DOCTOR APPOINTMENT MANAGEMENT SYSTEM

A Project Report submitted to

SCOTT CHRISTIAN COLLEGE (AUTONOMOUS)

(Affiliated to Manonmaniam Sundaranar University)

In partial fulfillment of the requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS.

Submitted by

Peer Mohammed Mubeen. S	**2000622**
Abishek Kumar. I	**2000603**
Abish Kumar. R	**2000602**

Under the Guidance of

Mrs. Suguna Jasmin. R

(Assistant Professor)

Department of Computer Applications

Nagercoil - 629 003

2022 - 2023

CERTIFICATE

This is to certify that the project entitled "**DOCTOR APPOINTMENT MANAGEMENT SYSTEM**" is a bonafide record work done by **PEER MOHAMMED MUBEEN. S (Reg.No.2000622), ABISHEK KUMAR. I (Reg. No.2000603), ABISH KUMAR (REG.NO:2000602)** in partial fulfillment of the requirement for the award of degree of Bachelor of Computer Applications during the academic year 2022-2023.

Internal Guide

Mrs. SUGUNA JASMIN. R, M.Sc., M.Phil.
Department of Computer Applications,
Scott Christian College (autonomous)
Nagercoil-3.

Head of the Department

Mrs. S.GNANA SOPHIA,
M.Sc, P.G.D.C.A, M.Phil.
Department of Computer Applications,
Scott Christian College (autonomous)
Nagercoil-3.

Submitted for the project viva-voce held on ______________ at Scott Christian college (Autonomous) Nagercoil – 629 003

Place: Nagercoil
Date:

Examiner

AKNOWLEDGMENT

All Praise, Glory and Honour to the Almighty, source of all knowledge for his gracious presence and guidance that enabled us to complete the project in due time.

We would like to express our gratitude and regards to our principal **DR.J. ROBERT VICTOR EDWARD** for his wishes and kind cooperation in all the aspects for the completion of the project.

We extend our hearty thanks to **Mrs. S.GNANA SOPHIA M.Sc., M.Phil. PGDCA**, Head of the Department, for her valuable guidance, suggestions and encouragement in all the aspects to complete this project successfully.

We extend our hearty thanks to **Mrs. SUGUNA JASMIN. R, M.Sc., M.Phil**, our internal guide for her constant encouragement and expert guidance that enabled us to complete the project effectively.

With great interest we express our thanks to the teaching faculties of our department **Dr. Sheeja**, **Mrs. Shiny** and technical staff **Mr. Lawrence** for providing necessary help and for the sustained interest shown for the completion of this academic exercise.

Our Wholehearted gratitude and thanks to our parents and other family members and friends for their perpetual support and encouragement.

Peer Mohammed Mubeen. S

Abishek Kumar. I

Abish Kumar. R

ABSTRACT

A web-based doctor appointment management system is a software application designed to streamline the scheduling, booking, and management of medical appointments. The system provides patients with an easy-to-use platform for booking appointments online, while also allowing doctors to manage their schedules and patient records in real-time. The web-based doctor appointment management system operates through a web-based platform, which means that it can be accessed from anywhere with an internet connection. This makes it convenient for patients to book appointments and for doctors to manage their schedules on the go. The system is designed to provide patients with a seamless and hassle-free experience when booking appointments. Patients can use the system to search for available appointment slots with their preferred doctor, select the date and time that works best for them, and receive an instant confirmation of their appointment. One of the key benefits of a web-based doctor appointment management system is that it can help to reduce wait times for patients. By providing patients with an easy-to-use platform for booking appointments, doctors can ensure that their schedules are filled more efficiently. This means that patients are more likely to be seen on time, which can help to improve overall patient satisfaction. In addition, the web-based doctor appointment management system can help to improve the accuracy of patient records. By providing doctors with an easy-to-use platform for recording patient information, the system can help to reduce errors and ensure that patient records are kept up to date. This can help to improve patient outcomes and reduce the likelihood of medical errors.

DOCTOR
APPOINTMENT
MANAGEMENT SYSTEM

Contents

1.INTRODUCTION

A web-based doctor appointment management system is a digital platform that enables patients to schedule, reschedule or cancel appointments with healthcare providers. The system simplifies the appointment booking process, allowing patients to easily access medical care services at their convenience. With the advent of digital technology, this system has become an essential tool for healthcare providers who are looking to enhance patient care, streamline operations and boost productivity. The system's primary goal is to provide a hassle-free experience for patients by eliminating long queues and reducing wait times. Patients can access the system from their mobile phones, tablets, or desktop computers, allowing them to schedule appointments anytime and anywhere. They can also receive automated reminders about upcoming appointments and reschedule them if necessary. Additionally, they can access their medical history, prescription details, and test reports through the system, making it easier to manage their health information. For healthcare providers, the system provides a centralized platform to manage appointments, reducing the administrative burden associated with manual appointment booking. It allows healthcare providers to manage their schedules efficiently, reducing the likelihood of double bookings and improving patient flow. The system can also generate reports on patient appointments, cancellations, and rescheduling patterns, providing valuable insights into patient behavior and preferences. Furthermore, the system offers benefits such as reducing no-shows, optimizing resource utilization, enhancing patient engagement, and increasing revenue. By reducing no-shows, healthcare providers can optimize their resources and provide better care for patients who require medical attention. Patients who are engaged in their care tend to be more satisfied with the care they receive, and this increases the likelihood of them recommending the provider to others.

2. SYSTEM ANALYSIS

2.1 Existing System

There are several existing web-based doctor appointment management systems available in the market, such as Zocdoc, Doctolib, Practo, and HealthEngine. These systems provide patients with a platform to book appointments with healthcare providers and offer healthcare providers with tools to manage their schedules and patient data efficiently. And it requires an User account for the patients.

2.2 System Requirements :

2.2.1 Hardware Requirements :

PROCESSOR	: Intel Core i5
RAM	: 8GB
HARDWARE	: 1TB

2.2.2 Software Requirements :

OPERATING SYSTEM	: Windows 7 or above
FRONT-END	: PHP
BACK-END	: MYSQL
SERVER	: XAMPP

2.3 Proposed System :

A proposed system for a web-based doctor appointment management system would allow patients to easily schedule appointments online, view their medical appointments. The system would streamline the appointment process, reducing wait times and increasing patient satisfaction. It's totally cost free.

ADVANTAGES

- Increased Efficiency: A web-based system can automate many routine tasks, such as appointment scheduling, reminders, and patient record keeping. This can save time for both patients and staff and increase the overall efficiency of the healthcare facility.
- Improved Patient Satisfaction: Patients can easily book appointments online without waiting on hold or traveling to the healthcare facility to schedule an appointment. They can also receive reminders about upcoming appointments via email or SMS, which can reduce the risk of missed appointments.
- Access to Real-time Information: Healthcare providers and staff can access patient records, appointment schedules, and other relevant information in real-time from any location with an internet connection. This can facilitate better communication and coordination among staff members, leading to improved patient care.
- Reduced Costs: A web-based system can help reduce administrative costs associated with manual appointment scheduling and record keeping. It can also help reduce the number of missed appointments, which can save healthcare providers money.
- Increased Security: Web-based systems can offer enhanced security measures to protect patient information. This can include encryption, authentication, and access controls, which can help prevent unauthorized access to patient data.

▪ 3. SYSTEM SPECIFICATION

3.1. Operating System

Windows 7:

Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM- compatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows. The first version of Windows, released in 1985, was simply a GUI offered as an extension of Microsoft's existing disk operating system, or MS- DOS. Based in part on licensed concepts that Apple Inc. had used for its Macintosh System Software, Windows for the first time allowed DOS users to visually navigate a virtual desktop, opening graphical "windows" displaying the contents of electronic folders and files with the click of a mouse button, rather than typing commands and directory paths at a text prompt. Subsequent versions introduced greater functionality, including native Windows File Manager, Program Manager, and Print Manager programs, and a more dynamic interface Microsoft also developed specialized Windows packages, including the networkable Windows for Workgroups and the high-powered Windows NT, aimed at businesses. The 1995 consumer release Windows 95 fully integrated Windows and DOS and offered built-in Internet support, including the World Wide Web browser Internet Explorer. With the 2001 release of Windows XP, Microsoft united its various Windows packages under a single banner, offering multiple editions for consumers, businesses, multimedia developers, and others. Windows XP abandoned the long, used Windows 95 kernel (core software code) for a more powerful code base and offered a more practical interface and improved application and memory management The highly successful XP standard was succeeded in late 2006 by Windows Vista, which experienced a troubled rollout and met with considerable marketplace resistance, quickly acquiring a reputation for being a large, slow, and resource-consuming system. Responding to Vista's disappointing adoption rate, Microsoft in 2009 released Windows 7, an OS whose interface web as similar to that of Vista but was met with enthusiasm for its noticeable speed improvement and its modest system requirements.

3.2. Software Description

XAMPP server:

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes. Apache distribution for many servers and command- line executables along with modules such as Apache server, Maria DB, PHP, and Perl XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and Maria DB is the most vividly used database developed by MySQL. The detailed description of these components is given below. Components of XAMPP. defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection 12 of software contains a web server named Apache, a database management system named Maria DB and scripting/ programming languages such as PHP and

Perl. X denotes Cross- platform, which means that it can work on different platforms such as Windows, Linux, and mac OS.

Many other components are also part of this collection of software and are explained below:

- Cross-Platform: Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.

