

CHAPTER 1

INTRODUCTION

Patients with Liver disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. This dataset was used to evaluate prediction algorithms in an effort to reduce burden on doctors. The data set include Age of the patient, Gender of the patient, Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase, Alamine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin, Albumin and Globulin Ratio. Dataset: field used to split the data into two sets (patient with liver disease, or no disease). Given a dataset containing biological and diagnostic data of 583 Indian patients, this project aims to identify a suitable machine learning algorithm which is capable of identifying whether a person has liver disease or not.

1.1 RELEVANCE OF THE PROJECT

This software allows to predict a patient with liver disease or not using certain attributes like total bilirubin, direct bilirubin, alkaline phosphatase, total protein, albumin, globulin etc.

1.2 PURPOSE

Medical diagnoses have important implications for improving patient care, research, and policy. For a medical diagnosis, health professionals use different kinds of pathological methods to make decisions on medical reports in terms of the patients' medical conditions. The use of artificial intelligence and machine learning in combination with clinical findings has further improved disease detection. The purpose of this study was to extract significant predictors for liver disease of humans using ML algorithms. This methods predict liver disease by incorporating the risk factors, which may improve the inference-based diagnosis of patients.

1.3 SCOPE OF THE PROJECT

This is a binary classification problem to be solved using supervised learning. It consists of ten features for each data point and a label which identifies whether the patient is suffering from liver disease or not. In order to arrive at the solution, our aim should be to train various supervised learning models on this dataset so that we have a well performing model which is able to classify any new data point as a positive or negative with a reasonable degree of accuracy and perform better than the benchmarks.

1.4 PROBLEM STATEMENT AND DEFINITION

Problem definition deals with defining the actual problem involved in the existing system or the system to be developed. Studies on various areas covered by the existing system are classified into various divisions and the actual task to be performed in the new system is determined. The project will be able to demonstrate the ideas of a website which helps the public. The website is trying to revitalize and simplify the various functions and activities and make them more people friendly. This is dedicated to providing better and speedy services to the public.

1.5 OBJECTIVE

This project helps to predict liver disease using the random forest algorithm.

CHAPTER 2

LITERATURE REVIEW

1. Software-based Prediction of Liver Disease with Feature Selection and Classification Techniques. [JagdeepSingh 1970–1980]

Today's health care is very important aspect for every human, so there is a need to provide medical services that are easily available to everyone. In this paper, the main focus is to predict the liver disease based on a software engineering approach using classification and feature selection technique. The implementation of proposed work is done on Indian Liver Patient Dataset (ILPD) from the University of California, Irvine database. The different attributes like age, direct bilirubin, gender, total bilirubin, Alkphos, sgpt, albumin, globulin ratio and sgot etc, of the liver patient dataset, are used to predict the liver diseases risk level. The various classification algorithms such as Logistic Regression, Sequential minimal optimization and K-nearest neighbor are implemented on the Liver Patient dataset to find the accuracy.

2. Biochemical Evaluation of Patients of Alcoholic Liver Disease and Non-alcoholic Liver Disease. [PRASAD.P.TORKADI 1979–1983]

The main disadvantage of this system is that the KNN algorithm classifies the data in accordance with the majority in the dataset, this algorithm predicts the result, but the accuracy is very less. Alcoholic liver disease (ALD) is due to excessive alcohol intake for long duration. Distinguishing ALD from non-ALD (non-alcoholic steatohepatitis, hepatitis of viral origin) is difficult as patient may deny alcohol abuse. Accurate diagnosis is important as management of ALD differs from non-ALD patients. The aim of this system was (1) To evaluate the patients of ALD and non-ALD by biochemical parameters compared to controls, (2) To assess whether these parameters can differentiate ALD from non-ALD. Study was carried out on 50 patients of ALD in group I and 35 patients of NASH (non-alcoholic steatohepatitis) and acute viral hepatitis each in group II. Our study concludes that ALD patients can be differentiated from NASH and acute viral hepatitis with certainty by measuring serum AST/ALT ratio, GGT and ALP. These biochemical parameters may help clinicians to support the diagnosis of ALD and non-ALD.

3. Liver Disease Prediction using Naïve Bayes Algorithms. [Dr. S. Vijayarani 1816–1820]

In recent years in healthcare sectors, data mining became an ease of use for disease prediction. Data mining is the process of dredge up information from the massive datasets or warehouse or other repositories. It is a very challenging task to the researchers to predict the diseases from the voluminous medical databases. To overcome this issue the researchers use data mining techniques such as classification, clustering, association rules and so on. The main objective of this research work is to predict liver diseases using classification algorithms. The algorithms used in this work are Naïve Bayes. These classifier algorithms are compared based on the performance factors i.e. classification accuracy and execution time.

4. Evaluation of Abnormal Liver Tests [Tinsay A. Woreta 2014]

The use of serum biochemical tests plays an important role in the diagnosis and management of liver diseases. The routine use of such tests has led to the increased detection of liver diseases in otherwise asymptomatic patients, often providing the first clue of the presence of liver pathology. Such laboratory tests, in addition to a careful history, physical examination, and imaging tests, can help clinicians determine the cause of liver disease in most cases. Traditionally, liver injury has been characterized as primarily hepatocellular versus cholestatic based on the degree of elevation of aminotransferases compared with alkaline phosphatase. Although such a distinction can help direct initial evaluation, there is often significant overlap in the presentation of various liver diseases, which often have a mixed pattern.

5. Improving healthcare operations management with ML [OlegS. Pianykh 2020]

Healthcare institutions need modern and powerful technology to provide high-quality, cost-effective care to patients. However, despite the considerable progress in the computerization and digitization of medicine, efficient and robust management tools have yet to materialize. One important reason for this is the extreme complexity and variability of healthcare operations, the needs of which have outgrown conventional management. Machine learning algorithms, scalable and adaptive to complex patterns, may be particularly well suited to solving these. The main goal of this work was to study this relationship using a person having disease or not.

CHAPTER 3

EXISTING SYSTEM

From a detailed research about the topic, it came to know that there are only two systems exist on the same domain. The first system is fully manual. It can store the details of the patients and their medical reports. These are the main features of the first system. Second system is more productive than first system. From a similar work paper, it came to know that the system is built using KNN algorithm. This algorithm predicts the result, but the accuracy is very less. The main disadvantage of this system is that it classifies the data in accordance with the majority in the dataset. These are the main systems present in this domain. So it would be very much useful to common people if we introduce a system which predicts if there is any chance for liver disease or not. And that system should be very accurate.

LIMITATIONS: -

- The system was completely manual.
- Does not predict an accurate value using the KNN algorithm.
- This system uses a lot of time for giving an output to the user.

CHAPTER 4

PROPOSED SYSTEM

This system is used to predict liver disease using Random Forest algorithm. Compare among several statistical learning methods the ability to predict binary classifications of liver disease. Obtain confusion matrices for comparing actual classes with predictive classes, compare several ML approaches to assess a better performance of liver disease diagnosis. Evaluate receiver operating characteristic (ROC) curves for determining the diagnostic ability of binary classification of liver disease. Here the patient can book an appointment to a specified doctor and the doctor can prescribe what test are needed to be done. In this system it is used to predict the liver disease of a person based on the parameters such as total bilirubin, direct bilirubin, albumin, total protein etc. Also in this system it can add the donor and recipient details who need the liver and blood during the transplantation time.

The proposed system is mainly having the following modules:

☐ Registration Module:

In this module the patient and hospital can register into the site by giving the specified data.

☐ Login:

The registered patient, hospitals and doctors can login in order to access the sites and its contents using username and password.

☐ Liver Disease Prediction:

Predict the patient having the liver disease or not by using the attributes like Age of the patient, Total Bilirubin, Direct Bilirubin, Alkaline Phosphatase, Alamine Aminotransferase, Aspartate Aminotransferase, Total Proteins, Albumin, Albumin and Globulin Ratio.

☐ Donor Details:

View the details of blood group available and organs during transplantation.

BENEFITS OF THE PROPOSED SYSTEM

- This system predict a patient having liver disease or not.
- Machine learning increases system efficiency.

CHAPTER 5

SYSTEM REQUIREMENTS AND SPECIFICATIONS

The software requirements specification (SRS) is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties-the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development. Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client's problem and application area. SRS this communication gap.

Problem analysis is done to obtain a clear understanding of the needs of the clients and the users, and what exactly is desired from the software. Analysis leads to the actual specification. People performing the analysis called analysts, are also responsible for specifying the requirements.

The software project is initiated by the client's needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify their requirements by talking to these people and understanding their needs. These people and the existing documents about the current mode of operation are the basis source of information for the analyst.

5.1 HARDWARE SPECIFICATION

| | |
|-----------------|---------------------|
| MACHINE | : INTEL DUAL CORE |
| MOTHER BOARD | : INTEL 945 CHIPSET |
| MEMORY | : 4 GB |
| HARD DISK | : 500 GB |
| MONITOR | : 18.5" LED |
| MONITORKEYBOARD | : USB/3 |
| MOUSE | : USB/3 |

5.2 SOFTWARE SPECIFICATION

OPERATING SYSTEM : WINDOWS

WEB TECHNOLOGIES: PYTHON, DJANGO, HTML, CSS

DATABASE : MySQL

WEB BROWSER : GOOGLE CHROME/MOXILA FIREFOX

5.3 DEVELOPMENT TOOLS

5.3.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985 - 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python is a MUST for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. I will list down some of the key advantages of learning Python. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified

on Python 3. The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it. With Python 2's end-of-life, only python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains Python, an open source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and Python development.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a smallcore language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy. Alex Martello, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture." Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the Python reference implementation that would offer marginal increases in speed at the Cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use Py, a just-in-time compiler. Python is also available, which translates a Python script into C and makes direct c-level API calls into the python interpreter.

Features of Language

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at address. The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

5.1.1 Hypertext Mark-up Language (HTML)

It is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input/>` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programmes written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content.

5.1.2 JavaScript

It is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior. JavaScript can function as both a procedural and an object-oriented language. Objects are created programmatically in JavaScript, by attaching methods and Properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

5.1.3 Cascading Style Sheets (CSS)

It is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, Math or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

5.1.4 MYSQL

MySQL is an Oracle-backed open source relational database management system (RDBMS) based on Structured Query Language (SQL). MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

MySQL is an important component of an open source enterprise stack called LAMP. LAMP is a web development platform that uses Linux as the operating system, Apache as the web server, and MySQL as the relational database management system and PHP as the object-oriented scripting language. (Sometimes Perl or Python is used instead of PHP.)

Originally conceived by the Swedish company MySQL AB, MySQL was acquired by Sun Microsystems in 2008 and then by Oracle when it bought Sun in 2010. Developers can use MySQL under the GNU General Public License (GPL), but enterprises must obtain a commercial license from Oracle. Today, MySQL is the RDBMS behind many of the top websites in the world and countless corporate and consumer-facing web-based applications, including Facebook, Twitter and YouTube.

How MySQL works

MySQL is based on a client-server model. The core of MySQL is MySQL server, which handles all of the database instructions (or commands). MySQL server is available as a separate program for use in a client-server networked environment and as a library that can be embedded (or linked) into separate applications. MySQL operates along with several utility programs which support the administration of MySQL databases. Commands are sent to MySQL Server via the MySQL client, which is installed on a computer.

MySQL was originally developed to handle large databases quickly. Although MySQL is typically installed on only one machine, it is able to send the database to multiple locations, as users are able to access it via different MySQL client interfaces. These interfaces send SQL statements to the server and then display the results.

MySQL Features:

- **Relational Database Management System (RDBMS):** MySQL is a relational database management system.
- **Easy to use:** MySQL is easy to use. You have to get only the basic knowledge of SQL. You can build and interact with MySQL with only a few simple SQL statements.
- **It is secure:** MySQL consists of a solid data security layer that protects sensitive data from intruders. Passwords are encrypted in MySQL.
- **Client/ Server Architecture:** MySQL follows a client /server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc.
- **Free to download:** MySQL is free to use and you can download it from MySQL official website.
- **It is scalable:** MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.
- **Compatible on many operating systems:** MySQL is compatible to run on many operating systems, like Novell NetWare, Windows* Linux*, many varieties of UNIX*(such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).
- **Allows roll-back:** MySQL allows transactions to be rolled back, commit and crash recovery.
- **High Performance:** MySQL is faster, more reliable and cheaper because of its unique

storage engine architecture.

- **High Flexibility:** MySQL supports a large number of embedded applications which makes MySQL very flexible.

Core MySQL features:

MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands. Before 2016, the main difference between MySQL and SQL was that the former could be used on multiple platforms, whereas the latter could only be used on Windows. Microsoft has since expanded SQL to support Linux, a change which went into effect in 2017. When MySQL is installed via Linux, its package management system requires custom configuration to adjust security and optimization settings. MySQL also allows users to choose the most effective storage engine for any given table, as the program is able to utilize multiple storage engines for individual tables. One of MySQL's engines is InnoDB. InnoDB was designed for high availability. Because of this, it is not as quick as other engines. SQL uses its own storage system, but it does maintain multiple safeguards against loss of data. Both systems are able to run in clusters for high availability. SQL Server offers a wide variety of data analysis and reporting tools. SQL Server Reporting Services is the most popular one and is available as a free download. There are similar analysis tools for MySQL available from third-party software companies, such as Crystal Reports XI and Actuate BIRT.

