CITIZEN AI

Project

**📍Introduction :**

Project Title: CITIZEN AI

Team Leader : **JAINUL MUSBIRA J**

Team member : **JAMEELA JEFRIN H**

Team member : **PRADEESHA M**

Team member : **RAFIDA M**

Team member : **SUJANA H**

📍 **Project Overview:**

**• Purpose**: Citizen AI is designed to deliver fast, accurate, and accessible information about government services, civic initiatives, and public issues. By collecting citizen feedback and analyzing public sentiment, it empowers government officials with real-time insights, enabling data-driven policy decisions and fostering transparent citizen engagement

**• Features:**

Conversational Interface:

✔️Funtionality: Citizens can interact naturally with the system to receive instant, reliable answers about government service

Resource Forecasting:

✔️Functionality: The Resource Forecasting module predicts future demand and utilization of essential civic resources such as energy, water, waste management, and public services.

Sentiment & Opinion Analysis:

✔️ Functionality: Continuously tracks and evaluates public sentiment to understand trends and emerging issues.

Citizen Feedback Loop:

✔️ Functionality: Aggregates citizen input to help policymakers refine and adapt services based on real community needs.

Data Dashboards:

✔️ Functionality: Provides visualized analytics and actionable insights for officials to support evidence-based decision-making.

Web-Based (Gradio):

✔️ Functionality: A simple, interactive interface for testing and showcasing the system’s capabilities

📍 **Architecture:**

• Frontend (Gradio): A user-friendly web interface that allows citizens to interact with the AI system.

• Backend (Google Colab + Python): Hosts the application, runs the AI model, processes all inputs in real time, and generates accurate responses.

• Dashboards: Provide visual insights of citizen feedback, sentiment trends, and resource forecasts .

• Version Control (GitHub): Ensures project files, code, and documentation are securely stored, tracked for changes.

**📍Setup Instructions:**

**Prerequisites:**

• Python programming knowledge

• Gradio framework installed

• Access to IBM Granite Model via Hugging Face

• Google Colab with T4 GPU enabled

• GitHub account for version control

**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 all required libraries installed.

4. Upload the final project files to GitHub for version control and collaboration

**📍Folder Structure:**

• app/ – Contains backend logic and integration files

• ui/ – Holds Gradio app interface files to provide the frontend interaction layer.

• citizen\_ai.py – The main application file which connects frontend, backend, and the AI model.

• model\_loader.py – Manages IBM Granite model integration and loading.

• dashboard.py – Handles visualization of citizen feedback, sentiment trends, and resource forecasts

**📍Running Application:**

**•** Open Google Colab and load the Citizen AI project notebook.

• Install all required dependencies and configure the runtime with GPU enabled.

• Run the notebook cells to initialize the AI model and launch the Gradio interface.

• Access the generated link to interact with the Citizen AI web application

**📍API Documentation:**

Citizen AI following API capabilities:

• Submit and retrieve queries related to government services.

• Upload user feedback for automated sentiment analysis.

• Retrieve and view summarized policies and guidelines.

• Access dashboards, analytics, and generated reports.

📍 **Authentication:**

Demo Mode: Citizen AI currently runs without authentication for testing and demonstration purposes.

Production Mode:

• Supports API keys, OAuth 2.0, and role-based access control.

• Encrypted communication (HTTPS) to ensure data security.

• Fine-grained permissions for officials, admins, and citizens

**📍 User Interface:**

Citizen AI uses a simple and intuitive Gradio interface that includes:

• Allows citizens to ask questions and receive instant AI-generated responses.

• Displays reports, analytics, and real-time updates for government staff.

