# Smart SDLC - Gradio Python Code

import gradio as gr  
import torch  
from transformers import AutoTokenizer, AutoModelForCausalLM  
from PIL import Image, ImageDraw

# Added for image generation  
  
model\_name = "ibm-granite/granite-3.3-2b-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  
)  
  
# Modified to return both AI response and image  
def granite\_inference\_structured(phase, prompt):  
 try:  
 # Construct a more specific prompt based on the phase and user input  
 structured\_prompt = f"SDLC Phase: {phase}\nTask/Question: {prompt}\nOutput:"  
 # Use the existing tokenizer and model for inference  
 if model and tokenizer:  
 inputs = tokenizer(structured\_prompt, return\_tensors="pt").to(model.device)  
 outputs = model.generate(  
 inputs.input\_ids,  
 max\_new\_tokens=1000,  
 num\_return\_sequences=1,  
 temperature=0.7,  
 top\_p=0.9,  
 do\_sample=True,  
 early\_stopping=True  
 )  
 response = tokenizer.decode(outputs[0][inputs.input\_ids.shape[1]:], skip\_special\_tokens=True).strip()  
  
 # Create a simple image based on the input  
 img = Image.new("RGB", (600, 100), color=(128, 128, 128))  
 draw = ImageDraw.Draw(img)  
 display\_text = f"{phase}: {prompt[:500]}..." # Truncated prompt on image  
 draw.text((10, 40), display\_text, fill="white")  
  
 return response, img  
 else:  
 return "Model not loaded. Cannot perform inference.", None  
 except Exception as e:  
 return f"An error occurred during inference: {e}", None  
  
# Updated to display both AI response and generated image  
iface\_structured = gr.Interface(  
 fn=granite\_inference\_structured,  
 inputs=[  
 gr.Dropdown(  
 ["Requirements", "Design", "Development", "Testing", "Deployment", "Maintenance", "General/Other"],  
 label="Select SDLC Phase"  
 ),  
 gr.Textbox(lines=5, label="Enter your specific question or task:")  
 ],  
 outputs=[  
 gr.Textbox(label="Generated Output:"),  
 gr.Image(label="Generated Prompt-Based Image")  
 ],  
 title="Smart SDLC - AI Enhanced Software Development Lifecycle (Structured Demo)",  
 description="Interact with the ibm-granite/granite-3.3-2b-instruct model for SDLC tasks by selecting a phase.",  
 css="""  
 body {  
 background-color: CCFFFF;  
 color: #000000;  
 }  
 .gradio-container {  
 background-color: #CCFFCC !important;  
 border-radius: 10px;  
 padding: 20px;  
 box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.05);  
 }  
 textarea, input, select {  
 background-color: #ffffff !important;  
 color: #111111 !important;  
 font-weight: 500;  
 border: 1px solid #cccccc;  
 border-radius: 6px;  
 }  
 button {  
 background-color: #1976d2 !important;  
 color: white !important;  
 font-weight: normal;  
 border-radius: 8px;  
 }  
 """  
)  
  
# Launch the structured Gradio interface  
print("\nLaunching Structured Gradio Interface...")  
iface\_structured.launch(share=True)