Ayurvedic Doctor Appointment System

A Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Mr. Tukaram Satyawan Manjarekar

Seat Number :

Under the esteemed guidance of

Mr. K . A . Kubal

Head of Department

DEPARTMENT OF INFORMATION TECHNOLOGY

SANT RAWOOL MAHARAJ MAHAVIDYALAYA, KUDAL

(*Affiliated to University of Mumbai*)

KUDAL, 416520

MAHARASHTRA

2024-25

SANT RAWOOL MAHARAJ MAHAVIDYALAYA, KUDAL

(*Affiliated to University of Mumbai*)

KUDAL-MAHARASHTRA-416520

DEPARTMENT OF INFORMATION TECHNOLOGY

K.M.S.P. MANDAL'S

SANT RAWOOL MAHARAJ MAHAVIDYALAYA,

KUDAL

(NAAC Accreditation "B+" Grade)

.....

CERTIFICATE

The work done is satisfactory and is presented as per the specifications.

This is to certify that the project entitled, **"Ayurvedic Doctor Appointment System"**, is bonafied work of

Mr. Tukaram Satyawan Manjarekar

bearing Seat.No:

submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY

from University of Mumbai.

PROJECT GUIDE **EXTERNAL EXAMINER** **HEAD OF DEPARTMENT**

DATE:

College Seal

DECLARATION

To,

The Head,

Department of Information Technology,

S. R. M. College, Kudal.

Respected Sir,

I hereby declare that the project entitled, "Ayurvedic Doctor Appointment System" done at Sant Rawool Maharaj Mahavidyalaya Kudal, 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 other than me, no one has submitted to any other university.

The project is done in partial fulfilment 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.

Date: (Mr. Tukaram Satyawan Manjarekar)

Place:Kudal

ABSTRACT

The Ayurvedic Doctor Appointment System is an web application that is developed for improving patient experience and reduce wait times By implementing a streamlined appointment system, the goal is to enhance the overall patient experience, leading to shorter wait times and greater satisfaction during visits to Ayurvedic clinics.

The project employs a MySQL database to manage the data regarding the appointment booking, treatment, payment, etc. The use of MySQL databases provide advance query capabilities, a structured way to store data, and that lead to high performance. The Ayurvedic Doctor Appointment System aim to provide a structured framework for scheduling appointments with Ayurvedic practitioners, ensuring both patients and doctors can efficiently manage their time and commitments.

Date: (Mr. Tukaram Satyawan Manjarekar)

Place:Kudal

ACKNOWLEDGEMENT

Many people have helped me during my project and each of their contribution has been valuable. The timely completion of this project is mainly due to the interest and worth helping of **Mr. K.A.Kubal(Head of Department)** their help will be remembered forever. I would like to thank them to guide me during my project work. I would like to thank all those who directly and indirectly helped in completion of this project.

Thank You.

Mr. Tukaram Satyawan Manjarekar

T.Y.B.Sc. Information Technology

PREFACE

This report presents the development process and logic behind the Ayurvedic Doctor Appointment System, built using React js, PHP , Bootstrap, and MySQL. It includes a detailed explanation of the key functionalities, such as user authentication, appointment booking, payment, etc. To ensure clarity, various diagrams and screen layouts have been provided to illustrate the system's structure and user flow.

While every effort has been made to prepare this report with precision, it is important to note that the project is still open to enhancements and refinements based on future requirements. This document aims to offer a comprehensive understanding of the system, though further improvements may continue to evolve with additional features and optimization strategies.

Table of Contents

Table of Contents of Conceptual Models

CHAPTER 1

INTRODUCTION

1.1 Background:

Ayurvedic appointments traditionally involved in-person consultations where the doctor assessed the patient's unique constitution through physical exams, pulse reading, and detailed lifestyle discussions. Treatment plans included personalized herbal remedies, dietary changes, yoga, and meditation to restore balance and health. With online consultations, this process has become more accessible, allowing patients to connect with Ayurvedic doctors remotely. While physical assessments like pulse reading may be limited, doctors can still evaluate symptoms, provide personalized care, and offer treatment plans. Online consultations make Ayurveda more convenient, especially for managing chronic conditions and promoting holistic wellness globally.

1.2 Objectives:

An appointment system for Ayurvedic doctors was designed to provide a structured framework for scheduling appointments, ensuring both patients and doctors can efficiently manage their time and commitments. This streamlined system aims to improve patient experience and reduce wait times, leading to greater satisfaction during visits to Ayurvedic clinics. By optimizing schedules, the system also enhances doctor productivity and scheduling efficiency, reducing downtime and maximizing the number of patients seen without compromising the quality of care provided.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose:

A user-friendly appointment system for Ayurvedic experts was developed to be intuitive and accessible, ensuring that both patients and practitioners can navigate it easily, thus encouraging its use and facilitating better healthcare management. The system streamlines appointment scheduling and management by integrating features such as reminders and easy rescheduling, simplifying the appointment process and making it hassle-free for both patients and doctors.

1.3.2 Scope:

The system enables patient registration and profiling, allowing patients to create profiles and provide necessary information and preferences to enhance personalized care. Patients can conveniently book appointments online, viewing available times and selecting their preferred Ayurvedic doctor based on expertise and availability. The system also includes a management feature for Ayurvedic practitioners to set their availability, ensuring that patient bookings are aligned with their schedules. Additionally, secure payment processing options are incorporated, allowing patients to settle bills efficiently and streamlining the financial aspect of the appointment process.

1.3.3 Applicability:

The appointment system is versatile and can be implemented across various Ayurvedic facilities, including clinics and hospitals, to enhance patient management and operational efficiency. It also benefits private practices of Ayurvedic doctors by enabling them to manage patient appointments more effectively, allowing them to focus on patient care rather than administrative tasks. Furthermore, healthcare organizations that offer Ayurvedic services can utilize the system to integrate these services seamlessly with other healthcare practices, providing a comprehensive and streamlined patient experience.

1.4 Achievements:

The appointment system has yielded numerous benefits, including improved patient satisfaction due to smoother appointment processes, which have become a hallmark of Ayurvedic care. Additionally, efficient scheduling and reminders have significantly red uced wait times and no-shows,ensuring that patients receive timely care. The streamlined processes have also increased doctor productivity, enabling them to see more patients and provide better care. Furthermore, the system's user-friendly design has enhanced its usability, making it accessible to all and encouraging wider adoption.

CHAPTER 2

SURVEY OF TECHNOLOGIES

Frontend:

React JS:

Building reusable UI components for the appointment booking system makes it easier to manage complex interfaces. Modular React components were developed for various parts of the booking system, such as input fields, calendars, and buttons, which can be reused across the application. This approach simplifies maintenance and updates, ensuring a consistent user experience. Additionally, a user-friendly interface was created for patients to view and manage their appointments, featuring an intuitive dashboard with seamless navigation and quick access to information. Furthermore, a dedicated interface was designed for doctors to manage their schedules and appointments, enabling them to easily view appointments, adjust availability, and communicate with patients, facilitating better time management and improving the overall workflow.

Bootstrap

Bootstrap's pre-built UI components ensured layout and design consistency, reducing development time and effort. This enabled a uniform design throughout the application, making it easy to maintain and update. A responsive design was also ensured, providing a seamless user experience across devices. The layout adapts to different screen sizes and devices, optimizing the user experience.

CSS:

Utilize CSS to enhance the aesthetics of the webpage, employing color schemes, fonts, and spacing that align with the holistic and calming principles of Ayurveda. This will create an inviting environment for users and improve the overall user experience.

Backend:

PHP

PHP will help manage appointment booking, scheduling, and reminders in your **Ayurvedic Doctor Appointment System** by handling backend processes efficiently. It will allow patients to book appointments while checking doctor availability to prevent scheduling conflicts. PHP will also ensure data security using prepared statements and role-based access control. Additionally, it provides a scalable foundation for future enhancements like video consultations, UPI payments, and treatment reports.

Database:

MySQL:

MySQL will be used to efficiently manage appointment data in your **Ayurvedic Doctor Appointment System** by structuring and storing information systematically. The **bookedappointments** table will handle patient bookings, allowing quick retrieval and updates to support appointment scheduling. Patient details and appointment history will be stored in the **user** and **bookedappointments** tables, enabling easy access to past interactions for better treatment planning. Doctor schedules and availability will be maintained in the **doctors** table, ensuring real-time updates and efficient queries to prevent scheduling conflicts. MySQL's relational structure will allow seamless data management, ensuring optimized performance for appointment booking and retrieval.

Software Process Models

For my Ayurvedic Doctor Appointment System project, I have chosen to use the Incremental development model.

Why Incremental Model?

The incremental model is suitable for this project because it allows for the development of the system in smaller, manageable segments. Each increment can be developed, tested, and deployed, enabling immediate feedback from users and stakeholders. This ensures that adjustments can be made based on real-world usage before the entire system is finalized.

Why Incremental Model is Suitable for Ayurvedic Doctor Appointment System?

The incremental model aligns well with the needs of the Ayurvedic Doctor Appointment System, as it can accommodate the evolving requirements of both patients and doctors. By delivering functional increments, the system can be adjusted and refined based on user feedback, ensuring it meets the unique needs of Ayurvedic practices.

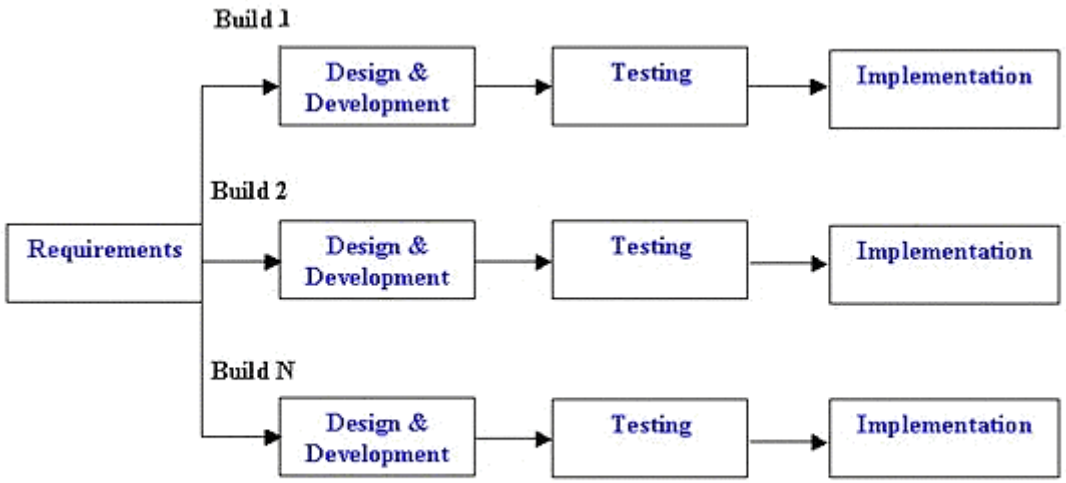
Benefits of Incremental Model:

1. Enhanced Flexibility: The ability to adapt to changes in requirements based on user feedback helps ensure that the final product is user-focused.

2. Reduced Risk: By breaking the project into smaller increments, risks are minimized, as potential issues can be identified and addressed early in the development process.

3. Faster Time to Market: Each increment can be deployed and used independently, allowing for quicker delivery of functionalities to users.

4. Continuous Improvement: Regular iterations facilitate ongoing enhancements, leading to a more polished and effective system by the end of development.



Incremental Life Cycle Model

CHAPTER 3

REQUIREMENTS AND ANALYSIS

3.1 Problem Definition

a. Difficulty in Managing Appointments for Ayurvedic Doctors: Ayurvedic practitioners often encounter challenges in efficiently scheduling and managing patient appointments due to a lack of centralized systems. This can lead to confusion regarding patient schedules, impacting their ability to provide timely care.

b. Inefficient Use of Time and Resources: The absence of a streamlined appointment management system results in inefficiencies that affect both doctors and patients. Doctors may find themselves overbooked or underutilized, while patients may experience longer wait times, leading to frustration and dissatisfaction.

c. Patients Face Challenges in Booking Appointments: Patients frequently struggle with cumbersome booking processes that can involve long wait times on the phone or unclear online forms. This complexity can lead to frustration and delays in receiving necessary care, which is counterproductive to their health needs.

d. Lack of Organization and Structure in Appointment Scheduling: Without a structured appointment scheduling system, Ayurvedic practices may face disorganization, leading to overlapping appointments and missed opportunities for patient care. This chaos can hinder the overall efficiency of healthcare delivery.

3.2 Requirements Specification

a. User-Friendly Interface for Patients: The appointment system must feature a simple and intuitive interface that allows patients to easily navigate through options to book, cancel, or reschedule appointments without confusion or technical difficulties.

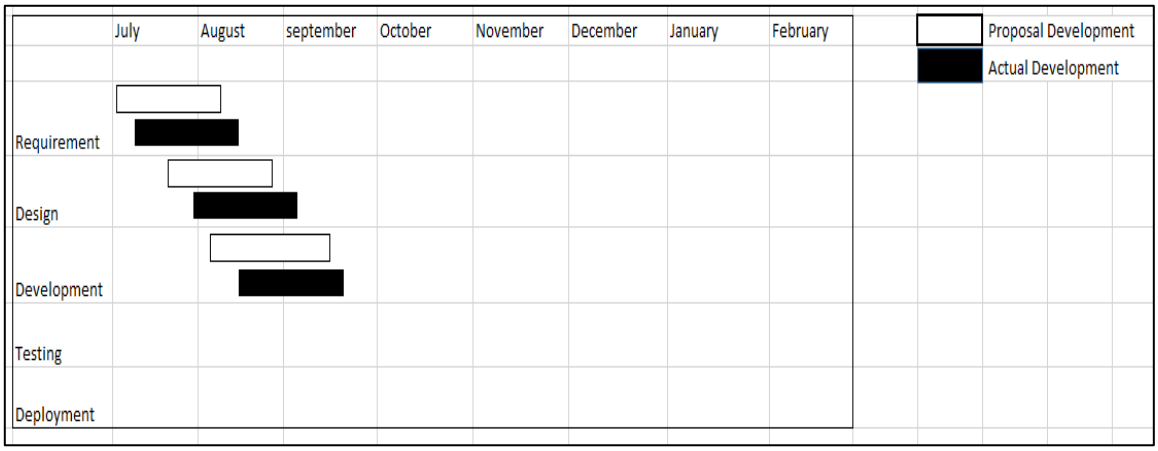
b. Easy Management for Doctors: Doctors should have access to effective tools that allow them to manage their appointments seamlessly, including functionalities to view patient records, set availability, and track their schedule efficiently.

c. Automated Reminders and Notifications: An automated reminder system is essential to send timely notifications to both patients and doctors, reducing no-shows and ensuring everyone is informed of upcoming appointments, thereby improving attendance rates.

d. Secure Storage of Patient Data: The system must prioritize the protection of patient information through secure data storage methods, ensuring compliance with privacy regulations and preventing unauthorized access to sensitive health records.

e. Scalable and Reliable System: The appointment system needs to be designed to handle an increasing number of users and data without performance degradation. Scalability ensures that as the practice grows, the system can accommodate new patients and doctors efficiently.

