Jabalpur Engineering College

DEPARTMENT OF

MASTER OF COMPUTER APPLICATION

(SESSION 2023-2024)

A Project Report On

HOSPITAL MANAGEMENT SYSTEM WEBSITE

HOD

DR. SAMAR UPADHYAY

DR. MAMTA LAMBERT

UNDER THE GUIDANCE OF

Mrs. ROSHNI VINODIA

Mrs. ANJALI SAHU

Mrs. MEGHA MALIK

Mrs. SWAPNILITA KASHYUP

Miss. SHIKHA SINGH NARELA

SUBMITTED BY –

LALIT PATIDAR – 0201CA221033

MCA 3rd SEMESTER

Jabalpur Engineering College

CERTIFICATE

This is to certified **LALIT PATIDAR** student of "**Master of Computer Application**" of "Jabalpur Engineering College, Jabalpur" has completed the project on "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**".

The project is being submitted by his in partial full film and for the award of the degree of "Master of Computer Application" at Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P) for the session 2023-24.

The report is up to the standard both in a respect of its contents and its literacy presentation for being referred to the examiner.

Dr. Samar Upadhyay
(HOD Department of MCA)

Jabalpur Engineering College

CERTIFICATE

This is to certified that **LALIT PATIDAR** student of "**Master of Computer Application**" of "**Jabalpur Engineering College**, Jabalpur" has completed the project on "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**".

They have done this project during the period of 2023-24 under the guidance and supervision of **Dr. Mamta Lambert**, **Professor,** Department MCA, Jabalpur Engineering College, Jabalpur.

They have completed the assigned project well within the time frame. They are sincere and hardworking and their conduct during this project is commendable.
I wish them all the best in their endeavors.

Dr. Mamta Lambert
(Professor MCA Department)
Jabalpur Engineering College

Jabalpur Engineering College

CERTIFICATE

This is to certified that **LALIT PATIDAR** student of "Master of Computer Application" of "Jabalpur Engineering College, Jabalpur" has completed the project on "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**".

The project is being submitted by his in partial full film and for the award of the degree of "Master of Computer Application" at Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P) for the session 2022-23.

The report is up to the standard both in a respect of its contents and its literacy presentation for being referred to the examiner.

INTERNAL EXAMINER

Date :

EXTERNAL EXAMINER

Date :

Jabalpur Engineering College

DECLARATION

I hereby declare that the project report entitled "STUDENT MANAGEMETN SYSTEM WEBSITE" which is being submitted in partial fulfillment for the award of degree of "Master of Computer Application" to "Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)" is an authentic record of my own work carried out under the guidance of Ms. Roshni Vinodia, Ms. Anjali Sahu, Ms. Swapnilita Kashyup, Miss. Shikha Singh Narela, Ms. Megha Malik Professors of Department of MCA, Jabalpur Engineering College, Jabalpur (M.P).

Project Submitted By

LALIT PATIDAR

(0201CA221033)

MCA 3rd SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **Dr. Samar Upadhyay** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3rd SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **Dr. MAMTA LAMBERT** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3rd SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **MRS. ROSHNI VINODIA** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3rd SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **MRS. ANJALI SAHU** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3rd SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **MRS. MEGHA MALIK** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3RD SEMESTER

SESSION 2023-24

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my project guide **MRS. SWAPINILITA KASHYUP** for the guidance, inspiration and constructive suggestions that helped me in the completion of my project entitled "**HOSPITAL MANAGEMENT SYSTEM WEBSITE**" which helped me in doing a lot of research and I came to know about so many new things.

I would also like to thank the college staff for their support and my friends who helped me a lot.

WITH REGARDS

LALIT PATIDAR

(0201CA221033)

MCA 3RD SEMESTER

SESSION 2023-24

Index

Sn.	TOPIC
1.	Preface
2.	Introduction
3.	Scope and Objective of the System
4.	Feasibility Study
5.	System Requirement 1. Hardware 2. Software
6.	Technologies Used 1. front-end 2. Back-end
7.	Goals
8.	Existing system
9.	Proposed system
10.	Modules
11.	Implementation of DFD (data flow diagram)
12.	ER Diagram - SOL Database
13.	SOL Database 0BJECT MODEL [Opted]
14.	Refrence

PREFACE

This preface introduces a transformative solution designed to streamline hospital operations and elevate patient care. This system amalgamates advanced technology with seamless administrative, clinical, and financial processes, redefining how hospitals function. It efficiently manages patient data, appointments, electronic health records, billing, and inventory. By automating tasks, it enhances staff efficiency, reduces errors, and optimizes resource allocation. The result is an environment where healthcare professionals can dedicate more time to patient well- being, leading to improved outcomes. With its integration capabilities and data security measures, the system ensures compliance with standards and regulations. This preface invites you to witness the convergence of healthcare and technology, where the Hospital Management System paves the way for operational excellence, cost-effectiveness, and exceptional patient-centric care.

Lalit Patidar 16/08/2023

INTRODUCTION

The introduction of the Hospital Management System marks a pivotal moment in this evolution, reshaping the landscape of healthcare administration and patient care.

The Hospital Management System is a sophisticated and comprehensive software solution that integrates a multitude of critical functions within healthcare institutions. This system transcends traditional administrative processes by automating tasks like patient registration, appointment scheduling, medical record management, billing, and inventory control. Its core purpose is to enhance operational efficiency, minimize errors, and empower healthcare professionals to focus on what truly matters – delivering exceptional patient care.

By centralizing diverse aspects of hospital management, this system facilitates seamless communication between departments, streamlines workflows, and enables data-driven decision-making through robust reporting and analytics. Additionally, it ensures compliance with privacy and security regulations, safeguarding sensitive patient information.

The potential benefits of the Hospital Management System are boundless. From reducing waiting times and improving resource allocation to optimizing financial processes and enhancing patient satisfaction, this system heralds a new era of healthcare management excellence.

As we embark on this transformative journey, we invite healthcare institutions, professionals, and stakeholders to embrace the Hospital Management System and witness the unfolding of a future where technology and healthcare converge to create a more efficient, patient-centric, and sustainable healthcare ecosystem.

SCOPE AND OBJECTIVE OF THE SYSTEM

Scope:

The scope of a Hospital Management System (HMS) is extensive, encompassing various administrative, clinical, and financial aspects of healthcare operations. The system aims to digitize and automate multiple processes within a hospital or healthcare facility, leading to improved efficiency, patient care, and overall management. The key components within the scope of an HMS include:

Patient Management:

Capturing, storing, and managing patient information, medical history, and treatment plans.

Appointment Scheduling:

Enabling patients to book appointments online and optimizing appointment slots for healthcare professionals.

Electronic Health Records (EHR): Creating and maintaining comprehensive digital health records for patients, accessible by authorized personnel.

Billing and Invoicing:

Automating billing processes, generating accurate invoices, and managing financial transactions.

Inventory Control:

Managing and tracking hospital inventory, including medical supplies and equipment, to ensure availability and prevent shortages.

Staff Management:

Efficiently scheduling and managing staff shifts, tracking performance, and handling payroll.

Reporting and Analytics:

Generating reports and analytics to aid decision-making, resource allocation, and process optimization.

Interoperability:

Facilitating data exchange between different departments and external systems for seamless collaboration.

Objectives:

The primary objectives of implementing a Hospital Management System are:

Efficiency Enhancement:

Streamlining administrative processes, reducing manual work, and improving overall operational efficiency.

Patient Care Improvement:

Enabling healthcare professionals to focus more on patient care by automating routine tasks.

Error Reduction:

Minimizing human errors in tasks such as billing, record-keeping, and inventory management.

Data Accessibility:

Providing authorized personnel with secure and quick access to patient records and medical information.

Resource Optimization:

Ensuring optimal utilization of hospital resources, including staff, equipment, and supplies.

Cost Reduction:

Cutting down on paperwork, streamlining processes, and optimizing resource allocation to reduce costs.

Compliance and Security:

Ensuring compliance with healthcare regulations and maintaining the security of patient data.

Patient Satisfaction:

Enhancing patient experience by reducing waiting times, improving appointment scheduling, and increasing the accuracy of medical records.

Decision Support:

Providing management with data-driven insights to make informed decisions about resource allocation and strategic planning.

FEASIBILITY STUDY

A feasibility study for a hospital management system involves assessing the viability and practicality of developing and implementing such a system. This study helps determine whether the project is worth pursuing, taking into account technical, economic, operational, and scheduling factors. Here's a step-by-step guide on how to conduct a feasibility study for a hospital management system:

Define the Scope and Objectives:

Clearly outline the goals and objectives of the hospital management system. Identify the specific functionalities you want the system to have, such as patient registration, appointment scheduling, medical records management, billing, and more.

Technical Feasibility:

Assess whether the technology required for the system is available and feasible. Consider factors such as hardware, software, networking, and any potential integration with existing systems. Evaluate if the development team has the required expertise to build the system.

Operational Feasibility:

Determine if the proposed system aligns with the hospital's operations. Identify potential challenges related to workflow, user training, and change management. Analyze how the system will impact daily tasks and whether staff will be able to adapt to it.

Economic Feasibility:

Conduct a cost-benefit analysis to determine whether the benefits of the system outweigh the costs. Consider development costs, hardware and software costs, ongoing maintenance costs, and potential savings or revenue generated through increased efficiency and improved patient care.

Financial Feasibility:

Evaluate the hospital's financial capability to fund the project. Determine whether the hospital can allocate the necessary resources, including budget, personnel, and time, for the system's development and implementation.