5.1.5 PhpMyAdmin

PhpMyAdmin is a (web application) client for MySQL. MySQL is server where your commands get executed and returns you data, It manages all about data while PHPMyAdmin is a web Application, with user friendly, easy to use GUI makes it easy to handle database, which is difficult to use on command line. PhpMyAdmin is the web application written primarily in PHP. It's used for managing MySQL database.

To be more specific, here is the detailed definition:

MySQL is the world's most popular open source database. With its proven performance, reliability, and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, and all five of the top five websites. Additionally, it is an extremely popular choice as embedded database, distributed by thousands of ISVs and OEMs. PhpMyAdmin is a free and open source administration tool for MySQL and MariaDB. As a portable web application written primarily

in PHP, it is one of the most popular MySQL administration tools, especially for web hosting services.

5.1.6 Windows

Windows 10 professional integrates the strengths of windows 2008 professional such as standards-based security, manageability, and reliability, with the best business features of windows 98 and windows Millennium Edition, such as plug and play, simplified user interface, and innovative support services. This combination creates the best desktop operating system for business. Whether your business deploys windows XP professional on a single computer or throughout a worldwide network, this new operating system increases your computing power while lowering cost of ownership for desktop computers. Some of the features in XP are as follows:

The Microsoft website provides tutorials that you can use to learn about and deploy the Windows XP operating system. This technical walk-through provides step-by-step instructions and illustrations for installing and configuring key features of Windows XP server. It is the most flexible and powerful operating system developed by Microsoft team. It is more users friendly and a stable operating system equipped with much more added features. The operating system supports new technologies such as digital video disks, multiple monitors etc. along with plug and play and multi display features. It has a graphical user interface operating environment. Faster computing, easy access to remote information and control remote computers are some added features. Following are the common features of Windows 10.

Faster computing, easy access to remote information and control remote computers. Built-in networking and messaging facility.

- Easier to set up, add or remove.
- Increase system security and control.
- Support advanced networking and communication.

CHAPTER 6

FEASIBILITY STUDY

During the system analysis study of the proposed system is carried out to see whether it is carried out to see whether it is beneficial to the organization. It is both needed and prudent to evaluate the feasibility of a project at the earliest time and minimum expenditure. Feasibility study is a test of system proposal access, its workability, impact on the organization, ability to meet the user needs, and effective use of resources. The different steps involved in feasibility analysis are:

- Formation of a project team
- Preparing the system flow chart
- Enumerating the potential candidate system
- Identifying the candidate system

The proposed system will help to solving the problem more efficiently and accurately. The reports obtained after feasibility studies are given below, they are:

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility
- Behavioral Feasibility

6.1 Economic Feasibility

It will reduce expenditure and improve the quality of service. A system can develop technically and that will used if the installed must still be a good investment for the organization. Financial benefits must exceed the cost. In the case of proposed system, performance of the system is effective of its accuracy, faster response and user friendly in nature. The campus-wide community for events and placements reduce unnecessary expenses and wastage of many hours by its capabilities of fast operations.

6.2 Technical Feasibility

Technical Feasibility checks the work for the project be done with current equipment, existing software technology and available personal. And if technology is required, what is the likelihood that it can develop. Also checks whether the proposed system guarantees accuracy,

reliability, data security and ease of access. All the resources or implementing this software is available in this project. So we can say it is technically feasibility.

6.3 Operational Feasibility

People are inherently to change, and computers have been known to facilitate change. An estimate should be made about the reaction of the user, staff towards the development of a computerized system. Computer installations have something to do with turnover, transfer and changes in job status. Proposed projects are beneficial only if that can be turned in to information system that will meet the organizations operating requirements. In-operational feasibility study the management and users where found to have interest for a change. Since the system is user friendly and training is less needed.

6.4 Behavioral Feasibility

People are inherently resistant to change and old computers have been known to facilitate change. An estimate should made of how strong the reaction the user is likely to have towards the development of the computer installation has sometimes to do with turnover transfers, retaining and changes. It is understandable that the introduction of a user system requires special effort to educate, sell and train. Thus the behavioral feasibility analysis plays a major role in the modification as well as other feasibility analysis to achieve success in the organization.

CHAPTER 7

SYSTEM DESIGN

The system design process partitions the requirements to either hardware or software systems. It establishes overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships. Design is the process of applying various techniques and principles for the purpose of defining a device, a process on a system in sufficient detail to permit its physical realization. It is a process through which requirements are translated into a representation of the software engineering process and is applied regardless of the development paradigm that is used. System design innovation information requirements and conceptual design into technical specification and general flow of processing, they are:

- Process design
- Database design
- User interface design

7.1 PROCESS DESIGN

Users are of 3 types:

- Guest Users
- Registered Users
- Admin

➤ **GUEST USERS:**

They will treat as an unauthorized user until they make registration. They are not provided with so much of features because of some safety concerns. But treat customers always happy. They can visit application services and other general details of the site only.

➤ **REGISTERED USERS:**

Only registered users can use the services such as, Consult new doctor, can view the available medical camps and view the prescription history.

➤ **ADMIN:**

Admin can login, and can predict the patient having liver disease or not.

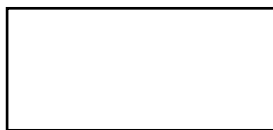
7.2 DATA FLOW DIAGRAM

Data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored.

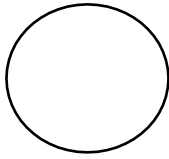
DFD is a designing tool used in the top-down approach to system Design. This context level DFD is next “exploded “, to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

- Function- An activity or a function that is performed for some specific reason; can be manual or computerized; ultimately each process should perform only one activity.
- Data Store- collection of data that is permanently stored.
- External Entity- A person, organization or system that is external to the system but interact with it.
- Data Flow- Single piece of data or logical collection of information like a bill.

The following are some DFD symbols used in the project



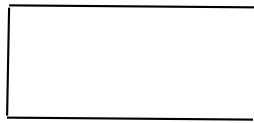
Rectangle: - It defines a source or destination of system data.



Circle: - It represents a process that transforms incoming data flow into outgoing data flow.



Arrow: - It defines data flow. It is a pipeline through which information flows.



Open rectangle: - It is used to store data or a temporary repository of data.

LEVEL 0 DFD:

LEVEL 0:DFD

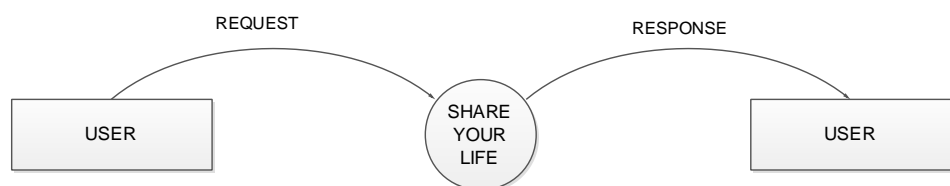


FIG 7.2.1: DATA FLOW DIAGRAM LEVEL 0

LEVEL 1 DFD FOR ADMIN:

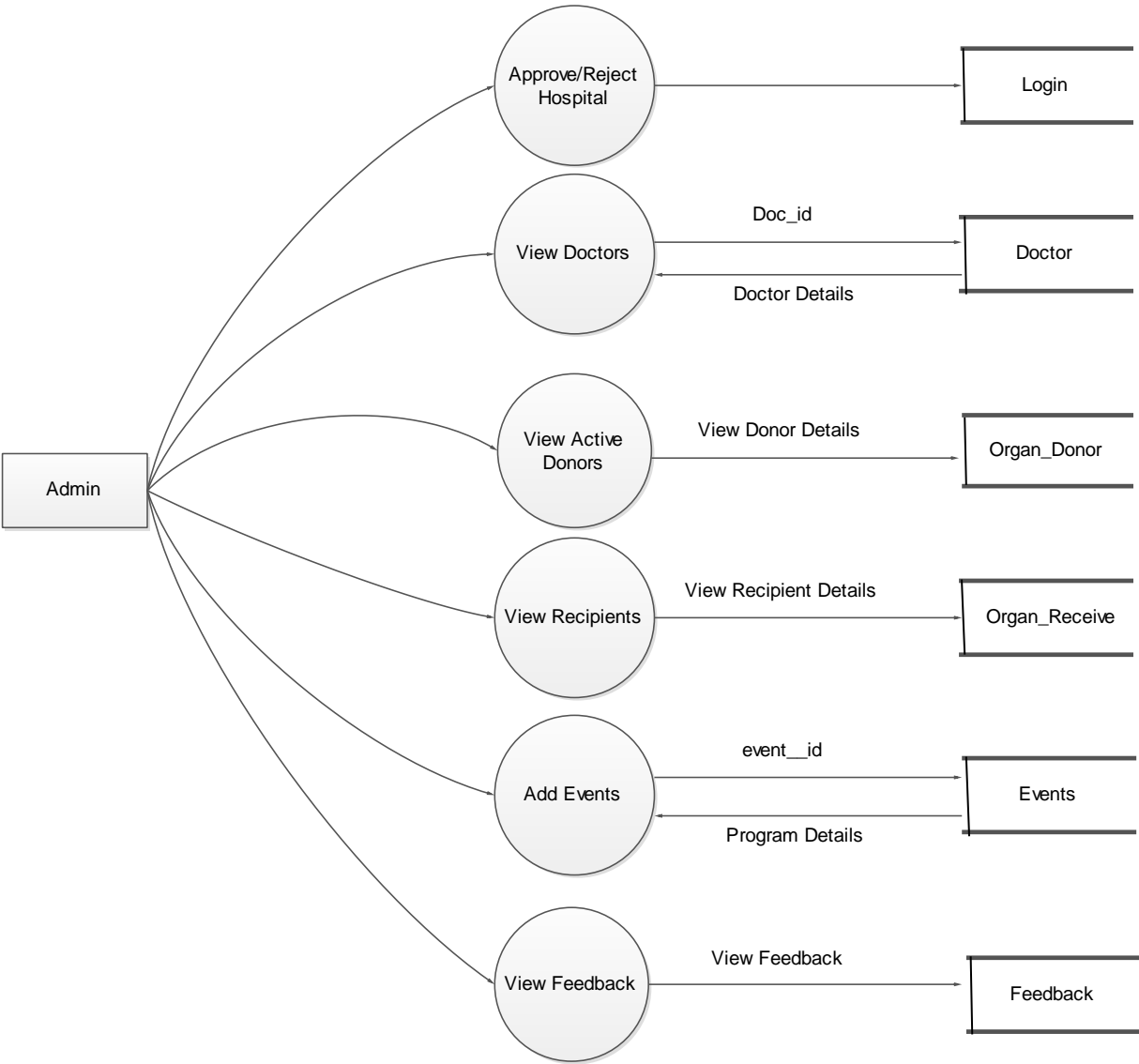


FIG 7.2.2: DATA FLOW DIAGRAM LEVEL 1 ADMIN

LEVEL 1 DFD FOR HOSPITAL:

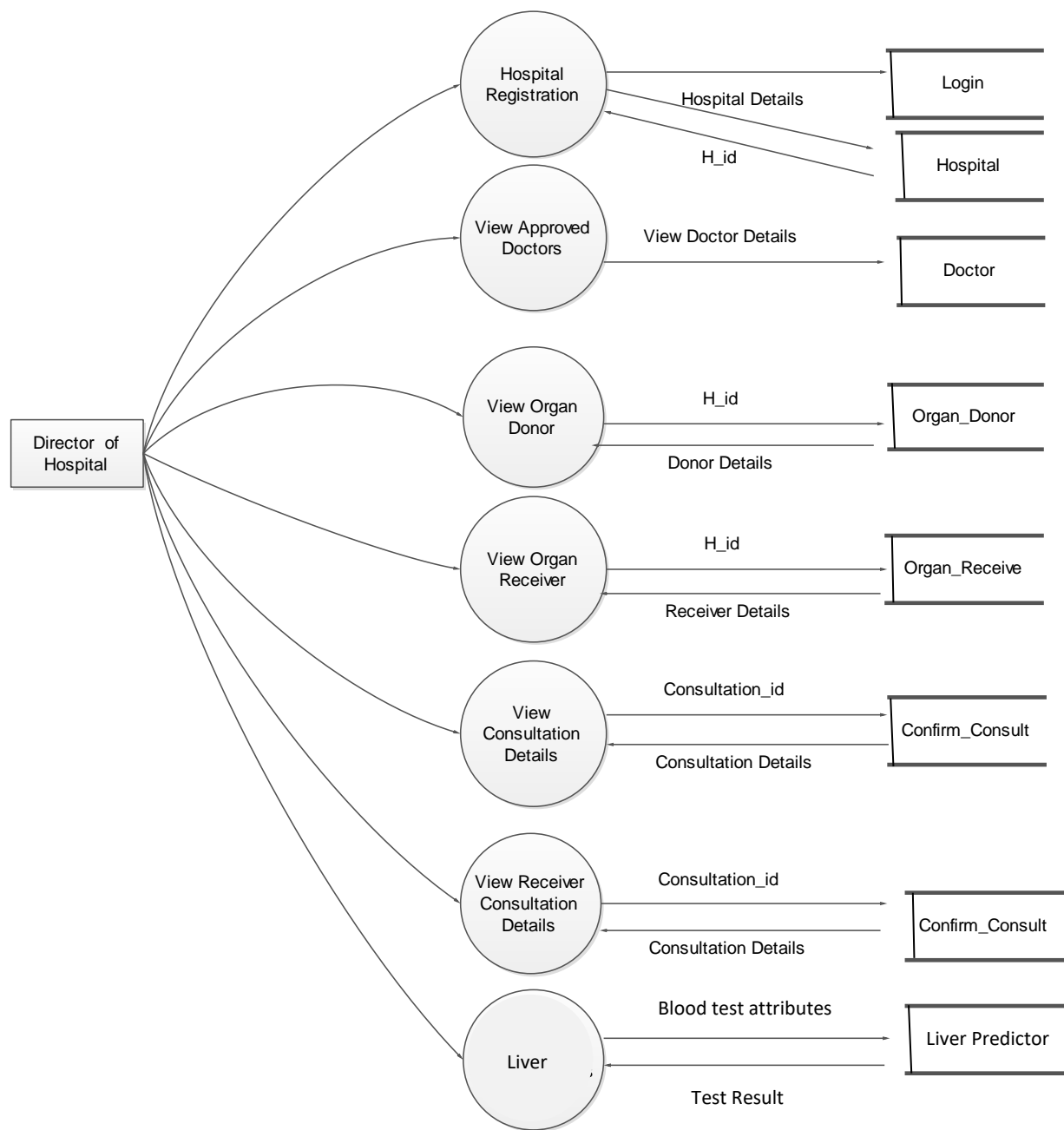


FIG 7.2.3: DATAFLOW DIAGRAM LEVEL 1 FOR HOSPITAL

LEVEL 1 DFD FOR DOCTOR:

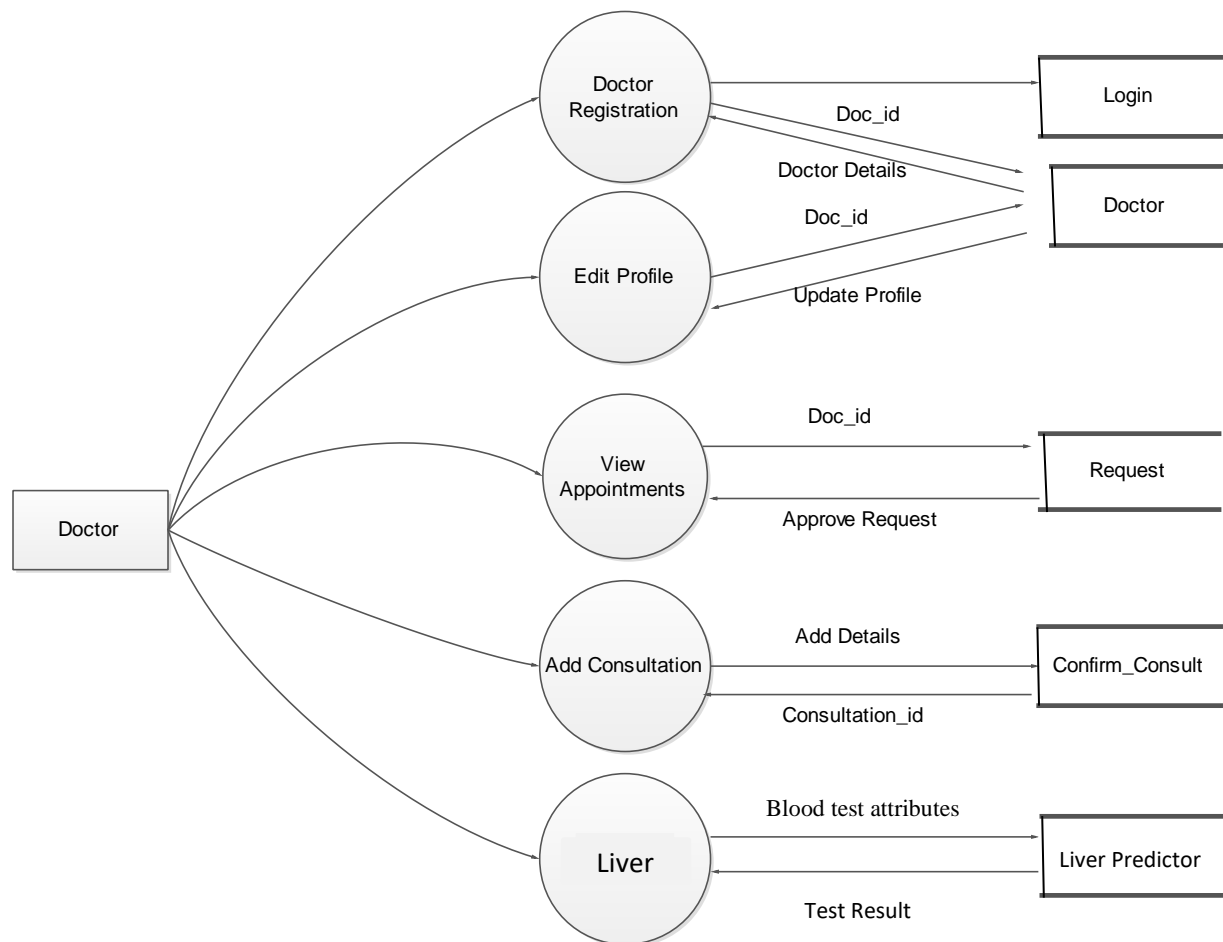


FIG 7.2.4: DATAFLOW DIAGRAM LEVEL 1 FOR DOCTOR

LEVEL 1 DFD FOR PATIENT:

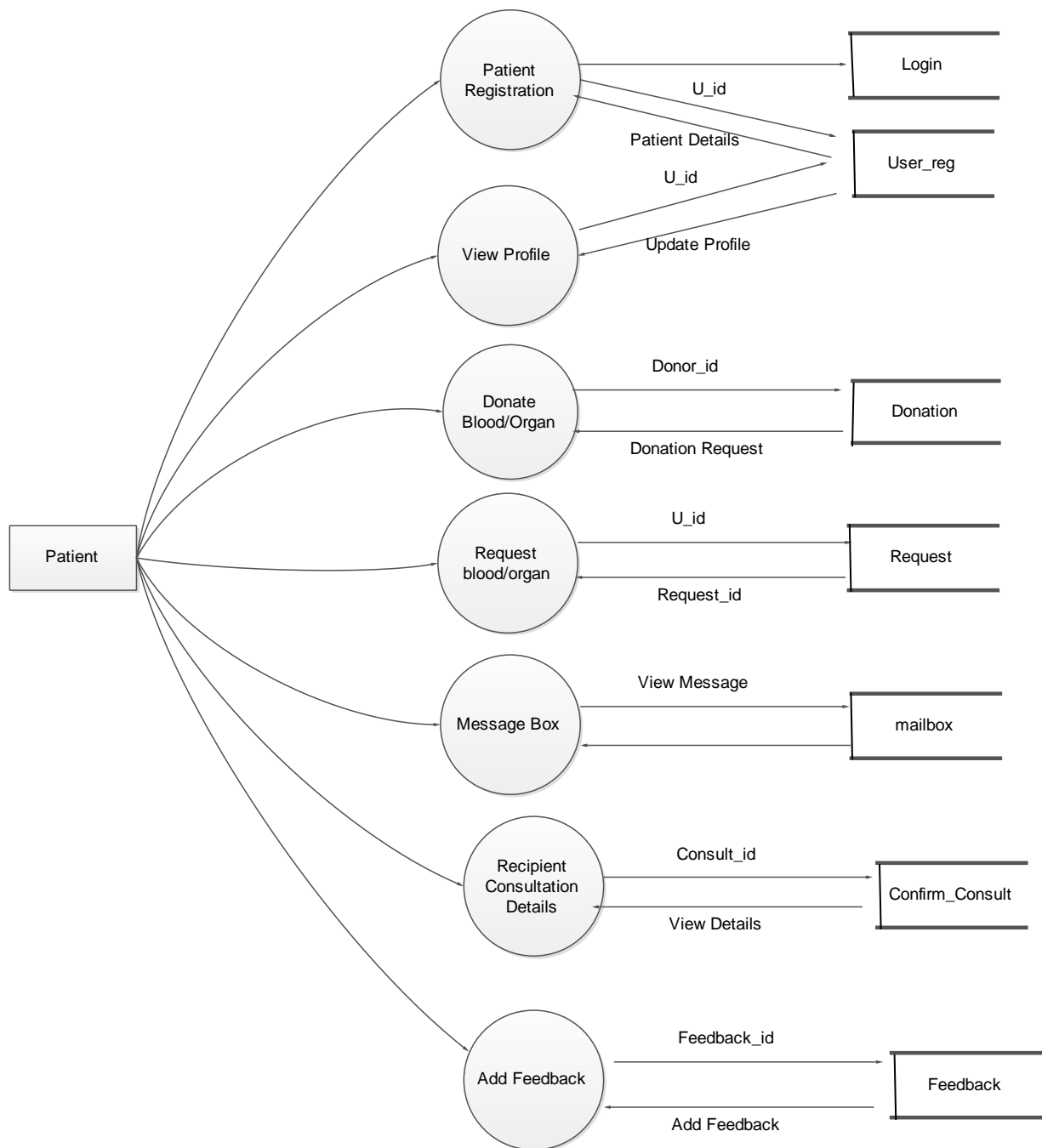


FIG 7.2.5: DATAFLOW DIAGRAM LEVEL 1 FOR PATIENT

7.3 USECASE DIAGRAM

Use case Diagram for Admin:

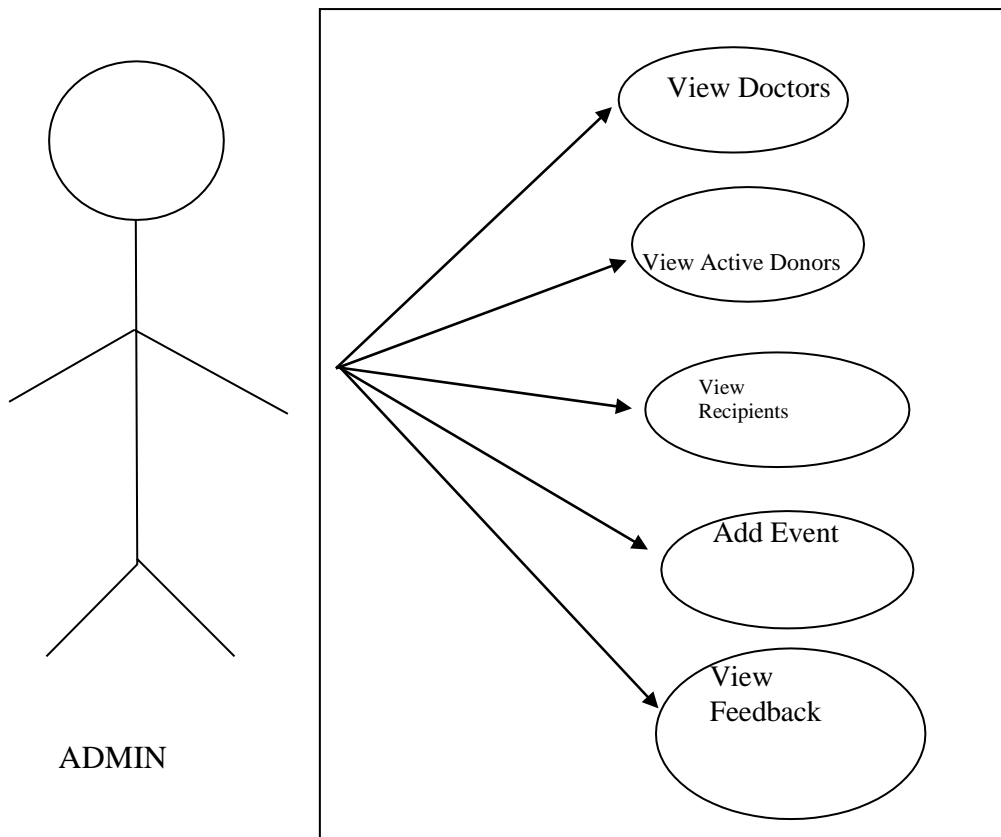


FIG 7.3.1 USECASE DIAGRAM FOR ADMIN

Use case Diagram for Hospital:

