

**18CSC303J DATABASE  
MANAGEMENT SYSTEM  
FINAL PROJECT REPORT**

**ONLINE APPOINTMENT  
SCHEDULING SYSTEM**

**FULLSTACK WEB DEVELOPMENT USING PHP AND MySQL**

**ANUSHKA RAY (RA1811003010266)  
SHAHBAZ SYED (RA1811003010280)  
CSE E1**

# ABSTRACT

Coronavirus disease (COVID-19), is an infectious disease caused by a newly discovered coronavirus (SARS-CoV-2), which has spread rapidly throughout the world. While countries, including India, have taken strong measures to contain the spread of COVID-19 through better diagnostics and treatment, vaccines will provide a lasting solution by enhancing immunity and containing the disease spread.

In the traditional appointment system patients have to come to the hospital and queue at the appointment window to make the appointment. The patient can, however, decide to schedule an appointment, but this option does not usually work well for all parties involved. Parties involved include: the patient, the medical personnel and the hospital. Thus, this project focuses on making a system which helps patients to book appointments online in order to practice safe social distancing and reduce the spread of the deadly COVID-19 virus.

# CONTENT

Abstract	(i)
Introduction .....	1
Technologies Used .....	2
a) Front-end (Client side) .....	2
b) Back-end (Server side) .....	3
c) Database .....	3
Architecture .....	4
Design .....	6
ER Diagram .....	8
Implementation .....	9
Implementation(Screenshots) .....	13
Conclusion .....	19
References .....	20

# Introduction

Our online vaccination appointment scheduling system is a system through which a patient can access the website of the doctor/clinic, and through this online portal, the patient can easily make their appointments. The clinic can update the appointment status of the patients making it more informative.

This online vaccination appointment system has the following features:

## **For patients:**

- Register as a patient account
- View patient profile along with past appointment history
- View slot availability
- Book an appointment

## **For admin:**

- Update status of appointments
- See appointment list
- Update schedule list
- Show all patients from database

# Technologies Used

## 1) Front-end (Client side)

- a) HTML:** The Hypertext Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It contains specific syntax, file and naming conventions that show the computer and the web server that it is in HTML and should be read as such. By applying these HTML conventions to a text file in virtually any text editor, a user can write and design a basic webpage, and then upload it to the internet.
- b) CSS:** Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in HTML. It is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics.
- c) JavaScript:** JavaScript is a client scripting language which is used for creating web pages. It is a standalone language developed in Netscape. It is used when a webpage is to be made dynamic and add special effects on pages like rollover, roll out and many types of graphics.
- d) Bootstrap:** Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. We

have used it to make our application more responsive and device friendly.

## 2) Back-end (Server side)

- a) **PHP:** PHP is a popular general-purpose scripting language that is especially suited to web development. It is widely used as a server-side language for creating dynamic web pages. PHP scripts may be embedded into HTML code, or it can be used in combination with various web template systems, web content management system and web frameworks and can only be interpreted on a server that has PHP installed.

## 3) Database

- a) **MySQL:** MySQL is an open-source relational database management system based on SQL – Structured Query Language. The application is used for a wide range of purposes but the most common use for MySQL however, is for the purpose of a web database.

The MySQLi extension of the PHP is used to connect and provide an interface with the MySQL database. It acts as the driver.

# Architecture

Our website is designed to make it easier for the patients to quickly register themselves and book an appointment for receiving the vaccine. The patients can register with us by providing a few mandatory details such as their Aadhar, phone and email. After registering themselves the users can then select a particular date to check for appointment availability. He can then confirm booking on any available date of his liking.

The other part of the site is the admin section which is handled by the administration. They are responsible for updating the profile of the doctors and their schedules. The admins also have access to the data of the users. They can allocate and make changes depending on the situation.

This has been implemented through the use of a MySQL database as the backend. 4 tables have been created to the data needed for the functioning of this website.

Their uses are:

1. **Patient table:** This table stores the details of the patients. When a user registers on the website, the details get inserted into the table. This helps keep a record of the patient's bio data. This info can then be viewed by the patient on the profile page.

2. **Doctor table:** It stores the info about the doctors. This table is managed by the admin. They update the table as and when needed.
3. **Doctor schedule table:** It stores the schedule of the doctor i.e. the day when he is available. If he gets booked then the table updates to show that he is unavailable. This table is also managed by the admin who updates the table depending on the schedule of the doctors.
4. **Appointment table:** When a patient books an appointment from the home page, the details of the appointment gets inserted into this table. It keeps a record of all the assignments. The admin can close the appointment once it has been completed.

