Chapter 1: Introduction

1.1 Company Profile

Registered Office (India)

4th floor, Dangat Patil Empire,Near Aksharwel Book Store.

Landmark: Navale Bridge, Vadgaon BK Pune-411 041, MH, India

Phone: 9561932445 9890132085

Email:hr@webmindsitsolution.com

webmindsitsolution@gmail.com

Website: https://webmindsitsolution.com

Our Mission:

To lead in various avenues of software development by implementing novel ideas for project development thus delivering customized cost effective solutions.

Our Vision:

To earn global admiration as a software & web development company, by building and maintaining long lasting relationship with people and technology.

Our Values:

To provide quality solutions for each problems to achieve the goal of organization with innovation and empowers IT individuals with competitive advantage.

Services:

- Software Development
- Web Development
- UX/UI Design
- AngularJS
- Java Application Development
- Salesforce Customization & Integration Services
- Mobile App Development
- Software Testing
- E- Commerce Development
- Big Data & Analytics
- Cloud Services

1.2Abstract

The proposed project is online doctor appointment system that provides patients or any user an easy way of booking a doctor's appointment online. This is an application project in web that overcomes the issue of managing and booking appointments according to user's choice or demands. The task sometimes become very tedious for the compounder or doctors himself in manually allotting appointments for the users as per their availability.

Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. This system is very useful to save the patient's timing and also easy to analyses the doctor availability. The patient can easily interact with doctor for their problems and get solutions from the doctor. Another advantage is that the doctor can easily know the duty time and duration.

1.3Existing System and Need For System

Existing System

Under manual Doctor Appointment System, you have to first wait in line to take appointment for the doctors and wait for your time to have meet with them and discuss on your health problems.

As you have to provide your information and other reports many times at different places such as the medicine store which is again a burden of carrying documents. You have to be present physically at the doctor's cabin.

Patients have to visit on another day of after some hours to take their health reports which involves extra care to the person with patients anytime.

Under manual system, perfect time and date for appointment cannot be got because of uncertainty of doctor's routine.

Limitations of existing system:

- Time consuming.
- Lack of efficiency.
- Needs to be physically present at the clinic.

Need for System

In “Online Doctor Appointment System” I have tried to consider all of the above problems with the existing system. My objective was to provide a platform where all doctors and patients can be brought together.

In this application, all doctors either new or working for a long time can sign up themselves through admin. This way people can get to know and choose to visit any doctor of their choice.

By using this application user will be able to know and access the professional profile of each registered doctor from all specialties. Doctor’s profile includes information regarding their professional experience, practicing license, educational background, clinic timings, and working days.

Users would be able to book appointment sitting at their homes. The user can choose a doctor close to them or have more professional experience. New practitioners can easily join the platform and start taking appointments without the need of any expensive advertisement.

1.4 Scope of System

With 462 million active internet users and 430 million active mobile internet users in India, the scope for e-health services and solutions can prove to be a game changer in the patient care space. And in the current urban milieu where internet access is far higher than the national average; the number of potential consumers especially from the corporate sector, provides an untapped opportunity for healthcare providers to pilot, facilitate, regulate, and expand the span of preventive medicine, primary health care, and wellness programs beyond conventional boundaries**.**

Real-time platforms like telemedicine, tele-consultation, video-conferencing, remote patient Monitoring, Medical Record Systems (MCRs) are facilitating better healthcare delivery systems. Trends such as bring-your-own-device (BYOD) and bring-your-own-application (BYOA) are also being integrated to enable better patient: doctor digital engagements. Other emerging interventions like store-and-forward technologies, remotely-hosted deployments, predictive analytics, Health Information Exchanges (HIEs), Early Warning Scores (EWS), Fast Healthcare Interoperability Resources (FHIR) can further provide optimal health-care connected solutions. Whether it is for individuals or the corporates, the presence of online doctor consultation platforms are benefitting employees.

Increasingly, consumers are being empowered to use different hand-held devices like phone, tab and PC to securely access their medical records anywhere anytime, find preventive information,

compare services offered by different players, purchase medicines, book appointments, use symptom checkers and get health checks, store medical history in health vaults, and efficiently and seamlessly.

This system is implemented for all the individuals who want to get treated by the city practitioners. The users can participate only if they have created an account through the registration form. Once they get registered themselves further, they can book appointments according to their choice.

1.5 Module Description

- **Admin**
  - Can add doctors.
  - Can view doctors.
  - Can view patients.
  - Can view doctor’s leaves.
  - Can view patient’s queries.
  - Reports
- **Doctor**
  - Can view approved patients appointments.
  - Can take leaves.
  - Can view own leaves.
  - Can add receptionist.
  - Can manage profile.
  - Can change password.
- **Receptionist**
  - Can approve appointment request.
  - Can view approved appointments.
  - Can manage payment.
  - Can manage profile.
  - Can change password.
- **Patient**
  - Can send appointment request.
  - Can make payment.
  - Can check appointment status.

- Can manage profile.
- Can change password.

- **Payment**
  - Patient can make payment.
  - Receptionist can manage payment.

1.6 Operating Environment – Hardware & Software

Software Requirements

Type	Client Side Configuration	Serve Side Configuration
Operating System	Intel core i3 7th generation 64-bit, windows 10	Windows XP/8/10, Pentium IV 2.4 GHz.
Web Server	Apache tomcat	Apache tomcat 10
Front-End	HTML,CSS,Bootstrap,JS ,J2EE (JSP, Servlet)	HTML,CSS,Bootstrap,JS, J2EE (JSP, Servlet)
Backend/Database	MYSQL	MYSQL 5.5

Hardware Requirements

Type	Client Side Configuration	Server Side Configuration
Processor	7th Gen Intel(R) Core(TM) i3- 1135G7 @ 2.40GHz, 2419 Mhz, 4 Core(s), Logical Processor(s)	Windows XP/8/10, Pentium IV 2.4 GHz.
RAM	8 GB	32 GB
Hard Disk	1 TB	2.5 TB

1.7 Brief Description of Technology Used

1.7.1 Operating System Used (Windows)

Windows 11 is the latest major release of Microsoft's Windows NT operating system, released in October 2021. It is a free upgrade to its predecessor, Windows 10 (2015), available for any Windows 10 devices that meet the new Windows 11 system requirements.

Windows 11 features major changes to the Windows shell influenced by the canceled Windows 10X, including a redesigned Start menu, the replacement of its "live tiles" with a separate "Widgets" panel on the taskbar, the ability to create tiled sets of windows that can be minimized and restored from the taskbar as a group, and new gaming technologies inherited from Xbox Series X and Series S such as Auto HDR and Direct Storage on compatible hardware. Internet Explorer (IE) has been replaced by the Chromium-based Microsoft Edge as the default web browser like its predecessor, Windows 10, and Microsoft Teams is integrated into the Windows shell. Microsoft also announced plans to allow more flexibility in software that can be distributed via Microsoft Store, and to support Android apps on Windows 11 (including a partnership with Amazon to make its app store available for the function).

HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to

improve the page load speed between the pages that share the file and its formatting.

Bootstrap

Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

JavaScript

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming

styles. JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

Ajax

Asynchronous JavaScript and XML (Ajax) refer to a group of technologies that are used to develop web applications. By combining these technologies, web pages appear more responsive since small packets of data are exchanged with the server and web pages are not reloaded each time that a user makes an input change. Ajax enables a web application user to interact with a web page without the interruption of constant web page reloading. Website interaction happens quickly with only portions of the page reloading and refreshing.

JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the

underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client–server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

JSP

JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet

such as expression language, JSTL, etc. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP that makes JSP development easy. JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic. If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application. In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

Eclipse

Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may

also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, C#, Clojure, COBOL, D, Erlang, Fortran, Groovy, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP,JSP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, and Scheme. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

Server

Tomcat is used for web applications written in Java that don't require full Java EE specifications, but still need a reliable tool. The reason Tomcat is not really a full application server is because it acts only as a Web server and Servlet container. It doesn’t provide the full feature set from the Java EE, but that isn't necessarily a disadvantage. Many applications require just the features that Tomcat provides, so it doesn't make sense to go with the heavier tools. You can use Apache Tomcat for production applications that process thousands requests if the features it provides is enough. In any case, Tomcat is a production ready tool.

1.7.2 Database

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more

often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, php BB, and Word Press. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

Chapter 2: Proposed System

2.1 Feasibility Study

- **Technical Feasibility**

  It is determined by evaluating the hardware and software cost. Since all these facilities are available, so implementation of the system is not a big factor.

  The technical issue usually raised during the feasibility stage of the investigation includes the following:

  Does the necessary technology exist to do what is suggested?

  Does the proposed equipment have the technical capacity to hold the data required to use the new system?

  Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

  Can the system be upgraded if developed? Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible.

- **Economic Feasibility**

  Economic feasibility is most frequently used for evaluating the effectiveness of a proposed system. The test of this analysis points out whether the proposed system is economically feasible or not.

  Development, Operating and hardware cost are very low in our system so given system is economically feasible.

- **Operational Feasibility**

  It includes the training of the users on the proposed system. Care has been taken to provide the user as much facility as possible. The system is user friendly so that users can be trained very easily. The user does not need any special training to run this application.

2.1 Objectives of Proposed System

The main objective of the system is to provide quality medical care to the patients by bringing all medical practitioners of the city to a single platform so that everybody can easily access them and make appointments.

The second objective is to replace the current manual file keeping system with an online medical patient database system.

We implement this system for better user experience. This system is very easy to access. Also, for establish real time communication, using modern and updated technology. So, user can see the update without reload or refresh.

This system will compatible with user device such as pc, laptop, tab & smart phone. So, user can easily access the system anytime anywhere. This system is very simple & user friendly so, any user can use this system easily.

2.3 Users of System

Patients

The patient can select any particular doctor and view his profile by clicking on the available doctors or by selecting the doctor's option from the menu screen. The list of the available doctors is displayed as shown below. Patient can contact the doctor by making a call by clicking on the doctor's phone number or patient can also sends an email.

By clicking in the book appointment button, a calendar and different available time slots are displayed on the screen. The patient has to send a request for appointment by selecting a day or time. The central database gets updated accordingly. The user will get notification message of "successfully added" if the appointment is successfully registered in the database.

Admin

Admin logs in and then he can add a doctor. Admin can view the doctor details, doctor leaves, patient details and the appointment details as well their queries.

Doctors

Doctor logs in and then he can view the patient details and the appointment details, can take and view leaves as well can add the receptionist of their hospitals.

Receptionist

Receptionist logs in and then he can view the patient details and the appointment details, can approve the appointments as well he can manage the payments.

Chapter 3: Analysis and Design

3.1 System Requirements (Functional and Non-Functional)

Functional Requirements

Such requirements describe system behavior under specific conditions and include the product features and functions which web app developers must add to the solution. Such requirements should be precise both for the development team. Functional requirements are needs related to the technical functionality of the system. Functional requirements state how the users will interact with the application, so the application must be able to comply and be testable.

The list of examples of functional requirements includes:

- Content to be created and published in the system (i.e., documents, blogs, etc.)
- Administrative functions
- Authentication
- Authorization levels
- External Interfaces
- Reporting Requirements
- Historical Data

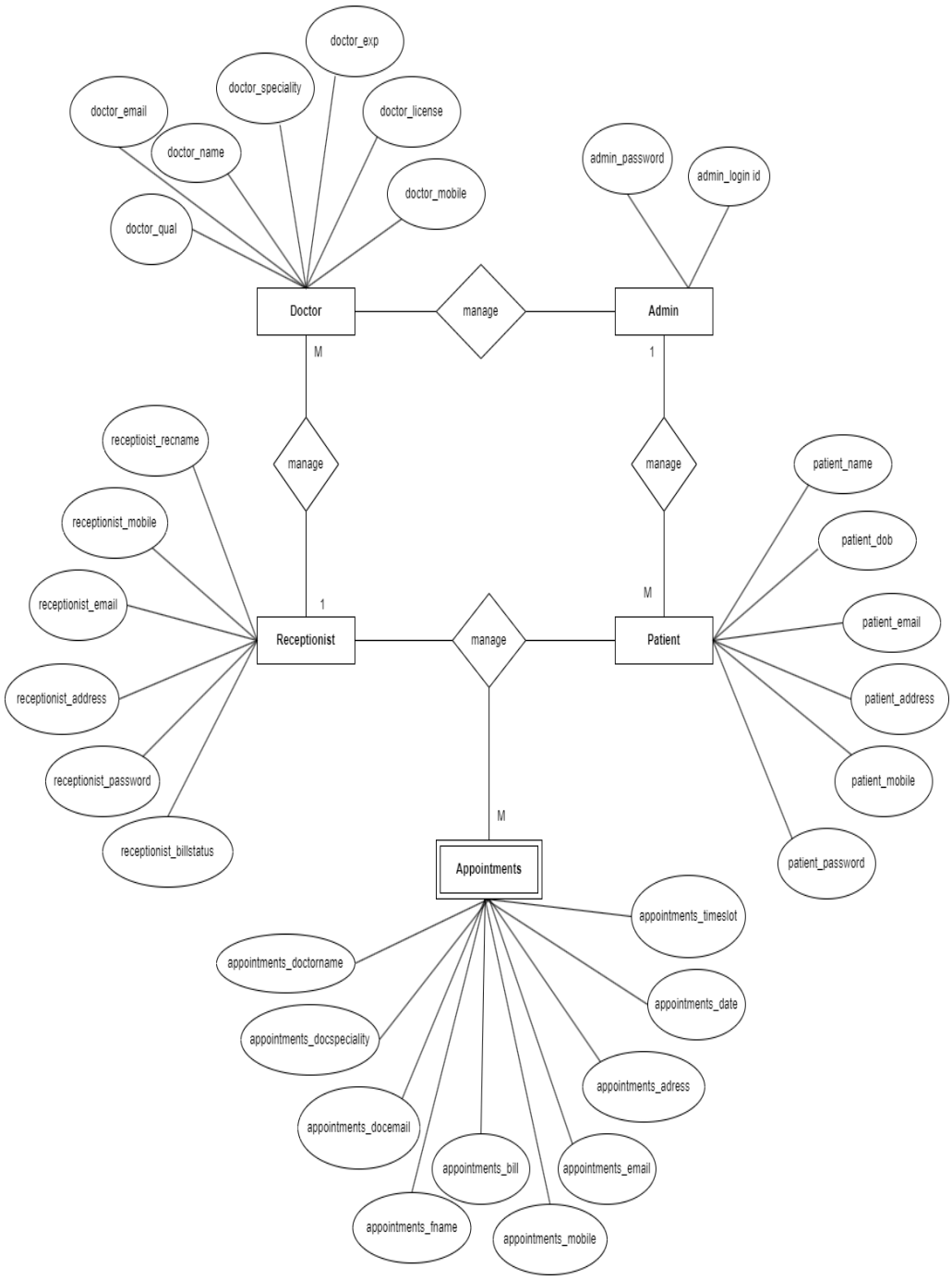
Non-Functional Requirements

Non-functional requirements specify criteria that can be used to judge the operation of a system in particular. These are conditions rather than specific behaviors. While functional requirements define what a system is supposed to do, non-functional requirements define how a system is supposed to be. Non-functional requirements are often called "quality attributes" of a system.

Non-functional requirements are quality attributes that describe ways your product should behave. The list of basic non-functional requirements includes:

- Performance - how much time each page should take to load
- Scalability - will the system be able to handle large volume of users that keeps increasing?
- Capacity - how much storage will be needed?
- Availability - availability and downtime of the application.
- Security - this includes security of the content and encryption.

3.2 Entity Relationship Diagram (ERD)



3.3 Table Structure

Table 1: Doctors

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
doctorname	Varchar(50)	Not Null	Name of the doctor
mobile	Varchar(10)	Not Null	Mobile number of doctor
emergencycontact	Varchar(10)	Not Null	Emergency contact number of doctor
email	Varchar(30)	Not Null	Email id of doctor
dob	Varchar(10)	Not Null	Date of birth of doctor
maritalstatus	Varchar(10)	Not Null	Marital status of doctor
specialty	Varchar(20)	Not Null	Specialty of doctor
license	Varchar(10)	Not Null	License number of doctor
qualification	Varchar(20)	Not Null	Qualification of doctor
experience	Varchar(50)	Not Null	Experience of doctor

state	Varchar(20)	Not Null	State of doctor
city	Varchar(20)	Not Null	City of doctor
postal	Varchar(10)	Not Null	Postal code of doctor
address	Varchar(100)	Not Null	Address of doctor
password	Varchar(30)	Not Null	Account password of doctor
entry_date	Varchar(10)	Not Null	Doctor registration date

Table 2: Leaves

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
doctorname	Varchar(50)	Not Null	Name of the doctor
email	Varchar(30)	Not Null	Email id of doctor
from_date	Varchar(10)	Not Null	Leave form date
to_date	Varchar(10)	Not Null	Leave to date
reason	Varchar(50)	Not Null	Reason of leave
entry_date	Varchar(10)	Not Null	Doctor leave data added date

Table 3: Patient

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
fname	Varchar(50)	Not Null	Name of the patient
dob	Varchar(10)	Not Null	Date of birth of patient
email	Varchar(30)	Not Null	Email id of patient
mobile	Varchar(10)	Not Null	Mobile number of patient

gender	Varchar(10)	Not Null	Gender of patient
state	Varchar(30)	Not Null	State of patient
city	Varchar(30)	Not Null	City of patient
address	Varchar(50)	Not Null	Address of patient
password	Varchar(30)	Not Null	Account password
entry_date	Varchar(10)	Not Null	Patient registration date

Table 4: Queries

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
name	Varchar(50)	Not Null	Name of the user
email	Varchar(30)	Not Null	Email id of user
topic	Varchar(20)	Not Null	Queries topic
mobile	Varchar(10)	Not Null	Mobile number of user
message	Varchar(100)	Not Null	Queries details

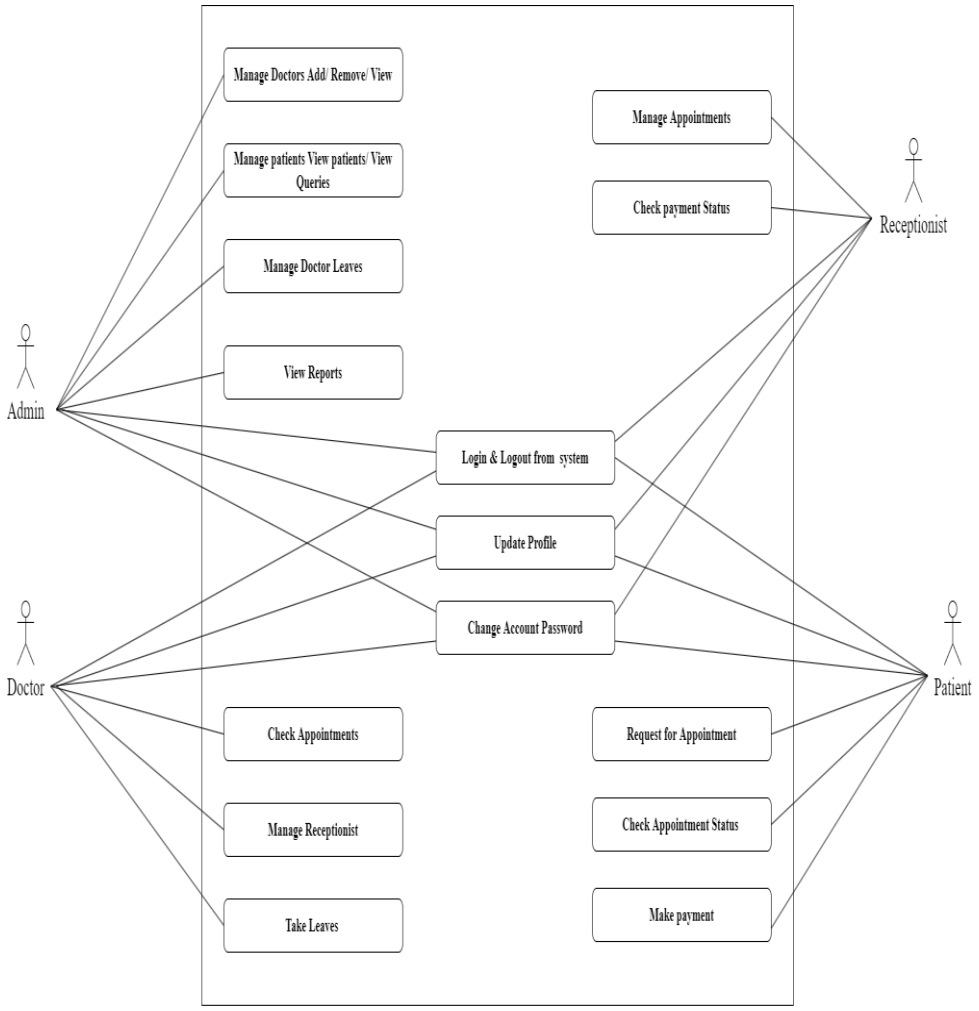
Table 5: Receptionist

FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
recname	Varchar(50)	Not Null	Name of the receptionist
mobile	Varchar(10)	Not Null	Mobile number of receptionist
docemail	Varchar(30)	Not Null	Email of doctor
address	Varchar(50)	Not Null	Address of receptionist
password	Varchar(30)	Not Null	Account password
entry_date	Varchar(10)	Not Null	Receptionist registration date

Table 6: Requests (Appointments)

