

*Green University of Bangladesh*

*Department of Computer Science and Engineering (CSE)*

*Semester: (Fall, Year: 2024), B.Sc. in CSE (Evening)*

Online News Portal

*Course Title: Web Programming Lab*

*Course Code: CSE 302*

*Section: 223 E2*

Students Details

|  |  |
| --- | --- |
| Name | ID |
| MD. ABU BAKKAR | 223015022 |

*Submission Date: 20-12-2024*

*Course Teacher’s Name: Umme Habiba*

*Designation: Lecturer*

[For teachers use only: Don’t write anything inside this box]

|  |
| --- |
| Lab Project Status |
| Marks: Signature:  Comments: Date: |

Contents

[1 Introduction 3](#_Toc10555)

[1.1 Overview 3](#_Toc10556)

[1.1.1 Key Features: 3](#_Toc10557)

[1.2 Motivation 3](#_Toc10558)

[1.3 Design Goals/Objectives 4](#_Toc10559)

[1.4 Application 4](#_Toc10560)

[1.4.1 User Interface: 4](#_Toc10561)

[1.4.2 Application Workflow 4](#_Toc10562)

[2 Design/Development/Implementation of the Project 5](#_Toc10563)

[2.1 Introduction 5](#_Toc10564)

[2.2 Project Details 5](#_Toc10565)

[2.2.1 Objective: 5](#_Toc10566)

[2.2.2 Scope: 5](#_Toc10567)

[2.2.3 Technologies Used: 5](#_Toc10568)

[2.3 Implementation 6](#_Toc10569)

[2.3.1 Development Environment Setup: 6](#_Toc10570)

[2.3.2 Project Structure: 6](#_Toc10571)

[2.4 Github Link 6](#_Toc10572)

[3 Performance Evaluation 42](#_Toc10573)

[3.1 Simulation Environment/ Simulation Procedure 42](#_Toc10574)

[3.1.1 Simulation Environment: 42](#_Toc10575)

[3.1.2 Simulation Procedure: 42](#_Toc10576)

[3.2 Results Analysis 42](#_Toc10577)

[3.3 Results Overall Discussion 43](#_Toc10578)

[3.3.1 Result\_portion 43](#_Toc10579)

[4 Conclusion 50](#_Toc10580)

[4.1 Discussion 50](#_Toc10581)

[4.2 Performance Evaluation 50](#_Toc10582)

[4.3 Limitations 50](#_Toc10583)

[4.4 Scope of Future Work 50](#_Toc10584)

[4.5 References 51](#_Toc10585)

Chapter 1

# Introduction

## 1.1 Overview

The proposed online fullstack news portal is a dynamic web application designed to deliver real-time news updates to users. Built with a robust combination of modern web technologies such as HTML, Tailwind CSS, PHP, MySQL, and JavaScript, the platform ensures a seamless user experience, responsiveness, and efficient data management.

The project leverages php on the backend, HTML, CSS, and JavaScript and tailwindcss and bootstrap for the frontend, and mariadb as the database.

### 1.1.1 Key Features:

* News Management: Add, update, view, and delete News records.
* Dan overview on details.
* Session management
* Efficiency: Improve the administrative workflow and reduce the manual efforts
* Interactivity: Provide a smooth user interface with dynamic interaction using JavaScript and php.

Chapter 2

# Design/Development/Implementation of the Project

## 2.1 Introduction

This section outlines the design, development, and implementation of the online news portal System, including technologies used, setup details, and project structure.

## 2.2 Project Details

### 2.2.1 Objective:

### The objective of the online fullstack news portal is to provide users with real-time, easily accessible news content through a dynamic and responsive platform. Utilizing HTML, Tailwind CSS, PHP, MySQL, and JavaScript, the portal aims to offer an intuitive user interface, secure authentication, and efficient content management. Key goals include delivering categorized and filtered news, enabling interactive features like comments and social sharing, and ensuring seamless performance across devices. The portal also focuses on scalability, allowing for future enhancements such as personalized recommendations, multilingual support, and integration with external APIs or mobile platforms.

### 2.2.2 Scope:

* **User Interface**:
* **Content Management**:
* Real-Time News Updates.
* Database Management

### 2.2.3 Technologies Used:

 **Frontend:**

* **HTML**: The foundation for structuring the web pages, providing the layout and content of the portal.
* **Tailwind CSS**: A utility-first CSS framework that allows for fast and responsive design, ensuring a modern and consistent look across devices.
* **JavaScript**: Adds interactivity to the portal, enabling features like dynamic search, filtering, and real-time content updates (via AJAX or WebSocket).

 **Backend:**

* **PHP**: Server-side scripting language that handles user authentication, content management, and dynamic generation of news articles.
* **MySQL**: Relational database management system for storing and managing data such as articles, categories, user information, and comments.

 **Additional Technologies:**

* **AJAX**: For asynchronous data fetching, enabling real-time updates without page reloads.
* **WebSocket (Optional)**: For live news feeds and real-time notifications.
* **JSON**: Used for exchanging data between the server and frontend, particularly in API integration and dynamic content loading.

## 2.3 Implementation

### 2.3.1 Development Environment Setup:

* Install php and xampp.
* A browser
* Install npm
* Installing Tailwind css
* Use Git for version control and collaboration.

### 2.3.2 Project Structure:

* /index.php: Root folder.
* /admin.php: Contains php project files (tables, dashboard, crud, etc.)
* /frontend: Contains HTML, CSS, and JavaScript, tailwindcss files
* /templates: HTML files
* /static: CSS, JavaScript, and image files

## 2.4 Github Link

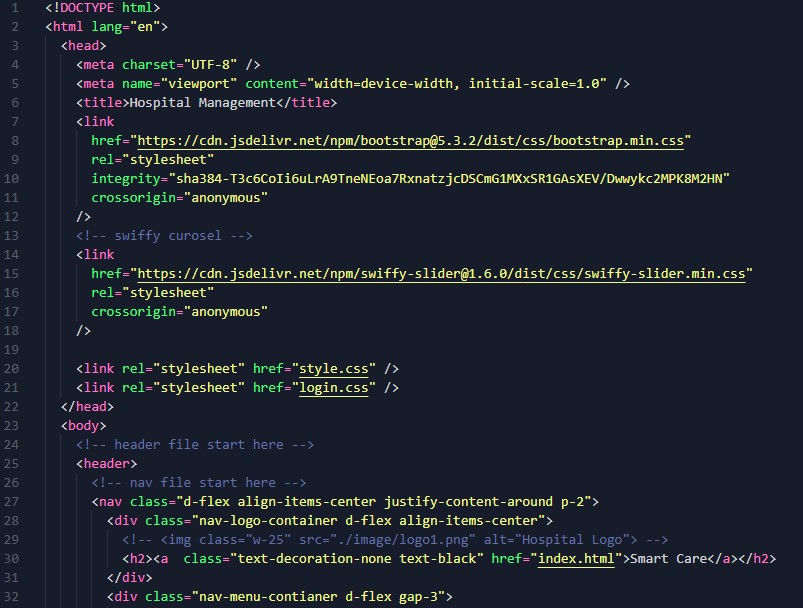
* Github Link Here in source code(Backend): [click here](https://github.com/abubakkar-cse/weblabfinal)

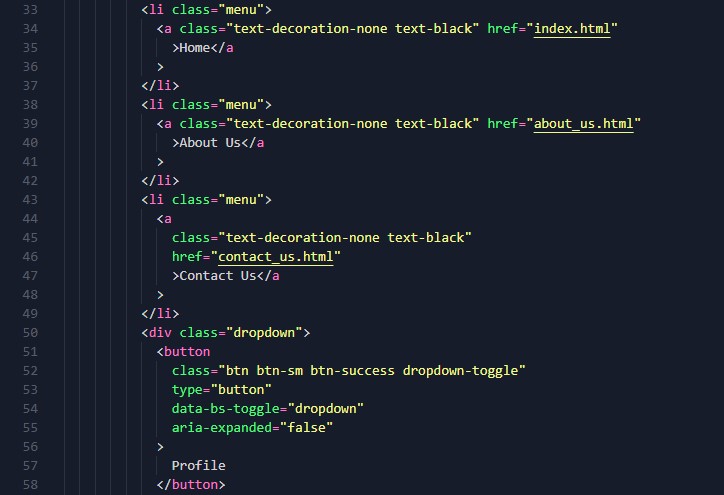
The workflow

Tools and libraries

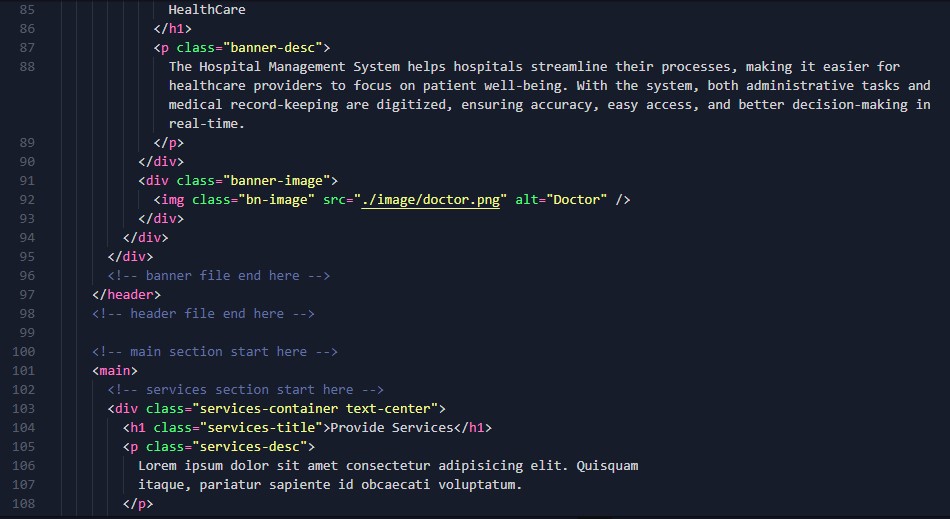
Implementation details (with screenshots and programming codes)

