# STEP 1: Install dependencies

!pip install transformers accelerate gradio causal-conv1d selective-state-update

# STEP 2: Download the model and tokenizer from Hugging Face

from transformers import AutoTokenizer, AutoModelForCausalLM

import os

HF\_TOKEN = ""  # 🔁 Replace with your actual token

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

model\_path = "/content/granite-model"

tokenizer = AutoTokenizer.from\_pretrained(model\_id, use\_auth\_token=HF\_TOKEN)

model = AutoModelForCausalLM.from\_pretrained(model\_id, device\_map="auto", use\_auth\_token=HF\_TOKEN)

tokenizer.save\_pretrained(model\_path)

model.save\_pretrained(model\_path)

# STEP 3: Load the model and tokenizer from local path

from transformers import pipeline

import torch

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

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

# Create generation pipeline

generator = pipeline("text-generation", model=model, tokenizer=tokenizer, device=0 if torch.cuda.is\_available() else -1)

def generate\_response(prompt):

    output = generator(prompt, max\_new\_tokens=256, do\_sample=True, temperature=0.7)

    return output[0]['generated\_text']

# STEP 4: Add Gradio interface

import gradio as gr

# Define Gradio function

def gradio\_chat(prompt):

    return generate\_response(prompt)

# Gradio UI

interface = gr.Interface(

    fn=gradio\_chat,

    inputs=gr.Textbox(lines=4, placeholder="Type your prompt here..."),

    outputs="text",

    title="🧠 Granite 3.3 - Local Chatbot",

    description="Ask anything and get responses from IBM Granite 3.3 LLM running locally."

)

# Launch the app

interface.launch(share=True)  # Set share=False if you don't want a public URL