

# EduTutor AI-Personalized Learning With IBM Granite

## 1.INTRODUCTION

**Project Title:** EduTutor AI

**Team Members:**

- [Team Member 1]     JEEVITHA S
- [Team Member 2]     MADHUVANTHI S
- [Team Member 3]     HEMA LATHA K
- [Team Member 4]     JANANI M

## 2.PROJECT OVERVIEW

**Purpose:**

EduTutor AI leverages IBM Granite models to provide personalized learning support. It offers interactive tools such as concept explainers, quiz generation, and adaptive learning experiences. The project is designed to simplify education through generative AI, enabling students to learn at their own pace with clear explanations, assessments, and progress tracking.

**Features:**

- Concept Explainer

Key Point: Simplified learning

Functionality: Breaks down complex topics into clear, easy-to-understand explanations.

- Quiz Generator

Key Point: Adaptive assessment

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

- Personalized Learning Flow

Key Point: Student-specific assistance

Functionality: Suggests learning paths and resources based on student needs.

- Low-Setup Deployment

Key Point: Easy accessibility

Functionality: Runs on Google Colab with minimal setup using Gradio interface.

### **3.ARCHITECTURE**

#### **Frontend (Gradio UI):**

The interface is built with Gradio, providing a simple, interactive environment for learners. Users can input queries, generate quizzes, and receive AI-based responses directly in the browser.

#### **Backend (Google Colab + Python):**

Google Colab provides a GPU environment (T4) for running the Granite model efficiently. Python scripts handle model integration, quiz generation, and user interaction.

#### **LLM Integration (IBM Granite Model):**

Granite models from Hugging Face are integrated for natural language processing. In this project, granite-3.2-2b-instruct is used due to its lightweight and fast performance.

#### **Version Control (GitHub):**

The project code is maintained in GitHub for collaboration, version tracking, and future enhancements.

### **4.SETUP INSTRUCTIONS**

#### **Prerequisites:**

- Python 3.9+
- Google Colab account with GPU access
- Hugging Face account (for IBM Granite models)

- GitHub account (for version control)
- Knowledge of Gradio Framework

### **Installation Process:**

1. Open Google Colab and create a new notebook.
2. Set runtime to T4 GPU.
3. Install dependencies:  
!pip install transformers torch gradio -q
4. Import and configure IBM Granite model from Hugging Face.
5. Run the notebook to launch the Gradio app.
6. Access the app through the provided URL.
7. Upload the project code to GitHub for version control.

## **5.FOLDER STRUCTURE**

- colab\_notebook/ – Contains Google Colab notebooks for project execution.
- app.py – Main Python script with model integration and Gradio interface.
- requirements.txt – List of dependencies (transformers, torch, gradio).
- /docs – Documentation and guides.

## **6.RUNNING THE INTERFACE**

1. Open Colab and run the notebook.
2. The Granite model downloads automatically.
3. A Gradio interface URL is generated.
4. Access the Gradio app in a new browser tab.
5. Use features like concept explainer and quiz generator interactively.

## **7.PROJECT WORKFLOW**

- Activity 1: Explore Naan Mudhalvan Smart Interz Portal.
- Activity 2: Select IBM Granite model from Hugging Face.
- Activity 3: Run application in Google Colab.
- Activity 4: Upload project files to GitHub.

## 8.USER INTERFACE

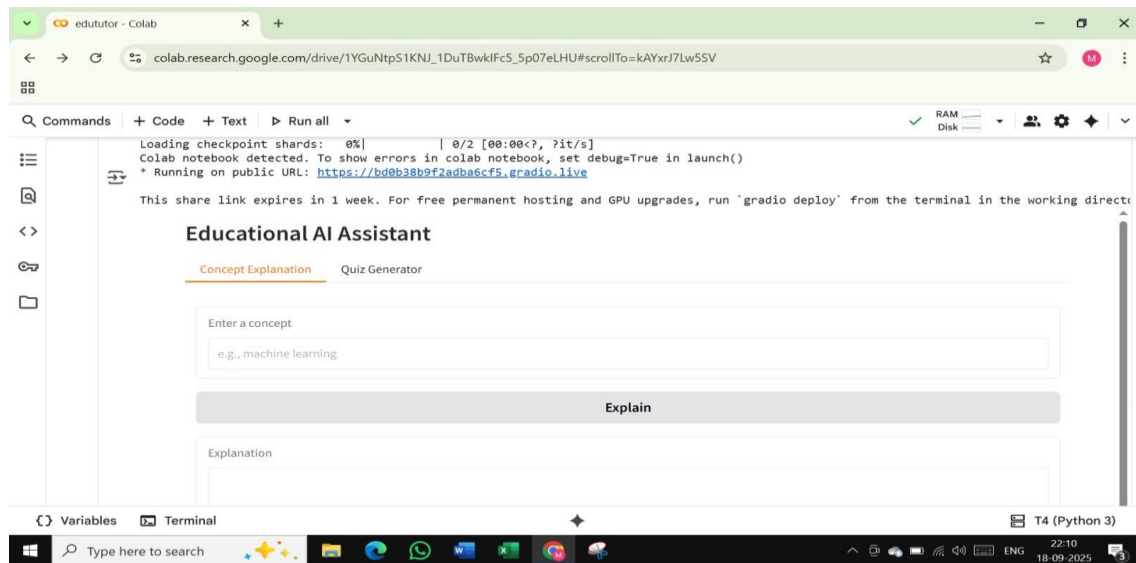
The Gradio interface is simple and intuitive:

