Sustainable Smart City Assistant Using IBM Granite LLM

TEAM ID : NM2025TMIDO3747

TEAM LEADER : ARUN G

TEAM MEMBER : ANANDAN K

TEAM MEMBER : DARUN VIGNESH M

TEAM MEMBER : LOKESHWARAN M

1. Executive Summary :

This project builds a **Sustainable Smart City Assistant (SSCA)** — a multimodal, privacy-preserving conversational and decision-support system powered by IBM Granite LLM. SSCA helps city officials, planners, utility operators, and residents make environmentally and socially sustainable decisions by combining real-time IoT telemetry, open urban data, GIS layers, policy documents, simulation outputs, and domain knowledge. The system provides natural-language Q&A, scenario simulations, automated reporting, energy & emissions suggestions, incident alerts, and citizen engagement tools.

The assistant is designed to be modular, auditable, and deployable on hybrid cloud or on-prem environments to comply with city governance and privacy requirements.

2. Project Objectives :

- Provide a single conversational interface (web + mobile + chat APIs) for stakeholders to query city sustainability metrics and receive actionable guidance.
- Integrate heterogeneous data: energy grids, traffic, waste management, public transit, air quality, weather, building energy use, and sensors.
- Use IBM Granite LLM for contextualized reasoning, summarization, policy-aware recommendations, and natural-language reporting.
- Offer scenario planning: "if we change traffic flows / adjust tariffs / add green roofs" and show predicted impacts on emissions, cost, and equity.
- Ensure privacy, fairness and explainability: data minimization, differential privacy

where required, provenance for model outputs.

- Deliver a production-ready pilot in 9–12 months, with evaluation on technical performance and stakeholder satisfaction.

3. Scope & Target Users :

- **Scope (pilot) :** one district of a mid-sized city (e.g., 50k–300k population) including utilities, traffic, public buildings, and citizen engagement channels.
- **Primary users :** city planners, sustainability officers, utility operators, emergency services, community organizations, and residents.

4. Key Capabilities :

- **Natural-language queries & conversational assistance**: Ask questions like "How did CO_2 emissions change last month in district A?" or "Suggest low-cost measures to reduce peak electricity demand.".
- **Real-time monitoring & alerts**: Ingest sensor feeds and notify on threshold breaches (air quality, water leaks, grid overload).
- **Scenario simulation & forecasting**: Create "what-if" scenarios using rule-based or learned models to estimate emissions, cost, and service-level impacts.
- **Automated reporting & compliance**: Generate auditable sustainability reports (PDF/CSV) and executive summaries aligned to local/regional standards.
- **Action recommendations**: Prioritized, budget-aware measures (e.g., building

retrofits, demand-response, transit incentives) with estimated impact.

- **Citizen engagement**: Chatbot for residents to report issues, receive personalized efficiency tips, and participate in local surveys.
- **Explainability & provenance**: For every recommendation, provide the data sources used, confidence scores, and an explanation in plain language.

5. System Architecture (high-level) :

- **Data ingestion layer :**
  - Stream connectors for IoT platforms (MQTT, Kafka), public open data APIs, GIS tiles, and batch uploads (CSV).
  - ETL & metadata catalog for schema normalization and lineage tracking.
- **Data storage & processing :**
  - Time-series DB for telemetry (e.g., InfluxDB / Timescale), relational DB for entities, data lake for raw archives.

    - Spatial DB (PostGIS) for GIS queries.
    - Feature store for ML features and scenario models.
- **Analytics & simulation engines :**
    - Rule-based calculators (emissions factors, tariff models).
    - ML forecasting models (demand, traffic, pollution) for scenario predictions.
- **LLM orchestration & contextualization layer :**
    - IBM Granite LLM used for: natural language understanding, multi-step reasoning, summarization, prompt-based retrieval-augmented generation (RAG), and generation of reports.
    - A **RAG** pipeline: retrieval from a vector DB (e.g., Milvus/FAISS) of policy, regulations, past reports, and local data summaries; Granite used to synthesize answers with provenance.

- Prompt templates, safety filters, and instruction tuning or lightweight adapters if necessary.