Legal and Regulatory Feasibility:

Research and assess any legal and regulatory requirements that the hospital management system must comply with, such as patient data privacy regulations (e.g., HIPAA) and medical standards. Ensure that the system design and functionality adhere to these regulations.

Scheduling Feasibility:

Develop a realistic timeline for the project, taking into account the various stages of development, testing, deployment, and training. Consider potential delays and dependencies that might affect the project schedule.

Risk Assessment:

Identify potential risks and challenges that could impact the success of the project. These might include technical issues, user resistance, budget overruns, or changes in requirements. Develop strategies to mitigate or address these risks.

SYSTEM REQUIREMENTS

H/W Required: →

- **Processor type** - Intel Pentium quad Core or above
- **Ram** -4 GB (at least)
- **Graphic card type** - If available (optional)

S/W Required: →

- Operating System- Windows/ Mac OS, Linux, etc.
- Web browser - Microsoft Edge, Mozilla Firefox, Google Chrome Etc., or any latest version of browser.

TECHNOLOGIES USED

Front End: →

1. **HTML**→

   Html or HyperText Markup Language, is a standard language used to create and structure content on the World Wide Web. It forms the foundation of web pages, allowing you to define the structure, layout, and elements that make up a webpage. HTML is not a programming language; it's a markup language used to describe the structure of content.

2. **CSS** →

   Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation of a document written in HTML or XML. CSS3 is the latest major revision of CSS, building upon the foundation of CSS2 and bringing many new features and enhancements to web styling.

3. **JAVASCRIPT** →

   JavaScript is a widely-used programming language that adds interactivity, dynamic behavior, and functionality to websites and web applications. It allows you to create dynamic content that can respond to user actions, manipulate the DOM (Document Object Model), and interact with servers.

4. **BOOTSTRAP** →

   Bootstrap is a popular open-source front-end framework that simplifies and speeds up the process of building responsive and visually appealing websites and web applications. Created by Twitter, Bootstrap provides a collection of pre-designed HTML, CSS, and JavaScript components that you can easily customize to create a modern and consistent user interface.

5. **JSP** →

JavaServer Pages (JSP) is a server-side technology that facilitates the creation of dynamic web pages using Java. It combines HTML templates with embedded Java code, enabling the generation of dynamic content. When a client requests a JSP page, the server processes it, executing embedded Java code and generating HTML to be sent back to the client's browser. This allows developers to build interactive web applications by seamlessly integrating Java's power and flexibility with web presentation. JSP promotes code reusability, scalability, and separation of concerns, making it a key tool for building robust and feature-rich web applications.

Back-end

1. JAVA →

   Java is a widely used, versatile, object-oriented programming language renowned for its platform independence and robustness. Developed by Sun Microsystems (now owned by Oracle), Java offers a write-once-run-anywhere capability, allowing code to be executed on various platforms. It emphasizes clean syntax, automatic memory management, and a rich standard library, simplifying software development. Java's core strength lies in its ability to build diverse applications, from desktop software to mobile apps and enterprise-level systems.

2. JDBC →

   Java Database Connectivity (JDBC) is an API that enables Java applications to interact with databases. It provides a

standardized interface for accessing relational database systems. By using JDBC, developers can establish connections, execute SQL queries, and retrieve results from databases. This facilitates the storage and retrieval of data within Java applications. JDBC's versatility allows applications to be database-independent, making it a crucial tool for building dynamic and data-driven software.

3. SERVLET →

   A servlet is a Java-based server-side component that extends the capabilities of a web server, processing incoming HTTP requests and generating dynamic responses. It forms a crucial part of Java EE (Enterprise Edition) or Jakarta EE web applications. Servlets handle tasks like user authentication, data processing, and content generation, interacting with web clients, databases, and other resources.

4. My Sql →

   MySQL is an open-source relational database management system. It stores and manages structured data, offering a reliable platform for data storage and retrieval. Widely used in web applications, MySQL supports various programming languages and provides features like data security, transactions, and indexing. Its versatility, scalability, and community support make

it a popular choice for powering dynamic and data-intensive applications

5. **.SQL →**

   Structured Query Language (SQL) is a domain-specific language used for managing and manipulating relational databases. It serves as a standard interface for interacting with databases, allowing you to create, modify, retrieve, and manipulate data.

GOALS

The goals of a hospital management system are to streamline and improve various aspects of hospital operations, patient care, and administrative tasks. These goals include:

Efficient Patient Care:

Enhance patient care by providing timely access to accurate medical records, treatment plans, and medications. Enable faster patient check-in, appointment scheduling, and care coordination.

Workflow Automation:

Automate routine tasks like appointment scheduling, billing, and prescription management to improve efficiency and reduce errors in administrative processes.

Data Centralization:

Centralize patient data, medical records, and administrative information in a secure and easily accessible digital format to eliminate paper-based systems and improve data accuracy.

Effective Resource Management:

Optimize resource allocation, including staff, equipment, and facilities, to ensure efficient use of hospital resources and minimize waste.

Improved Communication:

Facilitate better communication among healthcare providers, administrative staff, and patients. Provide tools for secure messaging, alerts, and notifications.

EXISTING SYSTEM

In the current student management system is a basic file system that is, Patient records, medical histories, test results, and other healthcare information were typically stored in paper files.

Paper-Based Records: Patient records, medical histories, test results, and other healthcare information were typically stored in paper files. These files were physically stored in filing cabinets, making data retrieval time-consuming and prone to errors.

Appointment Scheduling: Appointment scheduling was often done through manual methods, such as appointment books or receptionists writing down appointments in notebooks. This could lead to scheduling conflicts and inefficiencies.

Billing and Documentation: Billing processes were manual and required paper-based invoicing. Administrative staff manually filled out forms for insurance claims and billing statements, which increased the risk of errors.

Communication: Communication between different departments, healthcare providers, and administrative staff relied on in-person interactions, telephone calls, memos, and interoffice mail. This could sometimes result in delays and miscommunication.

Prescription Management: Prescription orders were handwritten, and patients had to physically visit pharmacies to get their medications. Doctors communicated treatment plans through paper prescriptions.

Patient Tracking: Tracking patient movements within the hospital, such as admissions, discharges, and transfers, was done manually through paper-based logbooks and forms.

PROPOSED SYSTEM

Hospital management systems offer a wide range of features to streamline operations, enhance patient care, and improve administrative efficiency within healthcare institutions. Here are some key features commonly found in modern hospital management systems:

Patient Registration and Records:
Create and manage patient profiles with demographics, medical history, and contact information.
Maintain electronic health records (EHR) for easy access to patient medical information.
Appointment Scheduling:
Enable patients to book appointments online or through the system.
Manage doctor availability and scheduling for efficient appointment allocation.
Billing and Invoicing:
Generate and manage bills for medical services and treatments.
Process insurance claims and integrate with insurance providers for billing and payments.
Prescription Management:
Electronically create and manage prescriptions for patients.
Integrate with pharmacies for prescription processing.
Laboratory and Test Integration:
Order and track lab tests, imaging, and other diagnostic procedures.
Store and retrieve test results and images within the system.
Medical Inventory Management:
Manage medical supplies, equipment, and medications.
Monitor stock levels and generate reorder alerts.

MODULES

Hospital management systems consist of various interconnected modules, each designed to address specific aspects of healthcare operations. These modules collectively contribute to efficient patient care, streamlined administration, and enhanced overall hospital management. While the specific modules can vary based on the system's design and the hospital's needs, here are some common modules found in hospital management systems:

Patient Registration and Admission:

Capture patient demographics, contact details, and medical history.

Facilitate patient admissions, transfers, and discharges.

Appointment and Scheduling:

Enable patients to book appointments online.

Manage doctor schedules and allocate appointments.

Electronic Health Records (EHR):

Store and manage patient medical records, including diagnoses, treatment plans, medications, and test results.

Billing and Invoicing:

Generate bills for treatments and services rendered.

Manage insurance claims and payments.

Prescription Management:

Create and manage electronic prescriptions.

Maintain medication history and drug interactions.

Laboratory and Imaging:

Order and track lab tests, radiology, and other diagnostic procedures.

Store and retrieve test results and images.

Inventory Management:

Manage medical supplies, equipment, and medications.

Monitor stock levels and automate reordering.

Pharmacy Management:

Dispense medications and manage pharmacy inventory.

Handle prescription processing and patient medication history.

Patient Portal:

Provide patients with online access to medical records, appointments, and communication with healthcare providers.

Doctor and Staff Management:

Manage doctor and staff profiles, schedules, and leave requests.

