TEAM ID- NM2025TMID11771

ARUN

ANAND

S.DEVENDRAN

T.DEVENDRAN

# -\*- coding: utf-8 -\*-

"""citizen.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1sJdbuT6MbL2IMAdRKDfcyZGn6f\_EgRJx

"""

!pip install transformers torch gradio -q

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=1024):

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 city\_analysis(city\_name):

prompt = f"Provide a detailed analysis of {city\_name} including:\n1. Crime Index and safety

statistics\n2. Accident rates and traffic safety information\n3. Overall safety assessment\n\nCity:

{city\_name}\nAnalysis:"

return generate\_response(prompt, max\_length=1000)

def citizen\_interaction(query):

prompt = f"As a government assistant, provide accurate and helpful information about the following

citizen query related to public services, government policies, or civic issues:\n\nQuery:

{query}\nResponse:"

return generate\_response(prompt, max\_length=1000)

# Create Gradio interface

with gr.Blocks() as app:

gr.Markdown("# City Analysis & Citizen Services AI")

with gr.Tabs():

with gr.TabItem("City Analysis"):

with gr.Row():

with gr.Column():

city\_input = gr.Textbox(

label="Enter City Name",

placeholder="e.g., New York, London, Mumbai...",

lines=1

)

analyze\_btn = gr.Button("Analyze City")

with gr.Column():

city\_output = gr.Textbox(label="City Analysis (Crime Index & Accidents)", lines=15)

analyze\_btn.click(city\_analysis, inputs=city\_input, outputs=city\_output)

with gr.TabItem("Citizen Services"):

with gr.Row():

with gr.Column():

citizen\_query = gr.Textbox(

label="Your Query",

placeholder="Ask about public services, government policies, civic issues...",

lines=4

)

query\_btn = gr.Button("Get Information")

with gr.Column():

citizen\_output = gr.Textbox(label="Government Response", lines=15)

query\_btn.click(citizen\_interaction, inputs=citizen\_query, outputs=citizen\_output)

app.launch(share=True)