Project Documentation

1. Introduction

Project Title : Citizen AI – Intelligent Citizen Engagement Platform

Team Members : Varshini A, Shruthi Chopra D, Neethu M, Yukti M Shah

2. Project Overview

- Purpose :

Citizen AI is designed to improve citizen engagement with government services using IBM's Granite Large Language Models (LLMs). It provides quick, conversational responses to queries, analyzes public sentiment, and generates dashboards for officials to support informed decision-making.

- Features :

  - Conversational interface via Gradio
  - Integration with IBM Granite models on Hugging Face
  - Lightweight deployment using Google Colab (T4 GPU)
  - Public sentiment tracking and visualization
  - GitHub integration for code sharing and version control

3. Architecture

- Frontend (Gradio): Simple and accessible UI for citizen queries
- Backend (Google Colab Runtime): Python + Transformers, Torch, Gradio
- Model Integration (IBM Granite): Uses granite-3.2-2b-instruct for fast inference
- Version Control (GitHub): Stores project notebook, scripts, and outputs

4. Setup Instructions

- Prerequisites :

  - Python (Google Colab runtime)
  - Hugging Face account for IBM Granite models
  - Gradio library
  - GitHub account for versioning

- Installation :

  - Open Google Colab → New Notebook

    - Change runtime → Select T4 GPU
    - Install dependencies: !pip install transformers torch gradio -q
    - Load Granite model from Hugging Face
    - Run notebook cells to start Gradio app

5. Folder Structure

- citizen_ai.ipynb – Main project notebook
- requirements.txt – Dependencies list
- README.md – Project overview (GitHub)
- output/ – Application screenshots / reports

6. Running the Application

- Launch Colab notebook
- Select T4 GPU runtime
- Install dependencies
- Run notebook cells
- Gradio generates a shareable URL → Open app in browser

7. API Documentation

- POST /ask – Accepts citizen query and returns AI-generated response
- GET /sentiment – Retrieves sentiment analysis dashboard (future)

8. Authentication

- Presently runs in open mode for demo
- Future security upgrades:
- - Hugging Face API keys
- - JWT-based authentication for access control

9. User Interface

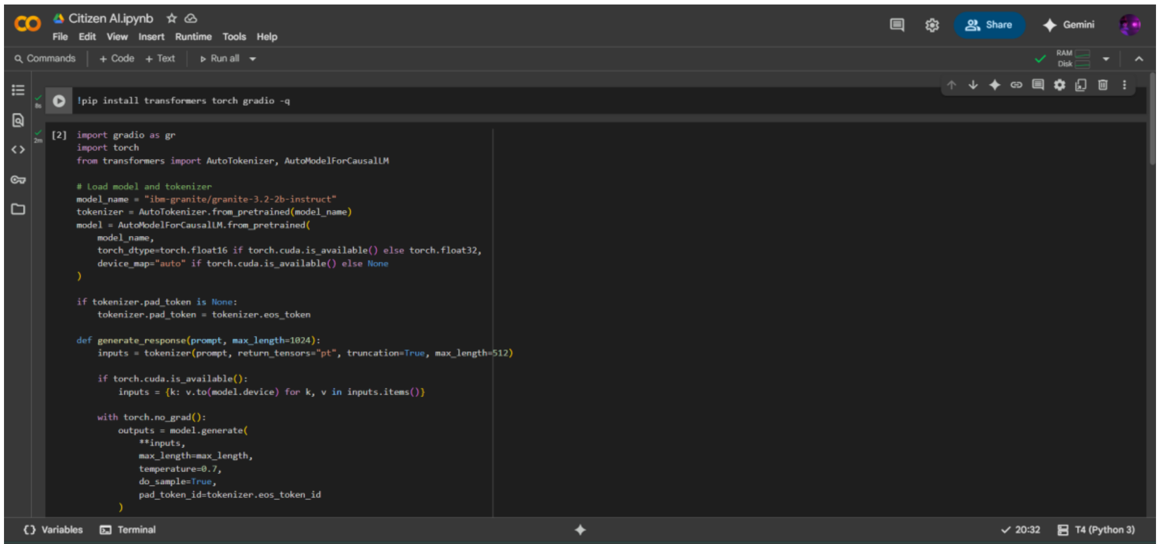
- Minimal Gradio interface with text input and response area
- Auto-generated shareable link from Google Colab
- Works on any browser

