

## **ABSTRACT**

The purpose of this project is to create a website which functions as a doctor appointment dashboard. The need for this project is to help all the people by booking their own appointments through a self-service tool. This website makes the entire business more efficient and ensure fewer scheduling errors and miscommunication.

The main objective of the project on Doctor Appointment System is to manage the details of Doctor, Appointment, Patient, Booking, Doctor Schedule.

The research problem that motivated throughout the project was the need to help the society and improving technology based ways to interact with people.

The main concern about the project was to design a creative, attractive and user-friendly website.

This project is completed with the aid of php and Visual Studio Code.

The execution of the project was a success, and the results were favourable to the original problem being addressed.

## OBJECTIVE

The ultimate aim of this project was to create a Doctor Appointment dashboard which should be easily accessed by all the people. This system improves patient satisfaction. Patients no longer have to be worried about the wait times in clinic. They can plan their daily schedule better. They can spend less time in waiting to meet the doctor.

The benefits of creating a doctor appointment dashboard:

### **1. 24\*7 booking: Anytime Anywhere**

A manual system to manage appointments necessarily requires the availability of a receptionist or office staff. A doctor appointment booking system, allows patients to book an appointment from the comfort of their homes, using their computer, laptop or mobile, and at any time. No matter where they are, they can contact doctors of their choice in any location. Even when they are traveling they can consult with their doctor. Doctors can also adopt technology in the form of teleconsulting to reach out to patients in remote corners.

### **2. Organise your schedule better**

Imagine a situation where multiple appointments are queued up because of manual error. As a doctor, you control the number of people who would be at your clinic at any point of time. This is particularly important given that social distancing is becoming the new normal. With an online appointment booking system, the doctor is in full control and can plan his/her day without any sense of trepidation.

### **3. Minimise patient no-shows**

One of the biggest challenges of a manual or telephone based appointment booking system are patient no-shows. Patients may simply forget their appointment given that there are no automated reminders. A doctor appointment booking system can be configured to send reminders to patients prior to the time of their appointment.

### **4. Making employees and staff more productive**

A doctor appointment booking system is a self-service tool for patients to book appointments. Your staff have their time released from handling phone calls from patients desiring appointments. Instead they can focus on other tasks and improve their productivity.

### **5. Seamlessly integrates with your E.M,R**

The appointment booking system should integrate well with your E.M.R. Doctors should be able to see the list of appointments at any time in addition to knowing the status of the appointment in terms of whether the patient has arrived, whether her vitals have been measured.

### **6. Access appointments on your computer, laptop and mobile phone**

As a doctor, you can access the patient appointment information on any device. You should have the ability to reschedule or cancel appointments in the event of any emergency. The online appointment system that is hosted on the cloud can give you all the appointment details at any day in the past, today or in the future.

# INTRODUCTION

## WEB DEVELOPMENT

Web development refers to the creating, building, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites.

The word Web Development is made up of two words, that is:

- Web: It refers to websites, web pages or anything that works over the internet.
- Development: It refers to building the application from scratch.

Web Development can be classified into two ways:

1. Frontend Development
2. Backend Development





## Frontend Development

The part of a website where the user interacts directly is termed as front end. It is also referred to as the 'client side' of the application.

HTML: HTML stands for HyperText Markup Language. It is used to design the front end portion of web pages using markup language. It acts as a skeleton for a website since it is used to make the structure of a website.

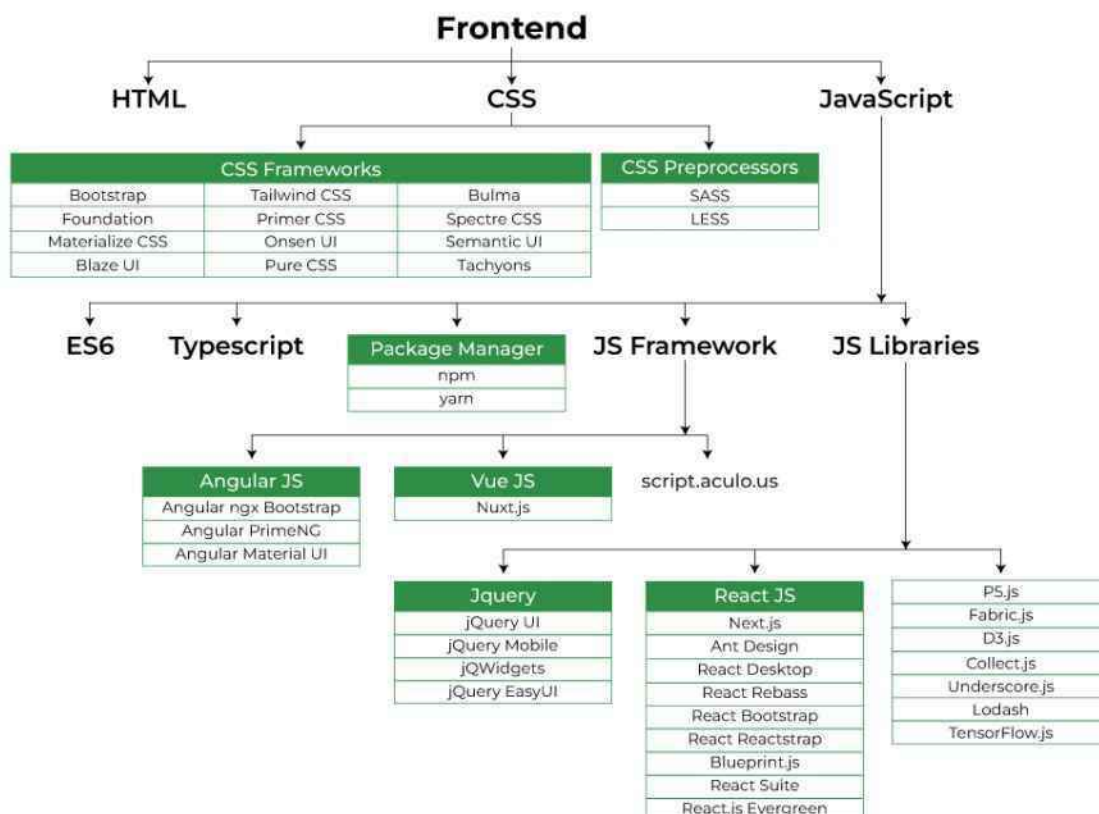
CSS: Cascading Style Sheets fondly referred to as CSS is a simply designed language intended to simplify the process of making web pages presentable. It is used to style our website.

