PROJECT REPORT

(Project Term August- November 2024)

Course Title: Server Side Scripting

Title of the Project
BookYourDoc - Doctor Appointment System

Submitted by

Rajesh Nahak

Regn No. 12220196
Course Code INT220

Submitted To:

Mr. Akshay Kumar

School of Computer Science and Engineering

LOVELY
PROFESSIONAL
UNIVERSITY

DECLARATION

We hereby declare that the project work entitled “BookYourDoc” is an authenticrecord of our own work carried out as requirements of Project for the award of Btech degree in Computer Science Engineering from Lovely Professional University, Phagwara, under the guidance of Akshay Kumar, during August to November 2024. All the information furnished in this project report is based on my own intensive work and is genuine.

Name of Student 1: Rajesh Nahak
Registration Number: 12216196

Name of Student 2: Arshad Zama
Registration Number: 12215798

CERTIFICATE

This is to certify that the declaration statement made by the student is correct to the best of my knowledge and belief. He /She have completed this Project under my guidance and supervision. The present work is the result of his/her original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Project is fit for the submission and partial fulfillment of the conditions for the award of B.tech degree in Computer Science Engineering from Lovely Professional University,Phagwara.:

Rajesh Nahak

Arshad Zama

Table of Contents

1. INTRODUCTION

The Doctor Appointment System is a **web-based application** designed to simplify and optimize the process of scheduling doctor appointments. Traditional methods of appointment scheduling, often involving phone calls or in-person visits, are time-consuming and inefficient. This system automates the scheduling process, allowing patients to book appointments online and doctors to manage their schedules with ease.

The project utilizes **PHP** for server-side processing and **MySQL** as its database, ensuring a stable and responsive user experience. The system supports multiple user roles, such as patients, doctors, and administrators, each with specific functionalities tailored to their needs. Patients can register, log in, and view available appointment slots in real-time, while doctors can approve or decline bookings and manage their schedules. Administrators have oversight of all users, schedules, and system operations, ensuring a smooth and secure workflow.

The purpose of this project is to **reduce administrative workload**, minimize scheduling conflicts, and enhance user satisfaction. By digitizing the appointment booking process, the system provides an accessible and reliable platform that benefits both healthcare providers and patients.

1.1. Scope and Objectives

The Doctor Appointment System aims to address several core objectives:

- **Improving Efficiency**: By automating the scheduling process, the system eliminates the need for manual booking, reducing errors and saving time.
- **Enhancing User Experience**: The platform offers a simple, user-friendly interface that allows patients to book appointments quickly and access medical services with ease.
- **Supporting Administrative Control**: Administrators can manage user data, track appointment bookings, and oversee system operations to ensure accuracy and reliability.

1.2. Project Features

The key features of the Doctor Appointment System include:

- **User Role Management**: Support for different roles (patients, doctors, and administrators) with specific permissions for each.
- **Real-time Scheduling**: Patients can view available slots and book appointments that are automatically reflected in the doctor’s schedule.

- **Appointment Management**: Doctors can review, approve, or cancel appointments and update their availability accordingly.
- **Administrative Control**: Administrators can manage users, monitor appointment history, and ensure data security.

1.2.1. User Roles and Functionalities

The system defines the following user roles and their corresponding functionalities:

- **Patients**: Can register, log in, view available doctors, book appointments, and receive booking confirmations.
- **Doctors**: Can manage their schedules, accept or decline appointments, and access patient information securely.
- **Administrators**: Have full control over the system, including user management, appointment tracking, and system maintenance.

1.2.2. Technical Specifications

The application is built with the following technologies and standards:

- **Backend**: PHP for server-side scripting and MySQL as the relational database management system.
- **Frontend**: HTML, CSS, and JavaScript for building a user-friendly interface.
- **Security**: Basic authentication mechanisms to protect user data and prevent unauthorized access

2. Profile of the Problem

2.1. Problem Statement

Managing doctor appointments using traditional methods presents a range of challenges for both medical professionals and patients. These challenges include:

- **Inefficiency in Scheduling**: Manual booking often involves phone calls, physical visits, or handwritten logs, leading to delays and time-consuming processes.
- **High Wait Times**: Due to overlapping appointments and lack of real-time updates, patients frequently experience long waiting periods.
- **Increased Administrative Burden**: Staff members face a heavy administrative workload when manually managing schedules, cancellations, and rescheduling requests.
- **Patient Dissatisfaction**: The lack of a streamlined appointment system can lead to frustration among patients, reducing overall satisfaction with healthcare services.

