EduTutor AI: Personalized Learning with Generative AI and LMS Integration:

Team members :

Team Leader: Avinash J L

Sanchana M

Praveena S

Lavanya S

Jayakumar K

DESCRIPTION:

A simple web application that explains concepts and generates quizzes using IBM's Granite 3.2B Instruct language model. Built using Gradio and Hugging Face Transformers.

🚀 Features

- ✍️ **Concept Explanation**: Enter any topic and get a detailed explanation with examples.
- 🧠 **Quiz Generator**: Generate 5-question quizzes (multiple choice, true/false, short answer) with an answer key.
- ⚡ Powered by IBM Granite 3.2B Instruct via Hugging Face.
- 🖥️ Easy-to-use Gradio web interface.

🛠️ How It Works (Code Breakdown)

1. Import Libraries

```
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM
```

- **gradio: Web UI**
- **torch: GPU/CPU model inference**

- **transformers: Pre-trained language models**

2. Load the 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
)
```

- **Loads the IBM Granite 3.2B Instruct model.**
- **Uses GPU if available for faster inference.**

3. Ensure Pad Token Exists

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

- **Prevents errors in token generation by assigning a pad token.**

4. Define Generation Function

```
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():
        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
```

- Tokenizes input and generates a model response.
- Temperature adds randomness.
- Strips the prompt from the response.

5. **Custom Prompt Functions**

```
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, provide all the answers in a separate ANSWERS section:"
    return generate_response(prompt, max_length=1000)
```

- Prompts the model to generate content based on task.

6. Build Gradio Interface

```
with gr.Blocks() as app:
    gr.Markdown("# Educational AI Assistant")

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

```
        explain_btn = gr.Button(...)
        explanation_output = gr.Textbox(...)
        explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanation_output)

    with gr.TabItem("Quiz Generator"):
        quiz_input = gr.Textbox(...)
        quiz_btn = gr.Button(...)
        quiz_output = gr.Textbox(...)
        quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)

app.launch(share=True)
```

- Two tabs: one for explanation, one for quizzes.
- Button triggers the corresponding model function.

