Citizen AI:Intelligent Citizen Engagement Platform

Team Leader:

Tarika S

Team Members:

Tawffeqa U

Thulasi Sre H G

1.SYSTEM ANALYSIS

1.1Introduction

This program is an **AI-powered web application** designed to assist with **city analysis** and **citizen services**. It leverages the **IBM Granite 3.2 Instruct model** integrated through **Hugging Face Transformers** and deployed via a **Gradio interface**.

The system provides two main services:

1. **City Analysis** – Generates detailed reports on a given city, focusing on **crime index** and **accident statistics**.
2. **Citizen Services** – Answers public queries related to **government policies, civic issues, and public services** in a clear and informative manner.

By combining natural language understanding and text generation, the program aims to deliver an **interactive, user-friendly, and accessible tool** for exploring city-level insights and improving communication between citizens and governance systems.

1.2Objective

The objective of this program is to develop an **AI-powered assistant** that:

1. **Analyzes cities** by generating insights on crime index, safety, and accident statistics.
2. **Supports citizens** by answering queries related to government services, civic issues, and public policies.
3. Provides a **simple and interactive interface** using Gradio for easy accessibility.
4. Utilizes the **IBM Granite AI model** to ensure natural language understanding and accurate response generation.
5. Enhances **citizen engagement** and **data-driven decision-making** through AI-driven text analysis.

1.3Existing System

In the existing system, citizens often rely on **manual searches, government portals, or static reports** to access information about cities, public services, or government policies. These systems usually provide **generalized data**, lack **personalized interaction**, and require users to navigate through multiple sources to find relevant information.

Current approaches also do not offer **real-time AI-driven assistance**, making it difficult for citizens to quickly analyze aspects like **crime statistics, accident rates, or civic policies**. Furthermore, traditional platforms lack **natural language interaction**, which limits accessibility for people unfamiliar with technical or bureaucratic processes.

As a result, the existing solutions are **time-consuming, fragmented, and less user-friendly**, highlighting the need for an **AI-powered interactive system** like the one proposed in this program.

1.4Proposed System

The proposed system introduces an **AI-powered web application** that simplifies access to **city insights** and **citizen services** through natural language interaction. By leveraging the **IBM Granite 3.2 Instruct model** with **PyTorch** and **Gradio**, the system provides:

- **City Analysis** → Generates AI-driven reports on crime index, accident rates, and safety statistics for any given city.
- **Citizen Services** → Offers instant responses to queries related to government policies, civic issues, and public services.
- **Interactive Interface** → A user-friendly Gradio interface with dedicated tabs for city analysis and citizen interaction.
- **AI-Powered Responses** → Context-aware answers that reduce dependency on manual searches and fragmented portals.

This system overcomes the limitations of existing solutions by providing a **centralized, intelligent, and conversational platform** that enhances **citizen engagement, accessibility, and decision-making**.

1.5Tools and Technologies Used

1. Programming Language

- **Python** → Core language used to implement the application, handle AI model integration, and build the web interface.

2. AI/ML Frameworks

- **PyTorch** → For deep learning model execution and efficient handling of GPU/CPU resources.

- **Hugging Face Transformers** → To load and use the **IBM Granite 3.2 Instruct model** for natural language processing and text generation.

3. Interface Development

- **Gradio** → To create an interactive and user-friendly web interface with tabs for *City Analysis* and *Citizen Services*.

4. Model

- **IBM Granite 3.2B Instruct** → A large language model (LLM) used for generating intelligent, context-aware responses.

5. Hardware & Runtime Environment

- **CPU/GPU Support**:
  - GPU (if available) for faster inference using FP16 precision.
  - CPU fallback for general execution.
- **Torch Device Management** → Automatic mapping to GPU/CPU using device_map.

6. Deployment & Accessibility

- **Gradio Launch (share=True)** → Enables local hosting with an option to generate a shareable public link.

2.SYSTEM DESIGN

2.1Project Description

The **AI-Powered City Analysis and Citizen Services Assistant** is an interactive web-based application that uses **artificial intelligence** to enhance civic engagement and information accessibility. The system integrates the **IBM Granite 3.2B Instruct model** through **Hugging Face Transformers** and provides a simple, intuitive interface using **Gradio**.