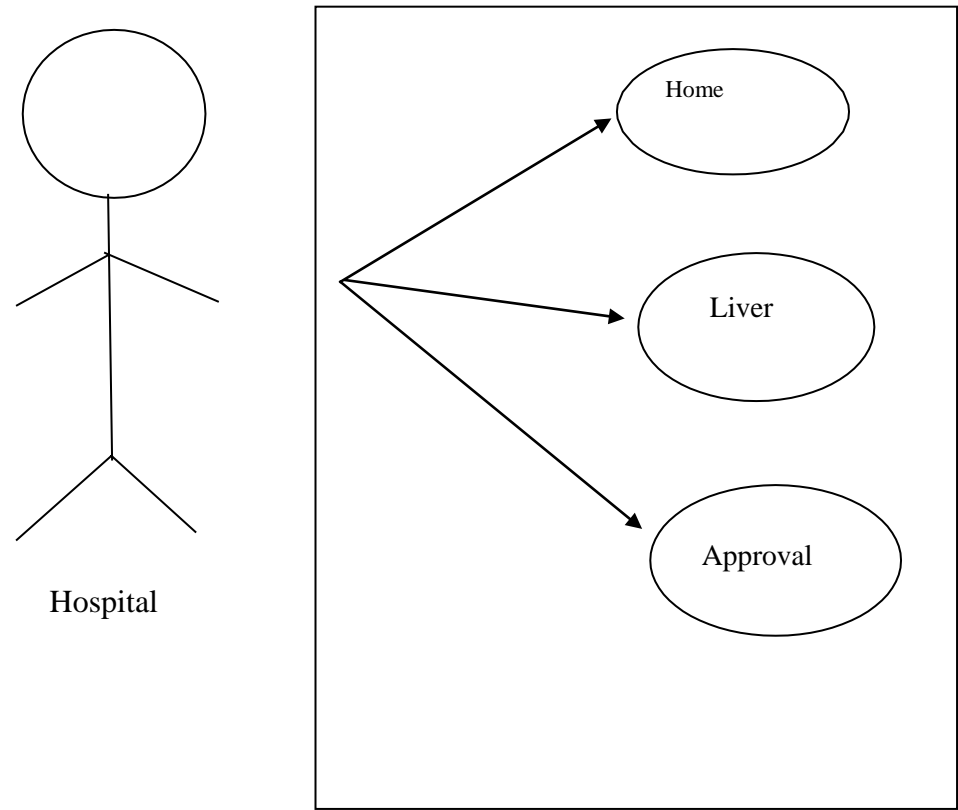


FIG 7.3.2 USECASE DIAGRAM FOR HOSPITAL

Use case Diagram for Doctor:

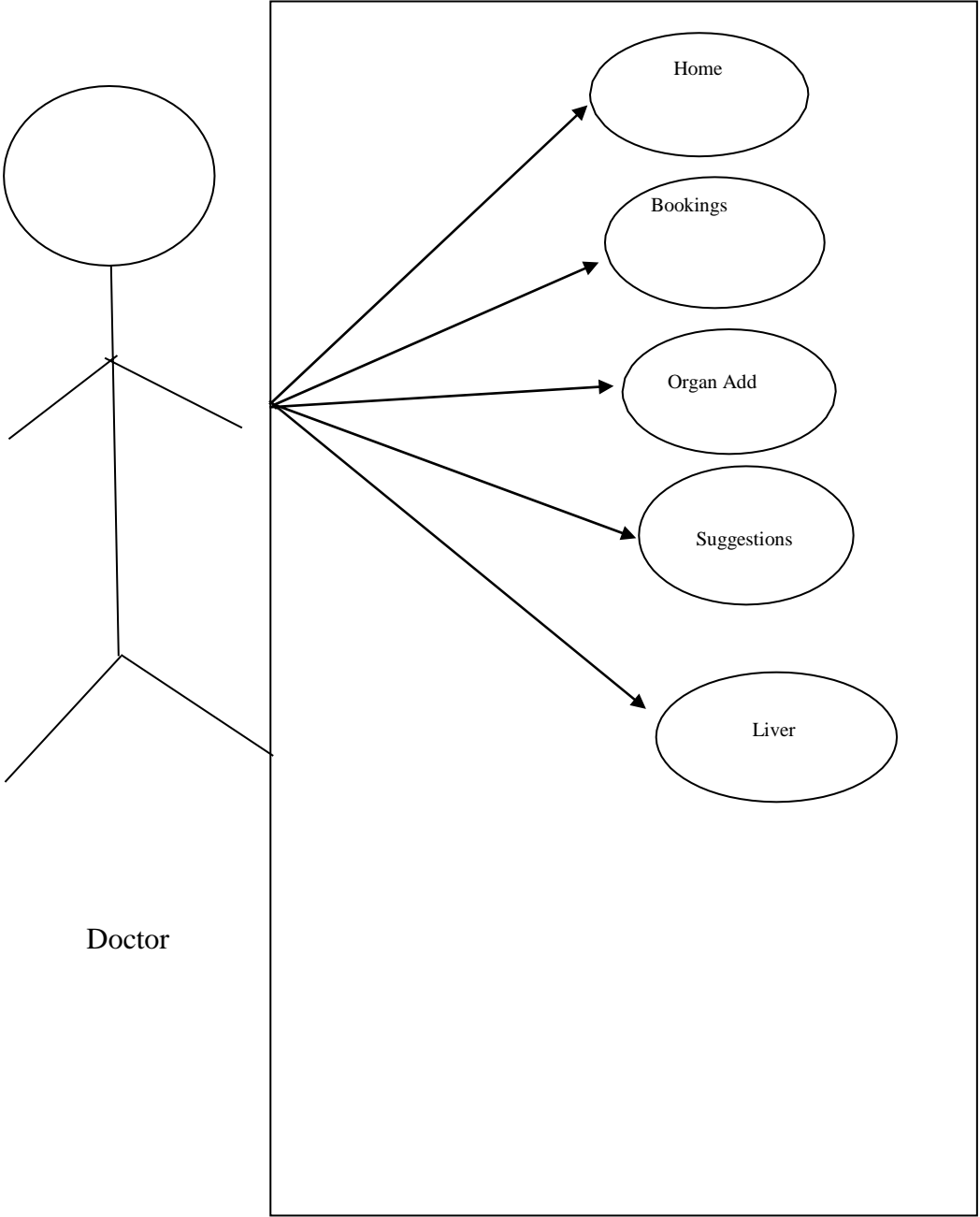


FIG 7.3.3 USECASE DIAGRAM FOR DOCTOR

Use case Diagram for Patient:

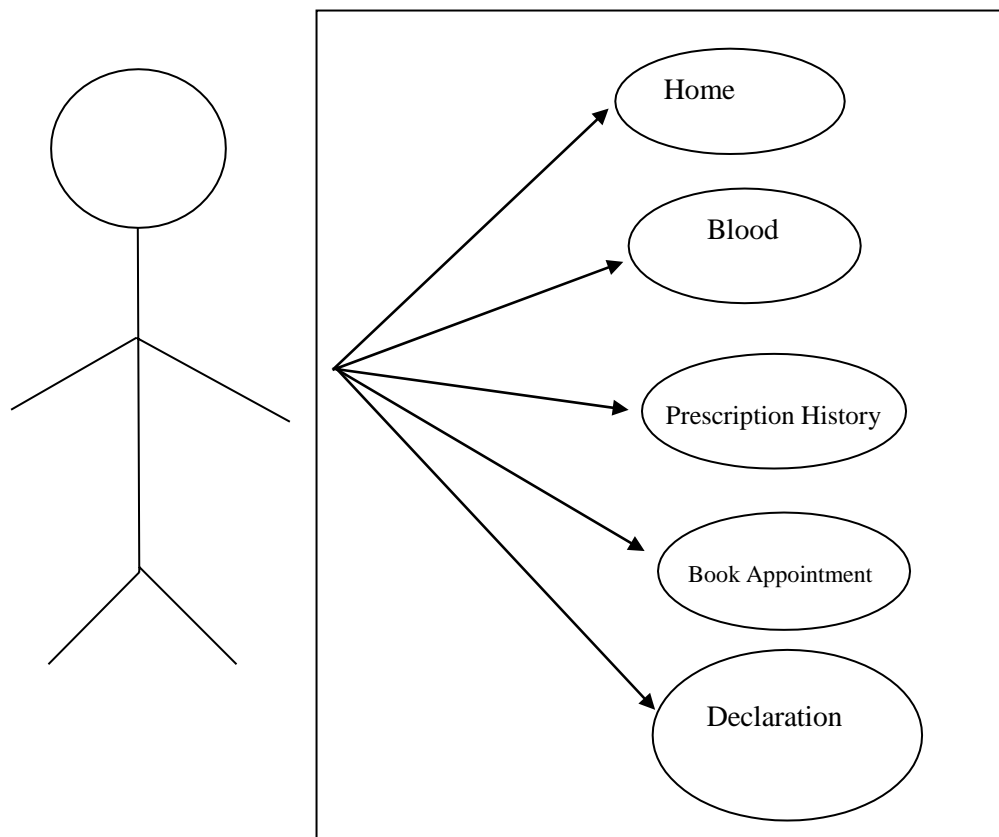


FIG 7.3.4 USECASE DIAGRAM FOR PATIENT

7.4 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objective are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at low-cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security

The design of the tables in the database is done according to the rules specified for database. In the proposed project, 10 tables are used and some of them are connected using foreign keys. Insertion and retrieval of values are easy by designing the database in this way.

Table name: Login

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------|--------------|--------------------|
| Username | varchar (10) | Username |
| Password | varchar (10) | Password |
| Type | varchar (10) | Type of the user |
| Status | int | Status of the user |

Table name: Hospital

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------------|--------------|--------------------------------|
| Hid | int | Hospital ID |
| Hinami | varchar (55) | Hospital name |
| City | varchar (20) | City |
| District | varchar (20) | District |
| State | varchar (20) | State |
| Telephone | varchar (15) | Telephone |
| Email | varchar (40) | Email |
| Website | varchar (70) | Website |
| Director | varchar (20) | Director of the hospital |
| H_d_number | varchar(15) | Mobile number of the director |
| Transp_lic_num | varchar (20) | Transplantation license number |
| Validator | date | License validate |
| Trans coordinator | varchar (20) | Transplantation coordinator |
| Unami | varchar (10) | Username |

Table name: Doctor

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------|--------------|-------------|
| Adocid | int | Doctor id |
| Name | varchar (30) | Name |
| Address | varchar (50) | Address |
| City | varchar (20) | City |
| District | varchar (20) | District |
| State | varchar (20) | State |
| Mobile | varchar (15) | Mobile |
| DOB | date | DOB |
| Gender | varchar (7) | Gender |

| | | |
|---------------|---------------|------------------------------|
| Qualification | varchar (15) | Qualification |
| Experience | float | Experience |
| Practicing at | int | Practicing Hospital |
| Ungenotyped | varchar (30) | Surgeon type |
| Dicarbenium | varchar (15) | Doctor's registration number |
| Unami | varchar (10) | Username |
| Photo path | varchar (500) | Photo path |

Table name: Registration

| | | |
|----------------|---------------|--------------------------|
| Urid | int | User id |
| Name | varchar (20) | Name |
| Address | varchar (30) | Address |
| City | varchar (20) | City |
| Dist. | varchar (20) | District |
| State | varchar (40) | State |
| Pin | int | Pin |
| Mobile | varchar (20) | Mobile |
| Email | varchar (40) | Email |
| DOB | Date | DOB |
| Gender | varchar (7) | Gender |
| Marital status | varchar (20) | Marital status |
| Blood Group | varchar (4) | Blood Group |
| Weight | Int | Weight of the user |
| Nazareth's | varchar (50) | Nearest hospital |
| Unami | varchar (10) | Username |
| Aadhar | varchar (12) | Aadhar card number |
| Photo path | varchar (500) | Photo path |
| Enamel | varchar (20) | Emergency contact person |
| Edmo | varchar (15) | Mobile number |

Table name: Donation

| COLUMN NAME | DATATYPE | DESCRIPTION |
|---------------|---------------|-----------------------------------|
| Donation | Int | Donation ID |
| Doridid | Int | User ID |
| Donated organ | varchar (10) | Donated organ |
| Blood Group | varchar (5) | Blood Group |
| Health path | varchar (MAX) | Preliminary health checkup record |

Table name: Donated

| COLUMN NAME | DATATYPE | DESCRIPTION |
|--------------|----------|--------------------|
| Did | Int | Donated id |
| donation | Int | Donation id |
| donated date | Date | Organ Donated date |

Table name: Request

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------|----------|-------------|
| Request | Int | Request id |
| Requestor | Int | User ID |
| Donation | Int | Donation id |
| Last date | Date | Last date |
| Hid | Int | Hospital id |
| Dottori | Int | Doctor id |
| Status | Int | Status |

Table name: Consultation

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------------|----------|-------------------|
| Consultation | int | Consultation id |
| Adocid | int | Doctor id |
| Reid | int | Request id |
| Consultation date | date | Consultation date |

Table name: Events

| COLUMN NAME | DATATYPE | DESCRIPTION |
|-------------|---------------|--------------|
| eventide | Int | Event id |
| Date | Date | Date |
| Venue | varchar (MAX) | Venue |
| Time | varchar (10) | Time |
| Program | varchar (MAX) | Program |
| Cond by | varchar (MAX) | Conducted by |