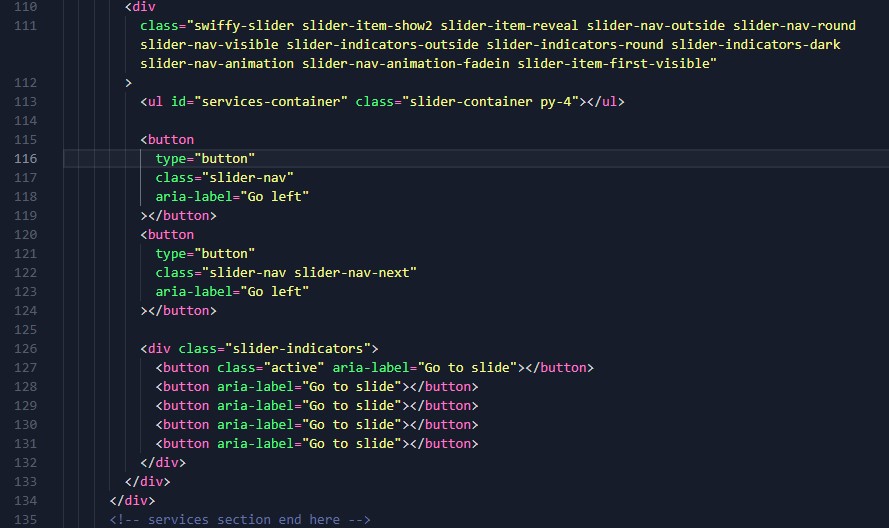
* index.php



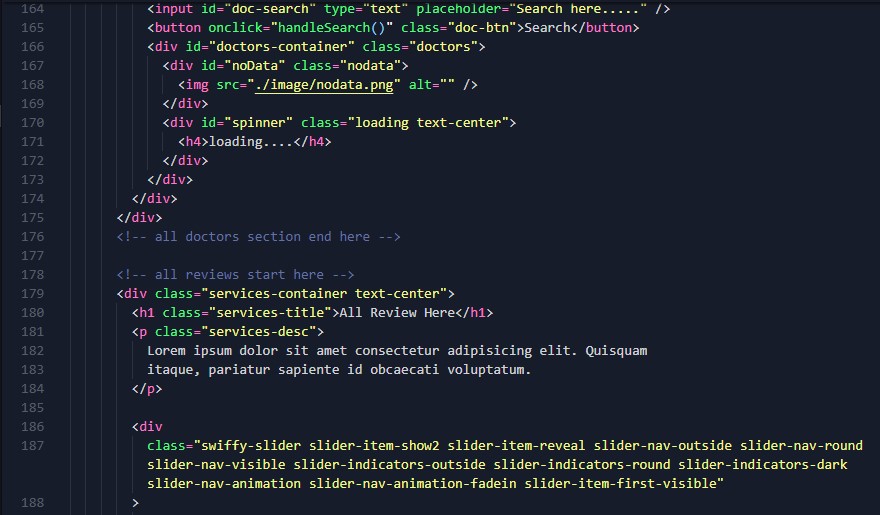




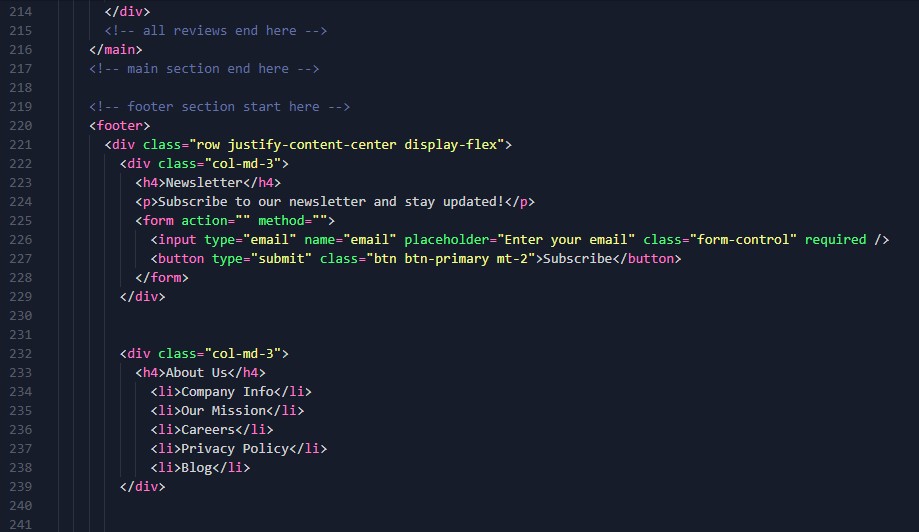


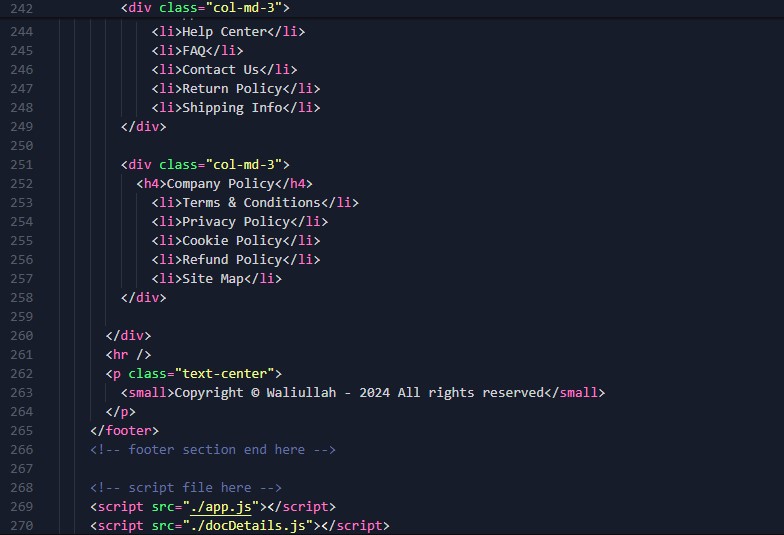


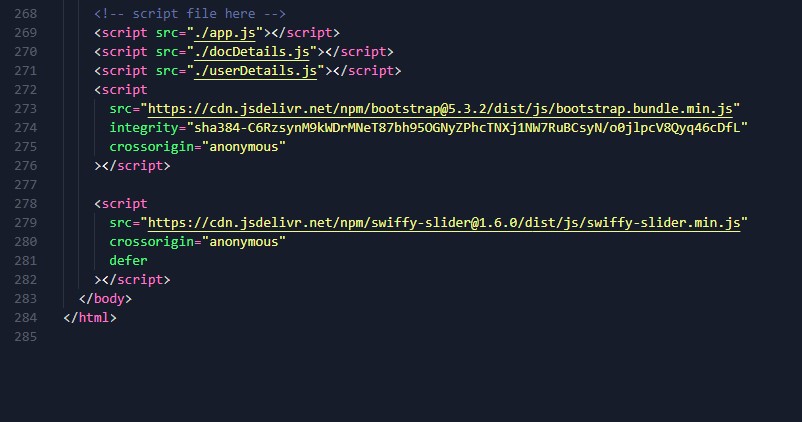




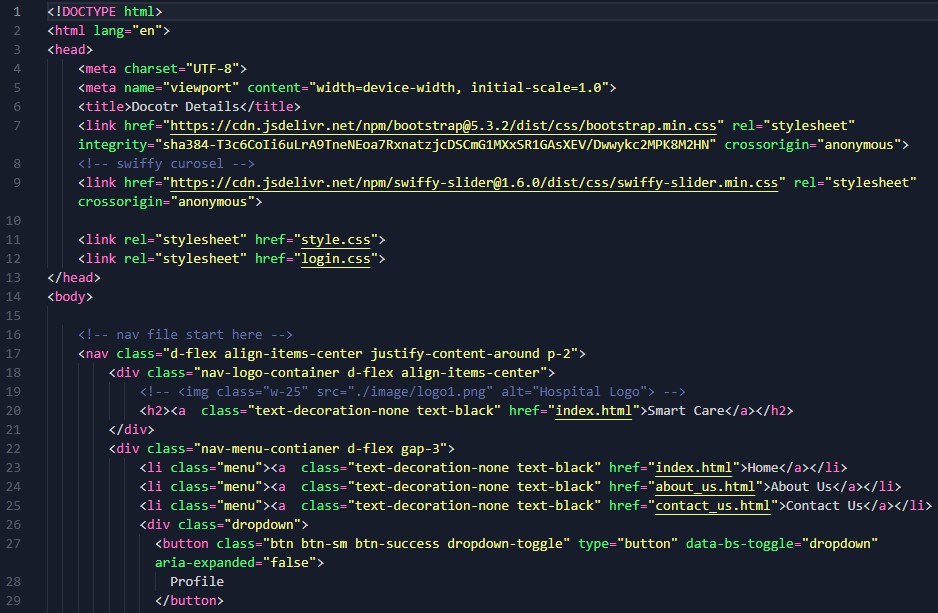


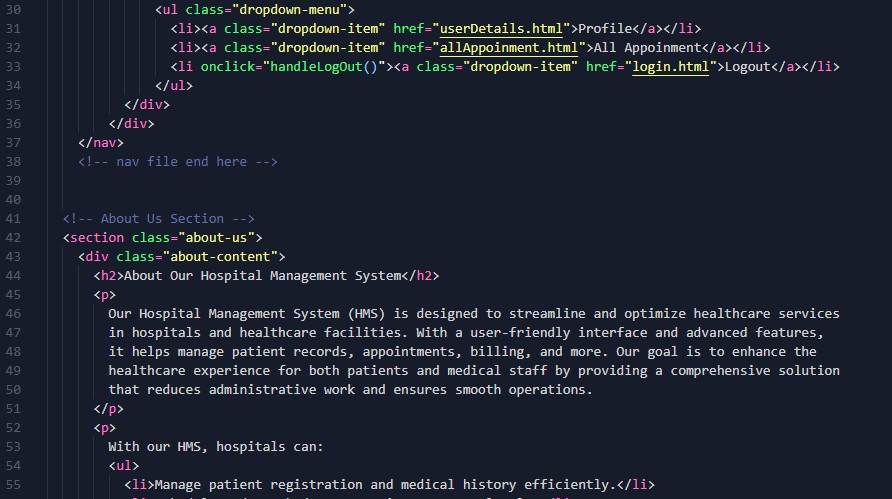


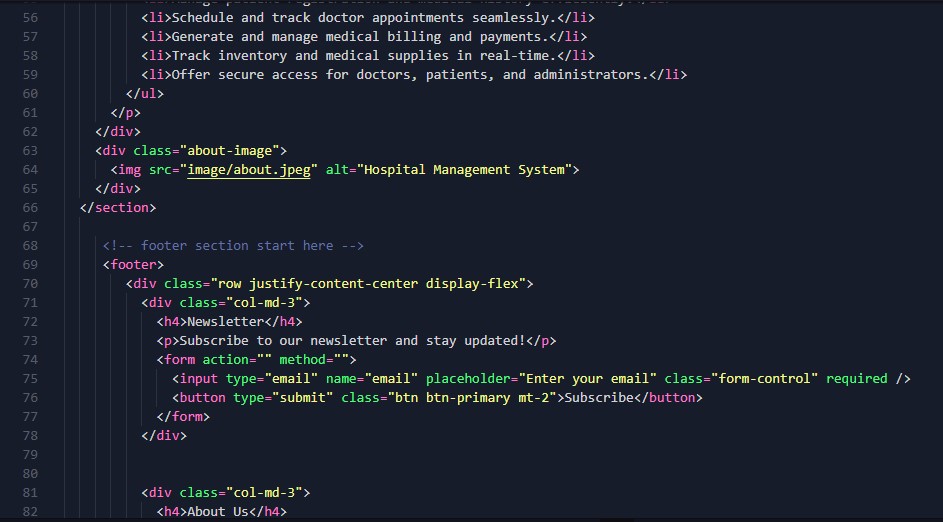


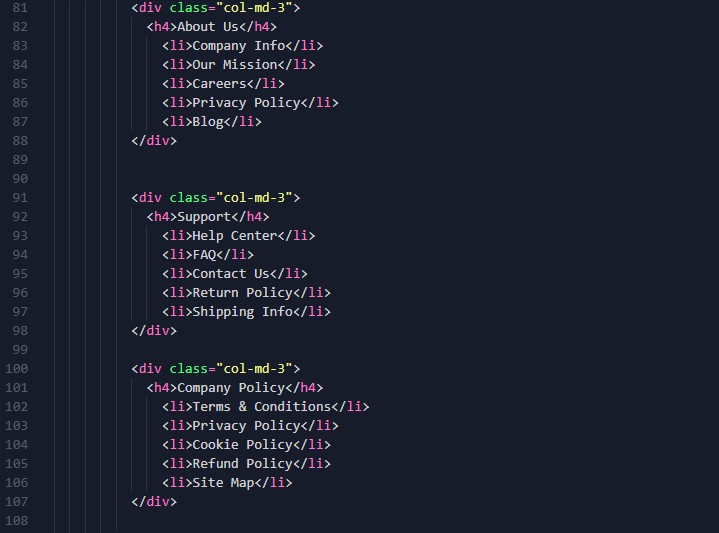


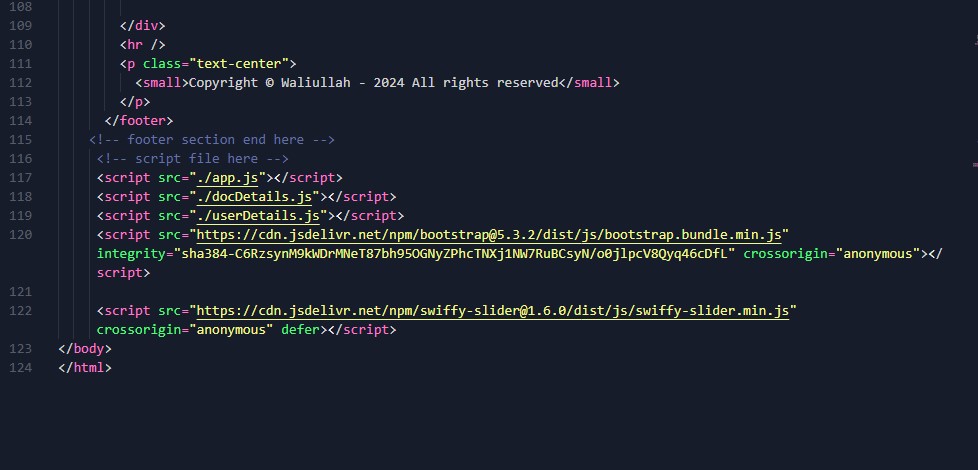
* about us.html



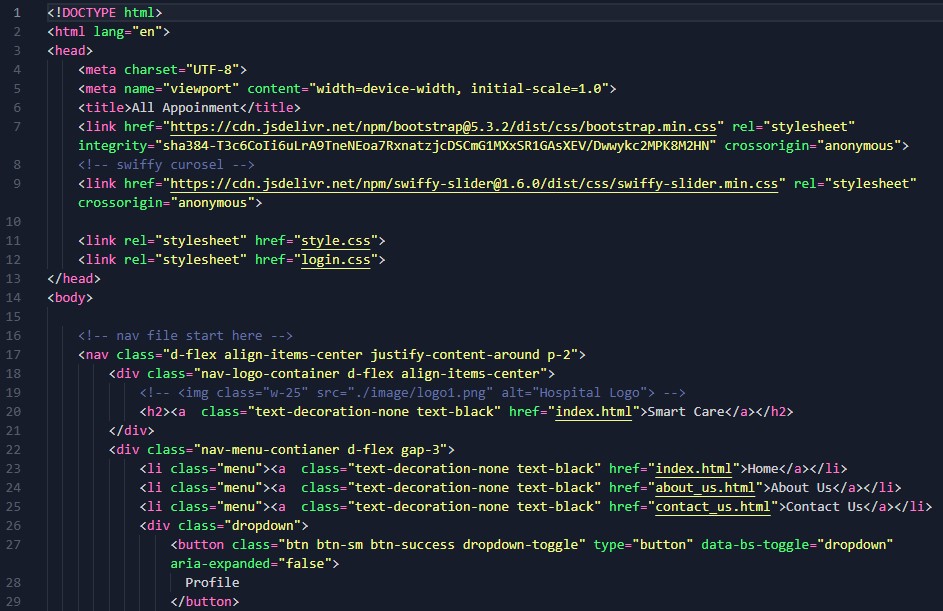


