

## Project Design Phase-II

### Technology Stack (Architecture & Stack)

Date	17 June 2025
Team ID	LTVIP2025TMID31087
Project Name	Sustainable Smart City Assistant Using IBM Granite LLM
Maximum Marks	4 Marks

### Technical Architecture Description

The architecture of the **Sustainable Smart City Assistant** is designed using a modular and scalable cloud-based approach. It uses IBM's Watsonx Granite LLM as the core language model for intelligent response generation and content summarization. The application consists of multiple functional components integrated through a user-friendly web interface powered by Gradio and deployed on Google Colab for easy testing and demonstration.

**Table 1: Components & Technologies**

S.No Component		Description	Technology
1	User Interface	Frontend where user interacts with the assistant	Gradio (Python-based UI)
2	Application Logic-1	Logic to process user inputs, handle file uploads, etc.	Python
3	Application Logic-2	Natural Language Processing and Text Summarization	IBM Watsonx Granite LLM
4	Application Logic-3	Sustainability Q&A, Eco Tips Generator, Anomaly Detection	IBM Watsonx Granite LLM
5	Database	Structured data storage (e.g., feedback submissions)	Pandas DataFrames + Excel (OpenPyXL)
6	Cloud Database	Temporarily handled using local files in Colab (for prototyping)	Local FileSystem (via Google Colab)
7	File Storage	Upload & process CSV for forecasting/anomaly detection	Google Colab Local Filesystem

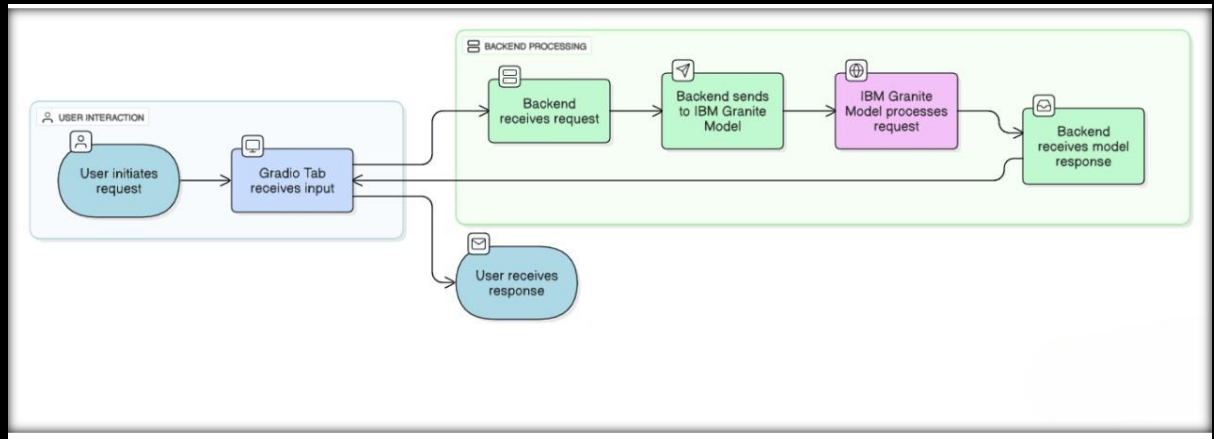
8	External API-1	PyNgrok used for sharing app via public URL	PyNgrok API
9	External API-2	Not used in current version	N/A
10	Machine Learning Model	Used for KPI forecasting and anomaly detection	Linear Regression (Scikitlearn), Statistical Z-Score
11	Infrastructure	Deployment & hosting	Google Colab (Jupyter Notebook Environment)

**Table 2: Application Characteristics**

S.No	Characteristics	Description	Technology Used
1	Open-Source Frameworks	Open-source libraries for model deployment, UI, and data handling	Gradio, Transformers, Scikit-learn, Pandas, PyNgrok
2	Security Implementations	Data handled in-memory; no sensitive information stored	File restrictions, no external access enabled
3	Scalable Architecture	Modular component-based architecture; each function can be containerized later	Python Modules, Gradio Tabs (can be scaled via FastAPI)
4	Availability	Accessible via Ngrok tunneling in Colab; scalable to any cloud in future	Google Colab + PyNgrok
5	Performance	Lightweight interface; fast response from IBM Granite LLM (under 3 sec per call)	Transformers Library + IBM Granite Model

**Solution Architecture diagram**





### Application flow:

