HOSPITAL APPOINTMENT BOOKING SYSTEM (HABS)

A MINI-PROJECT REPORT

Submitted by

G MADHUMITHA (221801030)

S MADHUVANTHIY (221801031)

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

ANNA UNIVERSITY,CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled "**HOSPITAL APPOINTMENT BOOKING SYSTEM (HABS)**" is the Bonafide work of **MADHUMITHA G (221801030), MADHUVANTHIY S (221801031)** who carried out the work under my supervision.

SIGNATURE

Dr. J.M. Gnanasekar M.E., Ph.D.,
Professor and Head
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

SIGNATURE

Mr. S. Suresh Kumar M.E., Ph.D.,
Professor,
Department of Artificial Intelligence and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Submitted for the project viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** and our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr.. J.M. GNANASEKAR., M.E., Ph.D.,** Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our coordinator **Mr. S. SURESH KUMAR M.E., Ph.D., Professor**, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

The Doctor Appointment System is a comprehensive web-based platform developed to modernize and streamline the traditional process of booking medical consultations in hospitals and clinics. The system addresses key issues such as long patient wait times, manual scheduling inefficiencies, and limited doctor-patient interaction by offering an automated solution that facilitates real-time appointment booking, doctor availability tracking, and user profile management. It features role-based access for patients, doctors, and administrators, enabling each group to interact with the system based on their specific needs—patients can search for doctors based on specialization and location, book or reschedule appointments, and maintain appointment history; doctors can manage their availability and patient list; and administrators can monitor platform activity and user data. Developed using web technologies like HTML, CSS, JavaScript, PHP, and MySQL, the system prioritizes data security, user-friendliness, and scalability. It not only reduces administrative overhead and booking errors but also enhances healthcare accessibility, especially for remote users. Designed with modular architecture, the platform is scalable and capable of supporting future enhancements such as mobile applications, online payments, teleconsultations, and multilingual interfaces. The system ultimately contributes to a more efficient, transparent, and accessible healthcare environment by leveraging digital innovation.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In the modern era, healthcare systems are undergoing a rapid digital transformation. This evolution is being driven by the growing need to improve efficiency, reduce human error, and enhance patient satisfaction. Traditional methods of hospital management and appointment scheduling are no longer sufficient in handling the increasing demands of both patients and healthcare providers. As a result, the integration of technology into healthcare infrastructure has become essential. From managing medical records to handling patient appointments and staff coordination, digital platforms are providing robust and scalable solutions that improve accuracy and transparency.

One of the critical areas in need of digital innovation is the appointment scheduling process. In many hospitals and clinics, appointments are still handled manually, leading to issues such as double bookings, lost records, and long waiting times. These inefficiencies not only affect patient care but also place a burden on administrative staff and doctors. A well-designed appointment management system eliminates these issues by allowing patients to schedule visits online, receive automatic confirmations, and reschedule with ease. For doctors, it provides a clear view of their daily schedule and allows better time management. This level of automation minimizes administrative overhead and ensures a smooth workflow for everyone involved.

The **Doctor Appointment and Hospital Management System** is designed to address these challenges by creating a centralized digital platform where all key activities are managed efficiently. The system caters to multiple stakeholders — patients, doctors, and administrators — with distinct roles and access privileges. Patients can register and book appointments based on doctor availability, doctors can update their schedules and manage appointments, and admins can monitor system usage, manage records, and generate reports.

1.2 NEED FOR THE STUDY

In today's fast-paced world, healthcare systems are under immense pressure to deliver timely and efficient services to patients. Traditional hospital management practices, especially appointment scheduling and patient record management, often lead to delays, miscommunication, and administrative confusion. Patients are frequently subjected to long queues, limited appointment slots, and inefficient communication with hospital staff. This not only affects their satisfaction but also compromises the quality of healthcare delivery. In this context, a dedicated digital solution is no longer optional—it is essential.

The lack of a centralized system in hospitals leads to data fragmentation, making it difficult for healthcare providers to access complete patient histories and appointment logs. Additionally, manually handled processes are more susceptible to human error, such as misplaced files, incorrect bookings, or scheduling overlaps. These shortcomings hinder the optimal use of hospital resources and contribute to patient dissatisfaction. The increasing complexity and patient load demand a system that can automate routine tasks and allow medical professionals to focus more on clinical care rather than paperwork.

A Doctor Appointment and Hospital Management System addresses these gaps by digitizing the entire workflow—right from user registration and appointment booking to doctor scheduling and patient history tracking. By streamlining operations and centralizing data, the system enhances transparency, minimizes the scope for errors, and ensures real-time communication between all parties involved. Thus, the need for this study stems from the urgency to improve healthcare infrastructure with a scalable and reliable digital system that meets both current and future demands.

1.3OBJECTIVES OF THE STUDY

1. **Develop a Comprehensive Web-Based System:** Create an integrated platform that simplifies hospital administration and enables efficient scheduling of appointments between patients and healthcare professionals.
2. **Enhance Patient Empowerment:** Provide a user-friendly interface that allows patients to self-manage their appointments, reducing dependency on front-desk staff and minimizing waiting times.
3. **Improve Accessibility:** Facilitate appointment scheduling for individuals with mobility challenges or those residing in remote areas, ensuring equitable access to healthcare services.
4. **Optimize Doctor and Administrator Workflows:** Equip doctors and hospital administrators with tools to manage schedules efficiently, including viewing, accepting, or rescheduling appointments through an intuitive dashboard.
5. **Automate Administrative Tasks:** Reduce administrative burdens by automating appointment confirmations, reminders, and status updates via email or SMS notifications.
6. **Ensure Data Security and Privacy:** Implement proper authentication and access controls to safeguard sensitive medical data, aligning with healthcare regulations and standards.
7. **Support Data-Driven Decision Making:** Provide analytical reports and insights to assist administrators in making informed decisions regarding hospital operations and resource allocation.
8. **Enhance Overall Healthcare Service Quality:** Improve patient access, optimize doctor schedules, and enhance administrative control to ensure the overall quality and efficiency of healthcare service delivery.

1.4 OVERVIEW OF THE PROJECT

The Doctor Appointment and Hospital Management System is designed as a comprehensive web-based platform that streamlines the operations of healthcare facilities, particularly focusing on the appointment scheduling process. The system provides three core user roles: patients, doctors, and administrators, each having dedicated modules with specific functionalities. Patients can create accounts, search for doctors by department or specialty, view availability, and book appointments. Upon booking, they receive confirmation and can track the status of their appointments through the system.

