

ONLINE APPOINTMENT SYSTEM FOR HEALTHCARE
A COURSE PROJECT REPORT

By

TANUMAY GHOSH (RA2011027010101)
NAMIT LODH (RA2011027010116)
ABDUL MUTHALLIB (RA2011027010123)

Under the guidance of

Dr. V. Vijayalakshmi

Assistant professor

Department of Data Science and Business Systems

In partial fulfilment for the Course

of

18CSC303J - DATABASE MANAGEMENT SYSTEM

in

Department of Data Science and Business Systems



COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
Kattankulathur, Chengalpattu District

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "Online Appointment System for HealthCare" is the bona fide work of **Tanumay Ghosh (RA2011027010101)**, **Namit Lodh (RA2011027010116)**, **Abdul Muthallib (RA2011027010123)** who carried out project work under my supervision.

M. Lakshmi

Dr. M. Lakshmi
Professor & HOD
Department of DSBS
SRM Institute of Science and
Technology.

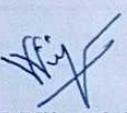

Dr. V. Vijayalakshmi
Assistant professor
Department of Data Science and Business Systems
SRM institute of science and technology

TABLE OF CONTENTS

| CHAPTERS | CONTENTS | PAGE NO. |
|-----------------|--------------------------------------|-----------------|
| 1. | ABSTRACT | 1 |
| 2. | GENERAL AND UNIQUE SERVICES | 2 |
| 3. | PROBLEM STATEMENT | 3 |
| 4. | OBJECTIVES | 4 |
| 5. | LITERATURE SURVEY | 6 |
| 6. | SCOPE OF PROJECT | 7 |
| 7. | PROPOSED ARCHITECTURE DIAGRAM | 9 |
| 8. | USE CASE DIAGRAM | 10 |
| 9. | ER DIAGRAM | 11 |
| 10. | FRONTEND DESIGN | 12 |
| 11. | BACKEND DESIGN | 21 |
| 12. | TYPES OF CONNECTIVITY | 27 |
| 13. | LIST OF MODULES | 29 |
| 14. | LIST OF DATABASE RELATIONS | 30 |
| 15. | CODING | 37 |
| 16. | CONCLUSION | 40 |
| 17. | REFERENCES | 41 |

1. Abstract

The Online Appointment Management system is a computerised management system. This system keeps the records of the appointments which are being registered by the customer and verified by the administrator. This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.

The project “Online Appointment Management System” is aimed to develop to maintain the daily appointments of customers with different doctors and clinics. It helps the customer to think wisely to visit the doctor of his choice.

It enables patients to schedule, modify, or cancel appointments with their healthcare providers. The system is designed to provide a convenient and efficient tool for patients to manage their appointments and improve access to care.

It can also help healthcare organisations reduce no-show rates, increase office efficiency, and enhance patient-provider communication. The system automates administrative tasks, such as appointment scheduling and reminders, which can reduce the workload on staff and improve clinic efficiency.

Furthermore, it can provide valuable data analytics that enable healthcare organisations to optimise their services and improve patient outcomes. However, successful implementation of such a system requires careful planning and consideration of potential challenges, such as security concerns and patient privacy. User-centred design is crucial to ensure that the system is user-friendly and meets the needs of patients and healthcare providers. Overall, an online appointment system for healthcare can provide significant benefits to both patients and healthcare organisations, and it is an important tool for improving access to care and enhancing patient satisfaction.

2. List of General and Unique Services in the database application

General

- Patient registration and profile management and Doctor registration and profile management.
- Appointment scheduling, cancellation and rescheduling.
- Patient medical history management.
- Prescription management.

Unique

- Appointment tracking and analytics for doctors and waitlist management for cancelled appointments.
- Health tracker for patients to monitor and track their health conditions.
- Get to know the doctor details and their availability.
- Referral management for doctors to refer patients to other specialists

3. Problem Statement

In the existing manual system, a lot of time is spent in communicating and sending the information across different branches and their independent website. The current world still works on the traditional, orthodox system of written entries of the registration and manually submitting all the information of booking appointments which is tedious and inefficient. There are more chances for the humans involved in the system to make an error, and this old-fashioned method also takes a long time to execute manually, even after not considering the high chances of mistakes. Also, due to the constant changing of the day to day life, it will lead to proportionate changes to the cost and appointments, which makes updating them dynamically everywhere tough.

All the old techniques and modus-operandi prove to be an inefficient and mammoth task, and we need to overcome this.

There is a need for an integrated automated system, which has some centralised control over the entire process. The Conventional System makes use of huge amounts of paper for recording transactions. The existing system is a manually maintained system. All the appointment records are to be maintained for the details of each customer, doctor's details, clinic details and appointments details etc. All these details are entered and retrieved manually.

The healthcare industry faces several challenges related to the traditional appointment scheduling process, such as long wait times, high no-show rates, and inefficient administrative tasks. Patients may experience difficulty in finding available appointments that fit their schedule, and healthcare providers may face a high workload in managing appointments manually. These challenges can lead to patient dissatisfaction, reduced clinic efficiency, and decreased access to care.

Additionally, the COVID-19 pandemic has highlighted the need for virtual care options, making online appointment systems an even more critical solution for healthcare organisations.

Therefore, the problem statement for an online appointment system for healthcare is to address these challenges by providing patients with a user-friendly tool for scheduling, modifying, or cancelling appointments with their healthcare providers, and enabling healthcare providers to streamline their administrative tasks, reduce no-show rates, and enhance patient-provider communication. The system must be secure, reliable, and compliant with healthcare regulations to ensure patient privacy and safety.

4. Objectives

Proposed system is a computerised version of the existing system which provides easy and quick access over the data keeping records of the land details so that the verification and registering the details is effective and efficient. Storing customer details and appointment details properly Maintains accuracy of data and reduces error. The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in a ‘Online Appointment Management System’, but utmost care has been taken to make the system efficient and user friendly.

This allows the user to book an appointment at his comfort and check for the availability of the appointment according to his/her comfort. So this project will help people not stand in queues and wait for a turn but go at the right turn with minimal waiting time.

The objective of an online appointment system for healthcare is to provide a convenient and efficient tool for patients to schedule, modify, or cancel appointments with their healthcare providers. The system should aim to improve patient satisfaction, reduce no-show rates, and increase office efficiency by automating the appointment scheduling process and reducing administrative costs. Other objectives may include:

- Enhancing accessibility: An online appointment system can improve accessibility by allowing patients to schedule appointments from anywhere and at any time.
- Improve patient satisfaction: By allowing patients to schedule appointments online, they can easily find available times that fit their schedule, which can lead to increased satisfaction and better patient experience.
- Improving patient-provider communication: The system can facilitate communication between patients and healthcare providers, such as sending appointment reminders, providing access to medical records, and enabling secure messaging.
- Streamlining administrative tasks: The system can automate administrative tasks, such as appointment scheduling, rescheduling, and cancellation, reducing the workload of healthcare staff.
- Increasing patient engagement: An online appointment system can encourage patients to be more engaged in their healthcare by allowing them to manage their appointments and medical information online.

- Enhancing data analytics: The system can provide healthcare organisations with valuable data on patient appointments, preferences, and behaviour, enabling them to optimise their services and improve patient outcomes.

Overall, the objective of an online appointment system for healthcare is to provide patients and healthcare providers with a user-friendly and efficient tool that improves access to care, enhances patient satisfaction, and reduces administrative burden.

5. Literature Survey

An online appointment system for healthcare is a web-based application that allows patients to schedule, modify, or cancel appointments with their healthcare providers. It is a convenient and efficient tool that helps both patients and healthcare providers save time and reduce administrative costs. In this literature survey, we will explore the existing research and literature on online appointment systems for healthcare.

- "Evaluation of an Online Appointment Scheduling and Reminder System for Patients in a Primary Care Practice": This study evaluated the impact of an online appointment scheduling and reminder system on patients' satisfaction, no-show rates, and office efficiency. The authors found that the system improved patient satisfaction and reduced no-show rates significantly.
- "A Review of Online Appointment Systems for Healthcare": This article provides an overview of the different types of online appointment systems available for healthcare and discusses their features, advantages, and limitations. The authors also highlight the challenges of implementing such systems and provide recommendations for successful implementation.
- "Design and Implementation of an Online Appointment System for a Multispecialty Clinic": This paper presents the design and implementation of an online appointment system for a multispecialty clinic. The authors describe the system's features, such as appointment scheduling, rescheduling, and cancelling, and discuss the system's benefits, including increased patient satisfaction and reduced administrative costs.
- "A Usability Evaluation of Online Appointment Scheduling System for Health Clinics": This study evaluated the usability of an online appointment scheduling system for health clinics from the perspective of patients and healthcare providers. The authors found that the system was easy to use and increased patient satisfaction and efficiency.

6. Scope of project

Online Appointment Management System is a computerised management system. This system keeps the records of the Online Appointment being booked in this organisation. The proposed system will keep a track of different appointment details including the doctors and clinic details.

