1. Introduction

This project documentation outlines the purpose, features, architecture, and deployment of the Citizen AI platform.

* Project Title: Citizen AI: Intelligent Citizen Engagement Platform
* Team ID :NM2025TMID07260
* Team Leader : HARIHARASUDHAN J
* Team member: BOJARAJAN V
* Team member: MOHAMMAD THUFAIL U
* Team member: MAHENDRAN T
* Team member: KARTHIK G

2. Project Overview

The purpose of the Citizen AI: Intelligent Citizen Engagement Platform is to empower citizens and city officials to thrive in a more connected and efficient urban environment. By leveraging AI and real-time data, the platform helps optimize essential city services like energy, water, and waste, while also guiding citizens toward sustainable behaviors through personalized tips and services. For city officials, it serves as a decision-making partner, offering clear insights, forecasting tools, and summaries of complex policies to support strategic planning. Ultimately, this platform bridges technology, governance, and community engagement to foster more efficient, inclusive, and resilient cities.

Key Features:

* Conversational Interface
  * Key Point: Natural language interaction
  * Functionality: Allows citizens and officials to ask questions, get updates, and receive guidance in plain language.
* Policy Summarization
  * Key Point: Simplified policy understanding
  * Functionality: Converts lengthy government documents into concise, actionable summaries.
* Resource Forecasting
  * Key Point: Predictive analytics
  * Functionality: Estimates future energy, water, and waste usage using historical and real-time data.
* Eco-Tip Generator
  * Key Point: Personalized sustainability advice
  * Functionality: Recommends daily actions to reduce environmental impact based on user behavior.
* Citizen Feedback Loop
  * Key Point: Community engagement
  * Functionality: Collects and analyzes public input to inform city planning and service improvements.
* KPI Forecasting
  * Key Point: Strategic planning support
  * Functionality: Projects key performance indicators to help officials track progress and plan ahead.

* Anomaly Detection
  * Key Point: Early warning system
  * Functionality: Identifies unusual patterns in sensor or usage data to flag potential issues.
* Multimodal Input Support
  * Key Point: Flexible data handling
  * Functionality: Accepts text, PDFs, and CSVs for document analysis and forecasting.
* Streamlit/Gradio UI
  * Key Point: User-friendly interface
  * Functionality: Provides an intuitive dashboard for both citizens and city officials to interact with the platform.

3. Architecture

The platform is built with a modular architecture to ensure scalability and maintainability.

Frontend (Streamlit)

The frontend is built with Streamlit, offering an interactive web UI with multiple pages including dashboards, file uploads, chat interface, feedback forms, and report viewers. Navigation is handled through a sidebar using the Streamlit option menu library. Each page is modularized for scalability.

Backend (Fast API)

Fast API serves as the backend REST framework that powers API endpoints for document processing, chat interactions, eco-tip generation, report creation, and vector embedding. It is optimized for asynchronous performance and easy Swagger integration.

LLM Integration (IBM Watsonx Granite)

Granite LLM models from IBM Watsonx are used for natural language understanding and generation. Prompts are carefully designed to generate summaries, sustainability tips, and reports.

Vector Search (Pinecone)

Uploaded policy documents are embedded using Sentence Transformers and stored in Pinecone. Semantic search is implemented using cosine similarity to allow users to search documents using natural language queries.

ML Modules (Forecasting and Anomaly Detection)

Lightweight ML models are used for forecasting and anomaly detection using Scikit-learn. Time-series data is parsed, modeled, and visualized using pandas and matplotlib.

4. Setup Instructions

To get the project running, follow these steps.

Prerequisites:

* Python 3.9 or later
* pip and virtual environment tools
* API keys for IBM Watsonx and Pinecone
* Internet access to access cloud services

Installation Process:

* Clone the repository.
* Install dependencies from requirements.txt.
* Create a .env file and configure credentials.
* Run the backend server using Fast API.

* Launch the frontend via Streamlit.
* Upload data and interact with the modules.

5. Folder Structure

The project is organized into the following directories and files.

* app/ – Contains all Fast API backend logic including routers, models, and integration modules.
* app/api/ – Subdirectory for modular API routes like chat, feedback, report, and document vectorization.
* ui/ – Contains frontend components for Streamlit pages, card layouts, and form UIs.
* smart_dashboard.py – Entry script for launching the main Streamlit dashboard.
* granite_llm.py – Handles all communication with the IBM Watsonx Granite model including summarization and chat.
* document_embedder.py – Converts documents to embeddings and stores in Pinecone.
* kpi_file_forecaster.py – Forecasts future energy/water trends using regression.
* anomaly_file_checker.py – Flags unusual values in uploaded KPI data.
* report_generator.py – Constructs AI-generated sustainability reports.