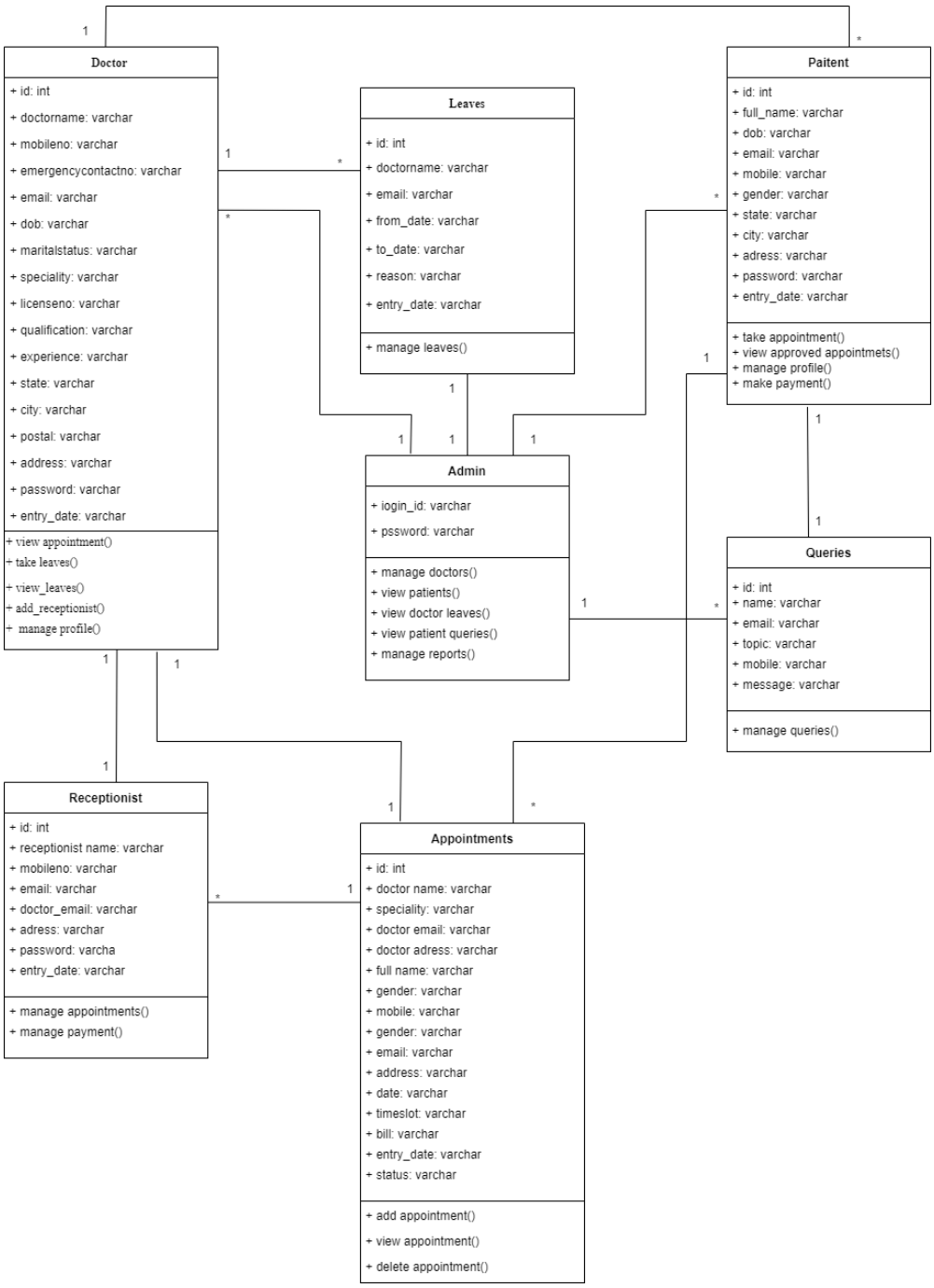
FIELD NAME	DATA TYPE	CONSTRAINTS	DESCRIPTION
id	int	Primary Key	Id of row
doctorname	Varchar(50)	Not Null	Name of the doctor
specialty	Varchar(20)	Not Null	Specialty of doctor
docemail	Varchar(30)	Not Null	Email id of doctor
docaddress	Varchar(50)	Not Null	Address of doctor
fname	Varchar(30)	Not Null	Name of the patient
gender	Varchar(10)	Not Null	Gender of the patient
mobile	Varchar(10)	Not Null	Mobile number of patient
email	Varchar(30)	Not Null	Email id of patient
address	Varchar(50)	Not Null	Address of patient
date	Varchar(10)	Not Null	Appointment date
timeslot	Varchar(20)	Not Null	Appointment timeslot
bill	Varchar(10)	Not Null	Appointment charges
entry_date	Varchar(10)	Not Null	Appointment registered date
status	Varchar(20)	Not Null	Appointment status

3.4 Use Case Diagram

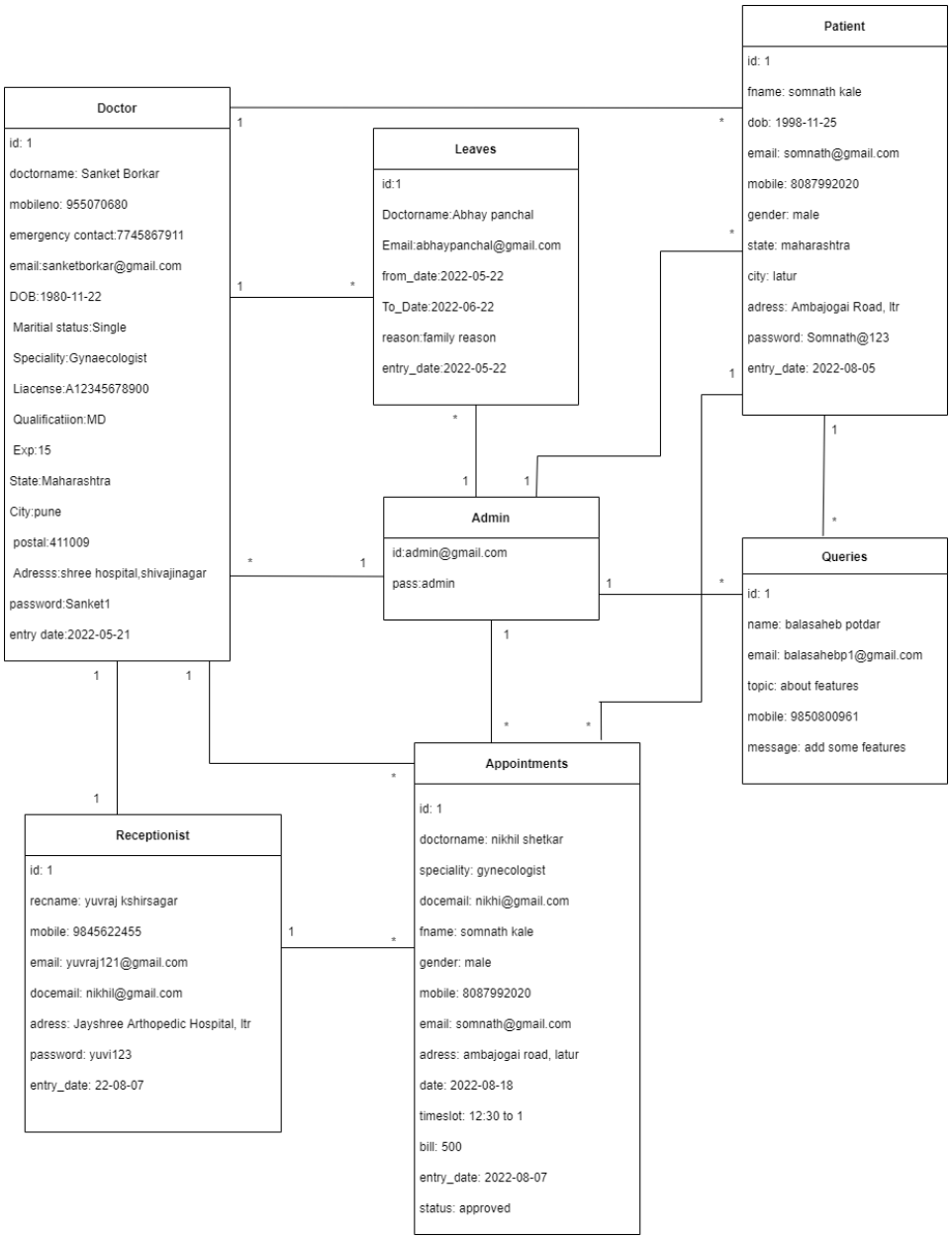


3.5 Class Diagram and Object Diagram

- **Class Diagram**



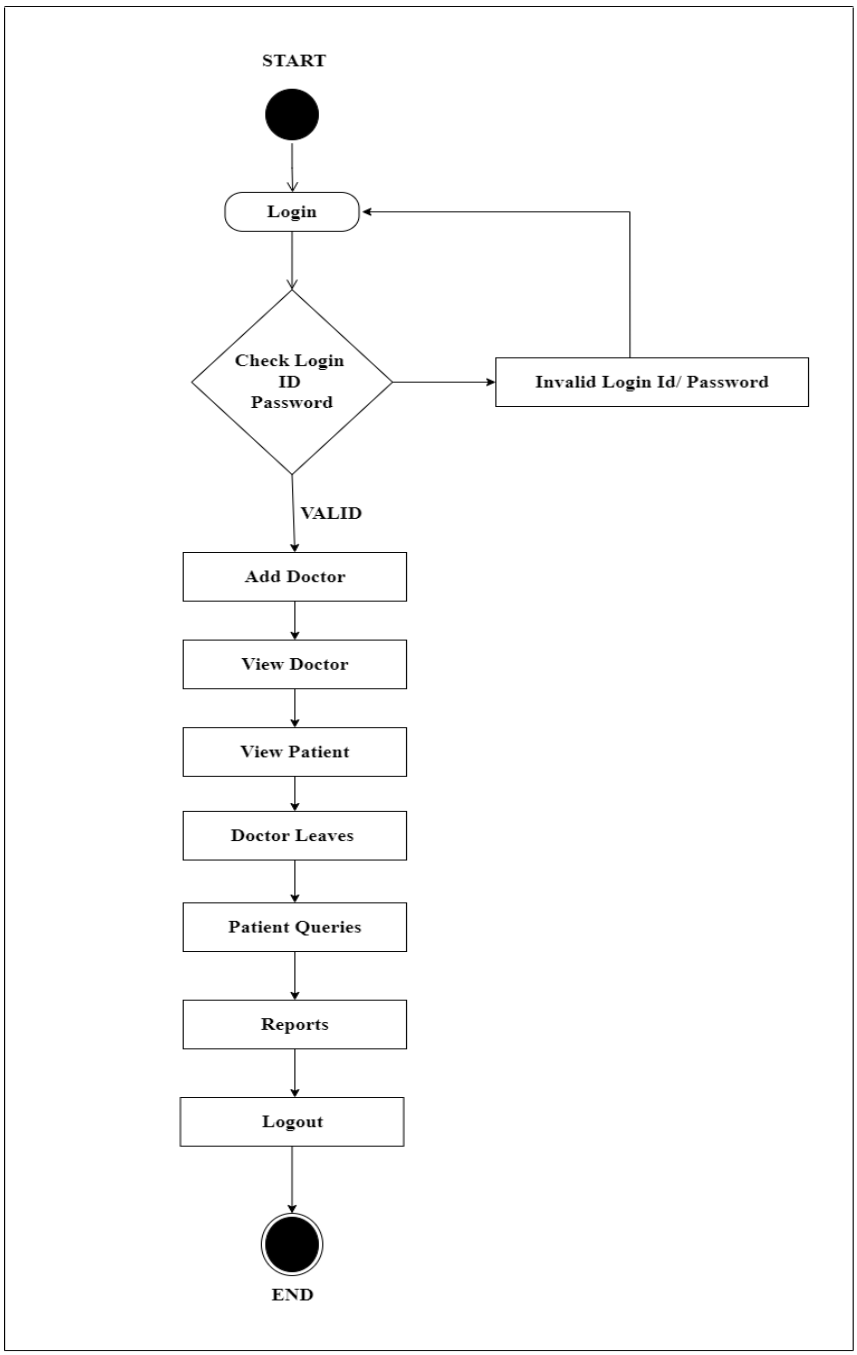
- **Object Diagram**



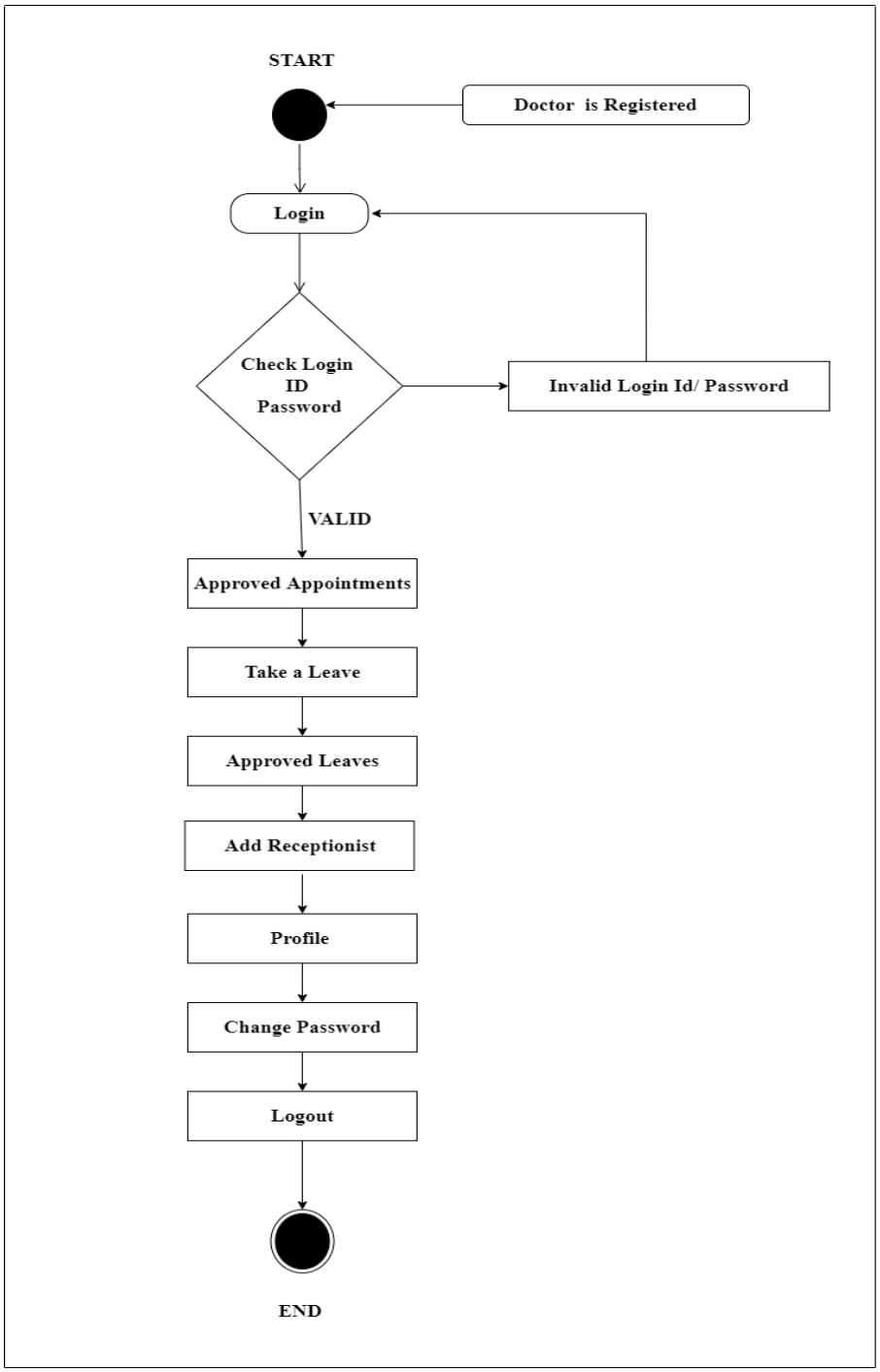
3.6 Activity Diagram and Sequence Diagram

