GEN AI WITH IBM CLOUD

Documentation format

# Introduction

* + **Project Title:** Sustainable Smart City Assistant Using IBM Granite LLM
  + **Team Members:**
  + **1.** Bodapati Devi Swamy Sai Pranai – Team Leader
  + 2. Swatesh – modules support

# Project Overview

* + **Purpose:** The Sustainable Smart City Assistant using IBM Granite LLM aims to promote environmentally responsible urban living by offering an AI-powered platform that simplifies access to sustainability insights and actions.To create a smart assistant that helps both citizens and city officials make informed, eco-friendly decisions by using natural language interaction and intelligent data processing.
  + **Features:**

**AI Chat Assistant:**

* + Users can ask sustainability-related questions in natural language.
  + Powered by IBM Granite LLM for accurate and context-aware responses.

**Eco-Tips Generator:**

* + Provides daily personalized tips for reducing environmental impact.
  + Suggestions based on user input or city data.

KPI Forecasting and Visualization

* + Predicts trends in energy, water, or waste usage.
  + Displays insights using interactive graphs and charts.
  +  Policy Summarization
  + Converts long government or city policy documents into easy-to-read summaries.
  + Helps users understand regulations and sustainability plans.
  +  Feedback Collection Module
  + Allows citizens to submit suggestions, complaints, or eco-ideas.
  + Data stored and optionally forwarded to authorities.
  +  Streamlit-Based Unified Interface
  + Combines all modules in a single web app.
  + Simple, responsive, and user-friendly UI.
  +  User Engagement & Personalization
  + Tracks past interactions to offer tailored advice.
  + Encourages continued participation with relevant content.

# Architecture

* + **Frontend:**  The frontend is built entirely using Streamlit, which serves interactive UI components such as:
  + Chat input/output interface . Buttons for generating eco-tips and submitting feedback
  + Graphs and charts for KPI visualization (using Matplotlib, Plotly, or Altair) It renders dynamic content directly in Python without needing separate frontend technologies like React.
  + **Backend:**  Handle API requests from the Streamlit frontend
  + Manage data storage, processing, and integration with city services
  + Ensure scalable, real-time communication between frontend and smart city systems

# Setup Instructions

* + **Prerequisites**: Python 3.8+  
    – Required to run Streamlit and related libraries.
  + Streamlit  
    – Web app framework for building the interactive UI.
  + Pandas *(optional but common)*  
    – For data manipulation and display.
  + Matplotlib / Plotly / Altair  
    – For creating visualizations like KPI charts and graphs.
  + Requests *(if connecting to APIs)*  
    – For calling external services (e.g., weather, pollution data).

1. **Installation:** Run the Streamlit AppFolder Structure

streamlit run app.py

* + **Client:** Streamlit interface
  + **Server**: Local Server

# Running the Application

* + Provide commands to start the frontend and backend servers locally.
    - **Frontend:** **streamlit run app.py**.
    - **Backend:** **streamlit run app.py**.

