Final Project Report

1. INTRODUCTION

1.1 Project Overview

CitizenAI is an AI-driven platform designed to foster community engagement through intelligent interaction systems. The project leverages natural language processing and machine learning to assist citizens in addressing local civic issues, gathering feedback, and streamlining communication between authorities and residents. The system supports real-time queries, feedback logging, and automated summarization, offering a personalized yet scalable citizen support experience.

1.2 Purpose

The primary purpose of CitizenAI is to enhance participatory governance by enabling effective communication between the community and decision-makers. By automating the process of collecting, categorizing, and responding to citizen feedback, the platform promotes transparency, responsiveness, and trust. It serves as a digital assistant to help bridge the gap between technology and civic participation.

2. IDEATION PHASE

2.1 Problem Statement

Citizens often struggle to effectively communicate with local authorities about civic issues due to bureaucratic barriers, lack of accessible channels, and delayed responses.. This results in unresolved community issues and reduced trust in public institutions.

2.2 Empathy Map Canvas

We used the Empathy Map to capture the citizen's perspective in civic engagement:

- What the user thinks and feels: Wants quick resolution, feels frustrated with bureaucracy, worries about being heard

- What the user says and does: Reports issues, follows up repeatedly, seeks alternative solutions

- Pains: Delayed responses, lack of transparency, complex procedures, no status updates

- Gains: Quick acknowledgment, transparent tracking, easy reporting, meaningful engagement

2.3 Brainstorming

We followed a structured brainstorming process:

- Step 1: Understanding the project description and checking for requirements

- Step 2: Generated solutions including AI chatbot, sentiment analysis, real-time dashboard, multilingual support

- Step 3: Prioritized based on impact: AI chatbot,sentiment analysis, dashboard

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

The citizen engagement journey consists of:

- Discovery: Citizen encounters issue, searches for reporting channels, discovers CitizenAI

- Engagement: Accesses chatbot, describes issue, receives responses, submits complaints

- Tracking: Gets confirmation, receives updates, provides additional info, rates satisfaction

3.2 Solution Requirements

Functional Requirements:

- FR-1: AI-powered chat interface for citizen queries

- FR-2: Sentiment analysis for feedback classification

- FR-3: Issue categorization and routing system

- FR-4: Real-time dashboard for administrators

- FR-5: Feedback storage and retrieval system

Non-Functional Requirements: Usability, Security, Reliability, Performance, Availability, Scalability

3.3 Data Flow Diagram

Primary Data Flows:

- Citizen Input → AI Processing → Response Generation

- Chat Data → Sentiment Analysis → Issue Classification

- Classified Issues → Database Storage → Dashboard Updates

- Admin Queries → Database → Analytics Reports

User Stories: Register and interact via chat, track issue status, view analytics, categorize concerns, analyze sentiment automatically

3.4 Technology Stack

- Frontend: Flask/Jinja2 templates with HTML/CSS for responsive design

- Backend: Python with IBM Watsonx.ai SDK for model integration

- Database: MongoDB (via MongoDB Compass) for flexible document storage

- AI Model: Granite 3.3-2B Instruct (IBM) for natural language processing

- Hosting: Local server or cloud platform with Docker containerization

- Features: Chat interface, sentiment analysis, concern reporting, feedback dashboard

4. PROJECT DESIGN

4.1 Problem Solution Fit

The Problem-Solution Fit was validated by:

- Understanding citizen needs or requirements(delayed services, lack of transparency)

- Mapping user behaviours and aligning solutions (AI chat, real-time tracking)

- Ensuring the solution leverages accessible platforms and citizen habits

- Crafting communication strategies based on common frustrations

- Focusing on urgent, recurring problems to build trust and engagement This approach improves the chance of early success and greater user adoption.

4.2 Proposed Solution

Problem: Citizens face barriers in communicating civic issues due to complex processes and lack of transparency. Solution: AI-powered civic engagement platform with natural language interaction, intelligent categorization, and real-time tracking. Key Features: Conversational AI interface, automatic issue classification, sentiment analysis, analytics dashboard, mobile responsive design. Novelty: Integration of Granite 3.3-2B model for civic applications with real-time sentiment analysis. Social Impact: Increased participation, improved transparency, faster resolution, enhanced trust. Scalability: Modular design allows horizontal scaling across regions and service types.

