**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 25 June 2025 |
| Team ID | LTVIP2025TMID30080 |
| Project Name | HealthAI: Intelligent Healthcare Assistant Using IBM Granite |
| Maximum Marks | 4 Marks |

## Technical Architecture:

HealthAI is an intelligent assistant for preliminary health assessments. Users input symptoms through a web-based interface. The data flows to a Flask backend which preprocesses the input and forwards it to a machine learning model (IBM Granite) hosted via API. The results are returned and displayed back to the user in a clean, informative UI. Data storage is handled locally or via IBM Cloudant for user interaction history.

## Table-1: Components & Technologies

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology Used |
| 1 | User Interface | Interface for users to input symptoms and view results | HTML, CSS, JavaScript |
| 2 | Application Logic-1 | Flask API backend to process input and return predictions | Python, Flask |
| 3 | Application Logic-2 | Connection to AI model API for prediction | IBM Granite Model API |
| 4 | Application Logic-3 | Simple routing and result formatting logic | Python Functions, Flask Routes |
| 5 | Database | Stores user inputs and results history | SQLite (for local), IBM Cloudant (optional) |
| 6 | Cloud Database | Optional cloud storage for scalability | IBM Cloudant |
| 7 | File Storage | Static files (model files, logs, data CSVs if used) | Local File System, IBM Block Storage |
| 8 | External API-1 | Optional API for health-related external validation (future scope) | OpenAI API / Symptom Checker APIs |
| 9 | External API-2 | Optional demographic info or location (future scope) | IBM Weather API, optional health records APIs |
| 10 | Machine Learning Model | Predicts disease risk or health feedback | IBM Granite-3.3-2B-Instruct |
| 11 | Infrastructure | Deployment of app and backend | Local Dev Server, IBM Cloud, Cloud Foundry |

## Table-2: Application Characteristics

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology Used |
| 1 | Open-Source Frameworks | Backend and frontend developed using open-source tools | Flask, HTML/CSS, Scikit-learn, Pandas |
| 2 | Security Implementations | Secure API calls, input validation, basic token/auth (optional) | Input sanitization, HTTPS, IAM (cloud-based) |
| 3 | Scalable Architecture | Frontend + Flask API + External ML Model makes the system modular and scalable | 3-Tier (UI → API → ML Model) |
| 4 | Availability | Can be deployed on IBM Cloud with high availability options | IBM Cloud Load Balancers, Uptime SLAs |
| 5 | Performance | Uses lightweight Flask server with optimized model; can handle moderate traffic with caching | Flask + JSON Caching, SQLite or Cloudant |