Apache: It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user. Maria DB: Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by Maria DB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion. XAMPP server XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the PS stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command- fine executables along with modules such as Apache server, MariaDB, PHP, and Perl XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to 13 test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

The detailed description of these components is given below. Components of XAMPP As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named Apache, a database management system named MariaDB and scripting/ programming languages such as PHP and Perl. X denotes Cross- platform, which means that it can work on different platforms such as Windows, Linux, and macOS. Many other components are also part of this collection of software and are explained below.

- Cross-Platform: Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.
- Apache: It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.
- MariaDB: Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.

PHP :

PHP is a popular server-side scripting language used for web development. It was originally created in 1994 by Rasmus Lerdorf and has since evolved into a mature and widely used language.

Some of the key features of PHP include:

- Simplicity: PHP is relatively easy to learn and use, making it a popular choice for beginners.
- Flexibility: PHP can be used to develop a wide range of web applications, from simple websites to complex web applications.
- Integration: PHP can be easily integrated with other technologies, such as HTML, CSS, JavaScript, and various databases.
- Open source: PHP is open source, which means that the source code is freely available and can be modified by anyone.
- Community: PHP has a large and active community of developers who contribute to the development of the language, provide support, and share knowledge.

PHP is commonly used in combination with the LAMP (Linux, Apache, MySQL, PHP) stack, which is a popular web development platform. It is also used in combination with other technologies such as WordPress, Drupal, Magento, and Laravel to create web applications.

Visual Studio Code (VS CODE) :

Visual Studio Code (VS Code) is a popular open-source code editor developed by Microsoft. It's available for Windows, macOS, and Linux operating systems. VS Code is designed to be lightweight and highly customizable, making it a popular choice for developers across different programming languages.

Some of the key features of VS Code include:

- Built-in debugging: VS Code has a powerful debugging system that allows developers to debug their code directly within the editor.
- Extensibility: VS Code has a rich ecosystem of extensions that allow developers to customize the editor to fit their specific needs.
- Git integration: VS Code has built-in Git support, which makes it easy to manage and version control code projects.
- IntelliSense: VS Code's IntelliSense feature provides code completion, syntax highlighting, and other helpful suggestions as you write code.
- Command Palette: VS Code's Command Palette provides a quick and easy way to access various editor features and settings.

Overall, Visual Studio Code is a highly popular and versatile code editor that can be used for a wide range of programming languages and development scenarios.

Advantages of VS CODE :

Visual Studio Code is a popular, lightweight, and versatile code editor that offers several advantages for developers. Here are some of the key advantages of using Visual Studio Code:

- Cross-Platform Support: Visual Studio Code is available on all major operating systems, including Windows, Linux, and macOS, making it an excellent choice for developers working on different platforms.
- Wide Language Support: Visual Studio Code supports a wide range of programming languages, including popular languages like JavaScript, Python, and Java, as well as less commonly used languages like R and TypeScript.
- Rich Extensions Ecosystem: Visual Studio Code has a rich ecosystem of extensions that provide additional features and functionality to the editor.

This means that developers can customize their coding experience to suit their specific needs.

- Integrated Debugging: Visual Studio Code comes with built-in debugging tools that allow developers to easily debug their code without having to switch to a different tool.
- Git Integration: Visual Studio Code has built-in Git integration, which makes it easy for developers to manage their code repositories and collaborate with other team members.
- Intuitive User Interface: Visual Studio Code has a clean and intuitive user interface that makes it easy for developers to navigate and use the tool effectively.
- Free and Open Source: Visual Studio Code is free to download and use, and it is also an open-source tool. This means that developers can contribute to the tool's development and help improve its functionality.

3.3. Database Description

SOL:

SQL is a short-form of the structured query language, and it is pronounced as S-Q-L or sometimes as See- Quell. This database language is mainly designed for maintaining the data in relational database management systems. It is a special tool used by data professionals for handling structured data (data which is stored in the form of tables). It is also designed for stream processing in RDSMS. You can easily create and manipulate the database, access and modify the table rows and columns, etc. This query language became the standard of ANSI in the year of 1986 and ISO in the year of 1987. If you want to get a job in the field of data science, then it is the most important query language to learn. Big enterprises like

Facebook, Instagram, and LinkedIn, use SQL for storing the data in the back-end. Why SQL? Nowadays, SQL is widely used in data science and analytics.

Following are the reasons which explain why it is widely used:

- The basic use of SQL for data professionals and SQL users is to insert, update, and delete the data from the relational database.
- SQL allows the data professionals and users to retrieve the data from the relational database management systems.
- It also helps them to describe the structured data. It allows SQL users to create, drop, and manipulate the database and its tables.
- It also helps in creating the view, stored procedure, and functions in the relational database.
- It allows you to define the data and modify that stored data in the relational database.
- It also allows SQL users to set the permissions or constraints on table columns, views, and stored procedures. MySQL MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server Oracle
- Database.

It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications. It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. The official pronunciation of MySQL is not the My Sequel; it is My Ess Que Ell. However, you can pronounce it in your way. Many small and big companies use MySQL.

MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

- It allows us to implement database operations on tables, rows, columns, and indexes. It defines the database relationship in the form of tables (collection of rows and
- columns)
- also known as relations.
- It provides the Referential Integrity between rows or columns of various tables.
- It allows us to updates the table indexes automatically.
- It uses many SQL queries and combines useful information from multiple tables for the end-users

How MySQL Works? MySQL follows the working of Client-Server Architecture. This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services. Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched. The process of MySQL environment is the same as the client- server model. The core of the MySQL database is the MySQL Server. This server is available as a separate program and responsible for handling all the database instructions, statements, or commands.

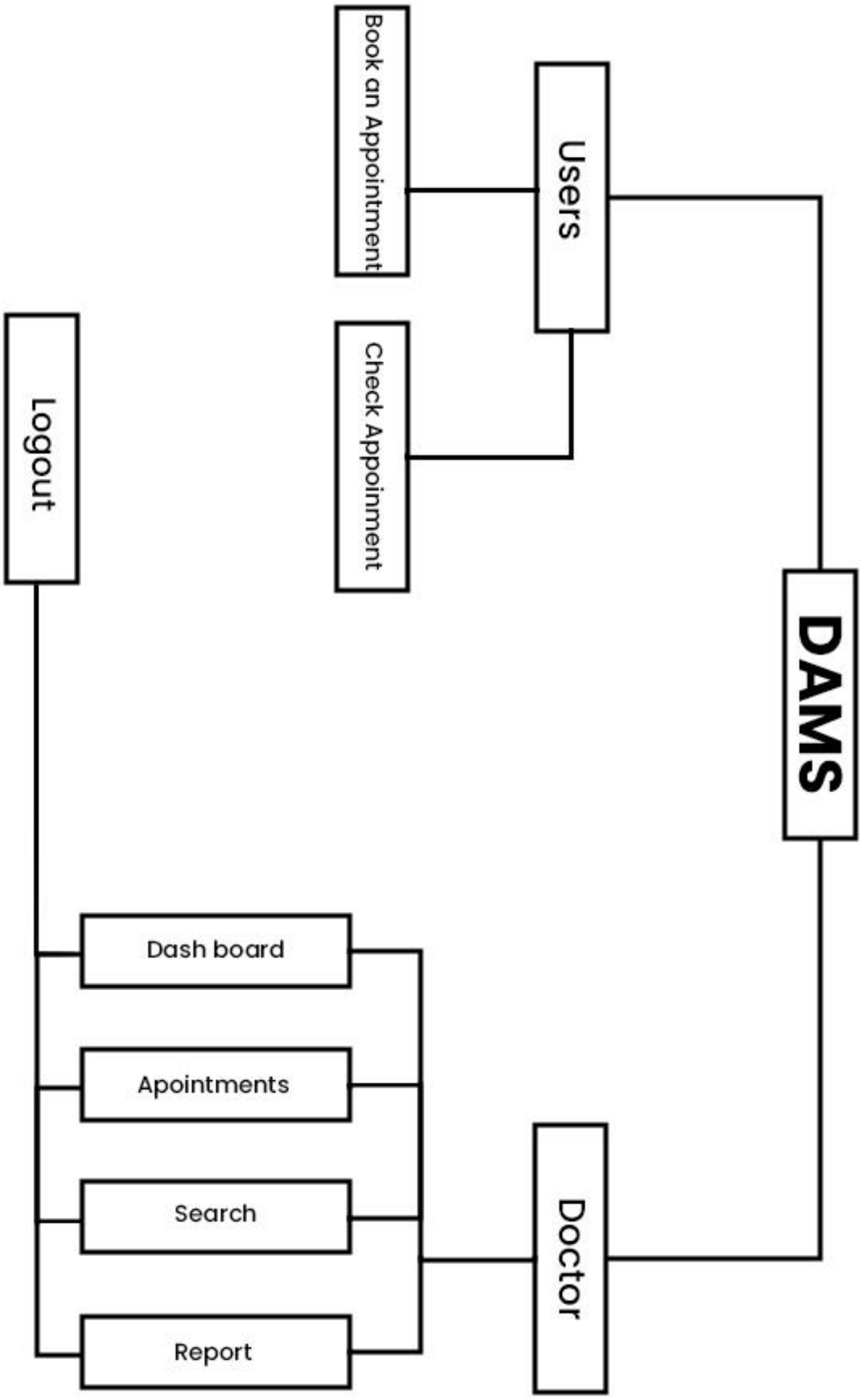
The working of MySQL database with MySQL Server are as follows:

- MySQL creates a database that allows you to build many tables to store and manipulate data and defining the relationship between each table.

