

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

Date	27 JUNE 2025
Team ID	TVP2025TMD42853
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

Our solution is a Flask-based Machine Learning Web App to predict liver cirrhosis using real medical data. The architecture includes backend processing, frontend UI, storage, and local deployment with scalable options.

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

<b>S.No</b>	<b>Component</b>	<b>Description</b>	<b>Technology</b>
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Handles ML prediction & input processing	Python (Flask)
3.	Application Logic-2	Preprocessing, encoding, and normalization	Python (Sklearn, Pandas)
4.	Application Logic-3	Patient input form handling, validation	HTML + Flask
5.	Database	Patient data loaded from CSV during training.	CSV (Locally), Pandas DataFrames
6.	Cloud Database	Optional upgrade for cloud storage	Firebase / MongoDB Atlas (optional)
7.	File Storage	Store model files, logs	Local File System
8.	External API-1	[Optional] to fetch live blood test norms	RapidAPI or medical APIs (future)

9.	External API-2	[Optional] authentication or login APIs	Aadhaar API / OAuth (future)
10.	Machine Learning Model	Liver cirrhosis prediction from 34 features	RandomForestClassifier (sklearn)
11.	Infrastructure (Server / Cloud)	App deployed locally or via cloud	Localhost / PythonAnywhere / Heroku

**Table 2 : Application Characteristics**

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Flask, Pandas, Scikit-learn, Bootstrap	Python, Flask, Sklearn
2.	Security Implementations	Local access control, form validation.	Flask, CSRF token (optional)
3.	Scalable Architecture	3-tier (Frontend, Backend, Model Layer), upgradeable to microservices	Flask, optional Docker/Kubernetes
4.	Availability	Deployable to platforms with load balancing support	PythonAnywhere, Heroku, AWS EC2

S.No	Characteristics	Description	Technology
5.	Performance	Lightweight model (RandomForest), fast response time, <1s prediction delay	Local Flask server, Scikit-learn