JavaScript: JavaScript is a scripting language used to provide a dynamic behavior to our website.

Bootstrap: Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular CSS framework for developing responsive, mobile-first websites. Nowadays, the websites are perfect for all browsers (IE, Firefox, and Chrome) and for all sizes of screens (Desktop, Tablets, Phablets, and Phones).

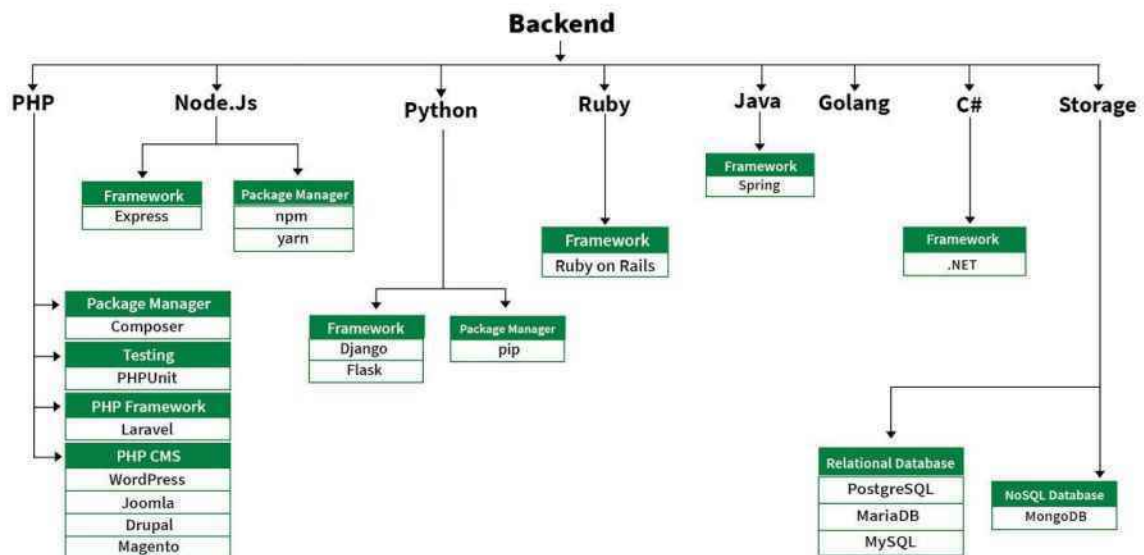
### Bootstrap 4

### Bootstrap 5



## Backend Development

Backend is the server side of a website. It is part of the website that users cannot see and interact with. It is the portion of software that does not come in direct contact with the users. It is used to store and arrange data.



## PHP

The term PHP is an acronym for Hypertext Preprocessor. It is an open-source, interpreted, object-oriented server-side scripting language that is used for web development. PHP is developed by the Rasmus Lerdorf in 1994 with the very first version of PHP that simply designed to set the Common Gateway Interface (CGI) binaries, which are written in C programming language. The latest version of PHP is PHP versions 8 which is released on November 24, 2022. It can be easily embedded with HTML files. HTML codes can also be written in a PHP file. The PHP codes are executed on the server-side whereas HTML codes are directly executed on the browser.

The term PHP is an acronym for PHP: Hypertext Preprocessor. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension “.php”.

Rasmus Lerdorf inspired the first version of PHP and participated in the later versions. It is an interpreted language and it does not require a compiler. PHP code is executed in the server.

It can be integrated with many databases such as Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, and Informix.

It is powerful to hold a content management system like WordPress and can be used to control user access.

It supports main protocols like HTTP Basic, HTTP Digest, IMAP, FTP, and others.

Websites like [www.facebook.com](http://www.facebook.com) and [www.yahoo.com](http://www.yahoo.com) are also built on PHP.

One of the main reasons behind this is that PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file.

The thing that differentiates PHP from the client-side language like HTML is, that PHP codes are executed on the server whereas HTML codes are directly rendered on the browser. PHP codes are first executed on the server and then the result is returned to the browser.

The only information that the client or browser knows is the result returned after executing the PHP script on the server and not the actual PHP codes present in the PHP file. Also, PHP files can support other client-side scripting languages like CSS and JavaScript.

**Other characteristics of PHP are as follows.**

- Simple and fast
- Efficient
- Secured
- Flexible
- Cross-platform, it works with major operating systems like Windows, Linux, and macOS.
- Open Source
- Powerful Library Support
- Database Connectivity

**Syntax:**

```
<?php  
    PHP code goes here  
?>
```

**Example:**

```
<html>  
<head>  
    <title>PHP Example</title>  
</head>  
<body>  
    <?php echo "Hello, World! This is PHP code";?>  
</body>  
</html>
```

**Output:**

Hello, World! This is PHP code

**Need of PHP:**

PHP can actually do anything related to server-side scripting or more popularly known as the backend of a website. For example, PHP can receive data from forms, generate dynamic page content, can work with databases, create sessions, send and receive cookies, send emails, etc. There are also many hash functions available in PHP to encrypt users' data which makes PHP secure and reliable to be used as a server-side scripting language. So these are some of PHP's abilities that make it suitable to be used as a server-side scripting language.

**Disadvantages of PHP:**

- PHP is not secure as it is open source.
- Not good to create desktop applications.
- Not suitable for large Web Applications- Php code is hard to maintain since it is not very modular.
- Modification Problem – PHP does not allow the change in the core behavior of the web applications.



## COMPOSER

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

Composer is **not** a package manager in the same sense as Yum or Apt are. Yes, it deals with "packages" or libraries, but it manages them on a per-project basis, installing them in a directory (e.g. `vendor`) inside your project. By default, it does not install anything globally. Thus, it is a dependency manager. It does however support a "global" project for convenience via the global command.

This idea is not new and Composer is strongly inspired by node's npm and ruby's bundler.

Suppose:

1. You have a project that depends on a number of libraries.
2. Some of those libraries depend on other libraries.

Composer:

1. Enables you to declare the libraries you depend on.
2. Finds out which versions of which packages can and need to be installed, and installs them (meaning it downloads them into your project).
3. You can update all your dependencies in one command.

### System Requirements:

Composer in its latest version requires PHP 7.2.5 to run. A long-term-support version (2.2.x) still offers support for PHP 5.3.2+ in case you are stuck with a legacy PHP version. A few sensitive php settings and compile flags are also required, but when using the installer you will be warned about any incompatibilities.

