HOSPITAL MANAGEMENT SYSTEM

A MINI PROJECT REPORT

[CASE STUDY]

Submitted by

TANYA YADAV[RA2211031010112]

JIYA GAYAWER[RA2211031010129]

Under the Guidance of

Dr. M. ANAND

[Assistant Professor, Department of Networking and Communications]

DEPARTMENT OF NETWORKING AND COMMUNICATIONS

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR– 603 203

NOV 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Certified that **21CSE354T – FULL STACK WEB DEVELOPMENT - Mini project report** titled "**HOSPITAL MANAGEMENT SYSTEM**" is the bonafide work of **Tanya Yadav [RA2211031010112]** and **Jiya Gayawer [RA2211031010129]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation.

SIGNATURE

Dr. M. ANAND
COURSE FACULTY
Assistant Professor
Department of Networking and Communications

SIGNATURE

Dr. M. LAKSHMI
HEAD OF THE DEPARTMENT
Department of Networking and Communications

DEPARTMENT OF NETWORKING AND COMMUNICATIONS
SCHOOL OF COMPUTING
College of Engineering and Technology
SRM Institute of Science and Technology

MINI PROJECT REPORT

ODD Semester, 2024-2025

Subject code & \Sub Name	:	21CSE354T & Full stack Web Development
Year & Semester	:	V
Project Title	:	Hospital Management System
Course Faculty Name	**:**	Dr. M. ANAND
Team Members	:	Tanya Yadav[RA2211031010112] and Jiya Gayawer[RA2211031010129]

Particulars	**Max. Marks**	**Marks Obtained**
FRONT END DEVELOPMENT	2.5	
BACK END DEVELOPMENT	2.5	
IMPLEMENTATION	3	
REPORT	**2**	

Date	**:**	
Staff Name	**:**	**Dr. M. Anand**
Signature	**:**	

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support. We extend our sincere thanks to **Dean-CET**, SRM Institute of Science and Technology, **Dr T.V.Gopal**, for his invaluable support.

We wish to thank **Dr Revathi Venkataraman**, **Professor and Chairperson**, School of Computing and **Dr M. Pushpalatha, Associate chairperson,** Department of Computing Technologies, SRM Institute of Science and Technology for their invaluable support.

We are grateful to our **Head of the Department**, **Dr. M. Lakshmi, Professor**, Dept. of NWC, SRMIST, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our **Audit Professor**, **Dr. R. K Pongiannan, Professor,** Department of Computing Technologies, SRM Institute of Scienceand Technology,for their inputs during the project reviews and support.

Our inexpressible respect and thanks to our course faculty, **Dr. M. Anand**, Assistant Professor, Department of NWC, SRMIST, for providing us with an opportunity to pursue our project under his mentorship. He provided us with the freedom and support to explore the research topics of our interest. His passion for solving problems and making a difference in the world has always been inspiring.

Tanya Yadav [RA2211031010112]

Jiya Gayawer[RA2211031010129]

TABLE OF CONTENTS

ABSTRACT

1. INTRODUCTION
   - 1.1 Overview
   - 1.2 Problem Statement
   - 1.3 Solution

2 PROJECT OVERVIEW AND OBJECTIVES
   - 2.1 Objectives
   - 2.2 Project Overview

3 ARCHITECTURE DIAGRAM AND TECHNOLOGIES USED
   - 3.1 Architecture Diagram
   - 3.2 Frontend Design
   - 3.3 Backend Design

4 PROJECT PLANNING
   - 4.1 Requirements
   - 4.2 Database Design

5 FRONTEND DEVELOPMENT
   - 5.1 HTML/CSS Structure
   - 5.2 JavaScript Structure

6 BACKEND DEVELOPMENT
   - 6.1 API Design, Authentication, and Database Connectivity
   - 6.2 Database Management and CRUD Operations

7 TESTING AND DEPLOYMENT

8. RESULTS AND CONCLUSION
   - 8.1 Conclusion
   - 8.2 Future Scope

REFERENCES

APPENDIX 1 - CODE

APPENDIX 2- OUTPUT

ABSTRACT

The Hospital Management System (HMS) is an integrated web-based solution designed to streamline and automate various administrative and medical processes within a hospital setting. Developed using MySQL for database management, PHP for server-side scripting, and front-end technologies such as HTML, CSS, and JavaScript, the system aims to improve hospital operations and enhance patient care through a centralized platform.The system's core functionality includes patient registration, appointment scheduling, medical records management, billing, and staff management. MySQL serves as the back-end database, storing critical information related to patients, doctors, treatment histories, and hospital inventory. It ensures secure, reliable, and scalable data management. PHP handles the logic for user authentication, database interactions, and dynamic page generation, facilitating smooth communication between the system and users.By automating manual processes, the HMS reduces administrative burdens, minimizes human error, and enhances communication across departments. Key features such as appointment scheduling, real-time medical record access, and billing management contribute to smoother operations, improved patient care, and optimized resource utilization. Additionally, the system's flexibility and scalability ensure it can be adapted to meet the evolving needs of the hospital.In conclusion, the Hospital Management System provides a robust solution for modern healthcare facilities, improving operational efficiency, data management, and overall service quality, while offering a user-friendly, accessible platform for hospital staff and patients alike.

1. INTRODUCTION

1.1 Overview

The HMS is a comprehensive web-based application designed to centralize and automate various administrative and medical tasks within a hospital. It utilizes MySQL as the database management system for storing critical data such as patient details, treatment records, and hospital inventory. PHP is employed for server-side scripting, processing user input and managing interactions between the front-end and database. The front-end is built using HTML, CSS, and JavaScript to create a responsive, interactive interface for users, including doctors, nurses, administrative staff, and hospital management.

The key features of the system include patient registration, appointment booking, medical record access, billing management, and inventory tracking. By providing a unified platform for these functions, the HMS aims to improve coordination, reduce administrative workload, and optimize hospital resources.

1.2 Problem Statement

Hospital administration involves managing a vast amount of data, including patient records, appointments, medical histories, and billing information. Traditional methods, such as paper records and isolated software systems, often lead to inefficiencies, data errors, and delays. These challenges can negatively impact patient care, increase administrative burden, and hinder the smooth functioning of hospital operations. The lack of centralized data management makes it difficult to retrieve accurate information promptly, leading to potential errors in medical decisions, delays in treatments, and increased operational costs.

1.3 Solution

The Hospital Management System (HMS) offers a comprehensive and integrated solution to the challenges faced by healthcare institutions in managing operations effectively. By leveraging modern technologies such as MySQL, PHP, and front-end development tools, the HMS centralizes hospital functions into a unified web-based platform, addressing issues related to data management, resource allocation, and administrative inefficiencies.

The solution is built around a robust database architecture powered by MySQL, ensuring secure and reliable storage of critical hospital data such as patient details, medical records, billing information, doctor schedules, and hospital inventory. The database is designed to support large-scale operations, enabling real-time updates and efficient data retrieval. With MySQL, hospitals can ensure that their data is protected through strict access controls, automated backups, and disaster recovery options.

PHP serves as the server-side scripting language for handling business logic, processing user requests, and interacting with the MySQL database. This technology enables dynamic content generation, such as appointment scheduling, real-time medical record access, and billing management. It also ensures that hospital staff, including doctors, nurses, and administrators, can interact with the system seamlessly and efficiently, even as the system scales with increasing users and data. PHP's flexibility allows for easy integration of new features or changes as hospital requirements evolve.

The front-end of the HMS is designed to provide an intuitive and user-friendly interface. Built using HTML, CSS, and JavaScript, the interface is both responsive and accessible across a variety of devices, from desktop computers to mobile phones. The system ensures that hospital staff can perform their tasks without needing specialized technical knowledge, as the user interface is designed to be simple, clear, and easy to navigate. The use of JavaScript adds interactivity, such as real-time validation of forms and instant updates to appointment schedules or patient information.

Key modules within the system include:

- **Patient Registration and Management**: Hospitals can efficiently register new patients, update their records, and track their treatment history.
- **Appointment Scheduling**: The system allows patients to book appointments with doctors, view available time slots, and receive confirmation notifications.
- **Medical Records Management**: All patient medical records, including diagnosis, treatments, and prescriptions, are stored digitally.
- **Billing and Financial Management**: The HMS automates the billing process, generating invoices for medical services and treatments.
- **Inventory Management**: The system allows hospitals to track medical supplies and equipment usage, ensuring that inventory is replenished in time and resources are used efficiently.
- **Staff Management**: Hospitals can manage staff schedules, track working hours, and oversee payroll and attendance.

2. PROJECT OVERVIEW AND OBJECTIVES

2.1 Objectives

The primary objectives of the Hospital Management System (HMS) are as follows:

- **Efficient Management of Hospital Data**: To create a centralized platform that integrates all hospital operations, such as patient registration, appointment scheduling, billing, medical records, and inventory management, into a single system to improve data organization and accessibility.
- **Automation of Administrative Tasks**: To automate routine administrative tasks, such as appointment bookings, billing, and staff scheduling, reducing the dependency on manual processes and minimizing errors and inefficiencies.
- **Real-Time Access to Patient Information**: To provide authorized medical professionals with instant access to up-to-date patient information, treatment histories, and prescriptions, ensuring timely and accurate medical decision-making.
- **Improved Communication and Coordination**: To enhance communication and coordination among hospital departments (doctors, nurses, administrators, and support staff), ensuring smooth workflows and optimized resource allocation.
- **Data Security and Privacy**: To ensure the security of sensitive patient data through role-based access control, data encryption, and secure database management, maintaining confidentiality and complying with healthcare regulations.
- **User-Friendly Interface**: To design an intuitive and easy-to-use interface for hospital staff and administrators, allowing them to navigate the system effortlessly without requiring specialized technical skills.
- **Scalability and Flexibility**: To develop a system that is scalable, allowing hospitals of different sizes to implement the system and tailor it according to their specific needs, with the ability to grow as the hospital expands.
- **Reporting and Analytics**: To enable hospital management to generate reports and analytics on key metrics such as patient demographics, financial performance, treatment outcomes, and staff productivity, facilitating better decision-making and hospital performance evaluation.

- **Cost and Time Efficiency**: To reduce operational costs and time spent on manual record-keeping, administrative work, and patient management, improving overall efficiency and cost-effectiveness.

2.2 Project Overview:

The Hospital Management System (HMS) is a web-based software solution designed to address the complexities of hospital operations through a comprehensive and integrated approach. Developed using MySQL for database management, PHP for server-side scripting, and front-end technologies such as HTML, CSS, and JavaScript, the system centralizes and automates hospital functions, enhancing overall operational efficiency and improving patient care.

The system provides various modules that streamline key hospital activities. The Patient Management module allows for efficient patient registration, updates, and tracking of treatment history, ensuring that patient information is accessible and up-to-date. The Appointment Scheduling module enables patients to book appointments, view available doctors, and receive notifications, while hospital staff can manage schedules and avoid conflicts.

The Medical Records Management module stores patient treatment histories, diagnoses, prescriptions, and other relevant health data, ensuring medical professionals have quick access to necessary information for effective treatment planning. The Billing and Financial Management module automates the process of generating invoices, tracking payments, and managing insurance claims, reducing the administrative workload and improving the accuracy of billing.

The Inventory Management module tracks hospital resources, including medical supplies and equipment, helping to ensure that the necessary materials are always available and properly utilized. The Staff Management module allows for easy management of hospital personnel, including staff scheduling, attendance tracking, and payroll management.

The Reporting and Analytics module provides hospital administrators with detailed reports on patient demographics, financial health, operational performance, and more, allowing for better resource planning and strategic decision-making.

In summary, the Hospital Management System enhances hospital management efficiency, streamlines workflows, improves patient care, and ensures better utilization of resources, contributing to overall better performance and sustainability in the healthcare sector.

3. ARCHITECTURE DIAGRAM AND TECHNOLOGIES USED

3.1 Architecture Diagram

The **Hospital Management System (HMS)** is a centralized and web-based solution designed to streamline and automate various hospital operations. The architecture of the system ensures efficient management of resources, accurate tracking of patient information, and smooth communication between different hospital departments.

The system consists of three core components:

- **Client-Side Interface (User Interface)**: The front-end of the system is designed for hospital staff, doctors, administrators, and other users. It allows them to interact with the system via a web browser. The user interface is designed to be intuitive and user-friendly, allowing individuals with varying technical expertise to navigate through tasks like scheduling appointments, managing patient records, and generating reports. Built using **HTML**, **CSS**, and **JavaScript**, the front-end is responsive and accessible across different devices, including desktops, tablets, and smartphones.
- **Server-Side Logic (Application Layer)**: On the back-end, the server-side logic processes the requests made by users from the front-end. This layer is powered by **PHP**, which handles business operations such as managing patient registration, handling appointment bookings, processing billing and payments, and generating reports. PHP is also responsible for managing the authentication and authorization of users to ensure that each individual has access only to the features they are permitted to use. It interacts with the database to retrieve, store, and update data as needed, ensuring that patient information is securely managed.
- **Database Management**: The **MySQL** database serves as the heart of the HMS, storing all critical hospital data, such as patient records, medical histories, treatment details, appointments, staff information, and billing records. MySQL is a relational database management system (RDBMS) that ensures data integrity, consistency, and easy retrieval. It supports complex queries to generate detailed reports on hospital operations, patient treatment, and financial performance. The database is securely managed with proper access controls to ensure that sensitive data is protected from unauthorized access.

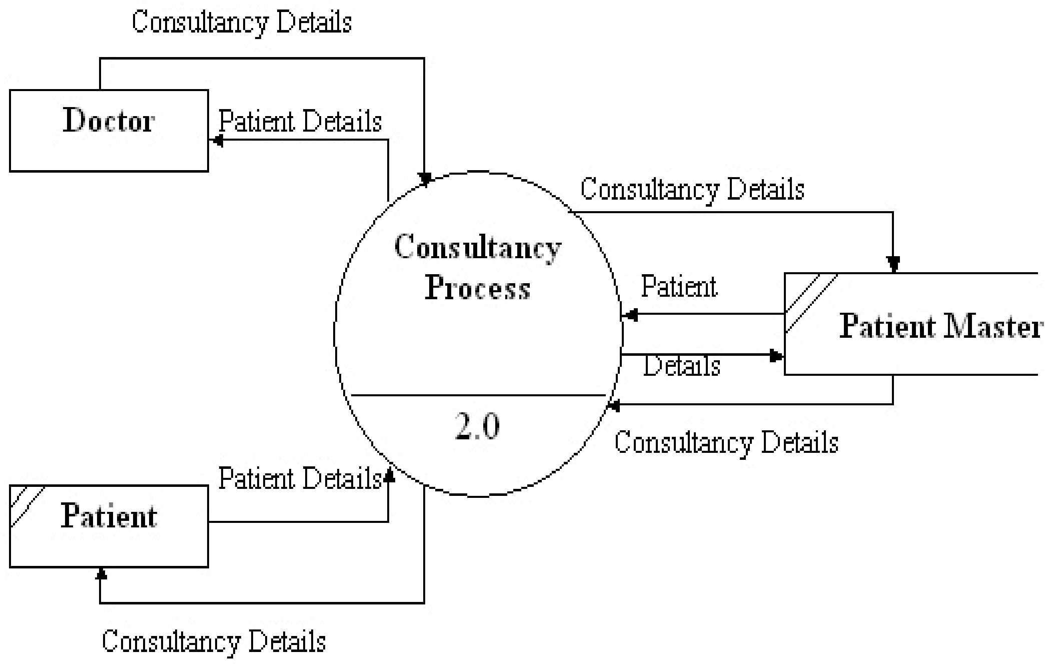


Figure 3.1
Architecture of HMS

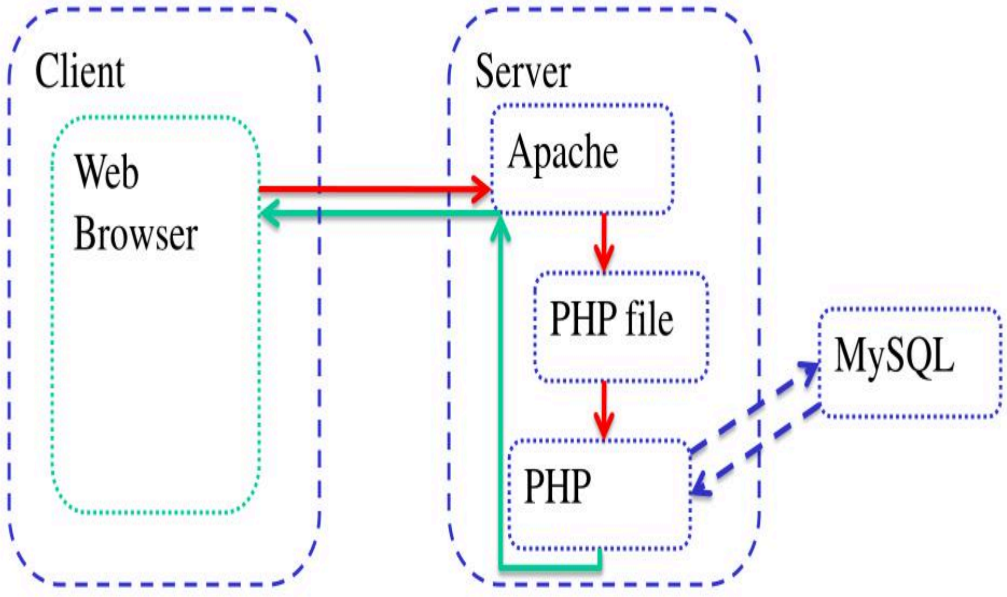


Figure 3.2
PHP and MYSQL Workflow

3.2 Technologies Used

A. **Front-End Technologies:**

- **HTML5**: Used for the structure and layout of web pages.
- **CSS3**: Used for styling and making the interface visually appealing and responsive to various screen sizes.
- **JavaScript**: Enhances user experience by adding interactivity to the system. JavaScript handles dynamic content like form validation, real-time updates of patient data, and user input handling.

B. **Back-End Technologies:**

- **PHP**: The server-side scripting language that processes requests from the front-end, interacts with the MySQL database, and generates dynamic content. It handles business logic such as appointment scheduling, patient record management, and billing calculations.

C. **Database:**

- **MySQL**: A powerful and widely-used relational database management system (RDBMS) that stores all data related to patients, hospital staff, medical records, billing, and inventory. It is used for data integrity, easy retrieval, and transaction management.

D. **Web Server:**

- **Apache/Nginx**: These are the web servers used to host the PHP application. Apache is commonly used for PHP hosting, while Nginx can be used for handling large traffic and providing high availability.

4. PROJECT PLANNING

Effective project planning is crucial to ensure that the Hospital Management System (HMS) meets the specific needs of the healthcare facility while delivering high-quality functionality, security, and user experience.

4.1 Requirements

4.1.1 Functional Requirements

These are the core functionalities the HMS must provide to meet hospital operations:

- **Patient Management**: Allow hospital staff to register new patients, update patient information, track medical histories, and view records.
- **Appointment Scheduling**: Enable patients to schedule appointments, and staff to manage and view available slots.
- **Doctor and Staff Management**: Maintain records of doctors, nurses, and other hospital staff, including schedules, shifts, and availability.
- **Billing and Payment Processing**: Automate the billing process, generate invoices, track payments, and integrate with online payment gateways.
- **Medical Records Management**: Securely store and manage patient medical records, including diagnosis, treatments, and prescriptions.
- **Reporting and Analytics**: Generate reports on various metrics, such as patient demographics, financial performance, treatment success rates, and inventory usage.
- **User Authentication and Role-Based Access**: Ensure secure access control, allowing different levels of access based on roles (e.g., doctors, nurses, administrators).

4.1.2 Non-Functional Requirements

These include quality attributes that define the overall performance and reliability of the system:

1. **Scalability**: The system should be able to handle an increasing number of patients, staff, and data as the hospital grows.
2. **Security**: Ensure data privacy and protection through encryption, secure authentication, and authorization mechanisms.

3. **Usability**: Provide a user-friendly interface that enables staff with limited technical expertise to operate the system efficiently.
4. **Performance**: Enable quick data retrieval, smooth transitions between functions, and efficient system responsiveness.
5. **Reliability**: The system should perform consistently with minimal downtime and quick recovery in case of failure.
6. **Compliance**: Ensure the system complies with health regulations and data protection laws.

4.2 Database Design

The database design is essential to ensure structured, secure, and efficient storage of all hospital-related data. For this HMS, the **MySQL** database is chosen for its reliability, scalability, and ease of use in handling large sets of structured data.

