

**SIMATS SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**AN ONLINE HOSPITAL APPOINTMENT BOOKING SYSTEM**

**A CAPSTONE PROJECT REPORT**

*Submitted in the partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**Computer Science and Engineering**

**Submitted by**

**M. Sai Prathap Reedy (192210657)**

**Sanikommu Venkata Sainadh (192210554)**

**Under the Supervision of**

**Dr. S.K Saravanan**

**JULY 2024**

**DECLARATION**

We, **M. Sai Prathap Reddy, S. Venkata Sainadh** students of **Bachelor of Engineering in CSE**, Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled **Hospital management system** is the outcome of our bonafide work and is correct to the best of our knowledge and this work has been undertaken taking care of Engineering Ethics.

**M. Sai Prathap Reddy (192210657)**

**Sanikommu Venkata Sainadh (192210554)**

Date:

Place:

**CERTIFICATE**

This is to certify that the project entitled **“Hospital Management System”** was submitted by **M. Sai Prathap Reedy (192210657) , S.Venkata Sainadh (192210554)**has been carried out under my supervision. The project has been submitted per the requirements in the current B. Tech Computer Science semester.

Dr. S.K Saravanan

Teacher-in-charge

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPICS** | **Page No** |
| 1 | **Abstract** | **5** |
| 2 | **Introduction** | **5** |
| 3 | **Project Description** | **6** |
| 4 | **Problem Description** | **6** |
| 5 | **Tool Description** | **7** |
| 6 | **Operations** 6.1 User Operations6.2 Administrator Operations | **7** |
| 7 | **Module Description** 7.1 Interaction Module (User)7.2 Appointment Management Module (User)7.3 Administrative Module (Administrator) | **9** |
| 8 | **Implementation** | **11** |
| 8 | **Result** | **19** |
| 9 | **Conclusion** | **19** |
| 10 | **Future Enhancements** | **19** |
|  | **Screenshots** | **20** |
|  | **References** | **24** |

1. **ABSTRACT**

This project presents the design and implementation of an integrated Hospital management system aimed at streamlining hospital operations, improving patient care, and enhancing data management. The front end, crafted with HTML, CSS, and JavaScript, provides an intuitive and responsive user experience, while the backend, developed using Node.js and Express.js, ensures secure and efficient data handling. Key features include patient registration, appointment scheduling, medical records management, billing, and reporting, all supported by a MySQL database for reliable data storage and integrity. Role-based access control protects sensitive information, ensuring privacy and security. Emphasizing scalability, security, and usability, this system adapts to hospitals of various sizes, reducing administrative burdens and enhancing operational efficiency, ultimately improving patient satisfaction. This comprehensive solution addresses the challenges of modern healthcare institutions, demonstrating significant advancements in healthcare technology.

1. **INTRODUCTION**

In today's rapidly evolving healthcare landscape, the need for efficient and integrated hospital management systems has never been more critical. Traditional paper-based systems and fragmented digital solutions often lead to inefficiencies, errors, and delays, adversely affecting patient care and hospital operations. To address these challenges, this project focuses on developing an integrated front-end and back-end web-based hospital management system that enhances operational efficiency, improves patient care, and streamlines administrative processes.

The proposed system aims to bridge the gap between different hospital departments, providing a seamless flow of information and facilitating better communication among healthcare providers, patients, and administrative staff. By leveraging modern web technologies, the system offers a user-friendly front-end interface designed with HTML, CSS, and JavaScript, ensuring a responsive and intuitive experience for all users, including patients, doctors, and administrative personnel.

The back-end system utilizes robust server-side technologies such as Node.js and Express.js to handle data processing, storage, and retrieval. The integration of a relational database like MySQL guarantees reliable and secure data management. Key functionalities of the system include patient registration, appointment scheduling, medical records management, billing, and comprehensive reporting.

Security and privacy are paramount in healthcare applications. Therefore, the system incorporates role-based access control to ensure that sensitive information is accessible only to authorized personnel. This feature not only protects patient data but also complies with healthcare regulations and standards.

1. **PROJECT DESCRIPTION**

#### The Online Hospital Appointment Booking Application streamlines the process of scheduling medical appointments by offering a user-friendly web interface for patient registration, appointment scheduling, and record searching. The application includes:

#### Proposed Method

* **Front-end Development**: Utilizing Visual Studio for designing responsive and intuitive user interfaces.
* **Back-end Development**: Using XAMPP stack (Apache, MySQL, PHP) to handle server-side scripting, database management via phpMyAdmin, and ensuring secure data storage and retrieval.

1. **Project Description**

**Existing System**

The existing manual hospital appointment booking systems are plagued with inefficiencies that hinder smooth and timely access to healthcare services. Long wait times for appointment scheduling are a common issue, as the manual process involves extensive paperwork and administrative steps that slow down the system, leading to frustration among patients. This not only causes delays in getting medical attention but also increases the workload for hospital staff, who must manage phone calls and in-person booking requests.

Additionally, manual appointment booking systems are susceptible to human errors, which can result in double bookings, misplaced records, and scheduling conflicts. These errors lead to patient dissatisfaction, potential loss of revenue for hospitals, and challenges in managing patient flow effectively. The lack of a centralized and streamlined record-keeping system further complicates the management of patient appointments, making it difficult to track bookings and handle cancellations or rescheduling efficiently. Moreover, the limited booking options in traditional systems can be inconvenient for users who prefer online and mobile solutions.