- Clients make requests through the GUI screen or command prompt by using specific SQL expressions on MySQL.
- Finally, the server application will respond with the requested expressions and produce the desired result on the client-side.

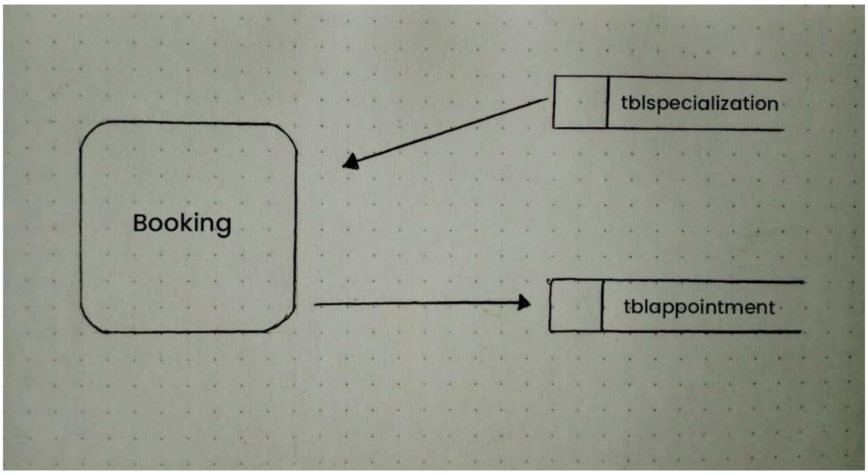
4. SYSTEM DESIGN

4.1 STRUCTURE CHART

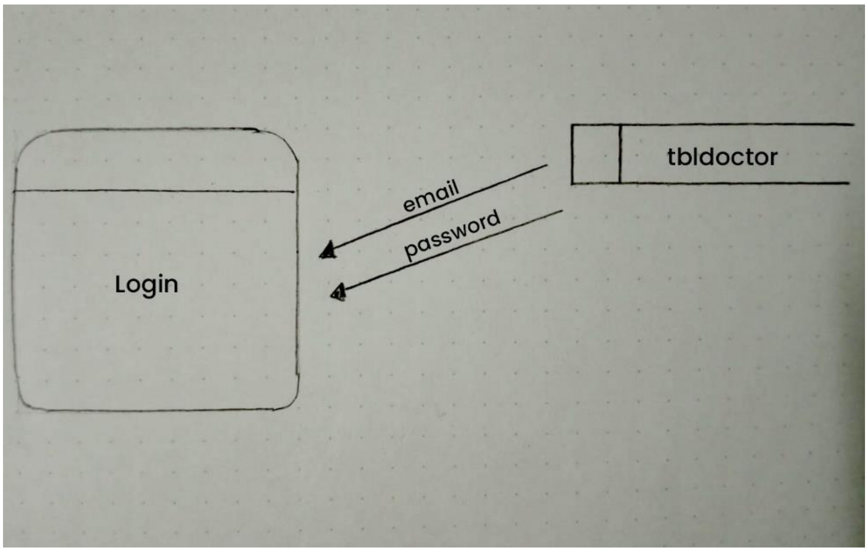


4.2 DATA FLOW CHART

Booking :



Admin Login:

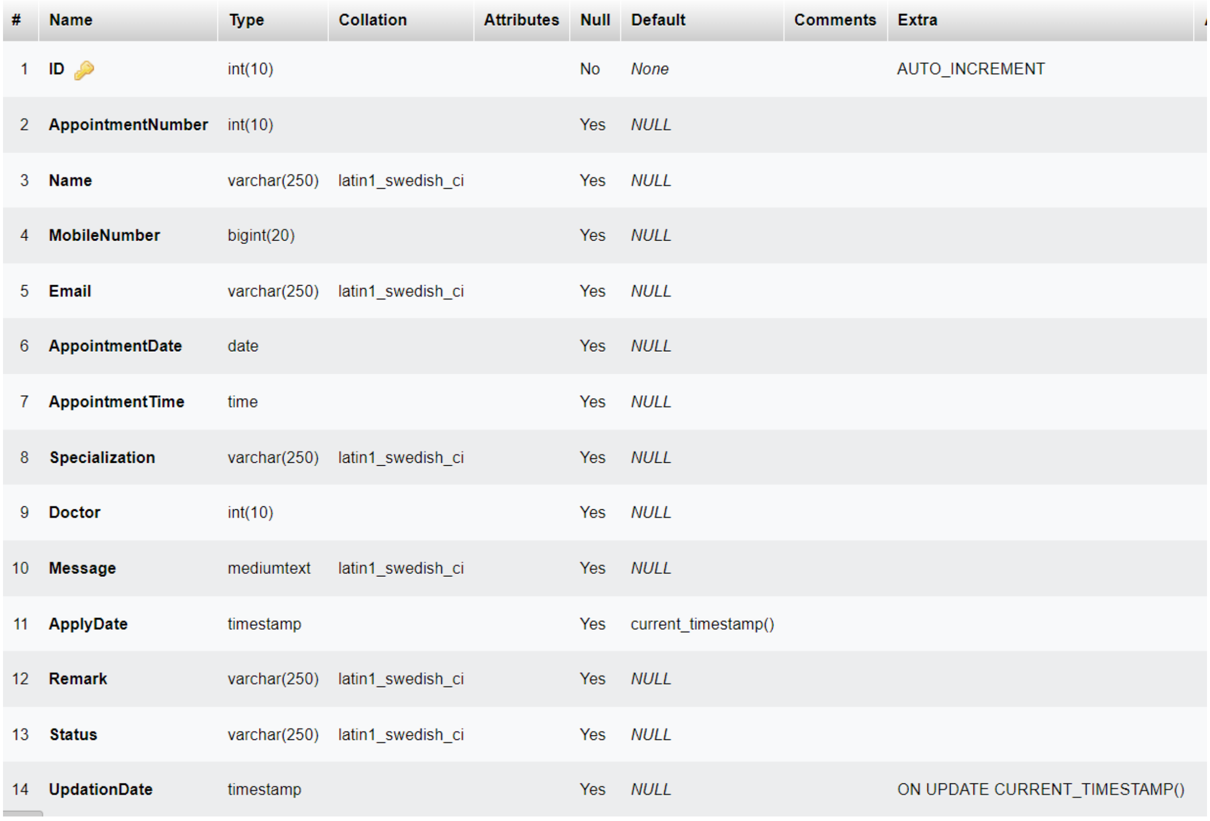


4.3 DATABASE DESIGN

It is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition language, which then can be used to create a database.