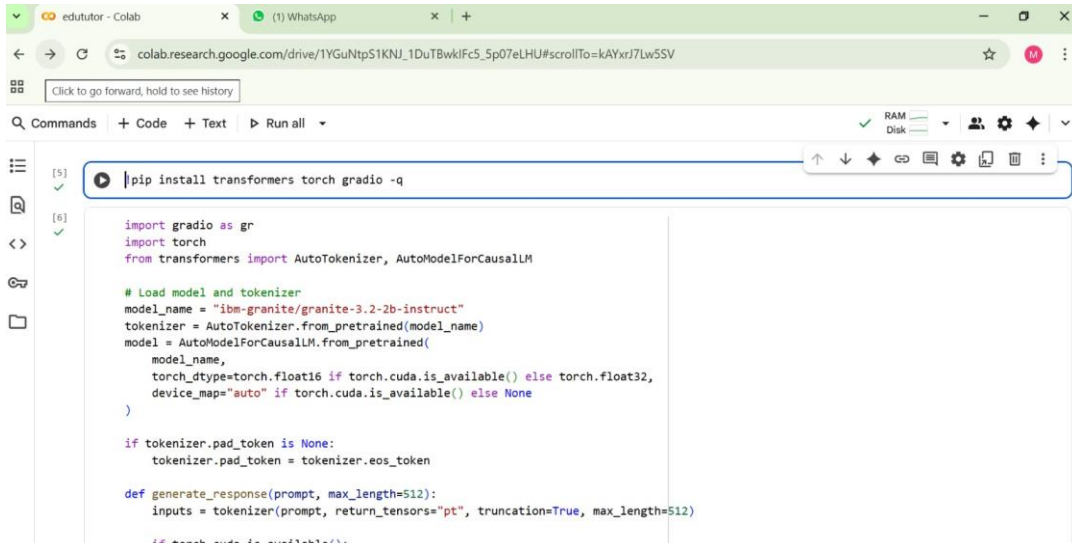
- Input box for queries
- Output section for explanations and generated quizzes
- Minimal design for easy navigation
- Browser-based access (no local installation needed)

## 9.TESTING

- Unit Testing: Verified model response quality.
- Manual Testing: Checked quiz generation and explanation accuracy.
- Edge Cases: Tested with difficult queries and large inputs.
- Performance Testing: Ensured smooth execution on Google Colab GPU.

