

# **E-MEDI APPOINTMENT BOOKING**

**A Project Report**

Submitted in partial fulfillment of the  
Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

Atharv Uttam Chavan

Seat Number – TT250006

**Under the esteemed guidance of**

**Mrs. S. S. Ketkar**

**Assistant Professor, Department of Information Technology**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

*University of Mumbai*

**R.E. Society's**

**R.P. Gogate College of Arts & Science And**

**R.V. Jogalekar College of Commerce (Autonomous), Ratnagiri.**

**RATNAGIRI, 415612**

**2024-25**

# **PROJECT PROPOSAL**

**PRN No.:2022016400785215**

**Roll No.: TTA-30**

**1. Name of the Student: Atharv Uttam Chavan.**

**2. Title of the Project: E-MEDI APPOINTMENT BOOKING**

**3. Name of the Guide: Mrs. S. S. Ketkar**

**4. Teaching experience of the Guide: 9 Years**

**5. Is this your first submission? Yes  / No**

Signature of the Student

Date: \_\_\_\_\_

Signature of the Guide

Date: \_\_\_\_\_

Signature of the Coordinator

Date: \_\_\_\_\_

*University of Mumbai*  
R.E. Society's  
R.P. Gogate College of Arts & Science And  
R.V. Jogalekar College of Commerce (Autonomous), Ratnagiri, MAHARASHTRA-415612  
**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, "**E-Medi Appointment Booking**", is bonafide work of **Atharv Uttam Chavan** bearing seat No:**TT250006** submitted in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY** from Gogate Jogalekar College (Autonomous), Ratnagiri.

**Internal Guide**

**Coordinator / HOD**

**External Examiner**

**Date:**

**College Seal**

## **ABSTRACT**

In recent years, the healthcare industry has increasingly adopted digital solutions to streamline processes and improve patient care. This paper presents an E-MEDI APPOINTMENT BOOKING web application designed to facilitate the booking and management of medical appointments. The system aims to enhance accessibility and efficiency for both patients and healthcare providers.

The project, "E-MEDI APPOINTMENT BOOKING," is a web-based application that allows patients to easily book their appointments at a particular clinic for their health issues. The current system requires patients to make appointments via phone call or by physically visiting the clinic. To address this inconvenience, we decided to develop an E-MEDI APPOINTMENT BOOKING, so clients no longer need to call the clinic to book their appointments. Patients also no longer have to worry about long wait times in the clinic. This system makes it easier to book appointments online, saving a lot of time.

This web application not only simplifies the appointment booking process but also aims to reduce the administrative workload and improve overall patient satisfaction. By providing a convenient and efficient platform for managing doctor appointments, this system represents a significant advancement in healthcare service delivery.

## **ACKNOWLEDGEMENT**

Presentation, inspiration and motivation have always played a key role in a success of any venture. I would like to express our sincere gratitude to all those who contributed to the development and success of this E-MEDI APPOINTMENT BOOKING web application.

First of all, I respect and thanks to Principal of our Gogate Joglekar college (Autonomous), Ratnagiri **Dr. M. R. Sakhalkar**, Vice Principal (Science) **Dr. A. M. Kulkarni**, Coordinator **Mrs. A. A. Gharpure** as well as our Head of Department **Mrs. M. A. Sahasrabuddhe** for giving us an opportunity to do this project and provide us all support to complete this project on time. I also thanks to my guide **Mrs. S. S. Ketkar** for her enthusiasm, patience, insightful comments and information, practical advice and unceasing ideas that have always helped me tremendously. Her immense knowledge, profound experience has enabled me to complete this project successfully. Without her support and guidance, this project would not have been possible.

I would also like to extend our heartfelt thanks to our all teacher of Information Technology Department, whose dedication and support have been invaluable. Their passion for teaching and constant encouragement have inspired us to achieve our best.

Finally, we are grateful to our families and friends for their unwavering support and encouragement, which motivated us to persevere and complete this project.

# DR. ZOPE'S CLINIC

2/43, 2nd Floor, Aashrivad Complex, Maruti Mandir, Ratnagiri – 41561. (Near Maruti Mandir)

## DR. ZOPE'S CLINIC

2/43, 2nd Floor, Aashrivad Complex,  
Maruti Mandir, Ratnagiri - 415612  
(Near Maruti Mandir)

**Date:** 1st March 2025

To,  
The Head of Department,  
Information Technology Department,  
Gogate Jogalekar College (Autonomous),  
Ratnagiri.

**Subject:** Appreciation for Developing Software for Dr. Zope's Clinic

Respected Ma'am,

We at Dr. Zope's Clinic were in need of a comprehensive software solution to efficiently manage our patients, appointments, and online consultancy. It is with great pleasure that we extend our heartfelt appreciation to the students of Gogate Jogalekar College, particularly Mr. Atharv Uttam Chavan, for their outstanding contribution in developing the required software.

The software developed by the students has far exceeded our expectations in terms of functionality and usability. We commend the students for their dedication, creativity, and attention to detail in creating such a valuable tool for our clinic. Their hard work and commitment are truly commendable, and we are immensely grateful for their contribution to the growth and efficiency of our practice.

Thank you once again for the remarkable work.

Yours sincerely,

Dr. Amol Zope      Dr. Sneha Zope

## **DECLARATION**

I hereby declare that the project entitled, "**E-MEDI APPOINTMENT BOOKING**" done at **Gogate Jogalekar College (Autonomous), Ratnagiri**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge otherthan me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

**Atharv Uttam Chavan**

## TABLE OF CONTENTS

Sr. No.	Content	Page No.
1.	<b>Introduction</b>	1-3
	1.1 Background	1
	1.2 Objectives	1
	1.3 Purpose, scope and Applicability	2
2.	<b>Survey Of Technologies</b>	4-5
	2.1 Existing Systems and its limitations	4
	2.2 Proposed System	5
3.	<b>Requirement Analysis</b>	6-14
	3.1 Problem Statement	6
	3.2 Requirements Specifications	6

	3.3 Preliminary Product Description	8
	3.4 ER and UML Diagrams	9
	3.5 Planning and scheduling – Gantt Chart	14
<b>4.</b>	<b>System Design</b>	15-88
	4.1 Event Table	15
	4.2 Modules	16
	4.3 Data Design (Data Tables)	17
	4.4 User Interface Design	21
	4.5 Coding and Implementation	38
	4.6 Test Case Design	57
<b>5.</b>	<b>Conclusion</b>	62
<b>6.</b>	<b>References</b>	63

## **TABLE OF FIGURES**

<b>Sr. No.</b>	<b>Figure Name</b>	<b>Page No.</b>
1	Entity Relationship Diagram	9
2	Use Case Diagram	9-11
	2.1 Sign up/in	10
	2.2 Patient	10
	2.3 Admin/Doctor	11
3	Class Diagram	12
4	Deployment Diagram	12
5	Sequence Diagram	13
6	Gantt chart	14

# INTRODUCTION

The healthcare industry is increasingly embracing digital solutions to improve efficiency and patient care. Traditional appointment booking methods, often involving phone calls and paperwork, are time-consuming and leading to dissatisfaction among patients and healthcare providers.

To address these issues, we have developed an E-MEDI APPOINTMENT BOOKING web application. This application streamlines the appointment booking process, providing a user-friendly interface for patients to book appointments with their preferred doctors. Key features include real-time scheduling, notifications, secure messaging, and management tools for healthcare providers.

## 1.1 Background:

In today's fast-paced world, the demand for efficient healthcare services is higher than ever. Traditional methods of booking doctor appointments often involve lengthy phone calls, manual record-keeping, and significant administrative effort, leading to inefficiencies and patient dissatisfaction. These conventional systems are not only time-consuming but also prone to errors and double bookings, which can negatively impact the overall patient experience.

## 1.2 Objectives:

The primary objectives of the E-MediAppointment Booking web application are as follows:

**Enhance Accessibility:** Provide patients with an easy-to-use platform for booking, rescheduling, and cancelling appointments online, thereby reducing the need for phone calls and in-person visits.

**Improve Efficiency:** Streamline the appointment booking process for both patients and healthcare providers, minimizing administrative burdens and reducing waiting times.

**24/7 Accessibility:** Enable patients to book appointments at any time, from any location, without the need for phone calls or personally visits to the clinic.

**Integration with Healthcare Systems:** Ensure compatibility and integration with existing healthcare management systems for seamless data exchange and workflow optimization.

### **1.3 Purpose & Scope:**

#### **1.3.1 Purpose-**

The purpose of the E-MEDI APPOINTMENT BOOKING web application is to revolutionize the way healthcare appointments are managed by leveraging digital technology. This system is designed to:

- **Simplify Appointment Booking:** Provide a convenient and accessible platform for patients to schedule, reschedule, and cancel appointments with their healthcare providers, eliminating the need for phone calls and reducing administrative burdens.
- **Enhance Operational Efficiency:** Streamline the workflow for healthcare providers by automating appointment management, reducing errors, and optimizing the use of their time and resources.

#### **1.3.2 Scope-**

- Allow patients and healthcare providers to create and manage accounts.
- Implement secure authentication mechanisms, including password protection and optional multi-factor authentication.
- Enable patients to search for and book appointments with available doctors based on various criteria such as specialty, location, and availability.
- Provide options for patients to reschedule or cancel appointments.
- Provide a secure messaging platform for patients and doctors to communicate directly regarding appointment details, pre-appointment instructions, and follow-up care.
- Offer online consultancy through video calling for remote medical consultations.

### **1.3.3 Applicability –**

The **E-MEDI APPOINTMENT BOOKING** offers wide applicability in enhancing healthcare service delivery. It significantly improves patient accessibility by allowing remote booking from any location at any time, eliminating the need for physical visits or phone calls to schedule appointments. The system is available 24/7, providing patients the convenience of booking outside normal clinic hours, and it is compatible across devices like desktops, smartphones, and tablets.

Additionally, the system reduces administrative burdens by automating appointment booking and providing real-time updates to prevent schedule conflicts. This automation not only saves time for clinic staff but also enhances accuracy in appointment handling. Integration with billing systems can further simplify payment processing, creating a smoother experience for both patients and healthcare providers. Overall, the system boosts patient satisfaction by making healthcare more accessible, reducing wait times, and providing a convenient and efficient platform for managing medical appointments.

# **SURVEY OF TECHNOLOGIES**

## **2.1 Existing System and its Limitations**

### **Existing System:**

Under manual appointing system, you have to first wait in line to take appointment to doctors and wait for your time to have meet with them and discuss on your health problem. As you have to provide information and other reports many times at different places such as medicine store which is again a burden of carrying.

### **Limitations in existing system:**

- Difficulty in finding immediate or same-day appointments due to high demand and limited real-time availability.
- There is no centralized database maintenance.
- Human Error: Manual record-keeping is susceptible to human errors such as data entry mistakes, illegible handwriting, and calculation errors. These errors can lead to inaccuracies in therecords.
- Time-Consuming: Creating, updating, and maintaining manual records is a time-consuming process. It requires significant effort to input data, cross-reference information, and ensure recordsare up-to-date.
- Limited Accessibility: Physical records are often stored in specific locations, making them lessaccessible compared to digital records. This can hinder quick retrieval of information when needed.
- Storage Space: Physical records require storage space, and as the volume of records grows, so does the need for space.
- Security Risks: Manual records are vulnerable to damage, theft, or loss due to factors like fire,floods, or mishandling. Ensuring their security can be challenging.

## **2.2Proposed System**

E-MEDI APPOINTMENT BOOKING has to meet with online doctors and can also meet physically with doctor. Patient have to fill online form and create account and login through this panel patients can choose doctors and have appointment with them on their time. In this system the doctors can also view patient's records, appointments etc. It is now easy to make appointments by using this web application.

### **Advantages with existing system:**

- Patients can schedule appointments by saving time and effort.
- This system provides centralized database maintenance.
- Automation: Data entry and management are automated, reducing the risk of human errors.
- Efficiency: Restaurant management system streamlines record-keeping, saving time and effort compared to manual processes.
- Accessibility: Digital records are easily accessible.
- Storage: Requires minimal physical storage space as data is stored electronically.
- Security: Protect against loss and unauthorized access.
- Data Retrieval: Provides efficient search and retrieval functions for finding specific information.
- Analysis: Enables data analysis and reporting for better decision-making.
- Scalability: Scales effortlessly with growing data volumes.
- Environmentally Friendly: Reduces paper usage and promotes sustainability.

# **REQUIREMENT AND SYSTEM ANALYSIS**

## **3.1 Problem Statement:**

Realizing you have a health issue that needs medical attention and treatment can cause anxiety. Visiting doctor physically takes too much time in this tech era and is also sometimes disappointing and frustrating. Instead of personally visiting the doctor and describing your symptoms in order to receive treatment, seeing a doctor online for any kind of periodical consultations seems to be a great option now. Improved technology has made telemedicine easier where remote video consultations between doctor and patient are technically possible and increasingly acceptable.

## **3.2 Requirement Specification:**

- Administrator/Doctor requirements-**

1. User management (View doctor, view patients, cancel appointment.).
2. Ability for the doctors to create and maintain their profiles and login.

- Patient requirements-**

1. Ability to register as a new patient and then login.
2. Patient can book their appointment, view booking and check appointment status.

- Functional requirements**

1. Appointment Booking: Ability for patients to select the doctors and book appointments.
2. Scheduling: Real-time updating of doctor availability, support for recurring appointments, and management of waiting lists.
3. Notifications: Automated reminders and notifications to patients and doctors about upcoming appointments, cancellations, or reschedules.
4. User Authentication: Secure login and registration processes for patients and healthcare providers.
5. Online Consultancy: Provide virtual medical consultations through secure video calling.

- **Non-functional requirements**

1. Performance: System responsiveness under various user loads, ensuring minimal downtime and fast loading times.
2. Security: Data encryption, secure storage of patient information, and compliance with healthcare data protection regulations (e.g., HIPAA).
3. Reliability: Dependable system operation, with minimal errors and robust backup and recovery mechanisms.
4. Usability: Intuitive user interface design, accessibility for users with disabilities, and support for multiple devices (desktops, tablets, smartphones).

- **User requirements**

1. Operating System: Any Operating System.
2. Browser: Any Browser (Chrome, IE, Microsoft edge, Opera, etc.).
3. Internet.

- **Developer requirements-**

1. Hardware requirements:

- a. PC with minimum 8 GB RAM.

2. Software requirements:

- a. Visual Studio 2022.
- b. SSMS (SQL Server Management Studio).
- c. HTML, CSS, Bootstrap.
- d. ASP.NET

### **3.3 Preliminary Product Description:**

The system is divided into six key modules based on functionality:

1. Login Module

- The patient login module includes a login screen with a "Forgot Password" option. If a patient forgets their password, they can reset it via email.

2. Appointment Booking Module

- Patients can book appointments based on available dates and times. The system provides real-time availability of doctors to ensure convenient scheduling.

3. Manage Patient Appointment Module

- Once an appointment is booked, the patient receives a confirmation notification via email.
- When the patient arrives at the clinic on the scheduled date, the receptionist verifies and accepts the appointment in the system.

4. Online Consultancy Module

- If a patient is unable to visit the clinic physically, they can opt for an online consultation via video call.

