



**Faculty of Computers  
& Artificial Intelligence**



**Benha University**

# **Medical Diagnosis System (Covid-19)**

A senior project submitted in partial fulfillment of the requirements for the degree of Bachelor of Computers and Artificial Intelligence.

**Information Systems Departement**

## ***Project Team***

1. Nada Fathy Mohamed Mubarak
2. Hader Magdy Mohamed Ebrahim
3. Ahmed Ahmed Hosny Mohamed Nafea
4. Abdelrhman Rizq Mohamed Afify
5. Omar Mahmoud Mohamed
6. Youssef Gamal Elsayed Mohamed

***Under Supervision of***  
**Dr. Noha Ezzat El-Attar**

Benha, **July 2021**

## **DECLARATION**

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of Computers and Artificial Intelligence in information systems is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of our knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

**Signed:** \_\_\_\_\_

**Signed:** \_\_\_\_\_

**Signed:** \_\_\_\_\_

**Signed:** \_\_\_\_\_

**Signed:** \_\_\_\_\_

**Signed:** \_\_\_\_\_

**Date:** July,2021.

## **ACKNOWLEDGMENTS**

First, and foremost, praise and thanks be to Allah almighty, The giver of generous blessing and gifts for his never-ending grace and mercy. Second, we would like to thank the Dean of the College, the Vice Dean of the College and the Head of the Information Systems Department. Third we want to thank our project supervisor Dr. Noha Ezzat El-Attar for her support, inspiration, advice, guidance, supervision and patience. She was a very powerful supporter for our project. Also, we would like to thank Eng. Mahmoud Sobhy for his proficient advice, appreciated comments and encouragement. And last but not least, our families for their understanding, encouragement and support.

## **ABSTRACT**

The automation of the medical process becomes urgent. Medical diagnosis is an important task that should be performed as accurately and efficiently as is possible. In this project we will be able to diagnose the disease (covid-19) we will be able to diagnose the disease(covid-19) by using machine learning algorithms. The user answers a specific question then the system returns the result. Also, the project help people to find a suitable communication channel with specialized doctor to select the proper treatment protocol.

## TABLE OF CONTENTS

<b>LIST OF FIGURES .....</b>	<b>III</b>
<b>LIST OF TABLES .....</b>	<b>V</b>
<b>1 INTRODUCTION.....</b>	<b>6</b>
<b>1.1 Motivation .....</b>	<b>6</b>
<b>1.2 Project outcome: .....</b>	<b>7</b>
<b>1.3 Problem STATEMENT:.....</b>	<b>7</b>
<b>1.4 Project Objectives:.....</b>	<b>8</b>
<b>1.5 Stakeholder List:.....</b>	<b>8</b>
<b>1.6 Proposed scope and Process model:.....</b>	<b>9</b>
1.6.1 In the case of patient:.....	9
1.6.2 In the case of doctor: .....	9
<b>2 PROJECT PLANNING AND REQUIREMENTS.....</b>	<b>10</b>
<b>2.1 Scope Initiation .....</b>	<b>10</b>
2.1.1 resource planning scheduling .....	11
<b>2.2 Development Requirements.....</b>	<b>11</b>
2.2.1 Cost Estimating & Budgeting.....	12
2.2.2 Project Risk List .....	12
2.2.3 Determining System Requirement .....	13
<b>3 PROJECT ANALYSIS AND DESIGN .....</b>	<b>15</b>
<b>3.1 Use Case Diagram .....</b>	<b>15</b>
<b>3.2 Sequence Diagram .....</b>	<b>22</b>
<b>3.3 Activity Diagram.....</b>	<b>22</b>
<b>3.4 Deployment Diagram.....</b>	<b>23</b>
<b>3.5 entity-relationship diagram (ERD) .....</b>	<b>24</b>
<b>3.6 ERD MAPPING .....</b>	<b>25</b>
<b>4 IMPLEMENTATION AND TESTING .....</b>	<b>26</b>
<b>4.1 Patient Path .....</b>	<b>26</b>
4.1.1 Home Page .....	26
4.1.2 Check.....	27
4.1.3 Check Result .....	29
4.1.4 Register Page.....	30
4.1.5 confirm sign up.....	31
4.1.6 Select Available Doctor to Start Chat.....	32
4.1.7 Start Chat with Doctor.....	33
4.1.8 Chat Details .....	33
4.1.9 Login Page.....	34
4.1.10 Reset Password.....	35
4.1.11 code verification for reset password .....	35
4.1.12 confirm password .....	35

<b>4.2</b>	<b>Admin Path .....</b>	<b>37</b>
4.2.1	Login Page.....	37
4.2.2	Admin Page.....	38
4.2.3	Doctor's Information Management .....	38
4.2.4	Add New Doctor .....	39
4.2.5	View patient Information .....	40
<b>4.3</b>	<b>Doctor Path .....</b>	<b>40</b>
4.3.1	Login Page.....	40
4.3.2	Doctor Page.....	41
4.3.3	View and management patient Information .....	42
4.3.4	Start Chat with Patients .....	43
4.3.5	Chat Details .....	43
<b>4.4</b>	<b>implementation code .....</b>	<b>44</b>
4.4.1	Patient Path.....	44
4.4.2	Admin Path.....	48
4.4.3	Doctor Path.....	50
4.4.4	API (application program interface) .....	52
<b>4.5</b>	<b>Database .....</b>	<b>54</b>
<b>4.6</b>	<b>Machine learning .....</b>	<b>55</b>
4.6.1	Methods.....	56
<b>4.7</b>	<b>Datasets.....</b>	<b>58</b>
4.7.1	First dataset .....	58
4.7.2	Second dataset .....	59
<b>5</b>	<b>CONCLUSIONS AND FUTURE IMPROVEMENTS .....</b>	<b>60</b>
<b>5.1</b>	<b>References .....</b>	<b>61</b>
<b>5.2</b>	<b>References to Electronic Sources .....</b>	<b>61</b>
	الملخص.....	62

## LIST OF FIGURES

Figure 2.1 Gantt chart .....	11
Figure 3.1 Use Case diagram .....	21
Figure 3.2 Sequence diagram.....	22
Figure 3.3 State diagram .....	23
Figure 3.4 Deployment diagram .....	24
Figure 3.5 ERD .....	25
Figure 3.6 ERD Mapping.....	25
Figure 4.1 Home Page.....	26
Figure 4.2 First step Check .....	27
Figure 4.3 Second Step Check .....	27
Figure 4.4 Choose check way .....	28
Figure 4.5 Enter analyze ratios .....	28
Figure 4.6 Negative Result .....	29
Figure 4.7 Positive Result .....	30
Figure 4.8 Register Page .....	30
Figure 4.9 Confirm Sign up .....	31
Figure 4.10 sending verification code.....	32
Figure 4.11 Select Doctor .....	32
Figure 4.12 Start Chat .....	33
Figure 4.13 Chat Details .....	33
Figure 4.14 Login Page.....	34
Figure 4.15 Reset Password.....	35
Figure 4.16 code verification for reset password.....	35
Figure 4.17 Send code to mail .....	36
Figure 4.18 Confirm Password .....	36
Figure 4.19 login page .....	37
Figure 4.20 Admin page .....	38
Figure 4.21 Doctor's information management.....	38
Figure 4.22 Add new doctor .....	39
Figure 4.23 view patient information.....	40
Figure 4.24 login page .....	40
Figure 4.25 doctor page .....	41
Figure 4.26 view and management patient information.....	42

Figure 4.27 start chat with patients .....	43
Figure 4.28 chat details .....	43
Figure 4.29 confirm signup.....	44
Figure 4.30 dashboard.....	45
Figure 4.31 Chat.....	46
Figure 4.32 Chat details .....	47
Figure 4.33 admin login front .....	48
Figure 4.34 admin login back .....	48
Figure 4.35 dashboard.....	49
Figure 4.36 doctor login front .....	50
Figure 4.37 doctor login back .....	50
Figure 4.38 doctor Dashboard.....	51
Figure 4.39 chat .....	52
Figure 4.40 API code .....	53
Figure 4.41 Database tables .....	54
Figure 4.42 patient table .....	54
Figure 4.43 doctor table .....	55
Figure 4.44 admin table .....	55
Figure 4.44 message table.....	55
Figure 4.45 Logistic regression algorithm .....	57
Figure 4.46 Naïve Bayes algorithm .....	58
Figure 4.47 Symptoms dataset.....	59
Figure 4.48 Analysis dataset .....	59



## **LIST OF TABLES**

TABLE 1.1. STAKEHOLDER LIST. ....	9
TABLE 3.1 Use Cases Id.....	17

## *Chapter One*

### **1 INTRODUCTION**

Human disease diagnosis is a complicated process and requires high level of expertise. Any attempt of developing a web-based expert system dealing with human disease diagnosis has to overcome various difficulties. Medical diagnosis is an important task that should be performed as accurately and efficiently as is possible. All doctors are unfortunately not equally skilled in every subspeciality and they are in many places a scarce resource. A system for automated medical diagnosis would enhance medical care and reduce costs. Rule-based expert system are well known examples of such automated diagnosis systems. They can however be very complicated to design since a physician often cannot express his reasoning in simple if-then rules. It would be desirable to learn models instead of manually constructing them. The learning could be done from a database of previous cases. By doing this we can overcome the knowledge acquisition problem. In our medical diagnosis system, we will be able to diagnose the disease by asking a specific question based on a very deep data from many doctors' experience and deep research in medical science then the system will Diagnose many diseases based on the answer of this specific questions.

#### **1.1 MOTIVATION**

What motivated us to do this idea are many reasons we live on the ground. Until the moment in our country, the cost of doing a medical smear to know your infection with the virus is very high, and on the other hand, isolation places in densely populated areas are very few, which exposes the people who have been affected by the virus more seriously. Prone to not finding a place in sanitary isolation hospitals for this, and by making the idea of electronic medical examination and providing treatment and advice, we can reduce overcrowding and make people who feel some non-serious symptoms and want reassurance not to leave their homes to avoid infection of others and to isolate themselves at home with follow-up Live through doctors through the website that we have implemented.

## **1.2 PROJECT OUTCOME:**

### **•Patient can:**

1. Know Information about The Covid-19 virus.
2. Do the test of infection with Covid-19 by enter the apparent symptoms.
3. Do the test of infection with Covid-19 by enter the medical test ratios.
4. Chat with a Doctor for Medical Advice and Medication.

### **•Doctor can:**

1. Easy Access to the chat rooms.
2. Follow-up the cases that the examination confirmed that they are infected.
3. Edit their Profiles and their schedules.

### **•Admin Can:**

1. Add doctor to the system.
2. Delete doctor from the system.
3. Show patient information.

## **1.3 PROBLEM STATEMENT:**

1. Covid-19, which affected on most countries of the world
2. No site introduces medication advice by communicate with real doctor.
3. The high cost of PCR.
4. Hospitals overcrowding.

## 1.4 PROJECT OBJECTIVES:

1. Easy to check and diagnose Covid-19.
2. Save time and effort for patients.
3. Effective communication between patients and doctors.
4. Reduce crowds in hospitals.

## 1.5 STAKEHOLDER LIST:

Stakeholder	Interest	Importance
<b>Patient</b>	<ul style="list-style-type: none"><li>- Easy to check disease and take some instruction to recover from it.</li><li>- Communicate with the specialist doctor when any symptoms appear.</li></ul>	Through our website, we were able to make communication between patient and doctor, and thus it became easy for him to diagnose Covid-19.
<b>Doctor</b>	It is difficult to communicate between doctor and patient at this time because hospitals are full of patients, so we want to reduce this crowd.	Through our website, the doctor able to follow patient status from login date to recovery date and introduce instructions for him.

TABLE 1.1. STAKEHOLDER LIST

## **1.6 PROPOSED SCOPE AND PROCESS MODEL:**

### **1.6.1 In the case of patient:**

As a patient, our website contains advantages like:

- Best-specialized doctors in this field.
- Control your status all time.
- Get notification from the doctor.
- Instructions to get recovery faster.
- Contact the doctor by messages.

### **1.6.2 In the case of doctor:**

As a doctor, our website contains advantages like:

- Can see patient symptoms.
- Add comments on patient profile.
- Monitor the patient's status and communicate with him at any time.
- Show previous comments on patient profile.
- Giving a patient a route of treatment, and stay connected with him.

## Chapter Two

# 2 PROJECT PLANNING AND REQUIREMENTS

## 2.1 SCOPE INITIATION

The project will design, develop and deliver a new website to check if he/she is infected with covid-19 through answering some Questions and also the doctor can use the same system to contact a patient and help him to get the suitable treatment protocol.

- The project aims to help covid 19 patient.
  - Patient can:  
  
login, sign up, enter personal information,  
  
select symptoms, find result, Contact with doctor.
  - Doctor can:  
  
login, show patient information,  
  
contact with patient, Send treatment recommendations to patient.
  - Admin can:  
  
login, sign up, Add/Delete doctor account,  
  
Update doctor's data, Show doctors information.

### 2.1.1 resource planning scheduling

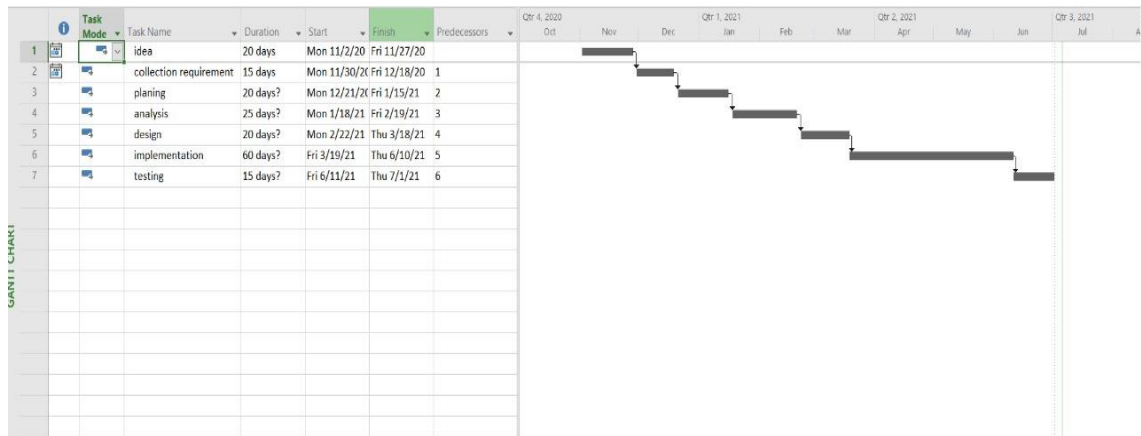


Figure 2.1 Gantt chart

## 2.2 DEVELOPMENT REQUIREMENTS

The system should satisfy the following requirements:

### 1. Accessibility

System should be easy to access. This achieved by, the system will be web-based and using by Android, it can be accessed from any device that has internet connection.

### 2. Availability

System should be in operable state at the start of a mission. Our system is always available and if any action occurs you get it.

### 3. Accuracy

System should provide accurate results, this is represented in use of our application to get the information about anything in the faculty.

### 4. Security

System should be able to protect the data and resources, this achieved by no one able to login to the system without having academic email.

## 5. Usability

System should be easy to use, Achieved by Use friendly

- UX/UI
  - User Experience Design
  - User Journey Map
  - Prototype
  - User Interface
- Front-End
  - Html, Html5
  - CSS
  - Java script
  - Bootstrap
- Back-End
  - Php
  - Php my admin (SQL)
- Machine learning
  - Python

### 2.2.1 Cost Estimating & Budgeting

- Laptops: 50,000 EGP.
- Software license: 2.000 EGP.
- Hosting website: 2.000 EGP.

### 2.2.2 Project Risk List

- Resource risk
- Skills risk
- Security risk



- Deadline risk
- Authentication risk
- Authorization risk
- Budget risk
- Failure in server
- Developers have health problems.