Doctors have a personalized dashboard where they can view upcoming appointments, accept or reject requests, and update their availability. This functionality helps doctors plan their schedules more efficiently, reduce overlapping appointments, and ensure they are not overbooked. Additionally, doctors can view brief patient details prior to the appointment, enabling better preparation and patient care. The administrative panel allows hospital staff or system administrators to manage doctor profiles, monitor system activity, oversee patient records, and generate useful reports on bookings, user traffic, and doctor performance.

The system architecture is built using standard web technologies like HTML, CSS, JavaScript for the frontend, PHP for server-side scripting, and MySQL for the backend database. It is designed with a responsive interface to ensure accessibility across devices—be it a desktop, tablet, or smartphone. The platform integrates features like secure login, automated email notifications, search filters, and real-time updates. Overall, this project serves as a scalable, secure, and efficient solution to digitalize and optimize hospital workflows, thus improving healthcare service delivery and operational transparency.

CHAPTER 2
REVIEW OF LITERATURE

2.1 INTRODUCTION

The healthcare industry has witnessed a significant shift in recent years with the adoption of Information and Communication Technologies (ICT) to manage services more effectively. Several studies and past projects have shown that digitized hospital systems lead to better patient outcomes, reduced waiting times, and streamlined administrative work. Research on electronic medical records, patient portals, and appointment scheduling systems underscores the importance of automation in reducing human error, increasing transparency, and improving resource utilization in healthcare settings.

Multiple healthcare organizations worldwide have already transitioned to digital appointment booking platforms. Literature from developed nations indicates a substantial increase in patient satisfaction due to reduced queuing and quicker access to specialized care. Moreover, these systems also alleviate pressure on healthcare staff by reducing manual tasks such as data entry, paper filing, and phone-based bookings. The benefits observed in such case studies validate the need for similar platforms in developing or under-resourced regions.

Additionally, previous academic work and commercial software solutions highlight the challenges faced during implementation, such as resistance to change, data security concerns, and integration with existing systems. These insights inform the design and development of this project by identifying common pitfalls and successful strategies. Incorporating lessons from earlier literature ensures that the Doctor Appointment and Hospital Management System is built with practicality, user experience, and sustainability in mind.

2.2 FRAMEWORK OF LCA

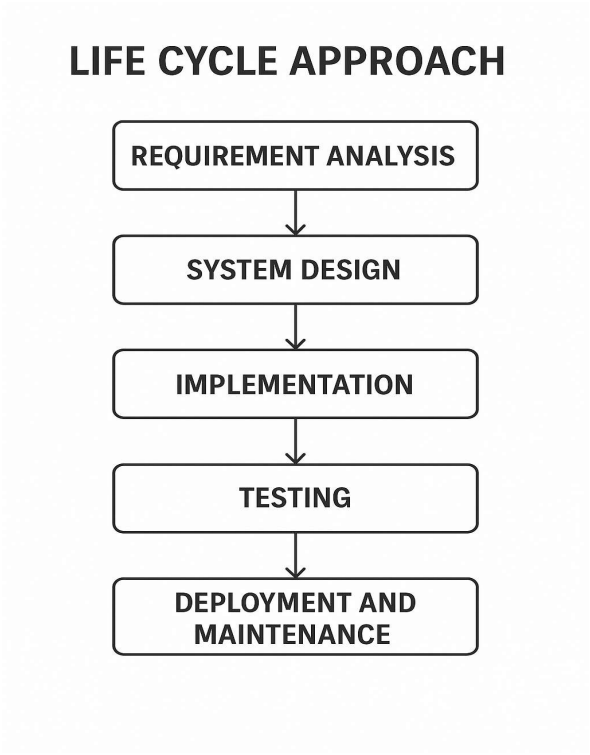

Fig 2.2.1: LCA

The Life Cycle Approach (LCA) is a systematic methodology used in software engineering to guide the development of projects through structured phases. It ensures that every aspect of the system—from initial requirements to final deployment—is addressed methodically. In the context of the Doctor Appointment and Hospital Management System, LCA helps to maintain clarity, reduce risks, and enable continuous improvement throughout the software's development and usage.

The framework begins with the Requirement Analysis Phase, where user expectations are gathered and documented. This includes understanding the needs of patients, doctors, and hospital staff, and identifying the functionalities they expect from the system. Following this, the System Design Phase involves planning the architecture of the application—defining databases, user interfaces, and process flows to meet those requirements effectively.

After the design is finalized, the project enters the Implementation Phase, where actual coding takes place using chosen technologies like PHP and MySQL. This is followed by the Testing Phase, where the system is rigorously tested for bugs, usability, and performance issues. Finally, the Deployment and Maintenance Phase involves installing the system in a real-world environment, training users, and continuously updating it based on feedback and evolving requirements. The structured nature of LCA ensures the developed solution is both efficient and scalable, suitable for long-term use in healthcare facilities.

Additionally, the Life Cycle Approach (LCA) emphasizes the importance of feedback loops and iterative improvements. Each phase of the LCA is not strictly linear; instead, the process is designed to allow for revisiting earlier stages based on insights gained during later phases. For instance, if new requirements emerge during the Testing Phase, the system design might be revisited to accommodate those needs. This iterative nature of the LCA ensures that the Doctor Appointment and Hospital Management System can evolve in response to user feedback and changing technological or regulatory demands, resulting in a more adaptable and future-proof solution. By continuously improving the system, LCA helps maintain its relevance and effectiveness over time, ensuring that the hospital's operational needs are consistently met and patient care remains efficient.

CHAPTER 3
SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

In many hospitals and clinics, the current system for managing appointments and administrative records is largely manual. Patients often have to physically visit the hospital or call the front desk to schedule appointments. This process is not only time-consuming but also prone to human errors, such as double-booking or lost appointment records. Staff members spend a considerable amount of time updating logs, managing schedules, and coordinating between departments. In peak hours, the inefficiencies are magnified, leading to long waiting lines and dissatisfied patients.

In addition, the lack of a centralized system means that patient data is often scattered across different departments, making it difficult for doctors to access comprehensive information during consultations. This fragmentation not only delays treatment but can also compromise the quality of care provided. Moreover, appointment cancellations or rescheduling often fail to reflect immediately, causing confusion and scheduling conflicts.

Manual systems also suffer from poor data security. Physical records are vulnerable to damage, loss, or unauthorized access. Without audit trails and proper user authentication, it becomes challenging to monitor who accessed or modified sensitive data. As healthcare becomes increasingly digital, the shortcomings of the traditional system highlight the urgent need for a more robust, centralized, and automated solution.