5. Doctor/Receptionist Login Module

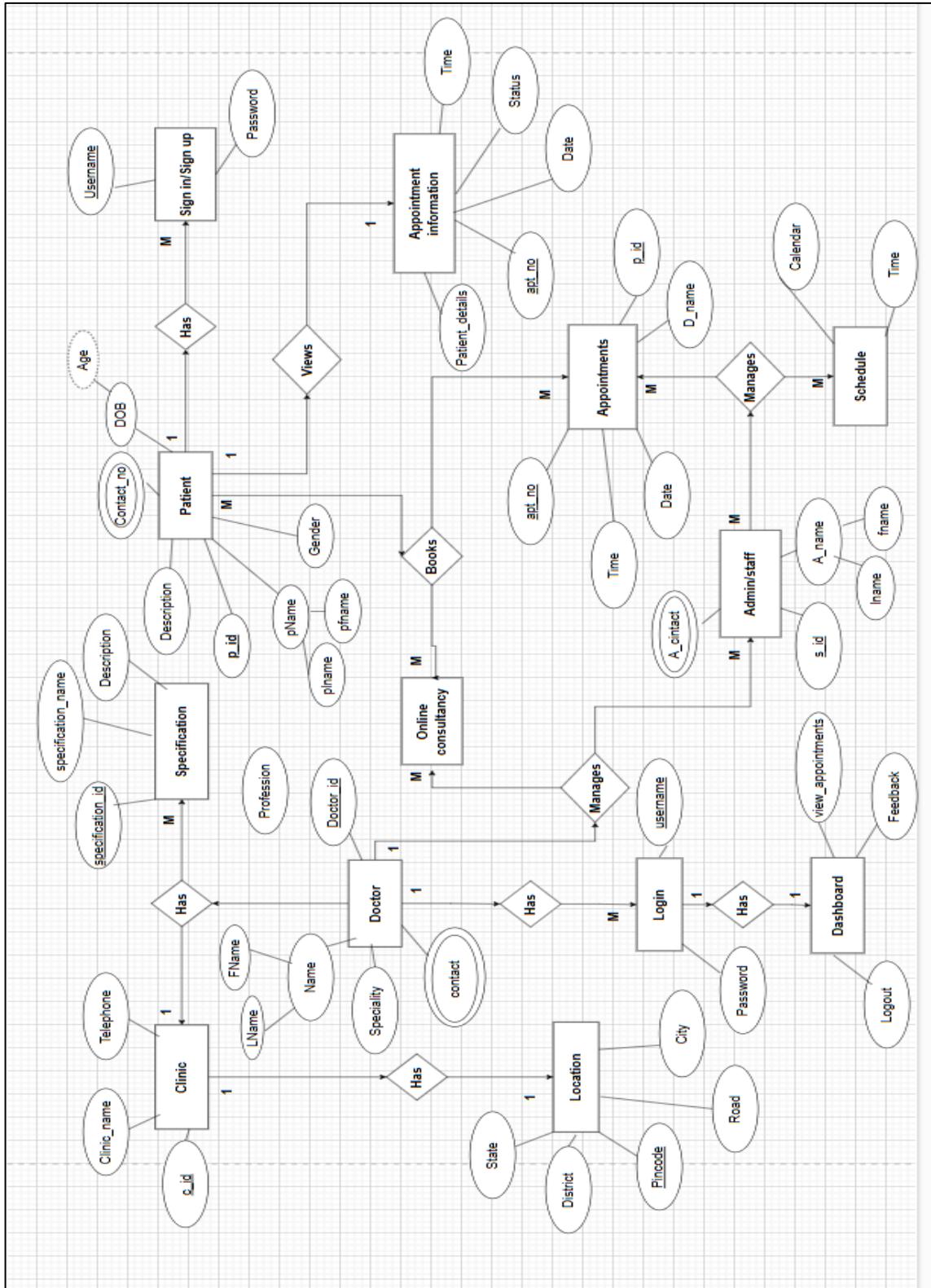
- Doctors and receptionists can log in to access their respective dashboards for managing appointments and consultations.

6. Admin Dashboard

- The admin panel allows doctors or receptionists to manage patient scheduling, send notifications, and oversee other operational activities.

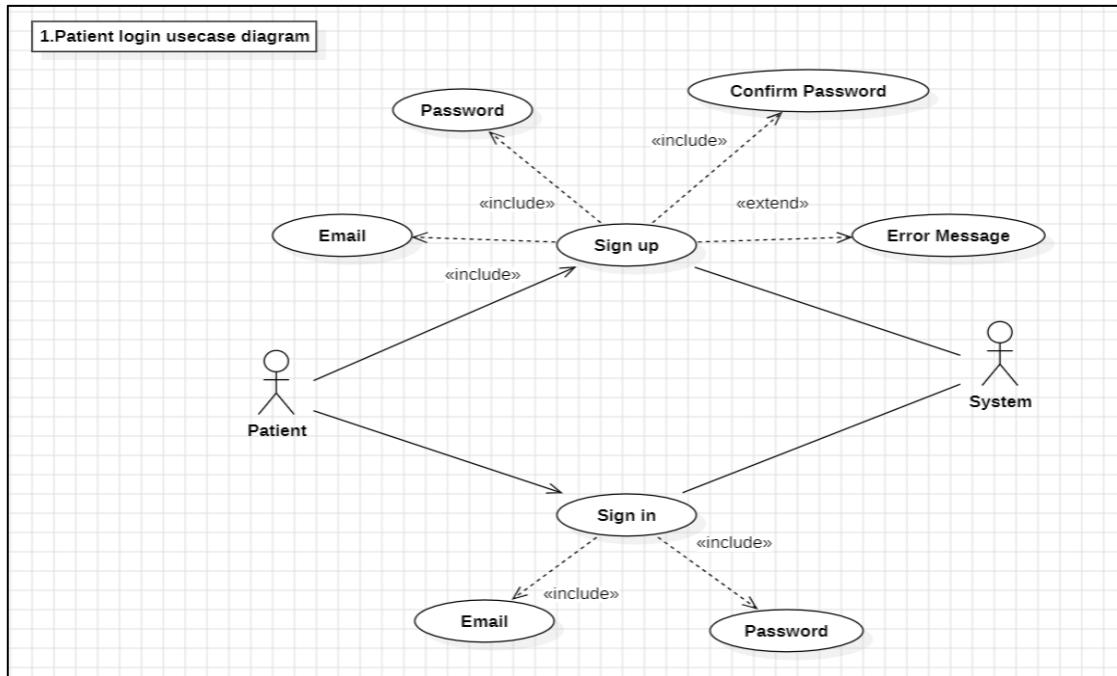
### 3.4 UML Diagrams:

#### 3.4.1 Entity Relationship Diagram-

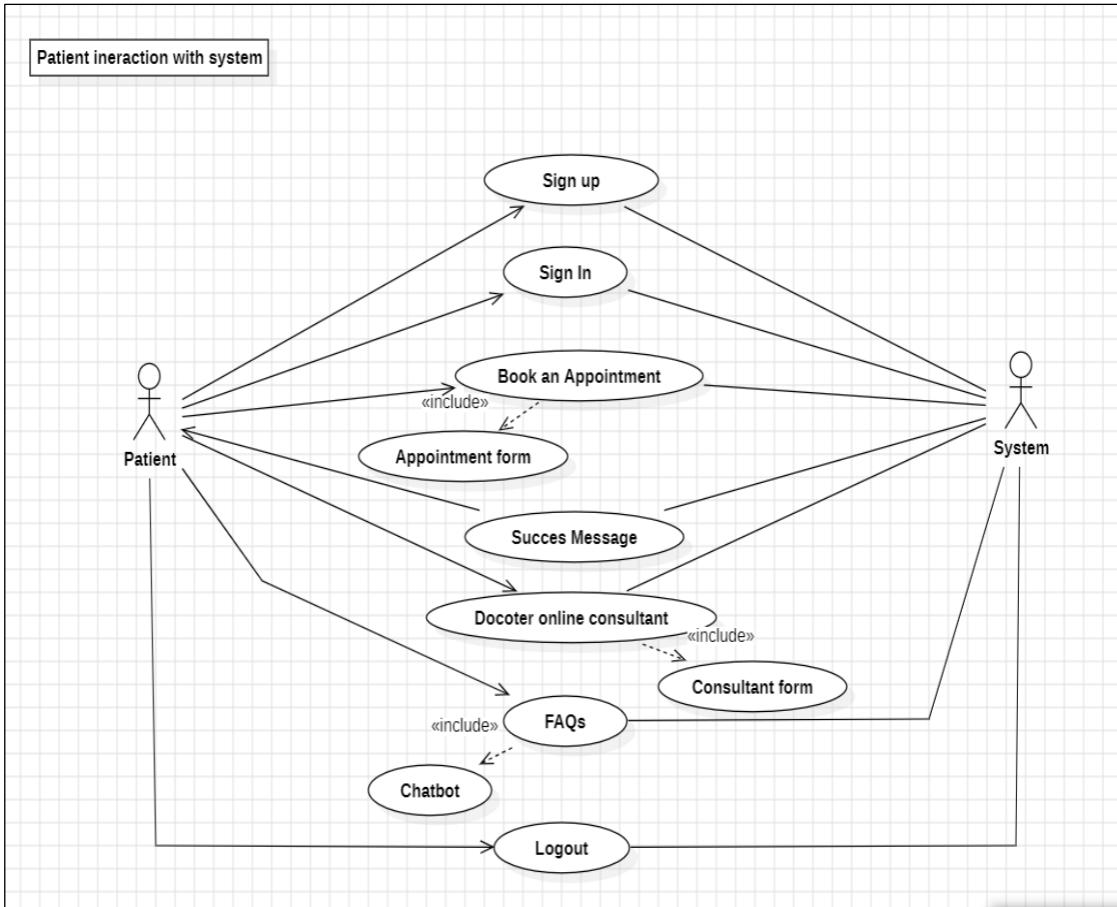


### 3.4.2 Use Case Diagram-

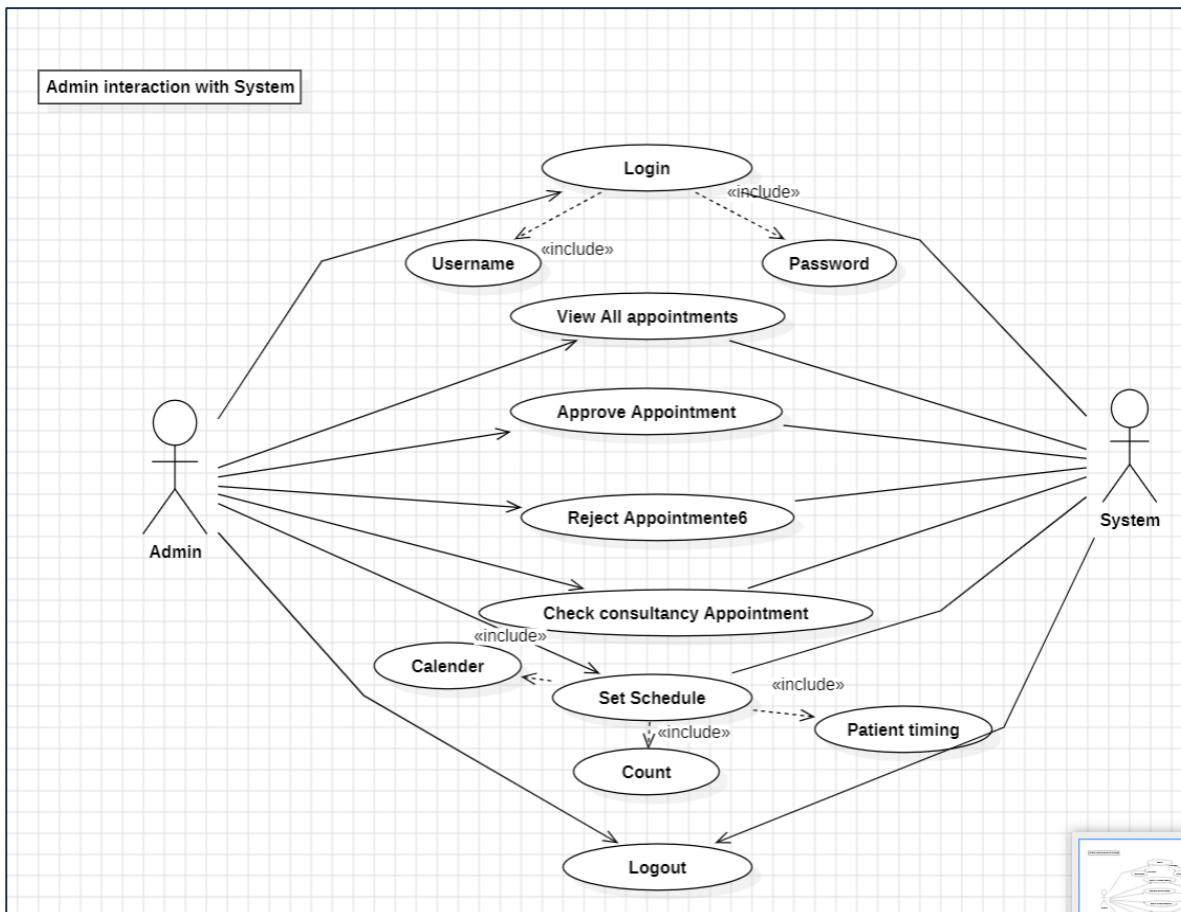
#### 1.Login



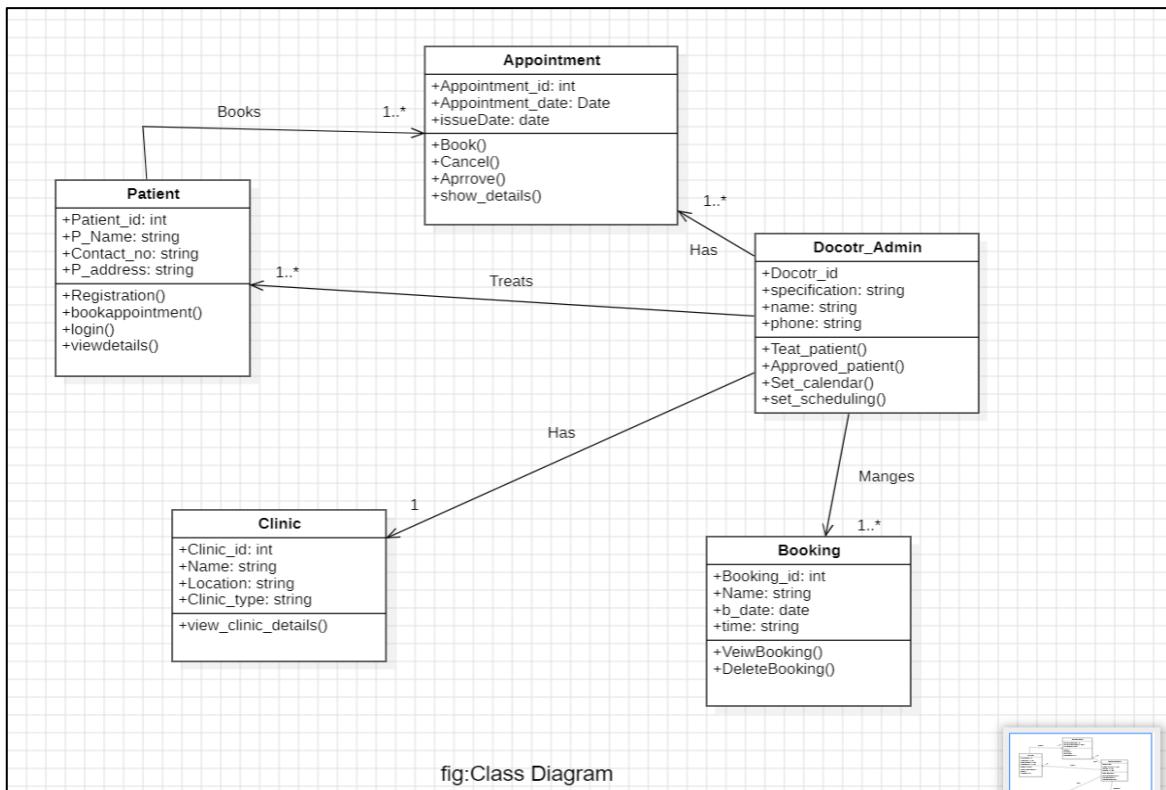
#### 2.Patient:



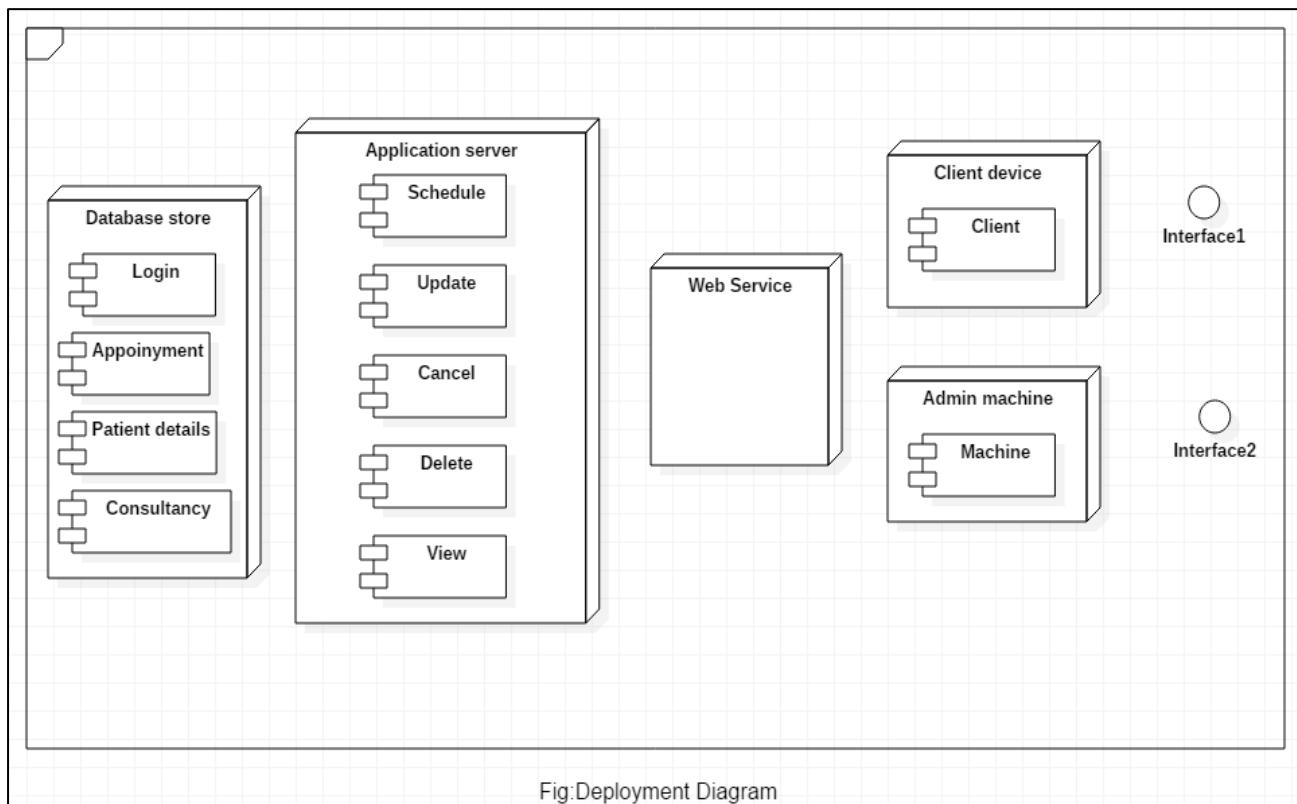
### 3.Admin/Doctor:



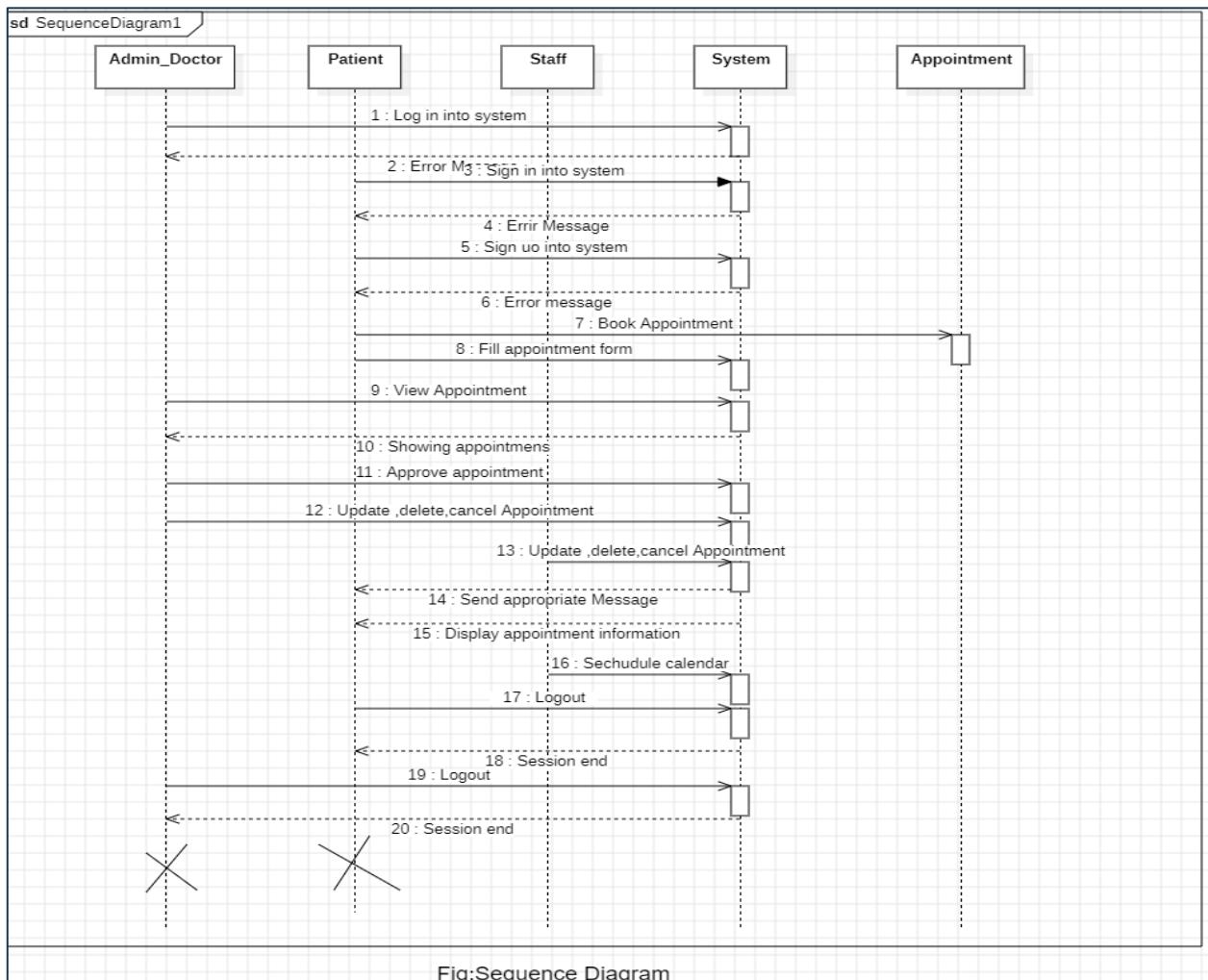
### 3.4.3 Class Diagram-



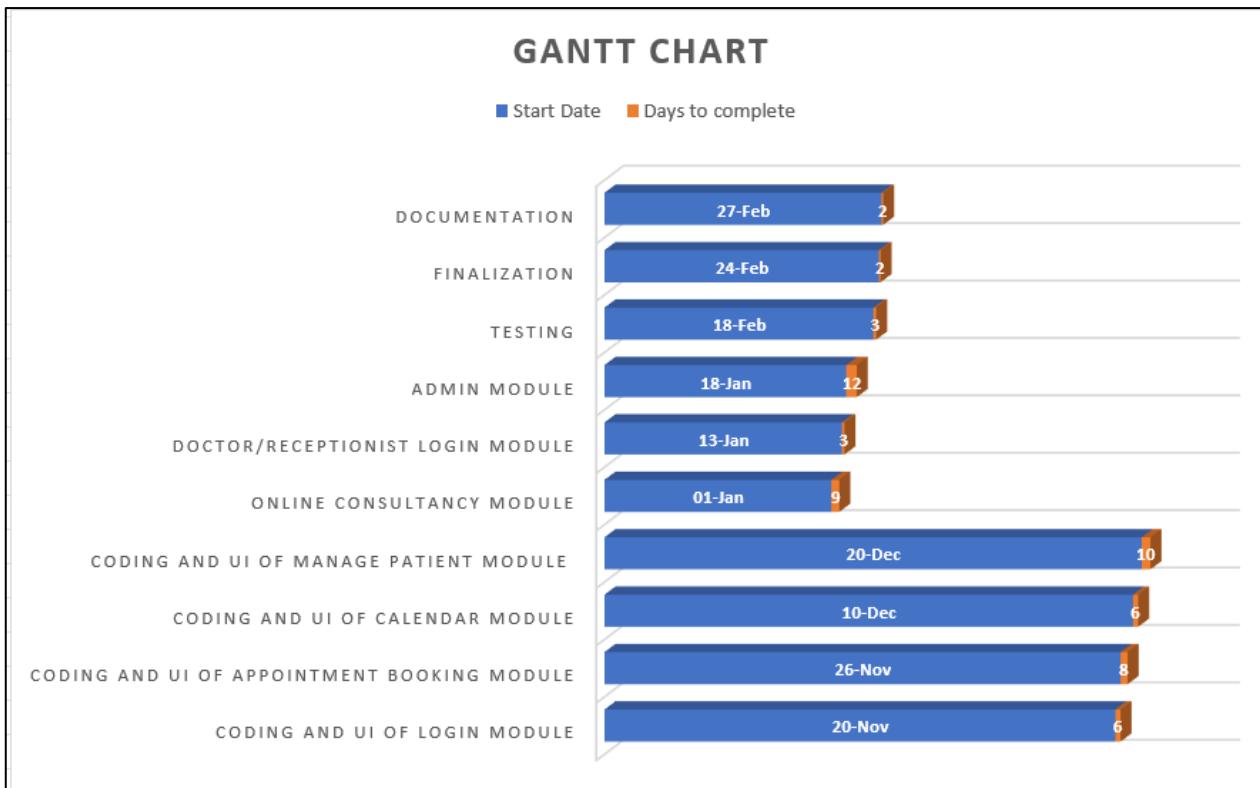
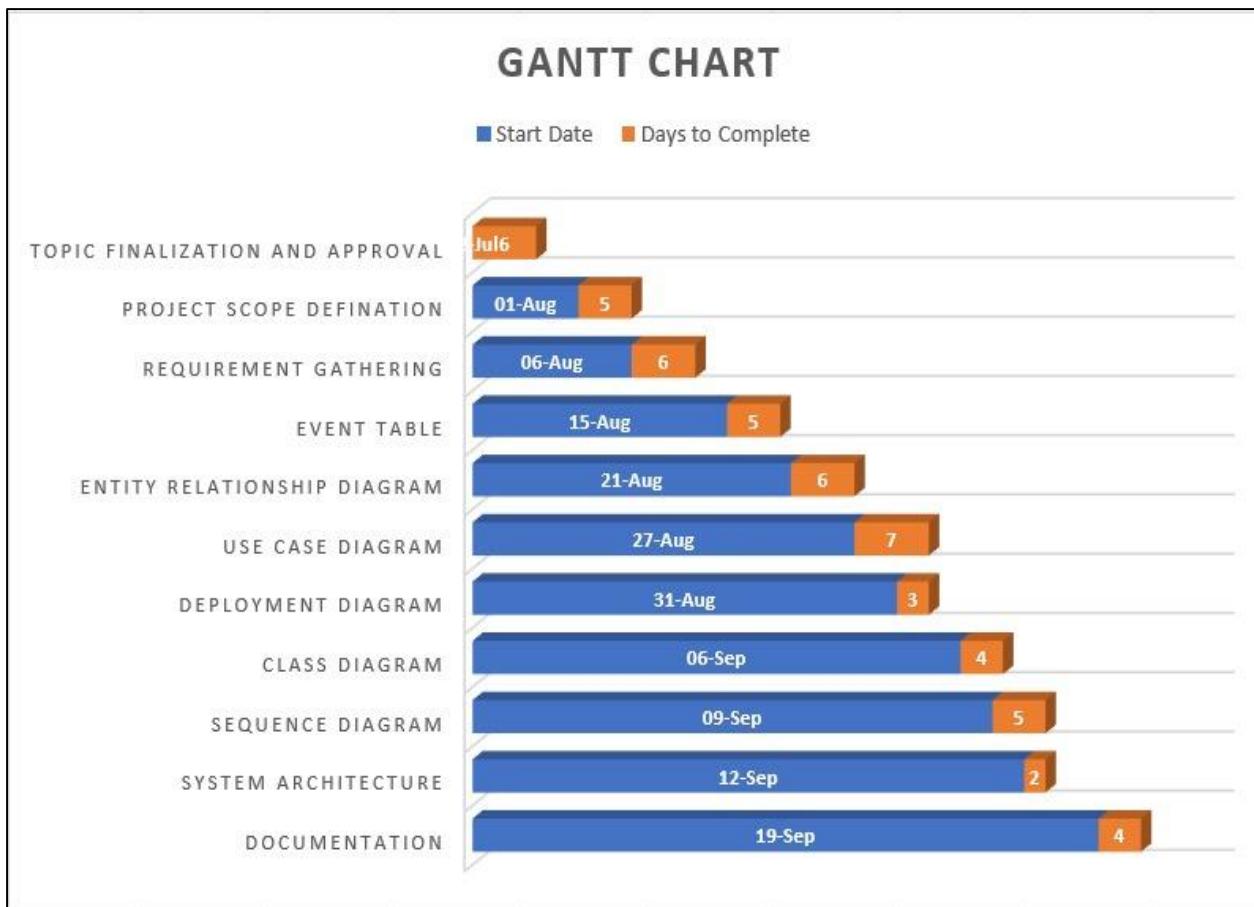
### 3.4.4 Deployment Diagram-



### 3.4.5 Sequence Diagram-



### 3.4 Gantt chart-



# SYSTEM DESIGN

## 4.1 Event Table

Event	Trigger	Source	Destination	Activity	Output
<b>Sign up into system</b>	User registration into system	New user	System	Sign up	a) Successfully sign up or b) Error Message.
<b>Sign in into System</b>	User sign in into system	Old user	System	Sign in	a) Successfully sign in or b) Error message
<b>Book Appointment</b>	User wants to book an appointment	User	System	Fill appointment details	a) Request sent or b) error message
<b>Approve appointment</b>	Admin/Doctor wants to approve appointment	Admin/Doctor	System	Admin checks patient details	a) Approved Or b) Not approved
<b>Display Status</b>	User wants to check her/his status	User	System	User checks own appointment profile	Display appointment details
<b>Online consultancy</b>	User wants to book online meeting	User	System	Fill consultancy form	a) Request sent or b) error message
<b>Approve Consultancy</b>	Admin wants to check his/her status	User	System	Admin checks schedule to grant consultancy appointment	a) Consultancy appointment granted or b) Not granted
<b>Update appointment calendar</b>	Admin/doctor wants to set calendar	Admin/doctor	System	Admin sets calendar	Calendar updated

## **4.2 Modules**

1. Doctor Login Module.
2. Patients Sign in/up Module.
3. Appointment booking Module.
4. Patient Appointment Management Module.
5. Patient Calendar management Module.
6. Admin dashboard Module.
7. Online Consultancy Module.

