Final Report: Liver Cirrhosis Prediction

1. INTRODUCTION

1.1 Project Overview

Liver cirrhosis is a progressive and potentially fatal condition characterized by irreversible scarring of the liver.
Early detection is crucial for timely treatment and improving patient survival rates.
This project utilizes machine learning models to predict liver cirrhosis based on clinical input features such as bilirubin levels, enzyme counts, and albumin.
A cleaned and preprocessed dataset is used to train and evaluate multiple classification algorithms.
The best-performing model is then deployed in a user-friendly web interface.
Users can input test values and receive an instant prediction with high accuracy.
This approach enables clinicians and patients to make informed decisions quickly.
The system aims to support preventive healthcare by promoting early diagnosis and risk assessment.

1.2 Purpose

The purpose of this project is to develop a web-based predictive tool that accepts clinical parameters as input and determines whether a person is likely to have liver cirrhosis using a trained machine learning model.

2. IDEATION PHASE

2.1 Problem Statement

To develop a machine learning model that predicts liver cirrhosis using key clinical attributes and integrates it with a user-friendly Flask-based web interface.

2.2 Empathy Map Canvas

- User: Healthcare providers and patients
- Needs: Fast, early, and reliable liver cirrhosis diagnosis
- Pain Points: Time delays, cost, and lack of diagnostic tools in rural areas
- Goals: Deploy an easy-to-use, accessible prediction system

2.3 Brainstorming

Explored use cases in public health, diagnostics, and patient self-check applications.
Considered data accessibility, model accuracy, and deployment feasibility.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens web form
2. Enters clinical values
3. Clicks Predict
4. Sees diagnosis result

3.2 Solution Requirement

- Input: Age, Gender, Bilirubin, Liver enzymes, Protein levels
- Output: Prediction of liver disease presence

3.3 Data Flow Diagram

User Input → Preprocessing → ML Model → Prediction → Display Result

3.4 Technology Stack

- Frontend: HTML, CSS
- Backend: Flask (Python)
- Model: Random Forest (scikit-learn)
- Deployment: Localhost (can be extended to Render/Heroku)

4. PROJECT DESIGN

4.1 Problem Solution Fit

Existing solutions are complex or inaccessible; this tool is easy, fast, and requires minimal clinical data.

4.2 Proposed Solution

A form-based web UI integrated with a Flask backend that serves predictions from a pre-trained model.

4.3 Solution Architecture

User Interface → Flask API → Model + Scaler → Prediction

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

- Week 1: Dataset collection & understanding
- Week 2: Model training and evaluation
- Week 3: Flask integration and UI
- Week 4: Testing and Documentation