### 2.2.3 Determining System Requirement

#### **Functional requirement:**

1. Patient basic functions:
  - Login and sign up.
  - Enter personal information.
  - Select symptoms.
  - Find result.
  - Contact doctor.
  - Exit.
2. Doctor basic functions:
  - Login.
  - Shaw patient information.
  - Contact with patient.
  - Send treatment recommendations to patient.
3. Admin basic functions:
  - Login and sign up.
  - Add/Delete doctor account.
  - Update doctors' data.
  - Show doctors' information.

**non-Functional requirement:**

1- Constraint

- Design.
- implementation.

2- External interface

- Software.
- user.
- communication.

3- Performance

- Response time.
- Throughput.
- Latency.
- Degraded modes.

4- Quality attribute

- Usability.
- Robustness.
- Instability.
- Integrity.
- Availability.
- Manufacturability.
- Serviceability.
- Size.

## *Chapter Three*

### **3 PROJECT ANALYSIS AND DESIGN**

This chapter presents the analysis phase of the proposed system. In this phase, the System requirements are discussed in order to fulfil the features and objectives that was previously mentioned.

Using Object Oriented Approach:

1. Use Case Diagram.
2. Sequence Diagram
3. State Diagram
4. Deployment Diagram
5. Database Design (ERD)
6. ERD Mapping

#### **3.1 USE CASE DIAGRAM**

The use case is described interactions between systems and users in a particular environment.

Use Case Id	Primary Actor	Use Cases
UC-1	Patient	Sign Up
UC-2	Patient/ Doctor / Admin	Login
UC-3	Admin	Add Doctor
UC-4	Admin	Update Doctor Data
UC-5	Admin	Delete Doctor
UC-6	Doctor / Admin	Show Doctor Information
UC-7	Doctor	Enter Work's Time
UC-8	Patient	Enter Symptoms
UC-9	Patient	Show Result
UC-10	Patient / Doctor	Chatting
UC-11	Admin / Doctor	Show Patient Information

Table 3.1 Use Cases Id

<b>Use Case ID:</b>	<b>UC-1</b>
Use Case Name:	Sign up
Actors:	Patient
Preconditions:	Home page
Post-conditions:	Patient Sign up to website by Enter First Name, Last Name, Email and Create Password.
Normal Flow:	If all information the patient enter is valid then patient can login and do the Covid-19 test.

<b>Use Case ID:</b>	<b>UC-2</b>
Use Case Name:	Login
Actors:	Patient / Doctor /Admin
Preconditions:	Sign Up
Post-conditions:	1. Patient login to website with email and password 2. Instructor login to website with email and password 3. Admin login to website with ID and password
Normal Flow:	1. If email and password valid can be logged in. 2. If user forget password. he can change password by sending confirm code for his Email, then change it and login again.

<b>Use Case ID:</b>	<b>UC-3</b>
Use Case Name:	Add Doctor
Actors:	Admin
Preconditions:	Admin Login
Post-conditions:	4. Admin Add New Doctor to the system to deal with the Patients by adding the information of the new doctor
Normal Flow:	3. If the information of the doctor is not valid admin can fill it again in the right way.

<b>Use Case ID:</b>	<b>UC-4</b>
Use Case Name:	Update Doctor Data
Actors:	Admin
Preconditions:	Admin Login
Post-conditions:	5. Admin Update the Data of existing Doctor in the system to deal with the Patients
Normal Flow:	4.The Updated Data Must Be in the right form

<b>Use Case ID:</b>	<b>UC-5</b>
Use Case Name:	Delete Doctor
Actors:	Admin
Preconditions:	Admin Login
Post-conditions:	6. Admin Delete a doctor from the system and the deleted doctor cannot deal with the system anymore.
Normal Flow:	5. Admin Must Be Logging in the system 6. Admin Click the Delete Button on the Doctor Record to Delete

<b>Use Case ID:</b>	<b>UC-6</b>
Use Case Name:	Show Doctor Information
Actors:	Doctor /Admin
Preconditions:	Login
Post-conditions:	7. Doctor Can Show His Info and Edit it 8. Admin Can Show Doctor Info.
Normal Flow:	7. If any issue in the information Doctor Can Update his record.

<b>Use Case ID:</b>	<b>UC-7</b>
Use Case Name:	Enter Work's Time
Actors:	Doctor
Preconditions:	Doctor Login
Post-conditions:	9. Doctor enter his work shifts to handle the times he will be available.
Normal Flow:	8. If the times he enters have objection with another doctor he can edit it.

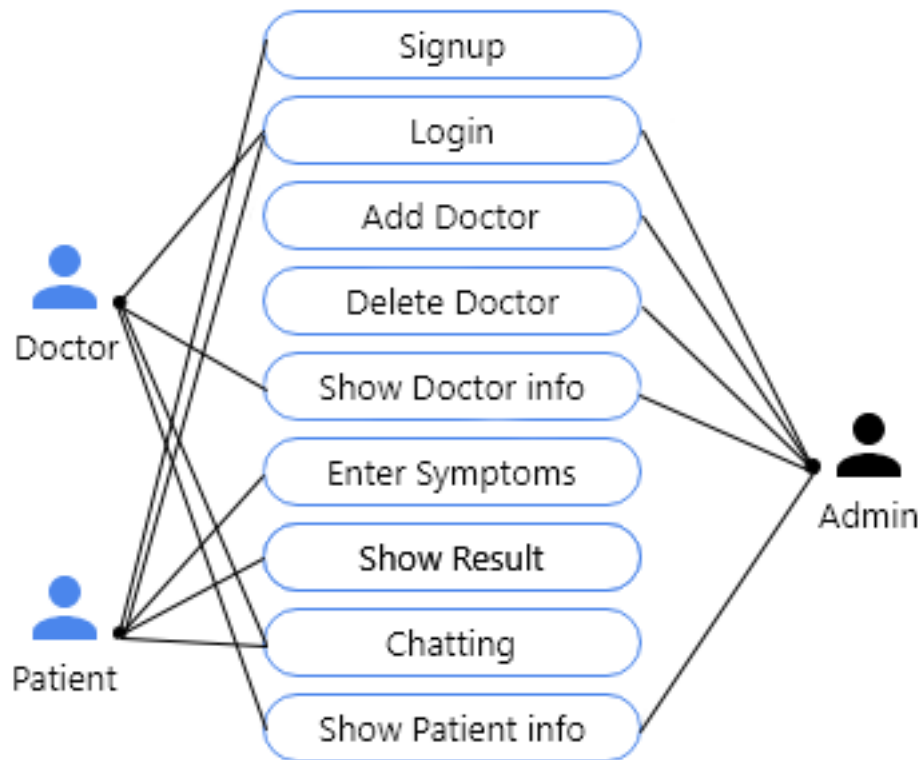
<b>Use Case ID:</b>	<b>UC-8</b>
Use Case Name:	Enter Symptoms
Actors:	Patient
Preconditions:	Sign Up or Login
Post-conditions:	10. Patient in the Steps of the Test The input He enters is the symptoms, the patient choose it depends on the pain he feels it.
Normal Flow:	9. If the patient has a mistake when he is choosing from the symptoms he can return and start the test from the beginning.

<b>Use Case ID:</b>	<b>UC-9</b>
Use Case Name:	Show Result
Actors:	Patient
Preconditions:	Sign Up or Login
Post-conditions:	11. The Patient can show his Test Result and choose to chat with a doctor or leaving the web site
Normal Flow:	10. If the Result is positive patient choose to chat with doctor or to leave. 11. If the result is negative patient leave the web site.

<b>Use Case ID:</b>	<b>UC-10</b>
Use Case Name:	Chatting
Actors:	Doctor / Patient
Preconditions:	After Showing the Test Result for Patients / Doctor Login
Post-conditions:	12. When Patient Finish the Test and Choose to start Chat with a doctor, He will enter a chat room with an online doctor and the conversation will begin.  13. When Doctor Login to the system he is appearing online, he will be ready to receive the messages in chats
Normal Flow:	12. If the patient leaves the chat, he can return to it again. 13. If the doctor is offline, he can't receive any messages.

<b>Use Case ID:</b>	<b>UC-11</b>
Use Case Name:	Show Patient Information
Actors:	Admin / Doctor
Preconditions:	Login
Post-conditions:	14. Admin Can Show the patient info and delete the non-useful records.  15. Doctor Can Show the information of the patient he is chatting with them to monitor their health
Normal Flow:	-----





**Figure 3.1 Use Case diagram**

## 3.2 SEQUENCE DIAGRAM

It's the sequence of messages exchanged between user, user interface, web server and database.

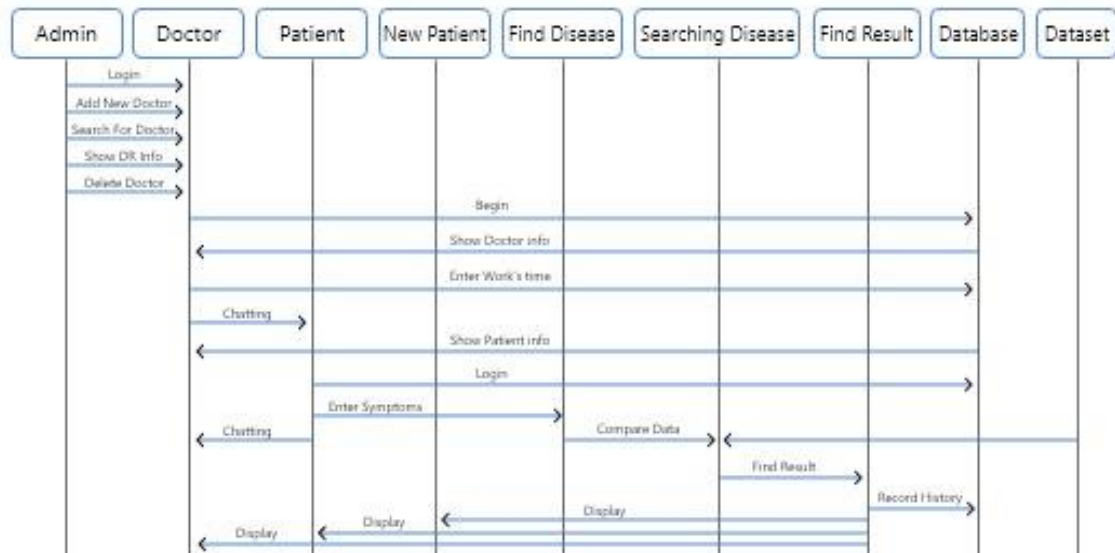


Figure 3.2 Sequence diagram

## 3.3 ACTIVITY DIAGRAM

It's used to give an abstract description of the behaviour of a system. This behaviour is analysed and represented as a series of events that can occur in one or more possible states.

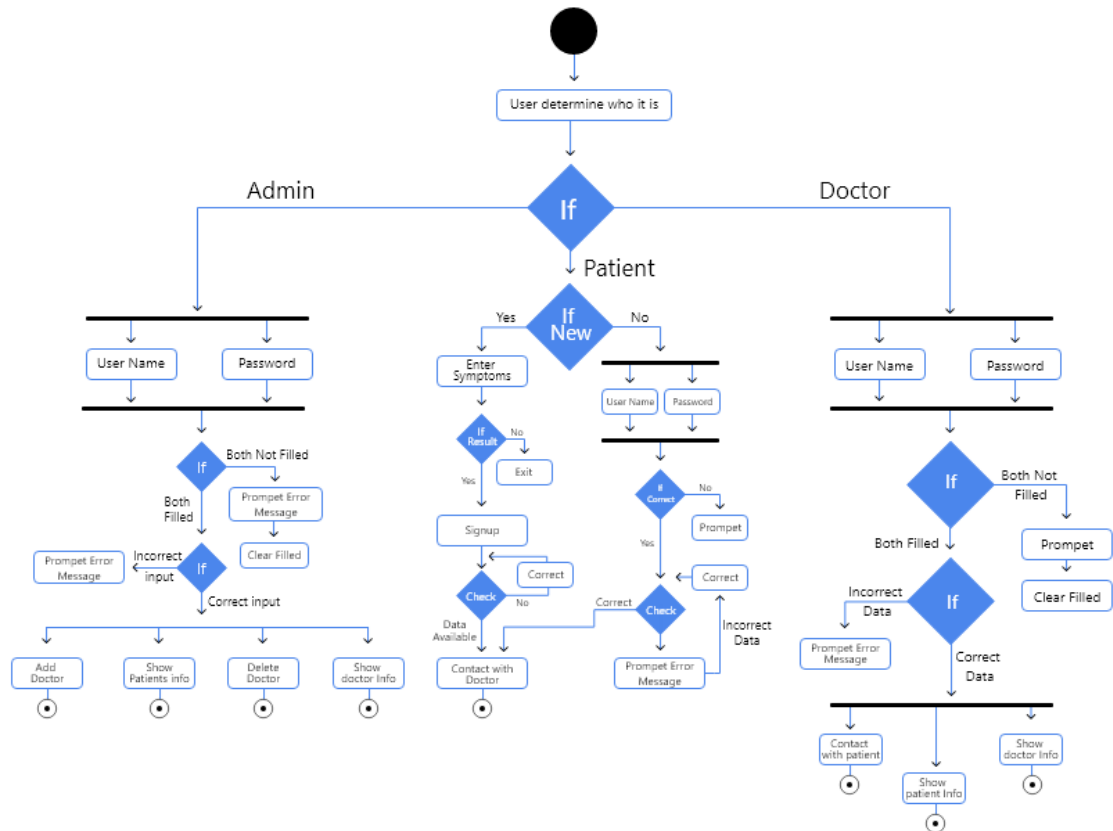


Figure 3.3 Activity diagram

### 3.4 DEPLOYMENT DIAGRAM

It is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets.

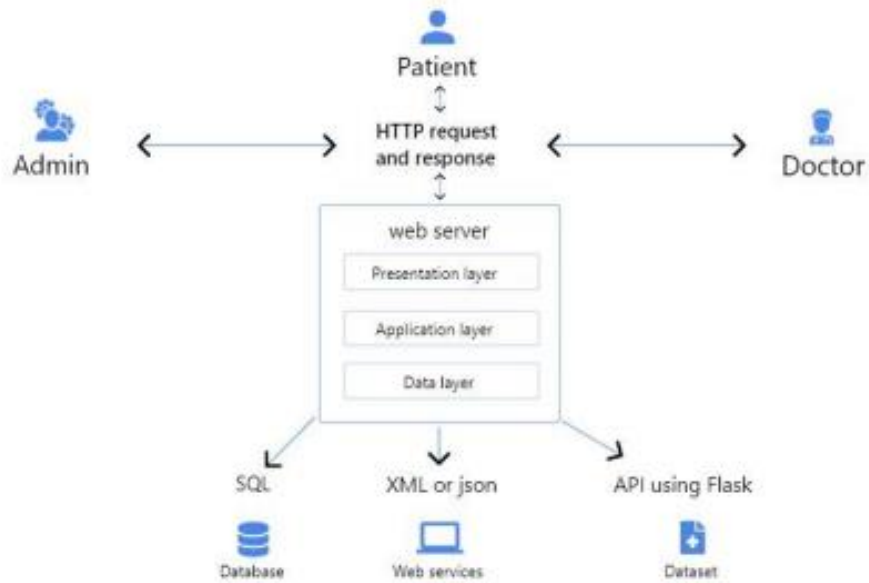


Figure 3.4 Deployment diagram

### 3.5 ENTITY-RELATIONSHIP DIAGRAM (ERD)

An entity-relationship diagram (ERD) is a data modeling technique that graphically illustrates an information system's entities and the relationships between those entities.