This project has GUI based software that will help in storing, updating and retrieving the information through various user-friendly menu-driven modules.

The project “Online Appointment Management System” is aimed to develop to maintain the day-to-day appointments. Main objective of this project is to provide appointments and clinic details to customers. This software application will help admin to handle customer information, doctor details and clinic details.

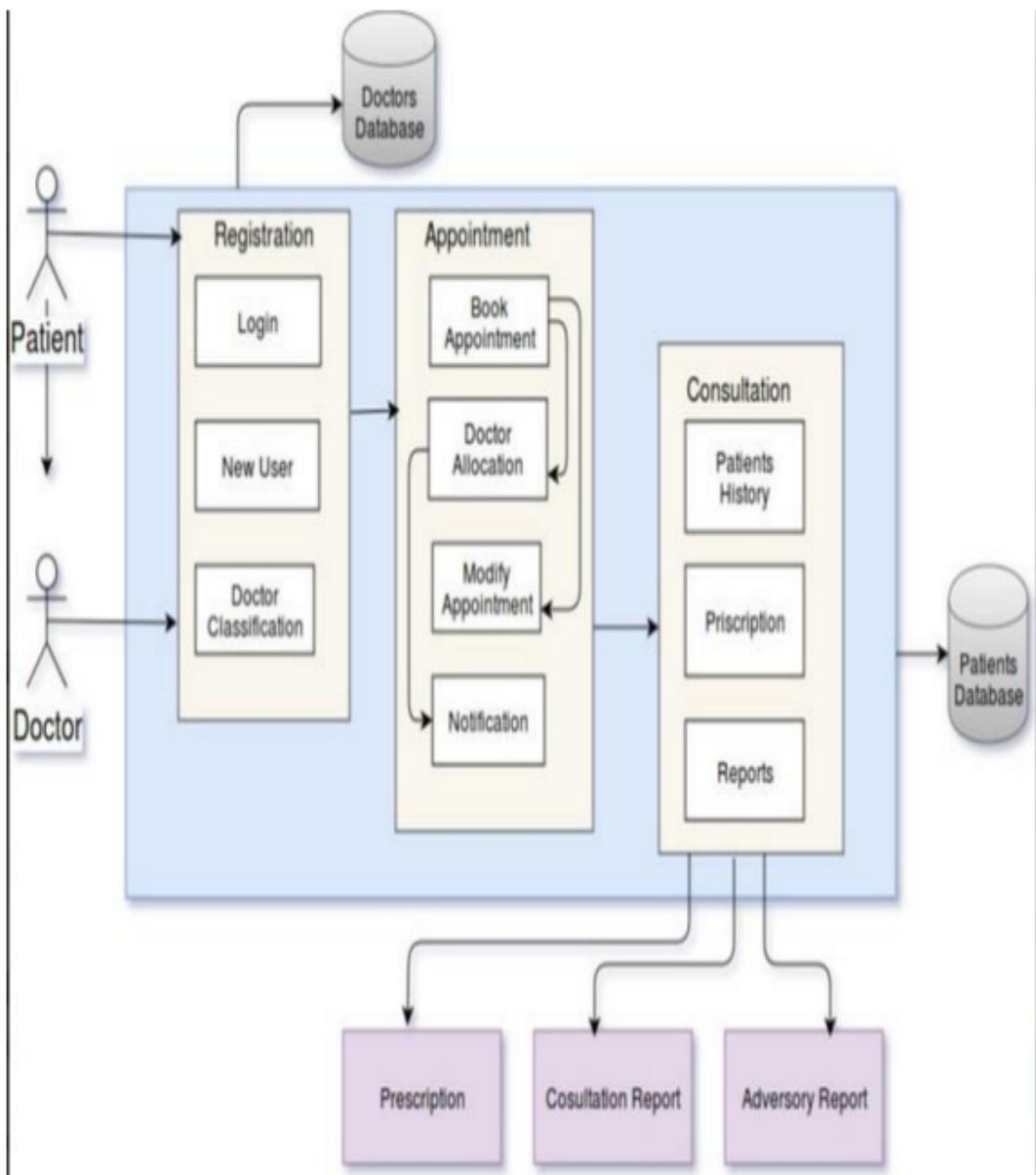
Detailed explanations about modules and design are provided in project documentation. The existing system is a manually maintained system. All the appointment records are to be maintained for the details of each customer, doctor details, appointment registration, etc. All these details are entered and retrieved manually, because of this there are many disadvantages like:

- 1 - Time Consuming
- 2 - Exhaustive and inefficient updating process
- 3 - Inaccuracy of data and room for human error in manual entry

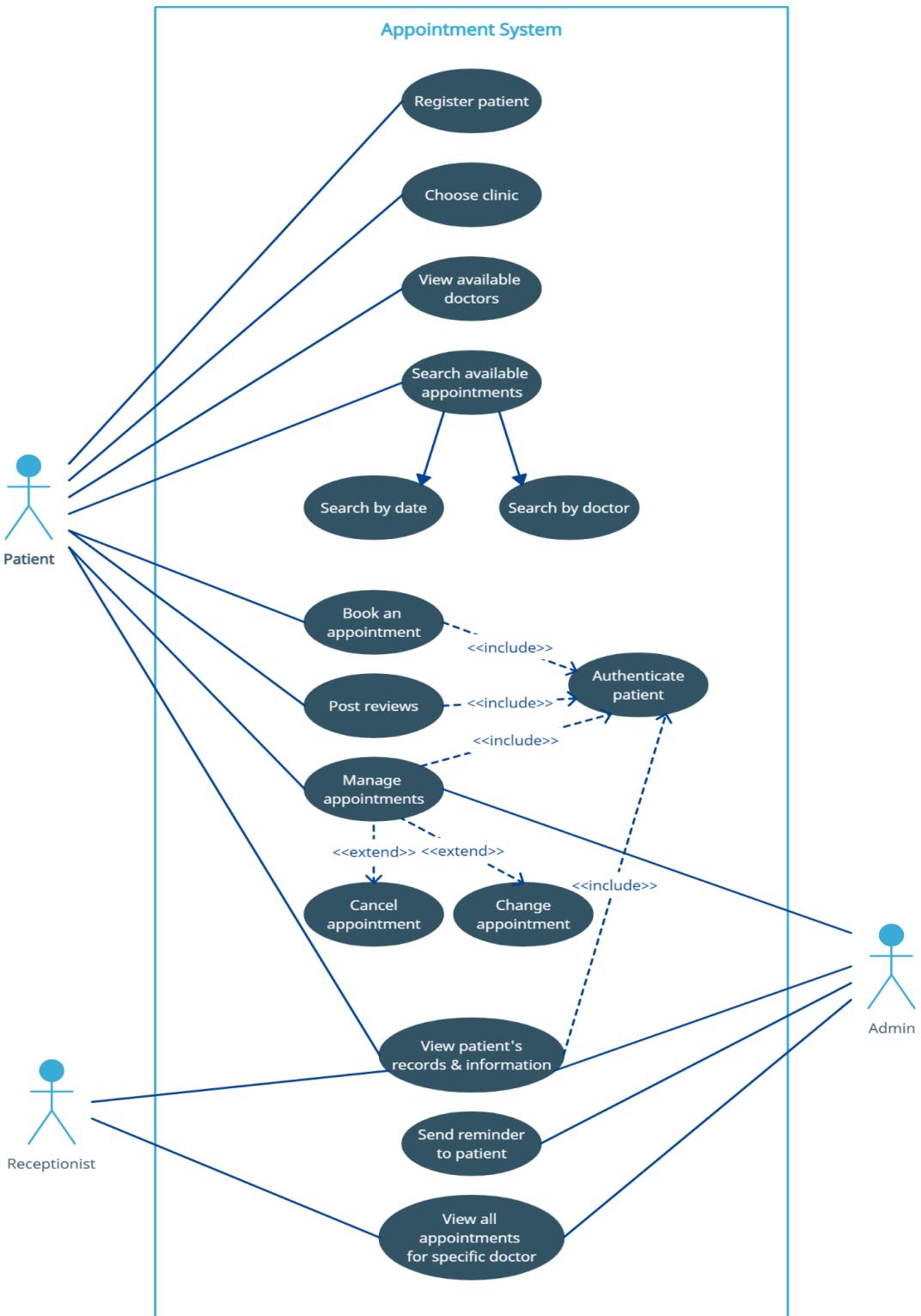
- The main aim of the project for an online appointment system for healthcare includes the development of a web-based application that allows patients to schedule, modify, or cancel appointments with their healthcare providers. The system should include features such as appointment reminders, secure messaging, and access to medical records.
- The project should also involve the development of an administrative dashboard for healthcare providers to manage appointments, view patient information, and generate reports. The system should be user-friendly, accessible, and responsive, and it should comply with healthcare regulations and data privacy laws.

