**Citizen AI with IBM**

**Project Documentation**

***1. Introduction***

*• Project Title: Citizen AI – Intelligent Citizen Engagement Platform*

*• Team Member(s):*

*Pavithra B*

*Kanimozhi R*

*Monica Mary J*

*Pavithra B*

***2. Project Overview***

*• Purpose:*

*The purpose of Citizen AI is to empower citizens and government officials with a simple, intelligent assistant that provides information about government services, tracks public sentiment, and enables data-driven decision-making. Using IBM Granite models and a Gradio interface, this project creates an interactive and accessible platform that is easy to deploy in Google Colab for low-cost, reliable performance.*

• Features:

Conversational Interface

Key Point: AI-powered chatbot

Functionality: Allows citizens to ask questions about government services and civic issues in natural language.

Public Sentiment Tracking

Key Point: Feedback analytics

Functionality: Collects, analyzes, and displays citizen feedback in an easy-to-read dashboard for officials.

Lightweight Deployment

Key Point: Low-cost, scalable setup

Functionality: Runs in Google Colab with GPU support for fast and cost-efficient execution.

Open Source & Version Controlled

Key Point: Collaborative development

Functionality: Project is stored and maintained on GitHub for version control and easy collaboration.

***3. Architecture***

*Frontend (Gradio):*

*Provides an intuitive and lightweight web UI for user interaction. Citizens can input queries and view responses in real time.*

*Backend (Hugging Face Granite):*

*Uses granite-3.2-2b-instruct model for natural language understanding and response generation.*

*Execution Environment:*

*Deployed in Google Colab with T4 GPU support for optimal performance.*

*Version Control (GitHub):*

*Code repository is maintained on GitHub, allowing for easy updates and collaboration.*

***4. Setup Instructions***

*Prerequisites:*

*Python 3.x*

*Knowledge of Gradio framework*

*IBM Granite model access (Hugging Face)*

*Google Colab with GPU runtime*

*GitHub account for version control*

*Installation Process:*

*1. Open Google Colab and create a new notebook.*

*2. Change runtime to T4 GPU (Runtime → Change runtime type).*

*3. Install dependencies:*

*!pip install transformers torch gradio -q*

*4. Copy and run the Citizen AI code provided in the guided project.*

*5. Launch the Gradio application by clicking the generated link.*

*6. Test the chatbot interface and verify outputs.*

*7. Download project as .py and upload it to GitHub repository.*

***5. Folder Structure***

*CitizenAI/*

*│*

*├── citizen\_ai.py # Main application code*

*├── requirements.txt # Dependencies*

*├── README.md # Project documentation*

*└── /notebooks # Colab notebooks (if saved).*

***6. Running the Application***

*1. Launch Google Colab notebook.*

*2. Install required libraries.*

*3. Run application cells sequentially.*

*4. Click the generated Gradio link to open the interface.*

*5. Interact with the chatbot and verify responses.*

***7. API / Model Usage***

*Hugging Face Model:*

*granite-3.2-2b-instruct – lightweight, instruction-tuned model from IBM Granite family.*