### **4.3 Data Tables-**

#### **1.patient\_appointment\_tbl**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
P_name	Nvarchar(255)		Patient Name
P_dob	Date		Patient date of birth
P_bloodtype	varchar(5)		Patient blood type
P_gender	varchar(10)		Patient gender
P_contact	Nvarchar(30)		Patient contact
P_emailid	Nvarchar(255)		Patient email
Appointment_date	Date		Date of appointment
Appointment_time	Nvarchar(255)		time of appointment
P_problem	Nvarchar(255)		Patient problem
P_state	Nvarchar(255)		Patient state
P_city	Nvarchar(255)		Patient city
P_pincode	Nvarchar(255)		Area pincode
P_password	Nvarchar(255)		password
P_username	Nvarchar(255)		Username
P_id	int	Primary key	Patient id
P_appointment_time	Time(7)		Patient appointment time
Status	Nvarchar(255)		Status
Height	Nvarchar(255)		Height
Weight1	Nvarchar(255)		weight

#### **2.patient\_signupin\_tbl**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
p_username	nvarchar(max)	Primary key	Patient Username
p_password	nvarchar(max)	unique	Password
p_confirm_password	varchar (60)	Not null	Conform password
p_email	nvarchar(max)	Not null	Patient email

### **3. doctor\_login\_table**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
Id	Char(10)	Primary_key	Doctor id
Name1	Nvarchar(100)		Doctor name
Joining_date	Date		Doctor joining date
Leave_date	Date		Doctor leave date
Designation	Nvarchar(20)		Designation
D_username	Nvarchar(20)		Username
D_password	Nvarchar(100)		Password

### **4.patient\_feedback\_tbl**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
p_fid	int	Primary key	Patient feedback id
p_name	Nvarchar(max)	Not null	Patient name
p_email	Nvarchar(max)	Not null	Patient email
p_subject	Nvarchar(max)	Not null	Patient subject
p_message	Nvarchar(max)	Not null	Patient message
ImageUrl	Nvarchar(255)		Patient Image

### **5.patient\_contact**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
pat_name	Nvarchar(255)	Not null	Patient name
Pat_subject	Nvarchar(255)	Not null	Patient subject
Pat_email	Nvarchar(255)	Not null	Patient email
Pat_mobile_no	Varchar(30)	Not null	Patient mobile number
Pat_message	Nvarchar(255)	Not null	Patient message
Status	Varchar(20)		Patient status

## **6.patient\_consultancy\_tbl**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
C_id	Int	Primary key	Patient id
c_dob	Date	Not null	Patient Date of Birth
c_patientname	Nvarchar(max)	Not null	Patient name
C_gender	Varchar(10)	Not null	Patient gender
C_email	Nvarchar(max)	Not null	Patient email
C_package	Nvarchar(max)	Not null	Patient selected package
C_address	Nvarchar(max)	Not null	Patient address
Status	Varchar(10)	Not null	Patient status
Ap_date	Date	Not null	Patient consultancy date
Ap_time	Varchar(20)	Not null	Patient consultancy time
C_contact	Nvarchar(13)	Not null	Patient contact
C_problem	Nvarchar(max)	Not null	Patient problem
Video_link	Nvarchar(max)	Not null	Patient video link

## **7.emergency\_appointment\_tbl**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
e_id	int	Primary key	Appointment id
E_name	Nvarchar(max)		Patient name
E_dob	Date		Patient birth date
E_bloodtype	Varchar(10)		Patient blood type
E_gender	Varchar(10)		Patient gender
E_emailid	nVarchar(max)		Patient email id
E_appointment_date	Date		Patient appointment date
E_appointment_time	Varchar(10)		Patient Appointment time
E_address	Nvarchar(max)		Patient address
Height	Varchar(10)		Patient height
Weight	Varchar(10)		Patient weight
Status	Varchar(10)		Patient status
E_contact	Varchar(50)		Patient contact number

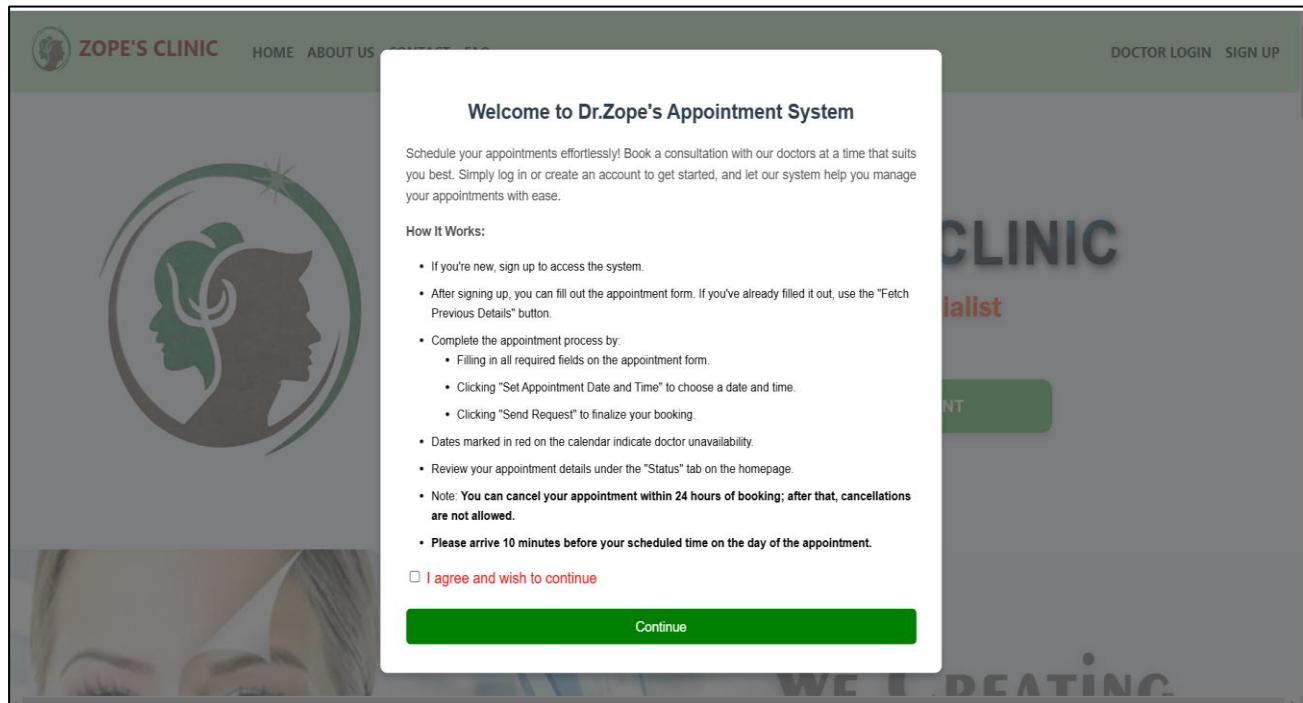
## **8.availability**

<b>Attribute</b>	<b>Datatype</b>	<b>Constraint</b>	<b>Description</b>
Id	int	Primary key	Availability id
Non_availability_date	Date		Date

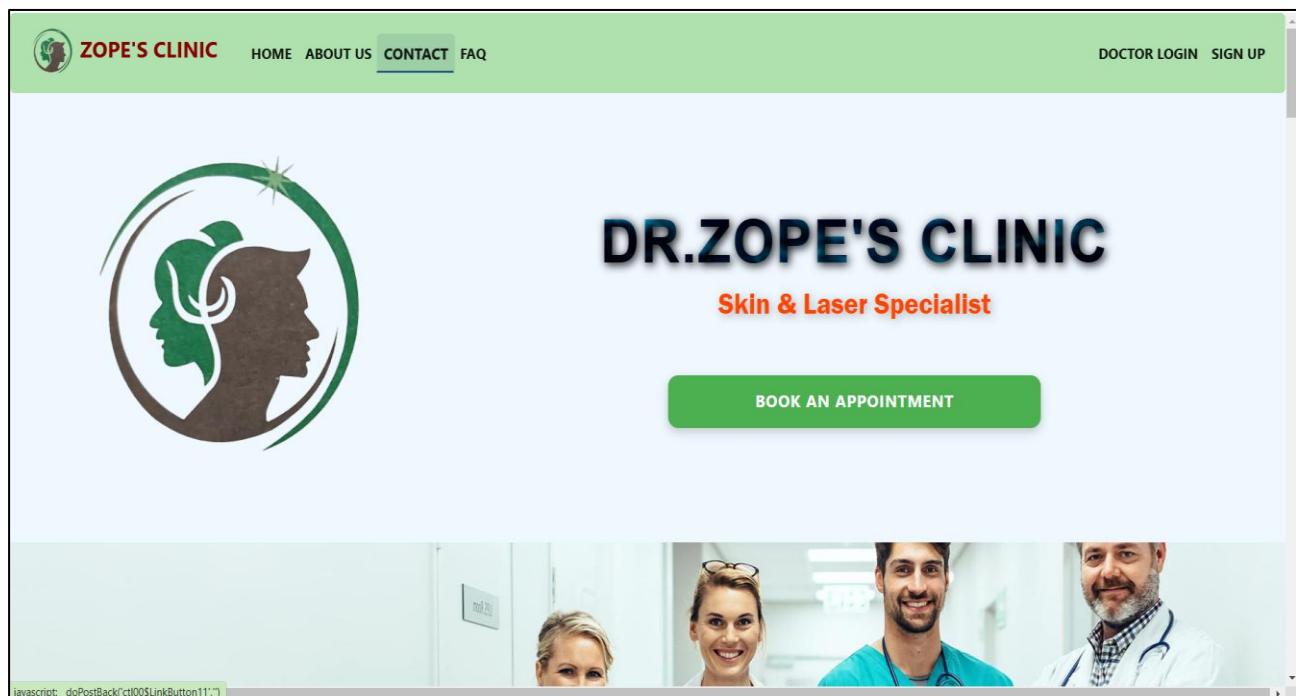
## 4.4 User Interfaces -

### USER PANEL -

Modal class window-



Home Page (Before Patient registration)-



## Sign up / Sign in Page-



### Patient SignUp

Enter Username

Enter Email ID

Enter Password  
 (

Enter Confirm Password  
 (

**Sign Up**



### Patient SignIn

Enter Username

Enter Password  
 (

**Sign In**

[Forgot Password?](#) | [New User?](#)

Home Page (After Patient registration)-

**ZOPE'S CLINIC** HOME ABOUT US CONTACT FAQ CONSULTANCY STATUS LOGOUT **HELLO Atharv\_Chavan1213**

**DR.ZOPE'S CLINIC**  
Skin & Laser Specialist

BOOK AN APPOINTMENT

**Beard Styling**  
A precise method to permanently reduce unwanted hair using concentrated laser light

Specified Service  
Lorem ipsum dol

24/7 Appointments  
Lorem ipsum dol

Online Consultation  
Lorem ipsum dol

Pharma  
Lorem ipsum dol

## Our Speciality

Our clinic specializes in providing advanced skincare and laser treatments tailored to individual needs. We use the latest technology to offer a wide range of services, from laser hair removal and skin rejuvenation to anti-aging therapies. Our team of experienced professionals is dedicated to helping you achieve healthy, radiant skin with personalized treatment plans. Whether you're seeking solutions for acne scars, pigmentation, or signs of aging, we prioritize safe and effective methods for lasting results. At our clinic, patient care and satisfaction are our top priorities, ensuring a comfortable experience in a professional environment.

**Face Cleaning**

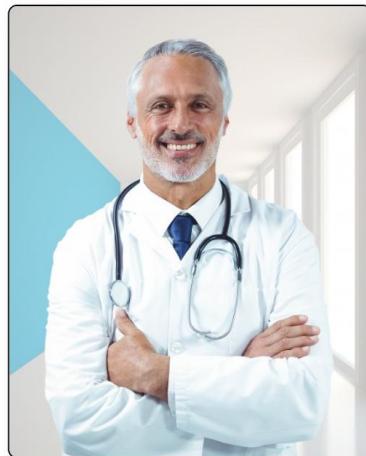
Face cleaning treatment involves deep cleansing of the face to remove dirt, oil, and impurities. It helps to unclog pores, prevent acne, and promote healthier-looking skin.

# Available Features in our clinic

Our skin and laser clinic offers a wide range of modern treatments and personalized care to help with various skin concerns. Our available services include laser hair removal for long-term hair reduction, skin resurfacing to reduce acne scars, fine lines, and uneven pigmentation, as well as specialized facials and chemical peels for brighter, healthier skin. We also provide anti-aging options, such as Botox and dermal fillers, and non-surgical body contouring to achieve a toned look. Each treatment is provided by skilled professionals using advanced equipment and safe, medically approved techniques to ensure effective results.



## Why Choose Us



### Advanced Laser Treatments

We offer the latest laser technology to effectively treat pigmentation, scars, and unwanted hair, delivering safe and lasting results.

### Personalized Skin Care

Our customized skin care regimens are designed to rejuvenate your skin, addressing acne, wrinkles, and other concerns.

### Expert Dermatologists

Our experienced team ensures professional care for all your dermatological needs, from consultations to advanced procedures.

## Our Special Doctors

Meet our experienced and specialized team of doctors who provide exceptional care.



**Dr. Aniruddha Zope**

Teledermatologist



**Dr. Sneha Zope**

Dermatologist



**Dr. Amol Zope**

Dermatologist



### Feedback Form

---

Name

Email

Subject

Message

**Upload Image (Optional)**

No file chosen

**Submit**

## What Our Patients Say About Us.



Nice infrastructure

**Atharv Uttam Chavan**



Excellent care and service

**Atharva Kulkarni**



Friendly and professional staff

**Nandini Chiplunkar**

• • • • •

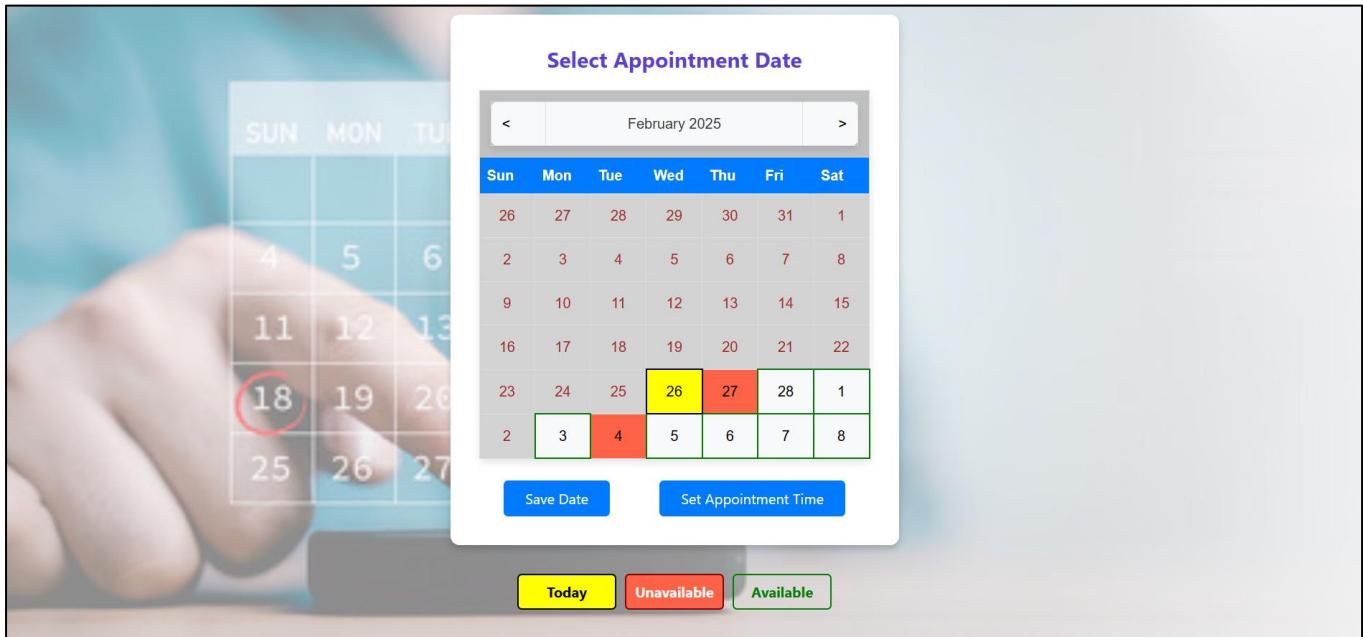
---

<b>QUICK LINKS</b> <hr/> <a href="#">HOME</a> <a href="#">DOCTORS</a> <a href="#">ABOUT US</a>	<b>GET HELP</b> <hr/> <a href="#">FAQ</a> <a href="#">MEDICINE</a>	<b>ADDRESS</b> <hr/> 42/43, 2nd Floor, Aashirwad Complex, Maruti Mandir, Ratnagiri-415612 Time: 9.30 AM to 1.30 PM & 3.00 PM to 6.00 PM	<b>CONTACT</b> <hr/>  Ratnagiri-415612, Maharashtra  zopesclinic123@gmail.com  9320882187, 8308826160
--	---	--	--

## Patient Appointment form –

The screenshot shows a web-based appointment form titled "Appointment Form". At the top, there is a "Login Credentials" section with fields for "Username" (containing "Atharv\_Chavan1213") and "Password" (represented by a series of dots). Below this is a "Fetch Previous Form Details" button. The main form area contains fields for "Patient Full Name" (Shruti Mahadik), "Date Of Birth" (12-04-2004), "Blood Type" (AB+), "Gender" (Female selected), "Contact Number" (9850115182), and "Email ID" (mratharv135@gmail.com).