These problems emphasize the need for a modernized, efficient solution that automates and optimizes the appointment scheduling process, minimizing errors and enhancing user experience.

2.2. Scope of the Study

The Doctor’s Appointment System is designed to solve the challenges identified in the problem statement by offering a comprehensive online platform that benefits both patients and healthcare providers. The system’s scope includes:

- **Online Appointment Scheduling**: Patients can easily schedule, modify, and view their appointments online, eliminating the need for in-person or phone-based booking. This feature also allows for real-time updates on doctor availability, reducing the risk of scheduling conflicts.
- **Efficient Schedule Management for Doctors**: Doctors have access to an organized interface for viewing and managing their appointments, allowing them to approve or reject requests, set availability, and update their schedules.
- **Reduced Wait Times**: By streamlining the scheduling process and providing real-time updates, the system minimizes the wait times for patients, enhancing the overall experience.
- **Administrative Oversight**: Administrators can monitor and manage all user roles,

appointment bookings, and system operations. This oversight helps maintain the accuracy, security, and efficiency of the entire system.

2.2.1. Benefits for Patients

- **Convenience**: Patients can schedule and manage appointments from any location, at any time, using the online platform.
- **Transparency**: The system provides real-time updates on appointment availability and status, reducing uncertainty.
- **Improved Access**: Patients have quick access to available doctors and services, facilitating a better healthcare experience.

2.2.2. Benefits for Doctors

- **Time Management**: Doctors can efficiently plan their schedules, minimize booking conflicts, and maximize appointment efficiency.
- **Organized Workflow**: The system helps doctors maintain an organized view of upcoming appointments, patient details, and availability.
- **Reduced Administrative Effort**: By automating scheduling, doctors and their staff spend less time managing appointments, allowing them to focus more on patient care.

2.2.3. Benefits for Administrators

- **Centralized Control**: Administrators have the ability to oversee all system operations, ensuring accurate user management and secure data handling.
- **Data Accuracy**: The system automates booking and scheduling processes, reducing human error in managing appointments.
- **Operational Efficiency**: By digitizing appointment processes, administrators can reduce workload and streamline operations within the healthcare facility.

3. Existing System

3.1. Introduction

Traditional methods of scheduling doctor appointments, or existing systems that operate without digital support, often involve the following challenges:

- **Manual Booking Process**: Appointments are typically scheduled via phone calls or in-person visits, relying heavily on human intervention and paper-based logs. This process can lead to miscommunication, overbooking, and scheduling conflicts.
- **Limited Accessibility**: Patients have to contact the healthcare facility during specific hours to book an appointment, making the process inflexible and inconvenient.
- **High Dependency on Staff**: Scheduling, rescheduling, and cancellations require staff involvement, which can lead to administrative overload, especially in busy clinics or hospitals.
- **Lack of Real-Time Updates**: Without a digital system, appointment availability isn't updated in real-time, which can cause discrepancies and delays in the scheduling process.

3.2. Existing Software

Several digital appointment scheduling systems are currently available, each with unique features and limitations. These systems aim to address the inefficiencies of manual scheduling, but they often fall short in certain areas:

3.2.1. Examples of Existing Systems

- **Hospital Management Systems (HMS)**: Many healthcare institutions use HMS platforms that include appointment scheduling modules. These systems provide a range of functionalities, such as managing patient data, scheduling visits, and recording medical history. However, they can be complex and costly for smaller clinics or private practices.
- **Third-Party Appointment Platforms**: Applications like Zocdoc and Practo provide patients the ability to search for doctors and book appointments online. While these platforms are user-friendly, they can be limited in terms of customization and may not integrate seamlessly with a clinic's specific needs.
- **Clinic-Specific Systems**: Some clinics use tailor-made software developed to suit their specific requirements. These systems often suffer from limited scalability, outdated user interfaces, and a lack of integration with other healthcare services.