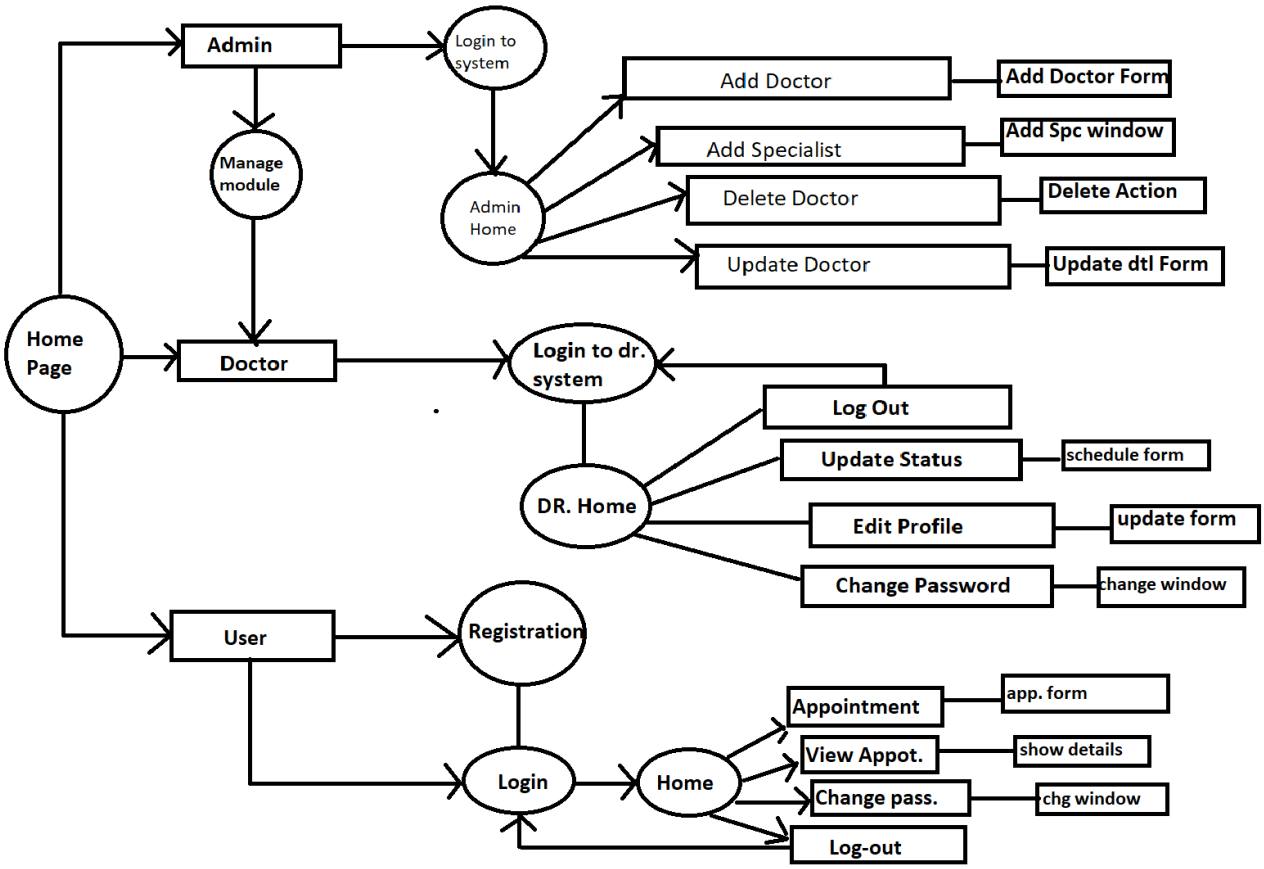
Assign roles and responsibilities within the system.

Ward and Bed Management:

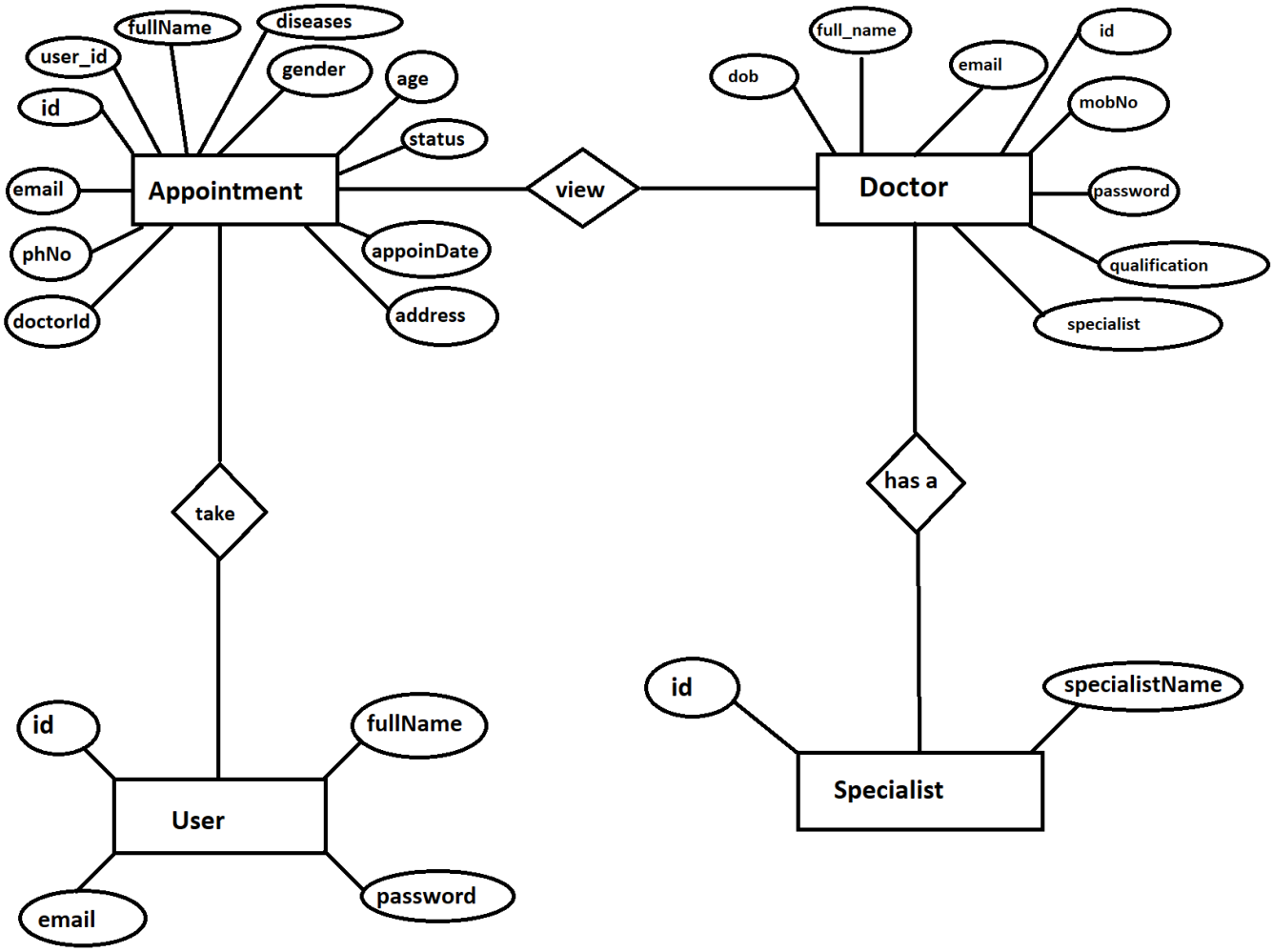
Allocate and manage patient beds and wards.

Track bed availability and occupancy.

DFD (DATA-FLOW DIAGRAM)



ER DIAGRAM SQL DataBase



Database Schemas

User table →

	id	full_name	email	password
▶	1	Lalit	lalitpatidar388@gmail.com	123
*	NULL	NULL	NULL	NULL

Doctor table →

	iddoctor	full_name	dob	qualification	specialist	email	mobno	password
	1	Lalit	2023-08-15	MBBS	cardiologist	lalitpatidar388@gmail.com	9575505209	123
▶*	NULL	NULL		NULL	NULL	NULL	NULL	NULL

Specialist table →

	id	spec_name
▶	1	Shubham Sharma
	2	cardiologist
*	NULL	NULL

Doctor table →

	id	userId	fullName	gender	age	appoinDate	email	phNo	diseases	doctorId	address	status
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Refrence

YouTube Channels - for Implementation
GitHub - for sample codes to practice
With the help of - **https://chat.openai.com**

Backend

AdminLogin.java

```
package com.admin.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.entity.User;

//@WebServlet("/loginAdmin")
public class AdminLogin extends HttpServlet {
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		try {
			String email = req.getParameter("email");
			String password = req.getParameter("password");

			HttpSession session = req.getSession();

			if ("admin@gmail.com".equals(email) && "admin".equals(password)) {
				session.setAttribute("adminObj", new User());
				resp.sendRedirect("admin/index.jsp");
			} else {
				session.setAttribute("errorMsg", "invalid email & password");
				resp.sendRedirect("admin_login.jsp");
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
	}
}
```

AdminLogout.java

```
package com.admin.servlet;
import java.io.IOException;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/adminLogout")
public class AdminLogout extends HttpServlet {
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		HttpSession session = req.getSession();
		session.removeAttribute("adminObj");
		session.setAttribute("succMsg", "Admin Logout Sucessfully");
		resp.sendRedirect("admin_login.jsp");
	}
}
```

AddDoctor.java

```
package com.admin.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.DoctorDao;
import com.db.DBConnect;
import com.entity.Doctor;

@WebServlet("/addDoctor")
public class AddDoctor extends HttpServlet {
@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		try {
			String fullName = req.getParameter("fullname");
			String dob = req.getParameter("dob");
			String qualification = req.getParameter("qualification");
```

```
                String spec = req.getParameter("spec");
                String email = req.getParameter("email");
                String mobno = req.getParameter("mobno");
                String password = req.getParameter("password");
                Doctor d = new Doctor(fullName, dob, qualification, spec, email, mobno,
password);
                DoctorDao dao = new DoctorDao(DBConnect.getConn());
                HttpSession session = req.getSession();
                if (dao.registerDoctor(d)) {
                        session.setAttribute("succMsg", "Doctor Added Sucessfully..");
                        resp.sendRedirect("admin/doctor.jsp");
                } else {
                        session.setAttribute("errorMsg", "something wrong on server");
                        resp.sendRedirect("admin/doctor.jsp");
                }
        } catch (Exception e) {
                e.printStackTrace();
        }
    }
}
```