## Selection of date



## Selection of time slot-

**MORNING SHIFT**

9:30 - 10:00    **10:00 - 10:30**    10:30 - 11:00    11:00 - 11:30    11:30 - 12:00    12:00 - 12:30

**EVENING SHIFT**

3:00 - 3:30    3:30 - 4:00    4:00 - 4:30    4:30 - 5:00    5:00 - 5:30    5:30 - 6:00

Selected time shift is 10:00 - 10:30

**SUBMIT**

*Note: All red shifts are already booked and cannot be selected.*

## Confirmation of Appointment-

Appointment Request Successfully Sent Inbox x

drzopeclinic2024@gmail.com 20:08 (4 minutes ago)

to me ▾

**Dear [mratharv135@gmail.com](mailto:mratharv135@gmail.com),**

Your appointment request has been successfully submitted. Your Appointment details are as follows:

- **Appointment Date:** 05-03-2025 00:00:00
- **Appointment Time:** 10:00 - 10:30

We will contact you shortly with further details. Thank you for choosing our services.

Best regards,  
Dr.Zope's Clinic

[Reply](#) [Forward](#)

## About us –



### DR.ZOPE'S CLINIC

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

**1000+**  
Happy Patients

**2+**  
Expert Doctors

**20+**  
Machinery

**800+**  
Patient Trust

## Contact page-

### Send a Message

**Full Name**

**Subject**

**Email**

**Mobile**

**Message**

**Send Message**

### Contact Info

Near the Clove Wine n Dine hotel, Maruti Mandir, Ratnagiri

drzopedclinic2024@gmail.com  
+91 987 654 3210

[Facebook](#) | [Twitter](#) | [LinkedIn](#) | [Instagram](#)



**WhatsApp**

## FAQ –

### FAQs

Answers to your most common questions about online doctor appointments for skin and laser treatments.

#### What is an online consultation with a skin and laser specialist?

An online consultation with a skin and laser specialist allows you to book a virtual appointment to discuss your skin concerns and treatment options. This consultation is convenient and can be done from the comfort of your home via video call or chat.

#### How do I book an appointment?

#### What skin treatments are available through online consultations?

#### Do I need to prepare anything before my online consultation?

#### How long does an online consultation last?

### How to book an Appointment?



## Online Consultancy-



### Online Consultancy Form

Patient Full Name

Date Of Birth

 dd-mm-yyyy

Email ID

Gender

 Male  Female  Other

Contact

Select Consultant Package

Problem Description

## OUR CONSULTATION

At Zope's Clinic, that is in our Skin and Laser Clinic, we understand that everyone's skin is unique and deserves personalized attention. With our online consultation services, you can now connect with our experienced dermatologists and laser specialists from the comfort of your home.

### Why Choose Our Online Consultation?

- Professional Guidance: Get expert advice from certified skin specialists for all your skin concerns.
- Convenience: No more waiting rooms—schedule an appointment at your convenience.
- Comprehensive Solutions: From acne and pigmentation to advanced laser treatments, we cover all your dermatological needs.

### Our Specialized Services Include:

- Advanced Laser Hair Removal
- Acne and Scar Treatments
- Pigmentation and Skin Brightening Solutions
- Anti-Aging Procedures
- Skin Rejuvenation and Hydration Therapy

## Packages/Therapy provided for Online consultancy-



### RADIANT GLOW PACKAGE

The Radiant Glow Package is your ultimate solution for achieving naturally luminous and flawless skin. This comprehensive treatment focuses on revitalizing dull, tired-looking skin, and enhancing your natural radiance.

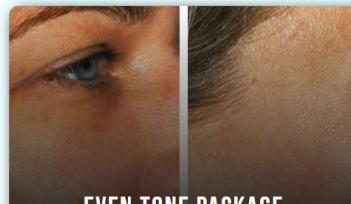
Cost- Rs.100/-



### ACNE FREE SKIN PACKAGE

The Acne-Free Skin Package offers advanced treatments to combat acne, reduce scars, and prevent breakouts. Achieve clearer, smoother, and healthier skin with our personalized care and expert guidance.

Cost- Rs.1000/-



### EVEN TONE PACKAGE

The Even Tone Package is designed to target pigmentation, dark spots, and uneven skin tone. Achieve radiant, balanced, and glowing skin with customized treatments tailored to your specific needs.

Cost- Rs.750/-



### HAIR FREE SMOOTH PACKAGE



### SCAR REVISION PACKAGE



### SPOTLESS SKIN PACKAGE

## Patient Status-

### PATIENT APPOINTMENT STATUS

Select an Appointment:

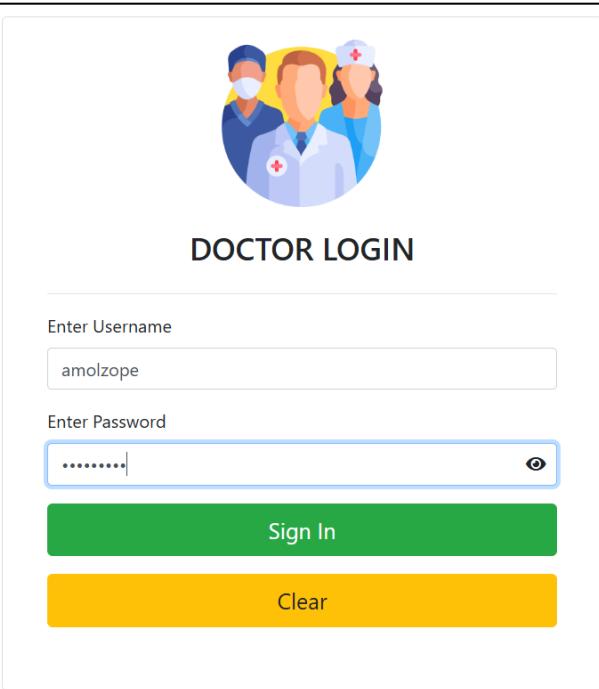
Appointment on 2025-03-05 ▾

#### Appointment Details

Appointment ID	5153
Appointment Issued Date	26-02-2025
Appointment Issued Time	20:04
Appointment Status	pending
Patient Name	Shruti Mahadik
Date of Birth	12-04-2004
Blood Type	AB+
Gender	Female
Contact Number	9850115182
Email ID	mratharv135@gmail.com
Appointment Date	05-03-2025

## ADMIN PANEL –

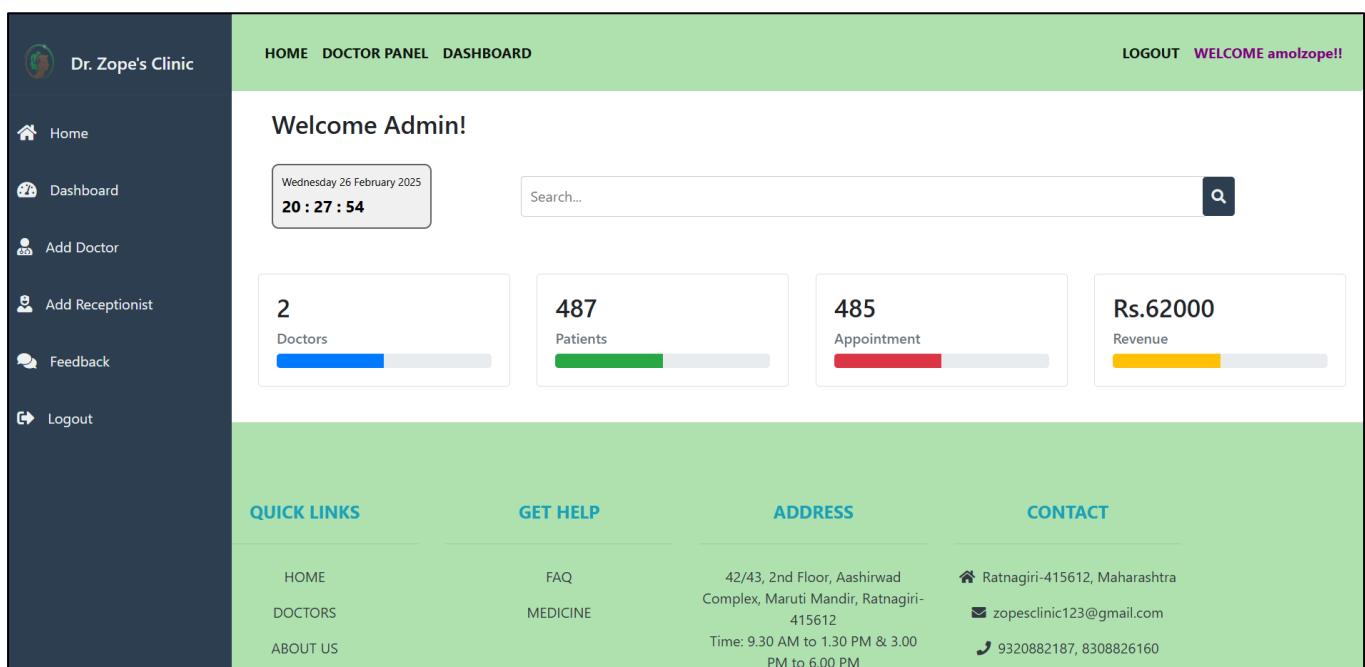
Doctor login-



The form is titled "DOCTOR LOGIN". It features a placeholder "Enter Username" with the value "amolzope" and a placeholder "Enter Password" with the value "\*\*\*\*\*". Below these are two buttons: a green "Sign In" button and a yellow "Clear" button.

[← Home page](#)

Doctor panel-



The dashboard is titled "HOME DOCTOR PANEL DASHBOARD". It includes a welcome message "Welcome Admin!", a date and time display ("Wednesday 26 February 2025 20 : 27 : 54"), a search bar, and a navigation menu on the left. Key statistics are shown in four boxes: 2 Doctors (blue bar), 487 Patients (green bar), 485 Appointment (red bar), and Rs.62000 Revenue (yellow bar). The bottom section contains "QUICK LINKS", "GET HELP", "ADDRESS", and "CONTACT" sections.

**QUICK LINKS**

- HOME
- DOCTORS
- ABOUT US

**GET HELP**

- FAQ
- MEDICINE

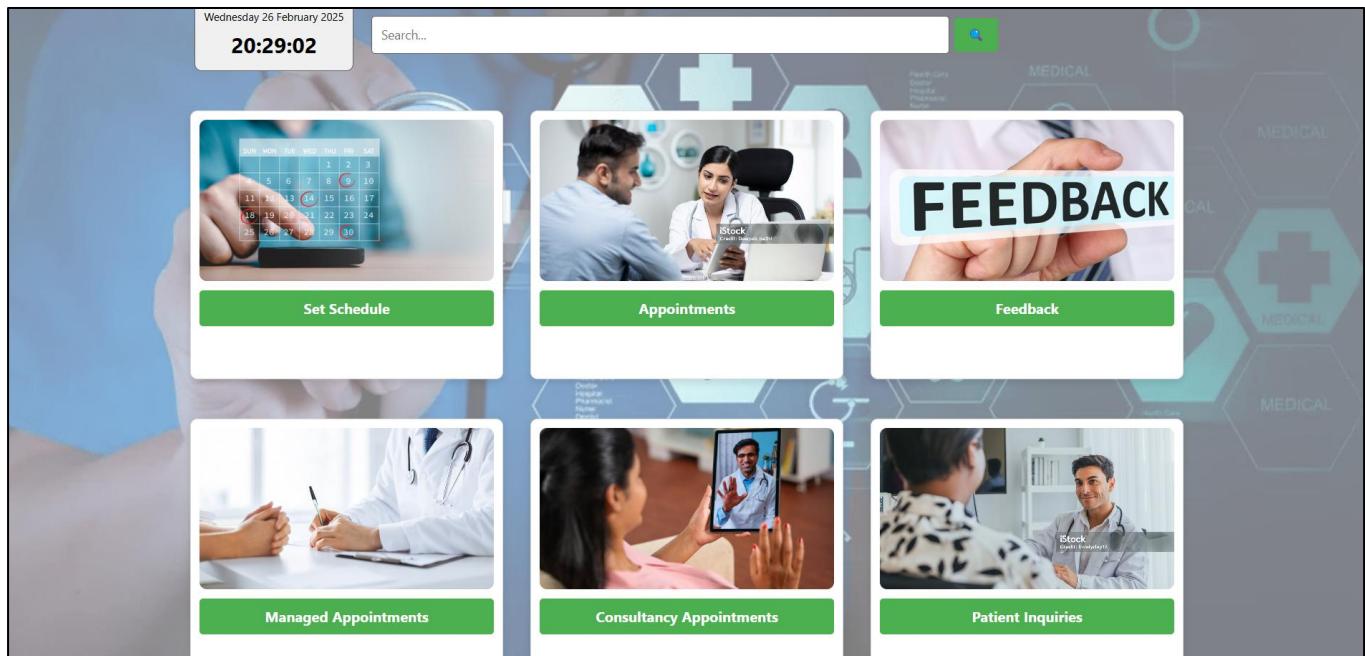
**ADDRESS**

42/43, 2nd Floor, Aashirwad Complex, Maruti Mandir, Ratnagiri-415612  
Time: 9.30 AM to 1.30 PM & 3.00 PM to 6.00 PM

**CONTACT**

- Ratnagiri-415612, Maharashtra
- zopesclinic123@gmail.com
- 9320882187, 8308826160

## Doctor Dashboard-



## Set schedule-

**Set Schedule** module:

February 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

DD/MM/YYYY

Add Date Emergency Leave

**Vacation date list** module:

Vacation ID	Date	Action
1	13-02-2025	Delete
2	21-02-2025	Delete
3	27-02-2025	Delete

Show 10 entries Search: Previous 1 Next

## Emergency leave –



Back

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

yyyy-MM-dd

9:30 - 10:00	10:00 - 10:30	10:30 - 11:00
11:00 - 11:30	11:30 - 12:00	12:00 - 12:30
3:00 - 3:30	3:30 - 4:00	4:00 - 4:30
4:30 - 5:00	5:00 - 5:30	5:30 - 6:00