These challenges underscore the need for a more efficient, reliable, and user-friendly solution to improve the appointment booking experience. By adopting an online hospital appointment booking application, these problems can be alleviated, ensuring faster scheduling, reducing errors, and providing a more convenient and modern approach to managing healthcare appointments.

### Proposed System

The proposed Online Hospital Appointment Booking Application is designed to enhance efficiency and user convenience by enabling secure online appointment scheduling and patient management. Users can register, book appointments with doctors, and access detailed reports of their medical history and upcoming appointments. The system automatically schedules appointments based on doctor availability and patient preferences.

Key features of the proposed system include user registration and authentication to ensure secure access and data protection. It incorporates a responsive design for compatibility across various devices and utilizes a robust architecture with a MySQL database for data storage, PHP for server-side processing, and HTML, CSS, and JavaScript for a user-friendly interface. This centralized platform will streamline appointment scheduling, reduce waiting times, and offer a secure, modern solution for hospital appointment management.

1. **Tool Description**

#### Hardware and Software Tools

To develop and deploy the online Hospital Appointment Booking web application, the following hardware and software tools were utilized:

**Hardware Specifications**

* **Laptop Model**: LENOVO LOQ
* **Graphics Card**: NVIDIA GeForce RTX 3050, 6GB
* **Storage**: 500GB SSD
* **RAM**: 16GB
* **Processor**: 12th Gen Intel(R) Core(TM) i5-12450H 2.00 GHz

**Software Tools**

* **Visual Studio Code**: An integrated development environment (IDE) used for writing and debugging code. Its extensions and integrated terminal enhanced the coding experience.
* **XAMPP**: A free and open-source cross-platform web server solution stack package developed by Apache Friends. It provided the necessary Apache, MySQL, PHP, and Perl support for local development and testing.
* **phpMyAdmin**: A free software tool written in PHP, intended to handle the administration of MySQL over the web. phpMyAdmin was used for database management, allowing for easy handling of the MySQL database used in the application.
* **Google Chrome**: The primary web browser used for testing and debugging the web application. Developer tools in Chrome facilitated real-time inspection and modification of the front-end code.

The combination of powerful hardware and a robust set of development tools provided a conducive environment for the efficient development, testing, and deployment of the recipe management web application.

1. **OPERATIONS**

The Hospital Appointment Booking Application provides various operations for users to manage appointments effectively and ensure a smooth user experience. Below are the detailed operations:

### 6.1 User Operations

**Interacting with Appointment Records**

* **View Appointment Records:** Users can browse and view detailed information about appointment records, including patient name, doctor name, appointment date, time, department, and consultation purpose.
* **Search Appointment Records:** Users can search for appointment records by patient name, doctor name, or appointment date using the search functionality.
* **Filter by Date:** Users can filter appointment records to display entries from a specific date, making it easier to find relevant appointments.

**Taking Actions on Appointment Records**

* **Book Appointment:** Users can book appointments through an online form by providing details such as patient name, doctor name, department, appointment date, and time. The records are then stored in the system's database for future reference and processing.
* **Cancel Appointment:** Users can cancel their appointments through the application by selecting the appointment they wish to cancel. This action updates the system's database accordingly.
* **Reschedule Appointment:** Users can reschedule their appointments by selecting a new date and time. The updated appointment details are then stored in the system's database.

### 6.2 Administrator Operations

#### Managing Appointment Records

* **Approve/Reject Appointment Requests:** Administrators can review appointment requests and either approve or reject them based on availability and other criteria. Approved appointments are confirmed and stored in the database, while rejected requests prompt users to choose another time.
* **View All Appointments:** Administrators have the ability to view all appointment records in the system, providing a comprehensive overview of scheduled consultations.
* **Search and Filter Records:** Administrators can search for and filter appointment records based on various criteria such as patient name, doctor name, department, and date. This feature helps in quickly locating and managing specific appointments.

#### System Management

* **Manage Users:** Administrators can manage user accounts by adding, updating, or deleting user information. This ensures that only authorized users have access to the system.
* **Generate Reports:** Administrators can generate detailed reports on appointment bookings, cancellations, and rescheduling activities. These reports aid in analyzing system usage and identifying areas for improvement.
* **Manage Doctor Schedules:** Administrators can update doctor schedules, ensuring that the system reflects the accurate availability of each doctor. This helps in preventing booking conflicts and ensuring efficient use of resources.

By structuring the operations around these roles, the Online Hospital Appointment Booking Application provides a seamless and efficient way for users to manage appointments and for administrators to maintain an organized and user-friendly appointment booking system.

1. **MODULE DESCRIPTION**

To develop the Hospital Appointment Booking Application, we will divide the project into distinct modules, each responsible for specific functionalities. By creating individual functions for every operation and unifying them, we can ensure modularity, maintainability, and scalability.

### Modules and Functionalities

#### 7.1 Interaction Module (User)

**Function: Home Page**

**Description:** Provides users with an introductory message about the online hospital appointment booking application.

**Functionalities:**

* Display a welcome message with brief information about the application.
* Include a call-to-action button to book an appointment.

**Function: About Us Page**

* **Description:** Provides detailed information about the online hospital appointment booking application and its mission.
* **Functionalities:**
* Display information about the purpose and benefits of the appointment booking application.
* Include background details and objectives.

**Function: Contact Form**

#### ****Description:**** Allows users to send inquiries or feedback through a contact form.

#### ****Functionalities:****

#### Input fields for name, email, subject, and message.

#### Validate and submit the contact form to the server.

#### ****Function: Search Appointment Records****

#### ****Description:**** Enables users to search for specific appointment records based on criteria such as patient name, doctor name, or appointment date.

