Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	31 January 3035
Team ID	LTVIP2025TMID29266
Project Name	Sustainable Smart City Assistant using IBM Granite LLM
Maximum Marks	4 Marks

Technical Architecture:

- Web interface (Gradio) allows citizen and admin interaction.
- Backend built in **Python**, hosted on **IBM Cloud Foundry**.
- Machine learning forecasts and feedback are handled using **IBM Granite LLM**.
- Weather and sensor data collected via **external APIs**.

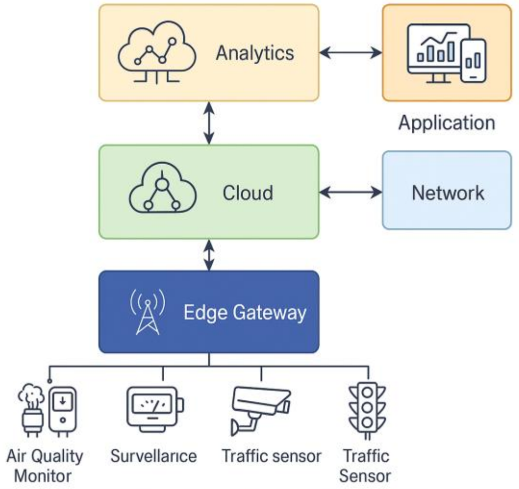

Figure 1: Architecture and Data Flow of the Smart City Application

Guidelines:

- Include all the processes (as application logic / technology blocks)
- Provide infrastructural demarcation (Local / Cloud)
- Indicate external interfaces (third-party APIs, etc.)
- Indicate data storage components / services
- Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	**Component**	**Description**	**Technology**
1.	User Interface	Web interface for citizens and admin via Gradio	Gradio, HTML, CSS
2.	Application Logic-1	Data Preprocessing and Feature Engineering	Python
3.	Application Logic-2	Model Inference using Granite LLM	IBM Granite LLM, Flask
4.	Application Logic-3	Citizen Feedback & Eco Tips generation logic	Python, Rule-Based NLP
5.	Database	Store KPI inputs, feedback, and logs	MongoDB (NoSQL)
6.	Cloud Database	Scalable and secure cloud database	IBM Cloudant
7.	File Storage	Store data, models, and config files	IBM Cloud Object Storage
8.	External API-1	Weather & Environmental data	IBM Weather API
9.	External API-2	Real-time Smart City Sensor integration	Indian Govt Smart City API
10.	Machine Learning Model	KPI Forecasting & Feedback NLP	IBM Granite LLM
11.	Infrastructure	App hosting and containerization	IBM Cloud Foundry, Docker

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Libraries & frameworks used for development	Flask, Pandas, Scikit-learn, Gradio
2.	Security Implementations	Authentication, Encryption, and IAM	OAuth 2.0, HTTPS, IBM IAM, SHA-256
3.	Scalable Architecture	Microservices and container-based deployment	Docker, Kubernetes (optional), REST APIs
4.	Availability	High uptime using IBM Cloud infra	Load Balancer, Distributed Cloud Services
5.	Performance	Fast response, caching, async I/O	Redis (optional), CDN, IBM Cloud Edge