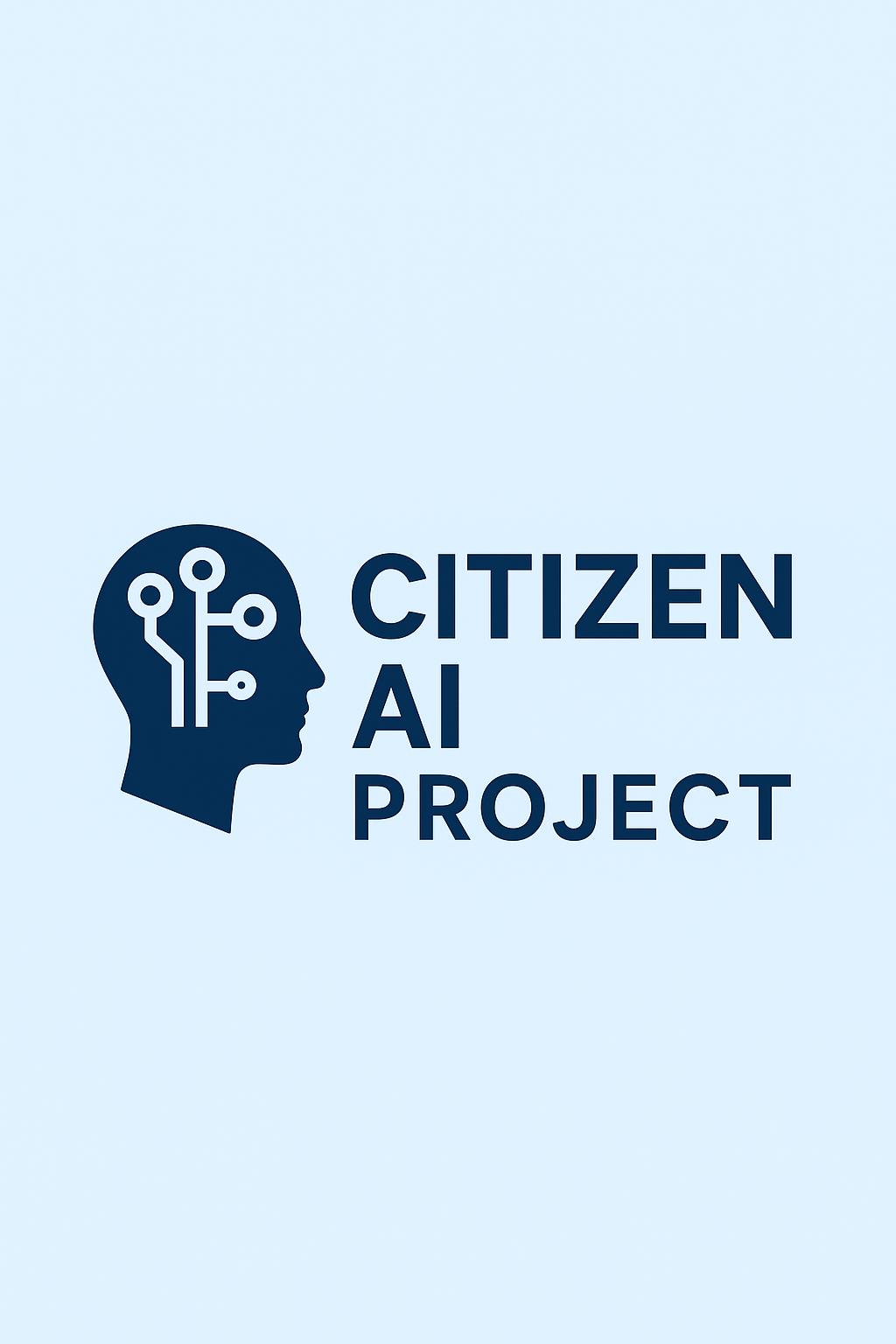
**CITIZEN AI -INTELLIGENT CITIZEN ENGAGEMENT PLATFORM**

****

**Team leader**:

Abinaya T

**Team members**:

Abinaya R

Aishwarya S

Akalya I

**1. SYSTEM ANALYSIS**

**1.1 Introduction**

The Citizen AI Project is designed to empower individuals and communities by making Artificial Intelligence (AI) more accessible, transparent, and beneficial for society. In today’s world, AI is not limited to research labs or big technology companies—it has become a crucial tool for solving everyday problems such as improving healthcare, supporting education, enhancing governance, and driving sustainable development.

Citizen AI focuses on bridging the gap between advanced AI technologies and ordinary citizens. The project aims to educate, engage, and enable people to understand how AI works, how it impacts their lives, and how they can actively participate in shaping its use for the greater good. By encouraging responsible AI development and usage, Citizen AI ensures that technology remains ethical, inclusive, and human-centered.