4.2.1 Database Structure

The database is divided into multiple tables, each representing a specific entity in the hospital. Key tables include:

1. **Patients Table**: Stores patient details, including patient ID, name, date of birth, contact information, medical history ID, and other personal data.
2. **Doctors Table**: Contains information about doctors, such as doctor ID, name, specialization, contact information, and assigned department.
3. **Appointments Table**: Manages scheduled appointments, including appointment ID, patient ID, doctor ID, date, time, and status.
4. **Medical Records Table**: Stores detailed medical information, such as treatment details, diagnosis, prescribed medications, and treatment outcomes.
5. **Billing Table**: Tracks billing information, including bill ID, patient ID, date, amount, status, and payment method.
6. **Staff Table**: Manages records of hospital staff, including staff ID, role (nurse, technician, etc.), contact details, and assigned shift timings.
7. **Inventory Table**: Maintains records of hospital resources, such as item ID, name, quantity, reorder level, and last updated date.
8. **User Accounts Table**: Manages user login credentials and role-based access information to ensure secure access control.

9. **Notifications Table**: Stores notification data for automated alerts, reminders, and system messages.

4.2.2 Relationships and Normalization

To avoid redundancy and ensure data integrity, the database is normalized into multiple related tables:

- **One-to-Many Relationships**: For example, one doctor may have multiple appointments, but each appointment is linked to only one doctor.
- **Many-to-Many Relationships**: For instance, a patient may see multiple doctors, and doctors may treat multiple patients. This relationship is managed through intermediary tables like **Patient-Doctor**.

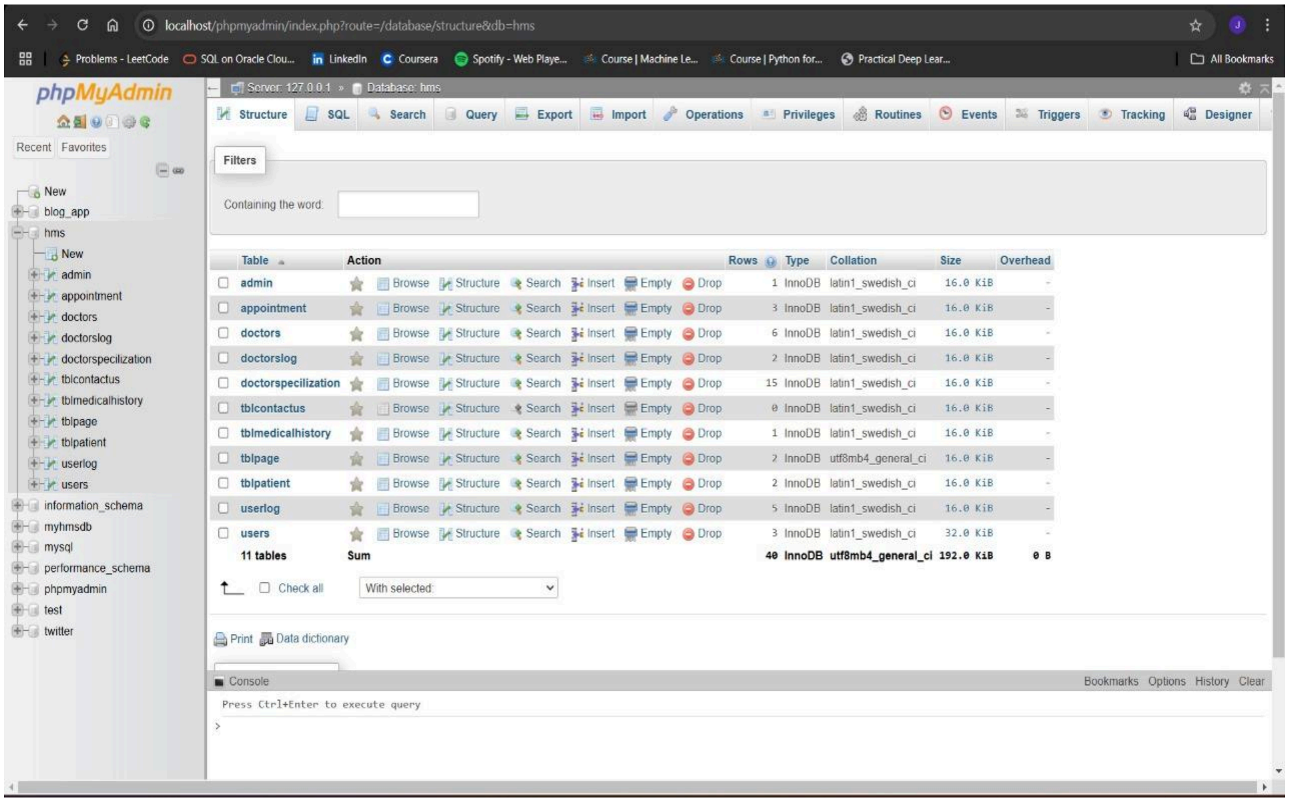
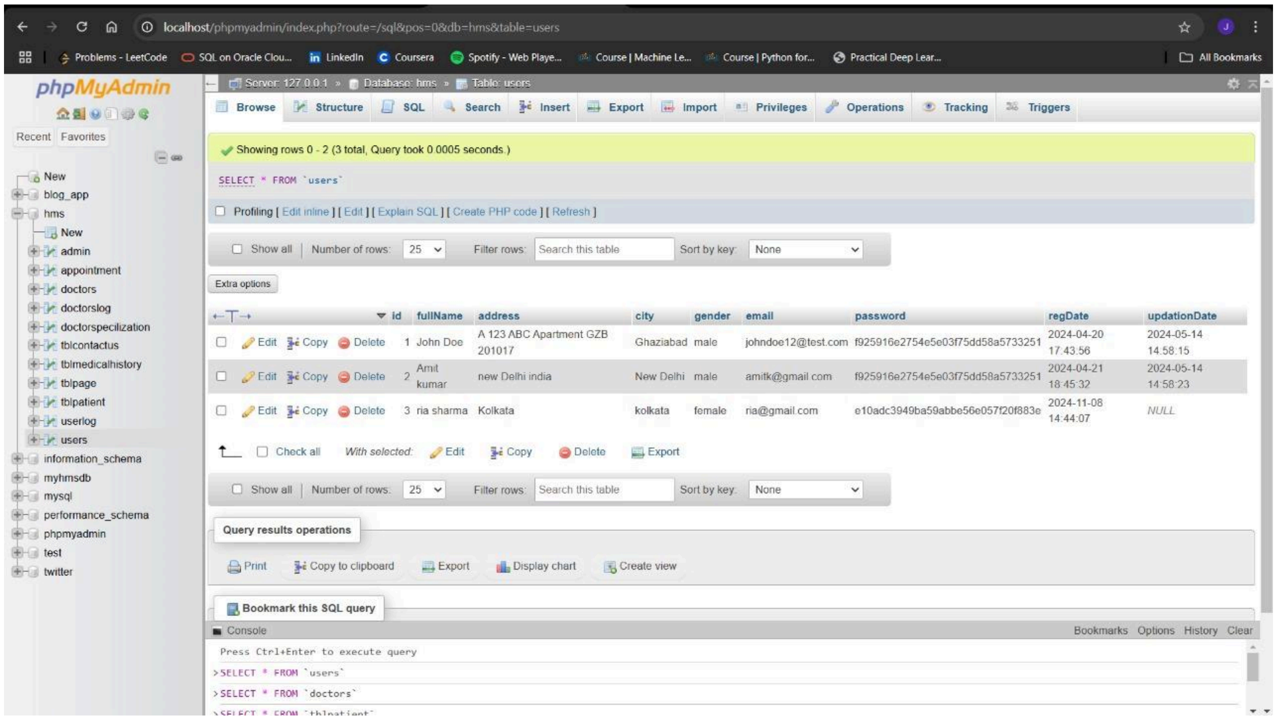

localhost/phpmyadmin/index.php?route=/database/structure&db=hms
Server: 127.0.0.1 » Database: hms
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking Designer
Filters
Containing the word:
Check all
With selected:
Print Data dictionary
Console
Press Ctrl+Enter to execute query

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
appointment	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB	-
doctors	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
doctorslog	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
doctorspecilization	Browse Structure Search Insert Empty Drop	15	InnoDB	latin1_swedish_ci	16.0 KiB	-
tblcontactus	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16.0 KiB	-
tblmedicalhistory	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 KiB	-
tblpage	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
tblpatient	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
userlog	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	32.0 KiB	-
11 tables	Sum	40	InnoDB	utf8mb4_general_ci	192.0 KiB	0 B

Figure 4.1

Database



Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

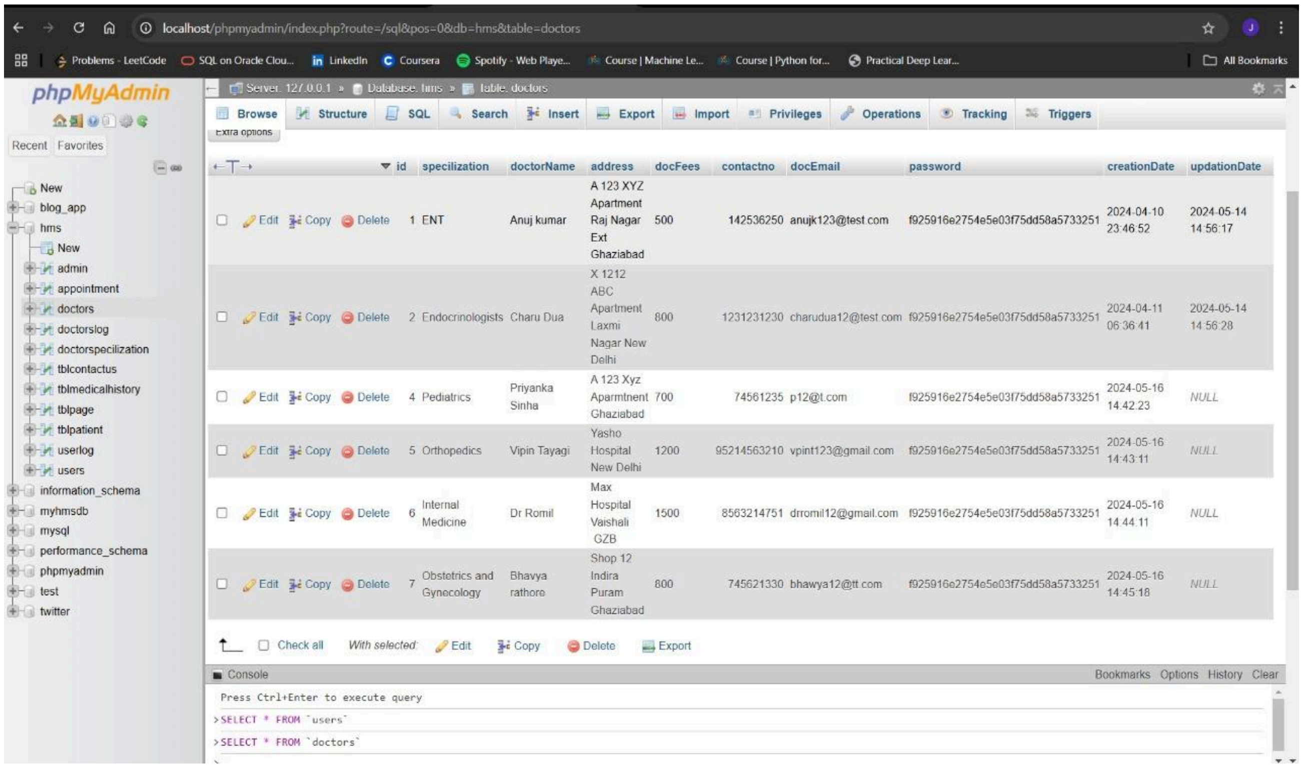
```
SELECT * FROM `users`
```

id	fullName	address	city	gender	email	password	regDate	updationDate
1	John Doe	A 123 ABC Apartment GZB 201017	Ghaziabad	male	johndoe12@test.com	f925916e2754e5e03f75dd58a5733251	2024-04-20 17:43:56	2024-05-14 14:58:15
2	Amit kumar	new Delhi india	New Delhi	male	amitk@gmail.com	f925916e2754e5e03f75dd58a5733251	2024-04-21 18:45:32	2024-05-14 14:58:23
3	ria sharma	Kolkata	kolkata	female	ria@gmail.com	e10adc3949ba59abbe56e057f20f883e	2024-11-08 14:44:07	NULL

```
Press Ctrl+Enter to execute query
>SELECT * FROM `users`
>SELECT * FROM `doctors`
>SELECT * FROM `tblpatient`
```

Figure 4.2

Patients Table

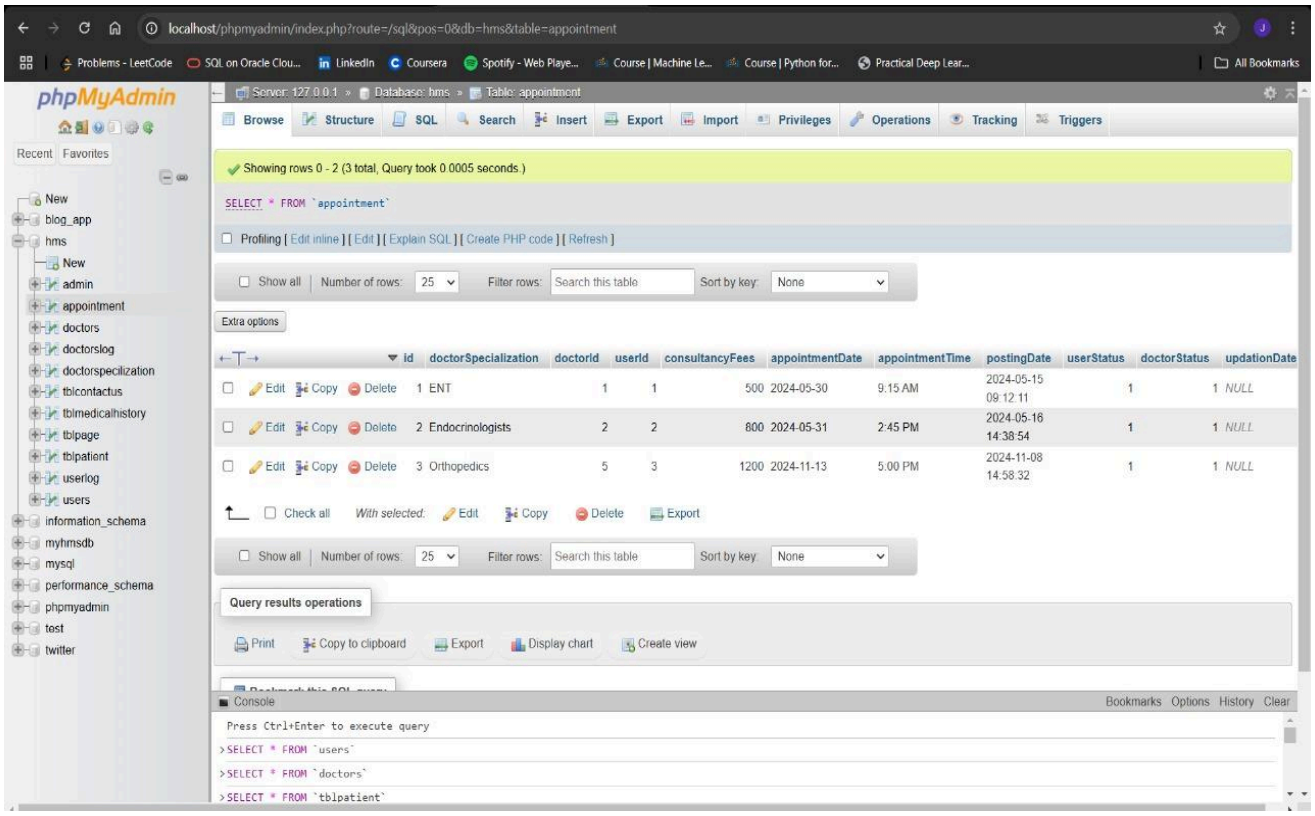


id	specilization	doctorName	address	docFees	contactno	docEmail	password	creationDate	updationDate
1	ENT	Anuj kumar	A 123 XYZ Apartment Raj Nagar Ext Ghaziabad	500	142536250	anujk123@test.com	f925916e2754e5e03f75dd58a5733251	2024-04-10 23:46:52	2024-05-14 14:56:17
2	Endocrinologists	Charu Dua	X 1212 ABC Apartment Laxmi Nagar New Delhi	800	1231231230	charudua12@test.com	f925916e2754e5e03f75dd58a5733251	2024-04-11 06:36:41	2024-05-14 14:56:28
4	Pediatrics	Priyanka Sinha	A 123 Xyz Aparmtnent Ghaziabad	700	74561235	p12@t.com	f925916e2754e5e03f75dd58a5733251	2024-05-16 14:42:23	NULL
5	Orthopedics	Vipin Tayagi	Yasho Hospital New Delhi	1200	95214563210	vpint123@gmail.com	f925916e2754e5e03f75dd58a5733251	2024-05-16 14:43:11	NULL
6	Internal Medicine	Dr Romil	Max Hospital Vaishali GZB	1500	8563214751	drromil12@gmail.com	f925916e2754e5e03f75dd58a5733251	2024-05-16 14:44:11	NULL
7	Obstetrics and Gynecology	Bhavya rathore	Shop 12 Indira Puram Ghaziabad	800	745621330	bhawya12@tt.com	f925916e2754e5e03f75dd58a5733251	2024-05-16 14:45:18	NULL

```
Press Ctrl+Enter to execute query
>SELECT * FROM `users`
>SELECT * FROM `doctors`
```

Figure 4.3

Doctors Table



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=hms&table=appointment

Server: 127.0.0.1 » Database: hms » Table: appointment

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

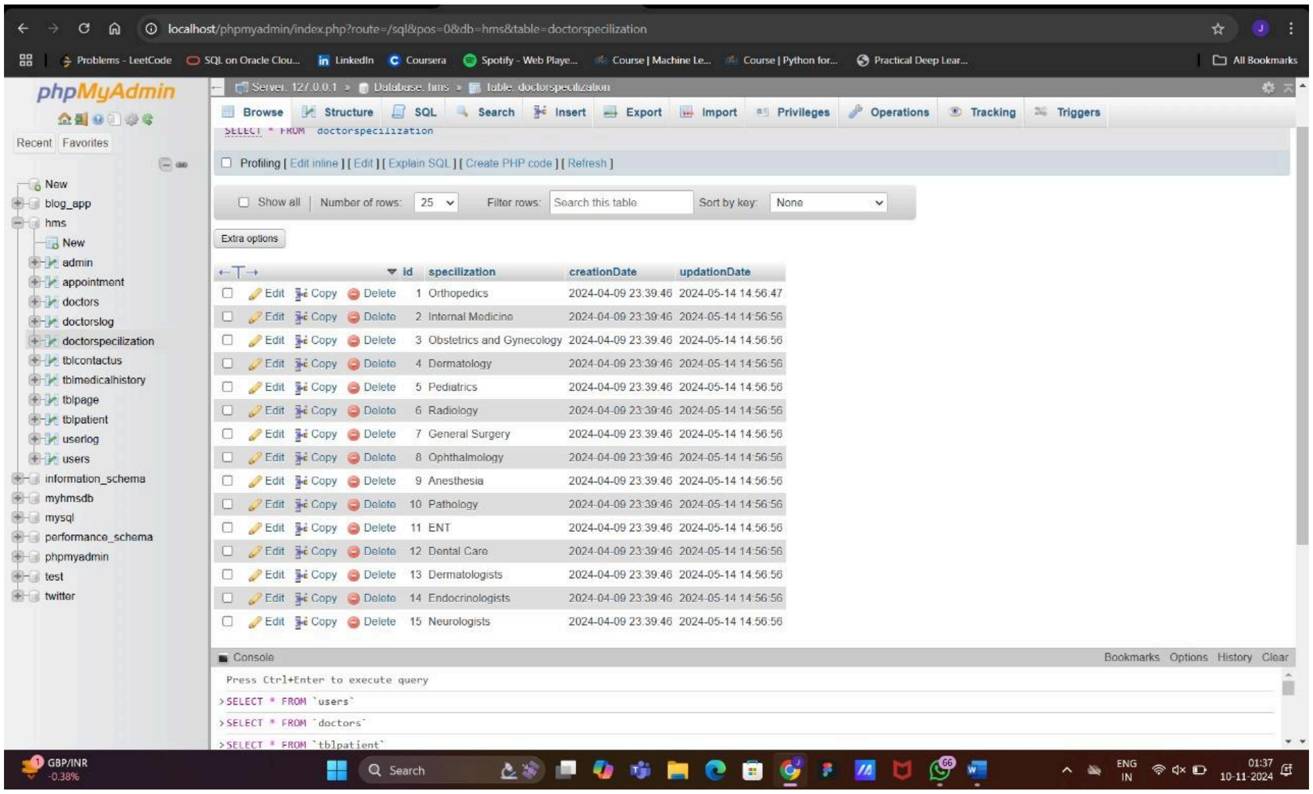
```
SELECT * FROM `appointment`
```

id	doctorSpecialization	doctorId	userId	consultancyFees	appointmentDate	appointmentTime	postingDate	userStatus	doctorStatus	updationDate
1	ENT	1	1	500	2024-05-30	9:15 AM	2024-05-15 09:12:11	1	1	NULL
2	Endocrinologists	2	2	800	2024-05-31	2:45 PM	2024-05-16 14:38:54	1	1	NULL
3	Orthopedics	5	3	1200	2024-11-13	5:00 PM	2024-11-08 14:58:32	1	1	NULL

```
Press Ctrl+Enter to execute query
>SELECT * FROM `users`
>SELECT * FROM `doctors`
>SELECT * FROM `tblpatient`
```

Figure 4.4

Appointments Table



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=hms&table=doctorspecilization

Server: 127.0.0.1 » Database: hms » Table: doctorspecilization

id	specilization	creationDate	updationDate
1	Orthopedics	2024-04-09 23:39:46	2024-05-14 14:56:47
2	Internal Medicine	2024-04-09 23:39:46	2024-05-14 14:56:56
3	Obstetrics and Gynecology	2024-04-09 23:39:46	2024-05-14 14:56:56
4	Dermatology	2024-04-09 23:39:46	2024-05-14 14:56:56
5	Pediatrics	2024-04-09 23:39:46	2024-05-14 14:56:56
6	Radiology	2024-04-09 23:39:46	2024-05-14 14:56:56
7	General Surgery	2024-04-09 23:39:46	2024-05-14 14:56:56
8	Ophthalmology	2024-04-09 23:39:46	2024-05-14 14:56:56
9	Anesthesia	2024-04-09 23:39:46	2024-05-14 14:56:56
10	Pathology	2024-04-09 23:39:46	2024-05-14 14:56:56
11	ENT	2024-04-09 23:39:46	2024-05-14 14:56:56
12	Dental Care	2024-04-09 23:39:46	2024-05-14 14:56:56
13	Dermatologists	2024-04-09 23:39:46	2024-05-14 14:56:56
14	Endocrinologists	2024-04-09 23:39:46	2024-05-14 14:56:56
15	Neurologists	2024-04-09 23:39:46	2024-05-14 14:56:56

```
Press Ctrl+Enter to execute query
>SELECT * FROM `users`
>SELECT * FROM `doctors`
>SELECT * FROM `tblpatient`
```

Figure 4.5

Doctor Specialization

5. FRONTEND DEVELOPMENT

Frontend development in the Hospital Management System (HMS) focuses on creating an accessible and user-friendly interface, enabling hospital staff, doctors, and administrators to interact with the system seamlessly. This phase involves structuring the layout, design, and interactivity of the system using HTML, CSS, and JavaScript.