To install packages from sources instead of plain zip archives, you will need git, svn, fossil or hg depending on how the package is version-controlled.

Composer is multi-platform and we strive to make it run equally well on Windows, Linux and macOS.

### Installation-Windows:

This is the easiest way to get Composer set up on your machine.

Download and run Composer-Setup.exe. It will install the latest Composer version and set up your PATH so that you can call `composer` from any directory in your command line.

## DOCTOR APPOINTMENT DASHBOARD

The Online Doctor Appointment Booking System PHP and Mysql is a simple project developed using PHP, JavaScript, and CSS. The project contains an admin and user sides. The admin side manages all the management like managing schedules, patient records and manage the appointment time, manage treatment facilities, and so on. The admin has an important role in the management of this Online Doctor Appointment Booking System PHP and Mysql.

Front end: HTML, CSS, JavaScript

1. HTML: HTML is used to create and save web document. E.g. Notepad/Notepad++
2. CSS : (Cascading Style Sheets) Create attractive Layout
3. Bootstrap : responsive design mobile freindly site
4. JavaScript: it is a programming language, commonly use with web browsers.

Back end: PHP, MySQL

1. PHP: Hypertext Preprocessor (PHP) is a technology that allows software developers to create dynamically generated web pages, in HTML, XML, or other document types, as per client request. PHP is open source software.
2. MySQL: MySql is a database, widely used for accessing querying, updating, and managing data in databases.

After Starting Apache and MySQL in XAMPP, follow the following steps.

**1st Step:** Extract file

**2nd Step:** Copy the main project folder

**3rd Step:** Paste in xampp/htdocs/

**4th Step:** Open a browser and go to URL “http://localhost/phpmyadmin/”

**5th Step:** Then, click on the databases tab

**6th Step:** Create a database naming “projectmms” and then click on the import tab

**7th Step:** Click on browse file and select “appointment.sql” file which is inside the “database” folder

**8th Step:** Click on go.

**After Creating Database,**

**9th Step:** Open a browser and go to URL “http://localhost/das/”

**For the admin page, go to URL “http://localhost/das/admin/adminlogin.php”**

## METHODOLOGY

The first step in beginning the project was understanding the problem statement. To design a Doctor Appointment Dashboard we should have the knowledge of how a Doctor appointment website looks like. Then, we must understand what are the requirements a patient would look for while using this website. So, firstly I began my project by looking for sample dashboards which would give me an idea of how to design.

The main steps followed during the project was :

Installing xampp

Installing composer

Installing visual studio code

With the help of many websites I got the idea for designing a doctor appointment dashboard.

My overall approach towards the project was to design a simple and elegant dashboard.

### **Steps to setup PHP environment on a local machine:**

There are basically two ways to set up PHP on a local machine which are:

Using all in one package (XAMPP & WAMP). (recommended)

Manually install all the required packages (MySQL, PHP & Apache) and configure them.

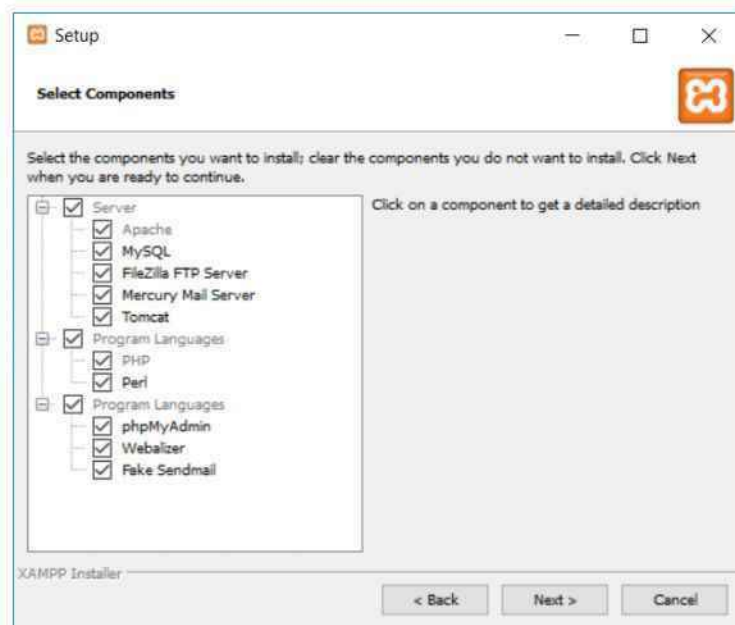
We are going to cover an easy and almost error-free method to install PHP on a local machine which is by using all in one package called XAMPP.

Steps to Installing XAMPP:

Open the downloaded .exe file: After opening the downloaded file you will see a popup from windows, click yes and proceed further, Click on 'next': You'll see a welcome window of XAMPP like below, click next.

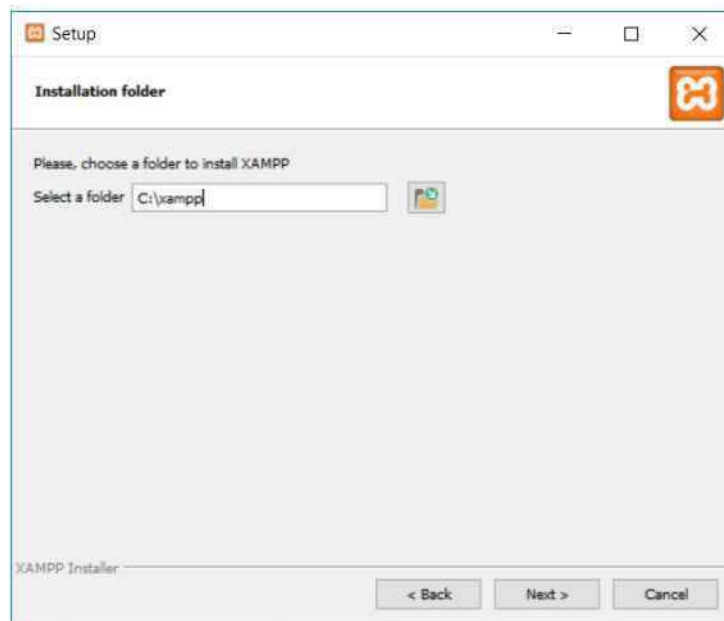


Select the components: Select the components you want to install. Please select the MySQL and phpMyAdmin components.



