

# **MEDICAL ASSISTANCE APPLICATION**

Major Project Report submitted in partial fulfillment of the requirement for the degree of Bachelor  
of Technology in

## **Computer Science and Engineering**

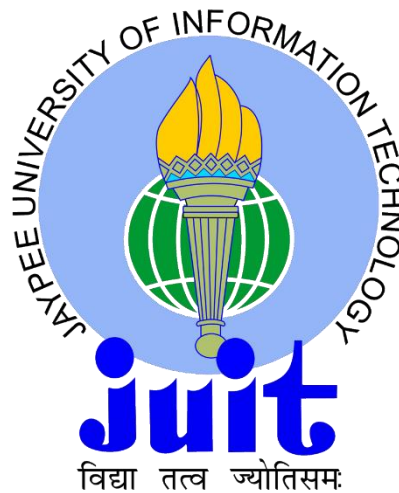
By

Aryan Malhotra 201153

Khushboo Gupta 201193

**UNDER THE SUPERVISION OF**

Dr. Amit Kumar



**Department of Computer Science & Engineering and Information Technology**

**Jaypee University of Information Technology,  
Waknaghat, 173234, Himachal Pradesh, INDIA**

# CERTIFICATE

I hereby declare that the work presented in this report entitled “Medical Assistance Application” in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering and Information Technology submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2020 to May 2024 under the supervision of (Dr. Amit Kumar) (Assistant Professor S.G of Department of Computer Science & Engineering/Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Aryan Malhotra

201153

Khushboo Gupta

201193

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Amit Kumar

Assistant Professor S.G.

Department of Computer Science & Engineering and Information Technology

Dated: 15<sup>th</sup> May 2024

# CANDIDATE'S DECLARATION

We hereby declare that the work presented in this report entitled '**MEDICAL ASSISTANCE APPLICATION**' in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering / Information Technology** submitted in the Department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to May 2024 under the supervision of **Dr. Amit Kumar** (Assistant Professor (SG), Department of Computer Science & Engineering and Information Technology).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature with Date)

Student Name: Aryan Malhotra

Roll No.: 201153

(Student Signature with Date)

Student Name: Kushboo Gupta

Roll No.: 201193

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature with Date)

Supervisor Name: Dr. Amit Kumar

Designation: Assistant Professor (SG)

Department: Computer Science & Engineering and Information Technology

Dated: 15<sup>th</sup> May 2024

# ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for his divine blessing that makes it possible to complete the project work successfully. I am grateful and wish my profound indebtedness to supervisor Dr. Amit Kumar, Assistant Professor S.G., Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of “Visualization, Machine Learning, and Neural Networking” helped me to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project. I would like to express my heartiest gratitude to Dr. Amit Kumar, Department of CSE, for his kind help to finish my project. I would also generously welcome each of those individuals who have helped me straightforwardly or in a roundabout way to make this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Aryan Malhotra

201153

Khushboo Gupta

201193

# TABLE OF CONTENTS

S. No.	Chapter	Sections	Page No.
1.	Chapter 1: Introduction	1.1 Introduction	1
		1.2 Problem Statement	2
		1.3 Objectives	2
		1.4 Significance and Motivation of the Project Work	3
		1.5 Organization of Project Report	4
2.	Chapter 2: Literature Survey	2.1 Overview of Relevant Literature	5
		2.2 Key Gaps in the Literature	7

3.	Chapter 3: System Development	3.1 Requirements and Analysis  3.2 Project Design and Architecture  3.3 Data Prepration  3.4 Implementation  3.5 Key Challenges	11  12  16  17  38
4.	Chapter 4: Testing	4.1 Testing Strategy  4.2 Test Cases and Outcomes	41  43
5	Chapter 5: Results and Evaluation	5.1 Results  5.2 Comparison with Existing Solutions	53  58
6.	Chapter 6: Conclusion and Future Scope	6.1 Conclusion  6.2 Future Scope	61  61

# LIST OF FIGURES

Fig No.	Figure Name	Page No.
1	Website Design	12
2	Diabetes Model flowchart	13
3	Heart disease Prediction flowchart	14
4	Medical insurance cost estimator flowchart	15
5	Appointment Booking Confirmed	18
6	Appointment Confirmed by doctor	19
7	Confirmation Email Preview	20
8	Logistic Regression Curve	33
9	SVM working	34
10	Linear Regression curve	36
11	Patient's Dashboard Home Page	53
12	Patient's Dashboard Diabetes Detection Page	54
13	Patient's Dashboard Heart Disease Prediction Page	54
14	Patient's Dashboard Medical Insurance Cost Estimator Page	55
15	Patient's Dashboard Appointment Booking Page	55
16	Patient's Dashboard Blog Page	56
17	Doctor's Dashboard Appointment List Page	57

# ABSTRACT

AI & ML Based Ailment Divination is a system that anticipates disease-related facts or symptoms that are fed into the system and provides accurate findings based on that information. It works well when the patient isn't seriously unwell, and the user merely has to know the type of ailment. It's an app that gives users guidance and techniques for staying on track with their health plans, as well as a way to diagnose disease via divination. Nowadays, the healthcare enterprise performs a critical function in treating patients' illnesses, so that is additionally a shape of help withinside the healthcare enterprise to inform the consumer and additionally beneficial to the consumer in case he does not wish to visit the medical institution or different clinics, so via way of means of including signs and all different beneficial information, they may be already in control, and the healthcare enterprise can benefit via way of means of virtually including this application into the traditional approach.



# **CHAPTER 1: INTRODUCTION**

## **1.1 Introduction**

Disease prediction based on artificial intelligence and machine learning is a system that uses information from users when registering to predict diseases. It also predicts the disease of the patient or user based on the information or symptoms entered into the system and returns appropriate results. If the patient is not seriously ill, the user only needs to know the type and outcome of the disease. It is a tool that provides users with advice and methods to manage their health care plans.

This is an important tool in the healthcare industry to provide information to users and also to help users if they do not want to go to the hospital or other hospitals because the medical industry plays an important role in the treatment of patients. responsibility. illness. This method is very useful for users. Using machine learning algorithms, Python programming language and existing hospital data, we completed the project and achieved the desired results.

Today, doctors use many techniques and research to diagnose and treat infectious and fatal diseases. Doctors sometimes make mistakes when diagnosing patients, so a disease prophecy system using machine learning can help achieve good results in these cases.

According to the research, an infection control system based on artificial intelligence and machine learning has been developed to combat infection at an early stage. As we all know, people focus on the competition for economic development and forget about our health. The main reason for not understanding personal health is laziness about going to the doctor and anxiety about not being able to make an appointment, which ultimately leads to a fatal disease.

## **1.2 Problem Statement**

Now a day's there are a variety of issues in the health industry that are connected to the equipment or gadgets that will supply. If we do not avoid those wrong or unaccepted results, we are going to build a program/project that will provide us with the exact and desired Divinations depending on the details offered by the user and also taking into consideration the existing data in that machine. There are lots of details to cover in that very large industry known as health sector, which is one of the poorest when it comes to data and intellectual input. With the aid of all those algorithms, techniques, and methodologies, this project has been achieved for whom it is meant. However, most people visit either hospitals or clinics to ascertain their health position concerning the coming day. Consequently, they must travel a long distance just to obtain feedback regarding their response, and sometime patients do not even receive their answers for different reasons like their doctors can be on leave, weather problems

## **1.3 Objectives**

The aim of this project, called "DocHub", is to accurately predict the patient's illness using the general information and symptoms of each patient. Using this information, we will compare it with the patient's previous information and predict the condition he or she may have. If the divination is done in the early stages of the disease with the help of this plan and other necessary precautions, the disease can be cured and most of the time this divination work is also useful in the healthcare sector. If doctors accept these plans, doctors' work will be reduced and they will be able to predict patients' conditions more easily. The main purpose of this disease classification is to identify many recurrent diseases that can be fatal if left untreated or ignored, as they pose a great danger to patients and their families. The system will predict most diseases based on symptoms. There is a lack of knowledge and experience about the healthcare industry, it is a huge job and a lot of work. Thus, we

have completed this project that will help those who need help with the help of all algorithms, technologies and methods.

#### **1.4 Significance and Motivation of the Project Work**

- This work will use datasets to predict the patients' diseases depending on their symptoms and other general data.
- This is majorly done using the previous datasets of the hospitals hence after testing, it will achieve up to 80% of correct results and the mission is still moving forward to obtain a hundred percent correct responses.
- It would anticipate ailment of the patient with support of DocHub, , solve problems and prevent other aspects.
- It offers protection to the system where nobody is allowed inside or to change anything within the system.
- For more precision, the user is supposed to put all the described symptoms to make it possible for the app to select the relevant one.
- This is where we can easily setup the data so that it can be transformed into an algorithm thereby improving our project's undertaking.
- It offers a variety of options that enable one to select from among the different properties and characteristics. In this case, the user must register first before being able to employ the Divination and then logging into the system by means of credential like username or password.
- First, the user opens the program to login and then registers by clicking on the Register or sign-up button.

Next, a basic registration data is entered which gets logged into the system.

## **1.5 Organization of Project Report**

- Chapter 1: It represents a system or project's introduction, problem statement, scope as well as objectives.
- Chapter 2: The literature review provides a brief study on the published articles and includes any criticisms, where applicable.
- Chapter 3: The detailed requirement of the problem identified for the major project, system architecture, and implementation details.
- Chapter 4: Performance analysis of the model. (TESTING)
- Chapter 5: Concludes the Report.
- Chapter 6: Future scope and conclusions.

## **CHAPTER 2: LITERATURE SURVEY**

### **2.1 Survey**

This is what a machine learning is as defined by Torres Mitchell (it is an “artificial intelligence or a computer program that is said to be able to learn from experiences and other actions and to do anything that creates a knowledge, or something as measured”). Machine Learning entails intermixing that is often related to relationships and links across different databases, many of which already exist. Thus, when a specific pattern is identified by Machine Learning Algorithms the model may take advantage of this relation to predict future observations and performance and some interesting interrelations are produced. Unstructured Reading in machine learning include Naïve Bayes Classifier, Bayes theorem, KNN (k-Nearest Neighbour Classifier) decisions trees, entropy, ID3, SVM i.e. support vector machines and random forest among others with data mining, One application area within Bulela, particularly data mathematics, is predictive modeling through machine learning in a commercial setting. Researchers, data scientists, engineers and analysts can derive ‘reliable results’ by learning from previous correlates and data styles to derive ‘unexpected insights. Algorithms used in Machine Learning. Arthur Samuel coined the term “learning machine” in 1959. Machine-learning is made up of algorithms which can comprehend and extract information. The math genius is closely linked to this subject and provides it with methods, teaching, and applications as background. Machine learning is frequently confused

In this case, there are too many chance variables associated with a variety of algorithms such as databases, programs, etc. The high chance and low chance affected person type is achieved based on the observations. achieved in a group. However, there is value for those fashions in healthcare settings and not the corporate world. Therefore, the Divinations system is built on Device learning and the supervised learning methods that facilitate covering the Ailment Divinations for various healthcare industries.

From an extensive review and comparison of all algorithms and machine learning theorems, we have discovered that logistic regression algorithm, K-Nearest Neighbors algorithm, Naive Bayes, regression and random forest algorithm are essential for establishing a system for divination concerning patients illness. Following research and confirming our initial findings, we established that one can achieve a maximum of 90 per cent accuracy by utilizing several approaches, including neural networks to detect disease. The patient data is statistical, diseases history as well as outcomes stored hence making it possible for us to identify a data solution at a low medical case studies cost. This approach allows one to predetermine the disease, but does not permit prediction of the disease's pathological modification or the human condition. Diagnosis of this disease is ill-defined, unknown, an enigma.

## 2.2 Key Gaps in the Literature

There would be several techniques, approaches, processes, software, and methods used to create a new symptom-based diagnostic system from patients and compare their symptoms with stored system's data. By using these data sets, we are able to determine the accurate proportion of sickness in the patient. A database with symbols contains the divination model of a system in which the data is processed, stored for future reference, and the user selects the feature where he will put the different symbols. Afterwards, various algorithms and techniques like logistic regression, KNN, naïve bayes, random forest and so on were used to separate the data. The system's risk analysis is then displayed through the reconsideration model showing systemic estimates, which are presented in different scenarios, including how the system behaves when the large number of petitions for the divorce is filed followed by patient reconsiderations basing on the outcome. We have built this structure and collected this informal data into the framework of Disease prediction for the overall risk analysis required for it. Systematic analysis can help us locate some chronic illnesses in a particular place or community. We use algorithms and techniques to select attributes spontaneously as informal analysis. The system notes the user's symptom and predicts disease based upon the previous database and present symptoms assisting the current diagnosis for viral infections, heart rate, blood pressure, diabetes, and many other internal and external signs and predicts diseases accordingly.

The following page shows the literature survey of various research papers related to my project field.

S. No.	Paper Title	Journal/Conference (Year)	Tools/Techniques/ Dataset	Results	Limitations
1.	HEALTH ASSISTANT	IJARIE-ISSN(O)-2395 -4396 , 2022	Voice recognition, speech recognition, Python programming, Tkinter GUI library, WebBrowser module, OS	successfully automates patient registration and doctor-patient interactions, enhancing efficiency and	Limited accuracy in voice recognition, security dependency on internet connectivity for some features.

S. No.	Paper Title	Journal/Conference (Year)	Tools/Techniques/ Dataset	Results	Limitations
			module, Requests module.	accessibility in healthcare.	
2.	Virtual Assistant For Health Care Services	Turkish Journal of Computer and Mathematics Education, 2021.	web development technologies, chat applications, and Attribute Based Encryption Algorithm	demonstrated the effectiveness of a virtual healthcare system, enhancing accessibility, reducing costs, and ensuring data security,	potential lack of internet access, hindering the system's usage in regions with limited connectivity
3.	Smart healthcare Ai assistant	ICSMDI 2021	Machine Learning, Artificial Intelligence, TensorFlow Lite, Chatterton Platform, Google Calendar API	Improved healthcare management through a chatbot incorporating disease prediction, medicine scanning, enhancing user self-care and medication adherence.	Limited accuracy in disease prediction, dependency on user- provided symptoms
4.	Virtual medical assistant using machine learning	IJRESM 2021	machine learning algorithms and cloud computing infrastructure for advanced medical data analysis and prediction	project achieved significant accuracy in predicting diseases, improving patient outcomes, and reducing healthcare costs.	Limited availability of comprehensive healthcare data
5.	Healthcare virtual assistant	IRJMETS 2021	the project utilized Python for programming, incorporating machine learning with decision trees, speech	successfully diagnosed diseases based on user symptoms, facilitated appointments with doctors,	accuracy heavily relies on the quality and quantity of input data



S. No.	Paper Title	Journal/Conference (Year)	Tools/Techniques/ Dataset	Results	Limitations
			recognition libraries, AMPP and phpMyAdmin	and sent email notifications, ensuring efficient healthcare assistance.	
6.	Medical Instructed Real-Time Assistant for Patient with Glaucoma and Diabetic Conditions	Applied sciences 2020	utilised deep learning models, natural language processing algorithms, and mobile application development frameworks for its implementation	MIRA demonstrated an 89.8% accuracy in disease prediction, using the PARADISE framework, and received positive user experience feedback	accuracy heavily relies on the quality and quantity of input data
7.	Virtual Assistant and Patient Monitoring System by using AI & Data Science	IRJET 2019	Artificial Intelligence, Data Science, Machine Learning, Naive Bayes algorithm, Hidden Markov Model, Data Mining techniques.	traditional methods, offering enhanced accuracy and speed, accessible anytime and anywhere for various patient issues.	The paper does not address potential ethical concerns, privacy issues, or challenges related to the implementation of the proposed virtual assistant and patient monitoring system.
8.	Medical Insurance Cost Prediction using Machine Learning	IJRASET 2022	Multiple Linear Regression, Decision Trees, and Gradient Boosting Regression for machine learning	Gradient Boosting Regression outperforms other models with an accuracy of 86.86% in predicting healthcare insurance costs	study does not address the potential biases in the dataset and does not explore the impact of socio-economic factors

S. No.	Paper Title	Journal/Conference (Year)	Tools/Techniques/ Dataset	Results	Limitations
				based on specific attributes.	