Through this project, citizens will gain exposure to practical AI applications, hands-on tools, and real-world problem-solving scenarios, ultimately helping them become contributors rather than just consumers of AI-driven innovations.

**1.2 Objective**

1. Enhance Public Awareness – To make Artificial Intelligence more accessible and understandable for citizens, enabling them to use AI tools for everyday decision-making.

2. Promote Civic Engagement – To provide a digital platform where citizens can interact with AI to get accurate, reliable, and timely information about government services, policies, and civic issues.

3. Support Data-Driven Safety Insights – To analyze city-specific data (crime index, accident rates, safety assessments) and deliver useful insights that help citizens and authorities make informed decisions.

**1.3 Existing System**

1. Government Websites & Portals – Citizens use official websites to check policies, schemes, and city-related updates. However, these sites are often complex, time-consuming, and difficult to navigate for the common user.

2. Manual Enquiries & Helplines – People depend on physical visits to government offices or calling helplines for information. This process is slow, sometimes unreliable, and not always citizen-friendly.

3. News & Reports – Safety data such as crime statistics or accident rates are usually found in reports, news articles, or survey databases. These are often technical and not easily understandable for non-experts.

4. Search Engines & Social Media – Many citizens turn to online searches or social media platforms for civic updates. While fast, this method can lead to misinformation, lack of authenticity, and scattered information sources.

**1.4 Proposed System**

* City Analysis Module – Citizens can input the name of a city and receive a detailed AI-generated analysis, including crime index, accident statistics, and overall safety assessment.
* Citizen Services Module – A virtual AI assistant responds to queries related to government services, policies, and civic issues in a clear and accessible manner.
* Interactive Interface – Built using Gradio, the platform provides a simple, intuitive, and web-based environment where users can easily interact with AI.
* Automated & Scalable – Unlike static websites or manual reports, the system generates responses dynamically and can be scaled to cover multiple cities and services.

**1.5 Tools and Technologies Used**

* Programming Language:

Python (for AI model integration and backend development)

* Libraries & Frameworks:

Transformers (Hugging Face) – for loading and fine-tuning AI models

* Torch (PyTorch) – for deep learning model operations

Gradio – for building the interactive user interface

* AI Model:

IBM Granite 3.2 2B Instruct – pre-trained large language model for natural language understanding and generation

* Platform & Environment:

Jupyter Notebook / Python IDEs (for development and testing)

Local Machine or GPU (for running the AI model efficiently)

**2. SYSTEM DESIGN**

**2.1 Project Description**

The system is designed with two core modules:

1. City Analysis Module – Citizens can enter the name of a city to receive a detailed AI-generated report covering the crime index, accident rates, and an overall safety assessment. This helps people gain quick insights into urban safety conditions without navigating multiple fragmented sources.

2. Citizen Services Module – Citizens can input queries related to public services, government policies, or civic concerns. The AI assistant responds with clear, relevant, and accessible information, thus bridging the gap between citizens and governing authorities.

* The project aims to modernize civic engagement by providing a centralized, interactive, and user-friendly system that reduces reliance on slow manual processes and scattered information sources. With features like instant responses, scalability, and ethical AI usage, the Citizen AI Project promotes transparency, inclusivity, and data-driven decision-making.
* This system can be further extended to integrate real-time government databases, multilingual support, and mobile applications, making it a scalable solution for smart city initiatives and digital governance.

**2.2 Testing**

* Unit Testing → Verified individual functions like generate\_response(), city\_analysis(), and citizen\_interaction().
* Integration Testing → Checked smooth interaction between AI model, tokenizer, and Gradio interface.
* Functional Testing → Ensured both modules (City Analysis & Citizen Services) work as intended.
* Input Testing → Tested with valid, invalid, and empty inputs to ensure stability and error handling.

**2.3 Sample Output**

1. City Analysis Module

Input (City Name): Mumbai

Output:

City: Mumbai

Crime Index: Moderate – higher than average in theft and petty crimes.

Accident Rates: Road accidents are frequent due to heavy traffic congestion.

Safety Assessment: Safe during daytime, but caution advised at night in crowded areas.

Overall: Requires improved traffic management and stricter law enforcement.

2. Citizen Services Module

Input (Citizen Query): What government schemes are available for students?

Output:

Response:

The government provides several schemes for students including:

1. National Scholarship Portal (NSP) for financial aid.

2. PM Scholarship Scheme for higher education.

3. Post-Matric Scholarship for economically weaker sections.

4. Skill India Initiative for vocational training.

You can apply through the official scholarship portal or respective state education websites.

**2.4 Future Enhancements**

1. Real-Time Data Integration – Connect with government and city databases for live updates on crime, traffic, and public services.

