CITIZEN AI WITH IBM

Project Documentation

– Introduction

• Project Title: Citizen AI with IBM

•Team Leader: BHARAKATHUL ELMIYA S

• Team Member: PAARKAVI M

•Team Member: ALRIFA H

• Team Member: DHILSHA B

• Team Member: YOPIKASRI S

– Project Overview

• Purpose: The purpose of Citizen Al is to provide quick and reliable information about government services and civic issues. It also collects citizen feedback and presents it through simple dashboards for officials, enabling smarter decision-making and better citizen engagement.

•Features:

-Conversational Interface: Citizens can interact naturally and get instant answers.

- Sentiment Analysis: Tracks public opinion and feedback.

- Citizen Feedback Loop: Helps governments listen to people's voices and adapt policies.

- Dashboards: Provides visual insights for officials to make data-driven decisions.

- gradio Interface: Simple web-based application for demonstration.

3. Architecture

Frontend (Gradio):A user-friendly interface for citizens to interact with the Al system.

Backend (Google Colab + Python): Hosts the application, runs the Al model, and processes inputs.

LLM Integration (1BM Granite Models): The core Al engine that generates responses and summaries.

Version Control (GitHub): Ensures project files are stored, tracked, and shared easily.

4. Setup Instructions

Prerequisites:

- Python Programming Knowledge
- Gradio Framework
- IBM Granite Model Access (via Hugging Face)
- Google Colab with T4 GPU
- GitHub Account Steps:

1. Access the Naan Mudhalvan Smart internz portal.
2. Choose an IBM Granite model from Hugging Face.
3. Run the application in Google Colab with required libraries.
4. Upload final project files to GitHub.
5. Folder Structure

• app/- Backend logic and integration.

• ui/ – Gradio app interface files.

• citizen AI.py– Main application file.

• model loader.py – Handles IBM Granite model integration.

• dashboard.py – Visualization of citizen feedback.

6. Running the Application

1. Open Google Colab and load the project notebook.
2. Install dependencies and configure runtime with GPU.
3. Run the notebook cells to start the Gradio app.
4. Access the provided link to interact with Citizen Al.

7. API Documentation

Citizen Al provides endpoints for:

- Asking questions about government services.
- Uploading feedback for sentiment analysis.
- Viewing summarized policies.
- Accessing dashboards and reports.

8. Authentication

For the demo, Citizen Al runs in an open setup. In real deployments, authentication methods like API keys, OAuth2, and role-based access would be used.

9. User interface

The Gradio interface is clean and simple with:

-A chat box for queries.

- Dashboard views for officials.

- Options to upload feedback and documents.

10. Testing

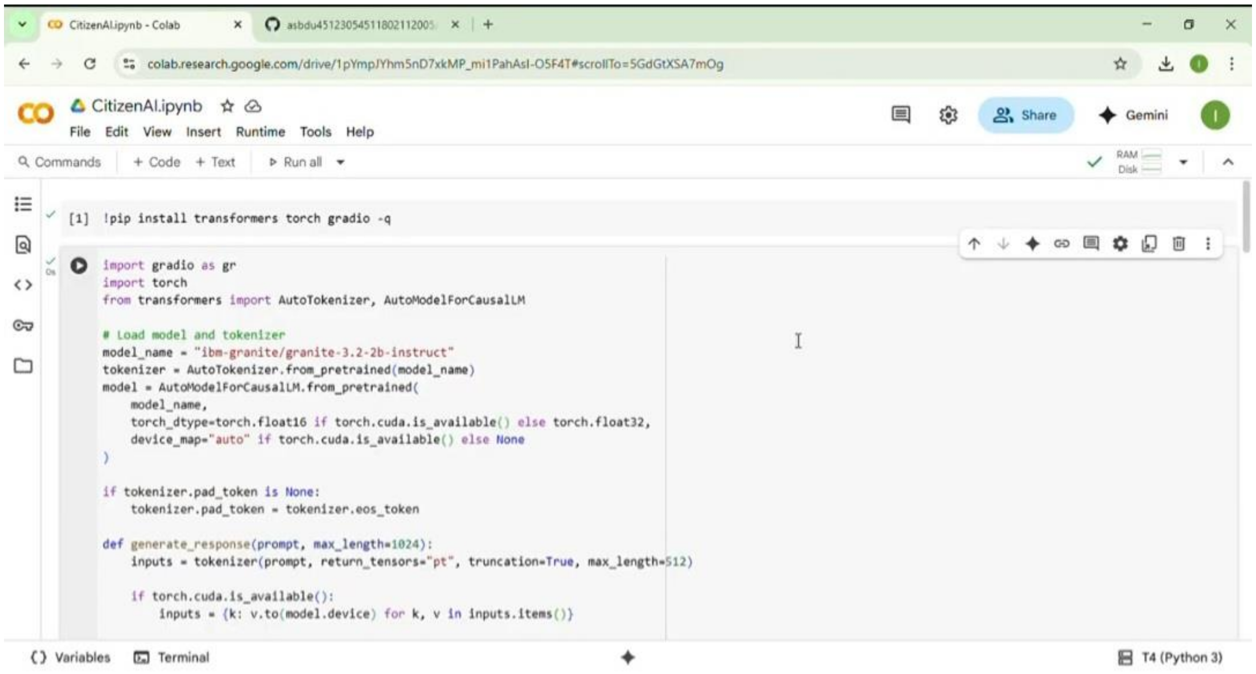
Testing included:

-Unit testing for Al responses.

