***🌍***Sustainable Smart City Assistant

Team Name: **EcoNova**

**Team ID: LTVIP2025TMID31087**

**Team Member:**

## NAME : BOLLAVARAM ANIL PREETHAM Register Number: ***SBAP0031072*** Hall Ticket No.: 22AT1A3113 College: G. Pullaiah College of Engineering and Technology

**📑 Table of Contents**

|  |  |  |
| --- | --- | --- |
| S. No. | Section Title |  |
| 1. | Introduction |  |
| 2. | Ideation Phase |  |
| 3. | Requirement Analysis |  |
| 4. | Project Design |  |
| 5. | Project Planning & Scheduling |  |
| 6. | Functional and Performance Testing |  |
| 7. | Results |  |
| 8. | Advantages & Disadvantages |  |
| 9. | Conclusion |  |
| 10. | Future Scope |  |
| 11. | Appendix |  |
|  | Source Code, Dataset Link, GitHub |  |

**1. INTRODUCTION**

**1.2 Purpose**

The main purpose of this project is to support eco-friendly choices and smarter city planning through an intelligent assistant.

Key objectives include:

* Educating users on sustainability topics
* Providing actionable, personalized recommendations
* Simplifying complex urban policies
* Automating feedback collection and resource analysis
* Promoting daily eco-friendly habits

This tool bridges the gap between AI and civic engagement, encouraging users to make more informed, sustainable decisions in daily life.

**2. IDEATION PHASE**

**2.1 Problem Statement**

Urban areas are under increasing pressure from challenges like poor waste management, inefficient energy use, and low public engagement in sustainable practices. Despite the availability of policies and initiatives, most citizens are either unaware of them or find them too complex to act upon. A lack of accessible digital tools that guide and assist in sustainable behaviour further widens the gap.

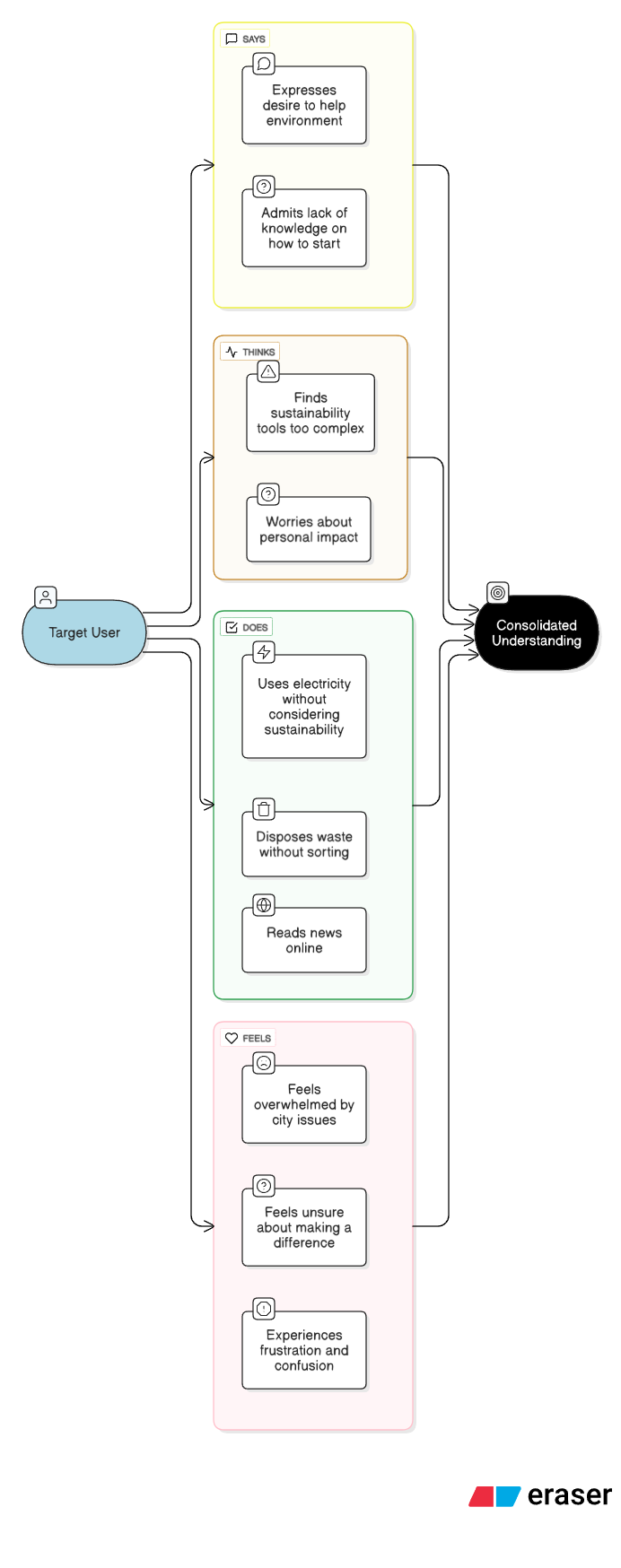
**2.2 Empathy Map Canvas**

To better understand the needs of target users, an empathy map was created based on the following key aspects:

**Says**: "I want to help the environment but don't know how."

* **Thinks**: "Sustainability tools are usually too complex."
* **Does**: Uses electricity, throws waste without sorting, reads news online
* **Feels**: Overwhelmed by city issues, unsure about making a difference

This helped identify opportunities to simplify sustainability through digital assistance.



**2.3 Brainstorming**

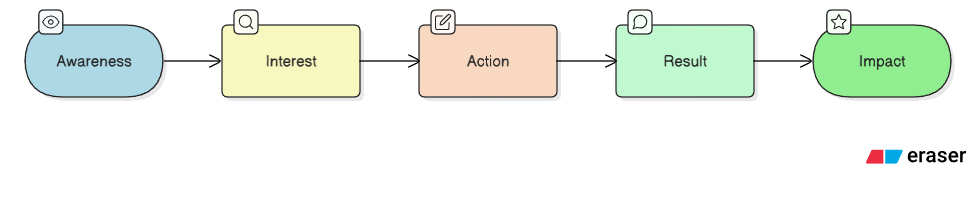
Initial brainstorming sessions focused on combining artificial intelligence with urban sustainability. Ideas were generated around making the system:

* **Conversational** (natural language-based)
* **Modular** (separate tools for waste, energy, policy, etc.)
* **Actionable** (providing tips, summaries, and forecasts)
* **Accessible** (no installation, simple UI)

The final idea evolved into a smart assistant using **Gradio for interface** and **IBM Granite for AI-powered responses**, deployed conveniently via Google Colab.

**3.1 Customer Journey Map**

To ensure user-centric design, a simple journey map was created:



* **Awareness**: User learns about the assistant via demo or campaign
* **Interest**: Visits the tool to explore features
* **Action**: Inputs data or questions (e.g., waste item, energy habits)
* **Result**: Receives personalized AI-driven response
* **Impact**: Learns, improves habits, or contributes civic feedback

This helped shape the feature layout and interface flow.

