**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 2025 |
| Team ID | LTVIP2025TMID33878 |
| Project Name | Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

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



**Table-1 : Components & Technologies:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web-based interface for input, prediction, and report visualization | HTML, CSS, JavaScript / Angular Js / React Js etc. |
|  | Application Logic-1 | Handles form input, validation, and client-server communication | JavaScript, React.js |
|  | Application Logic-2 | Backend logic for data preprocessing, feature engineering | Python (Pandas, NumPy) |
|  | Application Logic-3 | Machine Learning inference logic | Python (Scikit-learn / XGBoost / TensorFlow) |
|  | Database | Stores patient data and prediction logs | MySQL, NoSQL, etc. |
|  | Cloud Database | Optional cloud-hosted database | Firebase Firestore / AWS RDS |
|  | File Storage | Uploads & stores patient lab report files | AWS S3 / Google Cloud Storage / Local FS |
|  | External API-1 | Secure OTP/email confirmation system (if user registration is needed) | Firebase Auth / SendGrid / Twilio |
|  | External API-2 | Medical recommendations or drug interaction services | Health APIs (e.g., Infermedica API) |
|  | Machine Learning Model | Predicts liver cirrhosis and classifies risk levels | Trained ML model in Scikit-learn/TensorFlow |
|  | Infrastructure (Server / Cloud) | Application hosted on cloud or local system | AWS EC2 / Heroku / Render / Local Server |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | React.js (Frontend), Flask / FastAPI (Backend), Scikit-learn, TensorFlow | Technology of Opensource framework |
|  | Security Implementations | End-to-end encryption, HTTPS, access control, authentication | JWT, HTTPS, SHA-256, OAuth |
|  | Scalable Architecture | Designed using modular MVC pattern, easy to scale as microservices | 3-tier architecture, Docker (optional) |
|  | Availability | High availability ensured through cloud deployment and failover mechanisms | AWS/GCP Load Balancer, Auto-Scaling Groups |
|  | Performance | Fast model prediction (<3s), caching recent predictions for efficiency | Redis, CDN, API throttling |