

Full Stack Development Project Report

1. Introduction

Project Title: Citizen AI – Intelligent Citizen Engagement Platform

Team Members & Roles:

Name	Role
Tamaraana Sruthi	Team Lead, Flask Integration & UI Design
Pothabattthula Sandeep	LLM Integration & Backend Routes
Nakka Santhosh Kumar	Feedback Module & Sentiment Analysis
Thadicherla Chandra Manikanta Sri Devi Prasad	Admin Dashboard & Deployment

2. Project Overview

Purpose:

Citizen AI enables citizens to communicate with government services using a conversational assistant. They can ask questions, submit issues, and view public sentiment trends.

Features:

- AI-powered chatbot using IBM Granite
- Feedback and issue submission
- Sentiment analysis module
- Admin dashboard to monitor issues
- Login system for access control

3. Architecture

- Frontend: HTML, CSS, Bootstrap
- Backend: Python Flask
- AI: IBM Granite via Hugging Face

- Deployment: Render
- Data: In-memory (scalable to Firebase/Firestore)

4. Setup Instructions

Prerequisites:

- Python 3.10+
- pip
- Hugging Face token (optional)

Steps:

```
git clone https://github.com/shruthitamaraana/Citizen_AI-Intelligent-citizen-engagement-  
platform.git  
cd Citizen_AI-Intelligent-citizen-engagement-platform/code  
pip install -r requirements.txt  
python app/main.py
```

5. Folder Structure

```
code/  
├── app/  
│   ├── routes/  
│   ├── templates/  
│   ├── static/  
│   ├── ai_model.py  
│   └── main.py  
├── requirements.txt  
├── .render.yaml  
└── README.md
```

6. Running the Application

```
python app/main.py  
Visit: https://citizen-ai.onrender.com
```

7. API Documentation

Endpoint	Method	Description
/chat	POST	Sends user query to IBM Granite
/feedback	POST	Captures citizen feedback
/dashboard	GET	Displays admin dashboard
/login	GET/POST	Handles user authentication

8. Authentication

- Email-based login for users
- Admin authentication with route protection

9. User Interface

- Chat page for LLM Q&A
- Feedback submission form
- Sentiment-tagged admin dashboard
- Bootstrap-based responsive design

10. Testing

- Manual UI testing
- Backend logic tested with unit test scripts

11. Screenshots / Demo

Live App: <https://citizen-ai.onrender.com>

12. Known Issues

- Temporary in-memory feedback storage
- First LLM call is slow due to model warm-up
- No mobile optimization yet

13. Future Enhancements

- Persistent database storage
- Role-based user management
- SMS/Email notifications

- Voice input support
- Regional language expansion