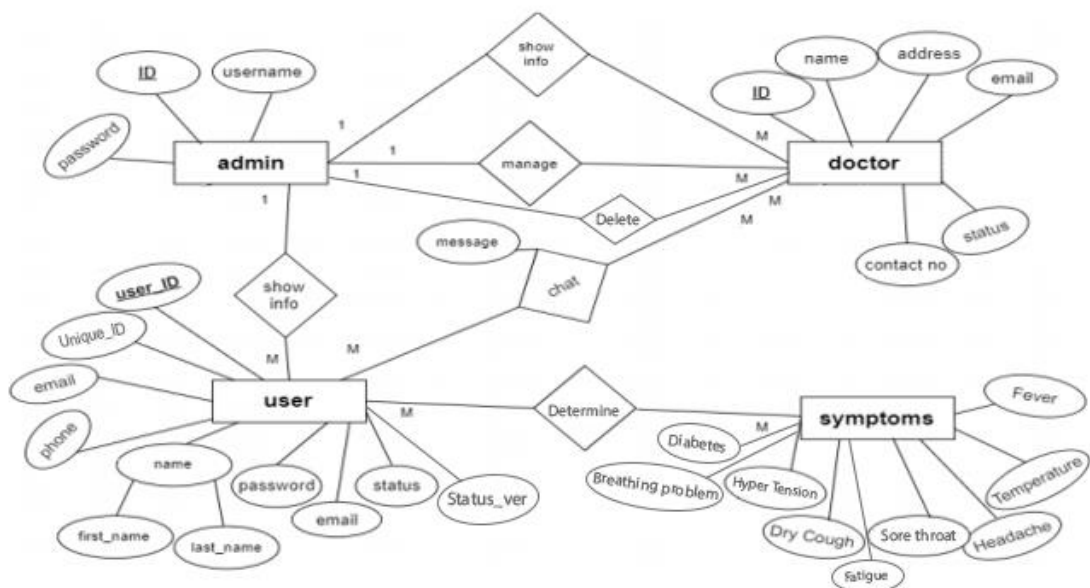


Figure 3.5 ERD

### 3.6 ERD MAPPING

Entity Relationship Diagram (ERD MAPPING) An entity relationship model is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases.

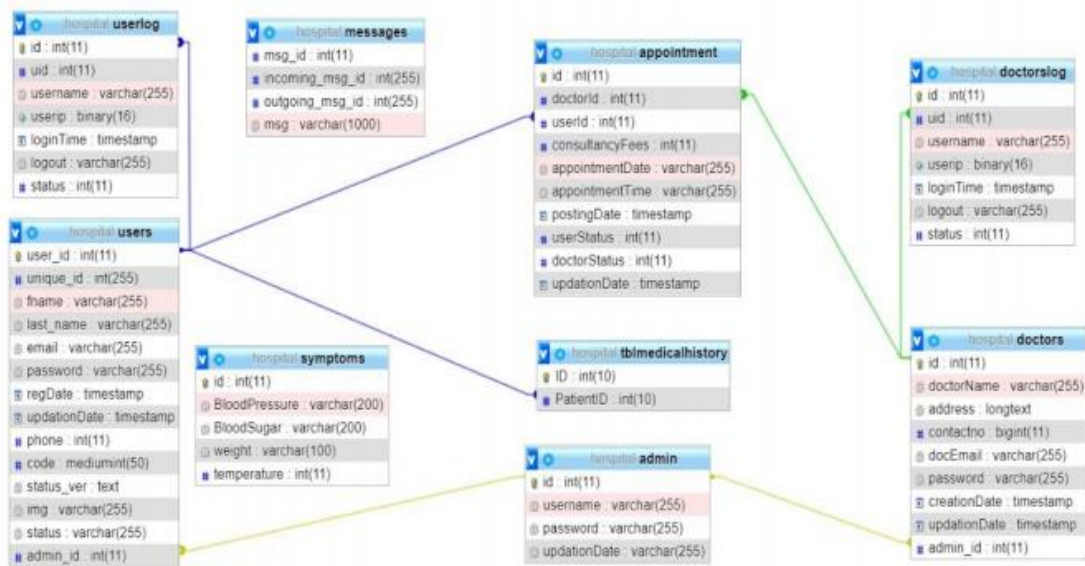


Figure 3.6 ERD Mapping

## Chapter Four

### 4 IMPLEMENTATION AND TESTING

#### 4.1 PATIENT PATH

##### 4.1.1 Home Page

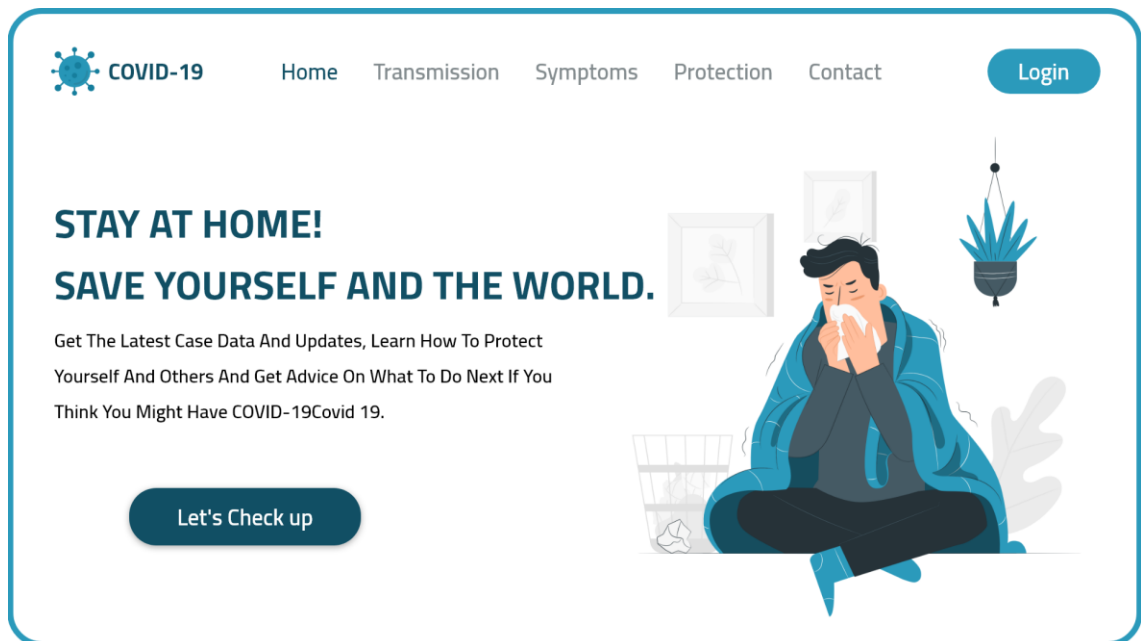







Figure 4.1 Home Page

Home page is the first page that appears to the patient, where he can enter to check his symptoms through this page when he clicks on button let's check up and get information about covid-19 and how to avoid infection.

### 4.1.2 Check



**Personal Information**  
Specify your gender and age

**Gendre \***

☐ Male

☐ Female

**Age \***

Years

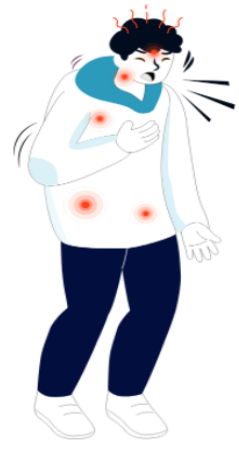
[Next](#)

1 of 6 answered

**Figure 4.2 First step Check**

In this page, we will know how to start the test steps:

- This is the first step in check the patient can determine her/his information.
- After this step the patient can enter to the second step.



**Select all the symptoms you currently have in this list.**

You can also select more than one symptom or tick the «No symptom» field.  
At least one field must be selected.

☐ No symptom (from the below)

☐ Breathing Problem

☐ Fever

☐ Dry Cough

☐ Sore throat

☐ Running Nose

☐ Asthma

[back](#) [Next](#)

**Figure 4.3 Second Step Check**

- In this step, the patient can determine the symptoms he is feeling.
- In the next step, there are other symptoms, the patient can choose the symptoms that he feels through and so on until he reaches the last step in determining the symptoms that appear on him.

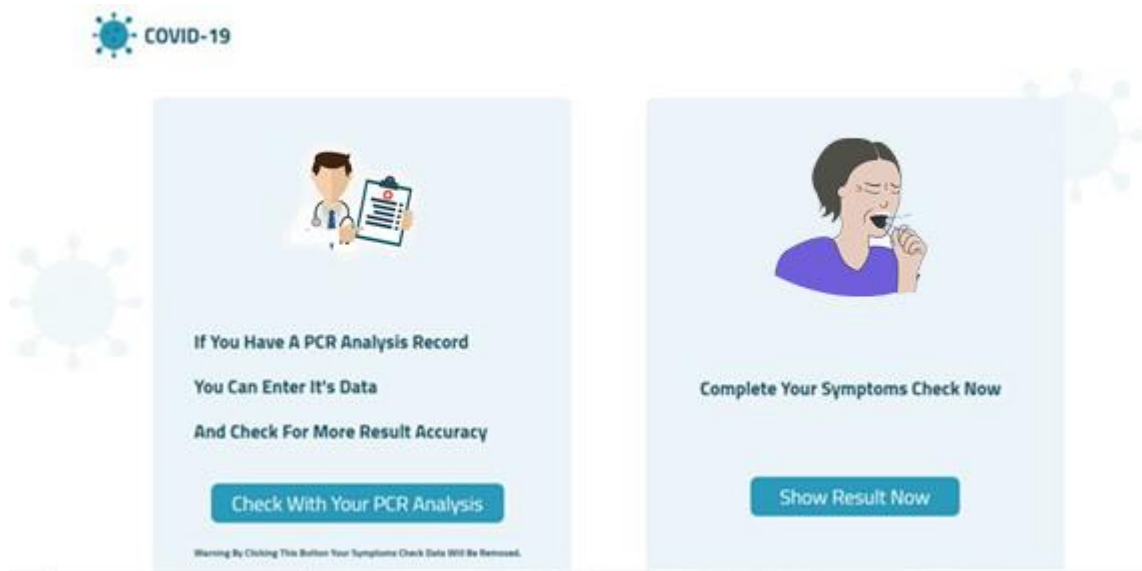


Figure 4.4 Choose check way

If patient have PCR analysis record, he can check using it or he can complete his symptoms check.

The screenshot shows a COVID-19 form titled 'Fill in the required proportions'. It contains a grid of input fields for various lab tests. The fields are arranged in four columns. The first column contains: CA, CK, CREA, ALP, GGT, GLU, AST, and ALT. The second column contains: LDH, PCR, KAL, NAT, UREA, WBC, RBC, and HGB. The third column contains: HCT, MCV, MCH, MCHC, PLT1, NE, LY, and MO. The fourth column contains: EO, BA, NET, LYT, MOT, EOT, BAT, and Suspect. A blue 'Next' button is located at the bottom center of the form.

Figure 4.5 Enter analyze ratios

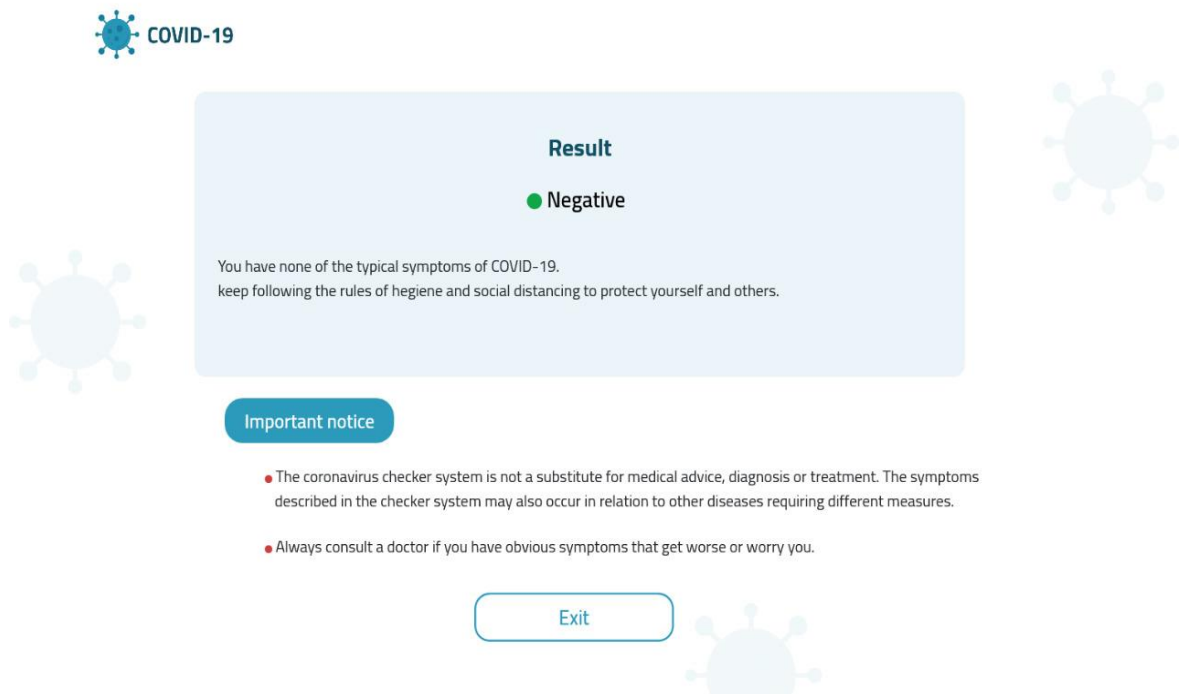


User can Enter PCR analysis data in this page and by predict, the check result will be appeared.

### 4.1.3 Check Result

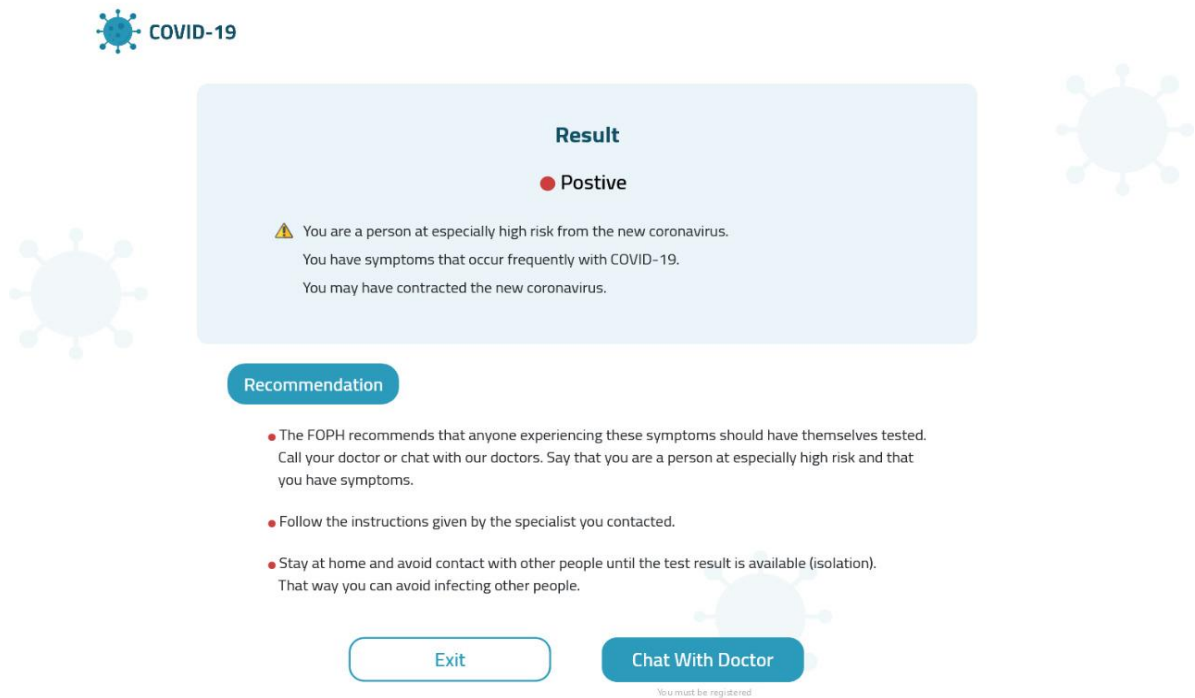
In this page, based on the symptoms or PCR analysis that have been identified, the result will appear to the patient:

- After determine all the symptoms that the patient feels or PCR analysis, the result will be appeared, whether they are negative or positive.
- If the result is negative, in which case the user can return to home page.



