Tribhuvan University

Faculty of Humanities and Social Sciences

HEART DISEASE PREDICTION SYSTEM USING LOGISTIC REGRESSION

A PROJECT REPORT

Submitted to:

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Ram Mahat (6-2-410-299-2019)

August, 2023 A.D

Under the supervision of

Mr. Pratik Chand

Tribhuvan University

Faculty of Humanities and Social Sciences

HEART DISEASE PREDICTION SYSTEM USING LOGISTIC REGRESSION

A PROJECT REPORT

Submitted to:

Department of Computer Application

Prime College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Ram Mahat (6-2-410-299-2019)

August, 2023 A.D

Under the supervision of

Mr. Pratik Chand

Tribhuvan University

Faculty of Humanities and Social Science

Prime College

SUPERVISOR RECOMMENDATION

I hereby recommend that this project prepared under my supervision by Ram Mahat entitled "**Heart Disease Prediction System Using Logistic Regression**" in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

SIGNATURE

Pratik Chand

Lecturer

Bachelor in Computer Application

Prime College

Tribhuvan University

Faculty of Humanities and Social Sciences

Prime College

LETTER OF APPROVAL

This is to certify that this project prepared by Ram Mahat entitled “Heart Disease Prediction System Using Logistic Regression” in partial fulfillment of the requirements for the degree of Bachelor in Computer Applications has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Pratik Chand, Lecturer Bachelor in Computer Application Prime College, Khusibu, Kathmandu	Ms. Rolisha Sthapit Bachelor in Computer Application Prime College, Khusibu, Kathmandu
…………………………………….	…………………………………….

ABSTRACT

Early detection of heart disease plays a crucial role in reducing mortality rates and improving patient outcomes. This abstract presents a heart disease prediction system which is developed using MERN stack and python that focuses on utilizing the logistic regression algorithm for accurate identification of individuals at risk. The system incorporates a wide range of patient data, including demographics, medical history, and diagnostic test results, to develop a predictive model. The heart disease prediction system harnesses the power of logistic regression, a popular statistical modeling technique, to capture the relationships and patterns within the data. The algorithm allows for the estimation of the probability of heart disease occurrence based on the input features. Feature selection methods are employed to identify the most influential predictors, enhancing the system's efficiency and interpretability. The heart disease prediction system presented in this abstract offer healthcare professionals a valuable tool for early risk assessment. By leveraging logistic regression and comprehensive patient data, the system enables timely interventions, personalized preventive strategies, and optimized allocation of healthcare resources. This approach holds significant potential in reducing the burden of heart disease and improving overall patient well-being.

Keywords: MERN, Python, Logistic Regression Algorithm

ACKNOWLEDGEMENT

For the partial fulfilment of the project this semester I would like to express my sincere gratitude to everyone who has directly and indirectly helped me to develop this project. There were times when I thought that this project was too difficult or may be not possible, but I am thankful to my supervisor Mr. Pratik Chand who continuously inspired me to give my best and perform the project to complete it. I could barely do anything without the help of Ms. Rolisha Sthapit who has helped me throughout and find and reconcile my mistakes.

I would also like to include the continuous support of my classmates who were always inspiring me to continue with the project and reminded me of the competition every time. I have also taken help from the seniors of Prime college so I would like to acknowledge them and really thank them from the bottom of my heart. It is because of all of you that I did not stop and give my best that has today brought the project here today.

In the end, I would also like to thank the Tribhuvan University for giving me this opportunity via the course of Computer Application to help me understand the project ethics at this early stage and helped me to evaluate my knowledge and expand it a little more.

Ram Mahat

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CSS- Cascading Style sheets

DFD- Data Flow Diagram

ERD- Entity Relationship diagram

HTML- Hypertext Markup Language

JS- JavaScript

LR – Logistic Regression

MERN – Mongo DB Express.js Rect.js Node.js

ML – Machine Learning

PHP- Hypertext Preprocessor

SQL- Structured Query Language

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Heart disease is a major global health concern, contributing to many deaths each year. Early detection and prevention of heart disease are crucial in reducing its impact and improving patient outcomes. In recent years, the integration of advanced technologies and data-driven approaches has shown promise in developing effective prediction systems. This introduction highlights a heart disease prediction system which will be developed using MERN (MongoDB, Express.js, React.js, Node.js) stack and Python, incorporating the logistic regression algorithm for accurate risk assessment.

The heart disease prediction system uses Logistic regression, a statistical modeling technique available in scikit-learn, is chosen as the predictive algorithm due to its interpretability and ability to handle binary classification tasks effectively and follows a data-driven approach, utilizing a comprehensive set of patient data including demographic information, medical history, and diagnostic test results. The system employs feature engineering techniques to extract meaningful features from the raw data, ensuring optimal performance of the logistic regression model. Furthermore, the system incorporates data preprocessing steps, such as handling missing values and scaling features, to enhance the quality and reliability of predictions.

The resulting heart disease prediction system offers healthcare professionals a powerful tool for early detection and risk assessment. By combining the advantages of modern web development technologies and machine learning, the system provides an intuitive and efficient interface to interact with patient data, enabling timely interventions and personalized preventive strategies. The system holds significant potential in reducing the burden of heart disease, improving patient outcomes, and optimizing healthcare resource allocation.

1.2 Problem Statement

The problem addressed in this project is the accurate and early prediction of heart disease. Current risk assessment methods often do not effectively use patient data and may not identify high-risk individuals promptly. The proposed heart disease prediction system aims to overcome these challenges by leveraging advanced machine learning algorithms and comprehensive patient data. The system seeks to develop a highly accurate predictive model that can identify individuals at risk of heart disease. By addressing these challenges, the heart disease prediction system can improve patient outcomes, optimize resource allocation, and reduce the burden of heart disease globally.

1.3 Objectives

- To develop a logistic regression model that can accurately identify the risk of heart disease.
- To enable users to schedule appointments with cardiologists who have heart disease.

1.4 Scope and Limitation

Scope:

- It can be used to identify individual risk of heart disease at an early stage, allowing them for timely intervention and preventive strategies.
- It can be integrated with existing healthcare infrastructures, such as electronic health records (EHRs) and clinical systems.
- It can also serve as a decision support tool for healthcare professionals, providing them with accurate risk assessments and predictive insights.

Limitations:

- Prone to cyberattacks because the system does not have a strong security layer and can breach the system.
- It requires stable internet connection.
- Prediction of heart disease gives average accuracy.

1.4 Development Methodology

A software development methodology or system development methodology is a framework used to structure, plan and control the process of developing a system. There are different models or methods used or followed during software development life cycle (SDLC) processes such as the waterfall model, prototyping model, spiral model, and others based on the nature or objective of the software.

As maximum requirements for the project were discussed and finalized before starting to work on the project and one stage would come after the completion of the previous steps, the author decided to use the incremental model for the completion of the project.

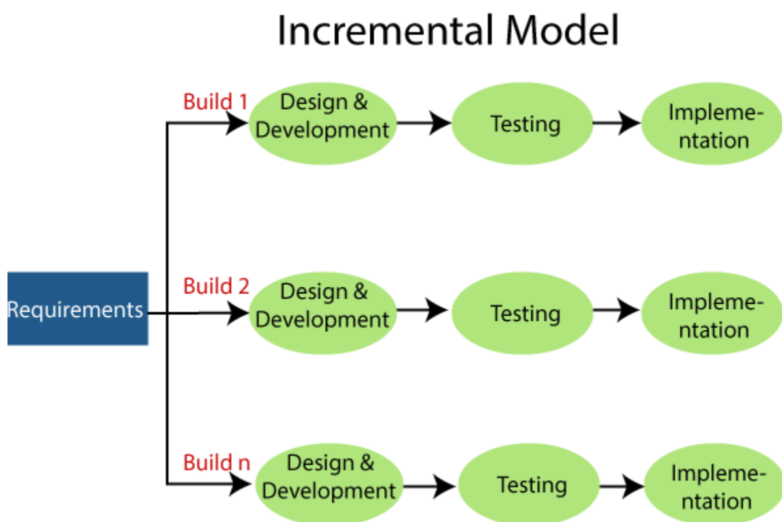


Figure 1.1 Incremental model

Source: https://www.sciencedirect.com/topics/computer-science/incremental-model

Requirement Analysis: This phase includes the gathering of all the requirements of the system. For this, some research was done, and some useful insights were pointed out so that the requirements, the functions, and the objectives of the system were well known.

Design: It includes the study and designing of the requirements gathered in the analysis phase for the system development. Thus, the formed design was then converted into use case-diagram, data modeling -diagram, component diagram and deployment diagram where each diagram depicted the design in their unique representation.

Implementation: This phase includes the implementation of the design done for system development in the previous phase. The features of the system were developed one at a time and then all the features were integrated to form a complete system in the next phase. All the modules were developed using React.js, MERN stack and Python.

Testing: The main objective of this phase is to see if the system is functioning as intended or not. This phase includes unit testing and system testing under various test case scenarios. In unit testing, each unit is tested one by one whereas in system testing the system is tested.

1.5 Report Organization

The report starts with a brief introduction to heart disease with problem statements and project objectives. Chapter 2 analyses the existing system. Chapter 3 discusses the data modeling and process modeling techniques used to give the information about the system requirement and feasibility study. The system design can be database component diagram, deployment diagram and activity diagram. Chapter 4 explains about the tools that are used on this project's front end, back end, and purpose of it. The modules used are also explained. The testing is also explained in this part. Chapter 5 discusses the conclusion of how the project is accomplished, its findings and many more. It also discusses the recommendation for future enhancements of the project. In conclusion, this chapter overviews the purpose of doing this project including its scope and objectives.

CHAPTER 2

BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background study

Heart disease is a major global health concern, leading to many deaths worldwide. The need for early detection and prevention of heart disease has prompted research in the development of prediction systems. Traditional risk assessment methods, such as the Framingham Risk Score, have limitations in utilizing comprehensive patient data and may not be suitable for early detection. In recent years, machine learning techniques have gained popularity in developing heart disease prediction models. Algorithms such as logistic regression, decision trees, random forests, support vector machines, and neural networks have been utilized, leveraging features including demographic information, medical history, laboratory test results, and imaging data. Feature engineering and selection techniques have been explored to extract meaningful features from raw patient data. Integration of advanced technologies, such as electronic health records (EHRs), wearable devices, and mobile health applications, has also been investigated to enhance risk assessment. Evaluation metrics such as accuracy, sensitivity, specificity, and AUC-ROC are used to assess model performance. Ethical considerations related to data privacy, informed consent, and algorithmic bias are important aspects to address in the development and implementation of heart disease prediction systems. Overall, the background study emphasizes the significance of accurate and reliable heart disease prediction systems in improving patient outcomes and the need for ongoing research in this field.

2.2 Literature Review

Till date different studies have been done on heart disease prediction. Various data mining and machine learning algorithms have been implemented and proposed on the datasets of heart patients and different results have been achieved for different techniques. But, still today we are facing a lot of problem faced by the heart disease. Some of the recent research papers are as follows:

In 2010, Authors applied machine learning algorithms such as Naive Bayes, KNN (K- nearest neighbors) and decision list for heart disease prediction. Tanagra tool is used to classify the data and the data evaluated using 10-fold cross validation and the results are compared in table 4. The data set consists of 3000 instances with 14 different attributes. The dataset is divided into two parts, 70% of the data are used for training and 30% are used for testing. The results of comparison are based on 10-fold cross validation. Comparison is made among these classification algorithms out of which the Naive Bayes algorithm is considered as the better performance algorithm. Because it takes less time to build model and also gives best accuracy as compared to KNN and Decision Lists [1].

Author developed a Decision Support in Heart Disease Prediction System (DSHDPS) using data mining modeling technique, namely, Naive Bayes. Using heart disease attributes such as chest pain, age, sex, cholesterol, blood pressure and blood sugar can predict the likelihood of patients getting a heart disease. It is implemented as web-based questionnaire application. Historical data set of heart patients from Cleveland database of UCI repository was used to train and test the Decision Support System (DSS). The reasons to prefer Naive Bayes machine learning algorithm for predicting heart disease are as follows: when data is high, when the attributes are independent of each other and when we want to achieve high accuracy as compared to other models. When the dimensionality of the inputs is high in that case Naive Bayes classifier technique is particularly suited. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods [2].

In 2013, Authors, applied data mining and machine learning algorithms namely Decision Tree (J48 algorithm), Naive Bayes and Artificial Neural Networks (ANN) for heart disease prediction. A dataset of 7339 instance with 15 attributes has been taken from PGI Chandigarh. WEKA 3.6.4 tool was used for the experiment. For model training and testing 10-Fold Cross

Validation techniques is used randomly. Best First Search method was used to select the best attributes from the already available 15 attributes and among them only 8 attributes has been selected. Each experiment was done on two different scenarios, first one containing all 15 attributes and the second case only 8 selected attributes. From all these experiments comparative results has been obtained and from these comparative results it has been found that J48 pruned in selected attributes case has performed best in accuracy with 95.56% and Naive Bayes with all attributes case gives less accuracy 91.96% but takes least time to build a model in the whole experiment [3].

In 2014, Authors applied data mining algorithms such as J48, Naive Bayes, REPTREE, CART, and Bayes Net in this research for predicting heart attacks. The patient dataset is collected from medical practitioners in South Africa. Only 11 attributes namely Patient Id, Gender, Cardiogram, Age, Chest Pain, Blood Pressure Level, Heart Rate, Cholesterol, Smoking, Alcohol consumption and Blood Sugar Level from the database are considered for the predictions required for the heart disease. WEKA data mining tool is used for experiments. The algorithms were applied on the data set using 10-fold cross validation technique in order to calculate average accuracy of all folds for each classification technique to predict a class. From the results it has been seen J48, REPTREE and SIMPLE CART algorithm performs best in this data set, while Bayes Net algorithm outperformed the Naïve Bayes algorithm [4].

In 2016, Author applied three machine learning algorithms viz Naïve Bayes, J48, and Artificial Neural Network (ANN) to achieve best accuracy in heart disease prediction for male patients. A dataset of 210 records with 8 attributes has been used in this experiment. In order to carry out experiments and implementations WEKA was used as the data mining tool. From the experiments comparative results has been drawn in table 8 and from the comparative result has been found that Naïve Bayes performed best as compared to J48 and ANN to predict heart disease with an accuracy of 79.9043% and takes less time 0.01 seconds to build a model [5].

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

System analysis is a process of studying a system or organization to understand its components, how they can be improved. It is a holistic approach that looks at the system and identifies the relationships between its parts. The goal of the system analysis is to identify problems and inefficiencies in the current system and to propose solutions for improvement.

3.1.1 Requirement Analysis

i. Functional Requirement

a) Patients: This application allows patients to login, register and predict heart disease.