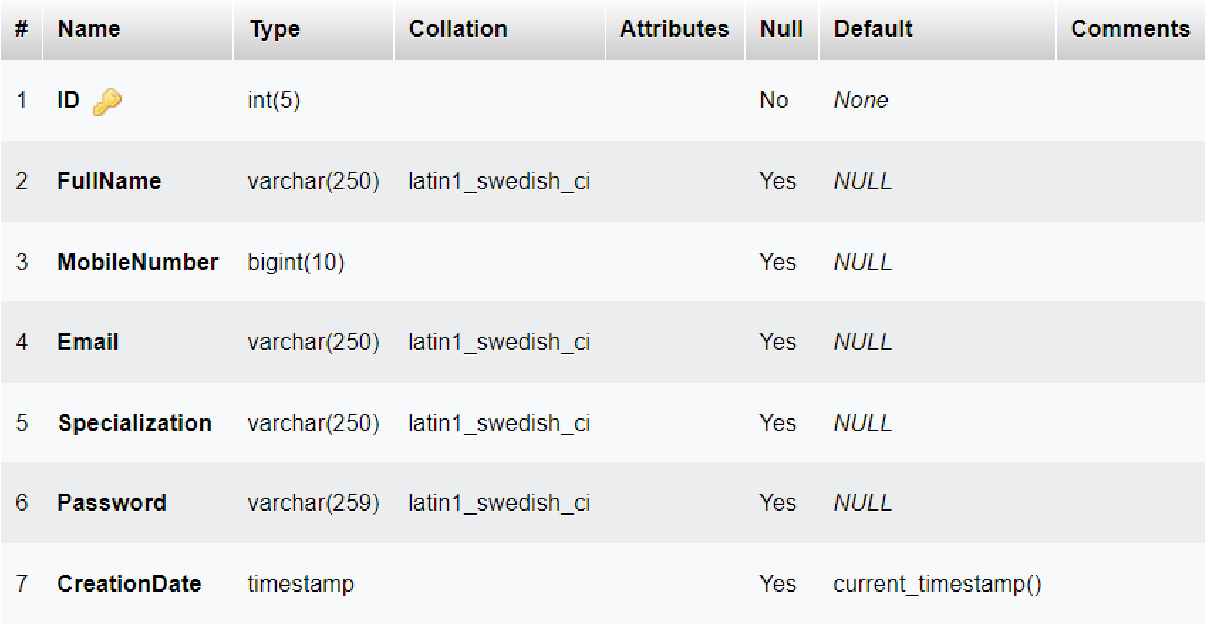
Database name : damsmsdb

Table name : tblappointment



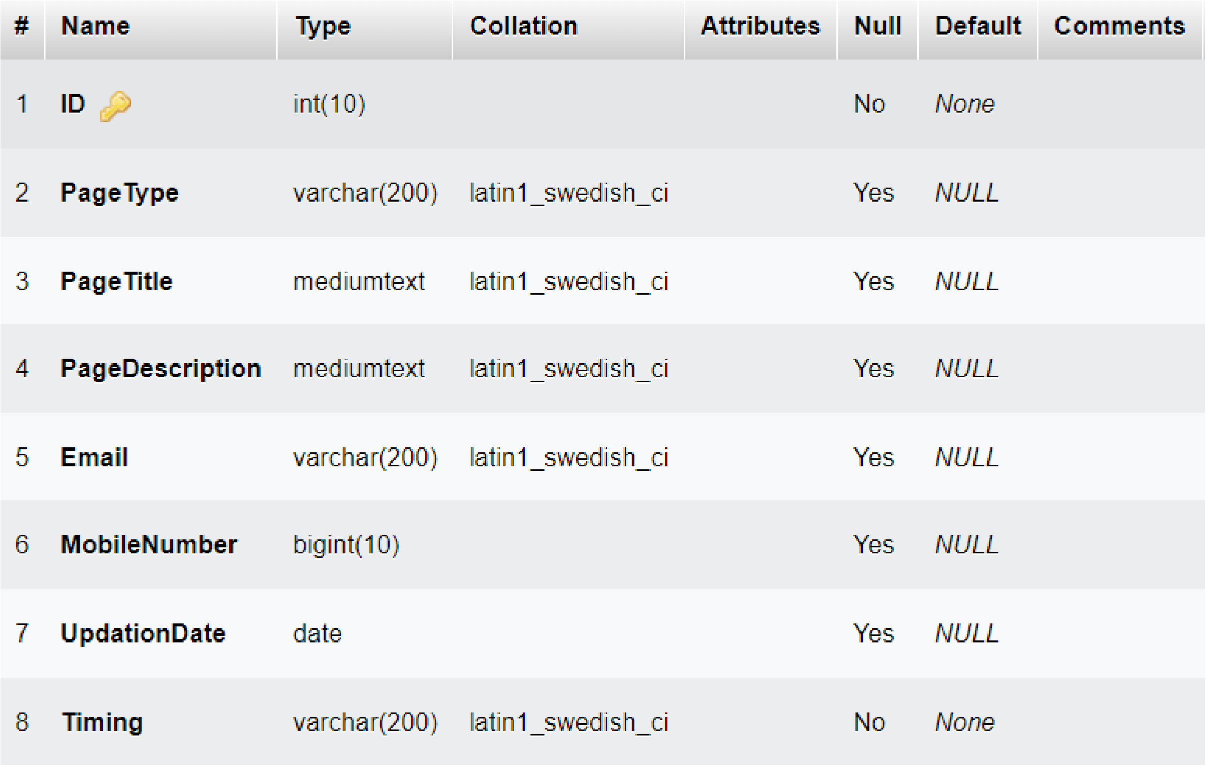
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID**	int(10)			No	*None*		AUTO_INCREMENT
2	**AppointmentNumber**	int(10)			Yes	*NULL*		
3	**Name**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
4	**MobileNumber**	bigint(20)			Yes	*NULL*		
5	**Email**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
6	**AppointmentDate**	date			Yes	*NULL*		
7	**AppointmentTime**	time			Yes	*NULL*		
8	**Specialization**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
9	**Doctor**	int(10)			Yes	*NULL*		
10	**Message**	mediumtext	latin1_swedish_ci		Yes	*NULL*		
11	**ApplyDate**	timestamp			Yes	current_timestamp()		
12	**Remark**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
13	**Status**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
14	**UpdationDate**	timestamp			Yes	*NULL*		ON UPDATE CURRENT_TIMESTAMP()

Table name : tbldoctor



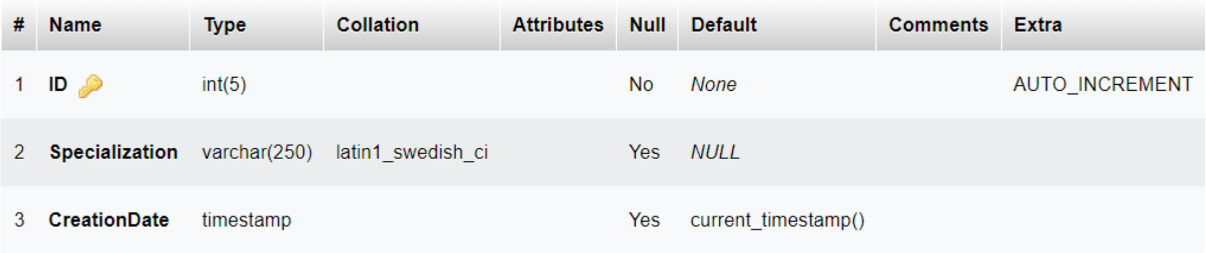
#	Name	Type	Collation	Attributes	Null	Default	Comments
1	**ID** 🔑	int(5)			No	*None*	
2	**FullName**	varchar(250)	latin1_swedish_ci		Yes	*NULL*	
3	**MobileNumber**	bigint(10)			Yes	*NULL*	
4	**Email**	varchar(250)	latin1_swedish_ci		Yes	*NULL*	
5	**Specialization**	varchar(250)	latin1_swedish_ci		Yes	*NULL*	
6	**Password**	varchar(259)	latin1_swedish_ci		Yes	*NULL*	
7	**CreationDate**	timestamp			Yes	current_timestamp()	

Table name : tblpage



#	Name	Type	Collation	Attributes	Null	Default	Comments
1	**ID** 🔑	int(10)			No	*None*	
2	**PageType**	varchar(200)	latin1_swedish_ci		Yes	*NULL*	
3	**PageTitle**	mediumtext	latin1_swedish_ci		Yes	*NULL*	
4	**PageDescription**	mediumtext	latin1_swedish_ci		Yes	*NULL*	
5	**Email**	varchar(200)	latin1_swedish_ci		Yes	*NULL*	
6	**MobileNumber**	bigint(10)			Yes	*NULL*	
7	**UpdationDate**	date			Yes	*NULL*	
8	**Timing**	varchar(200)	latin1_swedish_ci		No	*None*	

Table name : tblspecialization



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	**ID** 🔑	int(5)			No	*None*		AUTO_INCREMENT
2	**Specialization**	varchar(250)	latin1_swedish_ci		Yes	*NULL*		
3	**CreationDate**	timestamp			Yes	current_timestamp()		

6.SYSTEM DESCRIPTION

A web-based doctor appointment management system is a software application designed to facilitate the scheduling and management of appointments between patients and medical practitioners. The system provides a user-friendly interface for patients to schedule, reschedule or cancel appointments, view their medical history, and receive reminders. Additionally, the system provides features that allow medical practitioners to manage their schedules, track patient history, and receive alerts for appointments and cancellations. The system typically includes a database that stores patient and practitioner information, including medical history and appointment schedules. When a patient schedules an appointment, the system checks for availability of the chosen practitioner and books the appointment if available. If the chosen practitioner is unavailable, the system suggests alternative practitioners based on the patient's requirements. The system can also provide the option for patients to select a preferred time slot or practitioner based on previous appointments. The system provides notifications to patients and practitioners regarding upcoming appointments and cancellations. Patients receive notifications through email, text messages, or push notifications while practitioners receive notifications through email or through the application. The system also provides options for patients to request prescription refills, send messages to their practitioners, and view their medical records. The system allows medical practitioners to manage their schedules, view patient records, and update patient information. Practitioners can also view their schedules in real-time and make necessary changes. Additionally, the system provides practitioners with the ability to review patient medical records, create new records, and update existing records.

7. SYSTEM IMPLEMENTATION

A web-based doctor appointment management system can be implemented by following a few simple steps. First, a team of developers and designers should be assembled to create the system. They should work together to design a user-friendly interface that allows patients to easily schedule and manage appointments with their doctor. This interface should include features such as a calendar that displays available appointment times, the ability to cancel or reschedule appointments, and a notification system that reminds patients of upcoming appointments. Next, the team should develop the backend of the system. This will involve setting up a database to store patient information, appointment details, and other relevant data. The system should be designed to be scalable, so that it can handle a large number of users and appointments. Once the backend is complete, the team can start testing the system to ensure that it is working correctly. This testing should involve both automated and manual testing to identify any bugs or issues that need to be addressed. After testing is complete, the system can be launched. Patients should be informed of the new system and how to use it to schedule appointments with their doctor. The system should be monitored closely after launch to ensure that it is working correctly and to identify any issues that need to be addressed. Overall, implementing a web-based doctor appointment management system requires careful planning and execution. By following these steps, a team of developers and designers can create a system that is easy to use, scalable, and reliable.

8. USER MANUAL

Welcome to the user manual for the web-based doctor appointment management system. This system has been designed to simplify the process of scheduling and managing appointments for doctors and their patients. Here is a step-by-step guide to using the system:

- Login (admin): To access the system, you need to login with your username and password. If you are an admin/doctor, you can create a new account by providing your basic information.
- Dashboard (admin): After logging in, you will be taken to the dashboard where you can see all the important information about your appointments list of the patient.
- Book an appointment (Patients) : To book a new appointment, click on the "Book Appointment" button. You will be taken to a page where you can select the date, time, and doctor you wish to see. After selecting these details, you can confirm your appointment.
- Doctor's availability (Patients) : You can check the availability of a doctor by clicking on the "Doctor Availability" tab on the dashboard. This will show you the days and times when the doctor is available for appointments.
- Profile (admin): You can update your profile information, such as your name, phone number, and address, by clicking on the "Profile" tab on the dashboard.