Send Message

## Patient appointments-

**Patient Appointment Profile**



ID	Contact	
3155	9870675456	
Name	Email ID	
Virat Kohali	mratharv135@gmail.com	
DOB	Blood Type	Gender
11-02-2025	AB-	Male
Appointment Date	Appointment Time	
11-02-2025	5:30 - 6:00	
Height	Weight	
Height	Weight	
Address		
Mumbai		

**Issued Appointment List**

Patient ID	Name	Contact	Appointment Date	Status	Action
3155	Virat Kohali	9870675456	11-02-2025	pending	<button>Accept</button> <button>Reject</button>
3156	Virat Kohali	9870675456	18-02-2025	pending	<button>Accept</button> <button>Reject</button>
3157	Virat Kohali	9870675456	12-02-2025	pending	<button>Accept</button> <button>Reject</button>
3159	Virat Kohali	9870675456	11-02-2025	pending	<button>Accept</button> <button>Reject</button>
3162	Virat Kohali	9870675456	18-02-2025	pending	<button>Accept</button> <button>Reject</button>
3164	Madhuri dikshit	9870675456	11-02-2025	pending	<button>Accept</button> <button>Reject</button>
3165	Neha kakkar	9870675456	11-02-2025	pending	<button>Accept</button> <button>Reject</button>
3166	sana khan	9870675456	11-02-2025	pending	<button>Accept</button> <button>Reject</button>
3167	Virat Kohali	9870675456	19-02-2025	pending	<button>Accept</button> <button>Reject</button>
3168	Neha sonu	9870675456	18-02-2025	pending	<button>Accept</button> <button>Reject</button>

Show 10 entries Search:

Showing 1 to 10 of 13 entries Previous 1 2 Next

## Patient emergency appointment-

[← Back](#)



### Emergency Appointment Form

**Patient Details**

Patient Full Name	Date Of Birth
Full Name	dd-mm-yyyy <input type="button" value=""/>
Blood Type	Gender
Blood Type	<input type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Contact Number	Email ID
Contact	Email ID
Address	Select Time Slot
	<input type="button" value="Select"/>

## Managed emergency appointment-

[HOME](#) [DOCTOR PANEL](#) [DASHBOARD](#) [LOGOUT](#) **WELCOME amolzope!!**

[← Back](#)

### Emergency Appointment List

Patient ID	Name	Contact	Appointment Date	Status	Action
1007	Aruna kamble	6745932134	09-02-2025	pending	<input type="button" value="Accept"/> <input type="button" value="Reject"/>
2007	Atharv Chavan	9870435678	24-02-2025	pending	<input type="button" value="Accept"/> <input type="button" value="Reject"/>

Show 10 entries Search:

Showing 1 to 2 of 2 entries Previous  Next

---

**QUICK LINKS** **GET HELP** **ADDRESS** **CONTACT**

HOME	FAQ	42/43, 2nd Floor, Aashirwad Complex, Maruti Mandir, Ratnagiri-415612 Time: 9.30 AM to 1.30 PM & 3.00 PM to 6.00 PM	Ratnagiri-415612, Maharashtra zopesclinic123@gmail.com 9320882187, 8308826160
DOCTORS	MEDICINE		
ABOUT US			

## Feedback-

**PATIENT FEEDBACKS**

Patient Name	Email	Subject	Message	Action
Atharv Uttam Chavan	mratharv135@gmail.com	Clinic related	Nice infrastructure	<button>Delete</button>
Atharva Kulkarni	atharvkulkarni123@gmail.com	clinic	Excellent care and service	<button>Delete</button>
Nandini Chiplunkar	nandubhai34@gmail.com	clinic related	Friendly and professional staff	<button>Delete</button>
Vaishnavi chavan	vaish67@gmail.com	clinic	Highly recommend this clinic	<button>Delete</button>
Muskan Pagarkar	mratharv135@gmail.com	clinic related	clinic was excellent; the staff was caring	<button>Delete</button>

**QUICK LINKS**    **GET HELP**    **ADDRESS**    **CONTACT**

## Managed appointment-

**Managed Appointments**

ID: 101    Contact: 9822159627

Name: Nishant Kale    Email ID: atharvchavan106@gmail.com

DOB: 16-06-2004    Blood Type: AB+    Gender: Male

Appointment Date: 28-11-2024    Appointment Time: 5:30 - 6:00

Height: 150    Weight: 60

Address: 636,Renuka Niwas, Laxminaraynwadi, Kasarweli , Ratanagiri

**Appointment List**

Patient ID	Name	Contact	Status
101	Nishant Kale	9822159627	Appointed
104	Mihir Sawant	6745932134	Appointed
105	Yash naik	6745932134	Appointed
108	Poorva Kdam	6745932134	Appointed
1119	Mihir Sawant	6745932134	Appointed
1123	Yusuf Pathan	6745932134	Appointed
1124	Rudra karandikar	6745932134	Appointed
1125	Kavya deshmukh	6745932134	Appointed
1128	Mayur lokhande	6745932134	Appointed
1134	Mihir Sawant	6745932134	Appointed

Showing 1 to 10 of 35 entries

## Patient online consultancy appointments-

**Patient Appointment Profile**

ID: C016 | Contact: 9657434512 | Go

Consultancy Date: dd-mm-yyyy | Consultancy Time:

Patient Name: Atharv Chavan | Email ID: mratharv135@gmail.com

DOB: 13-02-2025 | Gender: Male

Video Link: [Empty] | Package: YouthFullLaserPackag

**Issued Consultancy List**

Patient ID	Patient Name	Calling
C016	Atharv Chavan	<a href="#">Get Link</a>
C018	amar chavan	<a href="#">Get Link</a>
C019	Virat Kohali	<a href="#">Get Link</a>

## Set date and Time-

Enter Doctor ID: C015

February 2025

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	1
2	3	4	5	6	7	8

2025-03-08

[Save Date](#) | [Emergency Leave](#)

**MORNING SHIFT**

9:00 - 10:00	10:00 - 11:00	11:00 - 12:00
--------------	---------------	---------------

**EVENING SHIFT**

3:00 - 4:00	4:00 - 5:00	5:00 - 6:00
-------------	-------------	-------------

Selected time shift is - 4:00 - 5:00

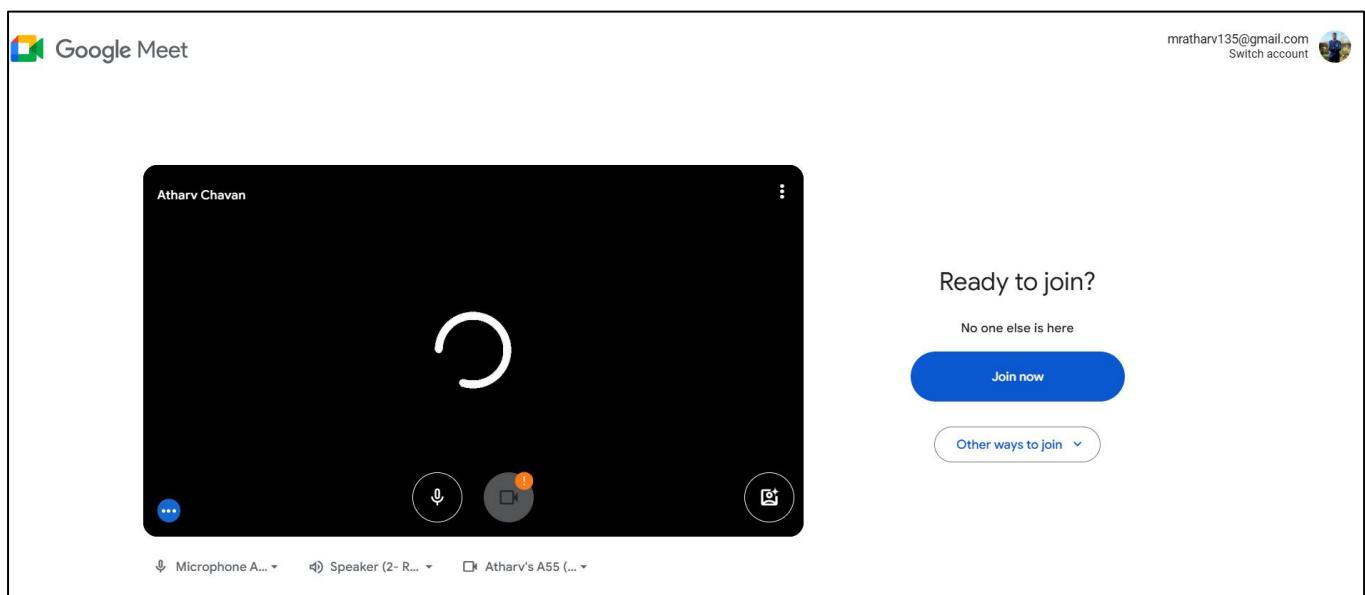
[SUBMIT](#)

Note: All red shifts are already booked and cannot be selected.

## Managed online consultancy appointment-

Issued Consultancy List										
Show 10 entries		Search:								
Patient ID	Patient Name	Contact Number	Date	Time	Status	Action	Video Link	Calling		
C001	Raj Waigankar	7056742378	25-11-2024 00:00:00	10:00 - 11:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/jzk-zcsk-wke">https://meet.google.com/jzk-zcsk-wke</a>	<button>Start Calling</button>		
C002	Atharv Chavan	9850115182	27-11-2024 00:00:00	10:00 - 11:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/ewx-nrgj-djq">https://meet.google.com/ewx-nrgj-djq</a>	<button>Start Calling</button>		
C003	Rahul vaidya	88679342134	07-11-2024 00:00:00	9:00 - 10:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/ssz-vbzcz-uv">https://meet.google.com/ssz-vbzcz-uv</a>	<button>Start Calling</button>		
C004	Vaishnavi Chavan	9552751623	03-01-2025 00:00:00	10:00 - 11:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/tmo-jtqj-cop">https://meet.google.com/tmo-jtqj-cop</a>	<button>Start Calling</button>		
C005	Atharv Chavan	9850115182	13-02-2025 00:00:00	4:00 - 5:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/phh-xdoc-bxr">https://meet.google.com/phh-xdoc-bxr</a>	<button>Start Calling</button>		
C006	Rekha Dikshit	9657434512	20-02-2025 00:00:00	9:00 - 10:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/msp-oxsy-zza">https://meet.google.com/msp-oxsy-zza</a>	<button>Start Calling</button>		
C007	Manorama	9657434512	28-11-2024 00:00:00	10:00 - 11:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/ajn-vsxt-mkn">https://meet.google.com/ajn-vsxt-mkn</a>	<button>Start Calling</button>		
C008	Jagadish shaikh	9657434512	30-11-2024 00:00:00	5:00 - 6:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/rpp-xavv-aje">https://meet.google.com/rpp-xavv-aje</a>	<button>Start Calling</button>		
C009	Haresh Shinde	9657434512	26-11-2024 00:00:00	9:00 - 10:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/tdr-yyxw-wso">https://meet.google.com/tdr-yyxw-wso</a>	<button>Start Calling</button>		
C010	Mangesh chalke	9657434512	28-11-2024 00:00:00	4:00 - 5:00	Appointed	<button>Accept</button> <button>Reject</button>	<a href="https://meet.google.com/gkm-rktx-ugj">https://meet.google.com/gkm-rktx-ugj</a>	<button>Start Calling</button>		

## Video Calling through Google meet for online consultancy-



## 4.5 Code and Implementation-

### Navfooter.master:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="navfooter.master.cs"
Inherits="WebApplication1.Site1" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>Doctor Appointment System</title>

    <!-- Bootstrap CSS -->
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet" />

    <!-- FontAwesome CSS -->
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" rel="stylesheet" />

    <!-- DataTables CSS -->
    <link rel="stylesheet" href="https://cdn.datatables.net/1.11.5/css/jquery.dataTables.min.css">

    <!-- Google Fonts -->
    <link href="https://fonts.googleapis.com/css2?family=Bebas+Neue&display=swap" rel="stylesheet">
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500&display=swap" rel="stylesheet">

    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

    <!-- Bootstrap JS & Dependencies -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

    <!-- Additional JS Libraries -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.4/moment.min.js"></script>
    <script src="https://cdn.datatables.net/1.13.4/js/jquery.dataTables.min.js"></script>

    <asp:ContentPlaceHolder ID="head" runat="server"></asp:ContentPlaceHolder>

    <style>
        /* Footer Background Color */
        section {
            background-color: #AFE1AF;
        }

        footer {
            width: 100%;
        }
    </style>
```

```

        clear: both;
        margin-top: 40px;
        flex: 0 0 100%;
    }

{
    .navbar-nav .nav-link:hover {
        color: #003366;
        background-color: rgba(0, 51, 102, 0.1);
        transform: scale(1.05);
        border-radius: 5px;
    }

    .navbar-nav .nav-link:hover::after {
        width: 100%;
    }
}
</style>
</head>

<body>
<form id="form1" runat="server">
    <div>
        <!-- Navbar -->
        <nav class="navbar navbar-expand-lg navbar-light" style="background-color:#AFE1AF;">
            <a class="navbar-brand" href="#">
                
                <span class="navbar-brand font-weight-bold" style="margin-left:-8px; color: darkred; font-size: 1.5rem;">
                    ZOPE'S CLINIC
                </span>
            </a>

            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav mr-auto">
                    <li class="nav-item active" runat="server">
                        <asp:LinkButton class="nav-link" ID="LinkButton8" runat="server" Visible="False" OnClick="LinkButton8_Click">HOME</asp:LinkButton>
                    </li>
                    <li class="nav-item active" runat="server">
                        <asp:LinkButton class="nav-link" ID="LinkButton6" runat="server" Visible="False" OnClick="LinkButton6_Click">DOCTOR PANEL</asp:LinkButton>
                    </li>
                    <li class="nav-item active" runat="server">
                        <asp:LinkButton class="nav-link" ID="LinkButton10" runat="server" Visible="False" OnClick="LinkButton10_Click">PATIENT</asp:LinkButton>
                    </li>
                </ul>
            </div>
        </nav>
    </div>
</form>

```

```

    </li>
    <li class="nav-item active" runat="server">
        <asp:LinkButton class="nav-link" ID="LinkButton11" runat="server" Visible="False" OnClick="LinkButton11_Click">CONTACT</asp:LinkButton>
    </li>
    <li class="nav-item active" runat="server">
        <asp:LinkButton class="nav-link" ID="LinkButton12" runat="server" Visible="False" OnClick="LinkButton12_Click">FAQ</asp:LinkButton>
    </li>
    <li class="nav-item active" runat="server">
        <asp:LinkButton class="nav-link" ID="LinkButton13" runat="server" Visible="False" OnClick="LinkButton13_Click">DASHBOARD</asp:LinkButton>
    </li>
    <li class="nav-item active" runat="server">
        <asp:LinkButton class="nav-link" ID="LinkButton14" runat="server" Visible="False" OnClick="LinkButton14_Click">CONSULTANCY</asp:LinkButton>
    </li>
    </ul>

    <ul class="navbar-nav">
        <li class="nav-item active" runat="server">
            <asp:LinkButton class="nav-link" ID="LinkButton1" runat="server" Visible="False" OnClick="LinkButton1_Click">DOCTOR LOGIN</asp:LinkButton>
        </li>
        <li class="nav-item active" runat="server">
            <asp:LinkButton class="nav-link" ID="LinkButton2" runat="server" Visible="False" OnClick="LinkButton2_Click">SIGN UP</asp:LinkButton>
        </li>
        <li class="nav-item active" runat="server">
            <asp:LinkButton class="nav-link" ID="LinkButton5" runat="server" Visible="False" OnClick="LinkButton5_Click">STATUS</asp:LinkButton>
        </li>
        <li class="nav-item active" runat="server">
            <asp:LinkButton class="nav-link" ID="LinkButton3" runat="server" Visible="False" OnClick="LinkButton3_Click">LOGOUT</asp:LinkButton>
        </li>
        <li class="nav-item active" runat="server">
            <asp:LinkButton class="nav-link" ID="LinkButton7" runat="server" Visible="False" OnClick="LinkButton7_Click">LOGOUT</asp:LinkButton>
        </li>
        <li class="nav-item active" runat="server" Visible="False">
            <asp:LinkButton class="nav-link" ID="LinkButton4" runat="server" Visible="False" OnClick="LinkButton4_Click">WELCOME USER</asp:LinkButton>
        </li>
    </ul>
</div>
</nav>
</div>