**3.2 Solution Requirements**

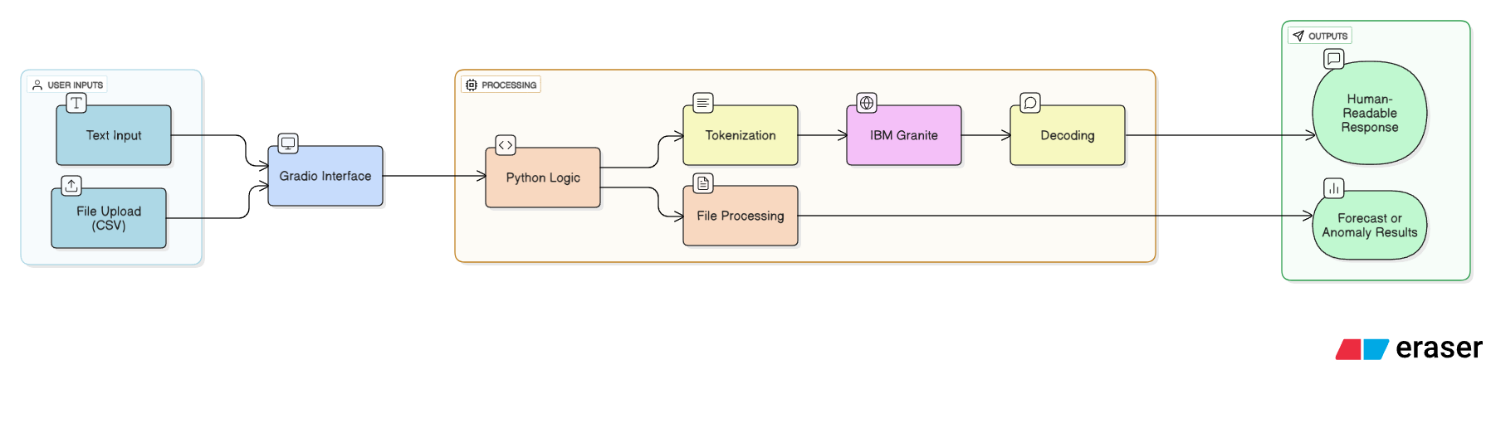
The tool was designed to offer the following core functionalities:

* Waste disposal and recycling guidance
* Energy usage analysis and tips
* Sustainability Q&A chatbot
* Policy text summarization
* Forecasting of utility consumption via CSV
* Anomaly detection in uploaded data
* Green challenge generator
* Citizen feedback collection and download

Each feature is designed to be modular, easy to use, and relevant to sustainable city living.

**3.3 Data Flow Diagram (DFD)**

The basic system flow is as follows:



**User Input** → **Gradio Interface** → **AI Model / Python Logic** → **Response Output**

* File Uploads (CSV) → Processed by Python → Forecast or Anomaly Results
* Text Inputs → Tokenized → Sent to IBM Granite → Decoded into human-readable response

**3.4 Technology Stack**

|  |  |
| --- | --- |
| Layer | Tools / Libraries Used |
| Language | Python |
| Frontend | Gradio (UI components, Tabs, Inputs, Buttons) |
| Backend | Custom Python Functions + IBM Granite 3.3-2B Instruct |
| AI Model | IBM’s open Granite family (via HuggingFace Transformers) |
| Data | CSV files, User text inputs |
| Hosting | Google Colab + Pyngrok for sharing public link |
| Storage | Feedback data saved as Excel using pandas + openpyxl |

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

This project directly addresses key sustainability pain points faced by urban residents:

|  |  |
| --- | --- |
| Problem | Solution |
| Waste disposal confusion | AI-powered waste sorting assistant |
| High energy consumption | Personalized energy-saving tips |
| Lack of eco-awareness | Daily green challenges and Q&A support |
| Complex policies | Summarization of documents in simple terms |
| Untracked utility trends | Forecasting and anomaly detection with CSVs |
| Low civic engagement | Easy citizen feedback collection and storage |

The assistant turns challenges into actionable insights using a conversational approach.

**4.2 Proposed Solution**

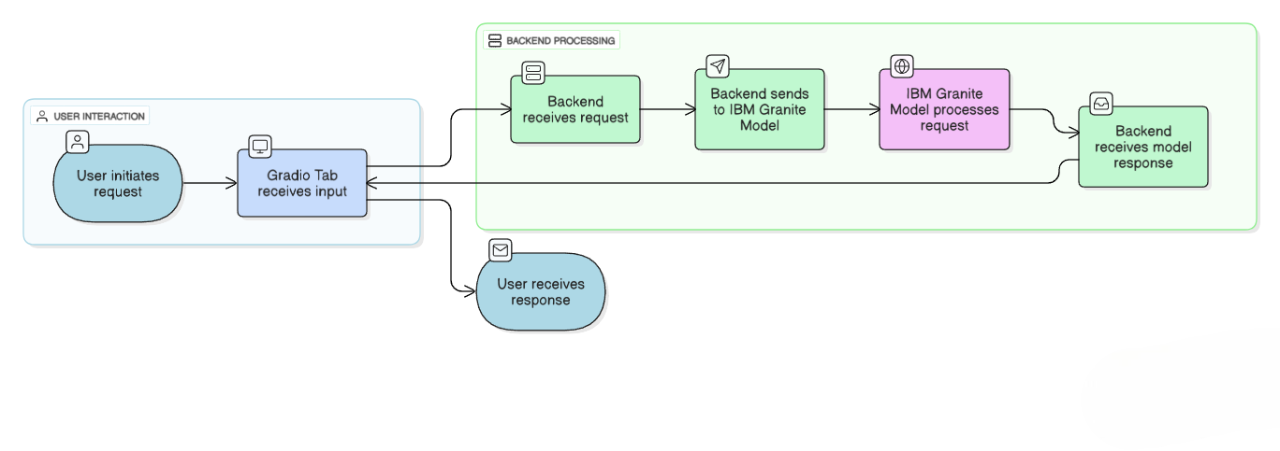
The solution is an **AI-driven, modular web assistant** that allows users to interact with sustainability features from a single, easy-to-use Gradio interface. Instead of creating multiple apps, all services are built as **individual tabs** with clean UI components and connected to a powerful language model (IBM Granite).