• Enables users to submit feedback, files

• Optimized for desktop and mobile use

**📍Testing:**

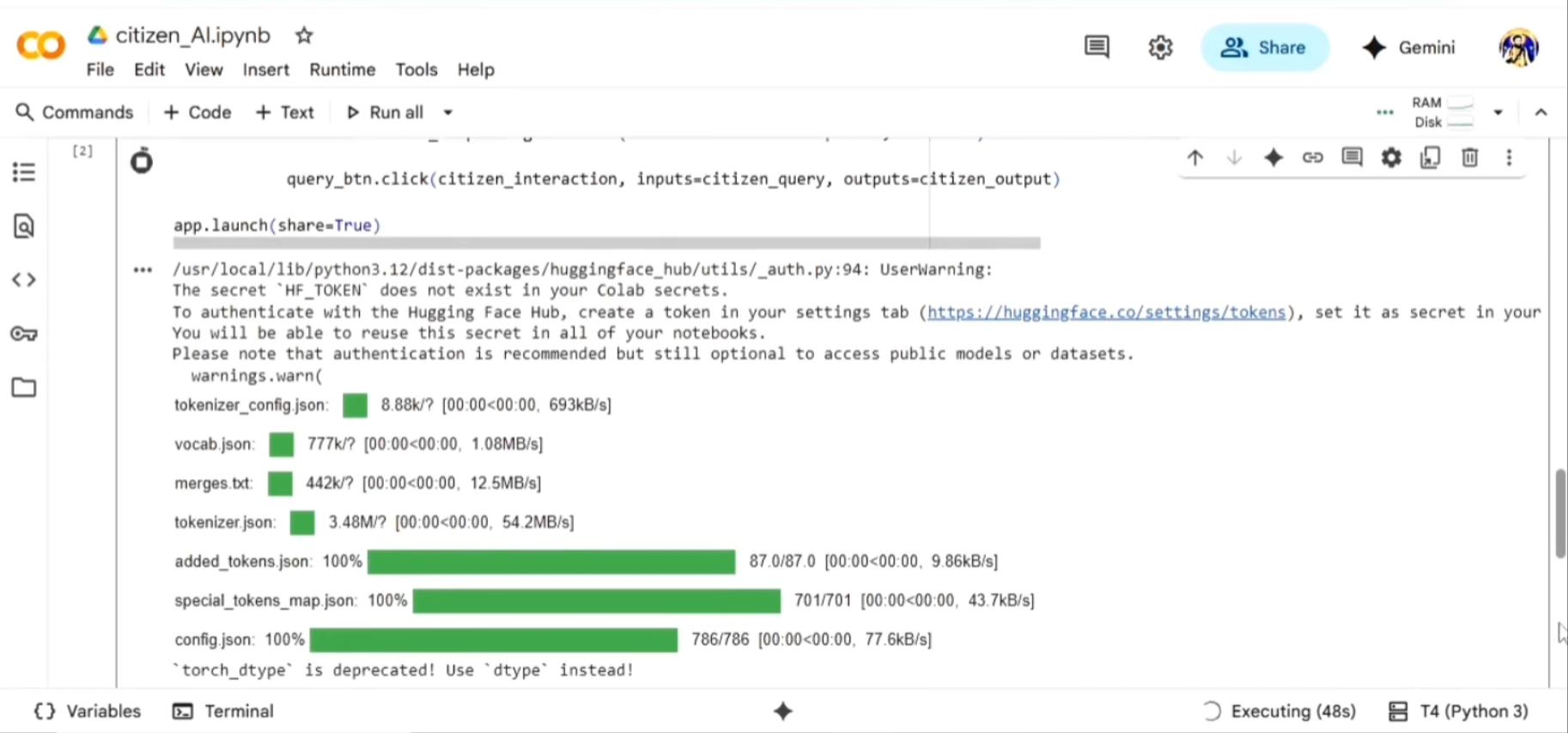
The Combination of automated and manual testing of Citizen AI:

