CITIZEN-AI

INTELLIEGENT CITIZEN ENGAGEMENT PLATFORM)

TEAM LEADER : DHANUSIYA

TEAM MEMBERS : RAGAVI R

ABINAYA P

JEEVITHA S

NITHIYASRI K

INTRODUCTION:

Citizen AI is an innovative digital platform designed to strengthen the relationship between citizens and governance through artificial intelligence. It aims to provide intelligent, transparent, and inclusive services that allow people to actively participate in decision-making, voice their needs, and access government resources more efficiently.

By using advanced AI technologies such as natural language processing, data analytics, and machine learning, Citizen AI can understand citizen concerns, provide personalized recommendations, predict community needs, and help authorities deliver timely solutions**.**

PROJECT DESCRTIPTION:

AI-Powered Virtual Assistants for answering queries in multiple languages . Grievance Redressal System with automated tracking and escalation mechanism. Data Analytics Dashboard for policymakers to understand citizen needs and sentiments Multichannel Access (web, mobile app, social media integration, IVR).Personalized Recommendations for government schemes and welfare program. Sentiment Analysis to capture public opinion and improve service design.

OBJECTIVES:

Enhance Communication

- Provide citizens with multiple channels (web, mobile app, chatbot, SMS) to interact with government and public service providers.

Improve Service Delivery

- Use AI and automation to process citizen queries, complaints, and requests quickly and efficiently.

Data-Driven Decision Making

- Collect, analyze, and visualize citizen feedback and engagement data to help authorities make informed policy decisions.

Personalized Citizen Experience

- Maintain citizen profiles and past interactions to deliver more relevant, customized responses.

Transparency & Accountability

- Ensure citizens can track the status of their requests and issues, improving trust in governance.

TOOLS:

Ensure Python (3.7+) and pip are installed for managing project dependencies.

Install Flask and Dependencies:

Use pip to install Flask and any other required backend libraries:

pip install Flask

Install AI/ML Libraries:

Install the necessary libraries for AI model integration:

pip install torch transformers accelerate bitsandbytes

(Ensure you install the correct PyTorch version for your CUDA setup if using a GPU).

PROJECT OVERVIEW:

An intelligent citizen engagement platform is a digital tool that uses AI and other technologies to improve two-way communication between governments and citizens, making participation in decision-making, policy

development, and access to services more efficient, personalized, and inclusive. These platforms feature AI-powered chatbots for quick responses to civic inquiries, sentiment analysis to gauge public opinion, interactive dashboards for visualizing trends, and support for multiple languages to break down barriers. Key goals include increasing transparency, building community trust, and enabling citizens to actively shape their communities by providing feedback and contributing to solutions in a scalable way.

- **AI-Powered Chatbots:**

- Provide instant, context-aware answers to common questions about public services, procedures, and documents**.**

Sentiment Analysis:

Analyze citizen feedback to understand public mood and concerns**.**

Interactive Dashboards:

Offer visual insights into feedback trends, engagement levels, and sentiment data.

Personalized Services:

Leverage AI and data to provide citizens with tailored information and access to relevant programs.

Multilingual Support:

Use AI to translate and present information in various languages, improving accessibility for diverse communities.

Mobile-First Design:

Ensure accessibility and user-friendliness through mobile-friendly interfaces.

RUNING IN THE APPLICTION:

An intelligent citizen engagement platform is a digital application that uses technology, including AI, to enable two-way communication and collaboration between citizens and government or community bodies for improved governance, decision-making, and service delivery. Examples include India's MyGov platform, which facilitates policy discussions and seeks public opinion, and AI-powered chatbots like Citizen AI that offer 24/7 support for citizen inquiries. These platforms help governments understand citizen needs, foster trust, increase transparency, and promote participatory development.

USER INTERFACE:

Citizen Login & Profile

Login / Register: via Email, Phone number, Aadhaar, or Social logins.

Profile Section: personal info, history of complaints, feedback given, participation in surveys.

Complaint & Feedback Module

 Form-based Input: Citizens can submit issues (roads, electricity, water, waste, etc.).

 Attachment option: Upload images/videos of issues.

 Geotagging: Auto-capture location with GPS.

 Track Status: Timeline view (Submitted → In Progress → Resolved).

Analytics & Insights (Citizen View)

 Personalized dashboard showing:

Number of issues reported.

Status of complaints.