**Figure 4.6 Negative Result**

If the result is positive, in this case the user can follow up with the doctor through the website.



The screenshot displays a web interface for COVID-19 test results. At the top left, there is a blue virus icon and the text "COVID-19". The main content area has a light blue background. A central box titled "Result" shows a red dot and the word "Positive". Below this, a warning icon is followed by text stating the user is at high risk and may have contracted the virus. A "Recommendation" section lists three bullet points: consulting a doctor, following specialist instructions, and staying home. At the bottom, there are "Exit" and "Chat With Doctor" buttons. A small note below the chat button says "You must be registered".

**COVID-19**

**Result**

● Positive

⚠ You are a person at especially high risk from the new coronavirus.  
You have symptoms that occur frequently with COVID-19.  
You may have contracted the new coronavirus.

**Recommendation**

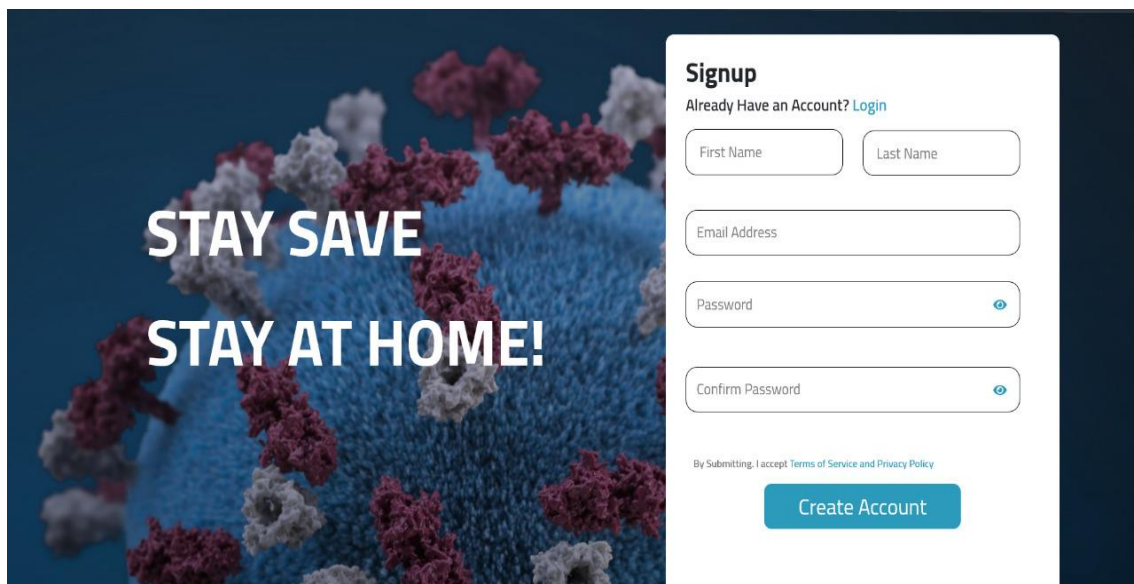
- The FOPH recommends that anyone experiencing these symptoms should have themselves tested. Call your doctor or chat with our doctors. Say that you are a person at especially high risk and that you have symptoms.
- Follow the instructions given by the specialist you contacted.
- Stay at home and avoid contact with other people until the test result is available (isolation). That way you can avoid infecting other people.

[Exit](#) [Chat With Doctor](#)

You must be registered

Figure 4.7 Positive Result

#### 4.1.4 Register Page



The screenshot shows a registration page. The background is dark blue with a 3D model of a virus. Large white text on the left says "STAY SAFE" and "STAY AT HOME!". On the right, a white "Signup" form is overlaid. It includes a link for existing users, input fields for first and last names, email, password, and confirm password, a terms and conditions link, and a "Create Account" button.

**STAY SAFE**  
**STAY AT HOME!**

**Signup**

Already Have an Account? [Login](#)

First Name  Last Name

Email Address

Password

Confirm Password

By Submitting, I accept [Terms of Service](#) and [Privacy Policy](#)

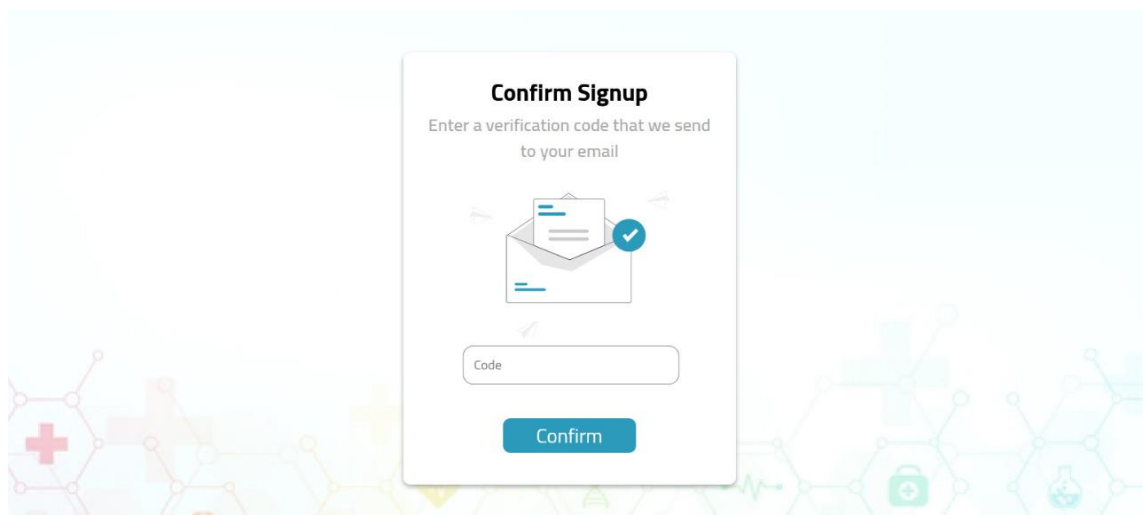
[Create Account](#)

Figure 4.8 Register Page

In this page we will know how register activity work:

- Before a patients can start talking to a doctor, they must first register.
- If the patient does not have an account, they must fill in all entries to create an account.
- If the patient leaves the fields blank, this message "All required fields" will appear.
- If the patient fills in all the required fields, it will be done after this step to ensure that the data form is correct.
- If the patient enters an invalid name, this message "Only alphabetic and white spaces for first name allowed" will appear.
- If the patient enters an email in an invalid format, the message "Invalid Email Format" will appear.
- If the password and confirm password do not match, this message "Confirm password does not match!" Will appear.
- If the patient enters all the required fields correctly, an account will be created and move to the next step.

#### 4.1.5 confirm sign up



**Figure 4.9 Confirm Sign up**

In this page Patient will enter confirm signup:

The website will send a code to the user's mail to validate the email address and improve the odds that it belongs to a real person.



Figure 4.10 sending verification code

#### 4.1.6 Select Available Doctor to Start Chat

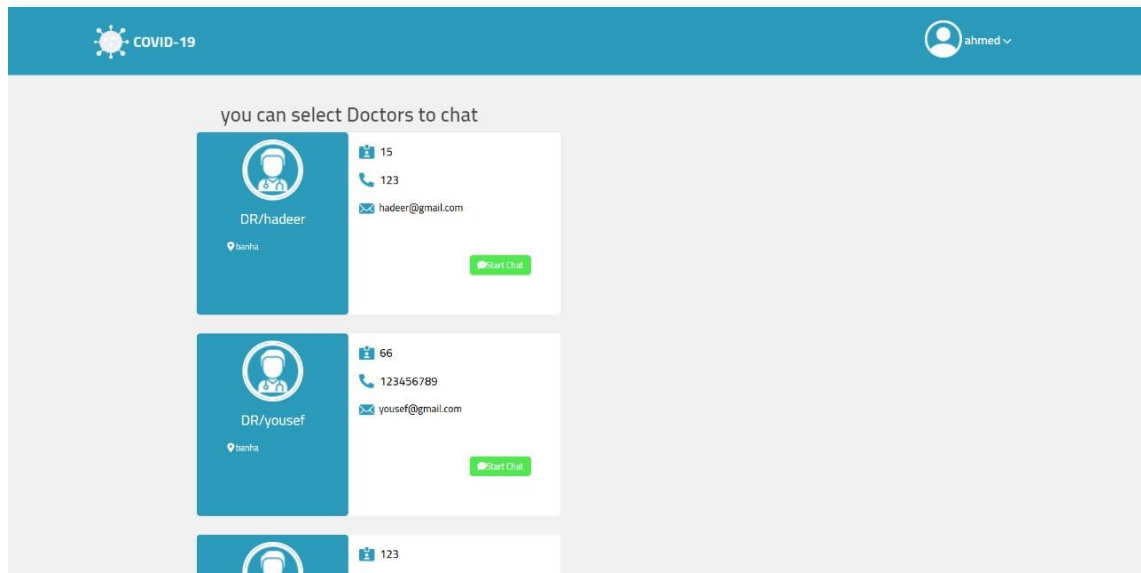


Figure 4.11 Select Doctor

### 4.1.7 Start Chat with Doctor

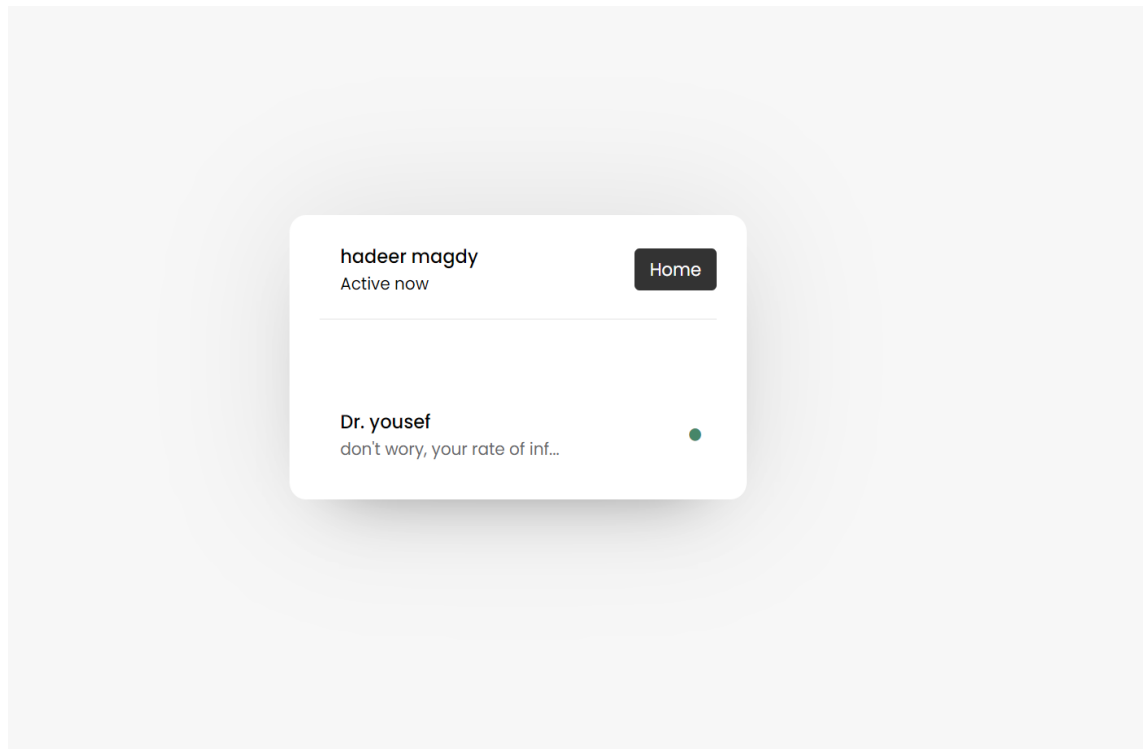


Figure 4.12 Start Chat

- If the patient clicks on the logout button, he will return to the login page.

### 4.1.8 Chat Details

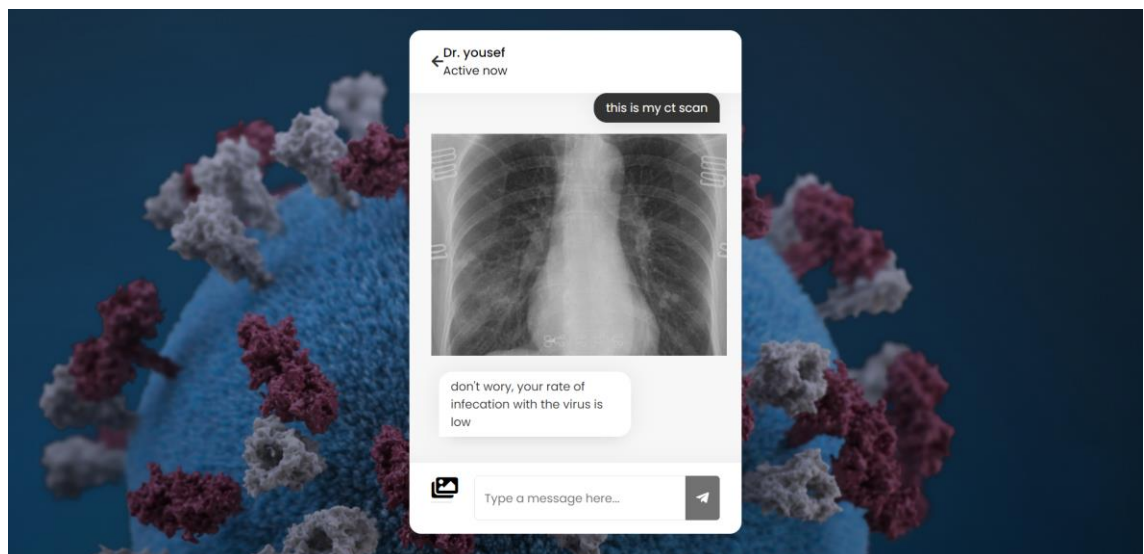
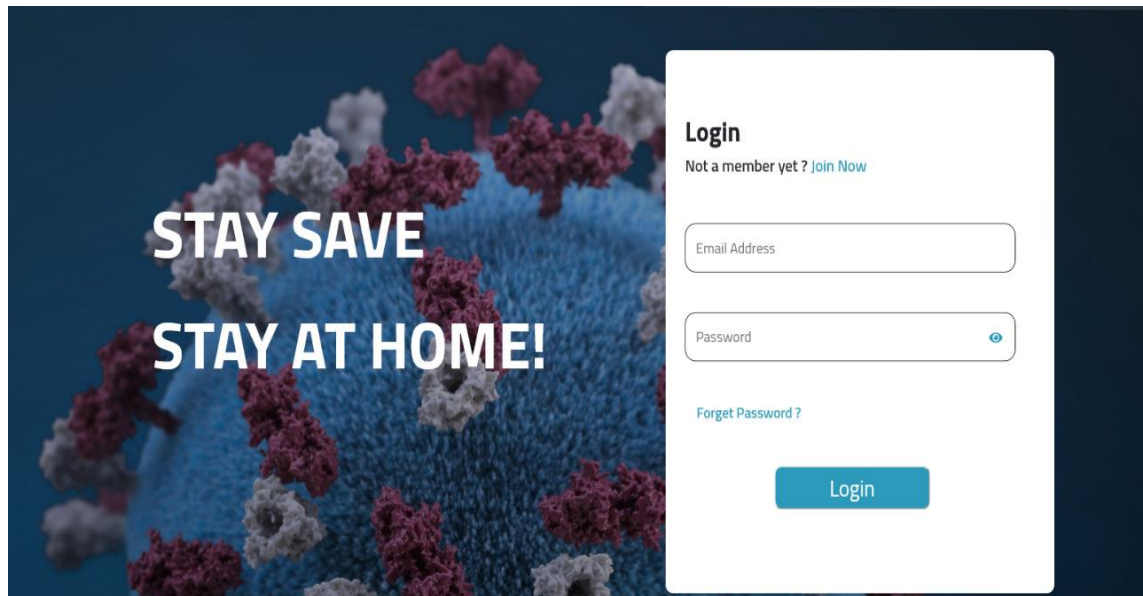


Figure 4.13 Chat Details

### 4.1.9 Login Page



**Figure 4.14 Login Page**

In this page we will know how login activity work:

- If you log in before registering, a message will appear to the user stating that this email does not exist!
- If you enter an incorrect password, a message will appear to the user with an incorrect email or password!
- If you forgot the password, you can reset the password by clicking the "Forgot password" button.
- After logging in successfully, you will be directed to the chat page.
- If not, you must create a new account by clicking the join now button.