Core modules include:

* Waste Sorting
* Energy Advisor
* Policy Expert
* Feedback Collector
* Forecasting & Anomaly Detection
* Sustainability Chatbot
* Eco Tips & Green Challenge Generator

**4.3 Solution Architecture**

Here’s how the system works:



* **Frontend**: Gradio for layout and user interaction
* **Model Integration**: IBM Granite via Hugging Face Transformers
* **Logic Layer**: Python functions for processing, summarizing, forecasting, etc.
* **Output**: Displayed or downloadable results (e.g., Excel for feedback)

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning**

* The project followed a milestone-based Agile workflow, divided into structured sprints to ensure steady progress within the given timeline of **May 26 to June 25, 2025**. Being a solo project, planning and execution were done independently with careful task prioritization.
* **🔹 Sprint-wise Breakdown**

|  |  |  |
| --- | --- | --- |
| Sprint | Focus Area | Tasks Covered |
| Sprint 1 | Setup & Initialization | Finalizing problem statement, Colab environment setup, installing dependencies |
| Sprint 2 | Core Feature Development | Waste sorting assistant, energy advisor, policy summarization, eco challenges |
| Sprint 3 | Advanced Modules | Forecasting and anomaly detection via CSV files |
| Sprint 4 | Feedback Module & File Integration | Citizen feedback form, Excel export functionality |
| Sprint 5 | AI Chat Assistant + UI Enhancements | Sustainability Q&A chatbot, Gradio tabbed layout, enhanced CSS |
| Sprint 6 | Final Testing & Submission Preparation | Debugging, file handling, test cases, documentation, and demo video creation |

|  |
| --- |
| 🧑‍💻 Task Allocation |
| Bollavaram Anil Preetham (Solo Developer) – Responsible for end-to-end development, UI design, testing, documentation, and deployment. |

**🕒 Project Timeline**

|  |  |
| --- | --- |
| Milestone | Date Completed |
| Requirements Finalized | May 26, 2025 |
| Setup & Model Integration | May 28, 2025 |
| Core Features Completed | June 5, 2025 |
| Forecasting & Anomaly Features | June 10, 2025 |
| Feedback & Chatbot Module | June 15, 2025 |
| Final Integration & Testing | June 20, 2025 |
| Report & Presentation Ready | June 24, 2025 |
| Final Submission | June 25, 2025 |

**6. FUNCTIONAL AND PERFORMANCE TESTING**

**6.1 Performance Testing**

The Sustainable Smart City Assistant was tested in **Google Colab** with the **IBM Granite 3.3-2B Instruct model** integrated through Hugging Face. Performance and stability were key priorities given the AI-driven response generation and file handling.

**✅ Functional Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| Module | Test Scenario | Expected Outcome | Result |
| Waste Sorting | Input: “Banana peel” | Suggests composting or biodegradable disposal | Passed |
| Energy Advisor | Input: “Leave fan on overnight” | Recommends switching off unused appliances | Passed |
| Policy Expert | Input: Sample EV policy document | Generates 3–5 citizen-friendly key points | Passed |
| Resource Forecasting | Upload: CSV with "usage" column | Forecasts next-period usage with +10% projection | Passed |
| Anomaly Detection | Upload: CSV with outlier values | Identifies values beyond ±2 SD from mean | Passed |
| Feedback Submission | Input: Issue text and category | Saves to Excel and confirms log | Passed |
| File Download | Click download button after feedback is saved | Downloads smart\_city\_feedback.xlsx | Passed |
| Sustainability Chatbot | Input: “How to reduce plastic in cities?” | AI returns informative, conversational answer | Passed |
| Green Challenge | Click challenge button | Returns one random eco-challenge | Passed |
| Eco Tips | Input: “Solar panels” | Provides 3 useful tips for city residents | Passed |

**🚫 Bug Fixes & Improvements**

|  |  |
| --- | --- |
| Issue | Fix Implemented |
| Long output getting truncated | Increased max\_length in model response |
| File error for Excel output | Added fallback for empty feedback list |
| Gradio layout overlapping on mobile | Adjusted column spacing and CSS padding |
| Colab timeout on long model generation | Optimized prompt size and reduced token limit |

**✅ Final Validation**

* All modules performed as expected
* AI-generated responses were contextually relevant
* CSV upload and file handling worked across multiple test cases
* The Gradio UI responded well on both desktop and mobile views

**🚀 Deployment**

* The app was deployed using **Gradio + Pyngrok** on **Google Colab**
* Link shared for testing and demonstration
* No external hosting required, ensuring free and fast access

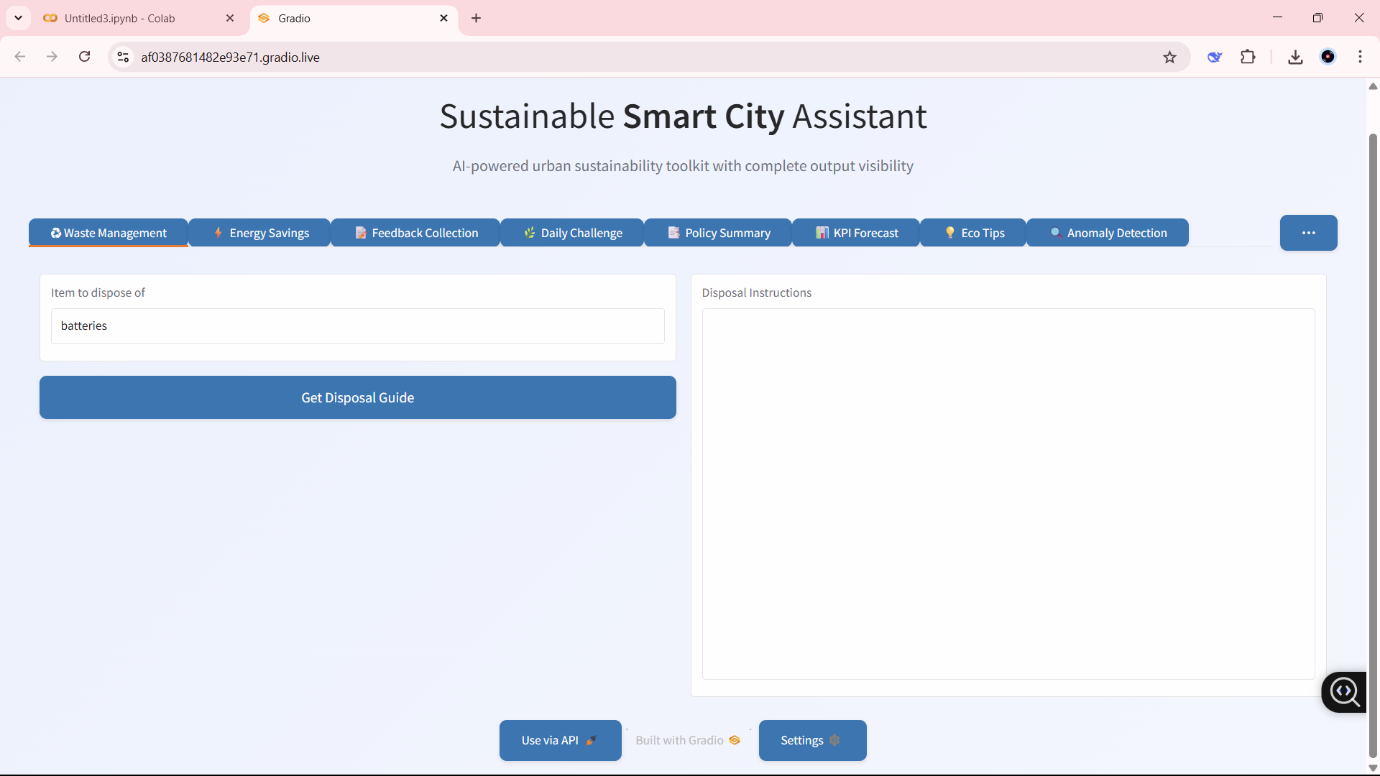
**7. RESULTS**

**7.1 Output Screenshots**

Below are the results of various modules successfully executed in the Sustainable Smart City Assistant. The screenshots validate both the functionality and the user interface design.

