**📘 Smart City Project - Technology Stack & Architecture**

**📅 Date: 31 January 3035**

**🧑‍💻 Team ID: LTVIP2025TMID37298**

**🏙️ Project Name: Sustainable Smart City Assistant Using IBM Granite LLM**

**🏆 Maximum Marks: 4 Marks**

**🔧 Technical Architecture Overview:**

You are expected to create a **technical architecture diagram** and fill in the required information in **Table-1 (Components & Technologies)** and **Table-2 (Application Characteristics)**.

**🧩 Table-1: Components & Technologies for Smart City Assistant**

| **S.No** | **Component** | **Description** | **Technology / Tools Used** |
| --- | --- | --- | --- |
| 1 | User Interface | User interacts with app via chatbot/web | Streamlit (Web UI), HTML, CSS |
| 2 | Application Logic-1 | Smart query processing | Python (FastAPI framework) |
| 3 | Application Logic-2 | Speech-to-text for voice commands | IBM Watson Speech to Text |
| 4 | Application Logic-3 | AI Assistant for responses and interaction | IBM Watson Assistant / Granite LLM |
| 5 | Database | Storage of user queries and analytics | Pinecone (Vector DB), MongoDB (for logging) |
| 6 | Cloud Database | Scalable DB service | IBM Cloudant / IBM DB2 |
| 7 | File Storage | File uploads, user logs | IBM Block Storage / Local FileSystem |
| 8 | External API-1 | Weather integration | IBM Weather API |
| 9 | External API-2 | Identity verification | Aadhar API or DigiLocker |
| 10 | Machine Learning Model | Understanding intent / context | Custom-trained LLM on urban datasets |
| 11 | Infrastructure | Deployment environment | IBM Cloud, Cloud Foundry, or Kubernetes |

**📊 Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology / Strategy Used** |
| --- | --- | --- | --- |
| 1 | Open-Source Frameworks | Tools used to reduce cost and ease development | Streamlit, FastAPI, Python, Pinecone, Docker |
| 2 | Security Implementations | Security for APIs, data access, and infrastructure | JWT Tokens, SHA-256 encryption, IBM IAM Controls |
| 3 | Scalable Architecture | Flexible design to handle more users & data | Microservices + Event-Driven Architecture |
| 4 | Availability | Ensure 24/7 access and fault-tolerance | Load Balancers, Redundant Servers, IBM Cloud Services |
| 5 | Performance | Response time, request handling, real-time data fetching | Caching with Redis, CDN, optimized LLM usage |

**📚 Reference Architecture Resources:**

* [IBM AI Patterns for Pandemic Response](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)
* [IBM Cloud Architecture Center](https://www.ibm.com/cloud/architecture)
* [C4 Model for Architecture](https://c4model.com/)
* [AWS Architecture Center](https://aws.amazon.com/architecture)
* [How to Draw Technical Diagrams](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)