Sustainable Smart City Assistant – Project Documentation

1. Introduction

**Project title:**

Team member: P.prsannadevi

Team member: M.Santhiya

Team member: B.Santhiya

Team member: R.Rakshana parveen

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

**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 or Gradio UI

**Key Point**: User-friendly interface

**Functionality:** Provides an intuitive dashboard for both citizens and city officials to interact with the assistant

3. Architecture

**Frontend (Streamlit):**

Built with Streamlit, offering an interactive web UI with dashboards, file uploads, chat interface, feedback forms, and report viewers. Navigation via sidebar (streamlit-option-menu). Modularized for scalability.

**Backend (FastAPI):**

FastAPI powers API endpoints for document processing, chat, eco-tip generation, report creation, and vector embedding. Optimized for async performance and Swagger integration

**LLM Integration (IBM Watsonx Granite):**

Granite LLMs handle summarization, sustainability tips, and reports.

**Vector Search (Pinecone):**

Policy documents are embedded with Sentence Transformers and stored in Pinecone. Semantic search via cosine similarity.

**ML Modules (Forecasting & Anomaly Detection):**

Built with Scikit-learn. Time-series data parsed, modeled, and visualized with pandas & matplotlib.

4. Setup Instructions

**Prerequisites:**

Python 3.9 or later

pip & virtual environment tools

API keys (IBM Watsonx & Pinecone)

Internet access

**Installation**:

1. Clone the repository

2. Install dependencies from requirements.txt

3. Create .env and configure credential

4. Run backend server (FastAPI)

5. Launch frontend (Streamlit)

6. Upload data & interact with module

5. Folder Structure

app/ – Backend logic

app/api/ – Modular API routes (chat, feedback, reports, vectorization)

ui/ – Frontend components

smart\_dashboard.py – Launch script (Streamlit dashboard)

granite\_llm.py – LLM communication (IBM Watsonx Granite)

document\_embedder.py – Document embeddings (Pinecone)

kpi\_file\_forecaster.py – Forecast trends

anomaly\_file\_checker.py – Detect unusual values

report\_generator.py – Generate sustainability reports

6. Running the Application

1. Launch FastAPI server

2. Run Streamlit dashboard

3. Navigate via sidebar

4. Upload docs/CSVs, chat with assistant, view reports/summaries/predictions

5. Real-time updates via backend APIs

7. API Documentation

POST /chat/ask – AI-generated response

POST /upload-doc – Embed documents in Pinecone

GET /search-docs – Retrieve similar policies

GET /get-eco-tips – Sustainability tips (energy, water, waste)

POST /submit-feedback – Store citizen feedback

8. Authentication

Supports secure deployments:

JWT/API key tokens

OAuth2 with IBM Cloud

Role-based access (admin, citizen, researcher)

Planned: user sessions & history tracking

9. User Interface

Sidebar navigation

KPI visualizations & summary card

**Tabs:** chat, eco tips, forecasting

Real-time forms

PDF report download

Minimalist & accessible design

10. Testing

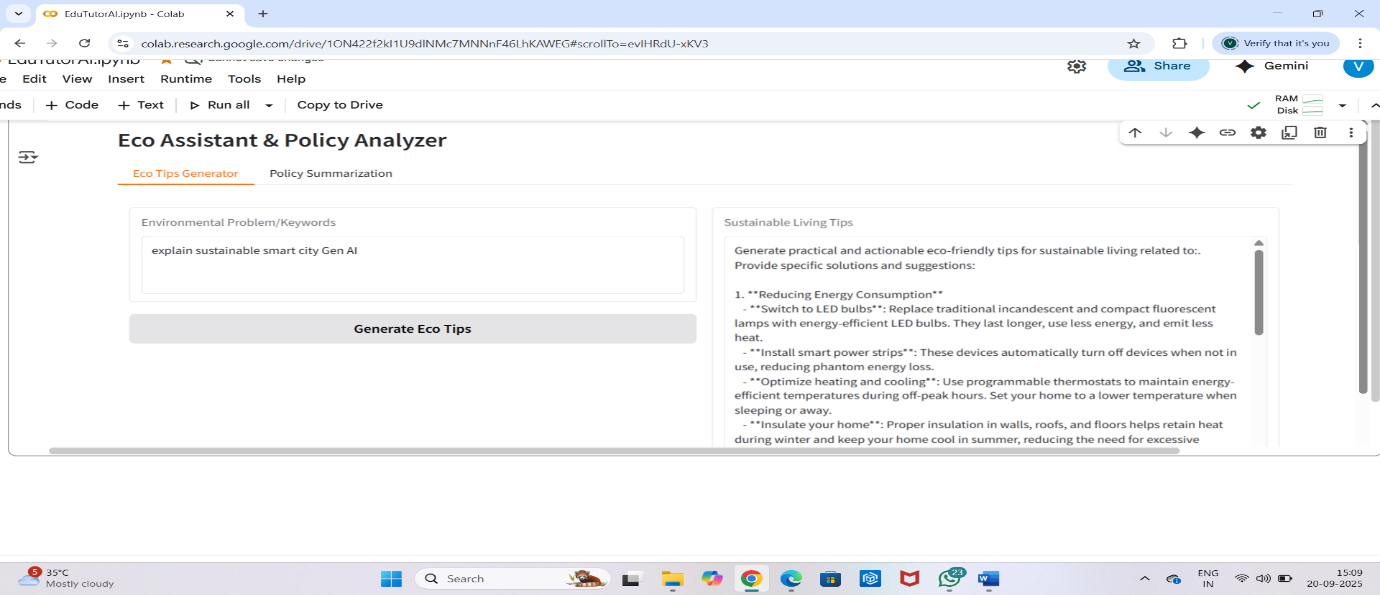
**Unit Testing:** Prompt functions & utilities

