# PROJECT REPORT

## Title: Citizen AI – Intelligent Citizen Engagement Platform

## 1. INTRODUCTION

### 1.1 Project Overview :

Citizen AI is an intelligent citizen engagement platform built using Python, Flask, and IBM Watson APIs. It enables real-time conversational interfaces for public service queries, sentiment analysis, and feedback management. The system integrates NLP, AI, and visual analytics to empower government officials with data-driven governance tools.

### 1.2 Purpose:

The platform aims to modernize civic engagement by providing a centralized, AI-powered solution where citizens can interact with public services naturally. It helps streamline issue reporting, improve feedback collection, and enhance transparency in governance.

## 2 IDEATION PHASE

### 2.1 Problem Statement:

PS-1: I am a citizen trying to access government service information online. I'm trying to get accurate and quick answers about policies and services. But the official websites are hard to navigate and have outdated information. Because there's no centralized, user-friendly, or real-time system in place. Which makes me feel frustrated and unheard.

### 2.2 Empathy Map Canvas

The empathy map captures what users say, think, feel, do, see, hear, and their pains and gains. It helps identify core frustrations and aspirations of both citizens and government officials.

| Category | Description |
|---|---|
| Says | "I need to report a civic issue quickly." "Who will respond to my complaint?" |
| Thinks | "Will the authorities take action?" "Is this platform reliable?" |
| Does | Tries to call helplines, visits government websites, posts complaints on social media |

| | |
|---|---|
| Feels | Frustrated by delays, unsure about where to report, hopeful for quick resolution |
| Pains | Slow or no responses, complex complaint systems, lack of transparency |
| Gains | Fast complaint registration, real-time updates, AI-based suggestions for better urban |

### 2.3 Brainstorming

The team explored various AI-powered solutions and prioritized ideas based on feasibility and impact. Key ideas included real-time chatbot, sentiment-based analytics, multilingual support, and a dynamic dashboard for officials.

---

## 3. REQUIREMENT ANALYSIS

### 3.1 Customer Journey Map:

The journey begins with a user query, proceeds through sentiment analysis and feedback submission, and ends at visualization for officials. Touchpoints include chatbot, feedback forms, and the dashboard.

### 3.2 Solution Requirement:

Real-time chatbot integration

Sentiment analysis service

Feedback storage and retrieval

Admin dashboard for visualization

### 3.3 Data Flow Diagram:

Citizen input flows into the Flask server, routes through the IBM Watson API, and results are stored in a SQLite/NoSQL DB. Insights are displayed on the dashboard.

