**Sustainable Smart City**

## Project Documentation

## 1.Introduction

# Sustainable smart city (Team ):

## Team ID : NM2025TMID02729

**Team Leader : Suresh G** (4F6BB45BA36C3A0BFD54D0B494AFCF42)

## Team member : Surya.K (BC4B45295CF05F092684471D805E17FA)

## Team member : Akathiyan A (5068B42ABC41136A397420A4F122091F)

**Team member : AGILAN.R** (24D3EF89919D6D7AF7A4FCB0215BC4C3)

## Team member : Sankaran.G (3DEDF8FECD28D838E98496ACDFFDFE4C)

## 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 Decisionmaking 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 (Stream lit):

The frontend is built with Stream lit, 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 stream lit-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**

### Prerequisites:

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

### Installation Process:

o Clone the repository o Install dependencies from requirements.txt o Create a .ENV file and configure credentials o Run the backend server using Fast API o Launch the frontend via Stream lit o Upload data and interact with the modules

## 5. Folder Structure

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 Stream lit pages, card layouts, and form UIs.

smart\_dashboard.py – Entry script for launching the main Stream lit dashboard.

granite\_llm.py – Handles all communication with IBM Watsonx Granite model including summarization and chat.

document\_embedder.py – Converts documents to embedding 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 Stream lit 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.

### Frontend (Stream lit):

The frontend is built with Stream lit, 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 stream lit-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.

## 7. API Documentation

Backend APIs available include:

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**

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

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.8. Authentication

## 9. User Interface

The interface is minimalist and functional, focusing on accessibility for nontechnical 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