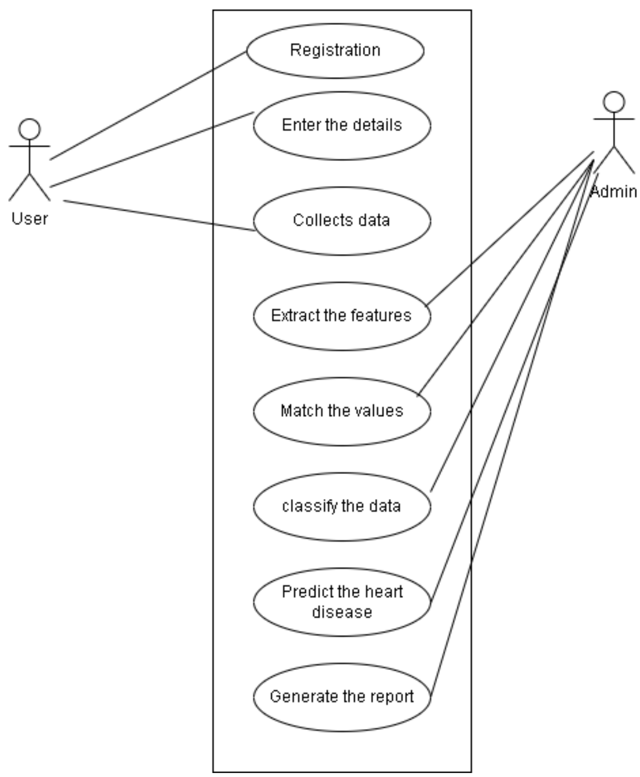


Figure 3. 1 Use Case Diagram of Patient

b) Doctors:

This application allows doctors to register, login, and help in viewing the patients' records and information. Doctor can also accept or reject appointments too.

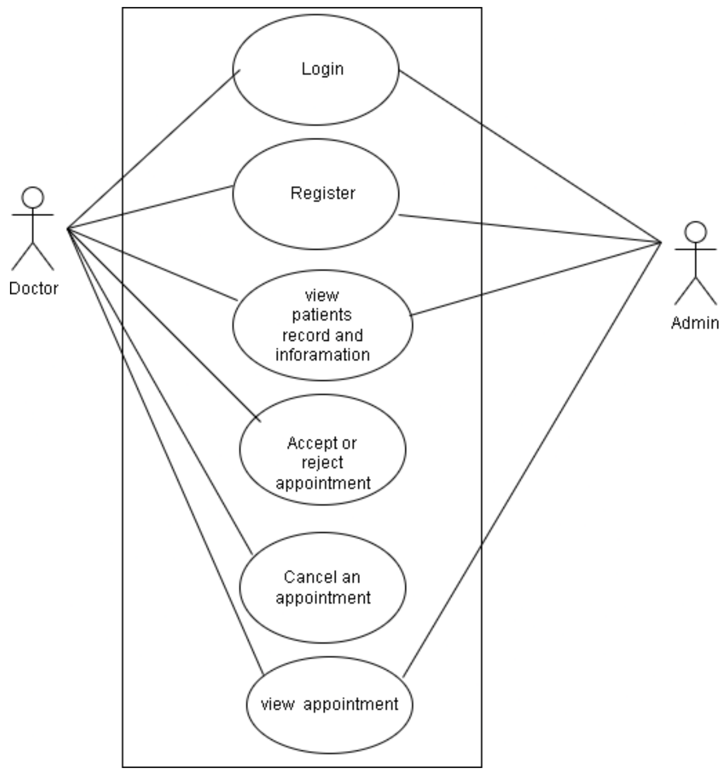


Figure 3. 2 Use Case Diagram of Doctor

- **a. Patient Registration**: Registration is open to all the visitors to e-Healthcare. Patients can register to the online system by filling up the form and entering the correct details.
- b. **Patient Login:** (general user/admin): Patient can login with username and password. Patient then selects appropriate login credentials to access their profile.
- **c. Take appointments:** General patients can take appointments by going to the specific doctor that the patients want to take appointments from, and schedules time for the appointments. General patients can view their appointment history and their details.
- **d. Cancel Appointments:** Patient and doctor themselves can cancel appointments in case of necessity.
- **e. Change Appointments:** Patient can change appointments in case of necessity.

f. **View all appointments for specific doctor:** Admin and doctor both can view the appointments taken by patients from specific doctor.
g. **View patients record information:** Doctor and admin can view patients record and information in case of any necessity.

ii. Non-functional Requirement

a. **User Friendly:** User Friendly is self-explanatory. When something is user friendly then it is easy to access and work with it. E-healthcare is user friendly. A person having basic knowledge and skills of computers can also easily use the web application.

b. **Simple and Easy to use:** Heart Disease Prediction System uses a simple design as well as simple language on the content to improve the user friendliness of the web application.

c. **Responsiveness**: Heart Disease Prediction System is responsive This nature could be very beneficial to people living in areas with limited access to computers.

3.1.1 Feasibility Analysis

The feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily made based on the future upcoming requirements.

i. Economic Feasibility

This is a particularly important aspect to be considered while developing a project. The author decided the technology based on the minimum possible cost factor.

- All the hardware and software cost have to be borne by the organization.
- The benefits the organization is going to receive from the proposed system will surely overcome the initial costs and later on running cost for the system.
- Also. the cost of the development of this system will be minimum which will benefit both users and developers.

ii. Technical Feasibility

It is an evaluation of the hardware and software and how it meets the needs of the proposed system. This includes the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, and checked if everything was possible using different types of frontend and backend platforms.

Table 3.1 Hardware Requirements:

SN	Hardware	Minimum System Requirements
1	Processor	2.4 GHz Processor speed
2	Memory	8 GB RAM
3	Disk Space	500GB

Table 3.2 Software Requirements:

SN	Software	Minimum Software Requirements
1	Operating System	Windows 10/Linux
2	Database Management System	Mongo DB
3	Runtime Environment	Visual Studio 2008

iii. Operational Feasibility

Operational feasibility is the measure of how well the project will support the customer and the service provider during the operational phase. No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory. Besides, proper training has been conducted to let the users know the essence of the system so that they feel comfortable with the new system. As far as our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

iv. Schedule Feasibility

This is one of the most important feasibility analyses as it helps an organization to estimate how much time the organization will take to complete the project and how much of it is on track to a given schedule.

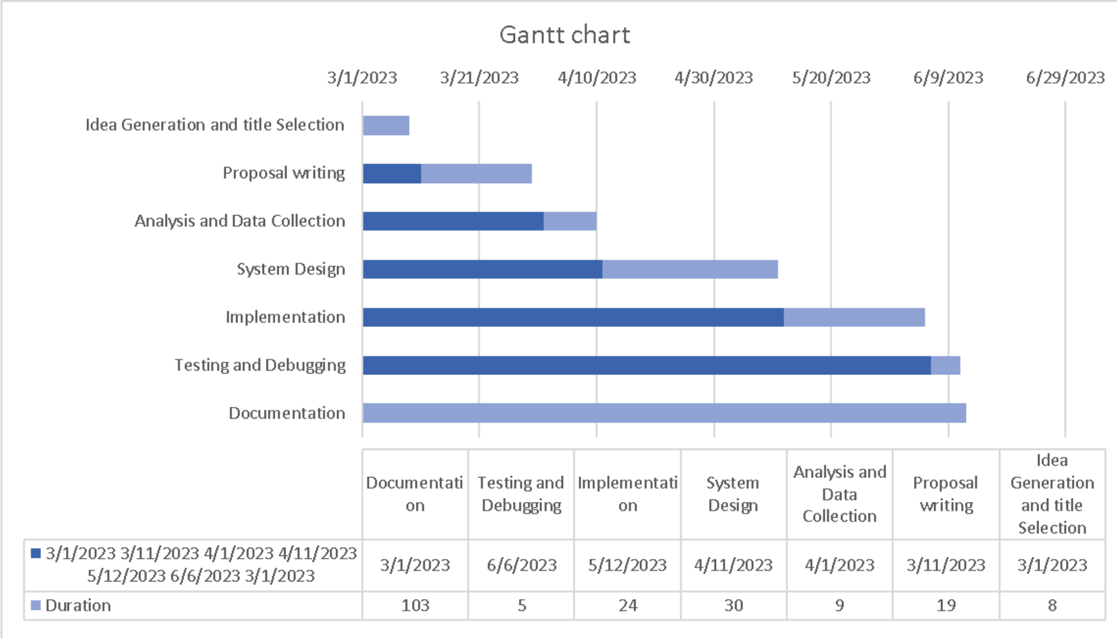


Figure 3.3 Gantt Chart

3.1.2 Object Modeling: Object and Class Diagram

A data model is a mechanism. Class diagrams are the type of UML diagram that describe a system's structure through the system classes, their attributes, operations and their relationship. Class Diagrams are also used for data modelling and each class in the diagram represents both main elements as well as its interaction with the environment.

The class diagram of Heart Disease Prediction System consists of 4 classes named Doctor, User, Admin and Appointments. Each class has their own attributes and since they are private, so it is represented by ‘- ‘. Each attribute is also followed by its type. The relationship between classes is shown by a solid line or dotted line.

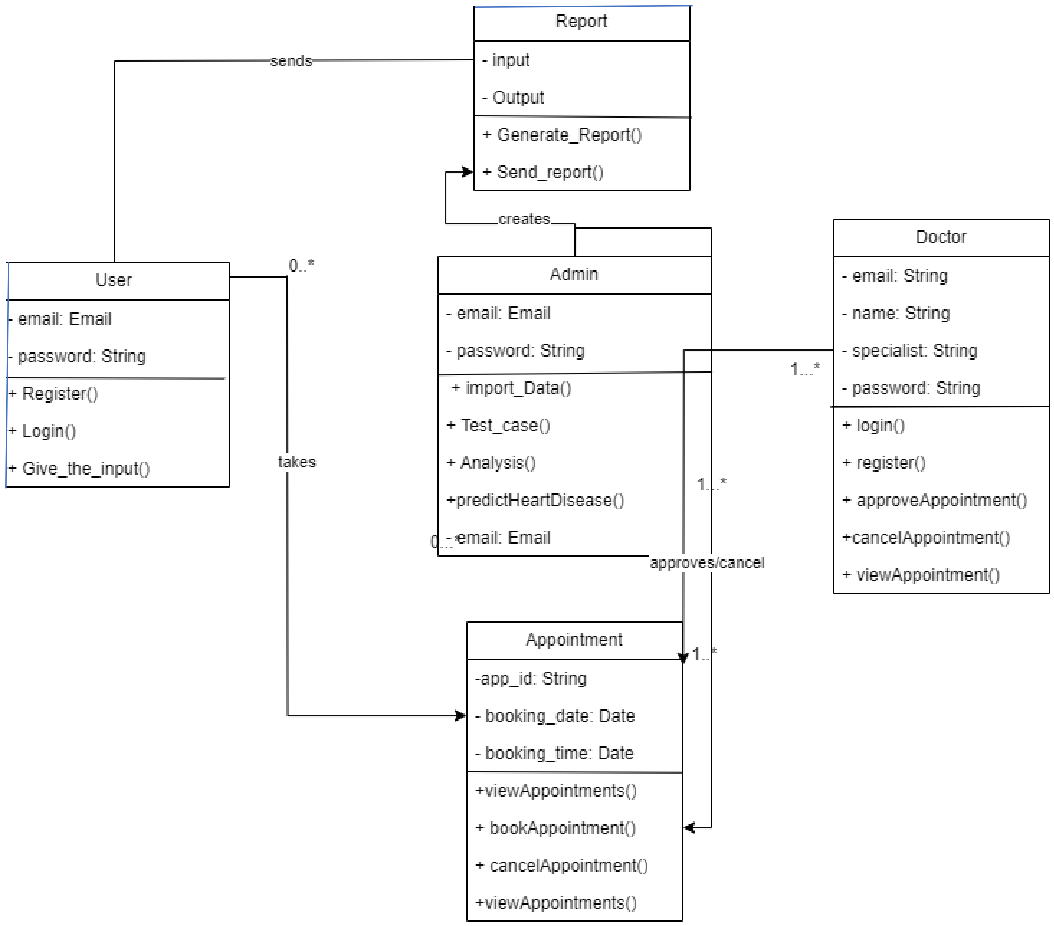


Figure 3.4 Class Diagram

3.1.3 Dynamic Modeling

Dynamic modeling can be referred to those aspects of the system that are concerned with the time and sequencing of the operations. State Diagrams describe the behavior of the system through sequence of finite state transitions. The above diagram represents the state diagram of Heart Disease Prediction System which starts through user inputs and ends up predicting the heart disease of the patient and taking appointments from the cardiologist.

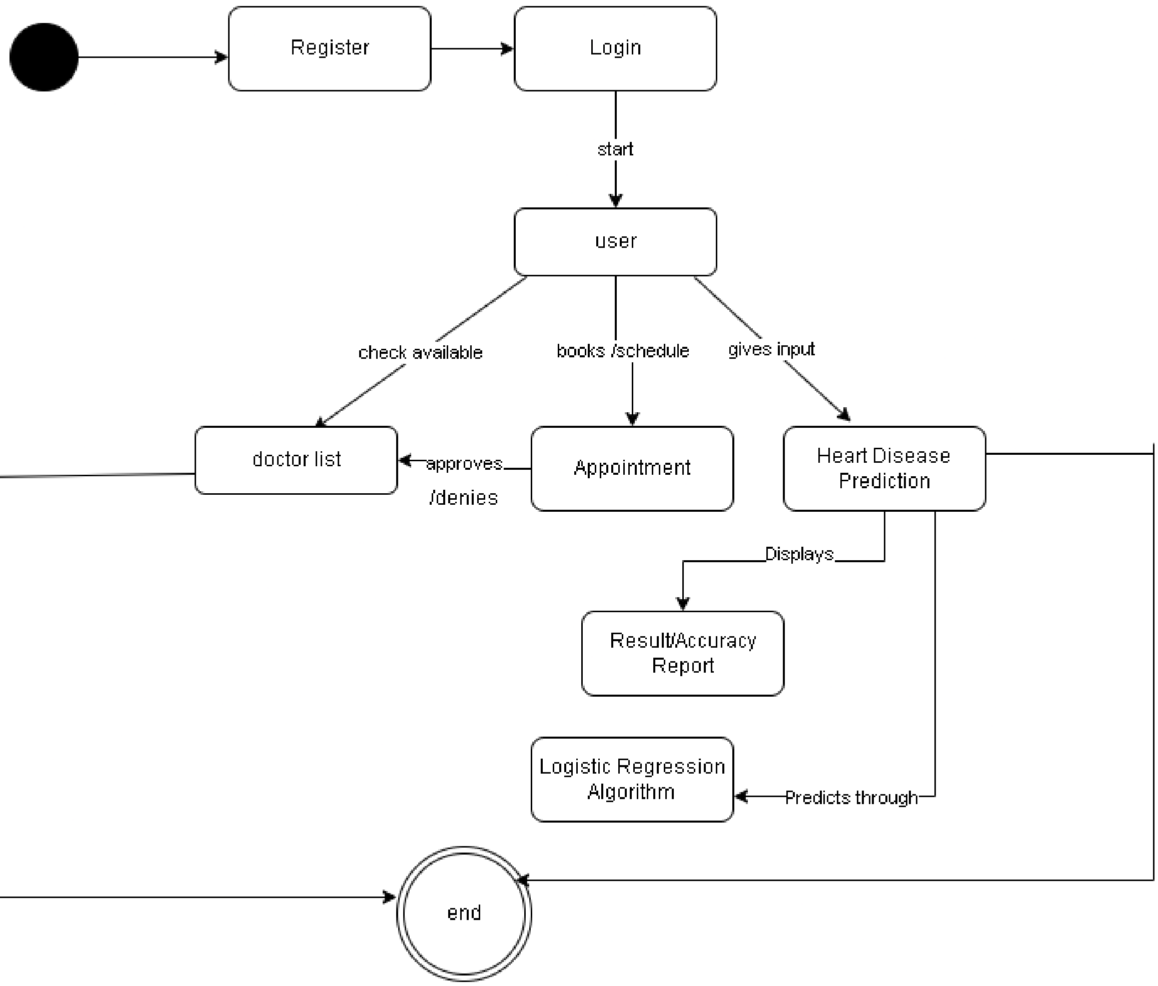


Figure 3.5 State Diagram

A sequence diagram is the interaction between the objects in a sequential order. Sequence diagrams describe how and in what order the objects in a system will function. Sequence diagram helps in understanding the existing or new system better.