### 3.4 Technology Stack:

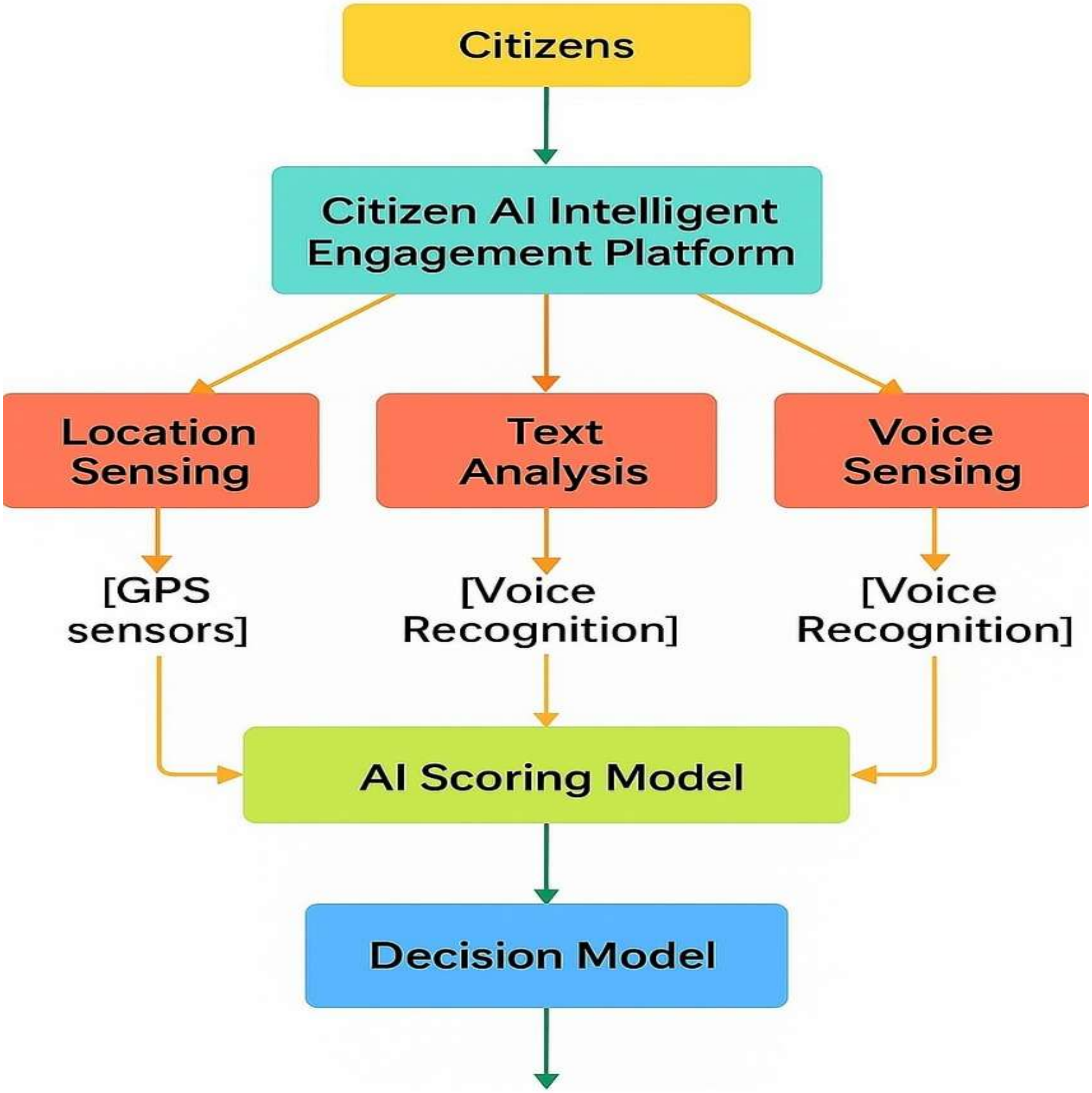
**Frontend:** HTML, CSS (via Flask templates)

**Backend:** Python Flask

**Database:** SQLite

**AI Integration:** IBM Watson/Granite APIs

**Data Flow Diagram:**



## 4. PROJECT DESIGN

### 4.1 Problem Solution Fit:

The solution directly addresses the lack of centralized, intelligent platforms for civic interaction and real-time feedback processing.

### 4.2 Proposed Solution:

An AI-based platform that enables citizens to engage with services, analyze sentiment, and allows officials to monitor and act on insights through a dashboard.

### 4.3 Solution Architecture:

Built using a layered structure:

**Frontend:** HTML/CSS

**Backend:** Flask routes and logic (app.py)

**AI Integration:** IBM Watson for NLP/sentiment

**Storage:** SQLite DB

## 5. PROJECT PLANNING & SCHEDULING

### 5.1 Project Planning:

The project followed an Agile approach, organized in multiple 5-day sprints with story points, backlog planning, and burndown tracking. Sprint velocity was calculated at 12 SP/sprint**.**

### *5.2 Project Planning*

| **Week Duration** | **Dates** | **Activities** |
|---|---|---|
| Week 1 | June 12 – June 19 | Idea finalization, architecture planning, frontend UI with javascript |
| Week 2 | June 20 – June 26 | Backend AI integration, testing, debugging, and documentation |

This two-week schedule allowed the team to focus on clear milestones and complete the HealthAI project within the planned timeline.