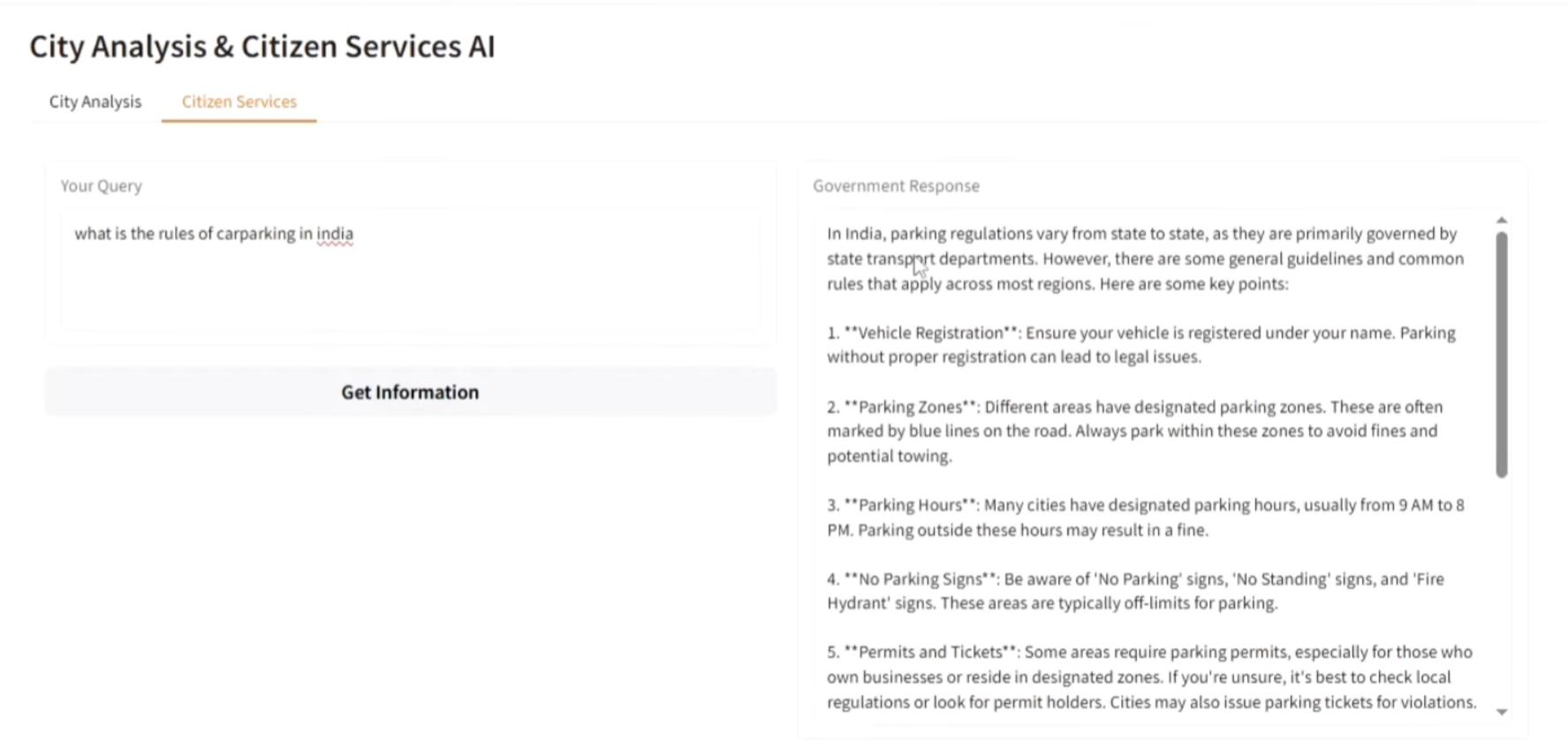
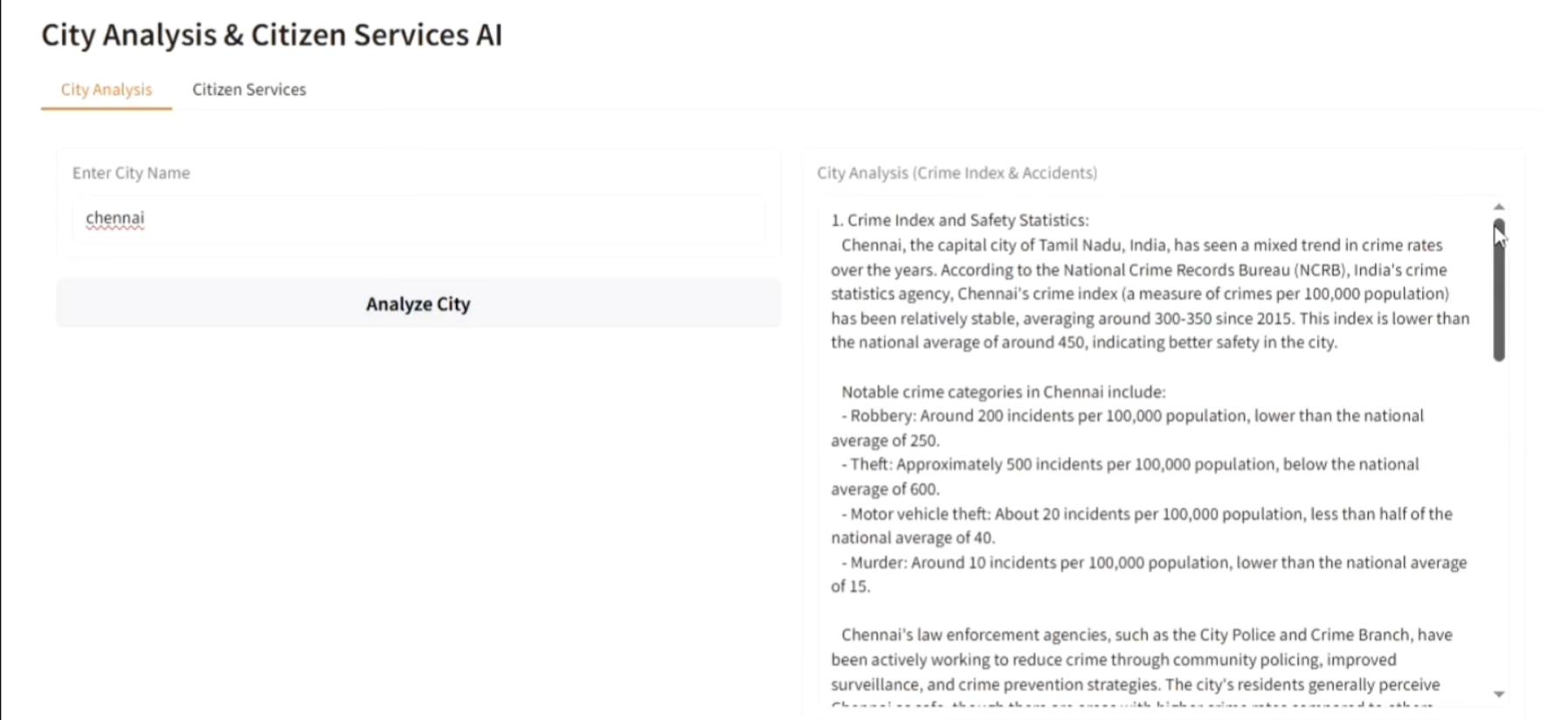
• Validates AI response accuracy and API endpoints

• Simulates unexpected or invalid inputs to ensure robust error handling

• Basic vulnerability and data privacy tests for authentication and data handling

📍 **Screenshots:**

****

****

**📍Known Issues:**

**•** The demo server may slow down under heavy concurrent user loads

**•** Limited Language Support

• Summarized policies may take time to refresh after backend updates

• Limited Authentication in Demo Mode

📍 **Future Enhancement:**

**•** Improved Error Handling: Enhance error messaging and fallback responses for better user experience.

• Mobile App Integration: Build a native mobile application with push notifications for updates.

• Authentication & Authorization: Implement OAuth 2.0, API keys, and role-based access for all production environments.