- The project should also include testing and quality assurance to ensure the system is reliable and secure. The implementation of the system should involve collaboration with healthcare providers and patients to ensure that their needs and requirements are considered in the design and development process.
- The project should also include training and support for healthcare providers and patients to ensure that they can effectively use the system. Overall, the scope of the project for an online appointment system for healthcare is to provide a convenient and efficient tool for patients to manage their appointments and improve access to care while reducing administrative burden for healthcare providers.

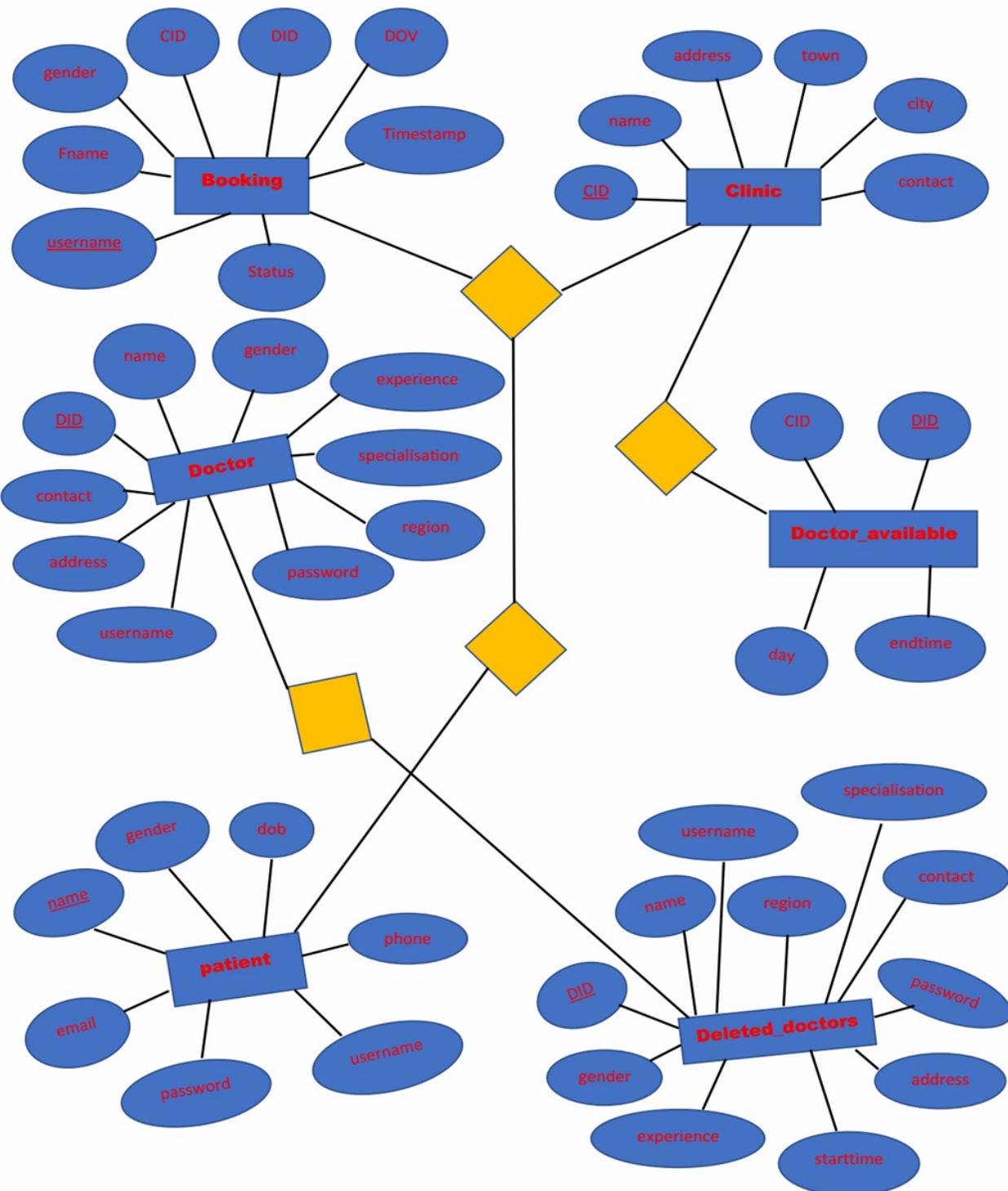
7. Proposed Architecture Diagram



8. Use Case Diagram



9. ER Diagram



10. Front end (Database) design, software used with explanation

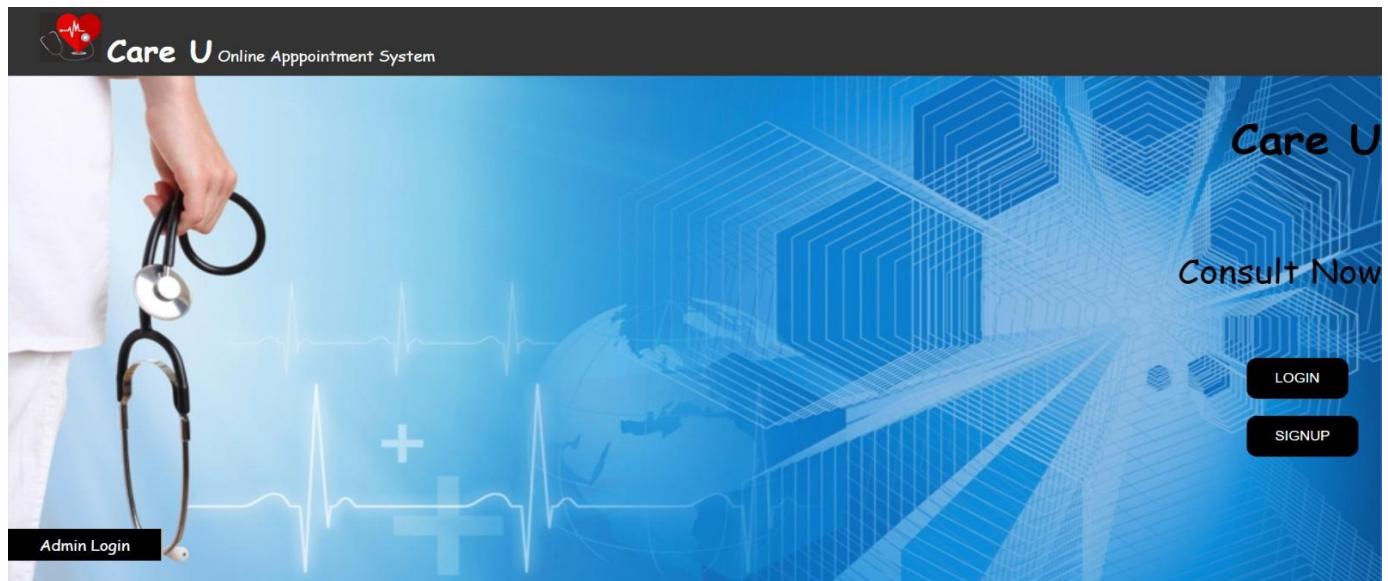
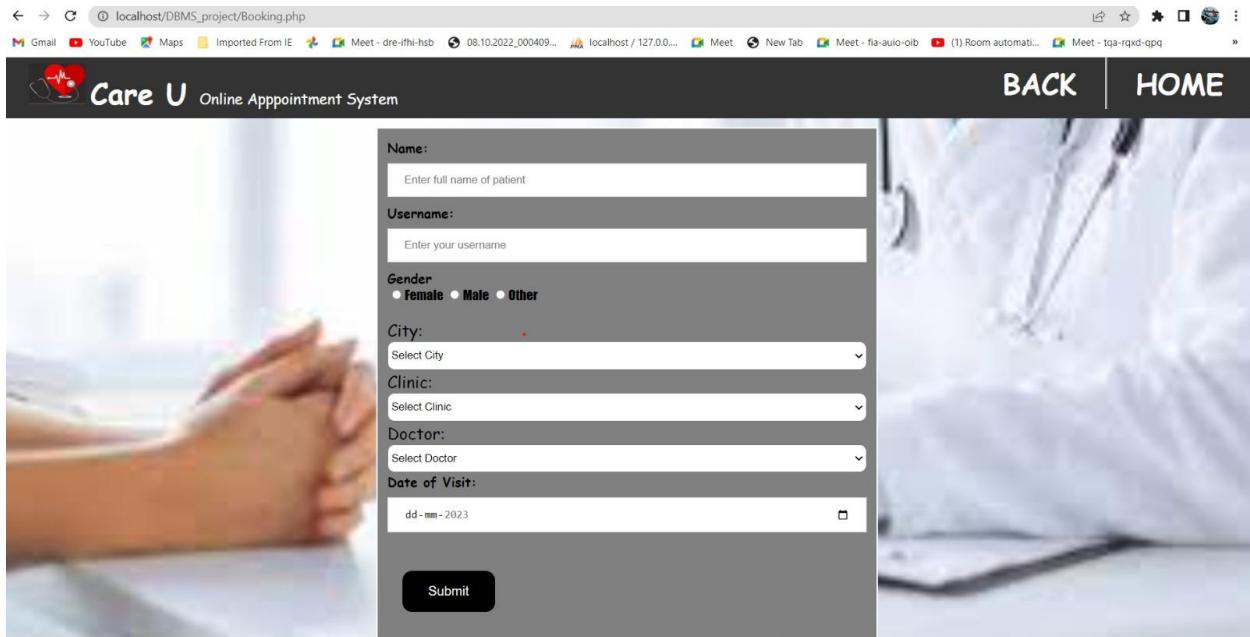


FIG.1:This is Home Page

The image shows the Patient Registration form. The header 'Care U Online Appointment System' and 'HOME' are visible. The form consists of several input fields: 'Enter Name:' with placeholder 'Enter your full name', 'Enter Date of birth:' with placeholder 'dd-mm-yyyy', 'Gender:' with radio buttons for 'Male', 'Female', and 'Other', 'Contact number:' with placeholder 'Enter your number', 'Enter Username:' with placeholder 'Enter your username', 'Enter email:' with placeholder 'Enter email-id', 'Enter Password:' with placeholder 'Enter the password', and 'Confirm Password:' with placeholder 'Enter the password'. To the right of the form, there is a partial view of a doctor wearing a white coat and holding a stethoscope.

