

EduTutor AI: Personalized Learning with Generative AI and LMS Integration

Team ID – NM2025TMID03774

222307708 **Palani Bharathi J** (TEAM LEADER)

222307663 **Abdul Kalam B**

222307711 **Rakshith K**

222307727 **Venkatesan K**

1. Introduction

EduTutor AI is an **AI-powered educational assistant** that helps learners by:

- Explaining complex concepts with examples.
- Generating interactive quizzes for practice.

It is built using **Hugging Face Transformers**, **PyTorch**, and **Gradio** for a simple and interactive web UI. The project is designed to run on **Google Colab** for easy accessibility and GPU acceleration.

2. Project Overview

The project has two main features:

1. **Concept Explanation** – Users input a concept (e.g., *Binary Search*, *Photosynthesis*), and EduTutor AI generates a detailed explanation with examples.
2. **Quiz Generator** – Users input a topic, and the system generates 5 quiz questions with an answer key.

Target Users: Students, teachers, and self-learners.

Goal: Provide a lightweight, interactive AI tutor that runs directly in a browser.

3. System Architecture

Components:

- **Frontend (Gradio UI):** User input and result display.
- **Backend (Hugging Face Model + PyTorch):** Generates explanations and quizzes.
- **Deployment:** Hosted via **Colab** (with optional `share=True` for a public link).

Flow:

1. User enters a concept/topic in Gradio interface.
2. Request sent to **Granite 3.2 Instruct model** (Hugging Face).
3. Model processes prompt → generates response.
4. Output displayed in Gradio textbox.

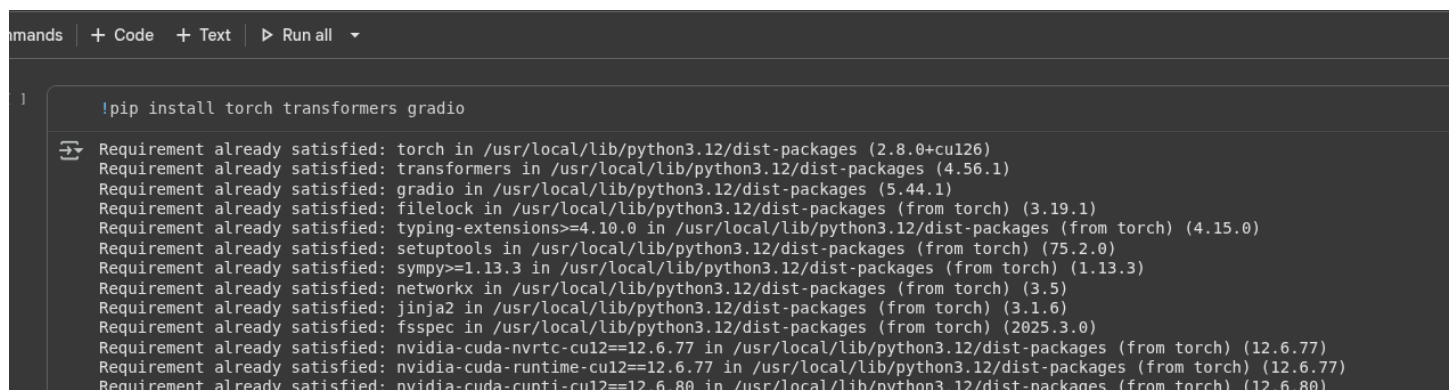
4. Setup Instructions

Using Google Colab

1. Open the Colab notebook.
2. Run the first cell to install dependencies:

```
!pip install torch transformers gradio
```

3. Copy the **project code** into a new Colab cell.
4. Run all cells.
5. Gradio will display a link to access the app.



```
Commands | + Code | + Text | ▶ Run all ▼

!pip install torch transformers gradio

Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.8.0+cu126)
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.56.1)
Requirement already satisfied: gradio in /usr/local/lib/python3.12/dist-packages (5.44.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch) (3.19.1)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch) (3.5)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch) (12.6.80)
```

5. Folder Structure

Since this is a **Colab project**, the structure is simple:

```
EduTutorAI/
├── U.ipynb          # Main Colab Notebook
├── edututorai.py    # Python file (optional modular script)
├── requirements.txt  # List of dependencies
├── README.md        # Documentation
└── assets/          # (Optional) images/screenshots/diagrams
```

6. Running the Application

- Open the Colab notebook.
- Run all cells.