5.1 HTML/CSS Structure

The HTML/CSS Structure defines the layout, appearance, and responsive design of the HMS user interface.

5.1.1 HTML Structure

HTML (Hypertext Markup Language) is used to define the structure and content of each page in the HMS interface. It organizes content into sections such as headers, navigation menus, content areas, forms, and footers. Each page is structured to be clear and easy to navigate, with dedicated sections for different functionalities:

1. **Login Page**: Allows users to securely log in to the system, entering credentials and selecting their role (e.g., doctor, nurse, administrator).
2. **Dashboard**: Displays an overview of system statistics, like patient counts, upcoming appointments, and alerts.
3. **Patient Registration and Profile**: A form for registering new patients and viewing detailed patient profiles, including medical history and contact details.
4. **Appointment Scheduling**: Allows scheduling of patient appointments with doctors, showing available slots and allowing easy booking.
5. **Medical Records**: Shows patient medical histories, including treatments, diagnoses, and prescriptions.
6. **Billing and Payment**: A section for processing patient bills and tracking payment statuses.
7. **Inventory Management**: Displays hospital resources, like medicine and equipment, with options for updating inventory levels.

Each page includes appropriate HTML elements like `<header>`, `<nav>`, `<main>`, `<section>`, `<form>`, and `<table>`, organized to provide a clean and intuitive structure for users.

5.1.2 CSS Styling

CSS (Cascading Style Sheets) is applied to enhance the visual appeal and usability of the HTML structure. Key considerations include:

- **Responsive Design**: CSS media queries ensure that the interface adjusts to different screen sizes (desktop, tablet, mobile), making the system accessible on various devices.
- **Consistent Branding and Color Scheme**: A consistent color palette, typography, and styling give the system a professional look and feel.
- **User-Friendly Layouts**: CSS grid and flexbox are used to create visually organized and well-structured layouts, making it easy for users to access information.
- **Form Styling**: Forms for patient registration, appointment booking, and billing are styled to be easy to read and fill out, with clear labels and inputs.
- **Button and Icon Design**: Buttons are styled with hover effects to provide visual feedback to users. Icons help represent actions visually (e.g., an appointment calendar or patient record icon), improving usability.

The HTML/CSS structure prioritizes simplicity and clarity to ensure that users can quickly access and understand the system's functionalities.

5.2 JavaScript

JavaScript adds interactivity and dynamic functionality to the HMS interface, enhancing the user experience by enabling real-time updates, form validation, and asynchronous operations.

5.2.1 Form Validation

JavaScript is used to validate user input on forms before they are submitted to the server. This ensures that data entered is complete and in the correct format, reducing errors and preventing incomplete data entries. Examples of form validation include:

Required Fields: Ensuring all required fields (e.g., patient name, contact information) are filled out.

Data Format Validation: Checking that email addresses, phone numbers, and dates are in the correct format.

Appointment Availability: Validating that the selected appointment time is available and doesn’t conflict with other bookings.

5.2.2 Dynamic Content and Interactivity

JavaScript enables dynamic updates to content on the page without reloading, providing a smoother user experience. Key JavaScript features include:

Appointment Scheduling: Real-time calendar integration allows users to view available slots and book appointments. JavaScript is used to check for available times and update the calendar accordingly.

Search and Filtering: Users can search for patients, appointments, and inventory items using search fields and filters, with results displayed instantly based on user input.

Inventory Alerts: JavaScript can trigger alerts when inventory levels are low, prompting the user to reorder supplies.

5.2.3 AJAX for Asynchronous Operations

AJAX (Asynchronous JavaScript and XML) is used to communicate with the server in the background, allowing data to be retrieved and displayed without refreshing the page. This is useful for:

Loading Patient Records: When a doctor or nurse selects a patient, AJAX retrieves their medical history from the database and displays it instantly.

Real-Time Data Updates: For example, when a user updates billing or inventory data, AJAX sends the data to the server, which then updates the displayed information without reloading.

Notifications and Alerts: Automated notifications, like appointment reminders, can be triggered and sent without page reloads.

5.2.4 Event Handling and User Feedback

JavaScript enables interactive elements that respond to user actions, enhancing the user experience by providing immediate feedback. For example:

Button and Link Events: When users click buttons (e.g., "Submit," "Cancel"), JavaScript manages the actions that follow, like submitting a form or resetting fields.

Navigation: JavaScript ensures smooth transitions between different sections of the interface, helping users navigate the system more intuitively.

6. BACKEND DEVELOPMENT

The backend development of the Hospital Management System (HMS) focuses on building the logic, database connectivity, and security measures that power the application's core functionalities. This layer is responsible for handling requests from the frontend, processing business logic, and interacting with the database to retrieve or store data.

6.1 API Design, Authentication, and Database Connectivity

Backend development begins with designing a robust API, setting up secure authentication mechanisms, and establishing a reliable database connection.

6.1.1 API Design

APIs (Application Programming Interfaces) allow the front-end interface to interact with the backend, sending requests and receiving responses. Key aspects of API design for the HMS include:

- RESTful API Structure: APIs are structured as RESTful services, meaning they use HTTP methods (GET, POST, PUT, DELETE) to perform actions on resources (e.g., patients, appointments, billing).
    a. GET: Retrieves information, such as patient details or appointment schedules.
    b. POST: Adds new entries, like registering a new patient or scheduling an appointment.
    c. PUT: Updates existing data, such as modifying patient information or updating billing status.
    d. DELETE: Removes records, such as canceling an appointment or deleting a patient profile.
- Endpoint Structure: Each API endpoint is designed to serve a specific purpose, with endpoints following a clear structure for ease of use. For example:
    a. `/api/patients` – handles CRUD operations for patients.
    b. `/api/appointments` – manages appointment booking, viewing, and updating.

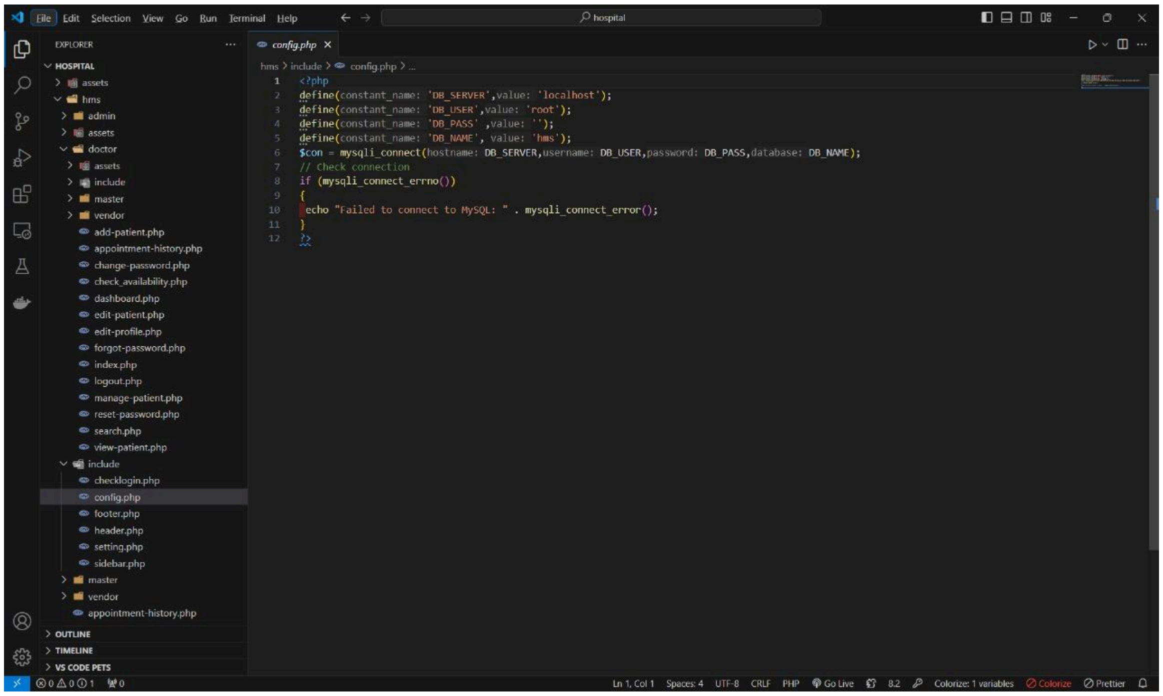
. `/api/billing` – manages payment processing and billing records.

- Response Formatting: APIs return data in JSON format for easy parsing and use by the frontend. Standardized response messages are used for success, error, and validation feedback.

6.1.3 Database Connectivity

The HMS backend is connected to a MySQL database, where patient, staff, appointment, and billing data is stored. Spring Boot provides an efficient way to establish and manage database connectivity through JDBC (Java Database Connectivity) or JPA (Java Persistence API).

- **Database Configuration**: Database connection details (e.g., URL, username, password) are stored in configuration files to establish a stable connection to MySQL.
- **Database Pooling**: Connection pooling is set up to manage multiple database connections efficiently, ensuring smooth handling of simultaneous requests.
- **Error Handling**: The backend includes error-handling mechanisms to manage database errors, such as connection timeouts or query failures, and provides appropriate responses to the client.



```php
<?php
define(constant_name: 'DB_SERVER',value: 'localhost');
define(constant_name: 'DB_USER',value: 'root');
define(constant_name: 'DB_PASS' ,value: '');
define(constant_name: 'DB_NAME', value: 'hms');
$con = mysqli_connect(hostname: DB_SERVER,username: DB_USER,password: DB_PASS,database: DB_NAME);
// Check connection
if (mysqli_connect_errno())
{
echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Figure 6.1

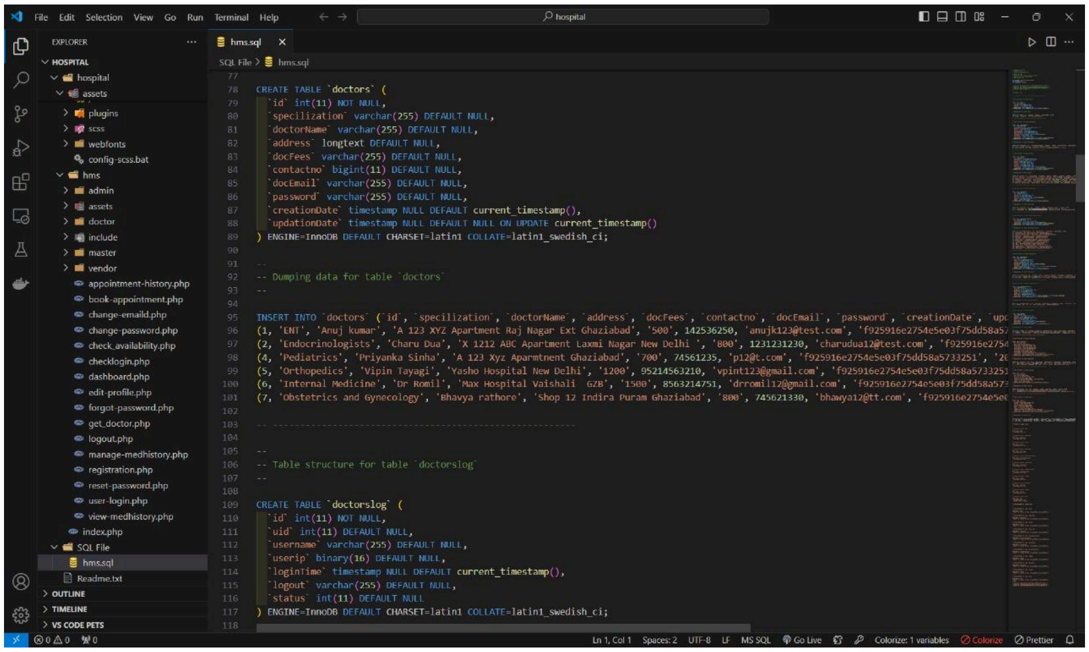
Database Connection

6.2 Database Management – CRUD Operations

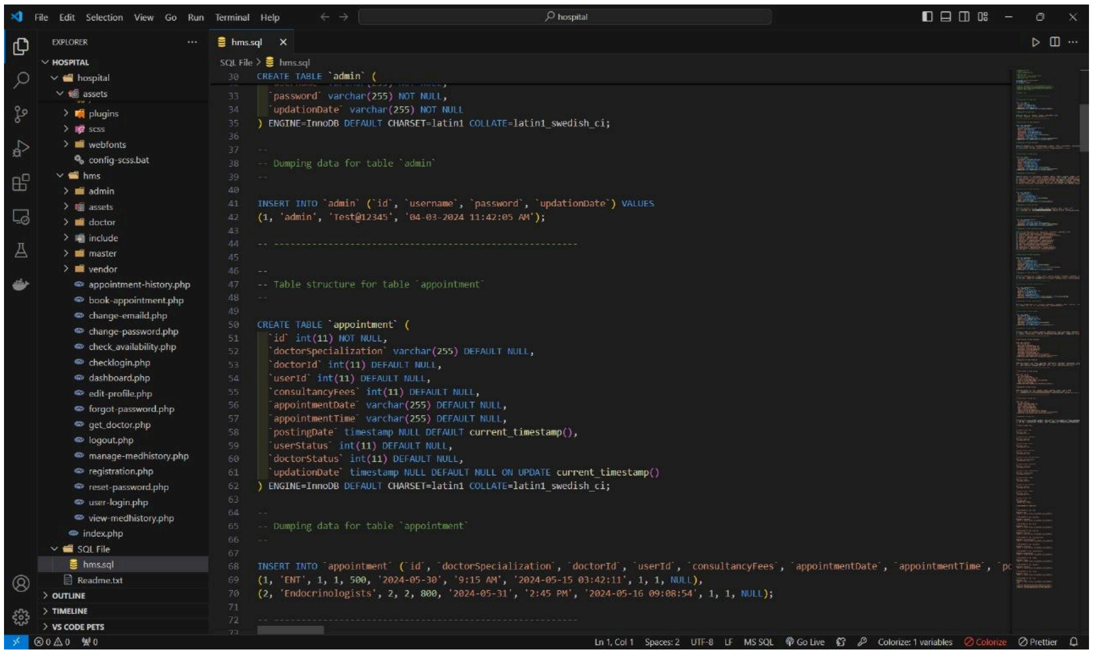
6.2.1 CRUD Operations

Each CRUD operation corresponds to a specific action in the system, enabling various functionalities such as patient registration, appointment booking, and billing.

- **Create**: This operation enables adding new records to the database. For example, when a new patient registers, a new patient record is created.

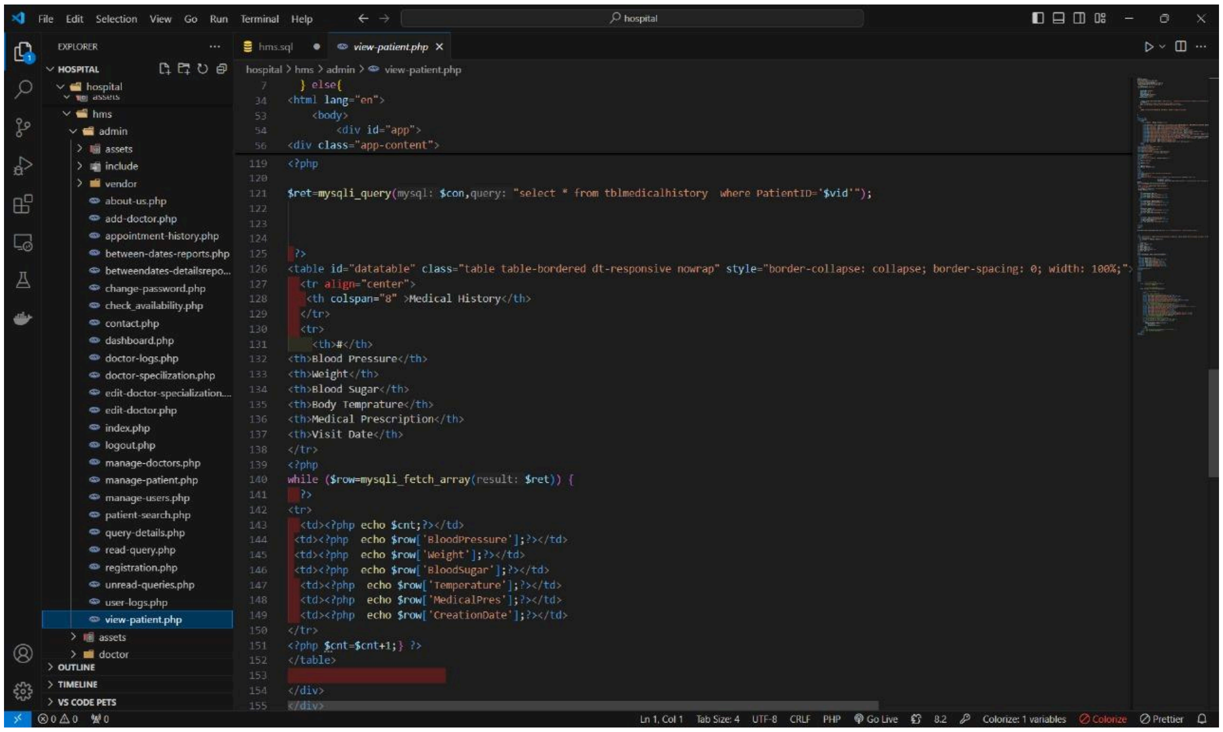


```sql

CREATE TABLE `doctors` (
  `id` int(11) NOT NULL,
  `specilization` varchar(255) DEFAULT NULL,
  `doctorName` varchar(255) DEFAULT NULL,
  `address` longtext DEFAULT NULL,
  `docFees` varchar(255) DEFAULT NULL,
  `contactno` bigint(11) DEFAULT NULL,
  `docEmail` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `creationDate` timestamp NULL DEFAULT current_timestamp(),
  `updationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--
-- Dumping data for table `doctors`
--

INSERT INTO `doctors` (`id`, `specilization`, `doctorName`, `address`, `docFees`, `contactno`, `docEmail`, `password`, `creationDate`, `upd
(1, 'ENT', 'Anuj kumar', 'A 123 XYZ Apartment Raj Nagar Ext Ghaziabad', '500', 142536250, 'anujk123@test.com', 'f925916e2754e5e03f75dd58a57
(2, 'Endocrinologists', 'Charu Dua', 'X 1212 ABC Apartment Laxmi Nagar New Delhi ', '800', 1231231230, 'charudua12@test.com', 'f925916e2754
(4, 'Pediatrics', 'Priyanka Sinha', 'A 123 Xyz Aparmtnent Ghaziabad', '700', 74561235, 'p12@t.com', 'f925916e2754e5e03f75dd58a5733251', '20
(5, 'Orthopedics', 'Vipin Tayagi', 'Yasho Hospital New Delhi', '1200', 95214563210, 'vpint123@gmail.com', 'f925916e2754e5e03f75dd58a5733251
(6, 'Internal Medicine', 'Dr Romil', 'Max Hospital Vaishali  GZB', '1500', 8563214751, 'drromil12@gmail.com', 'f925916e2754e5e03f75dd58a573
(7, 'Obstetrics and Gynecology', 'Bhavya rathore', 'Shop 12 Indira Puram Ghaziabad', '800', 745621330, 'bhawya12@tt.com', 'f925916e2754e5e0

-- --------------------------------------------------------


--
-- Table structure for table `doctorslog`
--

CREATE TABLE `doctorslog` (
  `id` int(11) NOT NULL,
  `uid` int(11) DEFAULT NULL,
  `username` varchar(255) DEFAULT NULL,
  `userip` binary(16) DEFAULT NULL,
  `loginTime` timestamp NULL DEFAULT current_timestamp(),
  `logout` varchar(255) DEFAULT NULL,
  `status` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

```



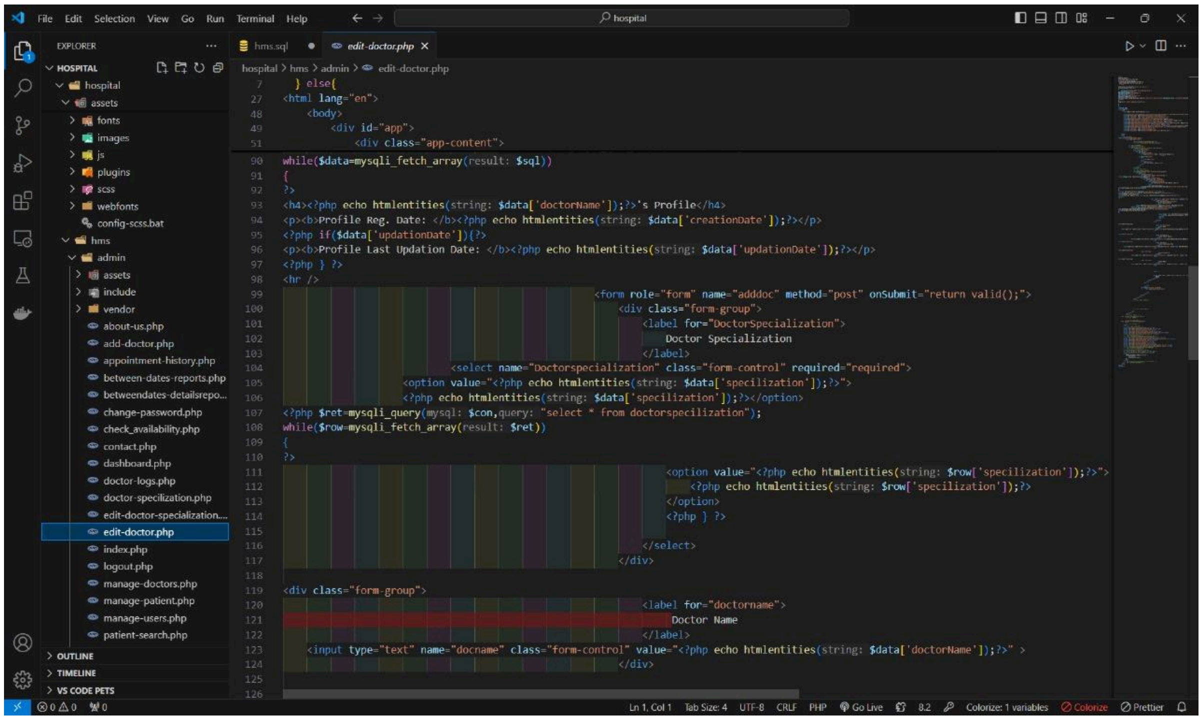
```sql
CREATE TABLE `admin` (
  `password` varchar(255) NOT NULL,
  `updationDate` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`id`, `username`, `password`, `updationDate`) VALUES
(1, 'admin', 'Test@12345', '04-03-2024 11:42:05 AM');

-- --------------------------------------------------------

--
-- Table structure for table `appointment`
--

CREATE TABLE `appointment` (
  `id` int(11) NOT NULL,
  `doctorSpecialization` varchar(255) DEFAULT NULL,
  `doctorId` int(11) DEFAULT NULL,
  `userId` int(11) DEFAULT NULL,
  `consultancyFees` int(11) DEFAULT NULL,
  `appointmentDate` varchar(255) DEFAULT NULL,
  `appointmentTime` varchar(255) DEFAULT NULL,
  `postingDate` timestamp NULL DEFAULT current_timestamp(),
  `userStatus` int(11) DEFAULT NULL,
  `doctorStatus` int(11) DEFAULT NULL,
  `updationDate` timestamp NULL DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci;

--
-- Dumping data for table `appointment`
--

INSERT INTO `appointment` (`id`, `doctorSpecialization`, `doctorId`, `userId`, `consultancyFees`, `appointmentDate`, `appointmentTime`, `pc
(1, 'ENT', 1, 1, 500, '2024-05-30', '9:15 AM', '2024-05-15 03:42:11', 1, 1, NULL),
(2, 'Endocrinologists', 2, 2, 800, '2024-05-31', '2:45 PM', '2024-05-16 09:08:54', 1, 1, NULL);

-- --------------------------------------------------------
```

- **Read**: Fetches data from the database, such as retrieving patient details, upcoming appointments, or a specific medical record.



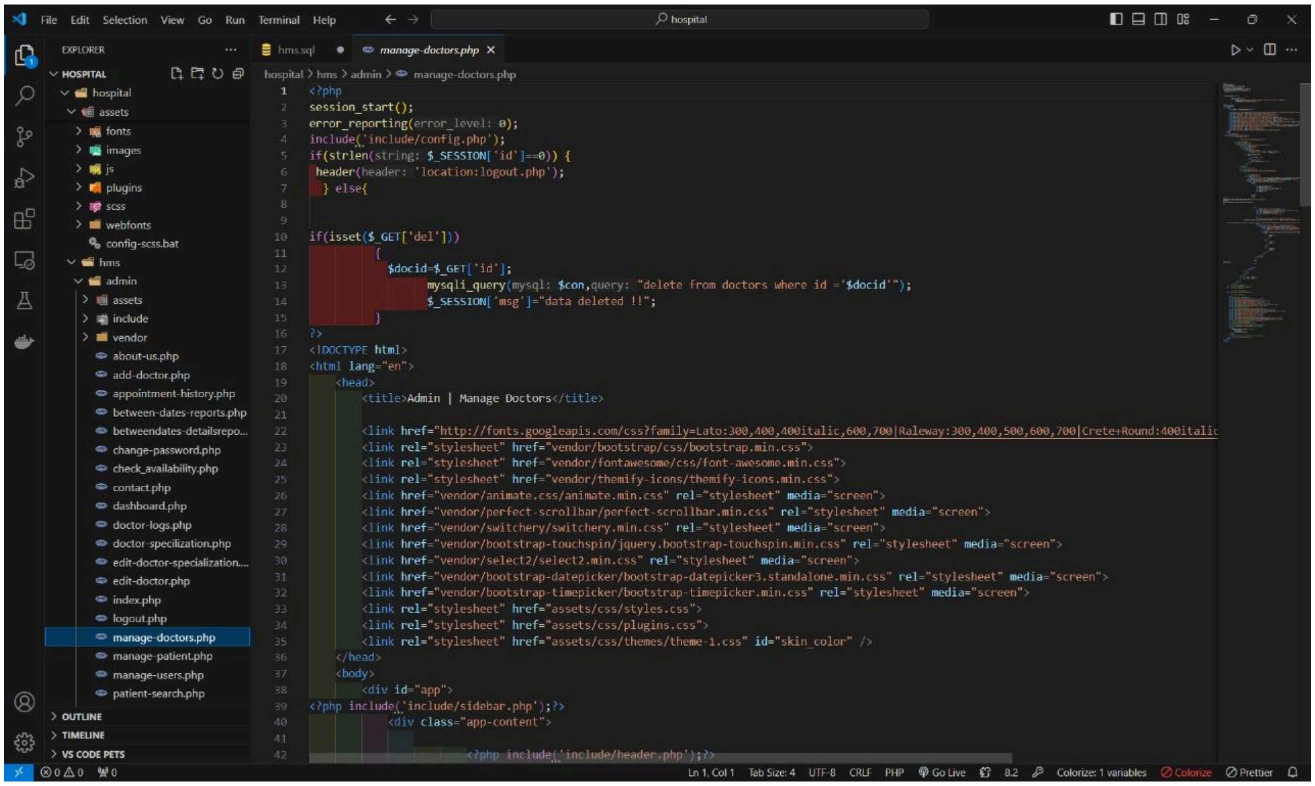
```php
 } else{
<html lang="en">
    <body>
        <div id="app">
<div class="app-content">
<?php

$ret=mysqli_query(mysql: $con,query: "select * from tblmedicalhistory  where PatientID='$vid'");



?>
<table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 100%;">
  <tr align="center">
   <th colspan="8" >Medical History</th>
  </tr>
  <tr>
    <th>#</th>
<th>Blood Pressure</th>
<th>Weight</th>
<th>Blood Sugar</th>
<th>Body Temprature</th>
<th>Medical Prescription</th>
<th>Visit Date</th>
</tr>
<?php
while ($row=mysqli_fetch_array(result: $ret)) {
  ?>
<tr>
  <td><?php echo $cnt;?></td>
 <td><?php  echo $row['BloodPressure'];?></td>
 <td><?php  echo $row['Weight'];?></td>
 <td><?php  echo $row['BloodSugar'];?></td>
  <td><?php  echo $row['Temperature'];?></td>
  <td><?php  echo $row['MedicalPres'];?></td>
  <td><?php  echo $row['CreationDate'];?></td>
</tr>
<?php $cnt=$cnt+1;} ?>
</table>

</div>
</div>
```

- **Update**: Modifies existing data, such as updating patient information or changing an appointment's date and time.



```php
 } else{
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
while($data=mysqli_fetch_array(result: $sql))
{
?>
<h4><?php echo htmlentities(string: $data['doctorName']);?>'s Profile</h4>
<p><b>Profile Reg. Date: </b><?php echo htmlentities(string: $data['creationDate']);?></p>
<?php if($data['updationDate']){?>
<p><b>Profile Last Updation Date: </b><?php echo htmlentities(string: $data['updationDate']);?></p>
<?php } ?>
<hr />
                                                  <form role="form" name="adddoc" method="post" onSubmit="return valid();">
                                                      <div class="form-group">
                                                          <label for="DoctorSpecialization">
                                                              Doctor Specialization
                                                          </label>
                                      <select name="Doctorspecialization" class="form-control" required="required">
                              <option value="<?php echo htmlentities(string: $data['specilization']);?>">
                              <?php echo htmlentities(string: $data['specilization']);?></option>
<?php $ret=mysqli_query(mysql: $con,query: "select * from doctorspecilization");
while($row=mysqli_fetch_array(result: $ret))
{
?>
                                                              <option value="<?php echo htmlentities(string: $row['specilization']);?>">
                                                                  <?php echo htmlentities(string: $row['specilization']);?>
                                                              </option>
                                                              <?php } ?>

                                                          </select>
                                                      </div>

<div class="form-group">
                                                          <label for="doctorname">
                                                              Doctor Name
                                                          </label>
    <input type="text" name="docname" class="form-control" value="<?php echo htmlentities(string: $data['doctorName']);?>" >
                                                      </div>
```

- **Delete**: Removes records from the database, like deleting a patient profile or canceling an appointment.



```php
<?php
session_start();
error_reporting(error_level: 0);
include('include/config.php');
if(strlen(string: $_SESSION['id']==0)) {
 header(header: 'location:logout.php');
  } else{


if(isset($_GET['del']))
          {
            $docid=$_GET['id'];
                  mysqli_query(mysql: $con,query: "delete from doctors where id ='$docid'");
                  $_SESSION['msg']="data deleted !!";
          }
?>
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Admin | Manage Doctors</title>

        <link href="http://fonts.googleapis.com/css?family=Lato:300,400,400italic,600,700|Raleway:300,400,500,600,700|Crete+Round:400italic
        <link rel="stylesheet" href="vendor/bootstrap/css/bootstrap.min.css">
        <link rel="stylesheet" href="vendor/fontawesome/css/font-awesome.min.css">
        <link rel="stylesheet" href="vendor/themify-icons/themify-icons.min.css">
        <link href="vendor/animate.css/animate.min.css" rel="stylesheet" media="screen">
        <link href="vendor/perfect-scrollbar/perfect-scrollbar.min.css" rel="stylesheet" media="screen">
        <link href="vendor/switchery/switchery.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-touchspin/jquery.bootstrap-touchspin.min.css" rel="stylesheet" media="screen">
        <link href="vendor/select2/select2.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-datepicker/bootstrap-datepicker3.standalone.min.css" rel="stylesheet" media="screen">
        <link href="vendor/bootstrap-timepicker/bootstrap-timepicker.min.css" rel="stylesheet" media="screen">
        <link rel="stylesheet" href="assets/css/styles.css">
        <link rel="stylesheet" href="assets/css/plugins.css">
        <link rel="stylesheet" href="assets/css/themes/theme-1.css" id="skin_color" />
    </head>
    <body>
        <div id="app">
<?php include('include/sidebar.php');?>
            <div class="app-content">

                    <?php include('include/header.php');?>
```

6.2.2 Additional Database Operations

In addition to basic CRUD operations, the HMS backend includes more complex database functions tailored to the system's specific requirements:

- **Search and Filtering**: Advanced search functions allow users to find patients by name, filter appointments by date or doctor, and view specific records.
- **Aggregate Functions**: Summary reports and analytics, such as total patient count or revenue generated, are generated using SQL aggregate functions like `SUM`, `COUNT`, and `AVG`.
- **Pagination and Sorting**: For large datasets, pagination and sorting are implemented to make it easy to navigate patient records, appointment logs, and inventory lists.

7. TESTING AND DEPLOYMENT

7.1 Testing

Testing is divided into various phases to ensure all system components meet quality and security standards.

- **Unit Testing**: This involves testing individual functions, classes, or components to ensure they perform correctly. For the HMS, unit tests validate core features like patient registration, appointment scheduling, and data retrieval. By isolating each function, unit testing allows developers to identify and resolve errors at a granular level, which simplifies debugging and ensures each feature works as expected.
- **Integration Testing**: Once individual components are verified, integration testing ensures they work together seamlessly. This is particularly important for the HMS, where interactions between the frontend, backend API, and database need to be smooth and error-free. Integration testing checks data flow between modules, such as ensuring appointment details entered on the frontend correctly update in the backend database.
- **Functional Testing**: Functional testing assesses the system against its functional requirements. It tests that every feature, such as patient profile management, billing, and appointment booking, functions according to the specified requirements. This step also verifies that different user roles (e.g., doctors, nurses, administrators) have the appropriate access levels and permissions to perform their tasks.
- **User Interface (UI) Testing**: UI testing ensures that the system's interface is user-friendly, intuitive, and consistent. This step checks responsiveness, ease of navigation, and that UI elements (like forms and buttons) respond correctly to user actions. Given that hospital staff use the system for daily tasks, UI testing is crucial to providing a positive user experience.
- **Performance Testing**: Performance testing evaluates how well the HMS handles load and responds to user requests. This includes **load testing**, which simulates multiple users accessing the system simultaneously, and **stress testing**, which pushes the system beyond normal conditions to check stability under high demand. Performance testing is essential for hospital management systems, as they often need to handle high volumes of data and users.
- **Security Testing**: Security testing identifies and addresses potential vulnerabilities, protecting sensitive patient and hospital data. This step includes checks for SQL injection, cross-site scripting (XSS), and token-based authentication mechanisms. A robust security testing process ensures that only authorized users can access protected information, and sensitive data is encrypted and stored securely.

7.2 Deployment

Deployment involves making the HMS available to users in a live environment. It includes configuring the production environment, deploying the application code, and testing in real-world conditions.

- **Preparing the Deployment Environment**: The deployment server needs to be configured with the necessary software and security protocols. Web servers (e.g., Apache, Nginx) and database configurations are set up, along with environment variables for secure connections. Connection pooling and caching mechanisms are also configured to ensure optimal performance.
- **Deployment Process**: The application code is built, packaged, and transferred to the server. The database is configured, with any necessary migrations applied to ensure data consistency. Once the code is deployed, basic **smoke testing** is conducted to verify that critical functions like user authentication, database connections, and API calls are working in the live environment.

Through thorough testing and structured deployment, the Hospital Management System is optimized for performance, security, and usability, providing a reliable solution that meets the complex needs of hospital management.

8. RESULTS AND CONCLUSION

8.1 Conclusion

The Hospital Management System (HMS) is designed to streamline hospital operations, improve patient care, and ensure efficient management of resources. By integrating the HMS with secure database management, intuitive user interfaces, and automated functions, the system enables real-time data handling, which is essential in today's healthcare landscape. Key modules—such as patient registration, appointment scheduling, billing, and reporting—work in harmony to enhance operational efficiency, reduce paperwork, and minimize errors. The successful implementation of HMS demonstrates how technology can transform healthcare processes, making them more reliable, accessible, and productive for medical staff and administrators alike.

By automating tasks and ensuring secure data handling, the HMS contributes to improving the quality of healthcare services. Additionally, it provides a foundation for scalable growth as the needs of hospitals expand, with data-driven insights that can be used to inform strategic planning and improve patient outcomes. This project demonstrates that a thoughtfully designed hospital management system can provide substantial benefits across patient experience, healthcare administration, and clinical effectiveness.

8.2 Future Scope

The HMS has a promising future scope, with numerous opportunities for enhancement and expansion. Potential developments include:

- **Integration with IoT and Wearable Devices**: By connecting with IoT-enabled medical devices, the HMS can automatically collect real-time patient data (e.g., heart rate, blood pressure), improving monitoring and reducing manual data entry.
- **Enhanced Data Analytics**: Advanced analytics could provide actionable insights for hospital administrators, such as predictive analytics to forecast patient inflow and staffing needs, or data visualization for identifying trends in patient outcomes and resource usage.
- **Telemedicine Integration**: As telehealth continues to grow, integrating telemedicine capabilities within the HMS can allow virtual consultations, remote monitoring, and patient follow-up care, making healthcare more accessible to remote patients.
- **AI and Machine Learning**: AI-powered diagnostics, predictive analysis, and patient management can improve operational efficiency and enable personalized treatment plans based on historical patient data and clinical patterns.

- **Mobile Application Development**: A mobile version of the HMS would allow patients to access their medical records, schedule appointments, and receive alerts on medications or upcoming visits, improving patient engagement and satisfaction.
- **Blockchain for Secure Data Management**: Integrating blockchain technology could enhance data security, ensuring that patient information is stored and shared securely and that access to sensitive information is well-regulated and transparent.
- **Multi-Language and Accessibility Support**: Adding multi-language support and accessibility features would make the system more user-friendly, accommodating diverse patient populations and ensuring that staff and patients can use it effectively.

With ongoing advancements in technology, the HMS can continuously evolve to meet emerging healthcare needs, making it an adaptable and essential tool for modern hospitals and clinics.

REFERENCES

1. Web References:

- W3Schools:
  - ➔ PHP Tutorial: https://www.w3schools.com/php/
  - ➔ MySQL Tutorial: https://www.w3schools.com/mysql/
  - ➔ HTML Tutorial: https://www.w3schools.com/html/
  - ➔ CSS Tutorial: https://www.w3schools.com/css/
  - ➔ JavaScript Tutorial: https://www.w3schools.com/js/
- MDN Web Docs: Web Development Learning: https://developer.mozilla.org/en-US/
- PHP.net: https://www.php.net/
- MySQL Documentation:https://dev.mysql.com/doc/
- Stack Overflow: https://stackoverflow.com/

2. Book References:

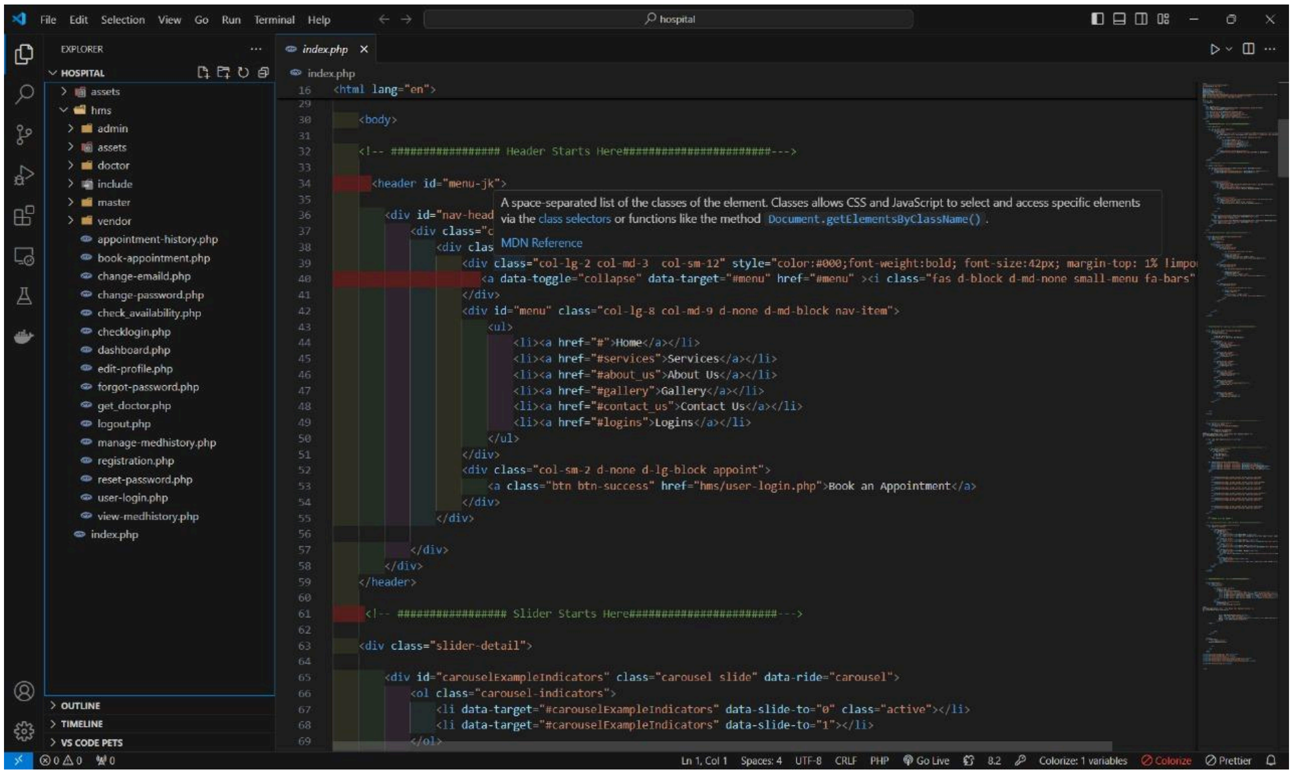
- Head First PHP & MySQL by Craig Walls
- PHP & MySQL Web Development by Dinesh Rajput

Journals:

- *Welling, L. and Thomson, L., 2003. PHP and MySQL Web development. Sams publishing.*
- *Groene, O., Botje, D., Suñol, R., Lopez, M.A. and Wagner, C., 2013. A systematic review of instruments that assess the implementation of hospital quality management systems. International journal for quality in health care, 25(5), pp.525-541.*

APPENDIX-1 :CODE

- **Index.php**

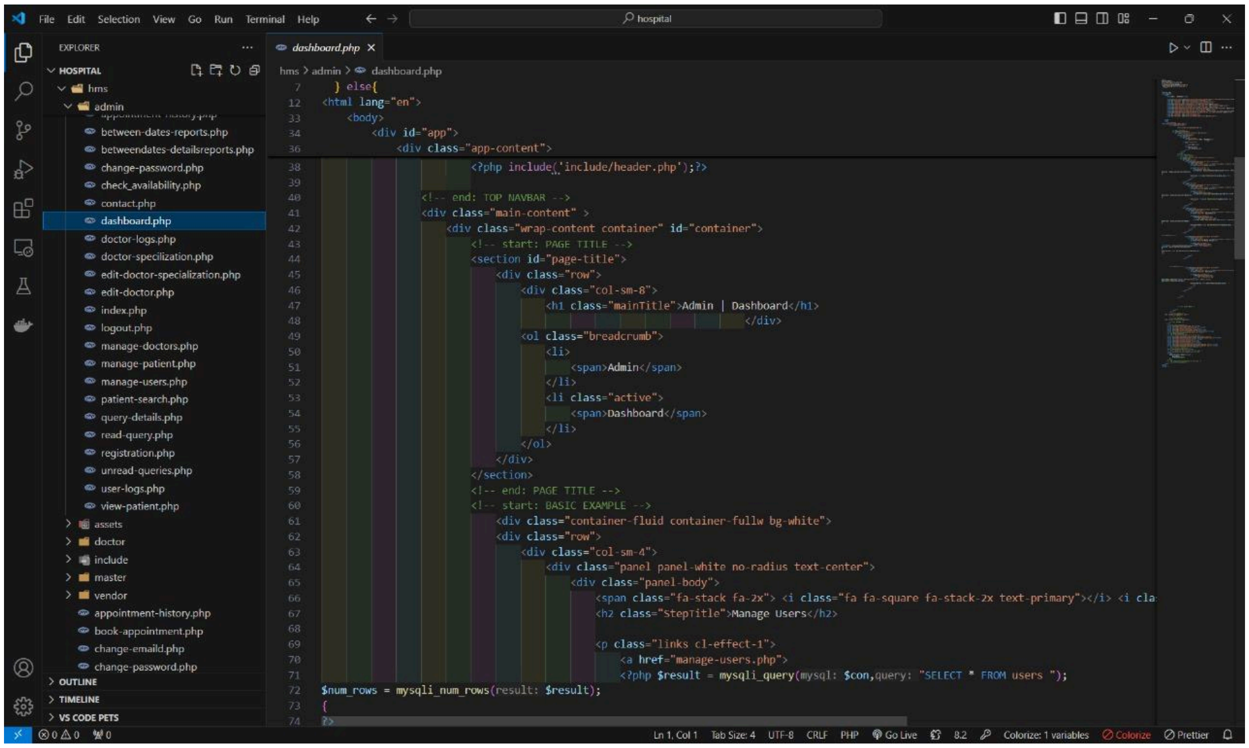


