Evolutionizing Liver Care - Project Documentation

# Objective

To predict liver disease risk using a machine learning model and provide a user interface through a Flask web application.

# Project Structure

The project is organized as follows:

Evolutionizing\_liver\_care/  
│  
├── dataset/  
│ └── HealthCareData.xlsx # Source dataset used for training  
│  
├── Flask/  
│ ├── app.py # Main Flask application  
│ ├── normalizer.pkl # Data normalization pickle file  
│ ├── rf\_acc\_68.pkl # Trained Random Forest model (68% accuracy)  
│ └── templates/ # HTML templates for UI  
│ ├── index.html  
│ ├── inner-page.html  
│ ├── portfolio-details.html  
│ └── result.html  
│  
└── Training/  
 └── train\_model.py # Python script for training the ML model

# Dataset

File: HealthCareData.xlsx

Description: Presumably includes health-related parameters (e.g., liver enzymes, bilirubin levels, age, etc.) for liver disease diagnosis.

# Machine Learning Model

Training Script: train\_model.py

This script likely performs the following tasks:  
- Loads and pre-processes the data  
- Trains a Random Forest classifier  
- Saves the trained model as rf\_acc\_68.pkl  
- Saves the pre-processing scaler as normalizer.pkl

# Flask Web Application

Main Application: app.py

This handles routing, user input, and prediction using the trained model.

Templates:

- index.html: Home page  
- inner-page.html: Likely a sub-page or info page  
- portfolio-details.html: Describes the model or dataset  
- result.html: Displays prediction output

# Working

1. User inputs health parameters via a web form.  
2. Flask app normalizes the inputs using normalizer.pkl.  
3. It predicts liver disease risk using the Random Forest model rf\_acc\_68.pkl.  
4. Result is shown to the user on result.html.