Project Title: Citizen AI: Intelligent Citizen Engagement Platform

Team Members:

- P. Indhumalini
- J. Janani
- M.Janani
- S.Jaslin Mary

1. Introduction

The purpose of a Sustainable Smart City Assistant is to empower cities and their residents to thrive in a more eco-conscious and connected urban environment. By leveraging AI and real-time data, the assistant helps optimize essential resources like energy, water, and waste, while also guiding sustainable behaviors among citizens through personalized tips and services. For city officials, it serves as a decision-making partner—offering clear insights, forecasting tools, and summarizations of complex polic...

2. Project Overview

Features: - Conversational Interface (Natural language interaction: Allows citizens and officials to ask questions, get updates, and receive guidance in plain language) - Policy Summarization (Simplified policy understanding: Converts lengthy government documents into concise, actionable summaries) - Resource Forecasting (Predictive analytics: Estimates future energy, water, and waste usage using historical and real-time data) - Eco-Tip Generator (Personalized sustainability advice: Recommends daily actions to reduce environmental impact based on user behavior) Citizen Feedback Loop (Community engagement: Collects and analyzes public input to inform city planning and service improvements) - KPI Forecasting (Strategic planning support: Projects key performance indicators to help officials track progress and plan ahead) - Anomaly Detection (Early warning system: Identifies unusual patterns in sensor or usage data to flag potential issues) Multimodal Input Support (Flexible data handling: Accepts text, PDFs, and CSVs for document analysis and forecasting) - Streamlit or Gradio UI (User-friendly interface: Provides an intuitive dashboard for both citizens and city officials to interact with the assistant)

3. Architecture

Frontend (Streamlit): Interactive web UI with dashboards, file uploads, chat, feedback, and reports. Backend (FastAPI): API endpoints for document processing, chat, eco tips, reports, and embeddings. LLM Integration (IBM Watsonx Granite): Natural language understanding and generation. Vector Search (Pinecone): Embeddings stored with semantic search capability. ML Modules: Forecasting and anomaly detection using Scikit-learn. 4. Setup Instructions - Python 3.9+ - pip and virtual environment tools - API keys for IBM Watsonx and Pinecone - Internet access

5. Folder Structure - app/: Backend logic - app/api/: API routes - ui/: Frontend components smart_dashboard.py: Entry script - granite_llm.py: LLM integration - document_embedder.py: Document embeddings - kpi_file_forecaster.py: Forecasting - anomaly_file_checker.py: Anomaly detection - report_generator.py: AI reports

6. Running the Application - Launch FastAPI server - Run Streamlit dashboard - Upload documents, interact with assistant, view outputs

7. API Documentation Includes /chat/ask, /upload-doc, /search-docs, /get-eco-tips, /submit-feedback
8. Authentication Supports JWT, API keys, OAuth2, and role-based access

9. User Interface Minimalist, accessible, with KPI visualizations, chat, eco tips, forecasting, PDF download

10. Testing Unit tests, API tests, manual tests, edge case handling

11. Screenshots

12. Known Issues

13. Future Enhancements