**AIML**

**Project Documentation format**

# 1. Introduction

### **. Project Title:** [Revolutionizing Liver Care]

# Kalyani Kamarajugadda – coding and Documentation

# Kandula Jahnavi – Demo and Coding

# Jarugumalla Rushi

# Gandham Johnson Paul

# 2. Project Overview

* **Purpose:**The **purpose of predicting liver cirrhosis** is to enable **early detection**, **timely intervention**, and **better management** of the disease.
* **Features:**

E**lectronic Health Records (EHR)**

**Blood Test Reports**

**Ultrasound/Imaging Reports**

**Lifestyle & Alcohol Consumption Data**

# 3. Architecture

* **Frontend:** The **frontend** of a liver cirrhosis prediction system is the **user interface** that allows doctors, patients, or researchers to input data and view results. It should be **simple, user-friendly, and informative**.
* **Backend:**The **backend** of a liver cirrhosis prediction system is responsible for **handling data processing, running prediction models, and returning results** to the frontend. It acts as the brain behind the user interface.
* **Database:**

The **database** of a liver cirrhosis prediction system is used to **store, retrieve, and manage**:

* Patient information
* Clinical/lab test results
* Prediction results
* System logs or feedback
* Model monitoring data (optional)

# 4. Setup Instructions

* **Prerequisites:**

Basic Python programming

Understanding of Machine Learning (ML)

Basics of HTML/CSS/JavaScript (for frontend)

* **Installation:**

**Download** Python: <https://www.python.org/downloads/>

 Make sure to check ✅**"Add Python to PATH"** during installation..

# 5. Folder Structure

* **Client:** Describe the structure of the React frontend.
* **Server:** Explain the organization of the Node.js backend.

# 6. Running the Application

• Provide commands to start the frontend and backend servers locally.

o **Frontend:**npm start in the client directory. o **Backend:**npm start in the server directory.

# 7. API Documentation

Endpoint Method Description

/api/predict POST Sends input data and receives prediction

/api/history GET Returns previously logged predictions

/api/export GET Exports data to CSV or PDF format

# Authentication

Method: Token-based Authentication (JWT)

 Usage:

o Tokens issued on login

o Middleware verifies tokens for protected routes

o Admin and user roles supported

1. **User Interface**

. Dark/light mode toggle

 Components:

o Input form for predictions

o Output cards and charts

o Admin dashboard with stats and export options

1. **Testing**

ools Used:

o Jest for React unit testing

o Postman for backend API testing

o Pytest for ML model evaluation

1. **Screenshots or Demo** 
   * https://drive.google.com/file/d/1aBM4qNDavIeq3B7LQeiaK52W-72Fvj9m/view?usp=drivesdk.
2. **Known Issues**

.Occasional lag on large dataset imports

 Limited dataset coverage in rural regions

 Requires retraining for seasonal data changes

# 13. Future Enhancements

•Integrate mobile app (React Native)

 Add real-time traffic camera feed analysis

 Smart signal automation via IoT integration