#### ****Functionalities:****

#### Filter and display appointment records based on search input.

#### Dynamically update the search results as the user types or selects criteria.

### 7.2 Appointment Management Module (User)

#### ****Function: Book Appointment****

#### ****Description:**** Allows users to book appointments with doctors.

#### ****Functionalities:****

#### Input fields for patient name, doctor name, department, appointment date, and time.

#### Validate and submit the appointment booking form to the server.

#### Confirm and display the appointment details to the user.

#### ****Function: Cancel Appointment****

* **Description:** Enables users to cancel their appointments.
* **Functionalities:**
  + Display a list of upcoming appointments.
  + Allow users to select and cancel an appointment.
  + Update the system's database and confirm the cancellation to the user.

#### ****Function: Reschedule Appointment****

* **Description:** Allows users to reschedule their appointments.
* **Functionalities:**
  + Display a list of upcoming appointments.
  + Allow users to select an appointment and choose a new date and time.
  + Validate and submit the rescheduling request to the server.
  + Confirm and display the updated appointment details to the user.

### 7.3 Administrative Module (Administrator)

#### Function: Approve/Reject Appointment Requests

* **Description:** Enables administrators to review and approve or reject appointment requests.
* **Functionalities:**
  + Display a list of pending appointment requests.
  + Allow administrators to approve or reject each request.
  + Update the system's database accordingly and notify the users of the decision.

#### Function: Manage Doctor Schedules

* **Description:** Allows administrators to manage and update doctor schedules.
* **Functionalities:**
  + Display a list of doctors and their schedules.
  + Allow administrators to update the availability of each doctor.
  + Ensure the system reflects the accurate availability of doctors to prevent booking conflicts.

#### ****Function: Generate Reports****

* **Description:** Enables administrators to generate detailed reports on appointment bookings, cancellations, and rescheduling activities.
* **Functionalities:**
  + Select criteria for generating reports (e.g., date range, department, doctor).
  + Generate and display the report based on selected criteria.
  + Provide options to export the report in various formats (e.g., PDF, Excel).

1. **IMPLEMENTATION**

**Coding**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Hospital Management System</title>

    <link rel="stylesheet" href="style1.css">

    <script src="script.js" defer></script>

    <style>

        body {

            font-family: Arial, sans-serif;

        }

        header {

            text-align: center;

            padding: 20px;

            background: #007BFF;

            color: white;

        }

        nav ul {

            display: flex;

            justify-content: center;

            list-style-type: none;

            background: #333;

            padding: 0;

        }

        nav ul li {

            margin: 0 15px;

        }

        nav ul li a {

            color: white;

            text-decoration: none;

            padding: 10px 20px;

            display: block;

        }

        nav ul li a:hover {

            background: #575757;

        }

        main {

            padding: 20px;

        }

        .form-section {

            max-width: 500px;

            margin: 20px auto;

        }

        .form-group {

            margin-bottom: 15px;

        }

        .form-group label {

            display: block;

            margin-bottom: 5px;

        }

        .form-group input,

        .form-group select,

        .form-group textarea {

            width: 100%;

            padding: 8px;

            box-sizing: border-box;

        }

        footer {

            text-align: center;

            padding: 10px;

            background: #007BFF;

            color: white;

            position: fixed;

            width: 100%;

            bottom: 0;

        }

    </style>

</head>

<body>

    <!-- Login Form -->

    <div id="login-form" class="form-section">

        <h1>Login</h1>

        <form id="login" onsubmit="return authenticate(event)">

            <div class="form-group">

                <label for="username">Username:</label>

                <input type="text" id="username" name="username" required>

            </div>

            <div class="form-group">

                <label for="password">Password:</label>

                <input type="password" id="password" name="password" required>

            </div>

            <input type="submit" value="Login">

        </form>

    </div>

    <!-- Frontend Page -->

    <div id="main-content" style="display: none;">

        <header>

            <h1>Hospital Management System</h1>

            <p>Welcome, <span id="user-display"></span> | <a href="#" onclick="logout()">Logout</a></p>

        </header>

        <nav>

            <ul>

                <li><a href="#home" onclick="showSection('home')">Home</a></li>

                <li><a href="#patients" onclick="showSection('patients')">Patients</a></li>

                <li><a href="#doctors" onclick="showSection('doctors')">Doctors</a></li>

                <li><a href="#appointments" onclick="showSection('appointments')">Appointments</a></li>

                <li><a href="#billing" onclick="showSection('billing')">Billing</a></li>

                <li><a href="#contact" onclick="showSection('contact')">Contact</a></li>

            </ul>

        </nav>

        <main>

            <!-- Home Section -->

            <section id="home" class="section-content">

                <h2>Welcome to Our Hospital</h2>

                <p>We provide the best medical care.</p>

            </section>

            <!-- Patients Section -->

            <section id="patients" class="section-content">

                <h2>Patient Information</h2>

                <form id="patients-form" class="form-section" action="process\_form.php" method="POST">

                    <input type="hidden" name="form\_type" value="patients">

                    <div class="form-group">

                        <label for="patient-name">Name:</label>

                        <input type="text" id="patient-name" name="patient-name" required>

                    </div>

                    <div class="form-group">

                        <label for="patient-age">Age:</label>

                        <input type="number" id="patient-age" name="patient-age" required>

                    </div>

                    <div class="form-group">

                        <label for="patient-gender">Gender:</label>

                        <select id="patient-gender" name="patient-gender" required>

                            <option value="male">Male</option>

                            <option value="female">Female</option>

                            <option value="other">Other</option>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="patient-address">Address:</label>