- **Activity Diagram**

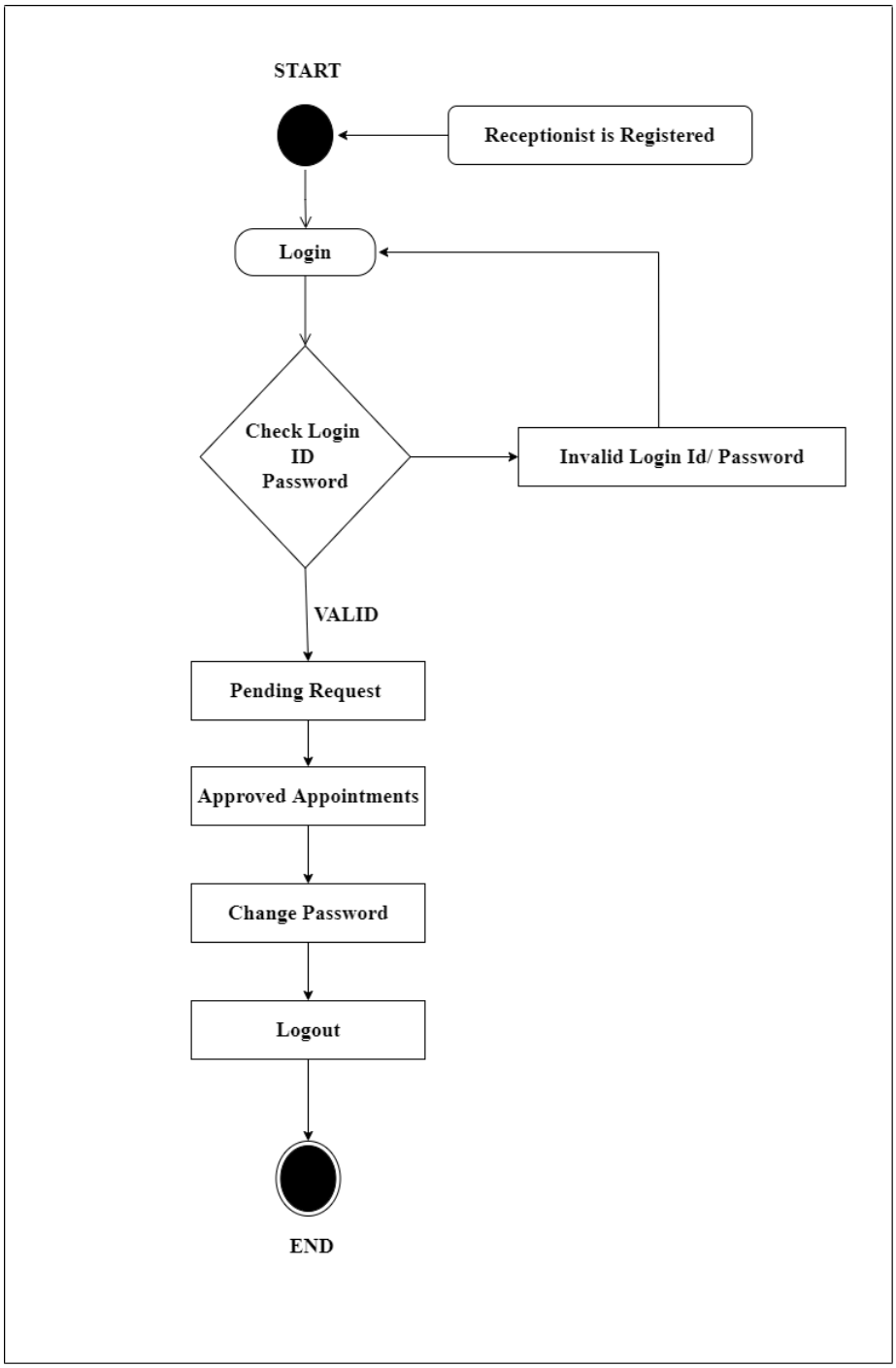
  Admin module



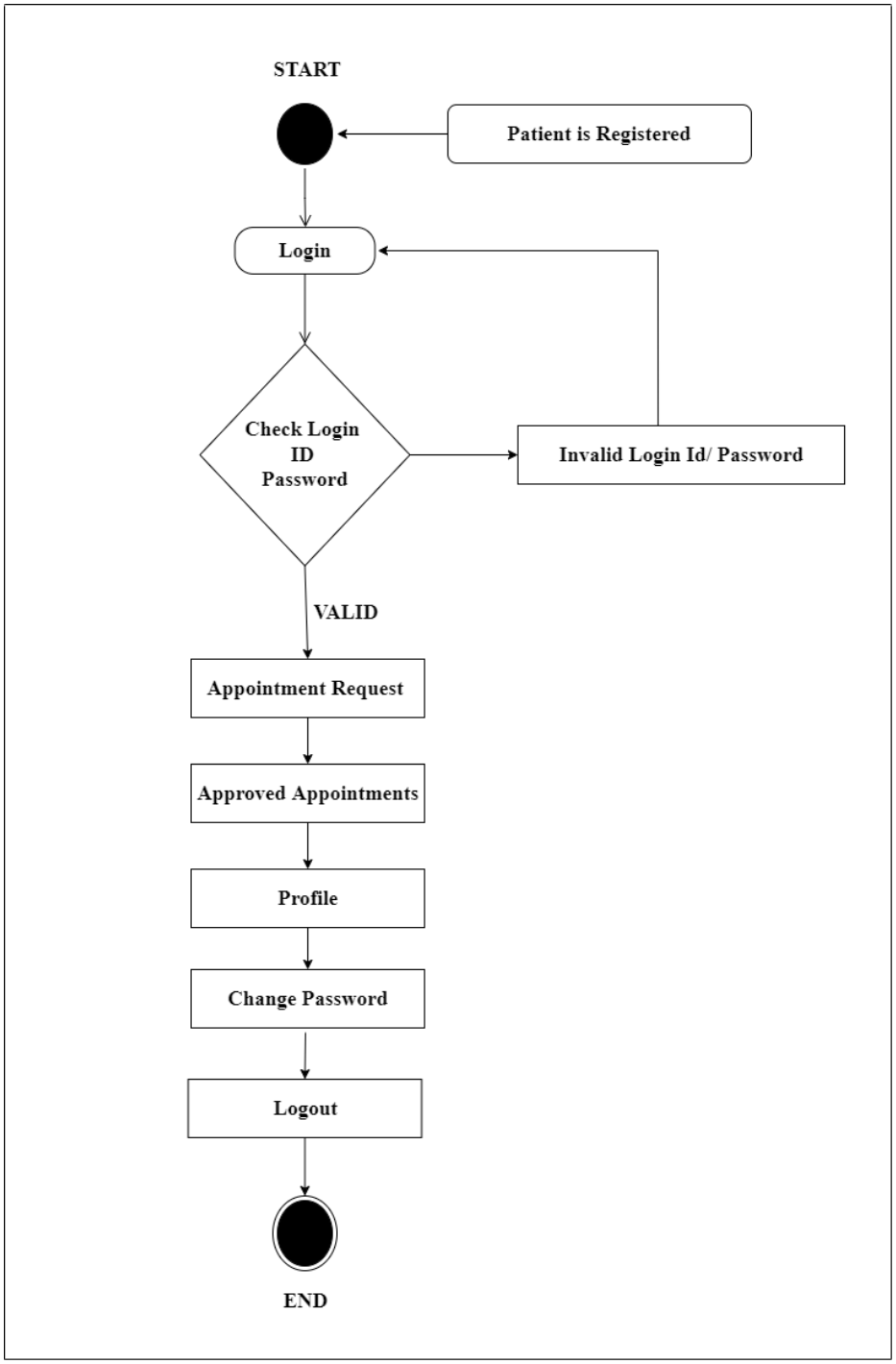
Doctor Module



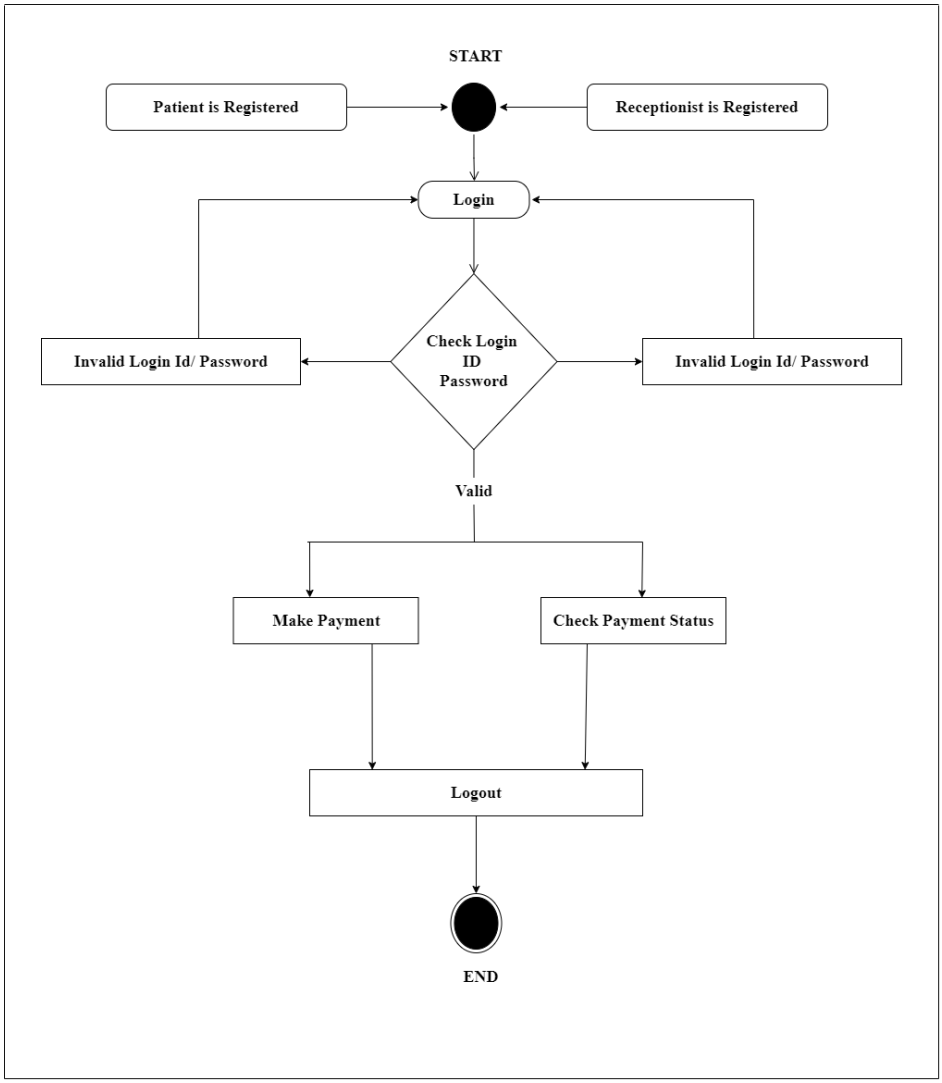
Receptionist Module



Patient Module

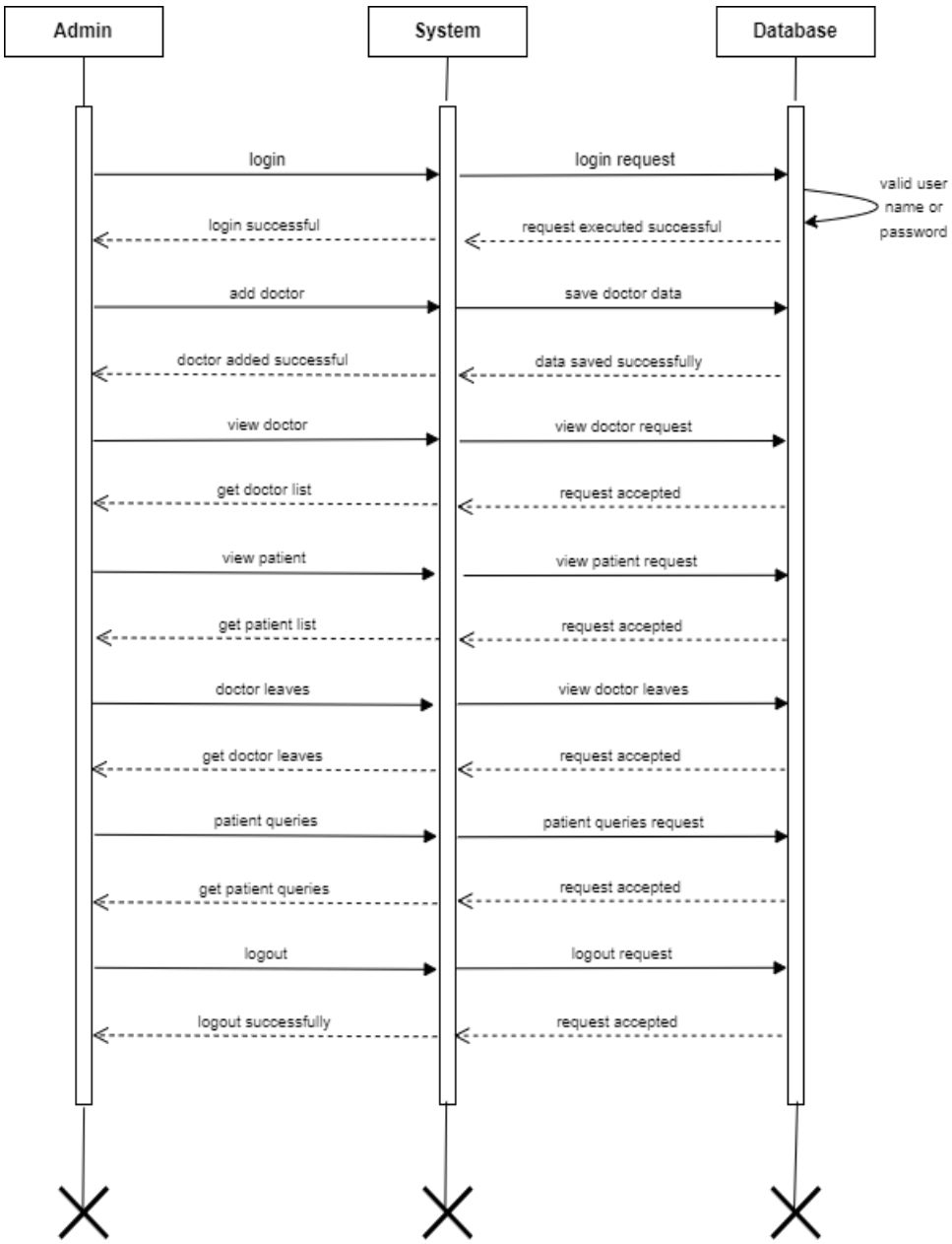


Payment Module

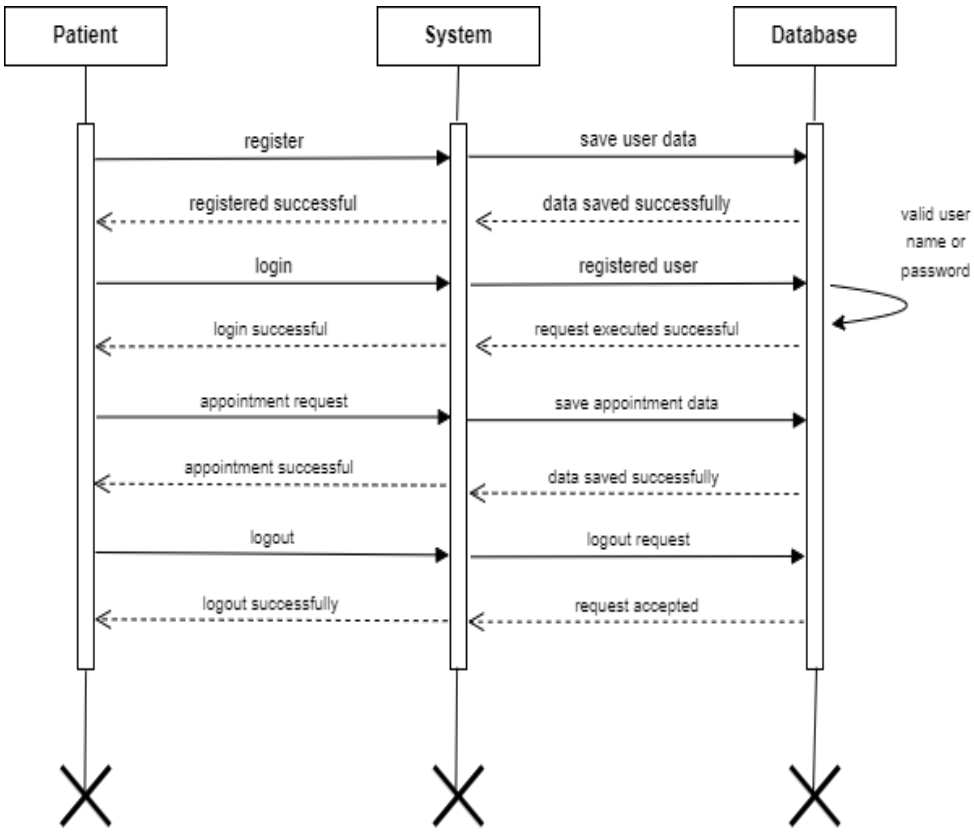


- **Sequence Diagram**

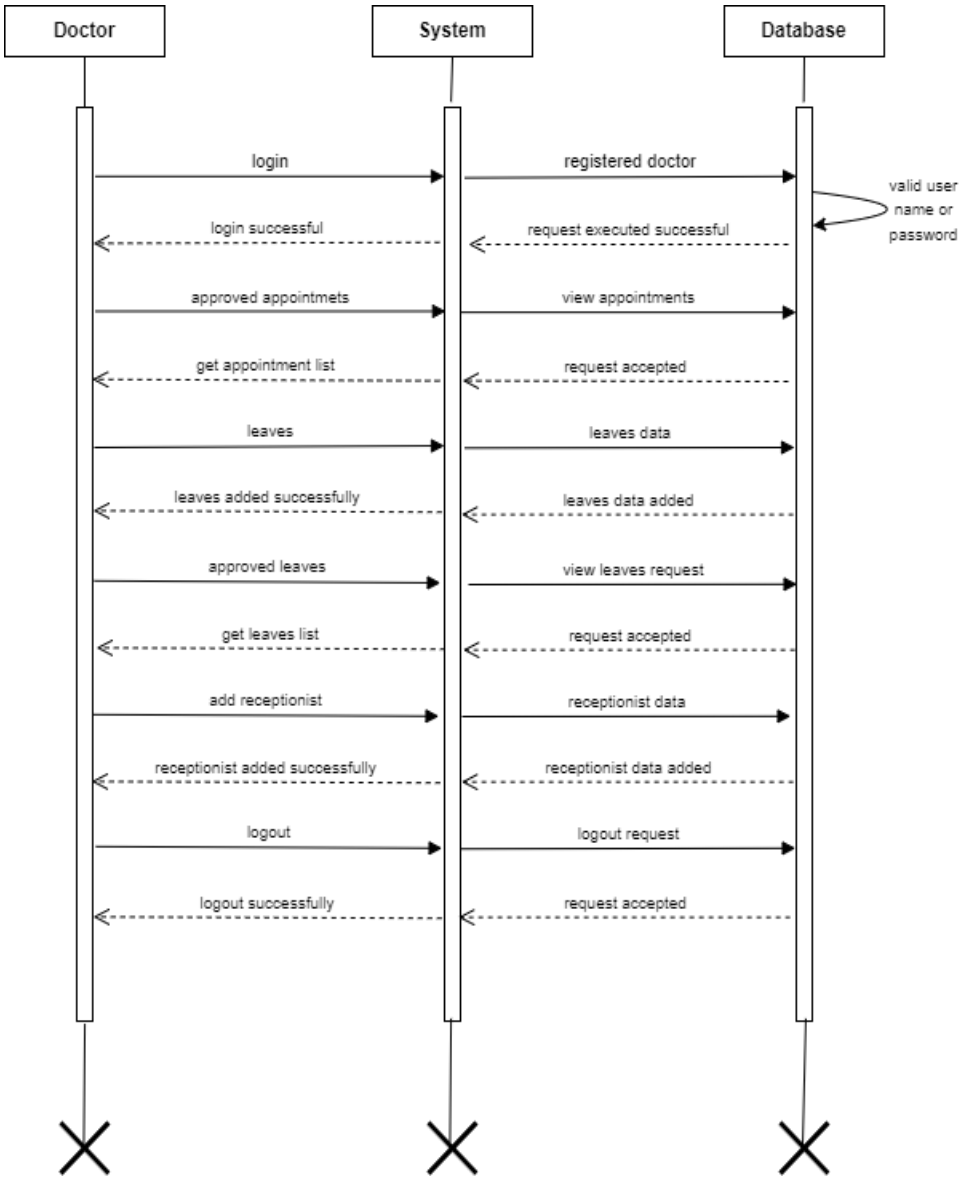
Admin



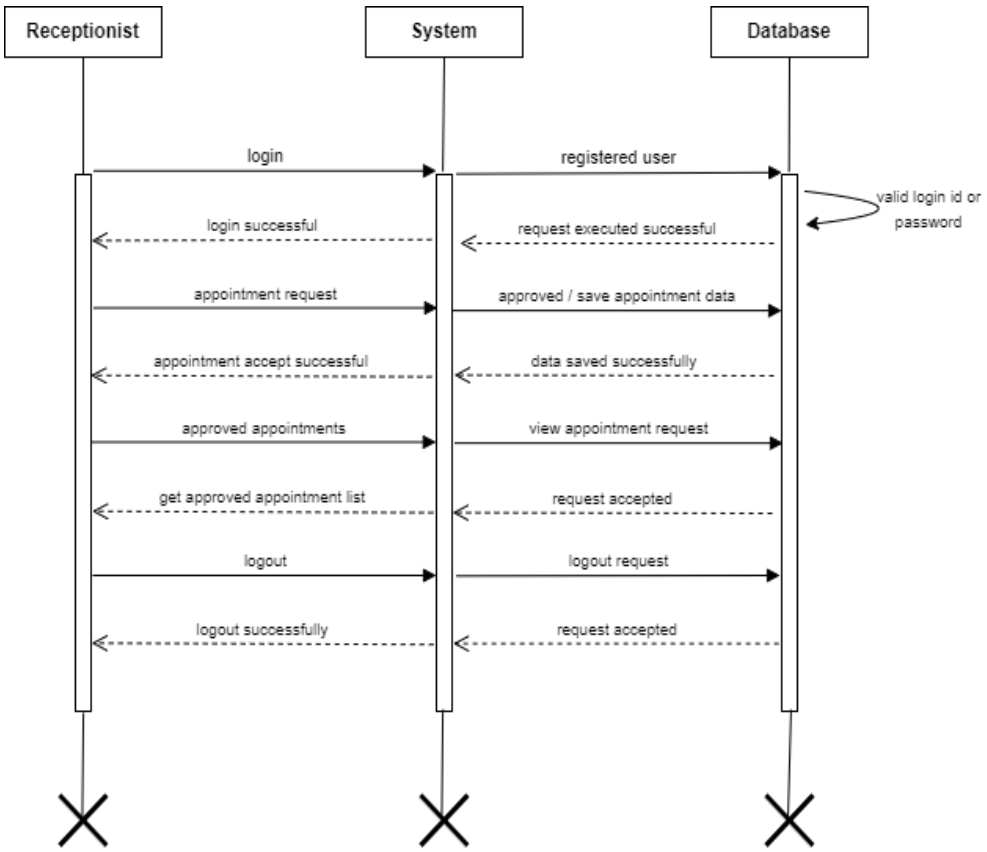
Patient


Patient
System
Database
register
save user data
registered successful
data saved successfully
valid user name or password
login
registered user
login successful
request executed successful
appointment request
save appointment data
appointment successful
data saved successfully
logout
logout request
logout successfully
request accepted

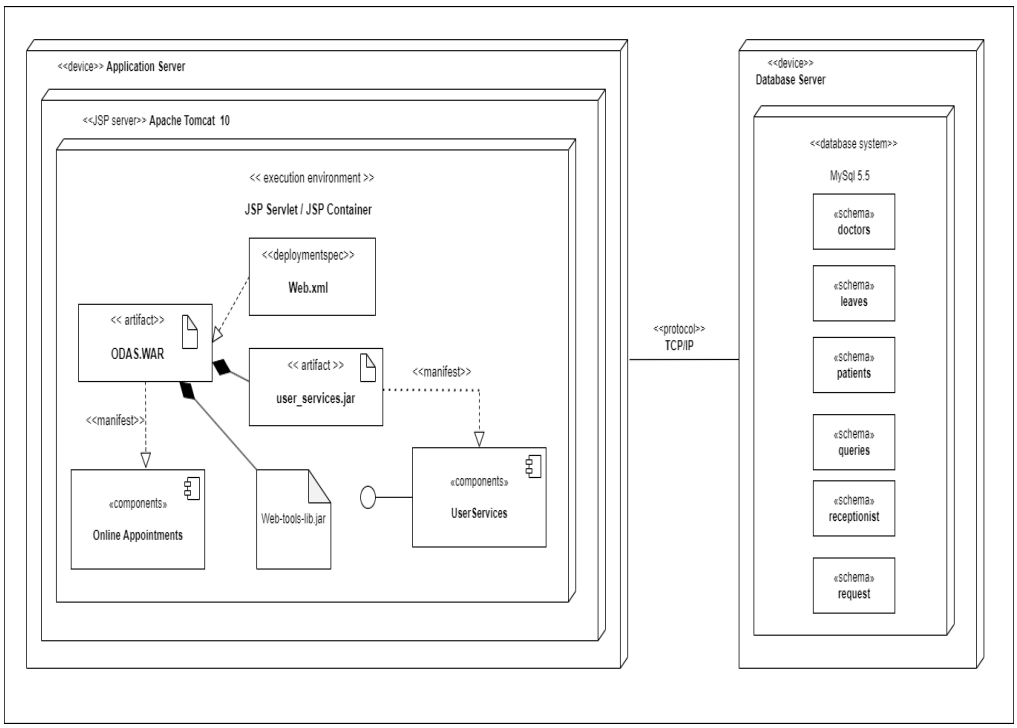
Doctor


Doctor
System
Database
login
registered doctor
valid user name or password
login successful
request executed successful
approved appointmets
view appointments
get appointment list
request accepted
leaves
leaves data
leaves added successfully
leaves data added
approved leaves
view leaves request
get leaves list
request accepted
add receptionist
receptionist data
receptionist added successfully
receptionist data added
logout
logout request
logout successfully
request accepted