3.2 PROPOSED SYSTEM

The proposed Doctor Appointment and Hospital Management System addresses the limitations of the existing manual system by offering a web-based platform that automates the end-to-end workflow. Patients can register, log in, search for doctors based on specialty and availability, and book appointments—all from the comfort of their homes. The system instantly confirms bookings and sends notifications via email or SMS. Patients also have the option to reschedule or cancel their appointments through the interface, which automatically updates the backend.

Doctors can view their schedules in real time through a dedicated dashboard. They can accept, decline, or reschedule appointments with a few clicks. The system allows them to access patient records prior to the appointment, enhancing their preparedness and improving diagnosis accuracy. The administrative module provides staff with complete visibility over hospital operations, including analytics, doctor availability, and patient flow.

Furthermore, the system emphasizes data security and user authentication. Role-based access controls ensure that users can only access information relevant to their role. All data is stored in a secure database with regular backups, reducing the risk of data loss. With its modular architecture and scalable design, the proposed system can be expanded to include additional functionalities such as electronic health records, billing integration, and telemedicine in the future.

3.3 FEASIBILITY STUDY

Technical Feasibility:

The system will be developed using widely supported web technologies such as HTML, CSS, JavaScript, PHP, and MySQL. These tools are robust, cost-effective, and well-documented, making the project technically feasible. The design will be responsive, ensuring compatibility with desktops, tablets, and mobile devices.

Economic Feasibility:

Compared to traditional systems, this digital platform reduces operational costs significantly. It minimizes paper usage, manual errors, and administrative staffing needs. Additionally, open-source tools and in-house development further reduce initial setup costs, making the project economically viable.

Operational Feasibility:

From a usability perspective, the system is intuitive and user-friendly. Training requirements for end-users (patients, doctors, and staff) are minimal due to the system's simplified interface. With automation of routine tasks, staff productivity is expected to increase, contributing to smoother hospital operations.

CHAPTER 4
SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

Processor: A minimum of 2.0 GHz processor is required for efficient request handling and smooth system operation. A more powerful processor (e.g., 2.5 GHz or higher) is recommended for handling higher data volumes and concurrent users, especially during peak times.

RAM: The system requires at least 4 GB of RAM to support multiple simultaneous processes (patient appointments, doctor schedules, and administrative tasks). For hospitals with a high volume of users, upgrading to 8 GB or more of RAM ensures better performance and scalability.

Hard Disk: A 500 GB hard disk is the baseline for data storage. This allows sufficient space for patient records, doctor profiles, appointment histories, and system logs. Larger hospitals or those with a high patient base may need additional storage via external or cloud-based solutions.

4.2 SOFTWARE REQUIREMENTS

Frontend Technologies:

- **HTML:** Used for structuring the web pages.
- **CSS:** Provides styling for the pages to ensure a user-friendly interface.
- **JavaScript:** Adds dynamic functionality (e.g., form validation, interactive search, appointment booking) to the system, making it responsive across modern browsers and devices.

Backend Technologies:

- **PHP 7.x:** A widely used scripting language for handling user requests, form submissions, and session management. It integrates seamlessly with the MySQL database and ensures improved performance and security over older PHP versions.

Database System:

- **MySQL:** An open-source relational database management system (RDBMS) that offers efficient data storage and retrieval. MySQL is capable of handling large datasets, crucial for managing patient records, doctor profiles, and appointments.

Server Environment:

- **Apache:** Web server software used to host the system, typically managed through XAMPP or WAMP stacks for easy setup and configuration. These software stacks allow the integration of PHP and MySQL, ensuring a stable and scalable platform for hospital operations and user interaction.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

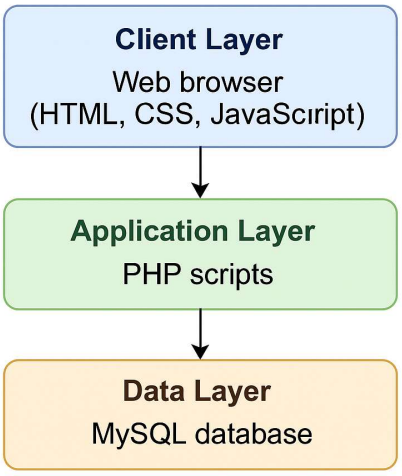

Fig 5.1.1 System Architecture

The System Architecture of the Doctor Appointment and Hospital Management System follows a three-tier design comprising the Client Layer, Application Layer, and Data Layer. The Client Layer consists of the user interface, built using HTML, CSS, and JavaScript, which allows users to interact with the system via a web browser. The Application Layer is responsible for processing business logic using PHP, handling user requests, managing sessions, and interacting with the database. Finally, the Data Layer uses MySQL to store and manage all system data, such as patient records, doctor profiles, and appointment schedules, ensuring fast and reliable data retrieval. This three-tier architecture ensures a well-organized, scalable, and maintainable system where each layer has a distinct responsibility, providing a seamless user experience and robust backend operations.

5.2 MODULE DESCRIPTION

Module 1: Admin Module

The Admin Module serves as the control center of the entire system. Administrators have comprehensive access to user management, including the ability to add, update, or remove both patient and doctor profiles. This module also allows the admin to review and monitor system activity, generate reports on appointments and system usage, and manage schedules or availability settings globally. The administrator ensures that the platform functions smoothly and that data remains consistent and secure across all modules.