# **CHAPTER 03:- SYSTEM DEVELOPMENT**

## **3.1 Requirements and Analysis**

- Pentium 4. Intel Core i3, i5, i7. and 2 GHz. processor RAM must be atleast 6 512MB.
- Hard disc with a capacity of at least 10 GB
- Input Keyboard and Mouse.
- Monitor for an output device
- Visual Studio Code for implementing Stream lit code.
- Google Collab for making machine learning models.
- Python as the programming language.
- MySQL for database.
- SMTPLIB and Gmail SMTP server and credentials for sending emails

## 3.2 Project Design and Architecture

### 3.2.1 Website Design:

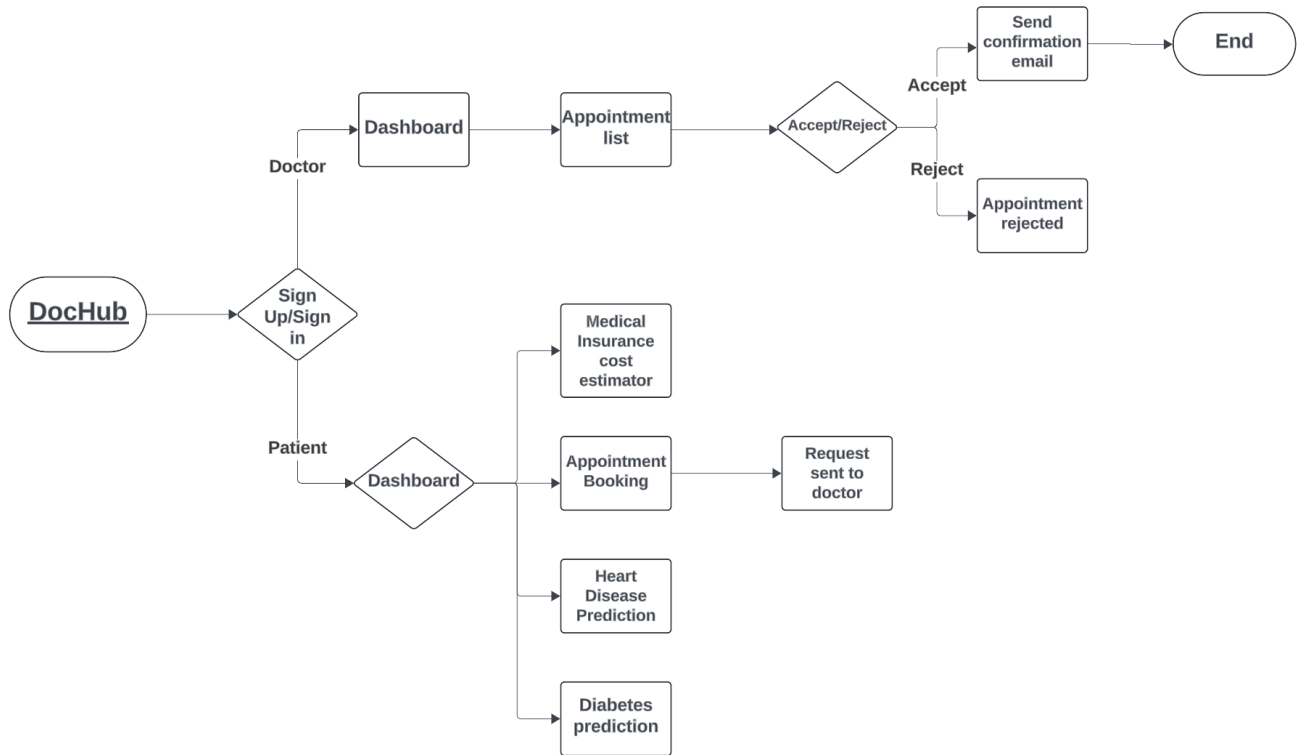


Fig 1: Website Design

### 3.2.2 Diabetes Model Flow Chart:

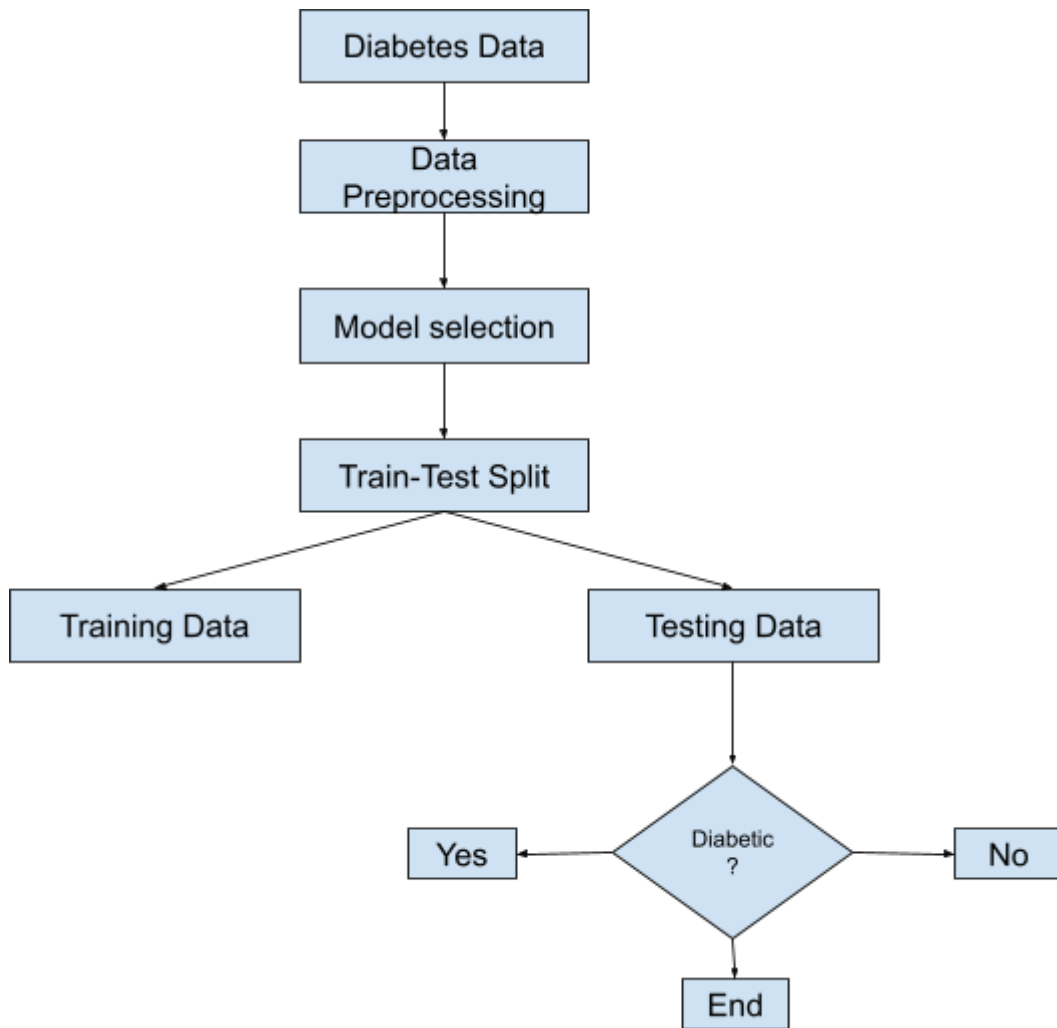


Fig 2: Diabetes Model flowchart

### 3.2.3 Heart Prediction Model Flow Chart:

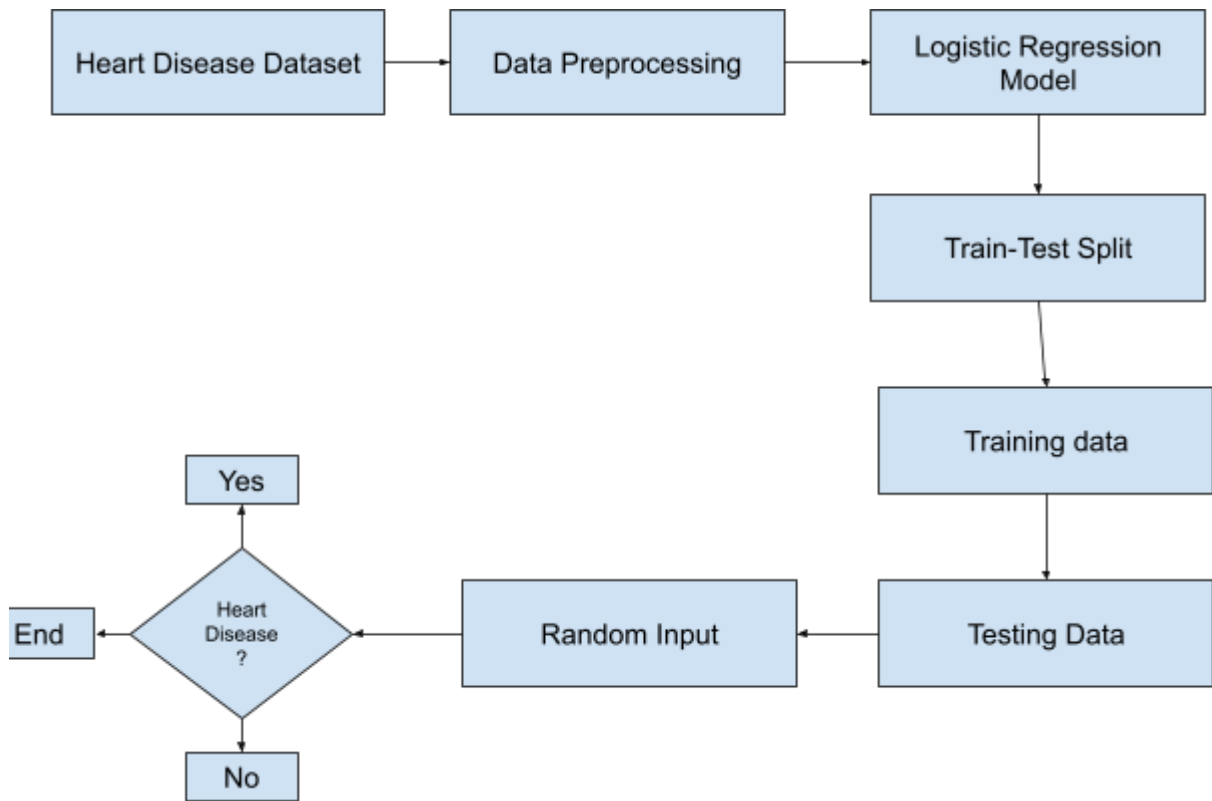
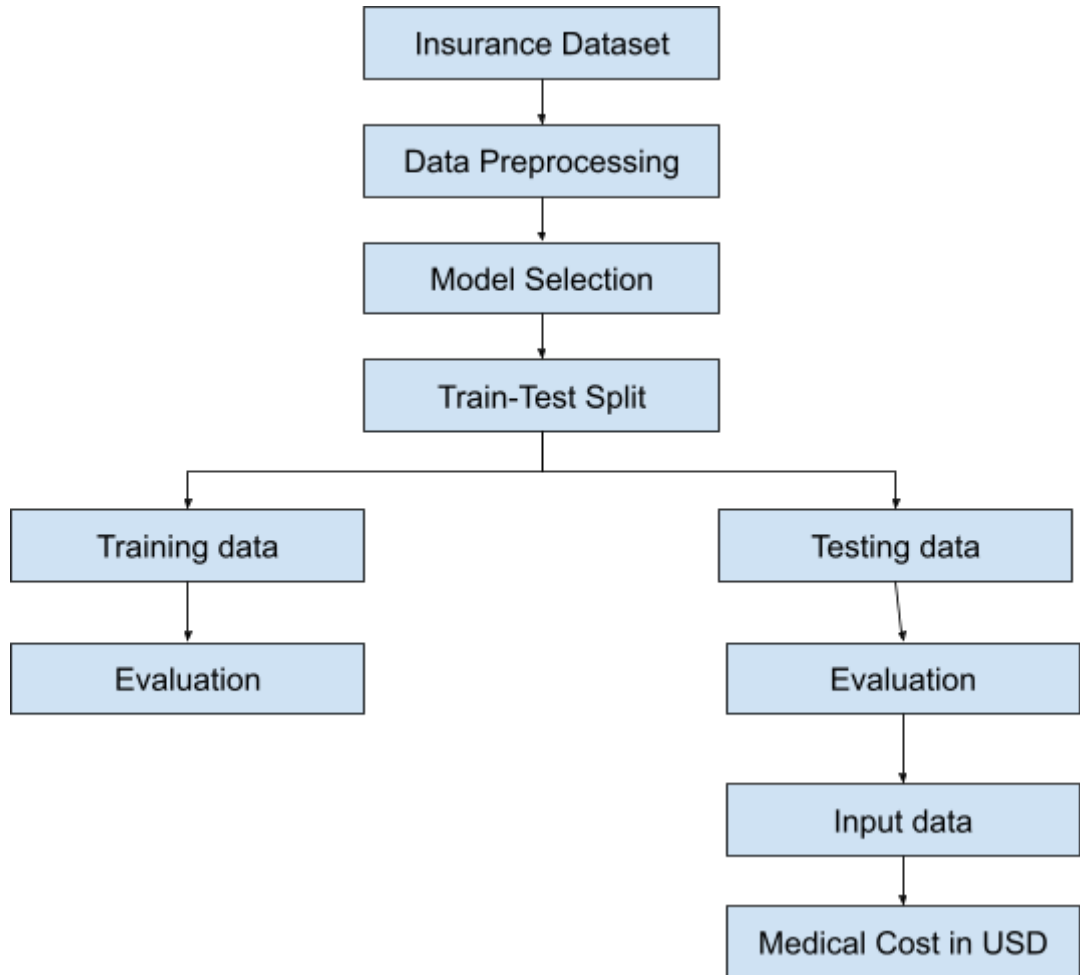


Fig 3: Heart Prediction Model Flowchart

### 3.2.4 Medical Insurance Cost Prediction Model Flow Chart:



*Fig 4: Medical Insurance cost estimator*

### 3.3 Data Preparation

#### **Data Collection:**

- Three Data set were taken from kaggle including Diabetes, Heart disease and Medical cost Insurance( in USD).