                        <textarea id="patient-address" name="patient-address" rows="4" required></textarea>

                    </div>

                    <input type="submit" value="Submit">

                </form>

            </section>

            <!-- Doctors Section -->

            <section id="doctors" class="section-content">

                <h2>Doctor Information</h2>

                <form id="doctors-form" class="form-section" action="process\_form.php" method="POST">

                    <input type="hidden" name="form\_type" value="doctors">

                    <div class="form-group">

                        <label for="doctor-name">Name:</label>

                        <input type="text" id="doctor-name" name="doctor-name" required>

                    </div>

                    <div class="form-group">

                        <label for="doctor-specialty">Specialty:</label>

                        <input type="text" id="doctor-specialty" name="doctor-specialty" required>

                    </div>

                    <div class="form-group">

                        <label for="doctor-phone">Phone:</label>

                        <input type="tel" id="doctor-phone" name="doctor-phone" required>

                    </div>

                    <div class="form-group">

                        <label for="doctor-email">Email:</label>

                        <input type="email" id="doctor-email" name="doctor-email" required>

                    </div>

                    <input type="submit" value="Submit">

                </form>

            </section>

            <!-- Appointments Section -->

            <section id="appointments" class="section-content">

                <h2>Appointments</h2>

                <form id="appointments-form" class="form-section" action="process\_form.php" method="POST">

                    <input type="hidden" name="form\_type" value="appointments">

                    <div class="form-group">

                        <label for="appointment-date">Date:</label>

                        <input type="date" id="appointment-date" name="appointment-date" required>

                    </div>

                    <div class="form-group">

                        <label for="appointment-time">Time:</label>

                        <input type="time" id="appointment-time" name="appointment-time" required>

                    </div>

                    <div class="form-group">

                        <label for="doctor-select">Select Doctor:</label>

                        <select id="doctor-select" name="doctor-select" required>

                            <option value="dr-smith">Dr. Smith</option>

                            <option value="dr-jones">Dr. Jones</option>

                            <option value="dr-doe">Dr. Doe</option>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="patient-id">Patient ID:</label>

                        <input type="text" id="patient-id" name="patient-id" required>

                    </div>

                    <input type="submit" value="Submit">

                </form>

            </section>

            <!-- Billing Section -->

            <section id="billing" class="section-content">

                <h2>Billing</h2>

                <form id="billing-form" class="form-section" action="process\_form.php" method="POST">

                    <input type="hidden" name="form\_type" value="billing">

                    <div class="form-group">

                        <label for="billing-amount">Amount:</label>

                        <input type="number" id="billing-amount" name="billing-amount" required>

                    </div>

                    <div class="form-group">

                        <label for="billing-method">Payment Method:</label>

                        <select id="billing-method" name="billing-method" required>

                            <option value="credit">Credit Card</option>

                            <option value="debit">Debit Card</option>

                            <option value="cash">Cash</option>

                        </select>

                    </div>

                    <div class="form-group">

                        <label for="billing-notes">Notes:</label>

                        <textarea id="billing-notes" name="billing-notes" rows="4"></textarea>

                    </div>

                    <input type="submit" value="Submit">

                </form>

            </section>

            <!-- Contact Section -->

            <section id="contact" class="section-content">

                <h2>Contact Us</h2>

                <form id="contact-form" class="form-section" action="process\_form.php" method="POST">

                    <input type="hidden" name="form\_type" value="contact">

                    <div class="form-group">

                        <label for="contact-name">Name:</label>

                        <input type="text" id="contact-name" name="contact-name" required>

                    </div>

                    <div class="form-group">

                        <label for="contact-email">Email:</label>

                        <input type="email" id="contact-email" name="contact-email" required>

                    </div>

                    <div class="form-group">

                        <label for="contact-message">Message:</label>

                        <textarea id="contact-message" name="contact-message" rows="4" required></textarea>

                    </div>

                    <input type="submit" value="Submit">

                </form>

            </section>

        </main>

    </div>

    <footer>

        <p>&copy; 2024 Hospital Management System</p>

    </footer>

    <script>

        function authenticate(event) {

            event.preventDefault();

            const username = document.getElementById('username').value;

            const password = document.getElementById('password').value;

            if (username === 'admin' && password === 'password') { // Example credentials

                document.getElementById('login-form').style.display = 'none';

                document.getElementById('main-content').style.display = 'block';

                document.getElementById('user-display').textContent = username;

            } else {

                alert('Invalid username or password');

            }

        }

        function logout() {

            document.getElementById('login-form').style.display = 'block';

            document.getElementById('main-content').style.display = 'none';

        }

        function showSection(sectionId) {

            document.querySelectorAll('.section-content').forEach(section => {

                section.style.display = 'none';

            });

            document.getElementById(sectionId).style.display = 'block';

        }

        document.addEventListener('DOMContentLoaded', () => {

            showSection('home');

        });

    </script>

</body>

</html>

1. **RESULT**

The implementation of the Online Hospital Appointment Booking Application successfully streamlined the appointment scheduling process, providing users with a convenient and efficient method to manage appointments and patient information. The system's automated scheduling feature based on doctor availability reduced the likelihood of errors and ensured consistency in appointment bookings. The user-friendly interface, along with secure registration and login functionalities, enhanced user experience and security. Compatibility across various devices ensured accessibility for a broader user base. Overall, the application demonstrated improved operational efficiency, reduced waiting times at hospitals, and offered a modern, reliable solution for appointment management.

