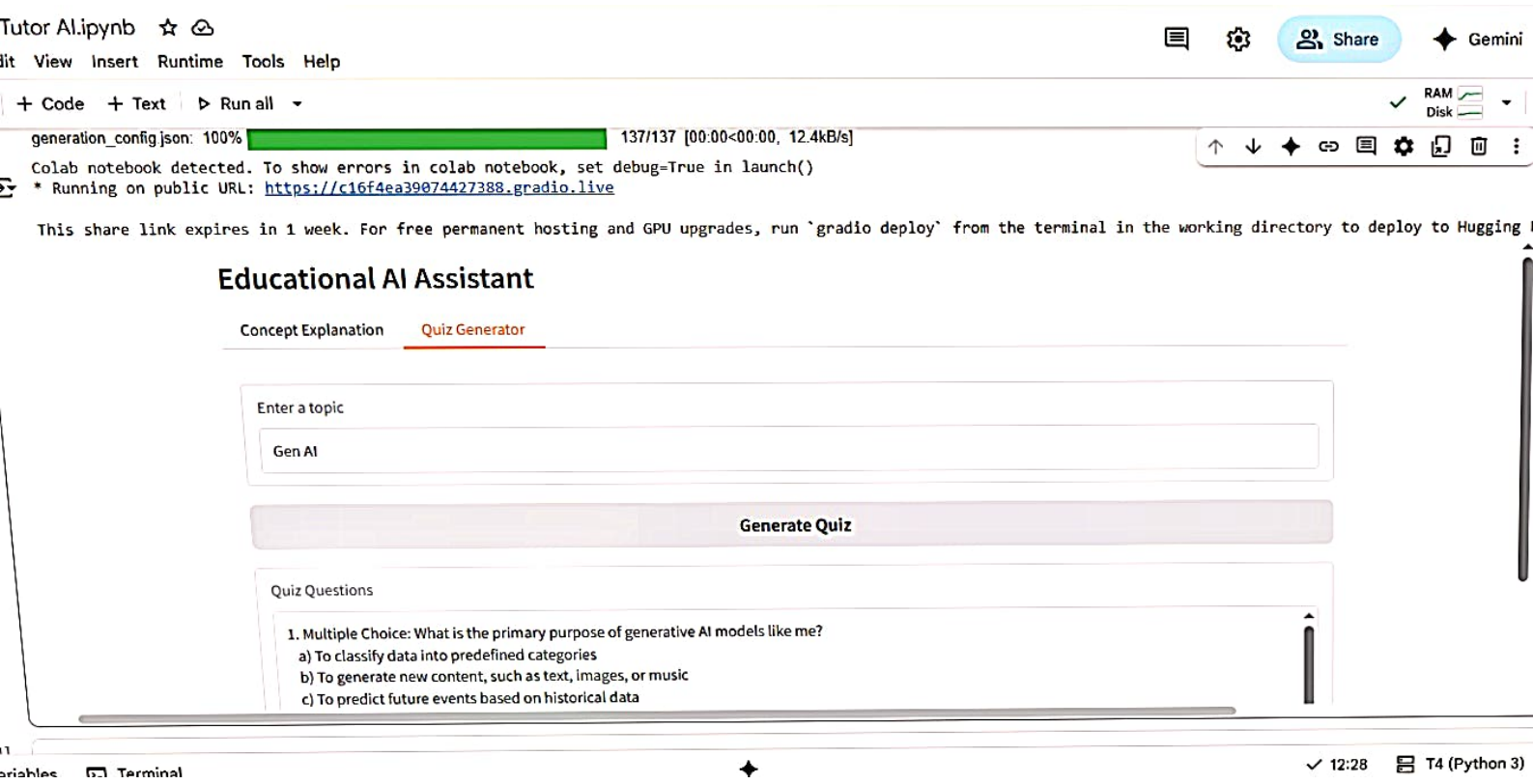
```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and re
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 8.88k/? [00:00<00:00, 800kB/s]
vocab.json: 777k/? [00:00<00:00, 10.2MB/s]
merges.txt: 442k/? [00:00<00:00, 26.6MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 88.2MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 7.74kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 57.2kB/s]
config.json: 100% 786/786 [00:00<00:00, 98.9kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.49MB/s]
Fetching 2 files: 100% 2/2 [02:57<00:00, 177.15s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:56<00:00, 118MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:08<00:00, 7.42MB/s]
Loading checkpoint shards: 100% 2/2 [00:19<00:00, 8.18s/it]
```

