Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	21 June 2025
Team ID	LTVIP2025TMID41434
Project Name	Revolutionizing Liver Care : Predicting Liver Cirrhosis using Advanced Machine Learning Techniques
Maximum Marks	4 Marks

[Frontend Layer]

Web UI / Mobile App

⬇️

[Application Logic Layer]

User Authentication, Data Validation, ML Inference

⬇️

[Backend Layer]

Flask API, ML Model, Database

⬇️

[Infrastructure Layer]

Local Server or Cloud

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Web-based interface for user input and result viewing	HTML, CSS, JavaScript
2	Application Logic-1	Backend logic for data preprocessing and routing	Python (Flask)
3	Application Logic-2	Machine learning model execution logic	Scikit-learn, NumPy, Pandas
4	Application Logic-3	Optional AI service for symptom check/chatbot	
5	Database	Stores user inputs, prediction results	SQLite / MySQL
6	Cloud Database	For scalable production deployments	IBM Cloudant / Firebase Realtime DB
7	File Storage	For model files, logs, and user documents	
8	External API-1	API for country-wise liver disease statistics (optional)	WHO Disease Statistics API (optional)
9	External API-2	Authentication (e.g., Aadhaar / Email verification API)	Email Verification API / Aadhaar API

10	Machine Learning Model	Predicts cirrhosis likelihood from clinical data	Random Forest / XGBoost Model
11	Infrastructure	Deployment of Flask app	Local (Flask server) / IBM Cloud Foundry

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Frameworks used for backend/frontend	Flask, Scikit-learn, Bootstrap, Pandas
2	Security Implementations	Data encryption and access control	SHA-256 hashing, HTTPS, JWT Auth, OAuth (if any)
3	Scalable Architecture	3-tier architecture allows independent scaling of layers	Flask (modular), DB separation, Docker optional
4	Availability	Can ensure uptime using cloud deployment and backups	
5	Performance	Fast inference, optimized ML model, input validation, caching	Flask + Gunicorn, Memcached (if scaling), CDN