#### 4.1.10 Reset Password

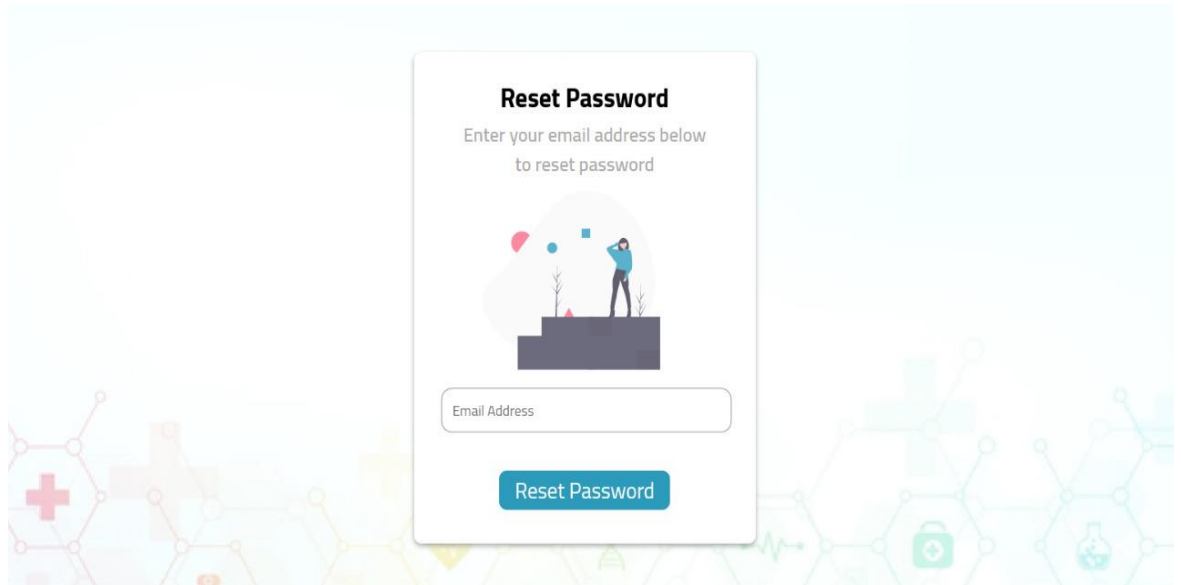


Figure 4.15 Reset Password

#### 4.1.11code verification for reset password

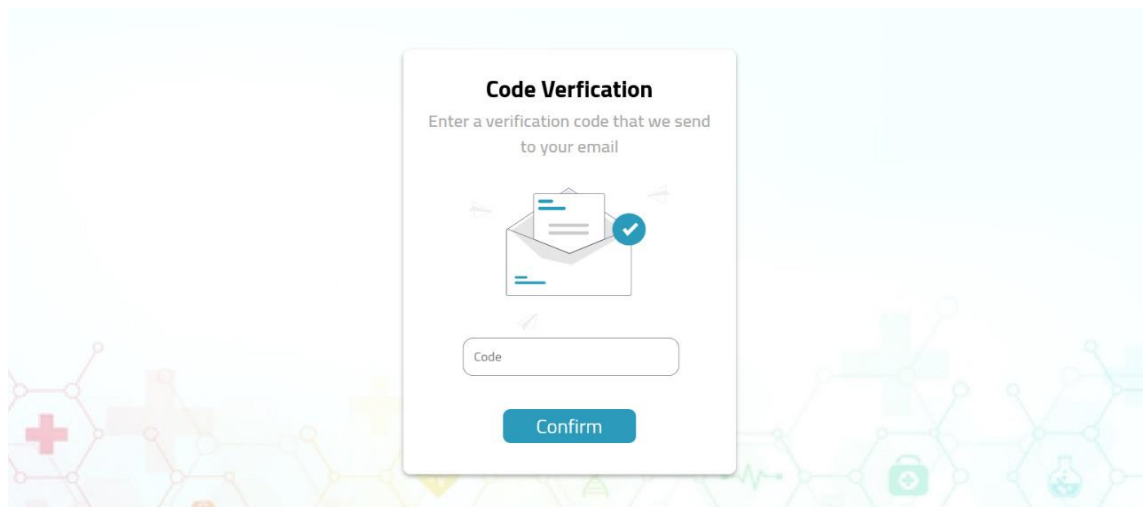


Figure 4.16 code verification for reset password

#### 4.1.12confirm password

In this page, you must enter the code that was sent to your e-mail:

- if you enter incorrect code, a message will appear "You've entered incorrect code!".
- if you enter correct code, you will be directed to the confirm password page.



## Password Reset Code

Your password reset code is 292340

**Figure 4.17 Send code to mail**

A 'Confirm Password' form with a light blue background. At the top, it says 'Confirm Password' in bold. Below this is an illustration of a person at a computer with a lock icon. The form contains two input fields: 'New Password' and 'Confirm Password'. At the bottom is a blue 'Confirm' button. The background of the entire image features a network of nodes and lines with various icons like a plus sign, a heart, and a flask.

**Figure 4.18 Confirm Password**

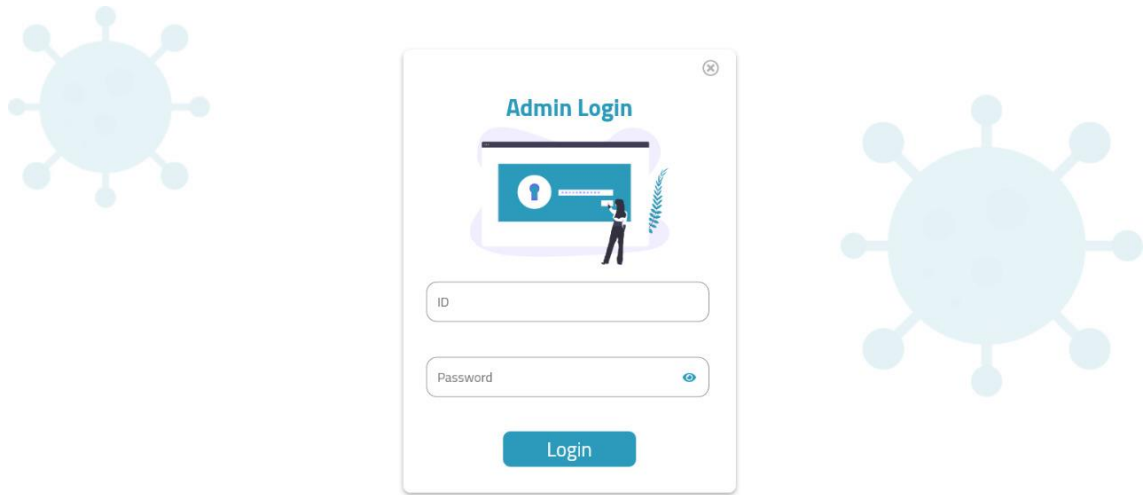
In this page, you must enter password and confirm password:

- if password not match with confirm password, a message will appear
- if password match with confirm password, you will be directed to the login page.



## 4.2 ADMIN PATH

### 4.2.1 Login Page

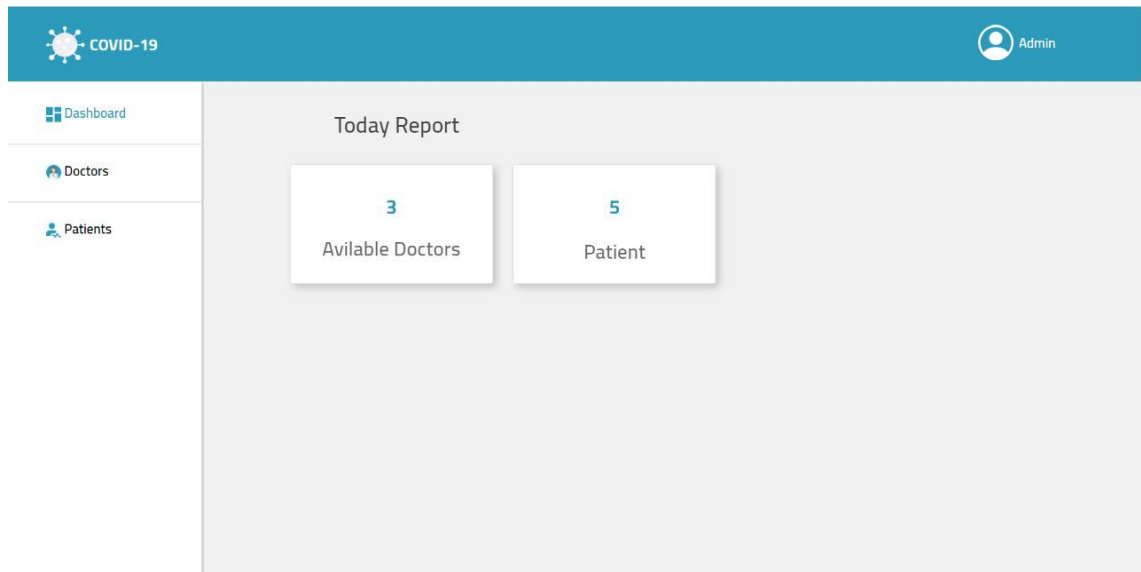


**Figure 4.19 login page**

In this page we will know how admin login work:

- If you enter an incorrect username or password, a message will appear to the user with an "Invalid username or password!".
- After logging in successfully, you will be directed to the dashboard page.

### 4.2.2 Admin Page

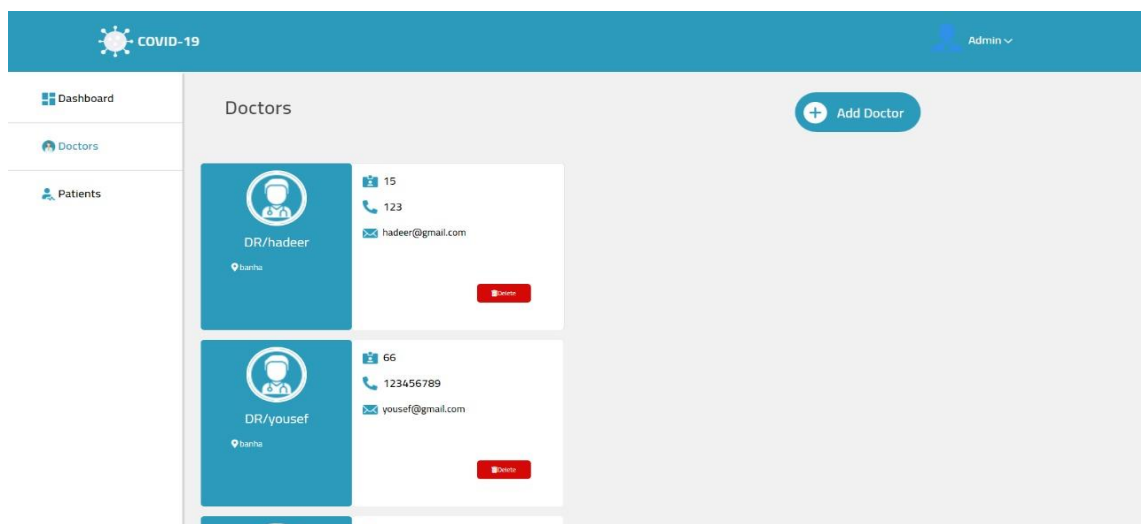


**Figure 4.20 Admin page**

In this page, the administrator can see the number of doctors and the number of patients available on the site.

- When he clicks on the Doctors or Patients button, he will direct entry to the next page.

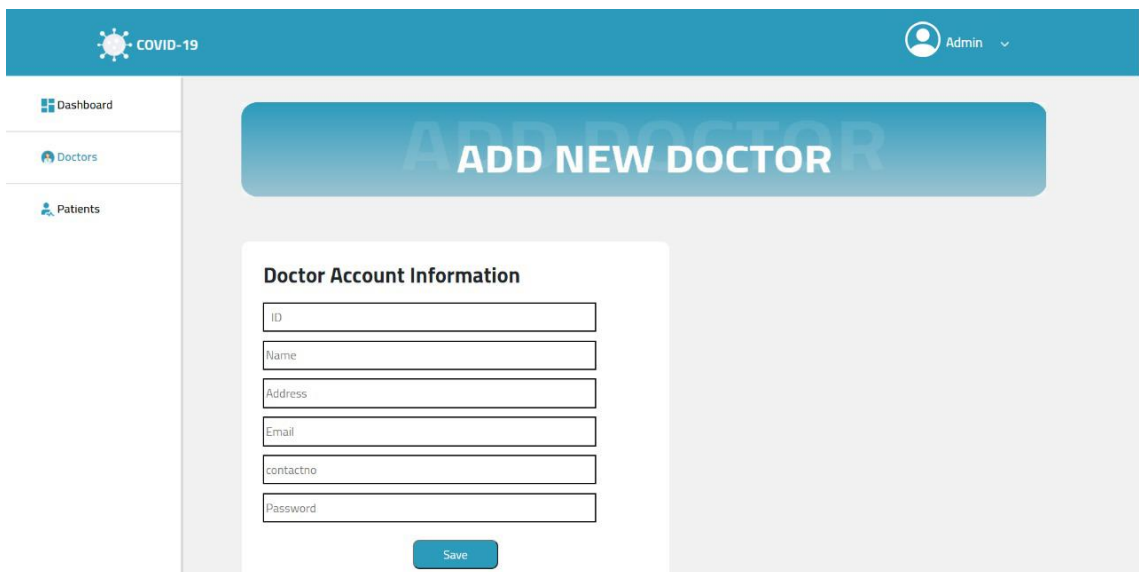
### 4.2.3 Doctor's Information Management



**Figure 4.21 Doctor's information management**

- If he clicks on the Doctors button, he will see the data of the doctors on the site and can modify or delete the data of any doctor.
- If he clicks on add doctor button, he can direct enter add doctor page.

#### 4.2.4 Add New Doctor



The screenshot shows a web application interface for managing COVID-19 related data. The top header is blue with a COVID-19 logo and the text 'COVID-19' on the left, and a user profile icon labeled 'Admin' with a dropdown arrow on the right. A left sidebar contains three menu items: 'Dashboard', 'Doctors', and 'Patients'. The main content area has a large blue banner at the top with the text 'ADD NEW DOCTOR'. Below this banner is a form titled 'Doctor Account Information'. The form contains six input fields: 'ID', 'Name', 'Address', 'Email', 'contactno', and 'Password'. A blue 'Save' button is located at the bottom of the form.

**Figure 4.22 Add new doctor**

- In this page, the administrator can add a doctor by entering his information.