```
<html lang="en">

    <body>

    <!-- ################# Header Starts Here#######################--->

      <header id="menu-jk">

        <div id="nav-head
            <div class="c
                <div clas
                    <div class="col-lg-2 col-md-3  col-sm-12" style="color:#000;font-weight:bold; font-size:42px; margin-top: 1% !impo
                       <a data-toggle="collapse" data-target="#menu" href="#menu" ><i class="fas d-block d-md-none small-menu fa-bars"
                    </div>
                    <div id="menu" class="col-lg-8 col-md-9 d-none d-md-block nav-item">
                        <ul>
                            <li><a href="#">Home</a></li>
                            <li><a href="#services">Services</a></li>
                            <li><a href="#about_us">About Us</a></li>
                            <li><a href="#gallery">Gallery</a></li>
                            <li><a href="#contact_us">Contact Us</a></li>
                            <li><a href="#logins">Logins</a></li>
                        </ul>
                    </div>
                    <div class="col-sm-2 d-none d-lg-block appoint">
                        <a class="btn btn-success" href="hms/user-login.php">Book an Appointment</a>
                    </div>
                </div>

            </div>
        </div>
    </header>

     <!-- ################# Slider Starts Here#######################--->

    <div class="slider-detail">

        <div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
            <ol class="carousel-indicators">
                <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
                <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
            </ol>
```

A space-separated list of the classes of the element. Classes allows CSS and JavaScript to select and access specific elements via the class selectors or functions like the method `Document.getElementsByClassName()`.

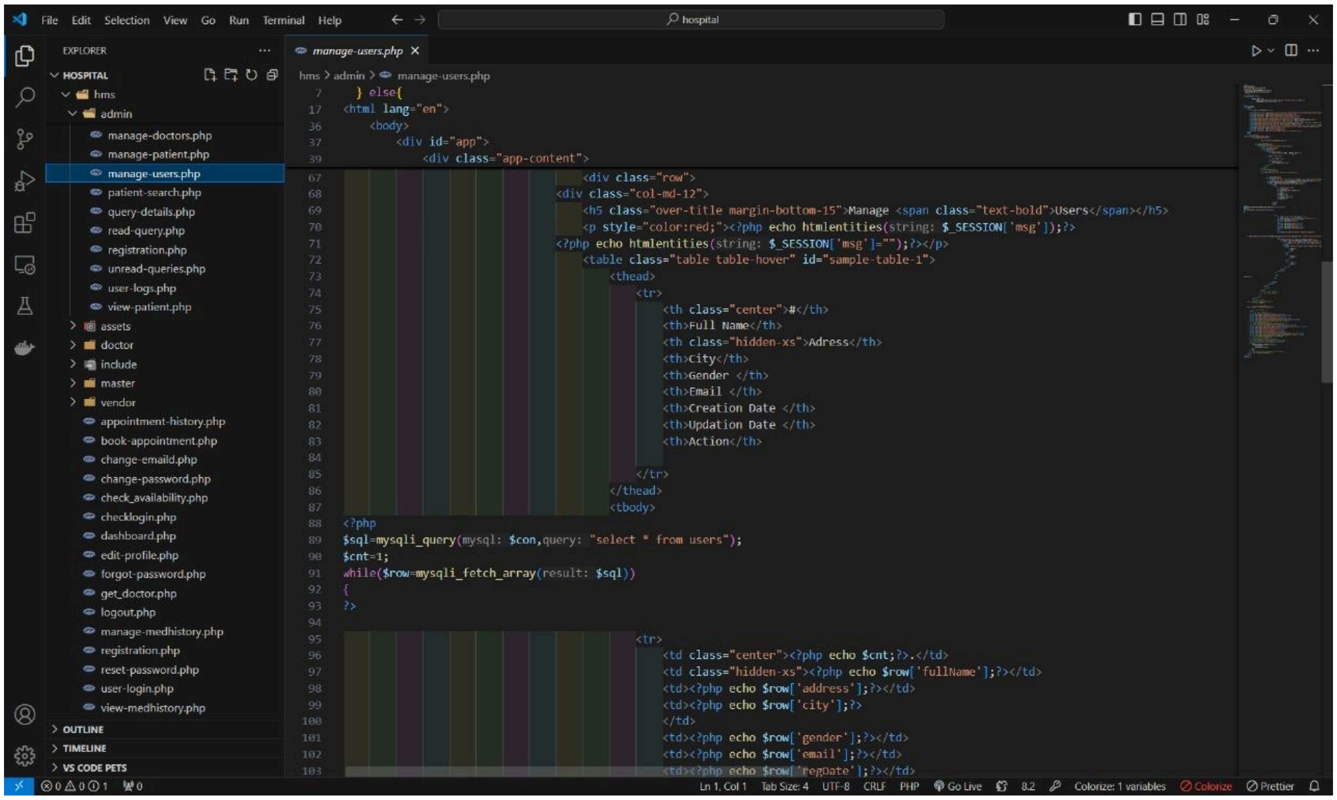
MDN Reference

- **Admin Dashboard**



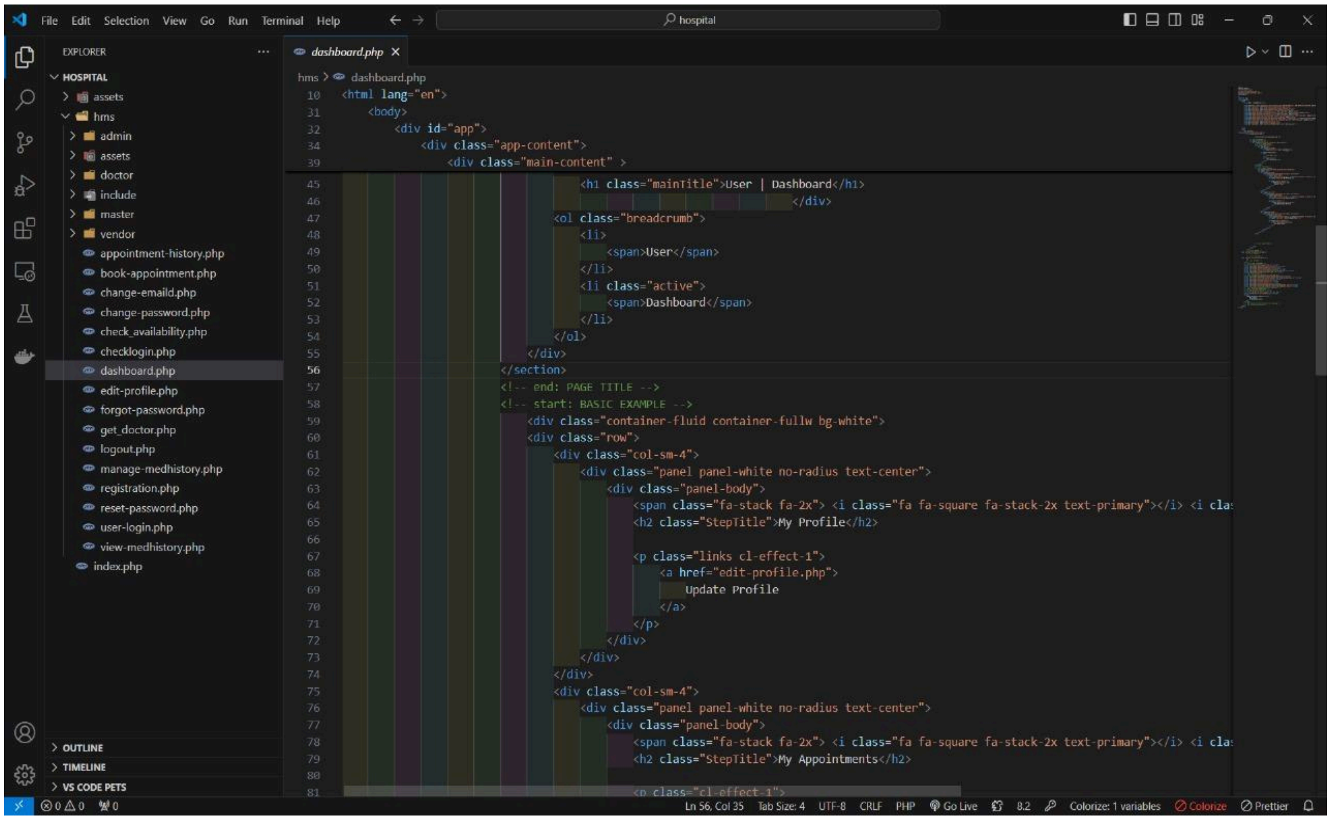
```
  } else{
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
                    <?php include('include/header.php');?>

                <!-- end: TOP NAVBAR -->
                <div class="main-content" >
                    <div class="wrap-content container" id="container">
                        <!-- start: PAGE TITLE -->
                        <section id="page-title">
                            <div class="row">
                                <div class="col-sm-8">
                                    <h1 class="mainTitle">Admin | Dashboard</h1>
                                                                </div>
                                <ol class="breadcrumb">
                                    <li>
                                        <span>Admin</span>
                                    </li>
                                    <li class="active">
                                        <span>Dashboard</span>
                                    </li>
                                </ol>
                            </div>
                        </section>
                        <!-- end: PAGE TITLE -->
                        <!-- start: BASIC EXAMPLE -->
                            <div class="container-fluid container-fullw bg-white">
                            <div class="row">
                                <div class="col-sm-4">
                                    <div class="panel panel-white no-radius text-center">
                                        <div class="panel-body">
                                            <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i cla
                                            <h2 class="StepTitle">Manage Users</h2>

                                            <p class="links cl-effect-1">
                                                <a href="manage-users.php">
                                                <?php $result = mysqli_query(mysql: $con,query: "SELECT * FROM users ");
$num_rows = mysqli_num_rows(result: $result);
{
?>
```

- **Manage Users**



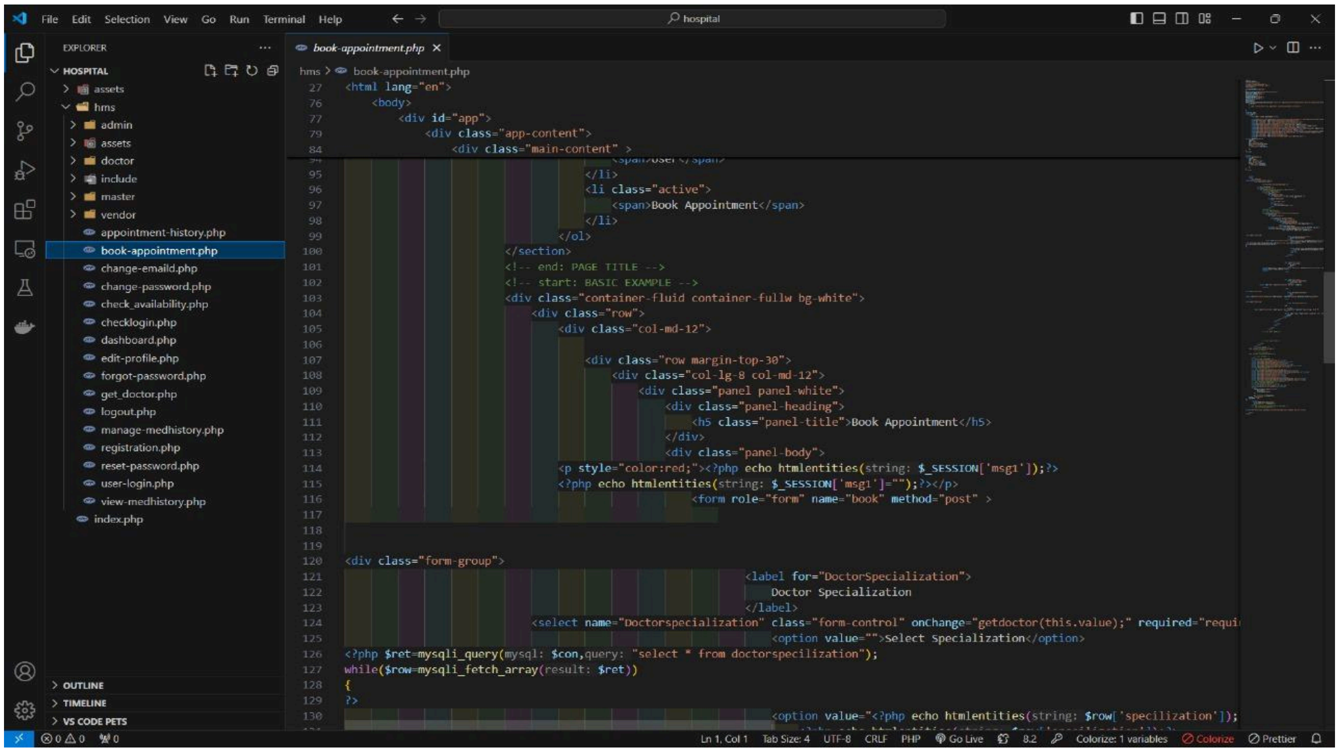
```
  } else{
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
                                    <div class="row">
                                <div class="col-md-12">
                                    <h5 class="over-title margin-bottom-15">Manage <span class="text-bold">Users</span></h5>
                                    <p style="color:red;"><?php echo htmlentities(string: $_SESSION['msg']);?>
                                <?php echo htmlentities(string: $_SESSION['msg']="");?></p>
                                    <table class="table table-hover" id="sample-table-1">
                                        <thead>
                                            <tr>
                                                <th class="center">#</th>
                                                <th>Full Name</th>
                                                <th class="hidden-xs">Adress</th>
                                                <th>City</th>
                                                <th>Gender </th>
                                                <th>Email </th>
                                                <th>Creation Date </th>
                                                <th>Updation Date </th>
                                                <th>Action</th>

                                            </tr>
                                        </thead>
                                        <tbody>
<?php
$sql=mysqli_query(mysql: $con,query: "select * from users");
$cnt=1;
while($row=mysqli_fetch_array(result: $sql))
{
?>

