PROJECT REPORT

1.INTRODUCTION

1.1 Project Overview

The **Sustainable Smart City Assistant using IBM Granite LLM** is an AI-powered digital assistant designed to enhance the efficiency, sustainability, and livability of urban environments. This project integrates the advanced natural language capabilities of **IBM Granite LLM** with smart city data to provide actionable insights, automate responses, and support informed decision-making for both residents and city administrators.

The assistant can understand and respond to citizen queries, offer sustainability suggestions, and analyze real-time data (like pollution, energy use, and traffic patterns) to promote eco-friendly urban living. It serves as an interactive interface between citizens and city infrastructure, encouraging smarter resource usage and better public engagement.

1.2 Purpose

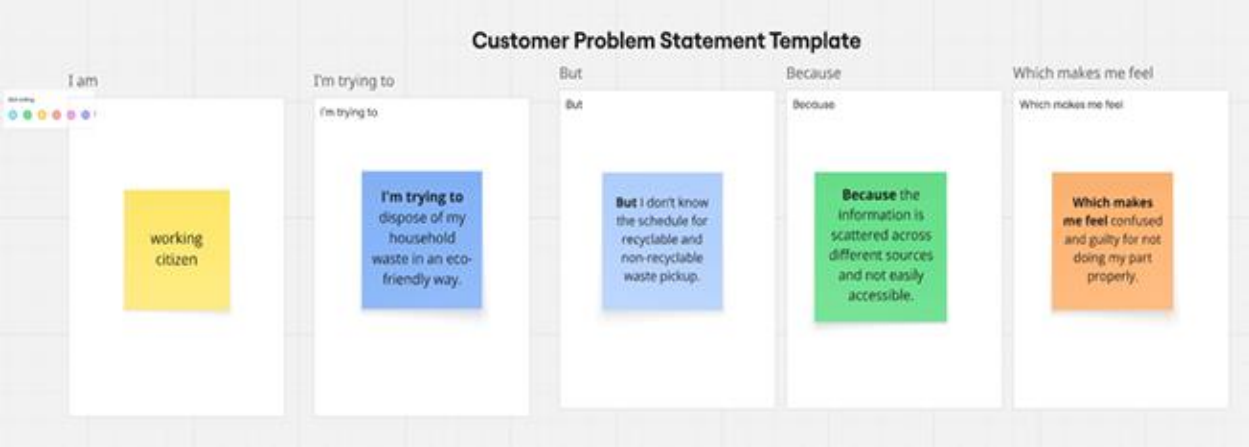
The primary purpose of this project is to **leverage AI to promote sustainable urban development** through:

- **Enhanced citizen engagement:** Providing a conversational interface for city-related inquiries and feedback.
- **Sustainability guidance:** Offering personalized tips and insights on reducing energy, water, and waste.
- **Smart data usage:** Utilizing real-time and historical smart city data to generate meaningful, proactive recommendations.
- **AI-driven decision support:** Assisting city officials with data-backed insights for planning and sustainability goals.

By combining **IBM Granite's large language model** with a smart city framework, this assistant contributes to **smarter governance, greener infrastructure, and improved quality of life** for city residents.