- **Application & APIs :**
  - REST / GraphQL APIs for integrations.
  - Event bus for alerts and action triggers.
- **UI / UX :**
  - Web dashboard (planner & operator modes) with maps, charts, and conversational panel.
  - Resident-facing mobile/web chat with location-aware functionality.
- **Security, privacy & governance :**
  - RBAC for user roles, encrypted storage, data minimization, and audit logs.
  - Model logging, prompt and response archiving with access controls for auditing.

6. IBM Granite LLM: Integration Strategy :

Note: This section assumes Granite is available via secure API or private-hosting. Implementation details should be finalized against IBM's latest developer docs during implementation.

- **Use-cases for Granite LLM :**
  - Natural language understanding (intent/entity extraction)
  - Answer synthesis using RAG (combine local data + knowledge base)
  - Summarization of long documents (contracts, policies)
  - Generating human-readable action plans and citizen messages
  - Translating technical model outputs into plain-language explanations
- **Workflow :**

- Preprocessing: normalize input (user query + user role + location + current context snapshot)
- Retrieval: query vector DB for relevant documents, sensor summaries, and policies
- Prompting: construct a structured prompt template with retrieved context (max tokens budget), explicit instruction to cite sources and provide confidence
- Post-processing: validate actionable suggestions through safety rules, compute numerical forecasts from simulation engines if required, attach provenance and confidence

➢ **Safety & Guardrails :**

- Apply instruction-level constraints, deny or escalate policy-sensitive requests (e.g., infrastructure sabotage, targeted surveillance recommendations).

- Use content filters and post-hoc validators for suggested interventions (legal/regulatory compliance checks).

- **Fine-tuning vs. Prompting :**
  - Start with prompt engineering + RAG for speed and safety.
  - Collect high-quality interaction logs (with consent) and consider instruction-tuning or local adapters on a small domain-specific corpus for improved performance.
- **Latency & Cost :**
  - Use caching for frequent queries and summarization of time-window snapshots.
  - For time-critical alerts, use deterministic rules or lightweight local models; reserve Granite LLM for explanation and complex reasoning.

7. Data Sources & Types :

- **Real-time telemetry :** smart meters (energy/water), traffic sensors, air-quality monitors, waste collection telemetry.
- **Public/open data :** census, land use, public transport schedules, weather.
- **City systems :** asset registries, building permits, zoning maps, utility network topologies.
- **Historical reports & policy docs :** sustainability plans, local regulations, procurement rules.
- **Citizen inputs :** incident reports, survey responses, consumption preferences.

    Privacy considerations : minimize personally identifiable information (PII) in the LLM context; when resident-specific tips are generated, use local templates and avoid exposing raw personal data to third-party model hosting.

8. Evaluation & Metrics Technical metrics:

- LLM answer quality (human-rated relevance & helpfulness)
- Retrieval relevance (R@k, MRR)
- Forecast accuracy (RMSE/MAPE for demand and pollution)
- System latency (median & tail latencies)

Operational & impact metrics :

- Reduction in peak electricity demand (kW) during pilot
- Reduced emissions (CO_2e) from implemented measures
- Response time to incidents
- Stakeholder satisfaction (surveys) and adoption rates

Safety metrics :

- Frequency of hallucinated claims (manual audit)

- Number of flagged policy non-compliant recommendations

9. Implementation Plan & Timeline (9 months — example) :

- **Month 0 (Planning & setup) :**
  - Stakeholder interviews, data access agreements, success metrics
  - environment provision
- **Month 1–2 (Data & infra) :**
  - Connect core data sources, build ETL, vector DB for documents
  - Deploy time-series DB and analytics pipelines
- **Month 3–4 (Core features & Granite integration) :**
  - Implement RAG pipeline, prompt templates, Granite LLM integration
  - Build conversational backend and basic UI prototypes
- **Month 5–6 (Simulations & recommendations) :**

- Integrate forecasting models and scenario engines
- Implement recommendation ranking and explainability layer
  - **Month 7 (Pilot deployment) :**
- Deploy to pilot district, on-board a small set of users
- Run monitoring, collect logs and feedback
  - **Month 8 (Evaluation & iteration) :**
- Measure KPIs, refine prompts, add features requested by users
  - **Month 9 (Wrap-up & handover)**
- Final report, documentation, knowledge transfer, and roadmap for scale

10. Sample code & Output :

- **Install libraries :**
  - transformers, torch, gradio, PyPDF2.
- **Import modules :**

- Load Gradio, Torch, Hugging Face Transformers, PDF reader**.**

- **Load model & tokenizer :**
  - Choose model → ibm-granite/granite-3.2-2b-instruct**.**
  - Load tokenizer and model with GPU/CPU support.
  - Set padding token if missing.