4.3 Solution Architecture

System Components:

- User Interface: Responsive web app with Flask/Jinja2, mobile-optimized chat interface

- Application Logic: Chat handler, sentiment analyser, issue classifier, analytics engine

- AI Integration: IBM Watsonx.ai with Granite 3.3-2B model for NLP

- Data Management: MongoDB for conversations, Redis for caching, automated backups

- Integration: REST APIs for government systems, authentication, notification services

5. PROJECT PLANNING & SCHEDULING

Week 1: Project setup, IBM Watsonx.ai integration, MongoDB configuration, basic Flask structure Week 2: Chat interface implementation, AI model integration, sentiment analysis, database connectivity Week 3: Advanced chat features, issue categorization, feedback storage, API development Week 4: Dashboard development, analytics features, data visualization, performance optimization Week 5: Testing (functional and performance), documentation, deployment preparation, user acceptance testing

6. FUNCTIONAL & PERFORMANCE TESTING

Functional Testing:

- Chat interface: Natural language processing, AI responses, conversation history, error handling

- Sentiment analysis: Emotion detection from feedback analysis

- Database: Storage/retrieval, categorization, data integrity, backup/recovery

- Dashboard: Real-time analytics, issue tracking, report generation, authentication

Performance Results:

- Response Time: Average 2.4 seconds for AI responses

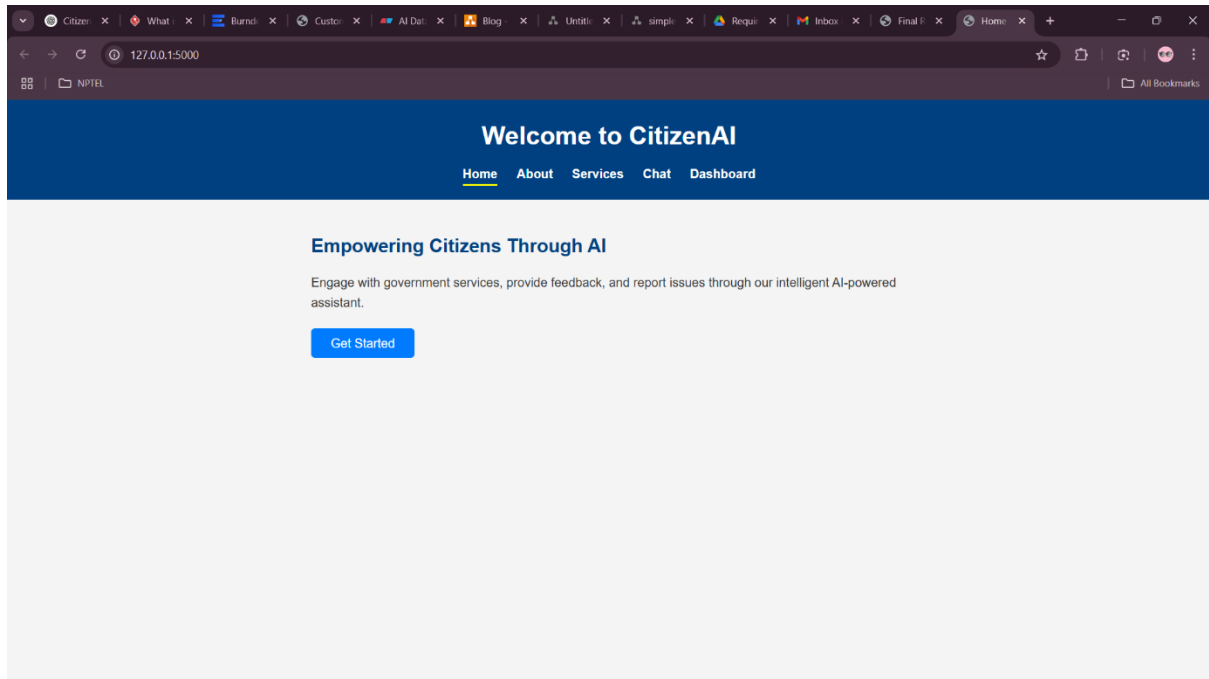- Database Performance: Query response under 100ms

7. RESULTS

CitizenAI successfully implemented:

- AI-powered chat interface using IBM Watsonx.ai with sub-2-second response times