#### **Data Cleaning:**

- Missing or Null values were handled.
- Outliers were checked and removed.

#### **Feature Engineering:**

- Encoding of categorical variable was done.

#### **Data Splitting:**

- Dataset was divided into Test data and training data.

#### **Model-Specific Preparations:**

- Three models were selected for Heart Disease Prediction and Diabetes Prediction including Logistic Regression, Random forest and Support Vector Machine.
- For Medical insurance cost estimation model linear regression along with other three is used.

#### **Evaluation:**

- Both Training and Testing data were evaluated on 4 metrics: Accuracy, Precision, Recall



## 3.4 Implementation

### 3.4.1 Models:

1. I chose three datasets from Kaggle such as the Heart disease file, Diabetes file and medical insurance cost estimation file
2. In each of my data sets I tried 3 models and assessed their accuracy, precision, recall and F1 scores.
3. Model used includes:
  - Support Vector Machine: The objective of SVM is to locate the hyperplane with the largest margin, which is defined as the amount of space in between. This margin becomes important as it makes the model more resistant and less susceptible to specific data points.
  - Logistic Regression: Logistic regression uses the logistic function, also referred to as the sigmoid function, to model the probability associated with being in class 1.
  - Linear regression: Linear regression aims at finding the best-fitting linear connection (line) between the dependent variable (target) and one or more independent variables (features).
  - Random Forest: A random forest ensemble learning technique used for regression and classification. It forms a significant number of decision trees during training and then outputs the averaged prediction (regression) or mode (classification) of every tree.

### 3.4.2 Website:

I used Streamlit for developing the website. A bit of HTML and CSS was used to enhance the website. Then, I loaded the above models and created different pages for each model. I also created an appointment booking page for the patients and appointment list page for doctors. A user can sign up/sign in as patient or doctor and the data they enter will be stored into respective tables in my database. Then I used the smtplib library to send confirmation emails to patients once a doctor accepts their appointments.

### 3.4.3 Appointment Booking, Appointment Confirmation and Email Sending:

I created different dashboards: one for patient and one for doctor.

The one for patient includes following services:

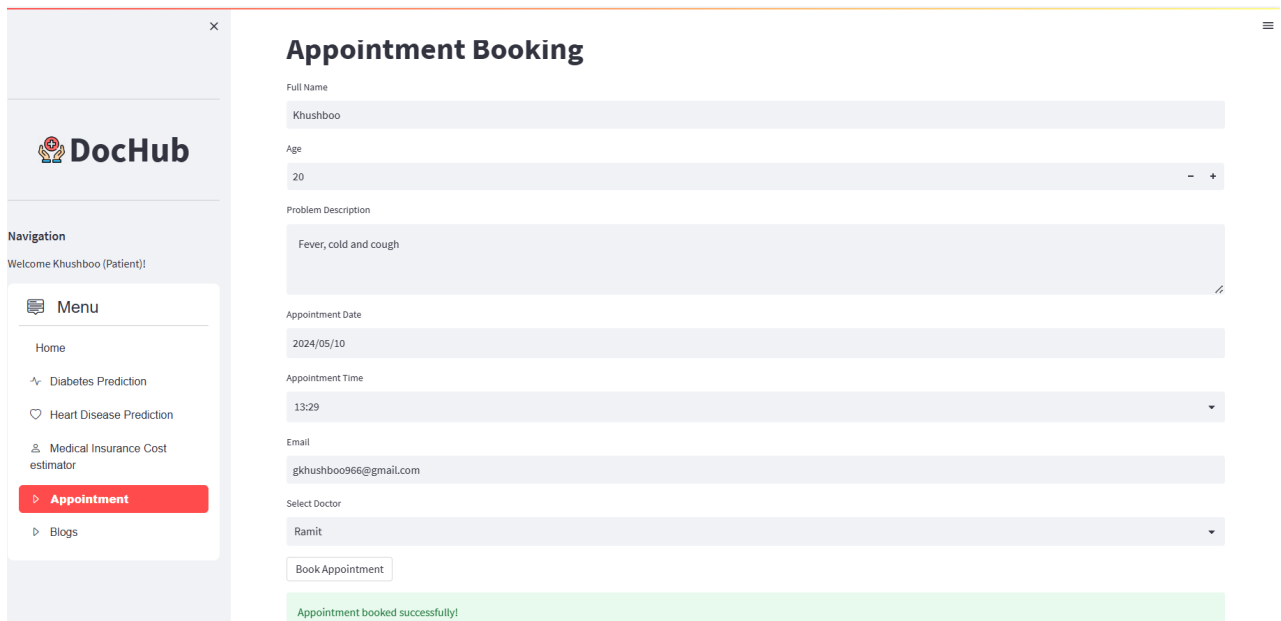
- Diabetes Prediction Model
- Heart Disease Prediction Model
- Medical Insurance Cost estimator
- Appointment Booking Page

The dashboard for doctor includes:

- Appointment Management

Doctor can use other services as well but signing up as patient. How it works:

1. Appointment Booking: Patient can book an appointment using appointment booking service provided on our patient dashboard. He needs to fill details such as Name, Age, Email, problem Description, can choose time and date and even the doctor from the drop-down menu that shows the list of doctors registered with us.



The screenshot displays the 'Appointment Booking' form on the DocHub patient dashboard. The form includes the following fields and values:

- Full Name:** Khushboo
- Age:** 20
- Problem Description:** Fever, cold and cough
- Appointment Date:** 2024/05/10
- Appointment Time:** 13:29
- Email:** gkhushboo966@gmail.com
- Select Doctor:** Ramit

A 'Book Appointment' button is located at the bottom of the form. Below the button, a green notification bar states: 'Appointment booked successfully!'.

The left sidebar of the dashboard shows the 'DocHub' logo and a navigation menu with the following items: Home, Diabetes Prediction, Heart Disease Prediction, Medical Insurance Cost estimator, **Appointment** (highlighted in red), and Blogs.

*Fig 5: Appointment Booking Confirmed*

2. Appointment Confirmation: When a doctor signs in, he can see the list of appointments that patients have booked on the appointment list page provided in the dashboard. He can accept/reject the appointment. Once the appointment is confirmed, an email from the dochub's support email is sent to the patient listing all the details of appointment and that his appointment is confirmed.

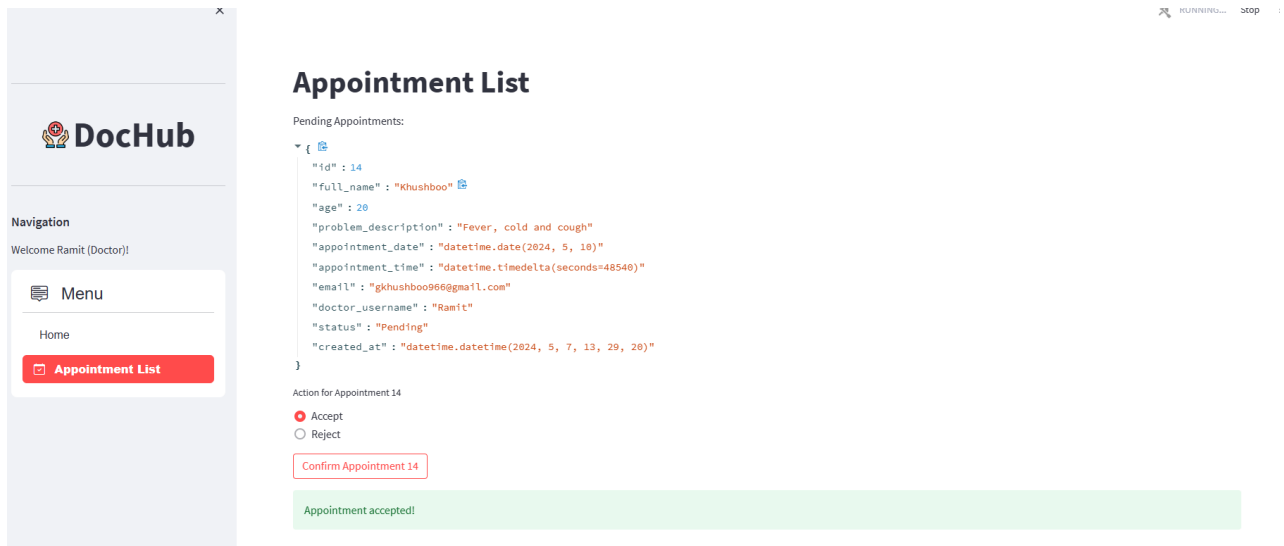


Fig 6: Appointment Confirmed by doctor

3. Email Sending: Details from the appointment booking page that the patient entered is stored in our database in appointment table. The table consists of additional column that specifies the status of the appointment: Pending, Accepted/ Rejected. Once the doctor accepts/ confirms the appointment, the status is updated to accepted and the appointment sending function of my code(send\_confirmation\_email) is triggered to send the respective email.

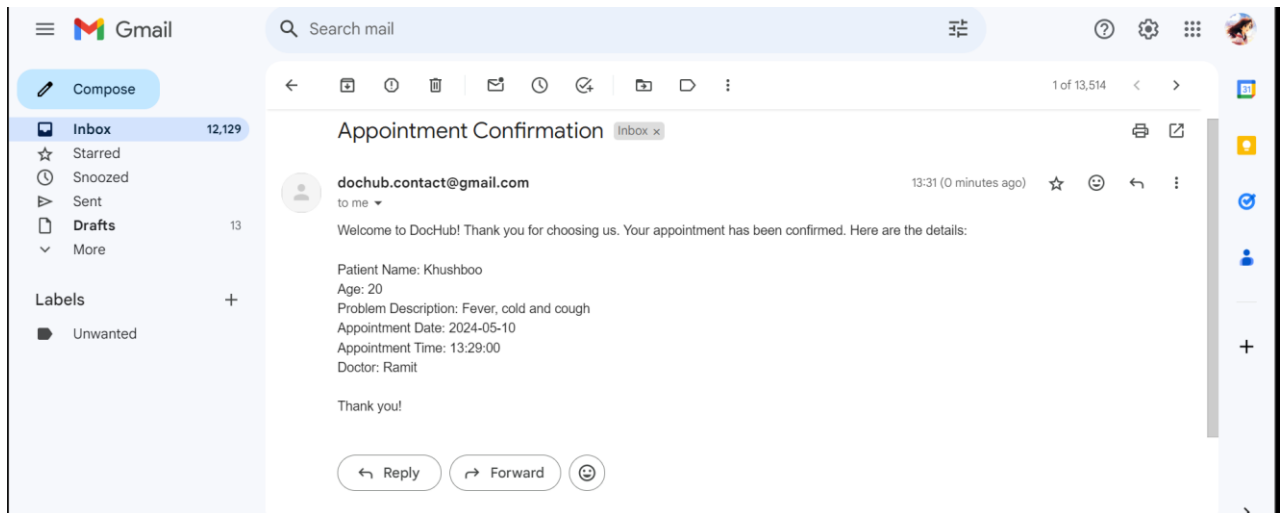


Fig 7: Confirmation Email Preview

### 3.4.4 Snippets of the models used:

#### Support Vector Machine

```
svm_classifier = SVC(kernel='linear')
svm_classifier.fit(X_train, Y_train)
Y_train_pred_svm = svm_classifier.predict(X_train)
Y_test_pred_svm = svm_classifier.predict(X_test)
```

#### Linear Regression

```
linear_reg_model = LinearRegression()
linear_reg_model.fit(X_train, Y_train)

Y_train_pred_linear_reg = linear_reg_model.predict(X_train)
Y_test_pred_linear_reg = linear_reg_model.predict(X_test)
```

## Logistic Regression

```
logreg_classifier = LogisticRegression()  
  
logreg_classifier.fit(X_train, Y_train)  
  
Y_train_pred_logreg = logreg_classifier.predict(X_train)  
Y_test_pred_logreg = logreg_classifier.predict(X_test)
```

## Random Forest

```
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=2)  
  
rf_classifier.fit(X_train, Y_train)  
  
Y_train_pred_rf = rf_classifier.predict(X_train)  
Y_test_pred_rf = rf_classifier.predict(X_test)
```

### 3.4.5 Snippets for user authentication and login/sign-up functionality:

#### Authenticate User Function

```
#Function to authenticate user type
def authenticate_user(conn, username, password, user_type):
    if conn:
        cursor = conn.cursor()
        try:
            if user_type == "Patient":
                cursor.execute("SELECT * FROM patients WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            elif user_type == "Doctor":
                cursor.execute("SELECT * FROM doctors WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            return False
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")
            return False
```

This function checks if the user is valid or not. It takes various parameters such as connection object, username, password and user type as input. Depending on whether the user is patient or doctors, it check the credentials against the respective tables. If they are found, it return true otherwise false. It also shows error messages if any error is found.

#### Login-User Functionality

```
#Function for login page
def login_user(conn, username, password, user_type):
    if conn:
        cursor = conn.cursor()
        try:
            if user_type == "Patient":
                cursor.execute("SELECT * FROM patients WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            elif user_type == "Doctor":
                cursor.execute("SELECT * FROM doctors WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            return False
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")
            return False
```

This function checks if the user can sign in. It takes parameters such as connection object, username, password and user type as input. Checks if the credentials are present in the database table. If they are found, it returns true, else returns false.

## Patient Sign-Up

```
#Function for patient signup page
def patient_signup(conn, username, email, password, age, gender):
    if conn:
        cursor = conn.cursor()
        try:
            cursor.execute("""
                INSERT INTO patients (username, email, password, age, gender) VALUES (%s, %s, %s, %s, %s)
            """, (username, email, password, age, gender))
            conn.commit()
            st.success("Patient account created successfully!")
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")
```

This function allows users to create new accounts. It takes a connection object and details like username, email, password, age and gender as input and inserts it in the patients table in the database. If they are saved, it commits the changes and displays a success message.