                                            <tr>
                                                <td class="center"><?php echo $cnt;?>.</td>
                                                <td class="hidden-xs"><?php echo $row['fullName'];?></td>
                                                <td><?php echo $row['address'];?></td>
                                                <td><?php echo $row['city'];?>
                                                </td>
                                                <td><?php echo $row['gender'];?></td>
                                                <td><?php echo $row['email'];?></td>
                                                <td><?php echo $row['regDate'];?></td>
```

- **Patients Dashboard**



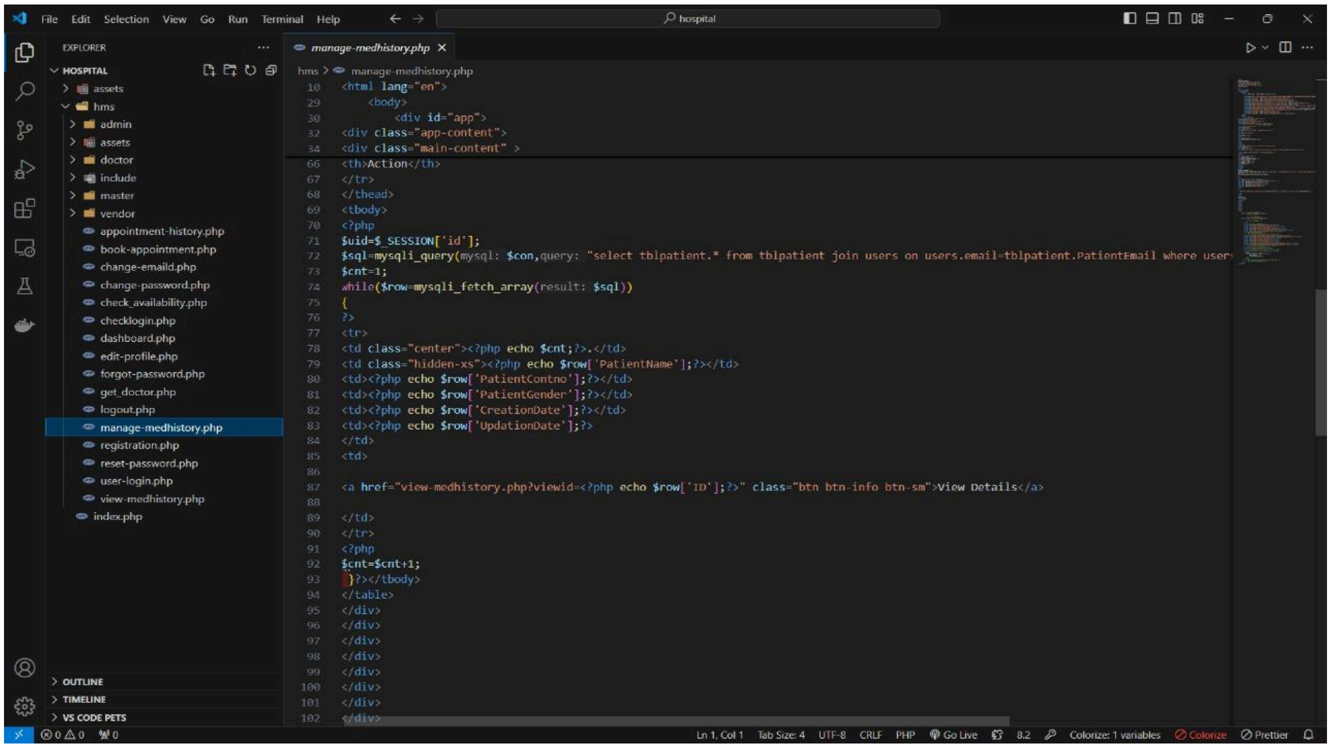
```
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
                <div class="main-content" >
                                    <h1 class="mainTitle">User | Dashboard</h1>
                                                                    </div>
                                <ol class="breadcrumb">
                                    <li>
                                        <span>User</span>
                                    </li>
                                    <li class="active">
                                        <span>Dashboard</span>
                                    </li>
                                </ol>
                            </div>
                        </section>
                        <!-- end: PAGE TITLE -->
                        <!-- start: BASIC EXAMPLE -->
                            <div class="container-fluid container-fullw bg-white">
                            <div class="row">
                                <div class="col-sm-4">
                                    <div class="panel panel-white no-radius text-center">
                                        <div class="panel-body">
                                            <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i cla
                                            <h2 class="StepTitle">My Profile</h2>

                                            <p class="links cl-effect-1">
                                                <a href="edit-profile.php">
                                                    Update Profile
                                                </a>
                                            </p>
                                        </div>
                                    </div>
                                </div>
                                <div class="col-sm-4">
                                    <div class="panel panel-white no-radius text-center">
                                        <div class="panel-body">
                                            <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i cla
                                            <h2 class="StepTitle">My Appointments</h2>

                                            <p class="cl-effect-1">
```

- **Book Appointment**



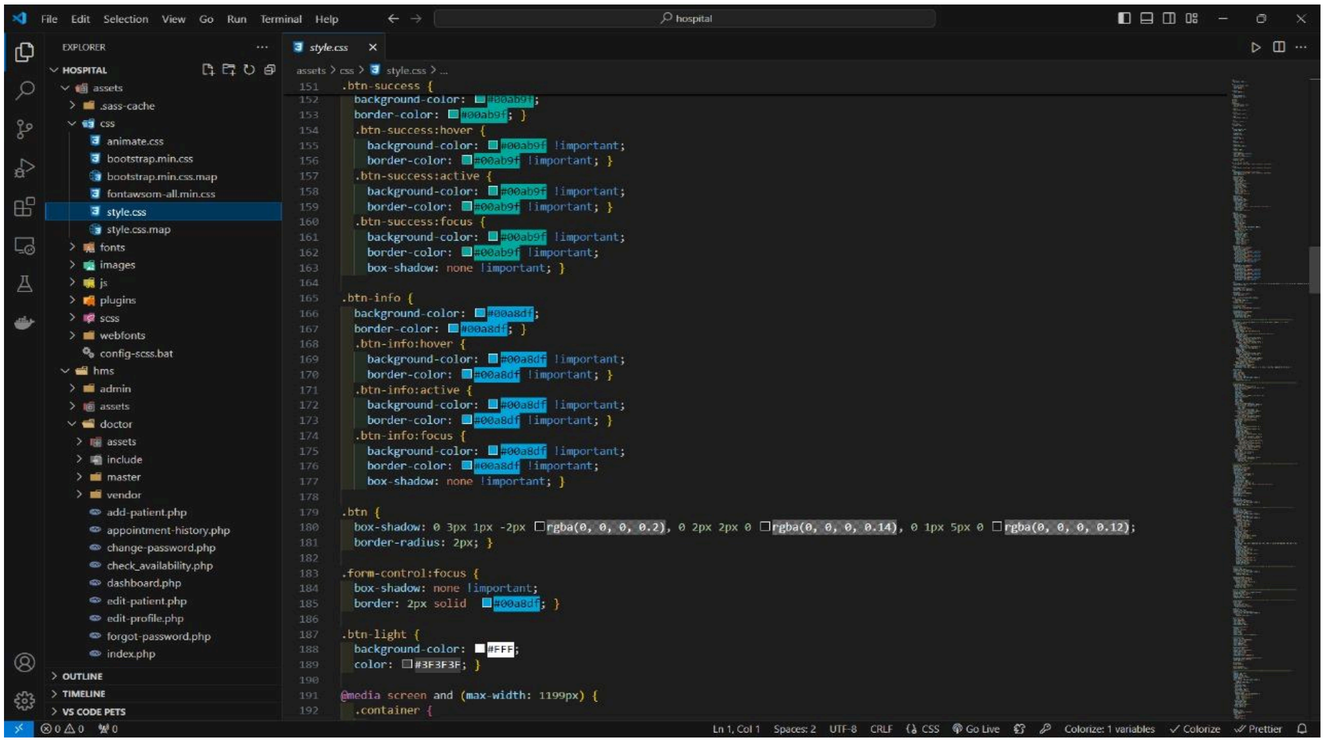
```
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
                <div class="main-content" >
                                    </li>
                                    <li class="active">
                                        <span>Book Appointment</span>
                                    </li>
                                </ol>
                            </section>
                            <!-- end: PAGE TITLE -->
                            <!-- start: BASIC EXAMPLE -->
                            <div class="container-fluid container-fullw bg-white">
                                <div class="row">
                                    <div class="col-md-12">

                                        <div class="row margin-top-30">
                                            <div class="col-lg-8 col-md-12">
                                                <div class="panel panel-white">
                                                    <div class="panel-heading">
                                                        <h5 class="panel-title">Book Appointment</h5>
                                                    </div>
                                                    <div class="panel-body">
                                    <p style="color:red;"><?php echo htmlentities(string: $_SESSION['msg1']);?>
                                    <?php echo htmlentities(string: $_SESSION['msg1']="");?></p>
                                                        <form role="form" name="book" method="post" >



<div class="form-group">
                                                              <label for="DoctorSpecialization">
                                                                  Doctor Specialization
                                                              </label>
                                <select name="Doctorspecialization" class="form-control" onChange="getdoctor(this.value);" required="requi
                                                                  <option value="">Select Specialization</option>
<?php $ret=mysqli_query(mysql: $con,query: "select * from doctorspecilization");
while($row=mysqli_fetch_array(result: $ret))
{
?>
                                                                  <option value="<?php echo htmlentities(string: $row['specilization']);
```

- **Manage med history**



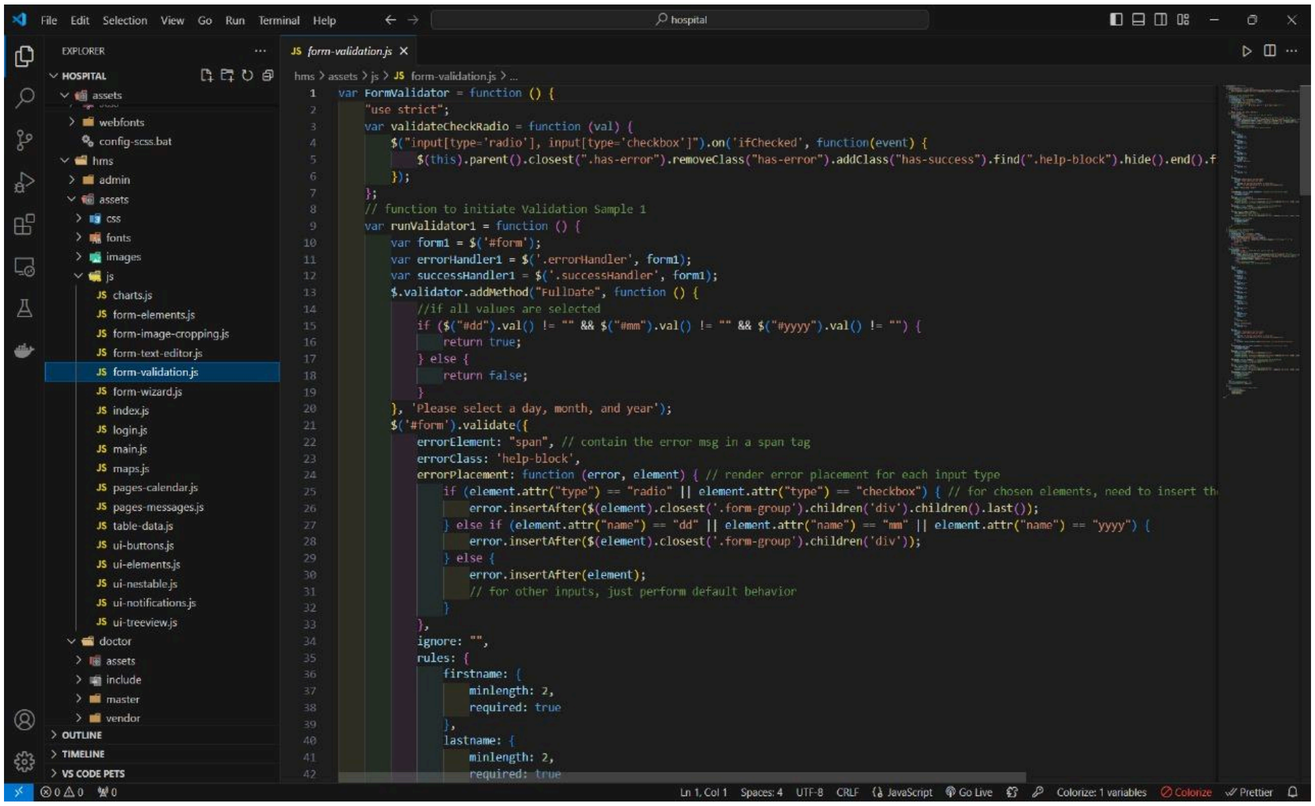
```
<html lang="en">
    <body>
        <div id="app">
<div class="app-content">
<div class="main-content" >
<th>Action</th>
</tr>
</thead>
<tbody>
<?php
$uid=$_SESSION['id'];
$sql=mysqli_query(mysql: $con,query: "select tblpatient.* from tblpatient join users on users.email=tblpatient.PatientEmail where user
$cnt=1;
while($row=mysqli_fetch_array(result: $sql))
{
?>
<tr>
<td class="center"><?php echo $cnt;?>.</td>
<td class="hidden-xs"><?php echo $row['PatientName'];?></td>
<td><?php echo $row['PatientContno'];?></td>
<td><?php echo $row['PatientGender'];?></td>
<td><?php echo $row['CreationDate'];?></td>
<td><?php echo $row['UpdationDate'];?>
</td>
<td>

<a href="view-medhistory.php?viewid=<?php echo $row['ID'];?>" class="btn btn-info btn-sm">View Details</a>

</td>
</tr>
<?php
$cnt=$cnt+1;
}?></tbody>
</table>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

- **CSS Files**



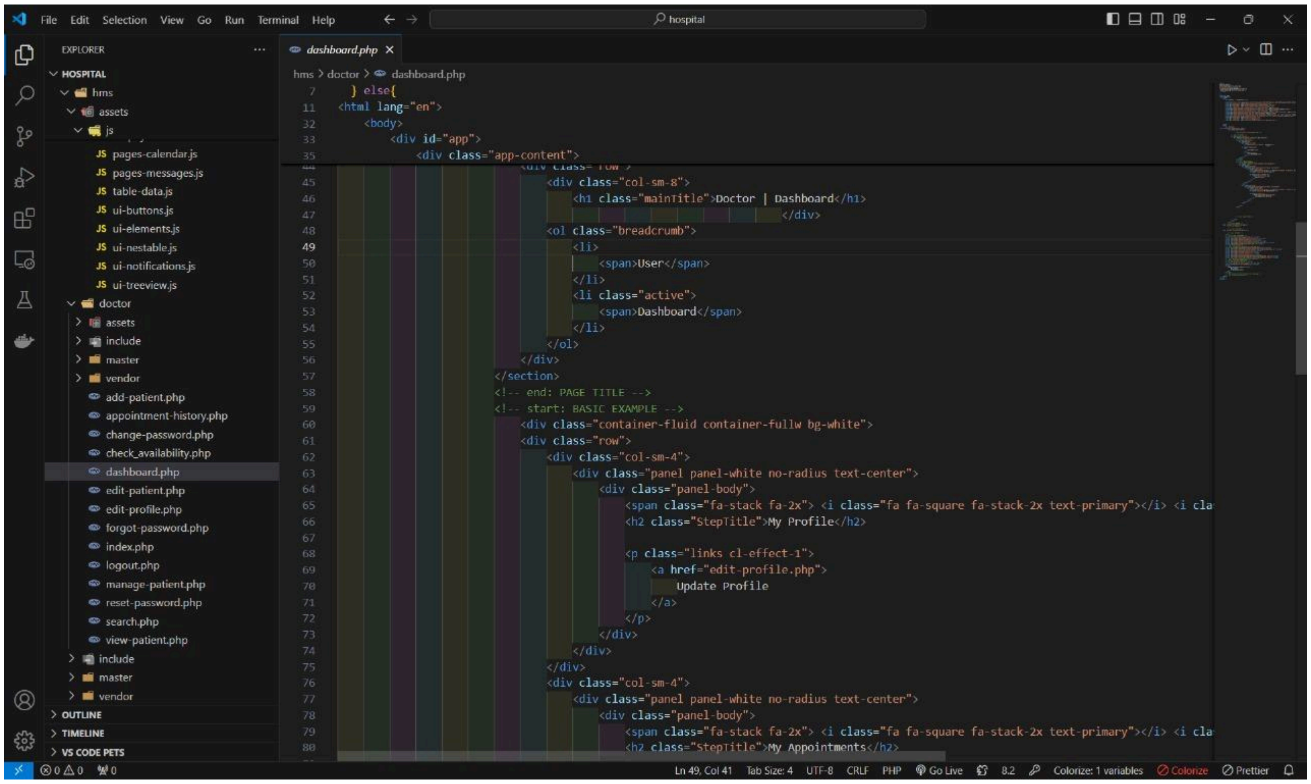
```css
.btn-success {
  background-color: #00ab9f;
  border-color: #00ab9f; }
  .btn-success:hover {
    background-color: #00ab9f !important;
    border-color: #00ab9f !important; }
  .btn-success:active {
    background-color: #00ab9f !important;
    border-color: #00ab9f !important; }
  .btn-success:focus {
    background-color: #00ab9f !important;
    border-color: #00ab9f !important;
    box-shadow: none !important; }

.btn-info {
  background-color: #00a8df;
  border-color: #00a8df; }
  .btn-info:hover {
    background-color: #00a8df !important;
    border-color: #00a8df !important; }
  .btn-info:active {
    background-color: #00a8df !important;
    border-color: #00a8df !important; }
  .btn-info:focus {
    background-color: #00a8df !important;
    border-color: #00a8df !important;
    box-shadow: none !important; }

.btn {
  box-shadow: 0 3px 1px -2px rgba(0, 0, 0, 0.2), 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 1px 5px 0 rgba(0, 0, 0, 0.12);
  border-radius: 2px; }

.form-control:focus {
  box-shadow: none !important;
  border: 2px solid  #00a8df; }

.btn-light {
  background-color: #FFF;
  color: #3F3F3F; }

@media screen and (max-width: 1199px) {
  .container {
```

- **Form Validator**



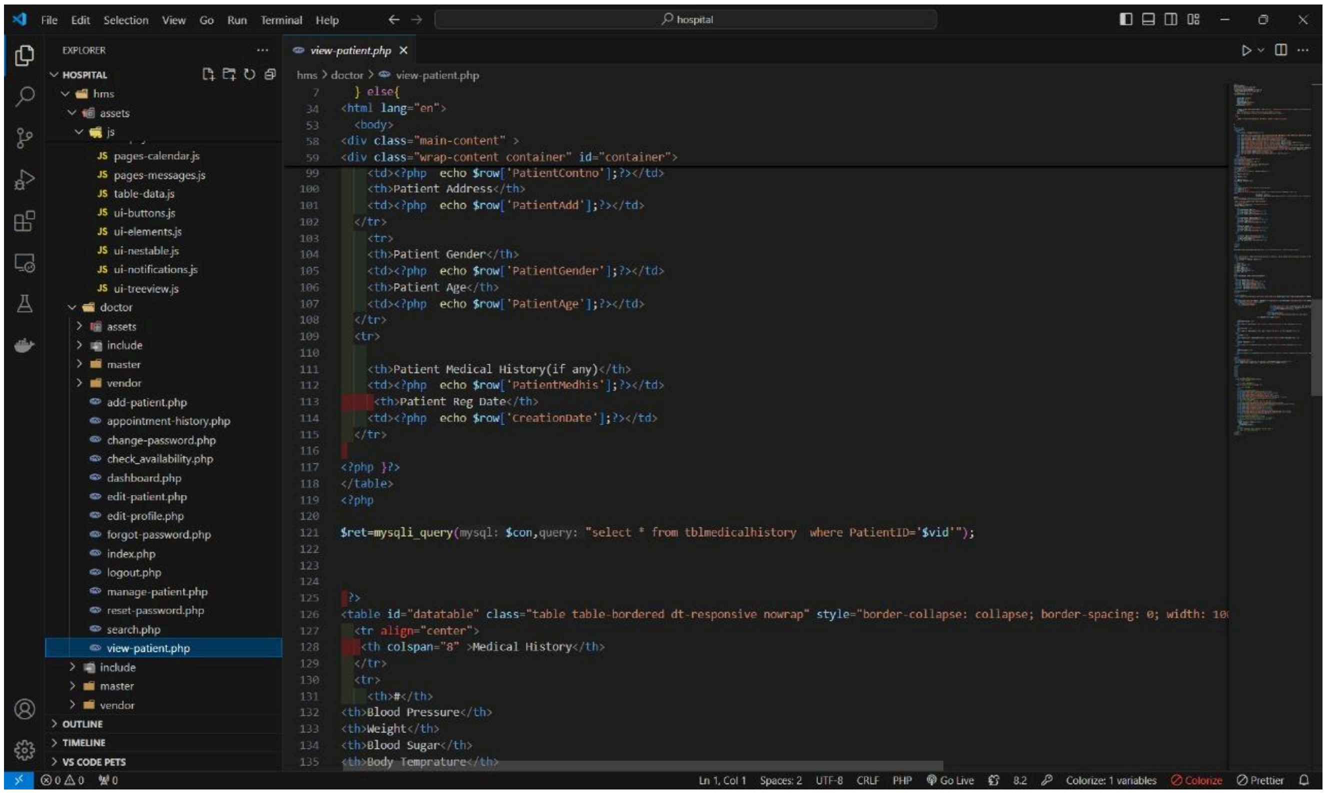
```javascript
var FormValidator = function () {
    "use strict";
    var validateCheckRadio = function (val) {
        $("input[type='radio'], input[type='checkbox']").on('ifChecked', function(event) {
            $(this).parent().closest(".has-error").removeClass("has-error").addClass("has-success").find(".help-block").hide().end().f
        });
    };
    // function to initiate Validation Sample 1
    var runValidator1 = function () {
        var form1 = $('#form');
        var errorHandler1 = $('.errorHandler', form1);
        var successHandler1 = $('.successHandler', form1);
        $.validator.addMethod("FullDate", function () {
            //if all values are selected
            if ($("#dd").val() != "" && $("#mm").val() != "" && $("#yyyy").val() != "") {
                return true;
            } else {
                return false;
            }
        }, 'Please select a day, month, and year');
        $('#form').validate({
            errorElement: "span", // contain the error msg in a span tag
            errorClass: 'help-block',
            errorPlacement: function (error, element) { // render error placement for each input type
                if (element.attr("type") == "radio" || element.attr("type") == "checkbox") { // for chosen elements, need to insert th
                    error.insertAfter($(element).closest('.form-group').children('div').children().last());
                } else if (element.attr("name") == "dd" || element.attr("name") == "mm" || element.attr("name") == "yyyy") {
                    error.insertAfter($(element).closest('.form-group').children('div'));
                } else {
                    error.insertAfter(element);
                    // for other inputs, just perform default behavior
                }
            },
            ignore: "",
            rules: {
                firstname: {
                    minlength: 2,
                    required: true
                },
                lastname: {
                    minlength: 2,
                    required: true
```