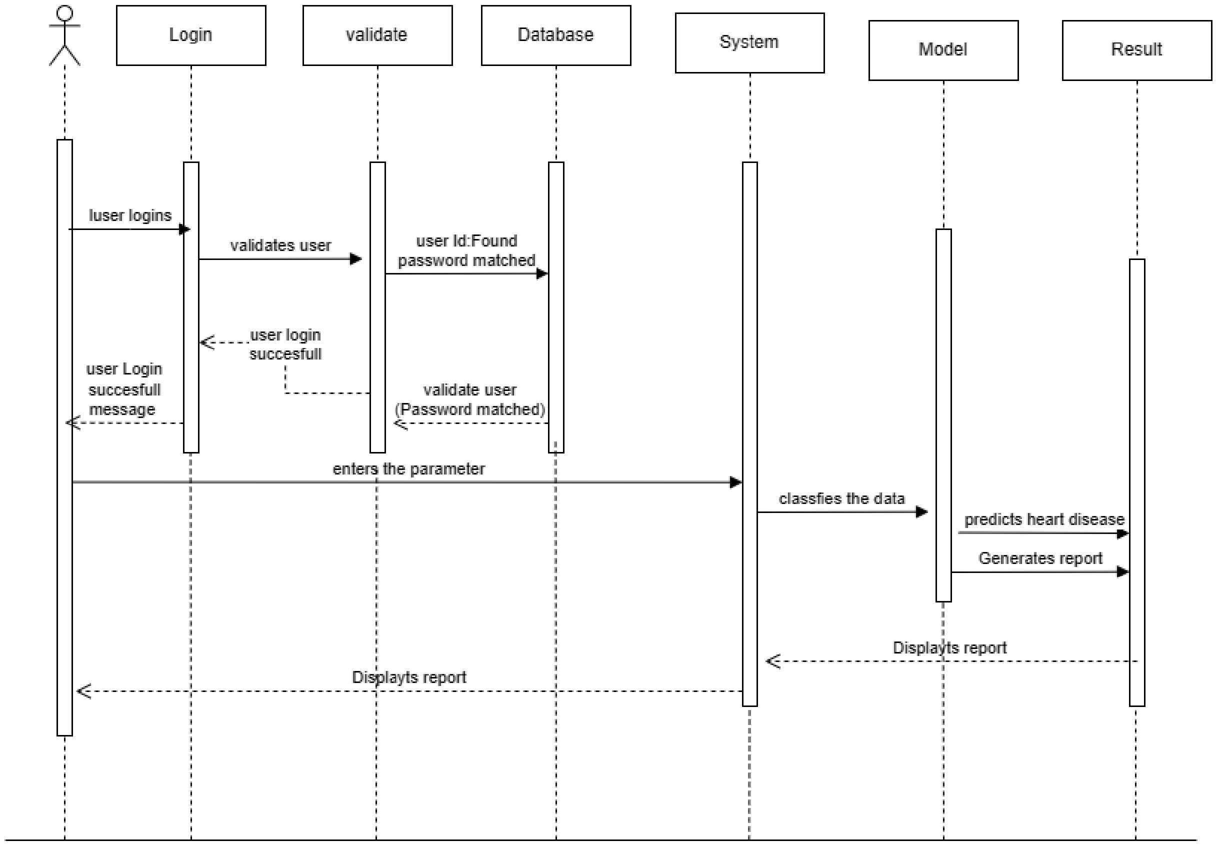


Figure 3.6 Sequence Diagram

3.1.4 Process Modeling: Activity Diagram

An activity diagram is basically a flowchart used to represent the flow from one activity to another. It helps to represent the dynamic perspective of the system. The activity diagram below shows how the working of the activity of a project comes together to predict the heart disease after the user successfully logins to the system.

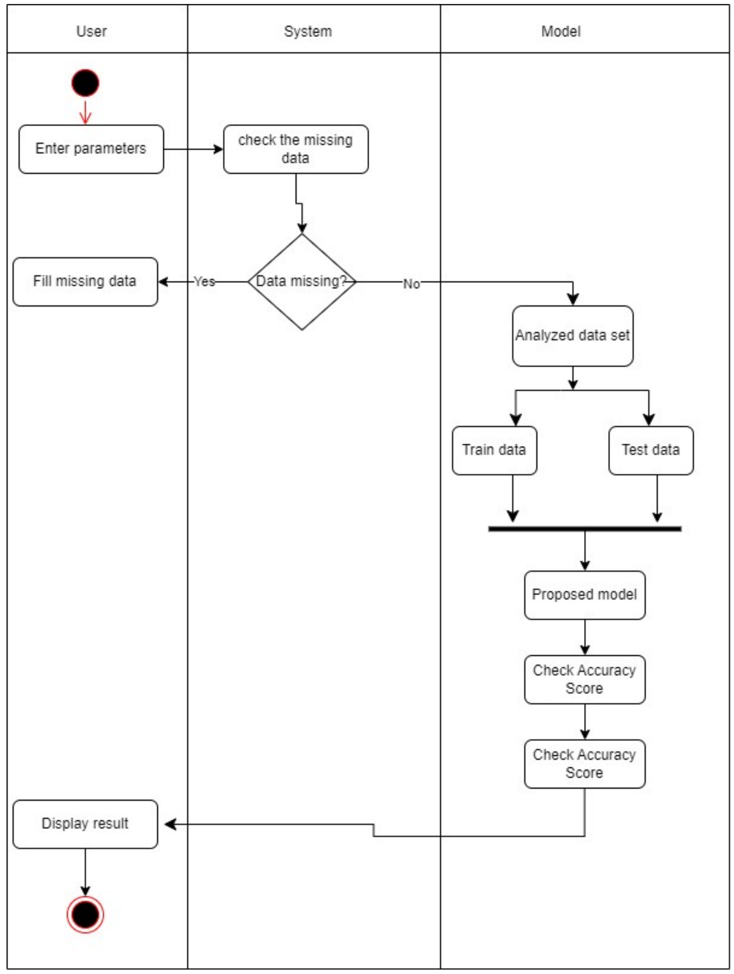


Figure 3.7 Activity Diagram

3.2 System Design

System Architecture is a framework that incorporates the interactions and relationships between application components such as databases, middleware systems and user interfaces. The system design servers the purpose of planning the way of how a system is to be implemented in order to solve the problem efficiently. The system design phase is further divided into smaller sub-phases which work together to achieve the goal of the system.

3.2.1 Architectural Design

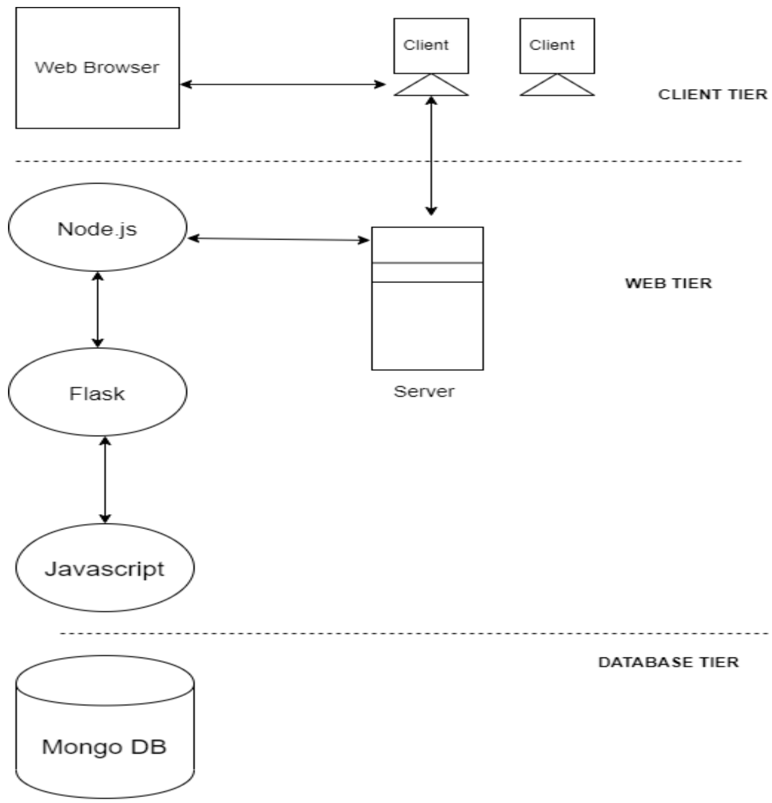


Figure 3.8 Architectural Design

This heart disease prediction system is three tier architecture that consists of client tier, web tier and database tier. Client basically refers to user who runs the system on specific browser and the web tier contains NodeJS, flask and JavaScript which runs on backend. For the application to run, it needs to run express server and flask server on the backend and in the database tier it used mongo dB cloud.

3.2.2 Component Diagram

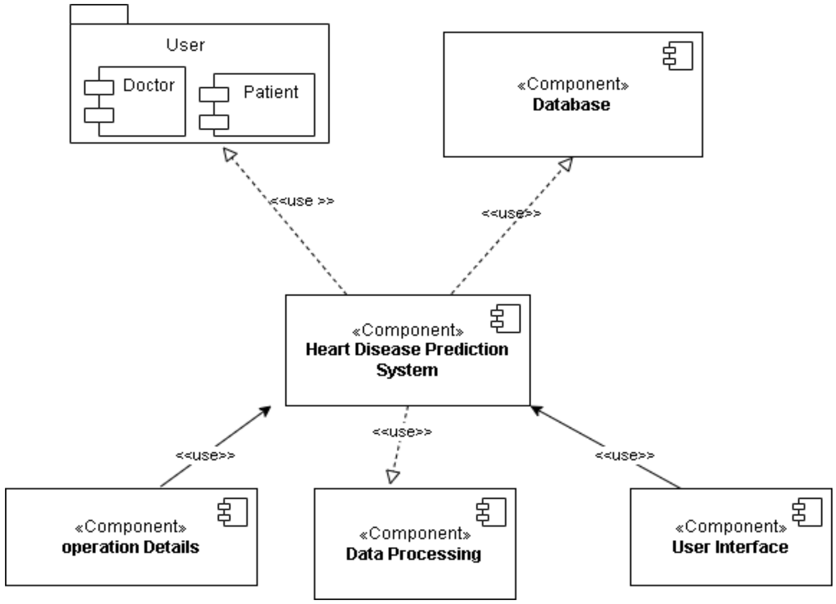


Figure 3.9 Component Diagram

User Component diagrams are the static implementation views of a system that does not describe the functionality of the system, but it describes the components used to make those functionalities. The components used in the system are user, database, data processing and heart disease prediction respectively.

3.2.3 Deployment Diagram

A deployment diagram shows the execution architecture of a system. The diagram represented above shows the deployment diagram of the system. It contains two nodes server side and client side respectively and inside client side there is internode which is web browser. The user gives input parameters through which the Heart Disease Prediction system provides the predicted output.

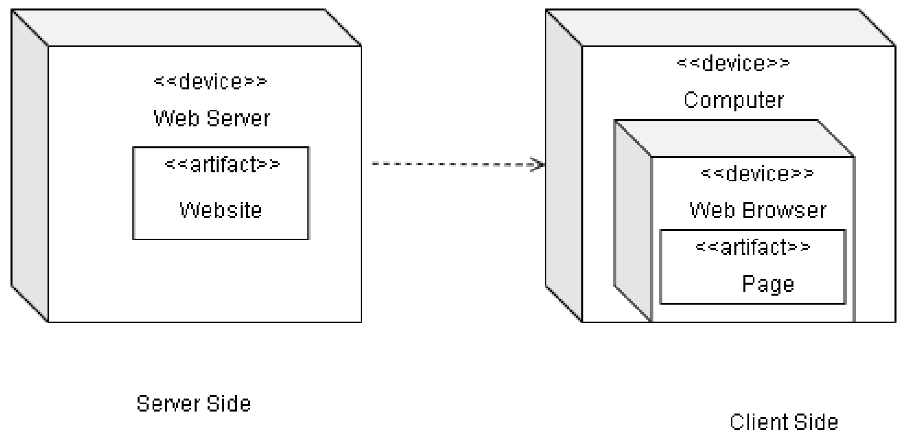


Figure 3.10 Deployment Diagram

3.1 Algorithm Details

Logistic Regression Algorithm

The logistic regression algorithm is a supervised learning algorithm used for classification tasks, where the goal is to predict the probability of an instance belonging to a certain class or category. It is widely used in various fields such as finance, healthcare, marketing, and social sciences.

Here's a step-by-step overview of the logistic regression algorithm:

Given a training set X = x1, ..., xn with responses Y = y1, ..., yn, logistic regression estimates the probabilities of binary outcomes by fitting a linear model to the data. The logistic regression algorithm follows these steps:

Set the number of iterations B and initialize an empty set of coefficients $\beta_0, \beta_1, \beta_2, ..., \beta_p$.