Community participation score.
Administrator Dashboard (Govt. View)

Monitor citizen complaints in real-time.

Categorize issues by type & location.

Data visualization (heatmaps of problem areas, response times).

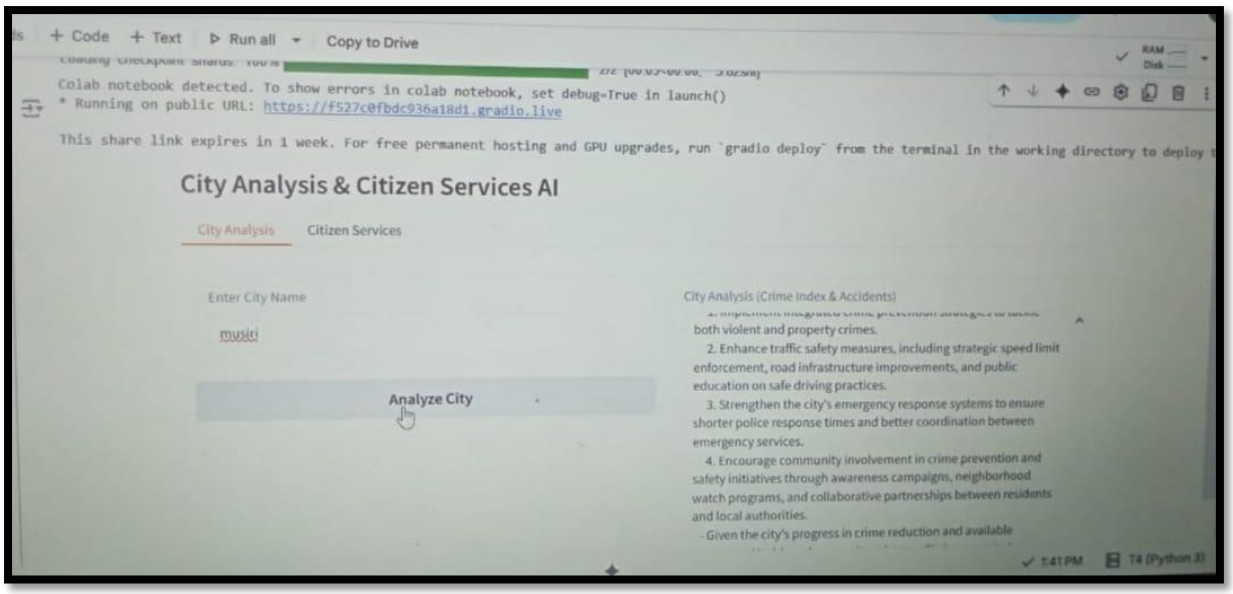
PROGRAM:

```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():
        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
        )
```

```
31
32 esponse = tokenizer.decode(outputs[0], skip_special_tokens=True)
33 esponse = response.replace(prompt, "").strip()
34 eturn response
35
36 ity_analysis(city_name):
37 rompt = f"Provide a detailed analysis of {city_name} including:\n1. Crime Index and safety statistics\n2. Accident ra
38 eturn generate_response(prompt, max_length=1000)
39
40 itizen_interaction(query):
41 rompt = f"As a government assistant, provide accurate and helpful information about the following citizen query relat
42 eturn generate_response(prompt, max_length=1000)
43
44 ate Gradio interface
45 gr.Blocks() as app:
46 r.Markdown("# City Analysis & Citizen Services AI")
47
48 ith gr.Tabs():
49     with gr.TabItem("City Analysis"):
50         with gr.Row():
51             with gr.Column():
52                 city_input = gr.Textbox(
53                     label="Enter City Name",
54                     placeholder="e.g., New York, London, Mumbai...",
55                     lines=1
56                 )
57                 analyze_btn = gr.Button("Analyze City")
58
59             with gr.Column():
60                 city_output = gr.Textbox(label="City Analysis (Crime Index & Accidents)", lines=15)
61
62         analyze_btn.click(city_analysis, inputs=city_input, outputs=city_output)
```

```
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)
```

OUTPUT: (GENERATING EXPLANATION)



CONCLUSION:

The Intelligent Citizen Engagement Platform builds a strong connection between government and citizens. It ensures transparency, quick service delivery, and active participation in decision-making. Citizens gain easy access and a voice in governance, while governments improve efficiency and trust. Overall, it is a transformative system for inclusive and sustainable governance.