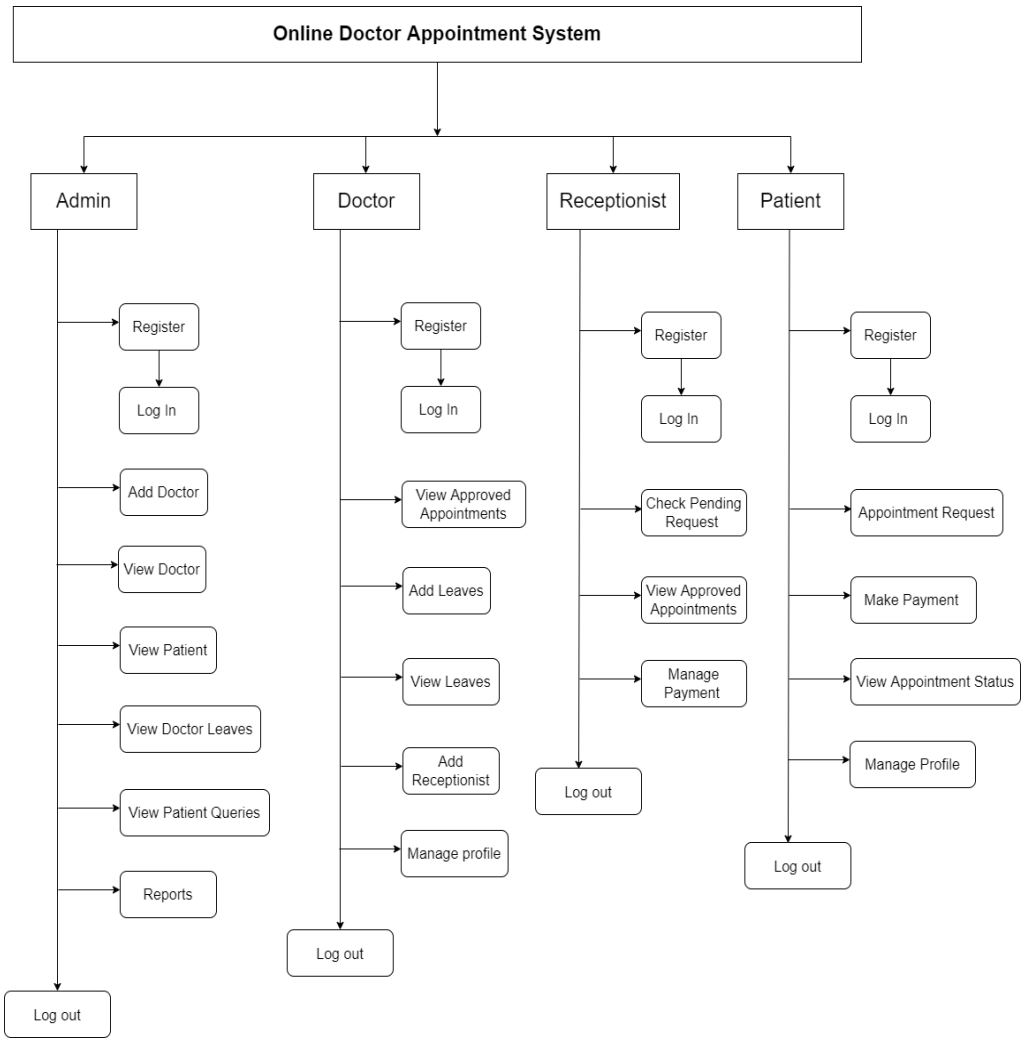
Receptionist



3.7 Deployment Diagram

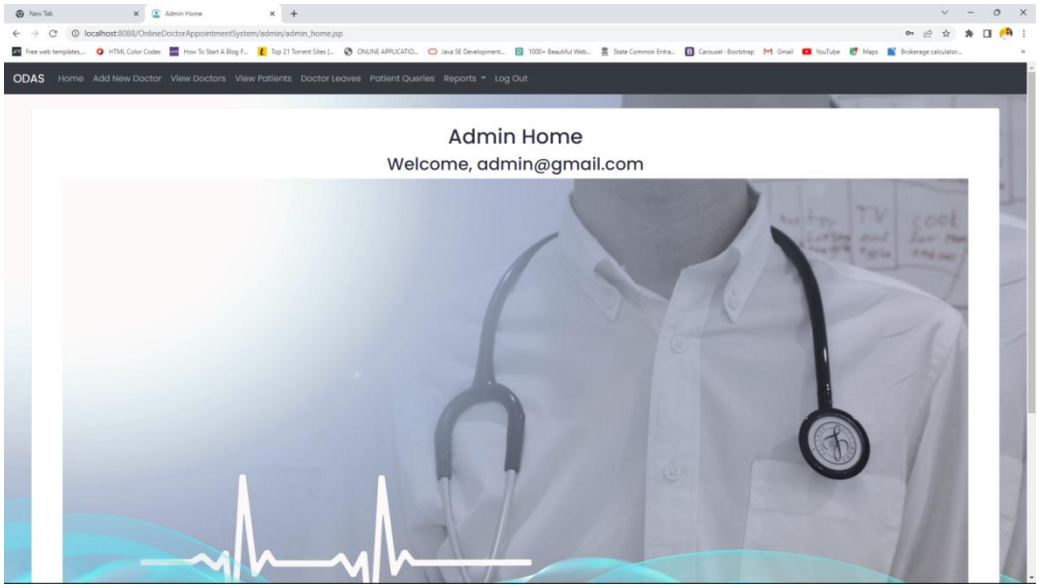


3.8 Module Hierarchy diagram

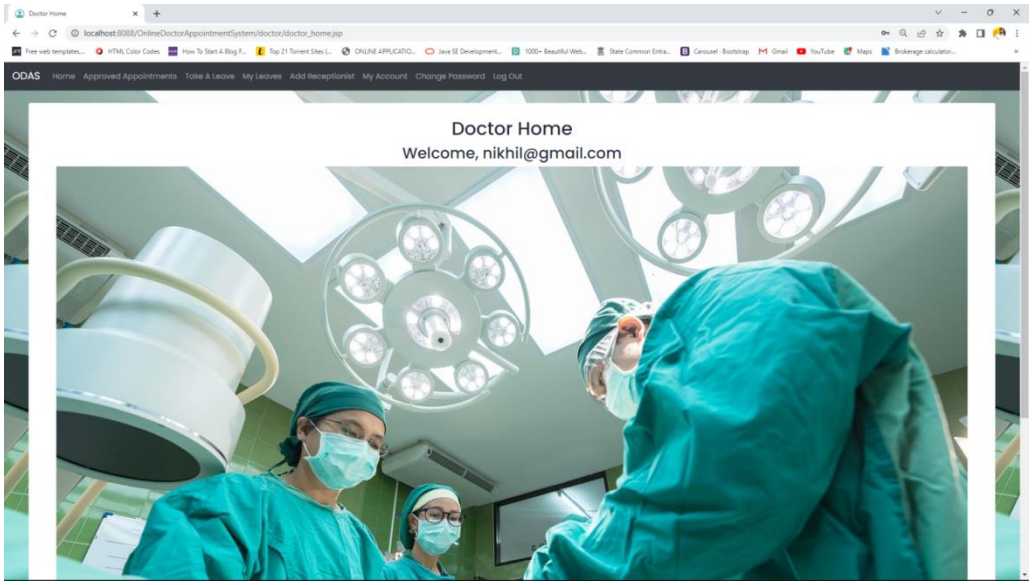


3.9 Sample Input and Output Screens

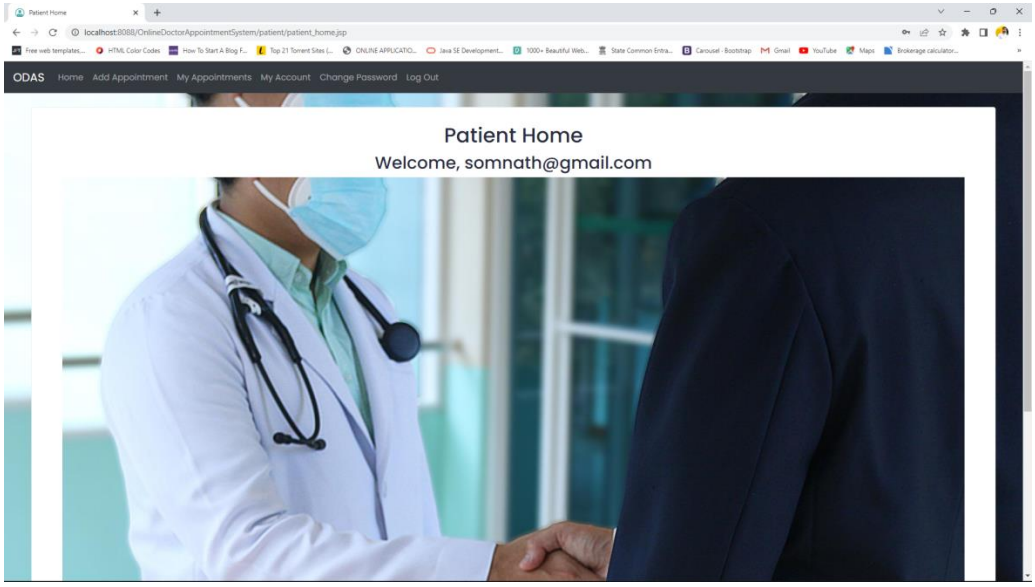
Admin home page



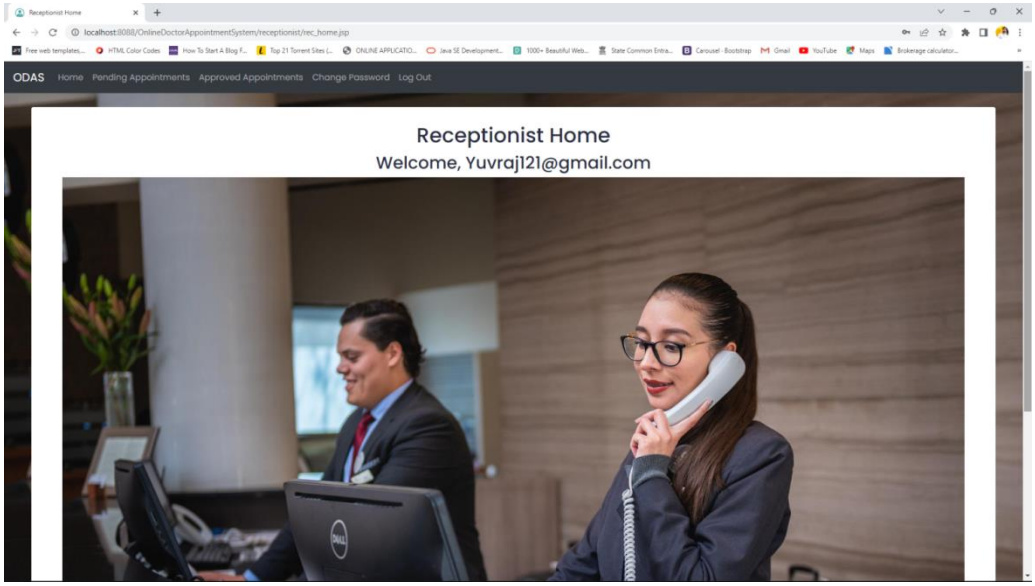
Doctor home page



Patient home page



Receptionist home page



Chapter 4: Coding

4.1 Code Snippets

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html lang="en">

<!-- doccure/  30 Nov 2019 04:11:34 GMT -->
<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
<title>Online Doctor Appointment System</title>

<!-- Favicons -->
<link rel="shortcut icon" type="image/x-icon" href="img/doc.ico" />

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="assets/css/bootstrap.min.css">

<!-- Fontawesome CSS -->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">
<link rel="stylesheet" href="assets/plugins/fontawesome/css/all.min.css">

<!-- Main CSS -->
<link rel="stylesheet" href="assets/css/style.css">

</head>
<body>

<!-- Main Wrapper -->
<div class="main-wrapper">

<jsp:include page="header.jsp"></jsp:include>
```

```
<!-- Home Banner -->
<section class="section section-search">
    <div class="container-fluid">
        <div class="banner-wrapper">
            <div class="banner-header text-center">
                <h1 class="mb-4">Online Doctor Appointment System</h1>
                    <h2>Search Doctor, Make an Appointment</h2>
                    <p>Discover the best doctors, clinic & hospital the city nearest to you.</p>
                </div>

        </div>
    </div>
</section>
<!-- /Home Banner -->

<!-- Clinic and Specialities -->
<section class="section section-specialities">
    <div class="container-fluid">
        <div class="section-header text-center">
            <h2>Clinic and Specialities</h2>
            <p class="sub-title">Health care services for day-to-day needs with facilities for consultation, diagnostics, health checks & more, all under one roof.</p>
        </div>
        <div class="row justify-content-center">
            <div class="col-md-9">
                <!-- Slider -->
                <div class="specialities-slider slider">

                    <!-- Slider Item -->
                    <div class="speicality-item text-center">
```

```
<div class="speicality-img">

<img src="assets/img/specialities/specialities-01.png"
class="img-fluid" alt="Speciality">

<span><i class="fa fa-circle" aria-hidden="true"></i></span>

</div>

<p>Urology</p>
</div>
<!-- /Slider
Item -->

<!-- Slider
Item -->
<div
class="speicality-item text-center">

<div class="speicality-img">

<img src="assets/img/specialities/specialities-02.png"
class="img-fluid" alt="Speciality">

<span><i class="fa fa-circle" aria-hidden="true"></i></span>

</div>

<p>Neurology</p>
</div>

<!-- /Slider
Item -->

<!-- Slider
Item -->
<div
class="speicality-item text-center">

<div class="speicality-img">
<img src="assets/img/specialities/specialities-03.png"
class="img-fluid" alt="Speciality">
```

```
<span><i class="fa fa-circle" aria-hidden="true"></i></span>

</div>

<p>Orthopedic</p>
                                                                    </div>

                                                                    <!-- /Slider
Item -->

                                                                    <!-- Slider
Item -->
                                                                    <div
class="speicality-item text-center">

<div class="speicality-img">

<img src="assets/img/specialities/specialities-04.png"
class="img-fluid" alt="Speciality">

<span><i class="fa fa-circle" aria-hidden="true"></i></span>

</div>

<p>Cardiologist</p>
                                                                    </div>

                                                                    <!-- /Slider
Item -->

                                                                    <!-- Slider
Item -->
                                                                    <div
class="speicality-item text-center">

<div class="speicality-img">

<img src="assets/img/specialities/specialities-05.png"
class="img-fluid" alt="Speciality">

<span><i class="fa fa-circle" aria-hidden="true"></i></span>
```

```
</div>

<p>Dentist</p>
                                                                    </div>

                                                                    <!-- /Slider
Item -->

                                                                    </div>
                                                                    <!-- /Slider -->

                                                                    </div>
                                                                    </div>
                                                                    </div>
                                                                    </section>
                                                                    <!-- Clinic and Specialities -->


                                                                    <!-- Availabe Features -->
                                                                    <section class="section section-features">
                                                                    <div class="container-fluid">
                                                                    <div class="row">
                                                                    <div class="col-md-5
features-img">
                                                                    <img
src="assets/img/features/medical.png" class="img-fluid mb-4"
alt="Feature">
                                                                    </div>
                                                                    <div class="col-md-7">
                                                                    <div
class="section-header">
                                                                    <h2
class="mt-2">Availabe Features in Our Clinics</h2>
                                                                    <p>Extra
touches in design and function make patients and their families feel
valued. We believe that even though their condition is undiagnosed
                                                                    and may
not be common, it is important to find answers for them in a setting
that is comfortable. </p>
                                                                    </div>
                                                                    <div
class="features-slider slider">
```

```
<!-- Slider Item -->
<div class="feature-item text-center">
<img src="assets/img/features/feature-01.jpg" class="img-fluid" alt="Feature">
<p>Patient Ward</p>
</div>
<!-- /Slider Item -->

<!-- Slider Item -->
<div class="feature-item text-center">
<img src="assets/img/features/feature-02.jpg" class="img-fluid" alt="Feature">
<p>Test Room</p>
</div>
<!-- /Slider Item -->

<!-- Slider Item -->
<div class="feature-item text-center">
<img src="assets/img/features/feature-03.jpg" class="img-fluid" alt="Feature">
<p>ICU</p>
</div>
<!-- /Slider Item -->

<!-- Slider Item -->
```

```
                                                    <div
class="feature-item text-center">

        <img    src="assets/img/features/feature-04.jpg"    class="img-
fluid" alt="Feature">

        <p>Laboratory</p>
                                                    </div>
                                                    <!-- /Slider
Item -->

                                                    <!-- Slider
Item -->
                                                    <div
class="feature-item text-center">

        <img    src="assets/img/features/feature-05.jpg"    class="img-
fluid" alt="Feature">

        <p>Operation</p>
                                                    </div>
                                                    <!-- /Slider
Item -->

                                                    <!-- Slider
Item -->
                                                    <div
class="feature-item text-center">

        <img    src="assets/img/features/feature-06.jpg"    class="img-
fluid" alt="Feature">

        <p>Medical</p>
                                                    </div>
                                                    <!-- /Slider
Item -->
                                                        </div>
                                                    </div>
                                                </div>
                                            </div>
                                        </section>
                                        <!-- Availabe Features -->
```

```
<jsp:include page="mainfooter.jsp"></jsp:include>
            </div>
            <!-- /Main Wrapper -->

                    <!-- jQuery -->
                    <script src="assets/js/jquery.min.js"></script>

                    <!-- Bootstrap Core JS -->
                    <script src="assets/js/popper.min.js"></script>
                    <script src="assets/js/bootstrap.min.js"></script>

                    <!-- Slick JS -->
                    <script src="assets/js/slick.js"></script>

                    <!-- Custom JS -->
                    <script src="assets/js/script.js"></script>

          </body>
</html>
```

Contactus.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
          pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html lang="en">

<!-- doccure/  30 Nov 2019 04:11:34 GMT -->
<head>

<meta charset="utf-8">
<meta name="viewport"
          content="width=device-width, initial-scale=1.0, user-scalable=0">
<title>Online Doctor Appointment System</title>

<!-- Favicons -->
<link rel="shortcut icon" type="image/x-icon" href="img/doc.ico" />

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="assets/css/bootstrap.min.css">
```

```
<!-- Fontawesome CSS -->
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">
<link rel="stylesheet" href="assets/plugins/fontawesome/css/all.min.css">

<!-- Main CSS -->
<link rel="stylesheet" href="assets/css/style.css">

<style type="text/css">
.buttonHolder {
	text-align: center;
}
</style>

<script type="text/javascript">
	function validateform() {

		var name = document.myform.name.value;
		var mobile = document.myform.mobile.value;
		var email = document.myform.email.value;
		atpos = email.indexOf("@");
		dotpos = email.lastIndexOf(".");

		var topic = document.myform.topic.value;
		var message = document.myform.message.value;

		if (name == null || name == "") {
			alert("Please Enter Your Name");
			document.myform.name.focus();
			return false;
		} else if (mobile.length < 10) {
			alert("Invalid Mobile Number");
			document.myform.mobile.focus();
			return false;
		} else if (email == "" || atpos < 1 || (dotpos - atpos < 2))
{
			alert("Invalid Email ID");
			document.myform.email.focus();
			return false;
		} else if (topic == null || topic == "") {
			alert("Please Enter Topic..");
```

```
                document.myform.topic.focus();
                        return false;
                }
                else if (message == null || message == "") {
                        alert("Please Enter Some Message..");
                        document.myform.message.focus();
                        return false;
                }
        }
</script>

</head>
<body>

        <!-- Main Wrapper -->
        <div class="main-wrapper">

                <jsp:include page="header.jsp"></jsp:include>

                <!-- Home Banner -->
                <section class="section section-search">
                        <div class="container-fluid">
                                <div class="banner-wrapper">
                                        <div class="banner-header text-
center">
                                                <h1>Contact Us For Any
Query</h1>
                                                <p>
                                                        Please complete
the form below to reach Admin. <br> An
                                                        asterisk (*)
indicates required information.
                                                </p>
                                        </div>
                                </div>
                        </div>
                </section>
                <!-- /Home Banner -->

                <div class="col-lg-8 mx-auto px-4 py-4">
                        <form action="queriesdb.jsp" name="myform"
method="post"
```

```
onsubmit="return validateform();">
<div class="row px-3">
    <div class="col-md">
        <label class="mb-1">
</label> <input class=" form-control mb-4"
            type="text"
name="name" maxlength="40"

    placeholder="*Your Full Name">
    </div>

    <div class="col-sm">
        <label class="mb-1">
</label> <input class="form-control"
            type="email"
name="email" placeholder="*Enter Your Email">
        <span  class=""  id="err">
</span>
    </div>


</div>

<div class="row px-3">
    <div class="col-md">
        <label class="mb-1">
            <h6  class="mb-0
text-sm"></h6>
        </label>  <input  class="
form-control mb-4" type="text" name="topic"
            maxlength="40"
placeholder="Your Query Topic">
    </div>

    <div class="col-md">
        <label class="mb-1">
</label> <input class="form-control mb-4"
            maxlength="10"
type="text" name="mobile"

    placeholder="*Your Phone Number"
```

```
oninput="this.value = this.value.replace(/[^0-9.]/g, ").replace(/(\..*)\./g, '$1');">
                    </div>


                </div>


                <div class="row px-3">
                    <div class="col-md">
                        <label class="mb-1">
</label>
                        <textarea
name="message" id="message" class="form-control"
                                rows="8"
placeholder="*Your Message"></textarea>
                    </div>
                </div>

                <div class="buttonHolder mt-4">
                    <input type="submit"
value="Send Message" class="btn btn-success">
                </div>

            </form>
        </div>

        <jsp:include page="mainfooter.jsp"></jsp:include>
    </div>
    <!-- /Main Wrapper -->

    <!-- jQuery -->
    <script src="assets/js/jquery.min.js"></script>

    <!-- Bootstrap Core JS -->
    <script src="assets/js/popper.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>

    <!-- Slick JS -->
    <script src="assets/js/slick.js"></script>

    <!-- Custom JS -->
```

```
        <script src="assets/js/script.js"></script>

</body>
</html>
```

Login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>
<html lang="en">

<head>
 <meta charset="utf-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">
                <title>Online Doctor Appointment System</title>

                <!-- Favicons -->
                <link rel="shortcut icon" type="image/x-icon" href="img/doc.ico" />

                <!-- Bootstrap CSS -->
                <link rel="stylesheet" href="assets/css/bootstrap.min.css">

                <!-- Fontawesome CSS -->
                <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css">
                <link rel="stylesheet" href="assets/plugins/fontawesome/css/all.min.css">

                <!-- Main CSS -->
                <link rel="stylesheet" href="assets/css/style.css">
                <script src="https://unpkg.com/sweetalert/dist/sweetalert.min.js"></script>
<script src="js/cities.js"></script>
   <style>
      #userimg {
         width: 150px;
```

```
            display: block;
            margin: auto;
        }
        .image {
    width: 380px;
    height: 445px;
}

    </style>
    <script>
function validateform(){
var email=document.myform.email.value;
var password=document.myform.password.value;
atpos = email.indexOf("@");
dotpos = email.lastIndexOf(".");


if (email==null || email==""){
  alert("Email can't be blank");
  return false;
}
else if(email == "" || atpos < 1 || ( dotpos - atpos < 2 )){
            alert("Invalid Email..");
              document.myform.email.focus() ;
              return false;
              }
else if(password=="" || password==null){
            alert("Please Enter Password!");
              document.myform.password.focus() ;
              return false;
              }
}

function myFunction() {
              var x = document.getElementById("myInput");
              if (x.type === "password") {
                x.type = "text";
              } else {
                x.type = "password";
              }
            }
</script>
```

```
</head>

<body>
<jsp:include page="header.jsp"></jsp:include>

      <div class="card card0 border-0">
         <div class="row d-flex">
            <div class="col-lg-6 border-line">
               <div class="card1 ">
                  <div class="row px-3 justify-content-center mt-5 mb-2
">
                     <img src="assets/img/a.svg" class="image">
                  </div>
               </div>
            </div>


            <div class="col-lg-6">
               <div class="card2 card border-0 px-4 py-5">

                  <div class="row mb-4 px-3 mx-auto">
                     <h1>Sign In</h1>
                   </div>
                   <hr>

