Citizen AI:- Intelligent Citizen Engagement Platform

Team Members

Team Leader: VIJAYALAKSHMI.C

Team member : VARSHINI.R

Introduction

Citizen AI is an AI-driven platform designed to support intelligent citizen engagement by analyzing city safety metrics and addressing citizen queries about public services and government policies. The system leverages Natural Language Processing (NLP) models with Gradio for interactive interfaces, providing citizens with real-time insights and assistance.

Objectives

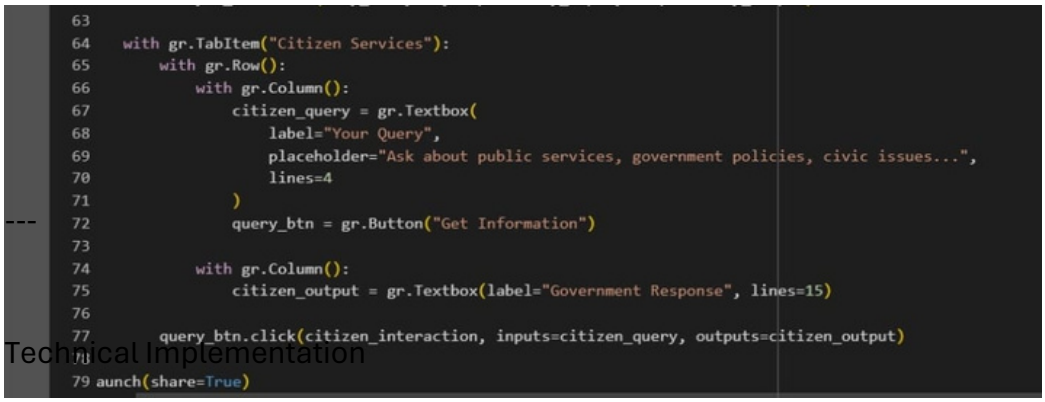
1. To provide detailed city analysis including crime index, accident rates, and overall safety.

2. Citizen Services Module

Accepts citizen queries related to public services, policies, or civic concerns.

Provides AI-generated government-style responses.

Ensures responses are accurate, structured, and citizen-friendly.



```
63
64  with gr.TabItem("Citizen Services"):
65      with gr.Row():
66          with gr.Column():
67              citizen_query = gr.Textbox(
68                  label="Your Query",
69                  placeholder="Ask about public services, government policies, civic issues...",
70                  lines=4
71              )
72              query_btn = gr.Button("Get Information")
73
74          with gr.Column():
75              citizen_output = gr.Textbox(label="Government Response", lines=15)
76
77      query_btn.click(citizen_interaction, inputs=citizen_query, outputs=citizen_output)
78
79 aunch(share=True)
```

Technical Implementation

Model Used: ibm-granite/granite-3.2-2b-instruct

Frameworks:

Transformers (Hugging Face) for model loading and text generation.

Gradio for building interactive user interfaces with tabs.

Device Support:

Utilizes GPU (torch.float16) if available for faster inference.

Falls back to CPU (torch.float32) when GPU is unavailable.

Interface Features:

Tab 1: City Analysis with input and analysis output.

Tab 2: Citizen Services with query input and government response output.

Shareable web interface with app.launch(share=True).

```python
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():
```

Workflow

1. User Input:

Enters a city name OR a query.

2. Processing:

AI model generates response using structured prompts.

3. Output:

Displays crime & accident analysis for cities.

Provides government-like responses for citizen queries.

Expected Output

City Analysis Tab:

Displays an AI-written report on safety, crime index, and accidents.

Citizen Services Tab:

Provides AI-generated, informative, and structured answers to public queries.

Conclusion

The Citizen AI platform offers an intelligent, user-friendly solution for civic engagement and city safety analysis. By integrating AI-driven insights with an interactive interface, it empowers citizens to make informed decisions and enhances communication between the government and the public.