3.3 Planning and Scheduling



3.4 Software and Hardware Requirements

Software Requirements(Developer side) :

a. Front-end: The development will utilize React JS, CSS Bootstrap to build a responsive and engaging user interface that meets the needs of both patients and doctors.

b. Back-end: PHP will serve as the backbone of the server-side logic and API management, ensuring smooth communication between the front-end and the database.

c. Database: A MySQL database will be implemented to store and manage all appointment-related data, allowing for efficient retrieval and manipulation of records for both patients and doctors.

Software Requirements(Client side) :

Browser : Google Chrome ,Microsoft Edge.

Operating System : Windows 7 and Above.

Hardware Requirements : Processor : Core i3 & above

Memory : Minimum 2 GB of RAM

3.5 Preliminary Product Description

a. Web-Based Appointment System for Ayurvedic Doctors:

This comprehensive platform will facilitate the efficient management of patient appointments, addressing the unique challenges faced by Ayurvedic practitioners.

b. Patient Features:

Patients will have the capability to book , providing them with flexibility and control over their healthcare scheduling.

c. Doctor Features:

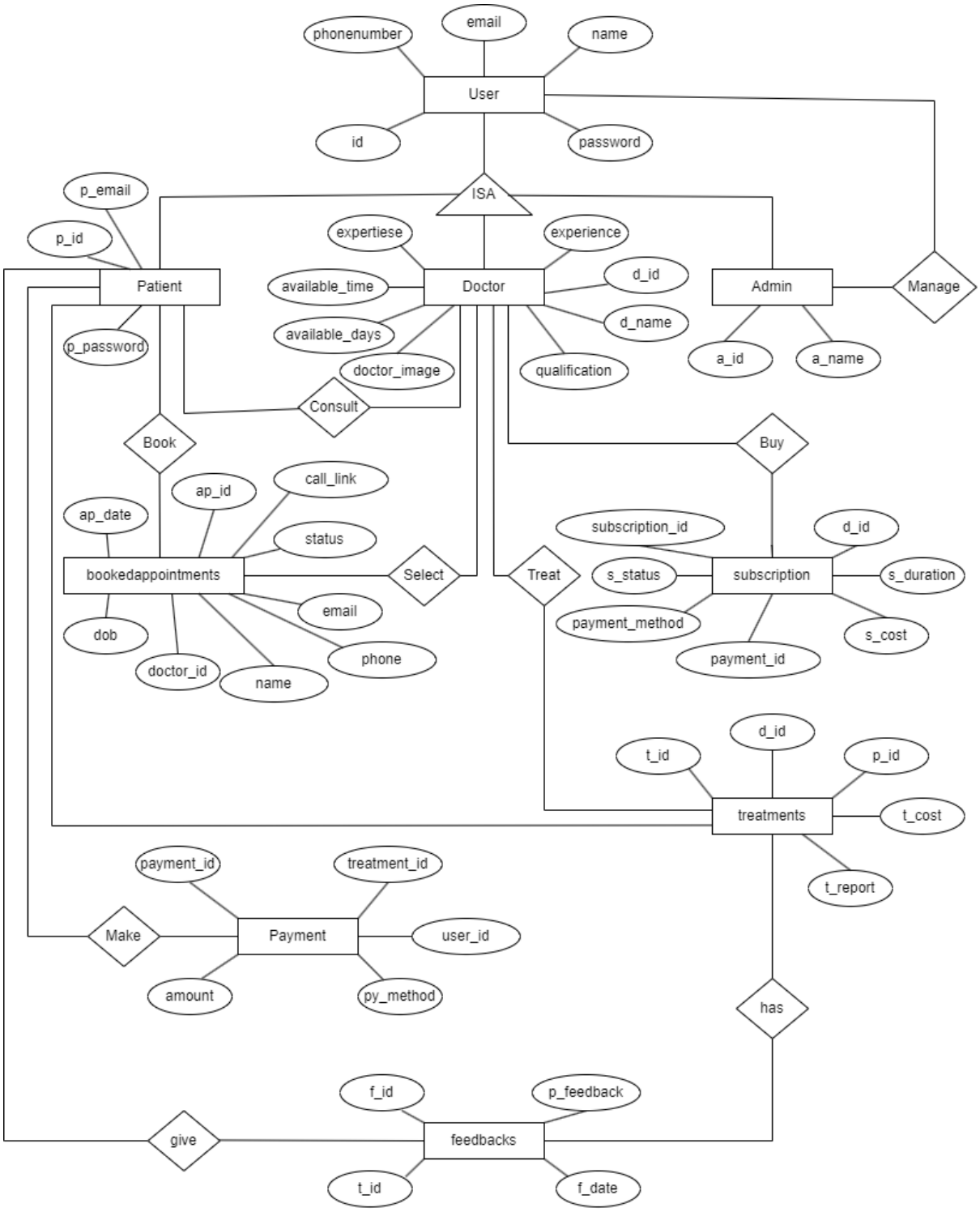
The system will provide doctors with essential tools to manage their appointments, access patient data securely, and update their availability as needed, improving their workflow and patient interaction.

d. Security and Scalability:

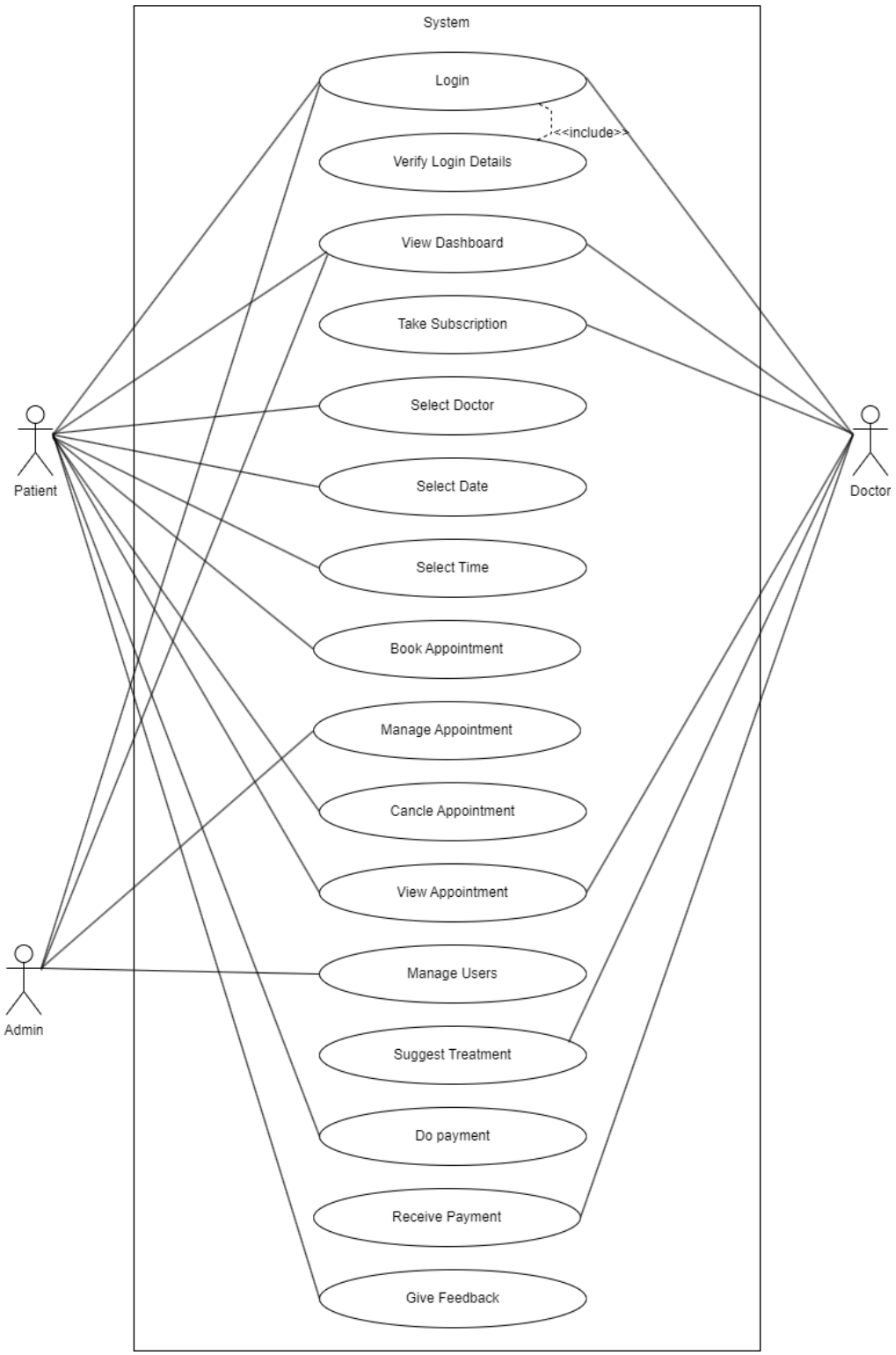
The appointment system will be built with a focus on security and scalability, ensuring patient data is protected while being robust enough to accommodate growth as user demand increases.