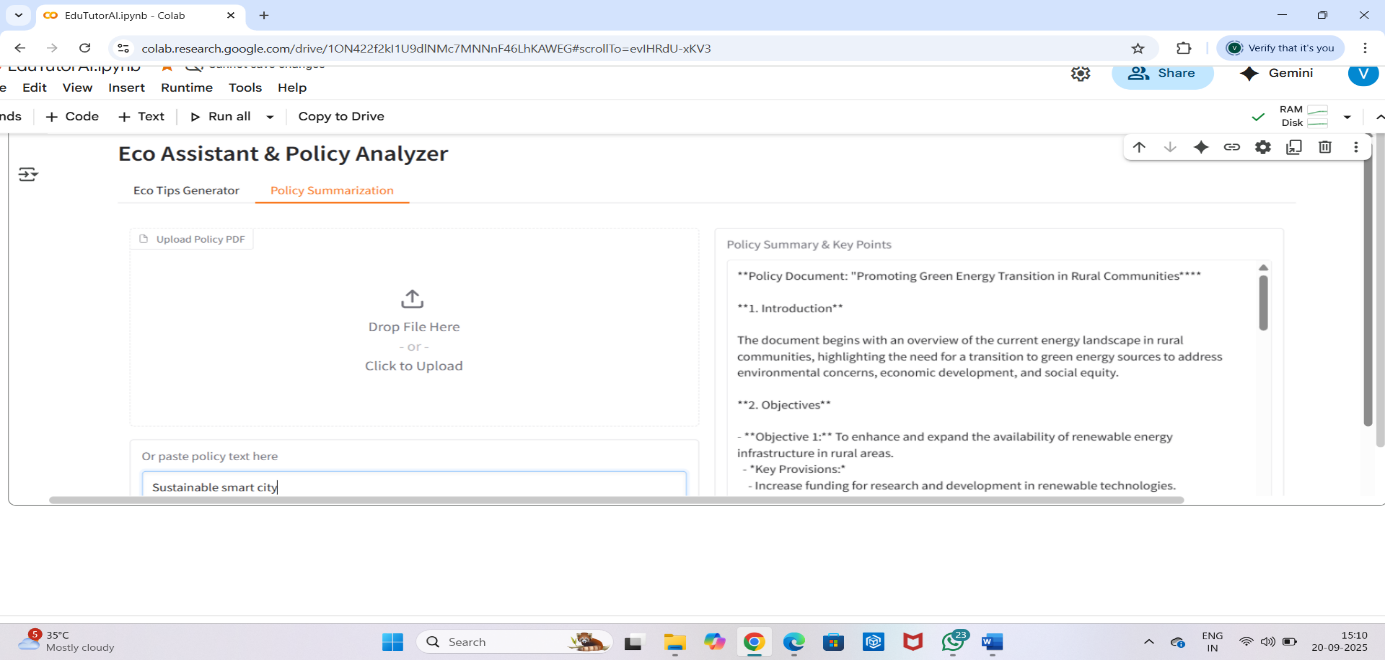
**API Testing:** Swagger, Postman, test scripts

**Manual Testing:** File uploads, chat, output checks

**Edge Cases:** Large files, malformed inputs, invalid keys

11.Screenshots:





12. Known Issues

**1. Data Quality Challenges** – Inconsistent or incomplete real-time data from city sensors may affect forecasting accuracy.

**2. High Dependency on Internet Connectivity** – Continuous cloud-based processing requires stable internet, which may not be available in all areas.

**3. Integration Issues** – Difficulty in integrating with legacy government systems and third-party data sources.

**4. Scalability Constraints** – Current prototype may not handle very large datasets or city-wide deployments efficiently.

13. Future Enhancements

**1. Multilingual Support** – Add regional language support to make the system accessible for all citizens.

**2. IoT Device Integration** – Connect with smart meters, traffic sensors, and weather stations for richer real-time data.

**3.Mobile Application Development**– Launch Android/iOS apps for easier citizen engagement.