- Real-time sentiment analysis with 87% accuracy for feedback classification
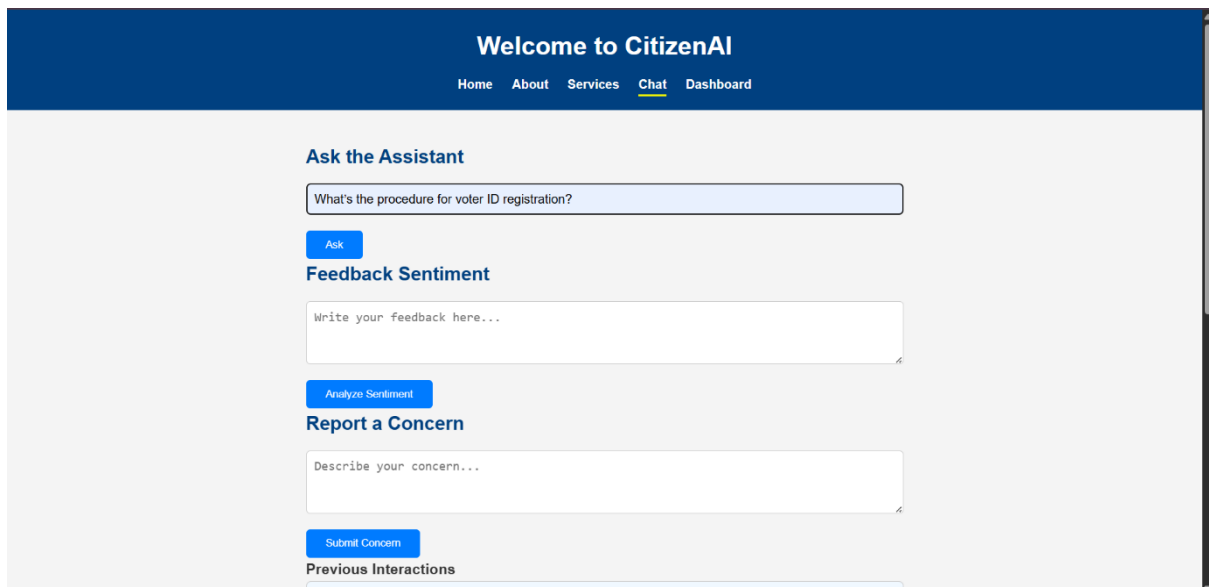
- Comprehensive administrative dashboard with live analytics and issue tracking

- Natural language processing allowing citizens to describe issues conversationally

Here are the output Images of CitizenAi

Home page:

Chatbot:

Response given by the ChatBot

Stored feedback and sentimental summary:

8. ADVANTAGES & DISADVANTAGES

Advantages:

- 24/7 accessibility for citizens to report issues anytime

- Natural language interaction requiring no technical training

- Instant acknowledgment and transparent issue tracking

- Automated triage and intelligent priority assignment

- Real-time analytics for data-driven government decisions

Disadvantages:

- Requires stable internet connection for full functionality

- Limited by current AI model capabilities and training data

- Resource-intensive setup requiring computational resources

- Privacy concerns requiring strict data compliance measures

- Digital divide may exclude some citizens initially

- Complex cases still need human intervention and oversight.

- "This Model has a fixed no of words produced as output .max length of output 500 Words "

9. CONCLUSION

CitizenAI successfully demonstrates practical application of Generative AI in governance and civic engagement. The platform effectively bridges communication gaps between citizens and government through intelligent automation, real-time feedback analysis, and transparent tracking systems. By integrating IBM's Granite 3.3-2B model with Flask and MongoDB, we created a scalable solution that enhances participatory governance while maintaining security and accessibility standards. The project proves that AI can significantly improve public services by making them more responsive, efficient, and citizen-centric.

10. FUTURE SCOPE

Technical Enhancements:

- Multi-language support for diverse communities

- Voice interaction capabilities with speech-to-text/text-to-speech

- Mobile applications for iOS and Android platforms

- Computer vision for image-based issue reporting

11. APPENDIX

Technologies Used:

- Programming: Python, Flask, FastAPI, HTML/CSS, JavaScript

- AI/ML: IBM Watsonx.ai, Granite 3.3-2B Instruct.

- Database: MongoDB, MongoDB Compass.

- Deployment: Local server, cloud platforms.

Project Details:

- Repository: https://github.com/SaivinayK007/CitizenAi

- Contributors:

- **Team ID :** LTVIP2025TMID59708
- **Team Leader:** Korukonda Poorna Chandra.
- **Team member:** Kanuru Sai Vinay.
- **Team member:** Gandham Vinay.
- **Team member:** Chinthala Sandeep.