### 4.2.5 View patient Information

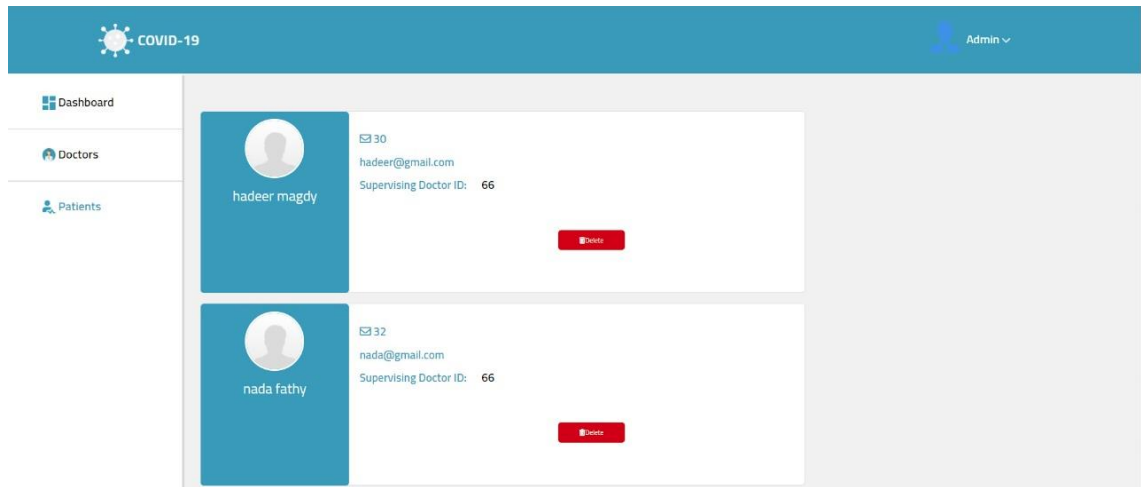


Figure 4.23 view patient information

If he clicks on the patient's button, he will see the data of the patients on the site and delete any patient.

## 4.3 DOCTOR PATH

### 4.3.1 Login Page

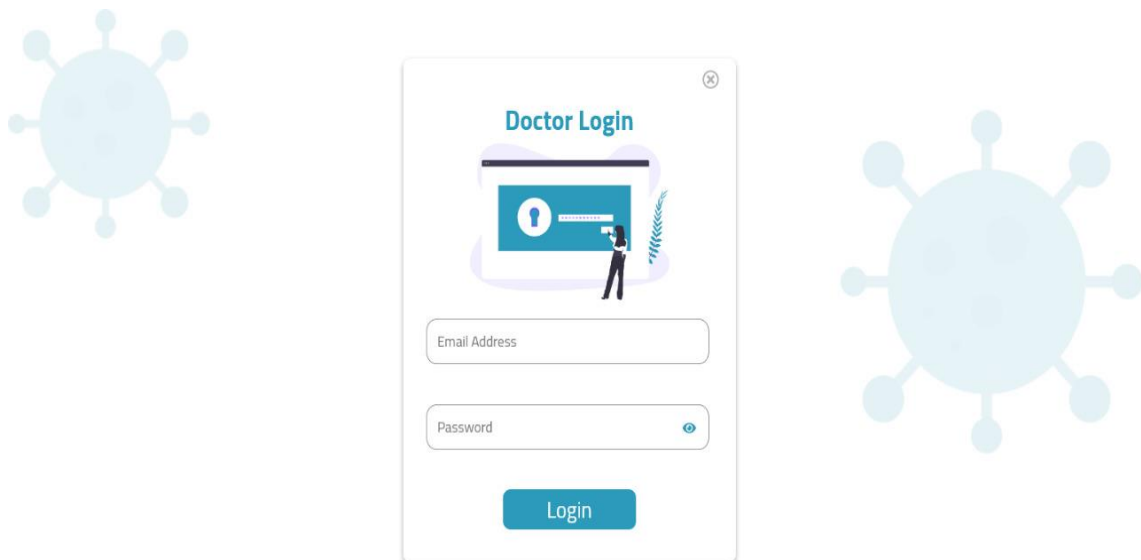
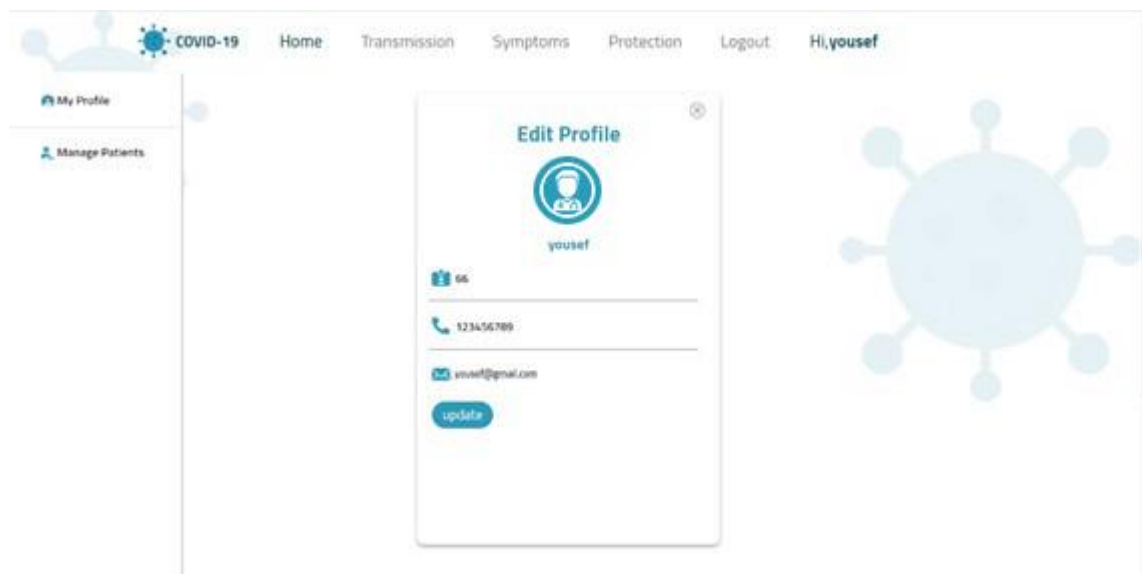


Figure 4.24 login page

In this page we will know how admin login work:

- If you do not enter one or both of the email and password, the message "All input fields are required!".
- If you enter an e-mail that has not been registered before by the admin, this message will appear to you "This email not Exist!".
- If you enter an incorrect password, a message will appear to the user with an "incorrect email or password!".
- After logging in successfully, you will be directed to the dashboard page.

### 4.3.2 Doctor Page

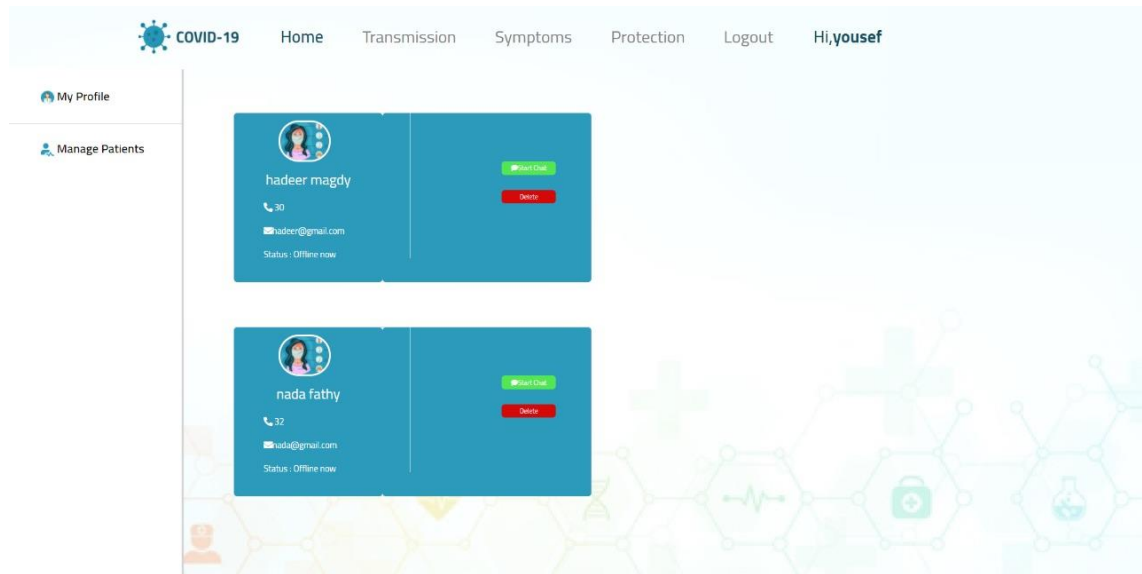


**Figure 4.25 doctor page**

If he clicks on my profile button:

- Doctor can show and update his information.

### 4.3.3 View and management patient Information



**Figure 4.26 view and management patient information**

If he clicks on manage patient's button:

- On this page the doctor can see all the patients who follow him and he can delete or write any comment for any patient.
- If the doctor clicks on the start chat button, he will be taken directly to the chat page.

### 4.3.4 Start Chat with Patients

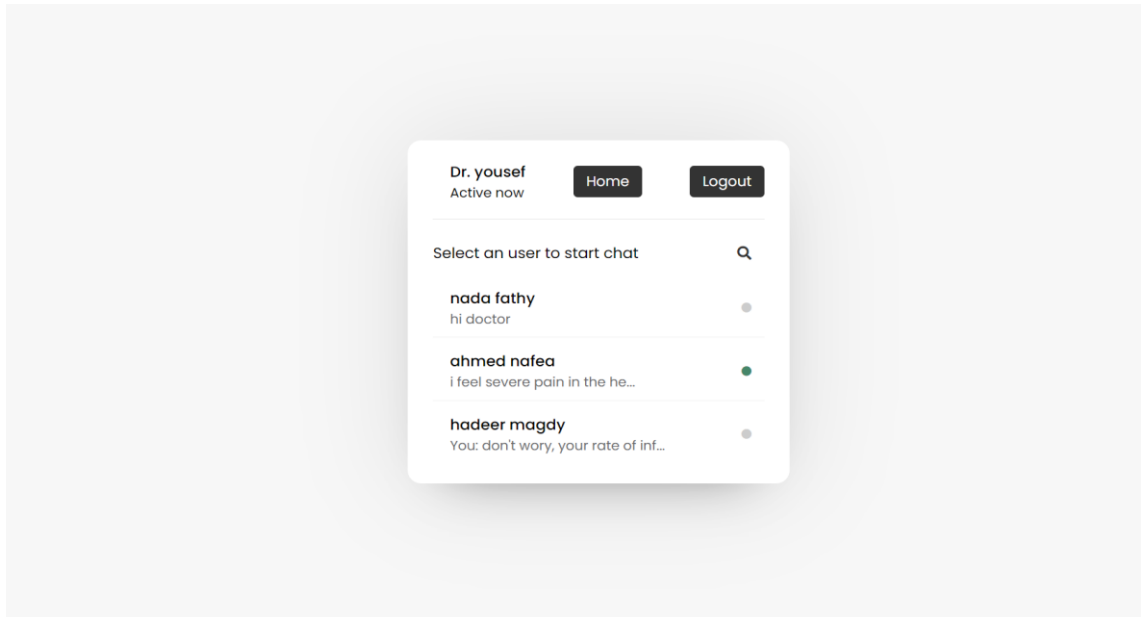


Figure 4.27 start chat with patients

-In this page:

- Patients who follow him will appear to him.
- If he chooses any patient, he will move to a private chat between them.
- If the doctor clicks on the logout button, he will return to the login page.

### 4.3.5 Chat Details

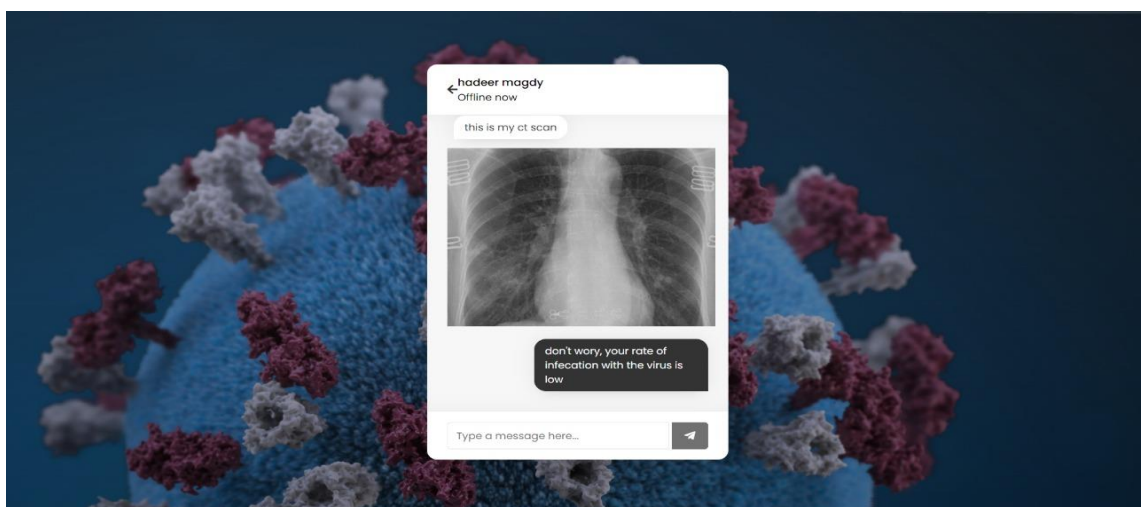


Figure 4.28 chat details

-In this page doctor will chat with patient.





```

4 include('include/config.php');
5 if(!isset($_SESSION['unique_id'])){
6     header("location: login.php");
7 }
8
9 if(isset($_GET['del']))
10 {
11     $mysql_query($con, "delete from doctors where id = '".$_GET['id']."'");
12     $_SESSION['msg']="data deleted !!!";
13 }
14 $sql = $mysql_query($con, "SELECT * FROM users WHERE unique_id = ($_SESSION['unique_id'])");
15 if($mysql_num_rows($sql) > 0)
16 {
17     $row = $mysql_fetch_assoc($sql);
18 }
19 if(isset($row['doctor_id']))
20 {
21     header("location: users.php");
22 }
23
24 <!DOCTYPE html>
25 <html lang="en">
26 <head>
27     <meta charset="UTF-8">
28     <meta name="viewport" content="width=device-width, initial-scale=1.0">
29     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-g1J6kkoiN00v6H24Jgth8i3Hj9vp7H/tdIUL7qTh1GE+H2414GfL+9CNq" crossorigin="anonymous">
30     <link rel="stylesheet" href="assets/css/style.css">
31     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" integrity="sha512-K1KjZiE2kZiADuRNMIEgkziznQrNah74XeaCY0thtak7AnPja0KYeR8w4L9kAp2zn" crossorigin="anonymous">
32     <link rel="icon" href="assets/images/virus.png">
33     <title>Doctors</title>
34
35 </head>
36 <body style="background-color: #f1f1f1;">
37     <header class="pp" style="background-color: #2b9abb; height: 100px; ">
38         <div class="cont">
39             <nav style="" class="navbar navbar-expand-lg navbar-light">
40                 <div class="container">
41                     <a style="position: relative; right: 60px; top: 10px;" class="navbar-brand" href="#">
42
43                     <a data-bs-toggle="dropdown" aria-expanded="false" style="text-decoration: none;" href="#">
44                         <span></span></a>
45                     <ul class="dropdown-menu" aria-labelledby="navbarDropdown">
46                         <li><a class="dropdown-item" href="logout.php"> Logout</li>
47                     </ul>
48                 </div>
49             </nav>
50         </div>

```

```

1 session_start();
2 include_once "php/config.php";
3 if(!isset($_SESSION['unique_id'])){
4     header("location: login.php");
5 }
6 }
7
8 <?php include_once "header.php"; ?>
9 <body>
10     <div class="wrapper">
11         <section class="users">
12             <header>
13                 <div class="content">
14                     <?php
15
16                         $sql = mysqli_query($conn, "SELECT * FROM users WHERE unique_id = {$_SESSION['unique_id']}");
17                         if(mysqli_num_rows($sql) > 0){
18                             $row = mysqli_fetch_assoc($sql);
19                         }
20                         if($row['doctor_id']!=0)
21                         {
22                             $user_id=$row['user_id'];
23                             $doc_id=$_GET['id'];
24                             $sql2 = mysqli_query($conn, "SELECT * FROM doctors WHERE id = {$doc_id}");
25                             if(mysqli_num_rows($sql) > 0){
26                                 $row2 = mysqli_fetch_assoc($sql2);
27                             }
28                             $admin_id=$row2['admin_id'];
29                             $upd=mysqli_query($conn,"Update users set admin_id ={$admin_id} , doctor_id={$doc_id} where user_id='{$user_id}'");
30
31                         }
32                     >?>
33                 <div class="details">
34                     <span><?php echo $row['fname']. " " . $row['lname'] ?></span>
35                     <p><?php echo $row['status']; ?></p>
36                 </div>
37             </div>
38             <a href="php/logout.php?logout_id=<?php echo $row['user_id']; ?>" class="logout">Logout</a>
39         </header>
40         <div class="search">
41             <span class="text">Select an user to start chat</span>
42             <input type="text" placeholder="Enter name to search...">
43             <button<i class="fas fa-search"></i></button>
44         </div>
45         <div class="users-list">
46
47         </div>
48     </section>

