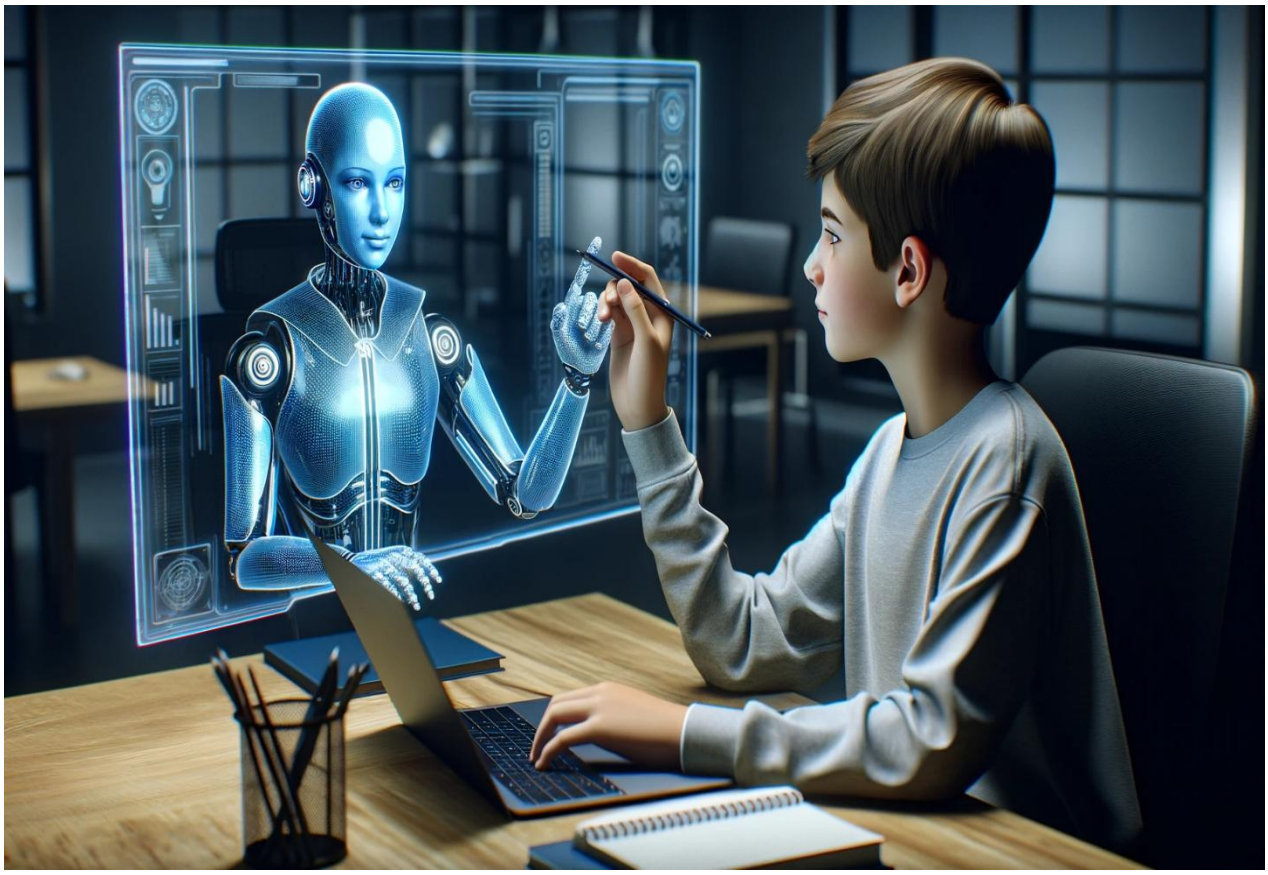
Edu Tutor AI: Personalized Learning Generative AI with IBM



Team Leader:

S.JANAVI

Team Members:

P.KAYALVIZHI and M.KEERTHANA

1. Project Overview

- EduTutor AI is a personalized learning platform powered by IBM Granite Models from Hugging Face. The system enables learners to access concept explainers, generate quizzes, and explore interactive learning tools.
- The project is implemented in Google Colab with Gradio framework for user interface, ensuring low setup effort and reliable performance.

2. Objectives

- Provide personalized learning support using AI.
- Simplify complex concepts into easy explainers.
- Automatically generate quizzes for practice.
- Deploy the solution with minimal setup via Colab and Gradio.
- Enable easy collaboration and version control using GitHub.