10. Testing

- Unit Testing: Verified installation and model loading in Colab
- Manual Testing: Citizen queries tested via Gradio UI
- Edge Cases: Long queries, irrelevant prompts tested

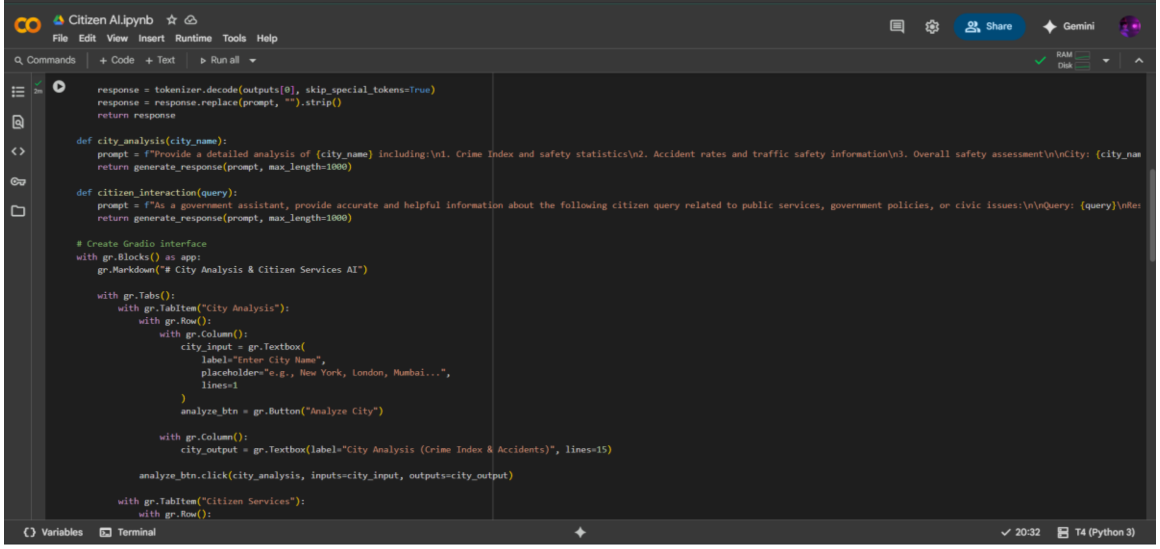
- Validation: Confirmed Hugging Face integration functions consistently