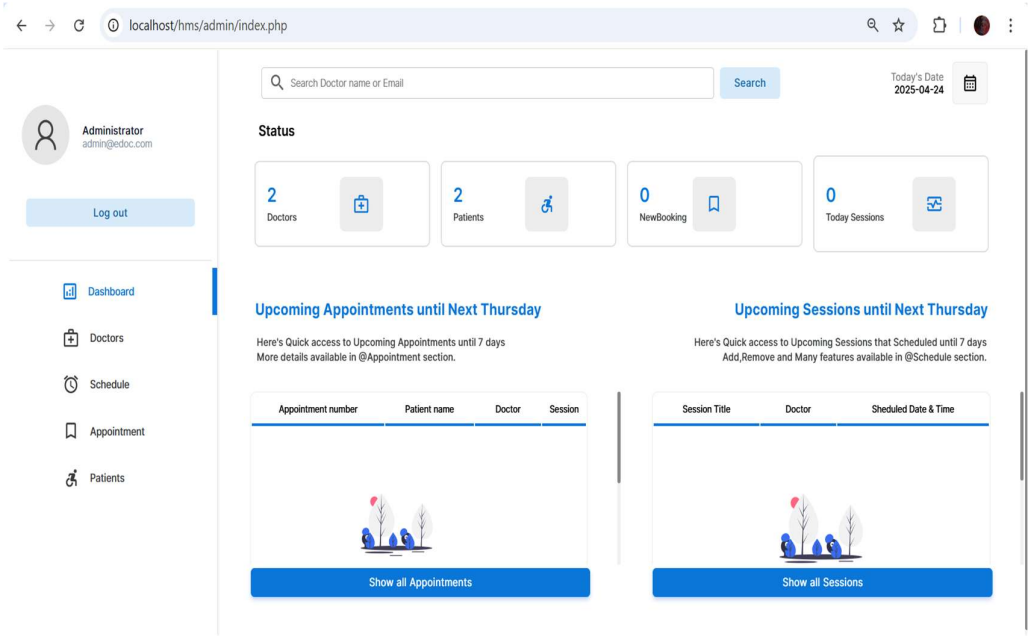

Fig 5.2.1 Admin Module

Module 2: Doctor Module

In the Doctor Module, medical professionals can manage their profiles, set their available days and time slots, and view the list of booked appointments. Doctors can accept, reschedule, or reject appointments based on their availability. Additionally, this module provides a space for doctors to view patient queries and respond accordingly, improving communication and patient satisfaction. Doctors can also review their past appointment history and manage their consultation slots dynamically.

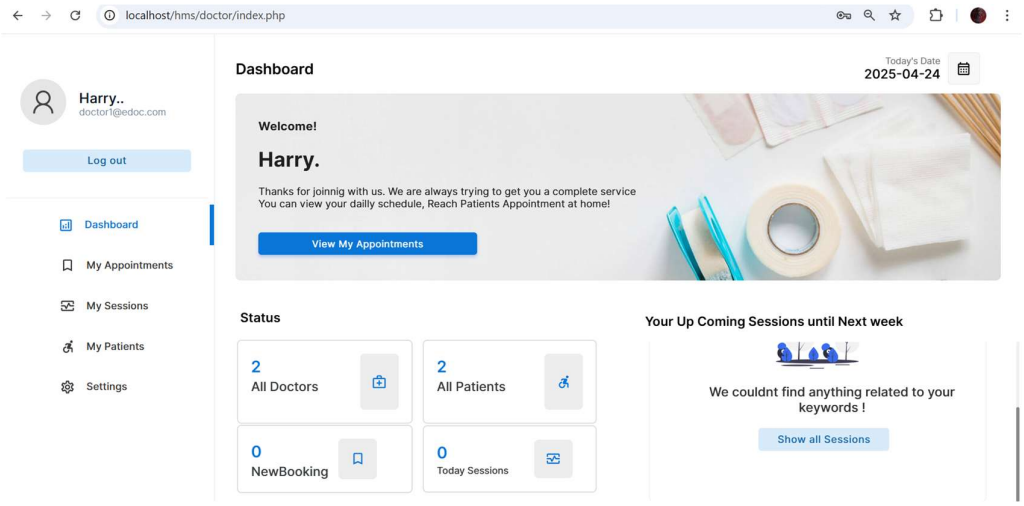

Fig 5.2.2 Doctor Module

Module 3: Patient Module

The Patient Module enables patients to register on the platform, log in securely, and access a user-friendly dashboard. They can browse the list of doctors filtered by specialization, location, or availability, and book appointments in real time. Patients can also manage their personal profiles, check appointment status, cancel or reschedule bookings, and view their history of consultations. This module ensures that patients have full control over their interactions with healthcare providers through an intuitive interface.

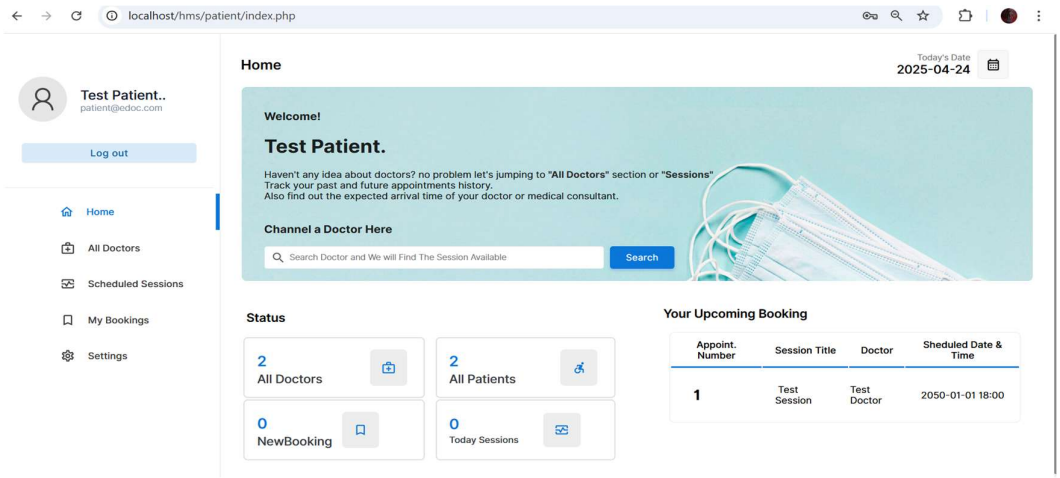

Fig 5.2.3 Patient Module

Module 4: Appointment Scheduling Module

This module handles the real-time appointment booking mechanism. It cross-references doctor availability with patient preferences and confirms slots dynamically. It prevents double bookings and notifies both doctors and patients upon confirmation, cancellation, or rescheduling. This module is crucial to ensuring smooth workflow and avoids clashes by checking the current status of appointments in the database. The scheduling engine also manages buffer times, slot durations, and availability blocks efficiently.