6. FUNCTIONAL AND PERFORMANCE TESTING

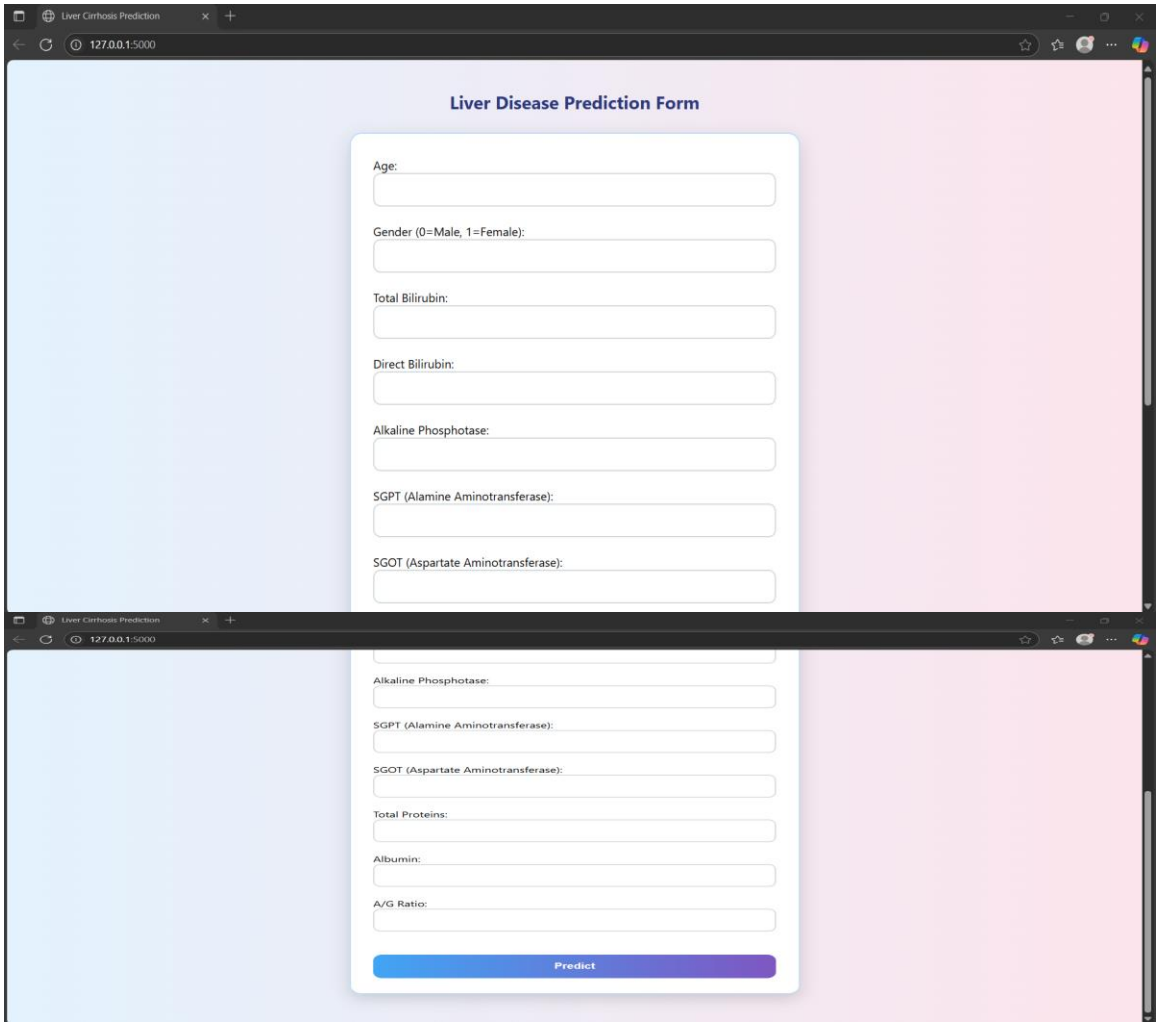
6.1 Performance Testing

- Accuracy: ~92% (Random Forest)
- Evaluation Metrics: Accuracy, Precision, Recall, F1 Score