6. Running the Application

To start the project:

* Launch the Fast API server to expose backend endpoints.
* Run the Streamlit dashboard to access the web interface.
* Navigate through pages via the sidebar.
* Upload documents or CSVs, interact with the chat assistant, and view outputs like reports, summaries, and predictions.
* All interactions are real-time and use backend APIs to dynamically update the frontend.

7. API Documentation

The backend APIs available are documented for easy integration.

* POST /chat/ask – Accepts a user query and responds with an AI-generated message.
* POST /upload-doc – Uploads and embeds documents in Pinecone.
* GET /search-docs – Returns semantically similar policies to the input query.
* GET /get-eco-tips – Provides sustainability tips for selected topics like energy, water, or waste.
* POST /submit-feedback – Stores citizen feedback for later review or analytics.

Each endpoint is tested and documented in Swagger UI for quick inspection and trial during development.

8. Authentication

This version of the project runs in an open environment for demonstration. However, secure deployments can integrate:

* Token-based authentication (JWT or API keys)
* OAuth2 with IBM Cloud credentials
* Role-based access (admin, citizen, researcher)
* Planned enhancements include user sessions and history tracking.

9. User Interface

The interface is minimalist and functional, focusing on accessibility for non-technical users. It includes:

* Sidebar with navigation

* KPI visualizations with summary cards
* Tabbed layouts for chat, eco tips, and forecasting
* Real-time form handling
* PDF report download capability

The design prioritizes clarity, speed, and user guidance with help texts and intuitive flows.