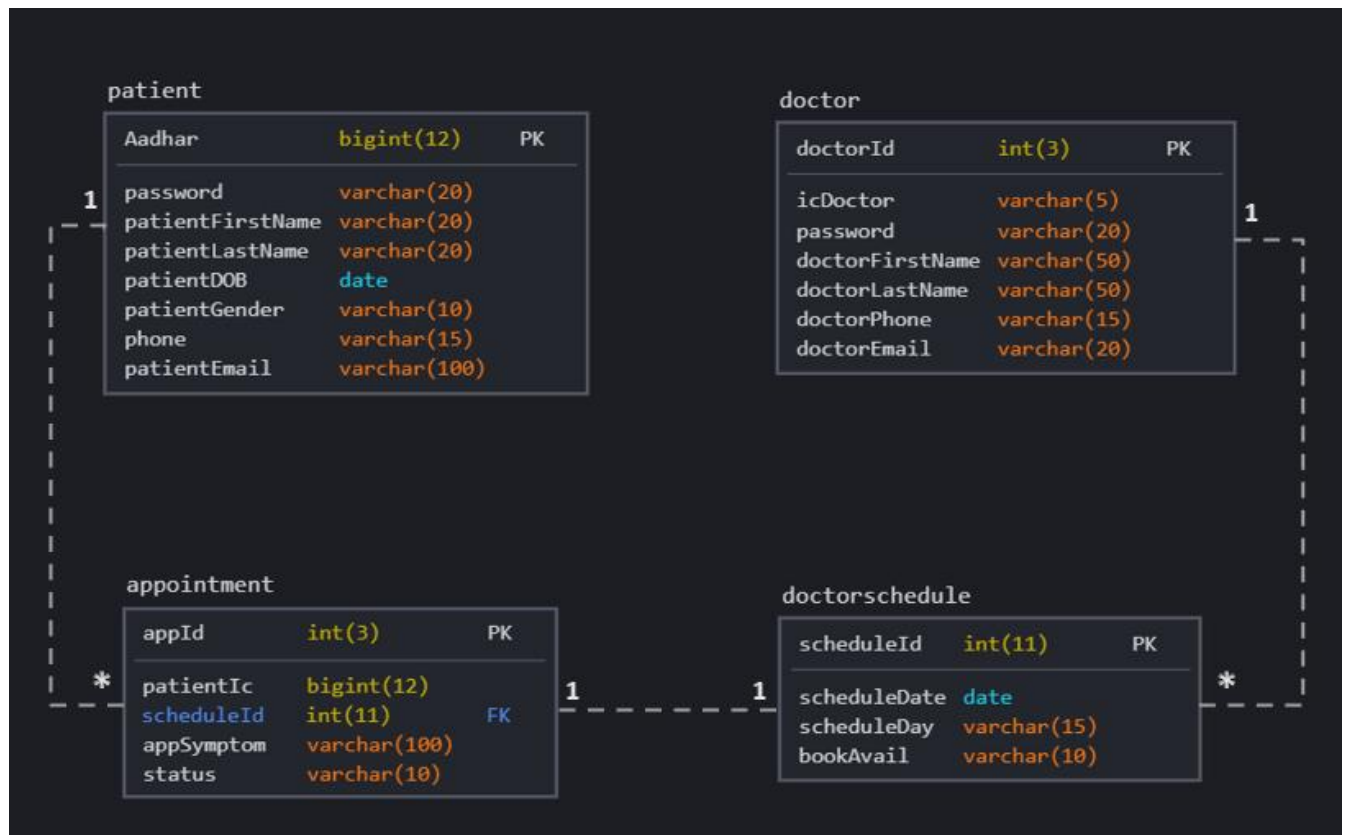
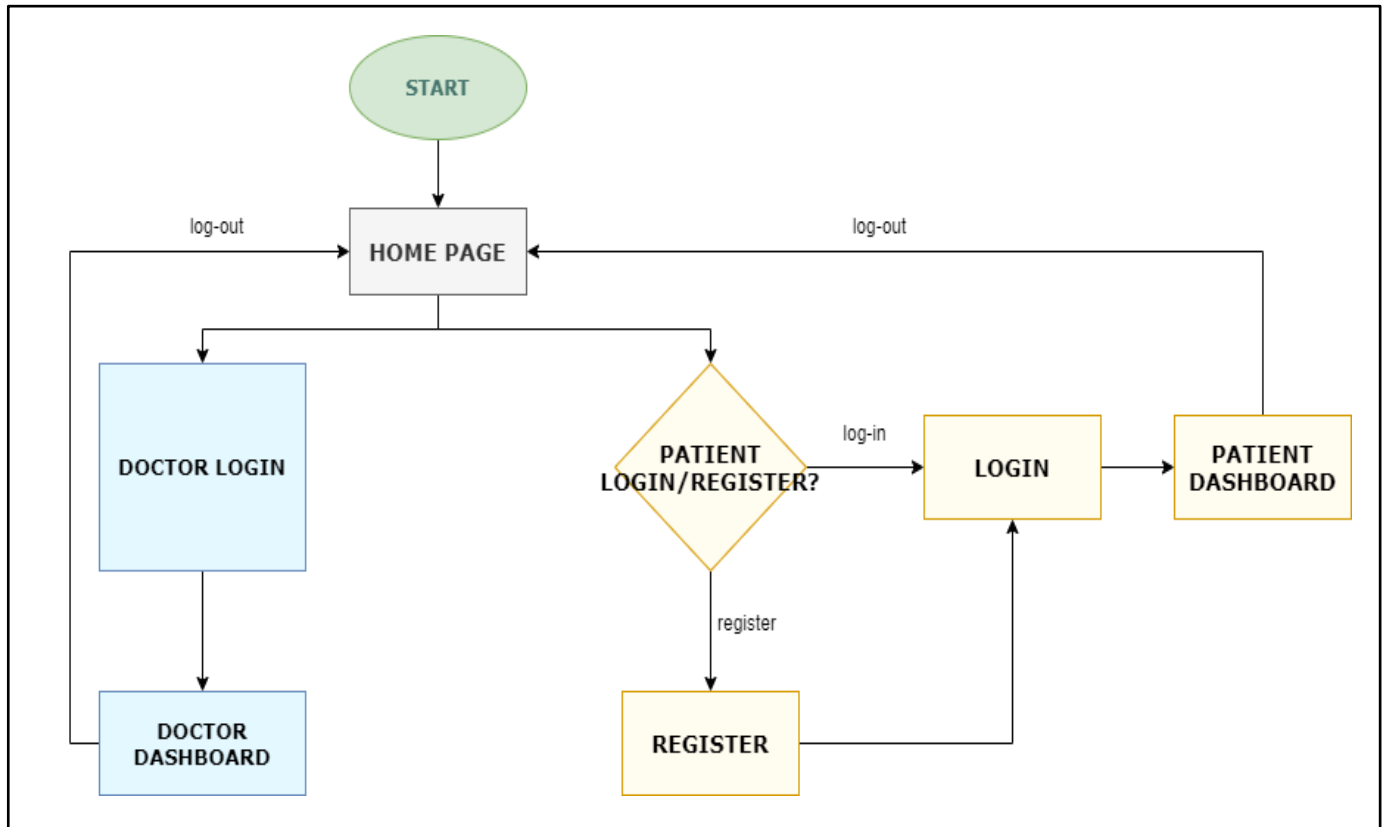