7.5 SYSTEM ARCHITECTURE

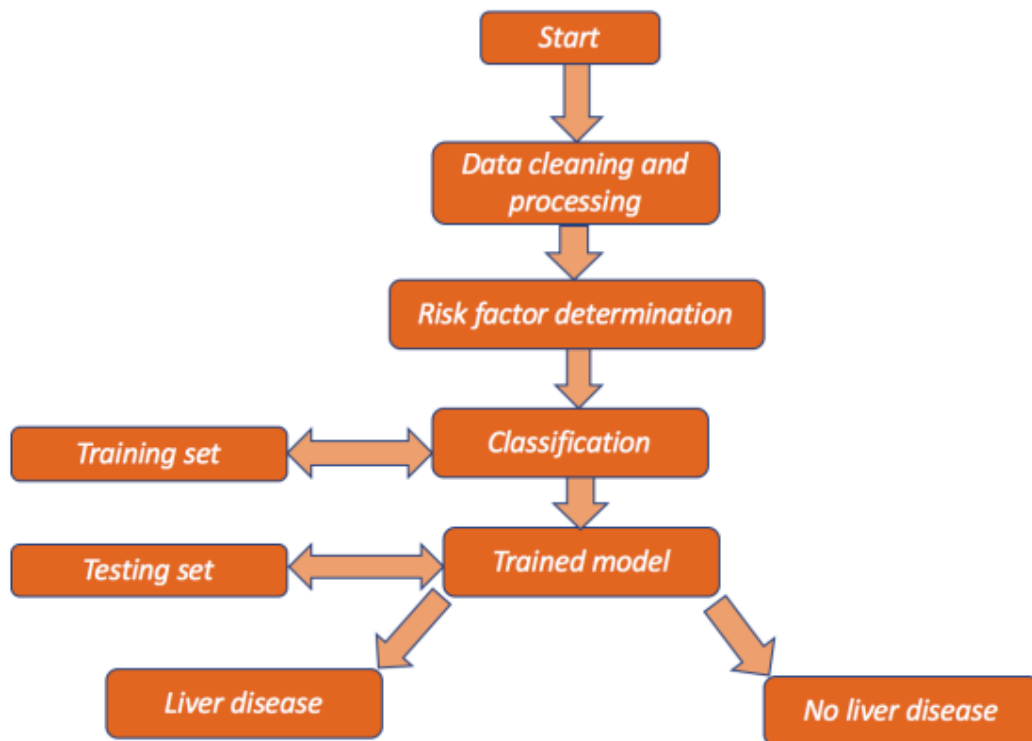


FIG 7.5.1 SYSTEM ARCHITECTURE

➤ **DATA CLEANING AND PROCESSING**

In this step it is used to remove the noisy data and the unwanted data from the database. The cleaned data is then processed and transferred to the next step.

➤ **RISK FACTOR DETERMINATION**

In this step here it is used to determine the data that are sufficient to the processing.

➤ **CLASSIFICATION**

Here we classify the data into two, training set and testing test.

➤ **TRAINING SET**

In training set train the data in the database and sent the trained data to the testing set.

➤ **TESTING SET**

Test the data that are trained and form a trained model.

➤ **TRAINED MODEL**

In the trained model by applying the random forest algorithm it is used to find out the patient who having liver disease or no.

CHAPTER 8

IMPLEMENTATION

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system, as well as, improper installation will prevent. There are four methods,

- Parallel approach: The old system is operated with the new system.
- Direct cut over method: The old system is replaced with the new system.
- Pilot approach: Working version of the system is implemented in one part of the organization based on the feedback, changes are made and the system is installed in the rest of the organization by one of the other methods.
- Phase-in-method: Gradually implements the system across all users.

8.1 Random Forest Algorithm

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. Random forest is a substantial modification of bagging that builds a large collection of de-correlated trees, and then they can be averaged. RF is very similar to boosting, and easy to train and tune. An average of p identical and independent random variables having variance σ^2 is used. Random forest helps to improve the variance reduction of bagging by reducing the correlation between trees without increasing the variance too much. Consider, any p from 1 to P , where there can be bootstrap sample W^* of size P from the training data. Then a random forest tree, T_p , can grow to the bootstrapped data. Afterwards, repetition of the following process for each terminal node of the tree occurs until the minimum node size n_{min} is reached. This gives m variables at random from the p variables and divides the node into two daughter nodes. Finally, the ensemble of trees by presenting the sequence $\{T_p\}_{1 \leq p \leq P}$ can be found. The prediction at a new point x is given. Therefore, for classification, $C_p(x)$ is the class prediction of the p th random forest tree.

CHAPTER 9

RESULT

Random Forest algorithm is used. This System aims to impute missing data using the (Multiple Imputation by Chained Equation) MICE algorithm. Compare among several statistical learning methods the ability to predict binary classifications of liver disease. Obtain confusion matrices for comparing actual classes with predictive classes, compare several ML approaches to assess a better performance of liver disease diagnosis. Evaluate receiver operating characteristic (ROC) curves for determining the diagnostic ability of binary classification of liver disease. Here the patient can book an appointment to a specified doctor and the doctor can prescribe what test are needed to be done. In this system it is used to predict the liver disease of a person based on the parameters such as total bilirubin, direct bilirubin, albumin, total protein etc. Also in this system it can add the donor and recipient details who need the liver and blood during the transplantation time.

CHAPTER 10

TESTING

The term software testing is defined as to find for the errors in the application that might lead to fault or failure of the whole application. There are testing conditions that the system must pass to say that it is tested and working properly. The quality and reliability is also attained by going through the process of testing.

➤ UNIT TESTING

Unit testing is a level of a software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class.(some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

➤ INTEGRATION TESTING

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of the level of testing is to expose faults in the interaction between integrated units. The purpose of this level of testing is to expose faults in the intersection between integrated units. The drivers and test stubs are used to assist in Integration Testing.

➤ SYSTEMS TESTING

System testing is a level of software testing where complete and integrated software is tested. The purpose of this test is to evaluate the system's compliance with the specified requirements.

➤ **ACCEPTANCE TESTING**

Acceptance testing is performed to ensure that the functional, behavioural, and performance requirements of the software are met. IEEE defines acceptance testing as a ‘formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.’

During acceptance testing, the software is tested and evaluated by a group of users either at the developer’s site or user’s site. This enables the users to site. The enables the users to test the software themselves and analyze whether it is meeting their requirements. To perform acceptance testing, a predetermined set of data is given to the software as input. It is important to know the expected output before performing acceptance testing so that outputs produced by the software as a result of testing can be compared with them. Based on the results of tests, users decide whether to accept or reject the software is considered to be correct and is accepted; otherwise, it is rejected.

➤ **REGRESSION TESTING**

Regression testing is the retesting of a software system to confirm that changes made to few parts of the codes has not any side effects on existing system functionalities. It is to ensure that old codes are still working as they were before introduction of the new change. The ideal process would be to create an extensive test suite and run it after each and every change.

❖ **VALIDATION CHECKS**

A validation check ascertains that the value (or data) input into a computer is valid. Validation checks are performed automatically by computer to ensure that entered data is correct and reasonable. Validation means check the input submitted by the user. There are two types of validation are available in PHP. They are as follows –

- Client-Side Validation – Validation is performed on the client machine web browsers.
- Server Side Validation – After submitted by data, the data has sent to a server and perform validation checks in server machine

- **Server Side Validation**

In the Server Side Validation, the input submitted by the user is being sent to the server and validated using one of server side scripting languages such as ASP.Net, PHP etc. After the validation process on the Server Side, the feedback is sent back to the client by a new dynamically generated web page. It is better to validate user input on Server Side because you can protect against the malicious users, who can easily bypass your Client Side scripting language and submit dangerous input to the server.

- **Client Side Validation**

In the Client Side Validation you can provide a better user experience by responding quickly at the browser level. When you perform a Client Side Validation, all the user inputs validated in the user's browser itself. Client Side validation does not require a round trip to the server, so the network traffic which will help your server perform better. This type of validation is done on the browser side using script languages such as JavaScript, VBScript or HTML5 attributes.

CHAPTER 11

FUTURE SCOPE OF PROJECT

11.1 SYSTEM MAINTENANCE

Maintenance means restoring something to its original conditions. Enhancement means adding, modifying the code to support the changes in the user specification. System maintenance conforms the system to its original requirements and enhancement adds to system capability by incorporating new requirements. Thus, maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system. It is an important part of system development that includes the activities which corrects errors in system design and implementation, updates the documents, and tests the data.

Maintenance Types

System maintenance can be classified into four types –

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

Corrective Maintenance

Corrective Maintenance deals with the repair of faults or defects found in day- today system functions. A defect can result due to errors in software design, logic and coding. Design errors occur when changes made to the software are incorrect, incomplete, wrongly communicated, or the change request is misunderstood. Logical errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow, or incomplete implementation of design specifications, faulty logic flow, or incomplete test of data. All these errors, referred to as residual errors, prevent the software from confirming to its agreed specifications. Note that the need for corrective maintenance is usually initiated by bug reports drawn by the users.

Adaptive Maintenance

Adaptive Maintenance is the implementation of changes in a part of the system, which has been affected by a change that occurred in some other part of the system. Adaptive Maintenance consists of adapting software to changes in the environment such as the hardware or the operating system. The term environment in this context refers to the conditions and the influences which act (from outside) on the system. For example, business rules, work patterns and government policies have a significant impact on the software system.

Perfective Maintenance

Perfective Maintenance mainly deals with implementing new or changed user requirements. Perfective Maintenance involves making functional enhancements to the system in addition to the activities to increase the system's performance even when the changes have not been suggested by faults. This includes enhancing both the function and efficiency of the code and changing the functionalities of the system as per the users changing needs.

Preventive Maintenance

Preventive Maintenance involves performing activities to prevent the occurrence of errors. It tends to reduce the software complexity thereby improving program understand ability and increasing software maintainability. It comprises documentation updating, code optimization and code restructuring. Documentation updating involves modifying the documents affected by the changes in order to correspond to the present state of the system. Code optimization involves modifying the programs for faster execution or efficient use of storage space. Code restructuring involves transforming the program structure for reducing the complexity in source code and making it easier to understand.

11.2 FUTURE ENHANCEMENT

- In future everyone will be busy.
- No one will have time to go to hospitals to find there diseases.
- In this situation this site will be of great help, for those who try to check there liver has any problem or not.
- It's just a click away.
- You can login to the site and can prescribe the doctor and find out if any problem for the liver.
- In future the various hospitals in the state can be linked together and thus we can include video chatting with famous doctors to clear our doubts regarding liver disease prediction.

CHAPTER 12

CONCLUSION

Chronic liver disease is detected by clinicians who are well trained in identifying significant observations and classifying them as normal or abnormal using background information and other context clues. ML algorithms can be trained to detect the possibility of liver disease in a similar way to assist healthcare workers. Using the correlation of each variable with the risk of liver disease to train the model, ML methods were able to identify which blood donors were healthy and which had liver disease with high accuracy. Application of the ML methods can help reduce the total burden of liver disease on public health worldwide by improving recognition of risk factors and diagnostic variables. More importantly, for chronic liver disease, detecting liver disease at earlier stages or in hidden cases by ML could decrease liver-related mortality, transplants, and/or hospitalizations.