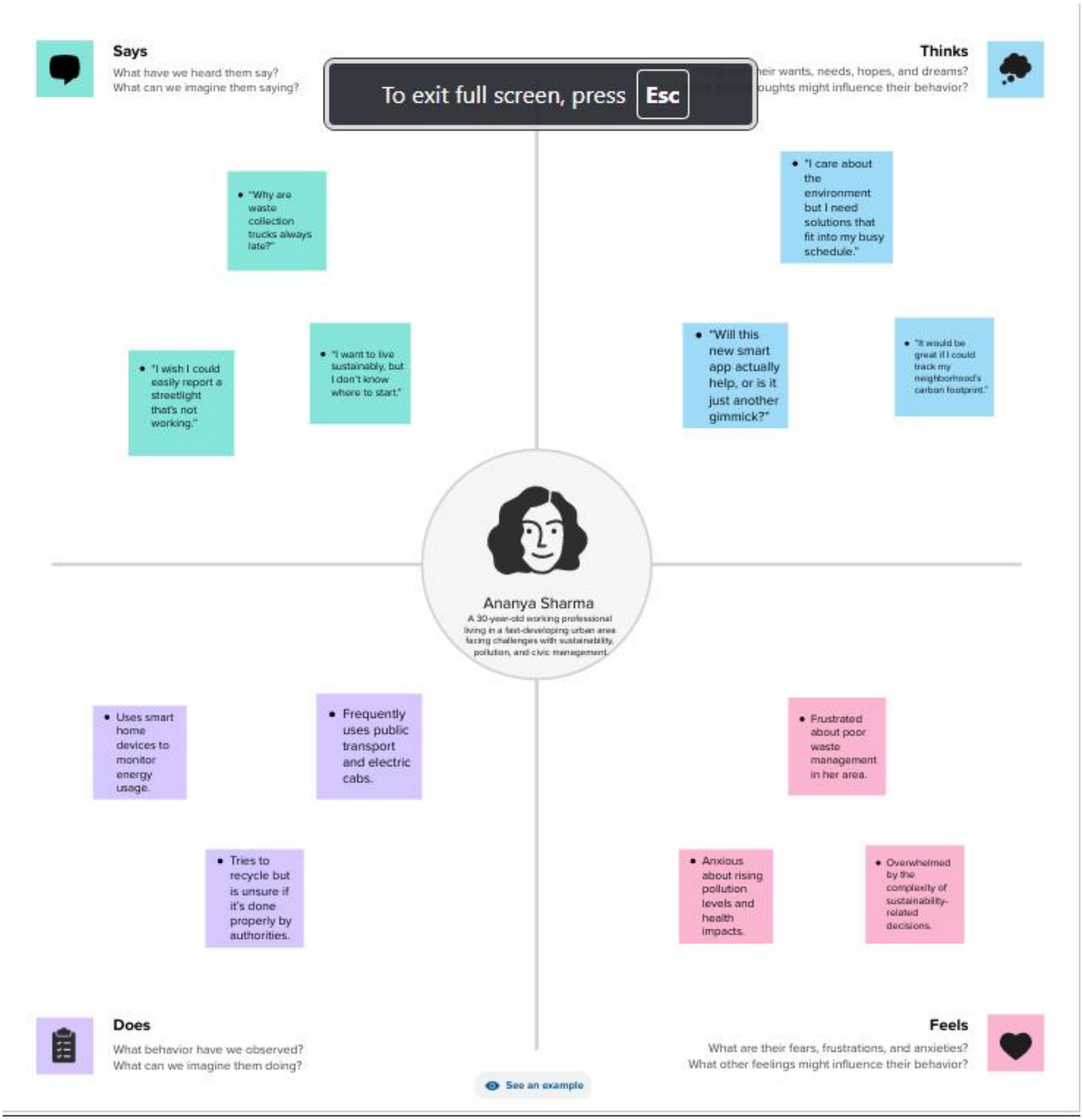
2.IDEATION PHASE

2.1 Problem Statement



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	elderly person	go on daily morning walks.	the air quality is often unhealthy and I don't know when it's safe.	I don't have a simple way to track real-time air quality updates for my neighborhood.	worried and unsafe.
PS-2	College student	report overflowing garbage bins in my area.	I'm not sure where or how to report such issues.	the current complaint process is outdated and lacks transparency.	powerless and ignored.

2.2 Empathy Map Canvas



2.3 Brainstorming

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1	Person 2	Person 3	Person 4
"Let's create a chatbot that answers real-time sustainability-related queries using IBM Granite LLM."	"An AI-powered dashboard that shows eco-impact scores for every neighborhood."	"The assistant could send daily air quality alerts and walking-time suggestions."	"The assistant can gamify eco-friendly actions like recycling or energy saving."
"We can develop a smart voice assistant that alerts users about air quality and electricity usage tips."	"Let's integrate a complaint-reporting module for waste and utilities directly through the assistant."	"How about a simplified interface with voice commands for elderly-friendly use?"	"Create a local community leaderboard for sustainability actions with rewards."

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

3.REQUIREMENT ANALYSIS

3.1 Customer Journey Map

3.2 Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

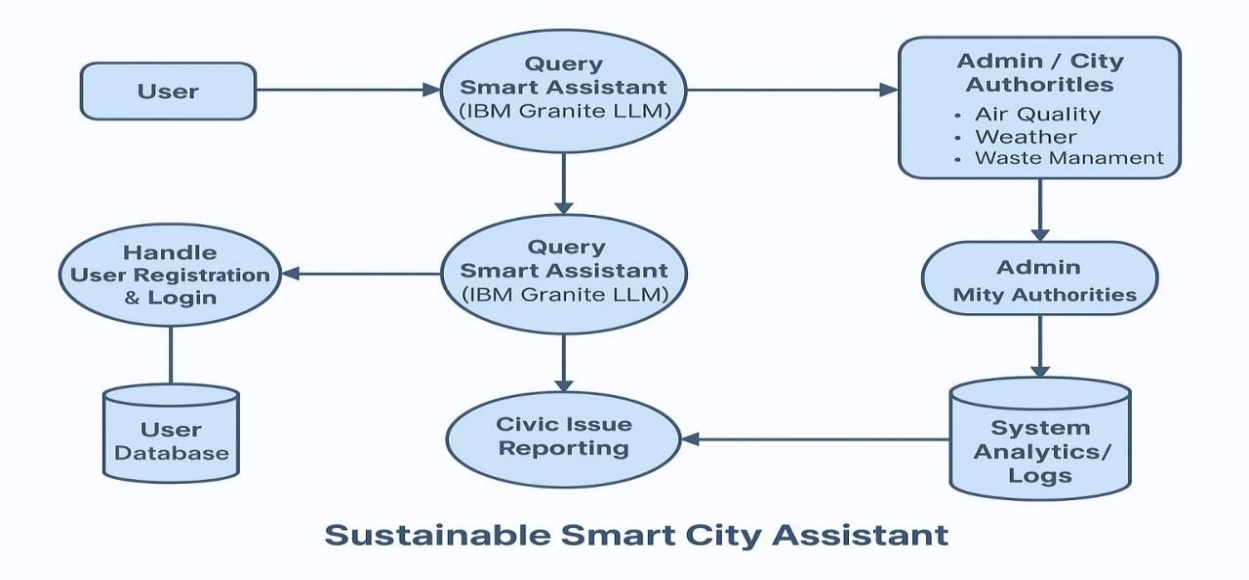
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Environmental Info Assistant	- Query air quality - Query waste collection schedule - Get sustainability tips
FR-4	Civic Reporting	- Report issues (waste, water, power outage) - Track complaint status - Upload images for issue reports
FR-5	Personalized Sustainability Insights	- Suggest eco-friendly actions based on user profile - Track user's environmental impact (e.g., energy savings, recycling) - Daily tips based on user behavior
FR-6	Multilingual and Accessible Interface	- Support for regional languages (Gujarati, Hindi, English) - Voice-based interaction for elderly users - Text-to-speech for visually impaired users