11. Screenshots

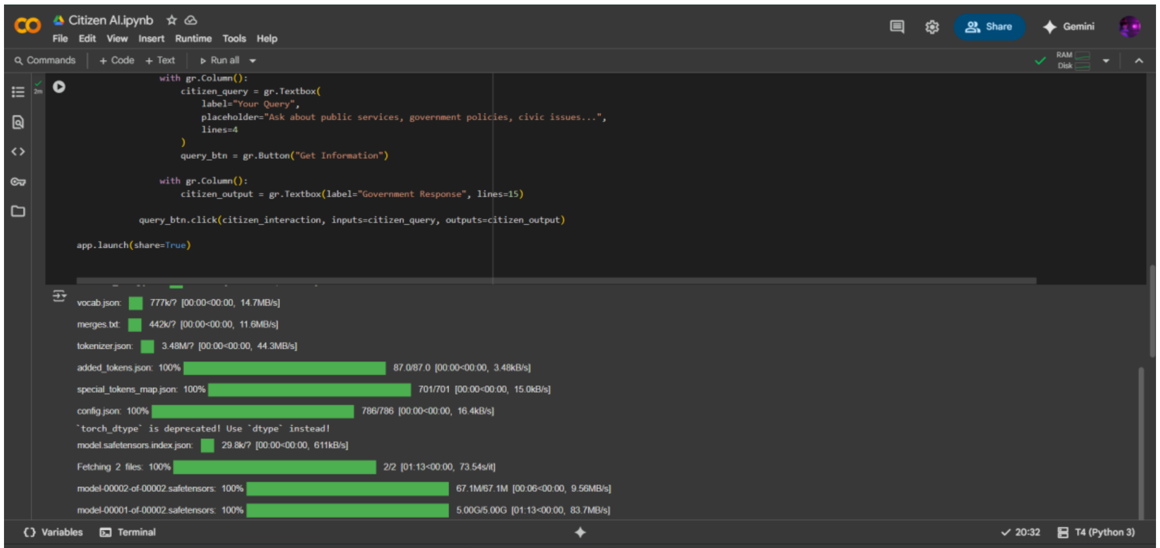


```
!pip install transformers torch gradio -q
```

```
[2] 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
            )
```

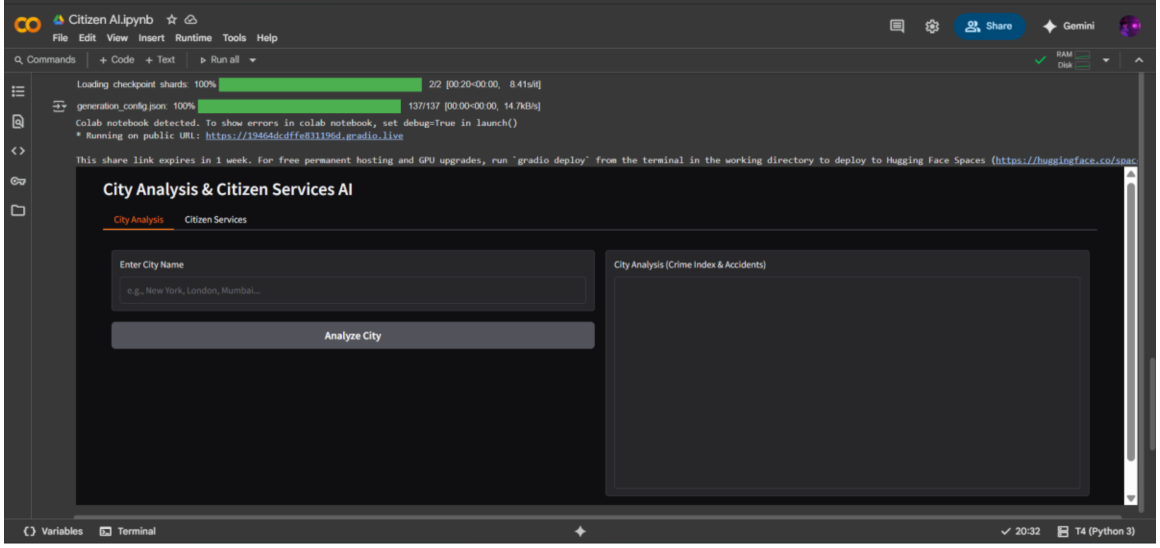


```
        response = tokenizer.decode(outputs[0], skip_special_tokens=True)
        response = response.replace(prompt, "").strip()
        return response

    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_nam
        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}\nRes
        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():
```


Citizen AI.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share Gemini
Variables Terminal 20:32 T4 (Python 3)

```
            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)
```