3.6 Conceptual Models

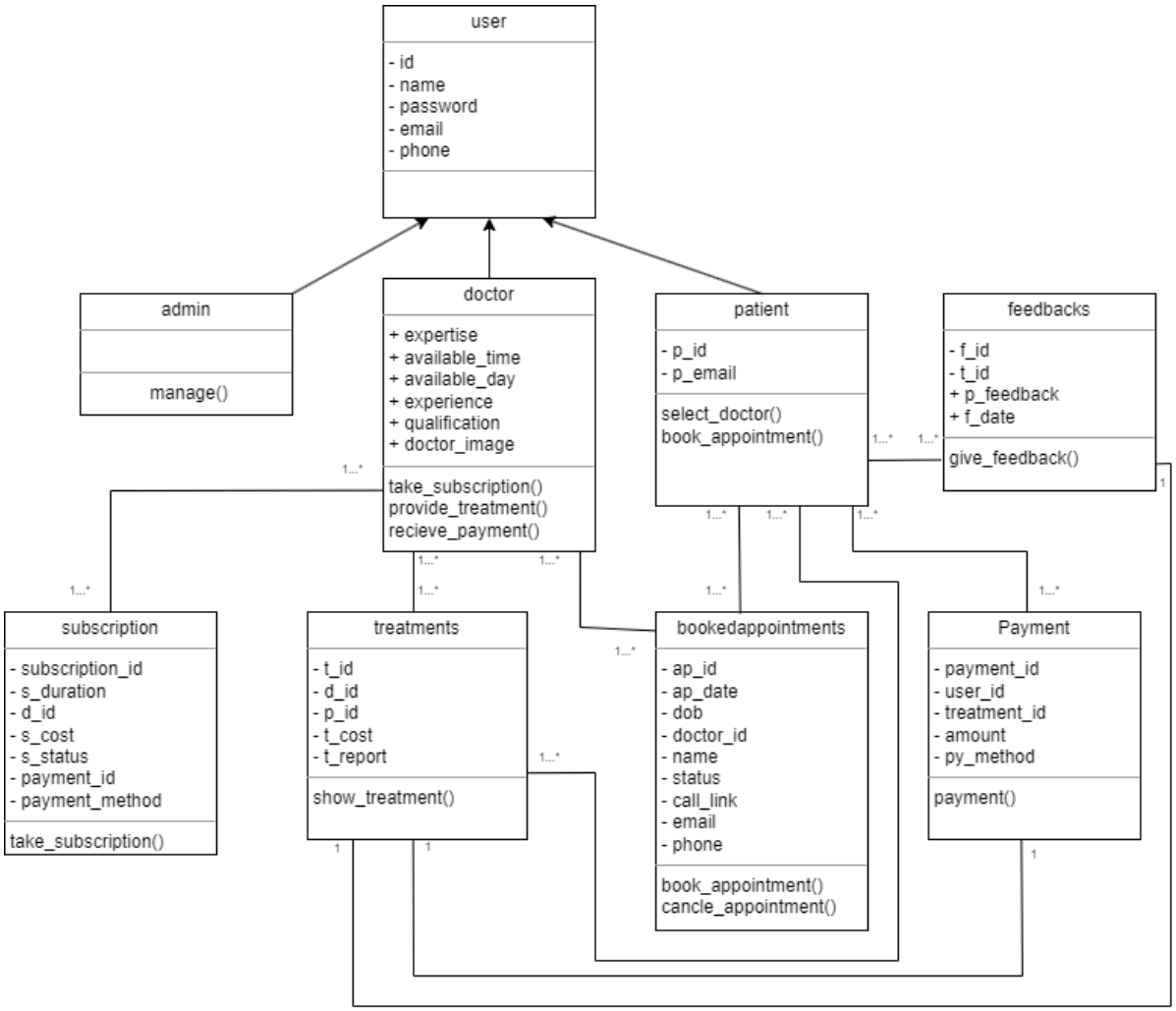
1. Entity Relationship Diagram



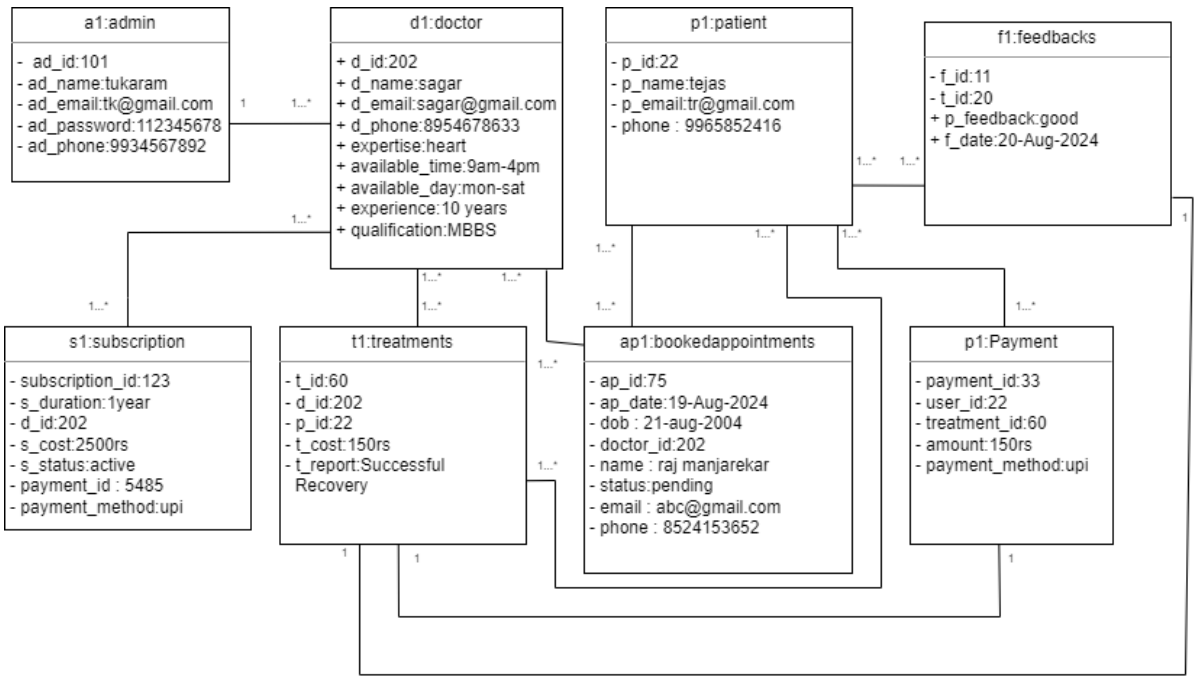
2. Use Case Diagram



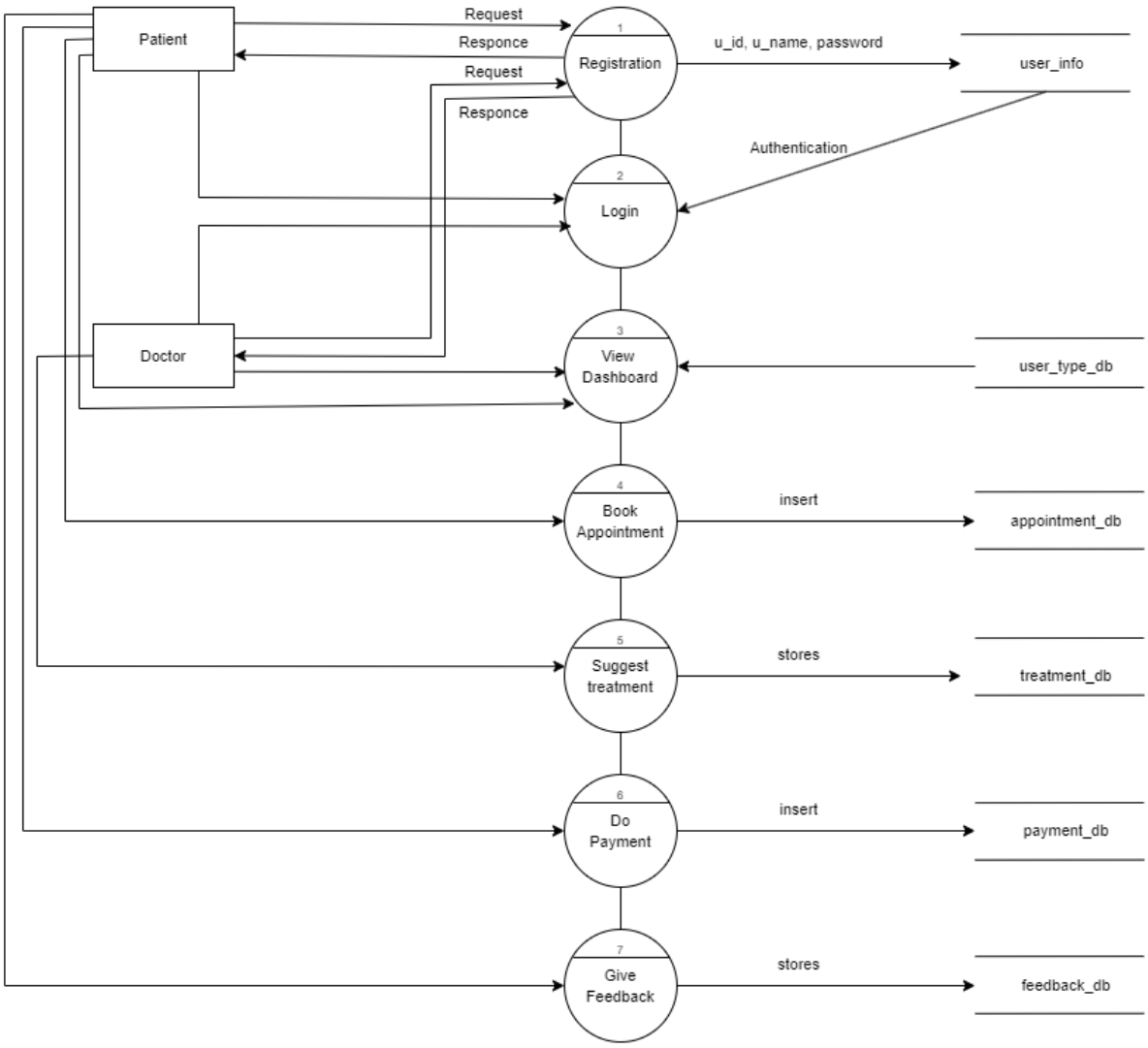
3. Class Diagram



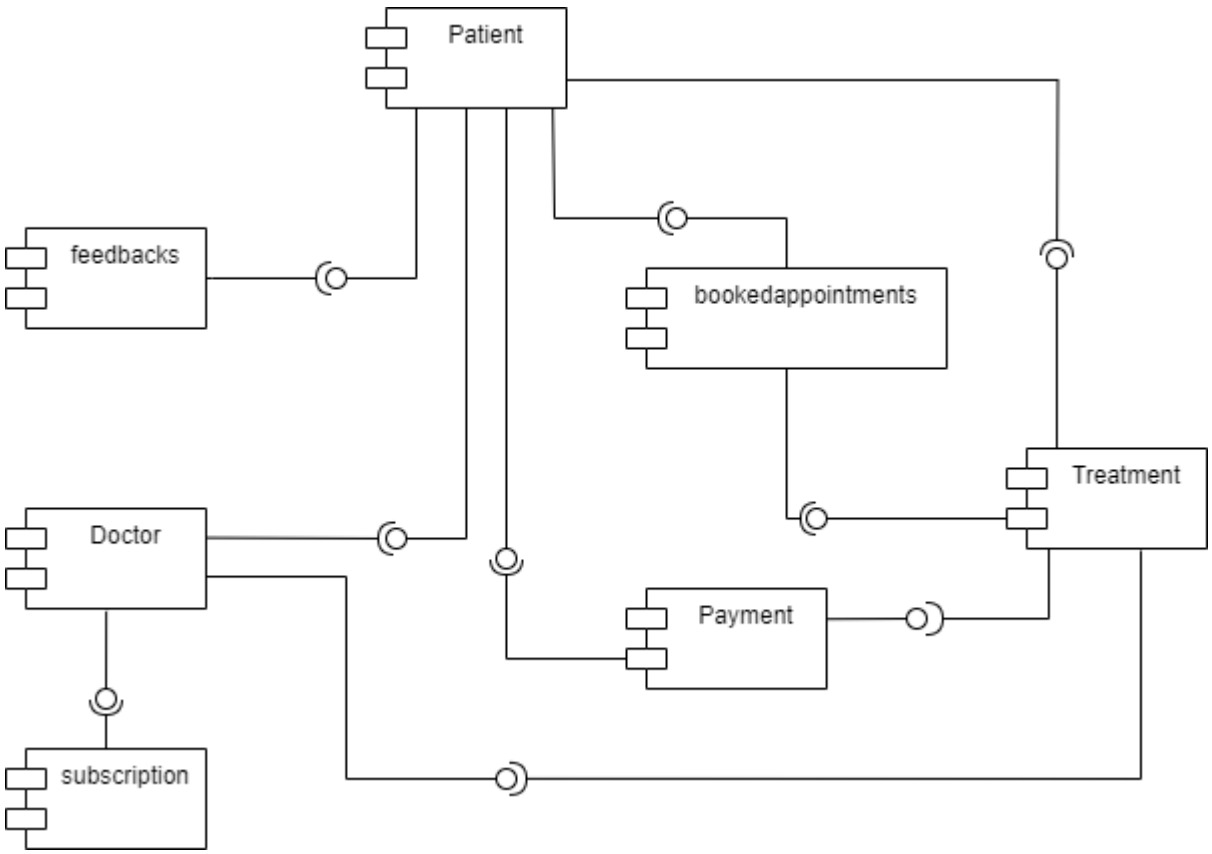
4. Object Diagram



5. Data Flow Diagram

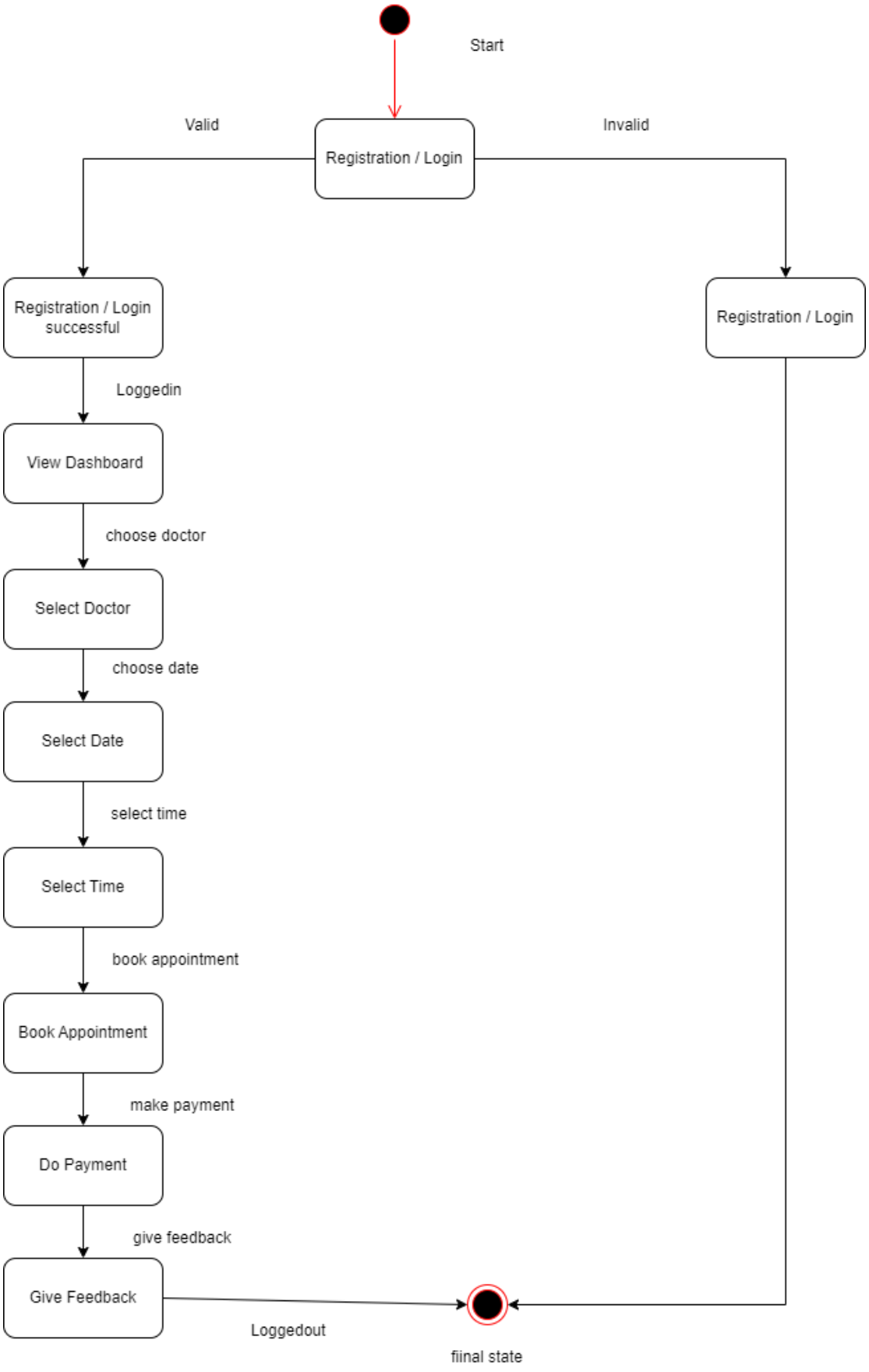


6. Component Diagram

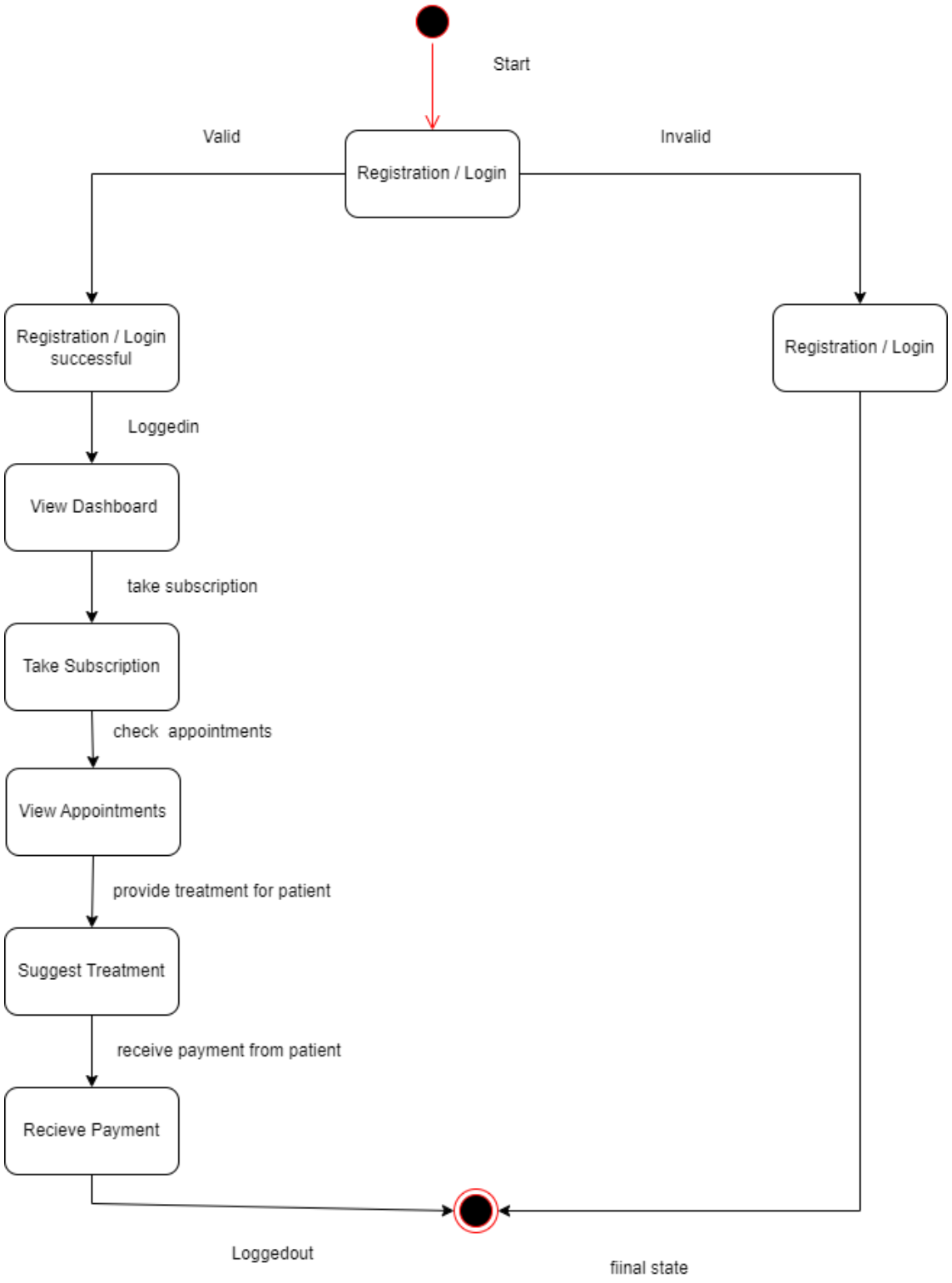


7. State Transition Diagra