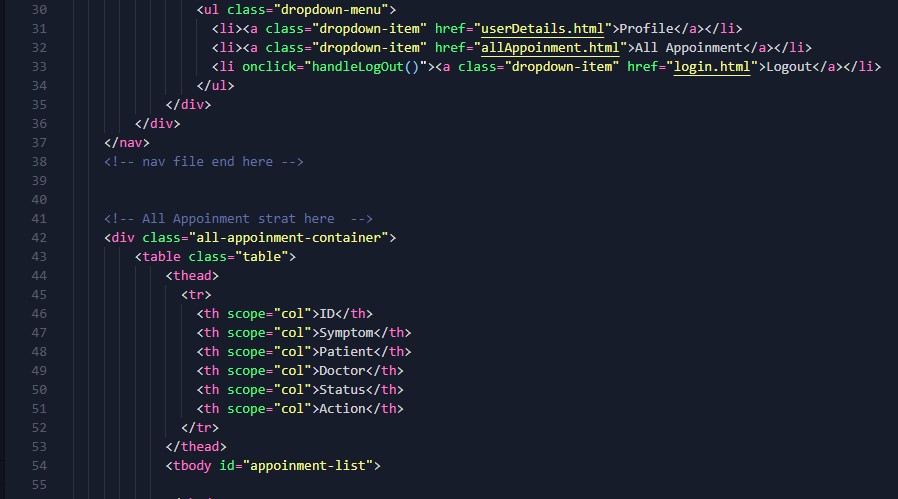


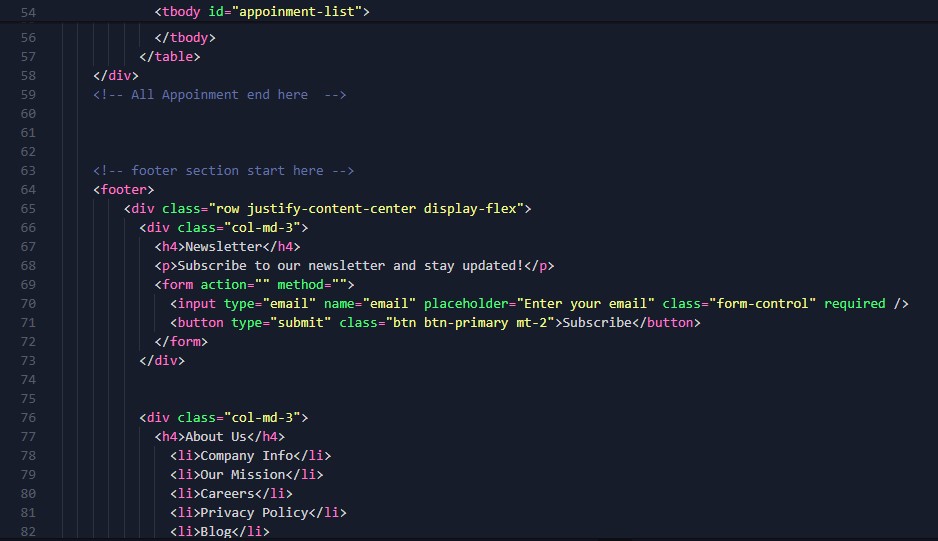




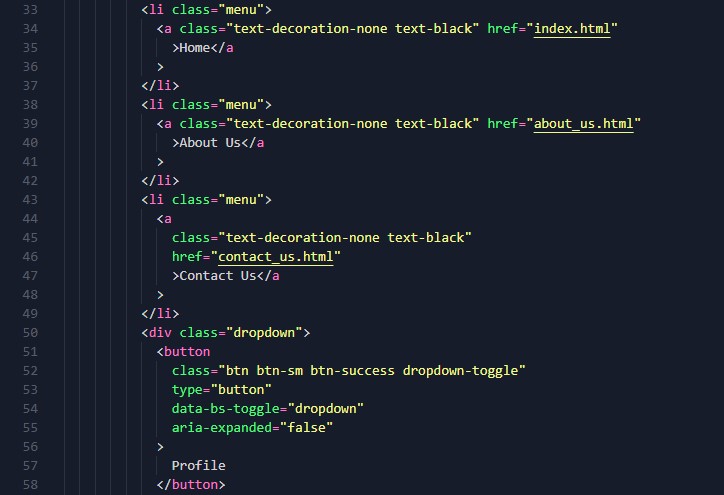
* allAppoinment.html



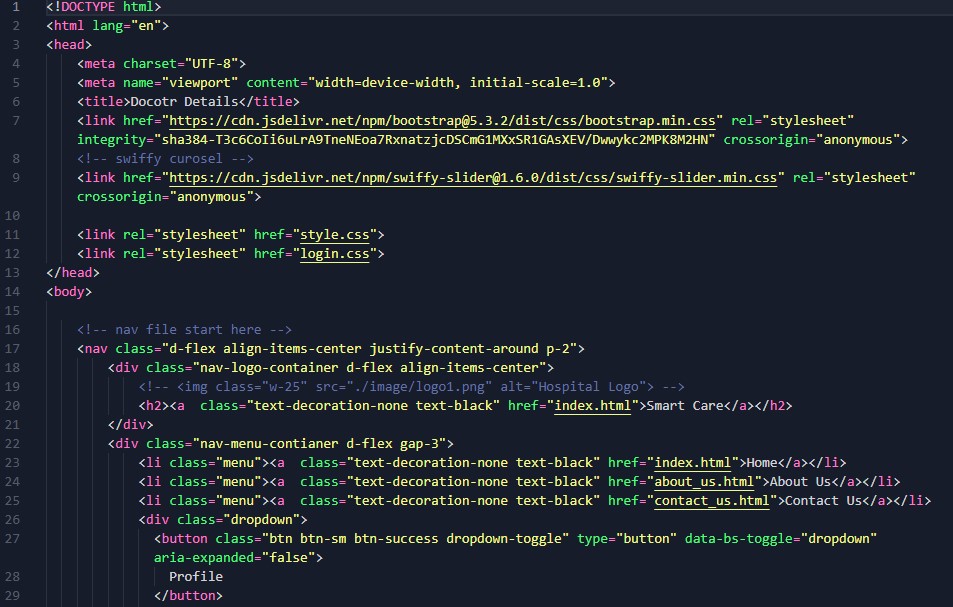


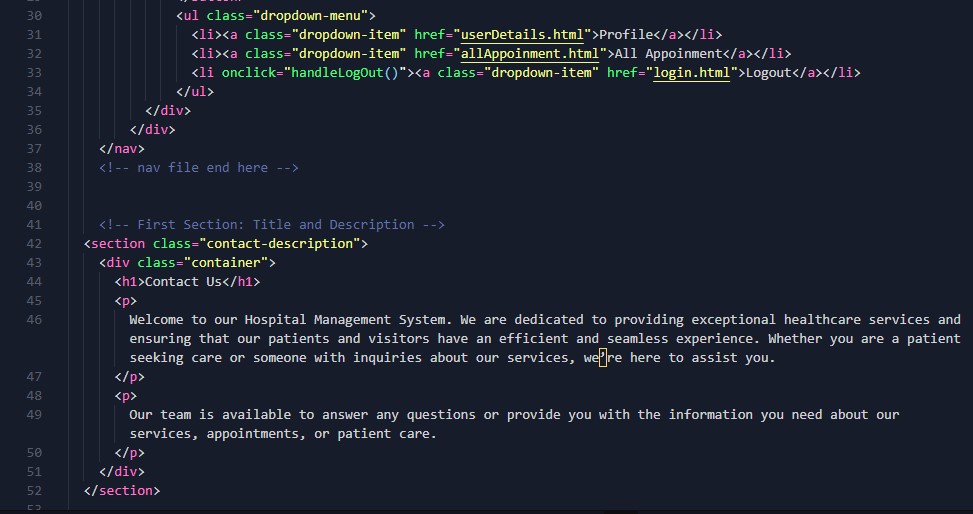


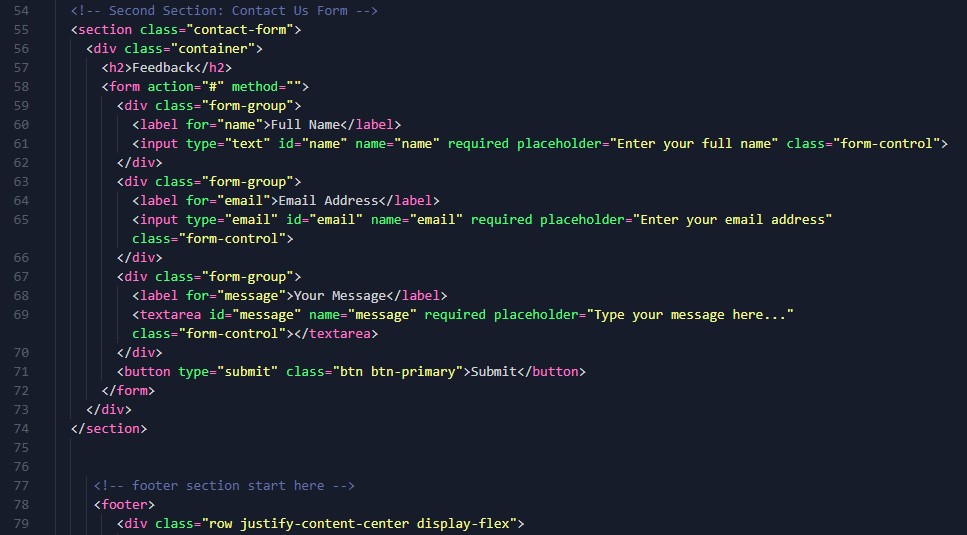


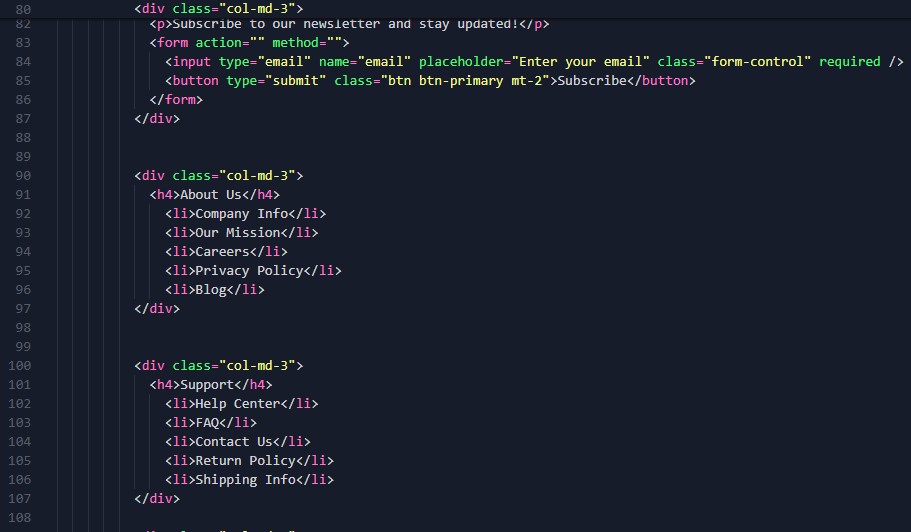


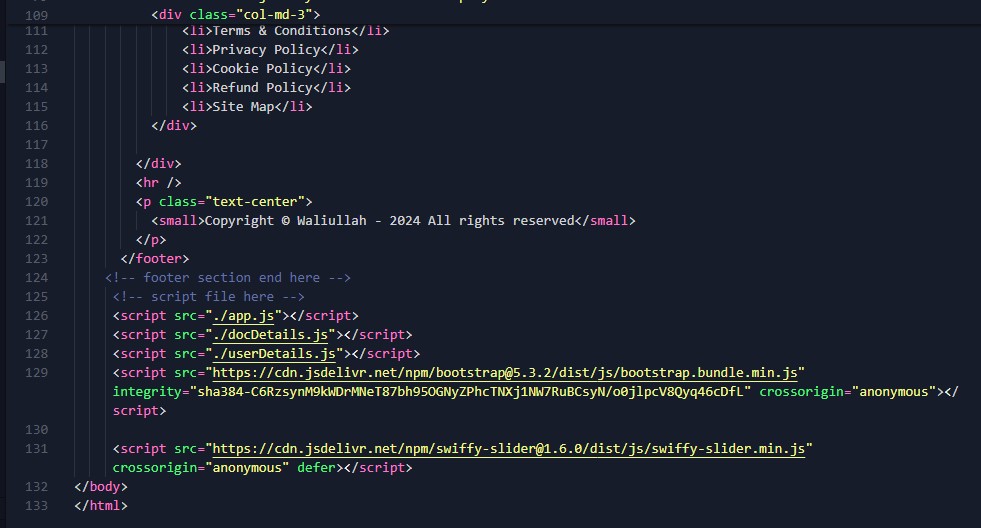
* contact us.html



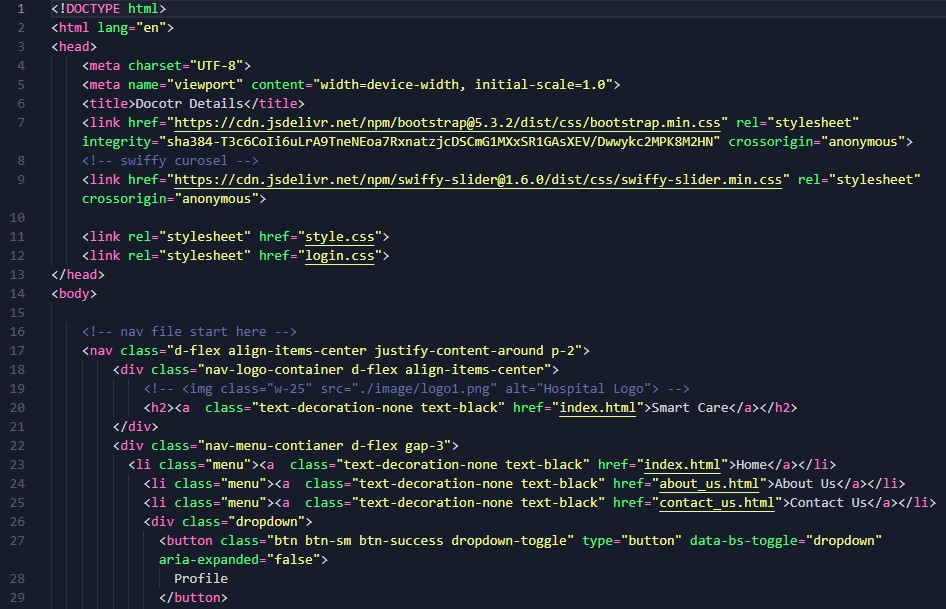


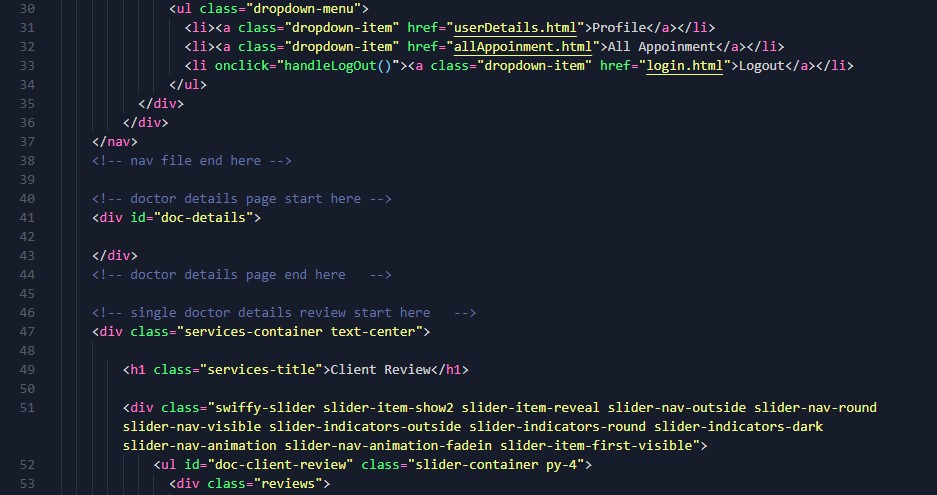


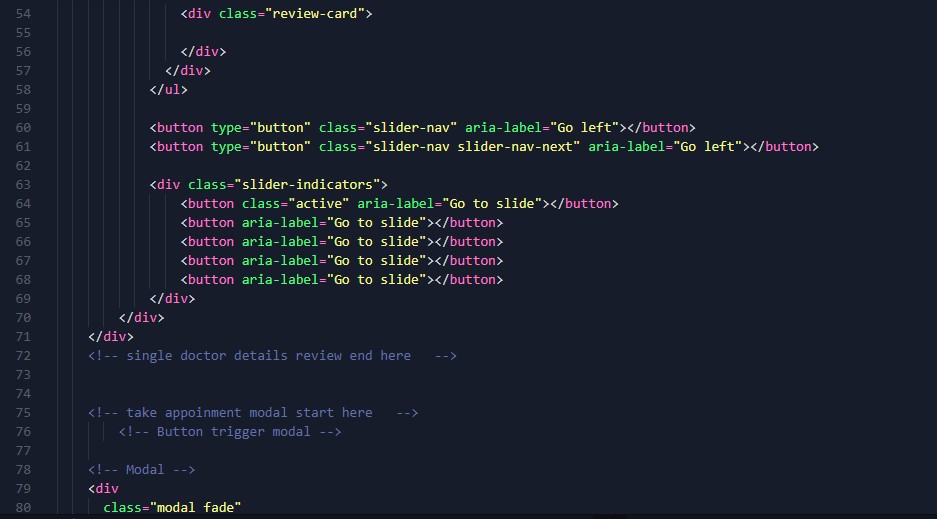