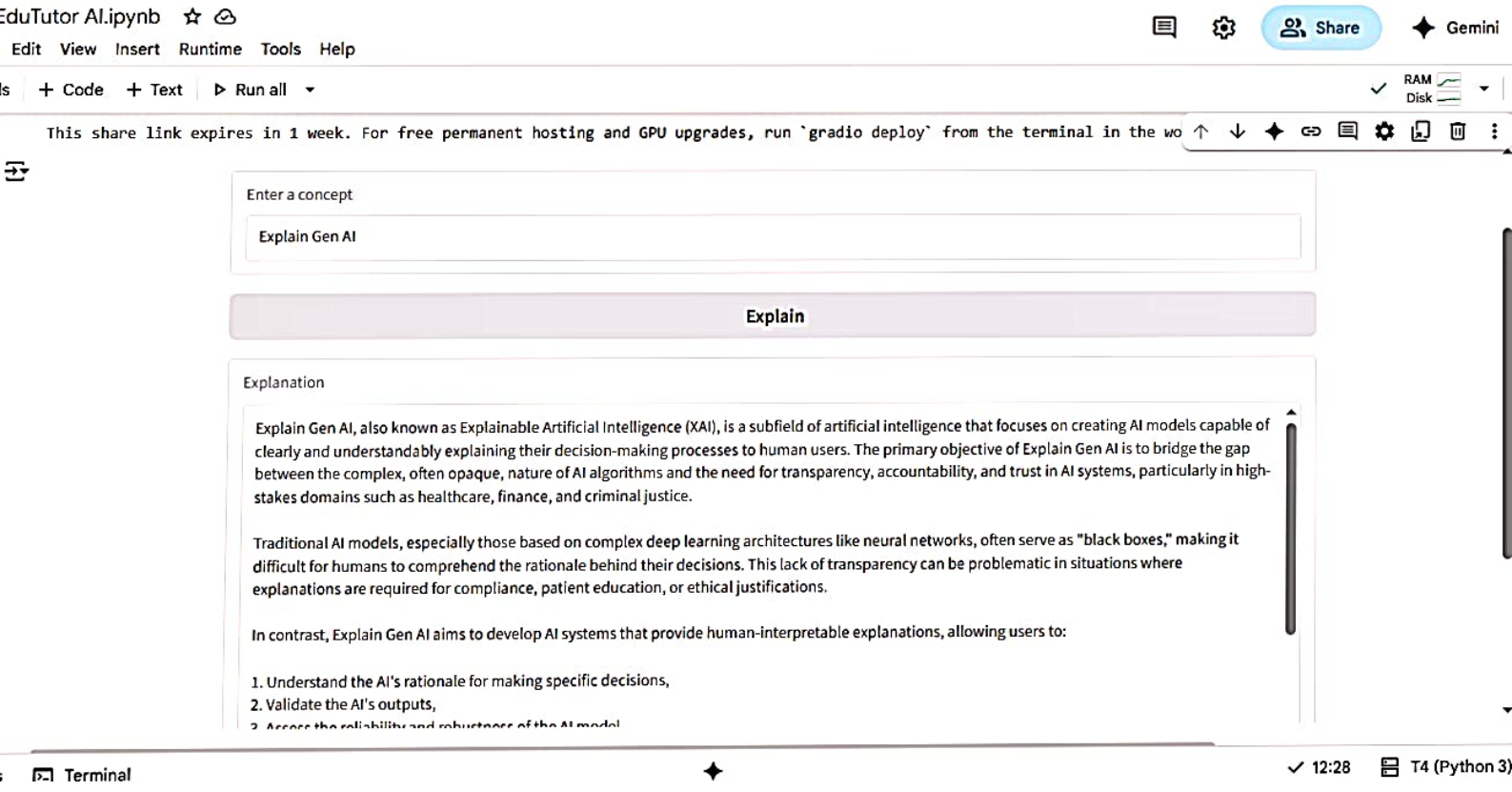
Variables Terminal

12:28 T4 (Python 3)


EduTutor AI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
+ Code + Text Run all
RAM
Disk
[2]
4m
app.launch(share=True)
vocab.json: 777k/? [00:00<00:00, 10.2MB/s]
merges.txt: 442k/? [00:00<00:00, 26.6MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 88.2MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 7.74kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 57.2kB/s]
config.json: 100% 786/786 [00:00<00:00, 98.9kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.49MB/s]
Fetching 2 files: 100% 2/2 [02:57<00:00, 177.15s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [02:56<00:00, 118MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:08<00:00, 7.42MB/s]
Loading checkpoint shards: 100% 2/2 [00:19<00:00, 8.18s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 12.4kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://c16f4ea39074427388.gradio.live
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging F
Educational AI Assistant
Variables
Terminal
12:28
T4 (Python 3)


duTutor AI.ipynb
Edit View Insert Runtime Tools Help
Share
Gemini
+ Code + Text Run all
RAM
Disk
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the wo
Educational AI Assistant
Concept Explanation
Quiz Generator
Enter a concept
e.g., machine learning
Explain
Explanation
Terminal
12:28
T4 (Python 3)


Tutor AI.ipynb
dit View Insert Runtime Tools Help
Share
Gemini
+ Code + Text Run all
RAM
Disk
generation_config.json: 100% 137/137 [00:00<00:00, 12.4kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://c16f4ea39074427388.gradio.live
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging
Educational AI Assistant
Concept Explanation
Quiz Generator
Enter a topic
Gen AI
Generate Quiz
Quiz Questions
1. Multiple Choice: What is the primary purpose of generative AI models like me?
a) To classify data into predefined categories
b) To generate new content, such as text, images, or music
c) To predict future events based on historical data
Terminal
12:28
T4 (Python 3)


EduTutor AI.ipynb
Edit View Insert Runtime Tools Help
Share
Gemini
+ Code + Text Run all
RAM
Disk
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the wo
Enter a concept
Explain Gen AI
Explain
Explanation
Explain Gen AI, also known as Explainable Artificial Intelligence (XAI), is a subfield of artificial intelligence that focuses on creating AI models capable of clearly and understandably explaining their decision-making processes to human users. The primary objective of Explain Gen AI is to bridge the gap between the complex, often opaque, nature of AI algorithms and the need for transparency, accountability, and trust in AI systems, particularly in high-stakes domains such as healthcare, finance, and criminal justice.
Traditional AI models, especially those based on complex deep learning architectures like neural networks, often serve as "black boxes," making it difficult for humans to comprehend the rationale behind their decisions. This lack of transparency can be problematic in situations where explanations are required for compliance, patient education, or ethical justifications.
In contrast, Explain Gen AI aims to develop AI systems that provide human-interpretable explanations, allowing users to:
1. Understand the AI's rationale for making specific decisions,
2. Validate the AI's outputs,
Terminal
12:28
T4 (Python 3)