FIG.2: This shows the Patient Registration

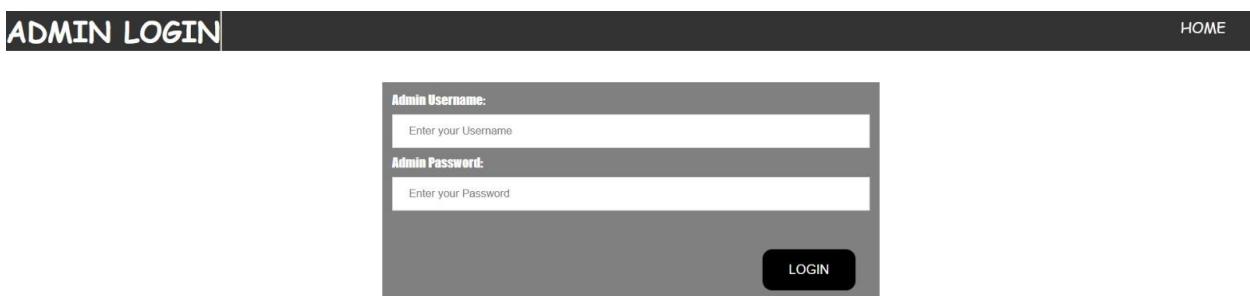


The screenshot shows a web browser window with the URL `localhost/DBMS_project/Booking.php`. The page title is "Care U Online Appointment System". On the right side of the header are "BACK" and "HOME" buttons. The main content area contains a form for appointment booking. The form fields include:

- Name:** A text input field with placeholder text "Enter full name of patient".
- Username:** A text input field with placeholder text "Enter your username".
- Gender:** Radio buttons for "Female", "Male", and "Other".
- City:** A dropdown menu labeled "Select City".
- Clinic:** A dropdown menu labeled "Select Clinic".
- Doctor:** A dropdown menu labeled "Select Doctor".
- Date of Visit:** A date input field with placeholder text "dd - mm - 2023".

At the bottom of the form is a "Submit" button.

FIG.3: This shows the Appointment Booking



The screenshot shows a web browser window with the title "ADMIN LOGIN" on the left and "HOME" on the right. The main content area contains a login form with two fields:

- Admin Username:** A text input field with placeholder text "Enter your Username".
- Admin Password:** A text input field with placeholder text "Enter your Password".

At the bottom of the form is a "LOGIN" button.

FIG.4: This is the admin login page

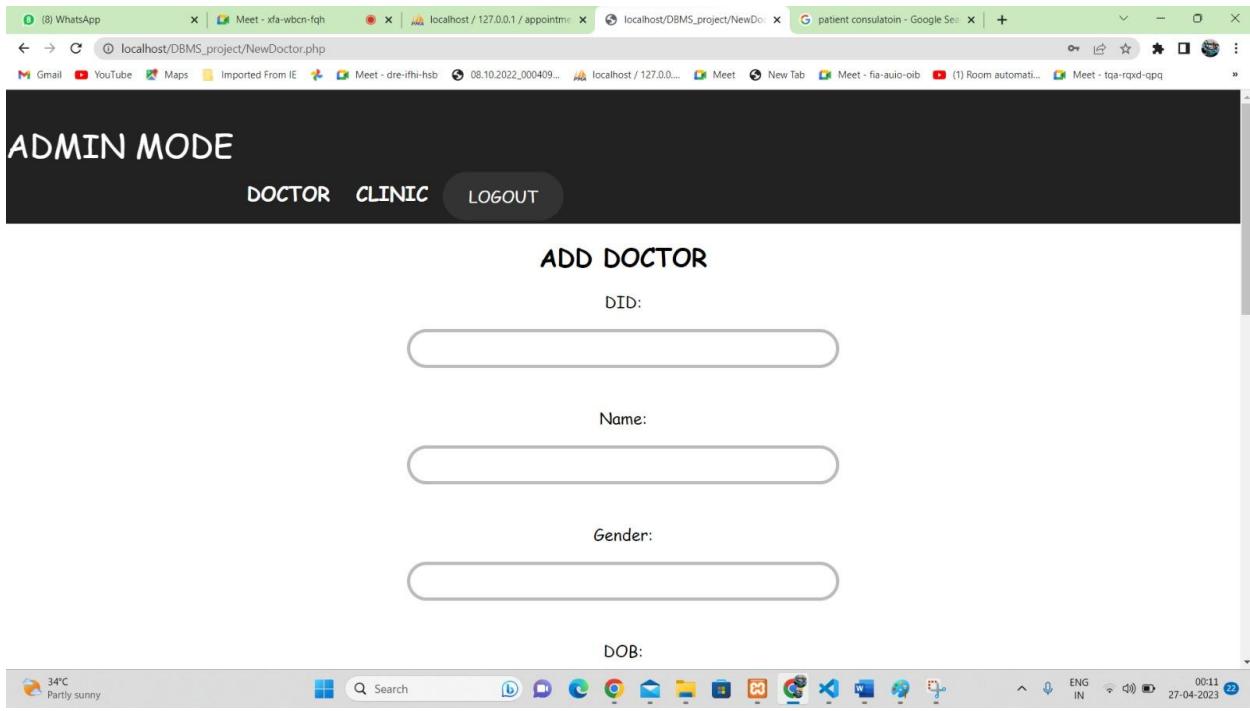


FIG.5: This shows the Doctor Registration by Admin

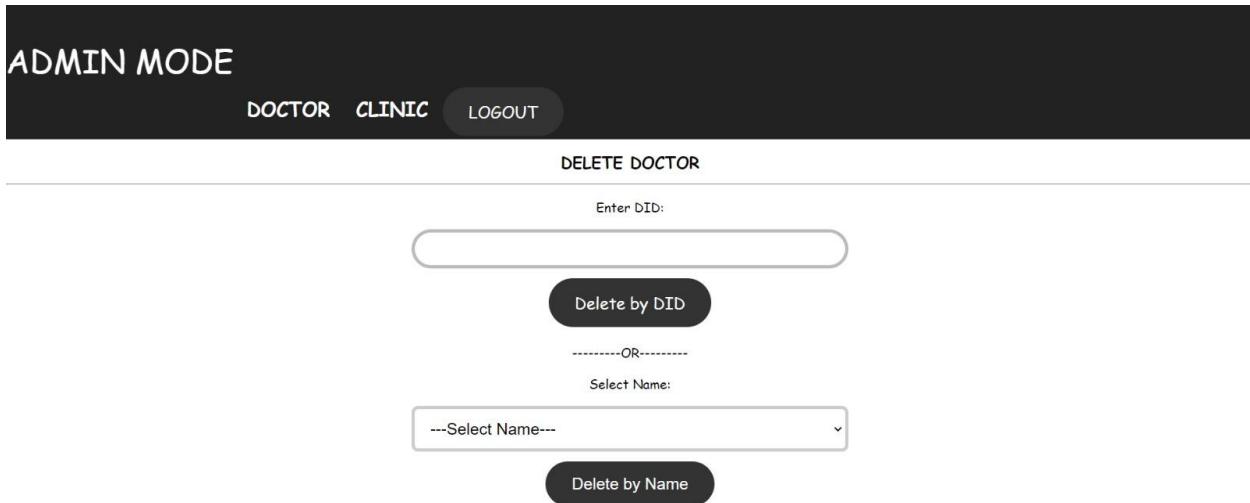


FIG.6:This shows the Deletion of Doctor Details by admin

ADMIN MODE

DOCTOR CLINIC

LOGOUT

SHOW CLINIC

TOTAL CLINICS AVAILABLE=2

| CID | Name | Address | Town | City | Contact |
|-----|--------------|---------|---------|---------|------------|
| 412 | santanclinic | Chennai | chennai | chennai | 7989889868 |
| 789 | lono | chennai | chennai | chennai | 798998989 |