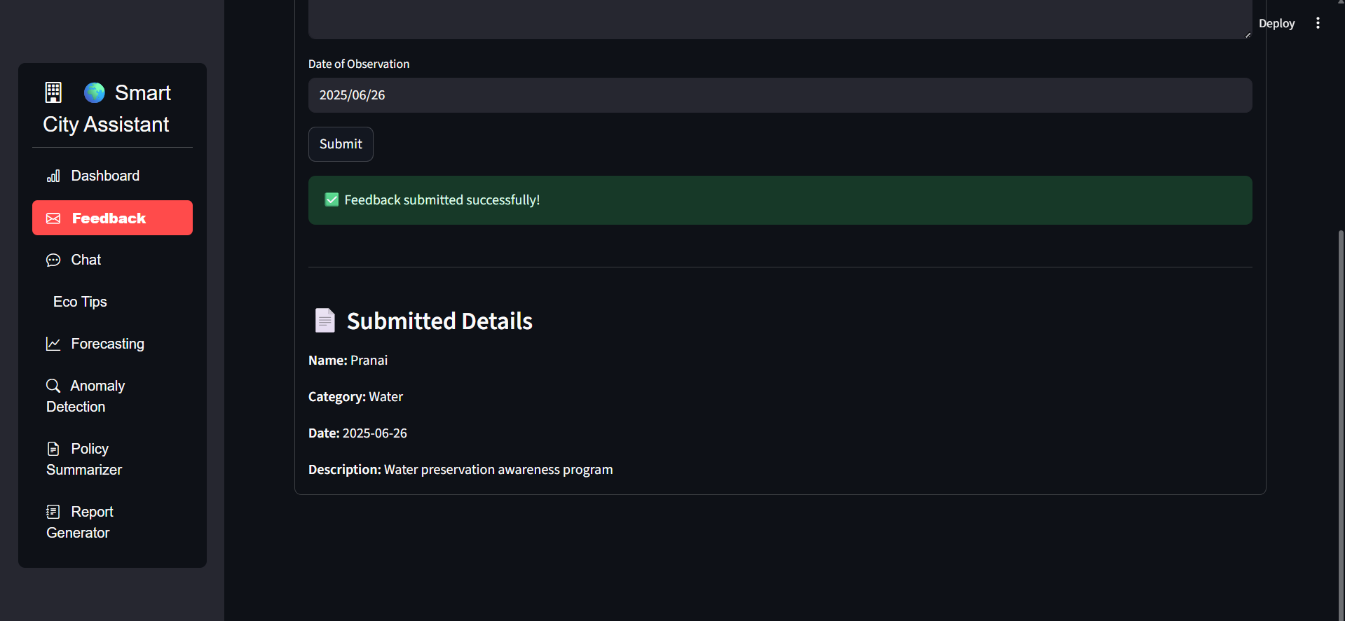
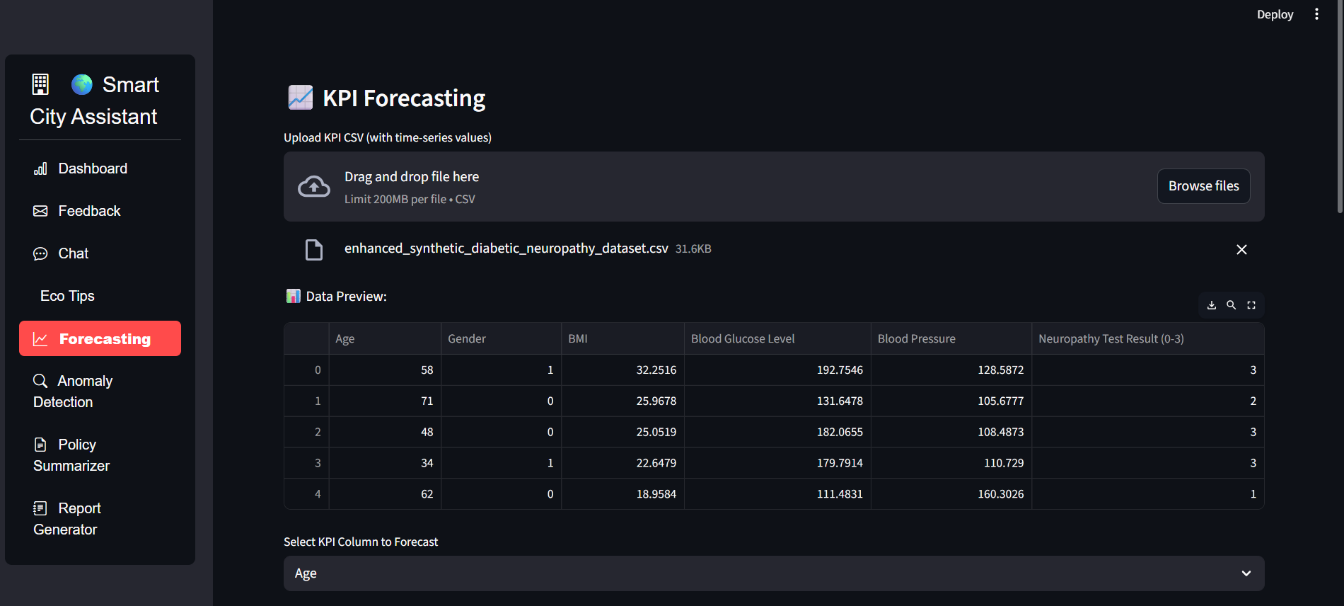
# API Documentation