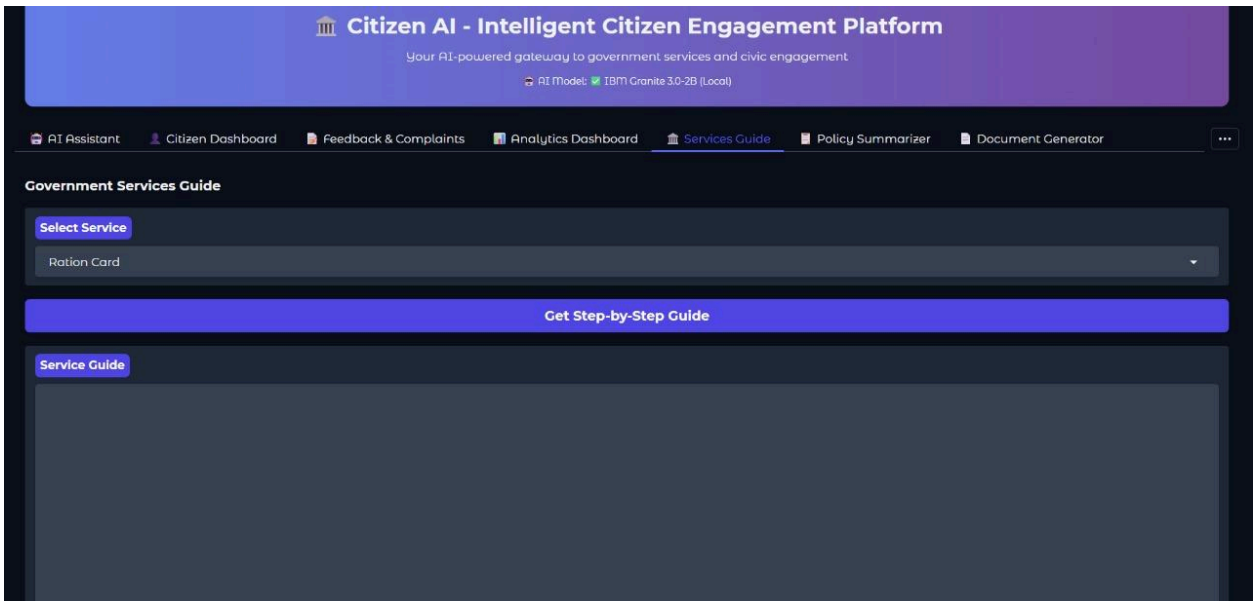
10. Testing

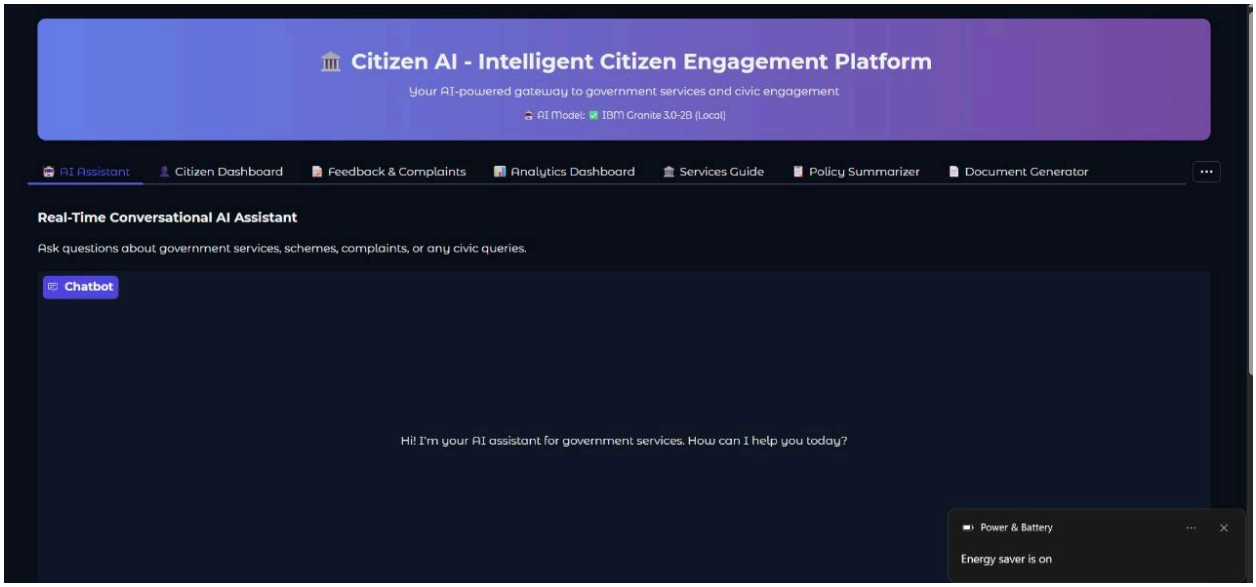
Testing was done in multiple phases:

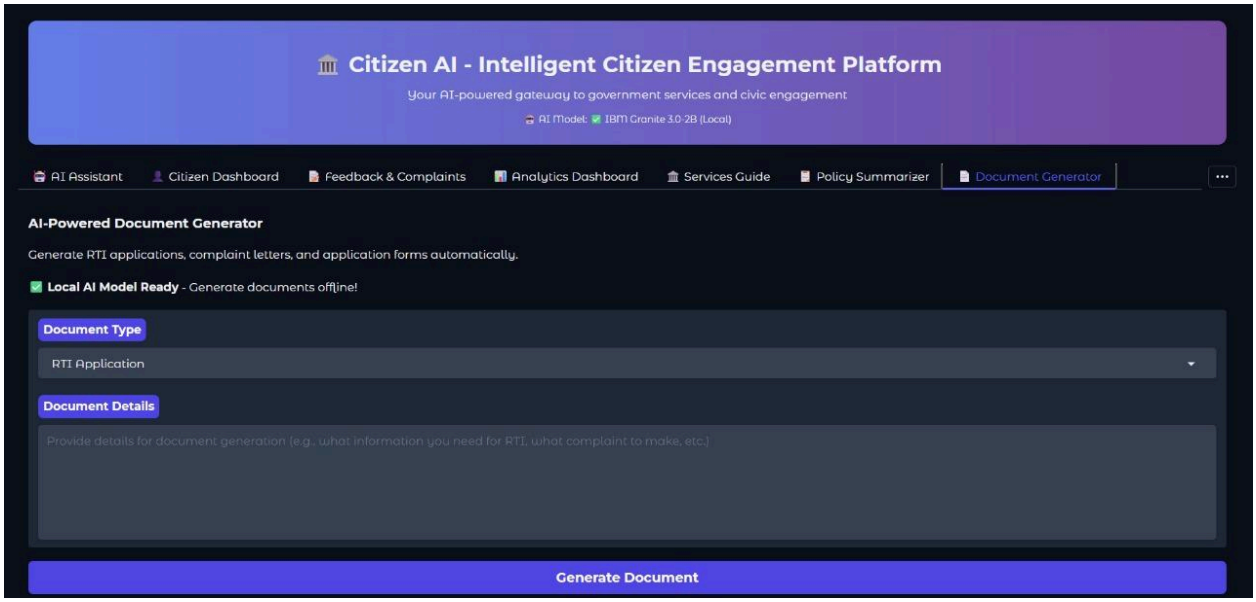
* Unit Testing: For prompt engineering functions and utility scripts.
* API Testing: Via Swagger UI, Postman, and test scripts.
* Manual Testing: For file uploads, chat responses, and output consistency.
* Edge Case Handling: Malformed inputs, large files, invalid API keys.