Select the installation location: Select the location you want to install the XAMPP, the default is C:\xampp.

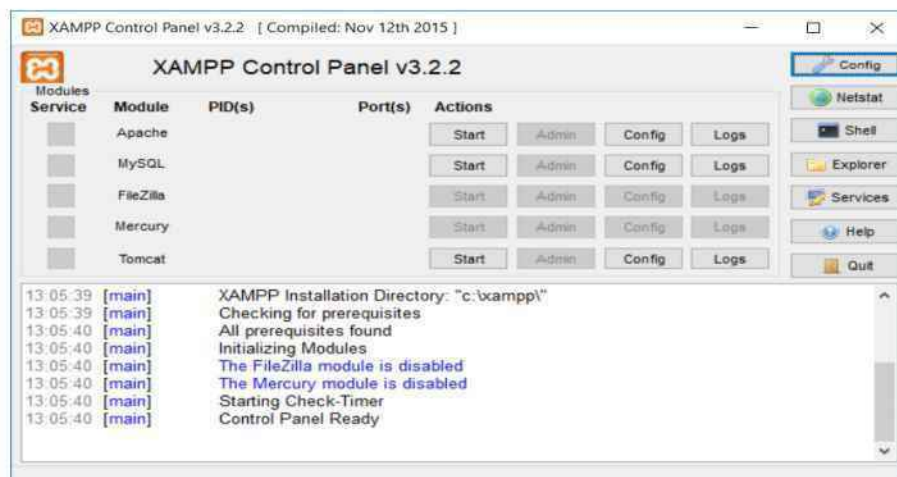




Click next: Your clicking next, your installation will begin.

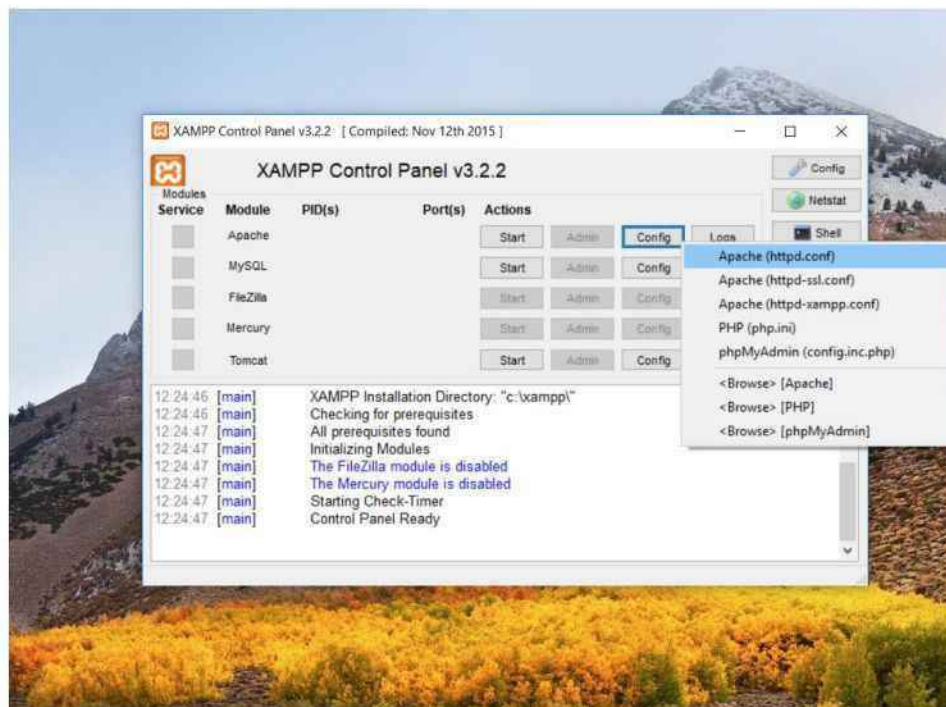


Open XAMPP control panel: After successfully installing the XAMPP in your local machine open the control panel by searching in the windows search bar for 'XAMPP control panel' or by going to the installation directory of XAMPP. You'll see a window like below.

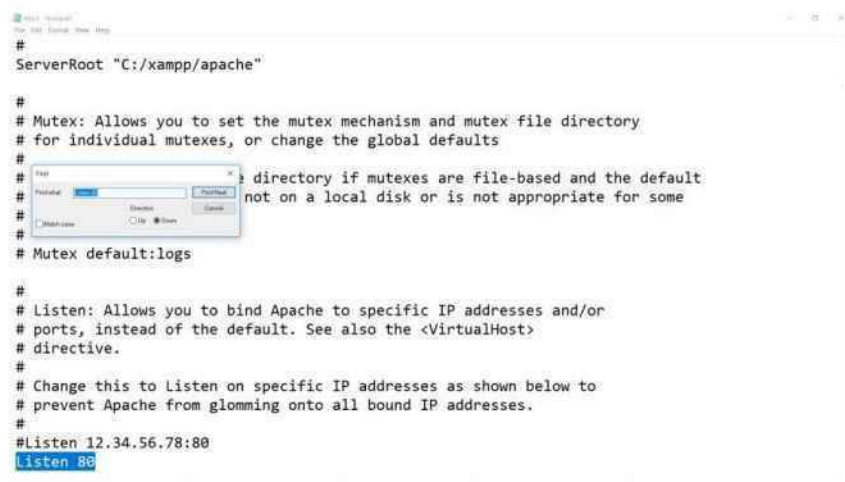


Configuring Apache\*: This step is optional i.e. if you are having some issues on windows 10 related to blocked port.

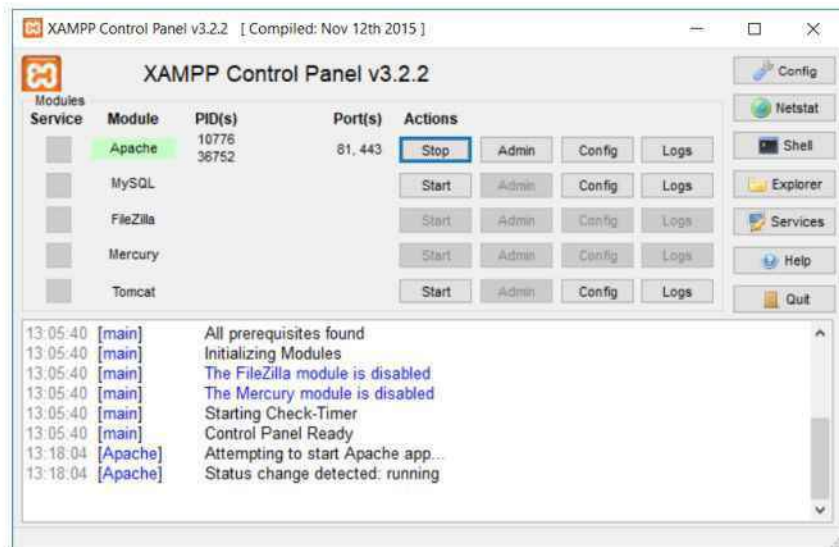
Open the configuration file for Apache as shown in the image below.