FIG.7: This table shows Clinic list

ADMIN MODE

DOCTOR CLINIC

LOGOUT

DOCTORS SCHEDULE

TOTAL CLINICS AVAILABLE=2

| CID | Clinic Name | DID | Doctor Name | Day | Time |
|-----|----------------------|-----|-------------|-----------|-------------------|
| 412 | santanclinic-chennai | 101 | Tanumay | Monday | 09:00:00-19:27:00 |
| 789 | lono-chennai | 120 | narashimha | Wednesday | 09:00:00-12:00:00 |

FIG.8: This Table shows Clinic List with appointed doctor

ADMIN MODE

DOCTOR CLINIC

LOGOUT

SHOW ALL DOCTORS

TOTAL DOCTORS IN DATABASE=2

| DID | Doctor Name | Date Of Birth | Experience | Specialisation | Address | Contact | Region |
|-----|----------------|---------------|------------|-----------------|----------------------|------------|---------|
| 101 | Dr. Tanumay | 2023-04-06 | 10 years | heartspecialist | guduvcanheri,chennai | 1234567890 | chennai |
| 120 | Dr. narashimha | 2003-10-06 | 3 years | orthopedic | chennai | 7894561223 | chennai |

FIG.9: This Table shows the Doctor List

ADMIN MODE

DOCTOR CLINIC

LOGOUT

ADD CLINIC

CID:

Name:

Address:

Town:

FIG.10: This shows the adding of clinic by admin

ADMIN MODE

DOCTOR CLINIC

LOGOUT

DELETE CLINIC

Enter CID:

Delete by CID

-----OR-----

Select Name:

---Select Name---

Delete by Name

FIG.11: This shows the deletion of clinic by admin

ADMIN MODE

DOCTOR CLINIC

LOGOUT

ASSIGN DOCTOR TO A CLINIC

City:

Select City

Clinic:

Select Clinic

Doctor:

Select Doctor

Available Days

- Monday:
- Tuesday:
- Wednesday:
- Thursday:
- Friday:
- Saturday:
- Sunday:

FIG.12: This shows the assignment of a doctor to a clinic

ADMIN MODE

DOCTOR CLINIC

LOGOUT

REMOVE DOCTOR FROM A CLINIC

City:

Select City

Clinic:

Select Clinic

Doctor & Time:

Select Day & Time

Submit

FIG.13: This shows the removal of a doctor from a clinic

We have used the following front-end languages:

- **HTML**

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other

objects, such as forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other elements are delineated by *tags*, written using angle brackets. Tags such as surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

- HTML is used to define the structure of web pages, including text, images, videos, and links. It uses a set of markup tags, also known as elements, to define the structure and content of the page. HTML tags are enclosed in angle brackets and are typically used in pairs, with the opening tag indicating the start of an element and the closing tag indicating the end.
- HTML includes a range of elements, such as headings, paragraphs, lists, tables, and forms. It also provides support for multimedia elements, such as images, audio, and video, and allows for the inclusion of scripts and stylesheets for enhanced functionality and styling.
- HTML is widely used by web developers to create static and dynamic web pages. It is constantly evolving, with new versions and features being released regularly. HTML5 is the latest version of HTML, which provides a range of new features and functionality, including support for multimedia elements and enhanced accessibility.

- CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications

- CSS is designed primarily to enable the separation of presentation and content, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.
- CSS, which stands for Cascading Style Sheets, is a style sheet language used to describe the presentation of web pages. It is used to control the layout, formatting, and appearance of web pages, including fonts, colours, spacing, and positioning.
- CSS is used in conjunction with HTML and JavaScript to create responsive and visually appealing web pages. CSS styles are typically defined in separate files or within the HTML file using the "style" tag. They are applied to HTML elements using selectors, which target specific elements or groups of elements on the page.
- CSS includes a wide range of features, such as layout and positioning of elements, typography, and colour management. It also provides support for responsive design, allowing web pages to adapt to different screen sizes and devices.
- CSS is widely used by web developers to create visually appealing and user-friendly web pages. It is supported by all modern web browsers and is constantly evolving to include new features and functionality.

11. Back end (Database) design, software used with explanation

The screenshot shows the phpMyAdmin interface for the 'appointment' database. The left sidebar lists various databases and tables. The main area displays a table of all tables in the 'appointment' database, including their names, types, rows, and sizes. A 'Create new table' form is also visible at the bottom.

| Table | Action | Rows | Type | Collation | Size | Overhead |
|------------------|---|------|--------|--------------------|----------|----------|
| booking | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| clinic | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| deleted_doctors | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| doctor | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| doctor_available | Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| patient | Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |

FIG.14: This shows the list of all the tables

The screenshot shows the phpMyAdmin interface for the 'booking' table in the 'appointment' database. The top bar shows the query: 'SELECT * FROM `booking`'. The main area displays the booking table with columns: username, Fname, gender, CID, DID, DOV, Timestamp, and Status. One row is selected for viewing.

| | username | Fname | gender | CID | DID | DOV | Timestamp | Status |
|--|----------|-------|--------|-----|-----|------------|---------------------|--|
| | abdul_90 | abdul | male | 412 | 101 | 2023-05-01 | 2023-04-26 20:03:31 | Booking Registered Wait for the update |

FIG.15: This table shows the booking details of the patient

The screenshot shows the phpMyAdmin interface for the 'appointment' database. The left sidebar lists various tables, and the main area is focused on the 'clinic' table. The SQL query at the top is:

```
SELECT * FROM `clinic`
```

The results table shows two rows of data:

| | id | CID | name | address | town | city | contact |
|--|-----------|------------|-------------|----------------|-------------|-------------|----------------|
| | 2 | 412 | santancinic | Chennai | chennai | chennai | 7989889868 |
| | 3 | 789 | liona | chennai | chennai | chennai | 7989989899 |

FIG.16: This table shows the no of clinics available

The screenshot shows the phpMyAdmin interface for the 'appointment' database. The left sidebar lists various tables, and the main area is focused on the 'doctor' table. The SQL query at the top is:

```
SELECT * FROM `doctor`
```

The results table shows two rows of data:

| | id | DID | name | gender | dob | experience | specialisation | contact | address | username | password | region |
|--|-----------|------------|-------------|---------------|------------|-------------------|-----------------------|----------------|-----------------------|-----------------|-----------------|---------------|
| | 3 | 101 | Tanumay | male | 2023-04-06 | 10 years | heartspecialist | 1234567890 | guduvcanheri, chennai | Tanumay_78 | Tanumay_90 | chennai |
| | 4 | 120 | norashimha | male | 2003-10-06 | 3 years | orthopedic | 7894561223 | chennai | KOTA_90 | Kotha@1234 | chennai |

FIG.17: This table shows the details of the doctor

The screenshot shows the phpMyAdmin interface for the 'appointment' database. The left sidebar lists various tables: New, appointment, booking, clinic, deleted_doctors, doctor, doctor_available, patient, cms_db, collegospace, crime_portal, index, information, just, login, music, mysql, performance_schema, phpmysadmin, sign, signup, songs, test, transaction, and wordpress. The 'doctor_available' table is selected. The main area displays the table structure with columns CID, DID, day, starttime, and endtime. Two rows are shown: one for Monday from 09:00 to 19:27 and another for Wednesday from 09:00 to 12:00.

FIG.18: This table shows the availability of doctor in clinic

The screenshot shows the phpMyAdmin interface for the 'appointment' database. The left sidebar lists various tables, including 'patient'. The 'patient' table is selected. The main area displays the table structure with columns id, name, gender, dob, phone, username, email, and password. One row is shown: a male patient named Abdul_90 born on 2002-10-06 with contact details and a password.

FIG.19: This table shows the patient details

The screenshot shows the phpMyAdmin interface on a web browser. The URL is `localhost/phpmyadmin/index.php?route=/sql&pos=0&db=appointment&table=deleted_doctors`. The left sidebar shows a tree view of databases and tables, including 'appointment' (which contains 'deleted_doctors'), 'cms_db', 'collegeinspace', 'crime_portal', 'index', 'information', 'just', 'login', 'music', 'mysql', 'performance_schema', 'phpmyadmin', 'signin', 'signup', 'songs', 'test', 'transaction', and 'wordpress'. The main panel shows the 'deleted_doctors' table under the 'appointment' database. The table has columns: id, DID, Name, gender, dob, experience, specialisation, contact, address, and username. A SQL query is entered in the query editor: `SELECT * FROM `deleted_doctors``. The results pane below the query editor displays the message: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0044 seconds.)". There are also options for Profiling, Edit inline, Explain SQL, Create PHP code, Refresh, and various operations like Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.