CHAPTER 13

REFERENCES

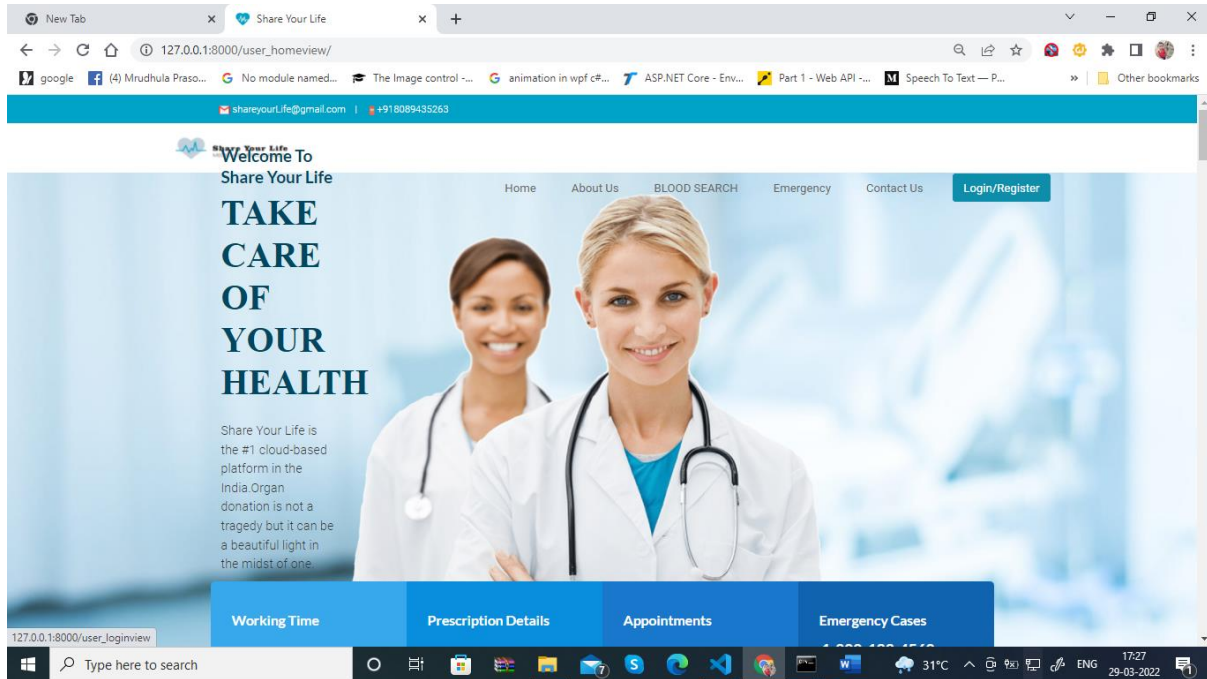
- Asrani, S.K.; Devarbhavi, H.; Eaton, J.; Kamath, P.S. “Burden of liver diseases in the world”. *J. Hepatol.* 2019.
- Chalasani, N.; Younossi, Z.; Lavine, J.E.; Charlton, M.; Cusi, K.; Rinella, M.; Harrison, S.A.; Brunt, E.M.; Sanyal, A.J. “The diagnosis and management of nonalcoholic fatty liver disease: Practice guidance from the American Association for the Study of Liver Diseases”. *Hepatology* 2018.
- Wang, Y.; Li, Y.; Wang, X.; Gacesa, R.; Zhang, J.; Zhou, L.; Wang, B. “Predicting Liver Disease Risk Using a Combination of Common Clinical Markers: A Screening Model from Routine Health Check-Up”. *Dis. Markers* 2020.