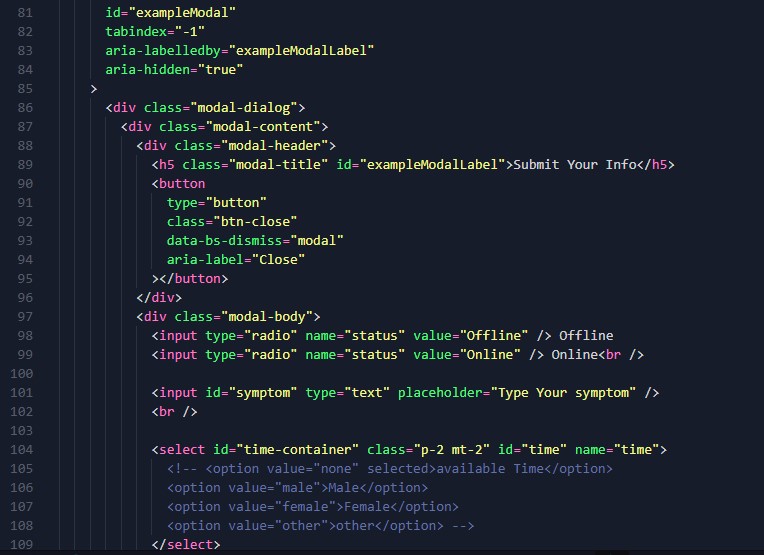


* docDetials.html



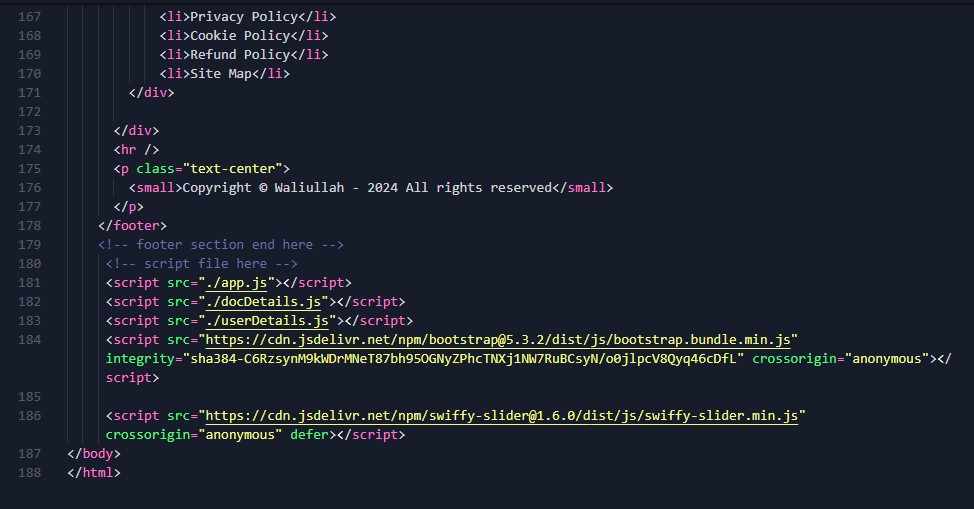




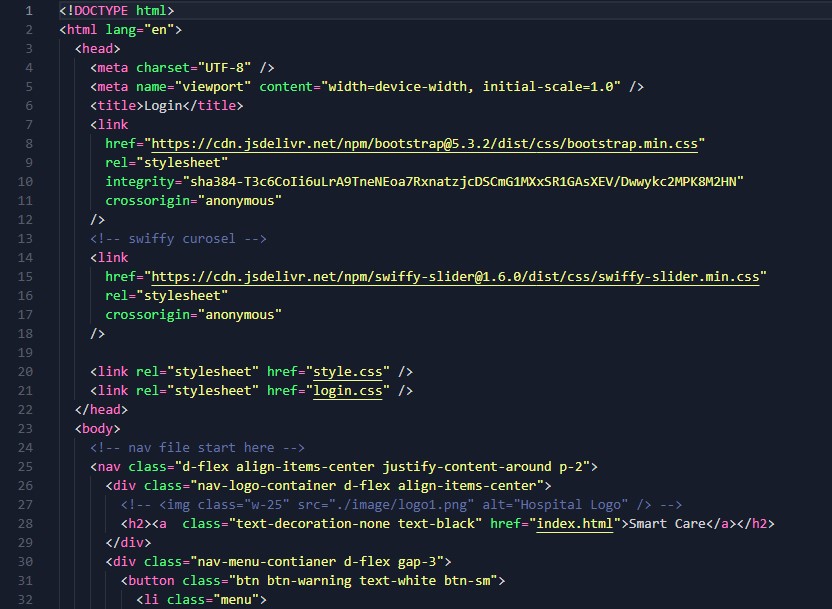


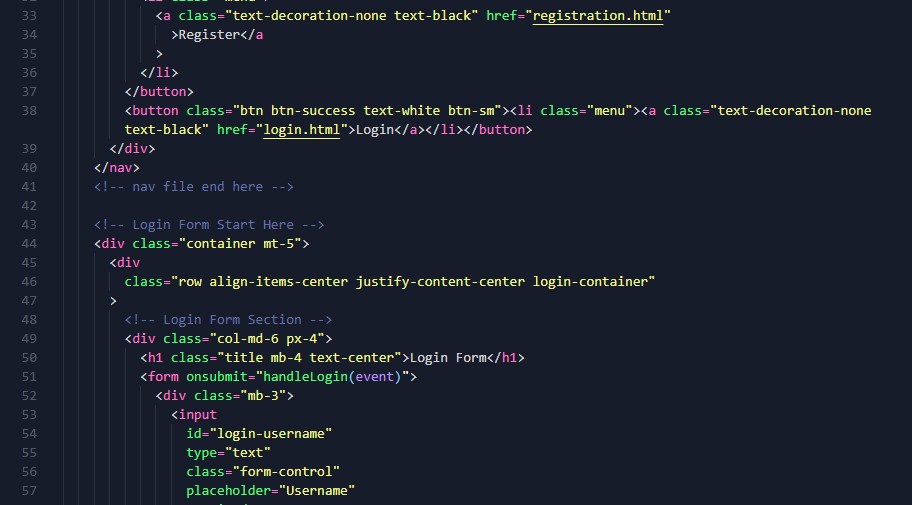


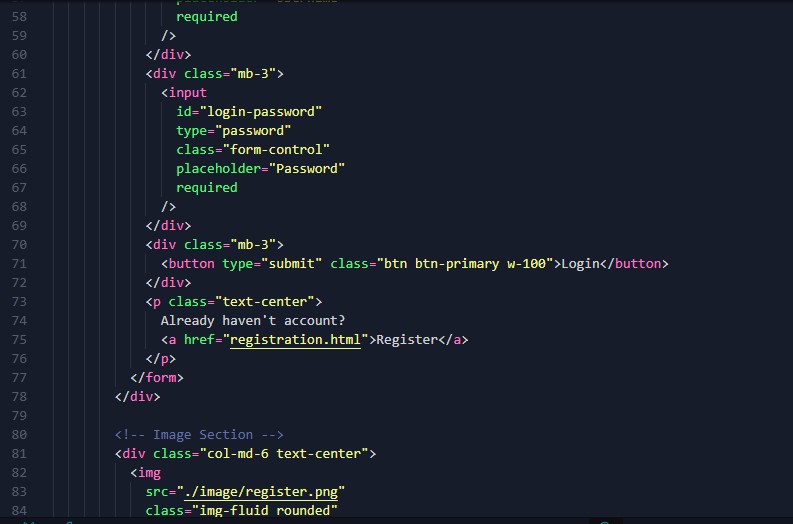


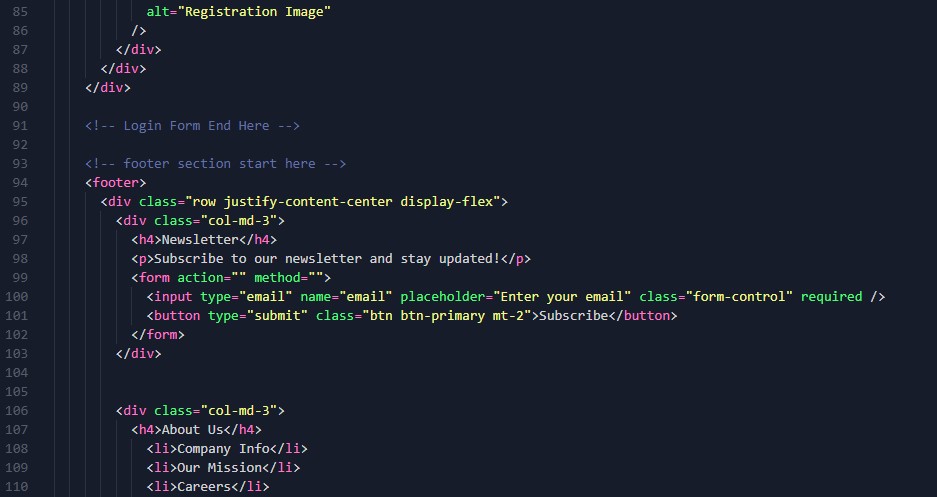


* login.html

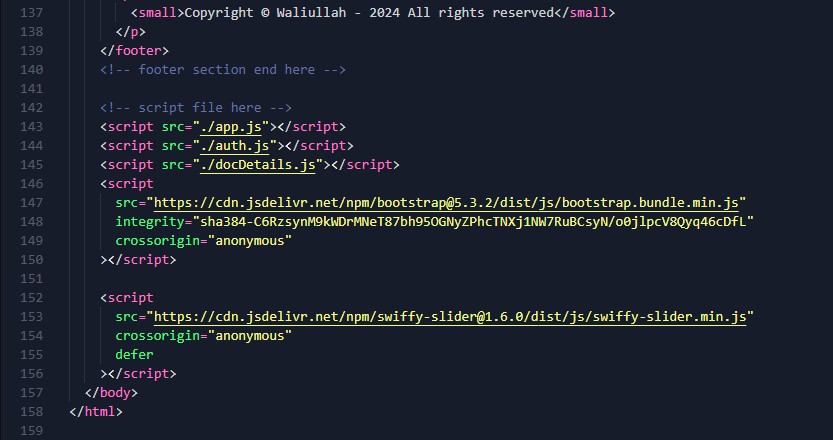




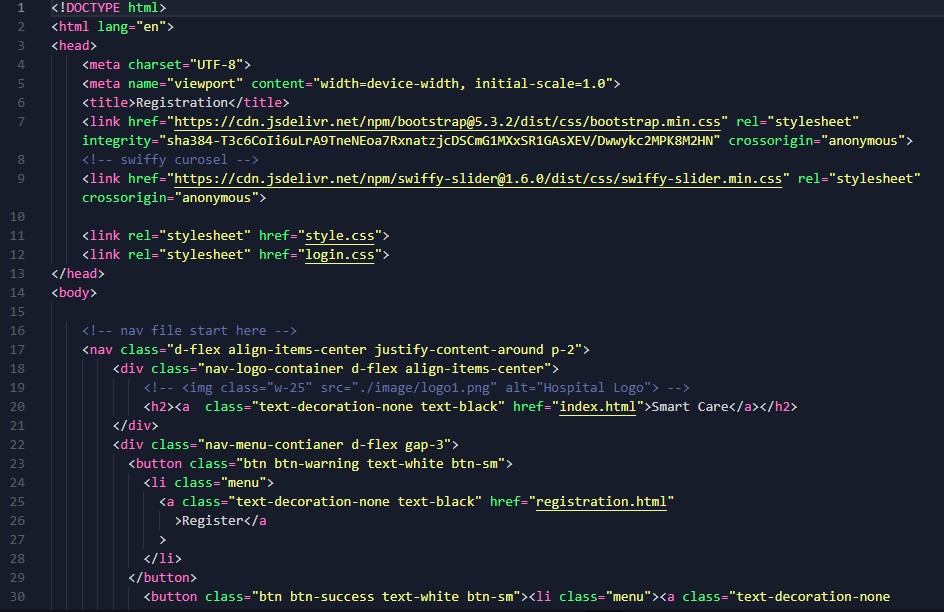




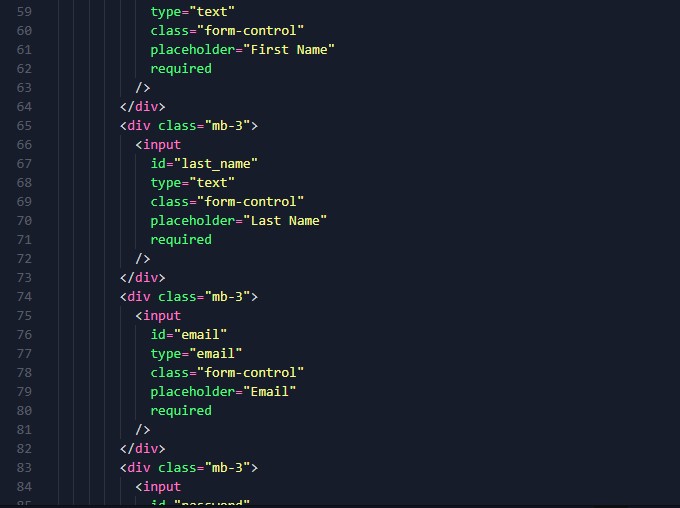


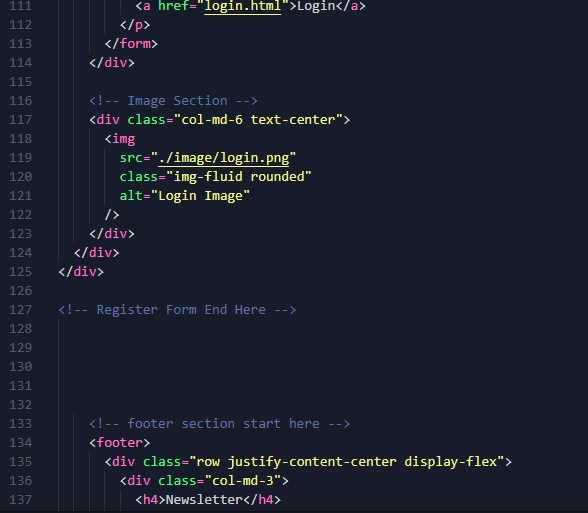
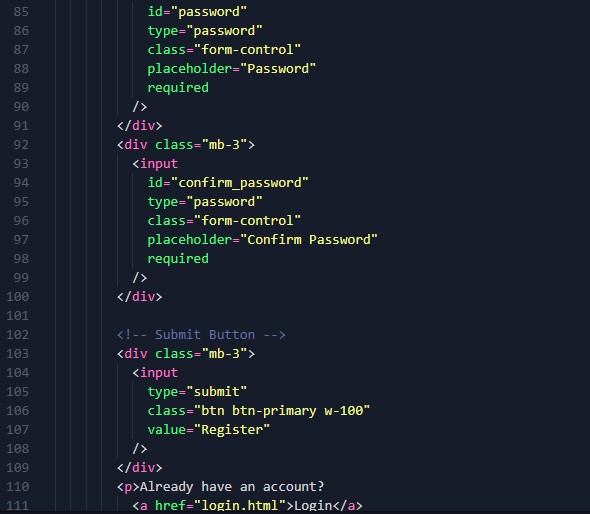