1. **CONCLUSION**

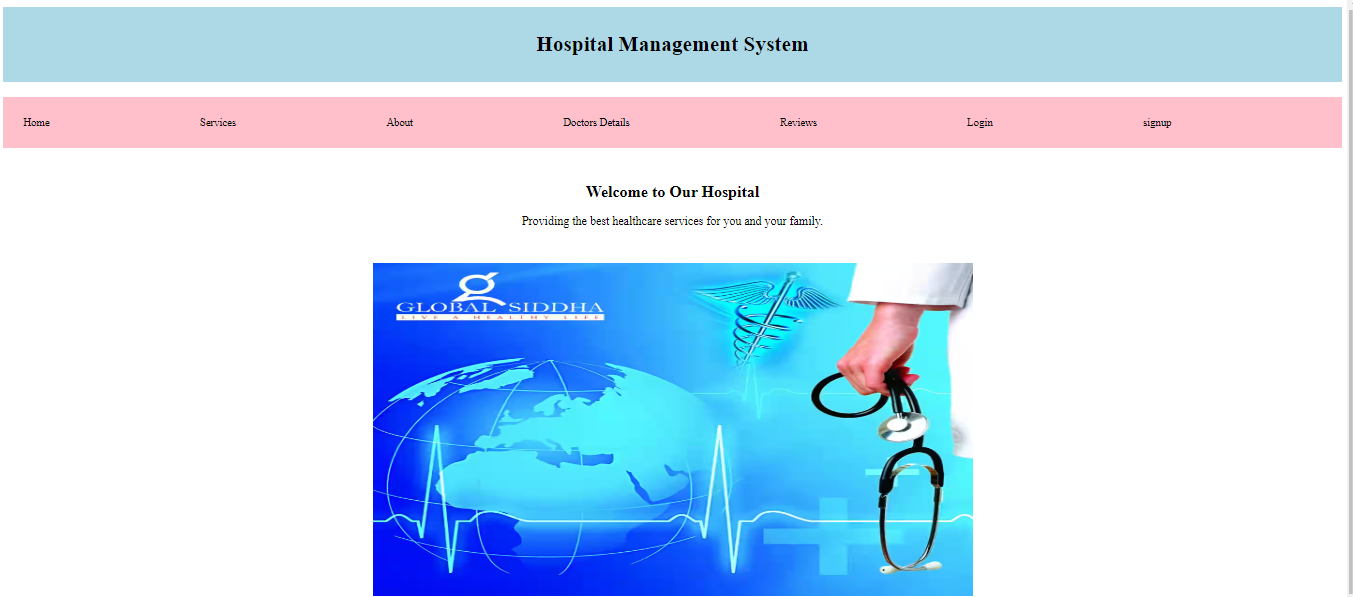
The "Online Hospital Appointment Booking Application" is a comprehensive web platform designed to simplify the appointment scheduling process for users, enabling them to manage medical appointments with ease and efficiency. Developed with a user-friendly interface, the application ensures a seamless experience for booking appointments, selecting healthcare providers, and accessing detailed records of past and upcoming visits. By offering features such as real-time scheduling, automated notifications, and secure patient information management, the application significantly reduces the time and effort associated with traditional appointment booking methods, thereby enhancing overall user convenience, satisfaction, and access to healthcare services.

1. **FUTURE ENHANCEMENTS**

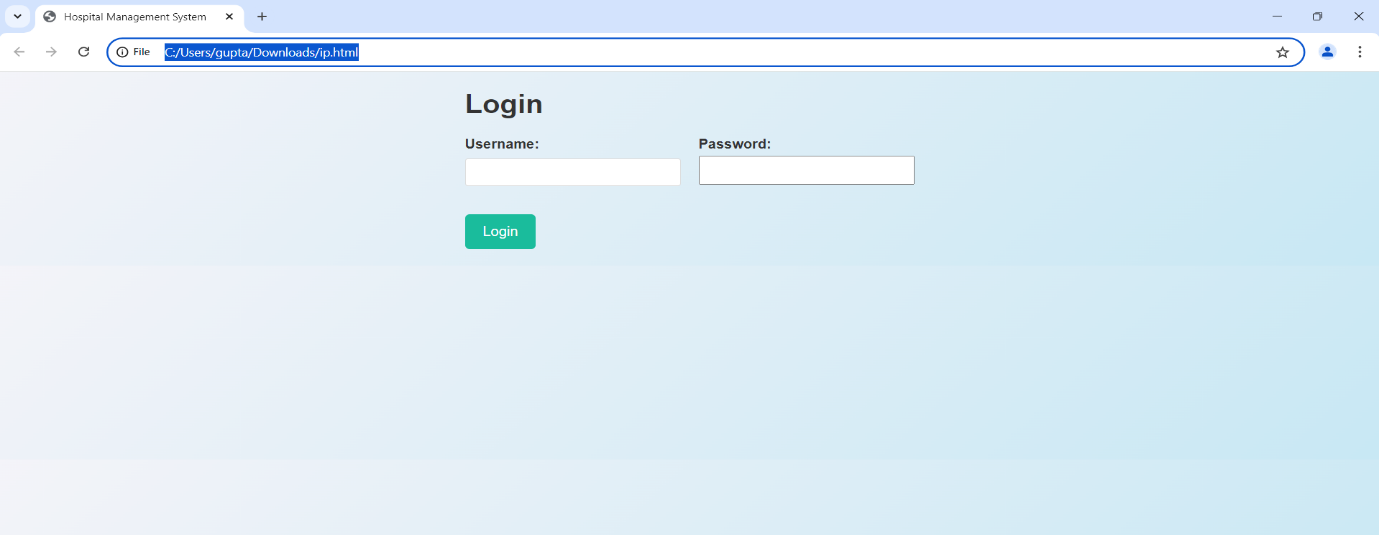
One major improvement is the integration of advanced search and filtering options. Additionally, implementing real-time appointment availability updates and notifications can keep users informed about any changes or cancellations, enhancing the application's utility.

