# Python Code - Educational AI Assistant

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():  
 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, provide all the answers in a separate ANSWERS section:"  
 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")  
 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)