Press CTRL + F and search for LISTEN 80. Replace the port 80 with something like 81 and save the file and restart control panel.



Start the Apache server: Start the Apache server by clicking the start button you'll see a port number just in front of the Apache column. You can stop the service whenever you want and start any service by just clicking the start button.



Checking PHP installation: Create a php file in htdocs folder in your installation directory. (C:/XAMPP/htdocs) and add the following code in it.

```
<?php
    echo phpinfo();
?>
```

### To create a php database :

A database consists of one or more tables.

The CREATE DATABASE statement is used to create a database in MySQL.

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

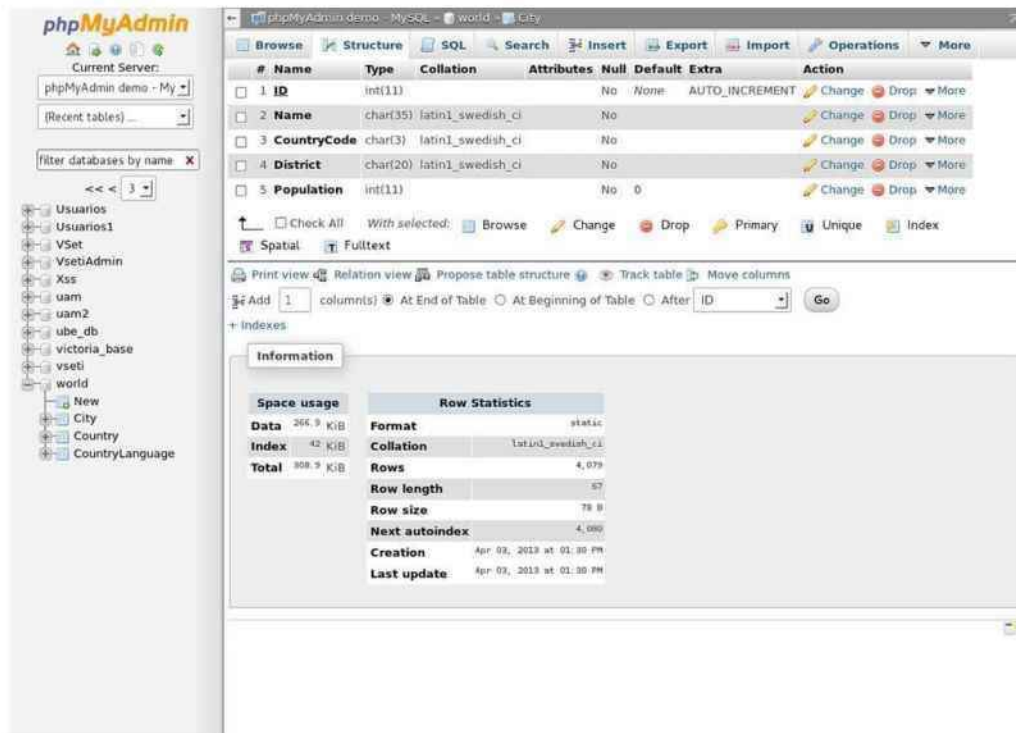
// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
}
```

```

} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>

```



Dr.Name	Contact	Expert at	Patient	PatientContact	Date	Time	Action
Md. Azharul Islam	01521470368	Cardiologist	azad	01521470368	2018-07-27	11.00am	<button>Update</button> <button>Delete</button>
Dr. Majedul Islam	01734761999	Medicine	patient	contact	2018-07-28	03.00pm	<button>Update</button> <button>Delete</button>
Md. Azharul Islam	01521470368	Cardiologist	mamun	01521470368	2018-07-26	11.00am	<button>Update</button> <button>Delete</button>
Dr. Badol Miah	01949389983	Kedney	mamun	01521470368	2018-07-20	11.00am	<button>Update</button> <button>Delete</button>
Dr. Azharul Islam	01764761919	Medicine	azad	01521470368	2018-07-26	03.00pm	<button>Update</button> <button>Delete</button>
Dr. Badol Miah	01949389983	Kedney	dezazad	dezazad	2018-07-07	11.00am	<button>Update</button> <button>Delete</button>
Dr. Rashid	01521670654	Medicine			2018-07-15	11.00am	<button>Update</button> <button>Delete</button>



## CODE

### ➤ home.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta name="keywords" content="Easiest way to book doctor appointment">
  <meta name="description" content="We are best website to provide doctor
appointment for all your monthly checkups">
  <title>Doctor appointment</title>
</head>
<body>

  <!--Navbar-->

  <title>Doctor First</title>
  <nav class="navbar navbar-expand-lg navbar-light fixed-top">
    <div class="container-fluid">
      <a class="navbar-brand" href="home.html">
        
      </a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavAltMarkup"
      aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav">
          <a class="nav-link active" href="sub-
pages/register.html">Register</a>
          <a class="nav-link" href="sub-pages/login.html">Login</a>
          <a class="nav-link" href="sub-pages/about.html">About Us</a>
          <a class="nav-link" href="sub-pages/contact-us.html">Contact
Us</a>
          <a class="nav-link" href="sub-
pages/specialist.html">Specialist</a>
        </div>
      </div>
    </div>
  </nav>
```

```

</nav><br><br>

<!--CSS-->
<div class="section1">
  <h1>#Doctor at your rescue</h1>
</div>

<!--bootstrap CSS-->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">

<!--Customised CSS-->
<link rel="stylesheet" href="style.css">

<!--Flex Box Model-->
<div class="main-section">
  <div class="box">
    <div class="row">
      <div class="box-img">
        
      </div>
    </div>
    <div class="box-content">
      <h3>Paediatrician</h3><br>
      <h4>Dr.Rahul</h4><br>
      <p>World's best child specialist</p><br>
      <a href="sub-pages/book.html" class="btn-style">Book appointment</a>
    </div>
  </div>
  <div class="box">
    <div class="box-img">
      
    </div>
    <div class="box-content">
      <h3>Surgeon</h3><br>
      <h4>Dr.Hina</h4><br>
      <p>World's best surgeon</p><br>
      <a href="sub-pages/book.html" class="btn-style">Book appointment</a>
    </div>
  </div>
  <div class="box">
    <div class="box-img">