Another significant enhancement is the incorporation of features to boost user interaction and patient engagement. Integrating the application with medicine services can provide users with options for virtual consultations, making healthcare more accessible. Furthermore, implementing a secure and efficient payment gateway with multiple payment options can enhance transaction security and user trust.

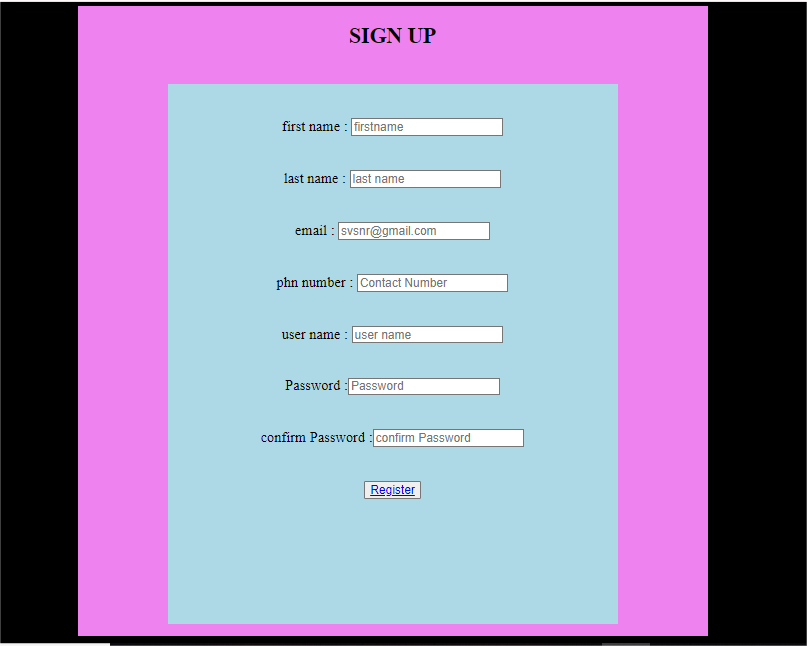
1. **SCREENSHOTS**

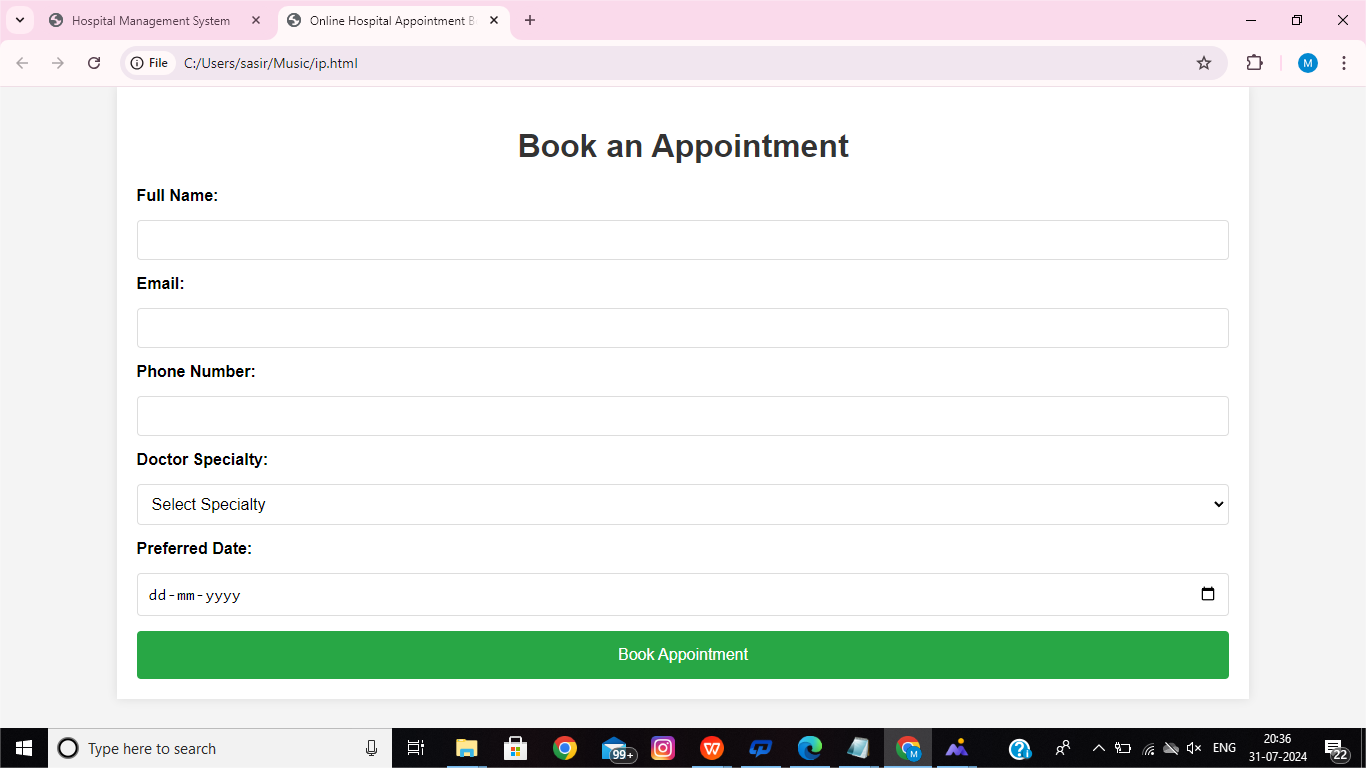
****

**Fig. 1: Home Page**

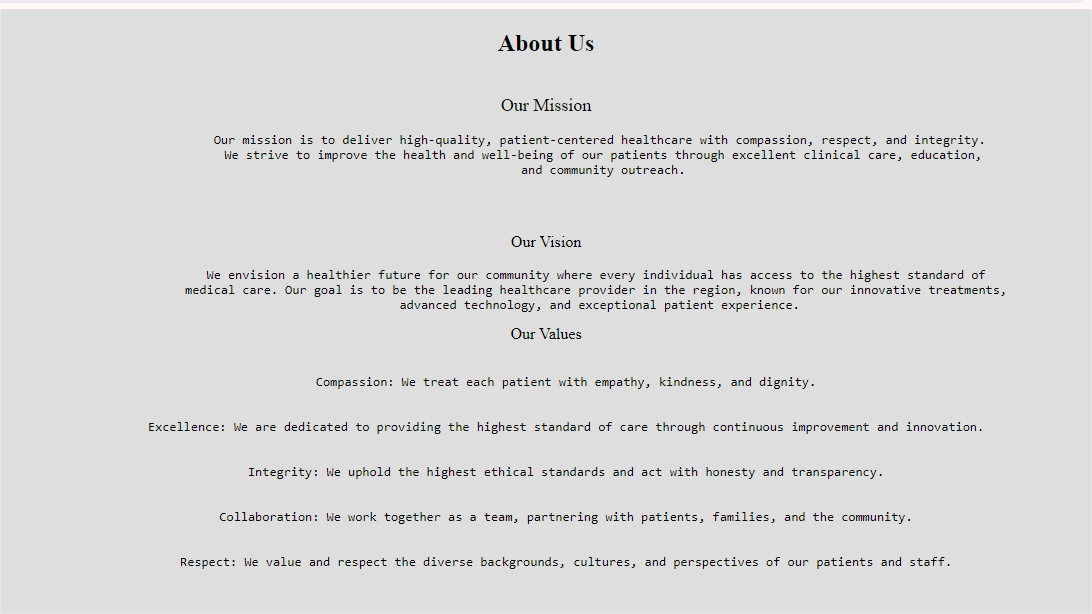