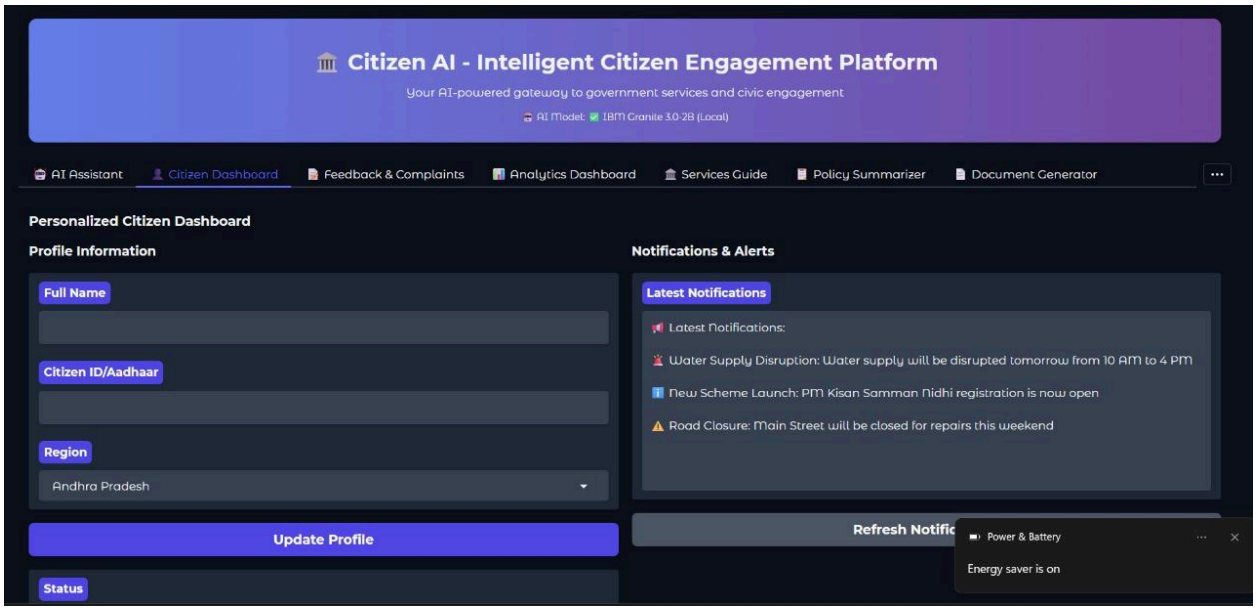
Each function was validated to ensure reliability in both offline and API-connected modes.