```

```

        
    </div>
    <div class="box-content">
        <h3>Gynecologist</h3><br>
        <h4>Dr.Jessica</h4><br>
        <p>World's best Gynecologist</p><br>
        <a href="sub-pages/book.html" class="btn-style">Book appointment</a>
    </div>
</div>
</div>
<div class="mainsection1">
    <div class="box">
        <div class="box-img">
            

        </div>

        <div class="box-content">
            <h3>Cardiologist</h3><br>
            <h4>Dr.Vasanth</h4><br>
            <p>World's best Cardiologist</p><br>
            <a href="sub-pages/book.html" class="btn-style">Book appointment</a>
        </div>
    </div>
    <div class="box">
        <div class="box-img">
            
        </div>
        <div class="box-content">
            <h3>Anesthesiologist</h3><br>
            <h4>Dr.Austin</h4><br>
            <p>World's best Anesthesiologist</p><br>
            <a href="sub-pages/book.html" class="btn-style">Book appointment</a>
        </div>
    </div>
    <div class="box">
        <div class="box-img">
            
        </div>
        <div class="box-content">
            <h3>Pathologist</h3><br>
            <h4>Dr.Annie</h4><br>
            <p>World's best Pathologist</p><br>
            <a href="sub-pages/book.html" class="btn-style">Book appointment</a>

```

```

        </div>
    </div>

    </div>

</body>
</html>

```

## ➤ register.php

```

<?php
if(isset($_REQUEST["name"]))
{

    $name = $_REQUEST["name"];
    $email= $_REQUEST["email"];
    $phone = $_REQUEST["phone"];
    $password = $_REQUEST["password"];
    $password = password_hash($password, PASSWORD_BCRYPT);
    include 'connect.php';

    $sql = "INSERT INTO `clients`(`id`, `name`, `email`,
`phone`, `password`) VALUES
(null,'$name','$email','$phone','$password')";
    mysqli_query($con,$sql) or die(mysqli_error($con));
    header("location:home.php");

}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Register</title>
</head>
<body>

```



```

    <!--bootstrap CSS-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

    <form>
      <div class="container ps-5 pe-5 m-5">
        <h1 class="text-center">Register here!!</h1>
        <div class="row">
          <div class="mb-3">
            <label for="exampleInput1" class="form-label">Name</label>
            <input type="text" class="form-control" id="exampleInput1">

          </div>
          <div class="mb-3">
            <label for="exampleInput2" class="form-label">Contact
Number</label>
            <input type="number" class="form-control" id="exampleInput2">

          </div>
          <div class="mb-3">
            <label for="exampleInputEmail1" class="form-label">Email
address</label>
            <input type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp">
            <div id="emailHelp" class="form-text">We'll never share your
email with anyone else.</div>
          </div>
          <div class="mb-3">
            <label for="exampleInput3" class="form-label">Blood
group</label>
            <input type="text" class="form-control"
id="exampleInput3" >
          </div>
          <div class="mb-3">
            <label for="exampleInput4" class="form-
label">Age</label>
            <input type="number" class="form-control"
id="exampleInput4">

          </div>
          <div class="mb-3">
            <div class="mb-3">

```

```

        <label for="exampleInputPassword1" class="form-
label">Password</label>
        <input type="password" class="form-control"
id="exampleInputPassword1">
    </div>
    <div class="mb-3 form-check">
        <input type="checkbox" class="form-check-input" id="exampleCheck1">
        <label class="form-check-label" for="exampleCheck1">Check me
out</label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
</form>

</body>
</html>

```

## ➤ login.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
</head>
<body>

    <?php include 'nav.php'?>
    <form>
        <div class="container ps-5 pe-5 m-5">
            <h1 class="text-center">Login here!!</h1>
            <div class="row">
                <div class="mb-3">
                    <label for="exampleInputEmail1" class="form-label">Email
address</label>
                    <input type="email" class="form-control" id="exampleInputEmail1"
aria-describedby="emailHelp">
                    <div id="emailHelp" class="form-text">We'll never share your email
with anyone else.</div>
                </div>
                <div class="mb-3">
                    <label for="exampleInputPassword1" class="form-
label">Password</label>
                    <input type="password" class="form-control"
id="exampleInputPassword1">

```

```

        </div>
        <div class="mb-3 form-check">
            <input type="checkbox" class="form-check-input" id="exampleCheck1">
            <label class="form-check-label" for="exampleCheck1">Check me
out</label>
        </div>
        <button type="submit" class="btn btn-primary">Submit</button>
    </form>