9.FUTURE ENHANCEMENT

There are several potential enhancements that could be made to a web-based doctor appointment management system to improve its functionality and user experience. Here are a few ideas:

- Integration with electronic health records (EHR): Integrating the appointment management system with an EHR would allow healthcare providers to easily access patients' medical histories and other relevant information during the appointment scheduling process. This could help to streamline the process and ensure that patients receive appropriate care.
- Automated reminders: Sending automated reminders to patients via email or text message could help reduce no-shows and improve appointment adherence. Reminders could include appointment details, directions to the office, and any pre-appointment instructions.
- Waitlist management: If a patient cancels an appointment, the system could automatically notify patients on a waitlist and offer them the newly available slot. This could help reduce appointment scheduling lead times and improve patient satisfaction.
- Real-time availability: Showing real-time availability of doctors and appointment slots could help patients schedule appointments more easily and reduce the need for back-and-forth communication with the office.
- Telemedicine integration: With the rise of telemedicine, integrating the appointment management system with a telemedicine platform could enable patients to schedule virtual appointments with their healthcare providers.
- Patient feedback: Collecting patient feedback after appointments could help healthcare providers identify areas for improvement and enhance the overall patient experience. The feedback could be integrated into the

appointment management system to help providers make data-driven decisions about their practice.

Overall, the key to enhancing a web-based doctor appointment management system is to focus on improving the patient experience and streamlining the appointment scheduling process for both patients and healthcare providers.

10.CONCLUSION

In conclusion, a web-based doctor appointment management system can provide numerous benefits to both patients and healthcare providers. The system can simplify the appointment scheduling process, reduce wait times, improve patient satisfaction, and increase the efficiency of healthcare delivery. By enabling patients to schedule appointments online and receive timely reminders, the system can also help reduce no-shows and improve overall clinic productivity. Additionally, the system can provide doctors and other healthcare providers with real-time access to patient information, allowing for better coordination of care and improved outcomes. Overall, implementing a web-based doctor appointment management system can lead to a more streamlined and effective healthcare delivery system. The system can also help reduce no-shows and cancellations, which can be costly for healthcare providers. By sending reminders and allowing patients to reschedule appointments online, the system can help reduce the number of missed appointments and improve patient outcomes.

Overall, a web-based doctor appointment management system can improve the patient experience, streamline healthcare provider operations, and ultimately lead to better patient outcomes. As technology continues to advance, it is likely that more healthcare providers will adopt these systems to improve patient care and efficiency in their practices.

11.APPENDIX

11.1 Bibliography

Reference websites :

- https://www.tutorialspoint.com/index.htm
- https://www.w3schools.com/php/
- https://www.hackerrank.com/skills-directory/php_basic
- https://github.com

Reference Book :

- "PHP and MySQL Web Development" by Luke Welling and Laura Thomson
- "Learning PHP, MySQL & JavaScript" by Robin Nixon
- "PHP Objects, Patterns, and Practice" by Matt Zandstra
- "PHP Programming with MySQL: The Web Technologies Series" by Don Gosselin

11.2 Source code

```
<?php
session_start();
//error_reporting(0);
include('doctor/includes/dbconnection.php');
   if(isset($_POST['submit']))
  {
 $name=$_POST['name'];
  $mobnum=$_POST['phone'];
 $email=$_POST['email'];
 $appdate=$_POST['date'];
 $aaptime=$_POST['time'];
 $specialization=$_POST['specialization'];
  $doctorlist=$_POST['doctorlist'];
 $message=$_POST['message'];
 $aptnumber=mt_rand(100000000, 999999999);
 $cdate=date('Y-m-d');

if($appdate<=$cdate){
      echo '<script>alert("Appointment date must be greater than todays date")</script>';
} else {
```

```
$sql="insert into tblappointment(AppointmentNumber,Name,MobileNumber,Email,AppointmentDate,AppointmentTime,Specialization,Doctor,Message)values(:aptnumber,:name,:mobnum,:email,:appdate,:aaptime,:specialization,:doctorlist,:message)";
$query=$dbh->prepare($sql);
$query->bindParam(':aptnumber',$aptnumber,PDO::PARAM_STR);
$query->bindParam(':name',$name,PDO::PARAM_STR);
$query->bindParam(':mobnum',$mobnum,PDO::PARAM_STR);
$query->bindParam(':email',$email,PDO::PARAM_STR);
$query->bindParam(':appdate',$appdate,PDO::PARAM_STR);
$query->bindParam(':aaptime',$aaptime,PDO::PARAM_STR);
$query->bindParam(':specialization',$specialization,PDO::PARAM_STR);
$query->bindParam(':doctorlist',$doctorlist,PDO::PARAM_STR);
$query->bindParam(':message',$message,PDO::PARAM_STR);

 $query->execute();
   $LastInsertId=$dbh->lastInsertId();
   if ($LastInsertId>0) {
    echo '<script>alert("Your Appointment Request Has Been Send. We Will Contact You Soon")</script>';
echo "<script>window.location.href ='index.php'</script>";
  }
  else
```

```
    {
        echo '<script>alert("Something Went Wrong. Please try again")</script>';
    }
}
}
?>
<!doctype html>
<html lang="en">
    <head>
        <title>Doctor Appointment Management System || Home Page</title>

        <!-- CSS FILES -->
        <link rel="preconnect" href="https://fonts.googleapis.com">

        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

        <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;60
0;700&display=swap" rel="stylesheet">

        <link href="css/bootstrap.min.css" rel="stylesheet">
        <link href="css/bootstrap-icons.css" rel="stylesheet">


        <link href="css/owl.carousel.min.css" rel="stylesheet">


        <link href="css/owl.theme.default.min.css" rel="stylesheet">


        <link href="css/templatemo-medic-care.css" rel="stylesheet">
        <script>
function getdoctors(val) {
  //  alert(val);
$.ajax({

type: "POST",
url: "get_doctors.php",
data:'sp_id='+val,
success: function(data){
$("#doctorlist").html(data);
}
});
}
</script>
    </head>
  <body id="top">

    <main>

      <?php include_once('includes/header.php');?>

      <section class="hero" id="hero">
        <div class="container">
          <div class="row">

            <div class="col-12">
              <div id="myCarousel" class="carousel slide carousel-fade" data-bs-ride="carousel">
                <div class="carousel-inner">
                  <div class="carousel-item active">
                    <img src="images/slider/portrait-successful-mid-adult-doctor-with-crossed-arms.jpg" class="img-fluid" alt="">
                  </div>

                  <div class="carousel-item">
```

```
                                <img src="images/slider/young-asian-female-dentist-white-coat-posing-clinic-equipment.jpg" class="img-fluid" alt="">
                            </div>

                            <div class="carousel-item">
                                <img src="images/slider/doctor-s-hand-holding-stethoscope-closeup.jpg" class="img-fluid" alt="">
                            </div>
                        </div>
                    </div>

                </div>

            </div>
        </div>
    </section>

    <section class="section-padding" id="about">
        <div class="container">
            <div class="row">

                <div class="col-lg-6 col-md-6 col-12">
```

```
                    <?php
$sql="SELECT * from tblpage where PageType='aboutus'";
$query = $dbh -> prepare($sql);
$query->execute();
$results=$query->fetchAll(PDO::FETCH_OBJ);

$cnt=1;
if($query->rowCount() > 0)
{
foreach($results as $row)
{            ?>
                    <h2 class="mb-lg-3 mb-3"><?php  echo htmlentities($row->PageTitle);?></h2>

                    <p><?php  echo ($row->PageDescription);?>.</p>

                   <?php $cnt=$cnt+1;}} ?>
                 </div>

                 <div class="col-lg-4 col-md-5 col-12 mx-auto">
                    <div class="featured-circle bg-white shadow-lg d-flex justify-content-center align-items-center">
```

```
                                <p class="featured-text"><span class="featured-number">12</span> Years<br> of Experiences</p>
                            </div>
                        </div>

                    </div>
                </div>
            </section>

            <section class="gallery">
                <div class="container">
                    <div class="row">

                        <div class="col-lg-6 col-6 ps-0">
                            <img src="images/gallery/medium-shot-man-getting-vaccine.jpg" class="img-fluid galleryImage" alt="get a vaccine" title="get a vaccine for yourself">
                        </div>

                        <div class="col-lg-6 col-6 pe-0">
                            <img src="images/gallery/female-doctor-with-presenting-hand-gesture.jpg" class="img-fluid galleryImage" alt="wear a mask" title="wear a mask to protect yourself">
                        </div>


                    </div>

                </div>

            </section>











            <section class="section-padding" id="booking">

                <div class="container">

                    <div class="row">


                        <div class="col-lg-8 col-12 mx-auto">

                            <div class="booking-form">


                                <h2 class="text-center mb-lg-3 mb-2">Book an
        appointment</h2>


                                <form role="form" method="post">

                                    <div class="row">
                                <div class="col-lg-6 col-12">
                                    <input type="text" name="name" id="name" class="form-control" placeholder="Full name" required='true'>
                                </div>


                                <div class="col-lg-6 col-12">
                                    <input type="email" name="email" id="email" pattern="[^ @]*@[^ @]*" class="form-control" placeholder="Email address" required='true'>
                                </div>


                                <div class="col-lg-6 col-12">
                                    <input type="telephone" name="phone" id="phone" class="form-control" placeholder="Enter Phone Number" maxlength="10">
                                </div>


                                <div class="col-lg-6 col-12">
                                    <input type="date" name="date" id="date" value="" class="form-control">


                                </div>


                                    <div class="col-lg-6 col-12">
```

```
                                        <input type="time" name="time" id="time"
value="" class="form-control">

                                    </div>

   <div class="col-lg-6 col-12">
<select onChange="getdoctors(this.value);"  name="specialization"
id="specialization" class="form-control" required>
<option value="">Select specialization</option>
<!--- Fetching States--->
<?php
$sql="SELECT * FROM tblspecialization";
$stmt=$dbh->query($sql);
$stmt->setFetchMode(PDO::FETCH_ASSOC);
while($row =$stmt->fetch()) {
  ?>
<option value="<?php echo $row['ID'];?>"><?php echo
$row['Specialization'];?></option>
<?php }?>
</select>
</div>
```

```
    <div class="col-lg-6 col-12">
<select name="doctorlist" id="doctorlist" class="form-control">
<option value="">Select Doctor</option>
</select>
</div>


                                        <div class="col-12">
                                            <textarea class="form-control" rows="5"
id="message" name="message" placeholder="Additional Message"></textarea>
                                        </div>

                                        <div class="col-lg-3 col-md-4 col-6 mx-auto">
                                            <button type="submit" class="form-control"
name="submit" id="submit-button">Book Now</button>
                                        </div>
                                    </div>
                                </form>


                            </div>
                        </div>
```

```
          </div>
        </div>
      </section>
    </main>
    <?php include_once('includes/footer.php');?>
    <!-- JAVASCRIPT FILES -->
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.bundle.min.js"></script>
    <script src="js/owl.carousel.min.js"></script>
    <script src="js/scrollspy.min.js"></script>
    <script src="js/custom.js"></script>
  </body>
</html>
<?php
include('doctor/includes/dbconnection.php');
if(!empty($_POST["sp_id"]))
{
$spid=$_POST["sp_id"];
$sql=$dbh->prepare("SELECT * FROM tbldoctor WHERE
Specialization=:spid");
$sql->execute(array(':spid' => $spid));
?>
<option value="">Select Doctor</option>

<?php

while($row =$sql->fetch())

{

?>

<option value="<?php echo $row["ID"]; ?>"><?php echo $row["FullName"];
?></option>

<?php

}

}

?>

<?php

session_start();

//error_reporting(0);

include('doctor/includes/dbconnection.php');


?>

<!doctype html>

<html lang="en">

    <head>

        <title>Doctor Appointment Management System || Home Page</title>
        <!-- CSS FILES -->
        <link rel="preconnect" href="https://fonts.googleapis.com">

        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

        <link
href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@300;400;600;700&display=swap" rel="stylesheet">

        <link href="css/bootstrap.min.css" rel="stylesheet">

        <link href="css/bootstrap-icons.css" rel="stylesheet">

        <link href="css/owl.carousel.min.css" rel="stylesheet">

        <link href="css/owl.theme.default.min.css" rel="stylesheet">

        <link href="css/templatemo-medic-care.css" rel="stylesheet">
        <script>
function getdoctors(val) {
    alert(val);
$.ajax({
```

```
type: "POST",

url: "get_doctors.php",

data:'sp_id='+val,

success: function(data){

$("#doctorlist").html(data);

}

});

}

</script>

   </head>


   <body id="top">


      <main>


         <?php include_once('includes/header.php');?>
      <section class="section-padding" id="booking">
        <div class="container">
          <div class="row">

            <div class="col-lg-12 col-12 mx-auto">
              <div class="booking-form">

                <h2 class="text-center mb-lg-3 mb-2">Search Appointment
History by Appointment Number/Name/Mobile No</h2>

                <form role="form" method="post">
                  <div class="row">
                    <div class="col-lg-6 col-12">
                      <input id="searchdata" type="text"
name="searchdata" required="true" class="form-control"
placeholder="Appointment No./Name/Mobile No.">
                    </div>

                    <div class="col-lg-3 col-md-4 col-6 mx-auto">
                      <button type="submit" class="form-control"
name="search" id="submit-button">Check</button>
```

```
                                        </div>
                                    </div>
                                </form>


                            </div>
                            <?php
if(isset($_POST['search']))
{


$sdata=$_POST['searchdata'];
  ?>
  <h4 align="center">Result against "<?php echo $sdata;?>" keyword </h4>


                     <div class="widget-body">
                         <div class="table-responsive">
                             <table class="table table-bordered table-hover js-basic-
example dataTable table-custom">
                                 <thead>
                                     <tr>
                                         <th>S.No</th>
                                         <th>Appointment Number</th>
                                         <th>Patient Name</th>
```

