import datetime

from transformers import AutoTokenizer, AutoModelForCausalLM

import torch

# Load IBM Granite model

model\_id = "ibm-granite/granite-3.3-2b-instruct"

tokenizer = AutoTokenizer.from\_pretrained(model\_id)

model = AutoModelForCausalLM.from\_pretrained(model\_id)

# Function to ask IBM Granite model

def ask\_granite(prompt):

inputs = tokenizer(prompt, return\_tensors="pt")

outputs = model.generate(\*\*inputs, max\_new\_tokens=200, temperature=0.7)

response = tokenizer.decode(outputs[0], skip\_special\_tokens=True)

return response.strip()

# Main health assistant function

def health\_assistant():

print("🤖 Hello! I'm your Health AI Assistant (powered by IBM Granite).")

name = input("What's your name? ")

print(f"Hi {name}, please describe your symptoms (comma-separated):")

symptoms\_input = input("Symptoms: ")

symptoms = [s.strip() for s in symptoms\_input.split(',')]

symptom\_text = ", ".join(symptoms)

print("\n🔍 Analyzing symptoms with AI...")

prompt = f"The user reports the following symptoms: {symptom\_text}. What are the most likely medical conditions or causes?"

granite\_response = ask\_granite(prompt)

print("\n📋 AI Suggested Conditions:")

print(granite\_response)

print("\n⚠️ Note: This is not a medical diagnosis. Please consult a healthcare professional.")

print(f"🕒 Timestamp: {datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")

# Run the assistant

if \_\_name\_\_ == "\_\_main\_\_":

health\_assistant()