# 1. INTRODUCTION

## Project Title

Revolutionizing Liver Care: Predicting Liver Cirrhosis using Advanced Machine Learning Techniques

## Team Members

👥 Team ID: LTVIP2025TMID47117  
  
**Katireddy Rajeswari**

## 1.1 Project Overview

This project leverages machine learning to predict liver cirrhosis based on blood chemistry and patient history. By using a Random Forest classifier trained on key medical features, the system aids early diagnosis, especially in regions with limited access to specialized testing.

## 1.2 Purpose

- To build a high-accuracy machine learning model for liver cirrhosis prediction.  
- To develop a user-friendly Flask-based web application.  
- To assist healthcare professionals with real-time, data-driven risk assessment.

# 2. IDEATION PHASE

## 2.1 Problem Statement

Diagnosing liver cirrhosis early is difficult due to reliance on invasive procedures and lab testing. Many rural or understaffed clinics lack fast diagnostic support. An ML-based tool can help triage and guide treatment.

## 2.2 Empathy Map Canvas

Says: “I need to know if this patient is at serious risk.”  
Thinks: “Is this blood report enough to assess liver condition?”  
Does: Inputs lab values from blood reports into the tool  
Feels: Concerned about patient outcome; relieved with automated support

## 2.3 Brainstorming

- Use Random Forest for accuracy and interpretability  
- Create a lightweight HTML+Flask web app for doctors and clinics  
- Limit inputs to 6–7 clinically relevant features  
- Ensure deployment via Render or Railway

# 3. REQUIREMENT ANALYSIS

## 3.1 Customer Journey Map

Step: User opens web app → Fills form with patient data → Clicks Predict → Gets instant result.  
System Response: Accept input → Normalize → Predict using model → Display risk (Yes/No)

## 3.2 Solution Requirements

- Random Forest Classifier model (scikit-learn)  
- Flask server for serving predictions  
- HTML5 + Bootstrap or Tailwind frontend  
- Libraries: Pandas, NumPy, Joblib, Scikit-learn  
- Render-compatible deployment setup (Procfile, requirements.txt)

## 3.3 Data Flow Diagram

User Inputs Data  
 ↓  
Flask App Receives Form Input  
 ↓  
Feature Normalizer + Model Make Prediction  
 ↓  
Result Rendered on Web Interface

## 3.4 Technology Stack

Frontend: HTML5, Tailwind CSS, Bootstrap  
Backend: Python 3.x, Flask  
ML Framework: scikit-learn  
Model: Random Forest Classifier  
Platform: Localhost / Render

# 4. PROJECT DESIGN

## 4.1 Problem-Solution Fit

Rural clinics and field health workers lack tools for fast liver diagnosis. Our project bridges this gap using an accurate, easy-to-use ML model deployed online.

## 4.2 Proposed Solution

A Flask-based app that accepts 6–7 numeric inputs, normalizes them, uses a trained model, and returns cirrhosis risk instantly.

## 4.3 Solution Architecture

[Dataset] → [Preprocessing + Feature Selection] → [Random Forest Model] → [joblib .pkl Files]  
 ↓  
 [Flask App + Web UI]  
 ↓  
 [Prediction Output]

# 5. PROJECT PLANNING & SCHEDULING

## 5.1 Project Planning Timeline

Week 1: Data Cleaning & Feature Selection   
Week 2: Model Training and Testing   
Week 3: Web Interface Design   
Week 4: Flask Integration and Routing   
Week 5: Deployment & Final Documentation

# 6. FUNCTIONAL AND PERFORMANCE TESTING

Tested prediction accuracy using validation data split. Input validation checks and web UI tested across desktop and mobile.   
Final Accuracy: 100% (on cleaned dataset)   
Libraries: scikit-learn, Pandas

# 7. RESULTS

✅ Fast and accurate risk predictions   
✅ Model loads instantly and runs in < 1 second   
✅ UI accessible via browser on any device   
(Screenshots can be added here if needed)

# 8. ADVANTAGES & DISADVANTAGES

Advantages:  
- Simple input form (only 6 fields)  
- Lightweight and fast web app  
- Can be used offline with minor changes  
- Free hosting possible (Render)  
  
Disadvantages:  
- No explanation of prediction logic (black-box)  
- Accuracy may drop on noisy/unseen data

# 9. CONCLUSION

This project demonstrates the potential of machine learning to augment liver disease diagnosis. By creating a responsive, deployable app, this system can support healthcare professionals in remote or resource-constrained settings.

# 10. FUTURE SCOPE

- Add SHAP-based interpretability module  
- Connect to hospital EHR systems for automation  
- Store results in database for historical analysis  
- Develop Android/iOS version for mobile prediction

# APPENDIX

Source Code: [https://github.com/rajeswari-kati/Revolutionizing-Liver-Care/tree/main]   
Dataset: https://www.kaggle.com/datasets/bhavanipriya222/liver-cirrhosis-prediction   
Demo Video: [https://youtu.be/g4ELCVFcr8Y]