- **Doc Dashboard**



```
    } else{
<html lang="en">
    <body>
        <div id="app">
            <div class="app-content">
                    <div class="col-sm-8">
                        <h1 class="mainTitle">Doctor | Dashboard</h1>
                                    </div>
                    <ol class="breadcrumb">
                        <li>
                            <span>User</span>
                        </li>
                        <li class="active">
                            <span>Dashboard</span>
                        </li>
                    </ol>
                </div>
            </section>
            <!-- end: PAGE TITLE -->
            <!-- start: BASIC EXAMPLE -->
                <div class="container-fluid container-fullw bg-white">
                <div class="row">
                    <div class="col-sm-4">
                        <div class="panel panel-white no-radius text-center">
                            <div class="panel-body">
                                <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i cla
                                <h2 class="StepTitle">My Profile</h2>

                                <p class="links cl-effect-1">
                                    <a href="edit-profile.php">
                                        Update Profile
                                    </a>
                                </p>
                            </div>
                        </div>
                    </div>
                    <div class="col-sm-4">
                        <div class="panel panel-white no-radius text-center">
                            <div class="panel-body">
                                <span class="fa-stack fa-2x"> <i class="fa fa-square fa-stack-2x text-primary"></i> <i cla
                                <h2 class="StepTitle">My Appointments</h2>
```

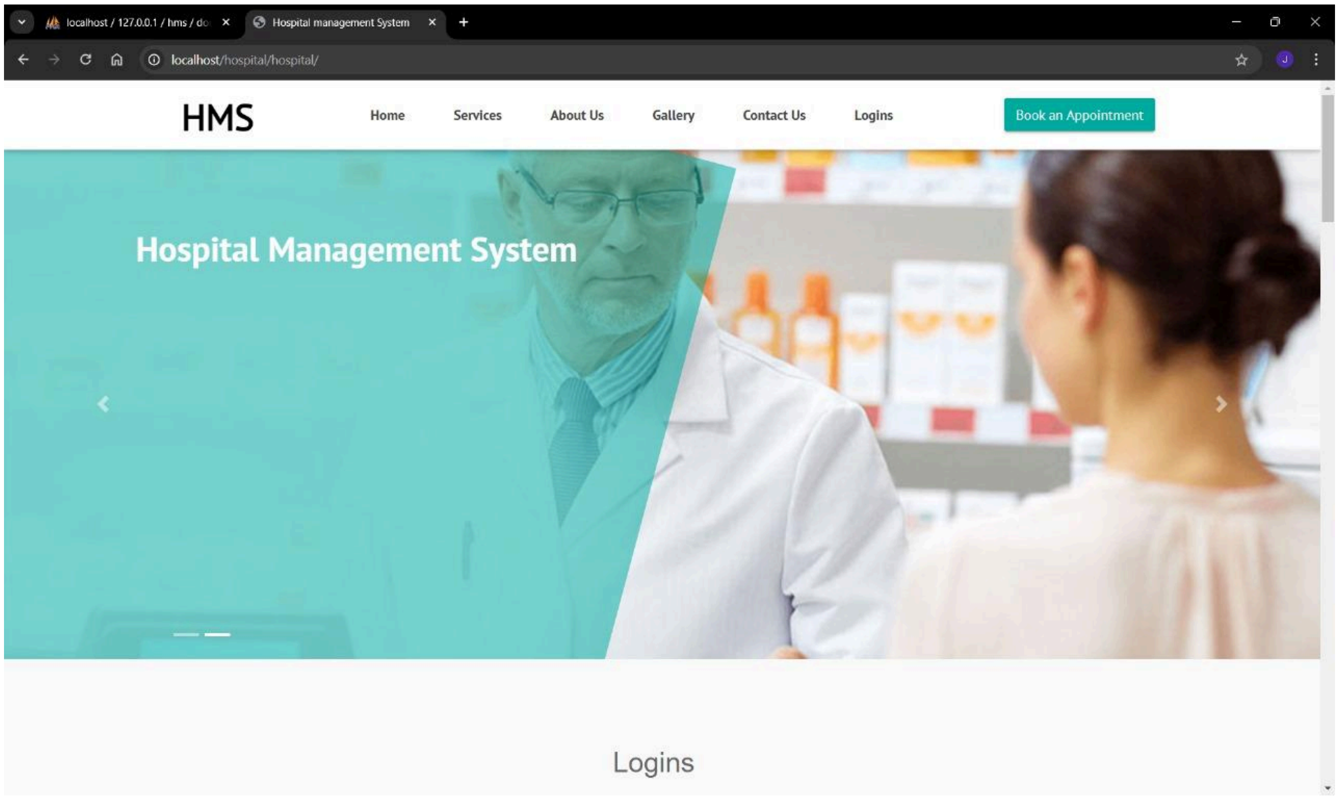
- **View Patients**



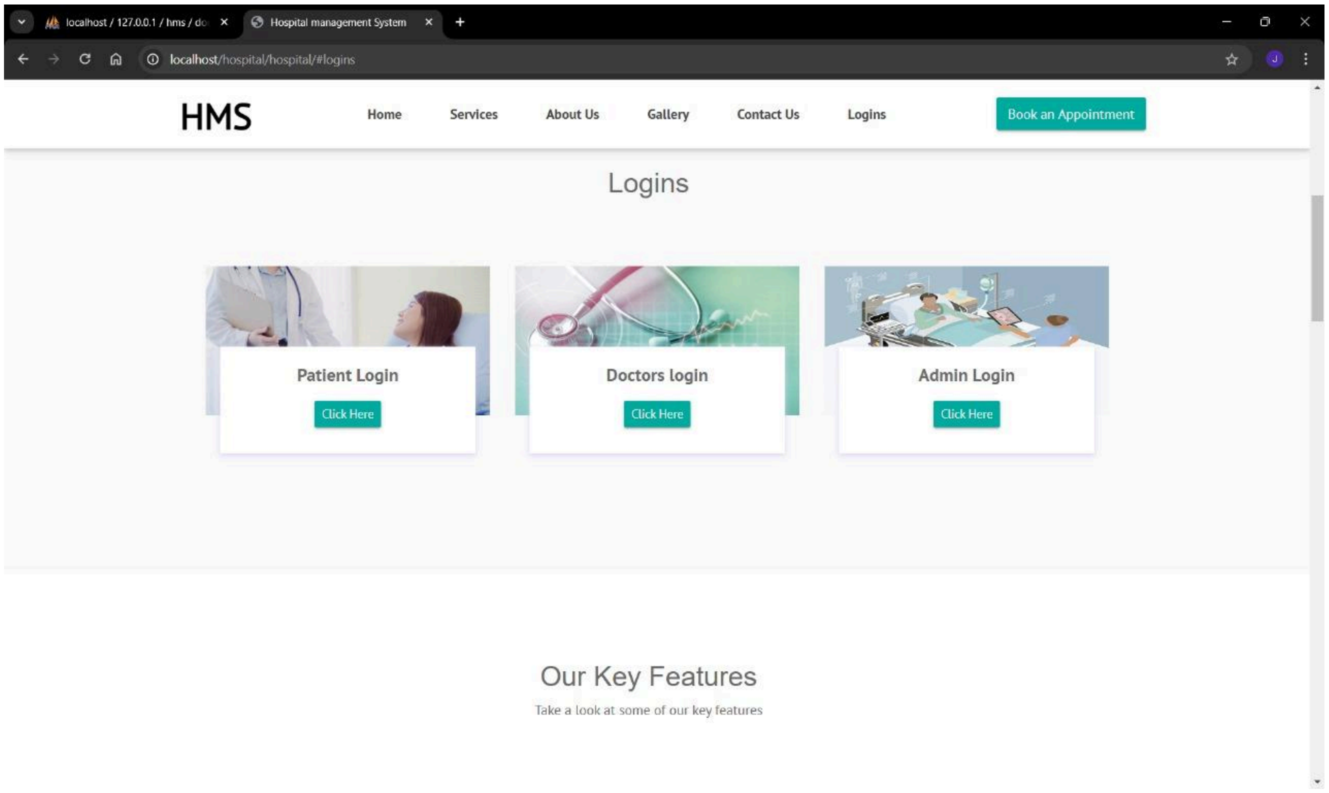
```
    } else{
<html lang="en">
  <body>
<div class="main-content" >
<div class="wrap-content container" id="container">
    <td><?php  echo $row['PatientContno'];?></td>
    <th>Patient Address</th>
    <td><?php  echo $row['PatientAdd'];?></td>
  </tr>
    <tr>
    <th>Patient Gender</th>
    <td><?php  echo $row['PatientGender'];?></td>
    <th>Patient Age</th>
    <td><?php  echo $row['PatientAge'];?></td>
  </tr>
  <tr>

    <th>Patient Medical History(if any)</th>
    <td><?php  echo $row['PatientMedhis'];?></td>
     <th>Patient Reg Date</th>
    <td><?php  echo $row['CreationDate'];?></td>
  </tr>

<?php }?>
</table>
<?php

$ret=mysqli_query(mysql: $con,query: "select * from tblmedicalhistory  where PatientID='$vid'");



?>
<table id="datatable" class="table table-bordered dt-responsive nowrap" style="border-collapse: collapse; border-spacing: 0; width: 10
  <tr align="center">
   <th colspan="8" >Medical History</th>
  </tr>
  <tr>
    <th>#</th>
<th>Blood Pressure</th>
<th>Weight</th>
<th>Blood Sugar</th>
<th>Body Temprature</th>
```

APPENDIX-2 :OUTPUT

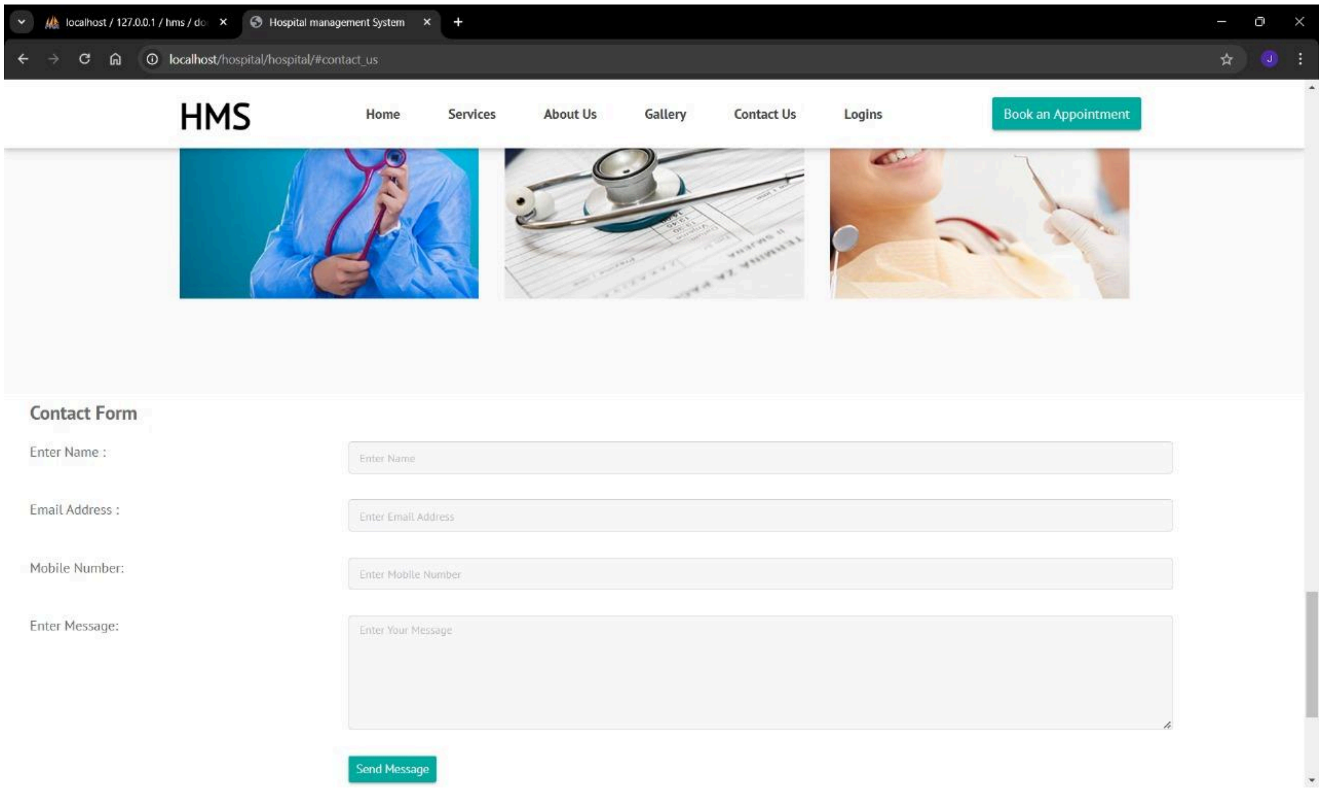
- **Home Page**


localhost / 127.0.0.1 / hms / do
Hospital management System
localhost/hospital/hospital/
HMS
Home
Services
About Us
Gallery
Contact Us
Logins
Book an Appointment
Hospital Management System
Logins

- **Login Section**



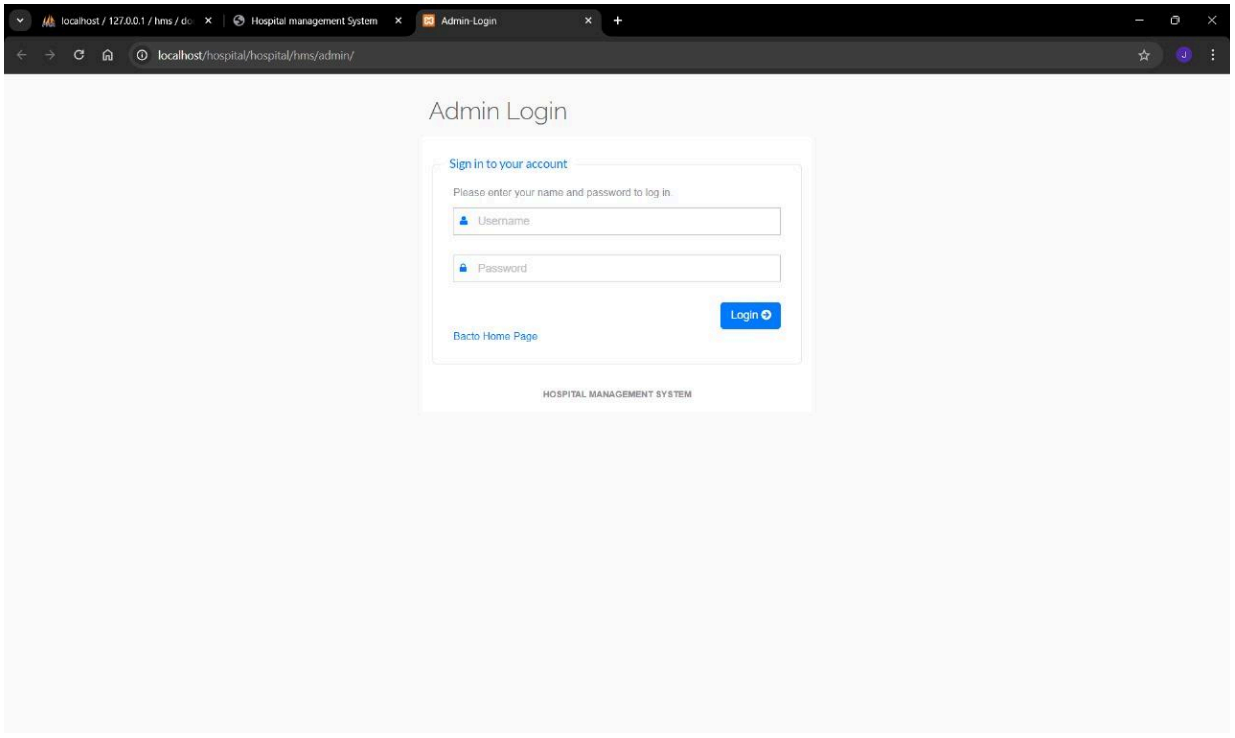
- **Contact Us**


localhost / 127.0.0.1 / hms / do
Hospital management System
localhost/hospital/hospital/#contact_us
HMS
Home
Services
About Us
Gallery
Contact Us
Logins
Book an Appointment
Send Message

Contact Form

Enter Name :	Enter Name
Email Address :	Enter Email Address
Mobile Number:	Enter Mobile Number
Enter Message:	Enter Your Message

- **Admin Login**


localhost / 127.0.0.1 / hms / do
Hospital management System
Admin-Login
localhost/hospital/hospital/hms/admin/
Login
Bacto Home Page

Admin Login

Sign in to your account

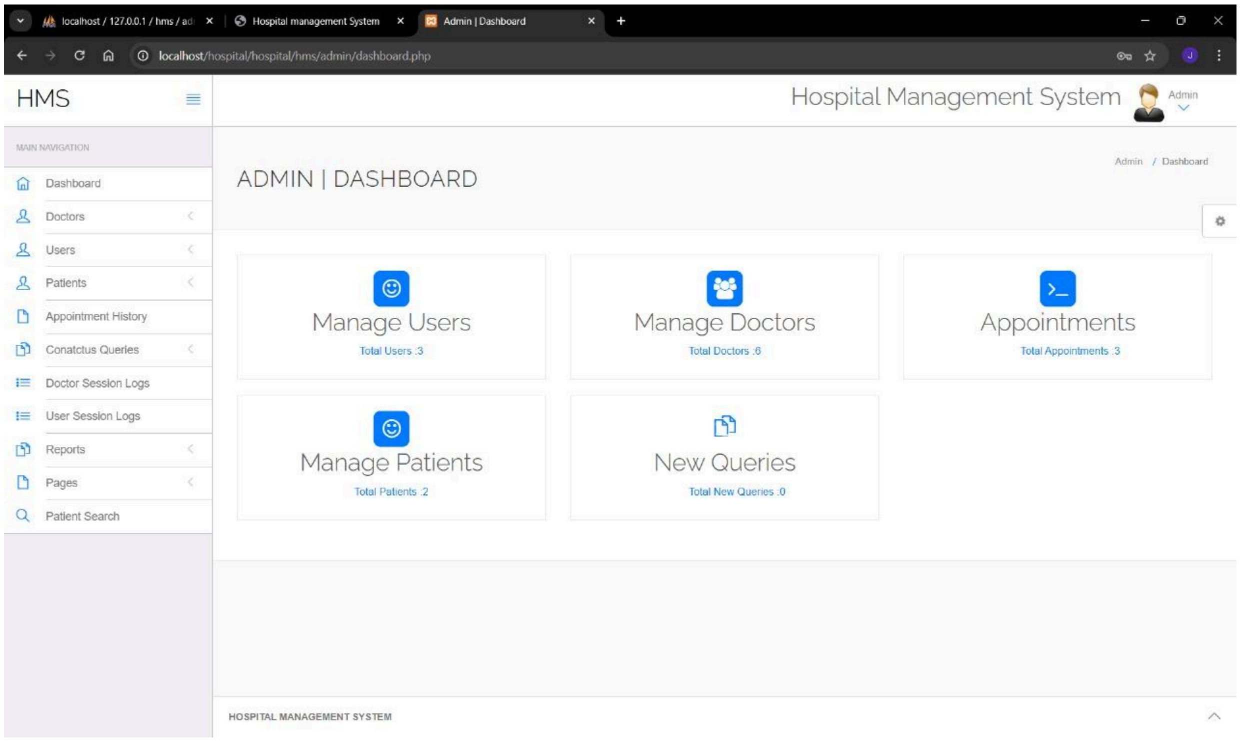
Please enter your name and password to log in.