🧪 How to Use

1. Run the app:

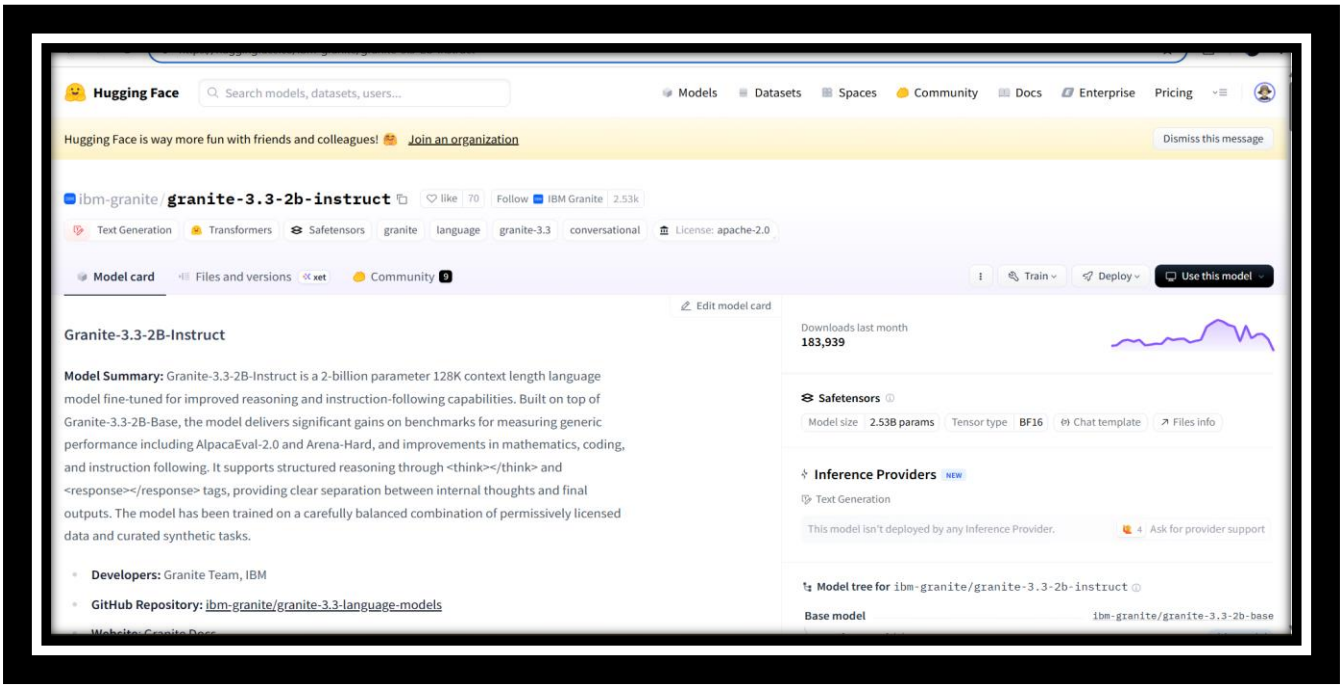
```
python app.py
```

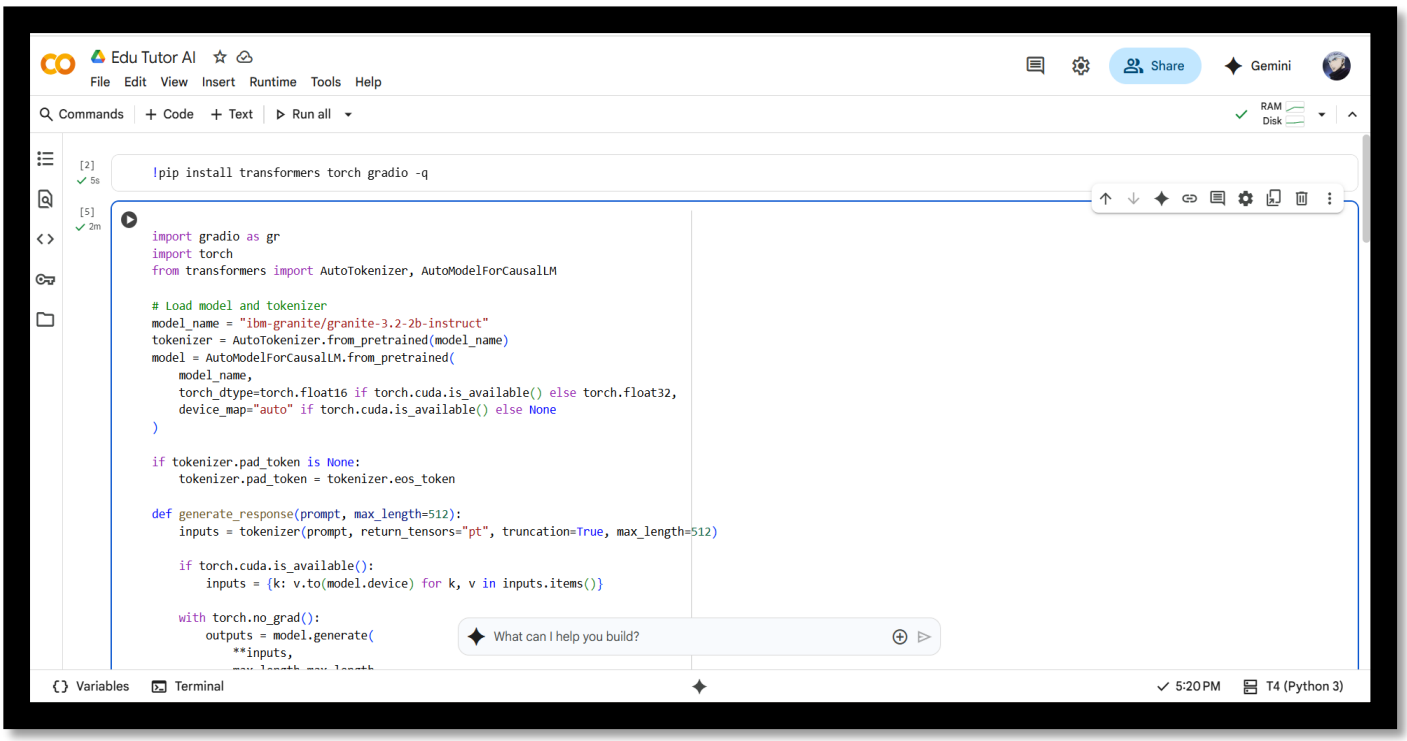
2. A browser tab will open with the Gradio interface.
3. Select a tab:
    - 🧠 Concept Explanation: Enter a concept (e.g., "neural networks").
    - ❓ Quiz Generator: Enter a topic (e.g., "photosynthesis").
4. Wait for the model to respond.

License & Model Info

- Model: Granite 3.2B Instruct by IBM
- License: Apache 2.0

SCREENSHOTS:





Edu Tutor AI

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
            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, provide all the answers in a separ
    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"):
            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")
```

What can I help you build?

Variables Terminal 5:20 PM T4 (Python 3)

Edu Tutor AI

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

```
added_tokens.json: 100%  87.0/87.0 [00:00<00:00, 9.76kB/s]
special_tokens_map.json: 100%  701/701 [00:00<00:00, 63.3kB/s]
config.json: 100%  786/786 [00:00<00:00, 85.6kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json:  29.8k/? [00:00<00:00, 2.68MB/s]
Fetching 2 files: 100%  2/2 [02:00<00:00, 120.27s/it]
model-00001-of-00002.safetensors: 100%  5.00G/5.00G [02:00<00:00, 110MB/s]
model-00002-of-00002.safetensors: 100%  67.1M/67.1M [00:01<00:00, 69.8MB/s]
Loading checkpoint shards: 100%  2/2 [00:20<00:00, 8.48s/it]
generation_config.json: 100%  137/137 [00:00<00:00, 17.3kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://58621e56c1edbc1622.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 Face Spaces (
```

Educational AI Assistant

Concept Explanation Quiz Generator

Enter a concept

e.g., machine learning

What can I help you build?

Variables Terminal 5:20 PM T4 (Python 3)