- At the end, Gradio will display two URLs:
 - **Local URL** (works only in Colab runtime).
 - **Public Shareable URL** (if share=True is enabled).

7. API Documentation

This project does not expose a REST API separately (since it's wrapped in Gradio).

But internally:


- **generate_response(prompt, max_new_tokens, temperature)**
 - Inputs: prompt (string), max_new_tokens (int), temperature (float)
 - Output: Generated response text
- **concept_explanation(concept)**
 - Input: Concept name
 - Output: Detailed explanation with examples
- **quiz_generator(concept)**
 - Input: Topic name
 - Output: Quiz (5 questions + answers)

8. Authentication

Currently, no authentication is implemented (open access).

Possible enhancements:

- Add **API key-based authentication**.
- Restrict **public share links** with passwords or tokens.

 EduTutor AI — Educational Assistant

[Concept Explanation](#) [Quiz Generator](#)

Enter a concept (e.g., `machine learning`, `binary search`, `photosynthesis`) and get a detailed explanation.

Enter a concept

e.g., machine learning

Explain


Explanation

Notes

- The model may sometimes produce slightly incorrect facts; always verify answers for critical use.
- If you run on CPU the model may be slow — using a GPU is recommended for faster response times.

9. Testing

- **Manual Testing:**
 - Enter different concepts (e.g., "binary search", "gravity") → check output relevance.
 - Generate quizzes and cross-check answers.
- **Performance Testing:**
 - Run on both **CPU and GPU** (Colab free tier + Pro).
 - Compare response times.
- **Future:**
 - Add **unit tests** for helper functions (using pytest).

 EduTutor AI — Educational Assistant

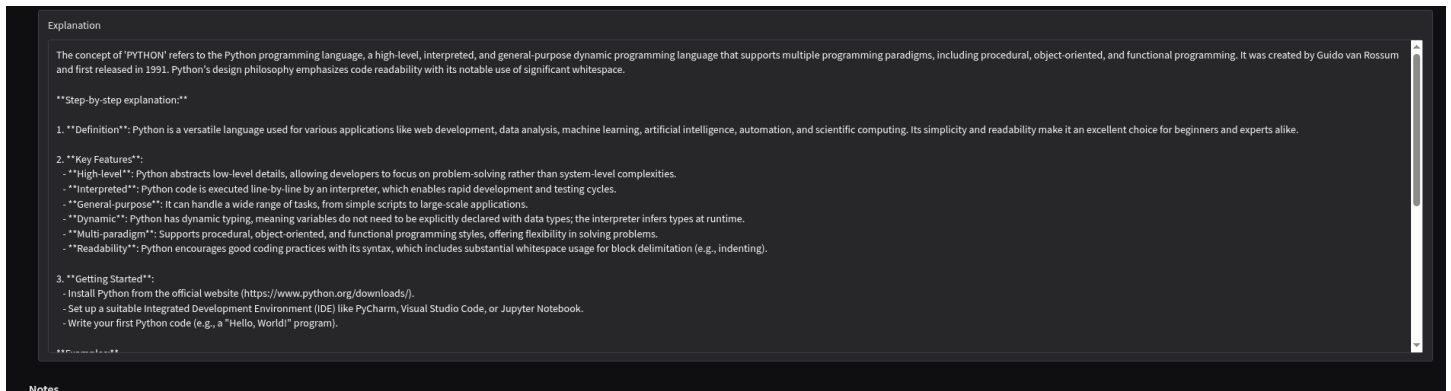
[Concept Explanation](#) [Quiz Generator](#)

Enter a concept (e.g., `machine learning`, `binary search`, `photosynthesis`) and get a detailed explanation.

Enter a concept

PYTHON

Explain



10. Conclusion & Future Enhancements

EduTutor AI is a lightweight educational AI app that:

- Explains concepts clearly.
- Creates quizzes for practice.
- Runs easily on Google Colab with Gradio.

Future Enhancements

- Add **multi-language support**.
- Save **generated quizzes to PDF/Google Drive**.
- Add **student performance tracking**.
- Implement **login system** for personalized learning.
- Deploy to **Hugging Face Spaces** or **Render** for permanent hosting.