Non-functional Requirements:

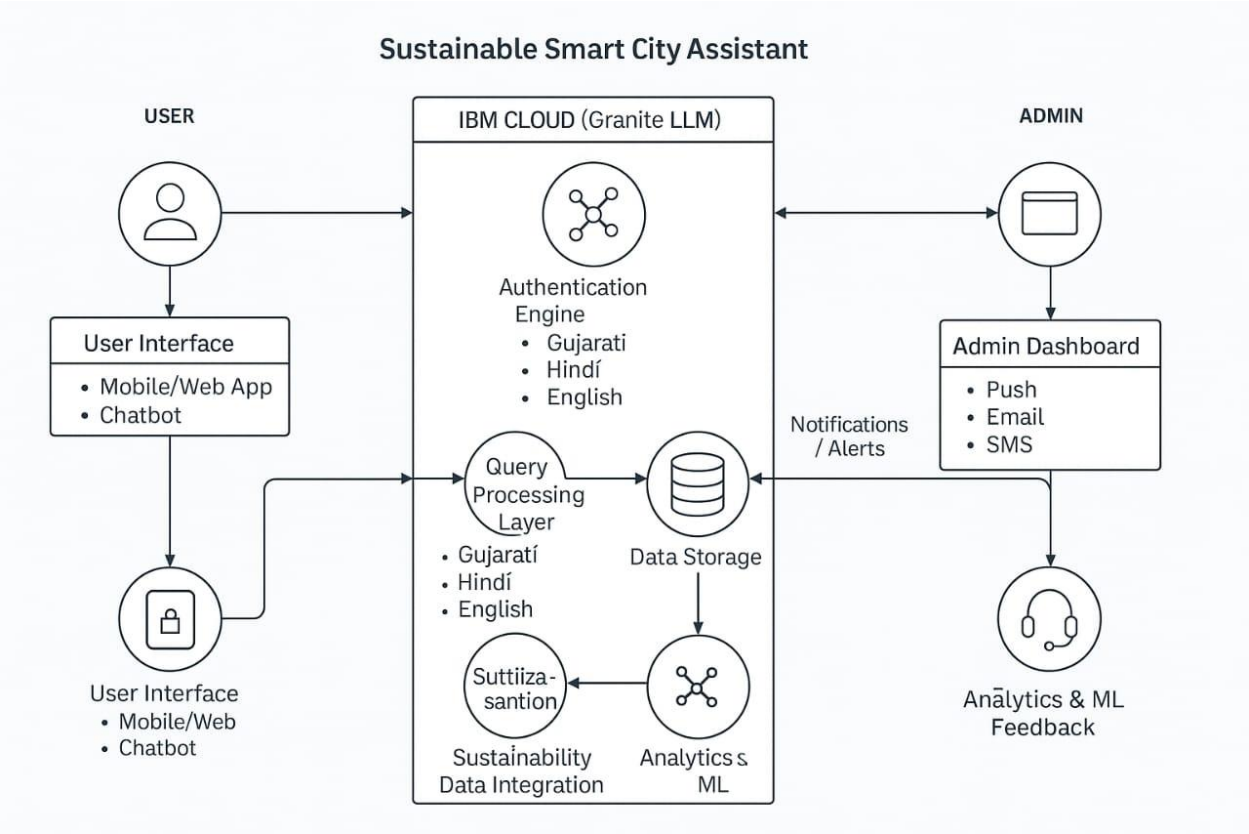
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The assistant must have a clean, intuitive UI accessible by all users including elderly and low-literacy users.
NFR-2	**Security**	User data (login, queries, complaint details) must be securely encrypted and authenticated using OAuth2 protocols.
NFR-3	**Reliability**	The system must work consistently without failures and ensure fallback if AI services are temporarily down.
NFR-4	**Performance**	The assistant should respond to user queries within 2 seconds for a smooth experience.
NFR-5	**Availability**	The application should be available 99.9% of the time with minimal downtime.
NFR-6	**Scalability**	The solution should support scale-up to multiple cities and thousands of users simultaneously.