## Doctor Sign-up

```
#Function for doctor sign up
def doctor_signup(conn, username, email, password, age, gender):
    if conn:
        cursor = conn.cursor()
        try:
            cursor.execute("""
                INSERT INTO doctors (username, email, password, age, gender) VALUES (%s, %s, %s, %s, %s)
            """, (username, email, password, age, gender))
            conn.commit()
            st.success("Doctor account created successfully!")
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")
```

Similar to patient sign-up, it creates new accounts for doctors. It takes connection object and various other details as input and inserts them into the doctor's table in the database. If insertion is successful, it commits the changes to the database and displays a message indicating success.

### 3.4.6 Database

```
#Function for connection to database
def connect_to_database():
    try:
        conn = mysql.connector.connect(
            host="LAPTOP-SKØRFN1L",
            user="Khushboo",
            password="Khush@1102",
            database="dochoop_db"
        )
        return conn
    except mysql.connector.Error as err:
        st.error(f"Error: {err}")
        return None
```

This function tries to connect our website to a database. It uses host, user, password and database name to connect. If the connection is successful, it returns a connection object, else it returns an error.

Database name is dochoop\_db. Tables in the database are:

#### Patients

```
mysql> SELECT * FROM patients;
```

id	username	email	password	age	gender
1	Ranvir	ranvirsorrot3005@gmail.com	hello123	21	Male
2	Khushboo	gkhushboo966@gmail.com	hello123	20	Female
3	Vaishali	vaishaligoyal1511@gmail.com	hello123	30	Female
4	Sanchit	mehtasanchit200@gmail.com	hello123	24	Male
5	Divya	divyasorrot3005@gmail.com	hello123	19	Female



## Doctors

```
mysql> SELECT * FROM doctors;
```

id	username	email	password	age	gender
1	Ajay	ajay123@gmail.com	hello123	43	Male
2	Ramit	ramit123@gmail.com	hello123	52	Male
3	Aryan	aryan123@gmail.com	hello123	41	Male
4	Sonali	sonali123@gmail.com	hello123	34	Female
5	Manvi	manvi123@gmail.com	hello123	37	Female

## Appointment

```
mysql> SELECT * FROM appointments;
```

id	full_name	age	problem_description	appointment_date	appointment_time	email	doctor_user
name	status	created_at					
15	Ranvir Sorrot	26	Fever, Cold and Cough	2024-05-18	20:48:00	ranvirsorrot3005@gmail.com	Manvi
	Pending	2024-05-14 20:48:04					
16	Khushi Sharma	28	Food Poisoning	2024-05-15	10:30:00	gkhushboo966@gmail.com	Sonali
	Pending	2024-05-14 20:49:22					
17	Vaishali	35	Pain in neck and back	2024-05-17	16:00:00	vaishaligoyal1511@gmail.com	Manvi
	Pending	2024-05-14 20:50:31					

### 3.4.7 Appointment Booking, Confirmation and Gmail SMTP Server for sending confirmation email

#### Book Appointment Function

```
def book_appointment(conn, appointment_details):
    try:
        cursor = conn.cursor()
        # Print the SQL query to check its correctness
        query = """
        INSERT INTO appointments (full_name, age, problem_description, appointment_date, appointment_time, email, doctor_username, status)
        VALUES (%s, %s, %s, %s, %s, %s, %s, 'Pending')
        """
        print("SQL Query:", query)
        print("Appointment details:", appointment_details)

        cursor.execute(query, (appointment_details['full_name'], appointment_details['age'], appointment_details['problem_description'],
                                appointment_details['appointment_date'], appointment_details['appointment_time'], appointment_details['email'],
                                appointment_details['doctor']))
        conn.commit()
        print("Appointment booked successfully!")
        return True
    except mysql.connector.Error as e:
        print(f"Error booking appointment: {e}")
        return False
```

This function is used for booking appointments. It takes several details as input such as the patient's name, age, problem description, date and time. It inserts the details in the appointment table in the database. If insertion is successful, it books the appointment.

## Get Appointment List to Doctors

```
def get_appointments_for_doctor(conn, doctor_username):
    try:
        cursor = conn.cursor(dictionary=True)
        cursor.execute("""
            SELECT * FROM appointments WHERE doctor_username = %s AND status = 'Pending'
            """, (doctor_username,))
        appointments = cursor.fetchall()
        return appointments
    except mysql.connector.Error as e:
        print(f"Error fetching appointments: {e}")
        return []
```

This function fetches the appointment list for doctors. The ones with the pending status are reflected in the doctor's dashboard appointment list. If there are no pending appointments, it displays the message "No appointments found".

## Accept Appointment Function

```
def accept_appointment(conn, appointment_id):
    try:
        cursor = conn.cursor()
        cursor.execute("""
            UPDATE appointments SET status = 'Accepted' WHERE id = %s
            """, (appointment_id,))
        conn.commit()
        return True
    except mysql.connector.Error as e:
        print(f"Error accepting appointment: {e}")
        return False
```

This functionality is defined for acceptance of the appointment. This changes the appointment status from pending to accepted. If everything goes well, it displays a success message.

## Reject Appointment Function

```
def reject_appointment(conn, appointment_id):
    try:
        cursor = conn.cursor()
        cursor.execute("""
UPDATE appointments SET status = 'Rejected' WHERE id = %s
""", (appointment_id,))
        conn.commit()
        return True
    except mysql.connector.Error as e:
        print(f"Error rejecting appointment: {e}")
        return False
```

This function is used for changing the pending status of the appointment to rejected. If everything goes well, it displays a success message.

## Send Confirmation Email

```
def send_confirmation_email(conn, appointment_id):
    # Fetch appointment details
    cursor = conn.cursor(dictionary=True)
    cursor.execute("""
SELECT * FROM appointments WHERE id = %s
""", (appointment_id,))
    appointment = cursor.fetchone()

    if appointment:
        if appointment['status'] == 'Accepted':
            # Your email sending logic here
            try:
                # Setup SMTP server
                server = smtplib.SMTP('smtp.gmail.com', 587)
                server.starttls()
                # Replace 'your_email' and 'your_password' with your Gmail credentials
                server.login('dochub.contact@gmail.com', 'obhi qztx dvvo kqwf')

                # Email content
                msg = MIMEText()
                msg['From'] = 'dochub.contact@gmail.com'
                msg['To'] = appointment['email']
                msg['Subject'] = 'Appointment Confirmation'
```

```

body = f"Welcome to DocHub! Thank you for choosing us. Your appointment has been confirmed. Here are the details:\n\n"
      f"Patient Name: {appointment['full_name']}\n\n"
      f"Age: {appointment['age']}\n\n"
      f"Problem Description: {appointment['problem_description']}\n\n"
      f"Appointment Date: {appointment['appointment_date']}\n\n"
      f"Appointment Time: {appointment['appointment_time']}\n\n"
      f"Doctor: {appointment['doctor_username']}\n\n"
      f"Thank you!"

msg.attach(MIMEText(body, 'plain'))
# Send email
server.sendmail('dochub.contact@gmail.com', appointment['email'], msg.as_string())
server.quit()
print("Email sent successfully!")
return True
except Exception as e:
    print(f"Error sending email: {e}")
    return False
else:
    print("Appointment status is not 'Accepted'. Email not sent.")
    return False
else:
    print("Appointment not found.")
    return False

```

This function is used for sending confirmation email to patients. It takes all the necessary information from appointment table in the database, prepares an email with all the information and sends it to the email provided by the patient. Gmail's smtp server is used to send these emails.

### 3.4.8 Tools & Technologies Used

#### 3.4.8.1 Python

This involves a robust programmable language with several attributes. sentences. It is featureful enough so that it can be applied in such areas as meta-programming, meta-objects and object-oriented programming. They include several paradigms including contract generation and logic programming among other extensions. Python advantage has been provided by joining a power-classification reference computation with waste gathering. It is called late binding, which means the functionality of the complex document management system in It.

Python developers refuse such a patch to not so significant sites of C – Python that will produce just little gain, however being quite costly before execution. Additionally, depending on how important speed is to the Python programmer, they may either use mod-written C languages which is a fast compiler, to ensure that their time-sensitive jobs are sent promptly. The truth is, Python is one of the actual Python interpreters that convert all scripts in Python to C and calling it on a C-level API from this Python interpreter. Python designers wanted to produce a language enjoyable to work with. Regarding its functional nature, Python's design supports Lisp. These comprise of filtering maps, reducing lists, comprehension expressions, generators, sets, and dictionaries.

Interestingly, two modules (itertools and functools) of the standard library apply realistic haskell and standard ml toolsets.

### **3.4.8.2 Why Python?**

Python being a cross-platform program, our program will be created using python since we can work with it anywhere. It therefore makes it a comparative language like the English just like the English one. It includes many libraries; it uses natural language. Nevertheless, python is more complex than java and C++.

codes. Consequently, the length of python programmes is smaller than those of some of the other programming languages. Hence, python works on data as a major way through which it interacts with artificial intelligence. Articles arranged in a language, Python.

### **3.4.8.3 Streamlit**

Streamlit is an open-source Python library used for building web applications focused science, machine learning, and data visualization. It's designed to make it easy for data professionals, especially those with knowledge of Python, to create interactive web applications for sharing and demonstrating data-driven insights without requiring expertise in web development.

#### **3.4.8.4 Machine Learning**

Artificial Intelligence consists of machine learning that comprises creation of methods and structures to train unbiased machines towards forecasting or suggesting by themselves without programming them. The main objective of machine learning is that it enables computers to learn by themselves without any human intervention.

#### **3.4.8.5 Scikit-Learn**

Possibly, scikit-learn might be one of the best machine learning packages for python. It has several useful machine learning and mathematical modeling techniques such as division, slowing, merging and size reduction.

One of Scikit-Learn's most popular APIs. Scikit-machine Learn employs the Estimator API with all its learning algorithms since this offers a single interface and many ML apps. The data (which contains data) reads the scale as the object.

The tensor flow is an upgrading version for a standard library. Scikit-Learn is such a package that enables you to develop learning algorithms which can be utilized to determine the distance between an iron point and some object, among other things.

#### **3.4.8.6 HTML and CSS**

The project utilizes some features of HTML and CSS incorporated within st to create this website.

### 3.4.8.7 Smtplib

Smtplib is a Python library for sending email using Simple Mail Transfer Protocol (SMTP). It provides an easy way to send email from a Python script or program. Using smtplib you can connect to an SMTP server, authenticate if necessary, and send email (including attachments) if necessary. I am using this to send confirmation emails to patients.

### 3.4.8.8 MySQL

MySQL is a database management system language used to create relational databases. I am using this to create databases and tables for my website from which data will be saved here.

### 3.4.8.9 Other libraries

**NumPy:** It should be noted that for a lot of people NumPy – one of the most powerful libraries for numerical calculations in Python – would be impossible to work without. Linear algebra to operate on the large multidimensional arrays and the matrix functions. NumPy forms the basis of most scientific computing libraries in Python and its application cuts across fields such as machine learning, data science, and engineering processes. Mathematics of matrix, logic processing, shape, selection, I/O function, differential transform, base of linear algebra are efficiently implemented by it.

**Pandas:** Pandas are a python package designed for data manipulation and analysis. Also, it provides efficient operations on Series and Data frame data structures for organized data. For example, Pandas works well with labelled data, missing data, analysis and data exploration. It can be used together with other python libraries, for instance, NumPy, and matplotlib.

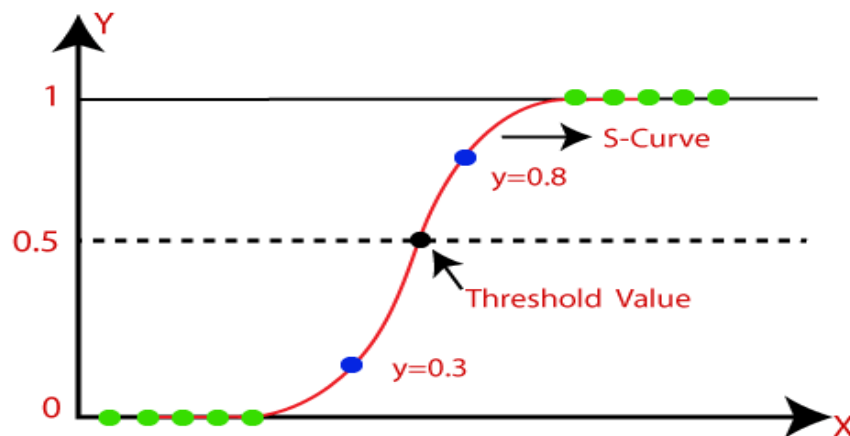
**Matplotlib:** Matplotlib is Python's 2D plotting library that creates different kinds of quality publication-level plots. It also generates plots, histograms, power spectra, bars, errors, and scatter plots among many others. On the other hand, matplotlib is very flexible and it can also be used with other libraries such as pandas, NumPy etc. in order to show a graphic of data.



## 3.4.9 Machine Learning Algorithms

### 3.4.9.1 Logistic Regression

Classification of a variable that has a probability of predicting a target variable by means of the supervised learning approach, logistic regression, is employed. Thus, this renders the goal or variable quantity as dichotomous denoting that there are two distinctive categories. Just put, the amount of the variable should be doubled, which means every bit of information would be presented in a binary format as 1 stands for 'yes' or failure, while 0 denotes 'no'. Predicted  $P(Y=1)$  is an element of  $X$  in a computed relapse model. This algorithm can be useful for mailing list monitoring, cancer forecasting, or diabetes prediction.



*Fig 8: Logistic Regression Curve*

#### Logistic Regression's Benefits

- **Simple and Interpretable:** It is rather simple to understand how different input features relate to this probability, using Logistic Regression.
- **Efficient for Linearly Separable Data:** Given that the straight line decision boundary is almost perfect, it proves to be functional when classes are normally linearly separable.
- **Probability Estimation:** Produces probabilities for outputs, enabling you to read the odds that an observation belongs in a certain class.
- **Low Variance:** When dealing with little data is more resistant, especially to overfitting the data provided.

### 3.4.9.2 Support Vector Machine

One of the supervised machine learning algorithms is the support vector machine (SVM) which uses a method known as the max-margin approach capable of performing both classification and regression tasks. It operates, in effect, by determining where one can draw a line in an n-dimensional space such that it cleanly bisects objects belonging to separate categories. Maximal separation of these classes happens in the way that the hyperplane is chosen, with the choice being supported by the fact that support vectors define the data points that are nearest the decision boundary.