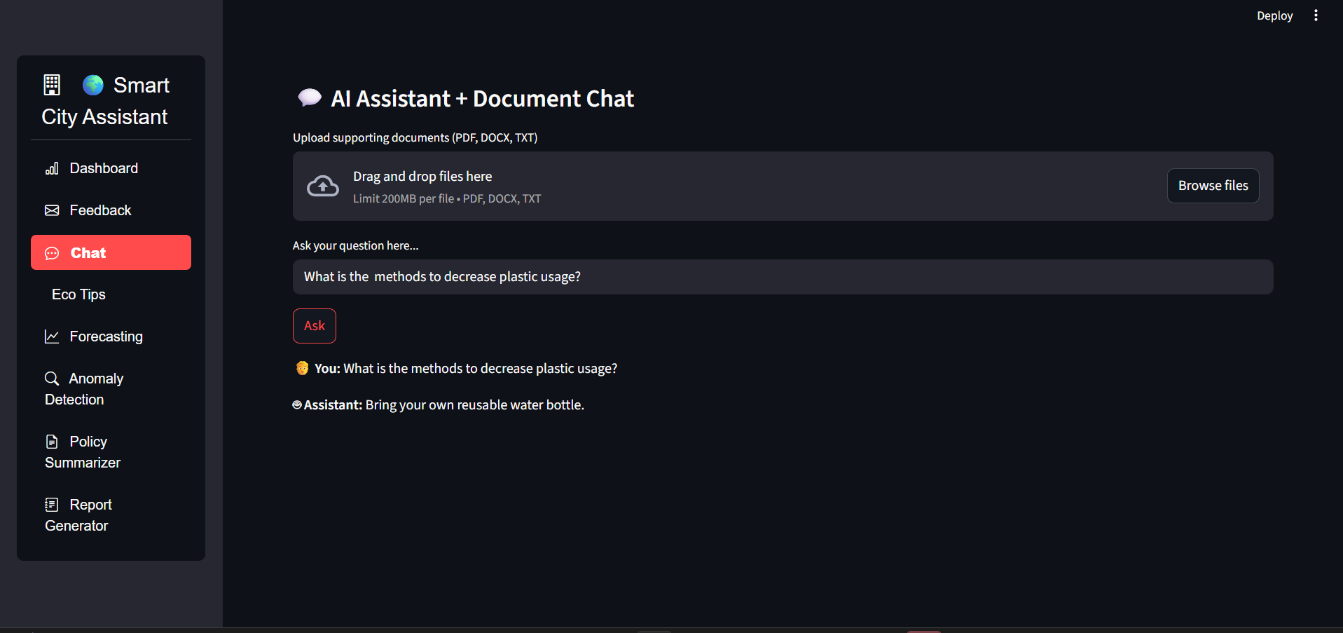
* + POST /generate\_eco\_tip, POST /chat, POST /submit\_feedback, GET /kpi\_data?area=city\_center&metric=energy\_usage
  + Include request methods, parameters, and example responses.

# Authentication

* + Once logged in, users are assigned a **role** (e.g., admin, user) stored in st.session\_state.
  + The app uses **Streamlit’s built-in login handling** *(if using Streamlit Community Cloud)*, or a **custom login system** using a username/password form.
  + Users log in through a **Streamlit form** with username and password.