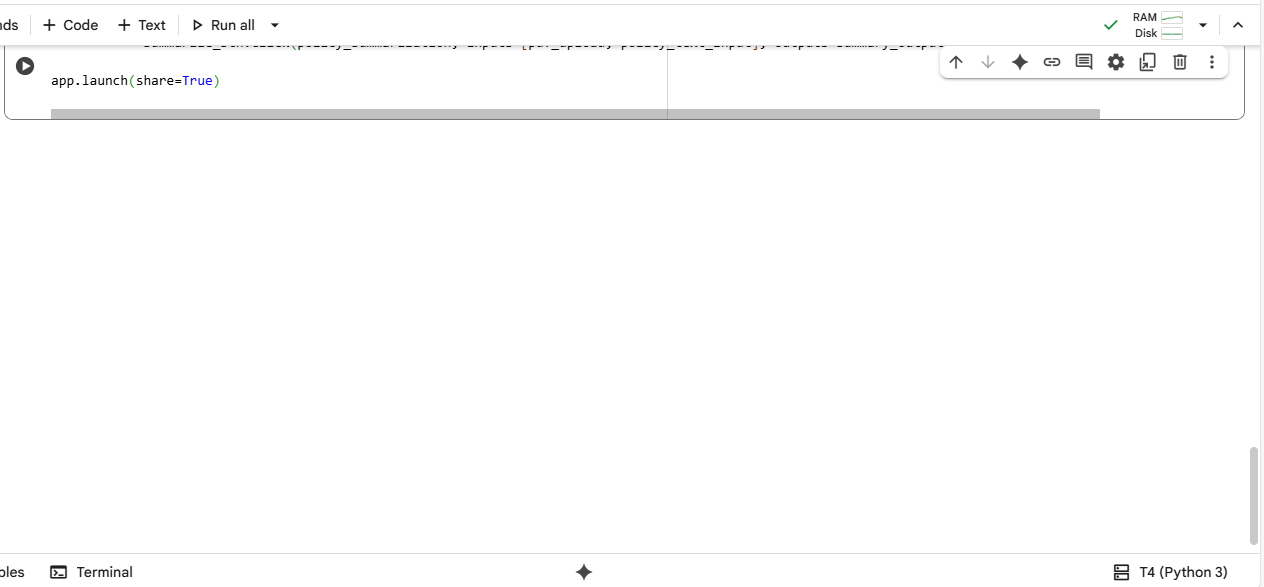
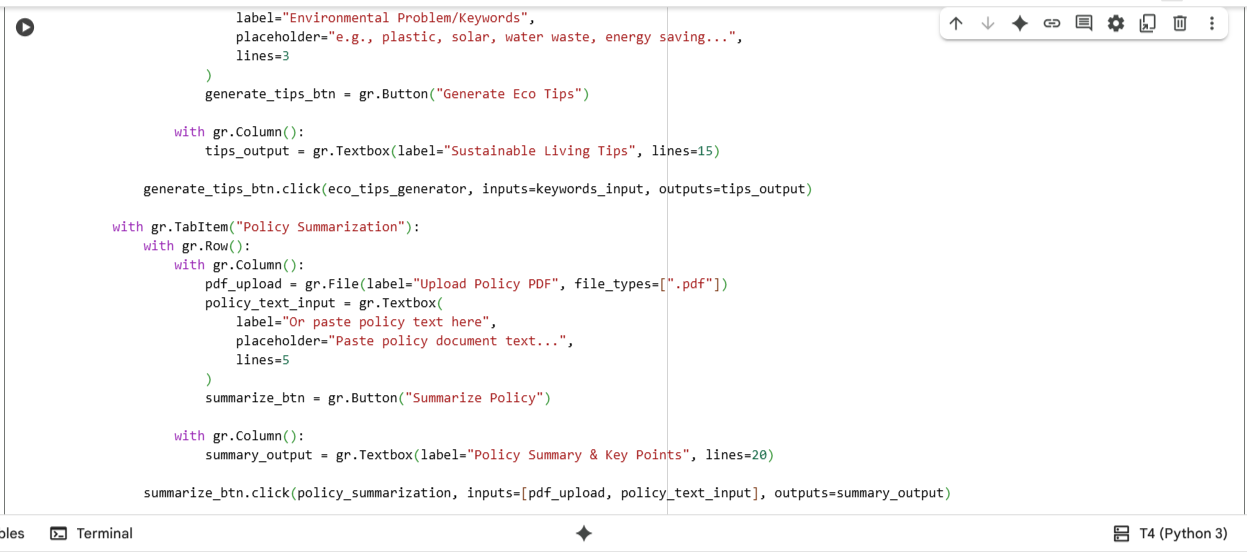
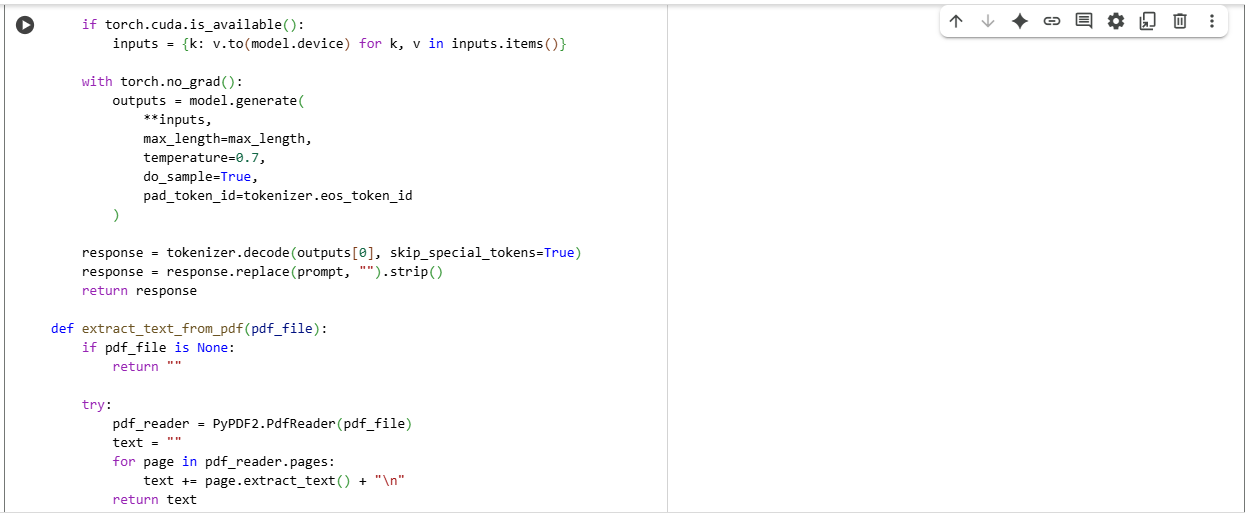
API Testing: Via Swagger UI, Postman, and test scripts

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**

****

1. **Known Issues**

**1. Data Privacy & Security**

* **Issue:** Collection of vast amounts of personal and urban data (e.g., through IoT devices, surveillance).
* **Risk:** Hacking, surveillance misuse, lack of consent.
* **Challenge:** Balancing efficiency with civil liberties.