</div>
</div>

</body>
</html>

```

### ➤ Connecting a database

**<?php**

**\$con=mysqli\_connect("localhost","root","","loginforms")or**

**Die(mysqli\_connect\_error());**

**?>**

### ➤ Creating a logout page

**<?php**

**session\_start();**

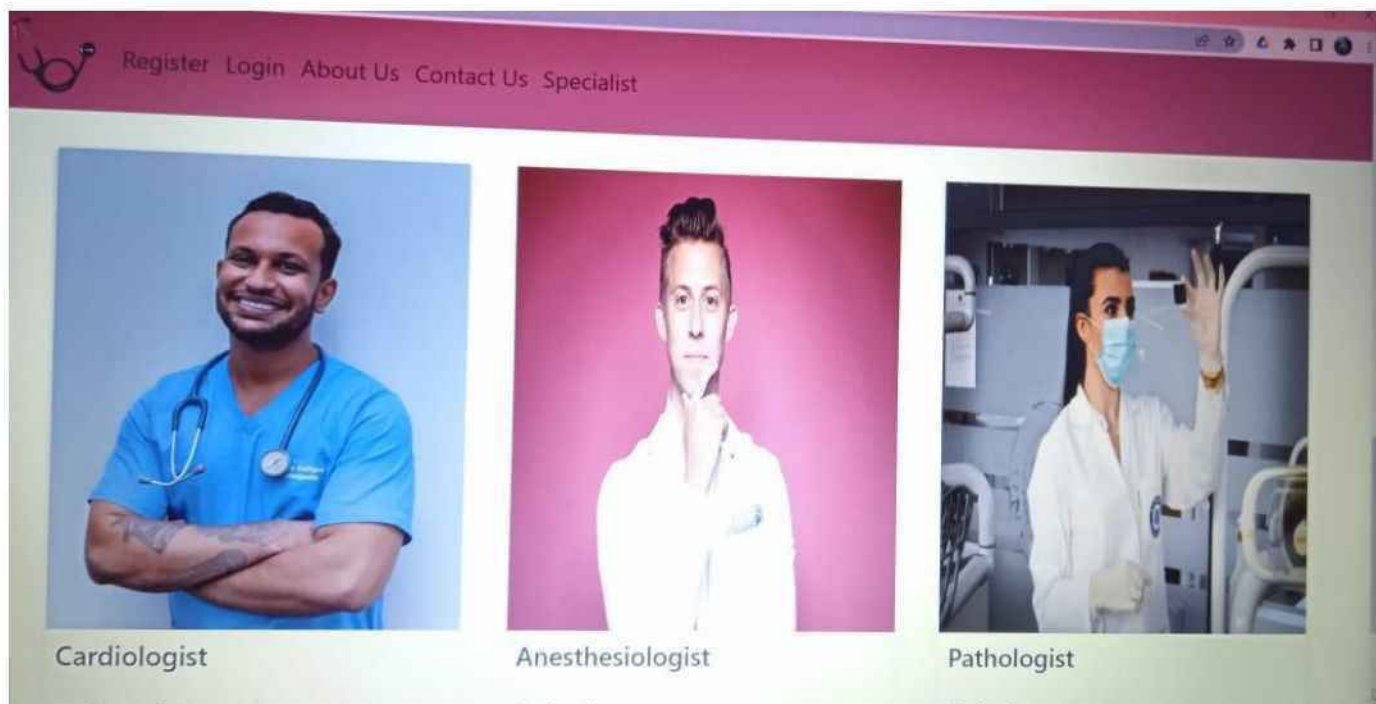
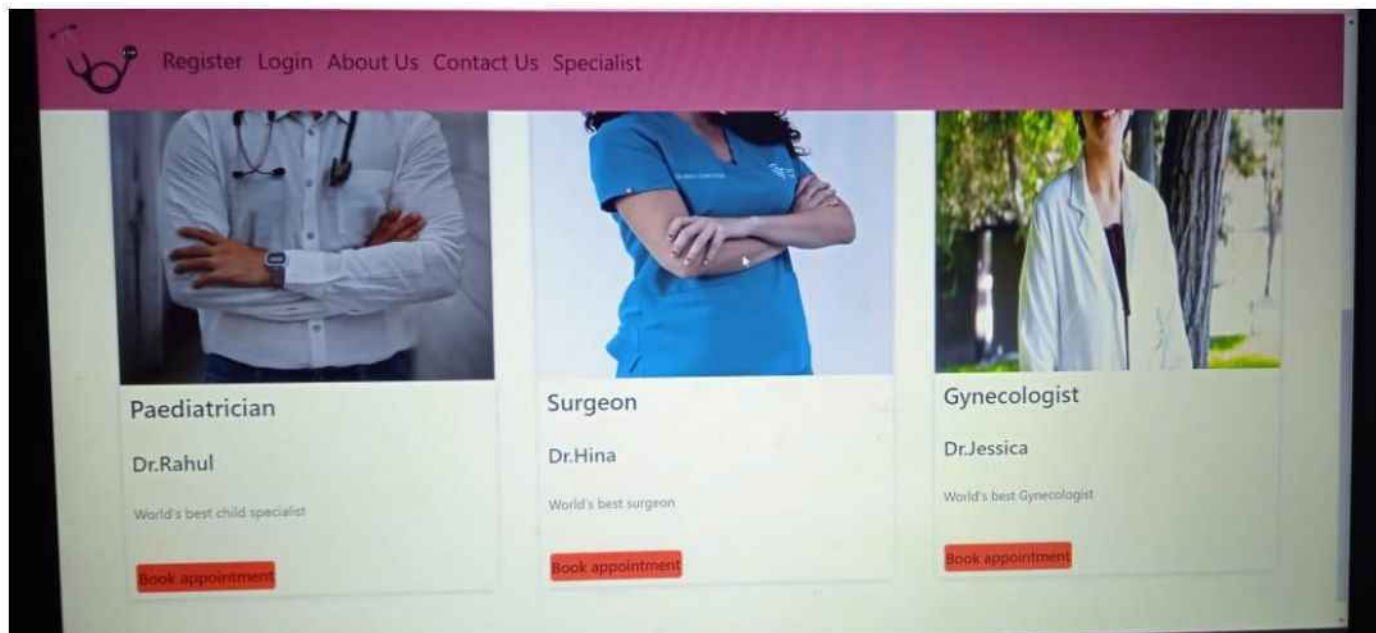
**session\_destroy();**

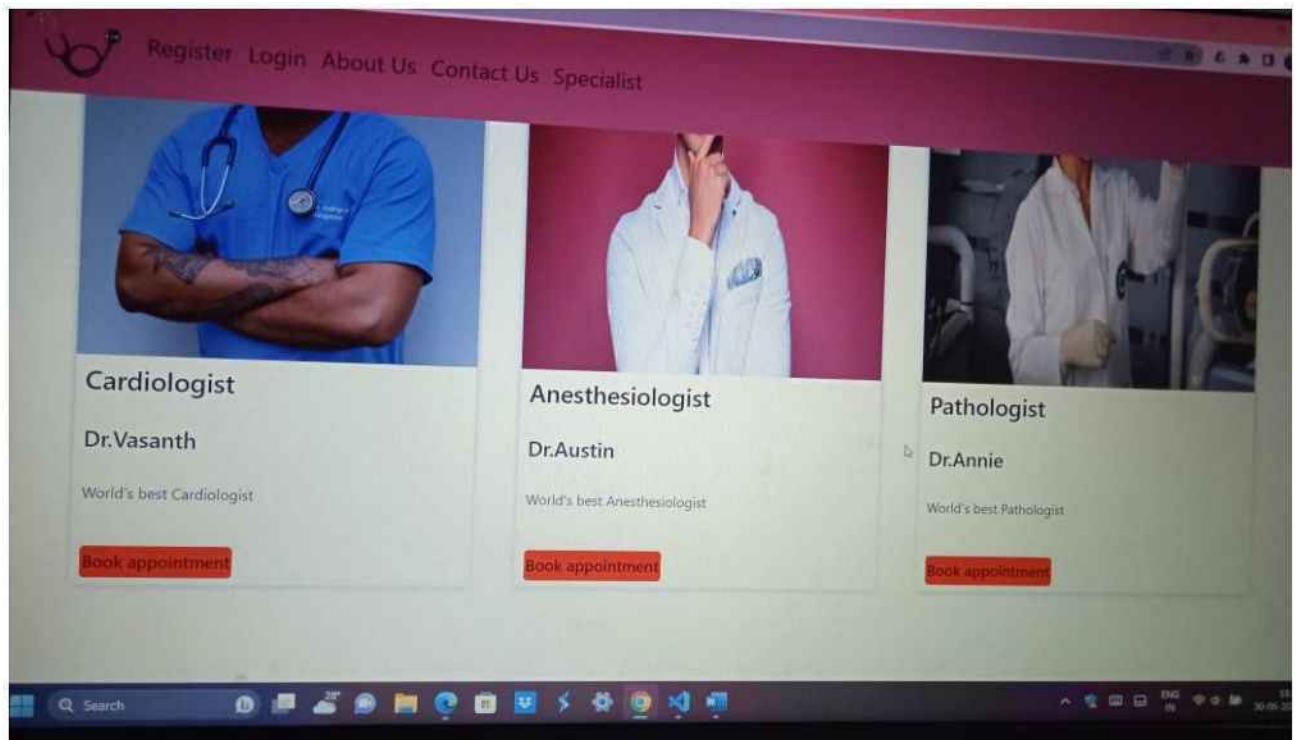
**Header("location:login.php");**

## OUTPUT









The screenshot shows a registration form titled "Register here!!". The form contains several input fields and a submit button. The fields are labeled as follows:

- Name: Ankit
- Contact Number: 2222222222
- Email address: ankit@gmail.com
- Blood group: O+ve
- Age: 15
- Password: .....
- Check me out: ☒

A "Submit" button is located at the bottom of the form. A small note below the email address states: "We'll never share your email with anyone else."

to view my sub pages/login.html

## Login here!!

Email address

anu@gmail.com

We'll never share your email with anyone else.

Password

\*\*\*\*

☒ Check me out

Submit

## Book appointment!!

Name

Ankitha

Category

Bone fracture