CHAPTER 14

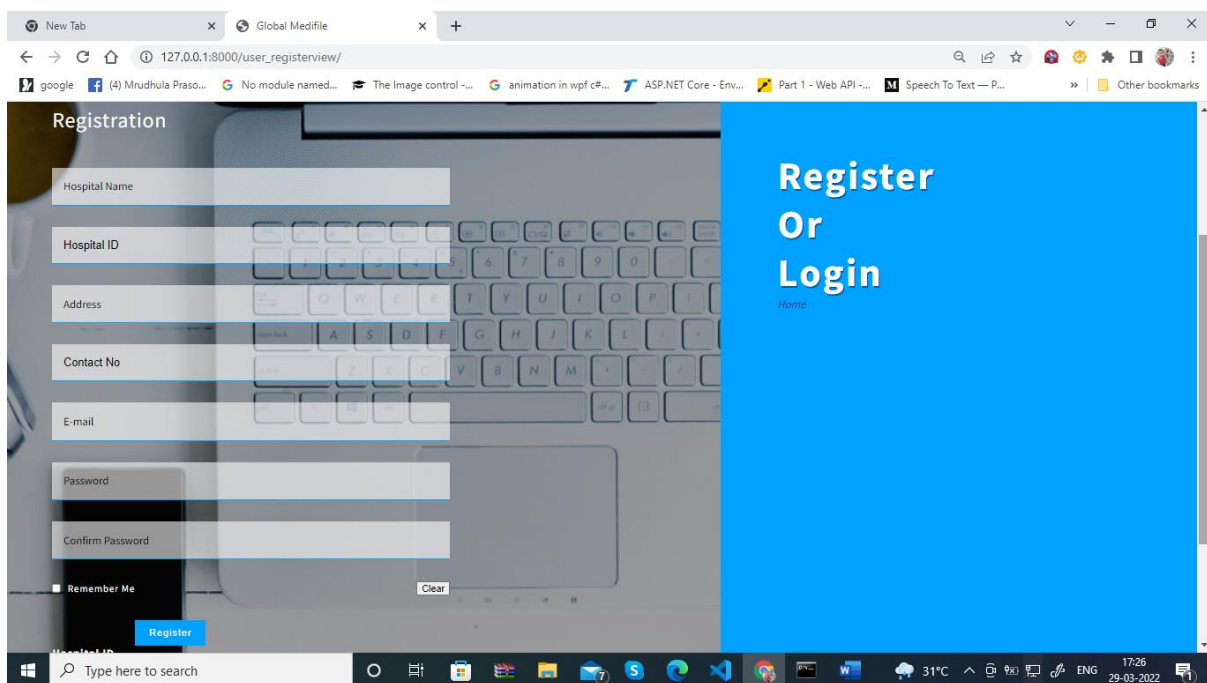
APPENDIX

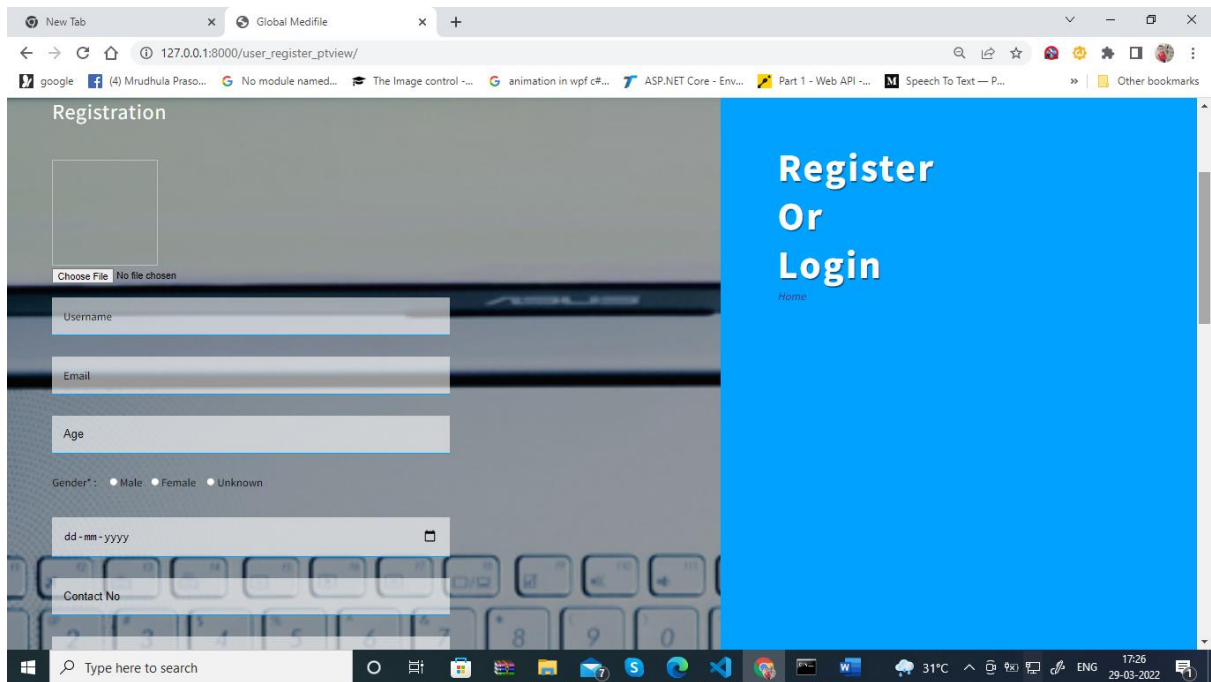
14.1 SCREENSHOTS

HOME PAGE

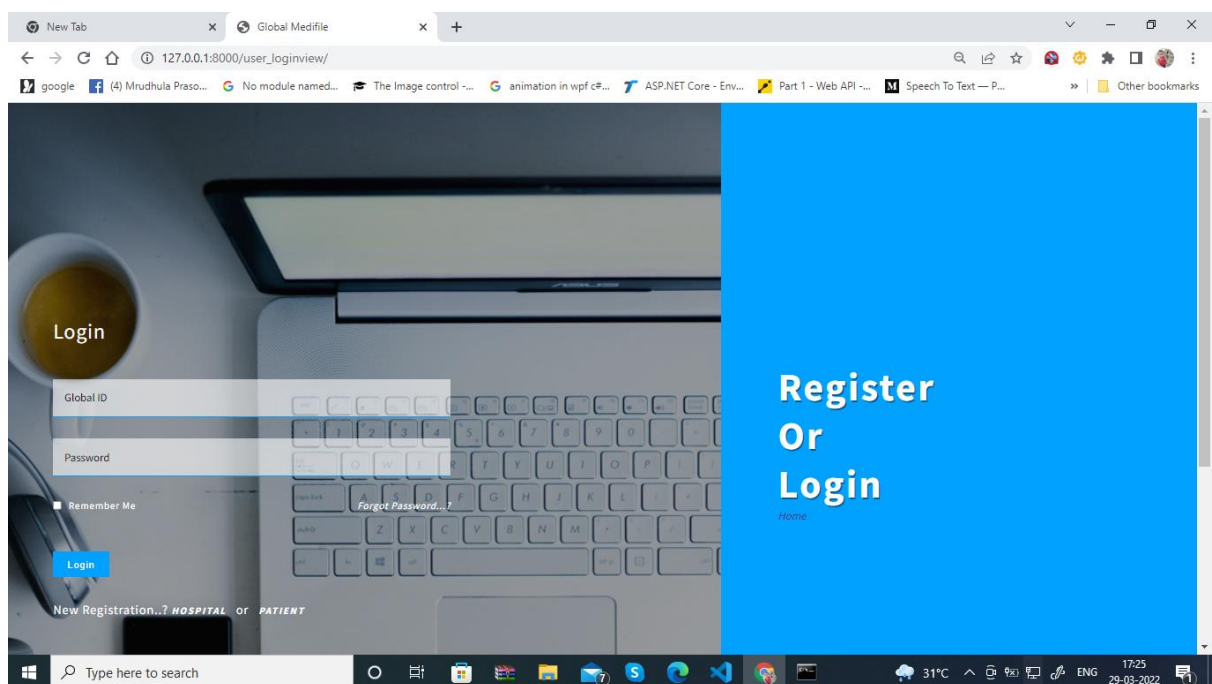


REGISTRATION PAGE

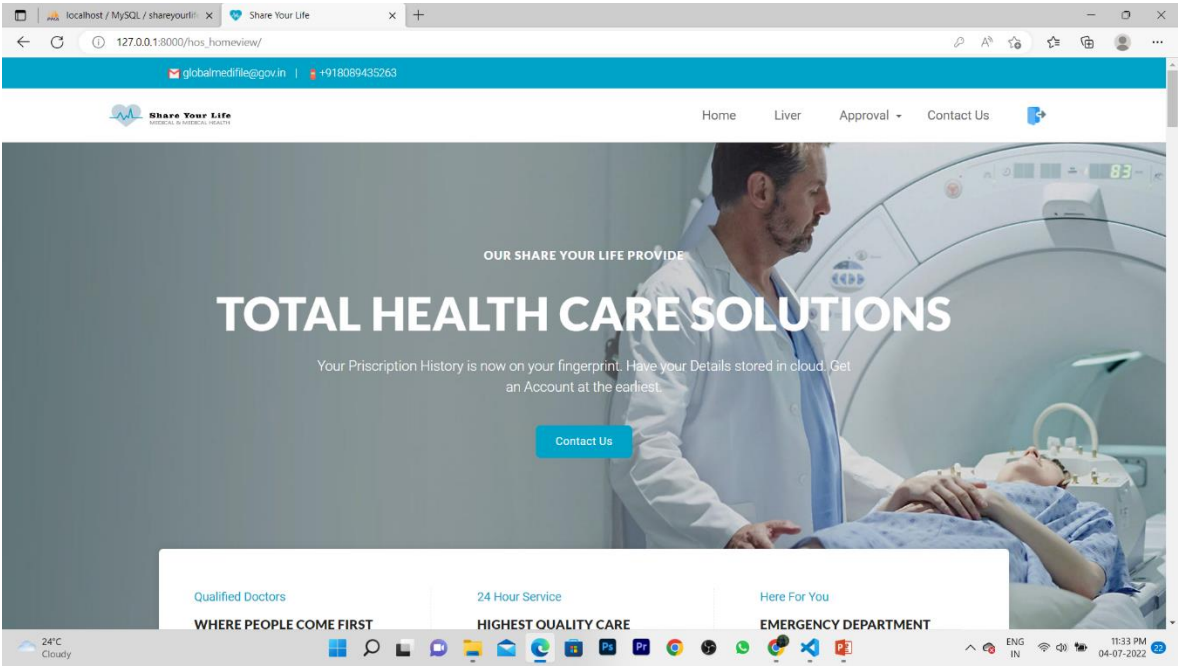




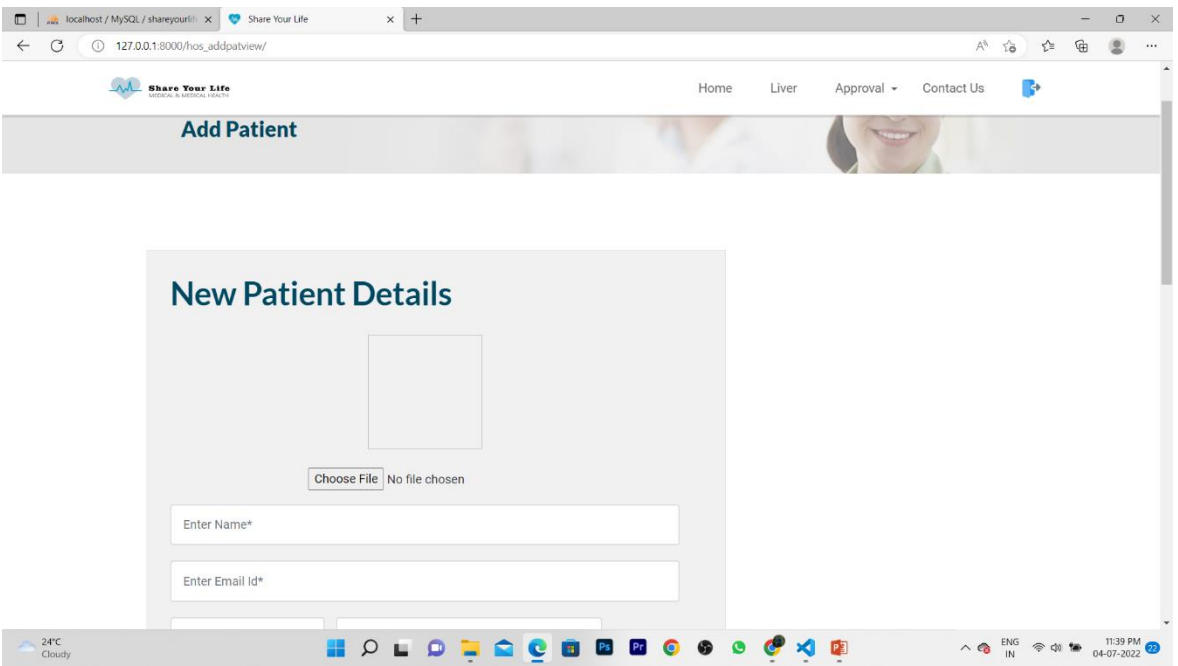
LOGIN PAGE



HOSPITAL HOME PAGE



ADD PATIENT



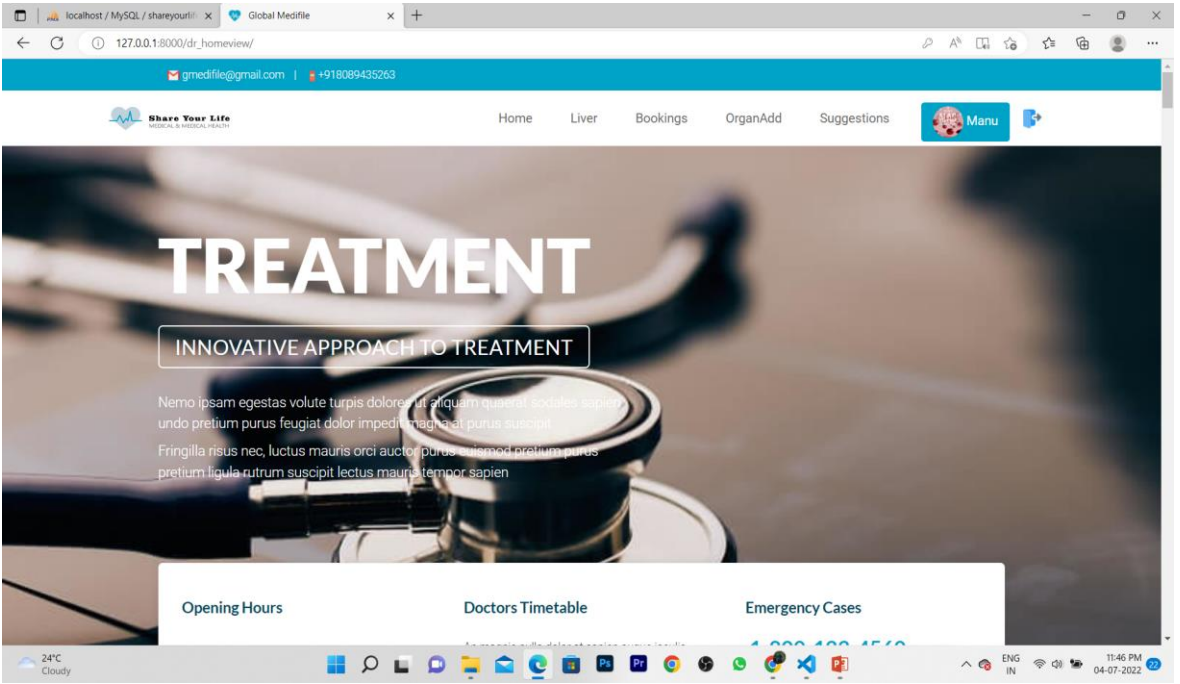
ADD NEW CAMP

The screenshot shows a web browser window with the URL `127.0.0.1:8000/ADDCAMP/`. The page features a navigation bar with links for Home, Liver, Approval, and Contact Us. The main content area is titled "New Camp Details" and contains a form with the following fields: "Enter Name*", "Location*", "Discription*", and a date field labeled "dd-mm-yyyy". A blue "Add New Camp" button is positioned below the form. A small note at the bottom of the form states: "* Add mandatory details carefully. Fill all details." The browser's taskbar at the bottom shows the system clock as 11:41 PM on 04-07-2022.

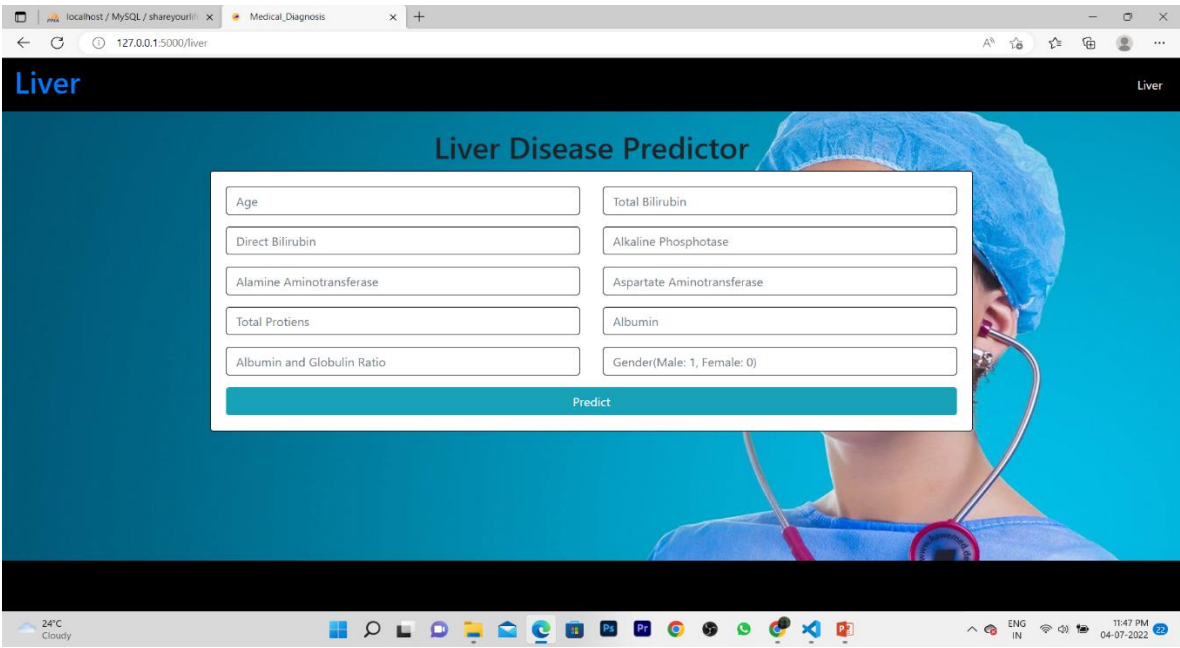
ADD NEW DOCTOR

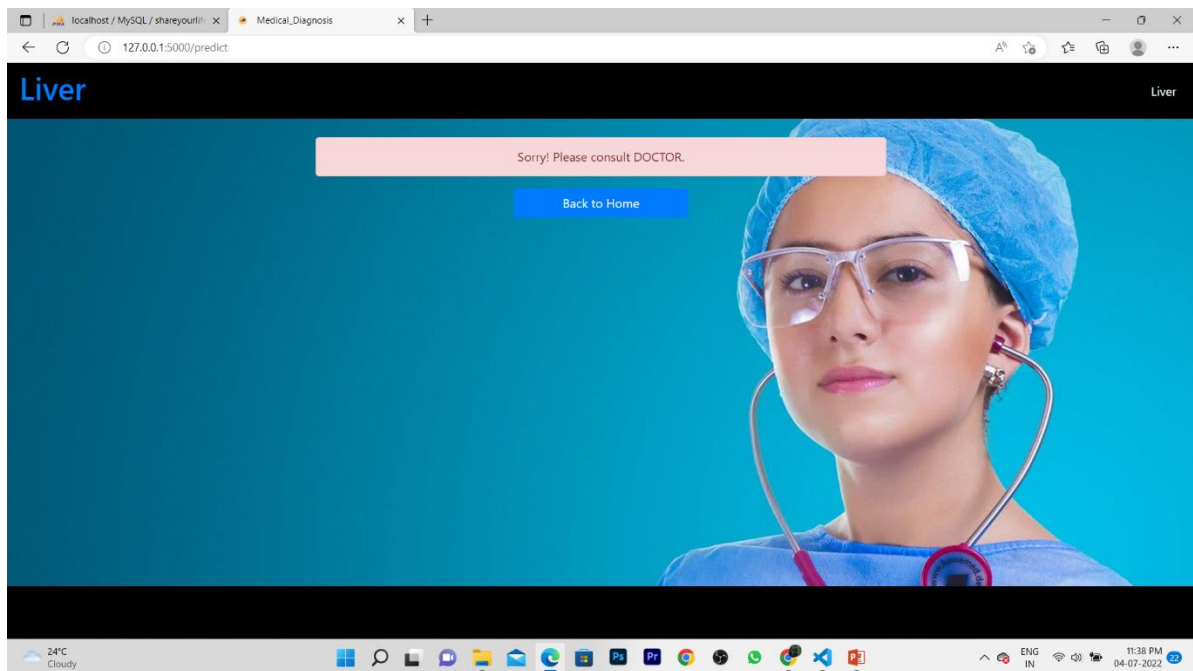
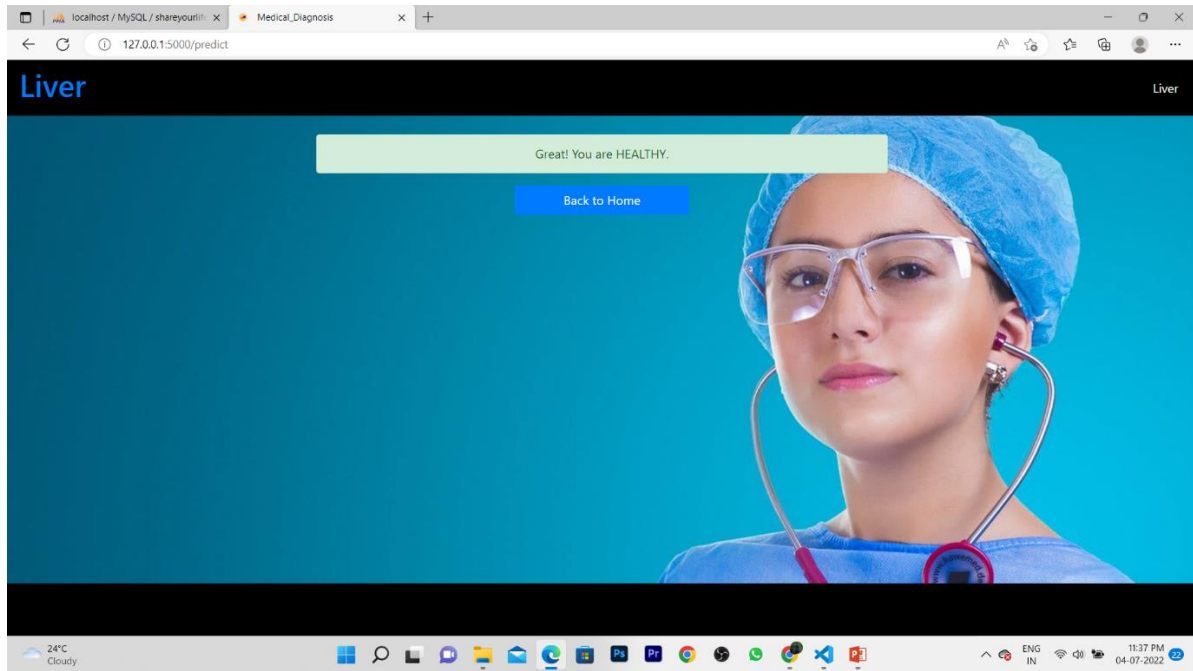
The screenshot shows a web browser window with the URL `127.0.0.1:8000/hos_adddocview/`. The page features a navigation bar with links for Home, Liver, Approval, and Contact Us. The main content area is titled "New Doctor Details" and contains a form with the following fields: a file upload section with a "Choose File" button and "No file chosen" text, "Enter Name*", "Enter Email Id*", "Age*", "Enter Phone Number*", and a "Select Department" dropdown menu. At the bottom of the form, there is a checkbox for "Forenoon" and a "Token No." input field. The browser's taskbar at the bottom shows the system clock as 11:43 PM on 04-07-2022.