For Patient

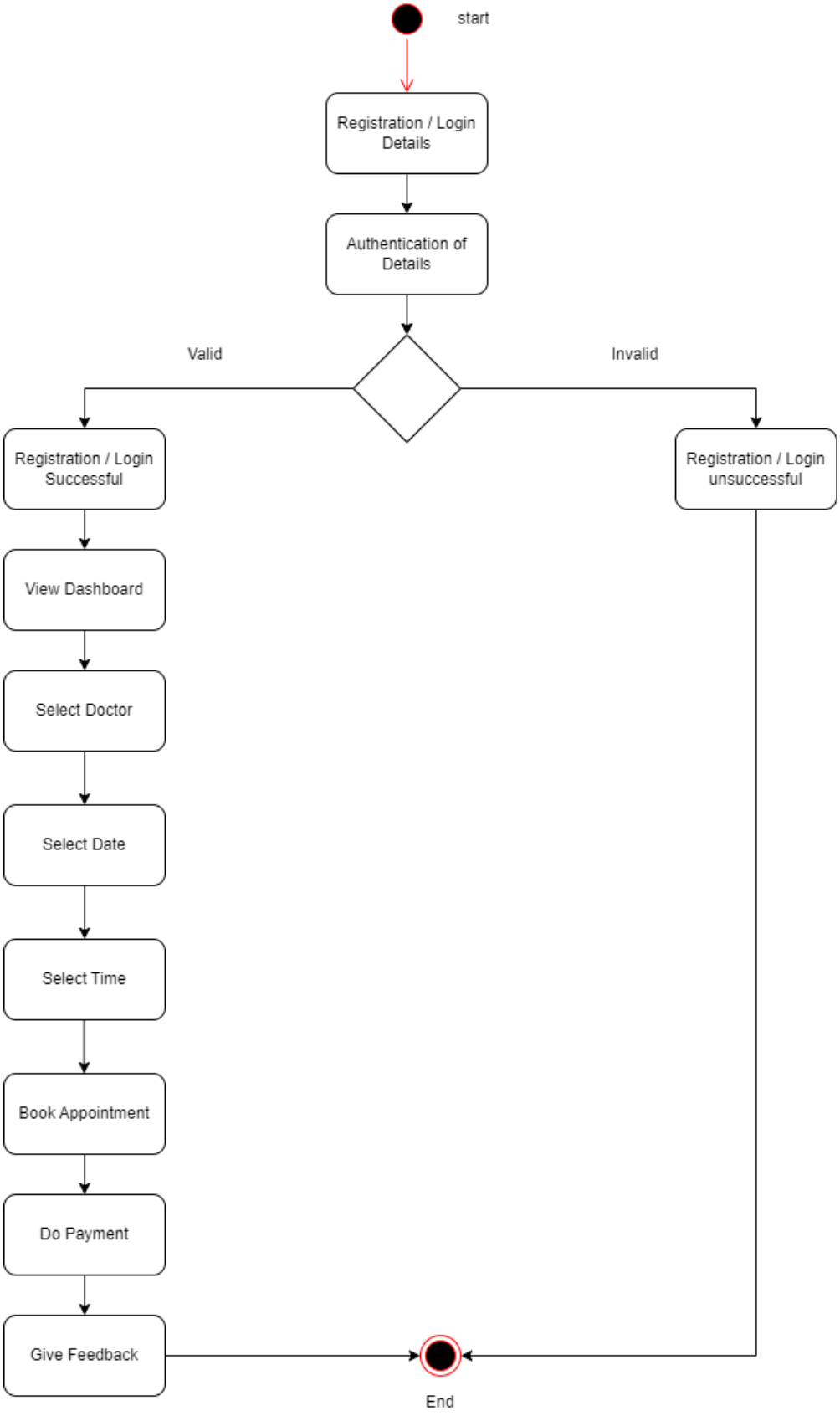


For Doctor

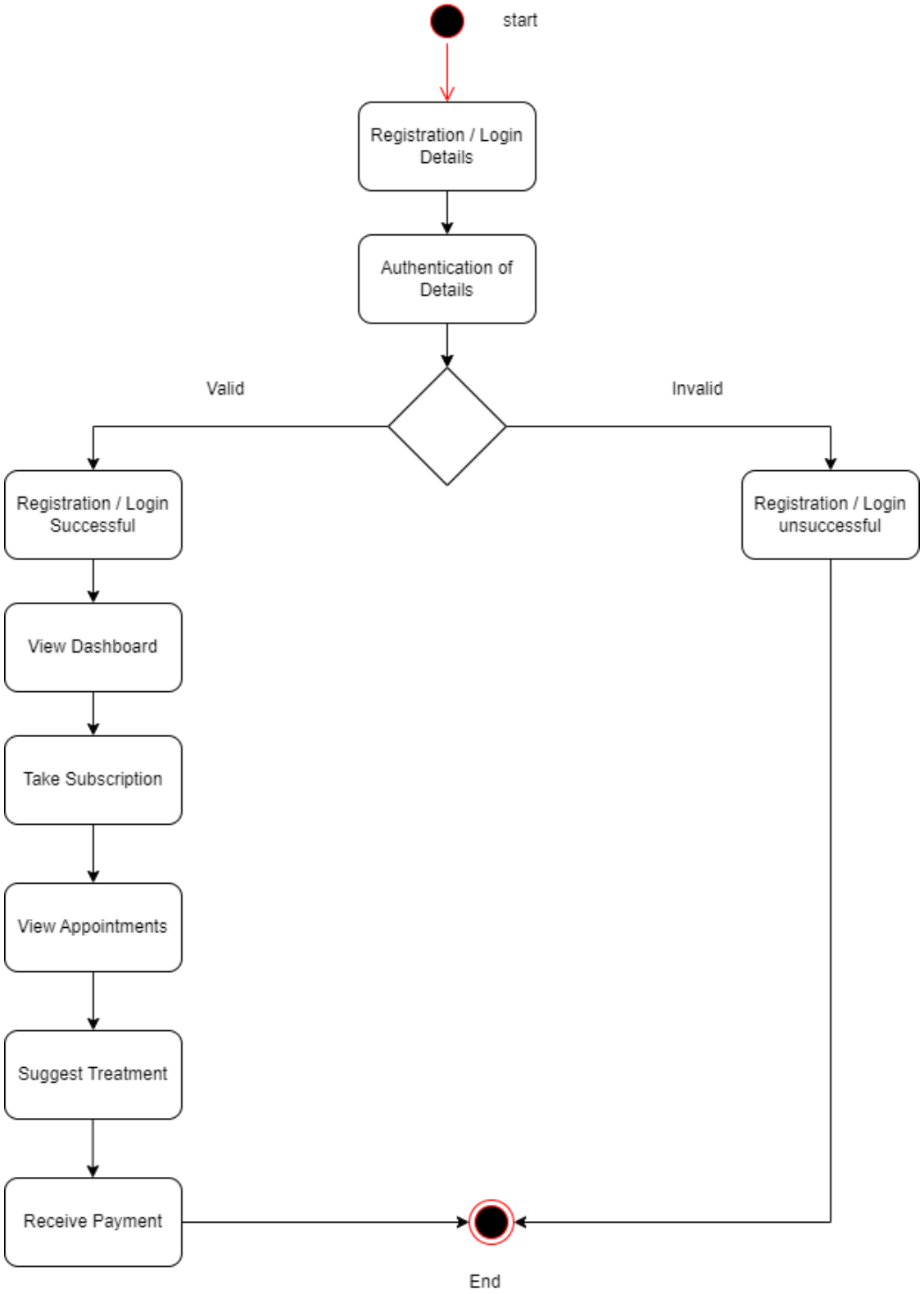

Start
Valid
Registration / Login
Invalid
Registration / Login successful
Registration / Login
Loggedin
View Dashboard
take subscription
Take Subscription
check appointments
View Appointments
provide treatment for patient
Suggest Treatment
receive payment from patient
Recieve Payment
Loggedout
fiinal state