*Gradio Interface:*

*Auto-generates a public URL for easy interaction and demo purposes.*

***8. Authentication***

*Currently, the application is deployed in an open environment for testing.*

*Future versions can include:*

*API key authentication*

*User session tracking*

*Role-based access for citizens and administrators.*

***9. User Interface***

*Minimalist and clean design*

*One-click query input and output display*

*Dashboard view for citizen sentiment analytics.*

***10. Testing***

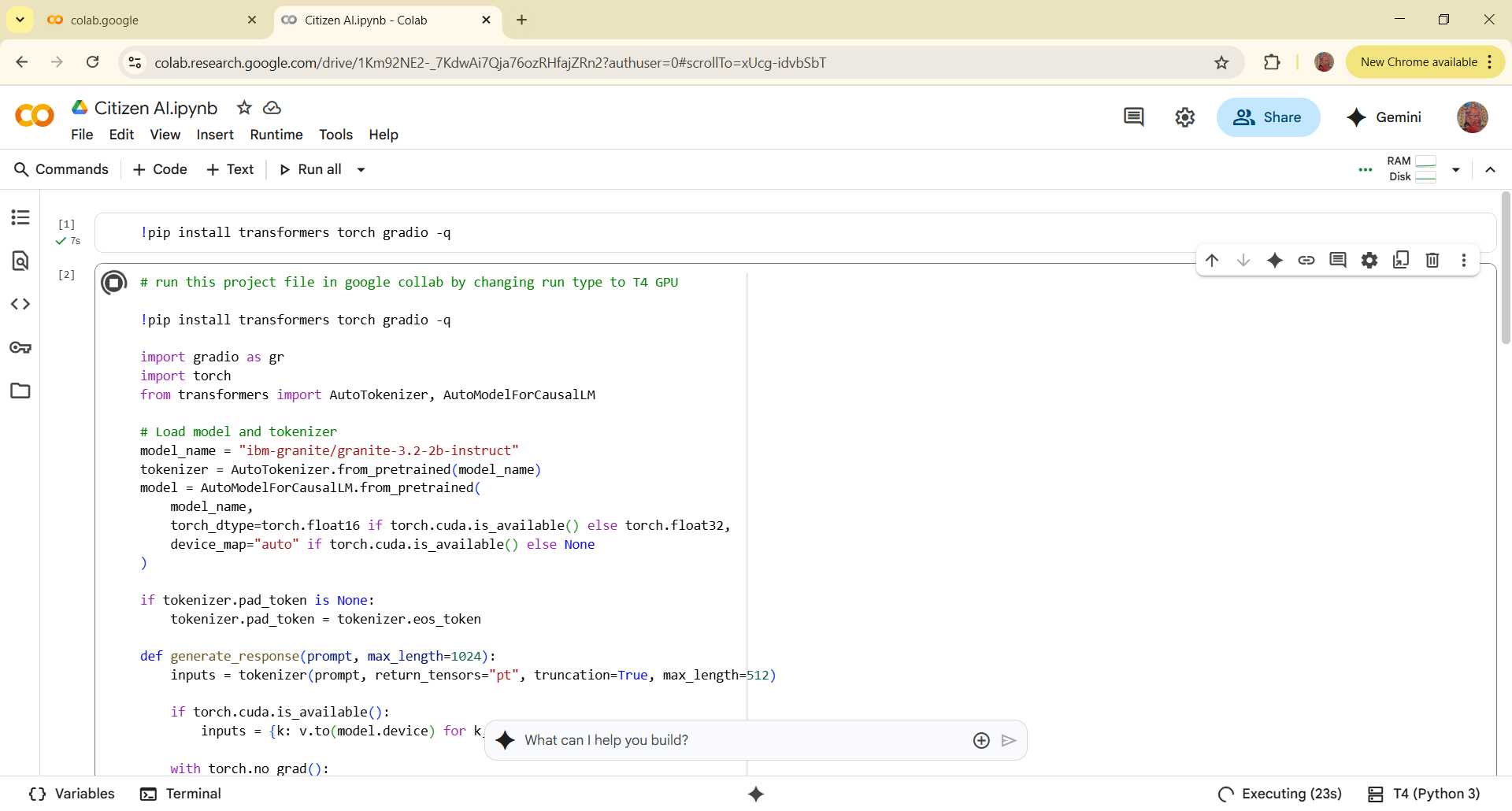
*Testing includes:*

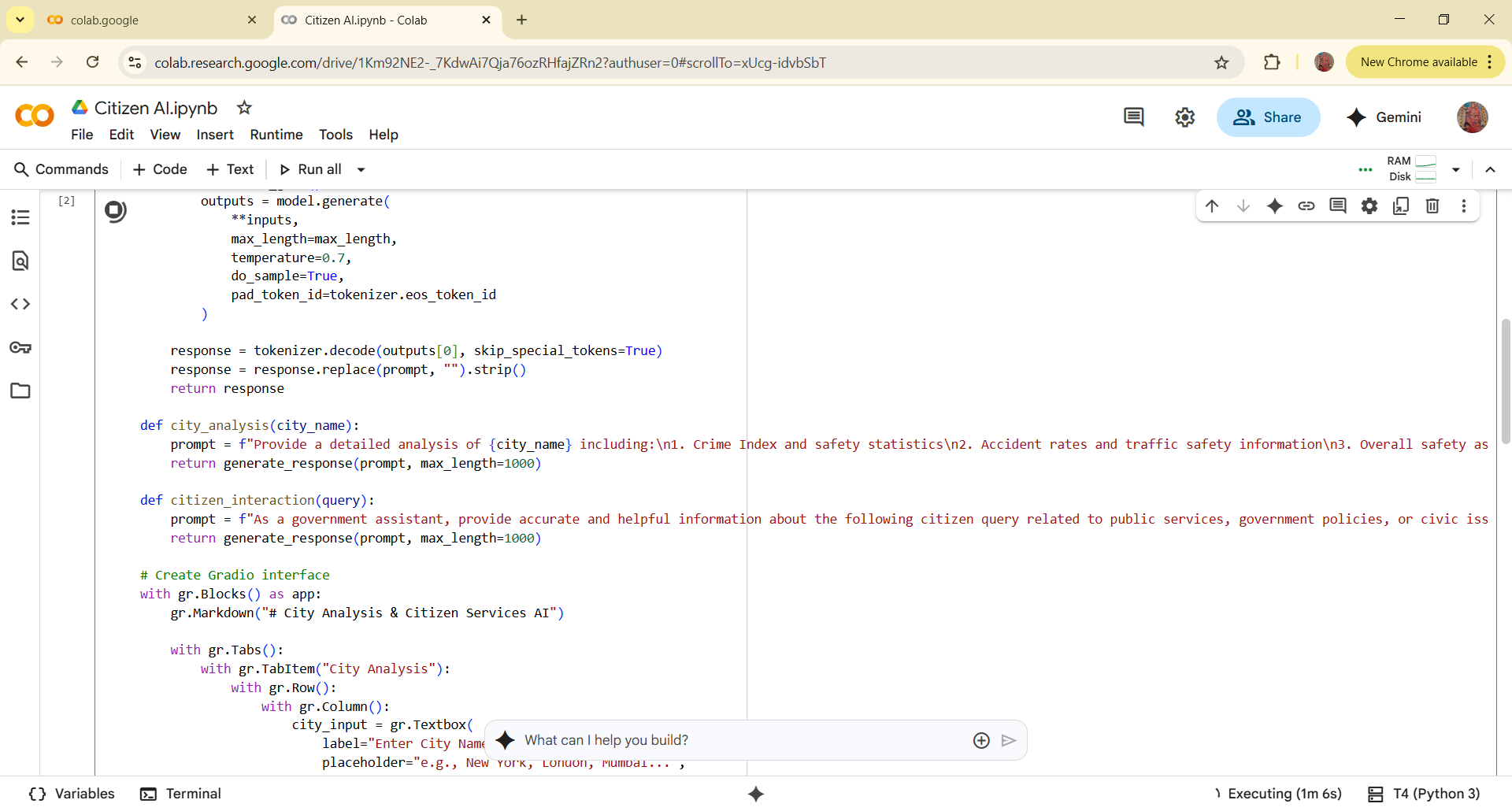
*Unit Testing: To verify core functions (model loading, response generation)*