DOCTORS HOME PAGE



LIVER DISEASE PREDICTOR





BOOKINGS

localhost / MySQL / shareyourlife | Global Medifile

127.0.0.1:8000/dr_bookview/

gmedifile@gmail.com | +918089435263

Share Your Life

Home | Liver | Bookings | OrganAdd | Suggestions | Menu

Home » Bookings

Bookings

Today's

Upcoming

Search Patient

Upcoming - Bookings

| Token No | Patient ID | Name | Date |
|----------|------------|------------|------------|
| 1 | 689914 | Anju | 2022-07-01 |
| 1 | 689914 | Anju | 2022-07-02 |
| 2 | 423709 | JOyal | 2022-07-02 |
| 3 | 765432 | JOyal | 2022-07-02 |
| 4 | 275647 | ROBIN BIJU | 2022-07-02 |

127.0.0.1:8000/dr_bookview/#tab-12

24°C Cloudy

Windows Taskbar

11:49 PM 04-07-2022

localhost / MySQL / shareyourlife | Global Medifile

127.0.0.1:8000/addorgany/

Share Your Life

Home | Liver | Bookings | OrganAdd | Suggestions | Menu

OrganAdd

roy mathew

Renal Medicity

BLOOD GROUP*

IF ANY DISEASE*

ORGAN

avail

Add New Doctor

* Add mandatory details carefully. Fill all details.

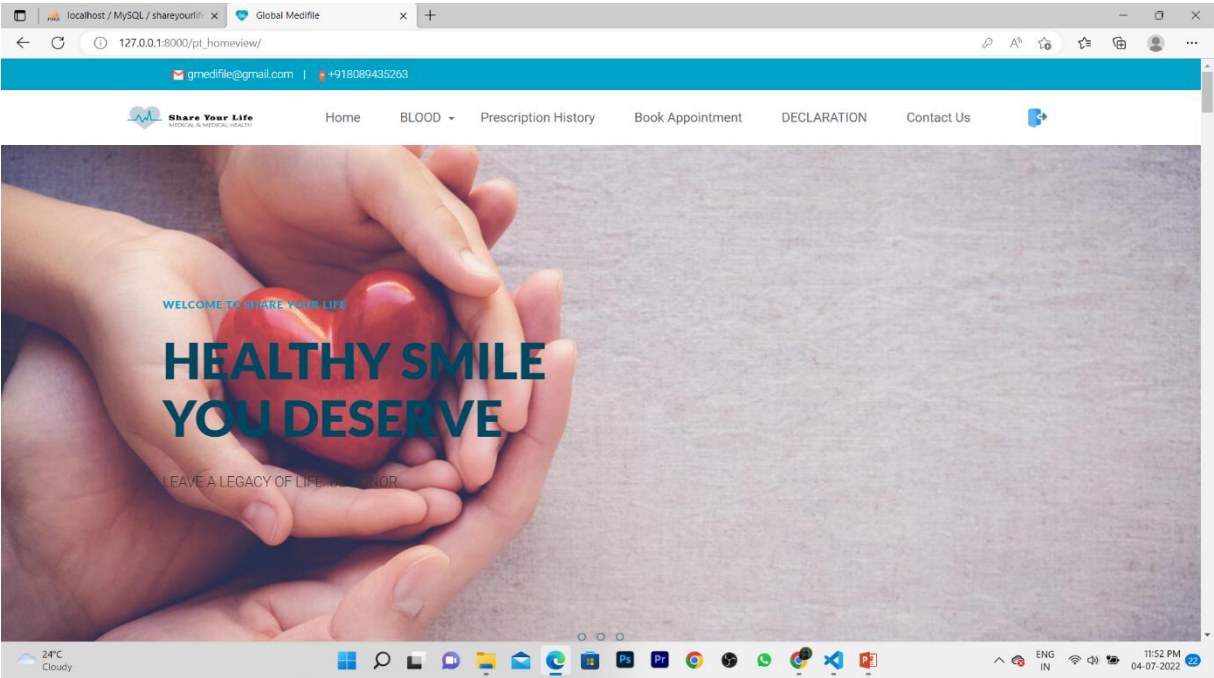
127.0.0.1:8000/addorgany/#tab-12

24°C Cloudy

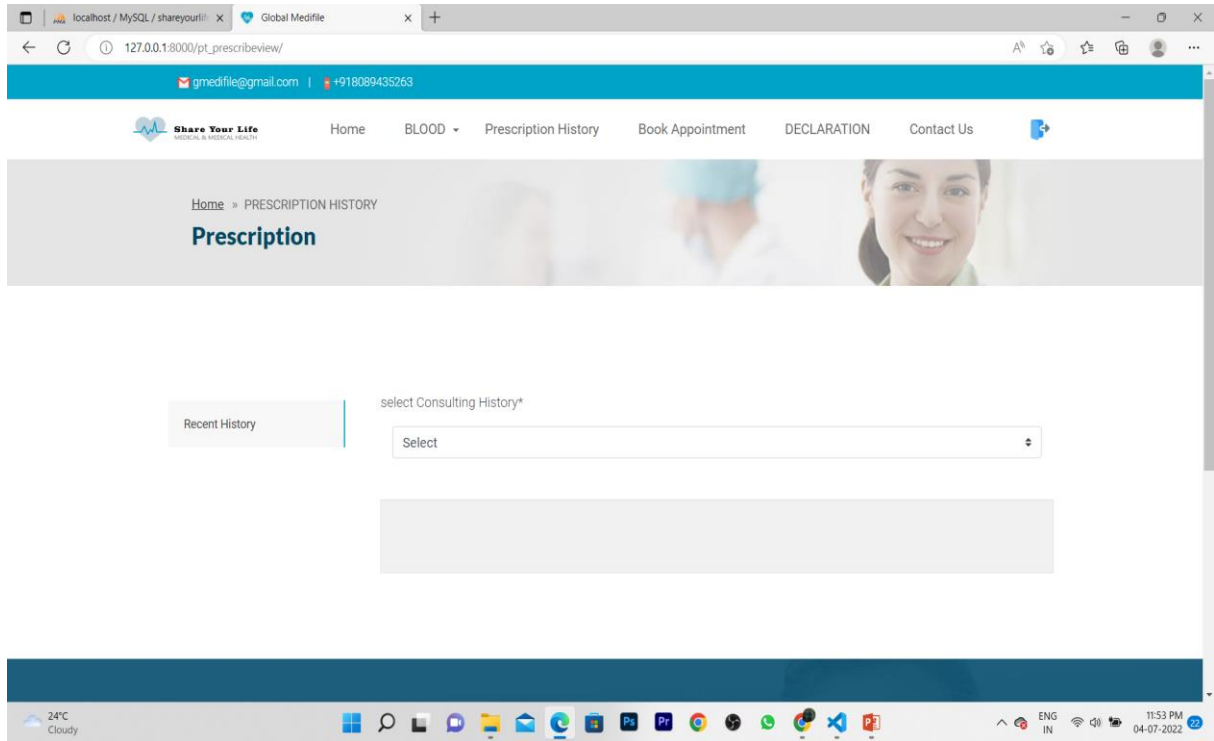
Windows Taskbar

11:50 PM 04-07-2022

PATIENT HOME PAGE



PATIENT PRESCRIPTION HISTORY



PATIENT BOOK APPOINTMENT

The screenshot shows a web browser window with the URL `127.0.0.1:8000/pt_bookview/`. The page features a navigation bar with links: Home, BLOOD, Prescription History, Book Appointment, DECLARATION, and Contact Us. The main heading is "Book Appointment". On the left, there is a sidebar with "Book appointment" and "Status". The main form area is titled "Book an Appointment" and includes the instruction "Fill out the Mandatory*". The form contains three fields: "Department Name*" with a dropdown menu showing "Select Department", "Hospital Name*" with a dropdown menu showing "Select Hospital", and "Booking Date*" with a text input showing "dd-mm-yyyy" and a calendar icon.

The screenshot shows a web browser window with the URL `127.0.0.1:8000/pt_bookview/`. The page features a navigation bar with links: Home, BLOOD, Prescription History, Book Appointment, DECLARATION, and Contact Us. The main form area is titled "Request an Appointment" and includes the instruction "Fill out the Mandatory*". The form contains five fields: "Token Number*" with a text input showing "Token Number", "Patient ID*" with a dropdown menu showing "Have You Visited Us Before?*", "Anju" with a text input, "689914" with a text input, and "Enter Your Phone Number*" with a text input. A blue button labeled "Request an Appointment" is at the bottom of the form. A scroll-to-top button is visible in the bottom right corner.

PATIENT DECLARATION

localhost / MySQL / shareyourlife X Global Medifile

127.0.0.1:8000/requestorgan/

Share Your Life
MEDICAL & MEDICAL HEALTHY

Home BLOOD Prescription History Book Appointment **DECLARATION** Contact Us

DETAILS

Hospital: Renal Medicity
Doctor: [v]

GIN*

Blood Group*

IF ANY DISEASE*

ORGAN DESCRIPTION

dd-mm-yyyy

DECLARATION

24°C Cloudy 11:56 PM 04-07-2022

14.2 SOURCE CODE

```
from flask import Flask, render_template, request, flash, redirect
import pickle
import numpy as np
from PIL import Image
from tensorflow.keras.models import load_model

app = Flask(__name__)

def predict(values, dic):
    if len(values) == 10:
        model = pickle.load(open('models/liver.pkl', 'rb'))
        values = np.asarray(values)
        return model.predict(values.reshape(1, -1))[0]

@app.route("/")
def home():
    return render_template('home.html')

@app.route("/liver", methods=['GET', 'POST'])
def liverPage():
    return render_template('liver.html')

@app.route("/predict", methods = ['POST', 'GET'])
def predictPage():
    p1=0
    p2=0
    try:
        if request.method == 'POST':
            to_predict_dict = request.form.to_dict()
            to_predict_list = list(map(float, list(to_predict_dict.values())))
            # pred = predict(to_predict_list, to_predict_dict)
            if(to_predict_list[2]<0.3):
                p1=p1+1
            else:
                p2=p2+1
            if(to_predict_list[1]>0.2 and to_predict_list[1]<1.2):
                p1=p1+1
            else:
                p2=p2+1
            if(to_predict_list[3]>20 and to_predict_list[3]<140):
                p1=p1+1
            else:
                p2=p2+1
            if(to_predict_list[4]>20 and to_predict_list[4]<60):
```

```

        p1=p1+1
    else:
        p2=p2+1
    if(to_predict_list[5]>10 and to_predict_list[5]<15):
        p2=p2+1
    else:
        p2=p2+1
    if(to_predict_list[6]>10 and to_predict_list[6]<15):
        p1=p1+1
    else:
        p2=p2+1
    if(to_predict_list[7]>4.0 and to_predict_list[7]<5.4):
        p1=p1+1
    else:
        p2=p2+1
    if(to_predict_list[6]>10 and to_predict_list[6]<15):
        p1=p1+1
    else:
        p2=p2+1
    if(p1>p2):
        pred=0
    else:
        pred=1
    print("#####3")
    print(pred)
except:
    message = "Please enter valid Data"
    return render_template("home.html", message = message)

return render_template('predict.html', pred = pred)

@app.route("/malariapredict", methods = ['POST', 'GET'])
def malariapredictPage():
    if request.method == 'POST':
        try:
            if 'image' in request.files:
                img = Image.open(request.files['image'])
                img = img.resize((36,36))
                img = np.asarray(img)
                img = img.reshape((1,36,36,3))
                img = img.astype(np.float64)
                model = load_model("models/malaria.h5")
                pred = np.argmax(model.predict(img)[0])
        except:
            message = "Please upload an Image"
            return render_template('malaria.html', message = message)
    return render_template('malaria_predict.html', pred = pred)

@app.route("/pneumoniapredict", methods = ['POST', 'GET'])
def pneumoniapredictPage():
    if request.method == 'POST':
        try:

```

```

if 'image' in request.files:
    img = Image.open(request.files['image']).convert('L')
    img = img.resize((36,36))
    img = np.asarray(img)
    img = img.reshape((1,36,36,1))
    img = img / 255.0
    model = load_model("models/pneumonia.h5")
    pred = np.argmax(model.predict(img)[0])
except:
    message = "Please upload an Image"
    return render_template('pneumonia.html', message = message)
return render_template('pneumonia_predict.html', pred = pred)

if __name__ == '__main__':
    app.run(debug = True)

```