```

```

1 <?php
2 session_start();
3 include_once "config.php";
4 $outgoing_id = $_SESSION['unique_id'];
5 $sql = mysqli_query($conn, "SELECT * FROM users WHERE unique_id = ({$_SESSION['unique_id']})");
6 $row = mysqli_fetch_assoc($sql);
7 $id=$row['doctor_id'];
8 $sql = "SELECT * FROM doctors WHERE id='$id'";
9 $query = mysqli_query($conn, $sql);
10 $output = "";
11 if(mysqli_num_rows($query) == 0){
12     $output .= "No users are available to chat";
13 }elseif(mysqli_num_rows($query) > 0){
14     include_once "data.php";
15 }
16 echo $output;
17 ?>

```

Figure 4.31 Chat

## chat details

```

1 <?php
2 session_start();
3 include_once "php/config.php";
4 if(!isset($_SESSION['unique_id'])){
5     header("location: login.php");
6 }
7 ?>
8 <?php include_once "header.php"; ?>
9 <body>
10 <div class="wrapper">
11 <section class="chat-area">
12 <header>
13 <?php
14     $user_id = mysqli_real_escape_string($conn, $_GET['user_id']);
15     $sql = mysqli_query($conn, "SELECT * FROM users WHERE unique_id = ({$_SESSION['unique_id']})");
16     $row = mysqli_fetch_assoc($sql);
17     $user_id=$row['user_id'];
18     $doctor_id=$row['doctor_id'];
19     $sql = mysqli_query($conn, "SELECT * FROM doctors WHERE id = ({$_doctor_id})");
20     $row = mysqli_fetch_assoc($sql);
21 ?>
22 <a href="users.php" class="back-icon"><i class="fas fa-arrow-left"></i></a>
23 <div class="details">
24 <span><?php echo "Dr. " . $row['doctorName'];></span>
25 <p><?php echo $row['status'];></p>
26 </div>
27 </header>
28 <div class="chat-box">
29 <div>
30 <form action="insert-chat.php" class="typing-area">
31 <input type="text" class="incoming_id" name="incoming_id" value="<?php echo $doctor_id; ?>" hidden>
32 <input type="text" name="message" class="input-field" placeholder="Type a message here..." autocomplete="off">
33 <button><i class="fab fa-telegram-plane"></i></button>
34 </form>
35 </div>
36 </div>
37 </section>
38 </div>
39 <script src="javascript/chat.js"></script>
40 </body>
41 </html>

```

```

1 <?php
2 while($row = mysqli_fetch_assoc($query)){
3     $sql2 = "SELECT * FROM messages WHERE (incoming_msg_id = {$row['id']}
4         OR outgoing_msg_id = {$row['id']}) AND (outgoing_msg_id = {$outgoing_id}
5         OR incoming_msg_id = {$outgoing_id}) ORDER BY msg_id DESC LIMIT 1";
6     $query2 = mysqli_query($conn, $sql2);
7     $row2 = mysqli_fetch_assoc($query2);
8     (mysqli_num_rows($query2) > 0) ? $result = $row2['msg'] : $result = "No message available";
9     (strlen($result) > 28) ? $msg = substr($result, 0, 28) . '...' : $msg = $result;
10    if(isset($row2['outgoing_msg_id'])){
11        ($outgoing_id == $row2['outgoing_msg_id']) ? $you = "You: " : $you = "";
12    }else{
13        $you = "";
14    }
15    ($row['status'] == "Offline now") ? $offline = "offline" : $offline = "";
16    ($outgoing_id == $row['id']) ? $hid_me = "hide" : $hid_me = "";
17
18    $output .= '
19 <a href="chat.php?user_id='.$row['id'].'"
20 <div class="content">
21 <div class="details">
22 <span>'. "Dr. ". $row['doctorName'] .'</span>
23 <p>'. $you . $msg .'</p>
24 </div>
25 <div class="status-dot '. $offline .'><i class="fas fa-circle"></i></div>
26 </a>';
27 }
28 ?>

```

```
1 <?php
2 session_start();
3 if(isset($_SESSION['unique_id'])){
4     include_once "config.php";
5     $outgoing_id = $_SESSION['unique_id'];
6     $incoming_id = mysqli_real_escape_string($conn, $_POST['incoming_id']);
7     $output = "";
8     $sql = "SELECT * FROM messages LEFT JOIN users ON users.unique_id = messages.outgoing_msg_id
9           WHERE (outgoing_msg_id = {$outgoing_id} AND incoming_msg_id = {$incoming_id})
10            OR (outgoing_msg_id = {$incoming_id} AND incoming_msg_id = {$outgoing_id}) ORDER BY msg_id";
11     $query = mysqli_query($conn, $sql);
12     if(mysqli_num_rows($query) > 0){
13         while($row = mysqli_fetch_assoc($query)){
14             if($row['outgoing_msg_id'] === $outgoing_id){
15                 $output .= '<div class="chat outgoing">
16                     <div class="details">
17                         <p>'. $row['msg'] . '</p>
18                     </div>
19                     </div>';
20             }else{
21                 $output .= '<div class="chat incoming">
22                     <div class="details">
23                         <p>'. $row['msg'] . '</p>
24                     </div>
25                     </div>';
26             }
27         }
28     }else{
29         $output .= '<div class="text">No messages are available. Once you send message they will appear here.</div>';
30     }
31     echo $output;
32 }else{
33     header("location: ../login.php");
34 }
35 }
36 ?>
```

```
1 <?php
2 session_start();
3 if(isset($_SESSION['unique_id'])){
4     include_once "config.php";
5     $outgoing_id = $_SESSION['unique_id'];
6     $incoming_id = mysqli_real_escape_string($conn, $_POST['incoming_id']);
7     $message = mysqli_real_escape_string($conn, $_POST['message']);
8     if(!empty($message)){
9         $sql = mysqli_query($conn, "INSERT INTO messages (incoming_msg_id, outgoing_msg_id, msg)
10                                VALUES ({$incoming_id}, {$outgoing_id}, '{$message}')") or die();
11     }
12 }else{
13     header("location: ../login.php");
14 }
15
16
17
18 ?>
```

Figure 4.32 Chat details

## 4.4.2 Admin Path

### Admin login

```

1 <?php
2 require_once "login.php";
3 if(isset($_SESSION['username']))
4 {
5     header("location: dashboard.php");
6 }
7
8 <!DOCTYPE html>
9 <html lang="en">
10 <head>
11 <meta charset="UTF-8">
12 <meta name="viewport" content="width=device-width, initial-scale=1.0">
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-gltf6kuglQ00y+H8Df7azDul8x6f1ca9wJkH-8R6DvddVhY7FAAarekkaP1" crossorigin="anonymous">
14 <link rel="stylesheet" href="assets/css/style.css">
15 <script type="text/javascript" src="assets/js/script.js"></script>
16 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" integrity="sha512-44CK96q23WU7jQ3WYdUwXW6p0VjTui6+kgS4KpVWPd0y77Aw0H3jO3uL657Lw5UV38S+jQTHbtrCE3Nc6tQ==" crossorigin="anonymous">
17 <link rel="icon" href="assets/images/virus.png">
18 <title>Admin Login</title>
19 </head>
20 <body style="background-image:url(assets/images/Group100.png); background-repeat: no-repeat; background-size: 100%;">
21 <div class="container">
22 <div class="row">
23 <div style="margin-top: 4.5%; class="col-md-12 col-sm-12">
24 <div style="background-color: white; height: 600px; width: 450px; border-radius: 10px; margin-left: 32%; box-shadow: 0 3px 6px 0 rgba(0, 0, 0, 0.33); method="POST" action="index.php">
25 <div style="text-align: center; padding-top: 4%; color: #2b9abb; font-family: Cairo; font-weight: bold;">Admin Login</div>
26 <div style="width: 60%; margin-left: 20%; text-align: center;">
27 <input style="width: 90%; height: 30px; border: 1px solid #ccc; border-radius: 5px;" type="text" placeholder="Email Address" name="username">
28 <input style="width: 90%; height: 30px; border: 1px solid #ccc; border-radius: 5px;" type="password" id="password-field" placeholder="Password" name="password">
29 <input style="margin-left: 10px; width: 100px; height: 30px; border: 1px solid #ccc; border-radius: 5px;" type="button" value="Login">
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>

```

Figure 4.33 admin login front

```

1 <?php
2 session_start();
3 $username="";
4 $password="";
5 $errors_val= array();
6 if(isset($_POST['submit'])){
7     include_once "include/config.php";
8     $username = mysqli_real_escape_string($con, $_POST['username']);
9     $password = mysqli_real_escape_string($con, $_POST['password']);
10     if(empty($username) && empty($password)){
11         $sql = mysqli_query($con, "SELECT * FROM admin WHERE username = '{$username}'");
12         if(mysqli_num_rows($sql) > 0){
13             $row = mysqli_fetch_assoc($sql);
14             $user_pass = md5($password);
15             $enc_pass = $row['password'];
16             if($password == $enc_pass){
17                 $_SESSION['username'] = $row['username'];
18             }else{
19                 $errors_val['email_password'] = "username or Password is Incorrect!";
20             }
21         }else{
22             $errors_val['email'] = "Username - This username not Exist!";
23         }
24     }else{
25         $errors_val['required'] = "All input fields are required!";
26     }
27 }
28 ?>

```

Figure 4.34 admin login back

## Admin dashboard

```
1 </php>
2 session_start();
3 error_reporting(0);
4 include('include/config.php');
5 $sql = mysql_query($con, "SELECT count(id) AS total_doctor FROM doctors ");
6 $row = mysql_fetch_assoc($sql);
7 $total_doctor=$row['total_doctor'];
8 $sql2 = mysql_query($con, "SELECT count(user_id) AS total_patient FROM users ");
9 $row2 = mysql_fetch_assoc($sql2);
10 $total_patient=$row2['total_patient'];
11 >
12 <!DOCTYPE html>
13 <html lang="en">
14 <head>
15 <meta charset="UTF-8">
16 <meta name="viewport" content="width=device-width, initial-scale=1.0">
17 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-g1J6KkoQ00y+H0P7az0ul0ctb1caT9wJ0r88bDvdV9yTFAAsrekukmP1" crossorigin="anonymous">
18 <link rel="stylesheet" href="assets/css/style.css">
19 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css" integrity="sha512-+42CK96+qNfURS+ck19EIR+20ht11oN19G1KS57V1MyYpYclwUqQ-9AfzsmV2B1aL13196P9sdh" crossorigin="anonymous">
20 <link rel="icon" href="assets/images/virus.png">
21 <title>Dashboard</title>
22 </head>
23 <body style="background-color: #f1f1f1;">
24 <div class="pp" style="background-color: #2b9abb; height: 100px; width: 100%;>
25 <div class="cont">
26 <nav class="navbar navbar-expand-lg navbar-light">
27 <div class="container">
28 <a style="position: relative; right: 60px; top: 10px;" class="navbar-brand" href="#"></a>
29 <a data-bb-toggle="dropdown" aria-expanded="false" style="text-decoration: none;" href="#"><span>
30 <ul class="dropdown-menu" aria-labelledby="dropdownMenuButton">
31 <li>
32 <a class="dropdown-item" href="logout.php"> Logout</a>
33 </li>
34 </ul>
35 </div>
36 </nav>
37 <div class="sidebar">
38 <a style="margin-left: 20px;" class="ooo" href="dashboard.php"> Dashboard</a>
39 
40 <a style="margin-left: 20px;" class="oooo" href="dashboard2.php"> Doctors</a><br>
41 
42 <a style="margin-left: 20px;" class="oooo" href="dashboard3.php"> Patients</a><br>
43 </div>
44 </div>
45 </div>
46 </body>
47 </html>
```

Figure 4.35 dashboard

## Login



## Chat

```
1 <?php
2 session_start();
3 include_once "include/config.php";
4 if(!isset($_SESSION['id'])){
5     header("location: index.php");
6 }
7 ?>
8 <?php include_once "header.php"; ?>
9 <body>
10 <div class="wrapper">
11     <section class="users">
12         <header>
13             <div class="content">
14                 <?php
15
16                 $sql = mysqli_query($con, "SELECT * FROM doctors WHERE id = {$_SESSION['id']}");
17                 if(mysqli_num_rows($sql) > 0){
18                     $row = mysqli_fetch_assoc($sql);
19                 }
20                 ?>
21             <div class="details">
22                 <span><?php echo "Dr. " . $row['doctorName']; ?></span>
23                 <p><?php echo $row['status']; ?></p>
24             </div>
25         </header>
26         <a href="dashboard.php?logout_id=<?php echo $row['id']; ?>" class="logout">Home</a>
27         <a href="php/logout.php?logout_id=<?php echo $row['id']; ?>" class="logout">Logout</a>
28     </section>
29     <div class="search">
30         <span class="text">Select an user to start chat</span>
31         <input type="text" placeholder="Enter name to search...">
32         <button<i class="fas fa-search"></i></button>
33     </div>
34     <div class="users-list">
35
36     </div>
37 </section>
38 </div>
39
40 <script src="javascript/users.js"></script>
41
42 </body>
43 </html>
44
45
1 <?php
2 session_start();
3 include_once "../include/config.php";
4 $outgoing_id = $_SESSION['id'];
5 $sql = "SELECT * FROM users WHERE doctor_id = {$_SESSION['id']} order by user_id DESC";
6 $query = mysqli_query($con, $sql);
7 $output = "";
8 if(mysqli_num_rows($query) == 0){
9     $output .= "No users are available to chat";
10 }elseif(mysqli_num_rows($query) > 0){
11     include_once "data.php";
12 }
13 echo $output;
14 ?>
```

Figure 4.39 chat

### 4.4.4 API (application program interface)

-We use http request between php (backend) and flask to classify input data from user and predict the status (positive or negative).



```

1 <?php
2 session_start();
3 //error_reporting(0);
4 include('include/config.php');
5
6
7 //The url you wish to send the POST request to
8 $url = 'http://127.0.0.1:5000/';
9
10
11 $data = array('x1' => 'Fatigue', 'x2' => 'Headache');
12
13 // use key 'http' even if you send the request to https://...
14 $options = array(
15     'http' => array(
16         'header' => "Content-type: application/x-www-form-urlencoded\r\n",
17         'method' => 'POST',
18         'content' => http_build_query($data)
19     )
20 );
21 $context = stream_context_create($options);
22 $response_data = file_get_contents($url, false, $context);
23 if ($response_data === FALSE) { /* Handle error */ }
24
25 var_dump($response_data);
26

```

main.py X result1.html stats.html

```

main.py > classify
12 GET /home():
13
14     return render_template('stats.html')
15
16
17
18 @app.route('/classify', methods=['POST','GET'])
19 def classify():
20     x1=float(request.form['sex'])
21     x2=float(request.form['ldh'])
22     x3=float(request.form['hct'])
23     x4=float(request.form['eo'])
24     x5=float(request.form['mcv'])
25     x6=float(request.form['ba'])
26     x7=float(request.form['mch'])
27     x8=float(request.form['net'])
28     x9=float(request.form['plt1'])
29     x10=float(request.form['mot'])
30     x11=float(request.form['wbc'])
31     x12=float(request.form['ne'])
32     x13=float(request.form['eot'])
33     x14=float(request.form['rbc'])
34     x15=float(request.form['ly'])
35     x16=float(request.form['bat'])
36     x17=float(request.form['alt'])
37     x18=float(request.form['hgb'])
38     x19=float(request.form['mo'])
39     x20=float(request.form['mhc'])
40     x21=float(request.form['lyt'])
41
42
43     arr=np.array([[x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,x21]])