                  <div class="row mb-4 px-3">
                     <img id="userimg" src="assets/img/user.png" alt="">
                  </div>
                     <form action="logincheck.jsp" name="myform"
onsubmit="return validateform()">
                        <div class="row px-3">
                           <label class="mb-1">
                              <h6 class="mb-0 text-sm">User</h6>
                           </label>
                           <select class="form-control mb-4"
name="acctype">
                              <option value="admin"> Admin </option>
                              <option value="doctor"> Doctor</option>
                              <option value="patient"> Patient </option>
                              <option value="receptionist">  Receptionist
</option>
                           </select>
```

```
</div>

<div class="row px-3">
   <label class="mb-1">
      <h6 class="mb-0 text-sm">Email
Address</h6>
   </label>
   <input class="form-control mb-4" type="email"
name="email" placeholder="Enter a valid email address">
</div>

<div class="row px-3"> <label class="mb-1">
      <h6 class="mb-0 text-sm">Password</h6>
   </label> <input type="password"
class="form-control" id="myInput" name="password"
maxlength="15" placeholder="Enter password"> </div>

<div class="row px-3 mb-4 mt-2">
   <div class="custom-control custom-checkbox
custom-control-inline">
      <input id="chk1" type="checkbox"
name="chk" onclick="myFunction()" class="custom-control-input">
      <label for="chk1" class="custom-control-label
text-sm">Show Password</label>
   </div>
   <a href="forgetpassword.jsp" class="ml-auto
mb-0 text-sm">Forgot Password?</a>
</div>


                                    <div class="row
mb-3 px-3">
                                    <button
type="submit" class="btn btn-success text-center mr-4"
>Login</button>
                                    <button
type="reset" value="reset"

      class="btn btn-danger text-center">Reset</button>

                                    </div>
```

```
                    <div class="row mb-4 px-3"> <small class="font-
weight-bold">Don't have an account? <a
                        href="register.jsp"        class="text-danger
">Register</a></small> </div>
                  </form>
              </div>
            </div>
          </div>
          <jsp:include page="mainfooter.jsp"></jsp:include>
      </div>

  <!-- /Main Wrapper -->

                <!-- jQuery -->
                <script src="assets/js/jquery.min.js"></script>

                <!-- Bootstrap Core JS -->
                <script src="assets/js/popper.min.js"></script>
                <script src="assets/js/bootstrap.min.js"></script>

                <!-- Slick JS -->
                <script src="assets/js/slick.js"></script>

                <!-- Custom JS -->
                <script src="assets/js/script.js"></script>
</body>

</html>
```

Logincheck.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-
8859-1"
   pageEncoding="ISO-8859-1" %>
   <%@ page import="java.sql.*" %>

   <%
         //check condition admin login not back index.jsp(login) page
         if(session.getAttribute("admin_login")!=null){
```

```
                response.sendRedirect("admin/admin_home.jsp");
        }

        //check condition employee login not back index.jsp(login)
page
        if(session.getAttribute("doctor_login")!=null){
        response.sendRedirect("doctor/doctor_home.jsp");
        }

        //check condition user login not back index.jsp(login) page
                if(session.getAttribute("patient_login")!=null){
                response.sendRedirect("patient/user_home.jsp");
                }

                //check condition user login not back index.jsp(login)
page
                if(session.getAttribute("rec_login")!=null){
                response.sendRedirect("receptionist/rec_home.jsp");
                }

   %>



<%
Class.forName("com.mysql.jdbc.Driver");
Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/odas
","root","root");

String acctype=request.getParameter("acctype");
String email=request.getParameter("email");
String password=request.getParameter("password");
if(acctype.equals("admin")){
        if(email.equals("admin@gmail.com")                    &&
password.equals("admin") ){

                session.setAttribute("admin_login",email);      //session
name is "admin_login"
      response.sendRedirect("admin/admin_home.jsp");   //after   login
success redirect to "admin_home.jsp" page in admin folder
```

```
            }

}else if(acctype.equals("doctor")){


    PreparedStatement ps=con.prepareStatement("select * from doctors
where email=? and password=?");
    ps.setString(1, email);
    ps.setString(2, password);
    ResultSet rs=ps.executeQuery();
    if(rs.next()){
            session.setAttribute("doctor_login",email); //session name is
"doctor_login"
            response.sendRedirect("doctor/doctor_home.jsp"); //after login
success redirect to "doctor_home.jsp" page in doctor folder

    }
}else if(acctype.equals("patient")){
            PreparedStatement ps=con.prepareStatement("select * from
patient where email=? and password=?");
    ps.setString(1, email);
    ps.setString(2, password);
    ResultSet rs=ps.executeQuery();
    if(rs.next()){

            session.setAttribute("patient_login",email);//session name is
"patient_login"
        response.sendRedirect("patient/patient_home.jsp"); //after login
success redirect to "patient_home.jsp" page in patient folder

    }
}
else if(acctype.equals("receptionist")){
            PreparedStatement ps=con.prepareStatement("select * from
receptionist where email=? and password=?");
    ps.setString(1, email);
    ps.setString(2, password);
    ResultSet rs=ps.executeQuery();
    if(rs.next()){

            session.setAttribute("rec_login",email);//session name is
"patient_login"
```

```
        response.sendRedirect("receptionist/rec_home.jsp");  //after  login
success redirect to "patient_home.jsp" page in patient folder

    }
}

%>
        <script>

        alert("Please Check User name and Password");
        location.href="login.jsp";
        </script>
        <%

%>
```

Chapter 5: Testing

5.1 Test Strategy

Software Testing is critical element of s/w quality assurance & represent the ultimate review of the specification, design and code generation. It is the process of executing a program with a primary objective of finding errors or bugs. Testing gives the guarantee that the s/w does not fail and runs according to its specification and in the way the end user experts. The following techniques were used in order to uncover the errors in the system:

Black Box Testing

Black box testing is also known as behavioral testing. It focuses on the functionality requirements of the s/w. it is related to i/p and o/p only. This helps further in detecting following errors.

- Initializing and terminating errors.
- Behavior and performance errors.
- Interface errors.
- Incorrect or missing functions.
- Errors in data structure and databases.

White Box Testing

White box testing is also known as glass box testing. It is related with the Internal logic of the system TEST PROCEDURE -

Software Testing is critical element of s/w quality assurance & represent the ultimate review of the specification, design and code generation. It is the process of executing a program with a primary objective of finding errors or bugs. Testing gives the guarantee that the

s/w does not fail and runs according to its specification and in the way the end user experts. The following techniques were used in order to uncover the errors in the system.

Integration Testing

Testing is a major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests.

To uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design. In a non - incremental integration all the modules are combined in advance and the program is tested as a whole. Here errors will appear in an endless loop function. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected. Different incremental integration strategies are top – down integration, bottom – up integration, regression testing.

System Testing

Here all the pre-tested individual modules will be assembled to create the larger system and tests are carried out at system level to make sure that all modules are working in synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually are also running in cohesion with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test

paths throughout the system to make sure that no path are making its way into chaos.

Regression Testing

Regression testing focuses on finding defects after a major code change has occurred. Specifically, it seeks to uncover software regression, or old bugs that have come back. Such regressions occur whenever software functionality that was previously working correctly stops working as intended.

Each time a new module is added as a part of integration as the software changes. Regression testing is an actually that helps to ensure changes that do not introduce unintended behavior as additional errors.

Regression testing maybe conducted manually by executing a subset of all test cases or using automated capture play back tools enables the software engineer to capture the test case and results for subsequent playback and compression. The regression suit contains different classes of test cases. A representative sample to tests that will exercise all software functions. Additional tests that focus on software functions that are likely to be affected by the change

TEST PLAN

This activity is done mainly on 3 levels.

1. Computer operator level.
2. Programmer level.
3. System analyst and designer level.

Testing with test data

Dummy data is entered to check the overall functionality of various programs with proper validation checks the artificial test data was prepared based on the following checks:

Range checks: data was chosen in such a way that there were value exceeding range, outside the range and values at the boundaries.

Field Size check data was chosen in such a way that there were value exceeding the maximum width of the field.

Key Fields data was chosen so that primary key and foreign key Constraints would be violated.

Data type checks: data was chosen in such a way that it would not match with the data types.

Testing for logical errors

The system is tested and debugged accordingly (module wise debugging with serial documentation of bugs).

Testing for presentation errors

After regular Intervals, the system is tested with user better graphical user interface and customer needs. Changes are made in error messages and displaying help items.

5.2 Unit Test Plan

Unit testing is a type of software testing where individual units or components of software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object.

In SDLC, STLC, V Model, Unit testing is first level of testing done before integration testing. Unit testing is a White Box testing technique that is usually performed by the developer. Though, in a practical world due to time crunch or reluctance of developers to tests, QA engineers also do unit testing.

- Unit tests help to fix bugs early in the development cycle and save costs.
- It helps the developers to understand the testing code base and enables them to make changes quickly
- Good unit tests serve as project documentation
- Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code until the tests run again.

5.3 Acceptance Test Plan

Acceptance Testing is a method of software testing where a system is tested for acceptability. The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not. Standard Definition of Acceptance Testing:

It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not.

Types of Acceptance Testing:

User Acceptance Testing (UAT): User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose. This is also termed as End-User Testing.

Business Acceptance Testing (BAT): BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits which are quite challenging due to the changing market conditions and new technologies so the current implementation may have to being changed which results in extra budgets.

Contract Acceptance Testing (CAT): CAT is a contract that specifies that once the product goes live, within a predetermined period, the

acceptance test must be performed and it should pass all the acceptance use cases. Here is a contract termed a Service Level Agreement (SLA), which includes the terms where the payment will be made only if the Product services are in-line with all the requirements, which means the contract is fulfilled. Sometimes, this contract happens before the product goes live. There should be a well-defined contract in terms of the period of testing, areas of testing, conditions on issues encountered at later stages, payments, etc.

Regulations Acceptance Testing (RAT): RAT is used to determine whether the product violates the rules and regulations that are defined by the government of the country where it is being released. This may be unintentional but will impact negatively on the business. Generally, the product or application that is to be released in the market, has to go under RAT, as different countries or regions have different rules and regulations defined by its governing bodies. If any rules and regulations are violated for any country then that country or the specific region then the product will not be released in that country or region. If the product is released even though there is a violation then only the vendors of the product will be directly responsible.

Operational Acceptance Testing (OAT): OAT is used to determine the operational readiness of the product and is non-functional testing. It mainly includes testing of recovery, compatibility, maintainability, reliability, etc. OAT assures the stability of the product before it is released to production.

Alpha Testing: Alpha testing is used to determine the product in the development testing environment by a specialized tester's team usually called alpha testers.

Beta Testing: Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience.

Use of Acceptance Testing:

- To find the defects missed during the functional testing phase.
- How well the product is developed.
- A product is what actually the customers need.
- Feedback help in improving the product performance and user experience.
- Minimize or eliminate the issues arising from the production.

5.4 Test Case/ Test Script

Testing of project problem statement using generated test data (using mathematical models, GUI, Function testing principles, if any) selection and appropriate use of testing tools, testing of UML diagram's reliability.

Module-ID: 01

Modules to be tested: Registration

1. Enter the case insensitive Username click on Submit button.

Expected: It should display error.

2. Enter the case sensitive Username click on Submit button.

Expected: It should accept.

3. Enter the case insensitive Password click on Submit button.

Expected: It should display error.

4. Enter the case sensitive Password click on Submit button.

Expected: It should accept.

5. Enter the case insensitive Mobile Number click on Submit button.

Expected: It should display error.

6. Enter the case sensitive Mobile Number click on Submit button.

Expected: It should accept.

7. Enter the wrong address and click on Submit button.

Expected: It should display error.

8. Enter the correct address and click on Submit button.

Expected: It should accept.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
101	Enter the case insensitive Username click on Submit button.	Username	Error comes	Error Should come	P
102	Enter the case sensitive Username click on Submit button.	Username	Accept	Accept Username	P
201	Enter the case insensitive Password click on Submit button.	Password	Error comes	Error Should come	P
202	Enter the case sensitive Password click on Submit button	Password	Accept	Accept	P
301	Enter the case insensitive Mobile Number click on Submit button	Mobile Number	Error comes	Error Should come	P

302	Enter the case sensitive Mobile Number click on Submit button.	Mobile Number	Accept	Accept	P

Module-ID: 2

Modules to be tested: Login

1. Enter the correct username and wrong password click on Submit button.

Expected: It should display error.

2. Enter the wrong username and correct password and click on Submit button.

Expected: It should display error.

3. Enter the correct username and password and click on Login button.

Expected: It should display welcome page.

4. After login with valid credentials click on back button.

Expected: The page should be expired.

5. After login with valid credentials copy the URL and paste in another browser.

Expected: It should not display the user's welcome page.

6. Check the password with Lower case and upper case.

Expected: Password should be case sensitive.

Test Case_ID	Description	Test case I/P	Actual Result	Expected result	Test case criteria (P/F)
001	Enter the correct username and wrong password click on Login button.	Username Password	Error comes	Error Should come	P
002	Enter the wrong username and correct password click on Login button,	Username Password	Error comes	Error Should come	P
003	Enter the correct username and password and click on Login button.	Username Password	Accept	Accept	P

Chapter 6: Limitations of Proposed System

Internet Connection:

To book an appointment online, you definitely need internet connection; be that on your phone or using a computer. In the case of no internet, you will be unable to do this. Of course, in today's day and age, this doesn't seem like a major issue, since we have access to the internet almost at all times. However, if the connection is absent for some reason, you will run into this complication. For businesses, on the other hand, this could be a bit more problematic. If you lose access to the internet in the office, for example, you won't be able to see your bookings, cancellations, and what you have going on for the coming days. Here, we suggest that when you choose an internet provider, make sure that it is a reliable one that will give you the least amount of connection issues.

High Demand:

When you provide the option for online appointment scheduling, you open your doors to new customers. The easiness of booking an appointment online makes your service accessible to almost anyone. Depending on your business and how you market it, this can create a large influx of demand for your service. The result- a very large volume of bookings that need to be handled. As a business, this can be very overwhelming for you. This increase in demand may force you to expand your resources, hire new staff, upgrade your existing means, etc.

Uncertainty:

An online appointment scheduling system can create uncertainty and confusion for your customers. The extent of uncertainty definitely depends on the type of services you provide. Here is an example. If you are a doctor and you allow your patients to book appointments via an online scheduling platform, you will have given them the option to choose which service they want from you. However, some patients may not be able to identify exactly what they need from you, which could force them to book the wrong appointment type.

Chapter 7: Proposed Enhancement

- We will add feature for online counseling for patients so that they get proper guidance on time.

- Also, patient will able to upload their past treatment documents for reference of their disease because of which doctor will get prior knowledge of patient's treatment.

- We will try to add chat box to set up conversation between patient and doctor due to which patient's problems can be elaborated in detail If patient's medicines are about to over the patient will able to get consultation from doctor on time.

- Patient will get notification about their appointment before 1 hour of the booked appointment.

- We will add online fees paying through payment gateway so that doctor will get only him patients who really want to take treatment.

- Online system is always a changeable system. It develops day by day, getting better and better to easier for peoples.

- This could be a revolutionary web application that may help bonding between doctor and patient.

- We believe we can make this system more advanced in future. Advance features and User interface will be updated in future.

- Our system is already user friendly but we will try to make this system more users friendly in future.

Chapter 8: Conclusion

- Having the ability to help sick people get better is a rare privilege that only doctors and a few other professionals have. While it gives great fulfillment to see your patients regain their health, you want to make the process of receiving treatment easy for them. The online appointment system gives you a chance to show that you are interested in the welfare of your patients even before they get to meet you.

- The booking system also allows people in different locations to come to you for treatment. While being an online medical doctor will help expand your client base, remember that you will also be putting a smile on the faces of those who have benefited from your efficient service. The Online Appointment System is a trend that will continue to gain popularity. It has come to stay and will make appointment scheduling a piece of cake for clients.

- The core reason for the establishment of computerizing Online Doctor Appointment System is to enable the hospital administrators in a convenient, fair and timely manner. Therefore the IT used should support the core objective of the system if it is to remain relevant to the hospital. A lot still needs to be done in the IT department in order to make available technology effective. This may involve training of the hospital staffs on how to enter data in the right and relevant data in the system and the management to keep updating the hardware and

software requirements of the system. IT and computer systems need to be kept being upgraded as more and more IT facilities software are introduced in to days IT market. The researcher acknowledges the fact this system does not handle all patient doctor and hospital. The researcher therefore suggests that for further research, the following can be researched on. The most cost effective way of handling all Hospital Patient process.

Chapter 9: Bibliography

References Books

“Head First Servlets and JSP: Passing the Sun Certified Web Component Developer Exam “, by Bryan
Basham (Author), Kathy Sierra (Author), O'Reilly Media; Second Edition (April 1, 2008).

“Core Java™, Volume I--Fundamentals (8th Edition) “, by Cay S. Horstman, Prentice Hall; 8 Edition

Murach's Java Servlets and JSP, 3rd Edition

Matthew MacDonald, "HTML 5 - The Missing Manual", 2nd ed, 2014, O’Reilly.

David Sawyer McFarland, "CSS 3 - The Missing Manual", 3rd ed, 2013, O’Reilly.

ECMAScript (JavaScript) Specification: "Standard ECMA-262 ECMAScript Lang. Specification 5.1"

Website References

https://www.java.com/en

https://www.w3schools.com

https://www.javatpoint.com

https://www.geeksforgeeks.org

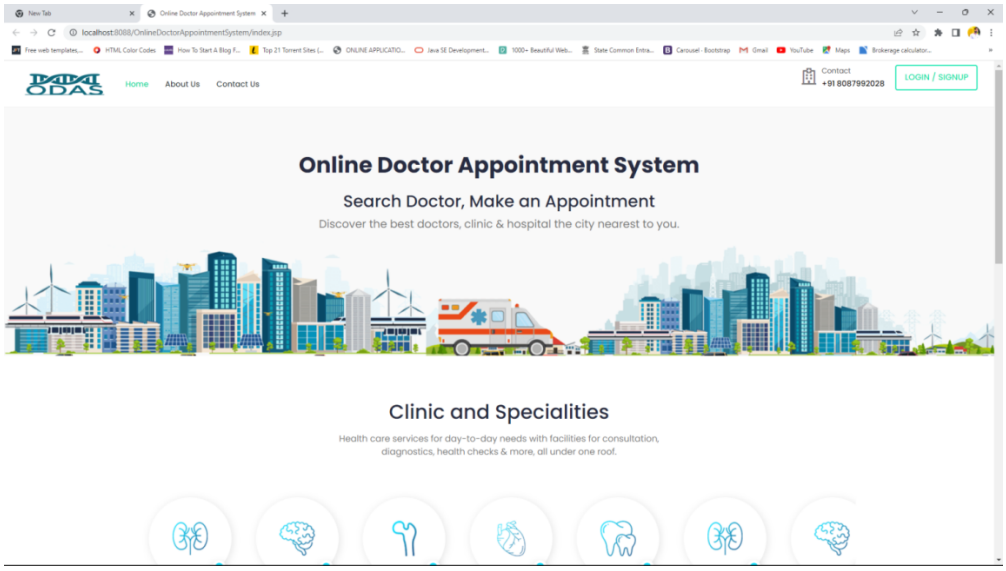
https://en.wikipedia.org/wiki

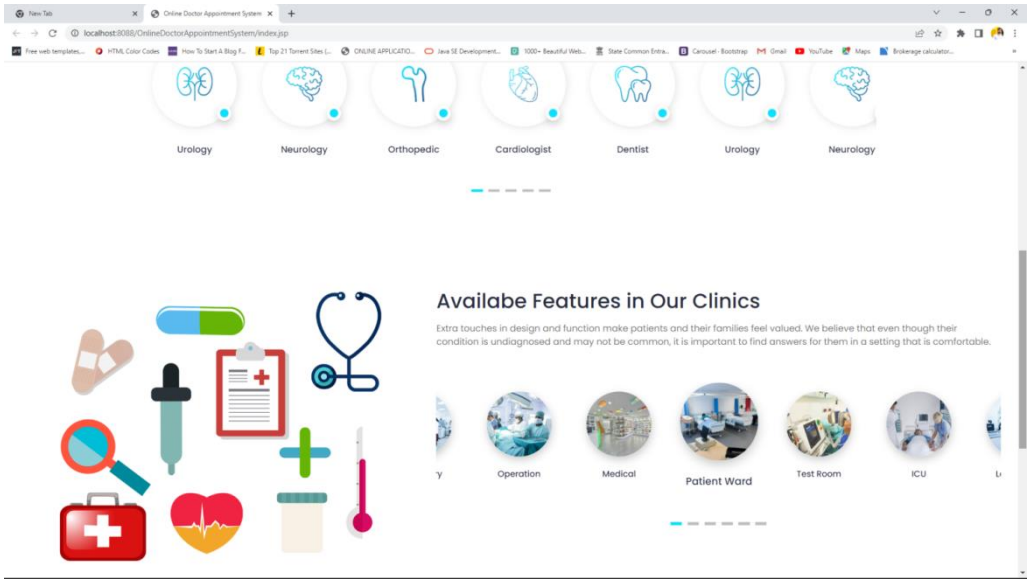
https://www.youtube.com/watch?v=Q33KBiDriJY&ab_channel=edureka%21