The project has two major functionalities:

1. **City Analysis**
   - Accepts a city name as input.
   - Generates detailed insights into **crime index, safety statistics, and accident rates**.
2. **Citizen Services**
   - Accepts queries related to **government policies, civic issues, or public services**.
   - Provides accurate and context-aware responses in natural language.

The system is designed to run on both **CPU and GPU**, ensuring flexibility and efficiency. GPU support allows faster response generation using **FP16 precision**, while CPU ensures broad compatibility.

By combining **AI-driven text generation** with a **user-friendly interface**, this project helps reduce the gap between citizens and governance by making information **accessible, interactive, and centralized**.

2.2Testing

1. Unit Testing

- **Function Testing**:
  - generate_response() → Verified that the function generates responses without errors for various prompts.
  - city_analysis() → Tested with city names (e.g., “Mumbai”, “New York”) to ensure output contains crime and accident-related information.
  - citizen_interaction() → Checked with queries like *“What are education schemes?”* and *“Explain traffic rules?”* for relevance.

2. Integration Testing

- Verified smooth integration between:
  - **Model & Tokenizer** → Ensured text is encoded/decoded correctly.
  - **Backend Functions & Gradio UI** → Confirmed inputs/outputs flow correctly between interface and functions.

3. System Testing

- Tested the **Gradio interface** for:
  - Proper loading of **City Analysis** and **Citizen Services** tabs.
  - Execution of queries with multiple users.
  - Response length control using max_length.

4. Performance Testing

- Tested on **CPU-only environment** → Verified slower but functional response.
- Tested on **GPU-enabled environment** → Ensured faster inference with reduced latency.

5. User Testing (UAT)

- Checked usability with non-technical users to confirm:
  - Inputs are easy to provide.
  - Outputs are understandable.
  - Interface is user-friendly and responsive.

2.3Sample Output

1. City Analysis

Input:

City Name: Mumbai

Output (Generated):

```
City Analysis for Mumbai:

1. Crime Index and Safety Statistics:
   - Mumbai has a moderate crime index compared to other major cities in India.
   - Common issues include petty theft, scams, and traffic violations.
   - Violent crimes are less frequent but do occur in certain high-density areas.
```

2. Accident Rates and Traffic Safety:
- Mumbai faces a high rate of road accidents due to heavy traffic congestion.
- Peak accident hours are during office commute times (8–10 AM, 6–8 PM).
- Government initiatives include stricter traffic monitoring and road safety campaigns.

---

2. Citizen Services

Input:

Query: What are the government schemes for higher education in India?

Output (Generated):

Government Response:

The Government of India provides several schemes for higher education, including:

1. National Scholarship Portal (NSP) – A single window for applying to multiple central and state-funded scholarships.
2. Central Sector Scheme of Scholarships – Supports meritorious students from low-income families.
3. AICTE and UGC Scholarships – For engineering, management, and postgraduate studies.
4. Study Abroad Assistance – Financial support through schemes like National Overseas Scholarship.

Students are advised to check eligibility criteria and apply online through official government portals.

2.4Future Enhancements

The current system provides AI-based city analysis and citizen services through natural language interaction. While effective, several enhancements can be introduced in the future to improve accuracy, usability, and scalability:

1. **Real-Time Data Integration**
   - Connect the system with **live government databases, open data APIs, and crime/traffic records** to provide updated and accurate statistics instead of static AI-generated insights.
2. **Multilingual Support**
   - Extend the model to handle multiple languages (e.g., Hindi, Tamil, Marathi, etc.) to make the system accessible to a **wider population**.
3. **Voice-Based Interaction**
   - Integrate **speech-to-text and text-to-speech** modules for voice-enabled citizen queries and responses.

4. **Mobile Application Deployment**
    - Develop Android/iOS apps for broader accessibility, in addition to the Gradio web interface.
5. **Personalized Recommendations**
    - Provide **personalized insights** based on user profiles, such as safety alerts for specific areas or customized government schemes.
6. **Enhanced Visualization**
    - Add **graphs, charts, and dashboards** for crime and accident statistics to improve understanding through data visualization.
7. **Feedback & Continuous Learning**
    - Allow users to rate responses, enabling the system to **improve accuracy over time** through fine-tuning and feedback loops.