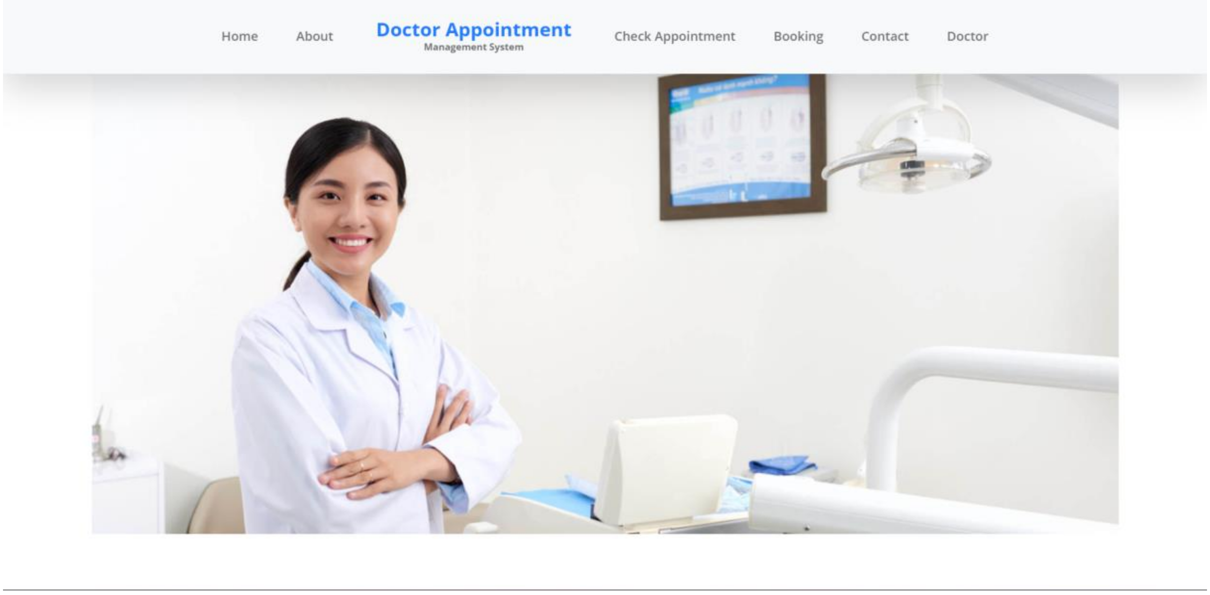
```
                    <th>Mobile Number</th>

                    <th>Email</th>

                  <th>Status</th>

                    <th>Remark</th>


                  </tr>

                </thead>


                <tbody>

         <?php


$sql="SELECT * from tblappointment where AppointmentNumber like
'$sdata%' || Name like '$sdata%' || MobileNumber like '$sdata%'";

$query = $dbh -> prepare($sql);

$query->execute();

$results=$query->fetchAll(PDO::FETCH_OBJ);


$cnt=1;

if($query->rowCount() > 0)

{

foreach($results as $row)

{           ?>
```

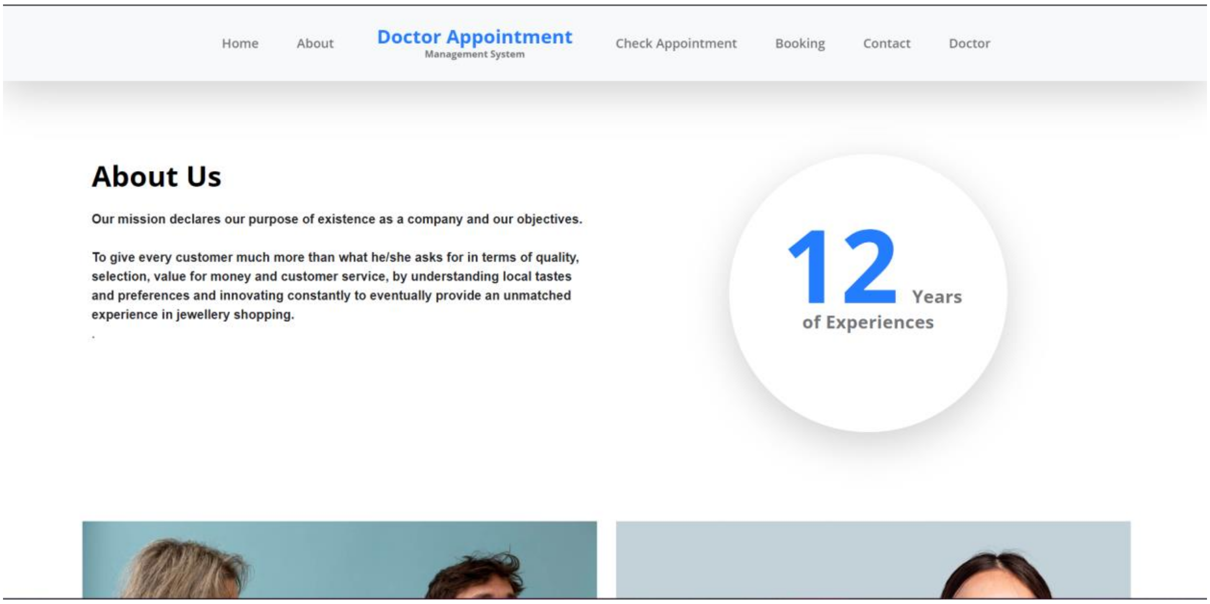
```
                              <tr>

                                <td><?php echo htmlentities($cnt);?></td>

                                <td><?php  echo htmlentities($row-
>AppointmentNumber);?></td>

                                <td><?php  echo htmlentities($row->Name);?></td>

                                <td><?php  echo htmlentities($row-
>MobileNumber);?></td>

                                <td><?php  echo htmlentities($row->Email);?></td>

                                <?php if($row->Status==""){ ?>


                   <td><?php echo "Not Updated Yet"; ?></td>

<?php } else { ?>                   <td><?php  echo htmlentities($row->Status);?>

                </td>

                <?php } ?>


                                <?php if($row->Remark==""){ ?>


                   <td><?php echo "Not Updated Yet"; ?></td>

<?php } else { ?>                   <td><?php  echo htmlentities($row->Remark);?>

                </td>

                <?php } ?>
```

```
                        </tr>




                      </tbody>


               <?php
$cnt=$cnt+1;
} } else { ?>
 <tr>
   <td colspan="8"> No record found against this search</td>


 </tr>
 <?php } }?>
                    </table>
                  </div>


                </div>
              </div>
            </section>


         </main>
         <?php include_once('includes/footer.php');?>
```

```
        <!-- JAVASCRIPT FILES -->
        <script src="js/jquery.min.js"></script>
        <script src="js/bootstrap.bundle.min.js"></script>
        <script src="js/owl.carousel.min.js"></script>
        <script src="js/scrollspy.min.js"></script>
        <script src="js/custom.js"></script>
    </body>
</html>
```