3.2.2. Gaps and Limitations in Existing Systems

- **Customization Issues**: Many existing platforms lack the flexibility to adapt to specific needs of smaller clinics or individual practices, requiring substantial adjustments to meet unique requirements.
- **Integration Challenges**: Current systems often face difficulties in integrating with other medical databases, insurance systems, or electronic health record (EHR) platforms, limiting the seamless flow of patient information.
- **Cost and Complexity**: Advanced systems with extensive features can be costly, making them inaccessible for small healthcare providers. Moreover, they may require specialized training for staff to operate, adding to the time and cost burdens.
- **User Experience Limitations**: Many systems focus more on the administrative side and often neglect a user-friendly interface for patients and doctors. This can lead to lower adoption rates and dissatisfaction among users.

The **Doctor's Appointment System** developed in this project addresses these gaps by offering a cost-effective, user-friendly, and easily customizable solution that supports seamless scheduling and administrative management, catering to both small clinics and larger medical facilities.

4. Problem Analysis

4.1. Product Definition

The **Doctor’s Appointment System** is a web-based application designed to optimize the scheduling and management of doctor appointments. It is built to cater to multiple user roles, each with specific functionalities that streamline the booking process and enhance the overall experience for both patients and healthcare providers. The primary features of the system include:

4.1.1. User Roles and Functions

- **Patients**:
    - Register and create an account.
    - View available doctors and their schedules.
    - Book, modify, and cancel appointments.
    - Receive confirmation notifications for bookings.
    - Access appointment history.
- **Doctors**:
    - Manage schedules, including setting availability and modifying time slots.
    - Approve or decline appointment requests.
    - View upcoming appointments and patient details securely.
    - Access patient appointment history for reference.
- **Administrators**:
    - Oversee user registration and maintain user accounts.
    - Manage doctors' schedules and monitor appointment activity.
    - Ensure the security of data and the integrity of the system.
    - Perform system maintenance and updates.

4.2. Feasibility Analysis

4.2.1. Technical Feasibility

The system leverages widely-used, reliable technologies to ensure stability and performance:

- **Backend**: PHP is used for server-side processing due to its efficiency in handling web applications.
- **Database**: MySQL provides a robust and scalable database solution for storing user data, schedules, and appointment details.
- **Frontend**: HTML, CSS, and JavaScript are utilized to create an intuitive and responsive user interface.

4.2.2. Operational Feasibility

The system is designed with simplicity in mind, making it easy for users across different roles:

- **Patients** can access a straightforward booking interface, view available slots in real-time, and receive instant updates on their appointments.
- **Doctors** have access to an organized schedule management system, reducing administrative overhead.
- **Administrators** benefit from a centralized dashboard, enabling them to oversee operations efficiently.

4.2.3. Economic Feasibility

Implementing the Doctor's Appointment System reduces costs in several areas:

- **Reduced Administrative Costs**: Automating appointment scheduling decreases the need for additional administrative staff.
- **Time Savings**: The streamlined booking process saves time for both patients and healthcare providers, allowing staff to focus on other critical tasks.
- **Cost-Effective Maintenance**: Using open-source technologies (like PHP and MySQL) lowers development and maintenance costs, making the system financially viable for smaller clinics.

4.3. Project Plan

The development of the Doctor's Appointment System follows a structured approach, with each stage ensuring that project objectives are met:

- **Planning**: Requirement gathering, defining system scope, and feasibility analysis.

- **Design**: Creation of wireframes, system architecture, database design, and user interface mockups.
- **Coding**: Development of the backend (PHP, MySQL) and frontend (HTML, CSS, JavaScript) components.
- **Testing**: Unit testing, integration testing, and user acceptance testing to ensure functionality and usability.
- **Deployment**: Deploying the system on a web server, followed by monitoring and optimization.

4.4. Flowcharts

Here are the core function flowcharts described in detail. I’ll provide a summary of each process:

4.4.1. Booking an Appointment

- **Start → Patient Login → Select Doctor → View Available Slots → Choose Slot → Confirm Booking → End**

4.4.2. Viewing and Managing Schedules (Doctor)

- **Start → Doctor Login → View Schedule → Modify Availability** (if needed) → **Approve/Decline Appointments → Save Changes → End**

4.5. Pseudocode

4.5.1. Checking Doctor Availability

```
Function CheckAvailability(doctorID, date, timeSlot):
    // Retrieve the schedule for the given doctor on the specified date
    doctorSchedule = GetSchedule(doctorID, date)

    // Check if the timeSlot is available
    If timeSlot in doctorSchedule.availableSlots:
        Return True
    Else:
        Return False
```