```

Figure 4.40 API code

## 4.5 DATABASE

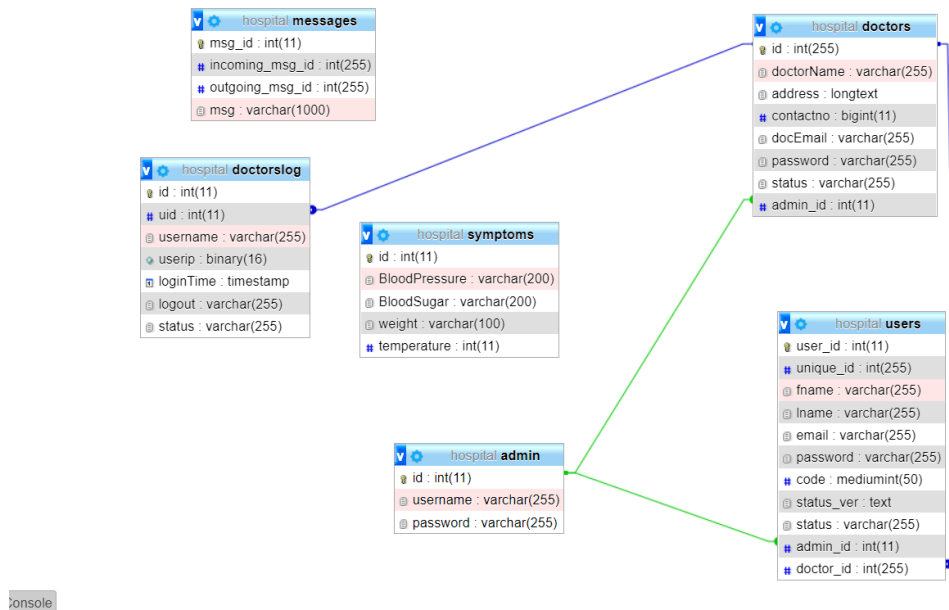


Figure 4.41 Database tables

### Patient table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	user_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	unique_id	int(255)			No	None			Change Drop More
3	fname	varchar(255)	utf8_general_ci		No	None			Change Drop More
4	lname	varchar(255)	utf8_general_ci		No	None			Change Drop More
5	email	varchar(255)	utf8_general_ci		No	None			Change Drop More
6	password	varchar(255)	utf8_general_ci		No	None			Change Drop More
7	code	mediumint(50)			No	None			Change Drop More
8	status_ver	text	utf8_general_ci		No	None			Change Drop More
9	status	varchar(255)	utf8_general_ci		No	None			Change Drop More
10	admin_id	int(11)			Yes	NULL			Change Drop More
11	doctor_id	int(255)			Yes	NULL			Change Drop More

Figure 4.42 patient table

## doctor table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>id</b>	int(255)			No	None			Change  Drop  More
<input type="checkbox"/> 2	<b>doctorName</b>	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 3	<b>address</b>	longtext	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 4	<b>contactno</b>	bigint(11)			Yes	NULL			Change  Drop  More
<input type="checkbox"/> 5	<b>docEmail</b>	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 6	<b>password</b>	varchar(255)	latin1_swedish_ci		Yes	NULL			Change  Drop  More
<input type="checkbox"/> 7	<b>status</b>	varchar(255)	latin1_swedish_ci		No	online			Change  Drop  More
<input type="checkbox"/> 8	<b>admin_id</b>	int(11)			No	None			Change  Drop  More

Figure4.43 doctor table

## admin table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>id</b>	int(11)			No	None			Change  Drop  More
<input type="checkbox"/> 2	<b>username</b>	varchar(255)	latin1_swedish_ci		No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>password</b>	varchar(255)	latin1_swedish_ci		No	None			Change  Drop  More

Figure 4.44 admin table

## Messages table

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	<b>msg_id</b>	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
<input type="checkbox"/> 2	<b>incoming_msg_id</b>	int(255)			No	None			Change  Drop  More
<input type="checkbox"/> 3	<b>outgoing_msg_id</b>	int(255)			No	None			Change  Drop  More
<input type="checkbox"/> 4	<b>msg</b>	varchar(1000)	utf8mb4_general_ci		No	None			Change  Drop  More

Figure 4.44 message table

## 4.6 MACHINE LEARNING

The concept of machine learning is something born out of this environment. Computers can analyse digital data to find patterns and laws in ways that is too complex for a human to do. The basic idea of machine learning is that a computer can automatically learn from experience. Although machine learning applications vary, its general function is similar throughout its applications. The computer analyses a large amount of data, and finds

patterns and rules hidden in the data. These patterns and rules are mathematical in nature, and they can be easily defined and processed by a computer.

The computer can then use those rules to meaningfully characterize new data. The creation of rules from data is an automatic process, and it is something that continuously improves with newly presented data.

#### **4.6.1 Methods**

There are numerous algorithms to create a classification model. While they all essentially have the same task, which is predicting a dependent variable based on independent variables, they are based on different mathematical methods.

- In our system patient have two ways to check so we used two models.

##### **First model (Symptoms)**

We tried some algorithms such as:

- Naive Bayes with accuracy 74%
- Random Forest with accuracy 89%
- Decision Tree with accuracy 84%
- Logistic Regression with accuracy 92%
- Support Vector Machine (SVM) with accuracy 89%

So, we used Logistic Regression which has the best accuracy.

```
#Logistic Regression

import sys
import pickle
print(sys.executable)
import pandas as pd
from sklearn.model_selection import cross_val_score

# Importing the dataset
dataset = pd.read_csv('G:\Covid Dataset.csv')
X = dataset.iloc[:, :20]
y = dataset.iloc[:, 20]

from sklearn.feature_selection import SelectPercentile
from sklearn.feature_selection import chi2
FeatureSelection = SelectPercentile(score_func = chi2, percentile=90) # score_func can = f_classif
X = FeatureSelection.fit_transform(X, y)

# Training the Logistic Regression model on the Training set
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X,y)

# Making the Confusion Matrix
scores = cross_val_score(classifier, X, y, cv=5)

print('Logistic Regression.mean: ',scores.mean() )
print('Logistic Regression.std: ',scores.std() )
model_path = 'model.pickle'
with open(model_path, 'wb') as f:
    pickle.dump(classifier, f)
    print('Model saved to ' + model_path)
```

Figure 4.45 Logistic regression algorithm

## Second model (using Naive Bayes algorithm)

We tried some algorithms such as:

- Naive Bayes with accuracy 80%
- Random Forest with accuracy 78.9%
- Decision Tree with accuracy 73%
- Logistic Regression with accuracy 77%
- Support Vector Machine (SVM) with accuracy 73%

So, we used Naive Bayes algorithm which has the best accuracy.

```
# Naive Bayes

# Importing the libraries
import pandas as pd
import sys
import pickle
print(sys.executable)
from sklearn.model_selection import cross_val_score

dataset = pd.read_csv('G:\DataCovid.txt',delimiter="\t")
X = dataset.iloc[:, :21]
y = dataset.iloc[:, 21]

# Feature Scaling
#from sklearn.preprocessing import StandardScaler
#sc = StandardScaler()

# Training the Naive Bayes model on the Training set
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X,y)

scores = cross_val_score(classifier, X, y, cv=5)
# Making the Confusion Matrix
print(' Naive Bayes.mean: ',scores.mean() )
print(' Naive Bayes.std: ',scores.std() )

model_path = 'model.pickle'
with open(model_path, 'wb') as f:
    pickle.dump(classifier, f)
    print('Model saved to ' + model_path)
```

**Figure 4.46 Naïve Bayes algorithm**

## 4.7 DATASETS

As our system include two model so we have two datasets.

### 4.7.1 First dataset

This dataset is about symptoms that appears on the patient and the answer of each question is yes or no.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Breathing	Fever	Dry Cough	Sore throat	Running	Asthma	Chronic L	Headache	Heart Dise	Diabetes	Hyper Ten	Fatigue	Gastroint	Abroad tr	Contact w	Attended	Visited Pu	Family wc	Wearing M	Sanitizati	COVID
2	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	0
3	1	1	1	1	1	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0
4	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	0	0	0	0	0
5	1	1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1	1	0	0	0
6	1	1	1	1	1	1	0	1	1	1	1	0	1	0	1	0	1	0	0	0	0
7	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
8	1	1	1	1	0	0	0	1	0	1	1	1	1	1	0	1	1	1	0	0	0
9	1	1	1	1	0	1	1	0	0	0	1	1	0	1	1	0	0	1	0	0	0
10	1	1	1	1	0	1	0	1	0	0	1	0	1	0	1	1	1	0	0	0	0
11	1	1	1	1	0	0	1	0	0	0	1	1	1	0	0	0	0	1	0	0	0
12	1	1	1	1	0	0	0	1	0	1	0	1	0	0	1	0	1	0	0	0	0
13	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	0	1	0	1	0	0
14	1	1	1	1	1	0	1	1	0	0	0	1	0	1	1	0	0	1	1	0	0
15	1	1	1	1	1	0	1	0	0	0	1	1	0	0	1	1	0	0	1	0	0
16	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	1	0	0
17	1	1	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
18	1	1	1	1	0	1	0	0	1	1	0	0	0	1	0	1	0	0	1	0	0
19	1	1	1	1	0	1	1	1	1	0	0	0	1	1	0	0	1	0	0	0	0
20	1	1	1	1	1	0	0	1	1	1	1	0	1	0	0	0	0	0	1	0	0
21	1	1	1	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	0	0
22	1	1	1	1	0	1	0	0	1	0	1	1	1	0	1	0	0	0	0	0	0
23	1	1	1	1	0	1	1	1	0	0	0	1	1	0	1	1	0	0	0	0	0

Figure 4.47 Symptoms dataset

## 4.7.2 Second dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Sex	Age	CA	CK	CREA	ALP	GGT	GLU	AST	ALT	LDH	PCR	KAL	NAT	UREA	WBC	RBC	HGB	HCT	MCV	MCH
2	1	82	2.09		1.15	95	40	78	26	21	307	61.9	4.16	144	29	9.9	3.82	13.25	38.5	100.85	
3	1	51	1.97	237	0.97	54	98	98	74	84	441	116.5	4.24	133.7	30	9.2	5.21	14.9	42.7	82	
4	1	58	2.11		1	80	147	106	41	36	359	86.65	3.94	139.3		6.7	4.905	14.05	41.35	84.3	2
5	0	82	2.27	138	0.755	123.5	176.5	106	114	63	281	1.6	4.4	135.75	28	9.7	5	14.5	42.3	84.6	
6	1	79	2.07	73	1.81	62	36.5	96	28	38.5	264	317.5	4.41	139.2	66	1	2.906667	9.233333	28.6	98.43333	31.76
7	1	84	2.06	115	1.283333	75	75	95.5	40.5	27	364.5	215.1	3.925	137.9	51.5	3.8	4.373333	13.16667	39.06667	89.33333	30.06
8	1	79			1.14	75	20	143	34	22		29.5	3.11	136.5		8.9	4.14	12	35	84.5	
9	0	9	2.29	104	0.64	131	16	105	25	13	345	146.9	4.64	135.5		15	4.8	12.6	36.8	76.7	
10	0	48	2.11		0.66	200	90	104	38	36	189	78.2	4.27	135.3	31	5.7	3.47	10.2	29.8	85.9	
11	0	67	1.98		0.61	47	23	106	54	27	356	56.1	3.59	135.5	16	4.3	4.36	12.5	35.9	82.3	
12	0	68	2.25		0.6	71	19	89	24	20		55.5	3.96	133.2		3.8	4.42	12.3	37.2	84.2	
13	1	68	2.3		1.19	52	42	160	30	34	233	64.1	4.33	135.1		10.5	4.67	14	40.9	87.6	
14	0	53	2.36	45.5	0.593333	62	21.5	91.66667	36	35.5	269.5	17.15	5.186667	140	18.5	6.05	4.325	12.55	37.8	87.4	
15	0	76	2.2	349	2.42	159.5	147	640	1018.5	559.5	694	19	5.345	134.05	159	17.3	4.125	8.95	29.9	72.95	
16	0	47	2.48	79	0.53	80	33	97	30	30	169	2.8	4.01	139.5	23	10.9	5.19	15.5	45.3	87.3	
17	1	85	2.185	55	1.075	226.5	226.5	96.5	108.5	104.5	284	56.9	3.355	141.3	32	6.85	2.945	9.4	28.15	95.65	
18	1	54	2.29		0.94				93	29	27	197	61.3	3.82	137.6		7.75	4.845	13.55	40	82.55
19	1	64	2.2	29	0.63	74.5	29	109	17.5	13.5	226	114.65	4.435	135.85		11.45	3.395	10.3	32.2	94.85	3
20	1	60	2.1		2.38	50	20	154	131	28	775	157.7	4.47	132.9	96	11.6	7.06	13.6	39.6	56.1	
21	0	41	2.25		1.97	99	19	115	20	15		177.7	3.19	154		13.6	4.07	13.4	41.5	102	
22	0	55	2	204	0.585	43	8	109	67	47.5	491	58.2	4.33	140.95	22	3.833333	3.866667	11.46667	34.33333	88.76667	29.66
23	0	81	2.07		1.95	58	18	178	46	17	524	112.6	5.79	137	84	5.1	3.65	11.4	35.4	97	

Figure 4.48 Analysis dataset

## **5 CONCLUSIONS AND FUTURE IMPROVEMENTS**

The project develops modern interactive diagnostic software. And provide an easy handling checker system to help people make sure whether they have positive or negative result of the virus through their symptoms. also help people who infected with the virus to find a suitable communication channel with specialized doctor to select the proper treatment protocol.

As a future work we can focus on the following aspects:

1. Adding more types of diseases
2. Predict if the user has covid-19 through x-ray on the chest.
3. . Enhancing the security of web application
4. . Developing an android app.
5. Enhancing our models.



## **5.1 REFERENCES**

- [1] H. Joo. “A study on understanding of UI and UX, and understanding of design according to user interface change”. In: International Journal of Applied Engineering Research 12 (Jan. 2017), pp. 9931–9935.
- [2] BOWLES, M., 2019. MACHINE LEARNING IN PYTHON. [Place of publication not identified]: JOHN WILEY & Sons.
- [3] Beginning Php6 Apache MySQL Web Development. (2021). Wiley India.
- [4] Aquino, C., & Gandee, T. (2017). Front-end web development: the Big Nerd Ranch guide. Big Nerd Ranch.

## **5.2 REFERENCES TO ELECTRONIC SOURCES**

- [5] Kaggle.com. 2021. Kaggle: Your Machine Learning and Data Science Community. [online] Available at: <<https://www.kaggle.com/>> [Accessed 6 July 2021]
- [6] [https://youtube.com/playlist?list=PL6-3IRz2XF5Uq7PkI\\_PWOm\\_DLC2CFvSzU](https://youtube.com/playlist?list=PL6-3IRz2XF5Uq7PkI_PWOm_DLC2CFvSzU)

### الملخص

التشخيص الطبي مهمة ضرورية يجب القيام بها بأكبر قدر ممكن من الدقة والفعالية. في هذا المشروع ، سنتمكن من تشخيص المرض (كوفيد -19) ، وسنكون قادرين على تشخيص المرض (كوفيد -19) باستخدام خوارزميات تعليم الآلة. يجب المستخدم على أسئلة محددة ثم يقوم النظام بإرجاع النتيجة. كما يساعد المشروع الأشخاص على إيجاد طريقة تواصل مناسبة مع طبيب متخصص لاختيار بروتوكول العلاج المناسب.