Fig.1: DATABASE SCHEMA DIAGRAM SHOWING THE TABLES IN THE DATABASE



# Design



**Fig.2: FLOWCHART OF THE PROPOSED SYSTEM**

When we open the website, the home page (index.php) is opened. It checks whether anyone is logged-in in its current session. If no one is logged in then there are two options depending on whether it's the user or admin.

The user has two options, he can login if he is already registered. Otherwise, he can register for the first time and then login. That

leads him to the patient dashboard (patient.php). Over there he can enter any particular date of his liking and check availability at the clinic. If there are doctors available then he can book an appointment.

The admin login leads the admin to the dashboard from where the admin can view, doctor's profile, update the schedules and check a patients' appointment status.

The logout option leads both to the homepage.

# ER Diagram

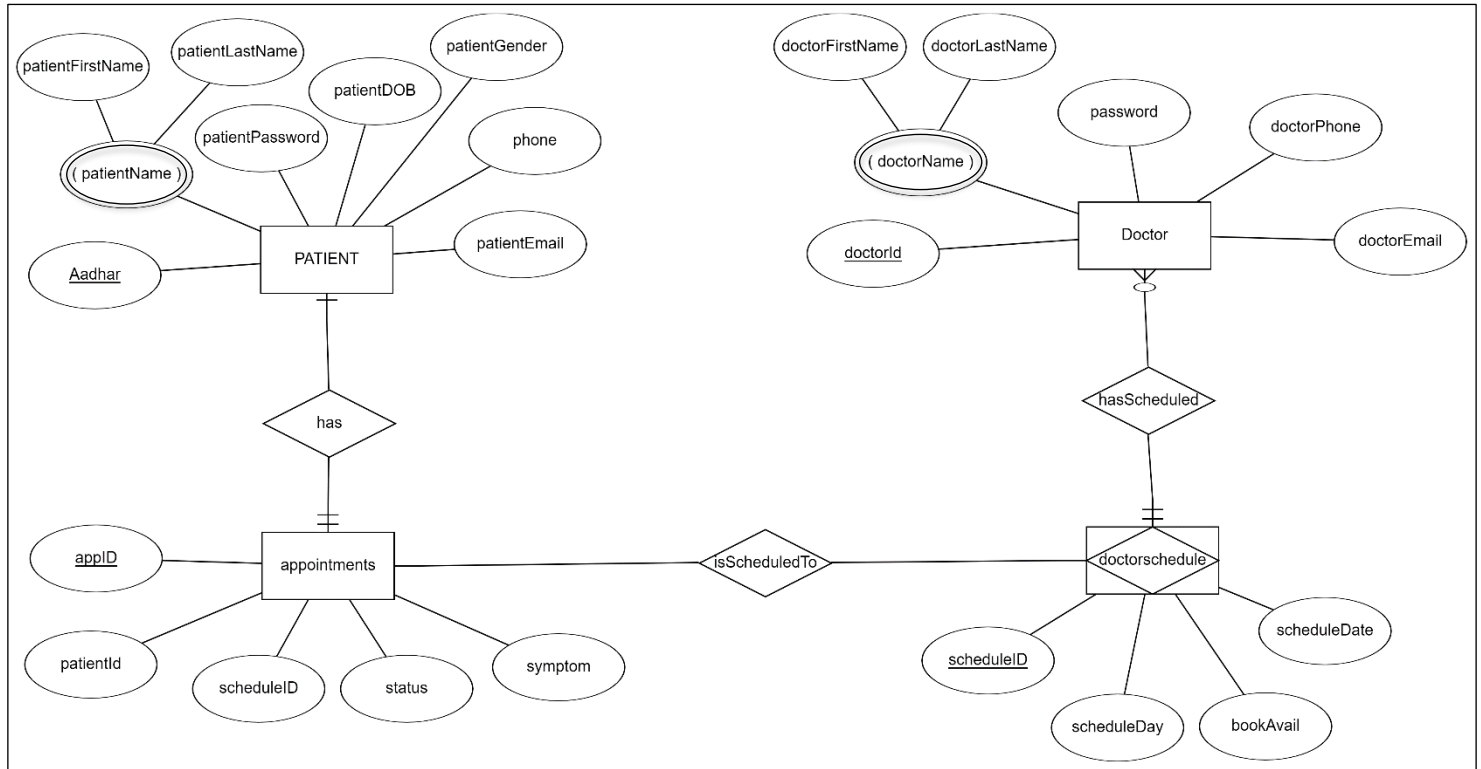


Fig.12: ER DIAGRAM FOR OUR DATABASE db\_healthcare

In the ER diagram above, the rectangular boxes are our entities (tables), i.e., **patient**, **doctor**, **appointment** and **doctorschedule**, and their attributes are represented by ellipse. Relationship between the entities is represented by the diamonds.