3.CODING

```
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
)

# Ensure tokenizer has a pad token
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token


# Function to generate response
```

```
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


# City analysis function
```

```
def city_analysis(city_name):
    prompt = f"""Provide a detailed analysis of {city_name} including:
1. Crime Index and safety statistics
2. Accident rates and traffic safety information"""
    return generate_response(prompt, max_length=1000)


# Citizen interaction function
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\n{query}"
    return generate_response(prompt, max_length=1000)


# Gradio Interface
with gr.Blocks() as app:
    gr.Markdown("# 🏙️ City Analysis & Citizen Services AI")

    with gr.Tabs():
        # Tab 1: City Analysis
        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)

        # Tab 2: Citizen Services
        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)


    # Launch app
    app.launch(share=True)
```

3.1Code Explanation

□ **Imports & Setup**

- Uses **Gradio** for UI, **PyTorch** for model execution, and **Transformers** to load the IBM Granite model.

□ **Model & Tokenizer**

- Loads "ibm-granite/granite-3.2-2b-instruct".
- Runs on **GPU (FP16)** if available, else **CPU (FP32)**.

□ **Response Generator**

- generate_response() → Tokenizes input, runs model, and decodes output into clean text.

□ Custom Functions

- city_analysis(city_name) → Generates crime & accident insights.
- citizen_interaction(query) → Provides government-style responses.

□Interface

- Built with **Gradio Blocks**.
- Two tabs: **City Analysis** & **Citizen Services**.
- Input (Textbox + Button) → Output (Textbox).

Launch

- app.launch(share=True) → Runs app locally and creates a shareable public link.

3.2Input and Output

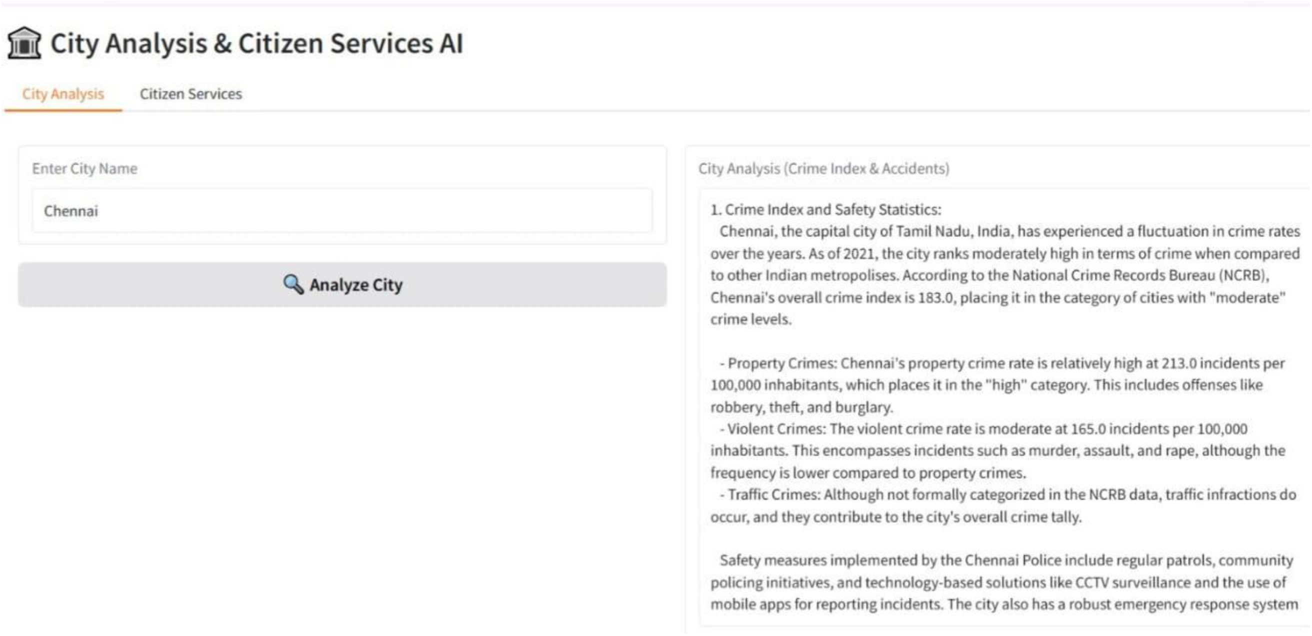
1. City Analysis

- **Input:** City name (e.g., *“Mumbai”*)
- **Output:** AI-generated report including **crime index, safety statistics, and accident rates**.