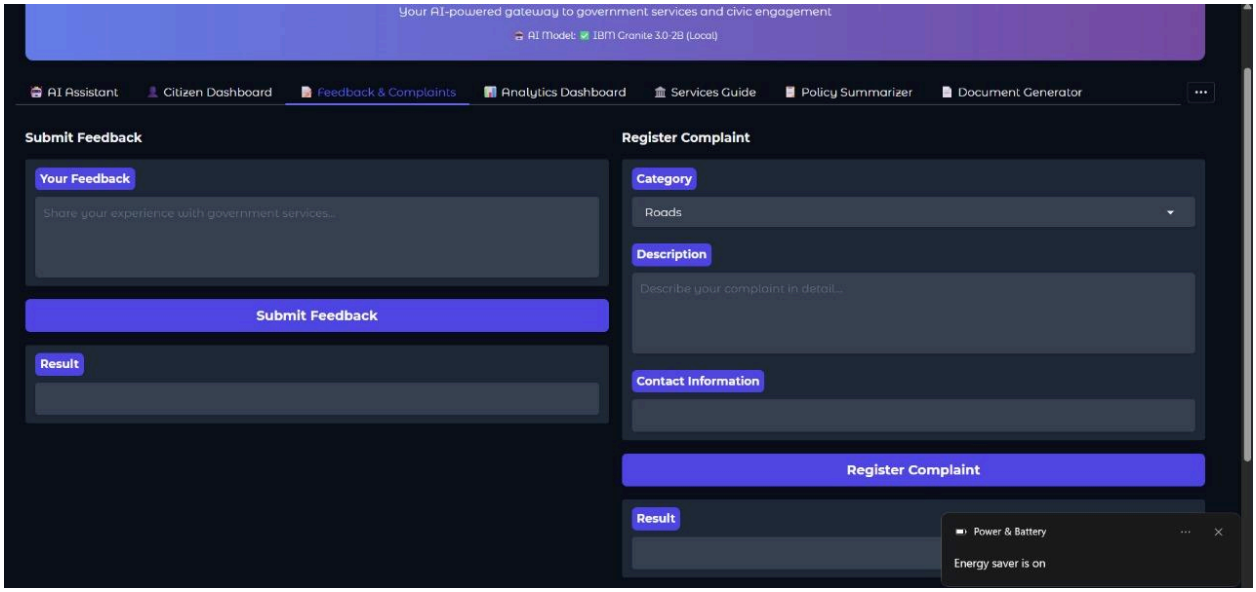
11. Screen Shots

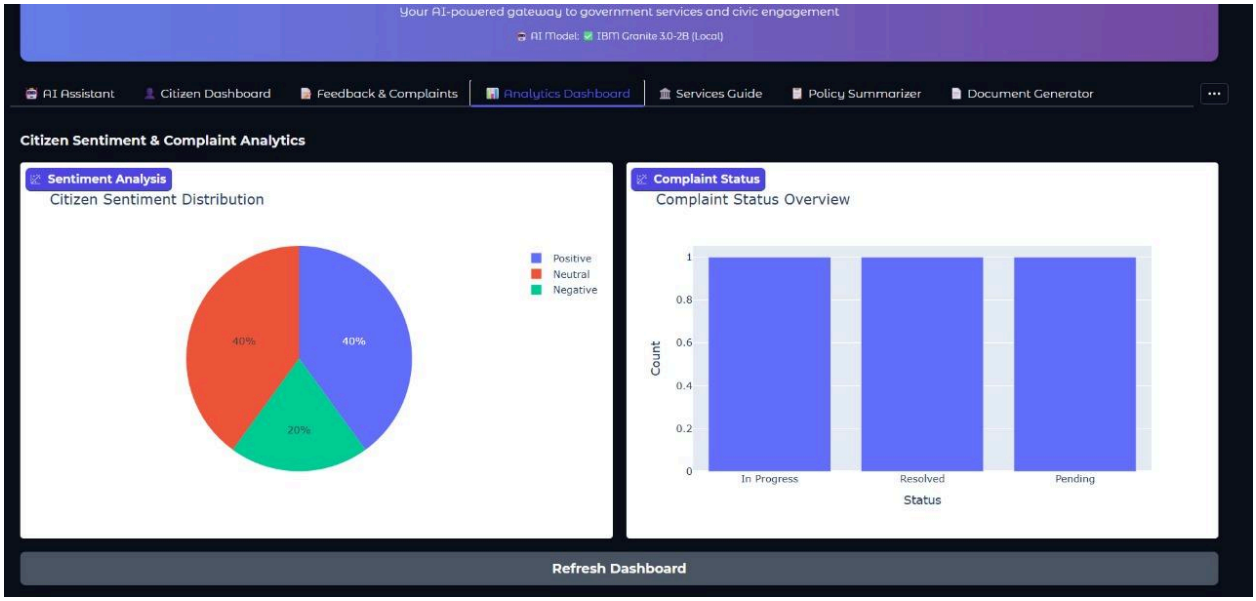


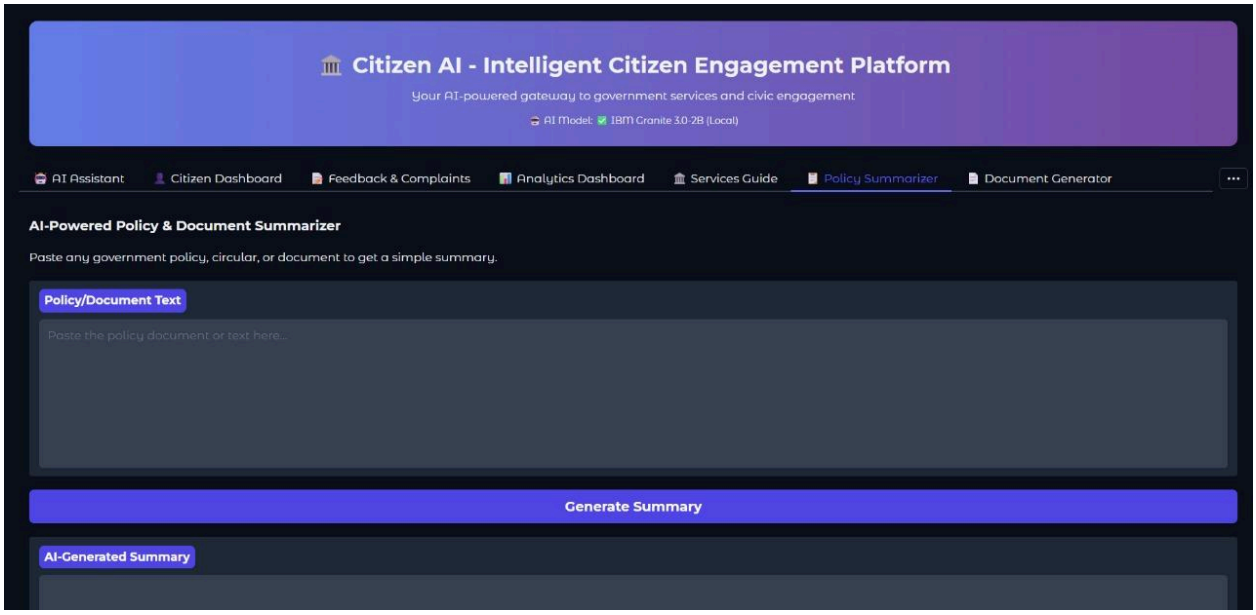



Citizen AI - Intelligent Citizen Engagement Platform
Your AI-powered gateway to government services and civic engagement
AI Model: IBM Granite 3.0-2B (Local)
AI Assistant
Citizen Dashboard
Feedback & Complaints
Analytics Dashboard
Services Guide
Policy Summarizer
Document Generator
AI-Powered Document Generator
Generate RTI applications, complaint letters, and application forms automatically.
Local AI Model Ready - Generate documents offline!
Document Type
RTI Application
Document Details
Provide details for document generation (e.g., what information you need for RTI, what complaint to make, etc.)
Generate Document


Citizen AI - Intelligent Citizen Engagement Platform
Your AI-powered gateway to government services and civic engagement
AI Model: IBM Granite 3.0-2B (Local)
AI Assistant
Citizen Dashboard
Feedback & Complaints
Analytics Dashboard
Services Guide
Policy Summarizer
Document Generator
Personalized Citizen Dashboard
Profile Information
Full Name
Citizen ID/Aadhaar
Region
Andhra Pradesh
Update Profile
Status
Notifications & Alerts
Latest Notifications
Latest Notifications:
Water Supply Disruption: Water supply will be disrupted tomorrow from 10 AM to 4 PM