## 10.SCREEN SHOTS





The screenshot shows a Google Colab notebook interface. The top bar includes tabs for 'edututor - Colab' and '(1) WhatsApp'. The address bar shows the URL 'colab.research.google.com/drive/1YGUNtpS1KNJ\_1DuTBwkiFc5\_5p07eLHU#scrollTo=kAYxrJ7Lw5SV'. The notebook has a left sidebar with icons for file explorer, search, and other functions. The main area shows a code cell with the following content:

```
[5] | pip install transformers torch gradio -q

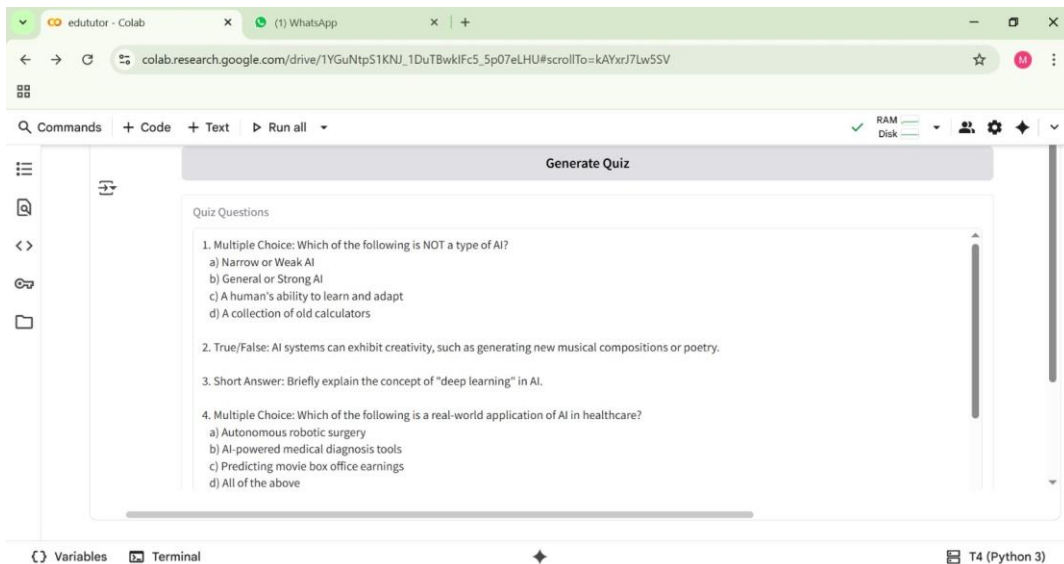
[6] import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# Load model and tokenizer
model_name = "ibm-granite/granite-3.2-2b-instruct"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    device_map="auto" if torch.cuda.is_available() else None
)

if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

def generate_response(prompt, max_length=512):
    inputs = tokenizer(prompt, return_tensors="pt", truncation=True, max_length=512)

    if torch.cuda.is_available():
```



The screenshot shows a Google Colab notebook interface. The top bar includes tabs for 'edututor - Colab' and '(1) WhatsApp'. The address bar shows the URL 'colab.research.google.com/drive/1YGUNtpS1KNJ\_1DuTBwkiFc5\_5p07eLHU#scrollTo=kAYxrJ7Lw5SV'. The notebook has a left sidebar with icons for file explorer, search, and other functions. The main area shows a code cell with the following content:

```
Generate Quiz

Quiz Questions

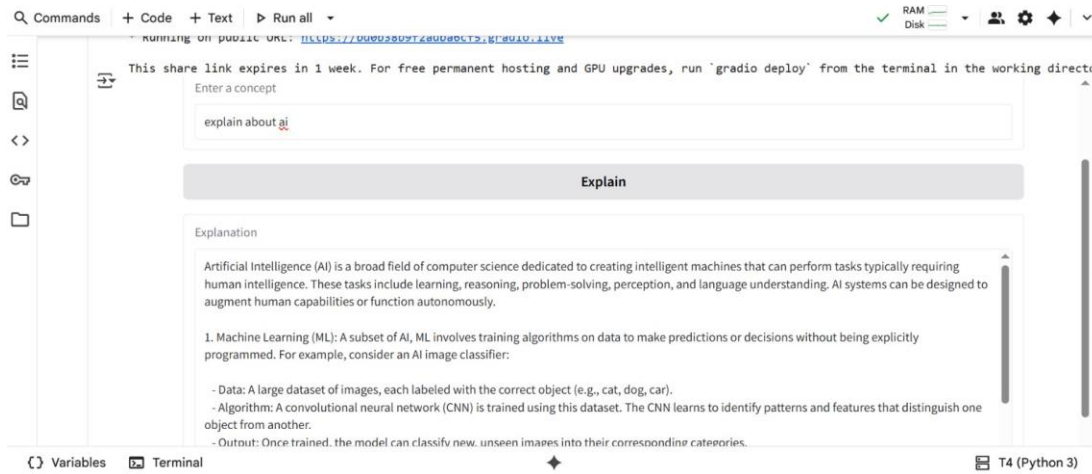
1. Multiple Choice: Which of the following is NOT a type of AI?
a) Narrow or Weak AI
b) General or Strong AI
c) A human's ability to learn and adapt
d) A collection of old calculators

2. True/False: AI systems can exhibit creativity, such as generating new musical compositions or poetry.

3. Short Answer: Briefly explain the concept of "deep learning" in AI.

4. Multiple Choice: Which of the following is a real-world application of AI in healthcare?
a) Autonomous robotic surgery
b) AI-powered medical diagnosis tools
c) Predicting movie box office earnings
d) All of the above
```

At the bottom of the notebook, there are tabs for 'Variables', 'Terminal', and 'T4 (Python 3)'.



## 11.FUTURE ENHANCEMENT

- Add support for multiple subjects and curricula.
- Integrate progress tracking for students.
- Provide teacher dashboards for monitoring performance.
- Expand to multimodal learning (text + images).