2. Citizen Services

- **Input:** Citizen query (e.g., *“What are government schemes for higher education?”*)
- **Output:** AI-generated **government-style response** with details on policies, services, or civic issues.

3.3Screenshot



Enter City Name

Chennai

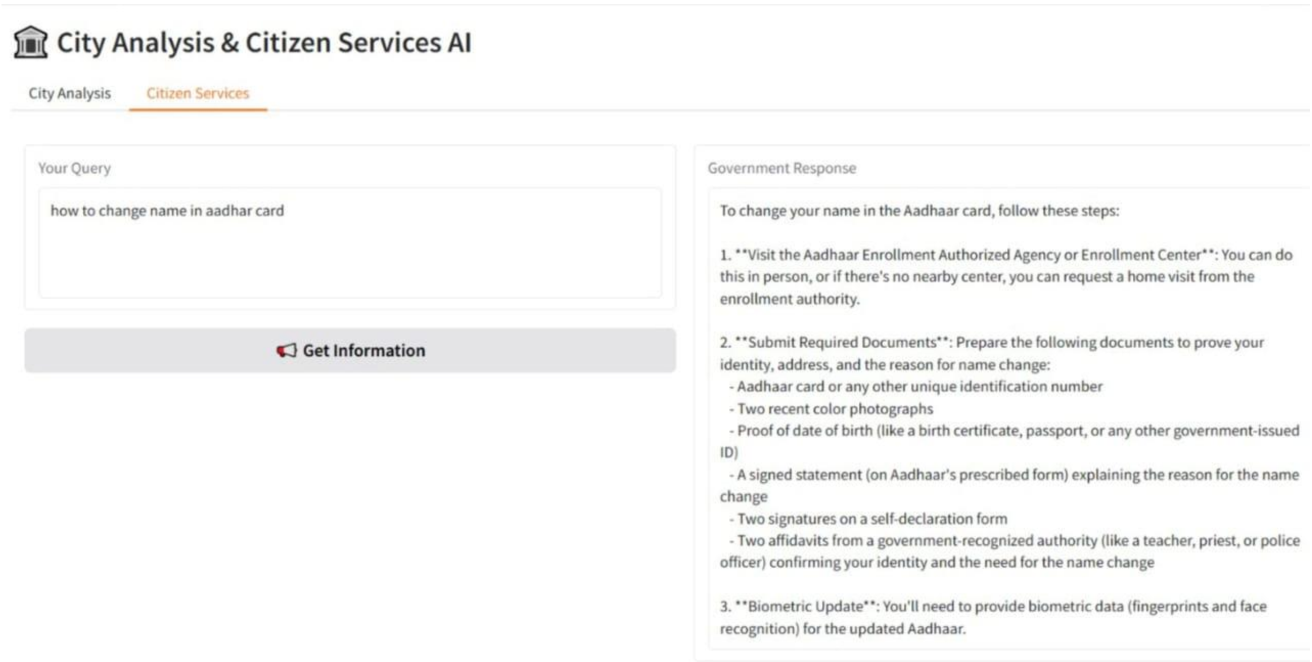
City Analysis (Crime Index & Accidents)

1. Crime Index and Safety Statistics:
 Chennai, the capital city of Tamil Nadu, India, has experienced a fluctuation in crime rates over the years. As of 2021, the city ranks moderately high in terms of crime when compared to other Indian metropolises. According to the National Crime Records Bureau (NCRB), Chennai's overall crime index is 183.0, placing it in the category of cities with "moderate" crime levels.