11.3 Screenshots

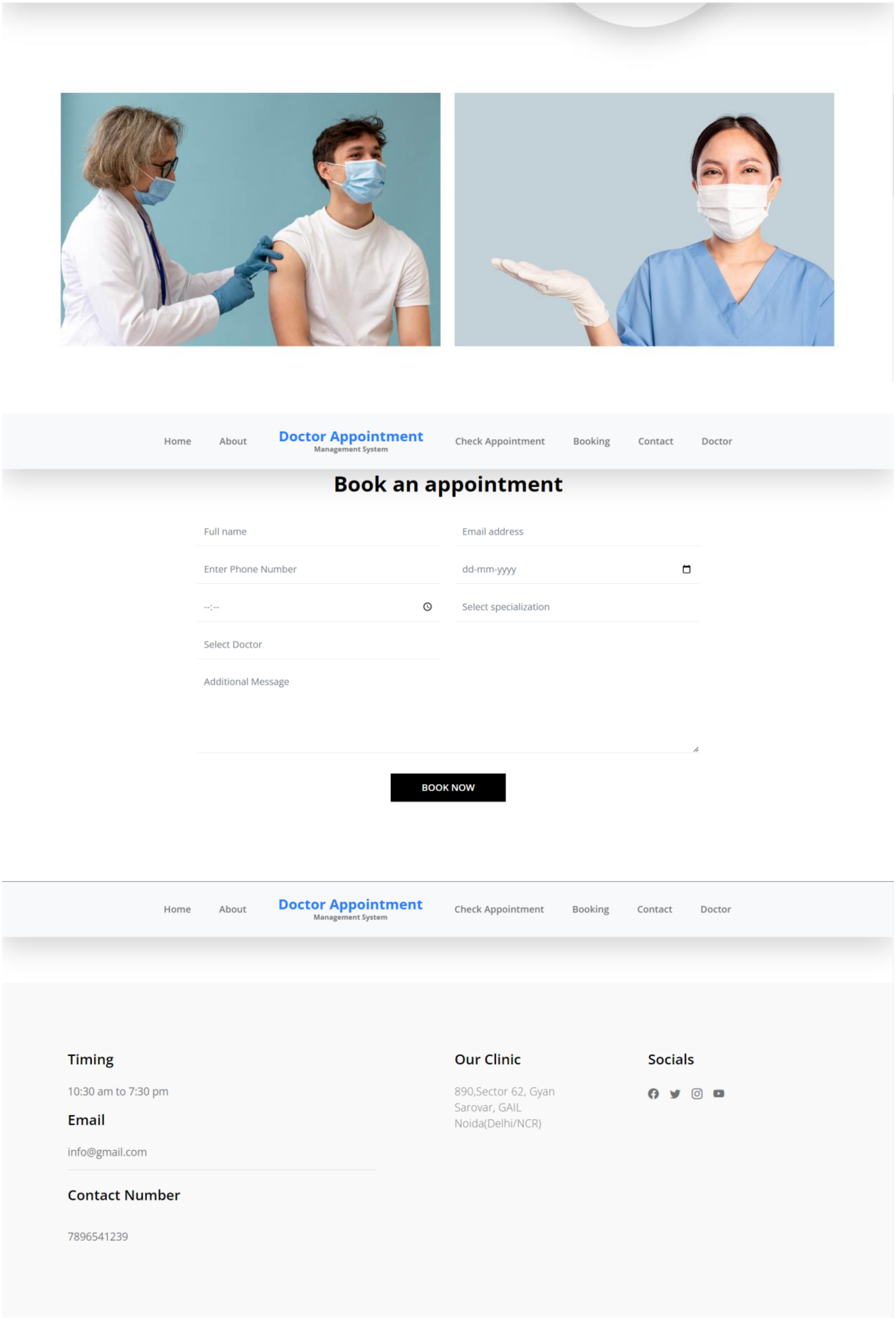




About Us

Our mission declares our purpose of existence as a company and our objectives.

To give every customer much more than what he/she asks for in terms of quality, selection, value for money and customer service, by understanding local tastes and preferences and innovating constantly to eventually provide an unmatched experience in jewellery shopping.



Book an appointment

Full name

Email address

Enter Phone Number

dd-mm-yyyy

--:--

Select specialization

Select Doctor

Additional Message

Timing

10:30 am to 7:30 pm

Email

info@gmail.com

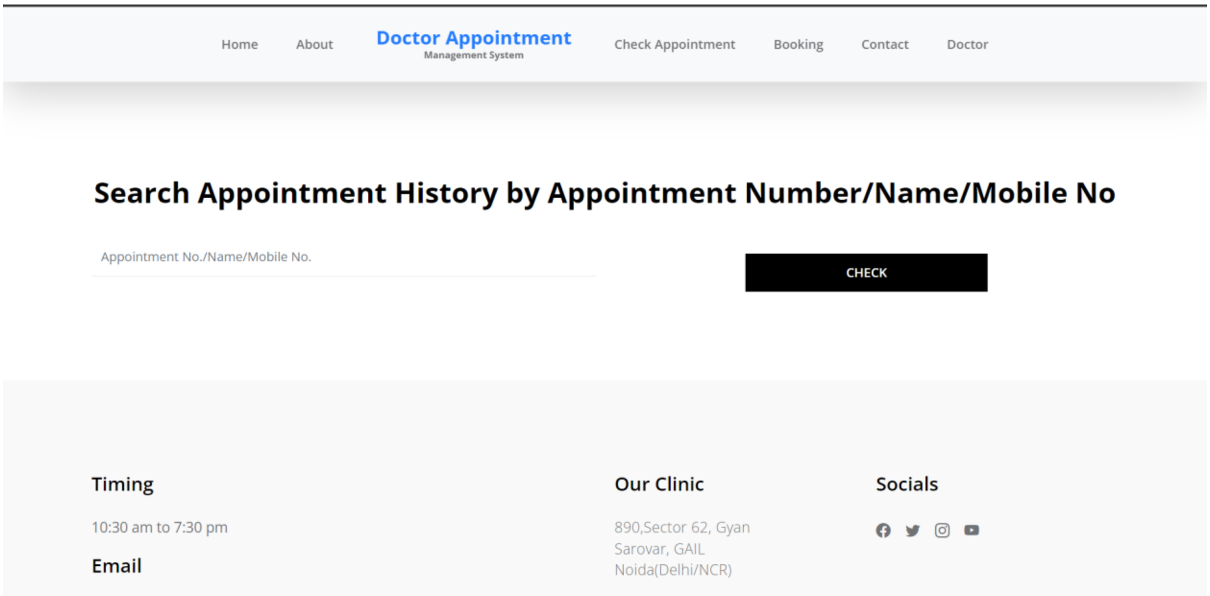
Contact Number

7896541239

Our Clinic

890,Sector 62, Gyan
Sarovar, GAIL
Noida(Delhi/NCR)

Socials



Search Appointment History by Appointment Number/Name/Mobile No

Appointment No./Name/Mobile No.