AdminSpecialist.java

```
package com.admin.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.SpecialistDao;
import com.db.DBConnect;
import com.entity.User;

@WebServlet("/addSpecialist")
public class AddSpecialist extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        String specName = req.getParameter("specName");
```

```
			SpecialistDao dao = new SpecialistDao(DBConnect.getConn());
			boolean f = dao.addSpecialist(specName);
			HttpSession session = req.getSession();
			if (f) {
				session.setAttribute("succMsg", "Specialist Added");
				resp.sendRedirect("admin/index.jsp");
			} else {
				session.setAttribute("errorMsg", "something wrong on server");
				resp.sendRedirect("admin/index.jsp");
			}
		}
}
```

DeleteDoctor.java

```
package com.admin.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.DoctorDao;
import com.db.DBConnect;
@WebServlet("/deleteDoctor")
public class DeleteDoctor extends HttpServlet {
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
			int id = Integer.parseInt(req.getParameter("id"));
			DoctorDao dao = new DoctorDao(DBConnect.getConn());
			HttpSession session = req.getSession();
			if (dao.deleteDoctor(id)) {
				session.setAttribute("succMsg", "Doctor Delete Sucessfully..");
				resp.sendRedirect("admin/view_doctor.jsp");
			} else {
				session.setAttribute("errorMsg", "something wrong on server");
				resp.sendRedirect("admin/view_doctor.jsp");
			}
	}}
```

Doctor-Backend

DoctorLogin.java

```
package com.doctor.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.DoctorDao;
import com.db.DBConnect;
import com.entity.Doctor;
//@WebServlet("/loginDoctor")
public class DoctorLogin extends HttpServlet {
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		String email = req.getParameter("email");
		String password = req.getParameter("password");
		HttpSession session = req.getSession();
		DoctorDao dao = new DoctorDao(DBConnect.getConn());
		Doctor doctor = dao.login(email, password);
		if (doctor != null) {
			System.out.println("hello");
			session.setAttribute("doctObj", doctor);
			resp.sendRedirect("indexDoctor.jsp");
		} else {
			session.setAttribute("errorMsg", "invalid email & password");
			resp.sendRedirect("doctor_login.jsp");
		}
	}
}
```

DoctorLogout.java

```
package com.doctor.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
```

```
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet("/doctorLogout")
public class DoctorLogout extends HttpServlet{
	@Override
	protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		HttpSession session = req.getSession();
		session.removeAttribute("doctObj");
		session.setAttribute("succMsg", "Doctor Logout Sucessfully");
		resp.sendRedirect("doctor_login.jsp");
	}
}
```

User-Backend

UserLogin.java

```
package com.user.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.UserDao;
import com.db.DBConnect;
import com.entity.User;
@WebServlet("/userLogin")
public class UserLogin extends HttpServlet {
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		String email = req.getParameter("email");
		String password = req.getParameter("password");
		HttpSession session = req.getSession();
		UserDao dao = new UserDao(DBConnect.getConn());
```

```
                User user = dao.login(email, password);
                if (user != null) {
                        session.setAttribute("userObj", user);
                        resp.sendRedirect("index.jsp");
                } else {
                        session.setAttribute("errorMsg", "invalid email & password");
                        resp.sendRedirect("user_login.jsp");
                }
        }
}
```

UserRegister.java

```
package com.user.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.UserDao;
import com.db.DBConnect;
import com.entity.User;

@WebServlet("/user_register")
public class UserRegister extends HttpServlet {
        @Override
        protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
                try {
                        String fullName = req.getParameter("fullname");
                        String email = req.getParameter("email");
                        String password = req.getParameter("password");
                        User u = new User(fullName, email, password);
                        UserDao dao = new UserDao(DBConnect.getConn());
                        HttpSession session = req.getSession();
                        boolean f = dao.register(u);
                        if (f) {
                                session.setAttribute("sucMsg", "Register Sucessfully");
                                resp.sendRedirect("signup.jsp");
```

```
                } else {
                        session.setAttribute("errorMsg", "Something wrong on server");
                        resp.sendRedirect("signup.jsp");
                }
        } catch (Exception e) {
                e.printStackTrace();
        }
}
```

UserLogout.java

```
package com.user.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/userLogout")
public class UserLogout extends HttpServlet {
        @Override
        protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
                HttpSession session = req.getSession();
                session.removeAttribute("userObj");
                session.setAttribute("succMsg", "User Logout Sucessfully");
                resp.sendRedirect("user_login.jsp");
        }
}
```

AppointmentServlet.java

```
package com.user.servlet;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
import javax.servlet.http.HttpSession;
import com.dao.AppointmentDAO;
import com.db.DBConnect;
import com.entity.Appointment;

@WebServlet("/appAppointment")
public class AppointmentServlet extends HttpServlet {
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {

		int userId = Integer.parseInt(req.getParameter("userid"));
		String fullname = req.getParameter("fullname");
		String gender = req.getParameter("gender");
		String age = req.getParameter("age");
		String appoint_date = req.getParameter("appoint_date");
		String email = req.getParameter("email");
		String phno = req.getParameter("phno");
		String diseases = req.getParameter("diseases");
		int doctor_id = Integer.parseInt(req.getParameter("doct"));
		String address = req.getParameter("address");
		Appointment ap = new Appointment(userId, fullname, gender, age,
appoint_date, email, phno, diseases, doctor_id,
				address, "Pending");
		AppointmentDAO dao = new AppointmentDAO(DBConnect.getConn());
		HttpSession session = req.getSession();
		if (dao.addAppointment(ap)) {
			session.setAttribute("succMsg", "Appointment Sucessfully");
			resp.sendRedirect("user_appointment.jsp");
		} else {
			session.setAttribute("errorMsg", "Something wrong on server");
			resp.sendRedirect("user_appointment.jsp");
		}
	}
}
```

Database-Models

Doctor.java

```
package com.user.servlet;
import java.io.IOException;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dao.AppointmentDAO;
import com.db.DBConnect;
import com.entity.Appointment;
@WebServlet("/appAppointment")
public class AppointmentServlet extends HttpServlet {
	@Override
	protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
		int userId = Integer.parseInt(req.getParameter("userid"));
		String fullname = req.getParameter("fullname");
		String gender = req.getParameter("gender");
		String age = req.getParameter("age");
		String appoint_date = req.getParameter("appoint_date");
		String email = req.getParameter("email");
		String phno = req.getParameter("phno");
		String diseases = req.getParameter("diseases");
		int doctor_id = Integer.parseInt(req.getParameter("doct"));
		String address = req.getParameter("address");
		Appointment ap = new Appointment(userId, fullname, gender, age,
appoint_date, email, phno, diseases, doctor_id,
				address, "Pending");
		AppointmentDAO dao = new AppointmentDAO(DBConnect.getConn());
		HttpSession session = req.getSession();

		if (dao.addAppointment(ap)) {
			session.setAttribute("succMsg", "Appointment Sucessfully");
			resp.sendRedirect("user_appointment.jsp");
		} else {
			session.setAttribute("errorMsg", "Something wrong on server");
			resp.sendRedirect("user_appointment.jsp");
		}
	}
}
```

Appointment.java

```
package com.entity;
public class Appointment {
	private int id;
	private int userId;
	private String fullName;
	private String gender;
	private String age;
	private String appoinDate;
	private String email;
	private String phNo;
	private String diseases;
	private int doctorId;
	private String address;
	private String status;

	public Appointment() {
		super();
		// TODO Auto-generated constructor stub
	}
	public Appointment(int userId, String fullName, String gender, String age, String
appoinDate, String email,
			String phNo, String diseases, int doctorId, String address, String status) {
		super();
		this.userId = userId;
		this.fullName = fullName;
		this.gender = gender;
		this.age = age;
		this.appoinDate = appoinDate;
		this.email = email;
		this.phNo = phNo;
		this.diseases = diseases;
		this.doctorId = doctorId;
		this.address = address;
		this.status = status;
	}

	public int getId() {
		return id;
	}
```

```
public void setId(int id) {
	this.id = id;
}
public int getUserId() {
	return userId;
}
public void setUserId(int userId) {
	this.userId = userId;
}
public String getFullName() {
	return fullName;
}
public void setFullName(String fullName) {
	this.fullName = fullName;
}
public String getGender() {
	return gender;
}
public void setGender(String gender) {
	this.gender = gender;
}
public String getAge() {
	return age;
}
public void setAge(String age) {
	this.age = age;
}
public String getAppoinDate() {
	return appoinDate;
}
public void setAppoinDate(String appoinDate) {
	this.appoinDate = appoinDate;
}
public String getEmail() {
	return email;
}
public void setEmail(String email) {
	this.email = email;
}
public String getPhNo() {
	return phNo;
```

```
	}
	public void setPhNo(String phNo) {
		this.phNo = phNo;
	}
	public String getDiseases() {
		return diseases;
	}
	public void setDiseases(String diseases) {
		this.diseases = diseases;
	}
	public int getDoctorId() {
		return doctorId;

	public void setDoctorId(int doctorId) {
		this.doctorId = doctorId;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public String getStatus() {
		return status;
	}
	public void setStatus(String status) {
		this.status = status;
	}
}
```