The primary keys for each entity are denoted by underlined attribute, e.g., **Aadhar** is the primary key for **patient** entity. Double ellipse are the composite attributes as these are made of more than one simple attribute, e.g., **patientName** is made of **patientFirstName** and **patientLastName**.

# Implementation

The front end was written with HTML and JavaScript and styled with bootstrap and CSS. The backend was coded with PHP and MySQLi was used as the driver for MySQL database.

The server was hosted using XAMPP. XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache. We use XAMPP for a local SQL and Apache server. Our database is hosted on that server.

## Database

The name of our database is: db\_healthcare

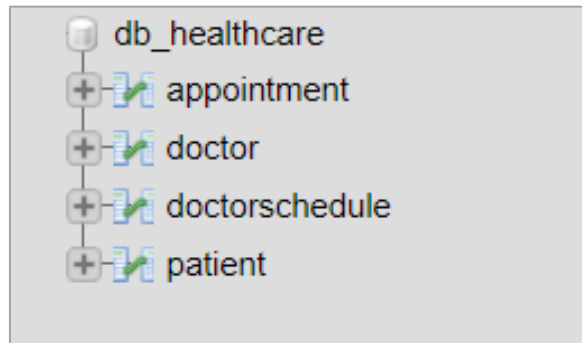


Fig.3: DATABASE db\_healthcare STRUCTURE

## Tables

Our database holds 4 tables, namely:

- patient
- doctor
- appointment
- doctorschedule

These tables were inserted into the database using the CREATE TABLE command.

```
CREATE TABLE `patient` (  
  `Aadhar` bigint(12) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  `patientFirstName` varchar(20) NOT NULL,  
  `patientLastName` varchar(20) NOT NULL,  
  `patientDOB` date NOT NULL,  
  `patientGender` varchar(10) NOT NULL,  
  `phone` varchar(15) NOT NULL,  
  `patientEmail` varchar(100) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `patient`  
  ADD PRIMARY KEY (`Aadhar`);|
```

Fig.4: CREATING TABLE patient

```
CREATE TABLE `doctor` (  
  `icDoctor` varchar(5) NOT NULL,  
  `password` varchar(20) NOT NULL,  
  `doctorId` int(3) NOT NULL,  
  `doctorFirstName` varchar(50) NOT NULL,  
  `doctorLastName` varchar(50) NOT NULL,  
  `doctorPhone` varchar(15) NOT NULL,  
  `doctorEmail` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `patient`  
  ADD PRIMARY KEY (`doctorId`);
```

Fig.5: CREATING TABLE doctor

```

CREATE TABLE `doctorschedule` (
  `scheduleId` int(11) NOT NULL,
  `scheduleDate` date NOT NULL,
  `scheduleDay` varchar(15) NOT NULL,
  `bookAvail` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `doctorschedule`
  ADD PRIMARY KEY (`scheduleId`);

```

Fig.6: CREATING TABLE doctorschedule

```

CREATE TABLE `appointment` (
  `appId` int(3) NOT NULL,
  `patientId` bigint(12) NOT NULL,
  `scheduleId` int(10) NOT NULL,
  `appSymptom` varchar(100) NOT NULL,
  `status` varchar(10) NOT NULL DEFAULT 'process'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `appointment`
  ADD PRIMARY KEY (`appId`);

ALTER TABLE `appointment`
  ADD FOREIGN KEY (`scheduleId`) REFERENCES doctorschedule(`scheduleId`);

```

Fig.7: CREATING TABLE appointment

### Inserting values into the table

When the user registers using the sign-up form or books an appointment from the patient dashboard, the values are entered into the tables **patient** and **appointment** respectively using INSERT INTO <table\_name> VALUES (value1, value2, value3, ...).

```

INSERT INTO `patient` VALUES(3672, 'qwerty', 'Shashank', 'Srivastava', '2011-02-01', 'male', '8901234567', 'shassy@gmail.com');
INSERT INTO `patient` VALUES(2646, 'qwerty', 'Anushka', 'Ray', '1985-03-06', 'female', '9205325712', 'anu@gmail.com');
INSERT INTO `patient` VALUES(5728, 'qwerty', 'Shahbaz', 'Syed', '1999-08-13', 'male', '9830807731', 'ss@gmail.com');
INSERT INTO `patient` VALUES(4512, 'qwerty', 'Khushi', 'Dave', '2000-09-18', 'female', '9876543212', 'khush@gmail.com');

```

**Fig.8: INSERTING INTO TABLE patient**

```

INSERT INTO `appointment` VALUES(87, 2646, 47, 'Fever', 'done');
INSERT INTO `appointment` VALUES(88, 5728, 49, 'Sneeze', 'done');
INSERT INTO `appointment` VALUES(89, 4512, 46, 'Diarrhea', 'process');

