

Citizen AI – Intelligent Citizen Engagement Platform Project Documentation

1. Introduction

- **Project Title:** Citizen AI – Intelligent Citizen Engagement Platform
- **Team Members:**
 - INDHUPRIYA B
 - KASTHURI V
 - KOUSALYA S
 - LAKSHMI M

2. Project Overview

Purpose:

The purpose of *Citizen AI* is to strengthen citizen–government interaction through an intelligent, AI-powered engagement platform. It empowers citizens to easily access government services, understand policies, provide feedback, and participate in decision-making processes. For government officials, it acts as a digital partner—offering insights from citizen feedback, summarizing large policy documents, and supporting transparent governance. Ultimately, *Citizen AI* bridges trust, technology, and governance, fostering a more inclusive, participatory, and efficient society.

Features:

- **Conversational Interface**
 - *Key Point:* Natural language interaction
 - *Functionality:* Citizens can ask queries about government services, track applications, and receive responses in simple language.
- **Policy Summarization**
 - *Key Point:* Simplified policy understanding
 - *Functionality:* Converts lengthy policies and government notices into short, easy-to-understand summaries.
- **Citizen Feedback Loop**
 - *Key Point:* Public engagement
 - *Functionality:* Collects feedback, grievances, and suggestions from citizens; provides analytics to officials for better planning.
- **Service Tracking & Notifications**
 - *Key Point:* Transparency and accountability
 - *Functionality:* Allows citizens to track the status of their applications, complaints, or requests and receive real-time notifications.
- **Community Insights Dashboard**
 - *Key Point:* Data-driven governance
 - *Functionality:* Provides officials with analytics on citizen issues, service delays, and sentiment analysis.
- **Multilingual Support**
 - *Key Point:* Accessibility
 - *Functionality:* Supports multiple languages to ensure inclusivity for diverse communities.

- **Anomaly Detection**
 - *Key Point:* Early issue identification
 - *Functionality:* Detects irregularities in service performance or recurring complaints.
- **User-Friendly Interface**
 - *Key Point:* Citizen-first design
 - *Functionality:* Intuitive dashboard for both citizens and government officials with chatbots, forms, and visual reports.

3. Architecture

- **Frontend (React / Streamlit / Gradio):**
Provides a web-based interface with citizen dashboards, policy summaries, feedback forms, and multilingual chatbots.
- **Backend (FastAPI / Node.js):**
Exposes REST APIs for service queries, feedback collection, policy summarization, and citizen engagement workflows.
- **LLM Integration (OpenAI / Watsonx / Llama):**
Used for natural language processing, summarization, and multilingual chat support.
- **Vector Search (Pinecone / FAISS):**
Enables semantic search across government policies, FAQs, and historical citizen queries.
- **ML Modules (Sentiment Analysis & Anomaly Detection):**
Analyzes citizen feedback sentiment and flags unusual complaint patterns for authorities.

4. Setup Instructions

Prerequisites:

- Python 3.9+
- Virtual environment tools
- API keys (for LLM & Vector DB)
- Internet access

Installation Steps:

1. Clone repository
2. Install dependencies (pip install -r requirements.txt)
3. Create .env file with credentials
4. Start backend server (FastAPI/Node.js)
5. Launch frontend (React/Streamlit)
6. Interact with platform modules

5. Folder Structure

citizen_ai/

```
| — app/           # Backend logic
|   | — api/       # API routes (chat, feedback, services)
|   | — models/    # Data models
|   | — utils/     # Helper functions
| — ui/           # Frontend pages
| — chatbot.py     # Conversational assistant
| — policy_summarizer.py # Summarizes government documents
| — feedback_analyzer.py # Sentiment and feedback analytics
| — anomaly_detector.py # Detects unusual service issues
| — report_generator.py # Generates citizen engagement reports
```

6. Running the Application

1. Start backend server (FastAPI).
 2. Launch frontend dashboard.
 3. Use sidebar to navigate: Chatbot, Feedback, Service Tracking, Policy Summaries.
 4. Upload documents or enter queries.
 5. View outputs: summaries, analytics, engagement reports.
-

7. API Documentation

- **POST /chat/ask** – Citizen queries → AI response
 - **POST /upload-policy** – Uploads government policy → summarized content
 - **GET /search-docs** – Search across policies and FAQs
 - **POST /submit-feedback** – Citizens submit feedback/complaints
 - **GET /citizen-insights** – Provides dashboards/analytics for officials
-

8. Authentication

- JWT-based authentication for citizens & officials
 - Role-based access (Admin, Citizen, Analyst)
 - OAuth2 support for secure government integration
-