Specialist.java

```
package com.entity;
public class Specalist {
	private int id;
	private String specialistName;
	public Specalist() {
		super();
		// TODO Auto-generated constructor stub
	}
```

```
	public Specalist(int id, String specialistName) {
		super();
		this.id = id;
		this.specialistName = specialistName;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getSpecialistName() {
		return specialistName;
	}
	public void setSpecialistName(String specialistName) {
		this.specialistName = specialistName;
	}
}
```

User.java

```
package com.entity;
public class User {
	private int id;
	private String fullName;
	private String email;
	private String password;
	public User() {
		super();
		// TODO Auto-generated constructor stub
	}
	public User(String fullName, String email, String password) {
		super();
		this.fullName = fullName;
		this.email = email;
		this.password = password;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
```

```
		this.id = id;
	}
	public String getFullName() {
		return fullName;
	}
	public void setFullName(String fullName) {
		this.fullName = fullName;
	}
	public String getEmail() {
		return email;
	}

	public void setEmail(String email) {
		this.email = email;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}
```

Databass-Connection

DBConnection.java

```
package com.db;

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnect {

	private static Connection conn;

	public static Connection getConn() {

		try {

			Class.forName("com.mysql.cj.jdbc.Driver");

			conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/hospital_management",
"root", "pass”
```

```
		} catch (Exception e) {
			e.printStackTrace();
		}
		return conn;
	}
}
```

Database-Operation

AppointmentDAO.java

```
package com.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
import com.entity.Appointment;

public class AppointmentDAO {
	private Connection conn;
	public AppointmentDAO(Connection conn) {
		super();
		this.conn = conn;
	}
	public boolean addAppointment(Appointment ap) {
		boolean f = false;
		try {
			String sql = "insert into
appointment(user_id,fullname,gender,age,appoint_date,email,phno,diseases,doctor_id,add
ress,status) values(?,?,?,?,?,?,?,?,?,?,?)";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setInt(1, ap.getUserId());
			ps.setString(2, ap.getFullName());
			ps.setString(3, ap.getGender());
			ps.setString(4, ap.getAge());
			ps.setString(5, ap.getAppoinDate());
			ps.setString(6, ap.getEmail());
			ps.setString(7, ap.getPhNo());
```

```
			ps.setString(8, ap.getDiseases());
			ps.setInt(9, ap.getDoctorId());
			ps.setString(10, ap.getAddress());
			ps.setString(11, ap.getStatus());
			int i = ps.executeUpdate();
			if (i == 1) {
				f = true;
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
		return f;
	}
	public List<Appointment> getAllAppointmentByLoginUser(int userId) {
		List<Appointment> list = new ArrayList<Appointment>();
		Appointment ap = null;
		try {
			String sql = "select * from appointment where user_id=?";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setInt(1, userId);
			ResultSet rs = ps.executeQuery();
			while (rs.next()) {
				ap = new Appointment();
				ap.setId(rs.getInt(1));
				ap.setUserId(rs.getInt(2));
				ap.setFullName(rs.getString(3));
				ap.setGender(rs.getString(4));
				ap.setAge(rs.getString(5));
				ap.setAppoinDate(rs.getString(6));
				ap.setEmail(rs.getString(7));
				ap.setPhNo(rs.getString(8));
				ap.setDiseases(rs.getString(9));
				ap.setDoctorId(rs.getInt(10));
				ap.setAddress(rs.getString(11));
				ap.setStatus(rs.getString(12));
				list.add(ap);
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
		return list;
```

```
}
public List<Appointment> getAllAppointmentByDoctorLogin(int doctorId) {
	List<Appointment> list = new ArrayList<Appointment>();
	Appointment ap = null;
	try {
		String sql = "select * from appointment where doctor_id=?";
		PreparedStatement ps = conn.prepareStatement(sql);
		ps.setInt(1, doctorId);
		ResultSet rs = ps.executeQuery();
		while (rs.next()) {
			ap = new Appointment();
			ap.setId(rs.getInt(1));
			ap.setUserId(rs.getInt(2));
			ap.setFullName(rs.getString(3));
			ap.setGender(rs.getString(4));
			ap.setAge(rs.getString(5));
			ap.setAppoinDate(rs.getString(6));
			ap.setEmail(rs.getString(7));
			ap.setPhNo(rs.getString(8));
			ap.setDiseases(rs.getString(9));
			ap.setDoctorId(rs.getInt(10));
			ap.setAddress(rs.getString(11));
			ap.setStatus(rs.getString(12));
			list.add(ap);
		}

	} catch (Exception e) {
		e.printStackTrace();
	}
	return list;
}
public Appointment getAppointmentById(int id) {
	Appointment ap = null;
	try {
		String sql = "select * from appointment where id=?";
		PreparedStatement ps = conn.prepareStatement(sql);
		ps.setInt(1, id);

		ResultSet rs = ps.executeQuery();
		while (rs.next()) {
			ap = new Appointment();
```

```
                ap.setId(rs.getInt(1));
                ap.setUserId(rs.getInt(2));
                ap.setFullName(rs.getString(3));
                ap.setGender(rs.getString(4));
                ap.setAge(rs.getString(5));
                ap.setAppoinDate(rs.getString(6));
                ap.setEmail(rs.getString(7));
                ap.setPhNo(rs.getString(8));
                ap.setDiseases(rs.getString(9));
                ap.setDoctorId(rs.getInt(10));
                ap.setAddress(rs.getString(11));
                ap.setStatus(rs.getString(12));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ap;
    }
    public boolean updateCommentStatus(int id, int doctId, String comm) {
        boolean f = false;
        try {
            String sql = "update appointment set status=? where id=? and
doctor_id=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, comm);
            ps.setInt(2, id);
            ps.setInt(3, doctId);

            int i = ps.executeUpdate();
            if (i == 1) {
                f = true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return f;
    }
    public List<Appointment> getAllAppointment() {
        List<Appointment> list = new ArrayList<Appointment>();
        Appointment ap = null;
        try {
```

```
                String sql = "select * from appointment order by id desc";
                PreparedStatement ps = conn.prepareStatement(sql);
                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                        ap = new Appointment();
                        ap.setId(rs.getInt(1));
                        ap.setUserId(rs.getInt(2));
                        ap.setFullName(rs.getString(3));
                        ap.setGender(rs.getString(4));
                        ap.setAge(rs.getString(5));
                        ap.setAppoinDate(rs.getString(6));
                        ap.setEmail(rs.getString(7));
                        ap.setPhNo(rs.getString(8));
                        ap.setDiseases(rs.getString(9));
                        ap.setDoctorId(rs.getInt(10));
                        ap.setAddress(rs.getString(11));
                        ap.setStatus(rs.getString(12));
                        list.add(ap);
                }
        } catch (Exception e) {
                e.printStackTrace();
        }
        return list;
    }
}
```

