

Project Final Report

Citizen AI – Intelligent Citizen Engagement Platform.

1. INTRODUCTION:

1.1 Project Overview

The Citizen AI Assistant is a civic-tech platform powered by IBM Watsonx AI and Streamlit. It helps citizens interact with local governance digitally by reporting issues, analyzing sentiment, and accessing public service information. The assistant uses predefined responses and AI fallback for dynamic citizen support.

1.2 Purpose

The purpose is to empower responsible citizens by providing an intelligent platform to raise civic concerns, gain awareness about public services, and promote government transparency through a real-time AI assistant backed by IBM Granite models and interactive dashboards.

2. IDEATION PHASE:

2.1 Problem Statement:

Citizens often face issues like potholes, waste, and service delays but struggle to report them or track resolution. Existing systems are inefficient, lack feedback, and don't analyze public sentiment. This leads to frustration, disengagement, and lack of trust in local governance.

2.2 Empathy Map Canvas

We analyzed citizen emotions, thoughts, and behaviors. Citizens want fast issue redressal, clear service updates, and easy complaint filing. They feel unheard, unaware of processes, and worry about corruption.

This inspired us to design a transparent, AI-powered assistant that listens, responds, and reports.

2.3 Brainstorming

We conducted team brainstorming to ideate civic-tech features: AI chat assistant, issue reporting form, real-time sentiment dashboard, public service FAQs, and smart data storage. Integration with IBM Watsonx emerged as a robust AI solution, enabling scalability, multi-language support, and trustworthy responses.

3. REQUIREMENT ANALYSIS:

Users visit the platform, ask civic questions, report issues, or give feedback. The AI processes the query, checks predefined responses, and replies accordingly. Sentiment is analyzed and stored, and the dashboard updates in real-time with user reports and analytics.

3.2 Solution Requirement

Functional requirements include AI chat interface, issue reporting module, login system, and a sentiment-based dashboard. Non-functional needs include scalability, data privacy, low latency, and cross-device support. The AI must fallback to IBM Watsonx if no predefined answer exists.

3.3 Data Flow Diagram

User inputs → Frontend → Predefined Issue Check → IBM Watsonx (fallback) → Output → Sentiment Analysis → Save to JSON → Dashboard displays updates. This flow ensures real-time response generation, data tracking, and visual reporting for administrators.

3.4 Technology Stack

Frontend: Streamlit, HTML, CSS

Backend: Python

AI: IBM Watsonx (Granite-3-3-2b-instruct)

Storage: JSON files (data.json, issues.json)

Hosting: Streamlit Cloud / GitHub

Version Control: GitHub
Deployment: Streamlit Share

4. PROJECT DESIGN:

4.1 Problem Solution Fit

The assistant fits the civic engagement gap by offering an intelligent, easy-to-use platform where citizens report, communicate, and see insights. The blend of static knowledge and AI ensures reliable information with real-time adaptability.

4.2 Proposed Solution

We propose a multi-page web assistant with login, chat, dashboard, and issue reporting modules. It uses predefined JSON responses for accuracy and IBM Watsonx for dynamic replies, backed by real-time sentiment analysis and graphical visualization.

4.3 Solution Architecture

Frontend interfaces interact with the backend Python modules. If a query matches a predefined issue, the answer is shown. If not, it's sent to IBM Watsonx via secure API. Sentiment is analyzed and all data is logged to JSON for dashboard visualization.

5. PROJECT PLANNING & SCHEDULING:

5.1 Project Planning

The project was executed in phases: Ideation, Design, Development, Testing, and Deployment. We used GitHub for version control and Streamlit Cloud for deployment. Task milestones were tracked weekly to ensure timely progress and delivery.

6. FUNCTIONAL AND PERFORMANCE TESTING:

6.1 Performance Testing

The AI assistant was tested with multiple queries, both matching and non-matching predefined issues. API latency with Watsonx remained

under 1.5 seconds. Dashboard updates were tested under concurrent user loads. JSON handling and real-time UI rendering passed stability benchmarks.