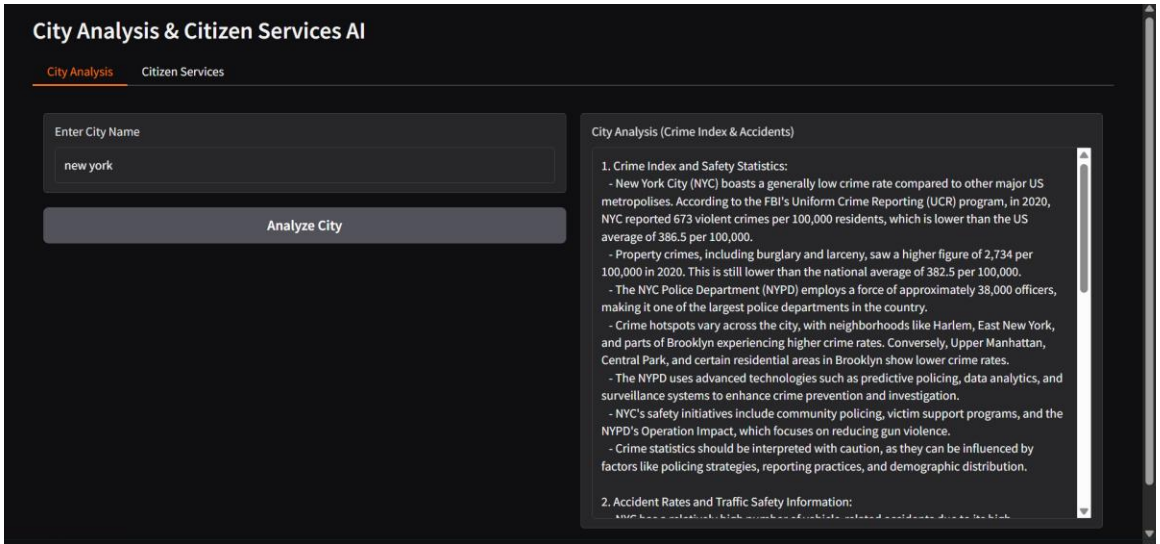
```
vocab.json: 🟩 777k/? [00:00<00:00, 14.7MB/s]
merges.txt: 🟩 442k/? [00:00<00:00, 11.6MB/s]
tokenizer.json: 🟩 3.48M/? [00:00<00:00, 44.3MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 3.48kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 15.0kB/s]
config.json: 100% 786/786 [00:00<00:00, 16.4kB/s]
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json: 🟩 29.8k/? [00:00<00:00, 611kB/s]
Fetching 2 files: 100% 2/2 [01:13<00:00, 73.54s/it]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:06<00:00, 9.56MB/s]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:13<00:00, 83.7MB/s]
```


Citizen AI.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share Gemini
Variables Terminal 20:32 T4 (Python 3)

```
Loading checkpoint shards: 100% 2/2 [00:20<00:00, 8.41s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 14.7kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://19464dcdffe831196d.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (https://huggingface.co/spac
```

City Analysis & Citizen Services AI

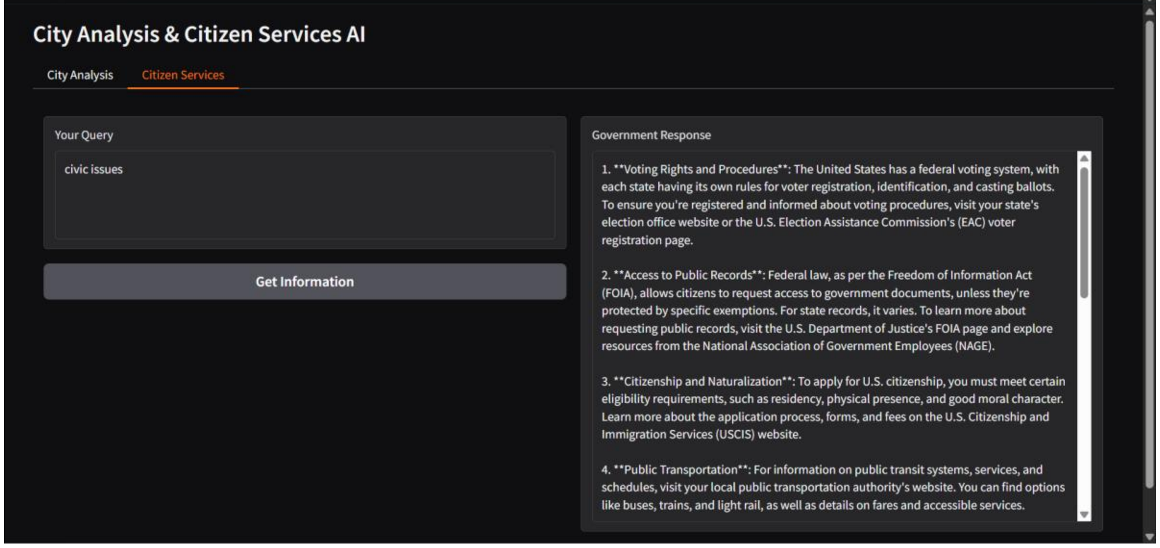
City Analysis
Citizen Services
Enter City Name
e.g., New York, London, Mumbai...
Analyze City
City Analysis (Crime Index & Accidents)