DoctorDao.java

```
package com.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import com.entity.Doctor;

public class DoctorDao {
    private Connection conn;
```

```
	public DoctorDao(Connection conn) {
		super();
		this.conn = conn;
	}

	public boolean registerDoctor(Doctor d) {
		boolean f = false;

		try {
			String sql = "insert into
doctor(full_name,dob,qualification,specialist,email,mobno,password) values(?,?,?,?,?,?,?)";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setString(1, d.getFullName());
			ps.setString(2, d.getDob());
			ps.setString(3, d.getQualification());
			ps.setString(4, d.getSpecialist());
			ps.setString(5, d.getEmail());
			ps.setString(6, d.getMobNo());
			ps.setString(7, d.getPassword());

			int i = ps.executeUpdate();
			if (i == 1) {
				f = true;
			}
		} catch (Exception e) {
			e.printStackTrace();
		}

		return f;
	}

	public List<Doctor> getAllDoctor() {
		List<Doctor> list = new ArrayList<Doctor>();
		Doctor d = null;
		try {

			String sql = "select * from doctor order by id desc";
			PreparedStatement ps = conn.prepareStatement(sql);

			ResultSet rs = ps.executeQuery();
			while (rs.next()) {
```

```
                d = new Doctor();
                d.setId(rs.getInt(1));
                d.setFullName(rs.getString(2));
                d.setDob(rs.getString(3));
                d.setQualification(rs.getString(4));
                d.setSpecialist(rs.getString(5));
                d.setEmail(rs.getString(6));
                d.setMobNo(rs.getString(7));
                d.setPassword(rs.getString(8));
                list.add(d);
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
        return list;
}

public Doctor getDoctorById(int id) {

        Doctor d = null;
        try {

            String sql = "select * from doctor where id=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                d = new Doctor();
                d.setId(rs.getInt(1));
                d.setFullName(rs.getString(2));
                d.setDob(rs.getString(3));
                d.setQualification(rs.getString(4));
                d.setSpecialist(rs.getString(5));
                d.setEmail(rs.getString(6));
                d.setMobNo(rs.getString(7));
                d.setPassword(rs.getString(8));

            }
```

```
		} catch (Exception e) {
			e.printStackTrace();
		}
		return d;
	}

	public boolean updateDoctor(Doctor d) {
		boolean f = false;

		try {
			String sql = "update doctor set
full_name=?,dob=?,qualification=?,specialist=?,email=?,mobno=?,password=? where id=?";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setString(1, d.getFullName());
			ps.setString(2, d.getDob());
			ps.setString(3, d.getQualification());
			ps.setString(4, d.getSpecialist());
			ps.setString(5, d.getEmail());
			ps.setString(6, d.getMobNo());
			ps.setString(7, d.getPassword());
			ps.setInt(8, d.getId());
			int i = ps.executeUpdate();

			if (i == 1) {
				f = true;
			}
		} catch (Exception e) {
			e.printStackTrace();
		}

		return f;
	}

	public boolean deleteDoctor(int id) {
		boolean f = false;
		try {
			String sql = "delete from doctor where id=?";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setInt(1, id);

			int i = ps.executeUpdate();
```

```
            if (i == 1) {
                f = true;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return f;
    }

    public Doctor login(String email, String psw) {
        Doctor d = null;
        try {

            String sql = "select * from doctor where email=? and password=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, email);
            ps.setString(2, psw);

            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                d = new Doctor();
                d = new Doctor();
                d.setId(rs.getInt(1));
                d.setFullName(rs.getString(2));
                d.setDob(rs.getString(3));
                d.setQualification(rs.getString(4));
                d.setSpecialist(rs.getString(5));
                d.setEmail(rs.getString(6));
                d.setMobNo(rs.getString(7));
                d.setPassword(rs.getString(8));
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return d;
    }
```

```
public int countDoctor() {
	int i = 0;
	try {
		String sql = "select * from doctor";
		PreparedStatement ps = conn.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();
		while (rs.next()) {
			i++;
		}

	} catch (Exception e) {
		e.printStackTrace();
	}

	return i;
}

public int countAppointment() {
	int i = 0;
	try {
		String sql = "select * from appointment";
		PreparedStatement ps = conn.prepareStatement(sql);
		ResultSet rs = ps.executeQuery();
		while (rs.next()) {
			i++;
		}

	} catch (Exception e) {
		e.printStackTrace();
	}

	return i;
}

public int countAppointmentByDocotrId(int did) {
	int i = 0;
	try {
		String sql = "select * from appointment where doctor_id=?";
		PreparedStatement ps = conn.prepareStatement(sql);
		ps.setInt(1, did);
		ResultSet rs = ps.executeQuery();
```

```
            while (rs.next()) {
                i++;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return i;
    }

    public int countUSer() {
        int i = 0;
        try {
            String sql = "select * from user_dtls";
            PreparedStatement ps = conn.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                i++;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }

        return i;
    }

    public int countSpecialist() {
        int i = 0;
        try {
            String sql = "select * from specialist";
            PreparedStatement ps = conn.prepareStatement(sql);
            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                i++;
            }

        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
        return i;
}

public boolean checkOldPassword(int userid, String oldPassword) {
        boolean f = false;

        try {
                String sql = "select * from doctor where id=? and password=?";
                PreparedStatement ps = conn.prepareStatement(sql);
                ps.setInt(1, userid);
                ps.setString(2, oldPassword);

                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                        f = true;
                }

        } catch (Exception e) {
                e.printStackTrace();
        }

        return f;
}

public boolean changePassword(int userid, String newPassword) {
        boolean f = false;

        try {
                String sql = "update doctor set password=? where id=?";
                PreparedStatement ps = conn.prepareStatement(sql);
                ps.setString(1, newPassword);
                ps.setInt(2, userid);

                int i = ps.executeUpdate();
                if (i == 1) {
                        f = true;
                }

        } catch (Exception e) {
                e.printStackTrace();
```

```
		}

		return f;
	}

	public boolean editDoctorProfile(Doctor d) {
		boolean f = false;

		try {
			String sql = "update doctor set
full_name=?,dob=?,qualification=?,specialist=?,email=?,mobno=? where id=?";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setString(1, d.getFullName());
			ps.setString(2, d.getDob());
			ps.setString(3, d.getQualification());
			ps.setString(4, d.getSpecialist());
			ps.setString(5, d.getEmail());
			ps.setString(6, d.getMobNo());
			ps.setInt(7, d.getId());
			int i = ps.executeUpdate();
			if (i == 1) {
				f = true;
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
		return f;
	}
}
```

SpecialistDao.java

```
package com.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;
```

```
import com.entity.Specalist;
public class SpecialistDao {
	private Connection conn;
	public SpecialistDao(Connection conn) {
		super();
		this.conn = conn;
	}
	public boolean addSpecialist(String spec) {
		boolean f = false;
		try {
			String sql = "insert into specialist(spec_name) values(?)";
			PreparedStatement ps = conn.prepareStatement(sql);
			ps.setString(1, spec);
			int i = ps.executeUpdate();
			if (i == 1) {
				f = true;
			}
		} catch (Exception e) {
			e.printStackTrace();
		}
		return f;
	}
	public List<Specalist> getAllSpecialist() {
		List<Specalist> list = new ArrayList<Specalist>();
		Specalist s = null;
		try {
			String sql = "select * from specialist";
			PreparedStatement ps = conn.prepareStatement(sql);
			ResultSet rs = ps.executeQuery();
```

```
                while (rs.next()) {
                        s = new Specalist();
                        s.setId(rs.getInt(1));
                        s.setSpecialistName(rs.getString(2));
                        list.add(s);
                }
        } catch (Exception e) {
                e.printStackTrace();
        }
        return list;
    }
}
```

UserDao.java

```
package com.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import com.entity.User;
public class UserDao {
    private Connection conn;
    public UserDao(Connection conn) {
        super();
        this.conn = conn;
    }
    public boolean register(User u) {
        boolean f = false;
        try {
```

```
                String sql = "insert into user_dtls(full_name,email,password)
values(?,?,?) ";
                PreparedStatement ps = conn.prepareStatement(sql);
                ps.setString(1, u.getFullName());
                ps.setString(2, u.getEmail());
                ps.setString(3, u.getPassword());
                int i = ps.executeUpdate();
                if (i == 1) {
                        f = true;
                }
        } catch (Exception e) {
                e.printStackTrace();
        }
        return f;
    }
    public User login(String em, String psw) {
        User u = null;
        try {
                String sql = "select * from user_dtls where email=? and password=?";
                PreparedStatement ps = conn.prepareStatement(sql);
                ps.setString(1, em);
                ps.setString(2, psw);
                ResultSet rs = ps.executeQuery();
                while (rs.next()) {
                        u = new User();
                        u.setId(rs.getInt(1));
                        u.setFullName(rs.getString(2));
                        u.setEmail(rs.getString(3));
                        u.setPassword(rs.getString(4));
```

```
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return u;
    }
    public boolean checkOldPassword(int userid, String oldPassword) {
        boolean f = false;
        try {
            String sql = "select * from user_dtls where id=? and password=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setInt(1, userid);
            ps.setString(2, oldPassword);

            ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                f = true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return f;
    }
    public boolean changePassword(int userid, String newPassword) {
        boolean f = false;
        try {
            String sql = "update user_dtls set password=? where id=?";
            PreparedStatement ps = conn.prepareStatement(sql);
            ps.setString(1, newPassword);
```

```
                ps.setInt(2, userid);
                int i = ps.executeUpdate();
                if (i == 1) {
                        f = true;
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
            return f;
        }
}
```

Frontend

Admin_login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Admin Login Page</title>
<%@include file="component/allcss.jsp"%>
<style type="text/css">
.paint-card {
    box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
</head>
<body>
    <%@include file="component/navbar.jsp"%>
    <div class="container p-5">
        <div class="row">
            <div class="col-md-4 offset-md-4">
```

```
                        <div class="card paint-card">
                            <div class="card-body">
                                <p class="fs-4 text-center">Admin Login</p>
                                <c:if test="${not empty succMsg }">
                                    <p class="text-center text-success fs-
3">${succMsg}</p>
                                    <c:remove var="succMsg" scope="session" />
                                </c:if>
                                <c:if test="${not empty errorMsg }">
                                    <p class="text-center text-danger fs-
5">${errorMsg}</p>
                                    <c:remove var="errorMsg" scope="session" />
                                </c:if>
                                <form action="loginAdmin" method="post">
                                    <div class="mb-3">
                                        <label class="form-label">Email
address</label> <input required
                                            name="email" type="email"
class="form-control">
                                    </div>
                                    <div class="mb-3">
                                        <label class="form-
label">Password</label> <input required
                                            name="password"
type="password" class="form-control">
                                    </div>
                                    <button type="submit" class="btn bg-success
text-white col-md-12">Login</button>
                                </form>
                            </div>
                        </div>
                    </div>
                </div>
            </body>
            </html>
```

Doctor_login.jsp

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

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Docotr Login Page</title>
<%@include file="component/allcss.jsp"%>
<style type="text/css">
.paint-card {
	box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
</head>
<body>
	<%@include file="component/navbar.jsp"%>
	<div class="container p-5">
		<div class="row">
			<div class="col-md-4 offset-md-4">
				<div class="card paint-card">
					<div class="card-body">
						<p class="fs-4 text-center">Doctor Login</p>

						<c:if test="${not empty succMsg }">
							<p class="text-center text-success fs-
3">${succMsg}</p>
							<c:remove var="succMsg" scope="session" />
						</c:if>

						<c:if test="${not empty errorMsg }">
							<p class="text-center text-danger fs-
5">${errorMsg}</p>
							<c:remove var="errorMsg" scope="session" />
						</c:if>

						<form action="loginDoctor" method="post">
							<div class="mb-3">
								<label class="form-label">Email
address</label> <input required
									name="email" type="email"
class="form-control">
							</div>
							<div class="mb-3">
```

```
                                                    <label class="form-
label">Password</label> <input required
                                                            name="password"
type="password" class="form-control">
                                                </div>
                                                <button type="submit" class="btn bg-success
text-white col-md-12">Login</button>
                                            </form>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
</body>
</html>
```