For each iteration (b = 1, ..., B), the logistic regression algorithm randomly selects a subset of the training data with replacement, creating a new training set Xb, Yb. Each sample in Xb is paired with its corresponding response in Yb and train a logistic regression model, denoted a

fb, using the new training set Xb, Yb. The model estimates the coefficients $\beta_0, \beta_1, \beta_2, ..., \beta_p$ that maximize the likelihood of the observed data. After training B logistic regression models, predictions for unseen samples x' can be made by averaging the predictions from all the individual models. For a given instance x', the predicted probability of the outcome can be calculated as:

$p = 1/B * \sum(fb(x'))$

where fb(x') represents the prediction of the b-th logistic regression model for the instance x'. To classify instances into specific classes, a decision threshold can be set. If the predicted probability exceeds the threshold, the instance is assigned to one class; otherwise, it is assigned to the other class. Logistic Regression Algorithm predicts the output in the following ways:

Step (1): Start by preprocessing the data, handling missing values, and performing feature scaling or normalization if necessary.

Step (2): Using the training data, the logistic regression model estimates the coefficients (β_0, β_1, β_2, ...) through an optimization algorithm such as maximum likelihood estimation. This process involves finding the optimal values for the coefficients that maximize the likelihood of the observed data.

Step (3): The logistic regression model assumes a linear relationship between the features and the log-odds of the outcome. It uses the logistic function (sigmoid) to map the linear combination of the features to a probability value between 0 and 1.

Step (4): Once the model coefficients are estimated, the log-odds (logit) of the outcome can be calculated for a given instance using the following equation:

$\text{logit}(p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$

where p is the probability of the outcome, β_0 is the intercept term, and β_1, β_2, ..., β_p are the coefficients for the corresponding features x_1, x_2, ..., x_p.

Step (5): To obtain the predicted probability of the outcome, the logistic regression model applies the sigmoid (logistic) function to the calculated log-odds:

$p = 1 / (1 + \exp(-\text{logit}(p)))$

This maps the log-odds to a probability value between 0 and 1.

Step (6): Depending on the specific problem and requirements, a decision threshold can be set to classify the instance into a specific class. For example, if the threshold is set at 0.5, instances with predicted probabilities above 0.5 are assigned to one class, while those below 0.5 are assigned to the other class.

Step (7): Finally, the logistic regression model predicts the output by comparing the predicted probability to the decision threshold. If the predicted probability exceeds the threshold, the instance is classified as belonging to one class; otherwise, it is classified as belonging to the other class.

CHAPTER 4

IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1. Tools Used (CASE tools, Programming languages, Database platforms)

4.1.1.1 Front End Tools

React.js: React.js is used to design the front-end views of this system.

CSS: CSS is used for describing the presentation of the front-page including layout,colors

JavaScript: JavaScript web framework is one of the best ways to stack backend and frontend frameworks and has been used for the same in the project.

4.1.1.2 Back End Tool

Node.js: Node.js is a JavaScript runtime environment that allows developers to build server-side applications. It utilizes an event-driven, non-blocking I/O model, making it highly efficient and scalable. Node.js enables the use of JavaScript on both the client and server sides, facilitating seamless communication and code sharing.

Express.js: Express.js is a web application framework built on top of Node.js. It provides a simple and minimalist approach to building web servers and APIs. Express.js offers a range of features, including routing, middleware support, and template engine integration, allowing developers to create scalable and maintainable web applications with ease.

MongoDB: MongoDB is a NoSQL database system that uses a flexible, document-based data model. It stores data in JSON-like documents, providing scalability and high performance for handling large amounts of data. MongoDB's schema-less nature allows for dynamic and flexible data structures, making it well-suited for applications that require frequent updates and evolving data models.

Flask: Flask, on the other hand, is a micro-framework for Python, designed for simplicity and extensibility. It provides the necessary tools to build web applications and APIs using Python. Flask offers routing, request handling, and template rendering capabilities, making it a lightweight and efficient choice for developing web applications.

4.1.2. Implementation Details of Modules (Description of procedures/functions)

There are different module descriptions. They are described below:

Sources of Data:

Dataset:

This system uses single dataset for the module Myenv.

Heart Disease Dataset:

The dataset used for heart disease prediction is ‘Heart Disease Dataset’ which was extracted from Kaggle [6] and it consists of 22 labels and 304 tuples. The labels in the dataset are:

1. Name of The Patient
2. Age: Age of patients in years.
3. Sex: The gender of the patient (1 refers to female and 0 to male).
4. Smoking: If the patient smoke or not (0=No, 1=Yes).

**5. Years: **Number of years of smoking if smoker.

6. LDL: LDL-Cholesterol ratio of the patient.
7. Chp: Chest pain type (1=Typical angina, 2=Atypical angina, 3=non-anginal pain,4=Asymptomatic).
8. Height: The height of the patient in cm.
9. Weight: The weight of patients in kg.
10. FH: Family history of heart disease.
    Active: If the patient is active or not (0=No, 1=Yes).
11. Lifestyle: The place of living (1=City, 2=Town, 3=Village).
12. CI: Does the patient has any cardiac catheterization or any intervention into the heart? (0=No, 1=Yes).
13. HR: Heart Rate ratio.
14. DM: Does the patient has diabetes (0=No, 1=Yes).
15. Bpsys: The ratio of Systolic Blood Pressure.
16. Bpdias: The Diastolic ratio of Blood Pressure.
17. HTN: Does the patient suffer from hypertension (0=No, 1=Yes).

18. IVSD: An Echo parameter (Interventricular Sepal and Diastole). IVSD is a measurement that is used to determine Left Ventricular Hypertrophy (LVH).
19. ECGpatt: Contains four categories of an ECG test which are (1=ST-Elevation, 2=ST-Depression, 3=T-Inversion, 4=Normal).
20. Qwave: The presence of the Q wave (0=No, 1=Yes).
21. Target: If the patient suffers from heart disease or not (0=without heart disease, 1=with heart disease).

Client Module:

This module is the front-end part which contains many directories. It contains node modules where npm packages are installed. Likewise, inside the public folder it contains the index.html file .Src folder contains all the code of the react which contains jsx applications. Likewise, inside the data, it contains data for the user, admin, and doctor. The image folder contains different images required in the system. The page folder contains the primary pages like the heart disease prediction page, appointment page etc. Likewise, in the styles folder it contains CSS files where all the designing part is carried out. Most importantly, app.js is the core component of the react application.

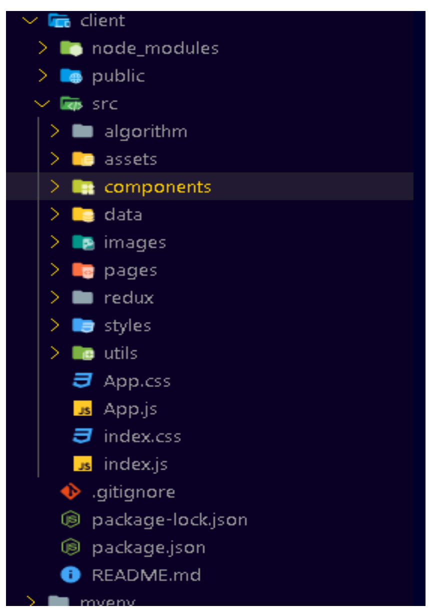


Figure 4.1 Client Module

Server Module:

This module is completely dedicated to the backend logic of the system. Whenever user, admin, doctor tries to login or register, it validates and save their data to the database. It also authenticates and authorizes the user based on their roles. Likewise, when user wants to know if he/she has heart disease or not, this module invokes the flask application where machine learning model is deployed and predicts the heart disease of the patient based on the input received from the frontend.

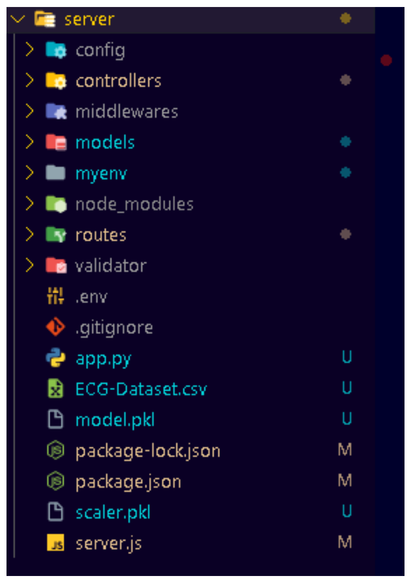


Figure 4.2 Server Module

Myenv Module:

This module contains the python code where machine learning model is trained, tested, and deployed using flask. It comprises the code for the logistic regression algorithm which helps predict the patient's heart disease. This module gets invoked using axios and child module package of npm in the express.js server.

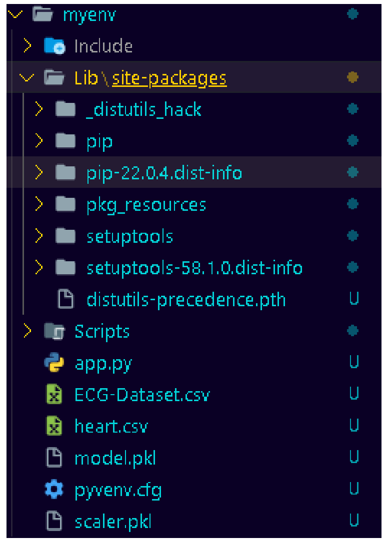


Figure 4.3 Myenv Module

```python
# Define the logistic regression algorithm with regularization
class LogisticRegressionScratch:
    def __init__(self, learning_rate=0.01, num_iterations=1000, lambda_val=0.01):
        self.learning_rate = learning_rate
        self.num_iterations = num_iterations
        self.lambda_val = lambda_val

    def sigmoid(self, z):
        return 1 / (1 + np.exp(-z))

    def fit(self, X, y):
        self.theta = np.zeros(X.shape[1])

        for _ in range(self.num_iterations):
            z = np.dot(X, self.theta)
            h = self.sigmoid(z)
            gradient = (np.dot(X.T, (h - y)) + self.lambda_val * self.theta) / y.size
            self.theta -= self.learning_rate * gradient

    def predict(self, X):
        z = np.dot(X, self.theta)
        return self.sigmoid(z)


# Train the logistic regression model from scratch
model_scratch = LogisticRegressionScratch(
    learning_rate=0.1, num_iterations=1000, lambda_val=0.1
)
model_scratch.fit(X_train_scaled, y_train)
```

Figure 4.4 Using Logistic Regression to predict and generate accuracy score

In Figure 4.4 after dataset is taken then it is split into features and target, respectively where X represents the removal of target parameter and Y represents the target parameter. Again, data is split into training and testing sets where there are altogether four variables are stored namely X_train, X_test, y_train, y_test respectively. Then data is preprocessed using StandardScaler function where X_train_scaled is transormed and X_test_scaled is also transformed. The model is trained afterwards using Logistic Regression function and fitted using the fit function. Finally, accuracy score is generated afterwards . For generating accuracy score , y_pred_train variable is used to predict and train_accuracy holds the accuracy information in percentage using accuracy_Score function.

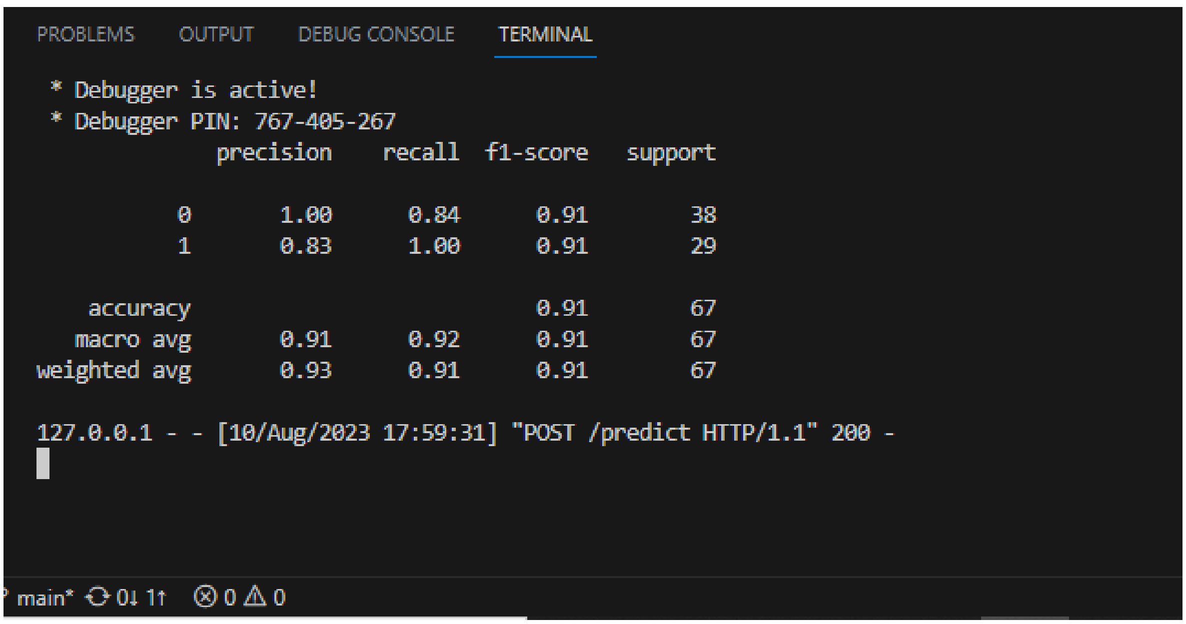


Figure 4.5 Generating classification report.

In Figure 4.5, classification report function is used to generate the classification report by putting the parameters y_test, y_pred_test, target_name which have already been defined in the above code respectively.

```
#  generate confusion matrix
cm = confusion_matrix(y_test, y_pred_test)
print(cm)
```

```
[[32  6]
 [ 2 27]]
```

```
cm = confusion_matrix(y_test, y_pred_test)
labels = ['0', '1']  # Assuming the labels are '0' and '1'
# Plot the confusion matrix
fig, ax = plt.subplots()
im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.set(xticks=np.arange(cm.shape[1]),
       yticks=np.arange(cm.shape[0]),
       xticklabels=labels, yticklabels=labels,  # Add label names
       xlabel='Predicted Label', ylabel='True Label',  # Update label names
       title='Confusion Matrix')

# Loop over data dimensions and create text annotations
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, str(cm[i, j]),
                ha="center", va="center",
                color="white" if cm[i, j] > np.max(cm) / 2 else "black")

fig.tight_layout()
plt.show()
```

Figure 4.6 Code for Generating confusion matrix.