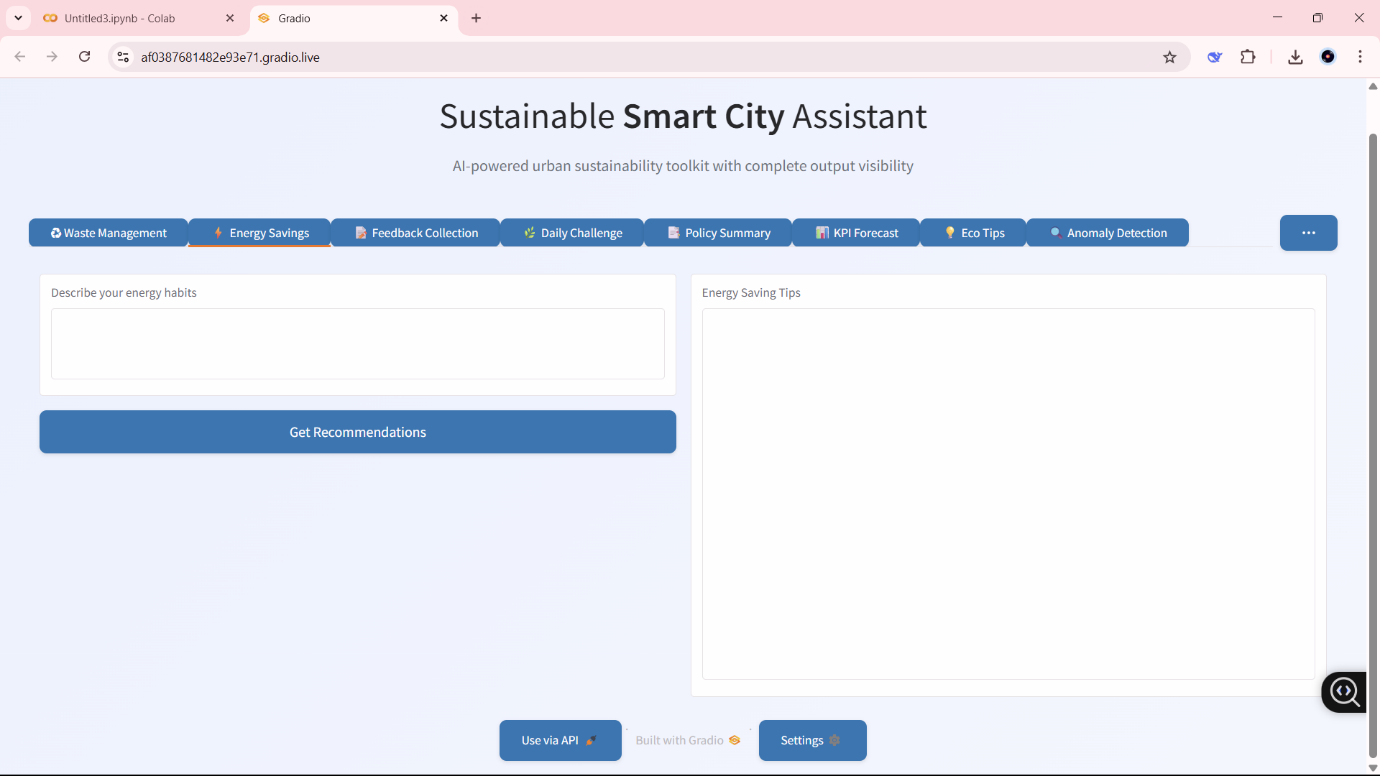
**📌 1. Waste Management Module**

**Description:** Input an item (e.g., "Plastic bottle") and receive eco-friendly disposal instructions.



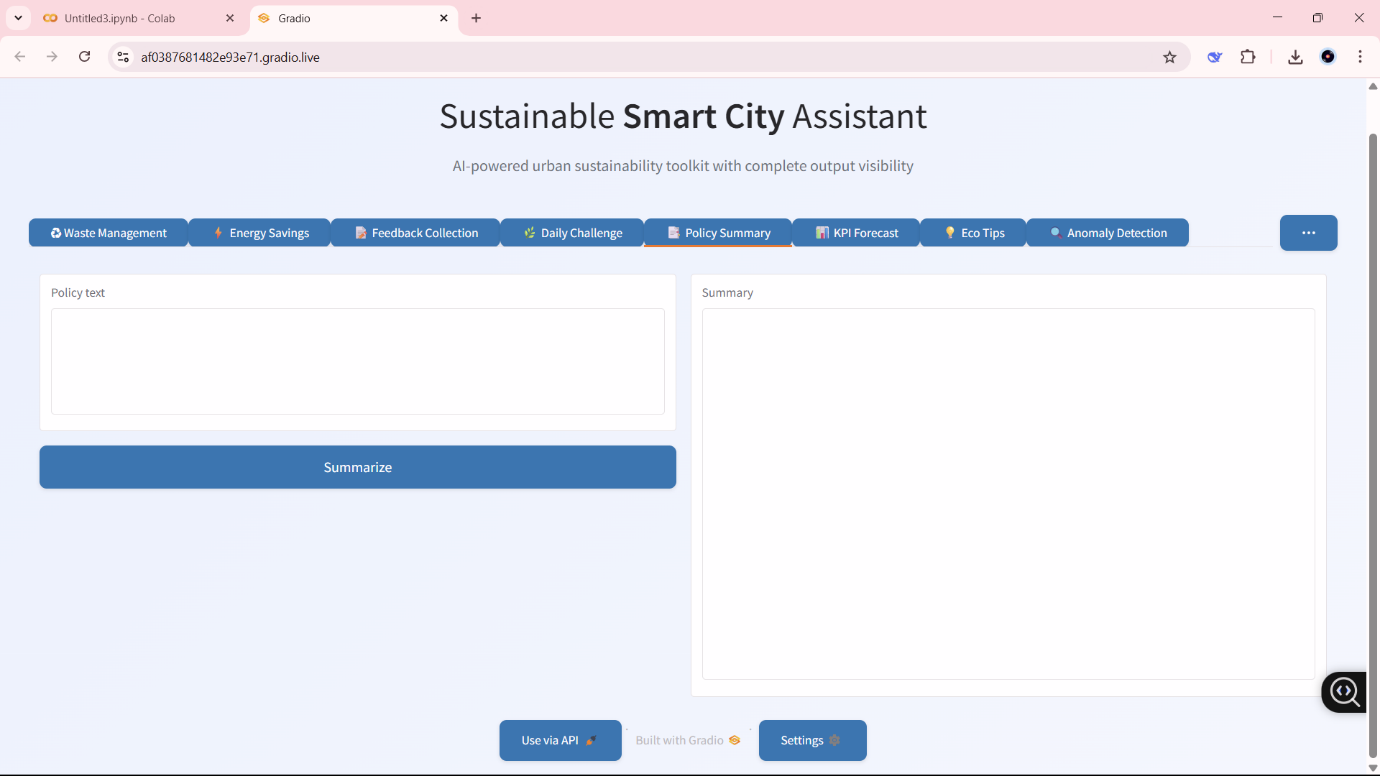
**📌 2. Energy Advisor**

**Description:** Users describe their