**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

Date: 21 January 2025

Team ID: LTVIP2025TMID38479

Project Name: Citizen AI – Intelligent Citizen Engagement Platform

Maximum Marks: 4 Marks

**Technical Architecture**

The architecture of Citizen AI integrates AI-powered quiz generation, role-based user experience, and visual analytics. The system includes a Streamlit frontend, Python backend, IBM Granite language model, and local JSON storage. Plotly is used for visualization, and future versions aim to migrate to cloud infrastructure for scalability.

**Table-1: Components & Technologies**

|  |  |  |
| --- | --- | --- |
| S.No | Component Description | Technology |
| 1 | User Interface – Web application with role-based tabs | Streamlit |
| 2 | Query Submission & Management Logic | Python, IBM Granite LLM |
| 3 | Workflow Management & Response System | Python, Streamlit |
| 4 | Authentication and Role Management | Streamlit session state |
| 5 | Local Storage / Database | JSON file-based storage |
| 6 | Data Visualization | Plotly, Pandas |
| 7 | AI Model Integration | IBM Granite, HuggingFace Transformers |
| 8 | Future Cloud Hosting | IBM Cloud |

**Table-2: Application Characteristics**

|  |  |  |
| --- | --- | --- |
| S.No | Characteristic Description | Technology |
| 1 | Open-Source Frameworks – Technologies used are freely available and community supported | Streamlit, Python, HuggingFace Transformers |
| 2 | Security – User role authentication and file isolation | Streamlit session, local file permissions |
| 3 | Scalable Architecture – Designed for future migration to microservices/cloud | Modular Python structure, IBM Cloud (future) |
| 4 | Availability – Local availability, with potential for 24/7 uptime on cloud | Cloud Foundry (future) |
| 5 | Performance – Optimized model calls and lightweight file I/O | Efficient API calls, caching in memory,lightweight file I/O |