In Figure 4.6, confusion matrix function is used to generate confusion matrix by taking two parameters y_test and y_pred_Test respectively. Since the labels are 0 and 1 in.System where 1 represents present of specific parameters defined in the system and 0 represents specific parameter is absent. Fig and ax are defined to plot the confusion matrix. Then label names are added and updated with “True label” and “Confusion Matrix, respectively. Then code loops over data dimensions and creates text annotations. Finally, the confusion matrix is shown by using the show function.

```
# Loop over data dimensions and create text annotations
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, str(cm[i, j]),
                ha="center", va="center",
                color="white" if cm[i, j] > np.max(cm) / 2 else "black")

fig.tight_layout()
plt.show()
```

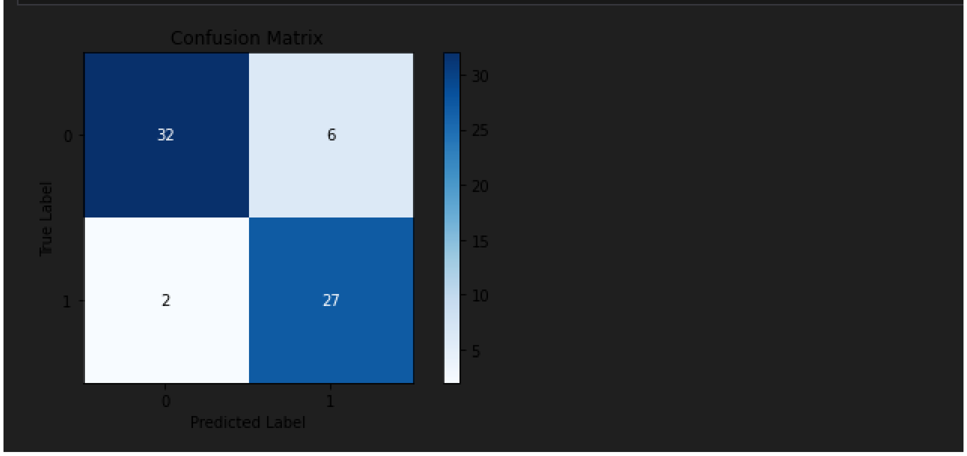


Figure 4.7 Code for showing Confusion matrix.

In the Figure 4.7 the confusion matrix is shown by the help of the show function by looping over data dimensions and creating text annotations.

4.2 Testing

Testing is the process of detecting errors. It performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later during maintenance also.

4.2.1 Test Cases of Unit Testing

Unit Testing is a software development process in which the smallest testable parts of an application called units are individually and independently scrutinized for proper operation. The system contains different types of individual parts that are tested. Some of the test cases are:

Table 4.1 User Registration

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Localhost:3000/	Login Page	Login Page	Pass
2	Submit without any Details	Null	Redirects to login page.	Redirects to login page.	Pass
3	Enter numeric value	Name: 123 Email: btycoon77@gmail.com Password: 123456789	Invalid name	Invalid name	Pass
4	Enter email of incorrect format	Name: Ram Email: b.com Password: 123456789	Please include @in the email	Please include @in the email	Pass
5	Enter correct details	Name: 123 Email: btycoon77@gmail.com Password: 123456789	Registration Successful	Registration Successful	Pass

Table 4.2 User Login

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Localhost:3000/	Login Page	Login Page	Pass
2	Submit without any Details	Null	This field cannot be empty	This field cannot be empty	Pass
3	Enter incorrect email	Email: btyn77@gmail.com Password: 123456789	Email does not exist	Email does not exist	Pass
4	Enter incorrect password	Email: btycoon77@gmail.com Password: 123456789	Password does not match	Password does not match	Pass
5	Enter correct details	Email: btycoon77@gmail.com Password: 123456789	Redirected to dashboard	Redirected to dashboard	Pass

Table 4.3 Edit User

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	http://localhost:3000/user /profile/userId	User profile	User profile	Pass
2	Submit without any Changes	Similar details	Redirected to dashboard	Redirected to dashboard	Pass
3	Change name	Previous Name: Ram New Name: Tycoon	Details updated	Details updated	Pass
4	Change password	Previous password: 123456789 New password: 123456	Redirected to dashboard	Redirected to dashboard	Pass
5	Change address	Previous address: Kirtipur New address: New York	Redirected to dashboard	Redirected to dashboard	Pass

Table 4.4 Heart Disease Prediction

Logistic Regression Algorithm				
SN	Action	Input	Output	Accuracy
1	Launch Application	Localhost:3000/predict Disease	Predict Disease page	-

2.	Submit without entering anything	Null	Please fill all the fields	Accuracy (%)
3	Submitting values other than 0 or 1 in the field where 0 and 1 is must	age:65, sex:0, smoke:50, years:10, ldl:69, chp:4, height:168, weight:111, fh:123, active:0, lifestyle:1, ihd:1, hr:98, dm:1, bpsys:120, bpdias:80, htn:1, ivsd:0, ecgpatt:4, qwave:0, target:0	Please either fill 0 or 1 in the required fields	-
4	Submitting the correct values	age:65, sex:0, smoke:0, years:10, ldl:69, chp:4, height:168, weight:111, fh:1, active:0, lifestyle:1, ihd:1, hr:98, dm:1, bpsys:120, bpdias:80, htn:1, ivsd:0, ecgpatt:4, qwave:0, target:0	Patient does not have heart disease	91%
5	Submitting the correct details	age: 65, sex: 1, smoke: 1, years: 0,	Patient has heart disease.	91%

		ldl: 200, chp: 8, height: 160, weight: 90, fh: 1, active: 0, lifestyle: 1, ihd: 1, hr: 120, dm: 1, bpsys: 140, bpdias: 90, htn: 1, ivsd: 1, ecgpatt: 2, qwave: 2, target: 1		

Table 4.5 Doctor Registration

SN	Action	Input	Expected Outcome		Actual Outcome	Test Result
1	Launch Application	http://localhost:3000/apply-doctor	Registration Page		Registration Page	Pass
2	Submit without any details	Null	This field cannot be empty		This field cannot be empty	Pass
3	Enter numeric value	First name: 123 Last name: Mahat Address: Kirtipur Contact:1234567 Specialization: Dentist feesPerConsultation:5000 Email: btycoon77@gmail.com Password: 123456789	Invalid name		Invalid name	Pass
4	Enter email of incorrect format	First name: 123 Last name: Mahat Address: Kirtipur Contact:1234567 Specialization: Dentist feesPerConsultation:5000 Email: b.com Password: 123456789	Please include @in the email		Please include @in the email	Pass
5	Enter correct details	First name: Ram Last name: Mahat Address: Kirtipur Contact:1234567 Specialization: Dentist feesPerConsultation:5000 Email: btycoon77@gmail.com Password: 123456789	Registration Successful		Registration Successful	Pass

Table 4.6 Doctor Login

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Localhost:3000/login	Login Page	Login Page	Pass
2	Submit without any Details	Null	This field cannot be empty	This field cannot be empty	Pass
3	Enter incorrect email	Email: btyn77@gmail.com Password: 123456789	Email not found	Email not found	Pass
4	Enter incorrect password	Email: btycoon77@gmail.com Password: 123456789	Password does not match	Password does not match	Pass
5	Enter correct details	btycoon77@gmail.com Password: 123456789	Redirected to dashboard	Redirected to dashboard	Pass

Table 4.7 Edit Doctor

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	http://localhost:3000/doctor/profile/doctorId	User settings	User Settings	Pass
2	Submit without any Changes	Similar details	Redirected to dashboard	Redirected to dashboard	Pass
3	Change name	Previous Name: Ram New Name: Tycoon	Details updated	Details updated	Pass
4	Change password	Previous password: 123456789 New password: 123456	Redirected to dashboard	Redirected to dashboard	Pass
5	Change address	Previous address: Kirtipur New address: New York	Redirected to dashboard	Redirected to dashboard	Pass

4.2.1 Test Cases for System Testing

System testing is an overall testing of the system after integrating all the functions of the project. When all the functions of the e-healthcare are integrated then system testing is done.

Table 4.8 Admin Interface

S. N	Action	Input	Expected Outcomes	Actual Outcomes	Test Result
1	Launch application	Localhost:3000/login	Login Page	Login Page	Pass
2	Admin Login	Email: admin@gmail.com Password:12345	Login Successful	Login Successful	Pass
3	Approve doctor registration request	Take Action to approve	Redirected to Approve	Redirected to approve	Pass
4	Reject Registration request	Are you sure to reject the Doctor registration? Yes	Registration Revoked	Redirected to dashboard	Pass
5	Approve doctor registration request	Take Action to approve	Redirected to Approve	Redirected to approve	Pass

Table 4.9 User Interface

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Localhost:3000/register	Registration Page	Registration Page	Pass
2	Register new user	First name: Ram Email: btycoon77@gmail.com Password: 1234	Registered successfully	Registered Successfully	Pass
3	Login by same user	Email: btycoon77@gmail.com Password: 1234	Login Successful	Redirected to dashboard	Pass
4	Edit user	Previous name: Ram New name: lasman	Redirected to same page (Details Updated)	Redirect to the same page	Pass
5	Take appointment	Booking date: 14-06-23 Booking time: 6:00 AM	Appointment booked successfully	Appointment booked successfully	Pass
6	Predict Heart Disease	age: 65, sex: 1, smoke: 1, years: 0, ldl: 200, chp: 8, height: 160, weight: 90, fh: 1, active: 0, lifestyle: 1, ihd: 1, hr: 120, dm: 1, bpsys: 140, bpdias: 90, htn: 1, ivsd: 1, ecgpatt: 2, qwave: 2, target: 1	Unknown	Patient has heart disease	Pass

Table 4.10 Doctor Interface

SN	Action	Input	Expected Outcome	Actual Outcome	Test Result
1	Launch Application	Localhost:3000/register	Registration Page	Registration Page	Pass
2	Register new doctor	First name: Ram Last name: Mahat Email: btycoon77@gmail.com Address: kirtipur Specialization: Dentist FeesPerConsultation:10000 Experience: 5 years. Timings: 10:00-5:00	Registered successfully	Registered Successfully	Pass
3	Login by same user	Email: btyn77@gmail.com Password: 1234	Login Successful	Redirected to dashboard	Pass
4	Edit doctor	Previous name: Ram New name: lasman	Redirected to same page (Details Updated)	Redirect to the same page	Pass
5	Approve appointment	Take action to approve User appointment	Appointment was successfully approved	Appointment was successfully approved	Pass

4.3 Result Analysis

The system was tested through unit testing and proved to be effective in executing its intended functions. The results show that the project was able to meet its goals, but there is still room for improvement in terms of expanding the system capabilities and increasing community involvement.

4.3.1 Evaluating Accuracy

In machine learning, accuracy is a common metric used to evaluate the performance of a logistic regression model. Accuracy measures the proportion of correctly classified instances among all the instances among all instances in the dataset. To calculate accuracy, the first step is to divide the dataset into two parts: a training set and a test set. The training set is used to train the model while the testing set is used to evaluate the model's performance. In logistic regression model the most common measure to evaluate accuracy are:

Precision:

Precision is a measure of how many of the predicted positive instances are actually true positives. In the context of heart disease prediction, precision tells you the ratio of correctly predicted positive cases (true positives) to all instances that were predicted as positive (true positives + false positives). A high precision indicates that when the model predicts a positive case, it is likely to be correct.

Precision for class 0: 1.00 (100%)

Precision for class 1: 0.83 (83%)

Recall (Sensitivity):

Recall is a measure of how many of the actual positive instances were correctly predicted by the model. In the context of heart disease prediction, recall tells you the ratio of correctly predicted positive cases (true positives) to all actual positive cases (true positives + false negatives). A high recall indicates that the model is effectively capturing a large portion of positive cases.

Recall for class 0: 0.84 (84%)

Recall for class 1: 1.00 (100%)

F1-score:

The F1-score is a balanced metric that takes into account both precision and recall. It is the harmonic mean of precision and recall. The F1-score provides a single value that balances the trade-off between precision and recall. It is particularly useful when you want to consider both false positives and false negatives in your evaluation.

F1-score for class 0: 0.91 (91%)

F1-score for class 1: 0.91 (91%)

Support:

Support represents the number of instances (samples) in each class. It tells you how many actual instances belong to each class in your dataset.

Support for class 0: 38

Support for class 1: 29

Accuracy:

Accuracy is a measure of overall correctness of the model's predictions. It calculates the ratio of correctly predicted instances to the total number of instances. However, it might not be the best metric when dealing with imbalanced datasets.

Accuracy: 0.91 (91%)

Macro Average:

Macro average calculates the mean of precision, recall, and F1-score for each class without considering class imbalance. It gives equal weight to each class.

Macro average precision: 0.91 (91%)

Macro average recall: 0.92 (92%)

Macro average F1-score: 0.91 (91%)

Weighted Average:

Weighted average is similar to the macro average, but it takes into account the proportion of each class in the dataset. It gives more weight to classes with more instances.

Weighted average precision: 0.93 (93%)

Weighted average recall: 0.91 (91%)

Weighted average F1-score: 0.91 (91%)

4.3.1 .1 Evaluating Accuracy of Logistic Regression Model

Table 4. 11 LR Heart Disease Prediction Classification Report

Logistic Regression Heart Disease Prediction Report				
	Precision	Recall	F1-score	support
0	1.00	0.84	0.91	38
1	0.83	1.00	0.91	29
Accuracy			0.91	67
Macro avg	0.91	0.92	0.91	67
Weighted avg	0.93	0.91	0.91	67

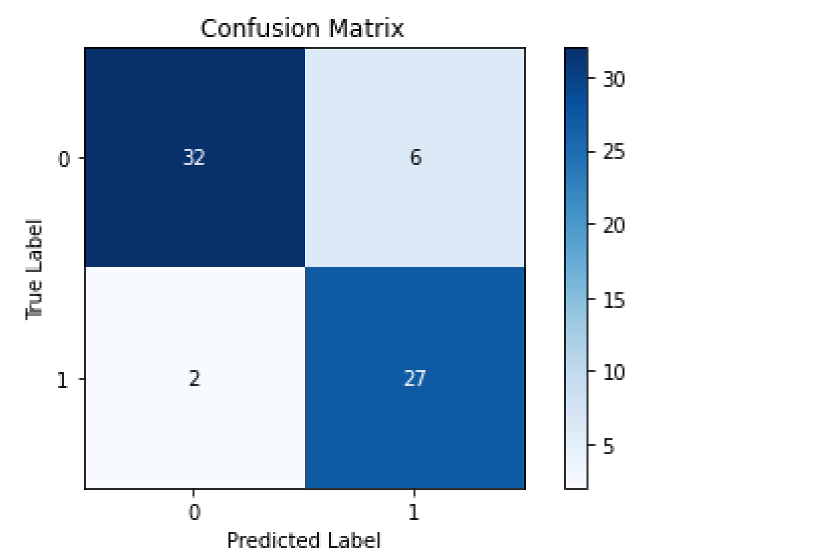


Figure 4. 8 Confusion matrix for Heart Disease Prediction from LR Algorithm

The Figure 4.8 shows the binary class confusion matrix. We can interpret it as follows where Class 0 represents the presence of heart disease and 1 represents the absence of heart disease. 32 patient samples were correctly predicted as presence of heart disease. 6 patient samples were incorrectly predicted as absence of heart disease. 2 patient samples were incorrectly predicted as absence of heart disease while 27 patient samples were correctly predicted as presence of heart disease.