**2. Digital Divide**

* **Issue:** Unequal access to smart city technologies (e.g., internet, smart devices).
* **Impact:** Marginalizes lower-income or older populations.
* **Challenge:** Ensuring digital inclusivity across socioeconomic groups.

**3. High Implementation & Maintenance Costs**

* **Issue:** Upfront capital and ongoing costs for smart infrastructure, sensors, and tech.
* **Risk:** Overdependence on tech without clear ROI.
* **Challenge:** Financial sustainability and securing long-term funding.

**4. Interoperability & Legacy Systems**

* **Issue:** Integrating new smart technologies with outdated city infrastructure or siloed systems.
* **Challenge:** Compatibility, data exchange, and vendor lock-in.

**5. Environmental Trade-Offs**

* **Issue:** Electronic waste, high energy consumption of data centers and IoT devices.
* **Paradox:** Some smart systems harm the environment in other ways.
* **Challenge:** Ensuring tech itself is sustainably produced and maintained.

**6. Governance and Policy Gaps**

* **Issue:** Lack of clear regulatory frameworks for smart technologies and AI in cities.
* **Challenge:** Enacting adaptive and forward-looking policies that balance innovation and oversight.

**7. Citizen Engagement and Trust**

* **Issue:** Top-down planning without citizen involvement leads to mistrust and low adoption.
* **Challenge:** Creating participatory platforms for citizens to shape smart city strategies.

1. **Future enhancement**

**1. AI-Driven Urban Management**

* **What's coming:** Advanced AI for dynamic traffic control, waste management, and energy optimization.
* **Impact:** Real-time decision-making, reduced congestion and pollution.
* **Example:** AI models predicting pedestrian traffic to optimize walkways and crossings.

**2. Green Infrastructure & Net-Zero Goals**

* **Focus:** Integrating smart buildings, renewable energy, and carbon-neutral planning.
* **Enhancements:**
  + Smart grids with real-time energy monitoring.
  + Solar-integrated public infrastructure (e.g., benches, bus stops).
  + Vertical farming and green roofs for food and climate resilience.

**3. Digital Twin Technology**

* **What's next:** Full-scale city simulations that mirror real-time operations.
* **Uses:**
  + Predict infrastructure failures.
  + Model disaster responses.
  + Test urban planning decisions virtually before real-world implementation.

**4. Hyperconnected Transportation Systems**

* **Improvements:**
  + Autonomous electric vehicles (AVs).
  + AI-managed multimodal transport (bikes, buses, trains, scooters).
  + Smart traffic signals that prioritize emergency and public transport.
* **Goal:** Reduce emissions and travel times.

**5. Citizen-Centric Smart Governance**

* **New tools:**
  + Participatory platforms using blockchain for transparent voting, budgeting.
  + Real-time feedback systems through mobile apps.
* **Goal:** Empower citizens in decision-making and city planning.

**6. Edge & Quantum Computing for City Operations**

* **Edge Computing:** Localized data processing to reduce latency in smart services.
* **Quantum Computing:** Early-stage use for optimizing city logistics and complex simulations.

**7. Advanced Waste & Water Management**

* **Upgrades:**
  + IoT-enabled bins with dynamic pickup scheduling.
  + Smart leak detection and water reuse systems.
* **Goal:** Reduce resource waste and improve efficiency.

**8. 5G & Beyond Connectivity**

* **Coming soon:** 6G and satellite internet integration for seamless connectivity.
* **Impact:** Faster, more reliable services enabling autonomous systems, augmented reality (AR), and telemedicine.

**9. Urban Resilience & Climate Adaptation**

* **Enhancements:**
  + Predictive models for extreme weather.
  + Flood sensors and climate-responsive urban planning.
  + Heat-resilient building materials and cooling corridors.

**10. Inclusive Design & Accessibility**

* **Future Direction:**
  + Universal design standards embedded in tech and architecture.
  + AI-based accessibility tools (e.g., voice navigation for the visually impaired).
  + Multilingual digital services for diverse populations.

1. **Known Issues**
2. **Future enhancement**