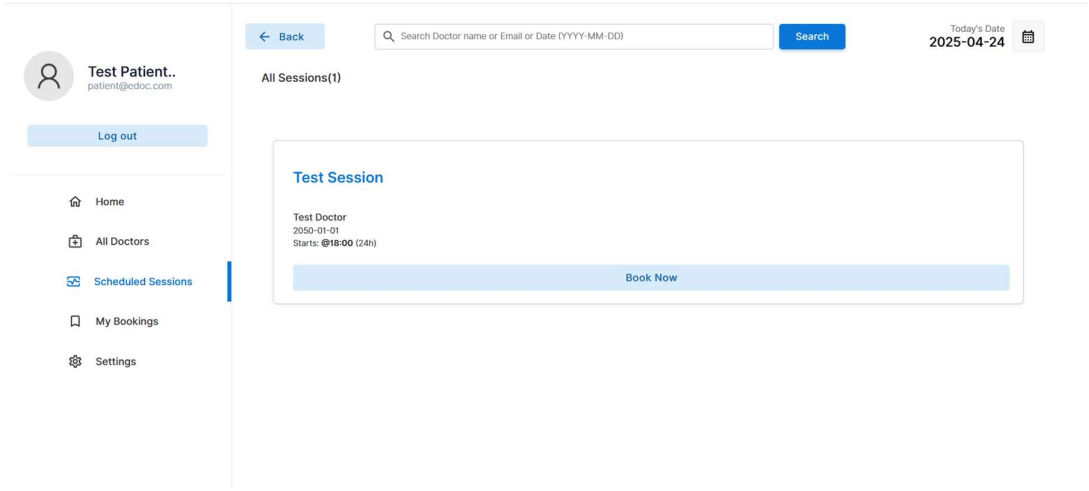

Fig 5.2.4 Appointment Schedule Module

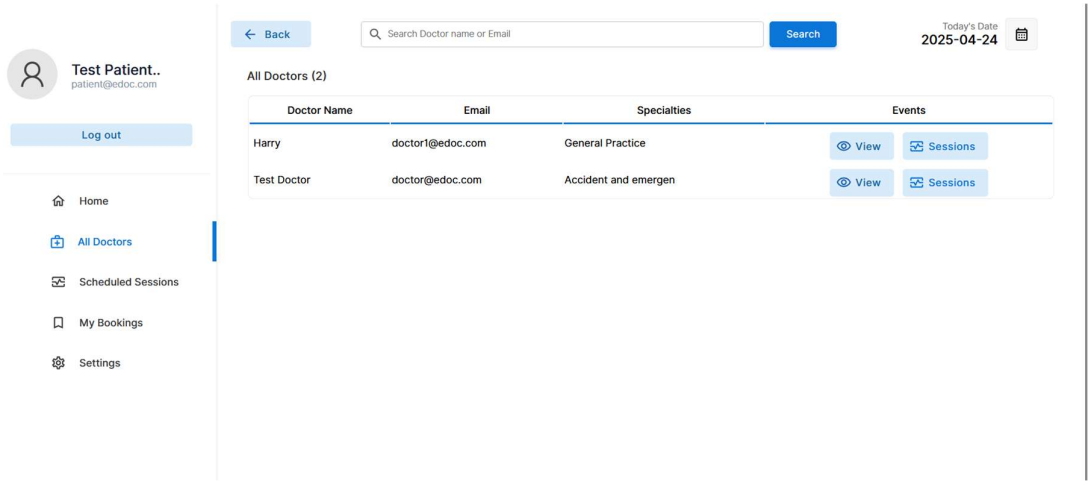

Fig 5.2.5 Doctor Schedule Module

Module 5: Login/Registration Module

The Login and Registration Module provides a secure gateway to access the system. New users can sign up by entering their credentials and personal details. This module handles role-based registration, assigning different privileges to patients, doctors, and administrators. Passwords are encrypted to maintain security, and role validation ensures that users only access functionalities relevant to their roles. The module also supports login sessions, password recovery, and profile verification for enhanced user management.