Index.jsp

```
<%@page import="com.db.DBConnect"%>
<%@page import="java.sql.Connection"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
        pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Index Page</title>
<%@include file="component/allcss.jsp"%>

<style type="text/css">
.paint-card {
        box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
</head>
<body>
        <%@include file="component/navbar.jsp"%>
        <div id="carouselExampleIndicators" class="carousel slide"
                data-bs-ride="carousel">
                <div class="carousel-indicators">
                        <button type="button" data-bs-target="#carouselExampleIndicators"
```

```
                    data-bs-slide-to="0" class="active" aria-current="true"
                    aria-label="Slide 1"></button>
                <button type="button" data-bs-target="#carouselExampleIndicators"
                    data-bs-slide-to="1" aria-label="Slide 2"></button>
                <button type="button" data-bs-target="#carouselExampleIndicators"
                    data-bs-slide-to="2" aria-label="Slide 3"></button>
            </div>
            <div class="carousel-inner">
                <div class="carousel-item active">
                    <img src="img/hsopital1.webp" class="d-block w-100" alt="..."
                        height="500px">
                </div>
                <div class="carousel-item">
                    <img src="img/hosptalHome.webp" class="d-block w-100" alt="..."
                        height="500px">
                </div>
                <div class="carousel-item">
                    <img src="img/hs.jpg" class="d-block w-100" alt="..."
                        height="500px">
                </div>
            </div>
            <button class="carousel-control-prev" type="button"
                data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">
                <span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span
                    class="visually-hidden">Previous</span>
            </button>
            <button class="carousel-control-next" type="button"
                data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
                <span class="carousel-control-next-icon" aria-hidden="true"></span>
<span
                    class="visually-hidden">Next</span>
            </button>
        </div>

        <div class="container p-3">
            <p class="text-center fs-2 ">Key Features of our Hospital</p>

            <div class="row">
                <div class="col-md-8 p-5">
                    <div class="row">
```

```
                    <div class="col-md-6">
                        <div class="card paint-card">
                            <div class="card-body">
                                <p class="fs-5">100% Safety</p>
                                <p>Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
                                    Voluptatem, inventore</p>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6">
                        <div class="card paint-card">
                            <div class="card-body">
                                <p class="fs-5">Clean Environment</p>
                                <p>Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
                                    Voluptatem, inventore</p>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6 mt-2">
                        <div class="card paint-card">
                            <div class="card-body">
                                <p class="fs-5">Friendly Doctors</p>
                                <p>Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
                                    Voluptatem, inventore</p>
                            </div>
                        </div>
                    </div>
                    <div class="col-md-6 mt-2">
                        <div class="card paint-card">
                            <div class="card-body">
                                <p class="fs-5">Medical Research</p>
                                <p>Lorem ipsum dolor sit amet,
consectetur adipisicing elit.
                                    Voluptatem, inventore</p>
                            </div>
                        </div>
                    </div>
                </div>
```

```
                </div>
                <div class="col-md-4">
                    <img alt="" src="">
                </div>
            </div>
        </div>
        <hr>
        <div class="container p-2">
            <p class="text-center fs-2 ">Our Team</p>

            <div class="row">
                <div class="col-md-3">
                    <div class="card paint-card">
                        <div class="card-body text-center">
                            <img src="img/doc1.jpeg" width="250px"
height="300px">
                            <p class="fw-bold fs-5">Dr.Samuani Simi</p>
                            <p class="fs-7">(Heart Specialist)</p>
                        </div>
                    </div>
                </div>

                <div class="col-md-3">
                    <div class="card paint-card">
                        <div class="card-body text-center">
                            <img src="img/doc2.jpeg" width="250px"
height="300px">
                            <p class="fw-bold fs-5">Dr.Siva Kumar</p>
                            <p class="fs-7">(Neuro Surgeon)</p>
                        </div>
                    </div>
                </div>

                <div class="col-md-3">
                    <div class="card paint-card">
                        <div class="card-body text-center">
                            <img src="img/doc3.jpeg" width="250px"
height="300px">
                            <p class="fw-bold fs-5">Dr. Nargis Mehta</p>
                            <p class="fs-7">(Neuro Surgeon)</p>
                        </div>
```

```
                        </div>
                    </div>

                    <div class="col-md-3">
                        <div class="card paint-card">
                            <div class="card-body text-center">
                                <img src="img/doc4.jpeg" width="250px"
height="300px">
                                <p class="fw-bold fs-5">Dr. Pooja Sharma</p>
                                <p class="fs-7">(Heart Specialist)</p>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
<%@include file="component/footer.jsp" %>
</body>
</html>
```

IndexDoctor.jsp

```
<%@page import="com.entity.Doctor"%>
<%@page import="com.db.DBConnect"%>
<%@page import="com.dao.DoctorDao"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<style type="text/css">
.paint-card {
    box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
<%@include file="../component/allcss.jsp"%>
</head>
<body>
```

```
<c:if test="${empty doctObj }">
	<c:redirect url="../doctor_login.jsp"></c:redirect>
</c:if>


<%@include file="navbar.jsp"%>

<p class="text-center fs-3">Doctor Dashboard</p>

<%
Doctor d = (Doctor) session.getAttribute("doctObj");
DoctorDao dao = new DoctorDao(DBConnect.getConn());
%>
<div class="container p-5">
	<div class="row">
		<div class="col-md-4 offset-md-2">
			<div class="card paint-card">
				<div class="card-body text-center text-success">
					<i class="fas fa-user-md fa-3x"></i><br>
					<p class="fs-4 text-center">
						Doctor <br><%=dao.countDoctor()%>
					</p>
				</div>
			</div>
		</div>


		<div class="col-md-4">
			<div class="card paint-card">
				<div class="card-body text-center text-success">
					<i class="far fa-calendar-check fa-3x"></i><br>
					<p class="fs-4 text-center">
						Total Appointment <br>

<%=dao.countAppointmentByDocotrId(d.getId())%>
					</p>
				</div>
			</div>
		</div>

	</div>
```

```
	</div>
</body>
</html>
```

Navbar.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<nav class="navbar navbar-expand-lg navbar-dark bg-success">
	<div class="container-fluid">
		<a class="navbar-brand" href="index.jsp"><i
			class="fas fa-clinic-medical"></i> Medi Home</a>
		<button class="navbar-toggler" type="button" data-bs-toggle="collapse"
			data-bs-target="#navbarSupportedContent"
			aria-controls="navbarSupportedContent" aria-expanded="false"
			aria-label="Toggle navigation">
			<span class="navbar-toggler-icon"></span>
		</button>
		<div class="collapse navbar-collapse" id="navbarSupportedContent">
			<ul class="navbar-nav me-auto mb-2 mb-lg-0">
				<li class="nav-item"><a class="nav-link active"
href="index.jsp">HOME</a></li>
				<li class="nav-item"><a class="nav-link active"
href="doctor.jsp">DOCTOR</a></li>
				<li class="nav-item"><a class="nav-link active"
href="view_doctor.jsp">VIEW DOCTOR</a></li>
				<li class="nav-item"><a class="nav-link active"
href="patient.jsp">PATIENT</a></li>

			</ul>

			<form class="d-flex">
				<div class="dropdown">
					<button class="btn btn-light dropdown-toggle"
type="button"
						id="dropdownMenuButton1" data-bs-
toggle="dropdown"
						aria-expanded="false">Admin</button>
					<ul class="dropdown-menu" aria-
labelledby="dropdownMenuButton1">
```

```
                                        <li><a class="dropdown-item"
href="../adminLogout">Logout</a></li>
                                    </ul>
                                </div>
                            </form>
                        </div>
                    </div>
</nav>
```

Signup.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<%@include file="component/allcss.jsp"%>
<style type="text/css">
.paint-card {
    box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
</head>
<body>
    <%@include file="component/navbar.jsp"%>
    <div class="container p-5">
        <div class="row">
            <div class="col-md-4 offset-md-4">
                <div class="card paint-card">
                    <div class="card-body">
                        <p class="fs-4 text-center">User Register</p>

                        <c:if test="${not empty sucMsg }">
                            <p class="text-center text-success fs-
3">${sucMsg}</p>
                            <c:remove var="sucMsg" scope="session" />
                        </c:if>
```

```
                                        <c:if test="${not empty errorMsg }">
                                            <p class="text-center text-danger fs-
3">${errorMsg}</p>
                                            <c:remove var="errorMsg" scope="session" />
                                        </c:if>

                                        <form action="user_register" method="post">
                                            <div class="mb-3">
                                                <label class="form-label">Full
Name</label> <input required
                                                    name="fullname" type="text"
class="form-control">
                                            </div>
                                            <div class="mb-3">
                                                <label class="form-label">Email
address</label> <input required
                                                    name="email" type="email"
class="form-control">
                                            </div>
                                            <div class="mb-3">
                                                <label class="form-
label">Password</label> <input required
                                                    name="password"
type="password" class="form-control">
                                            </div>
                                            <button type="submit" class="btn bg-success
text-white col-md-12">Register</button>
                                        </form>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
</body>
</html>
```

userLogin.jsp

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

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>User Login Page</title>
<%@include file="component/allcss.jsp"%>
<style type="text/css">
.paint-card {
	box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
}
</style>
</head>
<body>
	<%@include file="component/navbar.jsp"%>
	<div class="container p-5">
		<div class="row">
			<div class="col-md-4 offset-md-4">
				<div class="card paint-card">
					<div class="card-body">
						<p class="fs-4 text-center">User Login</p>