energy habits and get personalized suggestions.



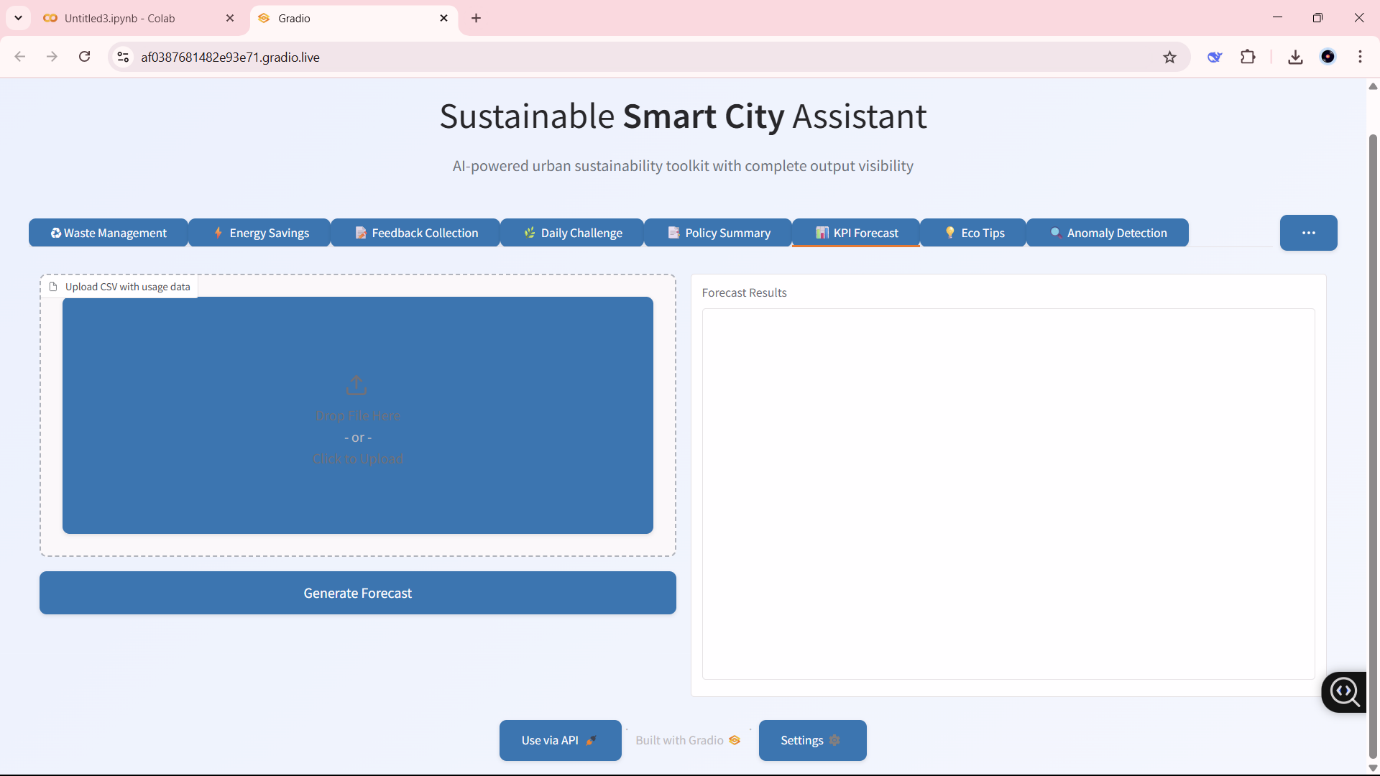
**📌 3. Policy Summary**

**Description:** Users paste policy text and receive simplified summaries in key points.

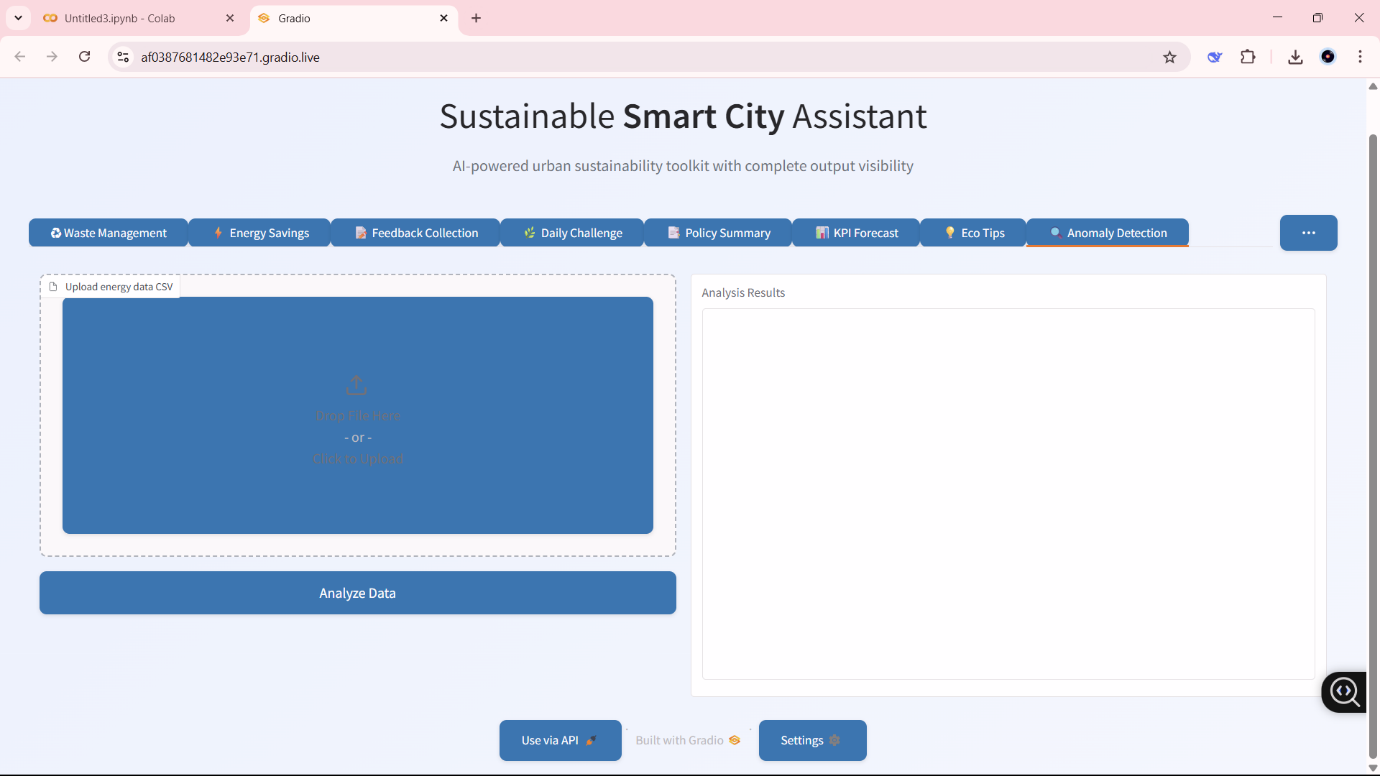