 - Property Crimes: Chennai's property crime rate is relatively high at 213.0 incidents per 100,000 inhabitants, which places it in the "high" category. This includes offenses like robbery, theft, and burglary.
 - Violent Crimes: The violent crime rate is moderate at 165.0 incidents per 100,000 inhabitants. This encompasses incidents such as murder, assault, and rape, although the frequency is lower compared to property crimes.
 - Traffic Crimes: Although not formally categorized in the NCRB data, traffic infractions do occur, and they contribute to the city's overall crime tally.

 Safety measures implemented by the Chennai Police include regular patrols, community policing initiatives, and technology-based solutions like CCTV surveillance and the use of mobile apps for reporting incidents. The city also has a robust emergency response system

Fig 1: City Analysis



Your Query

how to change name in aadhar card

Government Response

To change your name in the Aadhaar card, follow these steps:

1. **Visit the Aadhaar Enrollment Authorized Agency or Enrollment Center**: You can do this in person, or if there's no nearby center, you can request a home visit from the enrollment authority.

2. **Submit Required Documents**: Prepare the following documents to prove your identity, address, and the reason for name change:
 - Aadhaar card or any other unique identification number
 - Two recent color photographs
 - Proof of date of birth (like a birth certificate, passport, or any other government-issued ID)
 - A signed statement (on Aadhaar's prescribed form) explaining the reason for the name change
 - Two signatures on a self-declaration form
 - Two affidavits from a government-recognized authority (like a teacher, priest, or police officer) confirming your identity and the need for the name change

3. **Biometric Update**: You'll need to provide biometric data (fingerprints and face recognition) for the updated Aadhaar.

Fig 2:Citizen Services

3.4Advantages

1. **User-Friendly** → Simple Gradio interface with tabs for easy navigation.
2. **AI-Powered Insights** → Generates meaningful analysis and government-style responses.
3. **Time-Saving** → Reduces effort of searching multiple sources.
4. **Scalable** → Can be extended with new services or datasets.
5. **Cross-Platform** → Works on both CPU and GPU environments.
6. **Interactive** → Provides conversational responses instead of static data.

3.5Limitations

- **AI-Generated Data** → Information (crime stats, accidents) may not be real-time or 100% accurate.
- **Model Dependency** → Accuracy depends on IBM Granite model's knowledge base.
- **Resource Intensive** → Large models may run slowly on CPU-only machines.
- **No Live Database Integration** → Currently does not fetch official government or police data.
- **Language Limitation** → Works mainly in English (no multilingual support yet).

3.6Applications

- **Smart Governance Tools** → Helps governments provide AI-driven citizen services.
- **City Information Systems** → Useful for quick safety and traffic insights.
- **Educational Projects** → Demonstrates AI + NLP applications in civic engagement.
- **Public Awareness Platforms** → Citizens can easily query about policies and services.
- **Research & Planning** → Provides a base for future development of AI-driven dashboards.

4.CONCLUSION

The **AI-Powered City Analysis and Citizen Services Assistant** successfully demonstrates how artificial intelligence can be applied to improve **civic engagement and information accessibility**. By integrating the **IBM Granite 3.2B Instruct model** with a **Gradio-based interface**, the system provides citizens with easy-to-understand insights on **crime statistics, accident rates, and government services**.

The project proves that AI can act as a **virtual government assistant**, reducing the need for manual searches and simplifying access to public information. While the current system relies on **AI-generated knowledge** and lacks real-time data integration, it lays a strong foundation for future enhancements such as **live data connectivity, multilingual support, mobile apps, and interactive dashboards**.

Overall, this project highlights the **potential of AI in governance** and sets the stage for building smarter, more accessible, and citizen-friendly digital services.

4.1References

□ **Gradio Documentation** –
Gradio: Build and share delightful machine learning apps, in Python.
Available at: https://www.gradio.app

□ **Hugging Face Transformers** –
Hugging Face: Transformers library for Natural Language Processing.
Available at: https://huggingface.co/docs/transformers

□ **IBM Granite Models** –
IBM Granite LLMs on Hugging Face.
Available at: https://huggingface.co/ibm-granite

□ **PyTorch Documentation** –
PyTorch: An open source deep learning framework.
Available at: https://pytorch.org/docs

□ **Python Official Documentation** –
Python Programming Language Reference.
Available at: https://docs.python.org/3