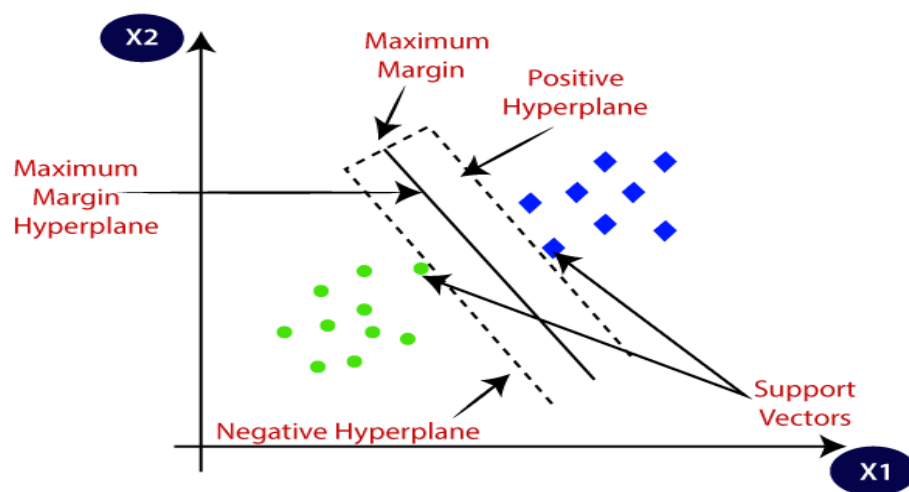


Fig 9: SVM working

#### Advantages of SVM:

- **Effective in High-Dimensional Spaces:** It is worth mentioning that when dealing with tasks containing many features, like image recognition, SVM shows the best performance in spaces of high dimensionality.
- **Versatility with Kernels:** SVM can deal with linear and non-linear relationships through the application of various kernels such as: polynomial kernel or the radial base function kernel.
- **Robust Performance:** First, SVM has proven itself to be a very powerful method, especially in situations when the separation of two classes is easily defined and clear.
- **Global Optimum:** SVM seeks out a hyperplane with the largest margin between classes thus resulting in a relatively robust solution despite being prone to local optima.

### **Disadvantages of SVM:**

- **Memory-Intensive:** SVM may consume a lot of memory, especially on larger sample sizes. It also makes it less desirable for huge scale applications.
- **Sensitivity to Noise:** Data noise may affect its performance, especially where the classes overlap, depending on SVM sensitivity.
- **Complexity in Choosing Kernels:** Choosing the right kernel and tuning its parameters is not an easy task as it will affect the model's performance.
- **Limited Interpretability:** Although, the decision boundary of SVM produces the optimal solution, it is also hard to explain compared with simpler models.

### **Applications of SVM:**

- **Classification:** SVM is a popular tool for solving both binary and multi-class classification issues across different industries including image recognition, text categorization, and bioinformatics.
- **Regression:** SVM can also be applied for regression tasks where it predicts a continuous or real-valued output rather than discrete class label.

### **3.4.9.3 Linear Regression**

Linear regression refers to a statistical technique of expressing mathematically the dependence of a stochastically selected phenomenon on one or several other quantitative indicators. The easiest type is simple linear regression with a single predictor, whose equation is just a straight line. With multiple linear regression, we have multiple independent variables and the equation becomes a hyperplane.

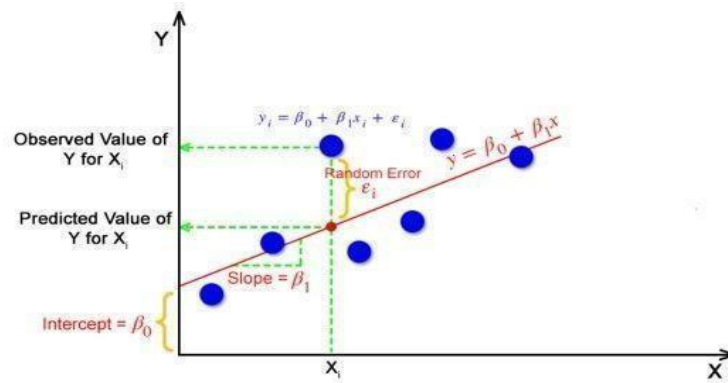


Fig 10: Linear Regression curve

### Advantages of Linear Regression:

- **Simplicity and Interpretability:** This linear regression method though straightforward. An intuitive understanding of the correspondence between input variables and outcome is easy to comprehend.
- **Efficiency for Simple Relationships:** Saturates if the relationship between the independent variable and the dependent variable is almost linear. This makes it more useful when the data is displayed in sequence form.
- **Fast Training and Prediction:** Linear regression is a quick method of training a model on large data sets, resulting in prediction solutions.
- **Useful for Feature Importance:** Therefore, the coefficients of the linear equation indicate on the influence of a particular feature in relation to the target variable.

### Disadvantages of Linear Regression:

- **Assumes Linearity:** Linear Regression assumes a linear relationship between the independent variables and the dependent variable. If the relationship is not linear, the model might not perform well.
- **Sensitive to Outliers:** Outliers in the data can strongly influence the model's coefficients and predictions, potentially leading to skewed results.
- **Limited Modeling of Complex Relationships:** Linear Regression may not capture complex relationships or interactions between features as effectively as more advanced models.

- **Homoscedasticity Assumption:** Assumes homoscedasticity, meaning that the variance of the residuals is constant across all levels of the independent variables. This assumption may not always hold.

#### **Applications of Linear Regression:**

- **Economics:** Linear regression is widely used in economics to model relationships between variables such as demand and supply, GDP and unemployment, etc.
- **Finance:** Linear regression is used in finance for tasks such as predicting stock prices, portfolio management, and risk assessment.

#### **3.4.9.4 Random Forest**

Random Forest is an ensemble learning method that is used to train many decision trees and helps in providing either the mode of classes or the modal prediction in classification problems.

#### **Advantages:**

- **High Accuracy:** However, Random Forest is known for its high accuracy in both classifications and regressions. Averaging the predictions by many trees reduces overfitting.
- **Robust to Overfitting:** The multiple trees in Random forest are better resisted hence providing an antidote for overfitting as opposed to individual decision trees.
- **Handles Non-Linearity and Interactions:** Random Forest is good for capturing non-linear relations as well as interactions between the characteristics of a dataset which in general are of high complexity.
- **Feature Importance:** MIT offers an insight into feature importance whereby users know the extent to which features affect the model's predictions.
- **Versatility:** Can be used in different tasks such as classification, regression, and feature selection.

### **Disadvantages:**

- **Computational Complexity:** Random forest is computationally exhaustive and takes a lot of time for real-time prediction, particularly for huge datasets.
- **Lack of Interpretability:** The internal operations of Random Forest provide for feature importance, but it is complicated to explain the model's decision-making process.

### **Applications:**

- **Classification:**
  - Image classification
  - Spam detection
  - Disease prediction
- **Regression:**
  - Predicting house prices
  - Financial forecasting
  - Demand prediction

## **3.5 Key Challenges**

Building a healthcare app has many challenges that need to be carefully considered to ensure that the app is not only functional and easy to use, but also legal and ethical. Some of the main problems encountered when developing applications and the strategies often used to solve them are:

### **3.5.1 Data Security and Privacy**

Protect sensitive medical information from unauthorized disclosure and access. data storage solutions and regular security reviews. Advise users on privacy practices.

### **3.5.2 Integration with existing systems**

Ensure the app integrates with existing healthcare systems, such as electronic health records (EHRs) and other health management tools. Use standard healthcare information exchange protocols such as HL7 or FHIR and work closely with IT providers to ensure compatibility and interoperability.

### **3.5.3 User Accessibility**

Create an application that is accessible to users of all ages and abilities, including people with disabilities. It includes features like best usability, commands, large texts, and making the interface simple and intuitive.

### **3.5.4 User Engagement and Protection**

Keep users engaged and using the app regularly. Notifications for health alerts and interactive content such as health tracking, activity or personal information.

### **3.5.5 Reliability and Performance**

Make sure the application performs well with no errors, no crashes and low downtime. including testing, integration testing and user detection. Increase capacity and reliability using cloud services.

### **3.5.6 Medical Accuracy and Responsibility**

Ensure that all medical information and advice provided by the app is accurate and up to date. Paraprofessionals review content, include evidence-based guidelines, and update information regularly. Clarify medical advice in practice to manage liability.

### **3.5.7 Multi-Language Support**

Provide multi-language support to appeal to different user groups. Culturally relevant content of the app to ensure greater integration and accessibility.

### **3.5.8 Financial Management**

Balance high development and maintenance costs with the need to make the application cost-effective for end users. Evaluate cloud solutions for application architecture, efficiency, and explore different models that can reduce costs without burdening users.

### **3.5.9 Response and continuous improvement**

Collect and integrate user input to improve the application. Quickly recommend and iterate based on this feedback using rapidly growing applications. Health awareness.



# **CHAPTER 04 : TESTING**

## **4.1 Testing Strategy:**

The datasets were gathered from kaggle.com. Models include:

**1.Diabetes Prediction Model:** Columns present in the dataset are:

- Pregnancies
- Glucose
- Blood Pressure
- Skin Thickness
- Insulin
- BMI
- DiabetesPedigreeFunction
- Age

**2.Heart Disease Prediction Model:** Columns present in the dataset are:

- Age
- Sex
- Chest pain types
- Resting blood pressure
- Cholesterol
- Fasting blood sugar
- Resting Electrocardiographic results
- Maximum heart rate achieved
- Exercise induced angina
- ST depression induced by exercise
- Slope
- Major vessel
- Thal

**3. Medical Insurance Cost Estimator Model:** Columns present in the dataset are:

- Age
- Sex
- BMI
- Children
- Smoker
- Region

## 4.2 Test Cases and Outcomes

### 4.2.1 Heart Disease Prediction Model:

For the Diabetes dataset, I used **Support Vector Machine, Logistic Regression and Random Forest** for prediction.

Models	Training Dataset				Testing Dataset			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.85	0.85	0.85	0.84	0.81	0.82	0.81	0.82
SVM	0.88	0.86	0.85	0.85	0.861	0.83	0.81	0.82
Random Forest	1.0	1.0	1.0	1.0	0.81	0.79	0.80	0.78

*Table Showing Comparison of Different models on basis of Accuracy, Precision, recall and F1 score.*

## Sample Test Cases:

### Sample test case 1:

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

input_data_as_numpy_array= np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = logreg_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')

[0]
The Person does not have a Heart Disease
```

### Sample test case 2:

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

input_data_as_numpy_array= np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = svm_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')

[0]
The Person does not have a Heart Disease
```

### Sample test case 3:

```
input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

input_data_as_numpy_array= np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = rf_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
    print('The Person does not have a Heart Disease')
else:
    print('The Person has Heart Disease')
```

[0]  
The Person does not have a Heart Disease

### 4.2.2 Diabetes Prediction Model:

For Heart Disease dataset, I used **Support Vector Machine, Logistic Regression and Random Forest** for prediction.

Model	Training Dataset				Testing Dataset			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
SVM	0.78	0.77	0.78	0.77	0.77	0.77	0.77	0.76
Logistic Regression	0.78	0.78	0.78	0.77	0.774	0.764	0.76	0.753

Random Forest	1.0	1.0	1.0	1.0	0.72	0.71	0.72	0.70
---------------	-----	-----	-----	-----	------	------	------	------

Table Showing Comparison of Different models on basis of Accuracy, Precision, recall and F1 score.

### Sample Test Cases:

#### Sample test case 1:

```

input_data = (5,140,72,30,175,25.8,0.890,51)
input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = logreg_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')

[0]
The person is not diabetic

```

#### Sample test case 2:

```

input_data = (5,140,72,30,175,25.8,0.890,51)
input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = svm_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')

[0]
The person is not diabetic

```

### Sample test case 3:

```
input_data = (5,140,72,30,175,25.8,0.890,51)
input_data_as_numpy_array = np.asarray(input_data)

input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = rf_classifier.predict(input_data_reshaped)
print(prediction)

if (prediction[0] == 0):
    print('The person is not diabetic')
else:
    print('The person is diabetic')

[0]
The person is not diabetic
```

### 4.2.3 Medical Insurance Cost Estimator model

For Medical Insurance cost dataset, I used **linear regression, Support Vector Machine, Logistic Regression and Random Forest** for prediction.

**Table for Linear Regression:**

Training Dataset			Testing Dataset		
MAE	MSE	R2	MAE	MSE	R2
4150.5	3617	0.79	4267.2	3833	0.78

Models	Training Dataset				Testing Dataset			
	Accuracy	Precision	Recall	F1 score	Accuracy	Precision	Recall	F1 score
Logistic Regression	0.25	0.41	0.26	0.16	0.04	0.03	0.03	0.03
SVM	0.97	0.96	0.97	0.96	0.03	0.02	0.03	0.031
Random Forest	1.0	1.0	1.0	1.0	0.02	0.03	0.03	0.031

*Table showing comparison of different models based on accuracy, precision, recall and F1 Score.*

Accuracy is most in linear regression as the in medical insurance cost estimation we need to find the cost which is a single value and not classify something.



## Sample Test Cases:

### Sample test case 1:

```
input_data = (31,1,25.74,0,1,0)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = regressor.predict(input_data_reshaped)
print(prediction)

print('The insurance cost is USD ', prediction[0])

[3760.0805765]
```

## 4.3 Email Sending And SMTPLIB Testing:

### 4.3.1 Booking an appointment

The screenshot shows the 'Appointment Booking' form in the DocHub application. The left sidebar contains the DocHub logo and a navigation menu with options: Home, Diabetes Prediction, Heart Disease Prediction, Medical Insurance Cost estimator, Appointment (highlighted in red), and Blogs. The main form area is titled 'Appointment Booking' and contains the following fields: Full Name (Khushboo), Age (20), Problem Description (Fever, cold and cough), Appointment Date (2024/05/10), Appointment Time (13:29), Email (gkhushboo966@gmail.com), and Select Doctor (Ramit). A 'Book Appointment' button is located below the doctor selection. A green success message at the bottom states 'Appointment booked successfully!'.