- Manual testing of the Gradio interface.

- Edge case handling with unexpected inputs.

11.Screenshots


CitizenAI.ipynb - Colab
asbdu45123054511802112005
colab.research.google.com/drive/1pYmpJYhm5nD7xkMP_mi1PahAsl-O5F4T#scrollTo=5GdGtXSA7mOg
CitizenAI.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text Run all
RAM Disk
Variables Terminal
T4 (Python 3)

```
[1] !pip install transformers torch gradio -q
```

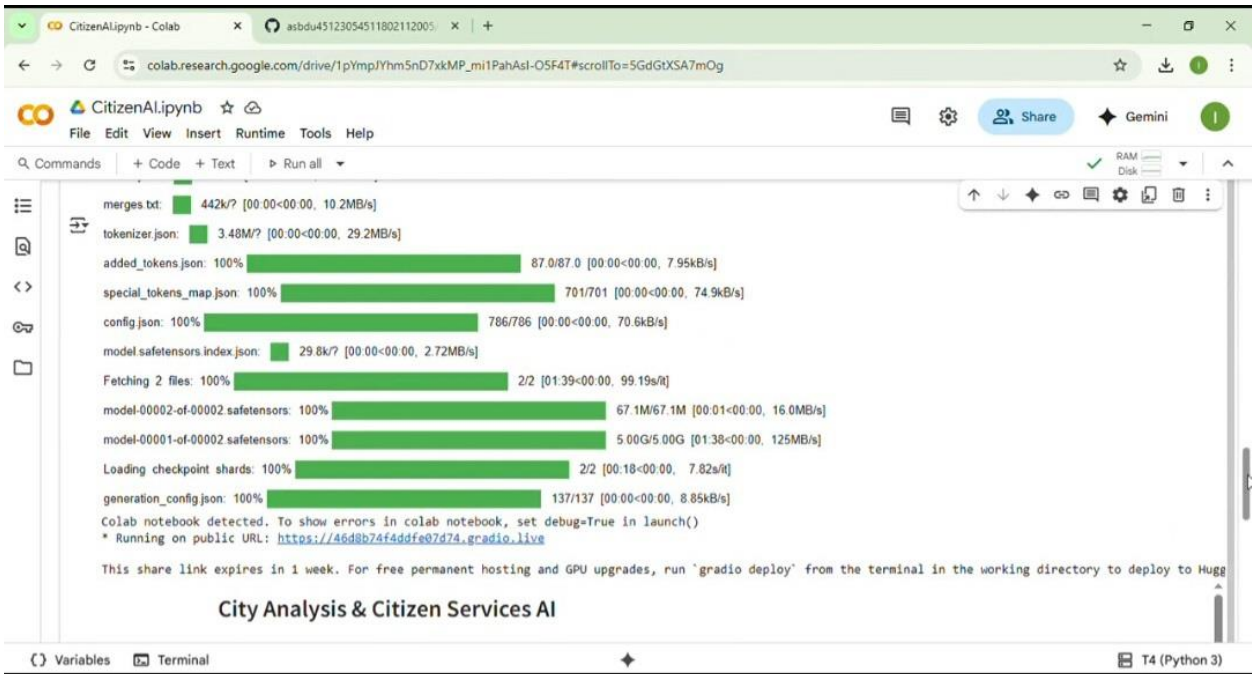
```
import gradio as gr
import torch
from transformers import AutoTokenizer, AutoModelForCausalLM

# 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()}
```