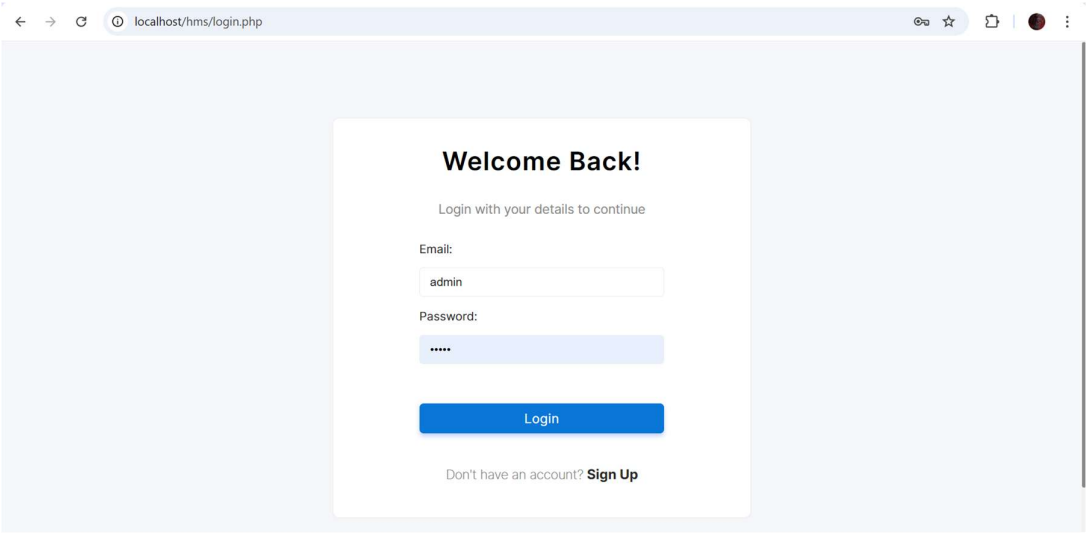

Fig 5.2.6 Login Module

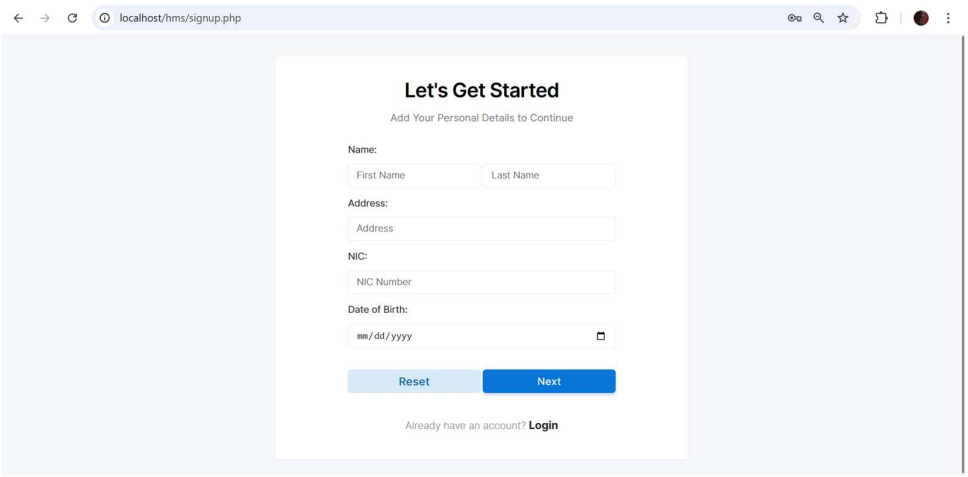

Fig 5.2.7 Registration Module

CHAPTER 6
CONCLUSION AND FUTURE ENHANCEMENT

6.1)CONCLUSION

The Doctor Appointment System represents a significant advancement in the integration of digital technologies into the healthcare appointment scheduling process. Through the creation of a centralized, user-friendly, and role-based web application, the system addresses key issues inherent in traditional and semi-digital methods of appointment booking—such as inefficiencies, lack of coordination, data redundancy, and excessive patient waiting times. By allowing patients to register, search for specialists, and book appointments online, the system drastically reduces the burden on hospital front desks and enhances user convenience. Doctors benefit from a structured platform that enables them to manage their schedules, view patient details, and communicate effectively without administrative bottlenecks. Administrators can monitor the system's performance, manage users, and ensure smooth operations. This well-structured digital workflow reduces human error, increases system transparency, and saves valuable time for all stakeholders. Furthermore, the modular and scalable architecture of the platform ensures that it is capable of incorporating emerging functionalities, such as telemedicine, payment gateways, and real-time notifications, in future iterations. The project’s success underscores the importance of user-centered design and agile development practices, which have enabled the creation of a system that not only meets current demands but is also positioned for long-term relevance and adaptability. Overall, the system contributes meaningfully to the evolution of healthcare delivery by bridging gaps between patients and providers through technology, improving service quality, operational efficiency, and healthcare accessibility.

6.2)FUTURE ENHANCEMENT

- **Mobile Application Development:** Creating dedicated apps for Android and iOS will enhance usability and allow access on the go.
- **Real-Time Communication:** Adding live chat features or integrating telemedicine capabilities will foster better interaction between patients and doctors.
- **Reminder Notifications:** Implementing SMS and email APIs for automatic appointment reminders and confirmations.
- **Online Payment Integration:** Integrating secure payment gateways to allow users to pay for services digitally.
- **Multi-language Support:** Supporting regional languages to cater to a diverse user base.
- **Patient Feedback and Rating System:** Allowing patients to rate doctors and provide feedback to improve service quality.
- **Data Analytics Dashboard:** Adding visual analytics for admins and doctors to monitor performance, trends, and utilization.

APPENDIX

A1.1 SAMPLE CODE

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="css/animations.css">
   <link rel="stylesheet" href="css/main.css">
   <link rel="stylesheet" href="css/index.css">
   <title>eDoc</title>
   <style>
      table{
         animation: transitionIn-Y-bottom 0.5s;
      }
   </style>

</head>
<body>

   <div class="full-height">
      <center>
      <table border="0">
         <tr>
            <td width="80%">
               <font class="edoc-logo">eDoc. </font>
               <font class="edoc-logo-sub">| AN IDT PROJECT</font>
            </td>
            <td width="10%">
```

```
            <a href="login.php" class="non-style-link"><p class="nav-item">LOGIN</p></a>
        </td>
        <td width="10%">
            <a href="signup.php" class="non-style-link"><p class="nav-item" style="padding-right: 10px;">REGISTER</p></a>
        </td>
    </tr>

    <tr>
        <td colspan="3">
            <p class="heading-text">Timely Appointments, Hassle-Free Service

            </p>

        </td>
    </tr>
    <tr>
        <td colspan="3">
            <p class="sub-text2">Your health matters.
                <br>Your time does too.<br>Feeling unwell? Don’t wait in line.
                <br>Book your doctor appointment online with eDoc — fast, simple, and 100% free.
                <br>Get the care you need, when you need it.</p>
        </td>
    </tr>
    <tr>

        <td colspan="3">
            <center>
            <a href="login.php" >
```

```
                    <input type="button" value="Make Appointment" class="login-btn
btn-primary btn" style="padding-left: 25px;padding-right: 25px;padding-top:
10px;padding-bottom: 10px;">
                </a>
            </center>
            </td>

        </tr>
        <tr>
            <td colspan="3">

            </td>
        </tr>
    </table>
    <p class="sub-text2 footer-hashen">Hospital Management System.</p>
  </center>

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

Login.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="css/animations.css">
    <link rel="stylesheet" href="css/main.css">
    <link rel="stylesheet" href="css/login.css">

    <title>Login</title>




</head>
<body>
    <?php

    //learn from w3schools.com
    //Unset all the server side variables

    session_start();

    $_SESSION["user"]="";
    $_SESSION["usertype"]="";

    // Set the new timezone
    date_default_timezone_set('Asia/Kolkata');
    $date = date('Y-m-d');
```

```
    $_SESSION["date"]=$date;


    //import database
    include("connection.php");




    if($_POST){

        $email=$_POST['useremail'];
        $password=$_POST['userpassword'];

        $error='<label for="promter" class="form-label"></label>';

        $result= $database->query("select * from webuser where email='$email'");
        if($result->num_rows==1){
            $utype=$result->fetch_assoc()['usertype'];
            if ($utype=='p'){
                //TODO
               $checker = $database->query("select * from patient where pemail='$email'
and ppassword='$password'");
                if ($checker->num_rows==1){

                    //   Patient dashbord
                    $_SESSION['user']=$email;
                    $_SESSION['usertype']='p';

                    header('location: patient/index.php');

                }else{
```

```
            $error='<label for="promter" class="form-label" style="color:rgb(255, 62, 62);text-align:center;">Wrong credentials: Invalid email or password</label>';
            }

        }elseif($utype=='a'){
            //TODO
           $checker = $database->query("select * from admin where aemail='$email' and apassword='$password'");
            if ($checker->num_rows==1){

                //   Admin dashbord
                $_SESSION['user']=$email;
                $_SESSION['usertype']='a';

                header('location: admin/index.php');

            }else{
                $error='<label for="promter" class="form-label" style="color:rgb(255, 62, 62);text-align:center;">Wrong credentials: Invalid email or password</label>';
            }

        }elseif($utype=='d'){
            //TODO
                    $checker = $database->query("select * from doctor where docemail='$email' and docpassword='$password'");
            if ($checker->num_rows==1){

                //   doctor dashbord
                $_SESSION['user']=$email;
                $_SESSION['usertype']='d';
                header('location: doctor/index.php');
```

```
            }else{
                $error='<label for="promter" class="form-label" style="color:rgb(255, 62, 62);text-align:center;">Wrong credentials: Invalid email or password</label>';
            }

        }

    }else{
        $error='<label for="promter" class="form-label" style="color:rgb(255, 62, 62);text-align:center;">We cant found any acount for this email.</label>';
    }




}else{
    $error='<label for="promter" class="form-label"> </label>';
}

?>



<center>
<div class="container">
    <table border="0" style="margin: 0;padding: 0;width: 60%;">
        <tr>
            <td>
                <p class="header-text">Welcome Back!</p>
```