<!-- Main Content -->

```

```

<div>
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
runat="server"></asp:ContentPlaceHolder>
</div>
</form>
</body>
</html>

```

### **Patient\_appointment\_form.aspx.cs**

```

using System;
using System.Data.SqlClient;
using System.Configuration;
using System.Web.UI;

namespace WebApplication1.Patient
{
    public partial class patient_appointment_form : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                AppointmentFormPanel.Visible = false;

                // Restore values from Session if they exist
                if (Session["FullName"] != null)
                    TextBox3.Text = Session["FullName"].ToString();
                if (Session["ContactNumber"] != null)
                    TextBox4.Text = Session["ContactNumber"].ToString();
                if (Session["Email"] != null)
                    TextBox5.Text = Session["Email"].ToString();
                if (Session["BirthDate"] != null)
                    TextBox1.Text = Session["BirthDate"].ToString();
                if (Session["BloodType"] != null)
                    TextBox2.Text = Session["BloodType"].ToString();
                if (Session["Gender"] != null)
                    RadioButtonList1.SelectedValue = Session["Gender"].ToString();
                if (Session["ProblemDescription"] != null)
                    TextBox11.Text = Session["ProblemDescription"].ToString();
                if (Session["State"] != null)
                    DropDownList1.SelectedValue = Session["State"].ToString();
                if (Session["City"] != null)
                    TextBox9.Text = Session["City"].ToString();
                if (Session["Pincode"] != null)
                    TextBox10.Text = Session["Pincode"].ToString();
                if (Session["Address"] != null)
                    TextBox8.Text = Session["Address"].ToString();
                if (Session["Username"] != null)

```

```

    TextBox12.Text = Session["Username"].ToString();

    // Store the password in HiddenField for postbacks
    if (Session["Password"] != null)
        HiddenPassword.Value = Session["Password"].ToString();

    // Disable autocomplete on sensitive fields
    TextBox12.Attributes.Add("autocomplete", "off");
    TextBox13.Attributes.Add("autocomplete", "new-password");

    // Set the initial value of password textbox
    TextBox13.Attributes["value"] = HiddenPassword.Value;
}

else
{
    // Set the password field's value during postbacks
    TextBox13.Attributes["value"] = HiddenPassword.Value;
}
}

// Insert record function
protected void InsertRecord()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

    // Retrieve form values
    string fullName = TextBox3.Text;
    string bloodType = TextBox2.Text;
    DateTime birthDate;
    DateTime.TryParse(TextBox1.Text, out birthDate);
    string gender = RadioButtonList1.SelectedValue;
    string contactNumber = TextBox4.Text;
    string email = TextBox5.Text;
    string problemDescription = TextBox11.Text;
    string state1 = DropDownList1.SelectedValue;
    string city = TextBox9.Text;
    string pinCode = TextBox10.Text;
    string fullAddress = TextBox8.Text;
    string username = TextBox12.Text;
    string password = TextBox13.Text;
    string Status = "pending";

    string query = @"INSERT INTO patient_appointment_tbl
        (p_name, p_bloodtype, p_gender, p_contact, p_emailid, p_problem,
        p_state, p_city, p_pincode, p_dob, p_address, p_username, p_password,
        p_appointment_time, p_appointment_date, status)
        VALUES (@FullName, @BloodType, @Gender, @Contact, @Email,
        @ProblemDescription, @State, @City, @PinCode, @BirthDate,
        @Address, @Username, @Password, CAST(GETDATE() AS TIME), GETDATE(),

```

```

    @Status);
}

using (SqlConnection connection = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        // Add parameters
        command.Parameters.AddWithValue("@FullName", fullName);
        command.Parameters.AddWithValue("@BloodType", bloodType);
        command.Parameters.AddWithValue("@Gender", gender);
        command.Parameters.AddWithValue("@Contact", contactNumber);
        command.Parameters.AddWithValue("@Email", email);
        command.Parameters.AddWithValue("@ProblemDescription", problemDescription);
        command.Parameters.AddWithValue("@State", state1);
        command.Parameters.AddWithValue("@City", city);
        command.Parameters.AddWithValue("@PinCode", pinCode);
        command.Parameters.AddWithValue("@BirthDate", birthDate);
        command.Parameters.AddWithValue("@Address", fullAddress);
        command.Parameters.AddWithValue("@Username", username);
        command.Parameters.AddWithValue("@Password", password);
        command.Parameters.AddWithValue("@Status", Status);

        try
        {
            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();

            if (rowsAffected > 0)
            {

                lblModalMessage.Text = "Record inserted successfully!";
                ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
                    "$('#messageModal').modal('show');", true);
            }
            else
            {

                lblModalMessage.Text = "No record inserted. Check field values or constraints.";
                ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
                    "$('#messageModal').modal('show');", true);
            }
        }
        catch (SqlException ex)
        {
            string sanitizedMessage = ex.Message.Replace("""", "\\\"");
            ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('SQL Error: {sanitizedMessage}');");
        }
    }
}

```

```

        }

    }

    // Clear fields

    // Redirects and stores form values in session
    protected void Button3_Click(object sender, EventArgs e)
    {
        StoreFormValuesInSession();
        Response.Redirect("select_appointment_date.aspx");
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        StoreFormValuesInSession();
        Response.Redirect("select_appointment_time.aspx");
    }

    // Stores form values in session
    private void StoreFormValuesInSession()
    {
        Session["FullName"] = TextBox3.Text;
        Session["ContactNumber"] = TextBox4.Text;
        Session["Email"] = TextBox5.Text;
        Session["BirthDate"] = TextBox1.Text;
        Session["BloodType"] = TextBox2.Text;
        Session["ProblemDescription"] = TextBox11.Text;
        Session["State"] = DropDownList1.SelectedValue;
        Session["City"] = TextBox9.Text;
        Session["Pincode"] = TextBox10.Text;
        Session["Address"] = TextBox8.Text;
        Session["Gender"] = RadioButtonList1.SelectedValue;
        Session["Username"] = TextBox12.Text;
        Session["Password"] = TextBox13.Text;
    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
        InsertRecord();

        // Uncomment this line once data is inserting successfully
        Response.Redirect("select_appointment_date.aspx");
    }

    private bool IsUserAuthenticated(string username, string password)
    {
        string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

```

```

        string query = "SELECT COUNT(*) FROM patient_signupin_tbl WHERE p_username =
@Username AND p_password = @Password";

        using (SqlConnection connection = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@Username", username);
        command.Parameters.AddWithValue("@Password", password);

        try
        {
            connection.Open();
            int userCount = (int)command.ExecuteScalar();
            return userCount > 0;
        }
        catch (Exception ex)
        {
            lblModalMessage.Text = "Error: " + ex.Message.Replace("'", "\\'");
            ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
"$('#messageModal').modal('show');", true);

            return false;
        }
    }
}

protected void LoginButton_Click(object sender, EventArgs e)
{
    if (IsUserAuthenticated(TextBox12.Text, TextBox13.Text))
    {
        // Show the remaining form fields if the user is authenticated
        AppointmentFormPanel.Visible = true;

        // Hide the Username, Password, and Login button after successful login
        TextBox12.Enabled = false;
        TextBox13.Enabled = true;
        LoginButton.Visible = false;

        // Store username in session for later use
        Session["Username"] = TextBox12.Text;
        Session["Password"] = TextBox13.Text;

        HiddenPassword.Value = Session["Password"].ToString();
    }
    else
    {
        AppointmentFormPanel.Visible = true;
    }
}

```

```

        }

    }

protected void Button3_Click1(object sender, EventArgs e)
{
    string username = TextBox12.Text;
    string password = TextBox13.Text;

    if (CheckAndLoadPreviousAppointment(username, password))
    {
        // Load the form with existing details
        AppointmentFormPanel.Visible = true;
    }
    else
    {
        lblModalMessage.Text = "You are new here, we do not have any details of you.";
        ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
        "$('#messageModal').modal('show');", true);
    }
}

private bool CheckAndLoadPreviousAppointment(string username, string password)
{
    string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;
    string query = @"SELECT p_name, p_bloodtype, p_gender, p_contact, p_emailid, p_problem,
                      p_state, p_city, p_pincode, p_dob, p_address
                      FROM patient_appointment_tbl
                      WHERE p_username = @Username AND p_password = @Password";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@Username", username);
            command.Parameters.AddWithValue("@Password", password);

            try
            {
                connection.Open();
                SqlDataReader reader = command.ExecuteReader();

                if (reader.HasRows && reader.Read())
                {
                    // Populate form fields
                    TextBox3.Text = reader["p_name"].ToString();
                    TextBox2.Text = reader["p_bloodtype"].ToString();
                    RadioButtonList1.SelectedValue = reader["p_gender"].ToString();
                    TextBox4.Text = reader["p_contact"].ToString();
                    TextBox5.Text = reader["p_emailid"].ToString();
                    TextBox11.Text = reader["p_problem"].ToString();
                }
            }
        }
    }
}

```

```

        DropDownList1.SelectedValue = reader["p_state"].ToString();
        TextBox9.Text = reader["p_city"].ToString();
        TextBox10.Text = reader["p_pincode"].ToString();
        TextBox1.Text = Convert.ToDateTime(reader["p_dob"]).ToString("yyyy-MM-dd");
        TextBox8.Text = reader["p_address"].ToString();
        return true;
    }
}
catch (Exception ex)
{
    lblModalMessage.Text = "Error: " + ex.Message.Replace("'", "\\'");
    ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
    "$('#messageModal').modal('show');", true);
}

}
}

return false;
}
}

}
}

```

## **Homepage2.aspx.cs**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using DlibDotNet.Dnn;
using System.IO;
using System.Xml.Linq;

namespace WebApplication1
{
    public partial class homepage2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                LoadTestimonials();
            }
        }

        protected void Button1_Click(object sender, EventArgs e)

```

```

{
    // Retrieve the username and password from the session variables
    string username = Session["Username"] != null ? Session["Username"].ToString() : null;
    string password = Session["Password"] != null ? Session["Password"].ToString() : null;

    if (!string.IsNullOrEmpty(username) && !string.IsNullOrEmpty(password))
    {
        // Verify the credentials against the database
        string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;
        using (SqlConnection conn = new SqlConnection(connectionString))
        {
            conn.Open();
            string query = "SELECT COUNT(1) FROM patient_signupin_tbl WHERE p_username =
@Username AND p_password = @Password";
            SqlCommand cmd = new SqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@Username", username);
            cmd.Parameters.AddWithValue("@Password", password);

            int count = Convert.ToInt32(cmd.ExecuteScalar());

            if (count == 1)
            {
                // Valid credentials, redirect to the appointment form
                Response.Redirect("~/Patient/patient_appointment_form.aspx");
            }
            else
            {
                // Invalid credentials, redirect to the sign-in form
                Response.Redirect("~/Patient/patient_signin_form.aspx");
            }
        }
    }
    else
    {
        // No username or password in session, redirect to the sign-in form
        Response.Redirect("~/Patient/patient_signin_form.aspx");
    }
}

protected void btnSubmit_Click(object sender, EventArgs e)
{
    if (Session["Username"] == null)
    {
        // Redirect to the sign-in page if not signed in
        Response.Redirect("~/Patient/patient_signin_form.aspx");
        return;
    }

    // Retrieve input values from the TextBox controls

```

```

string name = txtName.Text;
string email = txtEmail.Text;
string subject = txtSubject.Text;
string message = txtMessage.Text;

// Initialize ImageURL to a default image in the 'img' folder
string ImageURL = "img/feedback_icon.jpg";

// Check if an image is uploaded
if (fileUpload.HasFile)
{
    try
    {
        // Get the file extension and ensure it's a valid image file type
        string fileExtension = Path.GetExtension(fileUpload.PostedFile.FileName).ToLower();
        if (fileExtension == ".jpg" || fileExtension == ".jpeg" || fileExtension == ".png" || fileExtension
== ".gif")
        {
            // Generate a unique filename to avoid conflicts
            string fileName = "feedback_" + Guid.NewGuid().ToString() + fileExtension;
            string filePath = Server.MapPath("img/") + fileName;

            // Save the uploaded file to the 'img' folder
            fileUpload.SaveAs(filePath);

            // Set the ImageURL to the newly saved image path
            ImageURL = "img/" + fileName;
        }
        else
        {
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('Please upload a valid
image file (.jpg, .jpeg, .png, .gif).');", true);
            return;
        }
    }
    catch (Exception ex)
    {
        // Handle any errors during file upload
        ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('An error occurred while
uploading the image: {ex.Message}');", true);
        return;
    }
}

// Define the SQL connection string (update with your actual connection string if needed)
string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

// SQL query to insert data into patient_feedback_tbl
string query = "INSERT INTO patient_feedback_tbl (p_name, p_email, p_subject, p_message,
ImageUrl) " +

```

```

        "VALUES (@Name, @Email, @Subject, @Message, @ImageURL)";

using (SqlConnection connection = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        // Add parameters to avoid SQL injection
        command.Parameters.AddWithValue("@Name", name);
        command.Parameters.AddWithValue("@Email", email);
        command.Parameters.AddWithValue("@Subject", subject);
        command.Parameters.AddWithValue("@Message", message);
        command.Parameters.AddWithValue("@ImageURL", ImageURL);

        // Open the database connection, execute the command, and display a success message
        try
        {
            connection.Open();
            command.ExecuteNonQuery();

            // Display an alert message to the user
            ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('Feedback sent successfully!');", true);

            // Optionally, clear the form after submission
            txtName.Text = "";
            txtEmail.Text = "";
            txtSubject.Text = "";
            txtMessage.Text = "";
        }
        catch (Exception ex)
        {
            // Handle exceptions (e.g., log error, show error message)
            ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('An error occurred: {ex.Message}');", true);
        }
    }
}

private void LoadTestimonials()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

    using (SqlConnection con = new SqlConnection(connectionString))
    {
        string query = "SELECT p_message, ImageUrl,P_name FROM patient_feedback_tbl";
        SqlCommand cmd = new SqlCommand(query, con);

        con.Open();

```

```
SqlDataReader reader = cmd.ExecuteReader();
rptTestimonials.DataSource = reader;
rptTestimonials.DataBind();
    }
}
}
```