Contact Number

2222222

Email address

ankit@gmail.com

We'll never share your email with anyone else.

Blood group

O +ve

Age

15

Appointment ID

1345

Registration Form

Name:

Category:

Contact Number:

Email address:

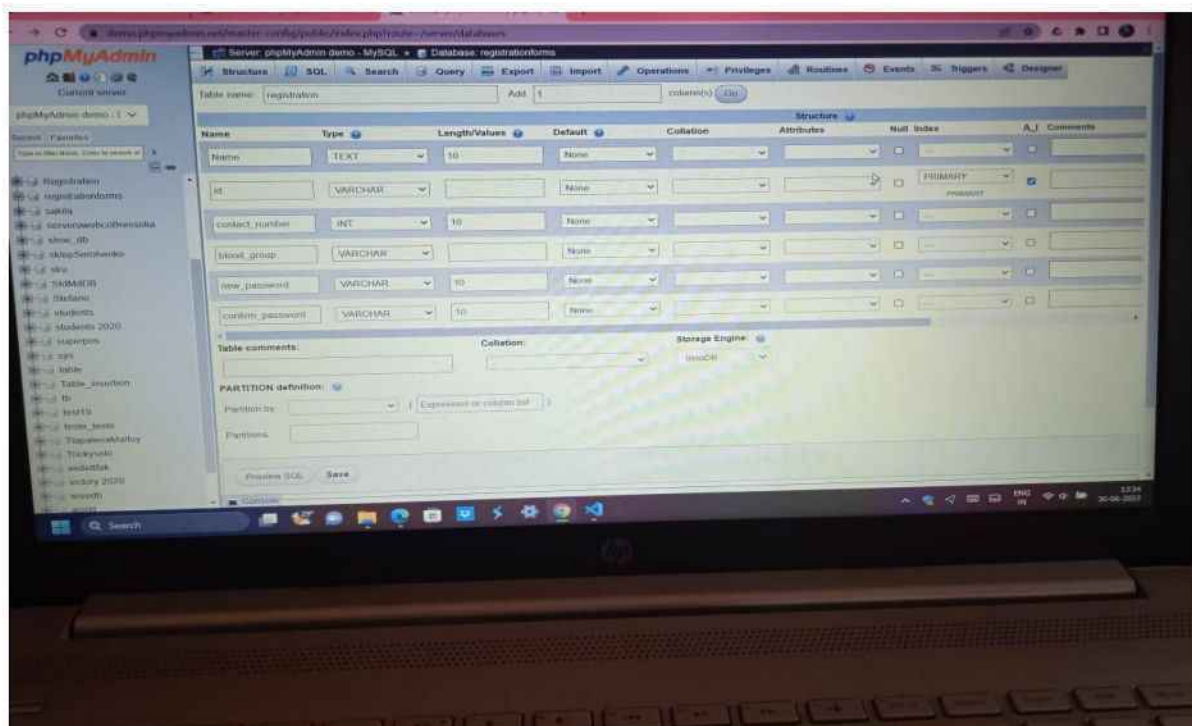
We'll never share your email with anyone else.

Blood group: ☐ O+ve

Age:

Appointment ID:

☒ Check me out





## **CONCLUSION**

**This project was very helpful. I learnt about xampp, composer, php and many more things.**

**I was able to think creatively. Through this project I learned how to create a Doctor Appointment Dashboard.**

**I am thankful for coincent for giving us the knowledge of Web Development. This project gave me the curiosity to create a real-time Websites .**

**The coincent classes gave me the knowledge of html codes. I am thankful for coincent for giving us the opportunity to execute the concepts we learnt by doing this project.**

**Through this project, I realized that websites we develop should be user-friendly and easily accessible to all users.**

**I learnt how technology can be implemented in day to day life activities.**

**Thank you.**