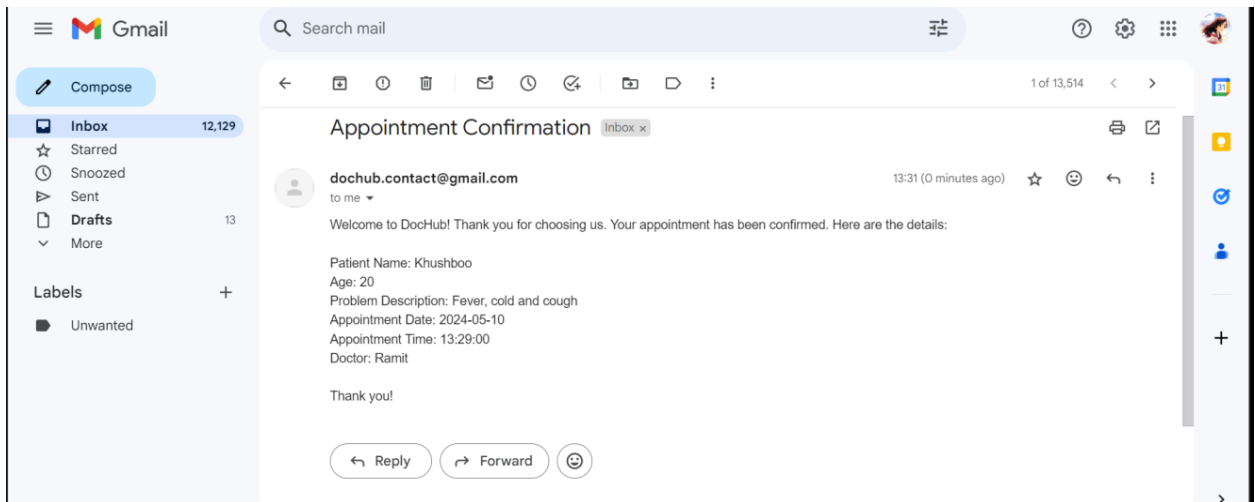
### 4.3.2 Acceptance of appointment by doctor

The screenshot shows the 'Appointment List' page in the DocHub application. The left sidebar is similar to the previous screenshot but with 'Appointment List' highlighted in red. The main content area is titled 'Appointment List' and shows 'Pending Appointments:'. A JSON object represents the appointment data: 

```
{ "id": 14, "full_name": "Khushboo", "age": 20, "problem_description": "Fever, cold and cough", "appointment_date": "datetime.date(2024, 5, 10)", "appointment_time": "datetime.timedelta(seconds=48540)", "email": "gkhushboo966@gmail.com", "doctor_username": "Ramit", "status": "Pending", "created_at": "datetime.datetime(2024, 5, 7, 13, 29, 20)" }
```

 Below the JSON, there is an 'Action for Appointment 14' section with radio buttons for 'Accept' (selected) and 'Reject'. A 'Confirm Appointment 14' button is present. A green success message at the bottom states 'Appointment accepted!'.

### 4.3.3 Confirmation Email



## 4.4 Sign-Up Page

### 4.4.1 Patient Sign-Up:

Are you a patient or a doctor?

- Patient  
 Doctor

Username

Divya

Email

divyasorrot3005@gmail.com

Password

\*\*\*\*\*



Age

19



Gender

Female



Signup

Patient account created successfully!

### 4.4.2 Doctor Sign-Up:

Are you a patient or a doctor?

- Patient  
 Doctor

Username

Manvi

Email

manvi123@gmail.com

Password

\*\*\*\*\*



Age

37



Gender

Female



Signup

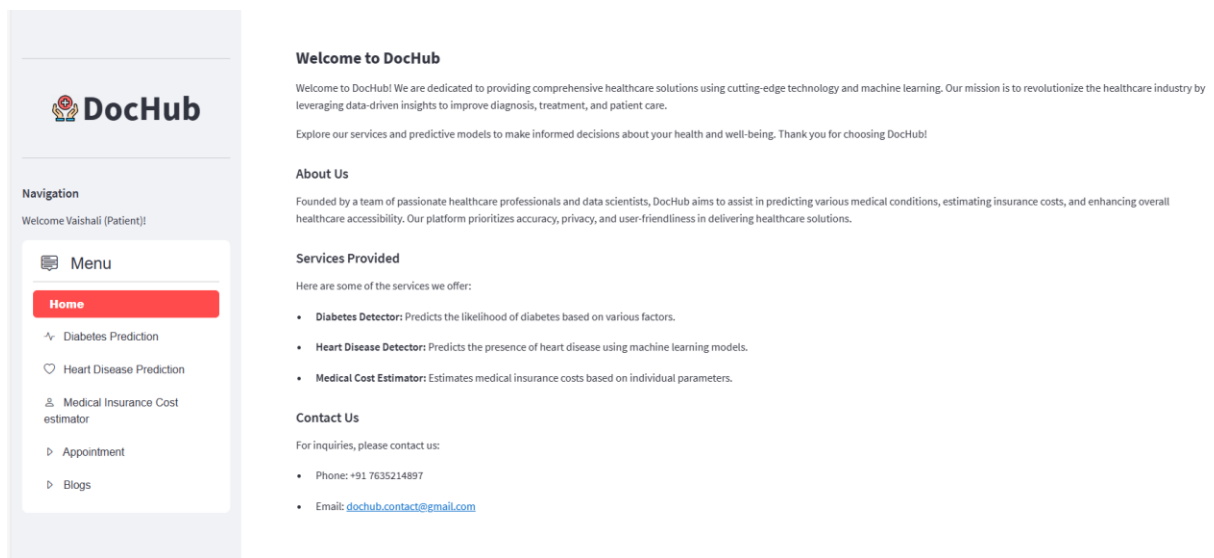
Doctor account created successfully!

# CHAPTER 05:- RESULT AND EVALUATION

## 5.1 Results

### 5.1.1 Patient's Dashboard:

#### Home Page



*Fig 11: Patient's Dashboard Home Page*

## Diabetes Detection page

The screenshot shows the 'Diabetes Prediction using ML' page. On the left is a navigation sidebar with the DocHub logo, a welcome message 'Welcome Vaishali (Patient)', and a menu with options: Home, Diabetes Prediction (highlighted in red), Heart Disease Prediction, Medical Insurance Cost estimator, Appointment, and Blogs. The main content area is titled 'Diabetes Prediction using ML' and contains a form with the following fields: Number of Pregnancies, Glucose Level, Blood Pressure value, Skin Thickness value, Insulin Level, BMI value, Diabetes Pedigree Function value, and Age of the Person. The values entered are 0 for Number of Pregnancies, Skin Thickness value, and Age of the Person; 0.0 for Diabetes Pedigree Function value and BMI value. A 'Diabetes Test Result' button is located at the bottom of the form.

Fig 12: Patient's Dashboard Diabetes Detection page

## Heart Disease Detection Page

The screenshot shows the 'Heart Disease Prediction using ML' page. On the left is a navigation sidebar with the DocHub logo, a welcome message 'Welcome Vaishali (Patient)', and a menu with options: Home, Diabetes Prediction, Heart Disease Prediction (highlighted in red), Medical Insurance Cost estimator, Appointment, and Blogs. The main content area is titled 'Heart Disease Prediction using ML' and contains a form with the following fields: Age, Sex (dropdown menu showing 'male'), Chest Pain types, Resting Blood Pressure, Serum Cholesterol in mg/dl, Fasting Blood Sugar > 120 mg/dl, Resting Electrocardiographic results, Maximum Heart Rate achieved, Exercise Induced Angina, ST depression induced by exercise, Slope of the peak exercise ST segment, and Major vessels colored by flourosopy. A legend below the form states: 'thal: 0 = normal; 1 = fixed defect; 2 = reversible defect'. The value entered for 'thal' is 0 (normal). A 'Heart Disease Test Result' button is located at the bottom of the form.

Fig 13: Patient's Dashboard Heart Disease Prediction Page

## Medical Insurance Cost Estimator Page

The screenshot shows the 'Medical Insurance Cost estimator' page. On the left is a navigation sidebar with the DocHub logo and a menu containing: Home, Diabetes Prediction, Heart Disease Prediction, **Medical Insurance Cost estimator** (highlighted in red), Appointment, and Blogs. The main content area is titled 'Medical Insurance Cost estimator' and contains a form with the following fields: Age (input field), Sex (dropdown menu with 'male' selected), BMI (input field with '0.00' and minus/plus buttons), Children (input field with '0' and minus/plus buttons), Smoker (dropdown menu with 'yes' selected), and Region (dropdown menu with 'southwest' selected). Below the form is a 'Insurance Cost' button.

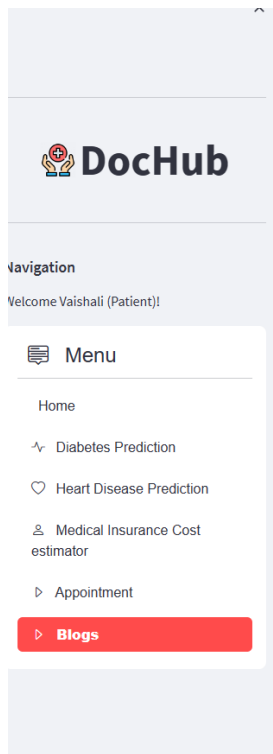
Fig 14: Patient's Dashboard Medical Insurance Cost Estimator Page

## Appointment Booking Page

The screenshot shows the 'Appointment Booking' page. On the left is a navigation sidebar with the DocHub logo and a menu containing: Home, Diabetes Prediction, Heart Disease Prediction, Medical Insurance Cost estimator, **Appointment** (highlighted in red), and Blogs. The main content area is titled 'Appointment Booking' and contains a form with the following fields: Full Name (input field), Age (input field with '0' and minus/plus buttons), Problem Description (text area), Appointment Date (input field with '2024/05/14'), Appointment Time (dropdown menu with '21:08' selected), Email (input field), and Select Doctor (dropdown menu with 'Ajay' selected). Below the form is a 'Book Appointment' button.

Fig 15: Patient's Dashboard Appointment Booking Page

## Blog Page



### DocHub Blog

#### 7 Steps to Take Care of Health

Maintaining good health involves a series of habits and lifestyle choices. Here are seven essential steps to take care of your health:

1. Eat a balanced diet.
2. Exercise regularly.
3. Get enough sleep.
4. Manage stress effectively.
5. Stay hydrated.
6. Avoid harmful habits (smoking, excessive drinking).
7. Regular health check-ups.

#### How to Take Care of Your Heart

Your heart is a vital organ that needs care. Here are a few tips to maintain a healthy heart:

- Eat a heart-healthy diet rich in fruits, vegetables, and whole grains.
- Exercise regularly to keep your heart strong.
- Control your blood pressure and cholesterol levels.
- Maintain a healthy weight.
- Avoid smoking and excessive alcohol intake.
- Manage stress effectively.
- Get enough sleep.

#### Why is Medical Cost Insurance Important?

Medical cost insurance is essential for several reasons:

- Protects against unexpected medical expenses.
- Provides access to quality healthcare.

*Fig 16: Patient's Dashboard Blogs Page*



## 5.1.2 Doctor's Dashboard

### Appointment List Page

**DocHub**

Navigation  
Welcome Manvi (Doctor!)

Menu

- Home
- Appointment List

### Appointment List

Pending Appointments:

```
{
  "id": 15,
  "full_name": "Ranvir Sorrot",
  "age": 26,
  "problem_description": "Fever, Cold and Cough",
  "appointment_date": "datetime.date(2024, 5, 18)",
  "appointment_time": "datetime.timedelta(seconds=74880)",
  "email": "ranvirsorrot3005@gmail.com",
  "doctor_username": "Manvi",
  "status": "Pending",
  "created_at": "datetime.datetime(2024, 5, 14, 20, 48, 4)"
}
```

Action for Appointment 15

Accept  
 Reject

Confirm Appointment 15

```
{
  "id": 17,
  "full_name": "Vaishali",
  "age": 35,
  "problem_description": "Pain in neck and back"
}
```

Fig 17: Doctor's Dashboard Appointment List Page

## **5.2 Comparison with Existing Solutions:**

The website I made using stream lit is DocHub. It consists of 5 sections:

### **Home:**

It contains subheadings including about us, services provided and contact info.

### **Diabetes Prediction:**

This page is connected with a diabetes prediction model made using a support vector machine in the backend. Users are needed to provide values in the various columns provided on the web page to check whether a person is Diabetic or not. The accuracy of the model is 0.77%. Accuracy is evaluated using accuracy score.

### **Heart Disease Prediction:**

This page is connected with heart disease prediction model made using logistic regression in the backend. Users are needed to provide values in the various columns provided on the webpage to check is a person is suffering from heart disease or not. The accuracy of model is 0.81%. Accuracy is evaluated using accuracy\_score.

### **Medical Insurance Cost prediction:**

This page is connected with a medical insurance cost prediction model made using linear regression in the backend. Users are needed to provide values in the various columns provided on the web page to predict medical cost. The accuracy of the model is 0.74%. Accuracy is evaluated using R2 score.

## Blogs:

This page contains sample blogs including:

- 7 steps to take care of health
- How to take care of your heart
- Why is Medical Cost Insurance important?
- Understanding mental health
- Tips for healthy lifestyle

### 5.2.1 Machine Learning Models:

In case of Diabetes Prediction, the highest accuracy is given by Support Vector machine, while in case of heart disease prediction model, highest accuracy is given by logistic regression and in case of medical cost insurance prediction is given by linear regression.

### Comparison of DocHub Model with existing models:

Name	Journal/year	Method/Accuracy (Their Model)	Method/accuracy (Developed Model(DocHub))
Diabetes prediction using machine learning and explainable AI techniques	NIH, 2022	Logistic Regression: 0.77  Random Forest: 0.78	Logistic regression: 0.774  Random Forest: 0.74

Diabetes Prediction using Machine Learning Algorithms	ICRTAC, 2019	Logistic Regression: 0.76 Random Forest: 0.72 SVM: 0.68	Logistic Regression: 0.774 Random Forest: 0.72 SVM: 0.77
Heart Disease Prediction Using Machine Learning	IJARST, 2019	SVM : 0.84 Random Forest: 0.80	SVM: 0.86 Random Forest: 0.805
Analysis of Heart Disease Prediction using Various Machine Learning Techniques	Research Gate, 2019	SVM : 0.656	SVM: 0.86
Health Insurance Cost Prediction by Using Machine Learning	SRNN, 2023	Linear Regression: 0.783 SVM: 0.072	Linear Regression: 0.781 SVM: 0.035
Medical Insurance Cost Prediction Using Machine Learning	Research Gate, 2023	Linear Regression: 0.744	Linear Regression: 0.781

# **CHAPTER 06:- CONCLUSIONS AND FUTURE SCOPE**

## **6.1 Conclusion**

Thus, it would be appropriate to conclude that DocHub can be considered as an everyday necessity especially for medical personnel, who utilize such software regularly to detect patients' diseases using their general data and symptoms. The healthcare plays an important role in the treatment of diseases in the modern day, hence a form of help to the healthcare to get informed about such program that will be helpful also to the user who doesn't wish to visit hospitals or clinics. Further addition of signs and other useful data to the They just have to specify the number of data values to enter their system before determining the type of disease that is inside one's body after just a few minutes. However, this project should have been accepted in the healthcare sector so that the doctor's work could be reduced and they have enough time to define the patient's condition. They include common diseases which if not treated can become fatal and could affect both the affected person and their families. Along with disease detection, it also provides a platform for appointment booking for patients. And for doctors it provides a platform for scheduling, accepting and rejecting appointments making it easier for doctors as well.

## **6.2 Future Scope**

The target of this project will be to create an accurate way of forecasting several diseases and medical insurance cost in the future. The above is predicted using a logistic regression technique, support vector machine, and linear regression technique in machine learning. More user-friendliness, better algorithms and better accuracy is the plan for future improvement.