City Analysis & Citizen Services AI
City Analysis
Citizen Services
Enter City Name
Analyze City
City Analysis (Crime Index & Accidents)

new york

1. Crime Index and Safety Statistics:
  - New York City (NYC) boasts a generally low crime rate compared to other major US metropolises. According to the FBI's Uniform Crime Reporting (UCR) program, in 2020, NYC reported 673 violent crimes per 100,000 residents, which is lower than the US average of 386.5 per 100,000.
  - Property crimes, including burglary and larceny, saw a higher figure of 2,734 per 100,000 in 2020. This is still lower than the national average of 382.5 per 100,000.
  - The NYC Police Department (NYPD) employs a force of approximately 38,000 officers, making it one of the largest police departments in the country.
  - Crime hotspots vary across the city, with neighborhoods like Harlem, East New York, and parts of Brooklyn experiencing higher crime rates. Conversely, Upper Manhattan, Central Park, and certain residential areas in Brooklyn show lower crime rates.
  - The NYPD uses advanced technologies such as predictive policing, data analytics, and surveillance systems to enhance crime prevention and investigation.
  - NYC's safety initiatives include community policing, victim support programs, and the NYPD's Operation Impact, which focuses on reducing gun violence.
  - Crime statistics should be interpreted with caution, as they can be influenced by factors like policing strategies, reporting practices, and demographic distribution.

2. Accident Rates and Traffic Safety Information:



civic issues

1. **Voting Rights and Procedures**: The United States has a federal voting system, with each state having its own rules for voter registration, identification, and casting ballots. To ensure you're registered and informed about voting procedures, visit your state's election office website or the U.S. Election Assistance Commission's (EAC) voter registration page.

2. **Access to Public Records**: Federal law, as per the Freedom of Information Act (FOIA), allows citizens to request access to government documents, unless they're protected by specific exemptions. For state records, it varies. To learn more about requesting public records, visit the U.S. Department of Justice's FOIA page and explore resources from the National Association of Government Employees (NAGE).

3. **Citizenship and Naturalization**: To apply for U.S. citizenship, you must meet certain eligibility requirements, such as residency, physical presence, and good moral character. Learn more about the application process, forms, and fees on the U.S. Citizenship and Immigration Services (USCIS) website.

4. **Public Transportation**: For information on public transit systems, services, and schedules, visit your local public transportation authority's website. You can find options like buses, trains, and light rail, as well as details on fares and accessible services.

12. Known Issues

- Free Colab tier has limited GPU availability
- Hugging Face model download requires internet
- No persistence across Colab restarts

13. Future Enhancements

- Deploy with IBM Watsonx API for enterprise use
- Use FastAPI backend for production
- Sentiment dashboard with real-time visualization
- Multi-language citizen query support
- Persistent cloud-based hosting