**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 25 JUNE 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID31965 |
| Project Name | Citizen AI – Intelligent Citizen Engagement Platform |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**Example: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)

****

Guidelines:

Include all the processes (As an application logic / Technology Block)

Provide infrastructural demarcation (Local / Cloud)

Indicate external interfaces (third party API’s etc.)

Indicate Data Storage components / services

Indicate interface to machine learning models (if applicable)

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| **1** | User Interface | Web UI for interacting with all features | Streamlit, streamlit-lottie |
| **2** | Application Logic-1 | Conversational AI assistant for civic queries | Python, transformers, IBM Granite |
| **3** | Application Logic-2 | Sentiment analysis from user inputs | Python, TextBlob |
| **4** | Application Logic-3 | Model fallback and error handling logic | Python, Groq API, conditional model routing |
| **5** | Database | Runtime session and feedback logging | JSON (file-based local storage) |
| **6** | Cloud Database | Not currently used | NA |
| **7** | File Storage | Storage of feedback and interactions | Local filesystem (JSON) via Python open() API |
| **8** | External API-1 | Fallback API for inference when IBM is unavailable | Groq API |
| **9** | External API-2 | Model access authentication | Hugging Face API, dotenv |
| **10** | Machine Learning Model | Model used for answering queries and generating responses | IBM Granite 3.3 Instruct, Hugging Face Transformers |
| **11** | Infrastructure (Server) | Deployment on Render/local with GPU support or CPU fallback | Flask, Uvicorn (if migrated), torch.cuda, Streamlit Cloud-ready |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks and libraries used | Streamlit, Flask, Transformers, TextBlob |
|  | Security Implementations | Token management via .env; external access restricted via backend | python-dotenv, Hugging Face API key, Groq API key |
|  | Scalable Architecture | Modular design with model fallback and lightweight frontend/backend separation | Flask (API), Streamlit (UI), fallback to Groq API |
|  | Availability | High availability ensured through model retry logic and dual-model setup | Groq API fallback, try/except with response validation |
|  | Performance | GPU detection, fast token generation, async fallback when required | torch.cuda.is\_available(), conditional routing, Flask app |

### Architecture Summary

* **Frontend**: Streamlit
* **Backend:** Flask (lightweight REST API server)
* **Core AI Engine:** Hugging Face LLMs (e.g., IBM Granite 3.3 – 2B Instruct)
* **Fallback Engine:** Groq API (LLaMA 3 model for response backup)
* **Deployment:** Local GPU machine (Render-supported); optionally cloud deployable
* **File Handling:** Local file system for session-based feedback storage (feedback.json)

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)