# REFERENCES

1. IEEE "Machine Learning Methods Used in Ailment" by [www.Wikipedia.com](http://www.Wikipedia.com)
2. Priyanka Sonar, Prof. K. JayaMalini," DIABETES PREDICTION USING DIFFERENT MACHINE LEARNING APPROACHES", 2019 IEEE ,3rd International Conference on Computing Methodologies and Communication (ICCMC)
3. Archana Singh ,Rakesh Kumar, "Heart Disease Prediction Using Machine Learning Algorithms", 2020 IEEE, International Conference on Electrical and Electronics Engineering(ICE3)
4. Zhang Y., et al., "Deep Learning for Diagnosis of Chronic Diseases: A Review," IEEE Transactions on Neural Networks and Learning Systems, 2022.
5. Wang S., et al., "AI-Assisted Diagnosis of Neurological Disorders Using Deep Neural Networks," IEEE Journal of Biomedical and Health Informatics, 2023.
6. Li X., et al., "Machine Learning Applications in Cancer Prognosis and Prediction," IEEE Computational Intelligence Magazine, 2022.
7. Kumar A., et al., "Intelligent IoT Framework for Clinical Decision Support in Hospitals," IEEE Access, 2023.
8. Chen M., et al., "AI Techniques for Drug Discovery and Pharmaceutical Research," IEEE Transactions on Industrial Informatics, 2022.
9. Gupta A., et al., "A Comprehensive Review on Medical Image Analysis using Machine Learning," IEEE Reviews in Biomedical Engineering, 2021.
10. Lee J., et al., "Machine Learning Approaches to Improve Patient Monitoring," IEEE Transactions on Information Technology in Biomedicine, 2023.
11. Zhao H., et al., "Predictive Models for Heart Disease Classification Using Machine Learning," IEEE Access, 2022.
12. Patel V., et al., "AI in Retinal Health Imaging," IEEE Spectrum, 2022.
13. Liu Y., et al., "Deep Reinforcement Learning for Personalized Healthcare Applications," IEEE Journal on Selected Areas in Communications, 2023.
14. Zhang R., et al., "AI-Enhanced ECG Analysis for Arrhythmia Detection," IEEE Transactions on Biomedical Circuits and Systems, 2022.
15. Xu B., et al., "Machine Learning for Early Detection of Infectious Diseases," IEEE Transactions on Neural Networks and Learning Systems, 2023.

16. Yang G., et al., "Artificial Intelligence in Healthcare: Past, Present, and Future," IEEE Access, 2021.
17. Wang Y., et al., "Federated Learning for Mobile Health Data: A Case Study," IEEE Transactions on Industrial Informatics, 2023.
18. Sharma A., et al., "AI-based Systems for Osteoporosis Risk Prediction," IEEE Reviews in Biomedical Engineering, 2022.
19. Nguyen T., et al., "AI for Stroke Prediction and Rehabilitation," IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2023.
20. Morris D., et al., "ML Techniques for Personalized Cancer Treatment," IEEE Computational Intelligence Magazine, 2022.
21. Lee H., et al., "Integrating AI with Clinical Practice in Gastroenterology," IEEE Journal of Biomedical and Health Informatics, 2022.
22. Jain A., et al., "Automated Machine Learning in Healthcare Decision-Making," IEEE Access, 2023.
23. He K., et al., "AI in Dermatology and Skin Care," IEEE Spectrum, 2021.
24. Guo Y., et al., "Machine Learning for Pain Management in Clinical Environments," IEEE Transactions on Affective Computing, 2022.
25. Fu Z., et al., "Deep Learning in Genomics and Biomedicine," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2022.
26. Edwards B., et al., "AI for Improved Diagnostic Accuracy in Medical Imaging," IEEE Signal Processing Magazine, 2023.
27. Das R., et al., "ML Algorithms for Predicting Treatment Outcomes in Depression," IEEE Journal of Selected Topics in Signal Processing, 2022.

# APPENDICES

```
main.py | appointment.py x | hey.py
appointment.py > book_appointment
1
2 import mysql.connector
3 import smtplib
4 from email.mime.multipart import MIMEMultipart
5 from email.mime.text import MIMEText
6 import streamlit as st
7
8
9 def book_appointment(conn, appointment_details):
10     try:
11         cursor = conn.cursor()
12         # Print the SQL query to check its correctness
13         query = """
14         INSERT INTO appointments (full_name, age, problem_description, appointment_date, appointment_time,
15         VALUES (%s, %s, %s, %s, %s, %s, %s, %s, 'Pending')
16         """
17         print("SQL Query:", query)
18         print("Appointment details:", appointment_details)
19
20         cursor.execute(query, (appointment_details['full_name'], appointment_details['age'], appointment_d
21         | appointment_details['appointment_date'], appointment_details['appointment_time'], appointment
22         | appointment_details['doctor']))
23         conn.commit()
24         print("Appointment booked successfully!")
25         return True
26     except mysql.connector.Error as e:
27         print(f"Error booking appointment: {e}")
28         return False
29
30
31 def get_appointments_for_doctor(conn, doctor_username):
32     try:
33         cursor = conn.cursor(dictionary=True)
34         cursor.execute("""
35         SELECT * FROM appointments WHERE doctor_username = %s AND status = 'Pending'
36         """, (doctor_username,))
37         appointments = cursor.fetchall()
38         return appointments
39     except mysql.connector.Error as e:
40         print(f"Error fetching appointments: {e}")
41         return []
42
43 def accept_appointment(conn, appointment_id):
44     try:
45         cursor = conn.cursor()
46         cursor.execute("""
47         UPDATE appointments SET status = 'Accepted' WHERE id = %s
48         """, (appointment_id,))
49         conn.commit()
50         return True
51     except mysql.connector.Error as e:
52         print(f"Error accepting appointment: {e}")
53         return False
54
55 def reject_appointment(conn, appointment_id):
56     try:
57         cursor = conn.cursor()
```



```

        cursor = conn.cursor()
        cursor.execute("""
        UPDATE appointments SET status = 'Rejected' WHERE id = %s
        """, (appointment_id,))
        conn.commit()
        return True
    except mysql.connector.Error as e:
        print(f"Error rejecting appointment: {e}")
        return False

def send_confirmation_email(conn, appointment_id):
    # Fetch appointment details
    cursor = conn.cursor(dictionary=True)
    cursor.execute("""
    SELECT * FROM appointments WHERE id = %s
    """, (appointment_id,))
    appointment = cursor.fetchone()

    if appointment:
        if appointment['status'] == 'Accepted':
            # Your email sending logic here
            try:
                # Setup SMTP server
                server = smtplib.SMTP('smtp.gmail.com', 587)
                server.starttls()
                # Replace 'your_email' and 'your_password' with your Gmail credentials
                server.login('dochub.contact@gmail.com', 'obhi qztx dvvo kwf')

                msg = MIMEMultipart()
                msg['From'] = 'dochub.contact@gmail.com'
                msg['To'] = appointment['email']
                msg['Subject'] = 'Appointment Confirmation'

                body = f"Welcome to DocHub! Thank you for choosing us. Your appointment has been confirmed\n"
                body += f"Patient Name: {appointment['full_name']}\n"
                body += f"Age: {appointment['age']}\n"
                body += f"Problem Description: {appointment['problem_description']}\n"
                body += f"Appointment Date: {appointment['appointment_date']}\n"
                body += f"Appointment Time: {appointment['appointment_time']}\n"
                body += f"Doctor: {appointment['doctor_username']}\n"
                body += f"Thank you!"

                msg.attach(MIMEText(body, 'plain'))
                # Send email
                server.sendmail('dochub.contact@gmail.com', appointment['email'], msg.as_string())
                server.quit()
                print("Email sent successfully!")
                return True
            except Exception as e:
                print(f"Error sending email: {e}")
                return False
        else:
            print("Appointment status is not 'Accepted'. Email not sent.")

```

```

        return True
    except Exception as e:
        print(f"Error sending email: {e}")
        return False
    else:
        print("Appointment status is not 'Accepted'. Email not sent.")
        return False
else:
    print("Appointment not found.")
    return False

```

```

main.py > ...
1  import pickle
2  import streamlit as st
3  from streamlit_option_menu import option_menu
4  import mysql.connector
5  from appointment import book_appointment, get_appointments_for_doctor, accept_appointment, reject_appointm
6
7
8  if 'user_type' not in st.session_state:
9      st.session_state.user_type = None
10
11
12
13  #Function for connection to database
14  def connect_to_database():
15      try:
16          conn = mysql.connector.connect(
17              host="LAPTOP-SK0RFN1L",
18              user="Khushboo",
19              password="Khush@1102",
20              database="dochoop_db"
21          )
22          return conn
23      except mysql.connector.Error as err:
24          st.error(f"Error: {err}")
25          return None
26
27  #Function to authetnticate user type
28  def authenticate_user(conn, username, password, user_type):
29      if conn:

```

```

def authenticate_user(conn, username, password, user_type):
    if conn:
        cursor = conn.cursor()
        try:
            if user_type == "Patient":
                cursor.execute("SELECT * FROM patients WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            elif user_type == "Doctor":
                cursor.execute("SELECT * FROM doctors WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            return False
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")
            return False

```

#Function for patient signup page

```

def patient_signup(conn, username, email, password, age, gender):

```

```

    if conn:
        cursor = conn.cursor()
        try:
            cursor.execute("""
            INSERT INTO patients (username, email, password, age, gender) VALUES (%s, %s, %s, %s, %s)
            """, (username, email, password, age, gender))
            conn.commit()
            st.success("Patient account created successfully!")

```

```

        except mysql.connector.Error as err:
            st.error(f"Error: {err}")

```

#Function for doctor sign up

```

def doctor_signup(conn, username, email, password, age, gender):

```

```

    if conn:
        cursor = conn.cursor()
        try:
            cursor.execute("""
            INSERT INTO doctors (username, email, password, age, gender) VALUES (%s, %s, %s, %s, %s)
            """, (username, email, password, age, gender))
            conn.commit()
            st.success("Doctor account created successfully!")
        except mysql.connector.Error as err:
            st.error(f"Error: {err}")

```

#Function for login page

```

def login_user(conn, username, password, user_type):

```

```

    if conn:
        cursor = conn.cursor()
        try:
            if user_type == "Patient":
                cursor.execute("SELECT * FROM patients WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():
                    return True
            elif user_type == "Doctor":
                cursor.execute("SELECT * FROM doctors WHERE username = %s AND password = %s", (username, password))
                if cursor.fetchone():

```

```

        if cursor.fetchone():
            return True
        return False
    except mysql.connector.Error as err:
        st.error(f"Error: {err}")
        return False

def login_page():
    st.markdown("**Are you a patient or a doctor?**")

    user_type = st.radio("", ("Patient", "Doctor"))

    username_login = st.text_input("Username")
    password_login = st.text_input("Password", type="password")
    if st.button("Login"):
        conn = connect_to_database()
        if login_user(conn, username_login, password_login, user_type):
            st.session_state.username = username_login
            st.session_state.user_type = user_type
            st.session_state.logged_in = True
            st.success(f"Logged in as {username_login} ({user_type})")
            return True
        else:
            st.error("Invalid username or password")
            return False

```

```

1  conn = connect_to_database()
2
3  def appointment_booking_page(conn):
4      st.title("Appointment Booking")
5
6      # Appointment details form
7      name = st.text_input("Full Name")
8      age = st.number_input("Age", min_value=0, max_value=150, step=1)
9      problem = st.text_area("Problem Description")
0      date = st.date_input("Appointment Date")
1      time = st.time_input("Appointment Time")
2      email = st.text_input("Email")
3
4      doctor_list = get_doctor_list(conn)
5      selected_doctor = st.selectbox("Select Doctor", doctor_list)
6
7      if st.button("Book Appointment"):
8          appointment_details = {
9              'full_name': name,
0              'age': age,
1              'problem_description': problem,
2              'appointment_date': date,
3              'appointment_time': time,
4              'email': email,
5              'doctor': selected_doctor
6          }
7          if book_appointment(conn, appointment_details):
8              st.success("Appointment booked successfully!")
9

```

```

    },
    if book_appointment(conn, appointment_details):
        st.success("Appointment booked successfully!")
    else:
        st.error("Failed to book appointment. Please try again.")

def get_doctor_list(conn):
    cursor = conn.cursor()
    cursor.execute("SELECT username FROM doctors")
    doctors = cursor.fetchall()
    doctor_list = [doctor[0] for doctor in doctors]
    return doctor_list

def appointment_list_page(conn):
    st.title("Appointment List")

    doctor_username = st.session_state.username
    appointments = get_appointments_for_doctor(conn, doctor_username)

    if appointments:
        st.write("Pending Appointments:")
        for appointment in appointments:
            st.write(appointment)
            action = st.radio(f"Action for Appointment {appointment['id']}", ("Accept", "Reject"))
            if st.button(f"Confirm Appointment {appointment['id']}"):
                if action == "Accept":
                    accept_appointment(conn, appointment['id'])
                    st.success("Appointment accepted!")
                    send_confirmation_email(conn, appointment['id'])
                elif action == "Reject":
                    reject_appointment(conn, appointment['id'])
                    st.warning("Appointment rejected!")
            else:
                st.write("No pending appointments.")

def signup():
    st.markdown("***Are you a patient or a doctor?***")

    user_type = st.radio("", ("Patient", "Doctor"))

    username = st.text_input("Username")
    email = st.text_input("Email")
    password = st.text_input("Password", type="password")
    age = st.number_input("Age", min_value=0, max_value=150, step=1)
    gender = st.selectbox("Gender", ("Male", "Female", "Other"))

    if st.button("Signup"):
        conn = connect_to_database()
        if user_type == "Patient":
            patient_signup(conn, username, email, password, age, gender)
        elif user_type == "Doctor":
            doctor_signup(conn, username, email, password, age, gender)

```

```

        st.session_state.signed_up = True
        return True
    return False

diabetes_model = pickle.load(open(r'C:\Users\gkhus\project\saved models\diabetes_model.sav', 'rb'))

heart_disease_model = pickle.load(open(r'C:\Users\gkhus\project\saved models\heart_model.sav', 'rb'))

insurance_model = pickle.load(open(r'C:\Users\gkhus\project\saved models\Insurance.sav', 'rb'))

def is_user_logged_in():
    return "username" in st.session_state

st.set_page_config(
    page_title="DocHub",
    page_icon="🏠",
    layout="wide",
    initial_sidebar_state="expanded"
)