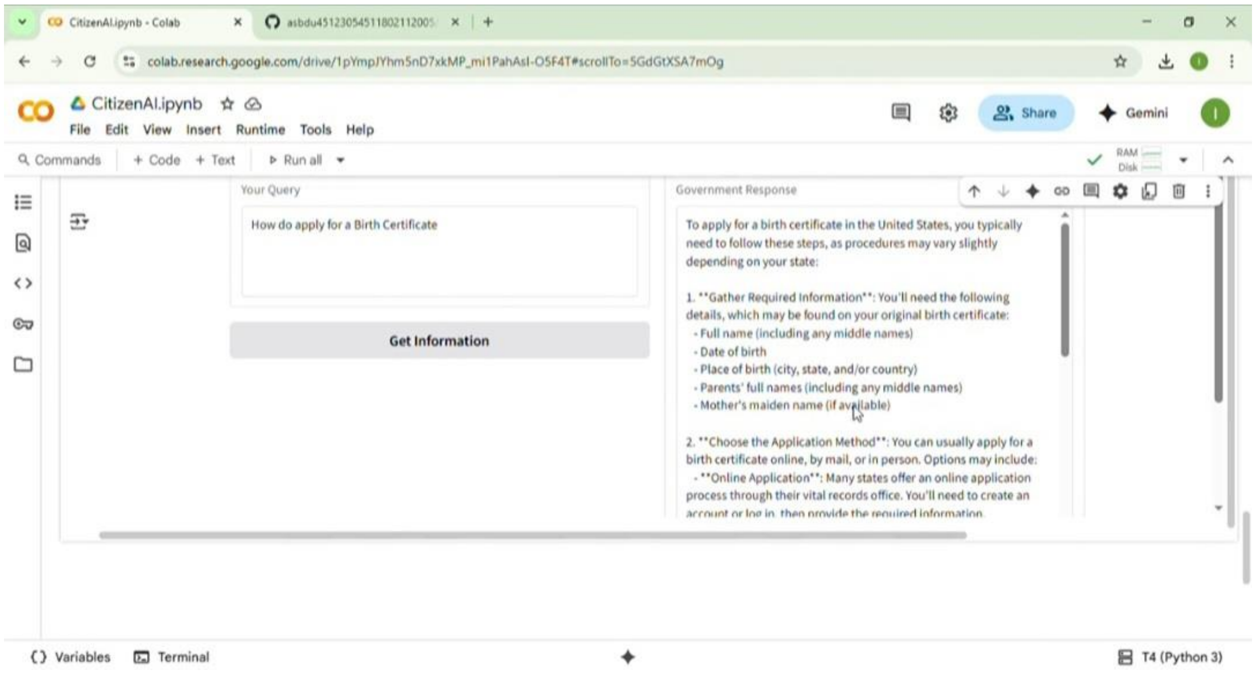


```
merges.txt:  442k/? [00:00<00:00, 10.2MB/s]
tokenizer.json:  3.48M/? [00:00<00:00, 29.2MB/s]
added_tokens.json: 100% 87.0/87.0 [00:00<00:00, 7.95kB/s]
special_tokens_map.json: 100% 701/701 [00:00<00:00, 74.9kB/s]
config.json: 100% 786/786 [00:00<00:00, 70.6kB/s]
model.safetensors.index.json:  29.8k/? [00:00<00:00, 2.72MB/s]
Fetching 2 files: 100% 2/2 [01:39<00:00, 99.19s/it]
model-00002-of-00002.safetensors: 100% 67.1M/67.1M [00:01<00:00, 16.0MB/s]
model-00001-of-00002.safetensors: 100% 5.00G/5.00G [01:38<00:00, 125MB/s]
Loading checkpoint shards: 100% 2/2 [00:18<00:00, 7.82s/it]
generation_config.json: 100% 137/137 [00:00<00:00, 8.85kB/s]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://46d8b74f4ddfe07d74.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 Hugg
```

City Analysis & Citizen Services AI

CitizenAI.ipynb - Colab
colab.research.google.com/drive/1pYmpJYhm5nD7xkMP_mi1PahAsl-O5F4T#scrollTo=5GdGtXSA7mOg
CitizenAI.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
Gemini
Variables
Terminal
T4 (Python 3)



Your Query

How do apply for a Birth Certificate

Get Information

Government Response

To apply for a birth certificate in the United States, you typically need to follow these steps, as procedures may vary slightly depending on your state:

1. **Gather Required Information**: You'll need the following details, which may be found on your original birth certificate:
  - Full name (including any middle names)
  - Date of birth
  - Place of birth (city, state, and/or country)
  - Parents' full names (including any middle names)
  - Mother's maiden name (if available)

2. **Choose the Application Method**: You can usually apply for a birth certificate online, by mail, or in person. Options may include:
  - **Online Application**: Many states offer an online application process through their vital records office. You'll need to create an account or log in, then provide the required information

CitizenAI.ipynb - Colab
colab.research.google.com/drive/1pYmpJYhm5nD7xkMP_mi1PahAsl-O5F4T#scrollTo=5GdGtXSA7mOg
CitizenAI.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
Gemini
Variables
Terminal
T4 (Python 3)

12. Known Issues

• Limited scope due to demo environment.

•Requires internet for Colab runtime and Hugging Face model.

13. Future Enhancements

- Integrate advanced analytics for deeper insights.
- Expand support for multiple languages.
- Deploy on cloud platforms for real-world scalability.