9. User Interface

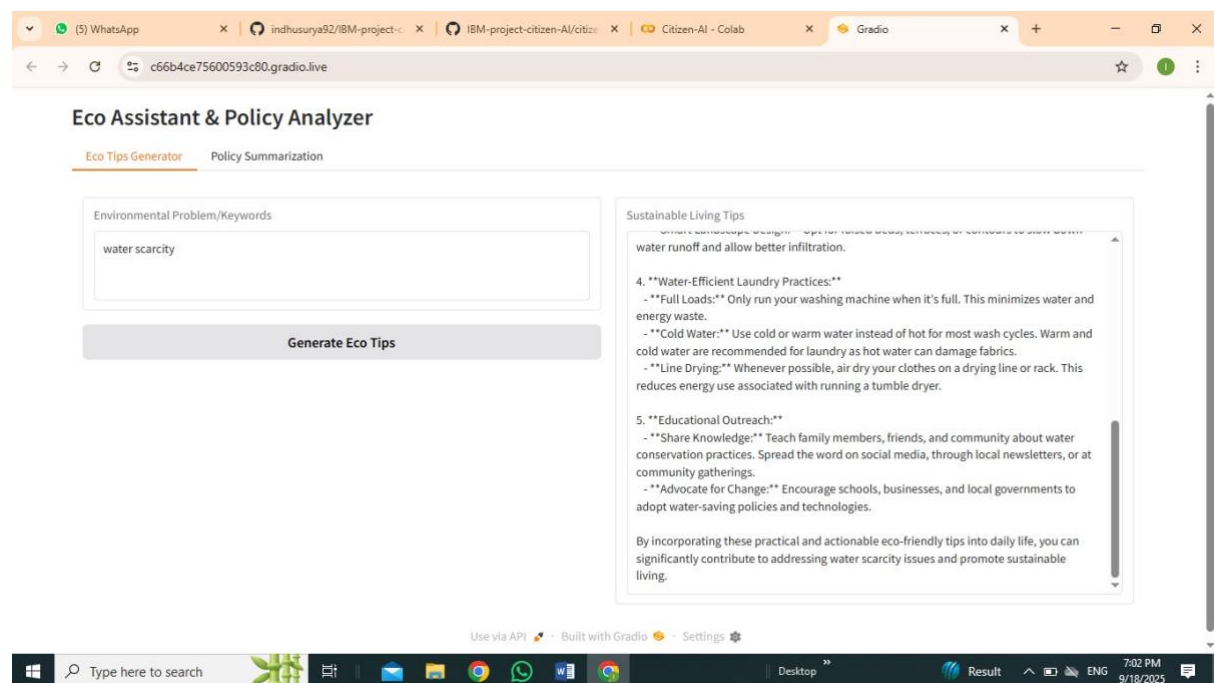
- Multilingual chatbot for citizens
- Service tracking dashboard

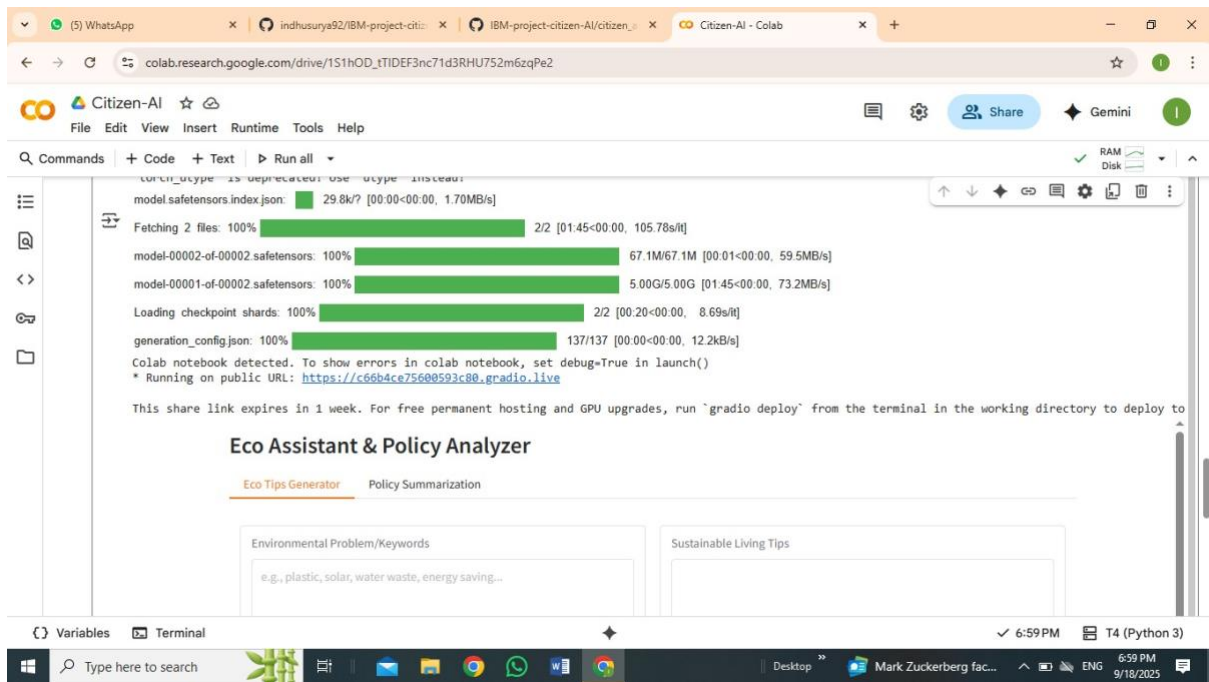
- Feedback submission form
- Visual analytics for officials
- PDF/CSV report download

10. Testing

- **Unit Testing** – Chat, policy summarization, feedback analysis
- **API Testing** – Swagger/Postman
- **Manual Testing** – Service queries, feedback loop
- **Edge Cases** – Large documents, multilingual inputs, incorrect queries

11. SCREEN SHOTS





12. KNOWN ISSUES

- ☐ Requires high-quality training data for policy understanding
- ☐ Multilingual NLP performance varies by language
- ☐ Dependency on API rate limits

13.FUTURE ENHANCEMENT

- ☐ Mobile app for wider accessibility
- ☐ Voice-based interactions for rural citizens
- ☐ Integration with WhatsApp/SMS channels
- ☐ Advanced predictive analytics for governance