# User Interface



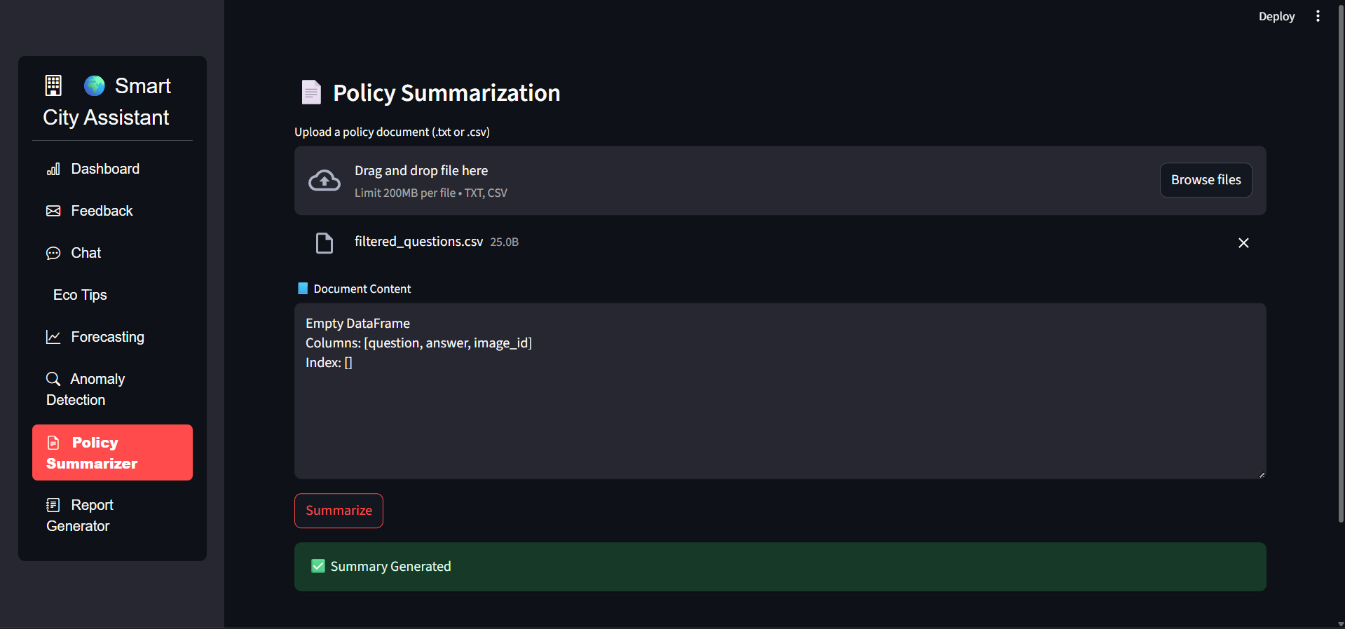


# Testing

* + Describe the testing strategy and tools used.

# Screenshots or Demo

Demo link: [Smartintern\_Project\_sdlc/Screen Recording 2025-06-26 130053 (1) (1) (2).mp4 at main · Pranai222/Smartintern\_Project\_sdlc](https://github.com/Pranai222/Smartintern_Project_sdlc/blob/main/Screen%20Recording%202025-06-26%20130053%20(1)%20(1)%20(2).mp4)



# Known Issues

**Session Reset on Refresh**

Streamlit clears session data when the page is reloaded.

* + *Solution:* Users can avoid refresh; persistent login is planned in future versions.

**Occasional API Timeout (if using external services like IBM Granite)**

* + Rare timeouts may happen due to internet issues or service load.

*Solution:* Handled with error messages and retry options.

**Mobile Layout**

* + While fully usable on mobile, some visualizations may need scrolling or resizing.

*Note:* Desktop or tablet gives the best experience.

# Future Enhancements

**Persistent Authentication System**  
Implement login with Google or Firebase to maintain user sessions across refreshes.

**Real-Time Data Integration**  
Connect to live data sources such as IoT sensors, weather APIs, or traffic feeds for up-to-date KPI tracking.

**Advanced AI Chatbot (IBM Granite Integration)**  
Enhance the chatbot with smarter, more context-aware responses using IBM Granite or similar large language models.

**Enhanced Analytics Dashboard**  
Add advanced filters, export options (CSV/PDF), and deeper insights using interactive visualizations.