FIG.20: This table shows the list of deleted doctors

- **PHP MY ADMIN**

PhpMyAdmin is a free and open-source web-based tool written in PHP used to manage MySQL and MariaDB databases. It provides an intuitive graphical user interface that allows users to perform various tasks related to database management, such as creating and deleting databases, tables, and fields, inserting, updating, and deleting data, running SQL queries, and managing user accounts and permissions.

PhpMyAdmin offers a range of features, including support for multiple languages, import and export of data in various formats, visual representation of database relationships, and the ability to create and execute SQL queries directly from the interface.

It also includes a set of security features, such as the ability to restrict access to specific databases and tables, and an audit log that records all actions performed by users.

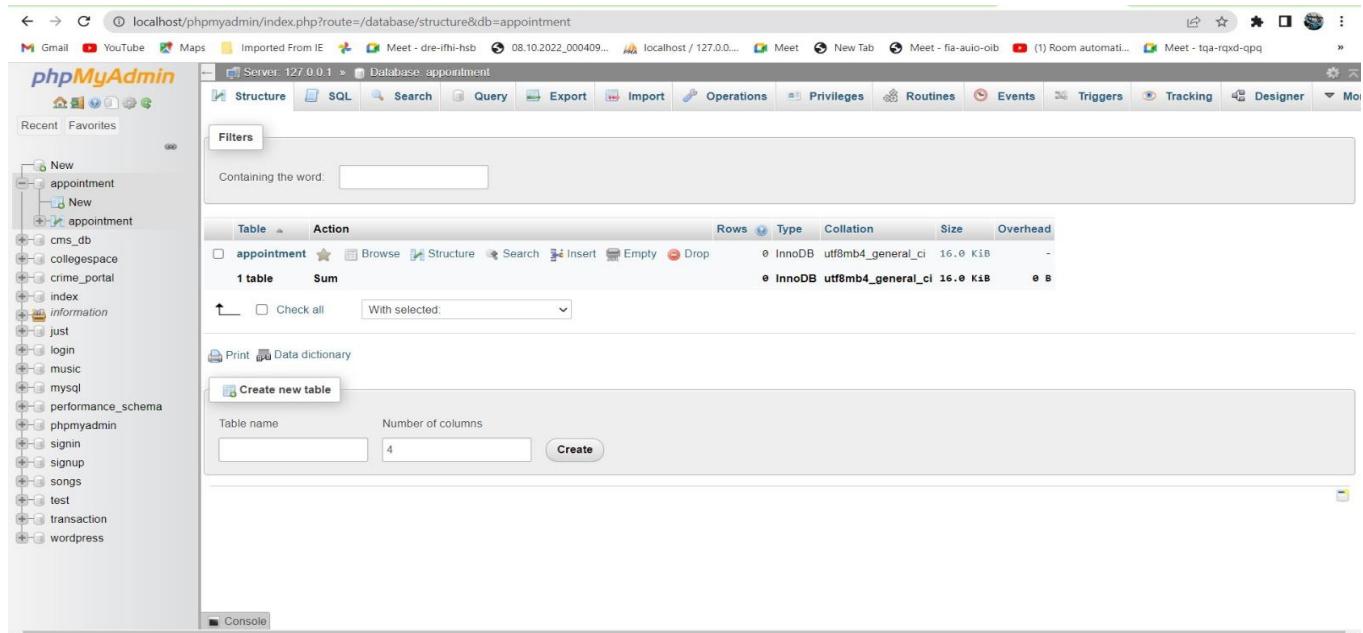
PhpMyAdmin is widely used by web developers, system administrators, and database administrators for managing and maintaining MySQL and MariaDB databases. It is available for download from the official website and can be installed on various web servers, including Apache and Nginx.

- **MY SQL**

We have used, in this project , MySQL which is an open-source relational database management system. MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

- MySQL is a popular open-source relational database management system (RDBMS) used to store, manage and retrieve data. It is widely used for web-based applications, including content management systems, e-commerce platforms, and online forums.
- MySQL supports various platforms, including Windows, Linux, and macOS, and provides a range of features, such as support for multiple databases, multi-threading, and indexing for efficient data retrieval. It also includes a set of programming interfaces and tools, such as JDBC, ODBC, and MySQL Workbench, to facilitate application development and database management.
- MySQL uses a structured query language (SQL) for managing data, which allows users to create, modify, and retrieve data in a structured and efficient manner. It also supports transactions and provides various security features, such as encryption and authentication, to ensure the security of data stored in the database.
- MySQL is widely used by web developers, system administrators, and database administrators for managing and maintaining data in various web applications. It is available for download from the official website and can be installed on various web servers, including Apache and Nginx.

12. Type of Connectivity used for database access with explanation



The screenshot shows the phpMyAdmin interface for a MySQL database named 'appointment'. The left sidebar lists various databases, including 'appointment' (selected), 'cms_db', 'collegeinspace', 'crime_portal', 'index', 'information', 'just', 'login', 'music', 'mysql', 'performance_schema', 'phpmyadmin', 'signin', 'signup', 'songs', 'test', 'transaction', and 'wordpress'. The main content area displays the 'Structure' tab for the 'appointment' database. It shows one table named 'appointment' with 1 row and 1 column. The table has an auto-increment primary key. The table structure is defined as follows:

| Field | Type | Collation | Size | Overhead |
|-------|---------|--------------------|------|----------|
| id | int(11) | utf8mb4_general_ci | 4 | 4 B |

Below the table structure, there is a 'Create new table' form with 'Table name' set to 'appointment' and 'Number of columns' set to '4'. The 'Create' button is visible.

phpMyAdmin is the most popular application for MySQL database management. We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operations like managing databases, relations, tables, columns, indexes, permissions, and users, etc., on MySQL and MariaDB. These operations can be performed via user interface, while we still have the ability to execute any SQL statement.

phpMyAdmin is translated into 72 languages and also supports both RTL and LTR languages so that a wide range of people can easily use this software. We can run MySQL queries, repair, optimise, check tables, and also execute other database management commands. phpMyAdmin can also be used to perform administrative tasks such as database creation, query execution.

phpMyAdmin is a web-based application used to manage MySQL databases. The type of connectivity used for phpMyAdmin is through the MySQL protocol, which uses the TCP/IP network protocol to establish a connection between the PHP-based web application and the MySQL database server.

When you access phpMyAdmin, it establishes a connection to the MySQL server using the MySQL protocol, and allows you to perform various database management tasks, such as creating and modifying tables, running queries, importing and exporting data, and more.

If we don't have access to a web hosting server and want to learn how to use this application to manage the MySQL database locally, we can install PhpMyAdmin on our PC using third-party

products, for example, XAMPP, which is the most popular PHP development environment for Windows, Linux and OS X, a completely free, Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be easy to install and to use. We also have other means of installation, which we can see in the PhpMyAdmin documentation.

Another third-party tool used to install PhpMyAdmin is the WampServer, which is a Windows-only web development environment that allows us to create web applications with Apache2, PHP and a MySQL database and also installs PhpMyAdmin to manage the MySQL database. In my example I have installed this tool to access PhpMyAdmin.

13. List of Modules and Functionalities in each of the modules with explanation

Here are the modules of our online appointment booking system:-

- **Home Page :-** As soon as the user enters, the first page will be the home page in which there will be three options: user login, user registration and admin login.
- **Sign Up Page :-** User can register through the sign up page by entering his/her details such as Name, DOB, Phone number etc, create his/her own password in this page.
- **Login page:-** This is an authentication page in which it checks for the credentials of the users.
- **Admin Login Page:-** In this page admin has to login with his username and password to further alter the data in the website.
- **Admin Main Page:-** Admin can choose between doctor clinic and logout to which he can add or delete the doctor, or add or delete the clinic.
- **Assigning New Doctor:-** Admin can add a new doctor and assign to a specific clinic.
- **Add Clinic:-** Here admin can add a clinic and view list of clinic which are present on the website.
- **Patient page:-** In this patient can book an appointment or delete an appointment and view the doctor details and his/her appointment history.
- **Assign Doctor to clinic:-** In this module admin will assign a doctor to a clinic.
- **Delete a Doctor from Clinic:-** Here admin can delete doctor details from assigned clinic.
- **Delete a clinic:-** In this Admin can delete a clinic.
- **List of Doctors:-** Here the admin can view all the doctors that he register in the website.
- **Delete a doctor:-** Here admin can delete a doctor details.
- **Doctor Schedule:-** Admin can see all the doctors' registered schedules.
- **Clinic List:-** Here it will show all the available clinics.