```

**Fig.9: INSERTING INTO TABLE appointment**

```

INSERT INTO `doctorschedule` VALUES(46, '2021-05-04', 'Tuesday', 'notavail');
INSERT INTO `doctorschedule` VALUES(47, '2021-05-05', 'Wednesday', 'notavail');
INSERT INTO `doctorschedule` VALUES(48, '2021-05-06', 'Thursday', 'notavail');
INSERT INTO `doctorschedule` VALUES(49, '2021-05-04', 'Tuesday', 'notavail');
INSERT INTO `doctorschedule` VALUES(50, '2021-05-04', 'Tuesday', 'available');
INSERT INTO `doctorschedule` VALUES(51, '2021-05-03', 'Monday', 'available');

```

**Fig.10: INSERTING INTO TABLE doctorschedule**

```

INSERT INTO `doctor` VALUES('D001', '123', 1, 'Tanya', 'Aggarwal', '9876543210', 'ta@medanta.com');
INSERT INTO `doctor` VALUES('D002', '123', 2, 'Anusri', 'Patti', '9876543210', 'pa@medanta.com');
INSERT INTO `doctor` VALUES('D003', '123', 3, 'Sujoy', 'Baitalik', '8976543210', 'sb@medanta.com');

```

**Fig.11: INSERTING INTO TABLE doctor**

# Implementation (screenshots)

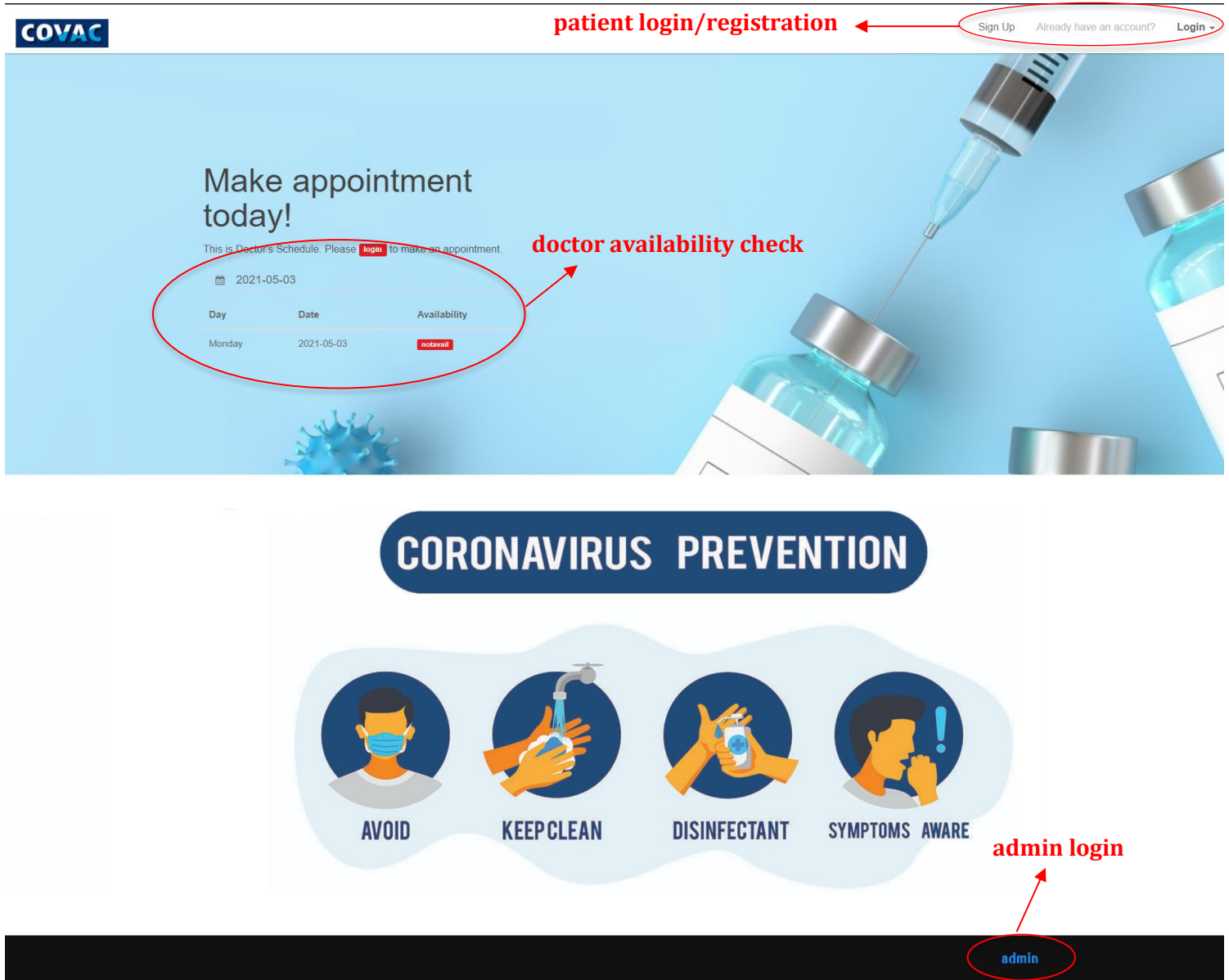
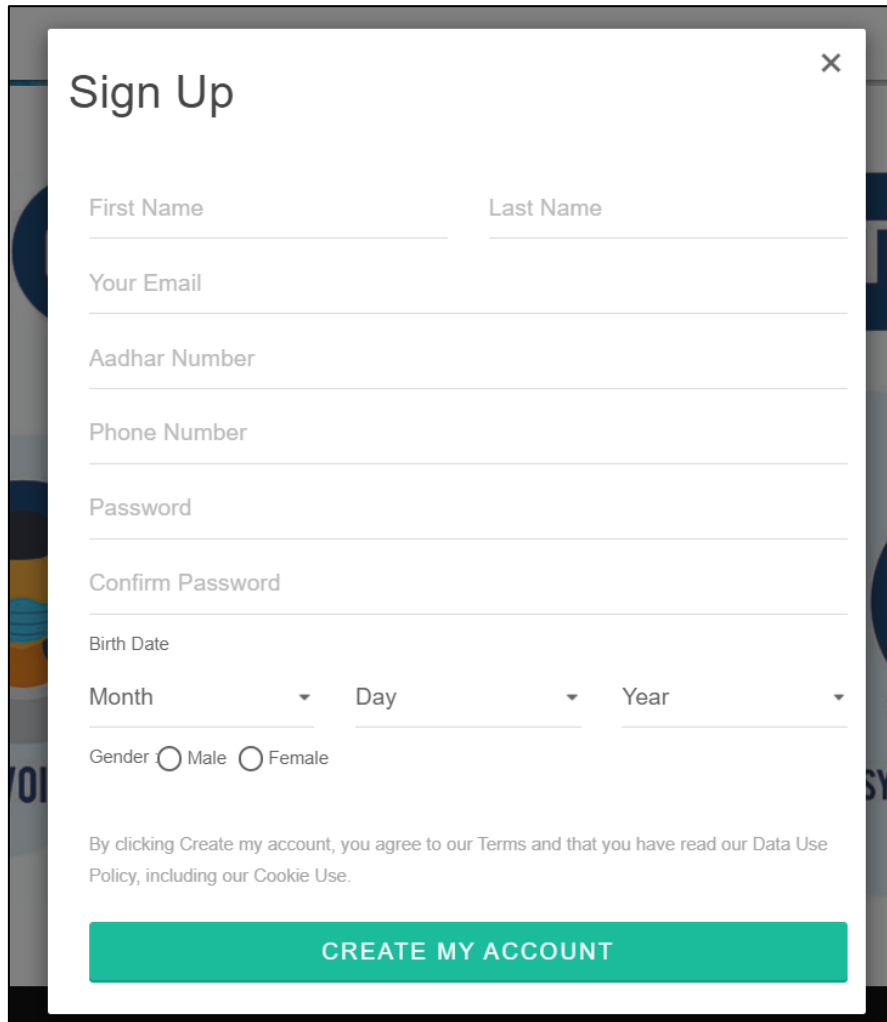