8. Activity Diagram

For Patient

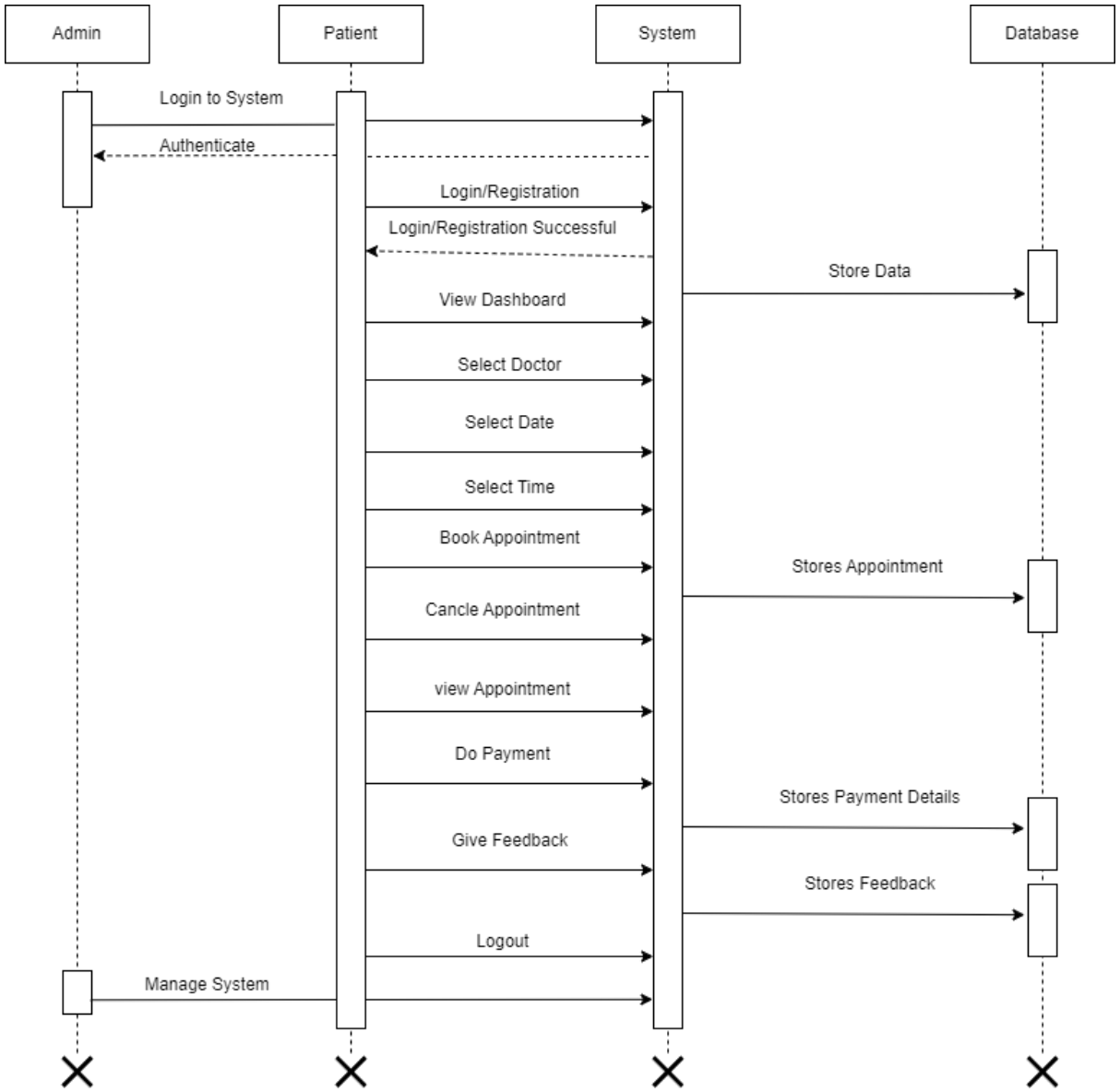


For Doctor

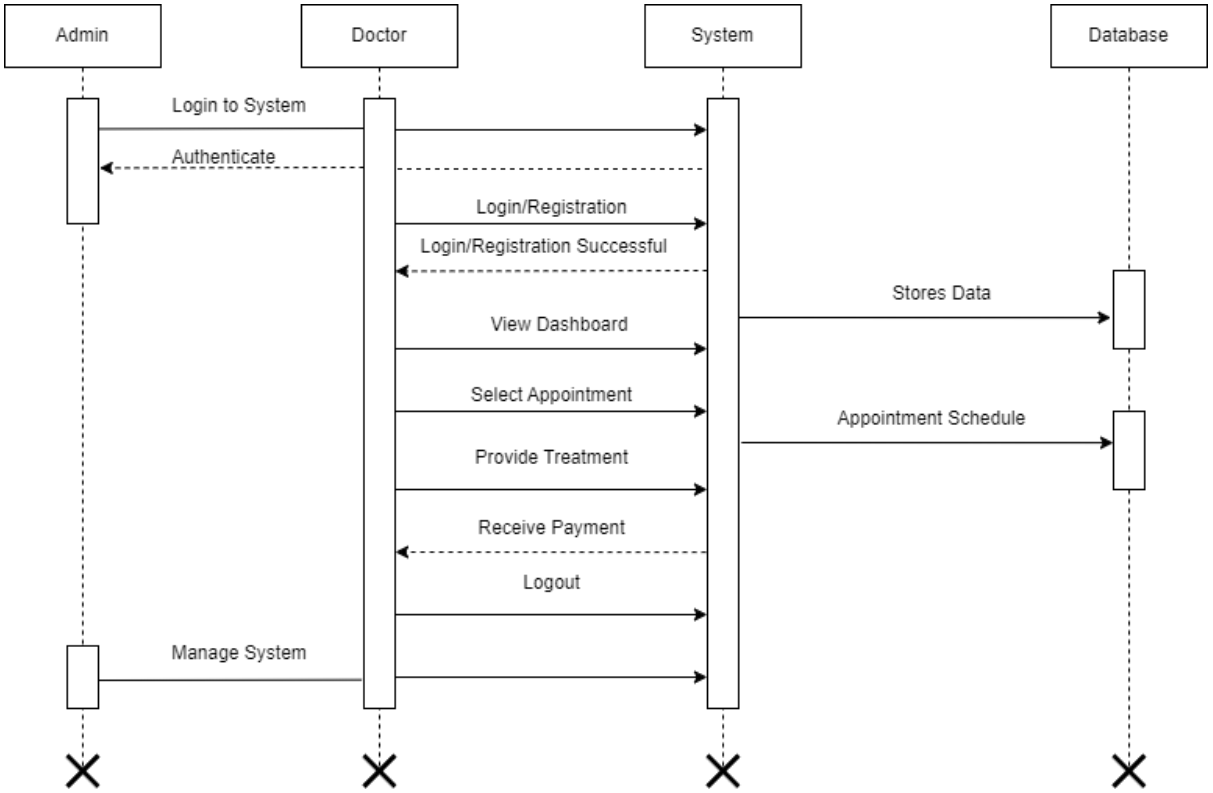

start
Registration / Login Details
Authentication of Details
Valid
Invalid
Registration / Login Successful
Registration / Login unsuccessful
View Dashboard
Take Subscription
View Appointments
Suggest Treatment
Receive Payment
End

9 Sequence Diagram

Patient



For Doctor


Admin
Doctor
System
Database
Login to System
Authenticate
Login/Registration
Login/Registration Successful
Stores Data
View Dashboard
Select Appointment
Appointment Schedule
Provide Treatment
Receive Payment
Logout
Manage System

CHAPTER 4

SYSTEM DESIGN

4.1 Basic Modules

Basic modules are essential for any website because they provide the foundational elements needed to deliver a functional, user-friendly, and engaging experience.

1. User Registration and Login

Allow patients to create accounts: Users can register by providing essential personal information, such as name, email, and contact details, to create secure accounts. This feature fosters a personalized experience, allowing patients to manage their health appointments conveniently.

Login/Logout functionality: The system includes a straightforward login and logout process, enabling users to access their accounts easily while ensuring that their personal information remains protected. This functionality enhances security and user control.

Validate user input and ensure password security: The system implements thorough validation checks for user inputs, ensuring that fields are filled out correctly to prevent errors. Additionally, strong password requirements protect user accounts against unauthorized access.

2. Home Page

Intuitive design and navigation: The home page is designed with a clean and organized layout, making it easy for users to navigate through different sections of the site. Clear menu options and a logical flow contribute to a positive user experience.

Featured doctors or specialties: Prominently displaying selected doctors or specialties on the home page helps direct users to popular or recommended options, facilitating their decision-making process and promoting specific services effectively.

Call-to-action (CTA) buttons for booking appointments: Strategically placed CTA buttons encourage users to take immediate action, such as booking an appointment or learning more about services. This accessibility fosters user engagement and convenience.

Essential information (e.g., contact, about us): Including vital information about the clinic, such as contact details, location, and background information, helps build trust and transparency with patients, encouraging them to feel more comfortable using the service.

3. Book Appointment

View doctor profiles (bio, expertise, availability): Patients can browse comprehensive profiles of doctors that include biographical information, areas of expertise, and their current availability. This aids patients in making informed decisions about their healthcare providers.

Select appointment date : An interactive calendar feature allows users to select their preferred appointment dates easily.

Fill appointment form (patient info, appointment date etc): Patients complete an appointment form detailing their personal information. This information is crucial for doctors to prepare adequately for the consultation, leading to more effective care.

4. Payment

Secure payment gateway integration: The system integrates secure payment gateways to facilitate online transactions, ensuring that all financial data is encrypted and protected. This builds trust with users regarding the safety of their payment information.

Support multiple payment methods (credit/debit cards, net banking): By offering various payment options, including credit/debit cards and net banking, the system caters to diverse user preferences, making the payment process more convenient and accessible.

Display payment amount and breakdown: Clearly presenting the total payment amount, along with any applicable fees or discounts, enhances transparency and helps users understand their financial commitments, reducing potential confusion during the checkout process.

Receipt generation: The system automatically generates and sends digital receipts to users following successful transactions, providing them with proof of payment for their records. This feature is crucial for financial tracking and accountability.

5. Patient Feedback

Text review/comment section: patients have the option to leave detailed text reviews about their experiences. This feedback not only helps other patients in choosing doctors but also provides valuable insights for practitioners to improve their services.

4.2 Data Design

I am using the structured database architecture, namely MySQL, in which the data will be stored. To ensure appropriate distribution of data, the database consists of several tables as outlined below:

4.2.1 Schema Design

1 . user

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	role	enum('patient', 'admin', 'doctor')	utf8mb4_general_ci		No	None			Change Drop More
3	name	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
4	email	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
5	phonenumber	varchar(15)	utf8mb4_general_ci		Yes	NULL			Change Drop More
6	password	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
7	username	varchar(50)	utf8mb4_general_ci		Yes	NULL			Change Drop More
8	qualification	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
9	expertise	varchar(100)	utf8mb4_general_ci		Yes	NULL			Change Drop More
10	experience	int(11)			Yes	NULL			Change Drop More
11	available_days	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
12	available_time	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
13	doctor_image	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
14	role_id	int(11)			No	3			Change Drop More
15	status	enum('Approved', 'Pending')	utf8mb4_general_ci		Yes	Pending			Change Drop More

2. bookedappointments

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	ap_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	doctor_id	int(11)			No	None			Change Drop More
3	name	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	phone	varchar(20)	utf8mb4_general_ci		No	None			Change Drop More
5	dob	date			No	None			Change Drop More
6	email	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
7	ap_date	date			No	None			Change Drop More
8	status	enum('Pending', 'Approved', 'Completed')	utf8mb4_general_ci		No	None			Change Drop More
9	created_at	timestamp			No	current_timestamp()			Change Drop More
10	call_link	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

3. treatments

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	t_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	d_id	int(11)			No	None			Change Drop More
3	p_id	int(11)			No	None			Change Drop More
4	t_cost	decimal(10,2)			No	None			Change Drop More
5	t_report	text	utf8mb4_general_ci		No	None			Change Drop More
6	created_at	timestamp			No	current_timestamp()			Change Drop More

4. payments

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	payment_id	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
2	user_id	int(11)			Yes	NULL			Change Drop More
3	treatment_id	int(11)			Yes	NULL			Change Drop More
4	amount	double(5,2)			Yes	NULL			Change Drop More
5	payment_time	datetime			Yes	current_timestamp()			Change Drop More

5. subscription

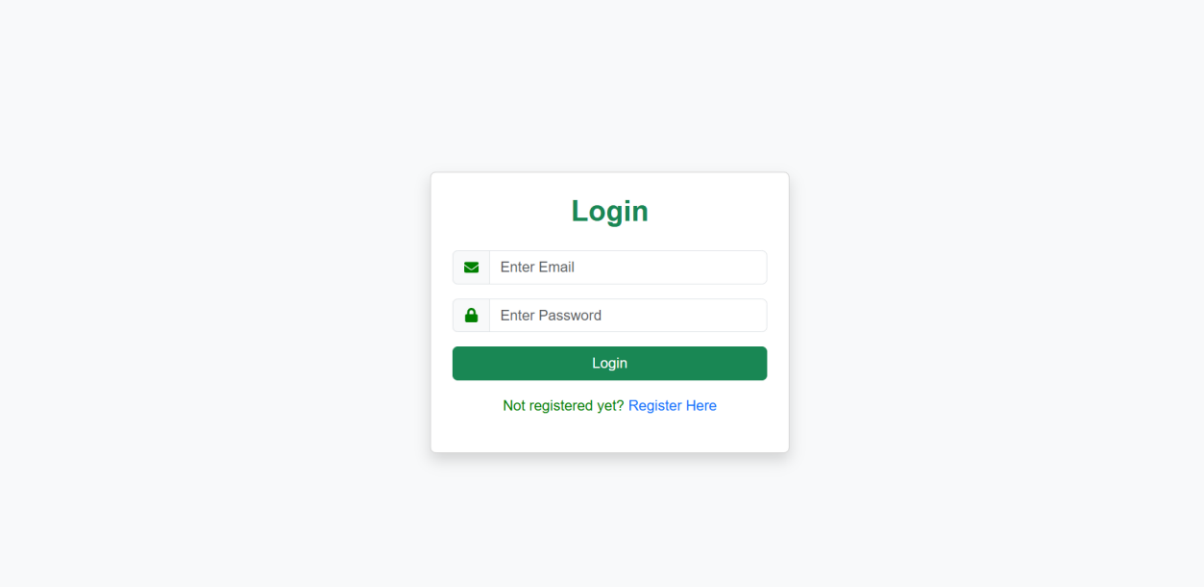
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	subscription_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	d_id	int(11)			No	None			Change Drop More
3	s_duration	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
4	s_cost	decimal(10,2)			No	None			Change Drop More
5	s_status	enum('Pending', 'Active', 'Expired')	utf8mb4_general_ci		Yes	Pending			Change Drop More
6	payment_method	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
7	payment_id	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
8	created_at	timestamp			No	current_timestamp()			Change Drop More

6. feedbacks

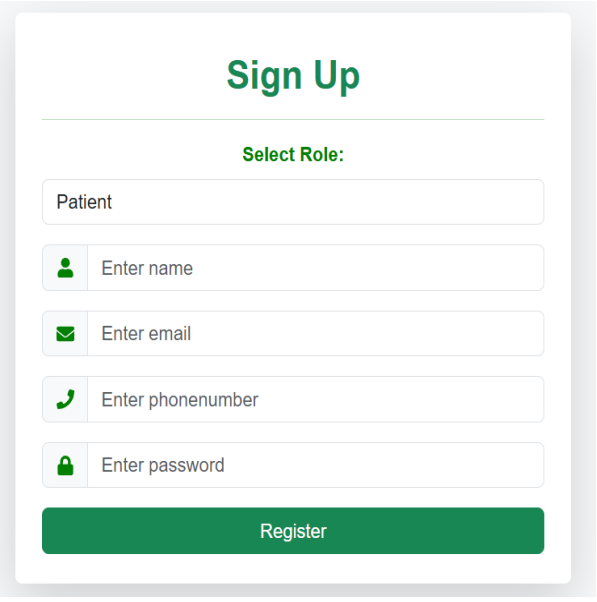
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	f_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	t_id	int(11)			No	None			Change Drop More
3	p_feedback	text	utf8mb4_general_ci		No	None			Change Drop More
4	f_date	timestamp			No	current_timestamp()			Change Drop More