2. Multilingual Support – Enable responses in regional languages to reach a wider citizen base.

3. Mobile Application Development – Extend the system as a mobile app for easy access anywhere.

4. Personalized Recommendations – Use AI to provide tailored safety tips, policy updates, and city-specific alerts to citizens.

**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

)

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\_name}\nAnalysis:"

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}\nResponse:"

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():

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)

**3.1 Code with Explanation**

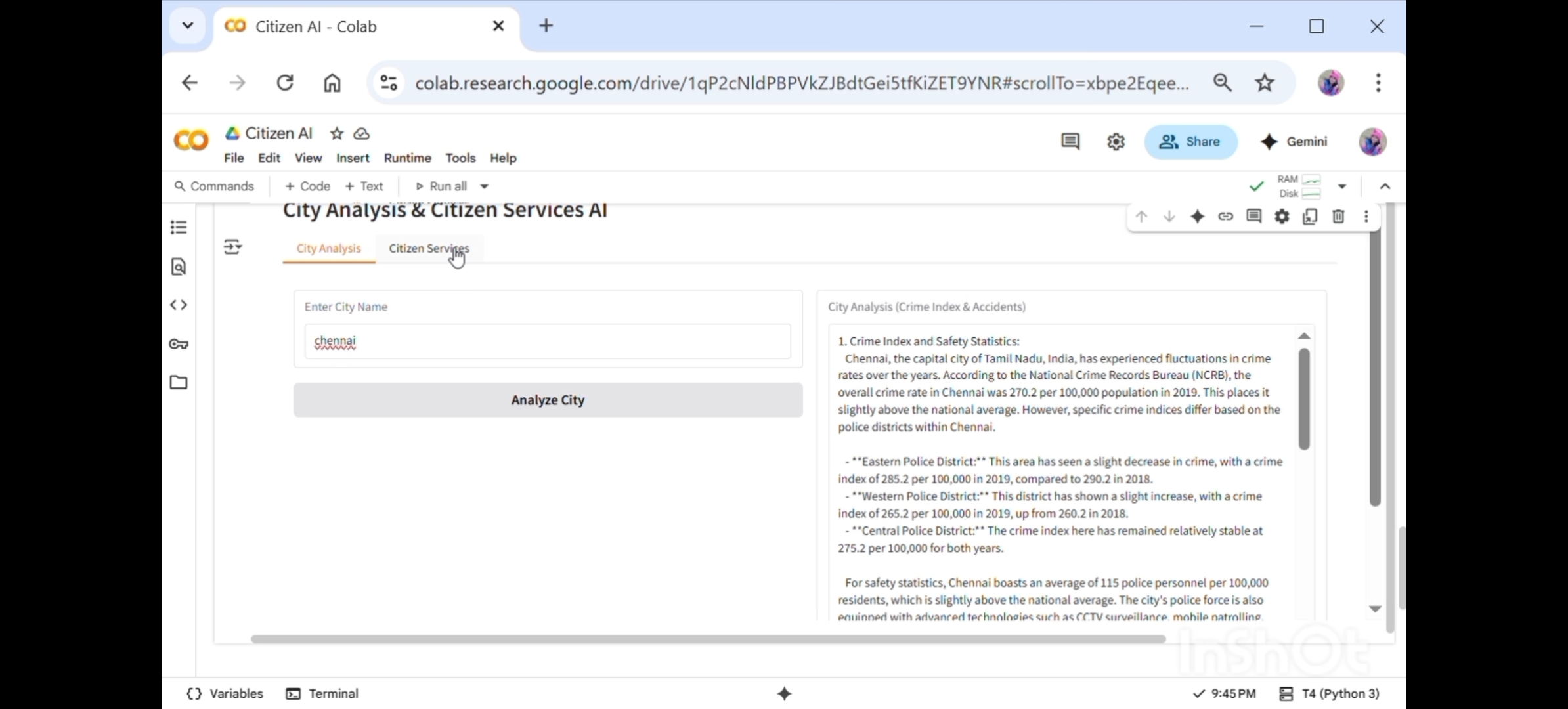
* **Importing Required Libraries**:Builds a simple web interface for the project.
* **Fixing Padding Issue:** the end-of-sequence (eos) token is used as the pad token to avoid errors.
* **Response Generation Function:** Decoding: Converts tokens back to human-readable text.
* Cleaning: Removes the input prompt from the output.
* **City Analysis Function :** Creates a custom prompt asking the model