3. Prerequisites

1. Gradio Framework Knowledge – Gradio Docs

2. IBM Granite Models (Hugging Face) – Granite Models

3. Python Programming Proficiency – Python Docs

4. Version Control with Git – Git Docs

5. Google Colab T4 GPU Knowledge – Google Colab Guide

4. Project Workflow

Activity 1:

- Exploring Naan Mudhalvan Smart Interz Portal
- Login to Naan Mudhalvan Smartinternz portal.
- Navigate to Projects → EduTutor AI.
- Access resources, guided project, and workspace.
- Track project progress and upload demo links.

Activity 2:

- Choosing an IBM Granite Model
- Create an account in Hugging Face.
- Search for IBM Granite Models.
- Select granite-3.2-2b-instruct (lightweight & fast).

Activity 3:

- Running Application in Google Colab
- Open Google Colab and create a new notebook.
- Rename notebook (e.g., Health AI).
- Set Runtime → Change Runtime Type → T4 GPU.

Install dependencies:

1. !pip install transformers torch gradio -q
2. Run provided EduTutor AI code (Download Code).

3. Launch Gradio application and access via generated URL.

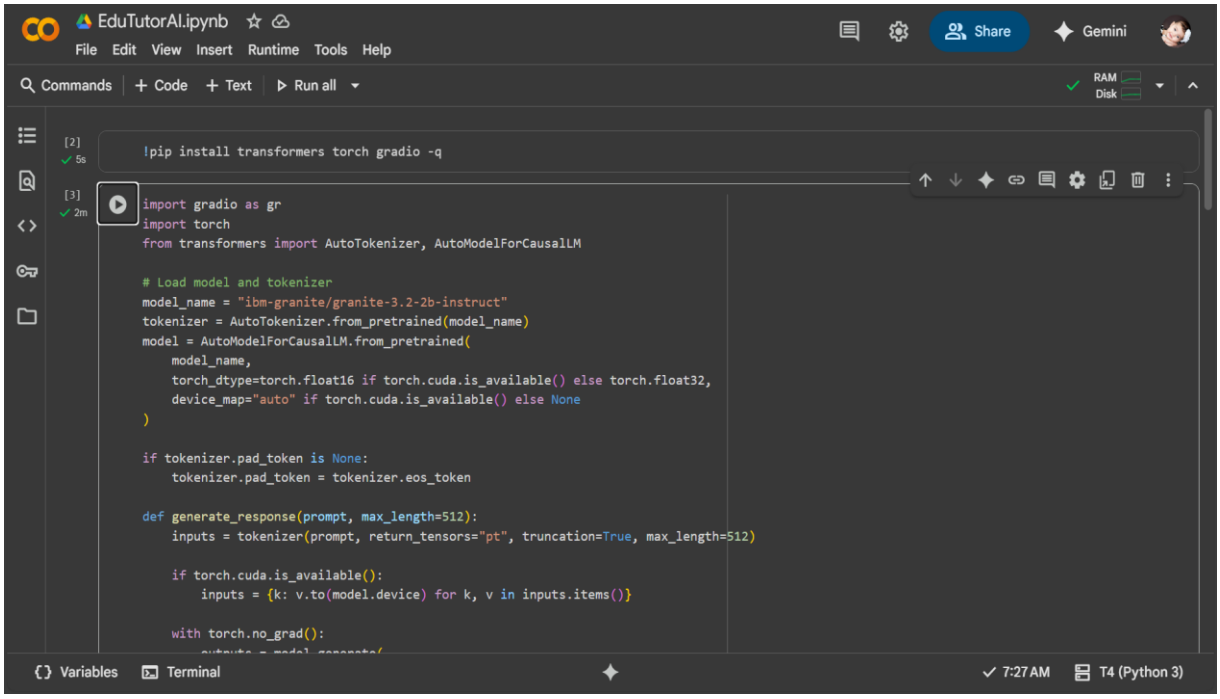
Activity 4:

- Upload Project to GitHub
- Sign up / log in to GitHub.
- Create a new repository (e.g., IBM-Project).
- Enable Add README file.
- Download .py file from Colab and upload it.
- Commit changes to finalize repository.

