```
importimportimportimport gradio as gr
import torch
from tratransformerstratransformersnsformersnsformers import AutoTokenizer, AutoModelForCausalLM
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, device_map="auto
def generate_response(prompt, max_length=1024):
    inputs = tokentokenizertokeniizerizer(prompt, return_tensors="pt", truncation=True, max_length=5
    if torch.cuda.is_available():
        inputs = {k: v.to(mmodelmmodelodelmmodelmmodelodelodelodel.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).replace(prompt, "").strip()
    return response
def city_analysis(city_name):
    prompt = f"Provide a detailed analysis of {city_name}..."
    return generate_response(prompt, max_length=1000)

def citizen_interaction(query):
    prompt = f"As a government assistant, provide accurate and helpful info about: {query}"
    return generate_response(prompt, max_length=1000)
with gr.Blocks() as app:
    gr.Markdown("# City Analysis & Citizen Services AI")

    with gr.Tabs():
        # City Analysis Tab
        with gr.TabItem("City Analysis"):
            with gr.Row():
                with gr.Column():
                    city_input = gr.Textbox(label="Enter City Name")
                    analyze_btn = gr.Button("Analyze City")
                with gr.Column():
                    city_output = gr.Textbox(label="City Analysis", lines=15)

            analyze_btn.click(city_analysis, inputs=city_input, outputs=city_output)

        # Citizen Services Tab
        with gr.TabItem("Citizen Services"):
            with gr.Row():
                with gr.Column():
                    citizen_query = gr.Textbox(label="Your Query")
                    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)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggin
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or d
  warnings.warn(
tokenizer_config.json:       8.88k/? [00:00<00:00, 230kB/s]
vocab.json:       777k/? [00:00<00:00, 7.94MB/s]
merges.txt:       442k/? [00:00<00:00, 6.26MB/s]
tokenizer.json:       3.48M/? [00:00<00:00, 48.5MB/s]
added_tokens.json: 100%                                                  87.0/87.0 [00:00<00:00, 1.80kB/s]
special_tokens_map.json: 100%                                            701/701 [00:00<00:00, 19.2kB/s]
config.json: 100%                                                        786/786 [00:00<00:00, 36.2kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json:       29.8k/? [00:00<00:00, 1.57MB/s]
Fetching 2 files: 100%                                                   2/2 [02:10<00:00, 130.52s/it]
model-00001-of-00002.safetensors: 100%                                   5.00G/5.00G [02:10<00:00, 28.3MB/s]
model-00002-of-00002.safetensors: 100%                                   67.1M/67.1M [00:08<00:00, 6.87MB/s]
Loading checkpoint shards: 100%                                          2/2 [00:20<00:00,  8.43s/it]
generation_config.json: 100%                                             137/137 [00:00<00:00, 13.1kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://b6fb7683e1905fdfad.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio de
```

City Analysis & Citizen Services AI

City Analysis Citizen Services

Enter City Name

Analyze City

City Analysis