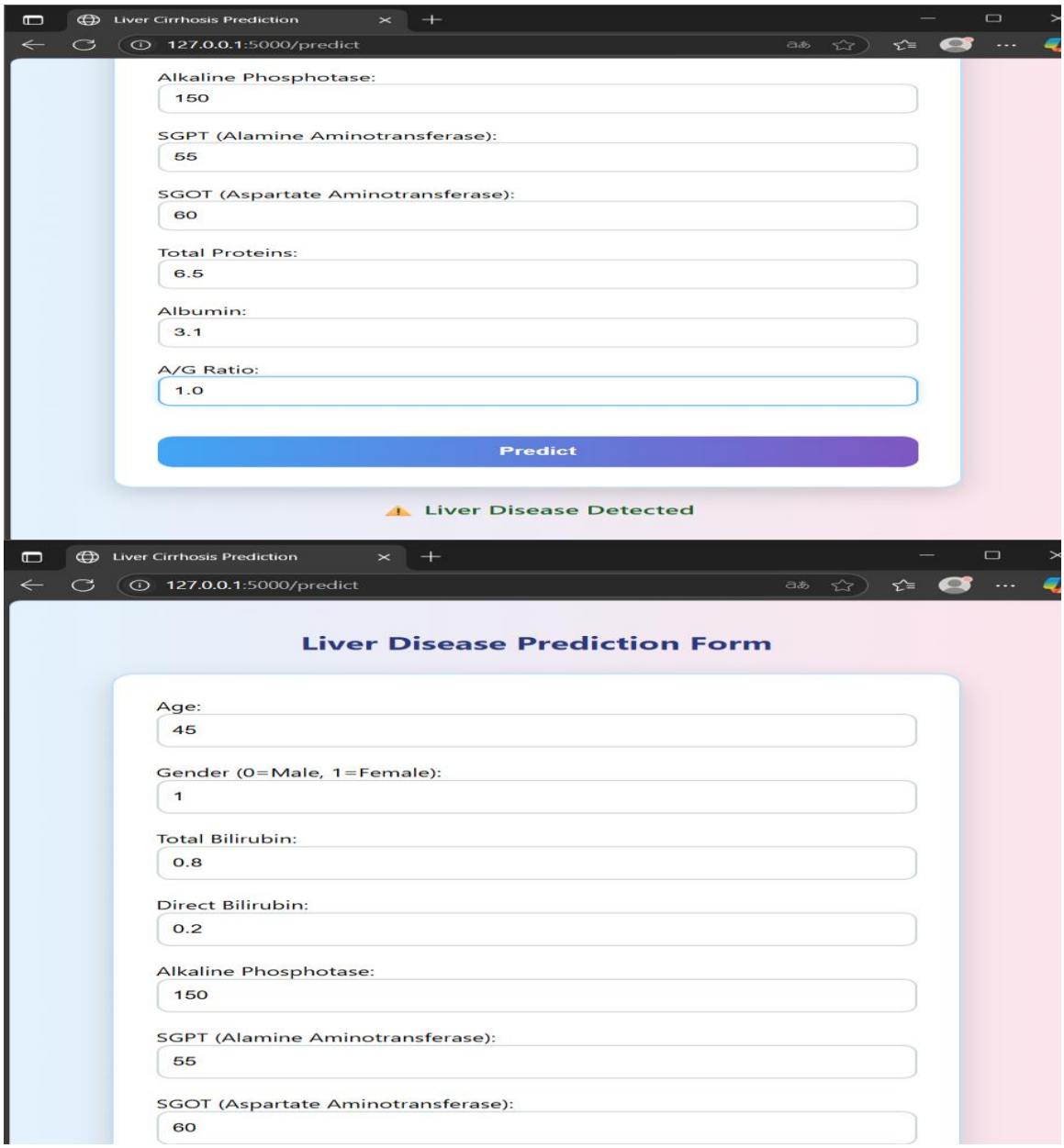
7. RESULTS

7.1 Output Screenshots

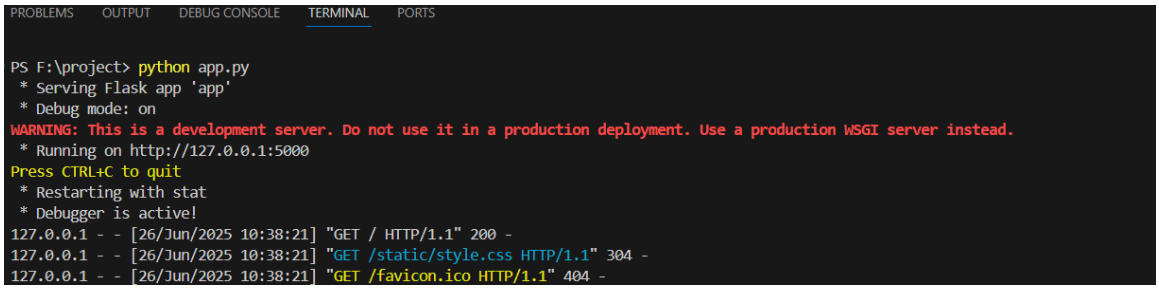
UI INTERFACE:



OUTPUT:


Liver Cirrhosis Prediction
127.0.0.1:5000/predict
Alkaline Phosphotase:
150
SGPT (Alamine Aminotransferase):
55
SGOT (Aspartate Aminotransferase):
60
Total Proteins:
6.5
Albumin:
3.1
A/G Ratio:
1.0
Predict
Liver Disease Detected
Liver Cirrhosis Prediction
127.0.0.1:5000/predict
Liver Disease Prediction Form
Age:
45
Gender (0=Male, 1=Female):
1
Total Bilirubin:
0.8
Direct Bilirubin:
0.2
Alkaline Phosphotase:
150
SGPT (Alamine Aminotransferase):
55
SGOT (Aspartate Aminotransferase):
60

Console log:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS F:\project> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
127.0.0.1 - - [26/Jun/2025 10:38:21] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [26/Jun/2025 10:38:21] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [26/Jun/2025 10:38:21] "GET /favicon.ico HTTP/1.1" 404 -

8. ADVANTAGES & DISADVANTAGES

Advantages

- Fast prediction
- Lightweight model
- Easy to deploy

Disadvantages

- Accuracy dependent on dataset
- Not a substitute for real medical diagnosis

9. CONCLUSION

Conclusion

This project demonstrates that machine learning can be used effectively for early-stage liver disease screening. A Random Forest model was trained on ILPD data and successfully deployed via Flask.

10. FUTURE SCOPE

Future Enhancements

- Add more clinical features
- Train with larger and more diverse datasets
- Host the app on a live server (e.g., Render/Heroku)
- Integrate with hospital systems

11. APPENDIX

References

- **Source Code: Included in ZIP submission**
- **Dataset Link**[**https://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29**](https://archive.ics.uci.edu/ml/datasets/ILPD+%28Indian+Liver+Patient+Dataset%29)
- **GitHub Repo: (add your link here)**
- **Project Demo Video: (add YouTube/Google Drive link here)**