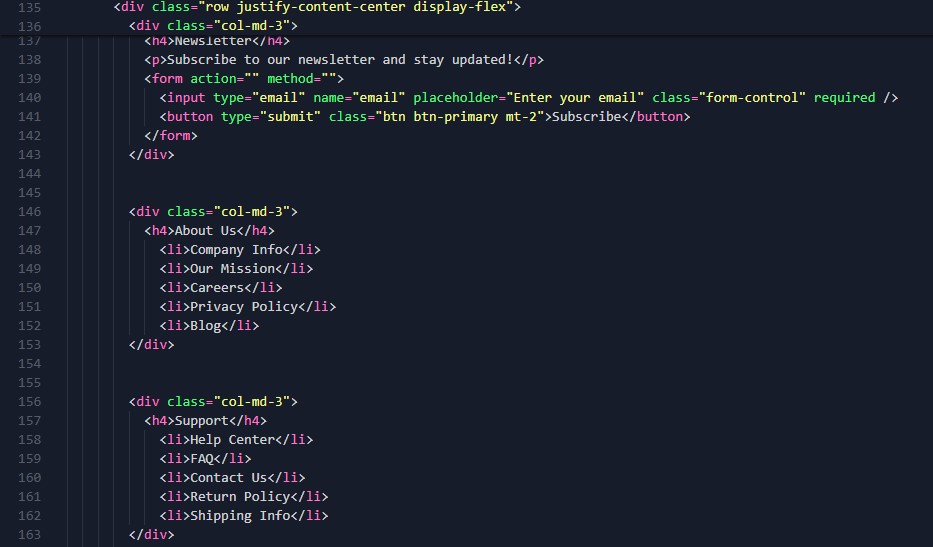
* register.html

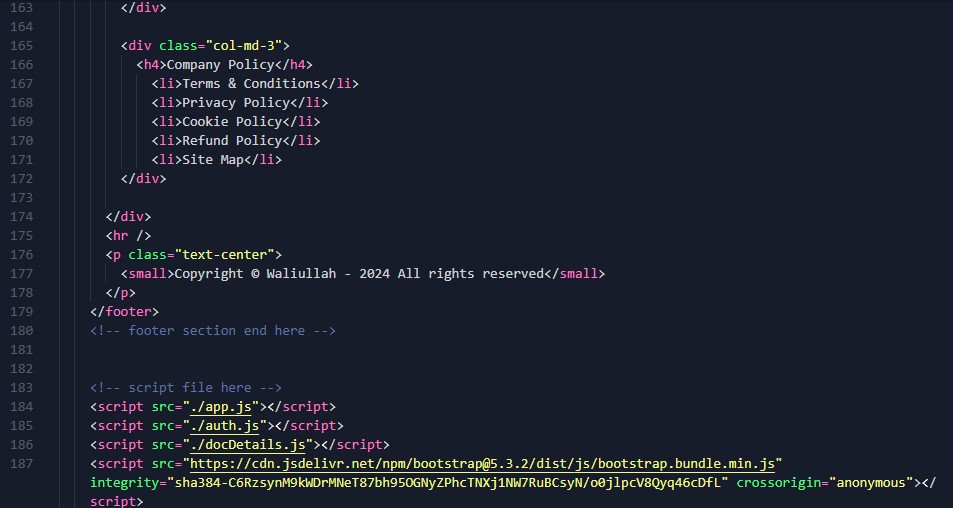


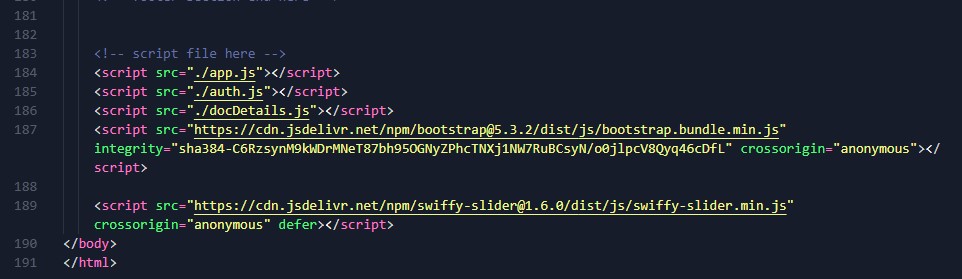




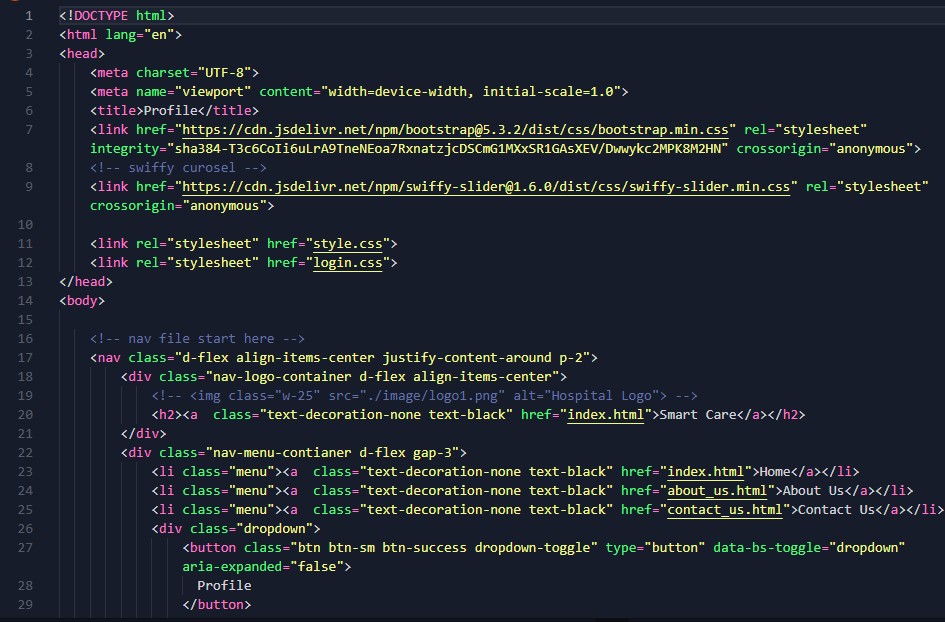


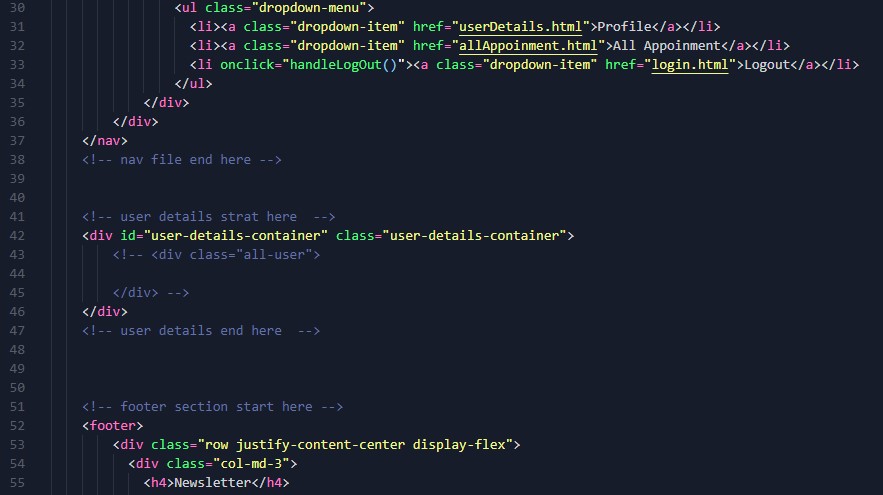


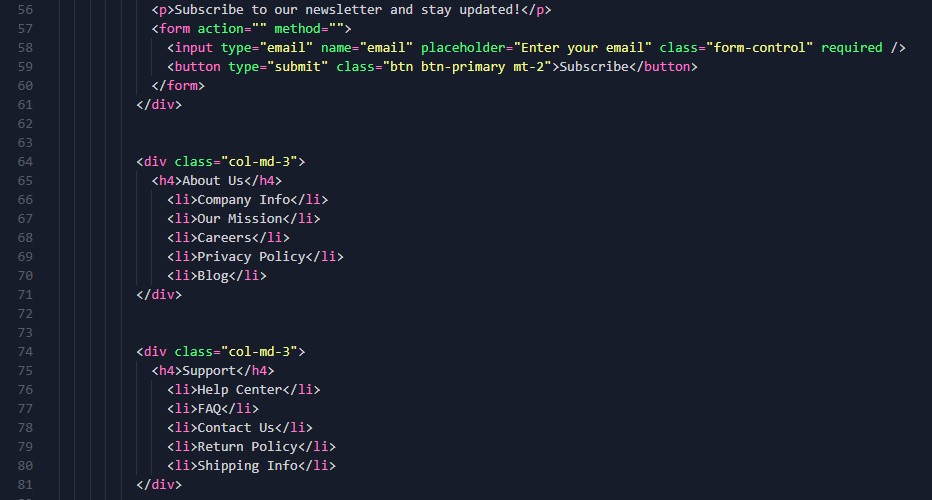


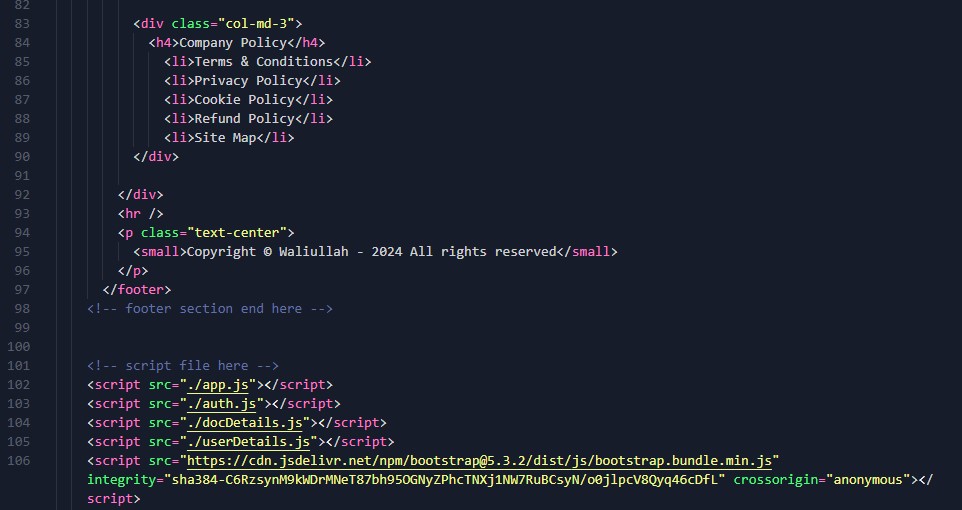


* userDetails.html

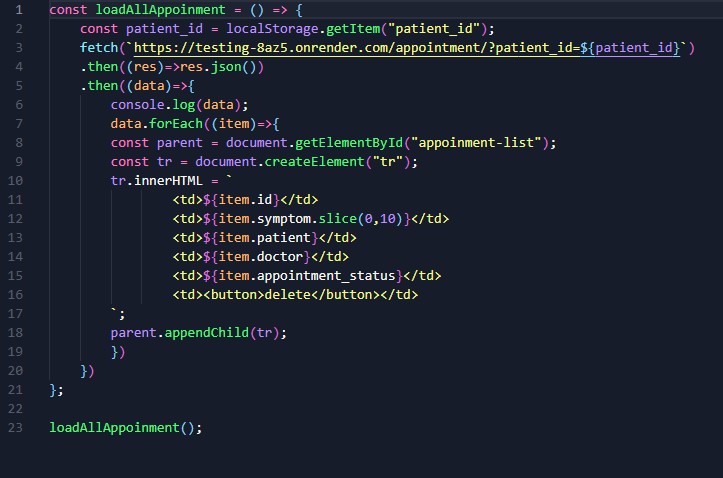




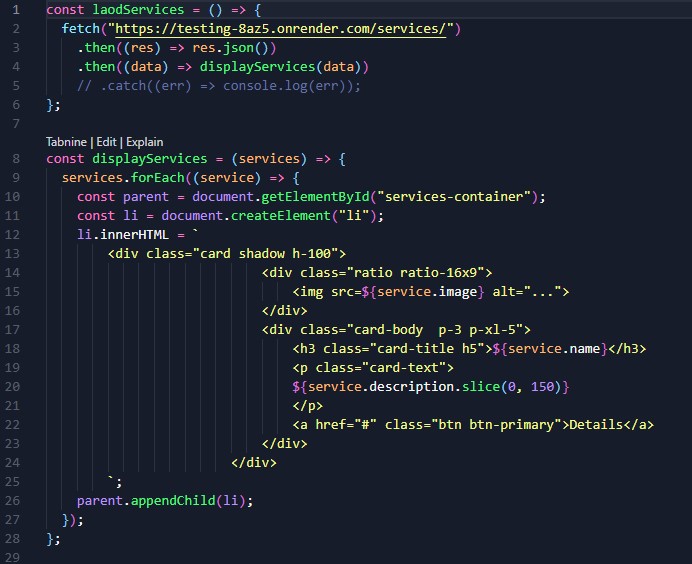




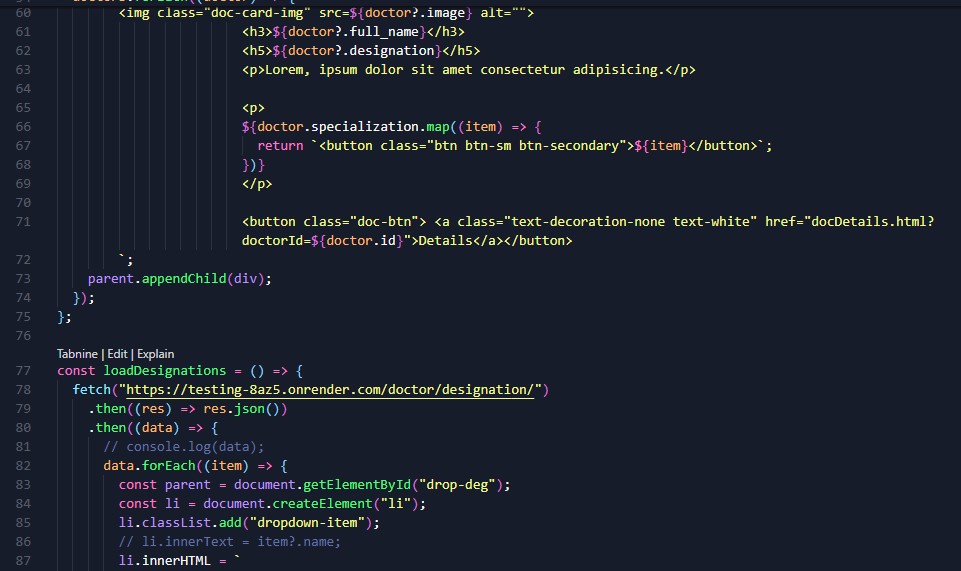
* allAppoinment.js

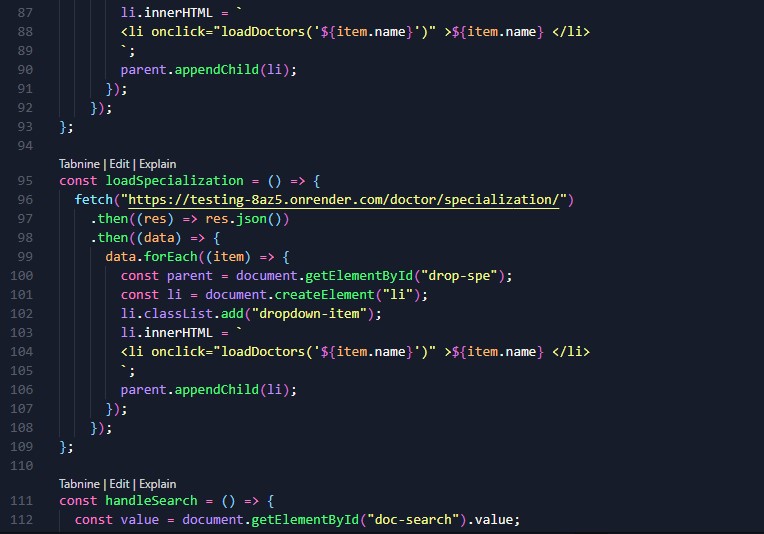


* app.js



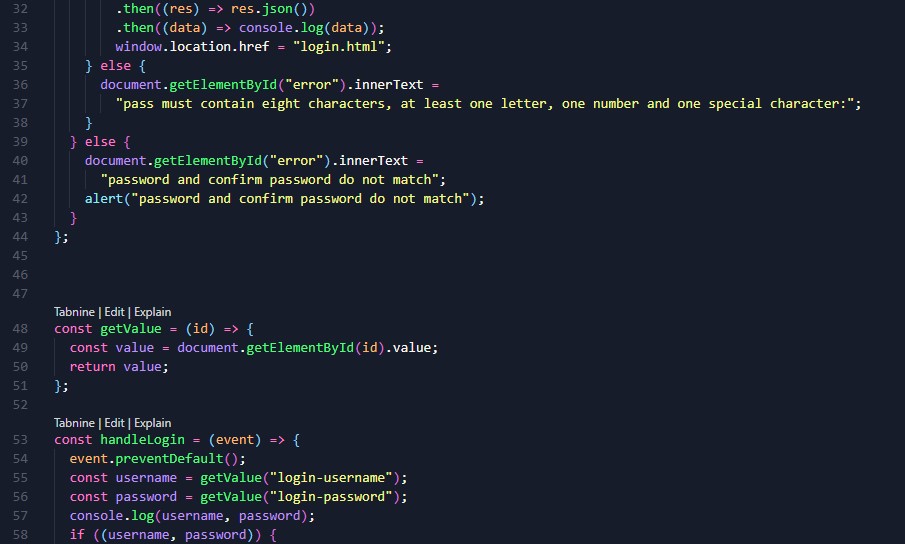
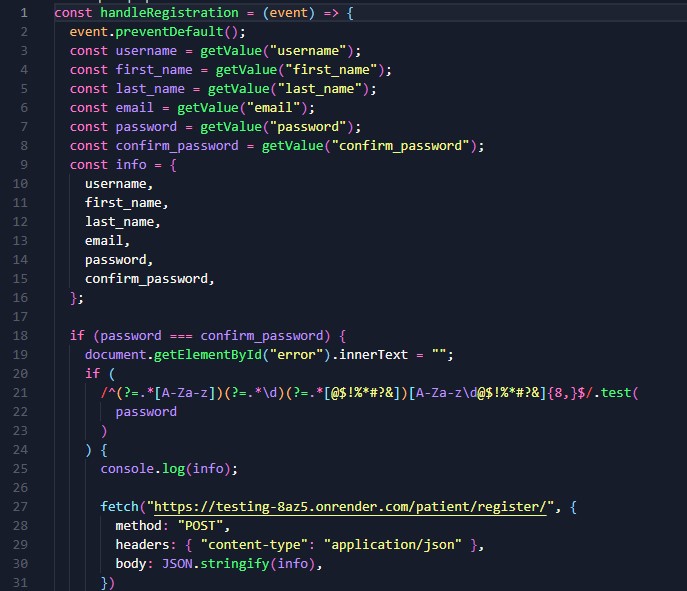