Username

Password

HOSPITAL MANAGEMENT SYSTEM

- **Admin Dashboard**



ADMIN | DASHBOARD

Manage Users
Total Users :3

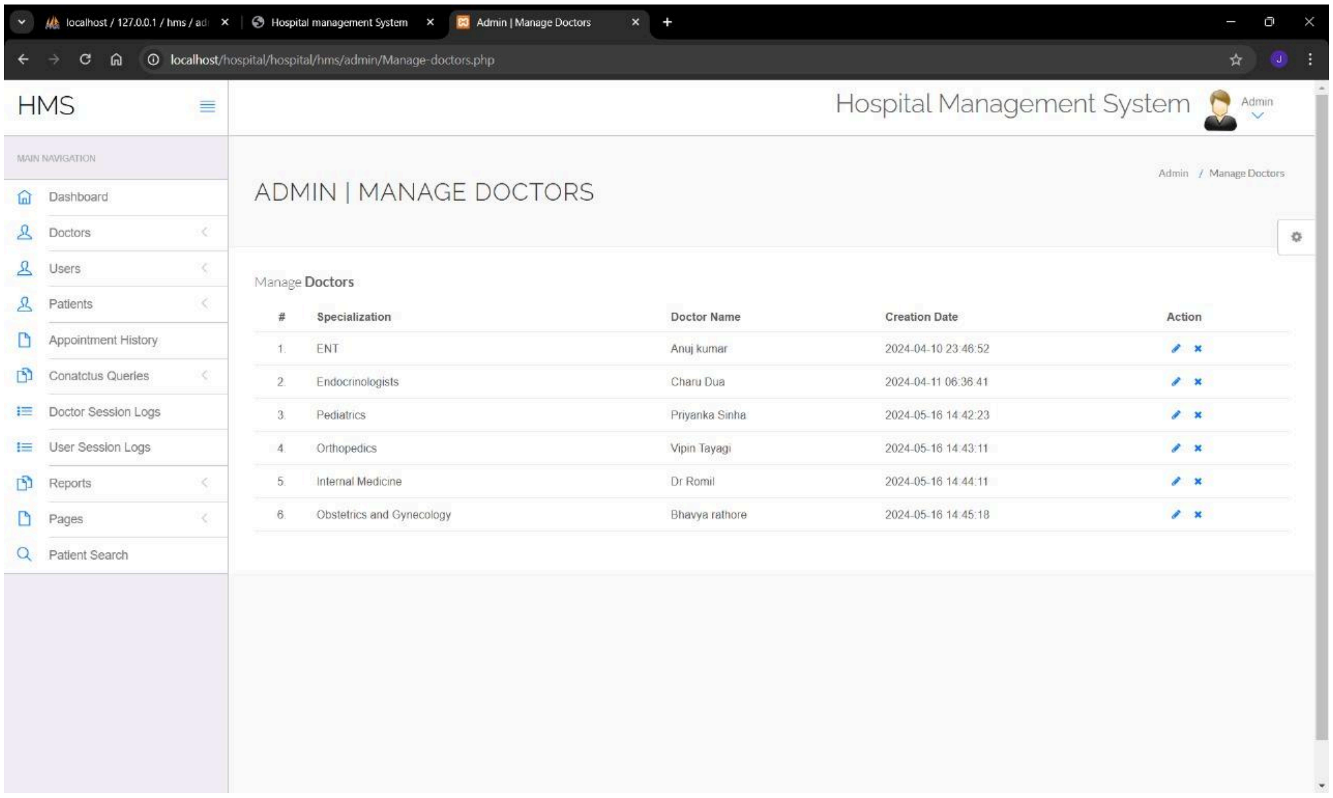
Manage Doctors
Total Doctors :6

Appointments
Total Appointments :3

Manage Patients
Total Patients :2

New Queries
Total New Queries :0

- **Admin Manage Doc**

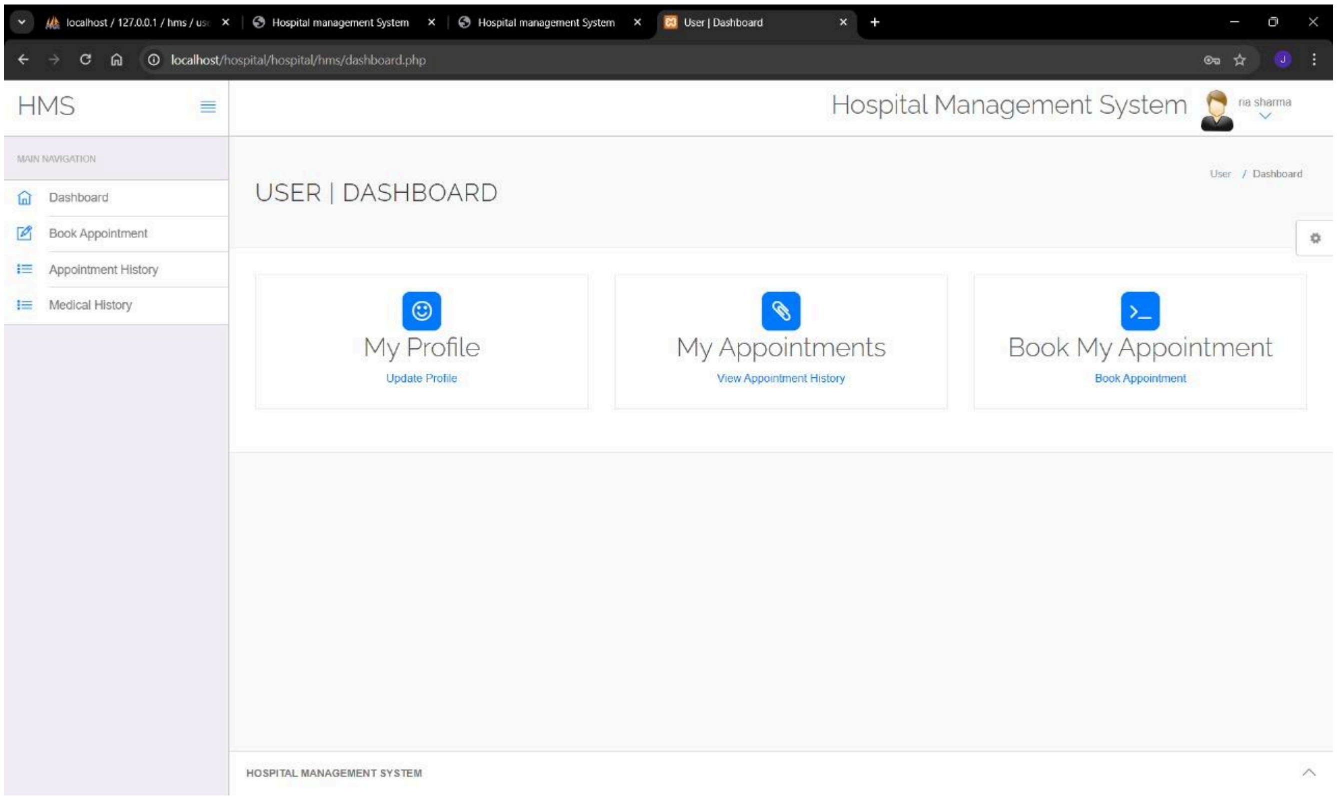


ADMIN | MANAGE DOCTORS

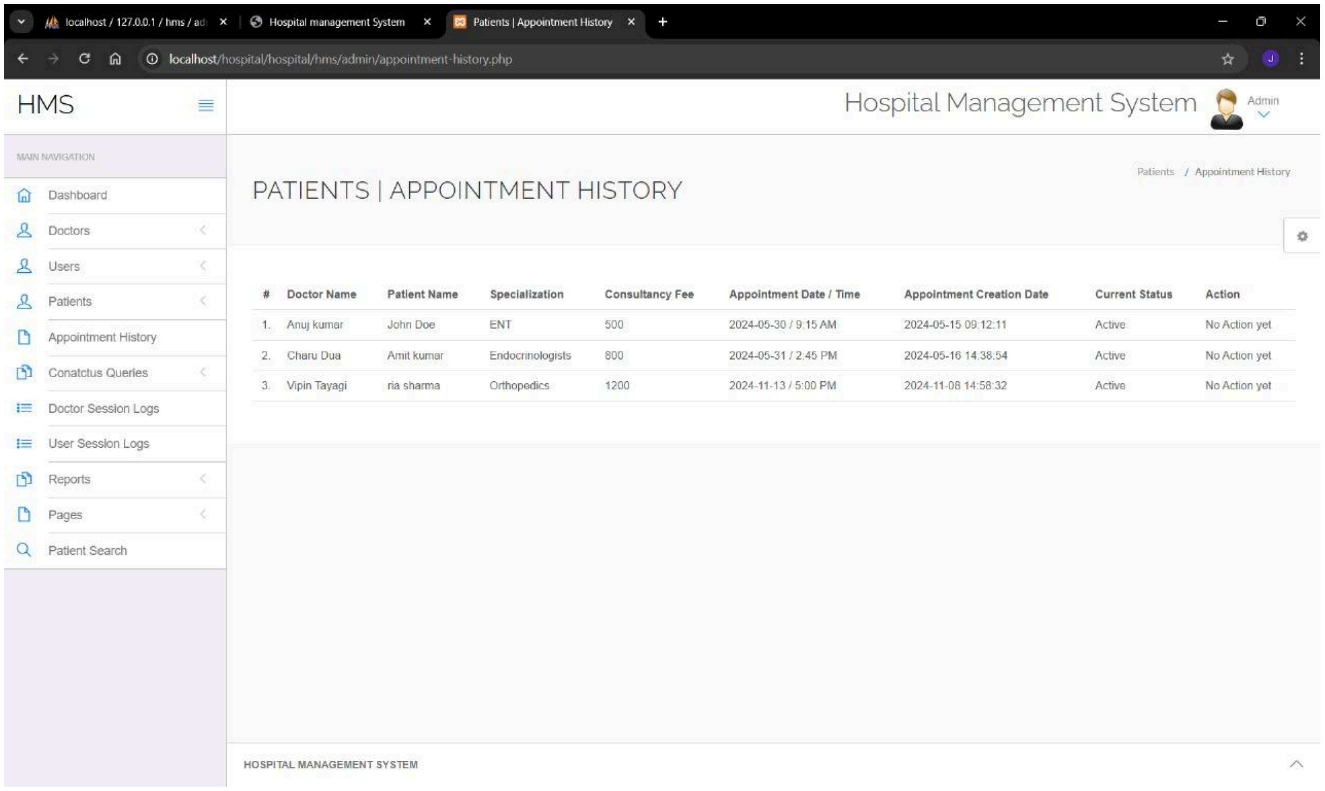
Manage **Doctors**

#	Specialization	Doctor Name	Creation Date	Action
1.	ENT	Anuj kumar	2024-04-10 23:46:52	
2.	Endocrinologists	Charu Dua	2024-04-11 06:36:41	
3.	Pediatrics	Priyanka Sinha	2024-05-16 14:42:23	
4.	Orthopedics	Vipin Tayagi	2024-05-16 14:43:11	
5.	Internal Medicine	Dr Romil	2024-05-16 14:44:11	
6.	Obstetrics and Gynecology	Bhavya rathore	2024-05-16 14:45:18	

- **User Dashboard**

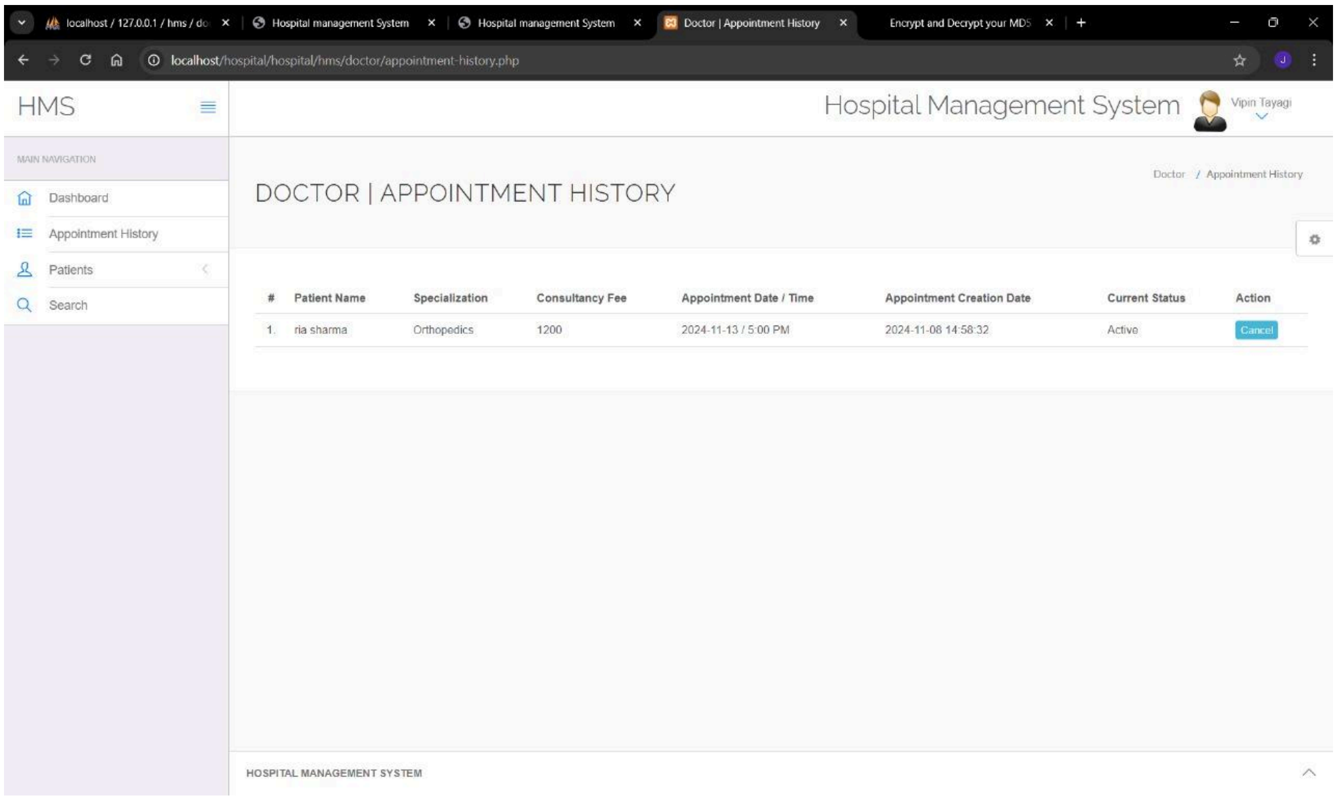


- **Patient History**


localhost / 127.0.0.1 / hms / ad
Hospital management System
Patients | Appointment History
localhost/hospital/hospital/hms/admin/appointment-history.php
HMS
Hospital Management System
Admin
MAIN NAVIGATION
Dashboard
Doctors
Users
Patients
Appointment History
Conatctus Queries
Doctor Session Logs
User Session Logs
Reports
Pages
Patient Search
PATIENTS | APPOINTMENT HISTORY
Patients / Appointment History
HOSPITAL MANAGEMENT SYSTEM

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Anuj kumar	John Doe	ENT	500	2024-05-30 / 9:15 AM	2024-05-15 09:12:11	Active	No Action yet
2.	Charu Dua	Amit kumar	Endocrinologists	800	2024-05-31 / 2:45 PM	2024-05-16 14:38:54	Active	No Action yet
3.	Vipin Tayagi	ria sharma	Orthopedics	1200	2024-11-13 / 5:00 PM	2024-11-08 14:58:32	Active	No Action yet

- **Appointment History**

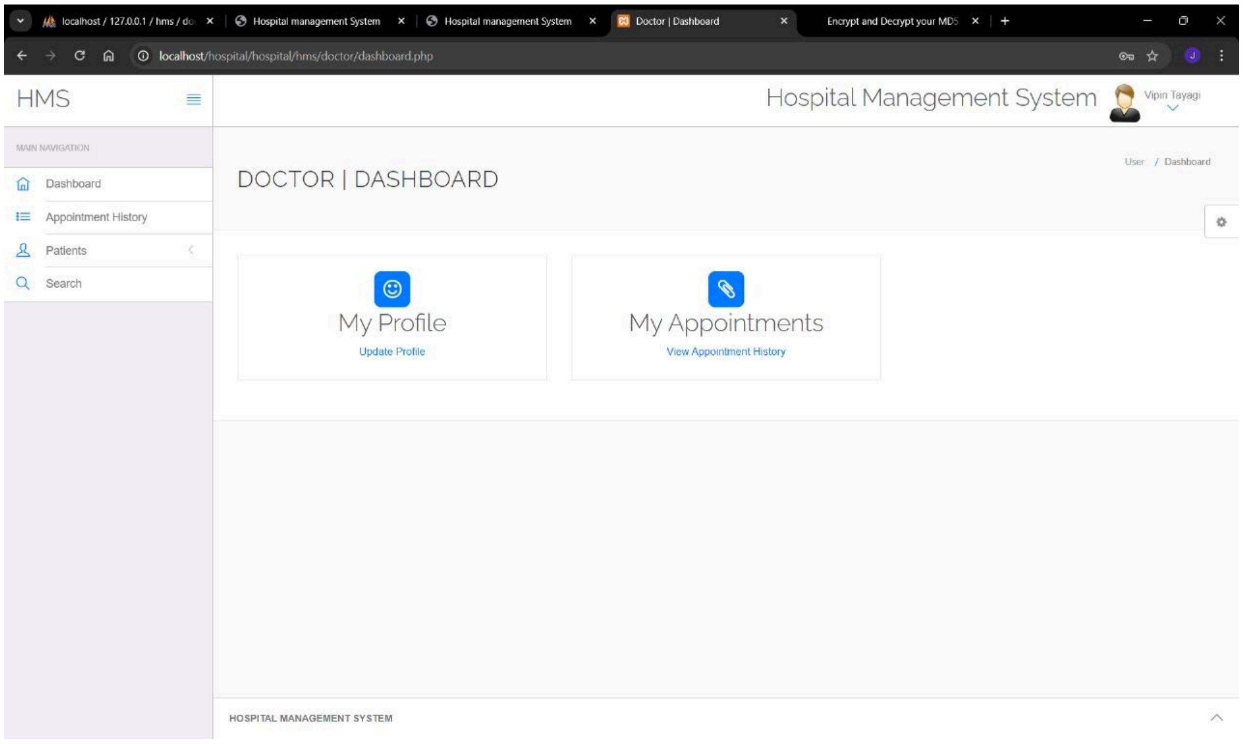


localhost / 127.0.0.1 / hms / do
Hospital management System
Hospital management System
Doctor | Appointment History
Encrypt and Decrypt your MD5
localhost/hospital/hospital/hms/doctor/appointment-history.php
HMS
Hospital Management System
Vipin Tayagi
MAIN NAVIGATION
Dashboard
Appointment History
Patients
Search
Doctor / Appointment History
HOSPITAL MANAGEMENT SYSTEM

DOCTOR | APPOINTMENT HISTORY

#	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	ria sharma	Orthopedics	1200	2024-11-13 / 5:00 PM	2024-11-08 14:58:32	Active	Cancel

- **Doctor Dashboard**



localhost / 127.0.0.1 / hms / do
Hospital management System
Hospital management System
Doctor | Dashboard
Encrypt and Decrypt your MD5
localhost/hospital/hospital/hms/doctor/dashboard.php
HMS
Hospital Management System
Vipin Tayagi
MAIN NAVIGATION
Dashboard
Appointment History
Patients
Search
User / Dashboard
DOCTOR | DASHBOARD
My Profile
Update Profile
My Appointments
View Appointment History
HOSPITAL MANAGEMENT SYSTEM