New Scheme Launch: PM Kisan Samman Nidhi registration is now open
Road Closure: Main Street will be closed for repairs this weekend
Refresh Notifi
Power & Battery
Energy saver is on


Your AI-powered gateway to government services and civic engagement
AI Model: IBM Granite 3.0-2B (Local)
AI Assistant
Citizen Dashboard
Feedback & Complaints
Analytics Dashboard
Services Guide
Policy Summarizer
Document Generator
Submit Feedback
Your Feedback
Share your experience with government services...
Submit Feedback
Result
Register Complaint
Category
Roads
Description
Describe your complaint in detail...
Contact Information
Register Complaint
Result
Power & Battery
Energy saver is on


Your AI-powered gateway to government services and civic engagement
AI Model: IBM Granite 3.0-2B (Local)
AI Assistant
Citizen Dashboard
Feedback & Complaints
Analytics Dashboard
Services Guide
Policy Summarizer
Document Generator
Citizen Sentiment & Complaint Analytics
Sentiment Analysis
Citizen Sentiment Distribution
Positive
Neutral
Negative
40%
40%
20%
Complaint Status
Complaint Status Overview
Count
1
0.8
0.6
0.4
0.2
0
In Progress
Resolved
Pending
Status
Refresh Dashboard



12. Known Issues

[List any known issues or bugs here]

13. Future Enhancements

[List planned future features and improvements here]