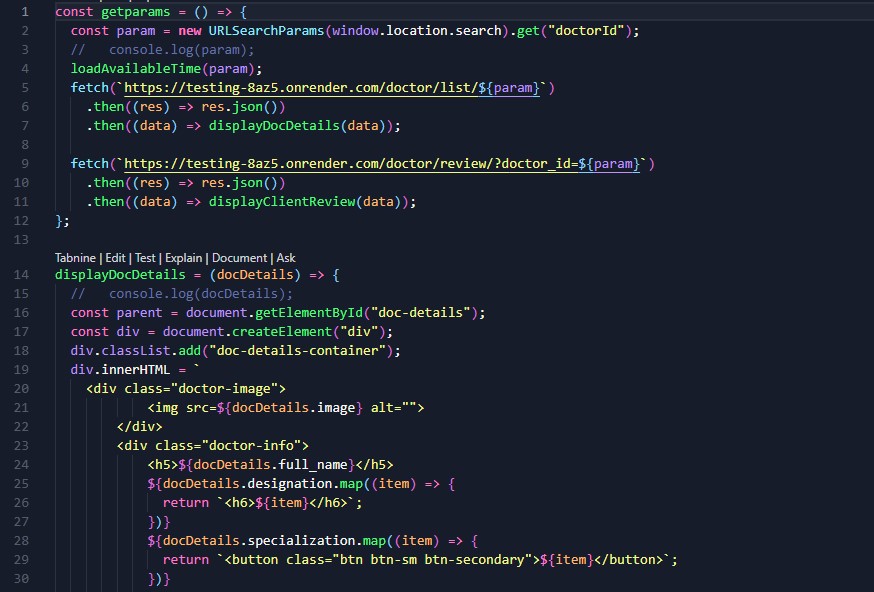


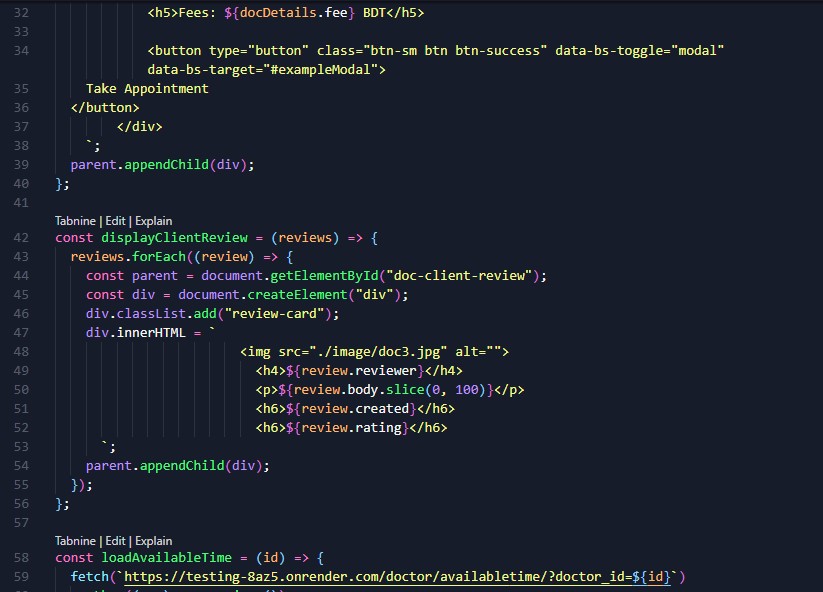
* auth.js

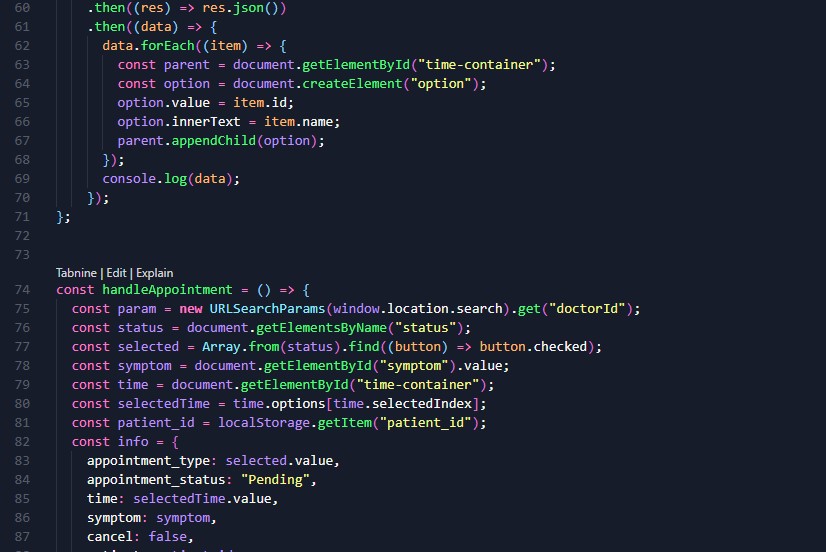


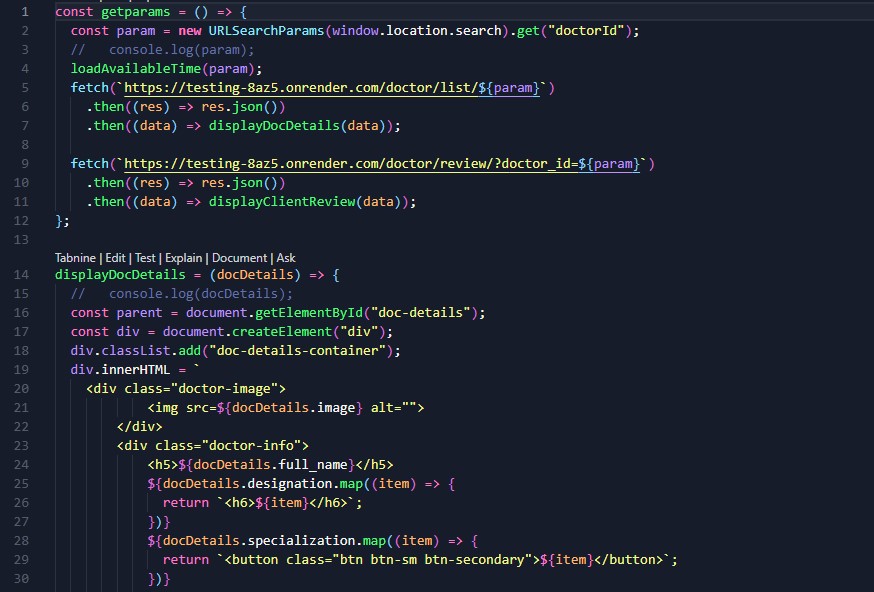


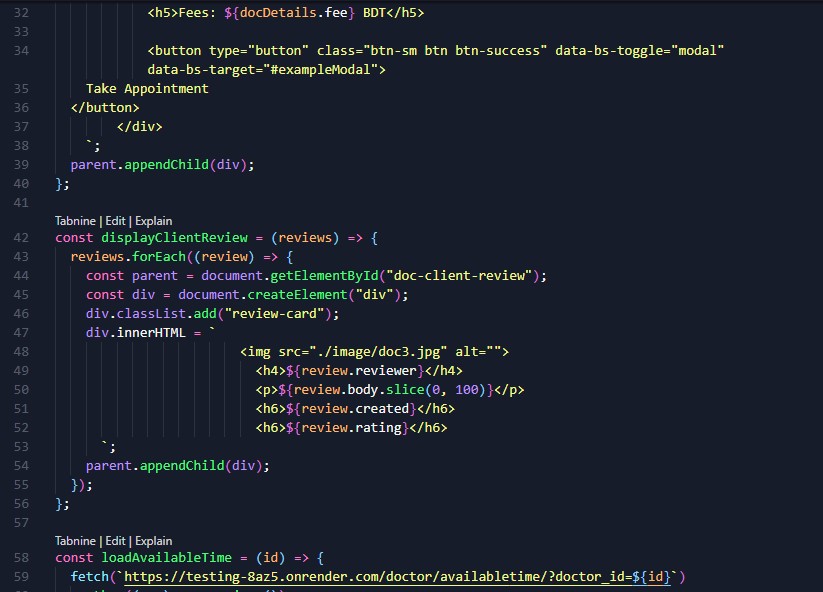
* docDetials.js

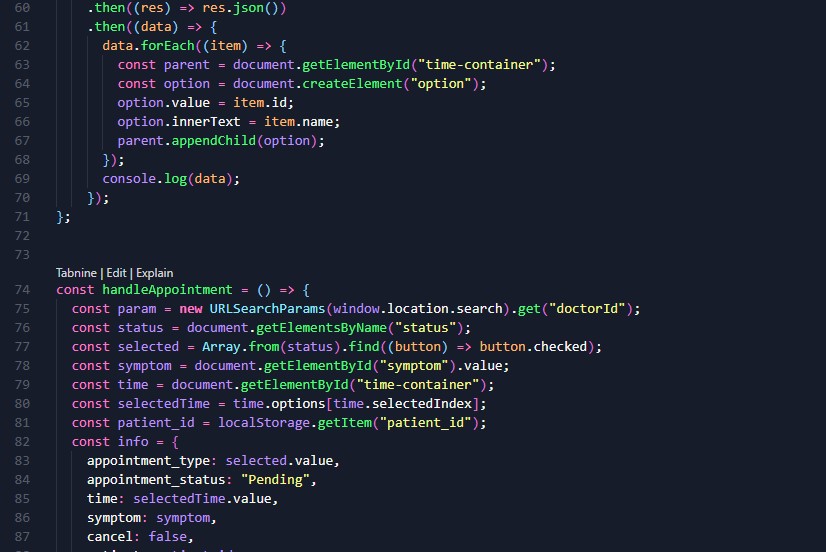


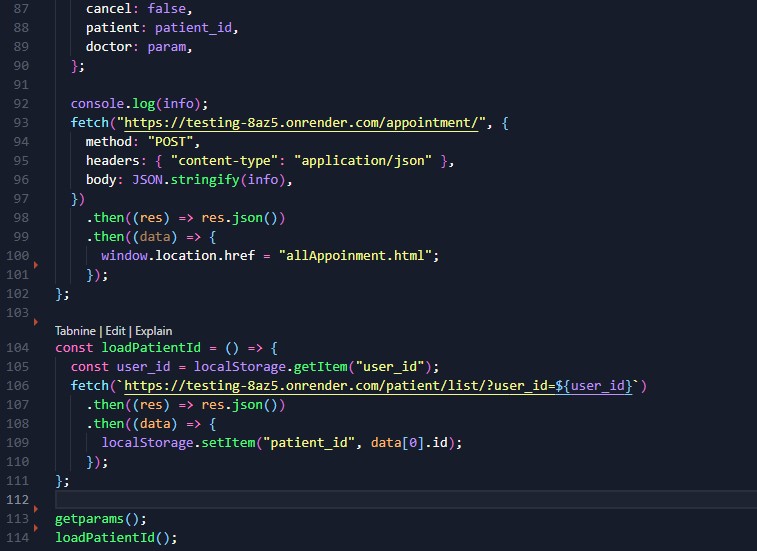




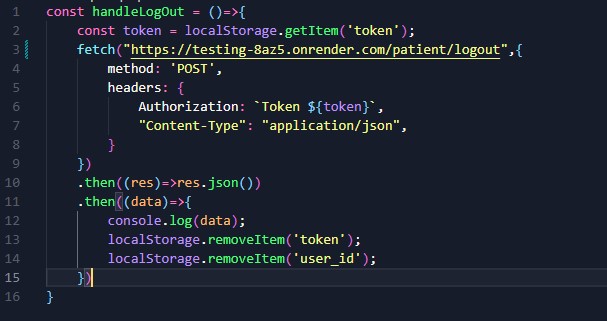








* logout.js



* userDetails.js



Chapter 3

# Performance Evaluation

## 3.1 Simulation Environment/ Simulation Procedure

### 3.1.1 Simulation Environment:

* Operating System: Windows/Linux/MacOS
* Web Server: Django development server (localhost)
* Database: SQLite for local testing and development • Browser: Google Chrome or Mozilla Firefox for frontend testing

### 3.1.2 Simulation Procedure:

* Set up the Django project environment with virtualenv.
* Migrate the database and create tables using Django’s migration tool.
* Test API endpoints using Django’s built-in testing utilities.
* Simulate appointment creation, patient record updates, and report generation.

## 3.2 Results Analysis

The system was tested by simulating real hospital operations, including:

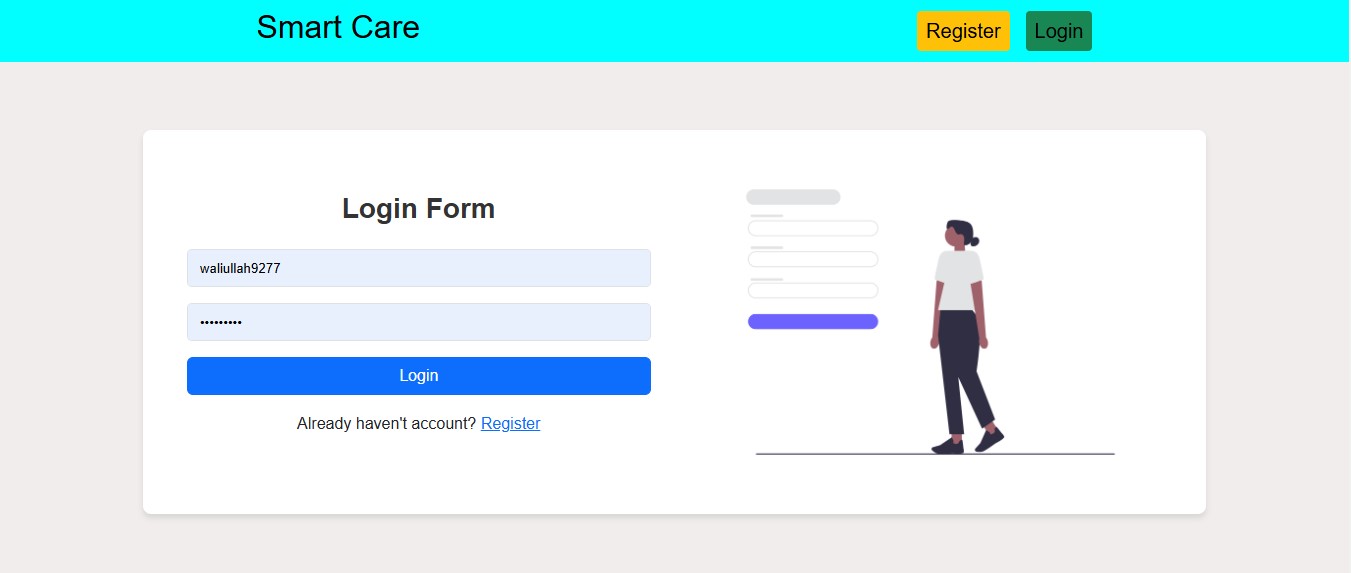
* Appointment booking: Ensured that appointments could be created, viewed, and updated by users.
* Patient management: Validated that patient records could be added, modified, and deleted.
* Reporting: Generated reports on patient visits, doctor schedules, etc.

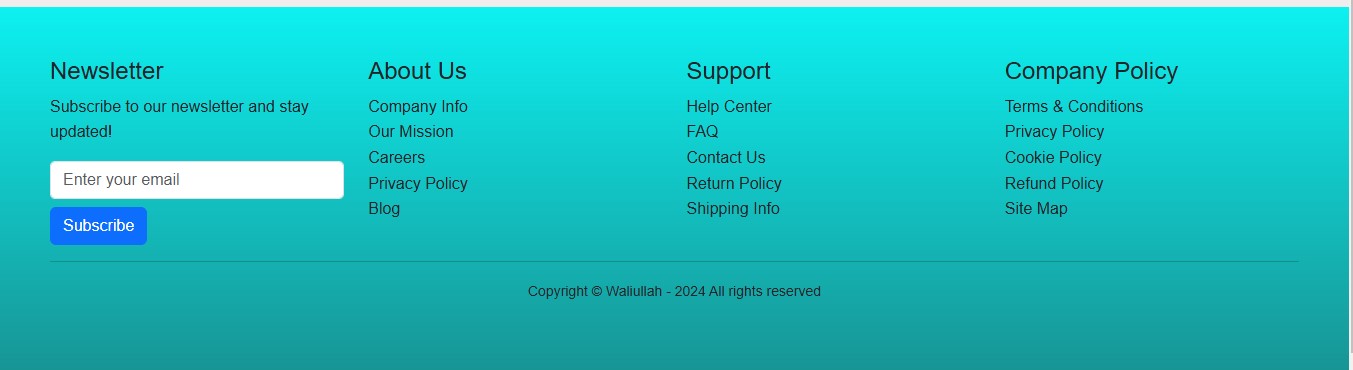
## 3.3 Results Overall Discussion

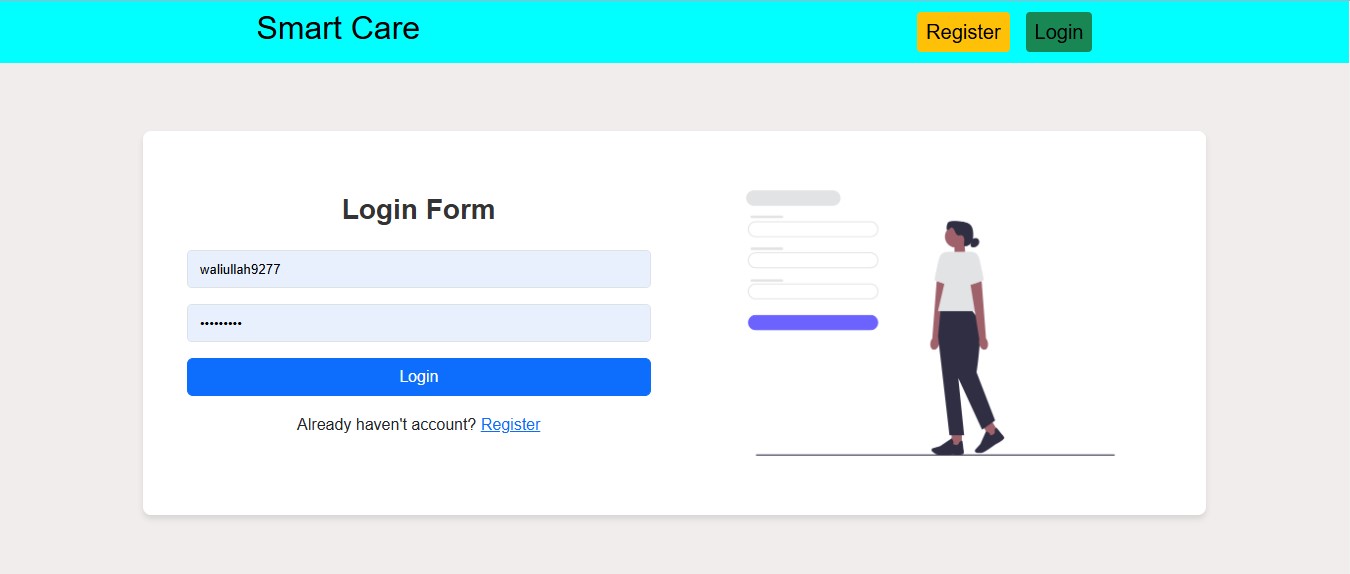
The system showed high efficiency in automating administrative tasks. The major functionality, including scheduling and patient management, worked seamlessly without any significant performance issues. There were some areas identified for future optimization, such as:

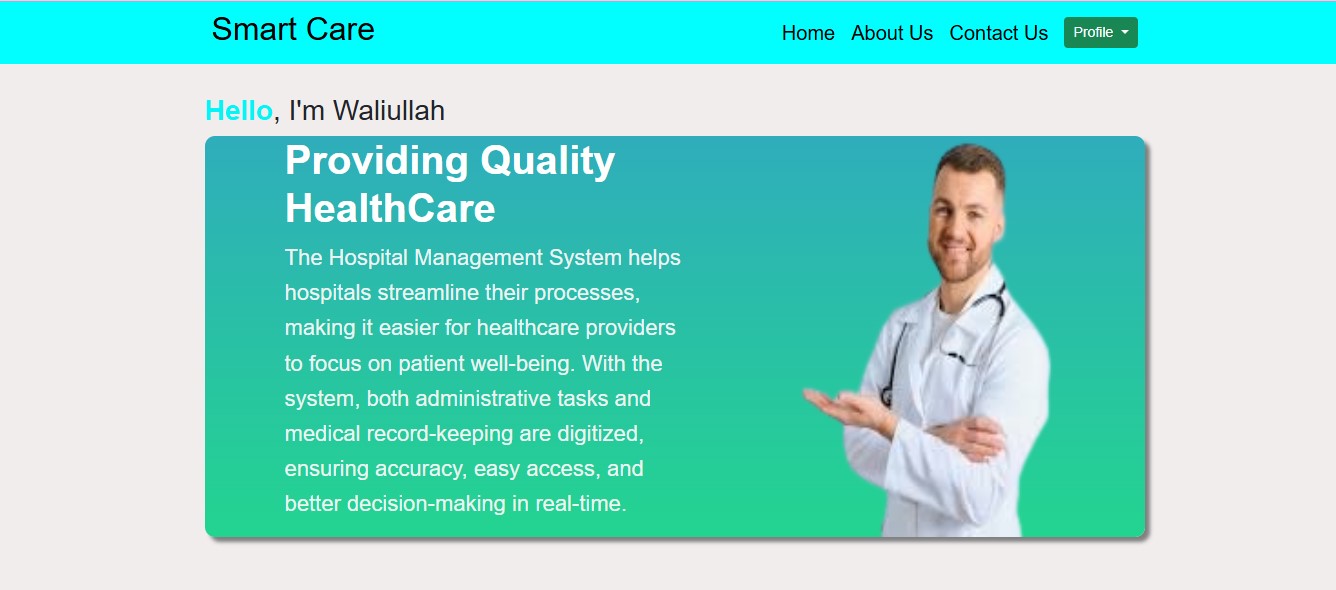
* Increasing the speed of report generation for larger datasets.
* Improving the mobile responsiveness of the frontend.