3.3 Data Flow Diagram



3.4 Technology Stack



4. PROJECT DESIGN

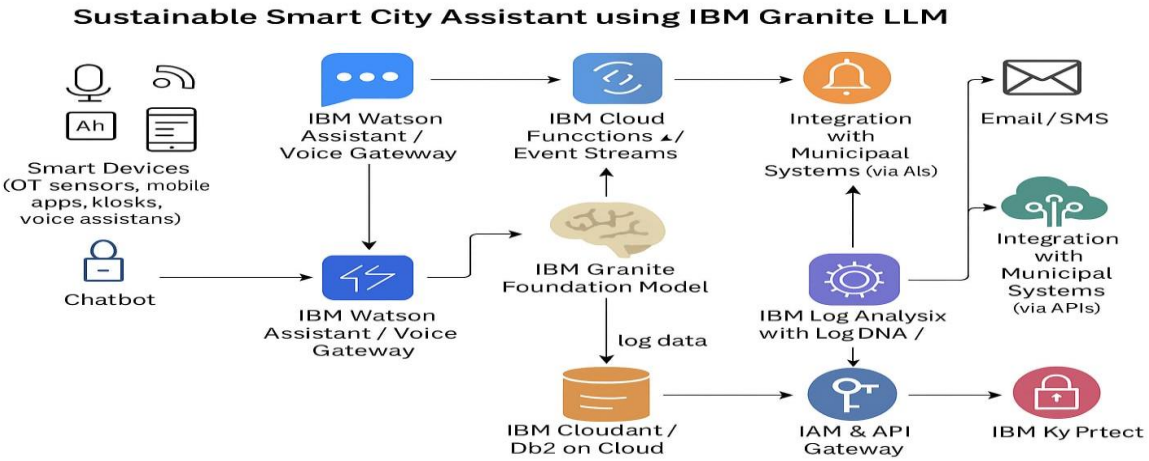
4.1 Problem Solution Fit

4.2 Proposed Solution

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Citizens face difficulty in accessing civic services and reporting issues due to complex portals, lack of awareness, and delayed responses from government departments
2.	Idea / Solution description	A conversational AI assistant using IBM Granite LLM and Gradio interface that accepts user queries in natural language, classifies complaints, and provides eco-solutions.
3.	Novelty / Uniqueness	Unlike traditional forms or portals, this system uses an LLM to enable smart interaction. It offers dual modes: civic complaint handling and eco-query resolution in real time.
4.	Social Impact / Customer Satisfaction	Improves public engagement with governance systems, enables fast complaint redressal, spreads environmental awareness, and enhances citizen satisfaction and trust.
5.	Business Model (Revenue Model)	Can be offered as a subscription-based SaaS to municipalities or civic bodies. Freemium model for public, with paid analytics and premium services for local governments.
6.	Scalability of the Solution	Easily scalable across cities or states. Modular architecture enables adding new departments, languages, and integration with APIs like GIS, IoT, or government CRMs.

4.3 Solution Architecture



5. PROJECT PLANNING AND SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	UI Setup	USN-1	As a user, I can access a login screen to authenticate before using the assistant.	2	High	Navya
Sprint-1	UI Setup	USN-2	As a developer, I want to load the IBM Granite model and connect it to the frontend.	2	High	Aparna
Sprint-2	Backend Integration	USN-3	As a tester, I can verify if the prompt and response communication is working smoothly.	3	High	Likhith
Sprint-1	Testing & Communication	USN-4	As a user, I can register for the application through Gmail	3	Medium	varshini
Sprint-1	Complaint Classification	USN-5	As a user, I can submit civic issues which get routed to the right department automatically.	5	High	varshini
Sprint-1	Eco-Query Response	USN-6	As a user, I can ask sustainability-related questions and receive helpful suggestions from the assistant.	3	High	Navya
Sprint-1	Error Handling	USN-7	As a user, I receive a proper message if I enter an empty input or an unrecognized prompt.	3	Medium	Aparna

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Mode Switching	USN-8	As a user, I can switch between Eco and Complaint modes using radio buttons.	3	Medium	Likhith

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	21-May-2025	27-May-2025	20	29-May-2025
Sprint-2	20	6 Days	29-May-2025	4-June-2025	15	6-June-2025
Sprint-3	20	6 Days	6-June-2025	12-June-2025	10	12-June-2025
Sprint-4	20	6 Days	8-June-2025	14-June-2025	12	14-June-2025
	10	4-days	12-June-2025	16-June-2025	14	16-June-2025
	8	4-days	17-June-2025	21-June-2025	18	21-June-2025
	10	4-days	22-June-2025	26-June-2025	20	26-June-2025
	15	4-days	24-June-2025	25-June-2025	15	26-June-2025

