# Project Design Phase-II

## Technology Stack (Architecture & Stack)

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID21127 |
| Project Name | Sustainable Smart City Assistant |
| Maximum Marks | 4 Marks |

## Technical Architecture:

The deliverable includes the architectural diagram and the following tables which explain the components and technologies used in the system, as well as its technical characteristics.

Reference: https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/

## Table-1 : Components & Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology |
| 1 | User Interface | Web interface for user interaction | Streamlit |
| 2 | Application Logic-1 | PDF upload, parsing, summarization | Python, PyPDF2, FLAN-T5 |
| 3 | Application Logic-2 | Semantic embedding and vector search | sentence-transformers, FAISS / Pinecone |
| 4 | Application Logic-3 | Chatbot for sustainability queries | Prompt engineering with Hugging Face |
| 5 | Database | Storage of user reports, tips | Local JSON/CSV or SQLite |
| 6 | Cloud Database | Semantic vector storage | Pinecone |
| 7 | File Storage | Uploaded PDFs and KPI data | Local filesystem |
| 8 | External API-1 | Optional: Weather or policy APIs | OpenWeatherMap (planned) |
| 9 | External API-2 | Optional: Civic info APIs | GovData APIs (future) |
| 10 | Machine Learning Model | Forecasting and anomaly detection | scikit-learn (Linear Regression, Isolation Forest) |
| 11 | Infrastructure | Local app deployment | Localhost / can be scaled to cloud |

## Table-2 : Application Characteristics

|  |  |  |
| --- | --- | --- |
| S.No | Characteristics | Description & Technology |
| 1 | Open-Source Frameworks | Streamlit, Hugging Face Transformers, FAISS, scikit-learn |
| 2 | Security Implementations | File type validation, API key security (future extension), local-only access |
| 3 | Scalable Architecture | Modular design with independent AI, DB, and frontend layers |
| 4 | Availability | Easily deployable on cloud or multiple servers if needed |
| 5 | Performance | Fast API calls, embeddings caching possible, optimized search using FAISS/Pinecone |

## References:

https://c4model.com/  
https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/  
https://www.ibm.com/cloud/architecture  
https://aws.amazon.com/architecture  
https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d