- **Define response function :**
  - Tokenize user input.
  - Send input to model (generate).
  - Decode tokens into text output**.**

- **Build Gradio interface :**
  - **Tab 1** : Eco Tips Generator

    Input : keywords.

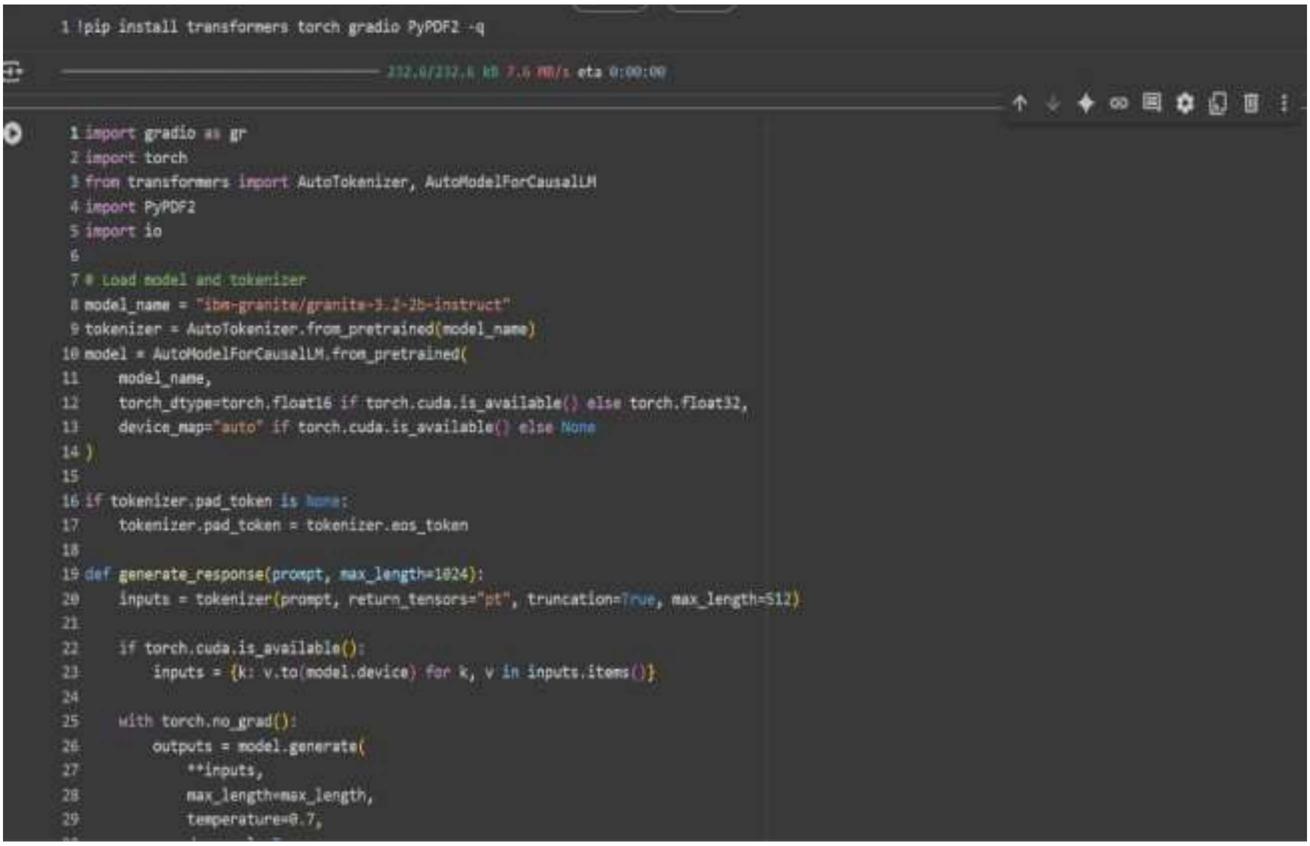
    Output : sustainability tips.