CHAPTER 5

CONCLUSION AND FUTURE RECOMMENDATION

5.1 Conclusion

This project "Heart Disease Prediction System" a web-based mobile application provides a convenient platform for users to predict heart disease based on user input and schedule appointments with cardiologist if heart disease is found in the patient. However, the system current accuracy levels have been found to be average, indicating the need for further improvement. To enhance the accuracy and reliability of heart disease prediction, additional algorithms should be incorporated into the system. By incorporating algorithms, the system can strive towards providing more reliable and accurate predictions, thereby enhancing its overall utility and effectiveness in the field of healthcare.

5.2 Lesson Learnt/ Outcome

During the application development process, there were many circumstances where the author felt like something more could be added but due to lack of enhancement and excellence in the programming language there might be some loopholes which could be fixed once the author gets to learn and explore this technology. The constraints that author have faced during the project development are time management and learning to build application using Mongo db, Node.js, experess.js, React.js and Python. Due to improper time management, the developed application is not of top-notch quality. Besides that, learning to build the application is a difficult task for a first timer with lack of knowledge as a lot needs to be learned. Although the project has not turned out to be exactly what the author has imagined it to be, there are some functions that the author would like to add in the forthcoming days and make it more user friendly and competitive

5.3 Future Recommendation

This project “Heart Disease Prediction System” has the potential to be improved in the future. The first and foremost thing it is recommended is to incorporate the Random Forest algorithm and Deep Learning models to improve its accuracy in predicting heart disease. Random Forest, with its ability to handle complex relationships within the data, can enhance the system's predictive capabilities by capturing a wider range of features and leveraging ensemble learning techniques. Secondly, Deep Learning models, such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), have demonstrated remarkable success in various medical applications. By integrating these models into the system, it can benefit from their ability to uncover intricate patterns in the data, leading to more accurate heart disease predictions.

REFERENCES

[1] Asha Rajkumar, and Mrs. G. SophiaReena, “Diagnosis of Heart Disease Using Data Mining Algorithm”, Global Journal of Computer Science and Technology, Vol. 10, pp. 38- 43, Sept. 2010.

[2] G.Subbalakshmi, K. Ramesh, and M.C. Rao, “Decision Support in Heart Disease Prediction System using Naive Bayes”, Indian Journal of Computer Science and Engineering (IJCSE), Vol. 2, No. 2, Apr-May 2011.

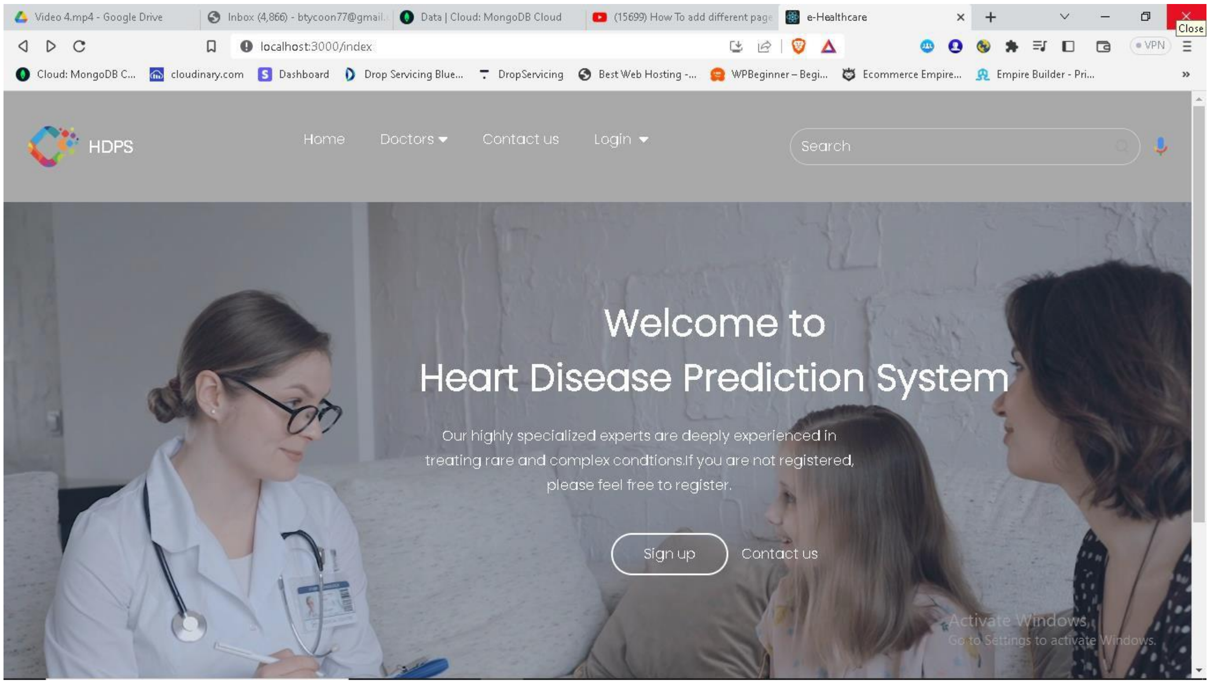
[3] Abhishek Taneja, “ Heart Disease Prediction System Using Data Mining Techniques”, Oriental Journal Of Computer Science and Technology, Vol. 6, pp. 457-466, Dec.2013.

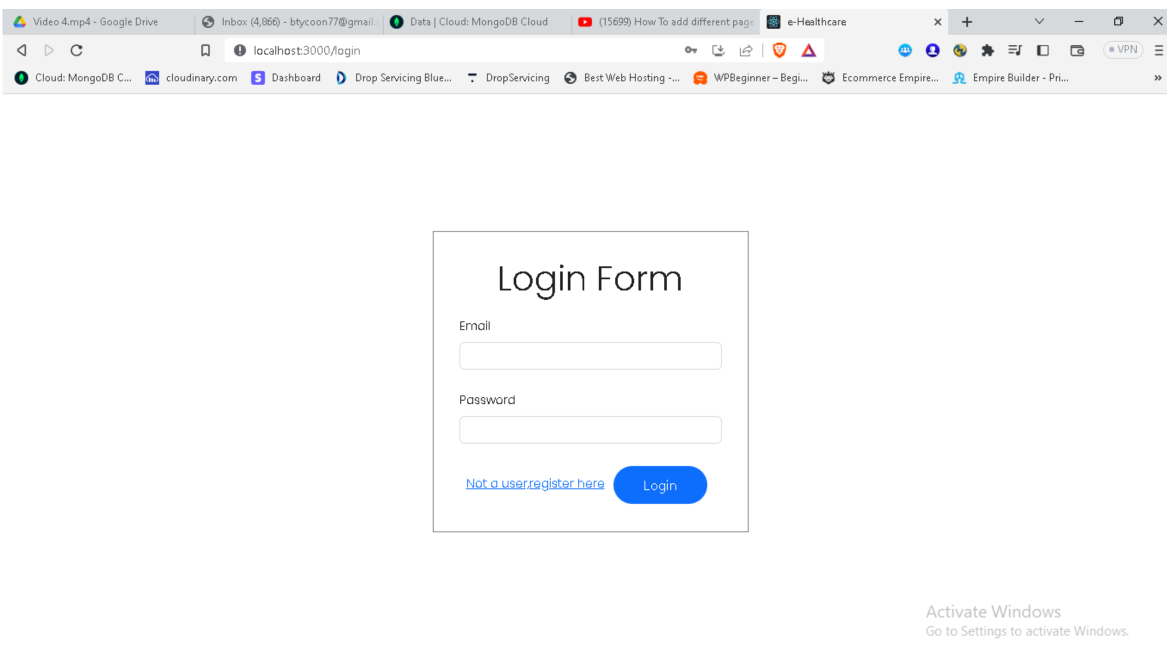
[4] Hlaudi Daniel Masethe, and Mosima Anna Masethe, “Prediction of Heart Disease Using Classification Algorithms”, in Proceedings of the World Congress on Engineering and Computer Science 2014 Vol. II WCECS 2014, 22-24 Oct. 2014, San Francisco, USA.

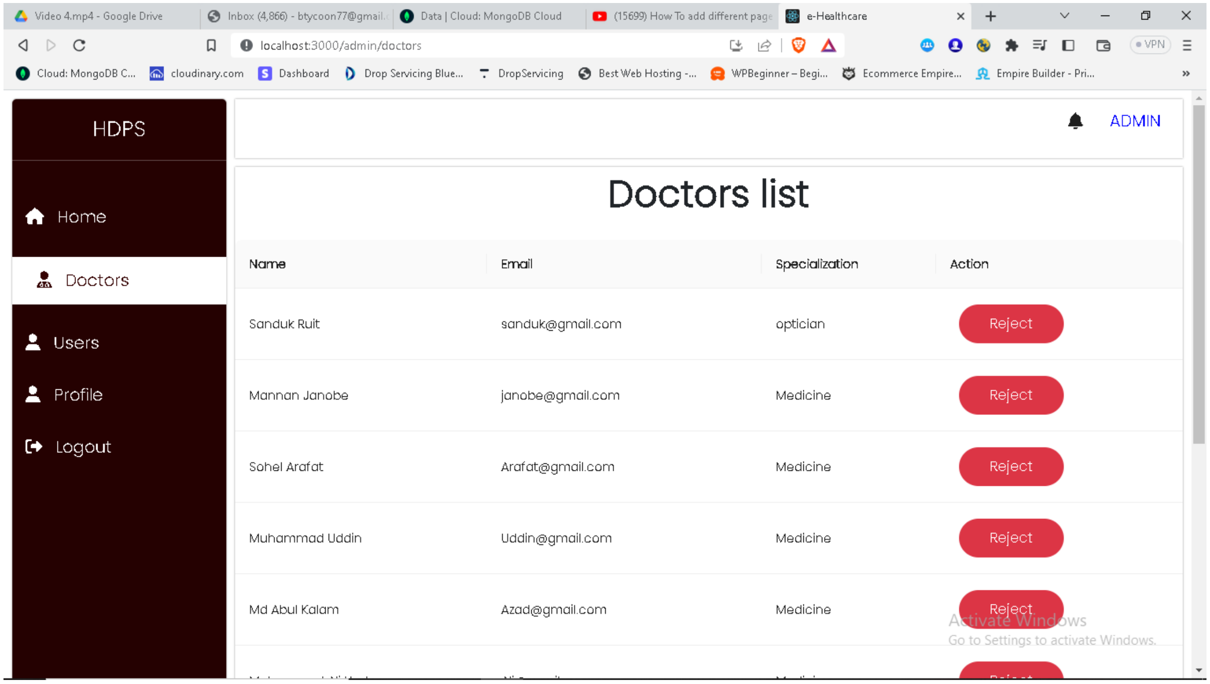
[5] K.Gomath, Dr. Shanmugapriyaa, “Heart Disease Prediction Using Data Mining Classification”, International Journal for Research in Applied Science & Engineering Technology (IJRASET), Vol.4, Issue 2, February-2016.

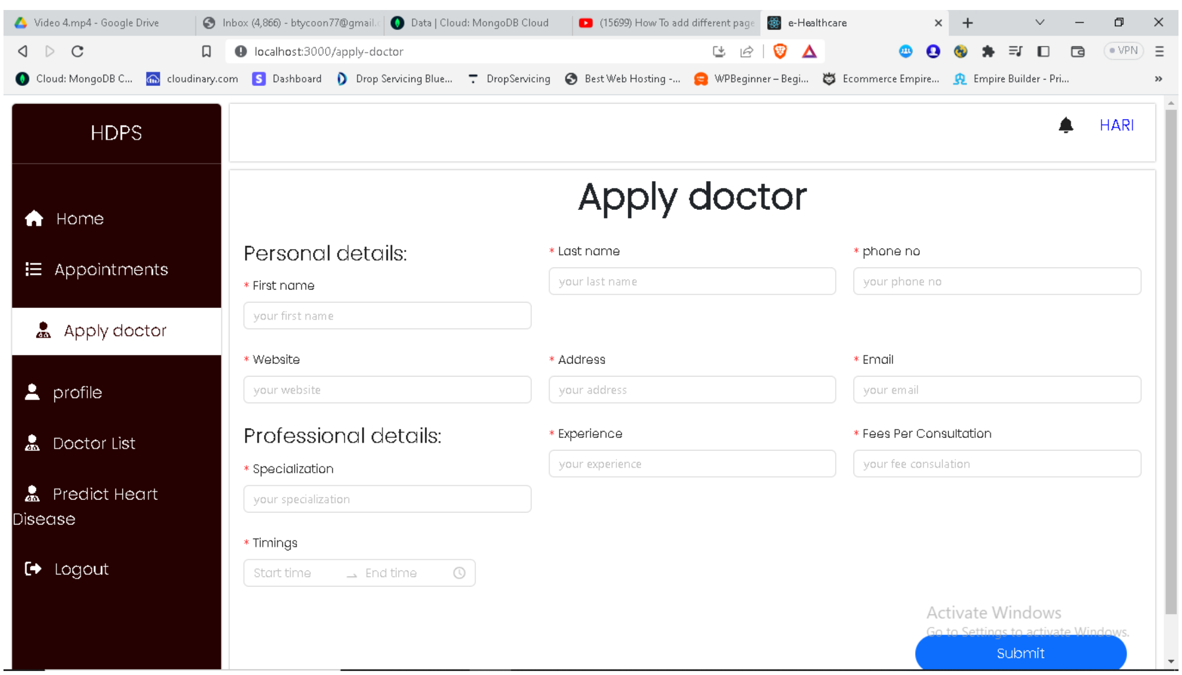
[6] “Erbil Heart Disease Dataset”, www.kaggle.com.
https://www.kaggle.com/datasets/hangawqadir/erbil-heart-disease-dataset
(accessed Jun 09,2023)