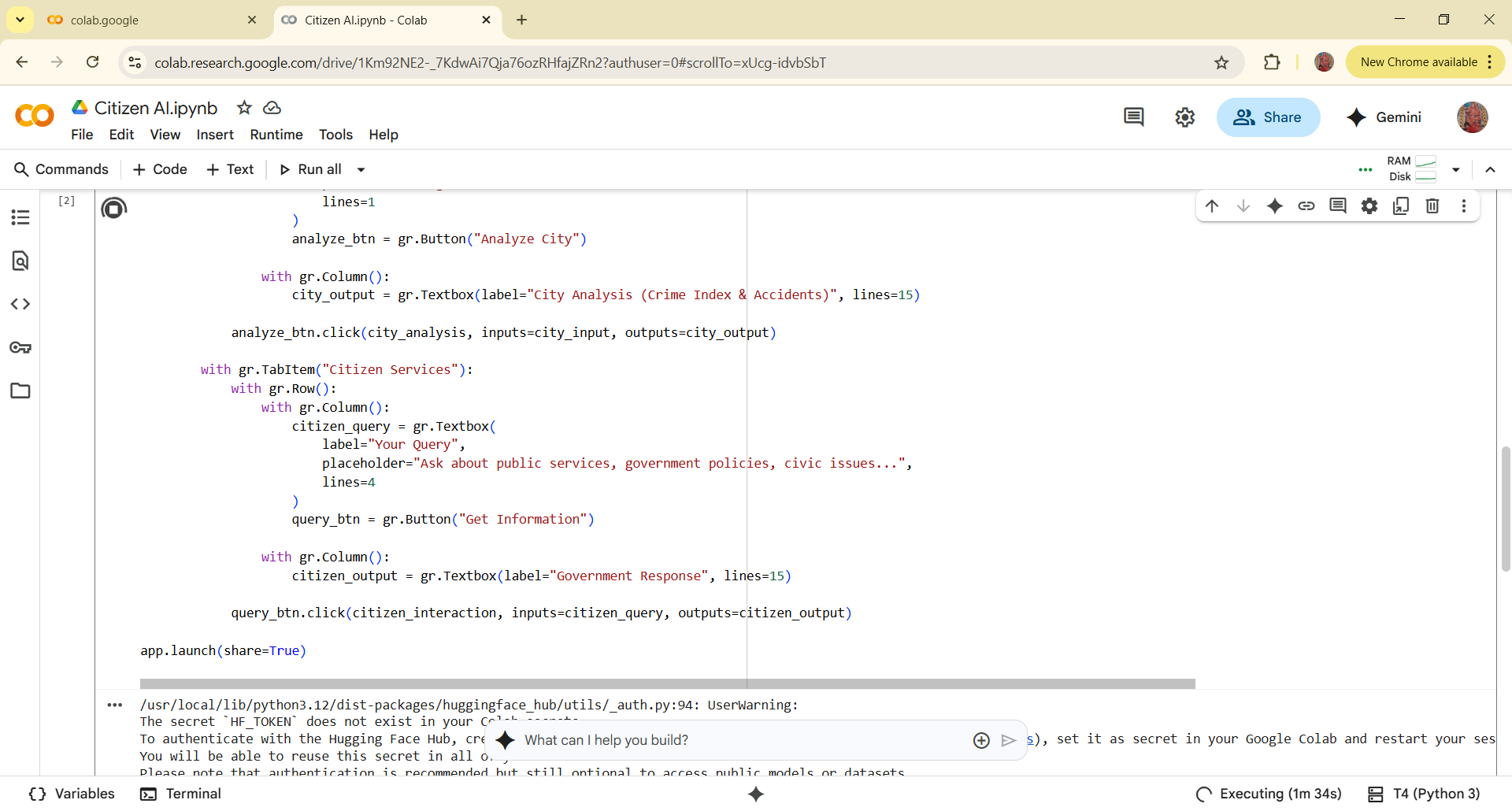
*Manual Testing: Running sample queries and validating responses*

