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

```
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/pythor
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/pythor
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.12/di
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.12/dis
Requirement already satisfied: fsspec in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-
Requirement already satisfied: networkx in /usr/local/lib/python3.12/dist-packa
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/li
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/li
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/p
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/p
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/py
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/pytho
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/pyth
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/p
Requirement already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-
```

```
     Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packa
     Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.
     Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/pythor
     Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.12/dist-pac
```

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

# Load model & 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
)

# Ensure pad token exists
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

# Function to generate response
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

# Concept explanation
def concept_explanation(concept):
    prompt = f"Explain the concept of {concept} in detail with example."
    return generate_response(prompt, max_length=800)

# Quiz generator
def quiz_generator(concept):
    prompt = f"Generate 5 quiz questions about {concept} with different question types (mul
    return generate_response(prompt, max_length=1200)

# Build Gradio app
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
            explain_btn = gr.Button("Explain")
            explanation_output = gr.Textbox(label="Explanation", lines=10)

            explain_btn.click(concept_explanation, inputs=concept_input, outputs=explanatio

        with gr.TabItem("Quiz Generator"):
            quiz_input = gr.Textbox(label="Enter a Topic", placeholder="e.g., Machine Learn
            quiz_btn = gr.Button("Generate Quiz")
            quiz_output = gr.Textbox(label="Quiz Questions & Answers", lines=15)

            quiz_btn.click(quiz_generator, inputs=quiz_input, outputs=quiz_output)

app.launch(share=True)
```

```
Loading checkpoint shards: 100%  2/2 [00:15<00:00,  6.50s/it]
Colab notebook detected. To show errors in colab notebook, set debug=True in lau
* Running on public URL: https://1f18c4c64855f2bce5.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades,
```