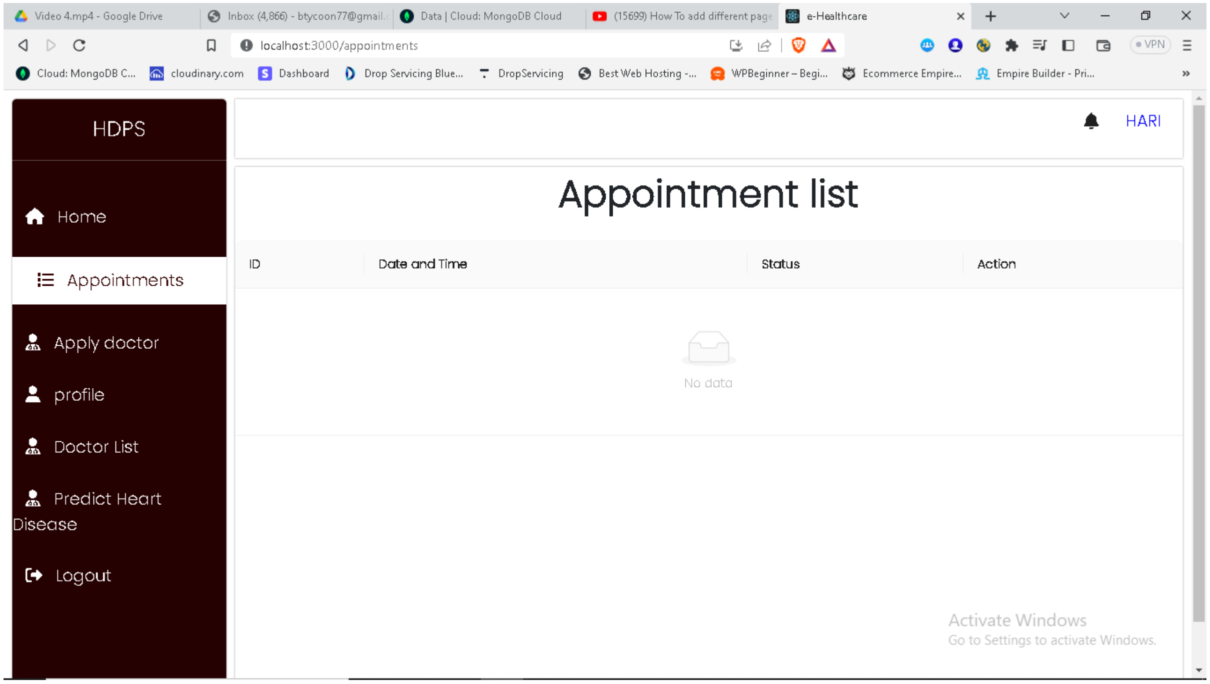
APPENDICES

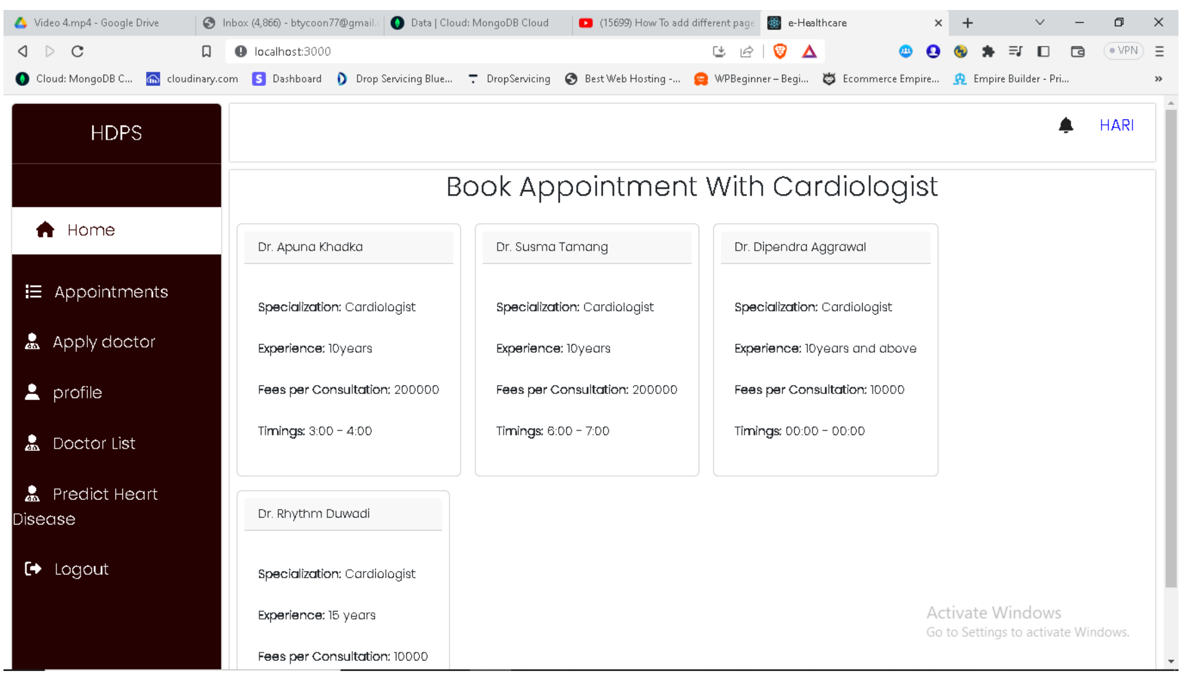

localhost:3000/index
HDPS
Home
Doctors
Contact us
Login
Search
Welcome to
Heart Disease Prediction System
Our highly specialized experts are deeply experienced in treating rare and complex condtions.If you are not registered, please feel free to register.
Sign up
Contact us

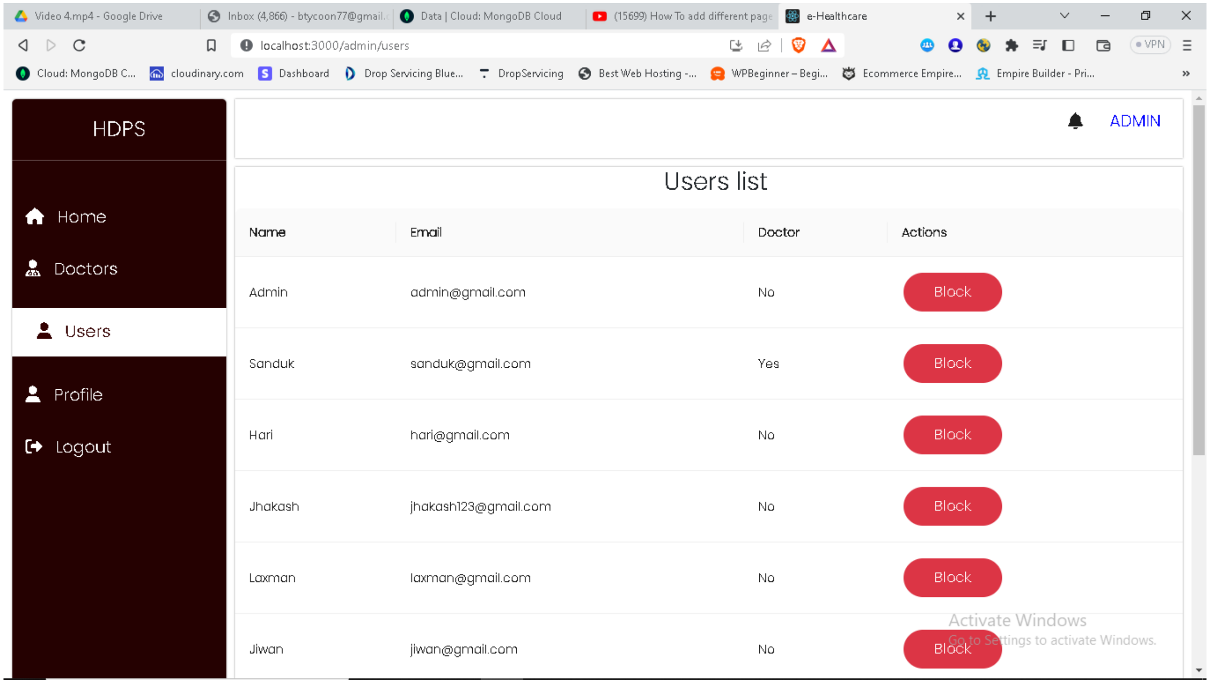

localhost:3000/login
Login Form
Email
Password
Not a user,register here
Login

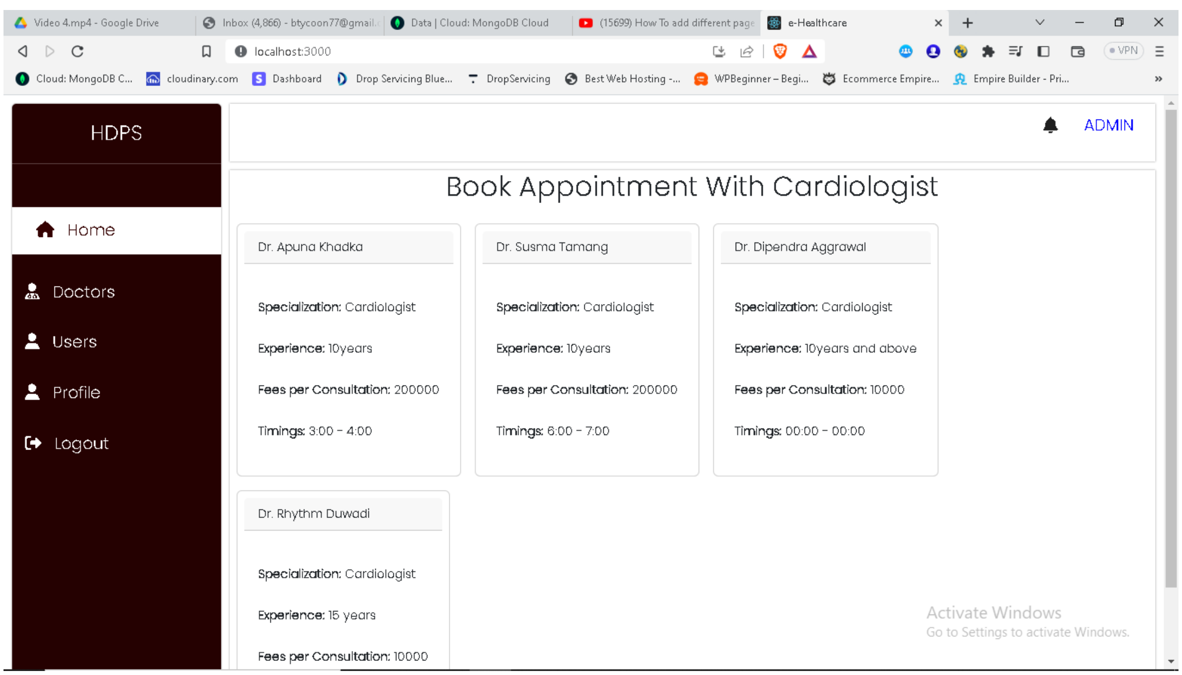

localhost:3000/admin/doctors
HDPS
Home
Doctors
Users
Profile
Logout
ADMIN
Doctors list
Name
Email
Specialization
Action
Sanduk Ruit
sanduk@gmail.com
optician
Reject
Mannan Janobe
janobe@gmail.com
Medicine
Reject
Sohel Arafat
Arafat@gmail.com
Medicine
Reject
Muhammad Uddin
Uddin@gmail.com
Medicine
Reject
Md Abul Kalam
Azad@gmail.com
Medicine
Reject
Activate Windows
Go to Settings to activate Windows.

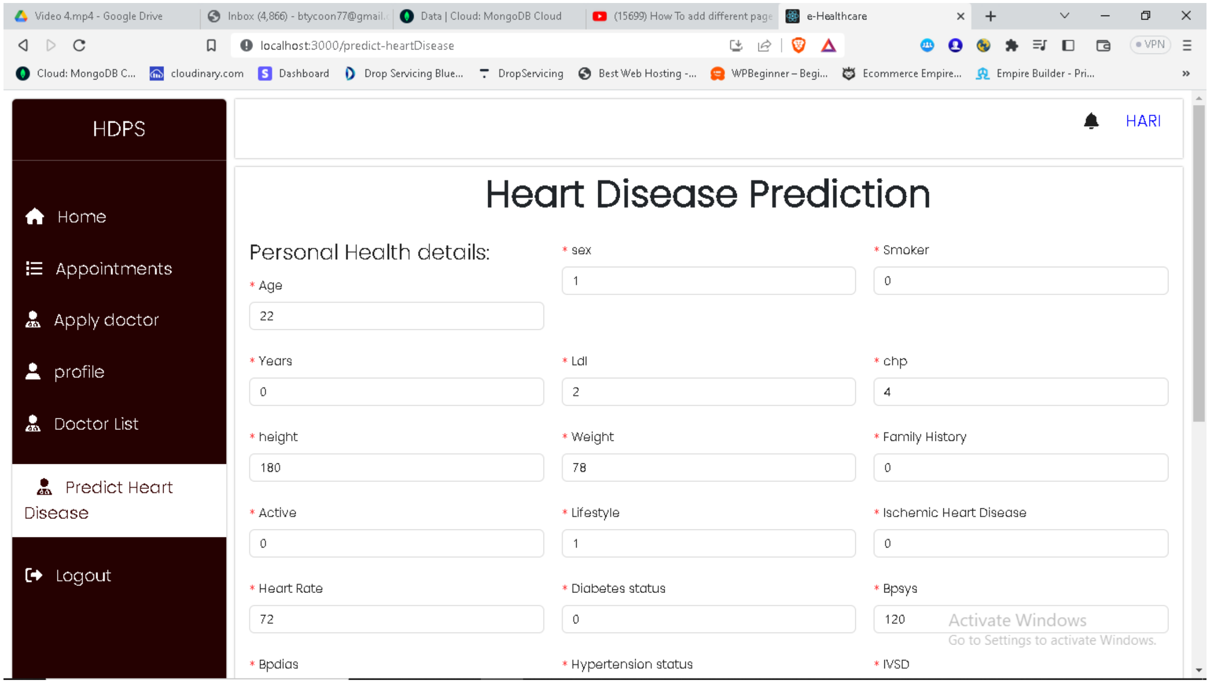

localhost:3000/apply-doctor
HDPS
Home
Appointments
Apply doctor
profile
Doctor List
Predict Heart Disease
Logout
HARI
Apply doctor
Personal details:
* First name
your first name
* Last name
your last name
* phone no
your phone no
* Website
your website
* Address
your address
* Email
your email
Professional details:
* Specialization
your specialization
* Experience
your experience
* Fees Per Consultation
your fee consulation
* Timings
Start time
End time
Activate Windows
Go to Settings to activate Windows.
Submit