```

```

conn = connect_to_database()

selected = None
with st.sidebar:
    st.markdown("---")
    st.markdown(
        '<div style="display: flex; justify-content: center; align-items: center;">'
        'label{color: #ff5733;}</style>', unsafe_allow_html=True)
    Pregnancies = st.text_input('Number of Pregnancies', value='0')
    SkinThickness = st.text_input('Skin Thickness value', value='0')
    DiabetesPedigreeFunction = st.text_input('Diabetes Pedigree Function value', value='0.0')

with col2:
    st.markdown('<style>label{color: #33ff8b;}</style>', unsafe_allow_html=True)
    Glucose = st.text_input('Glucose Level', value='0')
    Insulin = st.text_input('Insulin Level', value='0')
    Age = st.text_input('Age of the Person', value='0')

with col3:
    st.markdown('<style>label{color: #338bff;}</style>', unsafe_allow_html=True)
    BloodPressure = st.text_input('Blood Pressure value', value='0')
    BMI = st.text_input('BMI value', value='0.0')

diab_diagnosis = ''

if st.button('Diabetes Test Result'):
    diab_prediction = diabetes_model.predict([[Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin]])

    if (diab_prediction[0] == 1):
        diab_diagnosis = 'The person is diabetic'

```



```

        st.markdown("<h3 style='color: red;*>The person is diabetic</h3>", unsafe_allow_html=True)
    else:
        diab_diagnosis = 'The person is not diabetic'
        st.markdown("<h3 style='color: green;*>The person is not diabetic</h3>", unsafe_allow_html=True)
    pass

# Home page for website
if selected == 'Home':

    st.markdown("<h1 style='font-size: 24px;*>Welcome to DocHub</h1>", unsafe_allow_html=True)

    st.write("""
    Welcome to DocHub! We are dedicated to providing comprehensive healthcare solutions using cutting-edge
    Explore our services and predictive models to make informed decisions about your health and well-being
    """)

    st.markdown("<h2 style='font-size: 20px;*>About Us</h2>", unsafe_allow_html=True)
    st.write("""
    Founded by a team of passionate healthcare professionals and data scientists, DocHub aims to assist in
    """)
    st.markdown("<h2 style='font-size: 20px;*>Services Provided</h2>", unsafe_allow_html=True)

    st.write("Here are some of the services we offer:")
    st.write("- **Diabetes Detector:** Predicts the likelihood of diabetes based on various factors.")
    st.write("- **Heart Disease Detector:** Predicts the presence of heart disease using machine learning")
    st.write("- **Medical Cost Estimator:** Estimates medical insurance costs based on individual parameters")

```

```

    st.markdown("<h2 style='font-size: 20px;*>Contact Us</h2>", unsafe_allow_html=True)
    st.write("For inquiries, please contact us:")

    st.write("- Phone: +91 7635214897")
    st.write("- Email: dochub.contact@gmail.com")
    pass

# Heart disease Prediction Page
elif (selected == 'Heart Disease Prediction'):

    st.title('Heart Disease Prediction using ML')

    col1, col2, col3 = st.columns(3)

    with col1:
        age = st.text_input('Age')

    with col2:
        sex = st.selectbox('Sex', ['male', 'female'])

    with col3:
        cp = st.text_input('Chest Pain types')

    with col1:
        trestbps = st.text_input('Resting Blood Pressure')

    with col2:

```

```

with col2:
    chol = st.text_input('Serum Cholestoral in mg/dl')

with col3:
    fbs = st.text_input('Fasting Blood Sugar > 120 mg/dl')

with col1:
    restecg = st.text_input('Resting Electrocardiographic results')

with col2:
    thalach = st.text_input('Maximum Heart Rate achieved')

with col3:
    exang = st.text_input('Exercise Induced Angina')

with col1:
    oldpeak = st.text_input('ST depression induced by exercise')

with col2:
    slope = st.text_input('Slope of the peak exercise ST segment')

with col3:
    ca = st.text_input('Major vessels colored by flourosopy')

with col1:
    thal = st.selectbox('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect', ['0 (normal)', '1

```

```

with col1:
    thal = st.selectbox('thal: 0 = normal; 1 = fixed defect; 2 = reversable defect', ['0 (normal)', '1

age = int(age) if age else 0
cp = int(cp) if cp else 0
trestbps = int(trestbps) if trestbps else 0
chol = int(chol) if chol else 0
fbs = int(fbs) if fbs else 0
restecg = int(restecg) if restecg else 0
thalach = int(thalach) if thalach else 0
exang = int(exang) if exang else 0
oldpeak = float(oldpeak) if oldpeak else 0.0
slope = float(slope) if slope else 0.0
ca = int(ca) if ca else 0

heart_diagnosis = ''

if st.button('Heart Disease Test Result'):
    heart_prediction = heart_disease_model.predict([[age, sex, cp, trestbps, chol, fbs, restecg, thalach

        if (heart_prediction[0] == 1):
            heart_diagnosis = 'The person is having heart disease'
        else:
            heart_diagnosis = 'The person does not have any heart disease'

st.success(heart_diagnosis)
pass

```

```

412 #Medical Insurance Cost estimator page
413 elif (selected == 'Medical Insurance Cost estimator'):
414
415     st.title('Medical Insurance Cost estimator')
416
417     col1, col2, col3 = st.columns(3)
418
419     with col1:
420         age = st.text_input('Age')
421
422     with col2:
423         sex = st.selectbox('Sex', ['male', 'female'])
424
425     with col3:
426         bmi = st.number_input('BMI')
427
428     with col1:
429         children = st.number_input('Children', min_value=0, step=1)
430
431     with col2:
432         smoker = st.selectbox('Smoker', ['yes', 'no'])
433
434     with col3:
435         region = st.selectbox('Region', ['southwest', 'southeast', 'northwest', 'northeast'])
436
437     insurance_cost = ''
438
439     if st.button('Insurance Cost'):

```

```

39     if st.button('Insurance Cost'):
40         try:
41             sex_encoded = 1 if sex.lower() == 'male' else 0
42             smoker_encoded = 1 if smoker.lower() == 'yes' else 0
43
44             age = float(age)
45             bmi = float(bmi)
46             children = int(children)
47
48             sex_encoded = 1 if sex.lower() == 'male' else 0
49             smoker_encoded = 1 if smoker.lower() == 'yes' else 0
50
51
52             region_map = {'southwest': 0, 'southeast': 1, 'northwest': 2, 'northeast': 3}
53             region_encoded = region_map.get(region.lower(), -1) # Assign -1 for unknown regions
54
55             input_data = [[age, sex_encoded, bmi, children, smoker_encoded, region_encoded]]
56
57
58             insurance_prediction = insurance_model.predict(input_data)
59
60
61             insurance_cost = f"Predicted Insurance Cost: {insurance_prediction[0]}"
62
63         except ValueError as e:
64             insurance_cost = "Error: Please ensure all input values are in numeric format"
65
66     st.success(insurance_cost)

```

```

main.py > ...
467     pass
468 #Blogs page for website
469 elif selected == 'Blogs':
470
471     st.title('DocHub Blog')
472
473
474     st.markdown('<h3 style="color: #ff5733;">7 Steps to Take Care of Health</h3>', unsafe_allow_html=True)
475     st.write("""
476     Maintaining good health involves a series of habits and lifestyle choices. Here are seven essential st
477
478     1. Eat a balanced diet.
479     2. Exercise regularly.
480     3. Get enough sleep.
481     4. Manage stress effectively.
482     5. Stay hydrated.
483     6. Avoid harmful habits (smoking, excessive drinking).
484     7. Regular health check-ups.
485     """)
486
487
488     st.markdown('<h3 style="color: #33ff8b;">How to Take Care of Your Heart</h3>', unsafe_allow_html=True)
489     st.write("""
490     Your heart is a vital organ that needs care. Here are a few tips to maintain a healthy heart:
491
492     - Eat a heart-healthy diet rich in fruits, vegetables, and whole grains.
493     - Exercise regularly to keep your heart strong.
494     - Control your blood pressure and cholesterol levels.
495     - Maintain a healthy weight.

```

```

main.py > ...
95     - Maintain a healthy weight.
96     - Avoid smoking and excessive alcohol intake.
97     - Manage stress effectively.
98     - Get enough sleep.
99     """)
100
101     st.markdown('<h3 style="color: #338bff;">Why is Medical Cost Insurance Important?</h3>', unsafe_allow_
102     st.write("""
103     Medical cost insurance is essential for several reasons:
104
105     - Protects against unexpected medical expenses.
106     - Provides access to quality healthcare.
107     - Helps in managing and planning for healthcare costs.
108     - Ensures financial security during medical emergencies.
109     - Allows for preventive care and routine check-ups.
110     """)
111
112
113     st.markdown('<h3 style="color: #8b33ff;">Understanding Mental Health</h3>', unsafe_allow_html=True)
114     st.write("""
115     Mental health is as important as physical health. Here are some key aspects of understanding mental he
116
117     - Mental health includes emotional, psychological, and social well-being.
118     - Mental illnesses are common and can affect anyone.
119     - Seek professional help if you experience persistent mood changes or behavioral issues.
120     - Practice self-care, stress management, and healthy coping strategies.
121     - Reduce stigma and promote mental health awareness in society.
122     """)
123

```

```

524     st.markdown('<h3 style="color: #ff8b33;">Tips for a Healthy Lifestyle</h3>', unsafe_allow_html=True)
525     st.write("""
526     Embracing a healthy lifestyle can significantly impact your well-being. Here are some tips:
527
528     - Prioritize balanced nutrition and hydration.
529     - Stay physically active with regular exercise or movement.
530     - Get enough quality sleep each night.
531     - Manage stress through relaxation techniques or hobbies.
532     - Foster social connections and maintain healthy relationships.
533     - Practice mindfulness and engage in activities that bring joy.
534     """)
535     pass
536
537     #Appointment Booking Page if user is a patient
538     elif selected == "Appointment":
539         conn = connect_to_database() # Establish database connection
540         print("Database connection established successfully.")
541         appointment_booking_page(conn)
542
543         conn.close()
544
545     #Login Page for Website
546     elif selected == "login":
547         st.title('Login Page')
548         if login_page():
549             selected = "Home"
550
551     #SignUp page for website
552     elif selected == "signup":
553
554
555     #Login Page for Website
556     elif selected == "login":
557         st.title('Login Page')
558         if login_page():
559             selected = "Home"
560
561     #SignUp page for website
562     elif selected == "signup":
563         st.title('Signup')
564         if signup():
565             selected = "Home"
566
567     #Appointment list page if user is a doctor
568     elif selected == 'Appointment List':
569         appointment_list_page(conn)
570
571

```

# PLAGIARISM REPORT

Aryan\_Report

## ORIGINALITY REPORT

15%

SIMILARITY INDEX

15%

INTERNET SOURCES

4%

PUBLICATIONS

%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://www.ir.juit.ac.in:8080">www.ir.juit.ac.in:8080</a> Internet Source	5%
2	<a href="http://tnsroindia.org.in">tnsroindia.org.in</a> Internet Source	2%
3	<a href="http://export.arxiv.org">export.arxiv.org</a> Internet Source	1%
4	<a href="http://ir.juit.ac.in:8080">ir.juit.ac.in:8080</a> Internet Source	1%
5	<a href="http://www.coursehero.com">www.coursehero.com</a> Internet Source	1%
6	<a href="http://dokumen.pub">dokumen.pub</a> Internet Source	<1%
7	<a href="http://machinelearningmodels.org">machinelearningmodels.org</a> Internet Source	<1%
8	<a href="http://www.ijraset.com">www.ijraset.com</a> Internet Source	<1%
9	<a href="http://www.e3s-conferences.org">www.e3s-conferences.org</a> Internet Source	<1%

10	<a href="http://www.mdpi.com">www.mdpi.com</a> Internet Source	<1 %
11	<a href="http://www.geeksforgeeks.org">www.geeksforgeeks.org</a> Internet Source	<1 %
12	<a href="http://fastercapital.com">fastercapital.com</a> Internet Source	<1 %
13	<a href="http://www.sciencegate.app">www.sciencegate.app</a> Internet Source	<1 %
14	"Recent Findings in Intelligent Computing Techniques", Springer Science and Business Media LLC, 2018 Publication	<1 %
15	<a href="http://formative.jmir.org">formative.jmir.org</a> Internet Source	<1 %
16	<a href="http://www.cse.iitb.ac.in">www.cse.iitb.ac.in</a> Internet Source	<1 %
17	<a href="http://turcomat.org">turcomat.org</a> Internet Source	<1 %
18	<a href="http://www.jetir.org">www.jetir.org</a> Internet Source	<1 %
19	"Proceedings of International Conference on Communication and Computational Technologies", Springer Science and Business Media LLC, 2023 Publication	<1 %

20	<a href="http://uu.diva-portal.org">uu.diva-portal.org</a> Internet Source	<1 %
21	<a href="http://www.analyticsvidhya.com">www.analyticsvidhya.com</a> Internet Source	<1 %
22	<a href="http://www.ijert.org">www.ijert.org</a> Internet Source	<1 %
23	<a href="http://downloads.hindawi.com">downloads.hindawi.com</a> Internet Source	<1 %
24	<a href="http://www.listendata.com">www.listendata.com</a> Internet Source	<1 %
25	<a href="http://fdocuments.in">fdocuments.in</a> Internet Source	<1 %
26	<a href="http://www.tandfonline.com">www.tandfonline.com</a> Internet Source	<1 %
27	<a href="http://businessdocbox.com">businessdocbox.com</a> Internet Source	<1 %
28	<a href="http://koha.du.se">koha.du.se</a> Internet Source	<1 %
29	<a href="http://www.eurchembull.com">www.eurchembull.com</a> Internet Source	<1 %
30	<a href="http://www.ijrte.org">www.ijrte.org</a> Internet Source	<1 %
31	<a href="http://yiluheihei.r-universe.dev">yiluheihei.r-universe.dev</a> Internet Source	<1 %



32 123dok.org <1 %  
Internet Source

---

33 Eduardo Islas-Cota, J. Octavio Gutierrez-Garcia, Christian O. Acosta, Luis-Felipe Rodríguez. "A systematic review of intelligent assistants", Future Generation Computer Systems, 2022 <1 %  
Publication

---

34 medium.com <1 %  
Internet Source

---

35 www.hindawi.com <1 %  
Internet Source

---

36 www.researchgate.net <1 %  
Internet Source

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT**  
**PLAGIARISM VERIFICATION REPORT**

Date: .....

Type of Document (Tick):  PhD Thesis  M.Tech Dissertation/ Report  B.Tech Project Report  Paper

Name: \_\_\_\_\_ Department: \_\_\_\_\_ Enrolment No \_\_\_\_\_

Contact No. \_\_\_\_\_ E-mail. \_\_\_\_\_

Name of the Supervisor: \_\_\_\_\_

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): \_\_\_\_\_

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at .....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/Images/Quotes</li> <li>• 14 Words String</li> </ul>		Word Counts	
<b>Report Generated on</b>			Character Counts	
		<b>Submission ID</b>	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck.juit@gmail.com](mailto:plagcheck.juit@gmail.com)