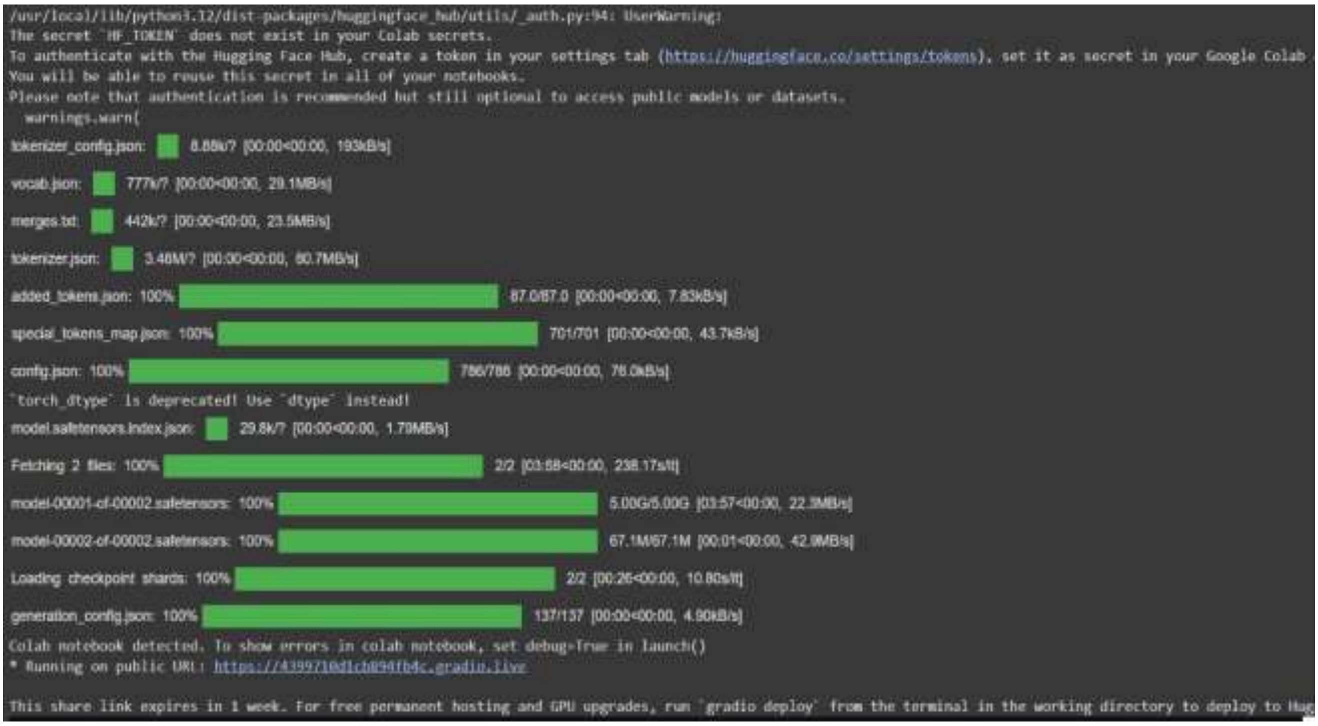
  - **Tab 2** : Policy Summarization

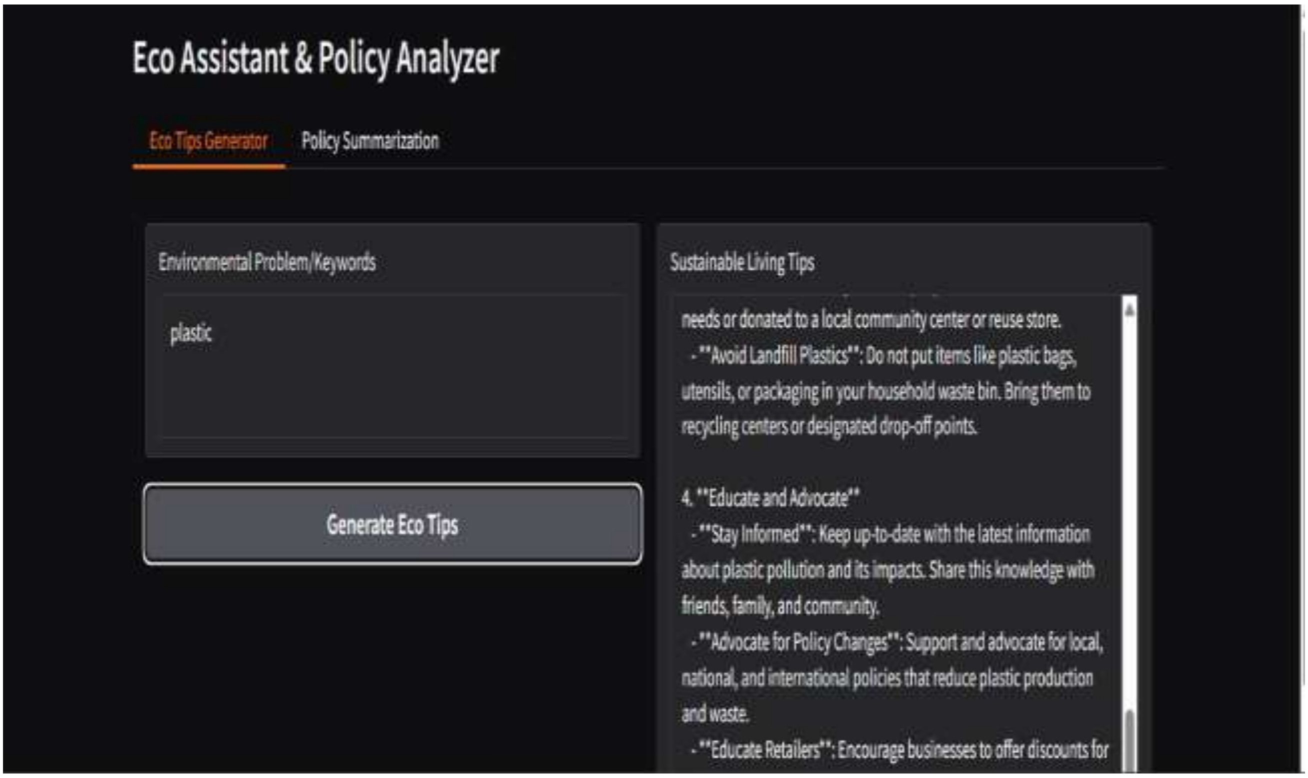
    Input : PDF file.

Output : short summary**.**

- **Launch app :**
  - Run Gradio demo → local/Colab share link is generated.




/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
warnings.warn(
tokenizer_config.json:
vocab.json:
merges.txt:
tokenizer.json:
added_tokens.json: 100%
special_tokens_map.json: 100%
config.json: 100%
`torch_dtype` is deprecated! Use `dtype` instead!
model.safetensors.index.json:
Fetching 2 files: 100%
model-00001-of-00002.safetensors: 100%
model-00002-of-00002.safetensors: 100%
Loading checkpoint shards: 100%
generation_config.json: 100%
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL:
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hug


Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
plastic
Generate Eco Tips
Sustainable Living Tips
needs or donated to a local community center or reuse store.
- **Avoid Landfill Plastics**: Do not put items like plastic bags, utensils, or packaging in your household waste bin. Bring them to recycling centers or designated drop-off points.
4. **Educate and Advocate**
- **Stay Informed**: Keep up-to-date with the latest information about plastic pollution and its impacts. Share this knowledge with friends, family, and community.
- **Advocate for Policy Changes**: Support and advocate for local, national, and international policies that reduce plastic production and waste.
- **Educate Retailers**: Encourage businesses to offer discounts for

Conclusion :

The Sustainable Smart City Assistant powered by IBM Granite LLM provides a practical AI-driven framework for urban sustainability. By covering key topics such as data integration, privacy, evaluation, risks, and stakeholder engagement, it ensures readiness for real-world deployment. The Colab + Gradio prototype validates its feasibility, enabling features like eco-tips, policy analysis, and scenario forecasting. This assistant represents a scalable, explainable, and citizen-focused tool for building smarter, greener cities.

PROJECT TEAM & ROLES :

ARUN – Code Developer

ANANDAN – Documentation Lead

DARUN VIGNESH – Voice Narration

LOKESHWARAN – Demo Production

END