**📌 4. KPI Forecasting Module**

**Description:** Upload CSV with usage data and forecast the next period’s resource consumption.

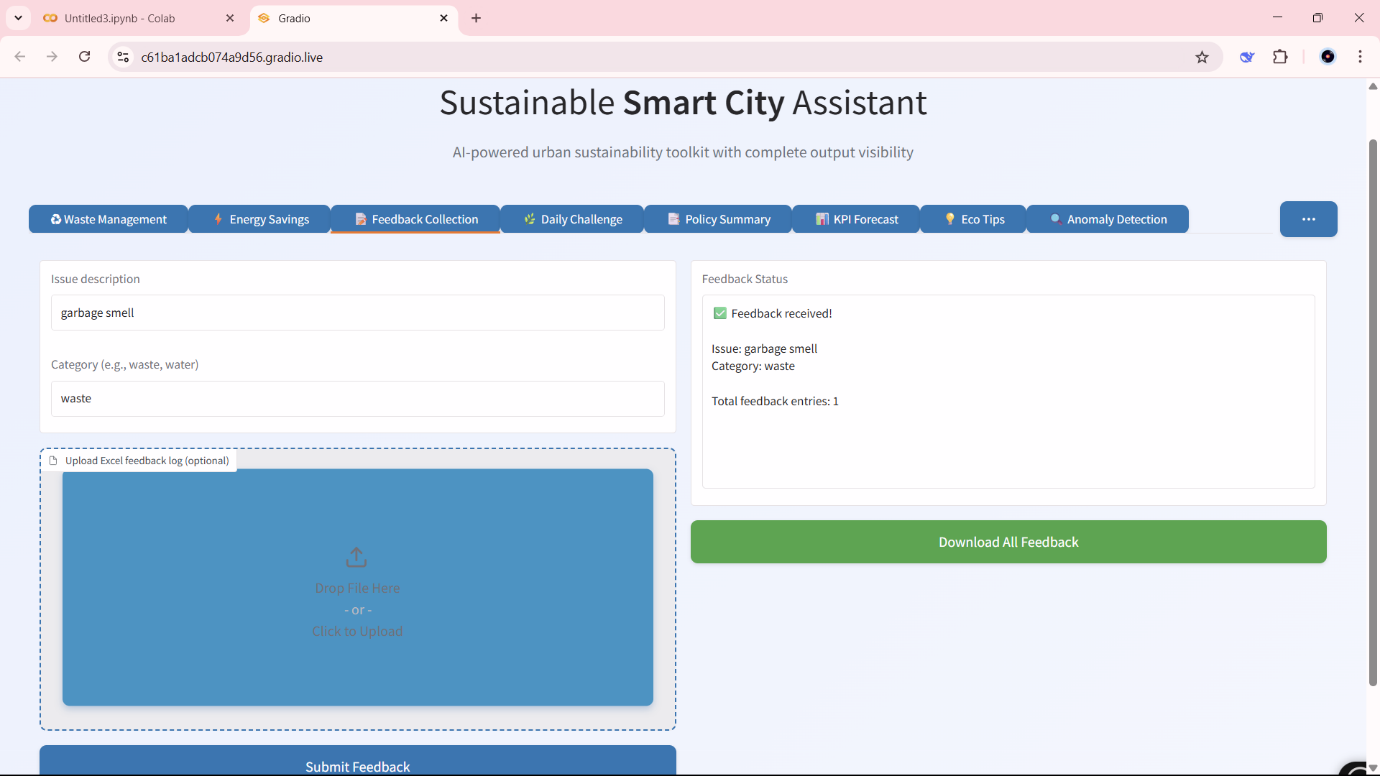


**📌 5. Anomaly Detection**

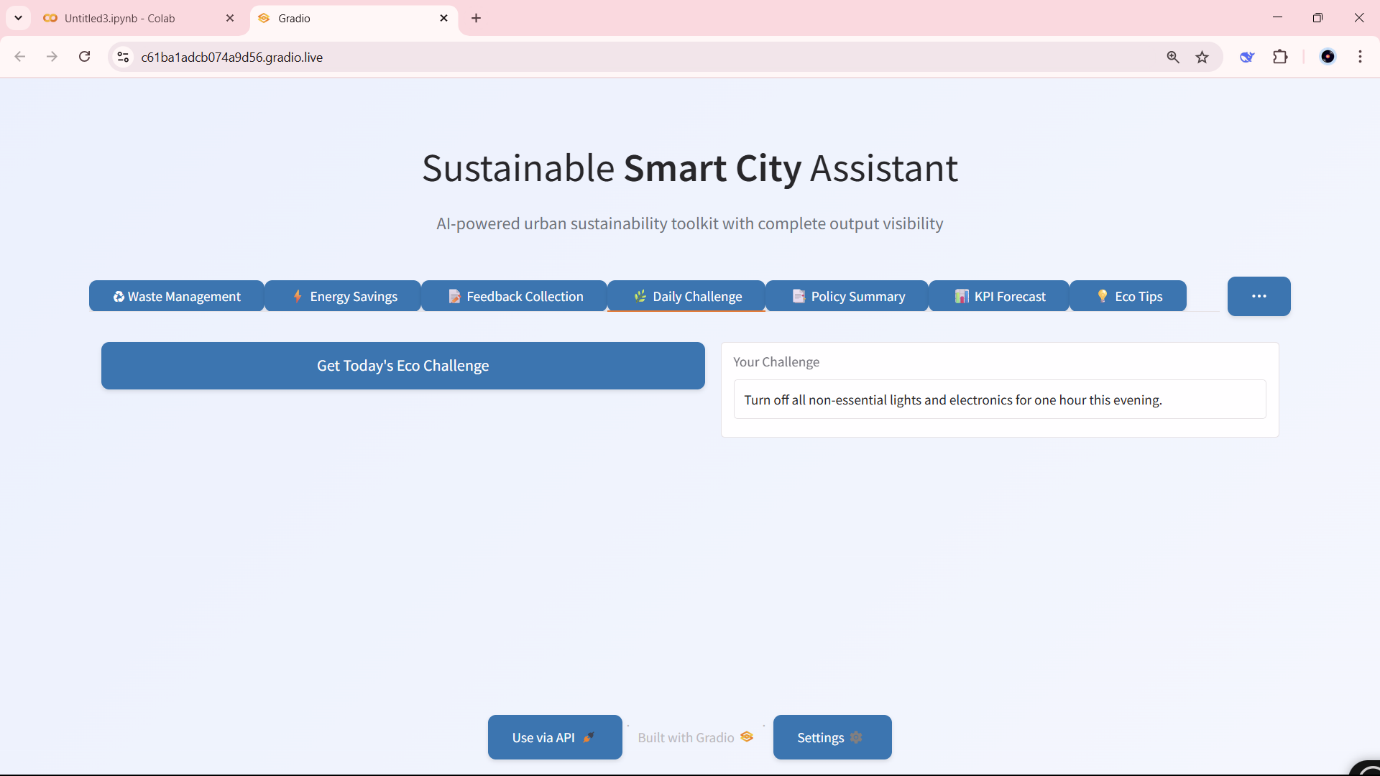
**Description:** Detects unusual data points in uploaded usage files using