7. RESULTS:

7.1 Output Screenshots

Screenshots include:

- Homepage with “Empowering Citizens” banner:

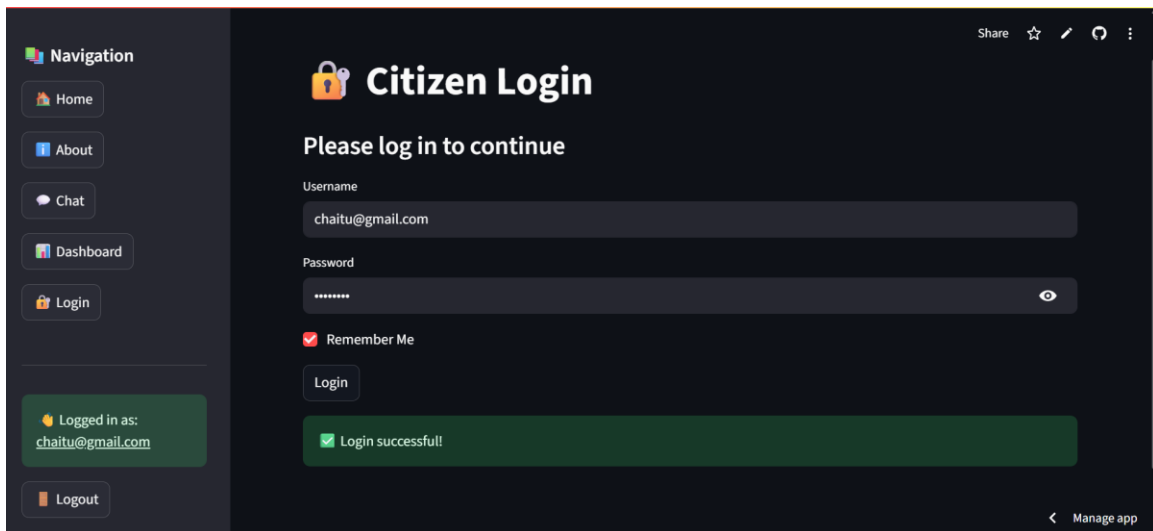


-

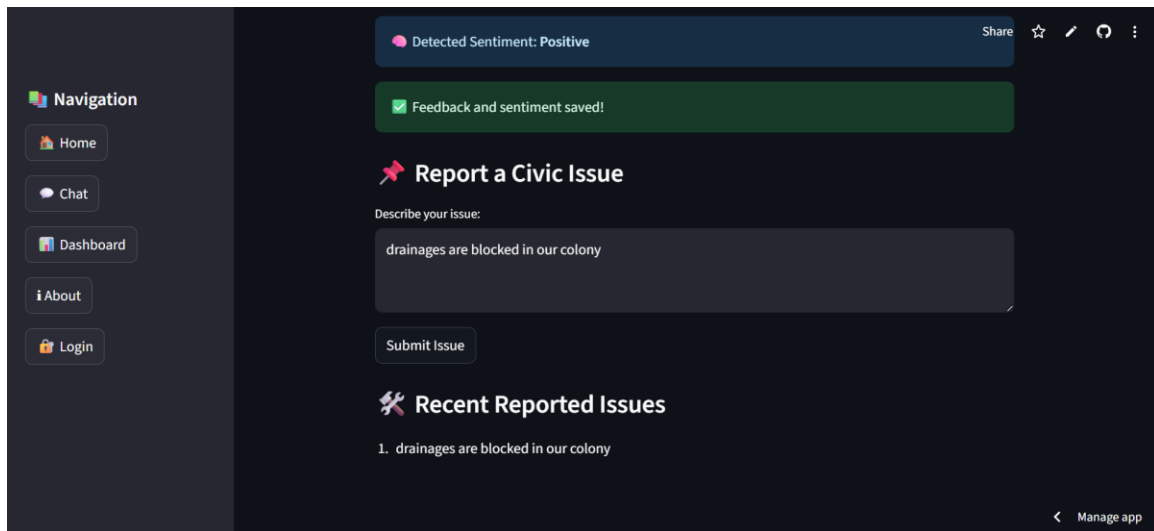
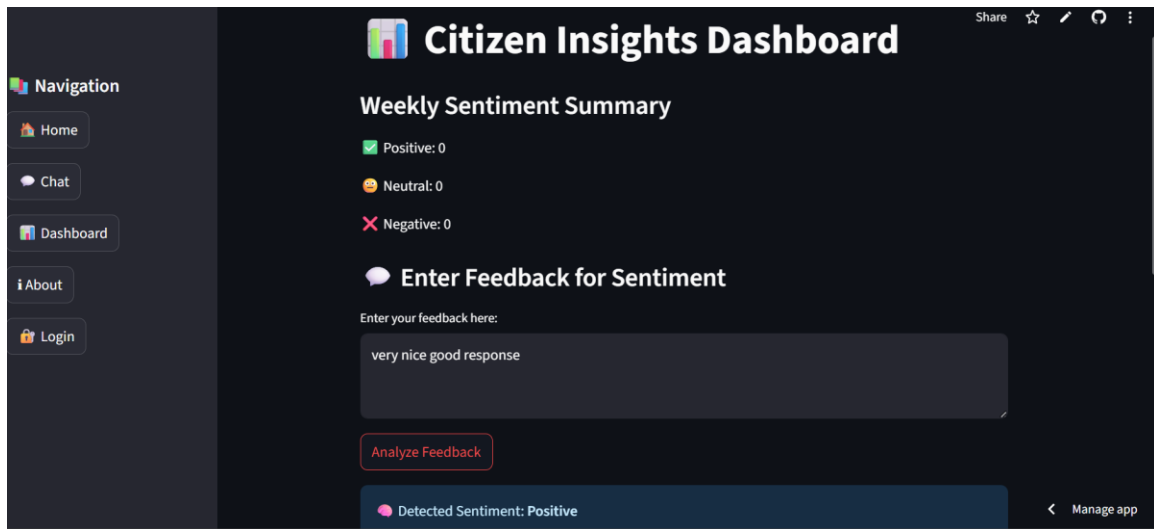
Chat page with AI conversations:



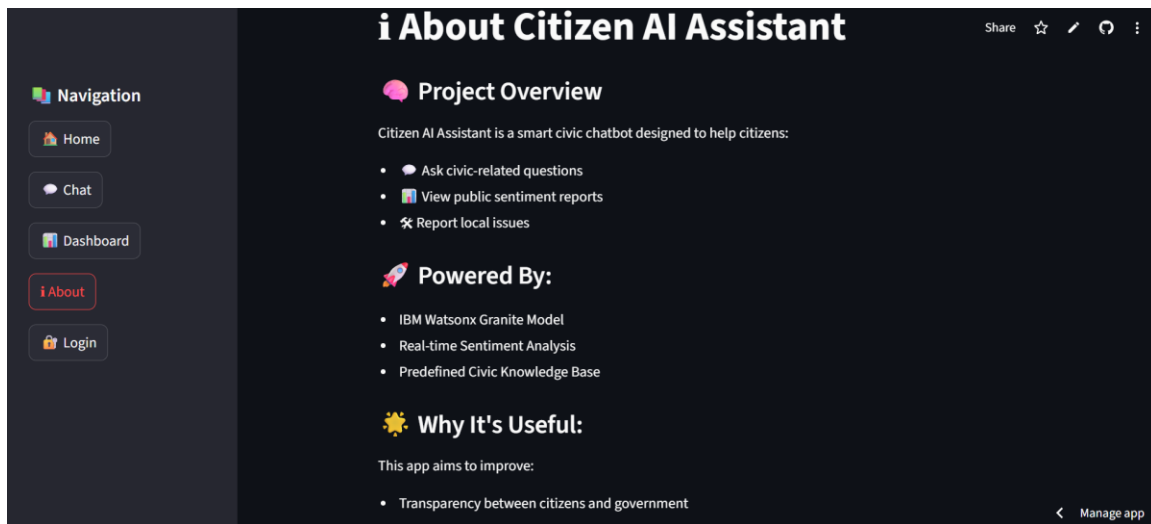
- Login page with validation:



- Dashboard showing weekly sentiment feedback, enter feedback for sentiment and citizen issues:



- About and Services pages detailing assistant capabilities:



8. ADVANTAGES & DISADVANTAGES:

Advantages:

- User-friendly AI interface
- Real-time sentiment and issue tracking
- Transparent governance support
- Uses trusted IBM AI models
- Predefined issue handling reduces hallucinations

Disadvantages:

- Internet dependency
- Requires up-to-date predefined data
- Limited to predefined civic domains
- JSON storage may not scale for large data without DB integration

9. CONCLUSION:

The Citizen AI Assistant enhances civic participation using AI. It allows users to raise concerns, get answers, and track public sentiments in one place. The integration with IBM Watsonx ensures trust and scalability, making it a valuable civic-tech innovation.

10. FUTURE SCOPE:

Add multilingual support

Replace JSON with cloud database (e.g., Firebase or DB2)

Integrate with municipal APIs (Swachh Bharat, MyGov)

Expand domain to legal rights, RTI, pensions

Add voice and WhatsApp chatbot interface

11. APPENDIX:

Source Code

GitHub Repository – Citizen-ai-assistant

Dataset Link

predefined_issues.json: civic issues and answers

data.json: sentiment and issue records

(Include these files in your appendix)

GitHub & Project Demo Link

GitHub: <https://github.com/chaitanyakanikicharla789/Citizen-ai-assistant>