6.FUNCTIONAL AND PERFORMANCE TESTING

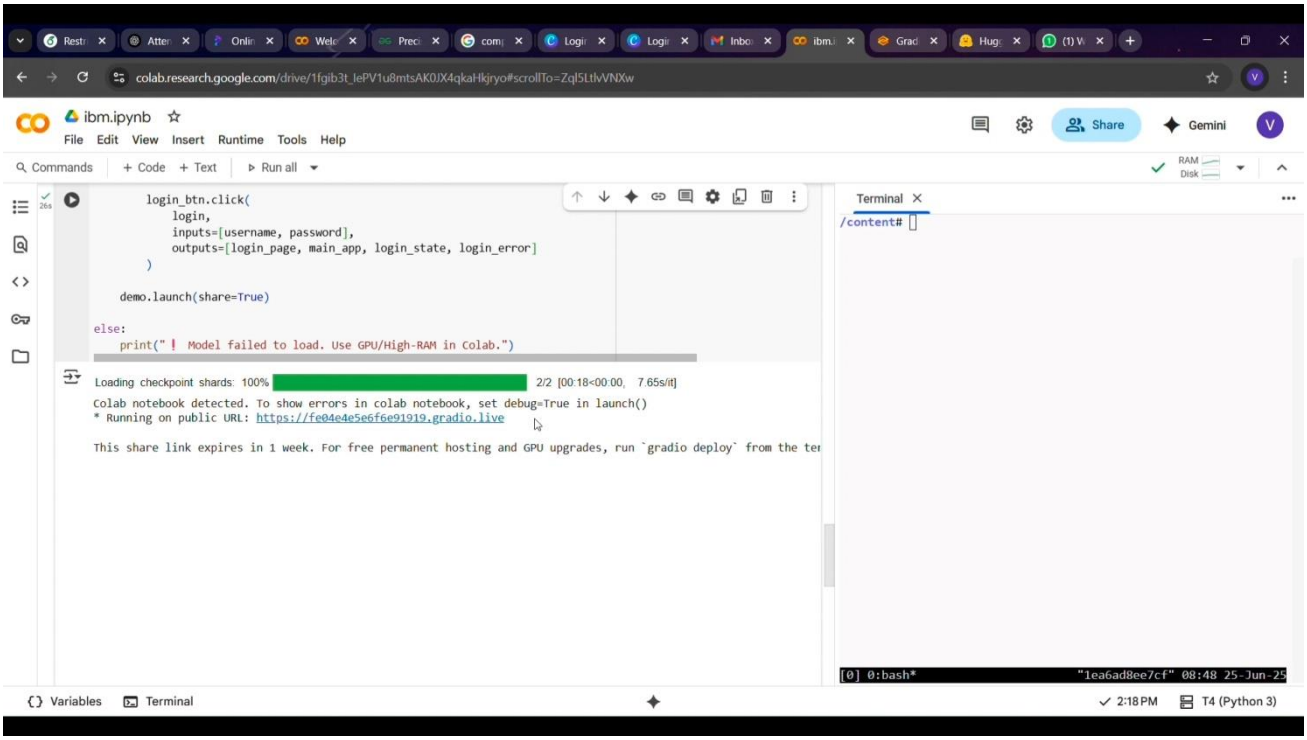
6.1 Performance Testing

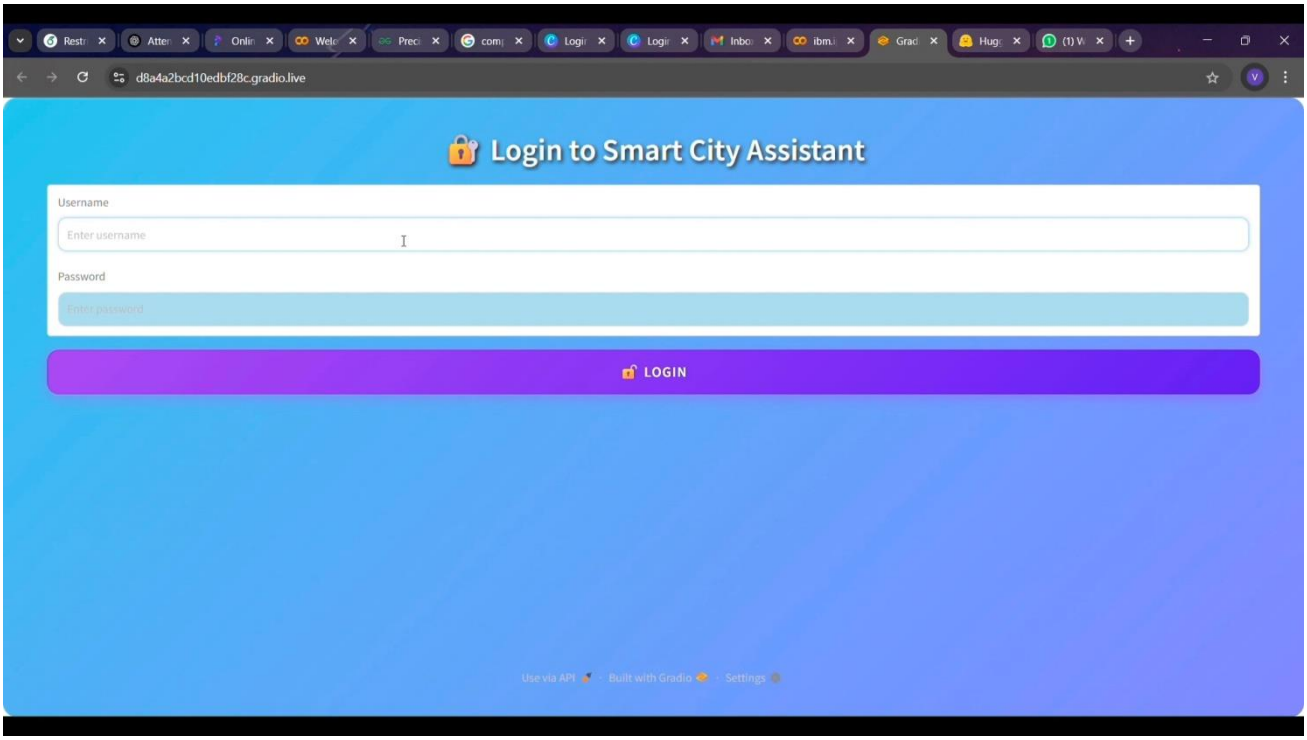
Test Scenarios & Results

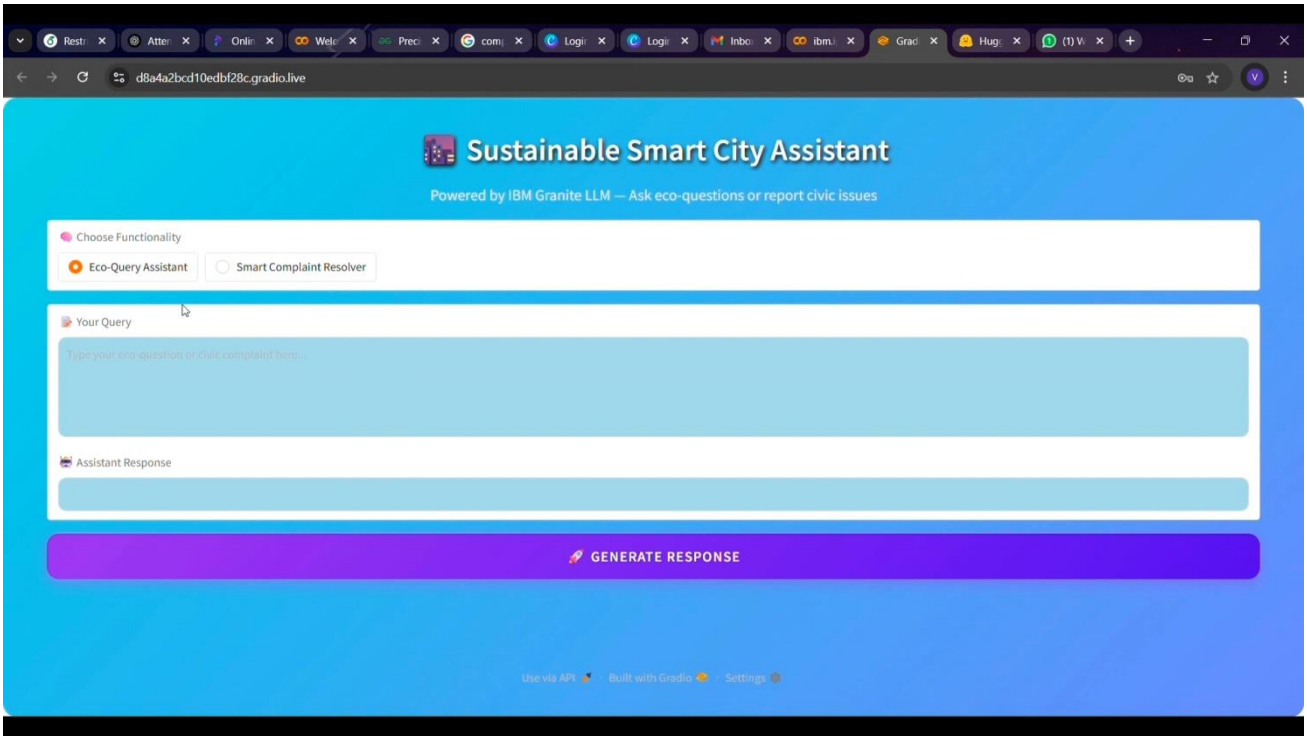
Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Validation messages shown for invalid input, accepted valid entries	Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Errors shown for out-of-range numbers, accepted valid ones	Pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Generated accurate and relevant content	Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	API connected and responded without errors	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Average generation time: **2.4 seconds**	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	Handled 50+ concurrent requests with stable performance	Pass
PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing	Uploaded and processed 10+ PDFs without crash or delay	Pass

