

Project Title: SUSTAINABLE SMART CITY ASSISTANT

Team Leader: S.Benasir Begum Team Members: S.Leelasri, A.Marial, S.Jemi Pricilla

1. Introduction Project Title: SUSTAINABLE SMART CITY ASSISTANT Team Leader: S.Benasir Begum Team Members: S.Leelasri, A.Marial, S.Jemi Pricilla

2. Project Overview Purpose: 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 policies to support strategic planning. Ultimately, this assistant bridges technology, governance, and community engagement to foster greener cities that are more efficient, inclusive, and resilient.

Features: - Conversational Interface: Natural language interaction for queries and guidance. - Policy Summarization: Simplified policy understanding via concise summaries. - Resource Forecasting: Predictive analytics for energy, water, and waste usage. - Eco-Tip Generator: Personalized sustainability advice. - Citizen Feedback Loop: Community engagement through feedback collection. - KPI Forecasting: Strategic planning support. - Anomaly Detection: Early warning system for potential issues. - Multimodal Input Support: Text, PDF, CSV data handling. - Streamlit/Gradio UI: User-friendly dashboard for interaction.

3. Architecture Frontend (Streamlit): Interactive web UI with dashboards, file uploads, chat, feedback, and reports. Modular pages with sidebar navigation. Backend (FastAPI): REST APIs for document processing, chat, eco-tips, and vector embedding. Optimized for performance with Swagger support. LLM Integration (IBM Watsonx Granite): Natural language understanding for summarization, tips, and reports. Vector Search (Pinecone): Semantic search for uploaded documents. ML Modules: Forecasting and anomaly detection using Scikit-learn, pandas, and matplotlib.

4. Setup Instructions Prerequisites: - Python 3.9+ - pip, virtual environments - IBM Watsonx & Pinecone API keys - Internet access

Frontend (Streamlit): Modular dashboards with real-time updates. Backend (FastAPI): Asynchronous endpoints, Swagger integration.

7. API Documentation Endpoints: - POST /chat/ask: AI-powered chat - POST /upload-doc: Document upload & embedding - GET /search-docs: Semantic search on policies - GET /get-eco-tips: Sustainability tips by topic - POST /submit-feedback: Store citizen feedback

8. Authentication Open environment for demo; future security: JWT, OAuth2, role-based access, session tracking.

9. User Interface Minimalist design: sidebar navigation, KPI visualizations, chat tabs, forecasting tools, PDF downloads.

10. Testing Unit, API, manual, and edge case testing via Postman, Swagger, and scripts.

11. Screenshots 12. Known Issues 13. Future Enhancements