**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | LTVIP2025TMID31802 |
| Project Name | Citizen AI |
| Maximum Marks | 4 Marks |

**Technical Architecture Overview**

Citizen AI is designed using a modular, scalable architecture integrating AI-driven natural language processing, sentiment analysis, and real-time data visualization. It utilizes both cloud-based APIs and open-source tools to deliver a robust and responsive citizen engagement platform

**Table-1: Components & Technologies**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Component Description** | **Technology** |
| 1 | User Interface | Gradio (Python-based Web UI) |
| 2 | Application Logic-1 | Python scripts for data processing and interaction |
| 3 | Application Logic-2 | Hugging Face Sentiment Analysis Pipeline |
| 4 | Application Logic-3 | IBM Granite model for conversational response |
| 5 | Database | Pandas DataFrame (in-memory), CSV/JSON files |
| 6 | Cloud Database | Not applicable (local storage used) |
| 7 | File Storage | Local filesystem for logs (CSV/JSON) |
| 8 | External API-1 | IBM Granite Inference API |
| 9 | External API-2 | Hugging Face Model Hub |
| 10 | Machine Learning Model | Granite 3.3b Instruct, Sentiment Classifier |
| 11 | Infrastructure (Server / Cloud) | Localhost via Gradio / Streamlit-compatible deployment |

**Table-2: Application Characteristics**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Characteristics Description** | **Technology** |
| 1 | Open-Source Frameworks | Gradio, Transformers, Matplotlib, Pandas |
| 2 | Security Implementations | API Token Authentication, Token Management in Environment Variables |
| 3 | Scalable Architecture | Modular component-based design; APIs can be containerized for scaling |
| 4 | Availability | Gradio can be deployed on scalable servers or behind load balancers |
| 5 | Performance | Fast local inference with GPU/CPU selection, lightweight architecture, minimal latency |