**📌 6. Feedback Collection**

**Description:** Users report civic issues and download feedback logs as an Excel file.

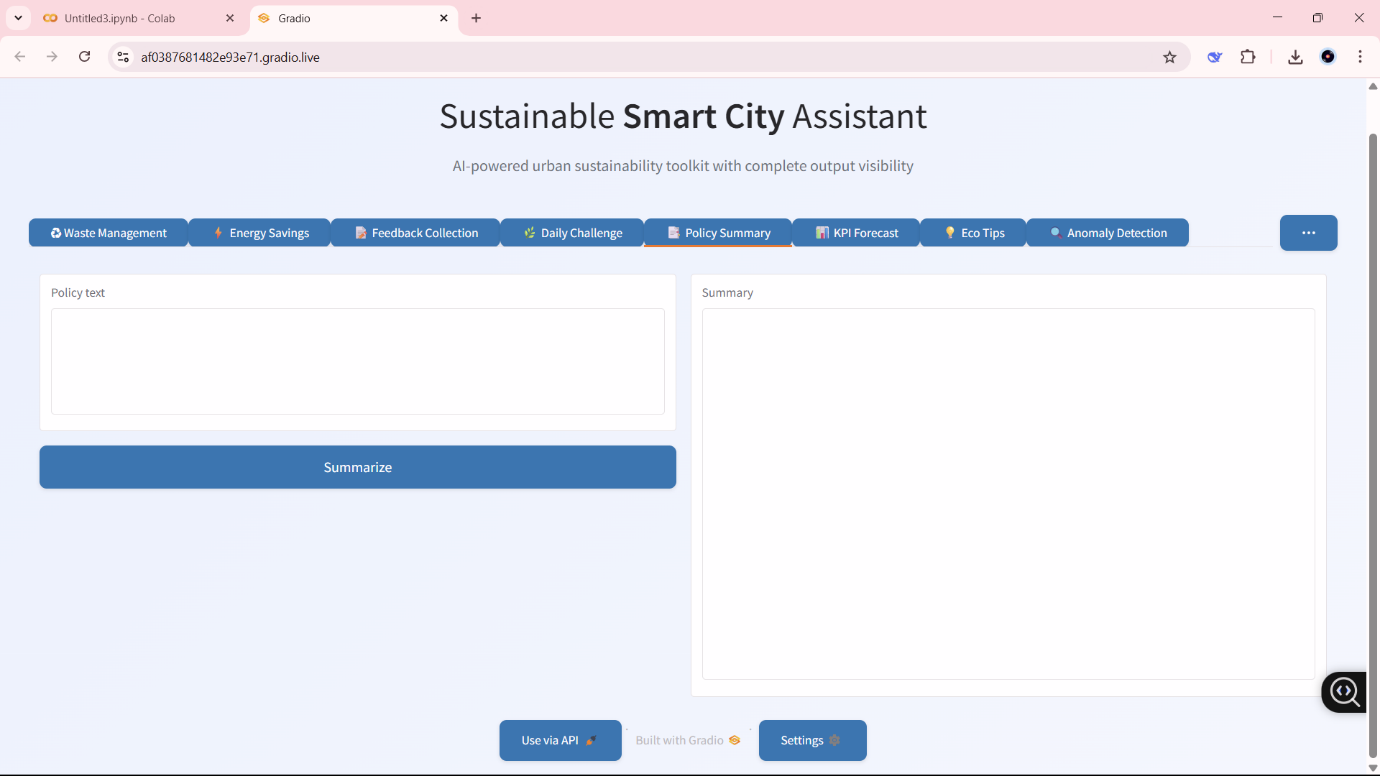


**📌 7. Daily Challenge**

**Description:** A random sustainability challenge is shown daily

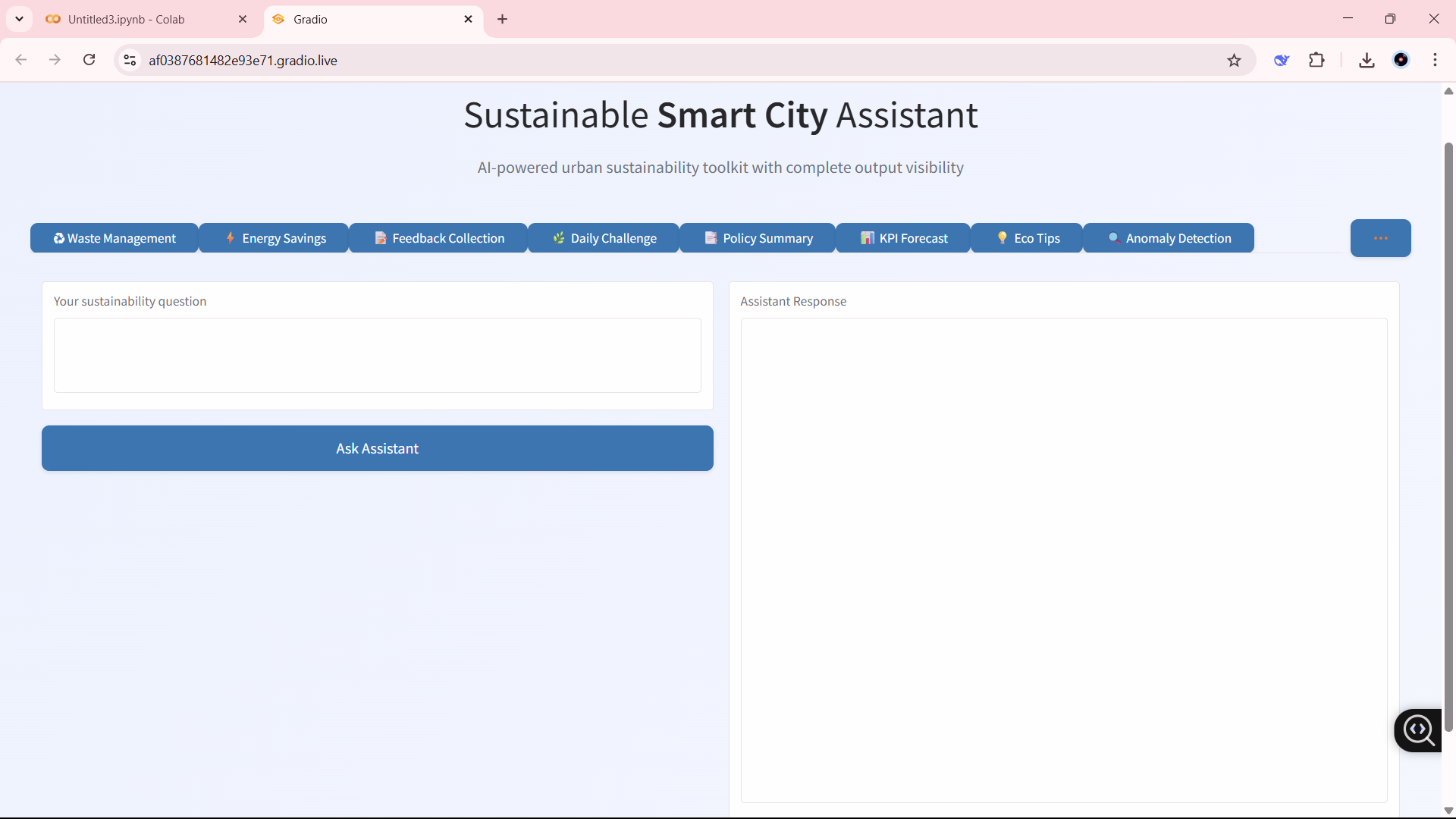
**📌 8. Eco Tips**

**Description:** Tips for eco-friendly practices based on a keyword (e.g., "solar").



**📌 9. Chat Assistant**

**Description:** Users ask questions like “How can my city reduce plastic?” and get AI responses.



**8. ADVANTAGES & DISADVANTAGES**

**✅ Advantages**

1. **AI-Powered Assistance**  
   The tool leverages IBM Granite to provide intelligent, real-time suggestions and summaries, making sustainability approachable for everyone.
2. **Modular Interface with Gradio**  
   The use of tabs keeps the UI organized and intuitive, offering a clean experience even with multiple functionalities.
3. **No Installation Needed**  
   Hosted on **Google Colab** and accessible via **Pyngrok**, users can run the assistant from any browser without setup.
4. **Dynamic CSV Analysis**  
   Upload-based forecasting and anomaly detection allow real-time resource monitoring for smart city planning.
5. **Citizen Engagement**  
   Enables feedback logging and civic participation, encouraging users to report local issues constructively.
6. **Educational Value**  
   Provides eco-tips, green challenges, and simplified policy summaries to boost awareness and behavioral change.
7. **Lightweight & Free to Use**  
   No costly infrastructure or servers needed; everything runs in the cloud with open-source tools.

**⚠️ Disadvantages / Limitations**

1. **Model Execution Delay**  
   Generating responses with IBM Granite may cause slight delays due to the model size and Colab limitations.
2. **Session-Based Feedback Loss**  
   Unless downloaded manually, feedback logs reset if the Colab session ends, limiting long-term data storage.
3. **Limited Real-Time Data**  
   The assistant does not yet integrate with live smart city APIs (e.g., pollution, transport, weather data).
4. **Dependency on External Hosting**  
   Requires stable internet and Colab runtime; not deployable offline or on local systems without modification.
5. **No Multi-language Support**  
   Currently, the assistant responds only in English, which may limit usability in multilingual urban settings.

**9. CONCLUSION**

The **Sustainable Smart City Assistant** successfully demonstrates how artificial intelligence can support urban sustainability through accessible and intelligent digital tools. By combining Gradio's interactive UI with the powerful IBM Granite model, the assistant offers practical solutions to key city challenges—such as waste sorting, energy saving, policy understanding, and data-driven forecasting.

The project achieved its intended goals of:

* Encouraging eco-friendly habits
* Simplifying sustainability knowledge for the public
* Empowering citizens to participate in smart city development
* Offering analytical insights for better resource planning