# 6. FUNCTIONAL AND PERFORMANCE TESTING

## 6.1 Performance Testing

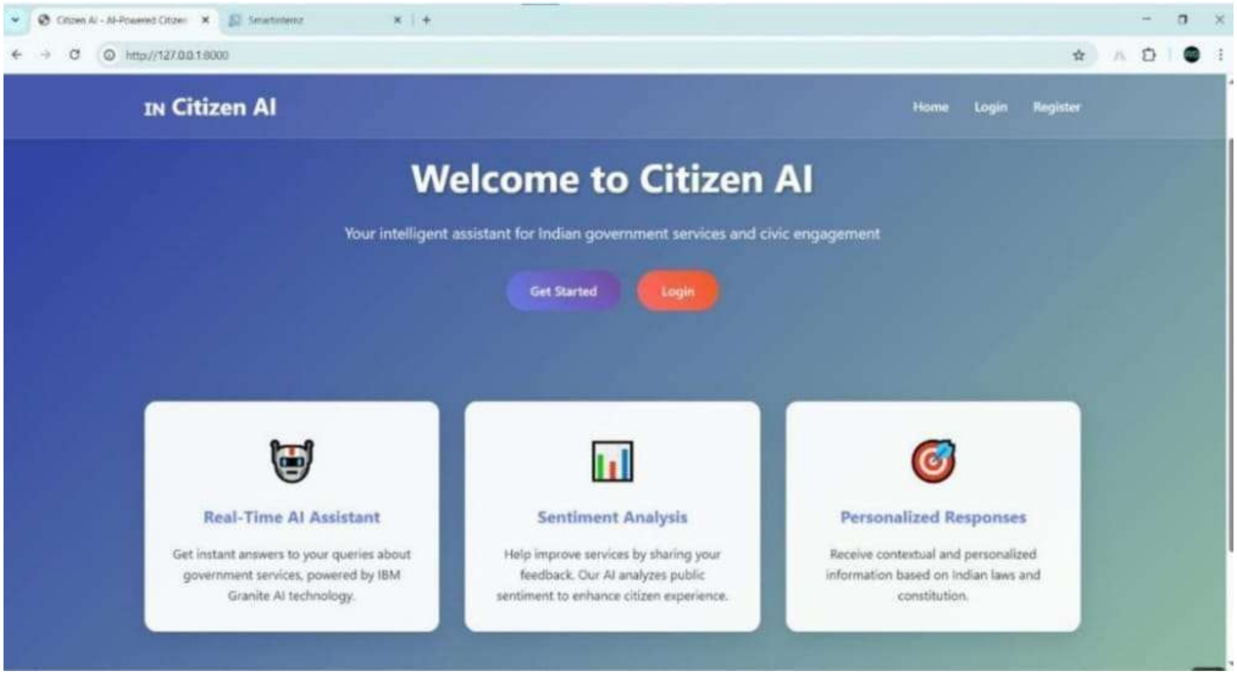
Manual testing confirmed input validation, API responsiveness, and sentiment accuracy.

Load and response time tests ensured stable performance under multiple requests.