## Patient\_appointment\_form.aspx.cs

```
using System;
using System.Data.SqlClient;
using System.Configuration;
using System.Web.UI;

namespace WebApplication1.Patient
{
    public partial class patient_appointment_form : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                AppointmentFormPanel.Visible = false;

                // Restore values from Session if they exist
                if (Session["FullName"] != null)
                    TextBox3.Text = Session["FullName"].ToString();
                if (Session["ContactNumber"] != null)
                    TextBox4.Text = Session["ContactNumber"].ToString();
                if (Session["Email"] != null)
                    TextBox5.Text = Session["Email"].ToString();
                if (Session["BirthDate"] != null)
                    TextBox1.Text = Session["BirthDate"].ToString();
                if (Session["BloodType"] != null)
                    TextBox2.Text = Session["BloodType"].ToString();
                if (Session["Gender"] != null)
                    RadioButtonList1.SelectedValue = Session["Gender"].ToString();
                if (Session["ProblemDescription"] != null)
                    TextBox11.Text = Session["ProblemDescription"].ToString();
                if (Session["State"] != null)
                    DropDownList1.SelectedValue = Session["State"].ToString();
                if (Session["City"] != null)
                    TextBox9.Text = Session["City"].ToString();
                if (Session["Pincode"] != null)
                    TextBox10.Text = Session["Pincode"].ToString();
                if (Session["Address"] != null)
                    TextBox8.Text = Session["Address"].ToString();
                if (Session["Username"] != null)
```

```

    TextBox12.Text = Session["Username"].ToString();

    // Store the password in HiddenField for postbacks
    if (Session["Password"] != null)
        HiddenPassword.Value = Session["Password"].ToString();

    // Disable autocomplete on sensitive fields
    TextBox12.Attributes.Add("autocomplete", "off");
    TextBox13.Attributes.Add("autocomplete", "new-password");

    // Set the initial value of password textbox
    TextBox13.Attributes["value"] = HiddenPassword.Value;
}

else
{
    // Set the password field's value during postbacks
    TextBox13.Attributes["value"] = HiddenPassword.Value;
}
}

// Insert record function
protected void InsertRecord()
{
    string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

    // Retrieve form values
    string fullName = TextBox3.Text;
    string bloodType = TextBox2.Text;
    DateTime birthDate;
    DateTime.TryParse(TextBox1.Text, out birthDate);
    string gender = RadioButtonList1.SelectedValue;
    string contactNumber = TextBox4.Text;
    string email = TextBox5.Text;
    string problemDescription = TextBox11.Text;
    string state1 = DropDownList1.SelectedValue;
    string city = TextBox9.Text;
    string pinCode = TextBox10.Text;
    string fullAddress = TextBox8.Text;
    string username = TextBox12.Text;
    string password = TextBox13.Text;
    string Status = "pending";

    string query = @"INSERT INTO patient_appointment_tbl
        (p_name, p_bloodtype, p_gender, p_contact, p_emailid, p_problem,
        p_state, p_city, p_pincode, p_dob, p_address, p_username, p_password,
        p_appointment_time, p_appointment_date, status)
        VALUES (@FullName, @BloodType, @Gender, @Contact, @Email,
        @ProblemDescription, @State, @City, @PinCode, @BirthDate,
        @Address, @Username, @Password, CAST(GETDATE() AS TIME), GETDATE(),

```

```

    @Status);
}

using (SqlConnection connection = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        // Add parameters
        command.Parameters.AddWithValue("@FullName", fullName);
        command.Parameters.AddWithValue("@BloodType", bloodType);
        command.Parameters.AddWithValue("@Gender", gender);
        command.Parameters.AddWithValue("@Contact", contactNumber);
        command.Parameters.AddWithValue("@Email", email);
        command.Parameters.AddWithValue("@ProblemDescription", problemDescription);
        command.Parameters.AddWithValue("@State", state1);
        command.Parameters.AddWithValue("@City", city);
        command.Parameters.AddWithValue("@PinCode", pinCode);
        command.Parameters.AddWithValue("@BirthDate", birthDate);
        command.Parameters.AddWithValue("@Address", fullAddress);
        command.Parameters.AddWithValue("@Username", username);
        command.Parameters.AddWithValue("@Password", password);
        command.Parameters.AddWithValue("@Status", Status);

        try
        {
            connection.Open();
            int rowsAffected = command.ExecuteNonQuery();

            if (rowsAffected > 0)
            {

                lblModalMessage.Text = "Record inserted successfully!";
                ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
                    "$('#messageModal').modal('show');", true);
            }
            else
            {

                lblModalMessage.Text = "No record inserted. Check field values or constraints.";
                ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
                    "$('#messageModal').modal('show');", true);
            }
        }
        catch (SqlException ex)
        {
            string sanitizedMessage = ex.Message.Replace("""", "\\\"");
            ClientScript.RegisterStartupScript(this.GetType(), "alert", $"alert('SQL Error: {sanitizedMessage}');");
        }
    }
}

```

```

        }

    }

    // Clear fields

    // Redirects and stores form values in session
    protected void Button3_Click(object sender, EventArgs e)
    {
        StoreFormValuesInSession();
        Response.Redirect("select_appointment_date.aspx");
    }

    protected void Button4_Click(object sender, EventArgs e)
    {
        StoreFormValuesInSession();
        Response.Redirect("select_appointment_time.aspx");
    }

    // Stores form values in session
    private void StoreFormValuesInSession()
    {
        Session["FullName"] = TextBox3.Text;
        Session["ContactNumber"] = TextBox4.Text;
        Session["Email"] = TextBox5.Text;
        Session["BirthDate"] = TextBox1.Text;
        Session["BloodType"] = TextBox2.Text;
        Session["ProblemDescription"] = TextBox11.Text;
        Session["State"] = DropDownList1.SelectedValue;
        Session["City"] = TextBox9.Text;
        Session["Pincode"] = TextBox10.Text;
        Session["Address"] = TextBox8.Text;
        Session["Gender"] = RadioButtonList1.SelectedValue;
        Session["Username"] = TextBox12.Text;
        Session["Password"] = TextBox13.Text;
    }

    protected void Button1_Click1(object sender, EventArgs e)
    {
        InsertRecord();

        // Uncomment this line once data is inserting successfully
        Response.Redirect("select_appointment_date.aspx");
    }

    private bool IsUserAuthenticated(string username, string password)
    {
        string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;

```

```

        string query = "SELECT COUNT(*) FROM patient_signupin_tbl WHERE p_username =
@Username AND p_password = @Password";

        using (SqlConnection connection = new SqlConnection(connectionString))
{
    using (SqlCommand command = new SqlCommand(query, connection))
    {
        command.Parameters.AddWithValue("@Username", username);
        command.Parameters.AddWithValue("@Password", password);

        try
        {
            connection.Open();
            int userCount = (int)command.ExecuteScalar();
            return userCount > 0;
        }
        catch (Exception ex)
        {
            lblModalMessage.Text = "Error: " + ex.Message.Replace("'", "\\'");
            ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
"$('#messageModal').modal('show');", true);

            return false;
        }
    }
}

protected void LoginButton_Click(object sender, EventArgs e)
{
    if (IsUserAuthenticated(TextBox12.Text, TextBox13.Text))
    {
        // Show the remaining form fields if the user is authenticated
        AppointmentFormPanel.Visible = true;

        // Hide the Username, Password, and Login button after successful login
        TextBox12.Enabled = false;
        TextBox13.Enabled = true;
        LoginButton.Visible = false;

        // Store username in session for later use
        Session["Username"] = TextBox12.Text;
        Session["Password"] = TextBox13.Text;

        HiddenPassword.Value = Session["Password"].ToString();
    }
    else
    {
        AppointmentFormPanel.Visible = true;
    }
}

```

```

        }

    }

protected void Button3_Click1(object sender, EventArgs e)
{
    string username = TextBox12.Text;
    string password = TextBox13.Text;

    if (CheckAndLoadPreviousAppointment(username, password))
    {
        // Load the form with existing details
        AppointmentFormPanel.Visible = true;
    }
    else
    {
        lblModalMessage.Text = "You are new here, we do not have any details of you.";
        ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
        "$('#messageModal').modal('show');", true);
    }
}

private bool CheckAndLoadPreviousAppointment(string username, string password)
{
    string connectionString =
ConfigurationManager.ConnectionStrings["defaultDB"].ConnectionString;
    string query = @"SELECT p_name, p_bloodtype, p_gender, p_contact, p_emailid, p_problem,
                    p_state, p_city, p_pincode, p_dob, p_address
                    FROM patient_appointment_tbl
                    WHERE p_username = @Username AND p_password = @Password";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@Username", username);
            command.Parameters.AddWithValue("@Password", password);

            try
            {
                connection.Open();
                SqlDataReader reader = command.ExecuteReader();

                if (reader.HasRows && reader.Read())
                {
                    // Populate form fields
                    TextBox3.Text = reader["p_name"].ToString();
                    TextBox2.Text = reader["p_bloodtype"].ToString();
                    RadioButtonList1.SelectedValue = reader["p_gender"].ToString();
                    TextBox4.Text = reader["p_contact"].ToString();
                    TextBox5.Text = reader["p_emailid"].ToString();
                    TextBox11.Text = reader["p_problem"].ToString();
                }
            }
        }
    }
}

```

```
DropDownList1.SelectedValue = reader["p_state"].ToString();
TextBox9.Text = reader["p_city"].ToString();
TextBox10.Text = reader["p_pincode"].ToString();
TextBox1.Text = Convert.ToDateTime(reader["p_dob"]).ToString("yyyy-MM-dd");
TextBox8.Text = reader["p_address"].ToString();
return true;
}
}
catch (Exception ex)
{
    lblModalMessage.Text = "Error: " + ex.Message.Replace("'", "\\'");
    ScriptManager.RegisterStartupScript(this, GetType(), "Popup",
    $"('#messageModal').modal('show');", true);
}

}
}

return false;
}
}
```

## 4.6 Test Cases

1) Patient signup/sign in-

Test-Case ID	Condition	Test Data	Expected Outcome	Actual Outcome	Pass /Fail	Remarks
1	User wants signup	Email:mratharv3@gmail.com Username: Atharv_Chavan 1213 Password: Atharv@12345 Confirm password:Atharv@12345	Alert message “Sign Up Successfully... Go to User Login to Login...!!” Should be shown.	Alert Message “Sign Up Successfully... Go to User Login to Login...!!” is shown	Pass	
		Username: Atharv_Chavan Password: Atharv@	Alert message “Password must contain Uppercase, lowercase , special symbol, digits” should be shown.	Alert message “Password must contain Uppercase, lower case, special symbol, digits” is shown	Pass	

		Username: Atharv_chavan 1213 Password:-	Alert message “Enter password” Should be shown.	Alert message “Enter password” is shown	Pass	
		If user enter duplicate data like username or email id that already exist in system.	Alert message “Username is already exist ,Try another username ” should be shown	Alert message “Username is already exist, Try another username” is shown	Pass	
		If user forgot password.User can enter email id.	Alert message “Password sent on registered email id” should be shown.	Alert message “Password sent on registered email id” is shown	Pass	

2)Patient Appointment form-

Test- Case ID	Condition	Test Data	Expected Outcome	Actual Outcome	Pass /Fail	Remarks
2	User wants to book an Appointment	If you are existing user.	All data fields will automatically fill when click on button.	All data fields are automatically filling when click on button	Pass	
		If you are new user	Alert message “You are new here, we do not have any details of you.” Should be shown.	Alert message “You are new here, we do not have any details of you.” is shown.	Pass	
		If you select red color date.	Alert message “Doctor not available on this date” should be shown.	Alert message “Doctor not available on this date” is shown.	Pass	
		If you select date with out of 15 days from current date.	Alert message “Please select a date within the next 14 days from current date” should be shown.	Alert message “Please select a date within the next 14 days from current date” is shown.	Pass	

		If you select Sunday's date.	Alert message "Appointments cannot be booked on Sundays." should be shown.	Alert message "Appointments cannot be booked on Sundays." is shown.	Pass	
		If you select regular date within 15 days from current date and non-red time slot.	Alert message "Appointment confirmation email sent successfully!" should be shown and email will come	Alert message "Appointment confirmation email sent successfully!" is shown and email came	Pass	
		If you select red time slot.	Alert message "This slot is already booked on --date." should be shown.	Alert message "This slot is already booked on --date. "is shown.	Pass	

3)Online consultancy form-

Test-Case ID	Condition	Test Data	Expected Outcome	Actual Outcome	Pass /Fail	Remarks
3	User wants to request for Online consultancy appointment	If user fill all the fields correctly.	Alert message "Consultancy appointment requested successfully" should be shown.	Alert message "Consultancy appointment requested successfully" is shown.	Pass	
		If you not fill all the fields.	Alert message "Please fill in all mandatory fields." Should be shown.	Alert message "Please fill in all mandatory fields." is shown.	Pass	

4)Patient Appointment status -

<b>Test-Case ID</b>	<b>Condition</b>	<b>Test Data</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass /Fail</b>	<b>Remarks</b>
4	User wants to check their appointment status.	If user try to cancel the appointment which is booked before 24 hours.	Alert message "You cannot cancel the appointment as the time has passed 24 hour." should be shown.	Alert message "You cannot cancel the appointment as the time has passed 24 hour." is shown.	Pass	
		If user cancel their appointment within 24 hours.	Alert message "Appointment canceled successfully." Should be shown.	Alert message "Appointment canceled successfully." Is shown.	Pass	

5)Doctor Login -

<b>Test-Case ID</b>	<b>Condition</b>	<b>Test Data</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass /Fail</b>	<b>Remarks</b>
5	Doctor wants to login into admin panel	If doctor entered correct details: Username: amolzope Password:@amolzope	Should redirect to doctor panel.	Redirecting to the doctor panel	Pass	
		If other than doctor try to entered details: Username:atharv Password:@atharv	Alert message "Invalid username or password." Should be shown.	Alert message "Invalid username or password." Is shown.	Pass	

6) Set schedule-

<b>Test-Case ID</b>	<b>Condition</b>	<b>Test Data</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass /Fail</b>	<b>Remarks</b>
6	Doctor wants to set the schedule.	If doctor directly click on ‘Add date button’	Message should show ‘Invalid date format’.	Message should show ‘Invalid date format’.	Pass	
		If doctor click on ‘delete’ button	Alert message “Holiday date canceled” Should be shown.	Alert message “Holiday date canceled” Is shown.	Pass	
		If doctor want emergency leave then click on time slots.	Alert message “Email has sent to patient for that time slot” should be shown	Alert message “Email has sent to patient for that time slot” is shown.	Pass	

# **CONCLUSION**

In conclusion, the E-MEDI APPOINTMENT BOOKING provides a seamless, efficient, and convenient solution for both patients and healthcare providers. By digitizing the appointment scheduling process, it reduces the time and effort involved in manual bookings, minimizes errors, and offers patients flexibility in choosing the right doctor based on availability, specialization, and reviews. The system ensures that patient data is securely managed while offering a user-friendly interface that enhances the overall experience. Furthermore, it streamlines clinic operations, allowing doctors to manage appointments and patient records effectively. This project bridges the gap between patients and healthcare professionals, contributing to the modernization of healthcare services and making healthcare more accessible to all.

# REFERENCES

## Websites & Blogs:

- HealthIT.gov – A valuable resource providing articles, case studies, and official guidelines on how information technology enhances healthcare efficiency, patient management, and overall medical practices.
- IEEE Xplore Digital Library – A premier repository for scholarly research papers on telemedicine, AI in healthcare, electronic health records (EHR), and innovative medical technologies.

## Books & Journals:

- "Healthcare Information Systems: A Practical Approach for Health Care Management"  
*Authors:* Karen A. Wager, Frances W. Lee, and John P. Glaser
  - A comprehensive guide covering the role of IT in healthcare management, implementation strategies, and the impact of digital transformation in medical environments.

## Technical Documentation & Online Resources:

- Stack Overflow – A developer-driven Q&A platform offering real-world solutions to coding challenges encountered while developing healthcare applications, including appointment booking systems and secure patient portals.
- Official API Documentation (FHIR, HL7, etc.) – Essential technical references for integrating standardized healthcare data exchange protocols into modern medical software solutions.

## Tools & AI Resources:

- StarUML / Draw.io – Industry-standard tools for designing UML and ERD diagrams, crucial for visualizing system architecture and database relationships.
- ChatGPT & DeepSeek – Advanced AI-driven platforms for code optimization, debugging, and generating structured insights related to healthcare IT solutions.
- AI-Powered Development Tools – Cutting-edge technologies leveraging artificial intelligence to enhance software development efficiency, error detection, and process automation in medical applications.