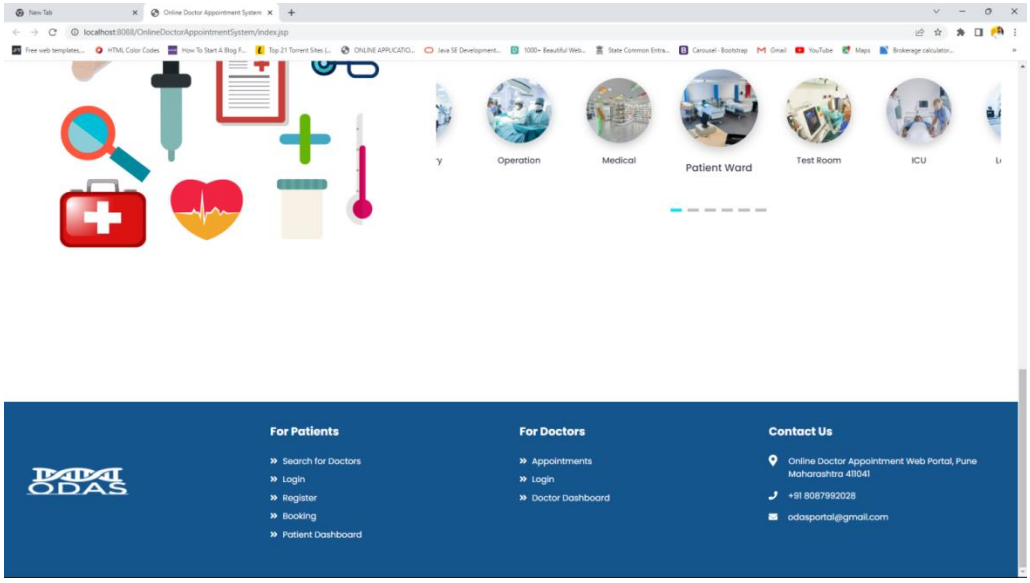
Chapter 10: User Manual

10.1 Input /Output Screens

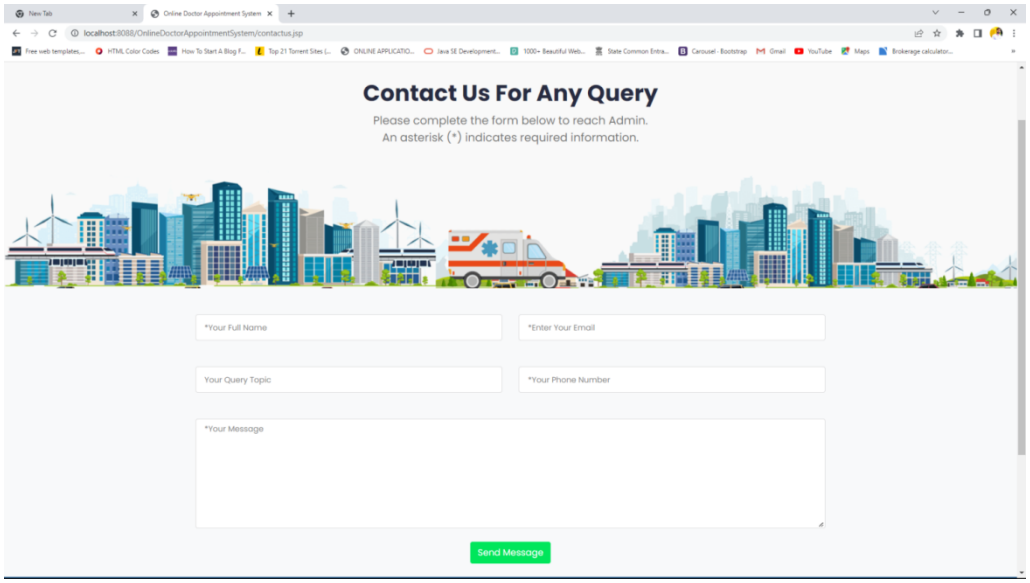
Home Page



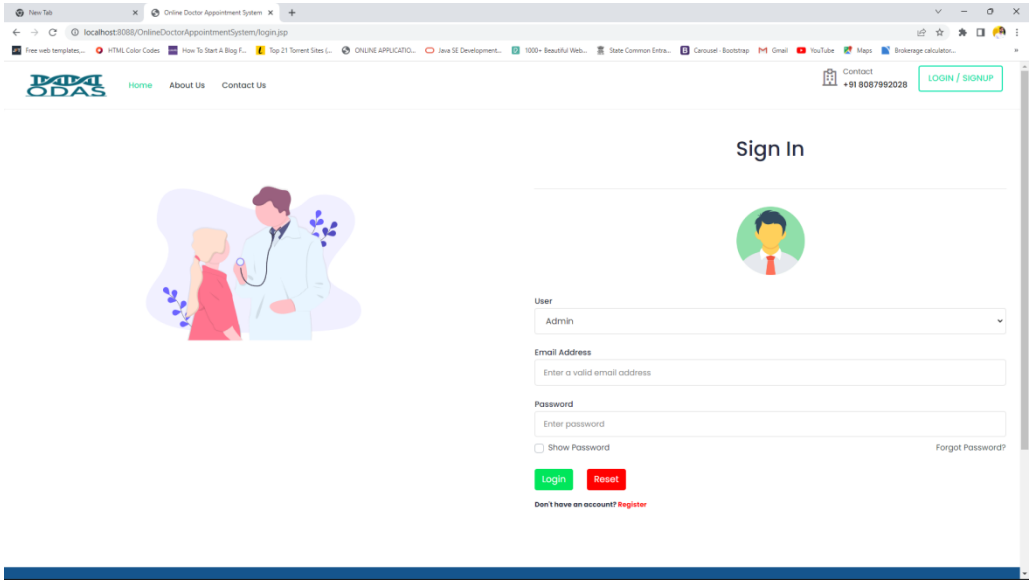




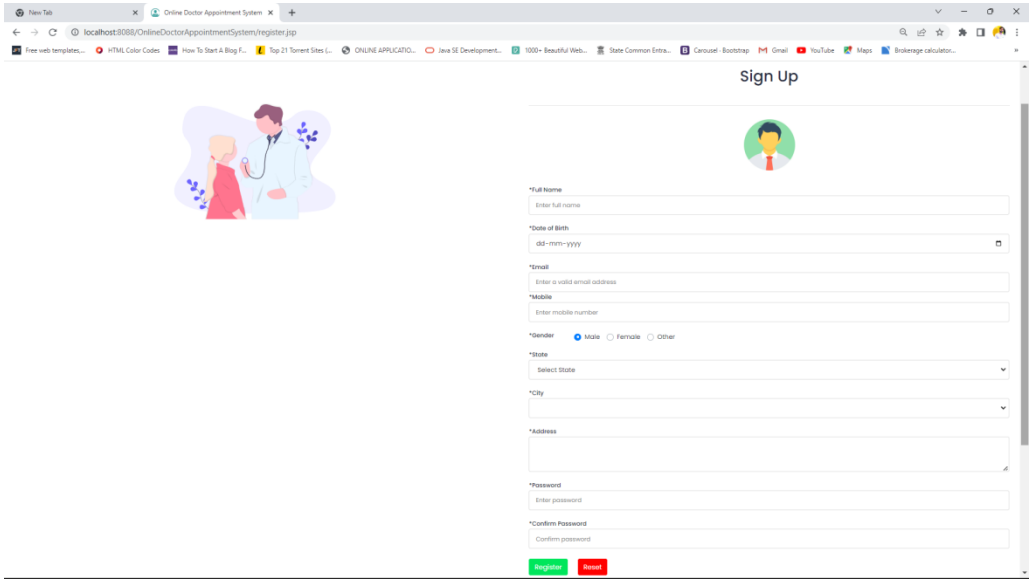
Contact us page



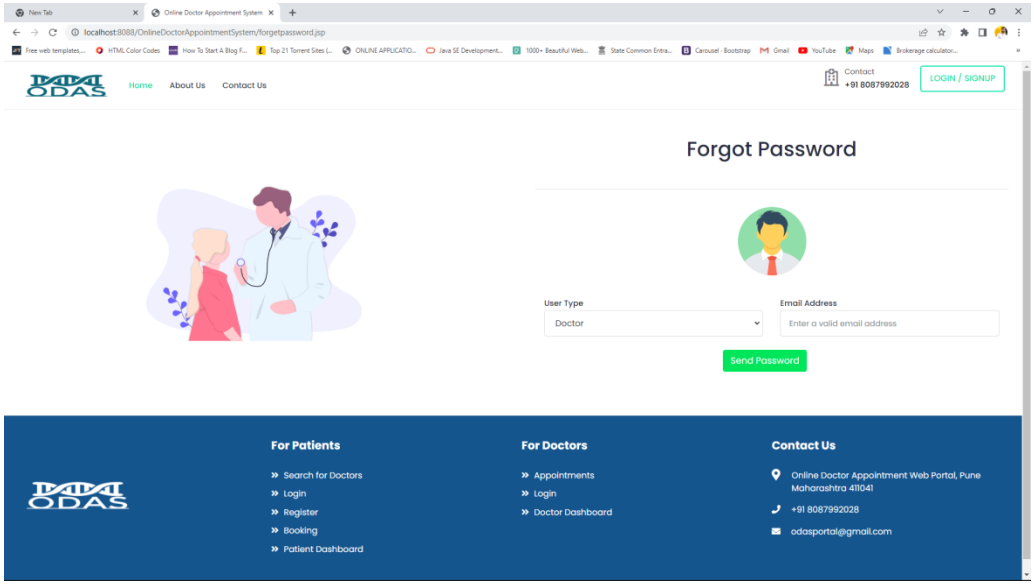
Login page



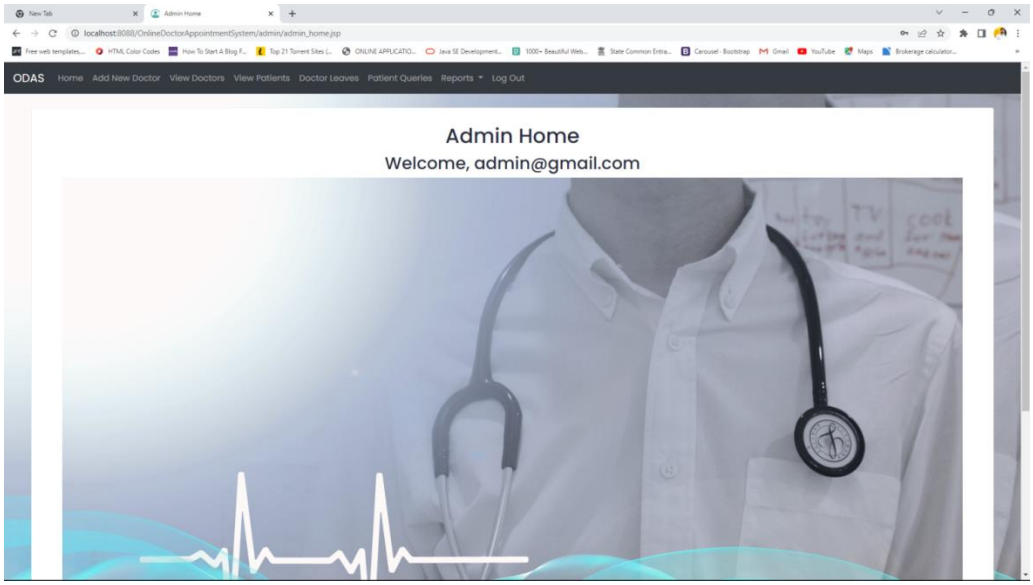
Registration page



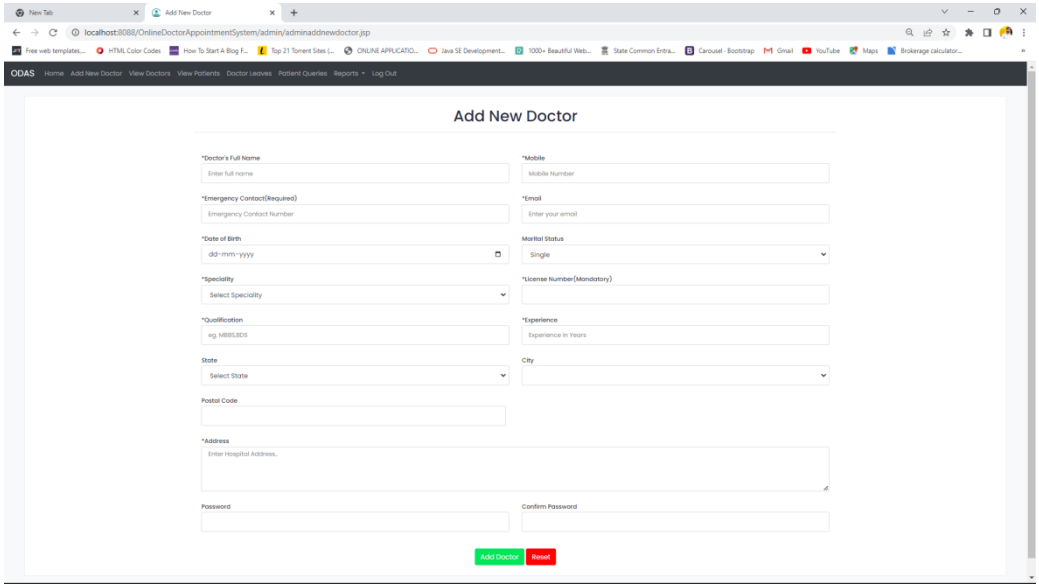
Forget password page



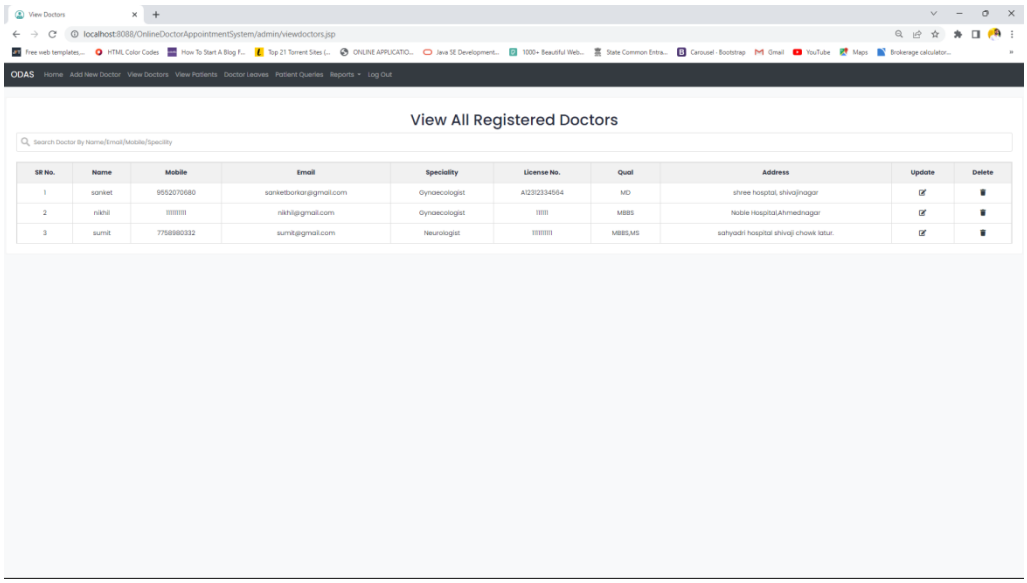
Admin home page



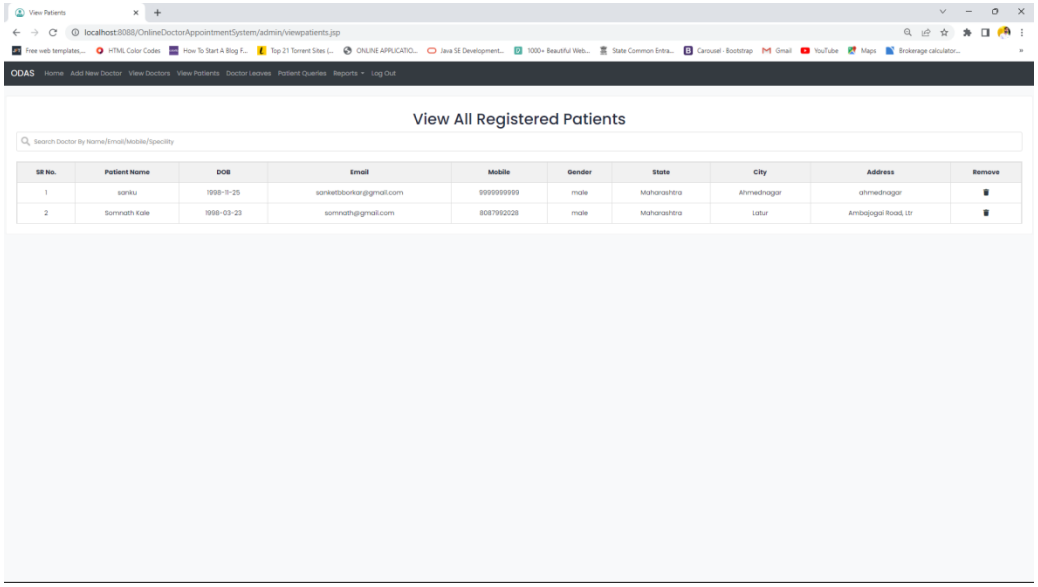
Add new doctor page



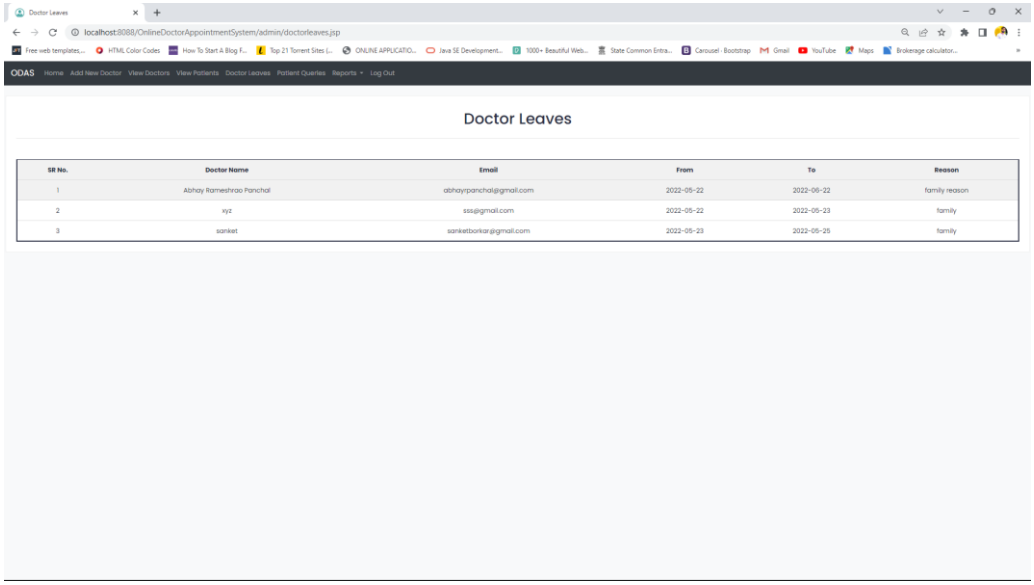
View Doctors page



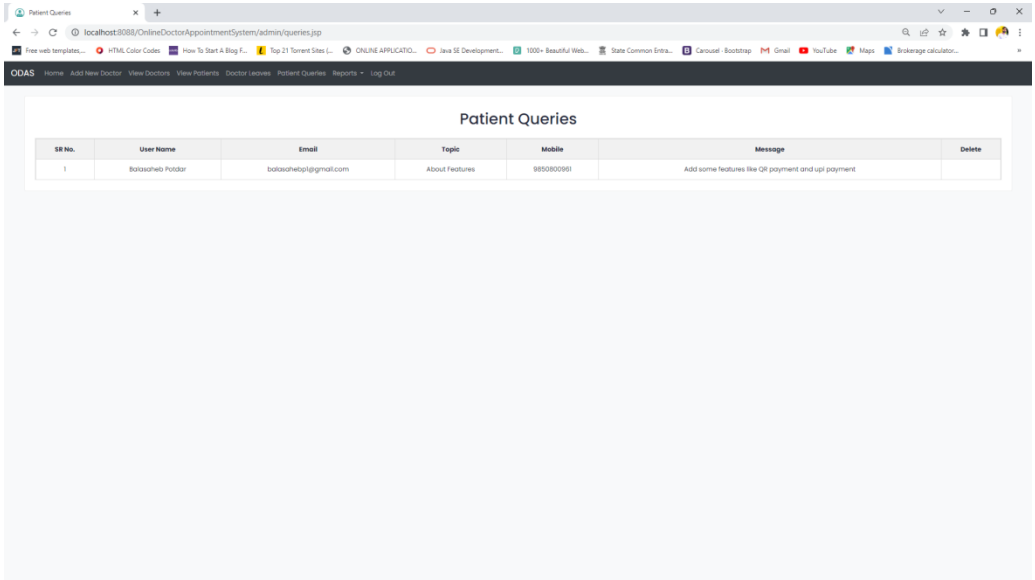
View patient page



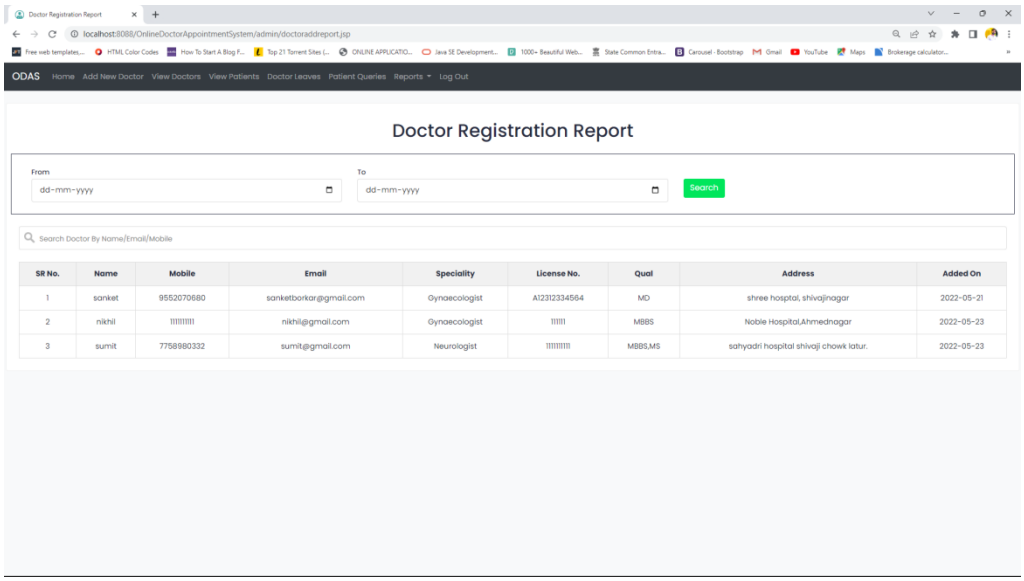
Doctor Leaves Page



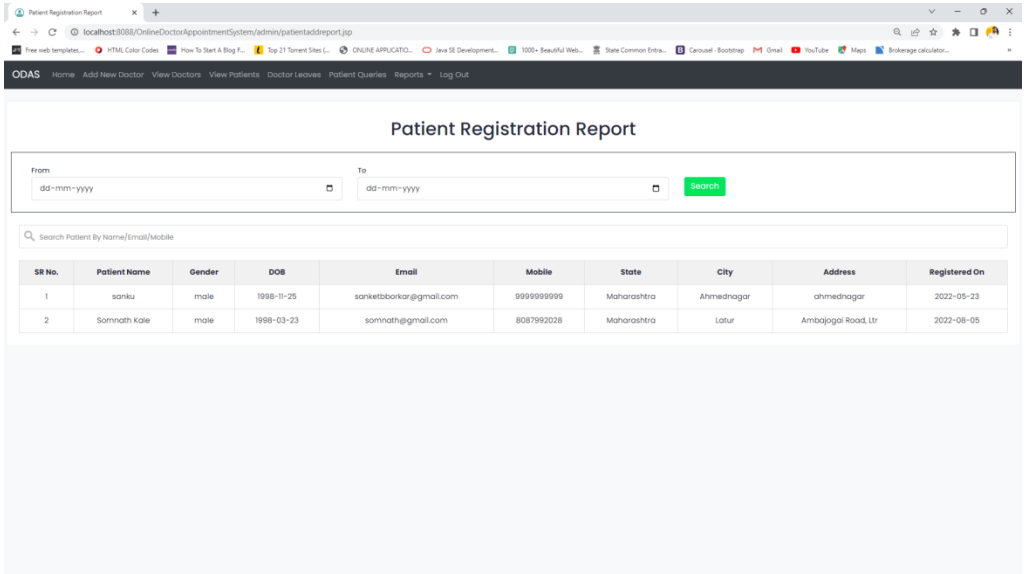
Patient Queries page



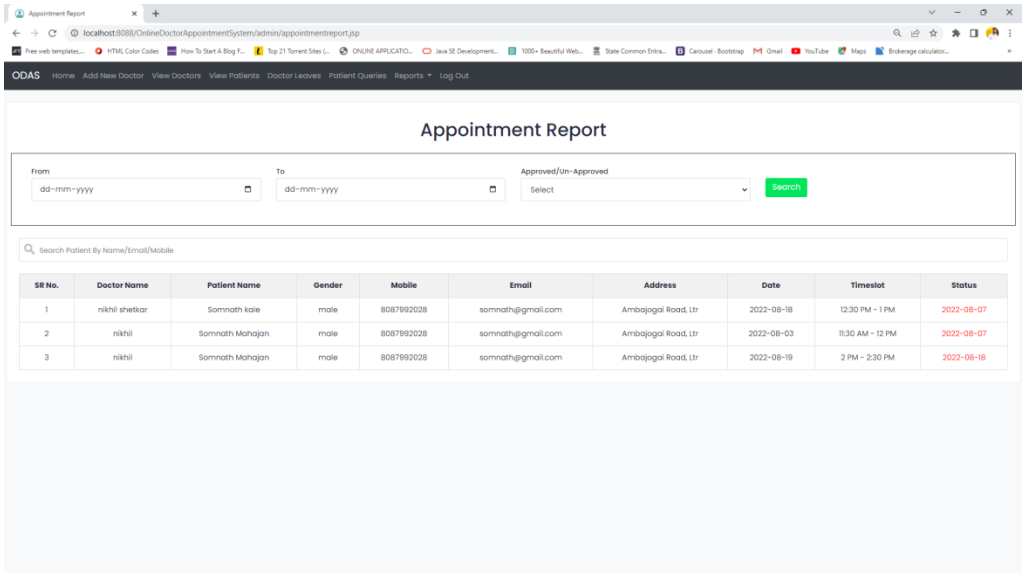
Doctor report page



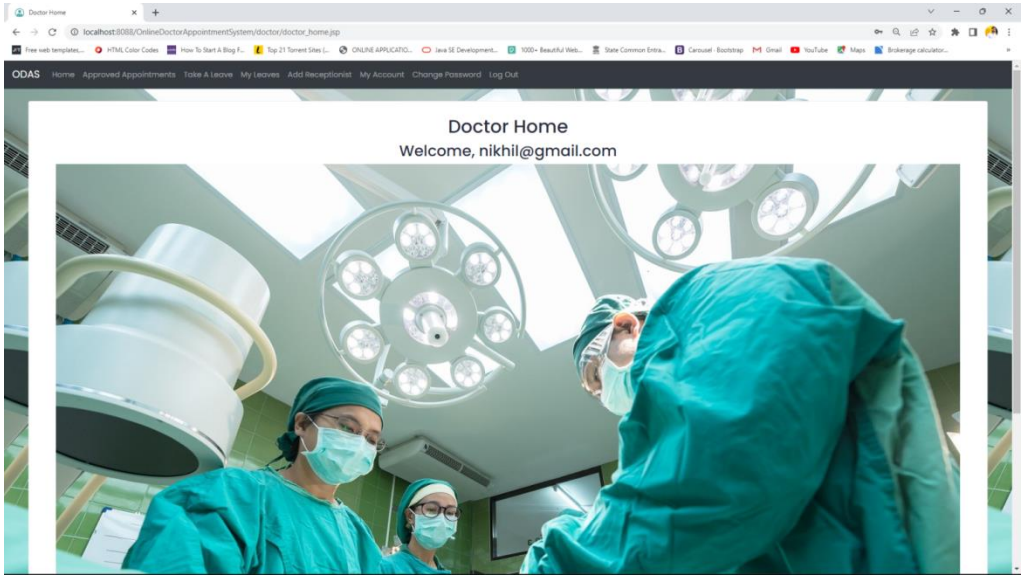
Patient repot page