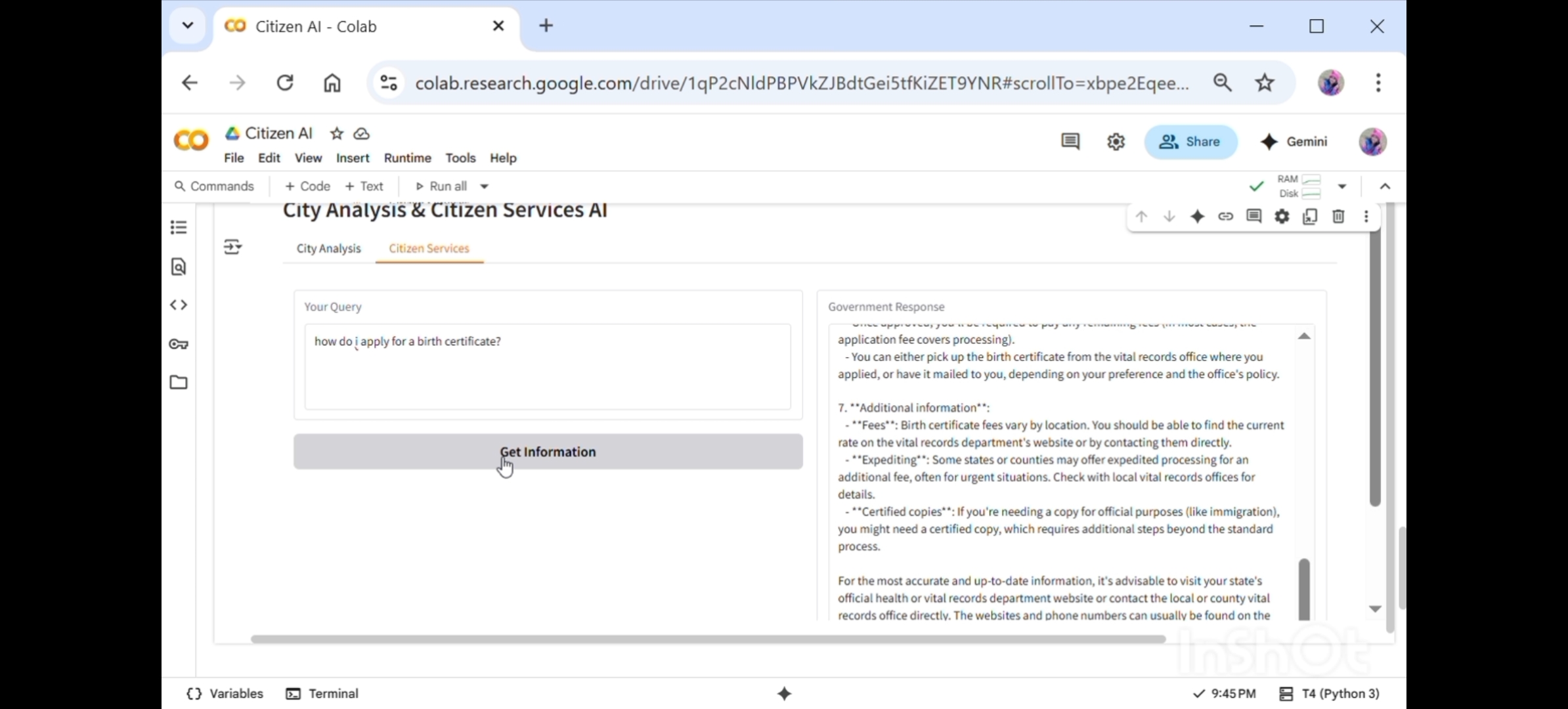
**3.2 Input and Output**

**Input:**User enters either a city name for analysis or a citizen query related to government services and civic issues.

**Output:** The system generates an AI-based city safety report or a government response to the citizen’s query.

**3.3 Screenshot**



****

**3.4 Advantages**

* Easy to use for all citizens.
* Gives quick answers.
* Helps in understanding city safety.
* Provides information about government services.

**3.5 Limitations**

* Depends on internet connection.
* May not always give 100% accurate data.
* Limited to the knowledge of the AI model.
* Cannot replace real government databases fully.

**3.6 Applications**

* Check city safety and crime information.
* Get answers about government schemes and services.
* Help citizens make informed decisions.
* Support digital governance and smart city initiatives.

**4. CONCLUSION**

The Citizen AI Project is a useful and interactive platform that helps citizens access city safety information and government services quickly and easily. It provides reliable, AI-generated responses, reducing the need for manual searches or visits to offices. With its user-friendly interface and centralized system, it promotes digital governance, informed decision-making, and civic engagement. In the future, it can be enhanced with real-time data, multilingual support, and mobile accessibility to serve a wider audience.

**4.1 References**

1. Hugging Face Transformers Documentation – https://huggingface.co/docs/transformers

2. PyTorch Documentation – https://pytorch.org/docs/stable/index.html

3. Gradio Documentation – https://gradio.app/docs/

4. IBM Granite Model Information – https://huggingface.co/ibm-granite/granite-3.2-2b-instruct

5. Digital Governance and Smart Cities Articles – Various sources from government portals and research papers.