14. List of Database Relations

Table 1 **Booking:** This table consists of details of the appointments

| COLUMN NAME | DATATYPE SIZE | CONSTRAINTS | DESCRIPTION |
|-------------|------------------|-------------|-----------------------------------|
| Username | Varchar (30) | Primary key | Username of customer |
| Fname | Varchar (30) | Not Null | Name of the customer |
| Gender | Varchar (10) | Not Null | Gender of customer |
| CID | Int (11) | Not Null | Clinic ID |
| DID | Int (11) | Not Null | Doctor ID |
| DOV | Date | Not Null | Date of Visit |
| Timestamp | Datetime | Not Null | Date and time the booking is done |
| Status | Varchar (50) | Not Null | Status of appointment |

Table 2 **Clinic**: This table consists of the clinic details.

| COLUMN NAME | DATATYPE SIZE | CONSTRAINTS | DESCRIPTION |
|-------------|------------------|-------------|--|
| CID | Int (11) | Primary Key | The ID accepts the integer which is used to identify the clinic. |
| Name | Varchar (30) | Not Null | Name of the clinic |
| Address | Varchar (30) | Not Null | Address of the clinic |
| Town | Varchar (20) | Not Null | Town where the clinic is located |
| City | Varchar (20) | Not Null | City where the clinic is located |
| Contact | Varchar (10) | Not Null | Contact of the clinic |

Table 3 **Doctor:** This table consists of the doctor details

| COLUMN NAME | DATATYPE SIZE | CONSTRAINTS | DESCRIPTION |
|----------------|------------------|-------------|--|
| DID | Int (11) | Primary Key | The ID accepts the integer which is used to identify the doctor. |
| Name | Varchar (30) | Not Null | Name of the doctor |
| Gender | Varchar (10) | Not Null | Gender of the doctor |
| DOB | Date | Not Null | Doctor's Date of Birth |
| Experience | Varchar (30) | Not Null | Doctor's working experience in years |
| Specialisation | Varchar (30) | Not Null | An expert or specialist of doctor |
| Contact | Varchar (10) | Not Null | Doctor contact details |
| Address | Varchar (40) | Not Null | Doctors address or home resident |
| Username | Varchar (30) | Not Null | Username of doctor |
| Password | Varchar (20) | Not Null | Doctors password |
| Region | Varchar (20) | Not Null | Doctor resident city |

Table 4 Doctor Available: Table consists of the doctors availability in clinic

| COLOUMN NAME | DATATYPE & SIZE | CONSTRAINTS | DESCRIPTION |
|--------------|-----------------|-------------|---|
| CID | Int (11) | Not Null | The id accepts the integer which is used to identify the clinic |
| DID | Int (11) | Primary key | The id accepts the integer which is used to identify the doctor |
| Day | Varchar (20) | Not Null | The day doctor is available in the clinic |
| Start time | Time | Not Null | Clinic opening time |
| End time | Time | Not Null | Clinic closing time |

Table 5 **Patients**: This table consists of the details about the customer

| COLUMN NAME | DATATYPE SIZE | CONSTRAINTS | DESCRIPTION |
|-------------|------------------|-------------|--|
| Name | Varchar (30) | Primary Key | Name of the customer |
| Gender | Varchar (10) | Not Null | Gender of the customer. |
| DOB | Varchar (50) | Not Null | Customer's Date of Birth |
| Phone | Varchar (10) | Not Null | Phone number of the customer. |
| Username | Varchar (20) | Unique | Username of the customer. |
| Password | Varchar (30) | Not Null | Customer's secured password used while logging in. |
| Email | Varchar (30) | Not Null | Customer's email used to send mails regarding appointments |

Table 6 **Deleted Doctors:** This table consists of the doctor details deleted by admin(Trigger).

| COLUMN NAME | DATATYPE SIZE | CONSTRAINTS | DESCRIPTION |
|----------------|------------------|-------------|--|
| DID | Int (11) | Primary Key | The ID accepts the integer which is used to identify the doctor. |
| Name | Varchar (30) | Not Null | Name of the doctor |
| Gender | Varchar (10) | Not Null | Gender of the doctor |
| DOB | Date | Not Null | Doctor's Date of Birth |
| Experience | Varchar (30) | Not Null | Doctor's working experience in years |
| Specialisation | Varchar (30) | Not Null | An expert or specialist of doctor |
| Contact | Varchar (10) | Not Null | Doctor contact details |
| Address | Varchar (40) | Not Null | Doctors address or home resident |
| Username | Varchar (30) | Not Null | Username of doctor |

Normalisation

- Process for evaluating and correcting table structures to minimize data redundancies
- Reduces data anomalies
- Works through a series of stages called normal forms:
 - First normal form (1NF)
 - Second normal form (2NF)
 - Third normal form (3NF)

1st NF

Database is in first normal form, if it satisfies the following conditions

1. Contains only atomic values.
2. There are no repeating groups.

2nd NF

A relation schema R is in second normal form, if every non-primary attribute A is fully functionally Dependant on Primary key of R

1. It is in 1NF
2. All non-key attributes are fully functionally dependant on primary key.

3rd NF

1. It is in 2NF
2. There are no transitive functional dependencies

The above tables satisfy all three normal forms

15. Coding

The screenshot shows the Visual Studio Code interface with the 'Booking.php' file open. The code handles patient booking by inserting data into the 'booking' table. It includes validation for required fields and checks if a doctor is available for the specified date.

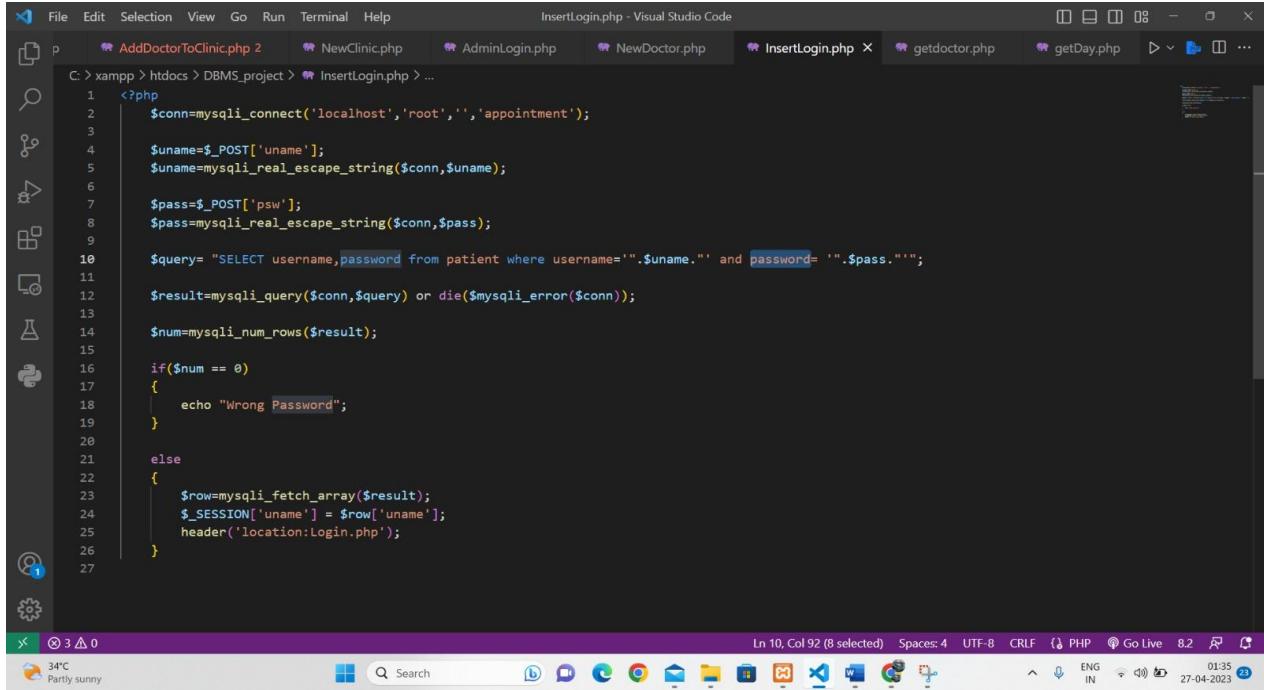
```
138 |         |     </div>
139 |     <?php
140 |     if(isset($_POST['submit']))
141 |     {
142 |
143 |         include 'DBconnect.php';
144 |         $fname=$_POST['fname'];
145 |         $gender=$_POST['gender'];
146 |         $username=$_POST['username'];
147 |         $cid=$_POST['cid'];
148 |         $did=$_POST['doctor'];
149 |         $dov=$_POST['DOV'];
150 |         $status="Booking Registered.Wait for the update";
151 |         $timestamp=date("Y-m-d H:i:s");
152 |         $sql = "INSERT INTO booking (username,Fname,gender,CID,DID,DOV,Timestamp,Status) VALUES ('$username','$fname','$gender','$cid','$di
153 |         if(!empty($_POST['fname'])&&!empty($_POST['gender'])&&!empty($_POST['username'])&&!empty($_POST['cid'])&&!empty($_POST['doctor']) &
154 |         {
155 |             $checkday = strtotime($dov);
156 |             $compareday = date("l", $checkday);
157 |             $flag=0;
158 |             require_once("DBconnect.php");
159 |             $query = "SELECT * FROM doctor_available WHERE DID = '" . $did. "' AND CID='".$cid."'";
160 |             $results = $conn->query($query);
161 |             while($rs=$results->fetch_assoc())
162 |             {
163 |                 if($rs["day"]==$compareday)
164 |                 {
165 |                     $flag++;
166 |                     break;
167 |                 }
168 |             }
169 |         }
170 |     }
171 | 
```