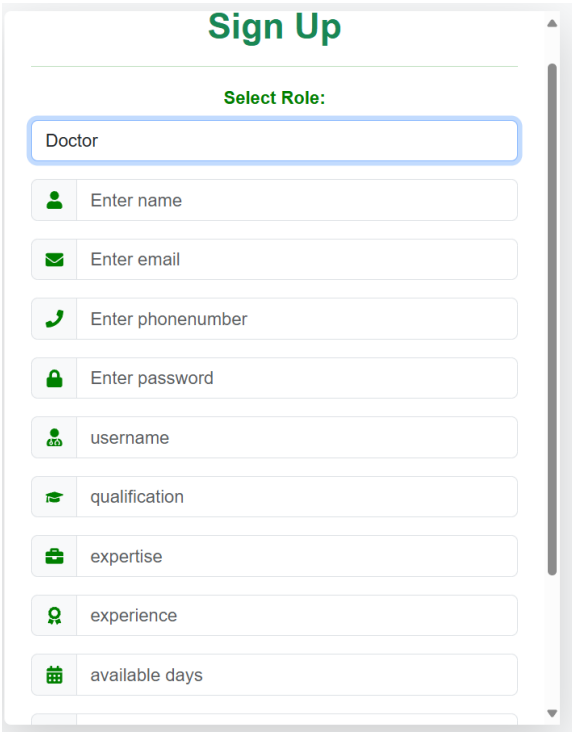
4.4 User Interface Design

Login Page



Registration Page

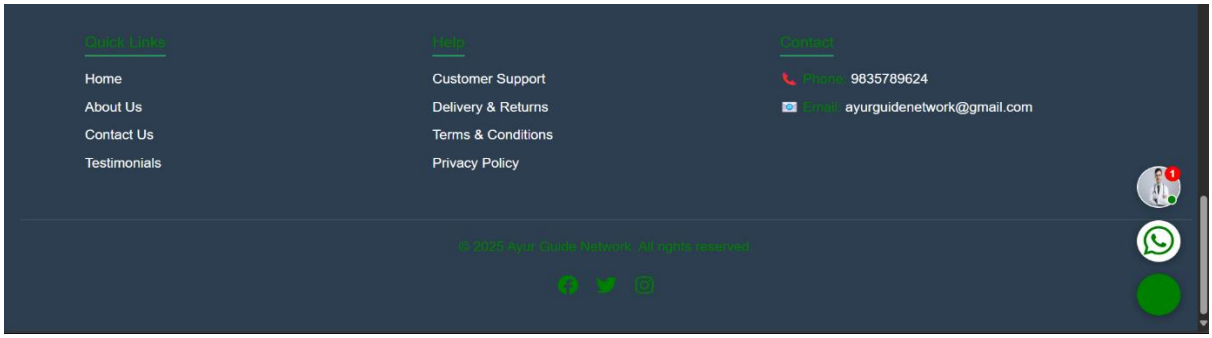




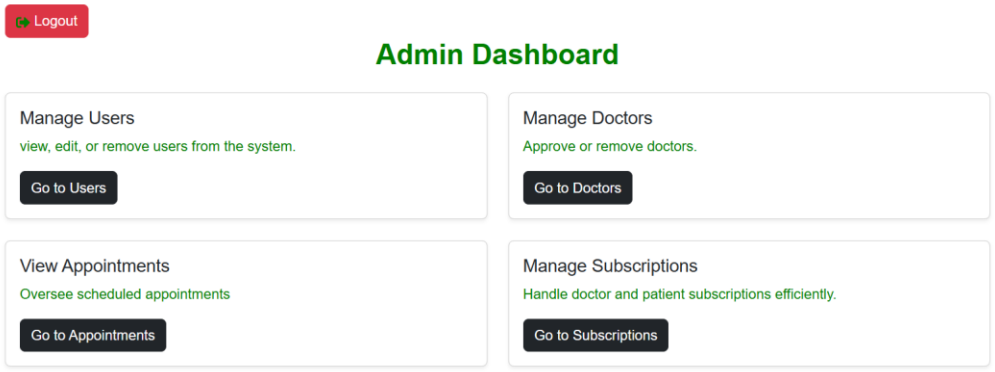
Home Page

About Us Page

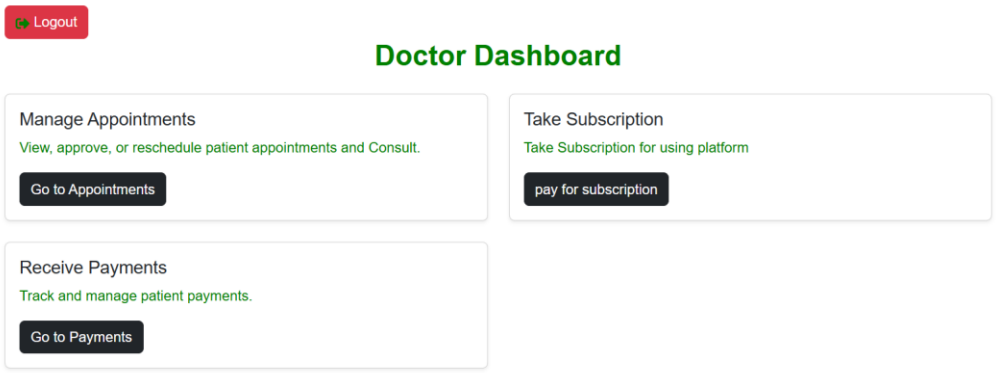
Footer Section



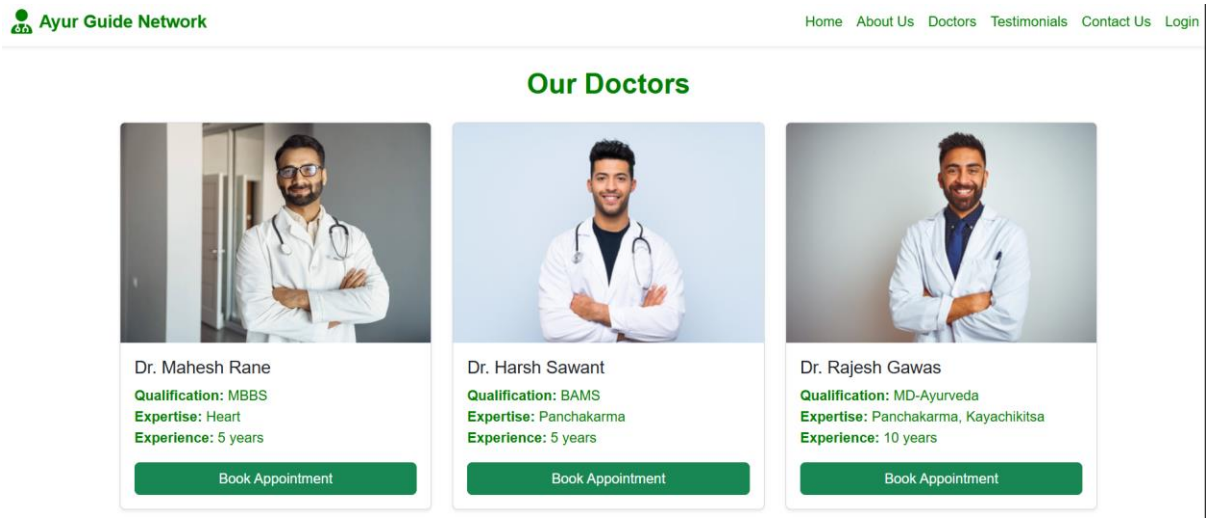
Admin Dashboard



Doctor Dashboard



Our Doctors



4.5 Test Cases Design

Tester Name : Tukaram Satyawan Manjarekar

Test Case ID	Test Steps	Input Data	Expected Results	Actual Results	Execution Status
1	User Registration	Enter name,email, Phone number, password	Registered Successfully	Registered Successfully	Completed
2	Verify email field restricting without @ symbol	Enter invalid email	Error message displayed	Error message displayed	Completed
3	Verify if all the fields have input	Submit without entering value in any of inputs	Error message displayed	Error message displayed	Completed
4	Verify password mismatch	Enter different password	Error message displayed	Error message displayed	Completed
5	User Login	Enter username and password	Login successfull	Login Successfull	Completed
6	Verify invalid login credentials	Enter incorrect username or password	Login Unsuccessful	Login Unsuccessful	Completed
7	Verify patient dashboard access	Login as Patient	Dashboard displayed with profile, appointments, feedback	Dashboard displayed	Completed
8	Verify doctor dashboard access	Login as Doctor	Dashboard displayed with appointements, schedule	Dashboard displayed	Completed
9	Verify admin dashboard access	Login as Admin	Dashboard Displayed with user, appointment	Dashboard displayed	Completed
10	Book Appointment	Enter valid patient details	Appointment booked successfully	Appointment booked successfully	Completed
11	Verify appointment booking restrictions	Try booking Past date	Error message displayed	Error message dislayed	Completed
12	Verify doctor selection	Select a doctor	Doctor assigned successfully	Doctor assigned successfully	Completed
13	Verify patient feedback submission	Enter feedback	Feedback submitted	Feedback submitted successfully	Completed

14	Verify treatment report generation	Doctor enters treatement details	Report generated and available for download	Report generated	Completed
15	Verify patient report download	Patient downloads treatment report	Report downloaded successfully	Report downloaded successfully	Completed
16	Verify payment processing	Enter payment details	Payment successful	Payment successful	Completed
17	Verify failed payment handling	Enter invalid payment details	Payment failed with error message	Payment failed	Completed

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Algorithms :

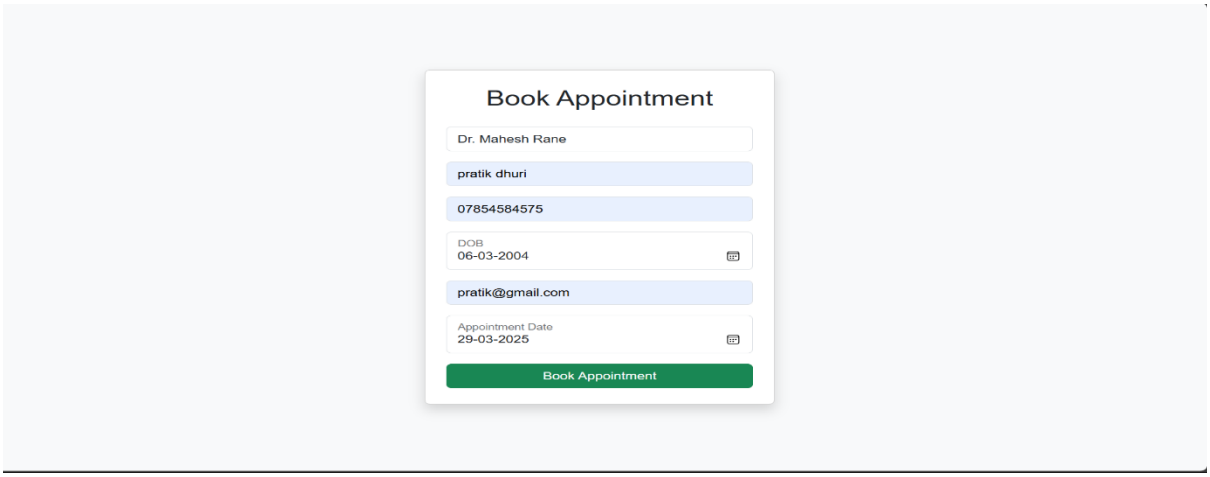
Following are some algorithms i have created :

1. A user books an appointment

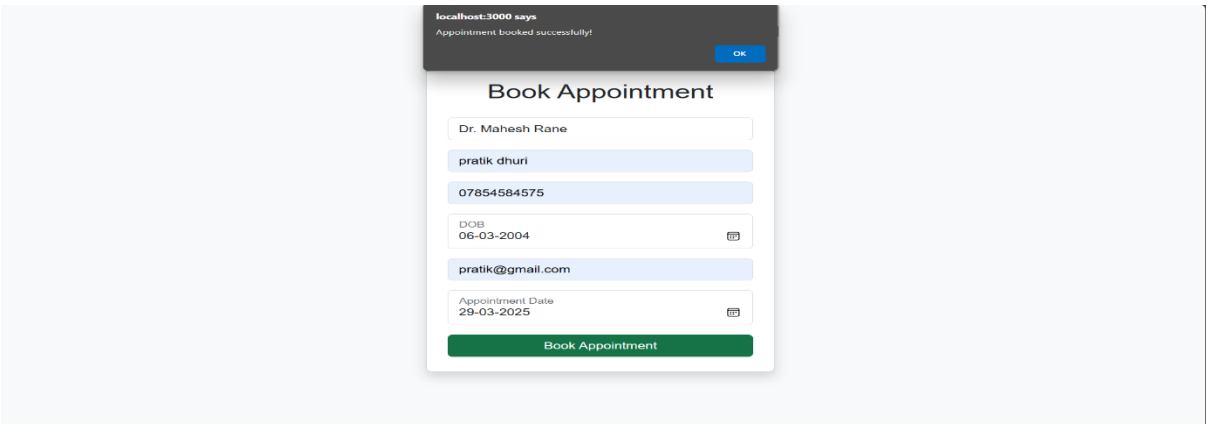
Code :