Its modular design, cloud-based deployment, and AI-driven responses make it a scalable and adaptable tool for both individual users and city planners. The assistant promotes not only environmental awareness but also responsible civic participation, making it a small but significant step toward smarter, greener cities.

**10. FUTURE SCOPE**

The Sustainable Smart City Assistant lays a strong foundation, but there are several opportunities to enhance its functionality, impact, and reach in future iterations:

**🔮 1. Real-time Data Integration**

* Connect to **live APIs** for pollution levels, electricity usage, traffic conditions, and water supply to provide context-aware suggestions.
* Example: Show energy-saving tips based on live grid load or suggest routes based on air quality.

**🌐 2. Multilingual Support**

* Integrate **language translation models** to support regional languages and increase accessibility for diverse populations across urban India and beyond.

**🤖 3. Voice Assistant Compatibility**

* Expand the assistant into **voice-based platforms** like Alexa, Google Assistant, or a mobile app using speech-to-text and text-to-speech models.

**📊 4. Admin Dashboard for Governments**

* Build an analytics dashboard for city officials using **collected feedback, anomaly data, and forecast insights** for smart decision-making.

**🧠 5. Advanced AI & Personalization**

* Use **user behavior and preferences** to personalize tips, forecast models, and daily challenges.
* Potential to train smaller, optimized AI models for faster edge deployment.

**🔁 6. Continuous Learning**

* Improve recommendations over time using **reinforcement learning** or **feedback loops**, allowing the assistant to learn from user inputs and choices.

**📱 7. Mobile App Deployment**

* Convert the web-based Gradio interface into a **cross-platform mobile application** (using tools like React Native or Flutter) for wider adoption.

These future enhancements can significantly increase the tool’s utility, adoption, and contribution toward building truly **sustainable smart cities**.

**11. APPENDIX**

**📂 11.1 Source Code**

The complete source code for the project, including all Gradio interface components, backend logic, and model integration using IBM Granite, is available in the GitHub repository below:

🔗 **GitHub Repository**:  
*https://github.com/anilpreetham/Sustainable-Smart-city-Assistant.git*

**📊 11.2 Dataset / Sample Files**

The project used user-uploaded .csv files for forecasting and anomaly detection modules. A sample dataset is included in the repository under /data.

* **Sample Usage File**:

|  |  |
| --- | --- |
| Month | usage |
| Jan | 1234 |
| Feb | 1180 |
| Mar | 1300 |
| Apr | 1275 |
| May | 1402 |
| Jun | 1500 |
| Jul | 1555 |
| Aug | 1532 |
| Sep | 1450 |
| Oct | 1370 |
| Nov | 1295 |
| Dec | 1250 |

**🧪 11.3 Model Information**

* **Model Used**: IBM Granite 3.3-2B Instruct
* **Hosted On**: Hugging Face
* **Integrated Via**: transformers library

🧠 Granite is an instruction-tuned open-source model well-suited for chat assistant, summarization, and text generation tasks in this project.