

```

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 sta
    return generate_response(prompt, max_length=1000)

def citizen_interaction(query):
    prompt = f"As a government assistant, provide accurate and helpful information about the follow
    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():
        # City Analysis Tab
        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)

# Citizen Services Tab
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)

```



Loading checkpoint shards: 100%

2/2 [00:06<00:00, 6.64s/it]

Colab notebook detected. To show errors in colab notebook, set debug=True in launch()

\* Running on public URL: <https://cf9667ac725de22884.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

Your Query

How to i apply death certificate

Get Information

Government Response

Dear Citizen,

Thank you for your question regarding the application process for a death certificate. The process may vary slightly depending on your location, so I'll provide a general outline that applies to many jurisdictions in the United States.

1. **\*\*Obtain the Death Certificate:\*\***
  - The healthcare facility, hospital, or funeral home where the death occurred typically issues the death certificate. It usually contains essential information such as the deceased's full name, date of birth, cause of death, and date of death.
2. **\*\*Notifv the Appnropriate Authorities:\*\***