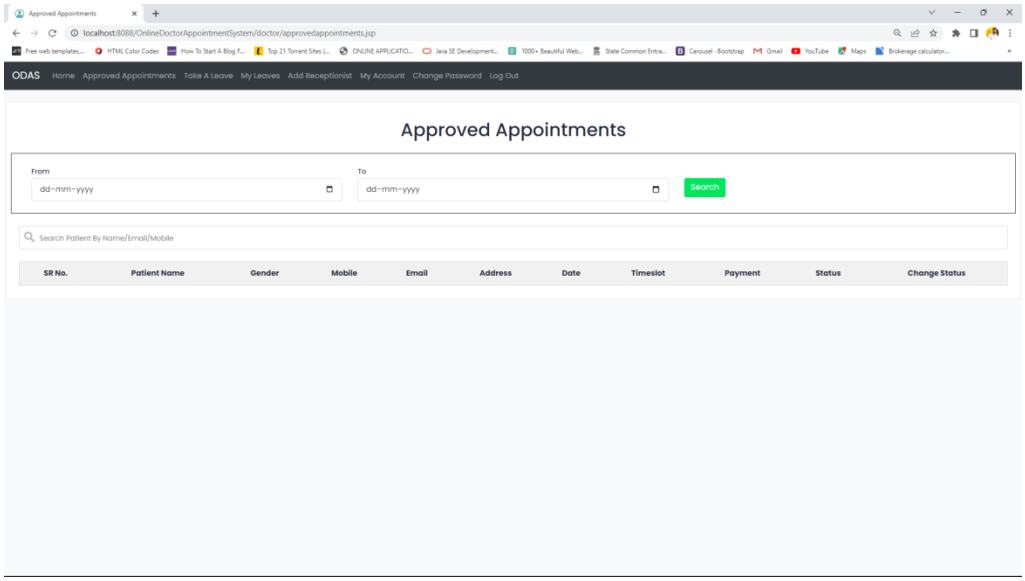
Appointment report page



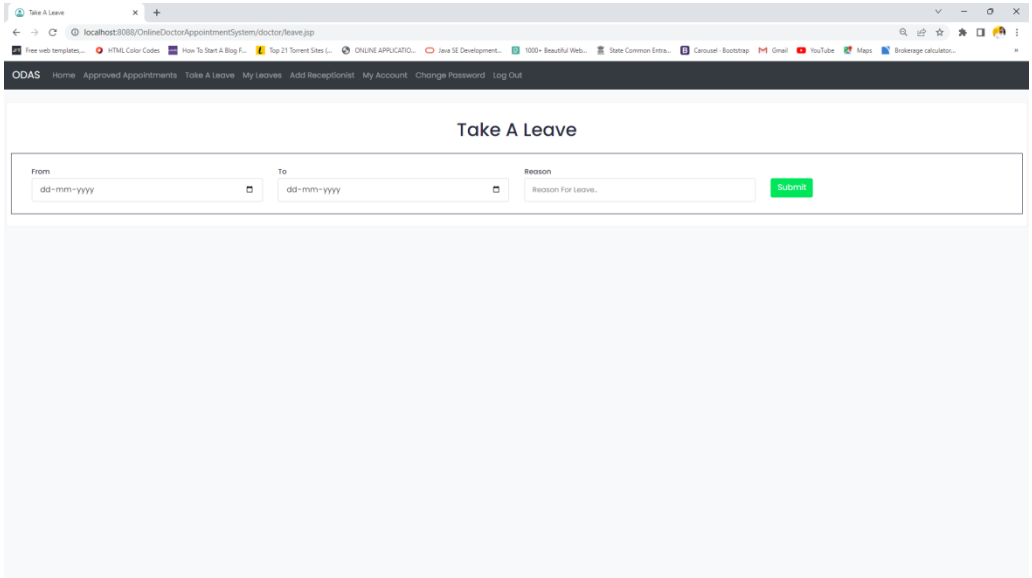
Doctor home page



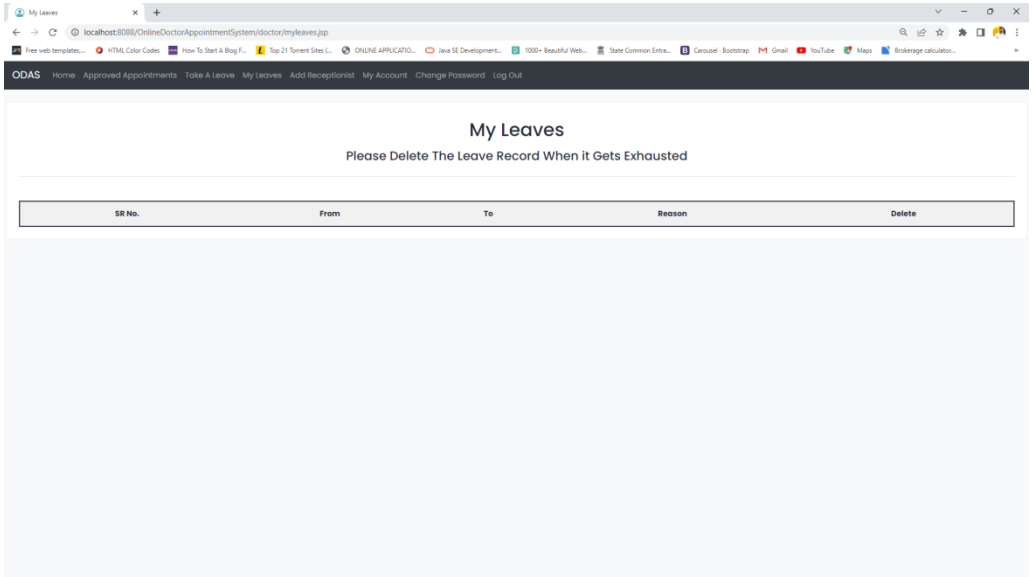
Approved Appointment page



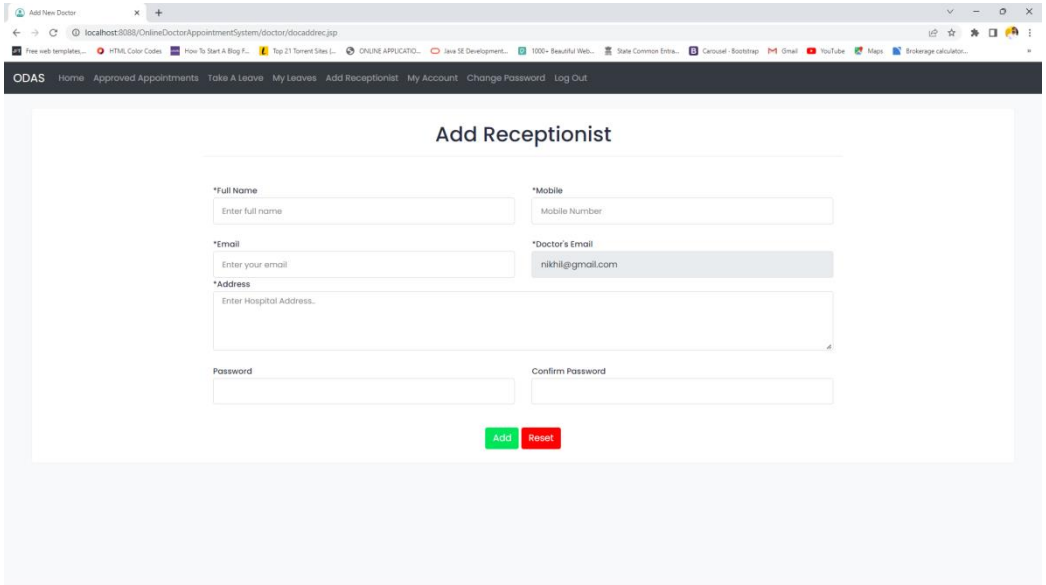
Doctor leave page



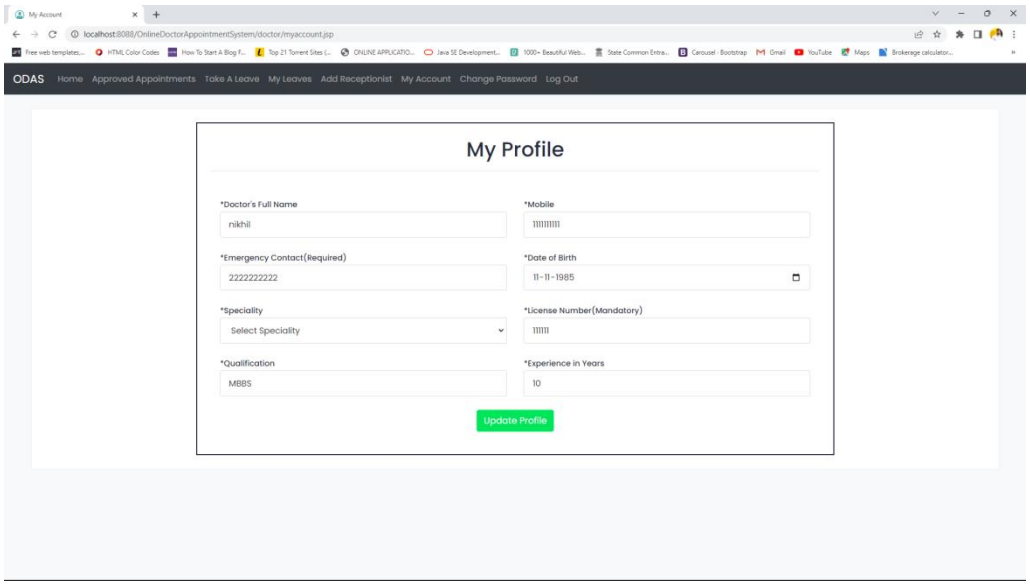
My leaves page



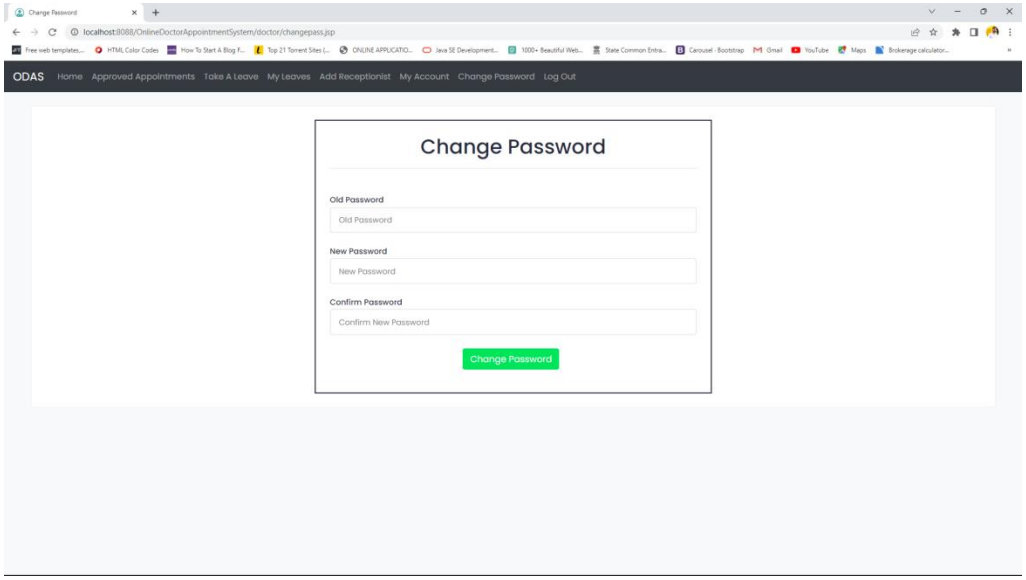
Add receptionist page



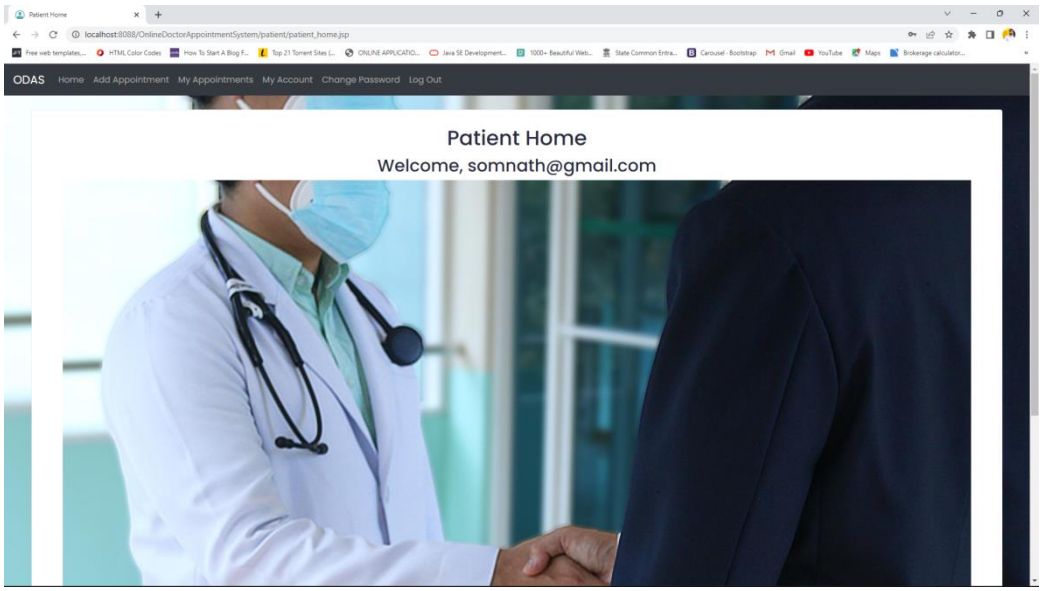
Doctor profile page



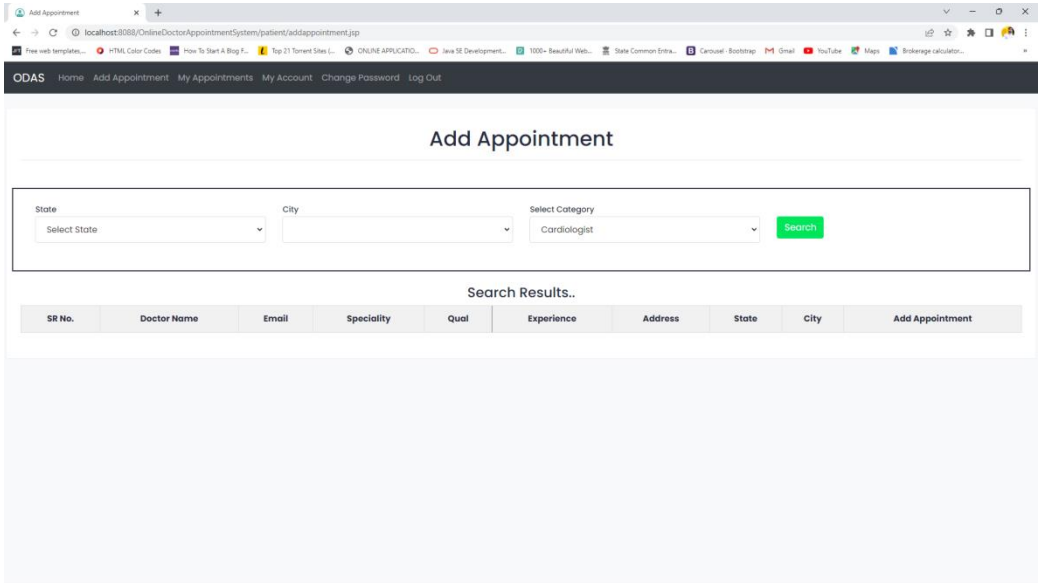
Change password page



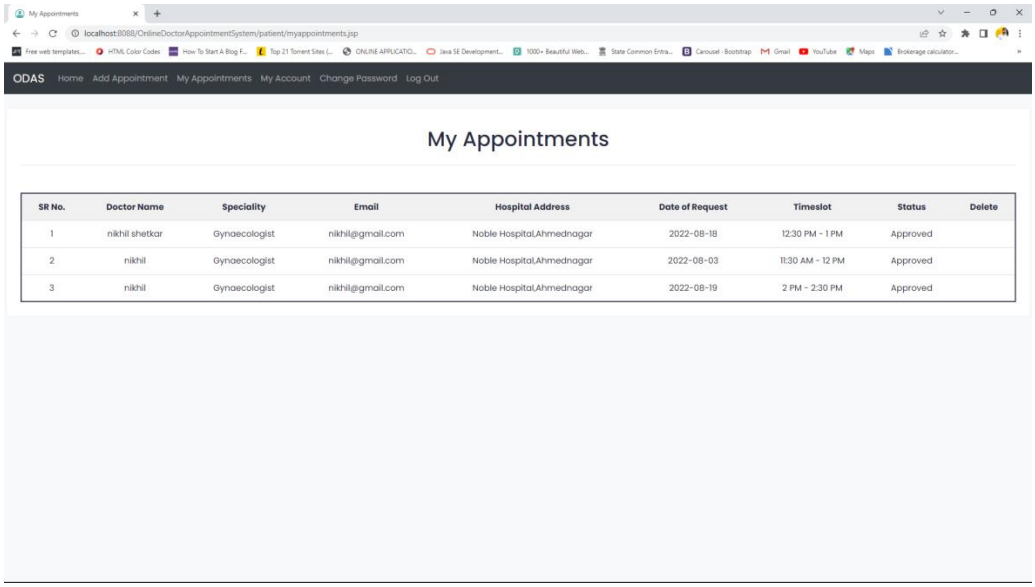
Patient home page



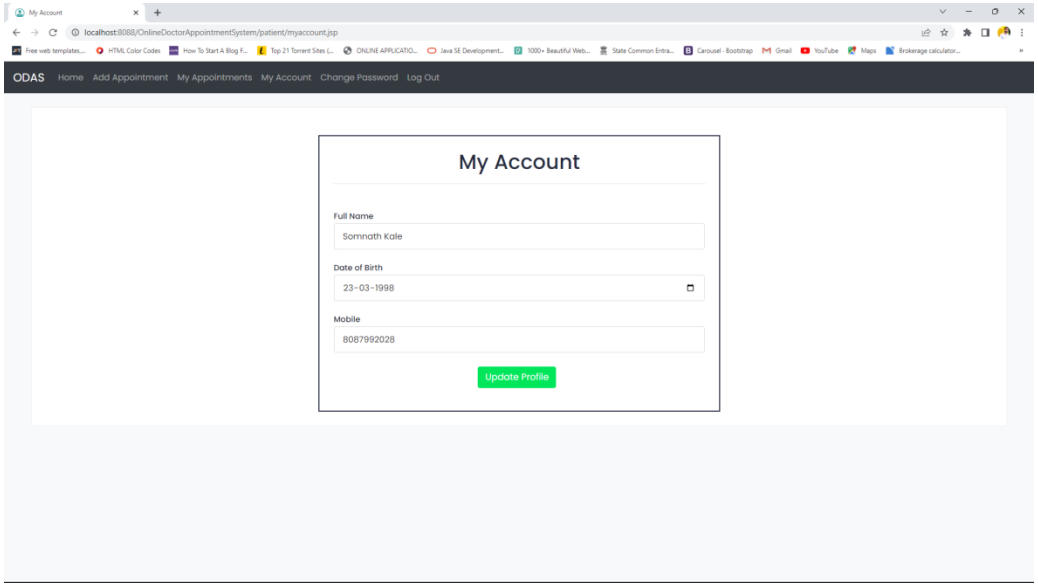
Add appointment page



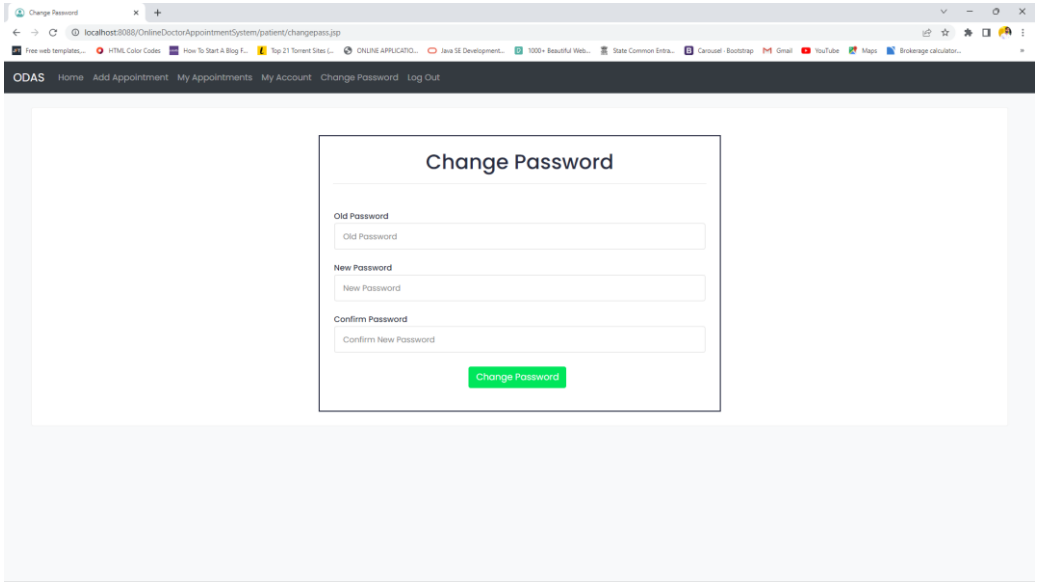
My appointment page



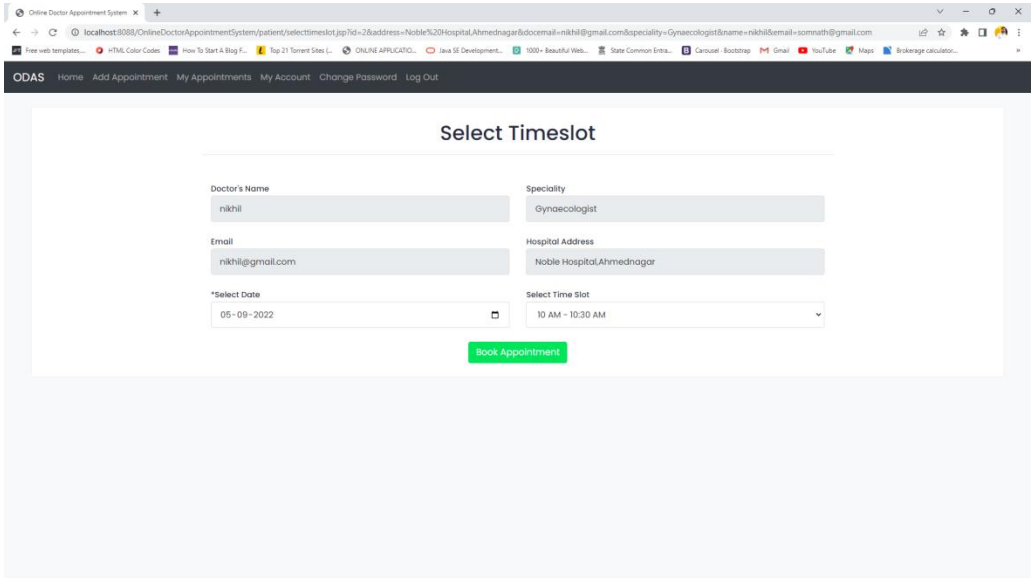
My account page



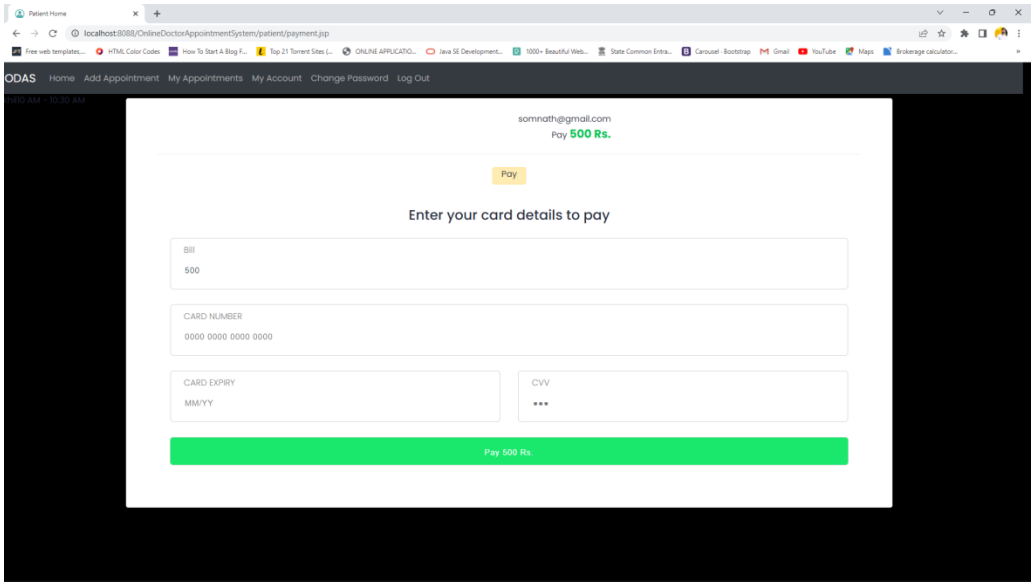
Change password page



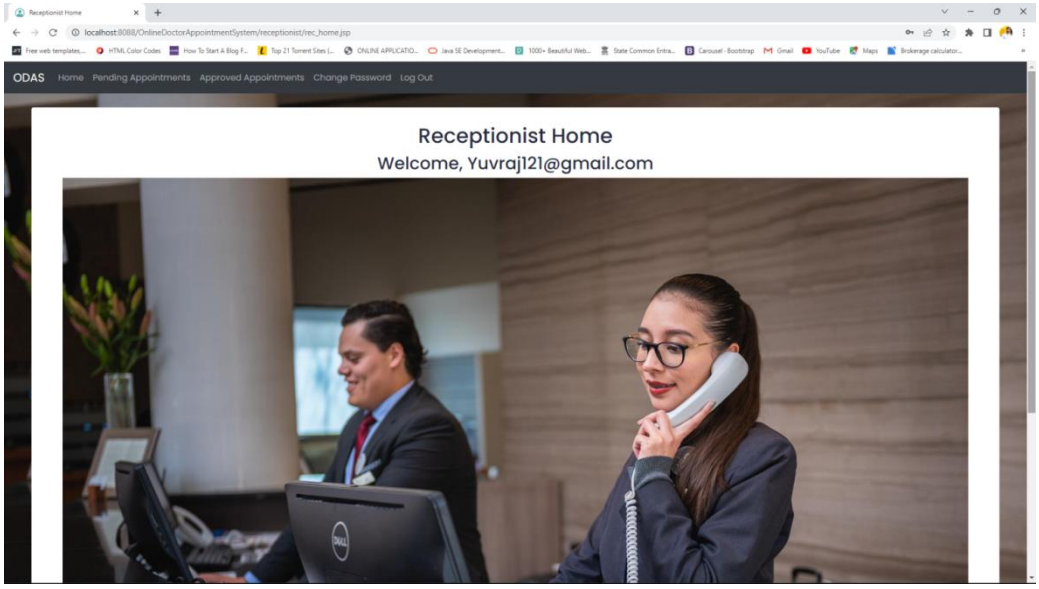
Timeslot page



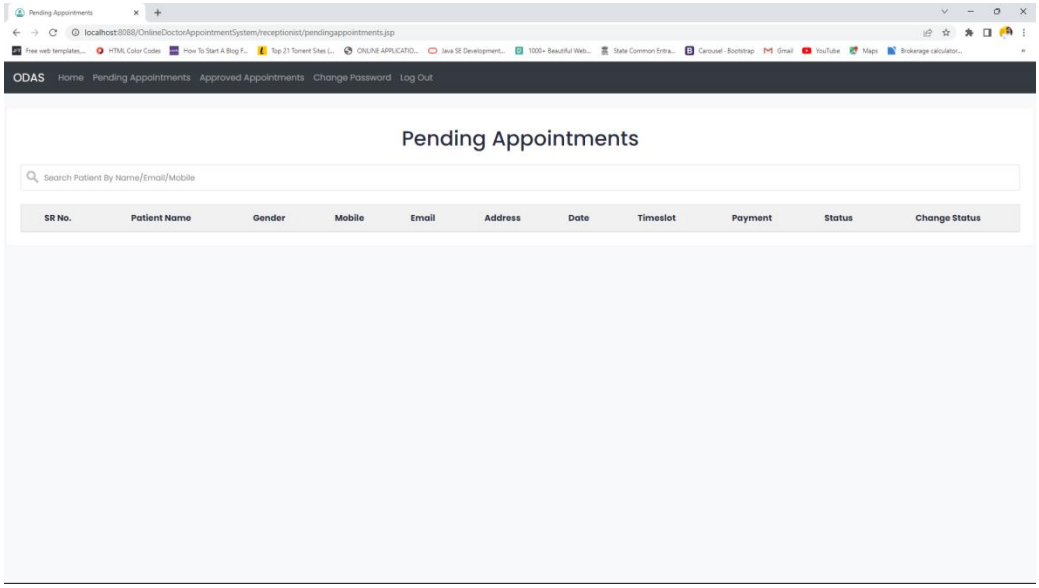
Payment Page