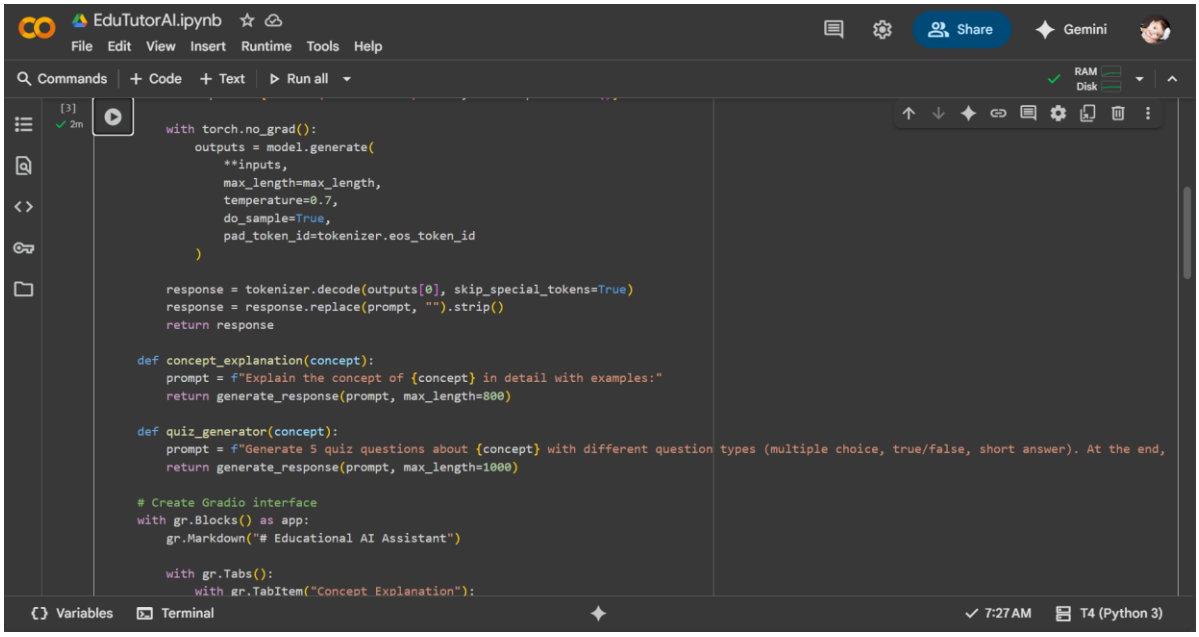
5. Output

- ✓ The EduTutor AI application runs successfully in Google Colab.
- ✓ Gradio provides an interactive interface for quizzes & explainers.
- ✓ The project is stored and version-controlled in GitHub.