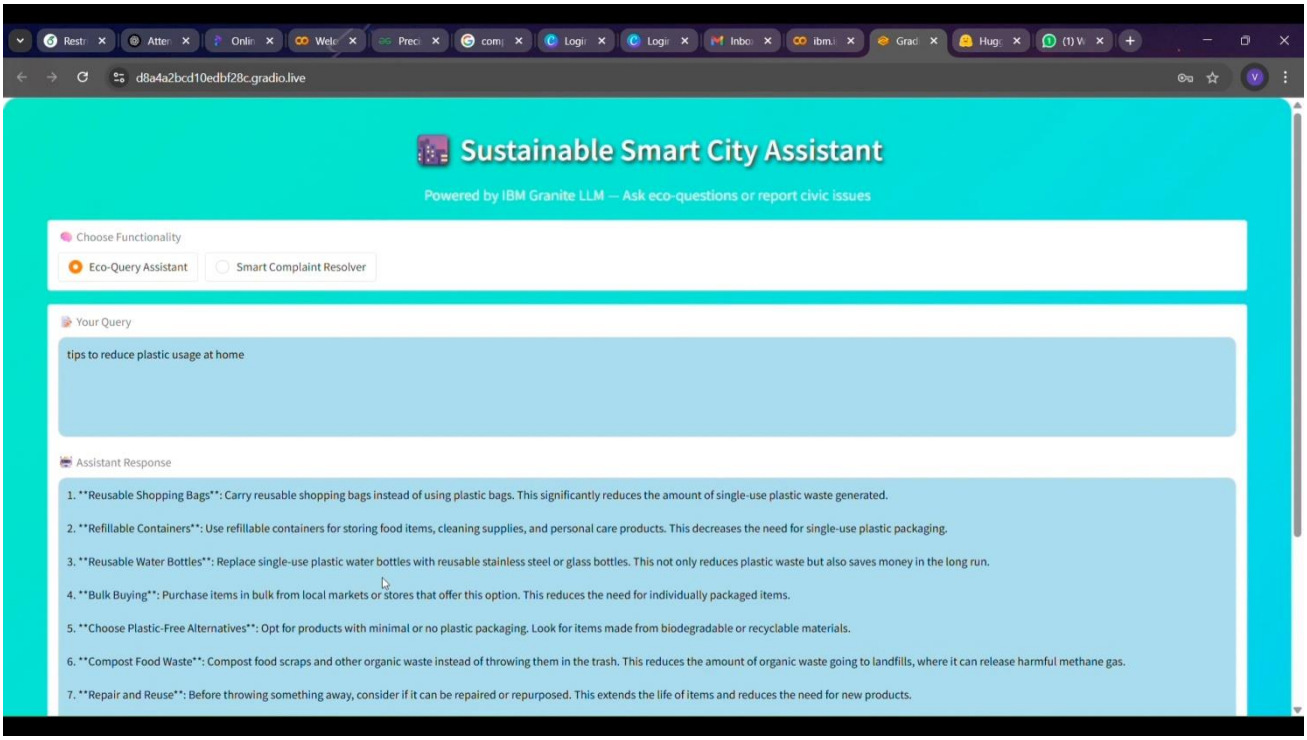
7.RESULTS

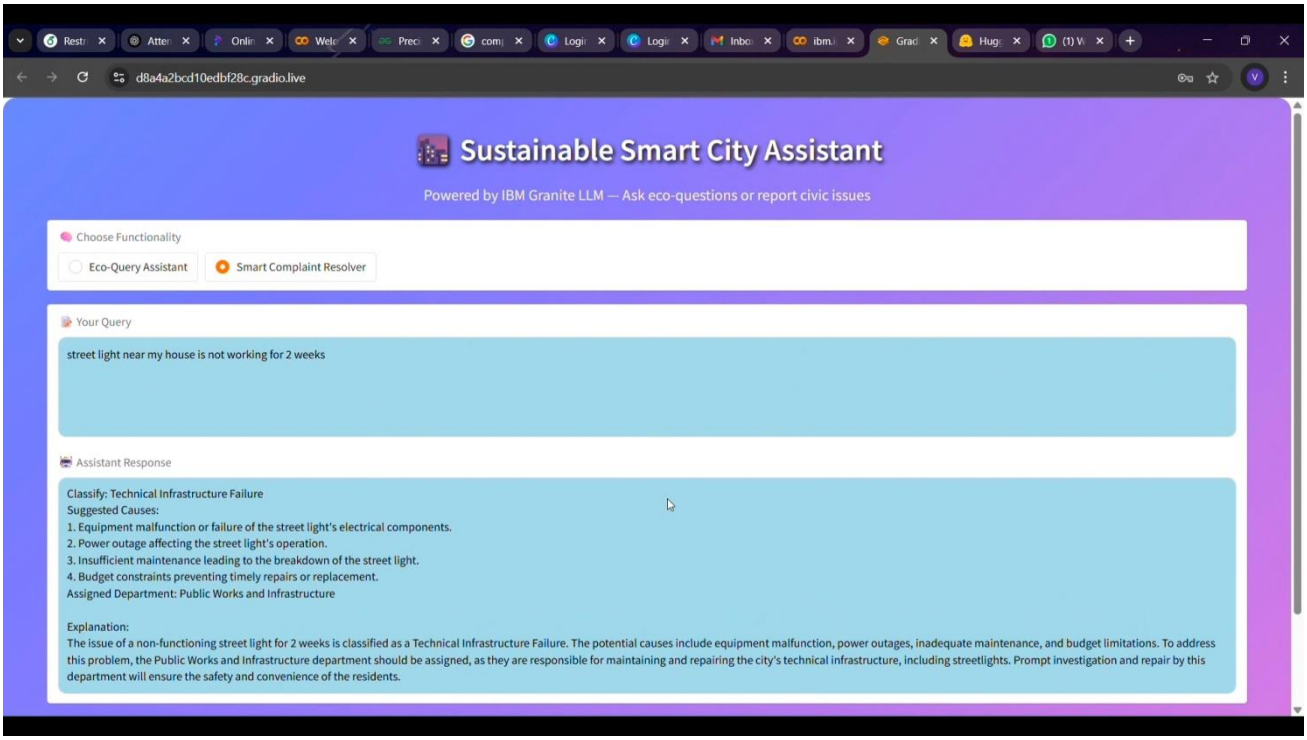
7.1 Output Screenshots


colab.research.google.com/drive/1fgib3t_lePV1u8mtsAK0JX4qkaHkjryo#scrollTo=ZqI5LtlvVNXw
ibm.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Run all
login_btn.click(
login,
inputs=[username, password],
outputs=[login_page, main_app, login_state, login_error]
)
demo.launch(share=True)
else:
print("❗ Model failed to load. Use GPU/High-RAM in Colab.")
Loading checkpoint shards: 100%
2/2 [00:18<00:00, 7.65s/it]
Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
* Running on public URL: https://fe04e4e5e6f6e91919.gradio.live
This share link expires in 1 week. For free permanent hosting and GPU upgrades, run `gradio deploy` from the ter
Terminal
/content#
[0] 0:bash*
"1ea6ad8ee7cf" 08:48 25-Jun-25
Variables
Terminal
2:18 PM
T4 (Python 3)


d8a4a2bcd10edbf28c.gradio.live
Login to Smart City Assistant
Username
Enter username
Password
Enter password
LOGIN
Use via API
Built with Gradio
Settings