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: POST, OPTIONS");
header("Access-Control-Allow-Headers: Content-Type, Authorization");
if ($_SERVER["REQUEST_METHOD"] == "OPTIONS") {
  http_response_code(200);
  exit();
}
include './dbconnection.php';
$data = json_decode(file_get_contents("php://input"), true);
// Debug: Log received data
error_log("Received Data: " . print_r($data, true));
// Check if required fields are present
if (!isset($data['doctor_id'], $data['name'], $data['phone'], $data['dob'], $data['email'], $data['ap_date'])) {
  echo json_encode(["message" => "All fields are required"]);
  exit();
}
// Sanitize inputs
$doctor_id = $conn->real_escape_string($data['doctor_id']);
$name = $conn->real_escape_string($data['name']);
$phone = $conn->real_escape_string($data['phone']);
$dob = $conn->real_escape_string($data['dob']);
$email = $conn->real_escape_string($data['email']);
$ap_date = $conn->real_escape_string($data['ap_date']);
$status = "Pending";
```

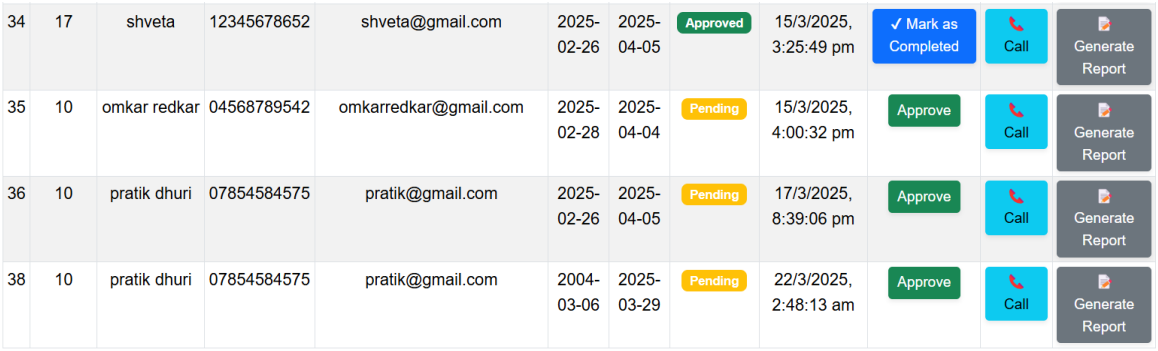
```
// Insert Query (without patient_id)
$sql = "INSERT INTO bookedappointments (doctor_id, name, phone, dob, email, ap_date, status, created_at)
        VALUES ('$doctor_id', '$name', '$phone', '$dob', '$email', '$ap_date', '$status', NOW())";
error_log("SQL Query: " . $sql);
if ($conn->query($sql) === TRUE) {
  echo json_encode(["message" => "Appointment booked successfully"]);
} else {
  error_log("Database Error: " . $conn->error);
  echo json_encode(["message" => "Database error: " . $conn->error]);
}
$conn->close();
?>
```



After successfully submitting the appointment form



Displayed the booked appointments on doctor’s dashboard



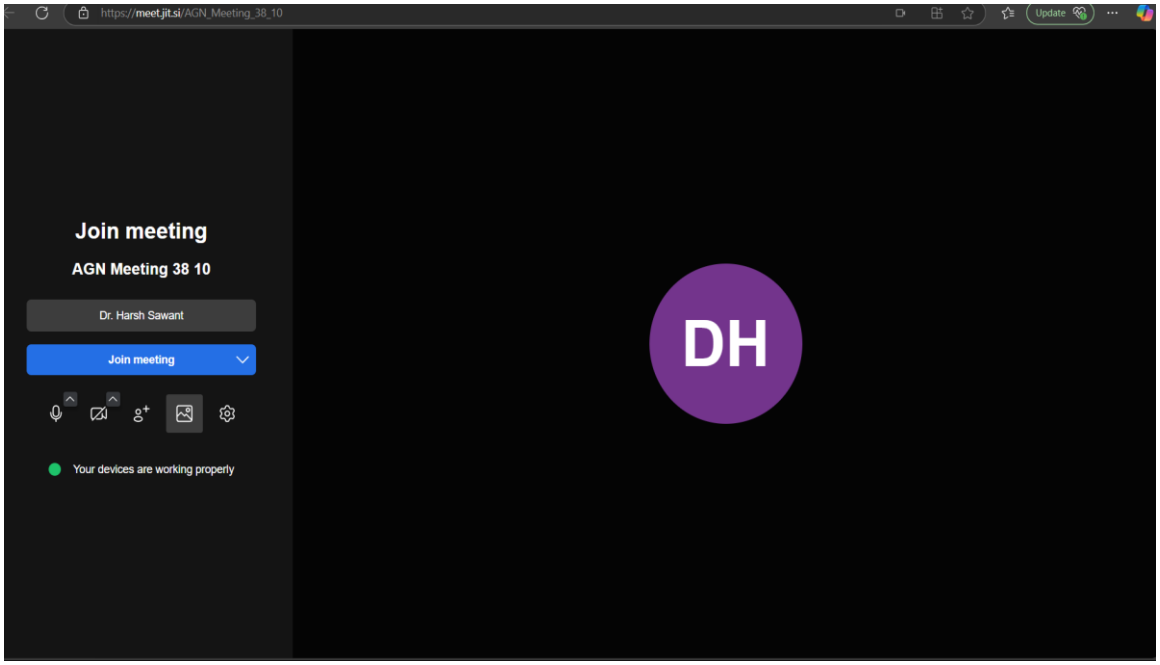
34	17	shveta	12345678652	shveta@gmail.com	2025-02-26	2025-04-05	Approved	15/3/2025, 3:25:49 pm	✓ Mark as Completed	Call	Generate Report
35	10	omkar redkar	04568789542	omkarredkar@gmail.com	2025-02-28	2025-04-04	Pending	15/3/2025, 4:00:32 pm	Approve	Call	Generate Report
36	10	pratik dhuri	07854584575	pratik@gmail.com	2025-02-26	2025-04-05	Pending	17/3/2025, 8:39:06 pm	Approve	Call	Generate Report
38	10	pratik dhuri	07854584575	pratik@gmail.com	2004-03-06	2025-03-29	Pending	22/3/2025, 2:48:13 am	Approve	Call	Generate Report

2. Doctor can call the patient through jitsi meet

Code :

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Allow-Headers: Content-Type");
header("Content-Type: application/json");
include './dbconnection.php';
$data = json_decode(file_get_contents("php://input"), true);
if (!isset($data['ap_id']) || !isset($data['doctor_id']) || !isset($data['call_link'])) {
  echo json_encode(["success" => false, "message" => "Missing required parameters"]);
  exit;
}
$ap_id = $data['ap_id'];
$doctor_id = $data['doctor_id'];
$call_link = $data['call_link'];
$sql = "UPDATE bookedappointments SET call_link = ? WHERE ap_id = ? AND doctor_id = ?";
$stmt = $conn->prepare($sql);
if (!$stmt) {
  echo json_encode(["success" => false, "message" => "SQL Prepare Failed: " . $conn->error]);
  exit;
}
```

```
$stmt->bind_param("sii", $call_link, $ap_id, $doctor_id);

if ($stmt->execute()) {

  echo json_encode(["success" => true, "message" => "Jitsi Meet link saved
successfully"]);

} else {

  echo json_encode(["success" => false, "message" => "Database update failed: " . $stmt-
>error]);

}

$stmt->close();

$conn->close();

?>
```



3. Doctor can generate report

Code :

```
<?php
header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: POST");
header("Access-Control-Allow-Headers: Content-Type");
header("Content-Type: application/json");
$conn = new mysqli("localhost", "root", "", "ayurguidenetwork");
if ($conn->connect_error) {
  die(json_encode(["success" => false, "message" => "Database connection failed"]));
}
$data = json_decode(file_get_contents("php://input"), true);
if (!isset($data["d_id"], $data["p_id"], $data["t_cost"], $data["t_report"])) {
  echo json_encode(["success" => false, "message" => "Missing required fields"]);
  exit();}
$d_id = intval($data["d_id"]);
$p_id = intval($data["p_id"]);
$t_cost = floatval($data["t_cost"]);
$t_report = $conn->real_escape_string($data["t_report"]);
$sql = "INSERT INTO treatments (d_id, p_id, t_cost, t_report) VALUES ('$d_id', '$p_id',
'$t_cost', '$t_report')";
if ($conn->query($sql) === TRUE) {
  echo json_encode(["success" => true, "message" => "Treatment report submitted
successfully"]);
} else {
  echo json_encode(["success" => false, "message" => "Error: " . $conn->error]);
$conn->close();
?>
```

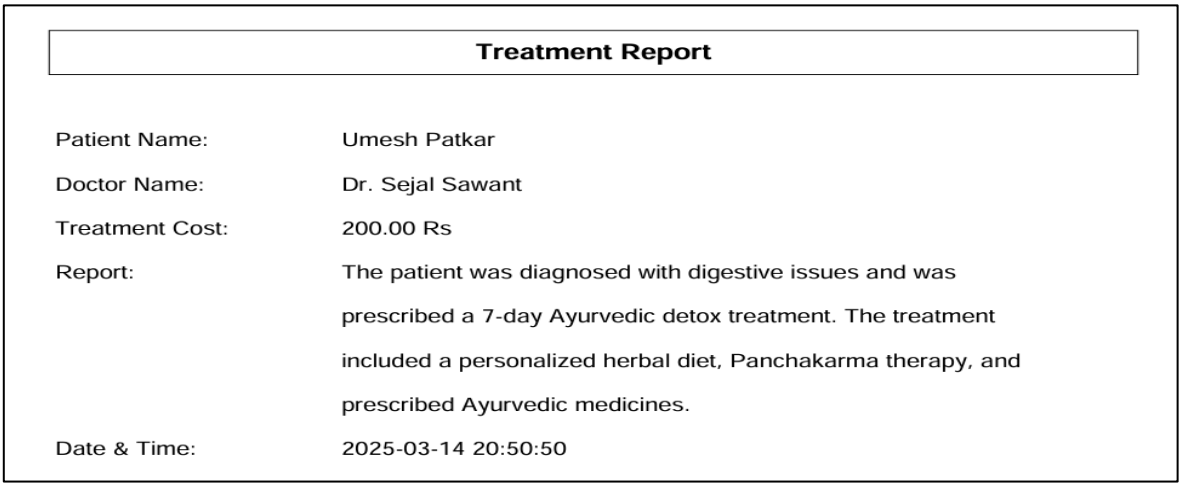
4. patient can download report

Code :

```
<?php
require(__DIR__ . '/fpdf186/fpdf.php');
include './dbconnection.php'; // Include database connection
// Check if p_id and t_id are provided in the request
if (!isset($_GET['p_id']) || empty($_GET['p_id']) || !isset($_GET['t_id']) || empty($_GET['t_id'])) {
  die("Error: Patient ID and Treatment ID are required.");
}
$p_id = $_GET['p_id'];
$t_id = $_GET['t_id'];
// Fetch treatment details including doctor and patient names
$query = "SELECT
        t.t_cost,
        t.t_report,
        t.created_at,
        p.name AS patient_name,
        d.name AS doctor_name
```

```
        FROM treatments t
        JOIN user p ON t.p_id = p.id  -- Get patient details
        JOIN user d ON t.d_id = d.id  -- Get doctor details
        WHERE t.p_id = ? AND t.t_id = ?
        LIMIT 1"; // Get specific treatment report
$stmt = $conn->prepare($query);
$stmt->bind_param("ii", $p_id, $t_id);
$stmt->execute();
$result = $stmt->get_result();
if ($result->num_rows > 0) {
  $data = $result->fetch_assoc();
  // Generate PDF
  class PDF extends FPDF
  {
    function Header()
    {
      $this->SetFont('Arial', 'B', 14);
      $this->Cell(190, 10, 'Treatment Report', 1, 1, 'C');
      $this->Ln(10);
    }
  }
  $pdf = new PDF();
  $pdf->AddPage();
  $pdf->SetFont('Arial', '', 12);
  // Patient Name
  $pdf->Cell(50, 10, 'Patient Name:', 0, 0);
  $pdf->Cell(100, 10, $data['patient_name'], 0, 1);
  // Doctor Name
  $pdf->Cell(50, 10, 'Doctor Name:', 0, 0);
  $pdf->Cell(100, 10, $data['doctor_name'], 0, 1);
```

```
  // Treatment Cost
  $pdf->Cell(50, 10, 'Treatment Cost:', 0, 0);
  $pdf->Cell(100, 10, $data['t_cost'] . ' Rs', 0, 1);
  // Treatment Report
  $pdf->Cell(50, 10, 'Report:', 0, 0);
  $pdf->MultiCell(130, 10, $data['t_report'], 0, 1);
  // Treatment Date & Time
  $pdf->Cell(50, 10, 'Date & Time:', 0, 0);
  $pdf->Cell(100, 10, $data['created_at'], 0, 1);
  // Output PDF
  $pdf->Output();
} else {
  echo "No report found for this patient and treatment ID.";
}
// Close DB Connection
$stmt->close();
$conn->close();
?>
```



Treatment Report

Patient Name:	Umesh Patkar
Doctor Name:	Dr. Sejal Sawant
Treatment Cost:	200.00 Rs
Report:	The patient was diagnosed with digestive issues and was prescribed a 7-day Ayurvedic detox treatment. The treatment included a personalized herbal diet, Panchakarma therapy, and prescribed Ayurvedic medicines.
Date & Time:	2025-03-14 20:50:50

5.2 Algorithm Efficient Aspects of Coding: Ayurvedic Doctor Appointment Sytem

1] Authentication & Role-Based Access

i. **Session-Based Authentication:** Implement session handling to ensure only logged-in users can access the system. Patients, doctors, and admins should have separate sessions.

ii. **Role-Based Access Control:**

- **Admin:** Can manage users, doctors, and appointments, and view feedback.
- **Doctor:** Can manage their profile, view appointments, provide treatment reports, and accept/reject video calls.
- **Patient:** Can book appointments, give feedback, and download treatment reports.

2] Session Management

i. Store user_id and role_id in sessions after login to track user authentication.

ii. Ensure sessions are cleared on logout to prevent unauthorized access.

3] Database Query Optimization

i. **Efficient Appointment Fetching:**

- Doctors see only their assigned appointments, while admins can see all bookings.

ii. **Optimized Search:**

- Patients can search for doctors based on specialization using indexed LIKE queries.

iii. **Treatment & Booking Validation:**

- Ensure that appointment dates and times do not overlap using optimized SQL queries.

4] Caching and Performance Enhancements

Optimize Database Queries:

- Reduce redundant queries when fetching doctor and appointment data.

5.3.1 Unit Testing

1. **User Authentication & Authorization**

    - Test login, registration, session management, and role-based access control.
    - Verify that unauthorized users cannot access protected routes.