```
            </td>
        </tr>
    <div class="form-body">
        <tr>
            <td>
                <p class="sub-text">Login with your details to continue</p>
            </td>
        </tr>
        <tr>
            <form action="" method="POST" >
            <td class="label-td">
                <label for="useremail" class="form-label">Email: </label>
            </td>
        </tr>
        <tr>
            <td class="label-td">
                        <input type="email" name="useremail" class="input-text"
placeholder="Email Address" required>
            </td>
        </tr>
        <tr>
            <td class="label-td">
                <label for="userpassword" class="form-label">Password: </label>
            </td>
        </tr>

        <tr>
            <td class="label-td">
                    <input type="Password" name="userpassword" class="input-text"
placeholder="Password" required>
            </td>
        </tr>
```

```
        <tr>
            <td><br>
            <?php echo $error ?>
            </td>
        </tr>

        <tr>
            <td>
                    <input type="submit" value="Login" class="login-btn btn-primary
btn">
            </td>
        </tr>
    </div>
        <tr>
            <td>
                <br>
                  <label for="" class="sub-text" style="font-weight: 280;">Don't have
an account&amp;#63; </label>
                <a href="signup.php" class="hover-link1 non-style-link">Sign Up</a>
                <br><br><br>
            </td>
        </tr>




                </form>
    </table>

  </div>
</center>
```

```
<!-- Code injected by live-server -->
<script>
   // <![CDATA[  <-- For SVG support
   if ('WebSocket' in window) {
      (function () {
         function refreshCSS() {
            var sheets = [].slice.call(document.getElementsByTagName("link"));
            var head = document.getElementsByTagName("head")[0];
            for (var i = 0; i < sheets.length; ++i) {
               var elem = sheets[i];
               var parent = elem.parentElement || head;
               parent.removeChild(elem);
               var rel = elem.rel;
                     if (elem.href && typeof rel != "string" || rel.length == 0 ||
rel.toLowerCase() == "stylesheet") {
                  var url = elem.href.replace(/(&|\?)_cacheOverride=\d+/, '');
                 elem.href = url + (url.indexOf('?') >= 0 ? '&' : '?') + '_cacheOverride='
+ (new Date().valueOf());
               }
               parent.appendChild(elem);
            }
         }
         var protocol = window.location.protocol === 'http:' ? 'ws://' : 'wss://';
         var address = protocol + window.location.host + window.location.pathname
+ '/ws';
         var socket = new WebSocket(address);
         socket.onmessage = function (msg) {
            if (msg.data == 'reload') window.location.reload();
            else if (msg.data == 'refreshcss') refreshCSS();
         };
                                                  if (sessionStorage &&
!sessionStorage.getItem('IsThisFirstTime_Log_From_LiveServer')) {
```