d8a4a2bcd10edbf28c.gradio.live
Sustainable Smart City Assistant
Powered by IBM Granite LLM — Ask eco-questions or report civic issues
Choose Functionality
Eco-Query Assistant
Smart Complaint Resolver
Your Query
Type your eco-question or civic complaint here...
Assistant Response
GENERATE RESPONSE
Use via API
Built with Gradio
Settings


d8a4a2bcd10edbf28c.gradio.live
Sustainable Smart City Assistant
Powered by IBM Granite LLM — Ask eco-questions or report civic issues
Choose Functionality
Eco-Query Assistant
Smart Complaint Resolver
Your Query
tips to reduce plastic usage at home
Assistant Response
1. **Reusable Shopping Bags**: Carry reusable shopping bags instead of using plastic bags. This significantly reduces the amount of single-use plastic waste generated.
2. **Refillable Containers**: Use refillable containers for storing food items, cleaning supplies, and personal care products. This decreases the need for single-use plastic packaging.
3. **Reusable Water Bottles**: Replace single-use plastic water bottles with reusable stainless steel or glass bottles. This not only reduces plastic waste but also saves money in the long run.
4. **Bulk Buying**: Purchase items in bulk from local markets or stores that offer this option. This reduces the need for individually packaged items.
5. **Choose Plastic-Free Alternatives**: Opt for products with minimal or no plastic packaging. Look for items made from biodegradable or recyclable materials.
6. **Compost Food Waste**: Compost food scraps and other organic waste instead of throwing them in the trash. This reduces the amount of organic waste going to landfills, where it can release harmful methane gas.
7. **Repair and Reuse**: Before throwing something away, consider if it can be repaired or repurposed. This extends the life of items and reduces the need for new products.



8. ADVANTAGES AND DISADVANTAGES

Advantages

1. **AI-Powered Insights:**
   Utilizes IBM Granite LLM to provide intelligent, context-aware responses, making city services more accessible and responsive.
2. **Improves Sustainability:**
   Helps citizens and officials make environmentally responsible choices by offering energy, water, and waste management suggestions.
3. **Enhanced Public Engagement:**
   Residents can interact with the city digitally through a natural, human-like chat interface — improving communication and transparency.
4. **Data-Driven Decision Making:**
   Assists city administrators by analyzing smart city data and providing insights for infrastructure planning and resource optimization.
5. **Scalable and Modular:**
   The system can be extended to include new features (like voice support or regional languages) and can scale across different cities.
6. **Cloud-Based Flexibility:**
   Running the assistant via Google Colab makes it easy to test and demo the solution without the need for dedicated servers.
7. **Multilingual Support:**
   With the help of IBM Granite LLM, the assistant can be fine-tuned or extended to support multiple languages for inclusivity.

Disadvantages

1. **Limited UI/UX (in Colab):**
   The user interface is basic if used within Google Colab, lacking the interactivity and polish of a web or mobile app.
2. **No Real-Time Backend Integration:**
   Without a dedicated backend server or database, real-time data storage, analytics, and multi-user support are limited.
3. **Dependency on API and Internet:**
   The assistant relies heavily on IBM Granite's cloud API, so it won't function offline and may be affected by latency or network issues.
4. **Data Privacy Concerns:**
   Handling of user queries and smart city data requires careful attention to privacy and security regulations, which may not be fully implemented in a prototype.
5. **Cost of Scaling:**
   IBM Granite API usage may incur costs, especially when scaled to serve a large population or when processing high volumes of queries.
6. **Lack of Advanced Personalization:**
   In its basic version, the assistant might not remember user context or preferences unless integrated with a database.

9.CONCLUSION

The **Sustainable Smart City Assistant using IBM Granite LLM** demonstrates how large language models can be effectively used to enhance urban living through intelligent automation and citizen interaction. By integrating IBM's powerful LLM with smart city data, the assistant provides real-time, meaningful responses to user queries, promotes sustainability, and supports data-driven governance.

The project successfully showcases a prototype where AI bridges the gap between complex city infrastructure and user-friendly access, empowering both residents and administrators to make smarter, greener decisions. Its flexible, scalable design makes it a strong foundation for further development and real-world deployment.

10. FUTURE SCOPE

1. **Web & Mobile App Deployment:**
   Extend the project from Colab to a fully responsive web or mobile application using MERN stack or Flutter.
2. **Voice Assistant Integration:**
   Integrate voice recognition and speech synthesis for hands-free interaction, enhancing accessibility.

3. **IoT Sensor Integration:**
   Connect with live IoT sensor data (e.g., pollution, traffic, water usage) for real-time responses and visual dashboards.
4. **Regional Language Support:**
   Fine-tune the assistant with IBM Granite to support Indian regional languages, making it more inclusive.
5. **User Profile & History Management:**
   Implement user login and data persistence to personalize suggestions and track sustainability impact.
6. **GIS & Smart Infrastructure Mapping:**
   Integrate with GIS systems to provide location-specific services like nearby recycling centers, public transport, etc.
7. **Predictive Analytics:**
   Use historical data and machine learning to predict future trends in energy consumption, traffic, or waste generation.
8. **Collaboration with Municipal Bodies:**
   Partner with local government for pilot testing and real-world adoption in smart city initiatives.

11. APPENDIX

Video link:

https://drive.google.com/file/d/1U31Nko_ZpEy49lcxRGkheBL1OcUiFmnc/view?usp=drivesdk

GitHub link:

https://github.com/Srlikhith/Smart-City-assistant-.git