Fig.12: HOMEPAGE index.php





**Sign Up** ×

First Name  Last Name

Your Email

Aadhar Number

Phone Number

Password

Confirm Password

Birth Date

Month  Day  Year

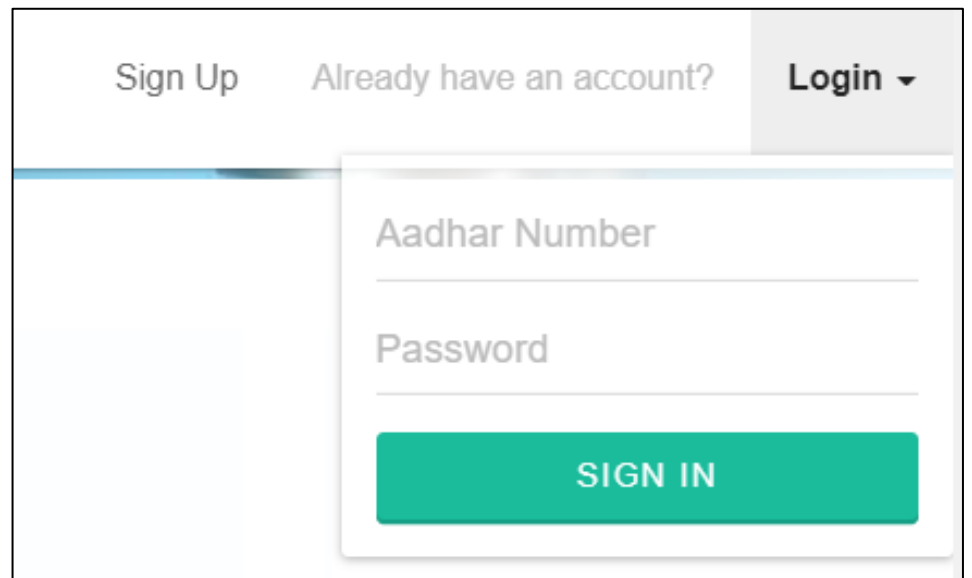
Gender ☐ Male ☐ Female

By clicking Create my account, you agree to our Terms and that you have read our Data Use Policy, including our Cookie Use.