```
            console.log('Live reload enabled.');
            sessionStorage.setItem('IsThisFirstTime_Log_From_LiveServer', true);
        }
    })();
}
else {
        console.error('Upgrade your browser. This Browser is NOT supported WebSocket for Live-Reloading.');
}
// ]]>
</script>
</body>
</html>
```

A1.2SCREENSHOTS

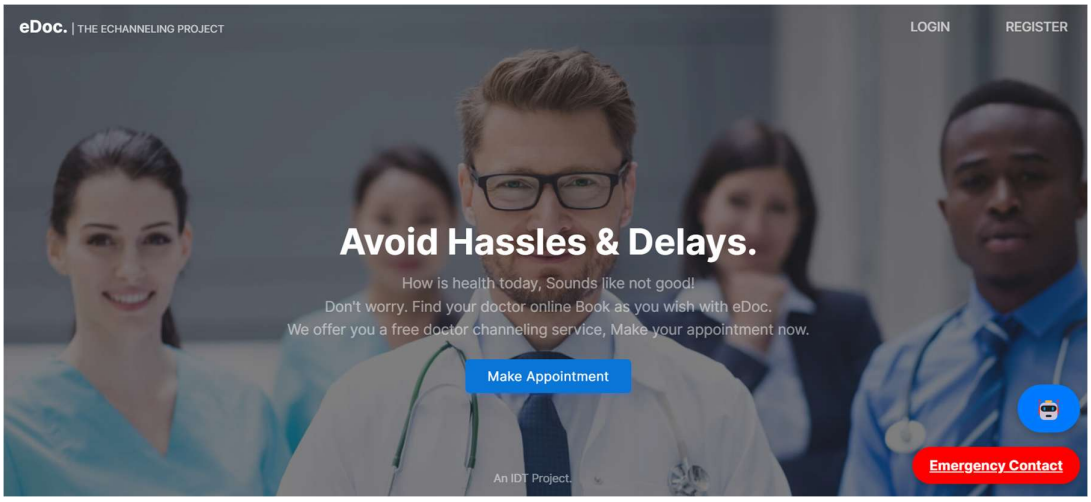

Fig A1.2.1 Main page

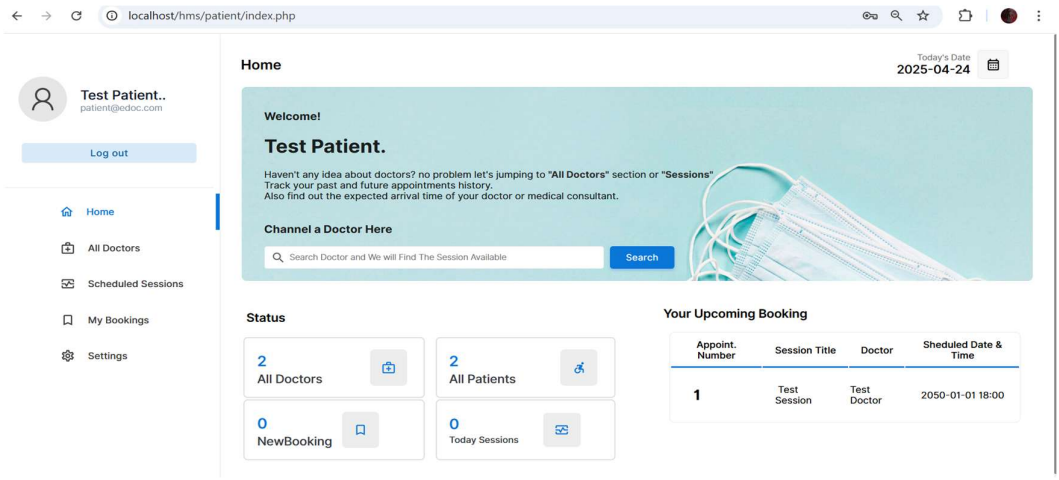

Fig A1.2.2 User Login

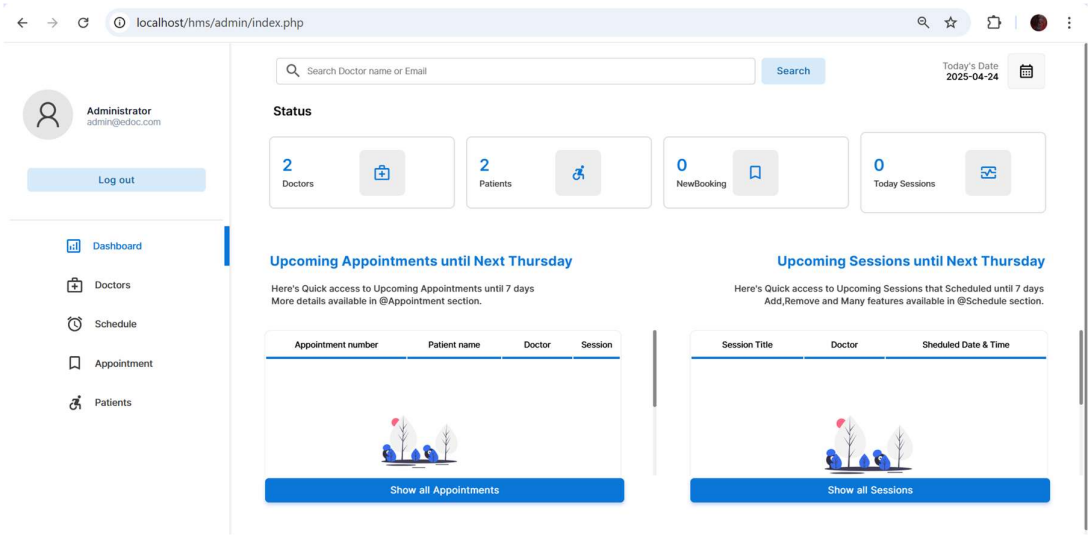

Fig A1.2.3 Admin Login

REFERENCES

[1] Shanika Wickramasinghe, “Choosing the Best Web Application Architecture”, October 6, 2020. [Online], Available: https://crowdbotics.com/posts/blog/how-to-choose-the-bestarchitecture-for-your-web-application/

[2] Chandrapal Singh, “Fundamentals Of Web Application Architecture”, July 14, 2020. [Online], Available: https://www.appventurez.com/blog/web-application-architecture

[3] Omer Imran Malik, “What is Sensitive Data Exposure Vulnerability & How to Avoid It?” August 12, 2019. [Online], Available: https://securiti.ai/blog/sensitive-data-exposure/

[4] Eugene Makieiev, “How to Create a Modern Web Application Architecture?” January 7, 2021. [Online], Available: https://integrio.net/blog/modern-web-application-architecture

[5] Google developer’s guide, “About PageSpeed Insights” September 21, 2020. [Online], Available: https://developers.google.com/speed/docs/insights/v5/about [Accessed March 19, 2021].

[6] Michael Chika Diyoke, “Healthcare Ethics Issues and Challenges in Nigeria” August 16, 2019. [Online], Available: https://www.researchgate.net/publication/340536886_Healthcare_Ethics_Issues_and_Challenges_in_Nigeria [Accessed March 20, 2021].

[7] G. Domantas, “What Is CSS and How Does It Work?” May 2, 2018. [Online], Available: https://www.hostinger.com/tutorials/what-iscss/ [Accessed March 15, 2021].

[8] Rock Content Writer, “What is a Scripting Language and What are the Most Common Ones?” December 15, 2020. [Online], Available: https://rockcontent.com/blog/scripting-languages/ [Accessed March 15, 2021].

[9] Kolade Chris, “What is PHP? The PHP Programming Language Meaning Explained” August 30, 2019. [Online], Available: https://www.freecodecamp.org/news/what-is-php-the-phpprogramming-language-meaning-explained/ [Accessed March 17, 2021].

[10] Server hub, “MySQL: the most popular open-source database” March 13, 2021. [Online], Available: https://serverhub.com/kb/mysql-themost-popular-open-source-database-system/ [Accessed March 17, 2021].

[11] Ahmed Shamel Noori and Ali Fadhil Najem, “Hospital Management System: Design and Implementation,” B.Eng. thesis, Dept. Comp. Eng., University of Technology, Johor Bahru, Malaysia, 2021.

[12] Saad Alami, “Hospital Management Control System,” B.Sc. thesis, Dept. Comp. Sci., Mohammed V University of Rabat, Morocco, 2018.

[13] M. Sowmya, D. Anilchhandra Varma, M. Venugopalarao, M. Sailaja and T. Prasanth, “The Hospital Magenememt System,” B.Eng. thesis, Dept. Software Eng., Sri Lanka Institute of Information Technology, SRI LANKA, 2021.

[14] W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional. Ohio, USA: Apress, 2010.

[15] Janet Valade and Steve Suehring, PHP, MySQL, JavaScript, HTML5: All in one for Dummies. California, USA: Wiley, 2014.

[16] Prof. Dharminder Kumar, “Overview of System Analysis & Design” Lesson no. 1, pp. 23-37, 2013.

[17] B. Rajanisade, “Hospital Management System,” February 3, 2016. [Online], Available: https://www.coursehero.com/file/p2r99h3/1- Hospital Management System Ernest J. Okoromi What-are-the-users-demonstrable-needs-and-how-does-acandidate-system-meet/ [Accessed March 19, 2021].

[18] Prof. Dharminder Kumar, “Overview of System Analysis & Design” Lesson no. 1, pp. 42-45, 2013.

[19] Prof. Dharminder Kumar, “Overview of System Analysis & Design” Lesson no. 1, pp. 51-56, 2013.

[20] Dr. Mohammed Najm Abdulla, Dr. Intisar Al-Mejibli, Shaimaa Khamees Ahmed, “An Investigation Study of Hospital Management Information System,” Comp. Eng. Dept., University of Technology, Iraq, IJARCCE, 2017.

[21] Michal Malewicz, “The User Interface (UI) Design” June 5, 2020. [Online], Available: https://www.interactiondesign.org/literature/topics/ui-design [Accessed March 19, 2021].

[22] Felipe Guimaraes e Equipe Aela, “Nielsen’s Heuristics: 10 Usability Principles To Improve UI Design” June 29, 2019. [Online], Available: https://aelaschool.com/en/interactiondesign/10-usabilityheuristics-ui-design [Accessed March 19, 2021].

[23] O. O. Lawal, B. O. Afeni, and J. O. Mebawondu, “Development of Hospital Information Management Systems,” B.Sc. thesis, Dept. Comp. Sci., Joseph Ayo Babalola Univ., Ado-Ekiti, Nigeria, 2016

[24] Adam Hyde, “The Case for HTML Word Processors” December 11, 2016. [Online], Available: https://www.adamhyde.net/html-wordprocessors/