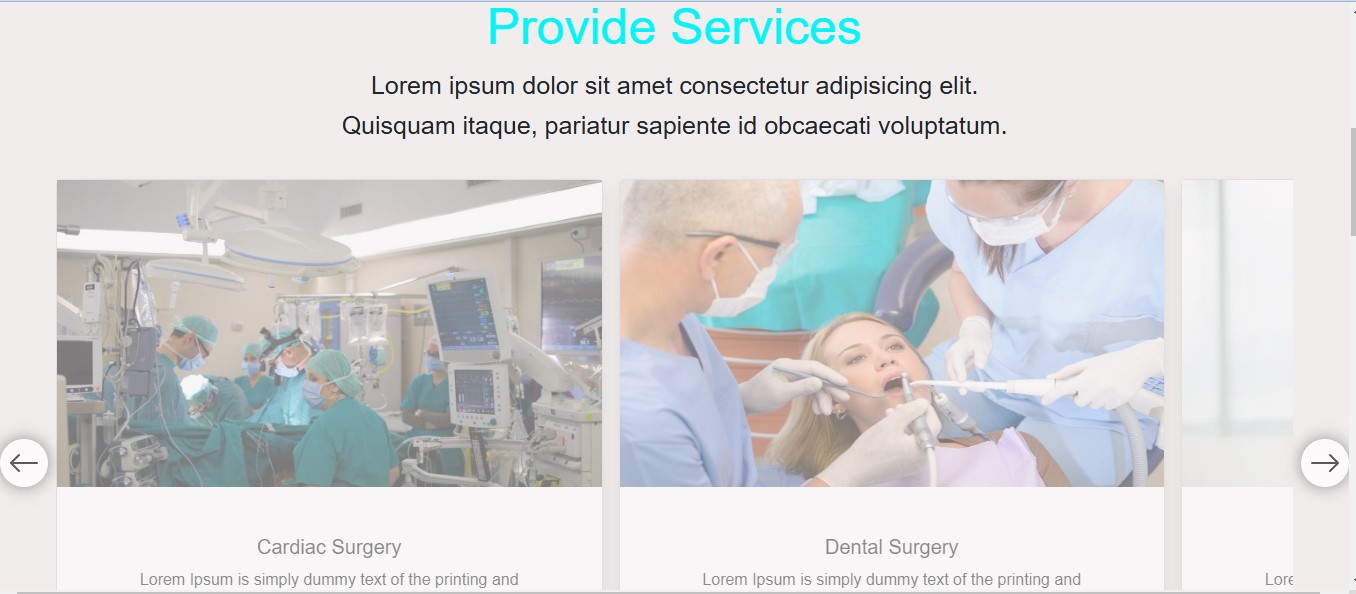
### 3.3.1 Result\_portion



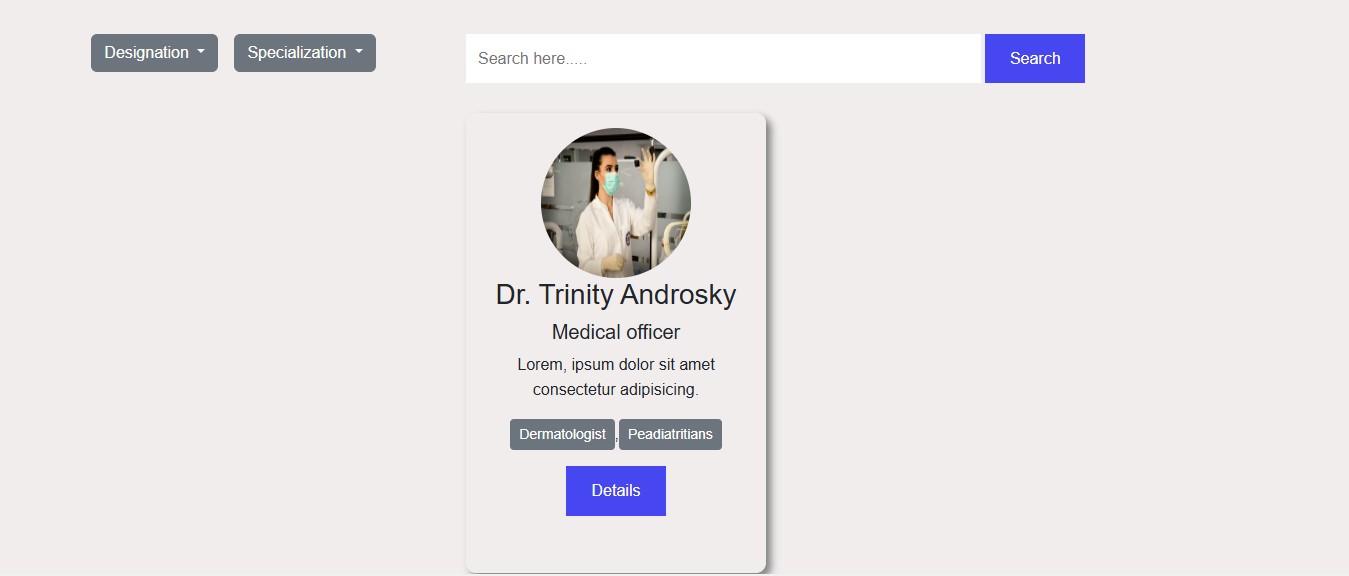


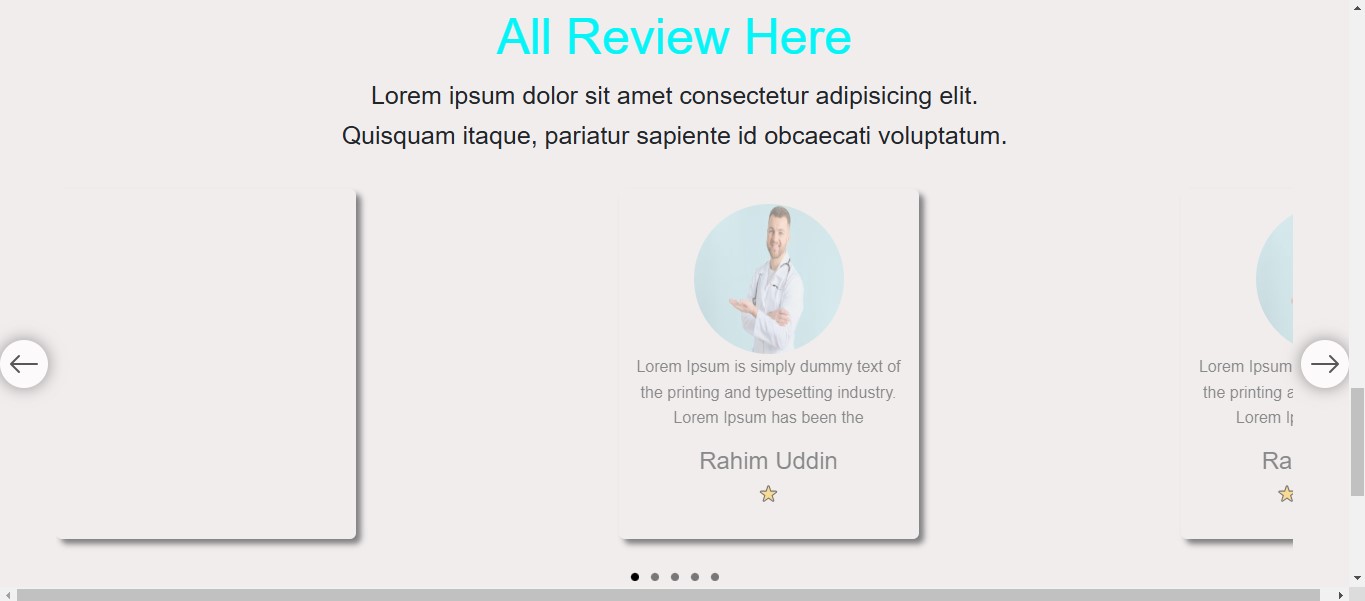




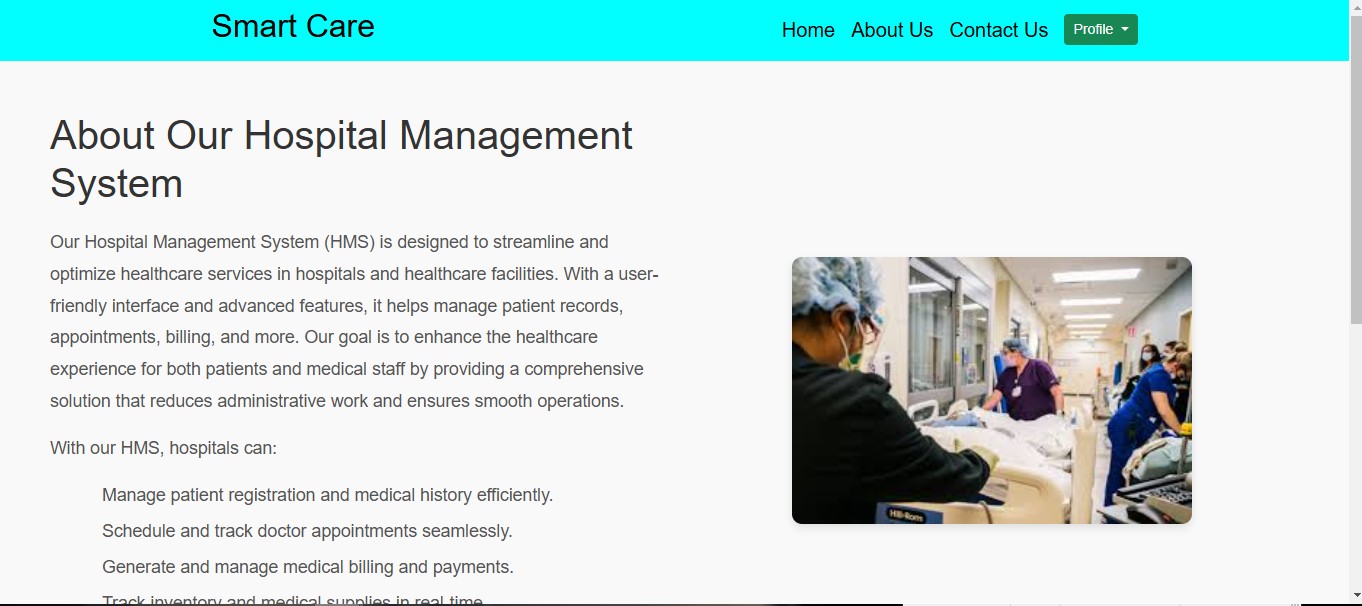


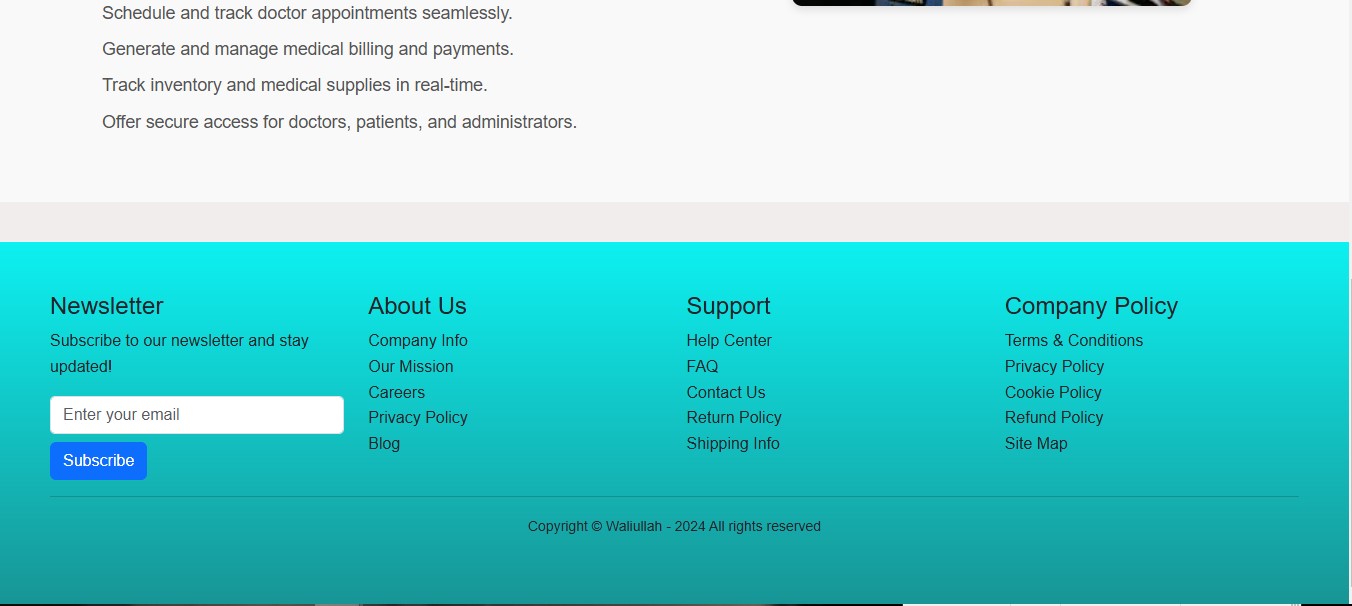


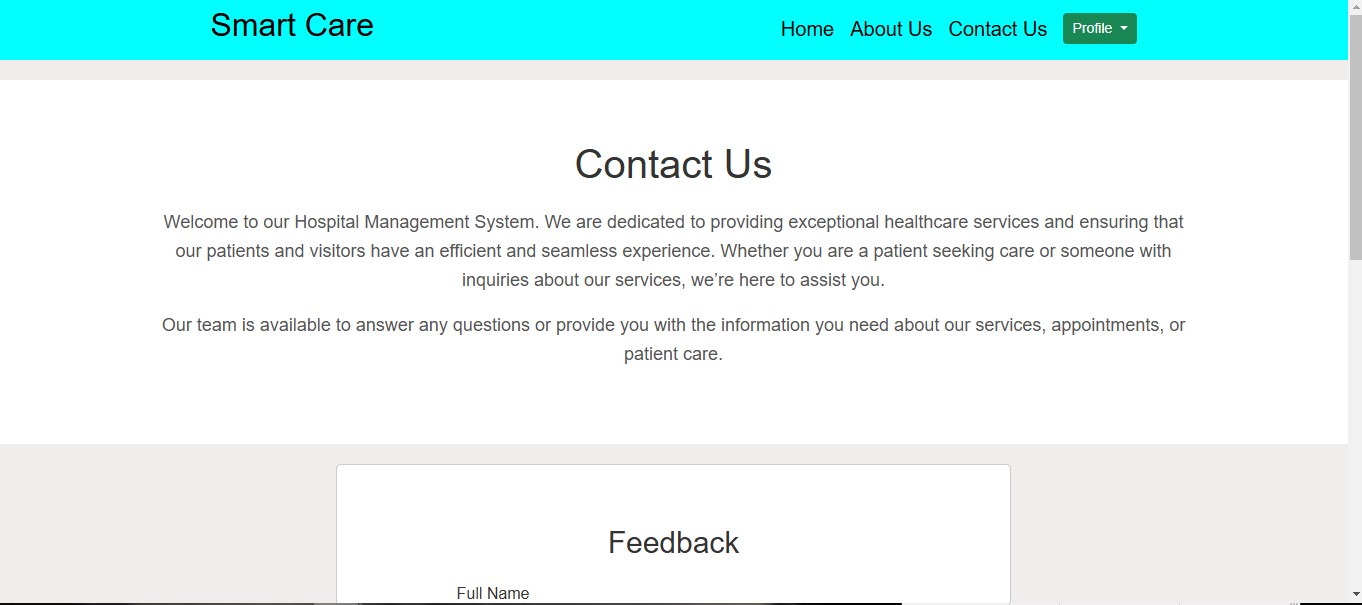


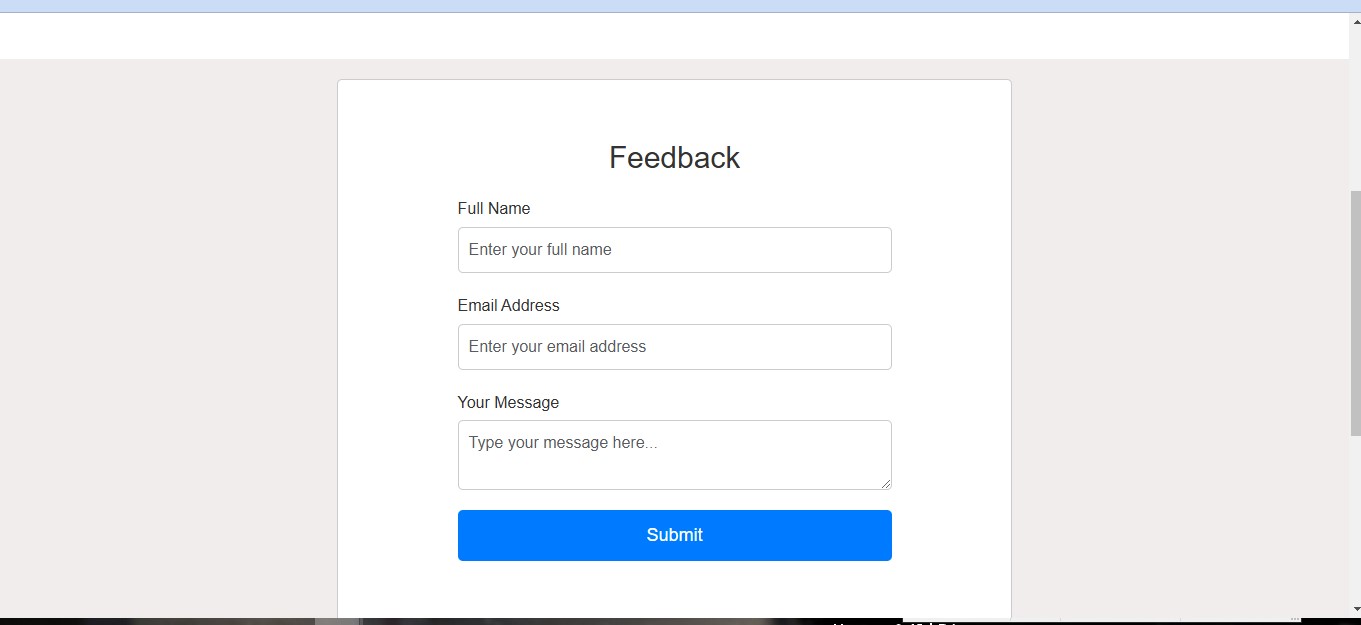


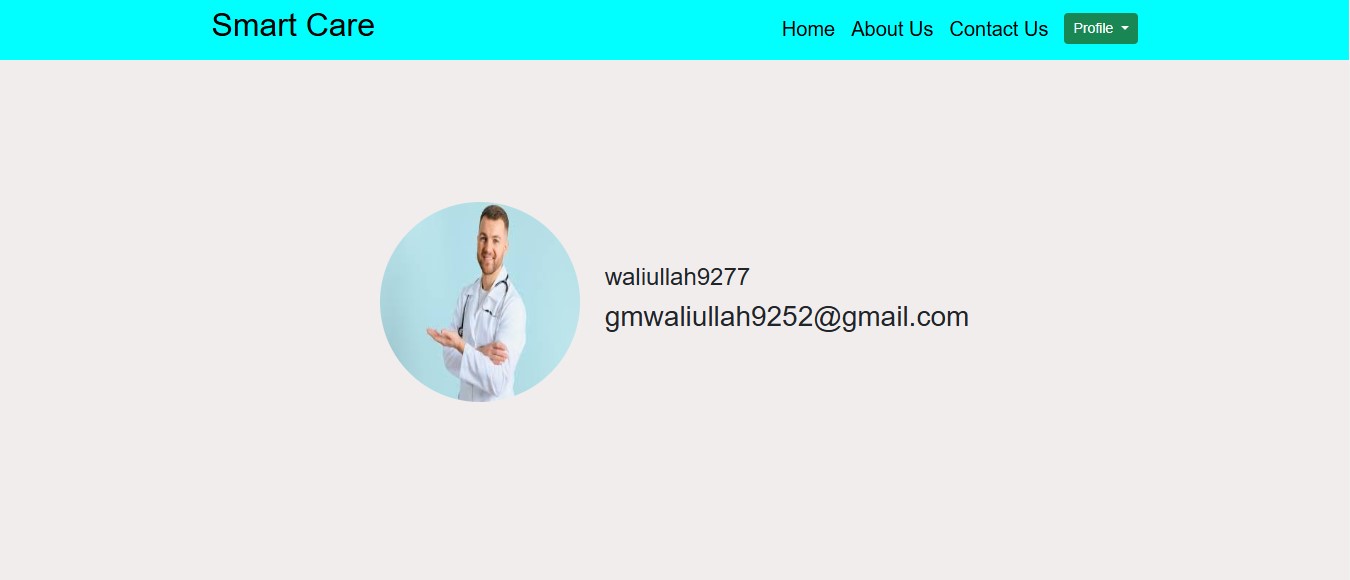


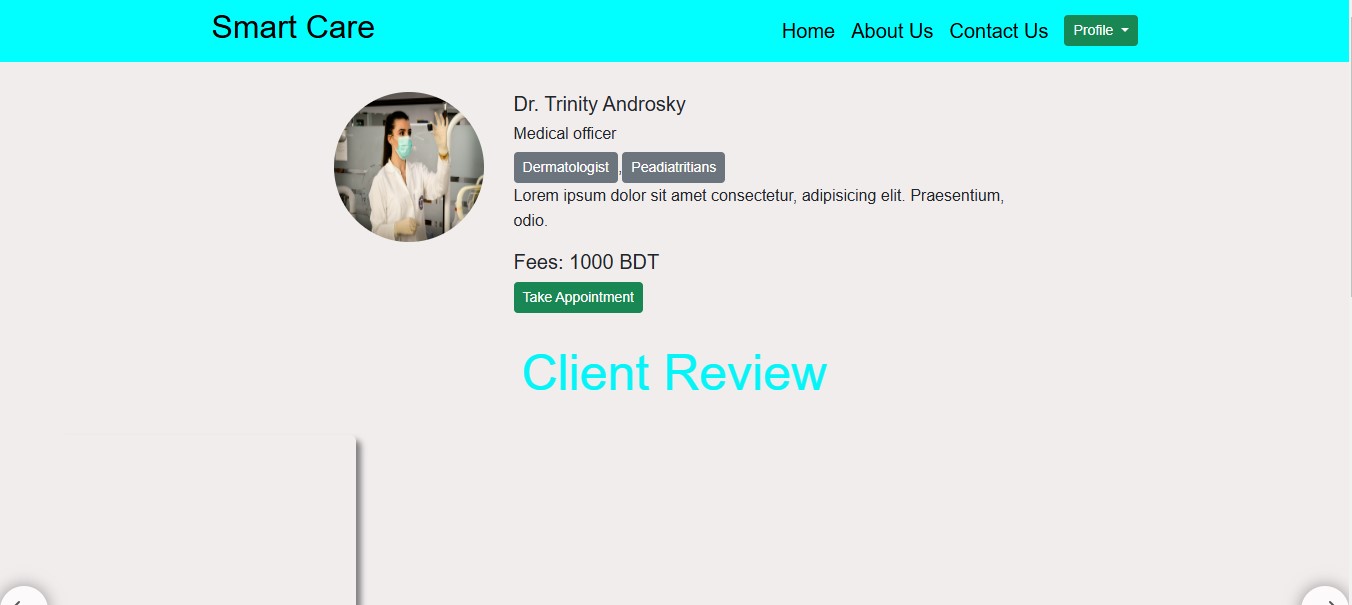


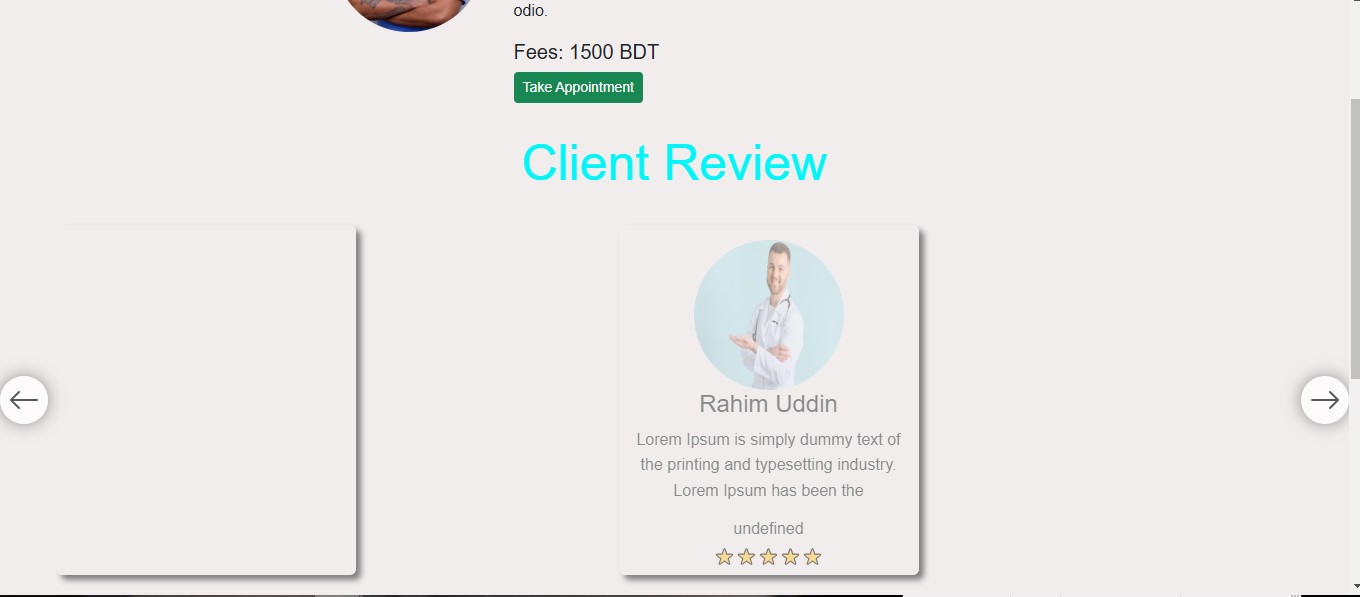


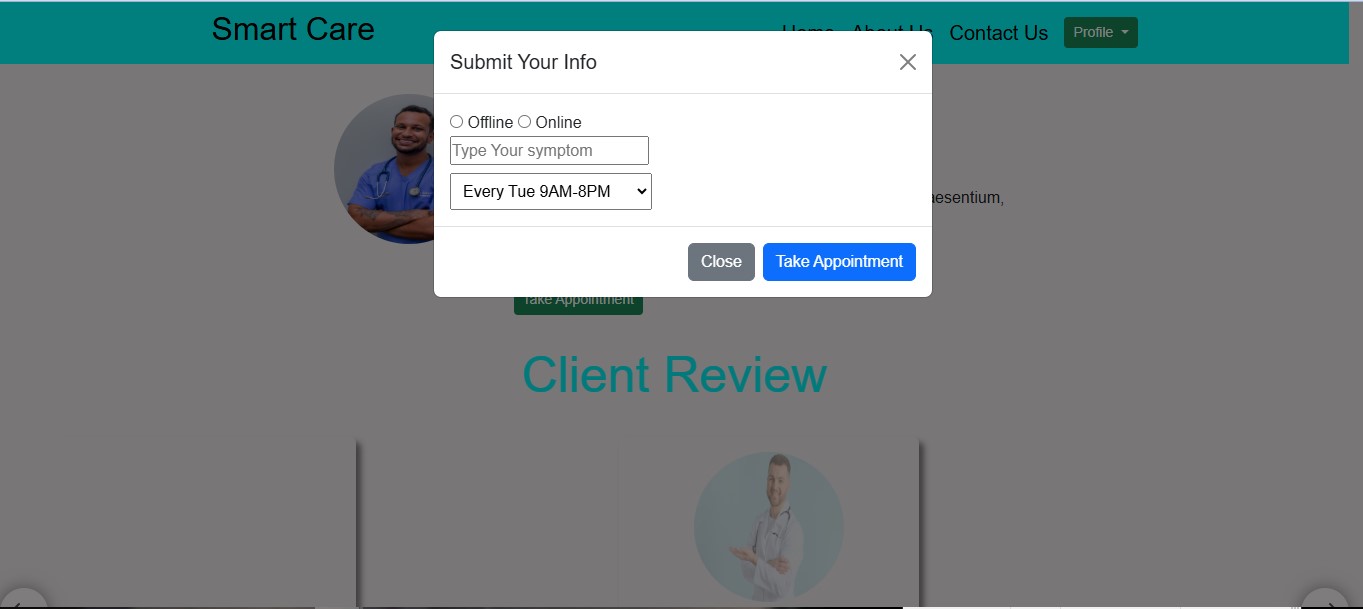


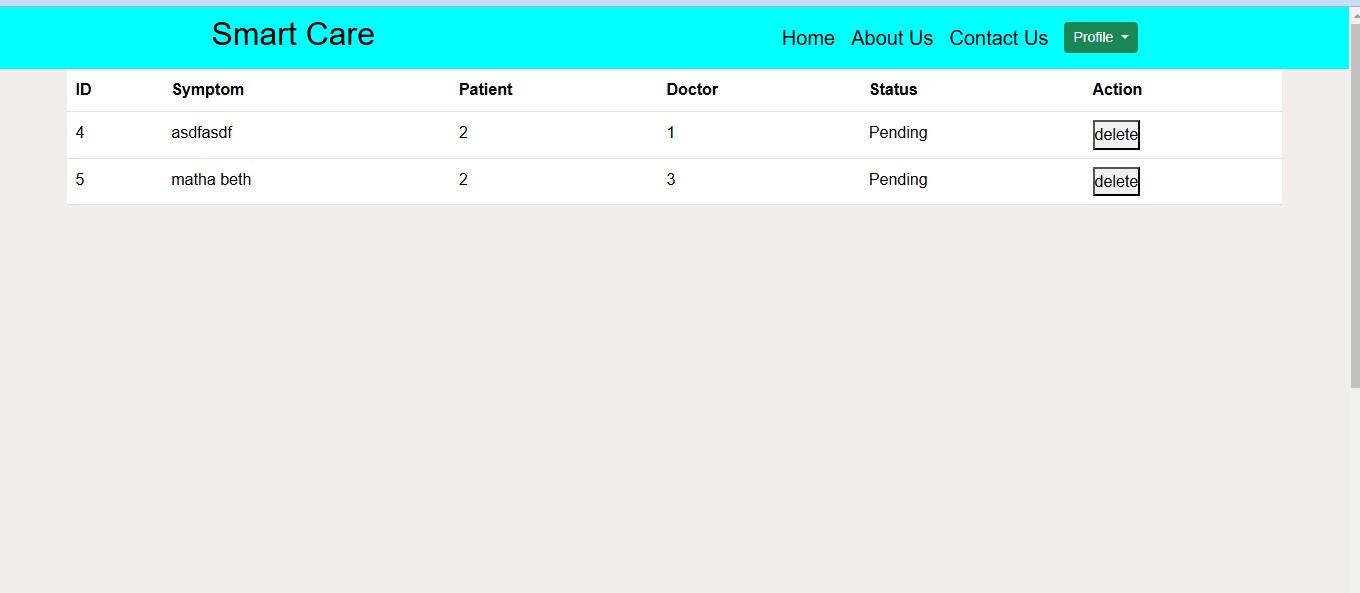












Chapter 4

# Conclusion

## 4.1 Discussion

The Hospital Management System provides a robust solution for automating the core operations of a hospital. It simplifies administrative workflows and improves data management. The system is scalable and can be expanded with additional features in the future.

## 4.2 Performance Evaluation

The performance evaluation shows that the system is responsive and efficient in managing hospital operations. Further optimizations could enhance the user experience, especially in mobile interfaces.

## 4.3 Limitations

* Limited functionality for patient-specific reports.
* The system needs more extensive testing with real hospital data.
* Mobile responsiveness needs improvement.

## 4.4 Scope of Future Work

* Mobile application integration for better accessibility.
* Real-time updates for doctor schedules and appointments.
* Integration with hospital billing systems and insurance claims.

## 4.5 References

List of references and resources used during the development and analysis of the Hospital Management System.

* Django REST Framework Documentation: https://www.django-rest-framework.org/
* SQLite Documentation: https://www.sqlite.org/docs.html
* https://www.google.com