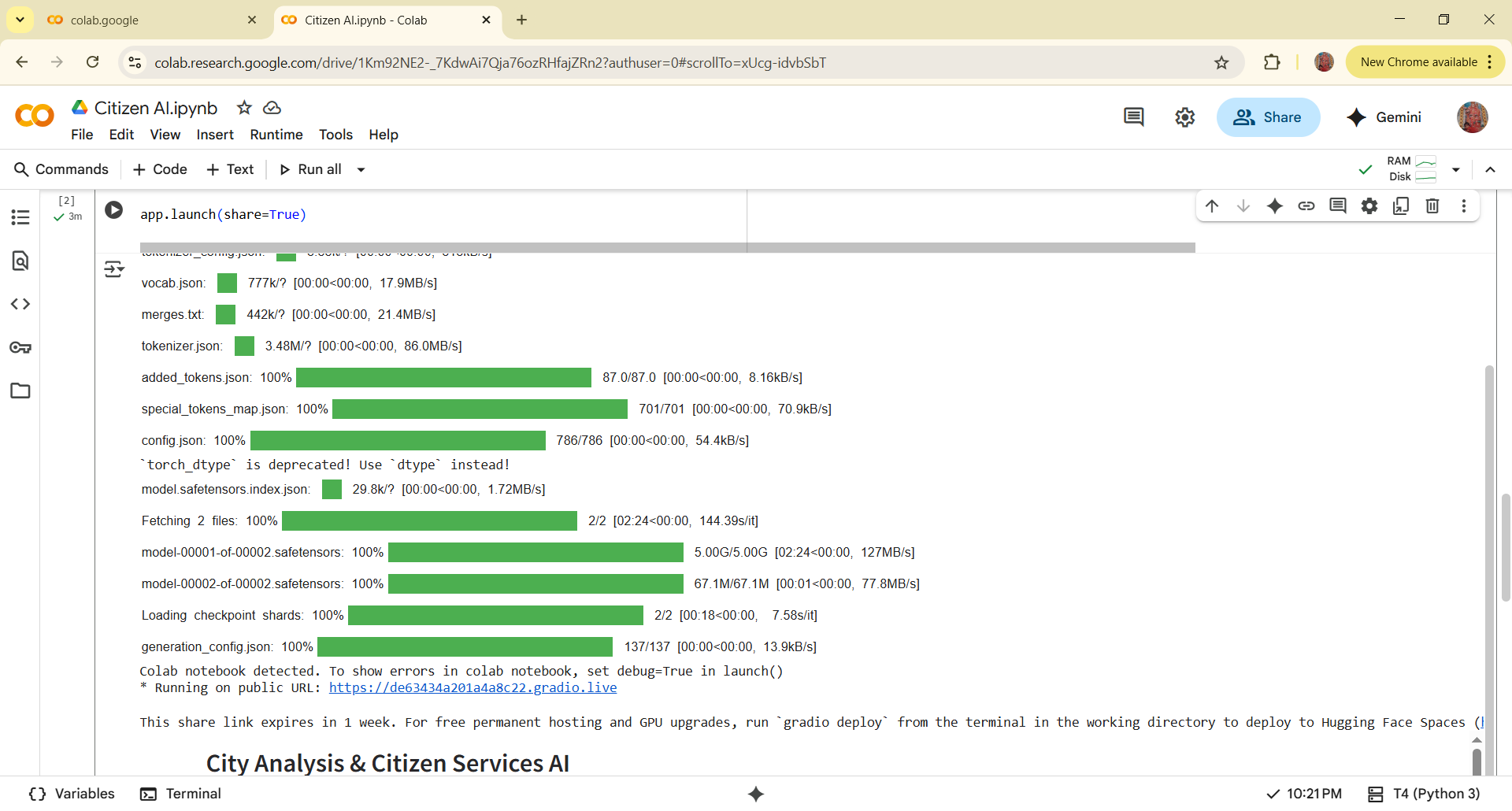
*Edge Case Testing: Handling empty queries or invalid inputs.*