Fig 21. Patient Booking

The screenshot shows the Visual Studio Code interface with the 'AddDoctorToClinic.php' file open. The code inserts data into the 'doctor_available' table for a specific doctor and clinic. It also handles errors and provides feedback to the user.

```
114 |         <label>
115 |             <button name="submit" type="submit" style="font-family:cursive">SUBMIT</button>
116 |         </form>
117 |     <center>
118 |     <?php
119 |     if(isset($_POST['submit']))
120 |     {
121 |         include 'DBconnect.php';
122 |         $cid=$_POST['cid'];
123 |         $did=$_POST['doctor'];
124 |         $starttime=$_POST['starttime'];
125 |         $endtime=$_POST['endtime'];
126 |
127 |         foreach($_POST['daylist'] as $daylist)
128 |         {
129 |             $sql = "INSERT INTO doctor_available(CID, DID, day, starttime, endtime) VALUES ('$cid', '$did', '$daylist', '$starttime', '$endtime')";
130 |             if (mysqli_query($conn, $sql))
131 |             {
132 |                 echo "<h2>Record created successfully( CID=$cid DID=$did Day=$daylist )!!</h2>";
133 |                 header("Refresh:3;url=AddDoctorToClinic.php");
134 |             }
135 |             else
136 |             {
137 |                 echo "Error: " . $sql . "<br>" . mysqli_error($conn);
138 |             }
139 |         }
140 |     }
141 |     ?>
142 |     </form>
```

Fig 22. Add Doctor to Clinic



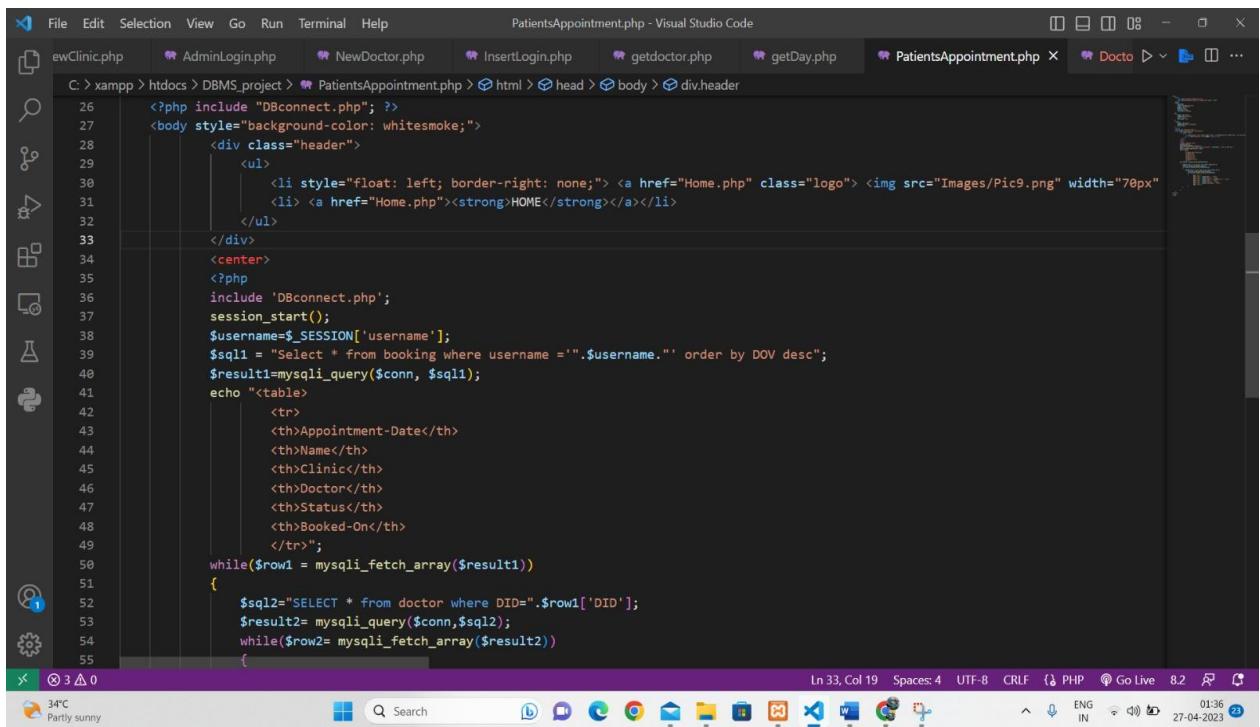
```
1 <?php
2     $conn=mysqli_connect('localhost','root','','appointment');
3
4     $uname=$_POST['uname'];
5     $uname=mysqli_real_escape_string($conn,$uname);
6
7     $pass=$_POST['psw'];
8     $pass=mysqli_real_escape_string($conn,$pass);
9
10    $query= "SELECT username,password from patient where username='".$uname."' and password = '".$pass."'";
11
12    $result=mysqli_query($conn,$query) or die($mysqli_error($conn));
13
14    $num=mysqli_num_rows($result);
15
16    if($num == 0)
17    {
18        echo "Wrong Password";
19    }
20
21    else
22    {
23        $row=mysqli_fetch_array($result);
24        $_SESSION['uname'] = $row['uname'];
25        header('location:Login.php');
26    }
27
```

34°C Partly sunny

Search

Spaces: 4 UTF-8 CRLF Go Live 8.2 ENG IN 01:35 27-04-2023

Fig 23. Patient Registration



```
1 ewClinic.php AdminLogin.php NewDoctor.php InsertLogin.php getdoctor.php getDay.php PatientsAppointment.php Doctor ...
26     <?php include "DBconnect.php"; ?
27     <body style="background-color: whitesmoke;">
28         <div class="header">
29             <ul>
30                 <li style="float: left; border-right: none;"> <a href="Home.php" class="logo"> 
31                 <li> <a href="Home.php"><strong>HOME</strong></a></li>
32             </ul>
33         </div>
34         <center>
35             <?php
36                 include 'DBconnect.php';
37                 session_start();
38                 $username=$_SESSION['username'];
39                 $sql1 = "Select * from booking where username ='".$username."' order by DOV desc";
40                 $result1=mysqli_query($conn, $sql1);
41                 echo "<table>
42                     <tr>
43                         <th>Appointment-Date</th>
44                         <th>Name</th>
45                         <th>Clinic</th>
46                         <th>Doctor</th>
47                         <th>Status</th>
48                         <th>Booked-On</th>
49                     </tr>";
50                 while($row1 = mysqli_fetch_array($result1))
51                 {
52                     $sql2="SELECT * from doctor where DID='".$row1['DID']';
53                     $result2= mysqli_query($conn,$sql2);
54                     while($row2= mysqli_fetch_array($result2))
55                     {
```

34°C Partly sunny

Search

Spaces: 4 UTF-8 CRLF Go Live 8.2 ENG IN 01:36 27-04-2023

Fig 24. Patient Appointment

The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** DoctorSchedule.php - Visual Studio Code
- Code Editor:** The main area contains PHP code for generating a doctor schedule table. The code includes database queries to fetch data from 'doctor' and 'clinic' tables based on a selected 'DID'.
- Left Sidebar:** Includes icons for file operations like Open, Save, Find, and Replace, along with a search bar and a preview pane.
- Right Sidebar:** Shows a preview of the generated HTML table and various status indicators.
- Bottom Status Bar:** Ln 101, Col 70 (7 selected), Spaces: 4, UTF-8, CRLF, PHP, Go Live, 8.2, 34°C Partly sunny, 01:38, 27-04-2023.

Fig 25. Doctor Schedule

16. Conclusion

In conclusion, an online healthcare appointment system can be an effective and efficient way to manage appointments for both patients and healthcare providers. It offers a number of benefits, including increased accessibility and convenience for patients, reduced administrative workload for healthcare providers, and improved patient outcomes through timely access to care.

However, the successful implementation of an online healthcare appointment system depends on several factors, such as user-friendly interface design, secure data management, and adequate technical support. It is also important to consider the needs of all stakeholders, including patients, healthcare providers, and administrative staff, in the design and implementation process.

Overall, an online healthcare appointment system can be a valuable tool for improving healthcare delivery and patient outcomes, but it should be carefully planned and implemented to ensure its effectiveness and success.

17. References

1. <https://ijisrt.com/wp-content/uploads/2018/04/Implementation-of-Doctor%20%80%99s-Appointment-Application-For-Life-Care-HospitalSion.pdf>
2. <https://ijiet.com/wp-content/uploads/2014/12/4.pdf>
3. https://www.researchgate.net/figure/The-architecture-of-proposed-patient-appointment-scheduling-system_fig1_249643353
4. <https://app.creately.com/d/IIB3y7rSN7n/edit>
5. <https://www.javatpoint.com/phpmyadmin>