# 7. RESULTS

## 7.1 Output Screenshots

- Chatbot UI
- Feedback submission
- Sentiment dashboard



Welcome to Citizen AI

Your intelligent assistant for Indian government services and civic engagement

Real-Time AI Assistant

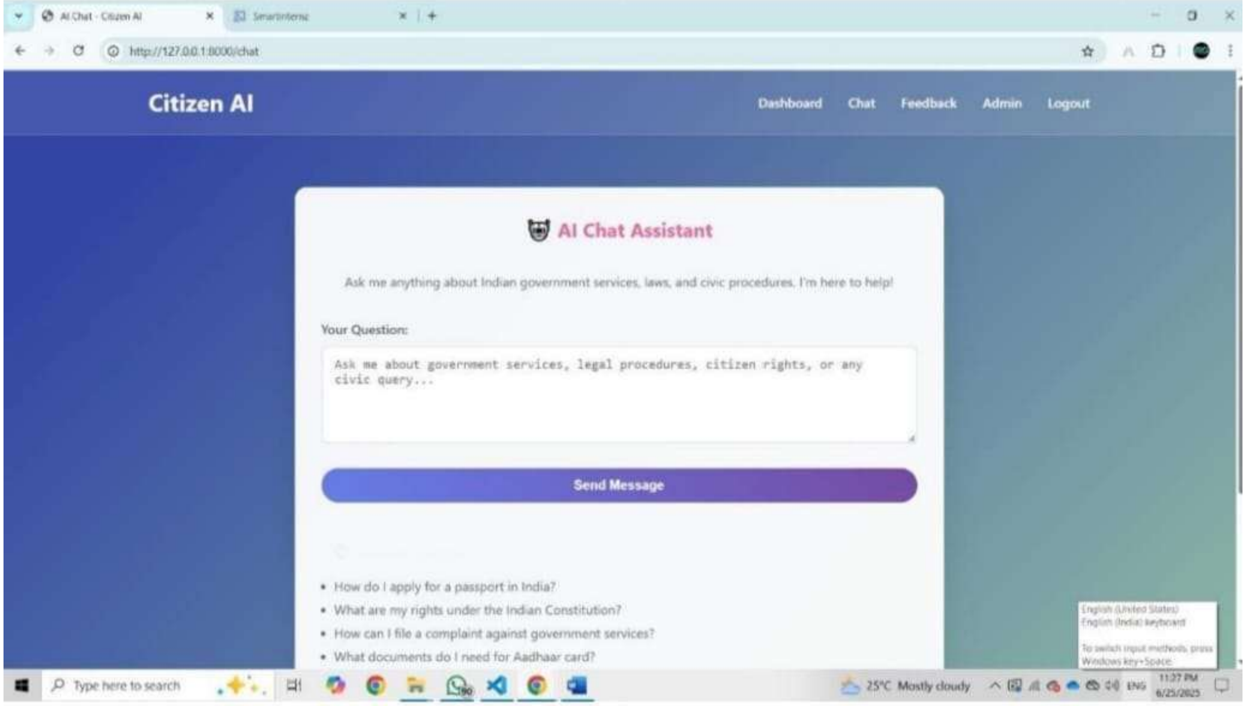
Get instant answers to your queries about government services, powered by IBM Granite AI technology.

Sentiment Analysis

Help improve services by sharing your feedback. Our AI analyzes public sentiment to enhance citizen experience.

Personalized Responses

Receive contextual and personalized information based on Indian laws and constitution.



AI Chat Assistant

Ask me anything about Indian government services, laws, and civic procedures. I'm here to help!

Your Question:

Ask me about government services, legal procedures, citizen rights, or any civic query...

- How do I apply for a passport in India?
- What are my rights under the Indian Constitution?
- How can I file a complaint against government services?
- What documents do I need for Aadhaar card?

## 8. ADVANTAGES & DISADVANTAGES

**Advantages:**

Real-time citizen support

Centralized dashboard for officials

Scalable and cloud-ready architecture

**Disadvantages:**

No authentication for admin dashboard

Basic error handling for API timeouts

## 9.CONCLUSION :

Citizen AI bridges the communication gap between governments and citizens using AI and automation. The platform enhances feedback management and governance with real-time, data-driven insights.

## 10. FUTURE SCOPE:

11. OAuth integration for secure login
12. Cloud deployment (AWS/GCP)
13. Fine-tuning with IBM Granite models
14. Multilingual support

## 15.APPENDIX :

**Source Code**: Available in `CITIZEN_AI (WITH API KEY)`

**Dataset**: NA (real-time inputs only)

**GitHub Link:** https://github.com/LukkaSanjay/sustainable-smart-city

**Source Code Files:main.py,index.html,style.css,map.js,chart.js**

**Source Code**: Available in CITIZEN_AI (WITH API KEY)

**Dataset**: NA (real-time inputs only)