**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 3035 |
| Team ID | LTVIP2025TMID60012 |
| Project Name | HealthAI: Intelligent Healthcare Assistant Using IBM Granite |
| 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** |
|  | User Interface | The front-end where users interact with HealthAI. They can chat, input symptoms, view treatment plans, and see health charts. | **Streamlit** (Python-based web framework) |
|  | Application Logic-1 | Receives and identifies the type of user input: a question, symptoms, health data, or a diagnosis. | 🔹 Python  🔹 Streamlit |
|  | Application Logic-2 | Based on input, it processes the request: sends to ML model, runs logic, or handles analytics. | 🔹 Python  🔹 IBM Watson API  🔹 Pandas |
|  | Application Logic-3 | Formats the result (text, chart, treatment) and sends it back to the UI for display. | 🔹 Python  🔹 Streamlit  🔹 Plotly / Matplotlib |
|  | Database | Stores user profiles, health history, symptom logs, and treatment plans. | 🔹 PostgreSQL / MongoDB / IBM Cloudant |
|  | Cloud Database | A managed database in the cloud that provides scalability and security. | 🔹 IBM Cloudant (NoSQL) or AWS RDS (SQL) |
|  | File Storage | For storing files like lab reports or uploaded health documents, if needed. | 🔹 IBM Cloud Object Storage  🔹 AWS S3 |
|  | External API-1 | Handles natural language understanding and response generation. | 🔹 IBM Watson Machine Learning  🔹 Granite-13b-instruct-v2 |
|  | External API-2 | Connects with external services to pull health data or validate symptoms. | 🔹 Fitbit/Apple Health API (optional)  🔹 Healthline or Infermedica API |
|  | Machine Learning Model | Processes inputs for predictions, chat answers, treatment suggestions. | 🔹 IBM Watson – Granite-13b-instruct-v2 (hosted via API) |
|  | Infrastructure (Server / Cloud) | Hosts the entire application, handles deployments, ensures uptime. | 🔹 IBM Cloud  🔹 Streamlit Cloud  🔹 AWS / Azure (optional) |

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | HealthAI uses free, community-supported tools for faster, cost-effective development. | 🔹 Streamlit (UI) 🔹 Python (Backend) 🔹 Pandas, Plotly (Data & Charts) |
|  | Security Implementations | Ensures user data is safe, private, and only accessible by authorized users. | 🔹 JWT (Login security) 🔹 HTTPS (Safe connection) 🔹 API Keys (Protects services) 🔹 Data Encryption |
|  | Scalable Architecture | The app can grow and handle more users and data. | 🔹 IBM Cloud / AWS (Cloud hosting) 🔹 Cloudant / MongoDB (Database) 🔹 Modular Python code |
|  | Availability | The app is always online and ready to use. | 🔹 IBM Cloud / AWS 🔹 Auto-restart tools 🔹 Uptime monitoring |
|  | Performance | The app is fast and smooth to use. | 🔹 Optimized Python code 🔹 Fast API responses 🔹 Lightweight UI (Streamlit) |

**Table-2: Application Characteristics:**

**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)