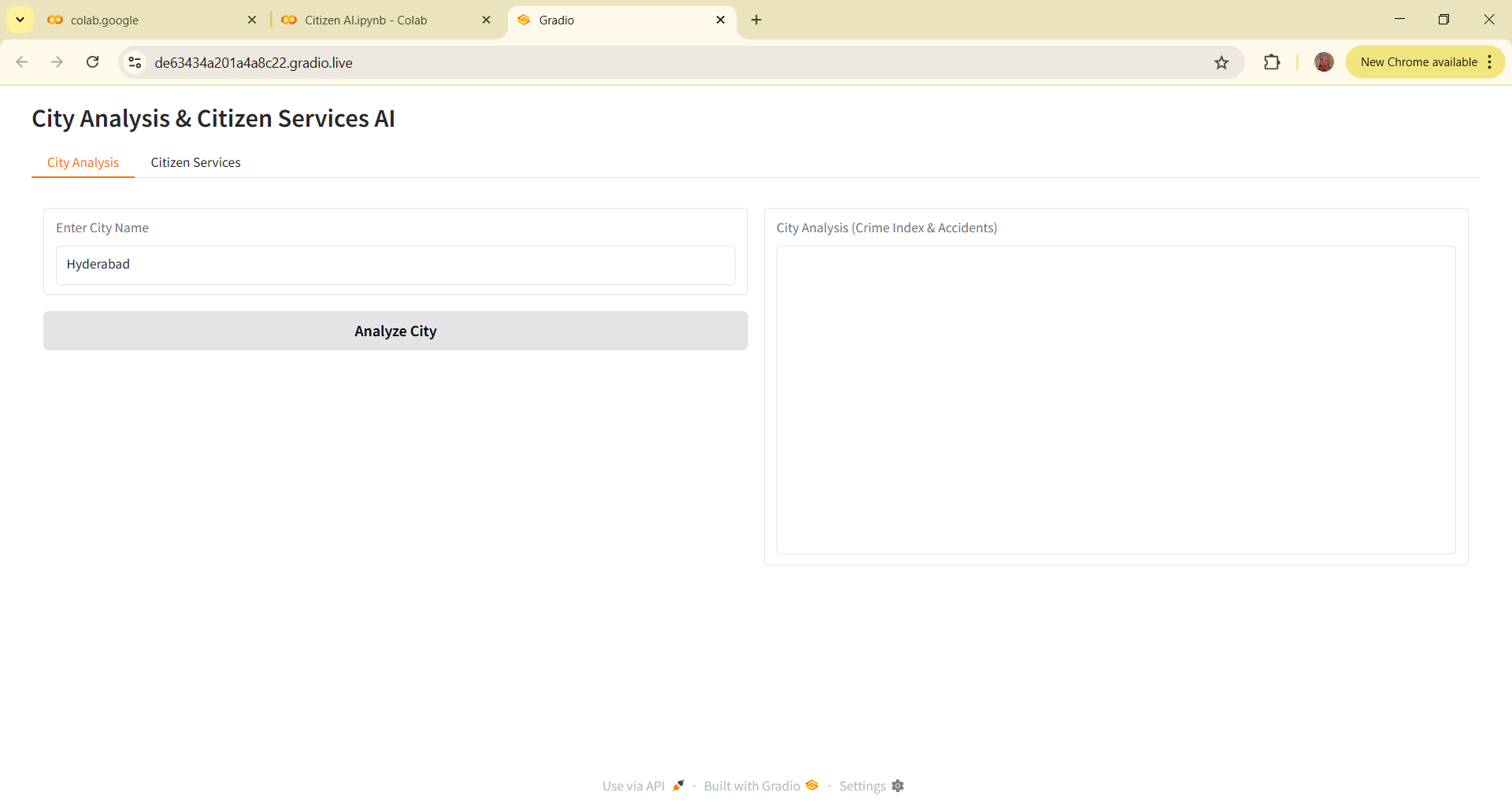
***11. Screenshots***

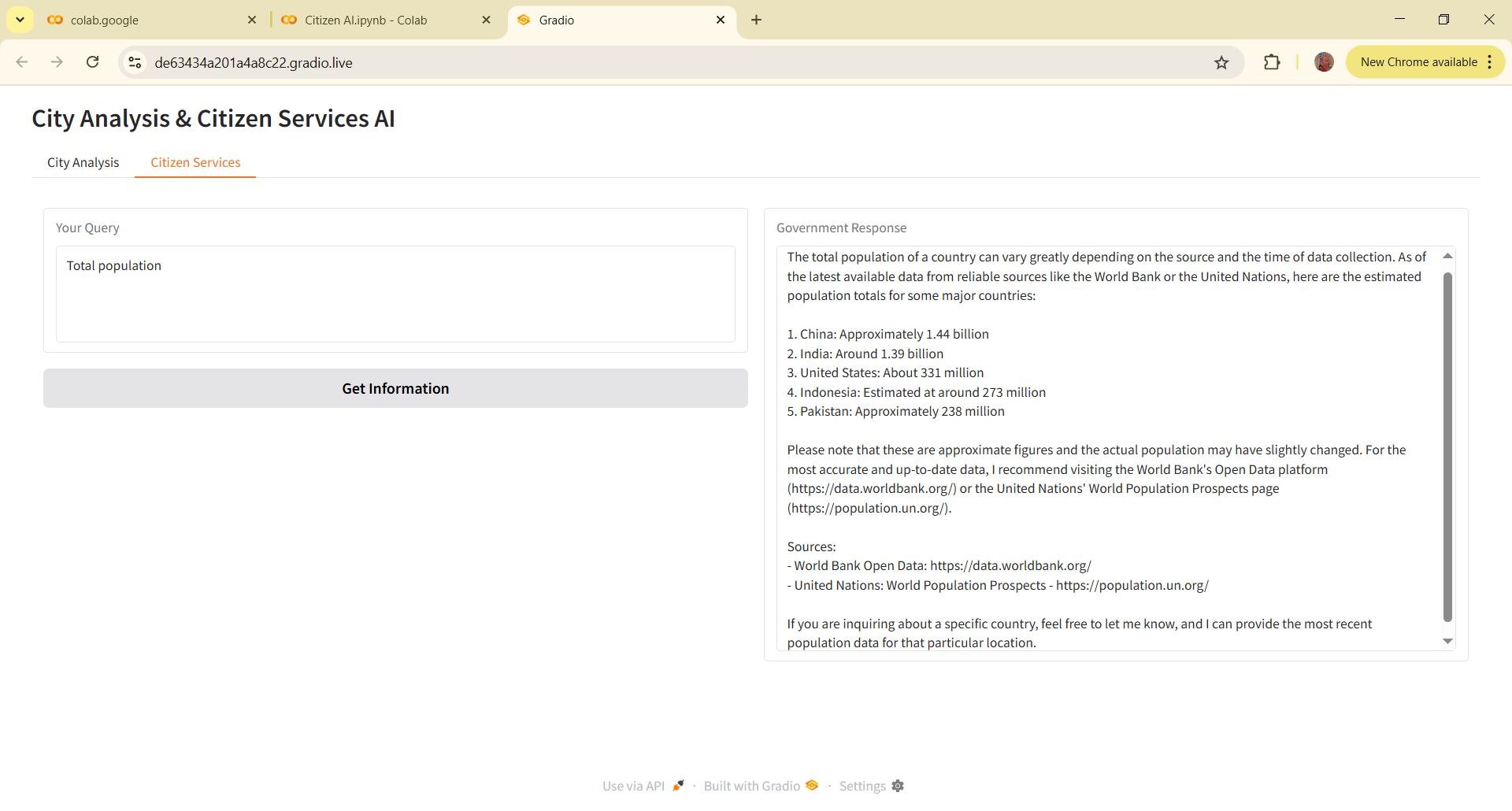
**

**

**

**

**

**

***12. Known Issues***

*Requires active internet connection to run Hugging Face model*

*Colab free tier may have limited GPU availability.*

***13. Future Enhancements***

*Add voice-based input/output for accessibility*

*Include multi-language support*

*Expand to real-time dashboard visualizations*

*Deploy on cloud platform for 24/7 availability.*