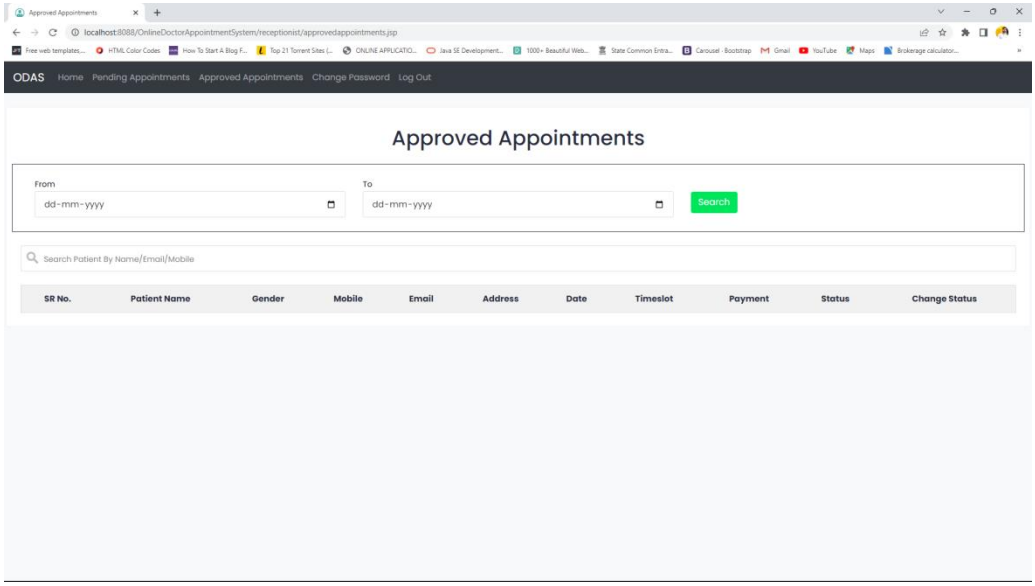
Receptionist home page



Pending appointment page

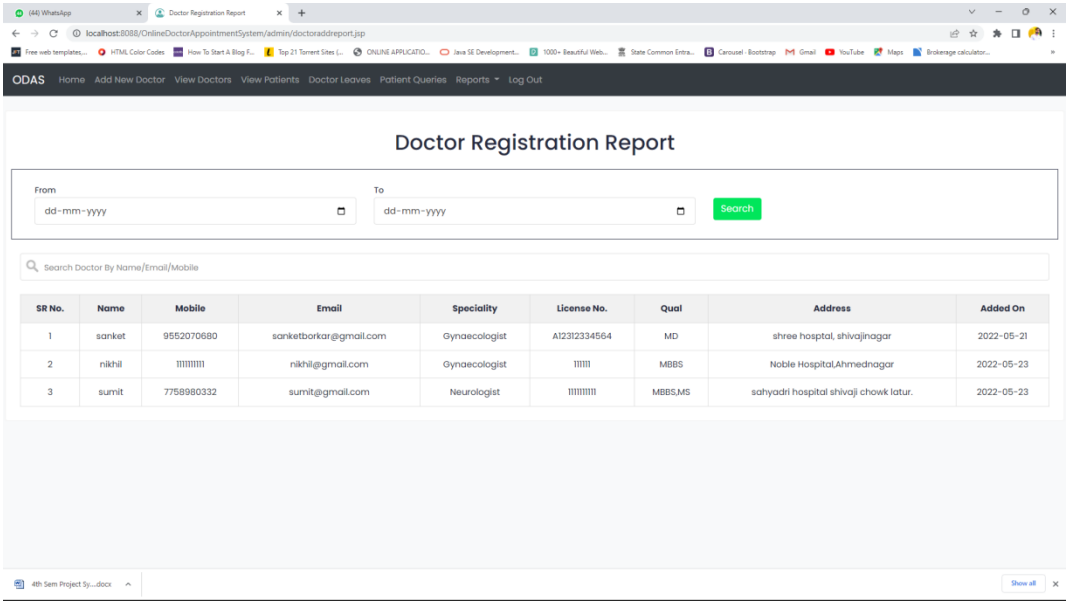


Approved appointment page



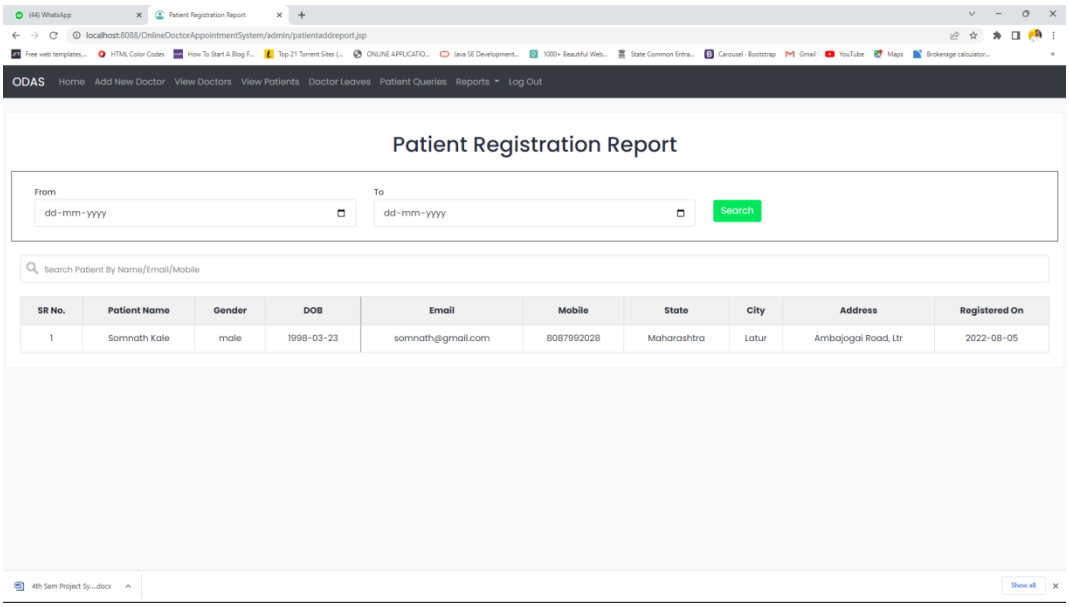
10.2 Data Reports

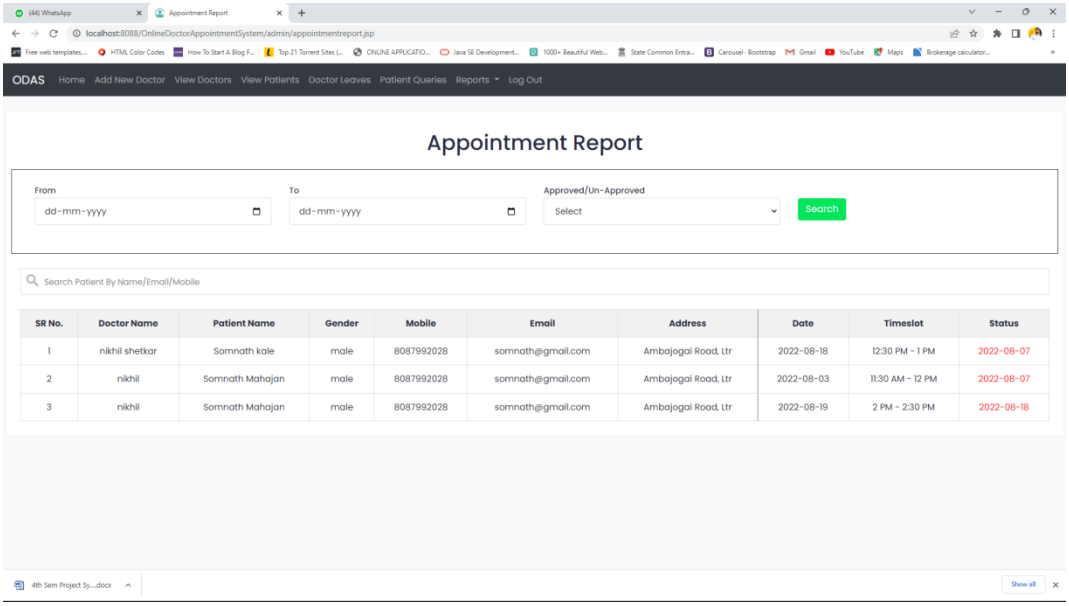
Doctor Registration Report



SR No.	Name	Mobile	Email	Speciality	License No.	Qual	Address	Added On
1	sanket	9552070680	sanketborkar@gmail.com	Gynaecologist	A12312334564	MD	shree hosptal, shivajinagar	2022-05-21
2	nikhil	1111111111	nikhil@gmail.com	Gynaecologist	111111	MBBS	Noble Hospital,Ahmednagar	2022-05-23
3	sumit	7758980332	sumit@gmail.com	Neurologist	1111111111	MBBS,MS	sahyadri hospital shivaji chowk latur.	2022-05-23

Patient Registration Report



SR No.	Patient Name	Gender	DOB	Email	Mobile	State	City	Address	Registered On
1	Somnath Kale	male	1998-03-23	somnath@gmail.com	8087992028	Maharashtra	Latur	Ambajogai Road, Ltr	2022-08-05

Appointment Report