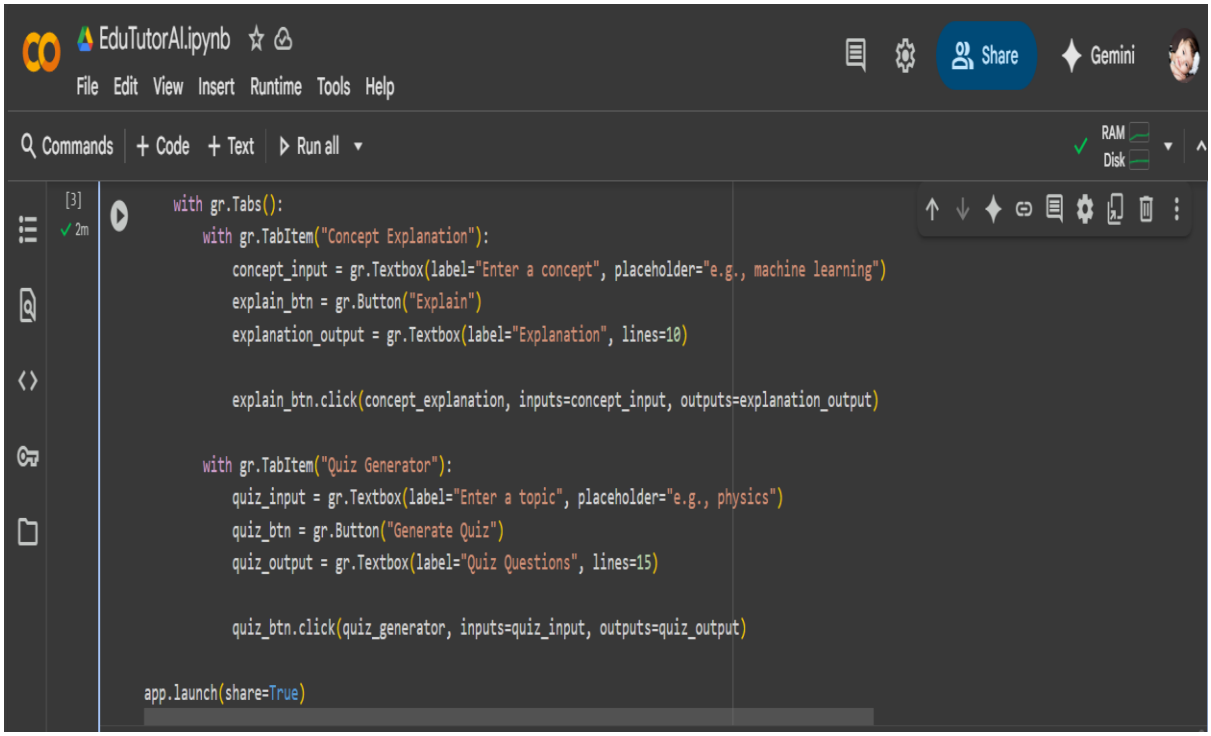
6.Project Screenshots



```
[2]
5s
!pip install transformers torch gradio -q

[3]
2m
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():
        inputs = {k: v.to(model.device) for k, v in inputs.items()}

    with torch.no_grad():
```



```
[3]
2m
    with torch.no_grad():
        outputs = model.generate(
            **inputs,
            max_length=max_length,
            temperature=0.7,
            do_sample=True,
            pad_token_id=tokenizer.eos_token_id
        )

    response = tokenizer.decode(outputs[0], skip_special_tokens=True)
    response = response.replace(prompt, "").strip()
    return response

def concept_explanation(concept):
    prompt = f"Explain the concept of {concept} in detail with examples:"
    return generate_response(prompt, max_length=800)

def quiz_generator(concept):
    prompt = f"Generate 5 quiz questions about {concept} with different question types (multiple choice, true/false, short answer). At the end,
    return generate_response(prompt, max_length=1000)

# Create Gradio interface
with gr.Blocks() as app:
    gr.Markdown("# Educational AI Assistant")

    with gr.Tabs():
        with gr.TabItem("Concept Explanation"):
```


EduTutorAI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
RAM
Disk
[3]
2m
with gr.Tabs():
with gr.TabItem("Concept Explanation"):
concept_input = gr.Textbox(label="Enter a concept", placeholder="e.g., machine learning")
explain_btn = gr.Button("Explain")
explanation_output = gr.Textbox(label="Explanation", lines=10)
explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation_output)
with gr.TabItem("Quiz Generator"):
quiz_input = gr.Textbox(label="Enter a topic", placeholder="e.g., physics")
quiz_btn = gr.Button("Generate Quiz")
quiz_output = gr.Textbox(label="Quiz Questions", lines=15)
quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)
app.launch(share=True)

7.Output

EduTutorAI.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
/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 yo
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, 162kB/s]
vocab.json: 777k/? [00:00<00:00, 6.01MB/s]
merges.txt: 442k/? [00:00<00:00, 9.34MB/s]
tokenizer.json: 3.48M/? [00:00<00:00, 36.9MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 4.74kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 28.7kB/s]
config.json: 100% 786/786 [00:00<00:00, 61.0kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 29.8k/? [00:00<00:00, 2.08MB/s]
Fetching 2 files: 100% 2/2 [01:24<00:00, 84.28s/it]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:24<00:00, 38.6MB/s]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 30.8MB/s]
Loading checkpoint shards: 100% 2/2 [00:19<00:00, 8.00s/it]
```

Variables Terminal 7:27 AM T4 (Python 3)

EduTutorAI.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 30.8MB/s]
Loading checkpoint shards: 100% 2/2 [00:19<00:00, 8.00s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 8.77kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://4c3a793835d6288e4b.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
```

Educational AI Assistant

Concept Explanation | Quiz Generator

Enter a concept

e.g., machine learning

Explain

Explanation

Variables Terminal 7:27 AM T4 (Python 3)

8.Conclusions

The Edu Tutor AI: Personalized Learning Generative AI with IBM using IMB granite LLM project was successfully done by: 3rd BCA 1st Section, Janavi.S ,Kayalvizhi.P , Keerthana.M .

9.Demo Video Link

https://drive.google.com/file/d/1zhKAJJKEBwv1PxddQxT6K2mZ-fa7WX-4/view?usp=drive_link