****

**Fig. 2: Login Page**

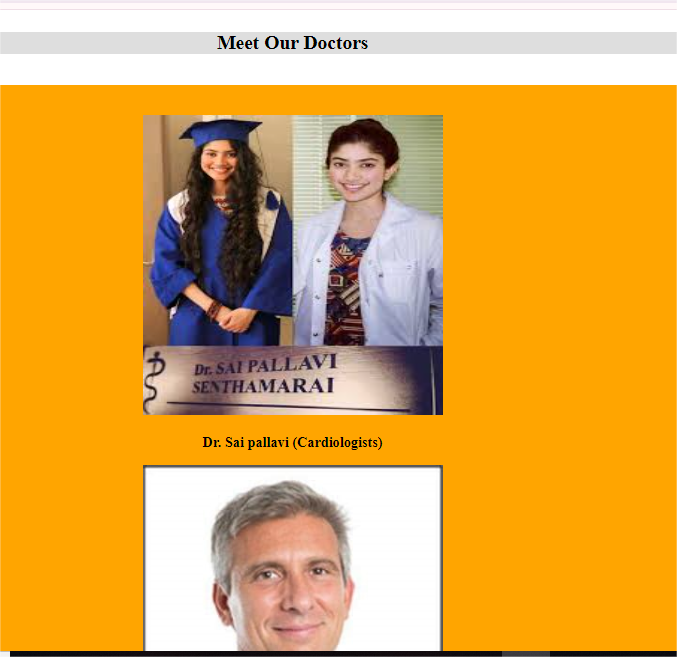
** Fig. 3: Sign-up page**

****

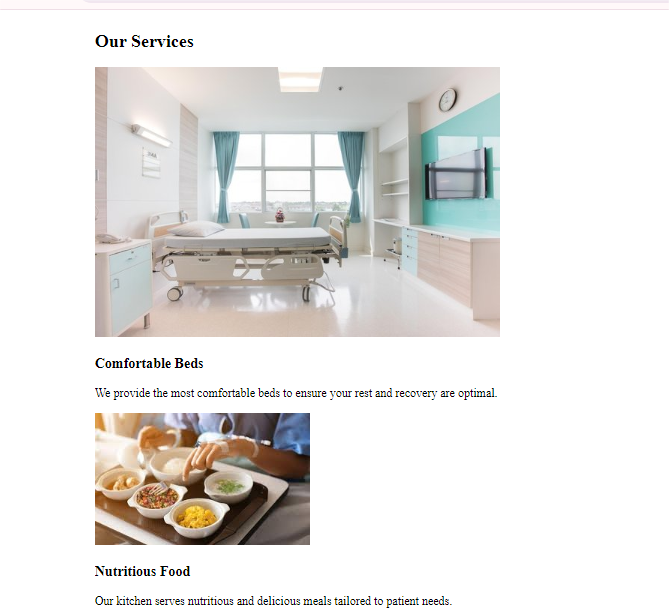
**Fig. 4: Booking page**

****

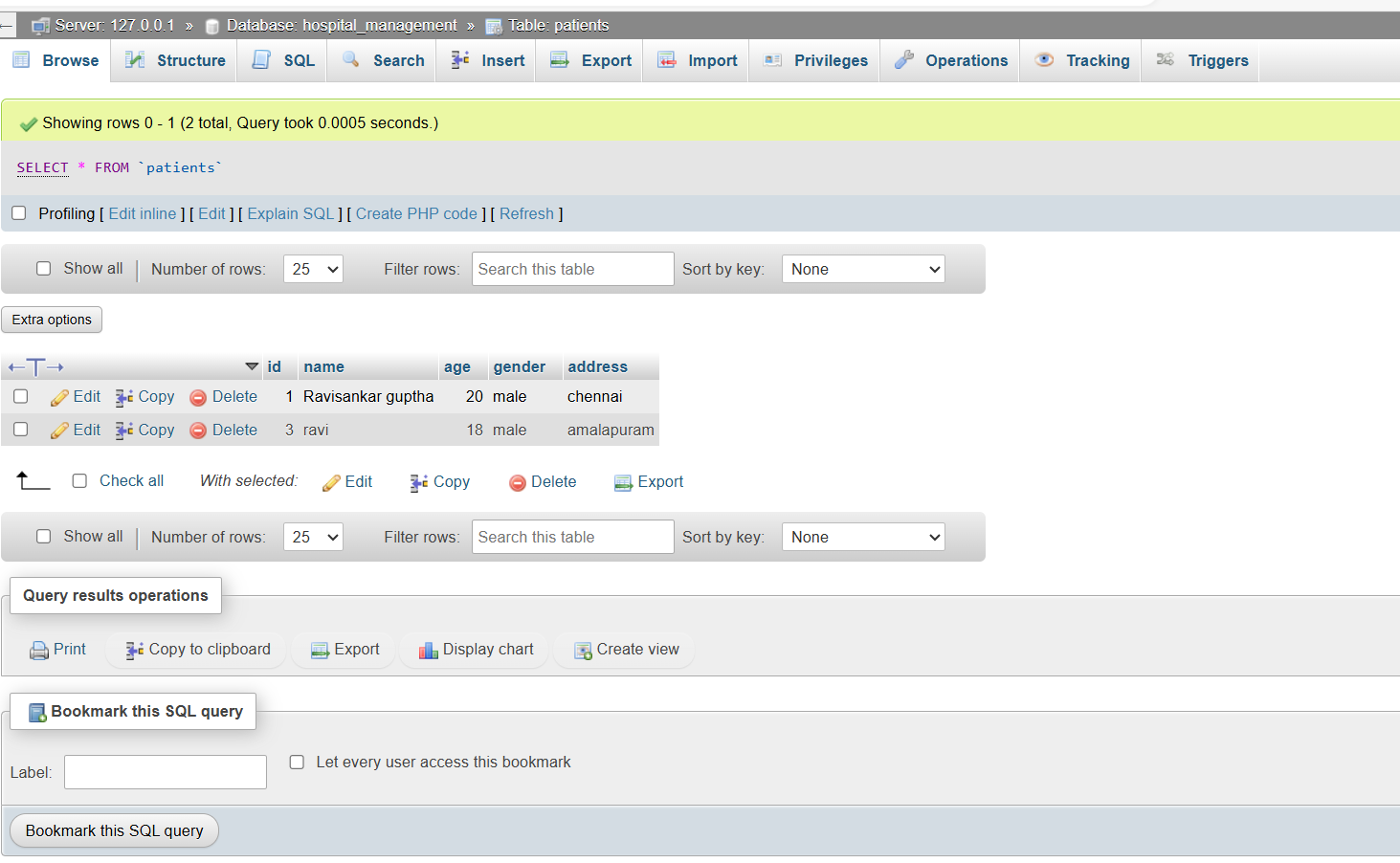
**Fig. 5: About page**

****

**Fig. 6: Doctors Page**

****

**Fig. 7: Services Page**



**Fig. 8: Report Page**

1. **References**

[1] Full stack Vue: The Complete Guide to Vue.js by Hassan Djirdeh, Nate Murray, and Ari Lerner

1. Full-Stack React, TypeScript, and Node: Build Cloud-Ready Web Applications Using React 17 with Hooks and GraphQL by David Choi

[3] Full-Stack Web Development with React and Node.js" by Eric Bishard Digital Ocean

 A Practical Guide to Node.js and React" by Vasan Subramanian SitePoint

 Full-Stack Web Development with React Specialization" by The Hong Kong Universityof Science and Technology on [Coursera](https://www.coursera.org/specializations/full-stack-react)

[4] Full-Stack Web Development with Node.js, Express, and MongoDB" by Colt Steele on Udemy.

[5] Full stack Vue: The Complete Guide to Vue.js by Hassan Djirdeh, Nate Murray, and Ari Lerner