**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID60665 |
| Project Name | Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

**Table 1: Project Component Details**

| **S.No** | **Component** | **Description** | **Technology Used** |
| --- | --- | --- | --- |
| **1** | **User Interface** | **How users interact with the web application for predictions** | **HTML, CSS, JavaScript, Bootstrap** |
| **2** | **Application Logic-1** | **Backend logic for processing inputs and returning predictions** | **Python, Flask** |
| **3** | **Application Logic-2** | **Converts user voice inputs to text (Optional speech-to-text feature)** | **IBM Watson STT Service** |
| **4** | **Application Logic-3** | **Intelligent chatbot for guiding users or FAQs** | **IBM Watson Assistant** |
| **5** | **Database** | **Stores input data, patient records, logs, etc.** | **SQLite (for local) / PostgreSQL (for cloud)** |
| **6** | **Cloud Database** | **Cloud-hosted database for production environment** | **IBM DB2, IBM Cloudant (alternative: Render PostgreSQL)** |
| **7** | **File Storage** | **Model files, report files, medical records** | **IBM Block Storage / Local Filesystem / GridFS** |
| **8** | **External API-1** | **Used for displaying local weather conditions (if used in UI)** | **IBM Weather API** |
| **9** | **External API-2** | **For user identity verification (if used in advanced versions)** | **Aadhar Verification API (UIDAI)** |
| **10** | **Machine Learning Model** | **Predicts whether liver cirrhosis is present based on clinical features** | **Random Forest / XGBoost (Trained in Python)** |
| **11** | **Infrastructure (Server / Cloud)** | **Web app hosting and deployment** | **Localhost (Dev), Render / IBM Cloud (Prod), Cloud Foundry** |

**Table 2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology / Notes** |
| --- | --- | --- | --- |
| **1** | **Open-Source Frameworks** | **Frameworks used in the development of UI, backend, ML, etc.** | **Flask, Scikit-learn, Pandas, Bootstrap, Jinja2** |
| **2** | **Security Implementations** | **Implements security at API and server levels** | **SHA-256, HTTPS, JWT Authentication (optional), IAM Roles, Firewalls** |
| **3** | **Scalable Architecture** | **Architecture allows easy scaling of services or load balancing** | **3-Tier Architecture (UI – Logic – Database), Docker, Microservices (if extended)** |
| **4** | **Availability** | **Ensures the app is always available through redundancy/load balancing** | **Cloud hosting (Render/IBM Cloud), Load Balancers, Auto Restart** |
| **5** | **Performance** | **Designed to handle real-time predictions and fast response** | **Caching (Redis - optional), CDN (for UI assets), Optimized ML Models, <100ms latency for prediction** |