**CREATE MY ACCOUNT**

**Fig.13: PATIENT SIGN-UP/  
REGISTRATION FORM**

**Fig.14: PATIENT SIGN-IN  
DROPDOWN**



Sign Up   Already have an account?   Login ▾

Aadhar Number

Password

**SIGN IN**

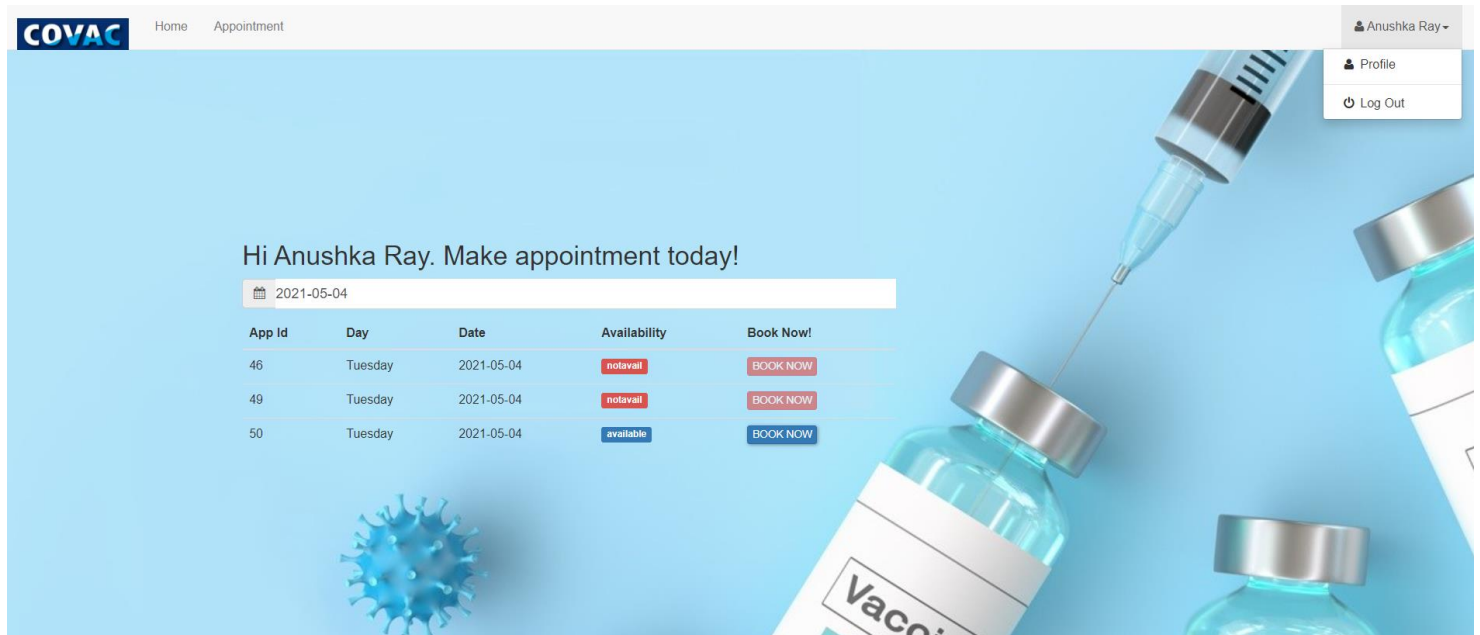


Fig.15: PATIENT DASHBOARD SHOWING DOCTOR AVAILABILITY

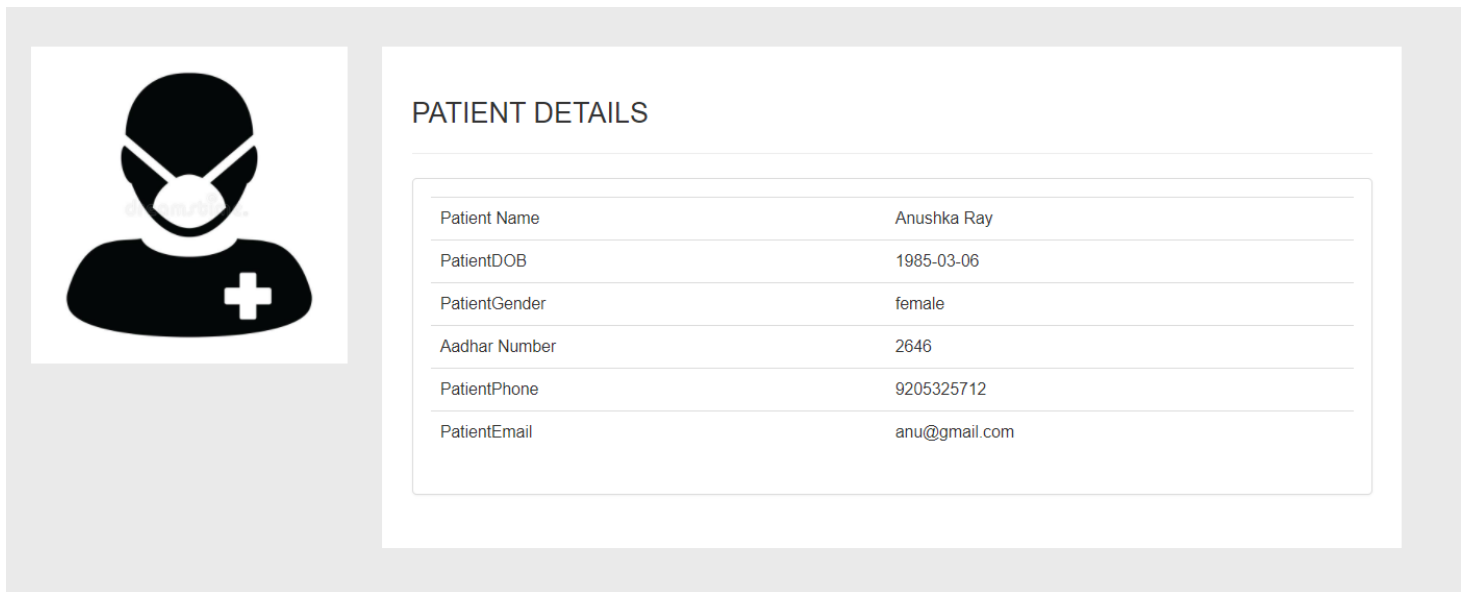



Fig.16: PATIENT PROFILE PAGE



Anushka Ray

Patient Information

Patient Name: Anushka Ray  
 Patient Aadhar: 2646  
 Contact Number: 9205325712

Appointment Information

Day: Tuesday  
 Date: 2021-05-04

Symptom:

Sneeze


Make Appointment

Fig.17: APPOINTMENT BOOKING PAGE

## List of Appointments

List of Appointment					
Appointment Id	Aadhar	First Name	First Name	scheduleDay	scheduleDate
87	2646	Anushka	Ray	Wednesday	2021-05-05
90	2646	Anushka	Ray	Monday	2021-05-03
91	2646	Anushka	Ray	Tuesday	2021-05-04

Fig.18: LIST OF APPOINTMENTS PAGE



Doctor ID

Password

Login

Fig.19: ADMIN SIGN-IN

Welcome Dr Tanya Aggarwal

Tanya Aggarwal

Profile

Log Out

Dashboard

Doctor Schedule

Patient List

Dashboard

Appointment List

Patient First Name	Patient Last Name	Contact No.	Email	Day	Date	Status	Complete
Anushka	Ray	9205325712	anu@gmail.com	Tuesday	2021-05-04	process	<input type="checkbox"/>
Anushka	Ray	9205325712	anu@gmail.com	Monday	2021-05-03	process	<input type="checkbox"/>
Khushi	Dave	9876543212	khush@gmail.com	Tuesday	2021-05-04	process	<input type="checkbox"/>
Shahbaz	Syed	9830807731	ss@gmail.com	Tuesday	2021-05-04	done	<input type="checkbox"/>
Anushka	Ray	9205325712	anu@gmail.com	Wednesday	2021-05-05	done	<input type="checkbox"/>

UPDATE

Fig.20: ADMIN DASHBOARD

## Doctor Schedule

Add Schedule

Date \*

Day \*

Monday

Availability \*

available

Submit

List of Patients

scheduleId	scheduleDate	scheduleDay	bookAvail	
46	2021-05-04	Tuesday	notavail	<input type="checkbox"/>
47	2021-05-05	Wednesday	notavail	<input type="checkbox"/>
48	2021-05-06	Thursday	notavail	<input type="checkbox"/>
49	2021-05-04	Tuesday	notavail	<input type="checkbox"/>
50	2021-05-04	Tuesday	notavail	<input type="checkbox"/>
51	2021-05-03	Monday	notavail	<input type="checkbox"/>

UPDATE

Fig.21: SCHEDULE UPDATE PAGE

## Patient List

List of Patients						
Aadhar Number	First Name	Last Name	ContactNo.	Email	Gender	Birthdate
2646	Anushka	Ray	9205325712	anu@gmail.com	female	1985-03-06
3672	Shashank	Srivastava	8901234567	shassy@gmail.com	male	2011-02-01
4512	Khushi	Dave	9876543212	khush@gmail.com	female	2000-09-18
5728	Shahbaz	Syed	9830807731	ss@gmail.com	male	1999-08-13

Fig.22: PATIENT LIST PAGE



Tanya Aggarwal  
Doctor

## Tanya Aggarwal

IC Number	D001
Contact Number	9876543210
Email	ta@medanta.com

Fig.23: DOCTOR PROFILE PAGE

# Conclusion

We have implemented a website that helps patients book appointments without any hassle in this emergency situation. Patients can register themselves with their information. They can login and book appointments according to their convenience. The administration can set up the schedules of the doctors as well.

The clinic stores the data of all the users. This was achieved with the help of MySQL database and PHP as the backend. The database stores the info of patients, doctors, schedules and appointments. An efficient website is when the frontend and the backend work in cohesion. Further work would be to ensure that there are multiple backups running to increase reliability. We also have to make sure that our system is scalable depending on the demand.

In this pandemic our best bet is to ensure a smooth running of our healthcare facilities and our vaccination drives. As social distancing is the norm, it is preferable to have online booking of appointments. Hence our project ensures that the citizens can easily book their appointments from the comfort of their homes.

# References

- [1] Welling, Luke, and Laura Thomson. *PHP and MySQL Web development*. Sams Publishing, 2003.
- [2] Nixon, Robin. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*. "O'Reilly Media, Inc.", 2014.
- [3] Valade, Janet. *PHP and MySQL for Dummies*. John Wiley & Sons, 2006.
- [4] Trichkova, Elisaveta. "Application of PHP and MySQL for search and retrieval Web services in Web information systems." *Proceedings of First International Conference on Information Systems & Datagrids, Sofia, Bulgaria*. 2005.
- [5] [https://www.w3schools.com/php/php\\_mysql\\_connect.asp](https://www.w3schools.com/php/php_mysql_connect.asp)
- [6] <https://www.php.net/manual/en/set.mysqlinfo.php>
- [7] [https://www.tutorialspoint.com/php/mysql\\_select\\_php.htm](https://www.tutorialspoint.com/php/mysql_select_php.htm)
- [8] <https://www.tutorialrepublic.com/php-tutorial/>
- [9] <https://getbootstrap.com/>
- [10] <https://stackoverflow.com/>