						<c:if test="${not empty succMsg }">
							<p class="text-center text-success fs-
3">${succMsg}</p>
							<c:remove var="succMsg" scope="session" />
						</c:if>
						<c:if test="${not empty errorMsg }">
							<p class="text-center text-danger fs-
5">${errorMsg}</p>
							<c:remove var="errorMsg" scope="session" />
						</c:if>
						<form action="userLogin" method="post">
							<div class="mb-3">
								<label class="form-label">Email
address</label> <input required
									name="email" type="email"
class="form-control">
							</div>
							<div class="mb-3">
```

```
                                                    <label class="form-
label">Password</label> <input required
                                                            name="password"
type="password" class="form-control">
                                                </div>
                                                <button type="submit" class="btn bg-success
text-white col-md-12">Login</button>
                                            </form>

                                            <br> Don't have an account? <a href="signup.jsp"
                                                class="text-decoration-none"> create one</a>
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
</body>
</html>
```

View_appointment.jsp

```
<%@page import="com.entity.User"%>
<%@page import="com.entity.Doctor"%>
<%@page import="com.dao.DoctorDao"%>
<%@page import="com.entity.Appointment"%>
<%@page import="java.util.List"%>
<%@page import="com.db.DBConnect"%>
<%@page import="com.dao.AppointmentDAO"%>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@page isELIgnored="false"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<%@include file="component/allcss.jsp"%>
<style type="text/css">
.paint-card {
    box-shadow: 0 0 10px 0 rgba(0, 0, 0, 0.3);
```

```
}

.backImg {
	background: linear-gradient(rgba(0, 0, 0, .4), rgba(0, 0, 0, .4)),
		url("img/hospital.jpg");
	height: 20vh;
	width: 100%;
	background-size: cover;
	background-repeat: no-repeat;
}
</style>
</head>
<body>
	<c:if test="${empty userObj }">
		<c:redirect url="user_login.jsp"></c:redirect>
	</c:if>
	<%@include file="component/navbar.jsp"%>

	<div class="container-fulid backImg p-5">
		<p class="text-center fs-2 text-white"></p>
	</div>
	<div class="container p-3">
		<div class="row">
			<div class="col-md-9">
				<div class="card paint-card">
					<div class="card-body">
						<p class="fs-4 fw-bold text-center text-
success">Appointment
							List</p>
						<table class="table">
							<thead>
								<tr>
									<th scope="col">Full Name</th>
									<th scope="col">Gender</th>
									<th scope="col">Age</th>
									<th scope="col">Appoint
Date</th>
									<th scope="col">Diseases</th>
									<th scope="col">Doctor
Name</th>
									<th scope="col">Status</th>
```

```
                    </tr>
                </thead>
                <tbody>
                    <%
                    User user = (User)
session.getAttribute("userObj");
                    AppointmentDAO dao = new
AppointmentDAO(DBConnect.getConn());
                    DoctorDao dao2 = new
DoctorDao(DBConnect.getConn());
                    List<Appointment> list =
dao.getAllAppointmentByLoginUser(user.getId());
                    for (Appointment ap : list) {
                        Doctor d =
dao2.getDoctorById(ap.getDoctorId());
                    %>
                    <tr>
                        <th><%=ap.getFullName()%></th>
                        <td><%=ap.getGender()%></td>
                        <td><%=ap.getAge()%></td>

        <td><%=ap.getAppoinDate()%></td>

                        <td><%=ap.getDiseases()%></td>
                        <td><%=d.getFullName()%></td>
                        <td>
                            <%
                            if
("Pending".equals(ap.getStatus())) {
                            %> <a href="#" class="btn
btn-sm btn-warning">Pending</a> <%
 } else {
 %> <%=ap.getStatus()%> <%
 }
 %>
                        </td>
                    </tr>
                    <%
                    }
                    %>
                </tbody>
```

```
                                    </table>

                                </div>
                            </div>
                        </div>
                        <div class="col-md-3 p-3">
                            <img alt="" src="img/doct.jpg">
                        </div>
                    </div>
            </div>
</body>
</html>
```

Allcss.jsp

```
<link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
        rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
        crossorigin="anonymous">
<link rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-
beta3/css/all.min.css"
        integrity="sha512-
Fo3rlrZj/k7ujTnHg4CGR2D7kSs0v4LLanw2qksYuRlEzO+tcaEPQogQ0KaoGN26/zrn20ImR1DfuLWnOo7aBA=="
        crossorigin="anonymous" referrerpolicy="no-referrer" />
<script
        src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
        crossorigin="anonymous"></script>
```

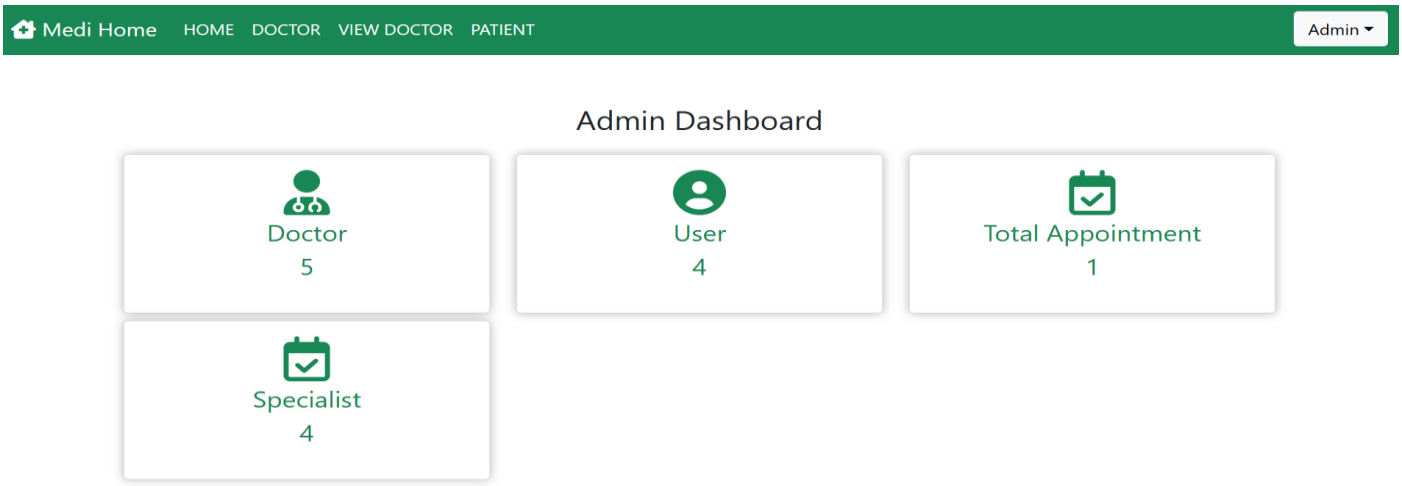
footer.jsp

```
<div class="container-fluid p-1 bg-success text-center text-white">
        <p>@copyright CityHospital.com</p>
        <p>for more - lalitpatidar@gmail.com</p>
</div>
```

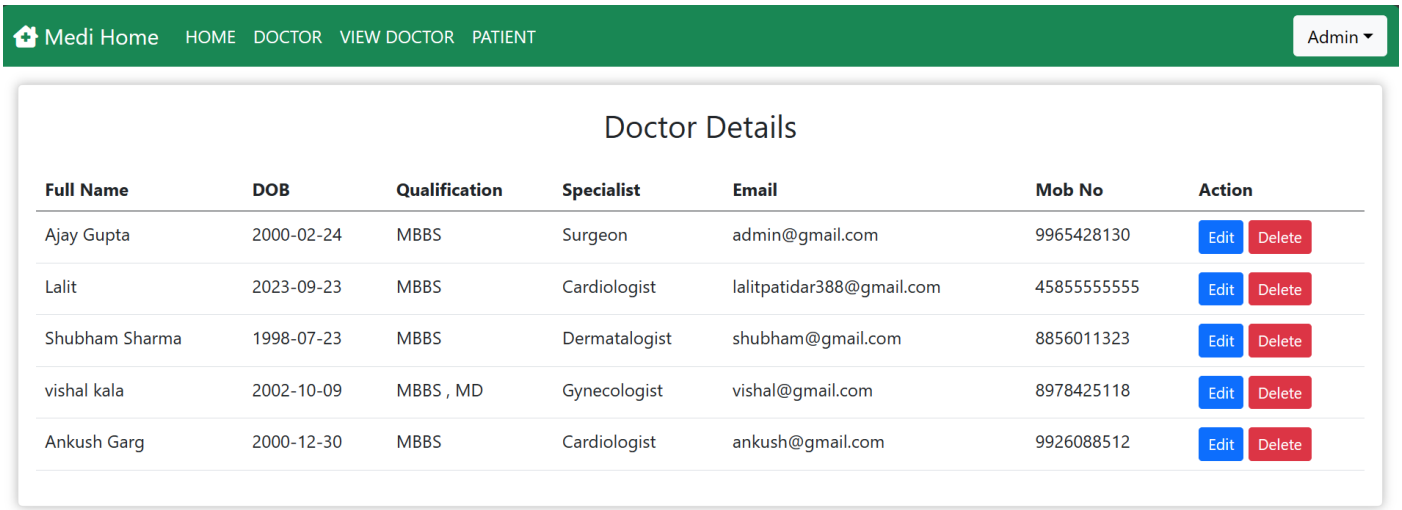
HomePage

TeamPage

Admin-Dashboard



Team Overview



Appointment Page