Timing

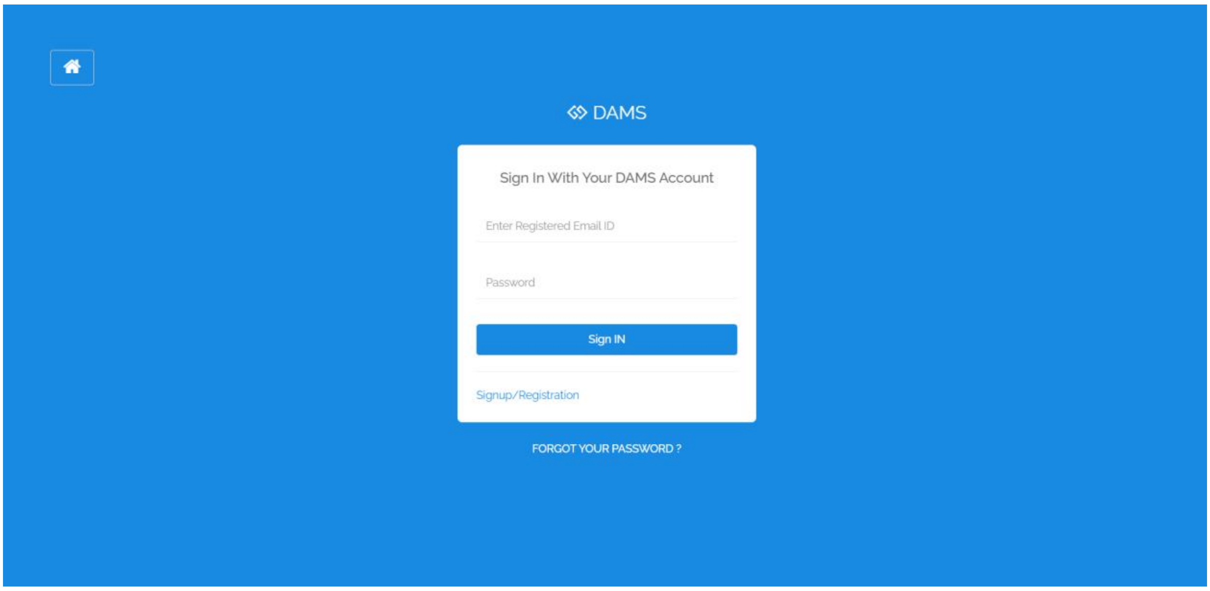
10:30 am to 7:30 pm

Email

Our Clinic

890,Sector 62, Gyan
Sarovar, GAIL
Noida(Delhi/NCR)

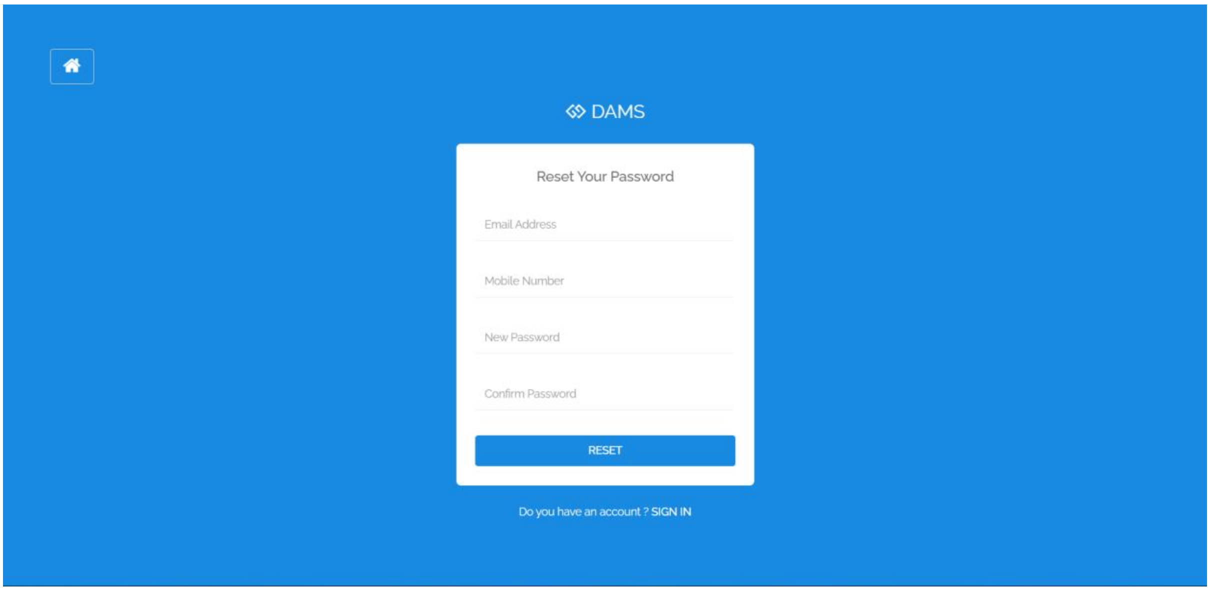
Socials



Sign In With Your DAMS Account

Enter Registered Email ID

Password



Reset Your Password

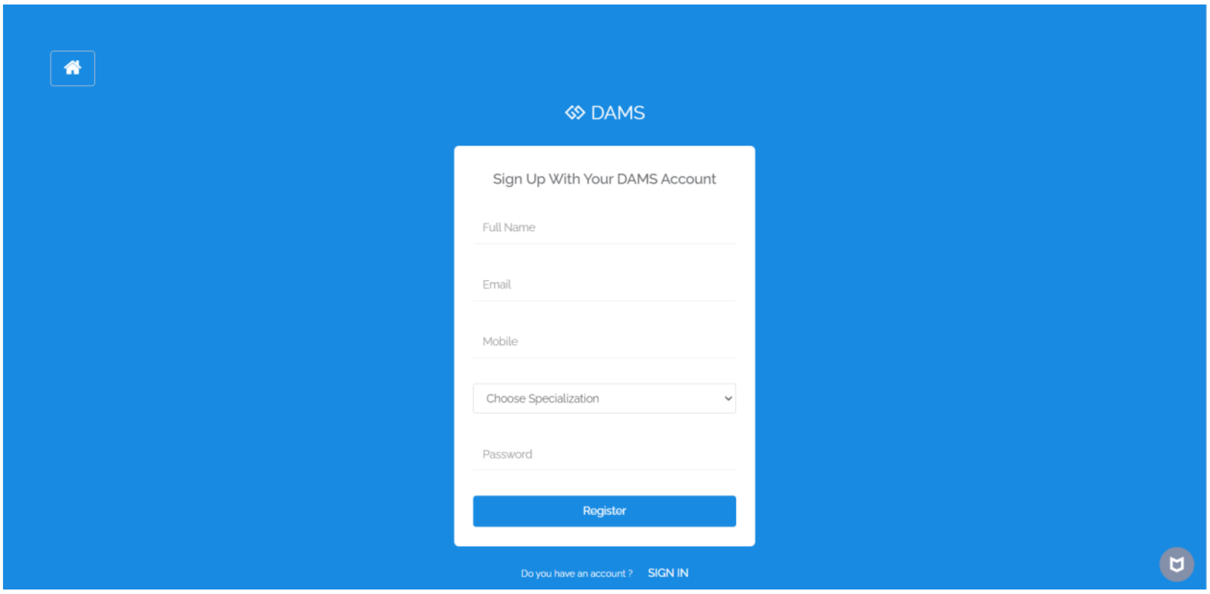
Email Address

Mobile Number

New Password

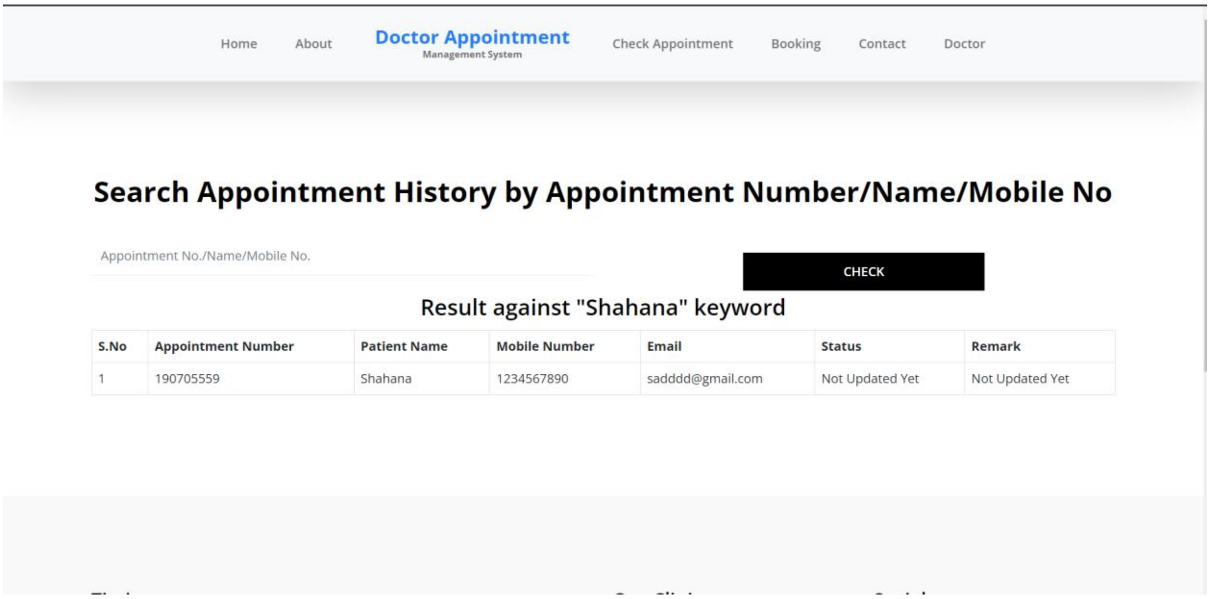
Confirm Password

Do you have an account ? SIGN IN



Sign Up With Your DAMS Account

Do you have an account ?



Search Appointment History by Appointment Number/Name/Mobile No

Result against "Shahana" keyword

S.No	Appointment Number	Patient Name	Mobile Number	Email	Status	Remark
1	190705559	Shahana	1234567890	sadddd@gmail.com	Not Updated Yet	Not Updated Yet