2. **Doctor & Patient Management**

    - Test adding, updating, and deleting doctor profiles.
    - Ensure patients can update their details correctly.

3. **Appointment Booking System**

    - Verify correct appointment date selection and validation.
    - Ensure a doctor cannot have overlapping bookings.

4. **Treatment & Feedback System**

    - Validate that doctors can add treatment reports and patients can download them.
    - Test patient feedback submission and retrieval.

5. **Admin Panel**

    - Test adding/editing/removing users (doctors, patients).
    - Ensure doctors can be activated/deactivated.

5.3.2 Integration Testing

1. **Appointment Booking & Availability Check**

    - Ensure that once an appointment is booked, the doctor is unavailable for the same slot.

2. **Doctor & Patient Interaction**

    - Verify that doctors can access only their assigned appointments.
    - Ensure patients can see only their treatment reports.

3. **Payment & Transaction Processing**

    - Ensure UPI payment works correctly and payments are logged in the database.

4. **Notification & Alerts**

    - Validate that patients receive appointment confirmations and treatment updates.
    - Ensure doctors get notifications when a patient books an appointment.

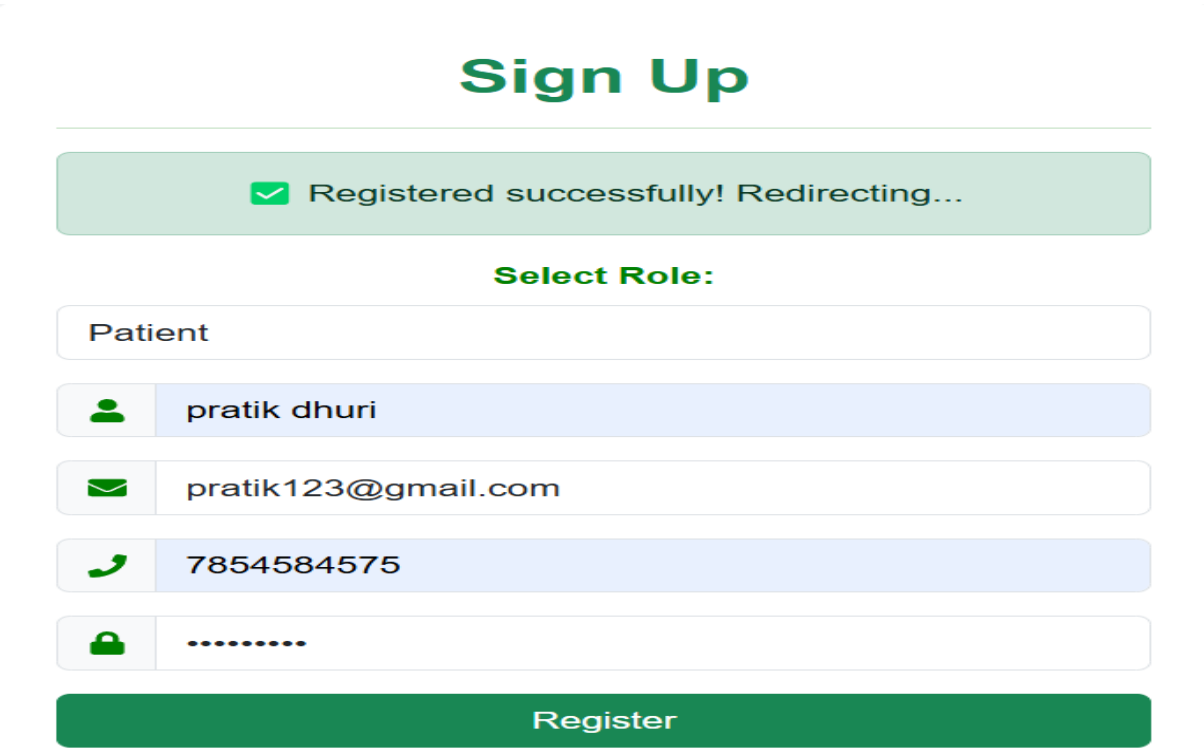
Chapter 6

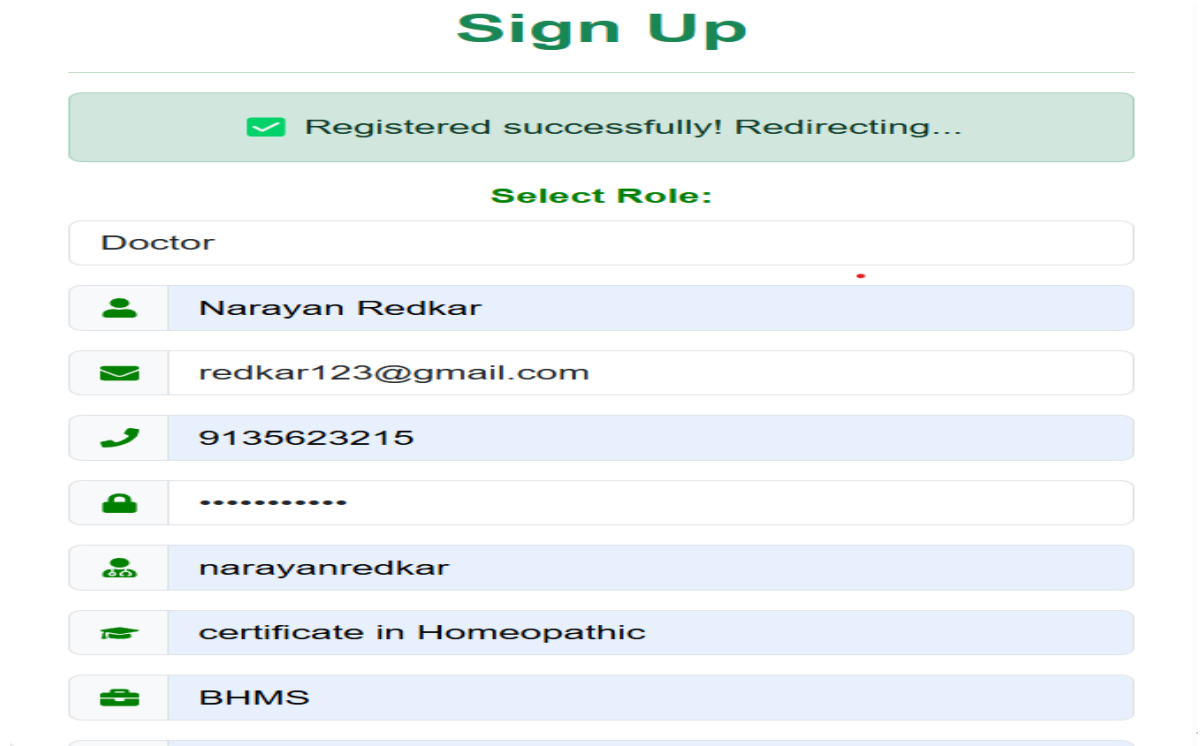
RESULTS AND DISCUSSION

6.1System UI:

User Registration:

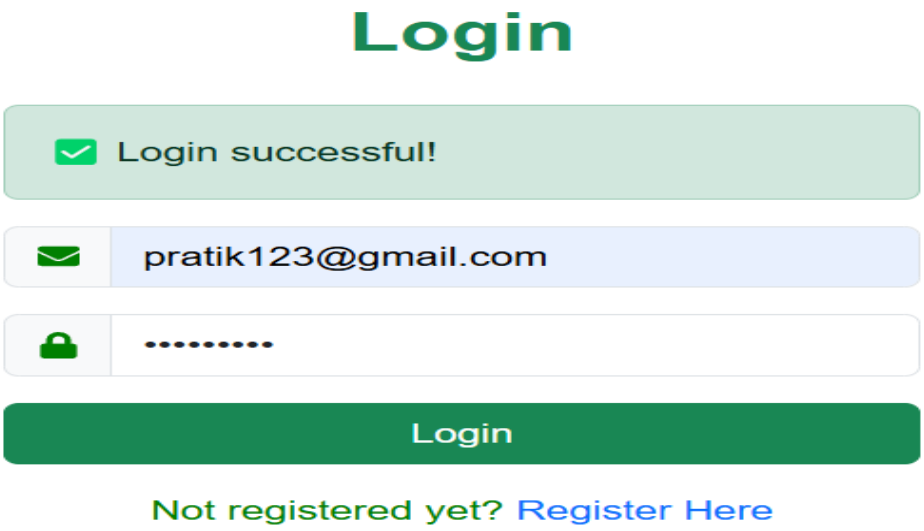
Used for inserting user data into the system and displaying a confirmation message upon successful registration.





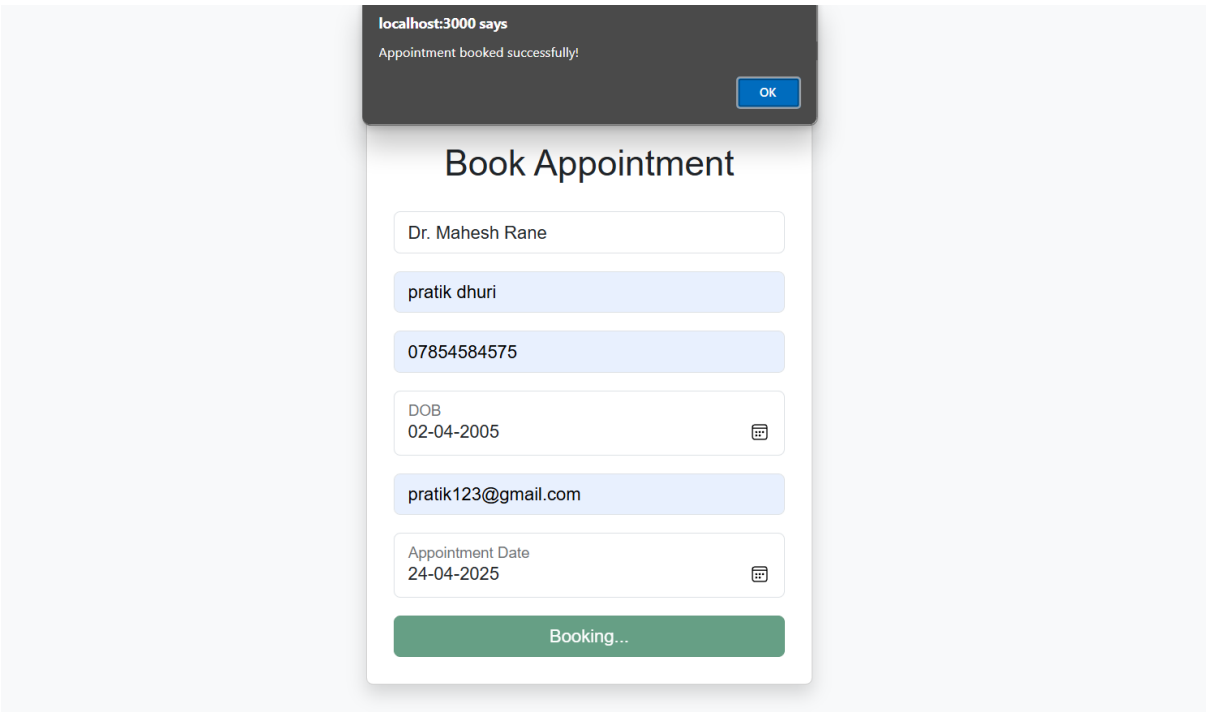
Login :

Login Form requiring Email and Password for secure user authentication and system access.



Appointment Booking :

Here user can book appointment and get alert message after successfully booked appointment.



Report Generation

Here doctor can generate the report according to treatment

Download Treatment Report

Here patient can download treatement report

Treatment Report

Patient Name:	Umesh Patkar
Doctor Name:	Dr. Sejal Sawant
Treatment Cost:	200.00 Rs
Report:	The patient was diagnosed with digestive issues and was prescribed a 7-day Ayurvedic detox treatment. The treatment included a personalized herbal diet, Panchakarma therapy, and prescribed Ayurvedic medicines.
Date & Time:	2025-03-14 20:50:50

Chapter 7

CONCLUSION

6.1 Conclusion

The **Ayurvedic Doctor Appointment System** is a user-friendly and efficient platform designed to streamline the process of booking consultations with Ayurvedic doctors. It provides a structured system where patients can book appointments, doctors can manage their schedules, and admins can oversee user activities.

The platform ensures a **secure and seamless experience** with:

- **Authentication & Role-Based Access** for patients, doctors, and admins.
- **Optimized Database Performance** for fast appointment scheduling and treatment history retrieval.
- **Appointment Management with Real-Time Status Updates**, ensuring smooth doctor-patient interactions.
- **UPI Payment Integration** for secure transactions between patients and doctors.
- **Treatment Reports & Feedback System** for enhanced patient care and continuous service improvement.

6.2 Future Scope of the Project

1. **Mobile Application Development**
    - Develop a dedicated mobile app for **Android & iOS** to allow patients to book appointments on the go and receive real-time updates.
2. **AI-Based Doctor Recommendation System**
    - Implement **AI-driven recommendations** to suggest the best doctors based on patient symptoms, preferences, and past consultations.
    - Implement **smart contracts** for automated and tamper-proof transactions.
3. **Advanced Search & Filtering**
    - Enhance the search functionality with filters based on **doctor specialization, experience, available slots, and patient reviews**.
4. **Enhanced Review & Rating System**
    - Improve the patient feedback system to ensure **authentic ratings and reviews** for doctors, increasing trust and credibility.
5. **Integration with Ayurvedic Pharmacies**
    - Partner with **Ayurvedic medicine suppliers** to allow patients to order prescribed treatments directly through the platform.
6. **Multiple Payment Gateways**
    - Expand payment options by integrating **Razorpay, PayPal, Stripe, Google Pay,** and UPI, ensuring a smooth and secure transaction process.

CHAPTER 8

REFERENCES

7.1 Referred Websites

1. https://getbootstrap.com/
2. https://www.google.com/
3. https://chatgpt.com/
4. https://www.youtube.com/