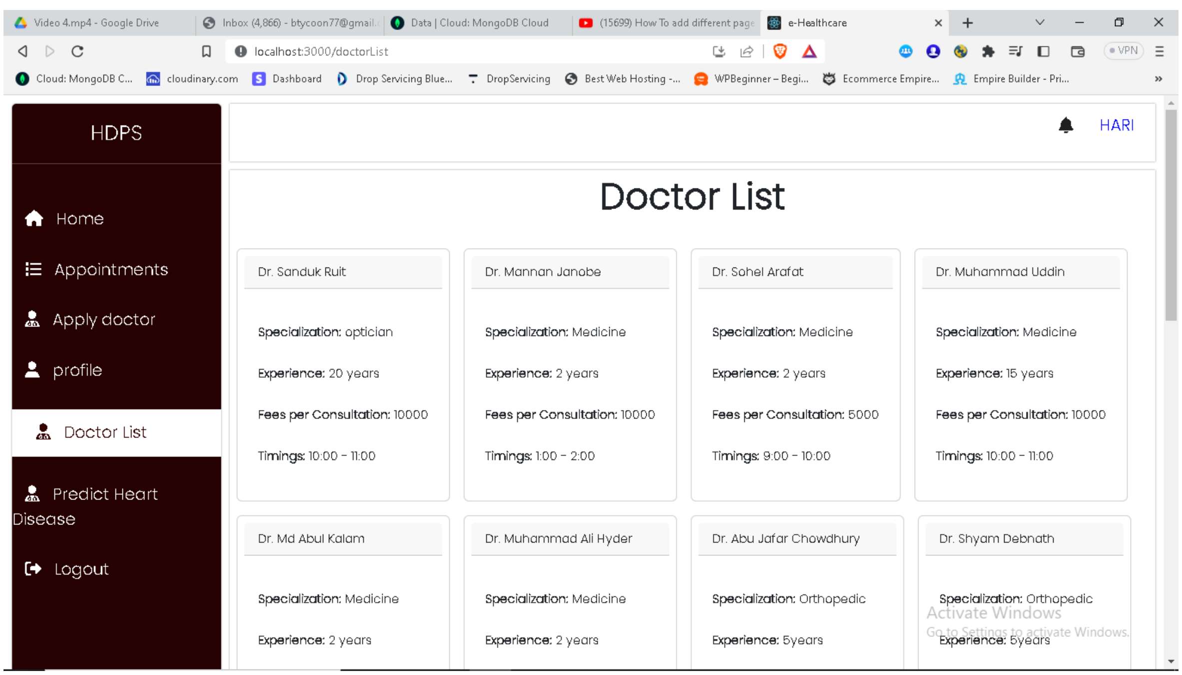

localhost:3000/appointments
HDPS
Home
Appointments
Apply doctor
profile
Doctor List
Predict Heart Disease
Logout
HARI
Appointment list
ID
Date and Time
Status
Action
No data
Activate Windows
Go to Settings to activate Windows.

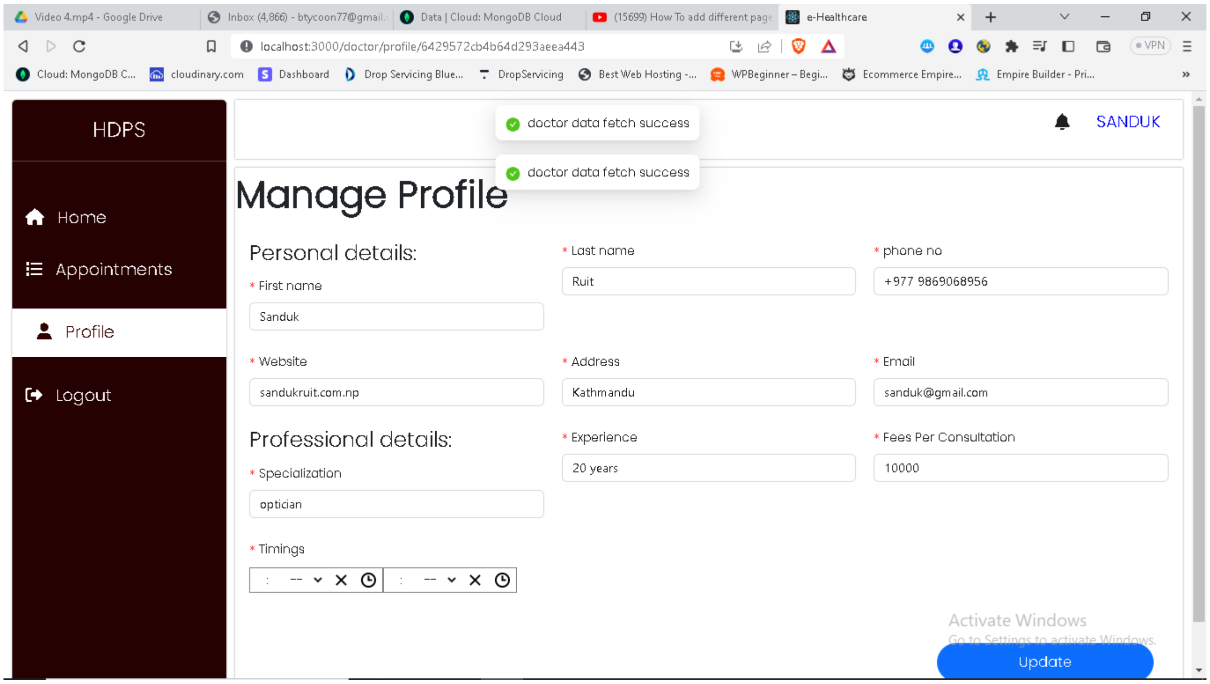

localhost:3000
HDPS
Home
Appointments
Apply doctor
profile
Doctor List
Predict Heart Disease
Logout
HARI
Book Appointment With Cardiologist
Dr. Apuna Khadka
Specialization: Cardiologist
Experience: 10years
Fees per Consultation: 200000
Timings: 3:00 - 4:00
Dr. Susma Tamang
Specialization: Cardiologist
Experience: 10years
Fees per Consultation: 200000
Timings: 6:00 - 7:00
Dr. Dipendra Aggrawal
Specialization: Cardiologist
Experience: 10years and above
Fees per Consultation: 10000
Timings: 00:00 - 00:00
Dr. Rhythm Duwadi
Specialization: Cardiologist
Experience: 15 years
Fees per Consultation: 10000
Activate Windows
Go to Settings to activate Windows.

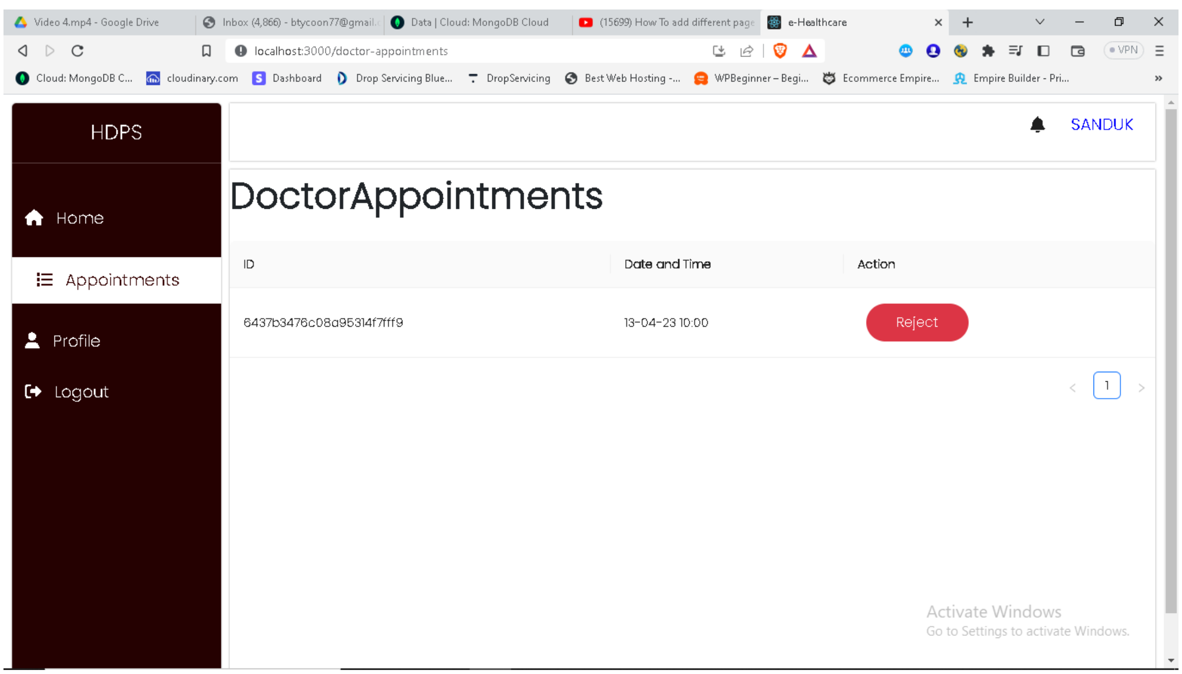

HDPS
Home
Doctors
Users
Profile
Logout
ADMIN
Users list
Name
Email
Doctor
Actions
Admin
admin@gmail.com
No
Block
Sanduk
sanduk@gmail.com
Yes
Block
Hari
hari@gmail.com
No
Block
Jhakash
jhakash123@gmail.com
No
Block
Laxman
laxman@gmail.com
No
Block
Jiwan
jiwan@gmail.com
No
Block

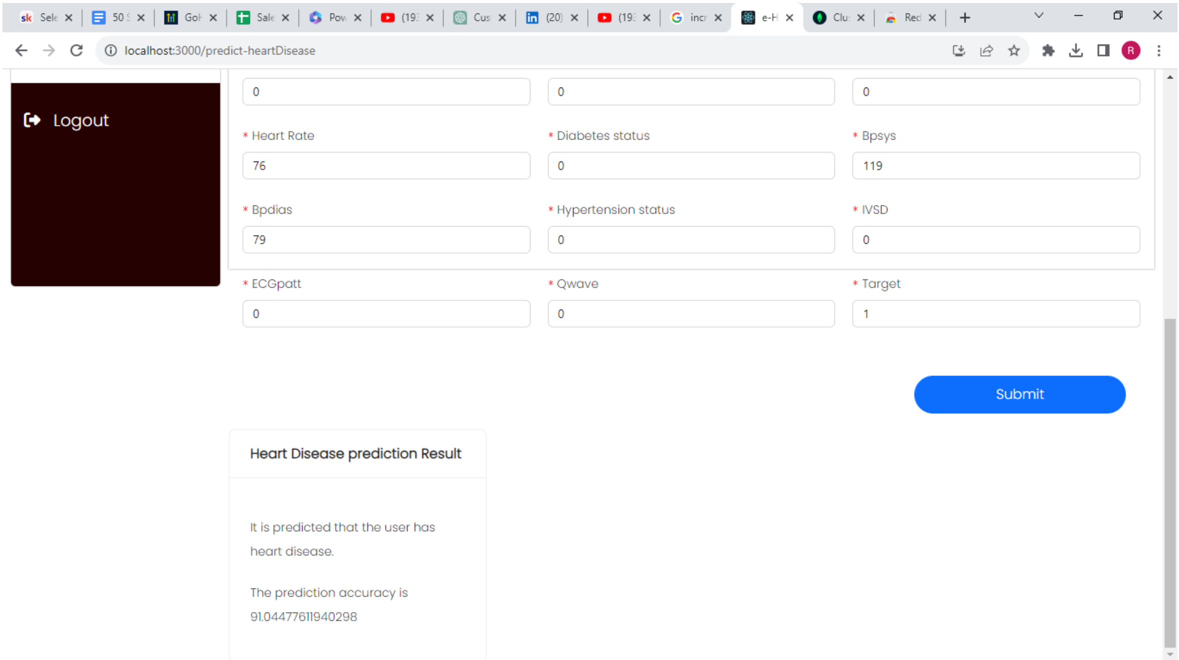

HDPS
Home
Doctors
Users
Profile
Logout
ADMIN
Book Appointment With Cardiologist
Dr. Apuna Khadka
Specialization: Cardiologist
Experience: 10years
Fees per Consultation: 200000
Timings: 3:00 - 4:00
Dr. Susma Tamang
Specialization: Cardiologist
Experience: 10years
Fees per Consultation: 200000
Timings: 6:00 - 7:00
Dr. Dipendra Aggrawal
Specialization: Cardiologist
Experience: 10years and above
Fees per Consultation: 10000
Timings: 00:00 - 00:00
Dr. Rhythm Duwadi
Specialization: Cardiologist
Experience: 15 years
Fees per Consultation: 10000


localhost:3000/predict-heartDisease
HDPS
HARI
Heart Disease Prediction
Home
Appointments
Apply doctor
profile
Doctor List
Predict Heart Disease
Logout
Personal Health details:
* sex
1
* Smoker
0
* Age
22
* Years
0
* Ldl
2
* chp
4
* height
180
* Weight
78
* Family History
0
* Active
0
* Lifestyle
1
* Ischemic Heart Disease
0
* Heart Rate
72
* Diabetes status
0
* Bpsys
120
* Bpdias
* Hypertension status
* IVSD


localhost:3000/doctorList
HDPS
HARI
Doctor List
Home
Appointments
Apply doctor
profile
Doctor List
Predict Heart Disease
Logout
Dr. Sanduk Ruit
Specialization: optician
Experience: 20 years
Fees per Consultation: 10000
Timings: 10:00 - 11:00
Dr. Mannan Janobe
Specialization: Medicine
Experience: 2 years
Fees per Consultation: 10000
Timings: 1:00 - 2:00
Dr. Sohel Arafat
Specialization: Medicine
Experience: 2 years
Fees per Consultation: 5000
Timings: 9:00 - 10:00
Dr. Muhammad Uddin
Specialization: Medicine
Experience: 15 years
Fees per Consultation: 10000
Timings: 10:00 - 11:00
Dr. Md Abul Kalam
Specialization: Medicine
Experience: 2 years
Dr. Muhammad Ali Hyder
Specialization: Medicine
Experience: 2 years
Dr. Abu Jafar Chowdhury
Specialization: Orthopedic
Experience: 5years
Dr. Shyam Debnath
Specialization: Orthopedic
Experience: 5years


HDPS
Home
Appointments
Profile
Logout
doctor data fetch success
doctor data fetch success
SANDUK
Manage Profile
Personal details:
* First name
Sanduk
* Last name
Ruit
* phone no
+977 9869068956
* Website
sandukruit.com.np
* Address
Kathmandu
* Email
sanduk@gmail.com
Professional details:
* Specialization
optician
* Experience
20 years
* Fees Per Consultation
10000
* Timings
Update


HDPS
Home
Appointments
Profile
Logout
SANDUK
DoctorAppointments
ID
Date and Time
Action
6437b3476c08a95314f7fff9
13-04-23 10:00
Reject


localhost:3000/predict-heartDisease
Logout
* Heart Rate
76
* Diabetes status
0
* Bpsys
119
* Bpdias
79
* Hypertension status
0
* IVSD
0
* ECGpatt
0
* Qwave
0
* Target
1
Submit
Heart Disease prediction Result
It is predicted that the user has heart disease.
The prediction accuracy is 91.04477611940298