4.5.2. Booking an Appointment

sql

```
Function Book Appointment(patient, doctor, date, timeslot):
    // Check if the selected time slot is available
    If CheckAvailability (doctorID, date, timeSlot) == True:
        // Reserve the time slot
        SaveAppointment(patientID, doctorID, date, timeSlot)
        Return "Appointment Confirmed"
    Else:
        Return "Time Slot Unavailable"
```

5. Implementation

5.1. Implementation of the Project

The coding phase of the Doctor’s Appointment System was focused on building a robust and user-friendly web-based application using PHP for the server-side and MySQL for database management. Here’s a breakdown of how the system was developed:

5.1.1. Coding Overview

- **PHP Scripts**: PHP was used to handle server-side logic, including user authentication, form submissions, appointment scheduling, and database interactions. The use of PHP allowed for seamless communication between the frontend and the backend.
    - **User Authentication**: A PHP script validates user credentials during login and registration, ensuring that only authorized users can access the system.
    - **Appointment Booking**: A script allows patients to select a doctor, choose a date, and confirm a booking. It also manages appointment conflicts by checking the doctor's availability.

    - **Doctor Schedule Management**: A PHP script enables doctors to set their availability, approve or decline appointment requests, and manage their calendars.

- **SQL Scripts**: MySQL was used to store and manage all data related to users, appointments, schedules, and system settings.
    - **Database Structure**: Tables were created for Users, Appointments, Doctors, and Schedules. Each table was designed with appropriate relationships and constraints to maintain data integrity.
    - **Data Retrieval and Manipulation**: SQL queries were employed to retrieve, update, delete, and insert data efficiently, ensuring the system operates smoothly with minimal load times.
    - **Backup and Recovery**: SQL scripts were also created to handle data backups and restore operations, safeguarding against data loss.

5.1.2. Key Features Implemented

- **User Registration and Login**: A secure registration system for patients, doctors, and administrators using PHP sessions and MySQL.
- **Appointment Booking System**: Automated appointment booking with real-time updates on availability, minimizing scheduling conflicts.
- **Doctor Dashboard**: A dedicated interface for doctors to manage appointments, set availability, and view patient history.
- **Admin Panel**: A control center for administrators to manage users, oversee appointments, and monitor system activity.

5.2. Conversion Plan

The transition from a traditional manual system to an online, web-based Doctor’s Appointment System involves several steps to ensure a smooth and efficient changeover:

5.2.1. Transition from Manual to Digital

1. **Data Collection and Entry**: Collect all relevant data from existing manual records, such as patient information, doctor schedules, and current appointment logs.
    - **Data Input**: Digitally input the collected information into the new system’s database using structured import tools or direct entry.

2. **Staff Training**: Provide training for all staff members, including doctors and administrative personnel, on how to use the new online system.
    - **Hands-on Demonstrations**: Conduct demonstrations on key features like booking appointments, managing schedules, and accessing patient data.
    - **Documentation and Guides**: Supply manuals or online guides that detail how to navigate the system and perform essential tasks.
3. **Pilot Testing**: Implement the system in a trial phase with a small number of users (selected patients and doctors) to identify any issues before full deployment.
    - **Feedback Collection**: Gather feedback from test users to address potential problems or make adjustments.
4. **Full Launch**: Roll out the system to all users, replacing manual booking entirely. Ensure that users have easy access to support during this transition phase.

5.2.2. Migrating from an Existing Digital Solution

1. **Data Export from Current System**: Extract all relevant data from the current digital system, including user information, appointments, and historical records.
2. **Data Transformation and Import**: Use data transformation tools to convert the extracted data into a format compatible with the new MySQL database.
3. **System Testing**: Verify that all data has been migrated accurately and perform integrity checks to ensure no data is missing or corrupted.
4. **System Integration**: Test the new system's compatibility with any existing infrastructure (e.g., medical records or billing systems) to ensure seamless integration.

5.3. Post-Implementation and Software Maintenance

Maintaining the Doctor’s Appointment System is crucial for ensuring its longevity, security, and continued user satisfaction. Here’s how ongoing maintenance will be handled:

5.3.1. Bug Fixes

- **Continuous Monitoring**: Regularly monitor the system for bugs or issues reported by users.
- **Issue Tracking**: Utilize a bug tracking system to log and prioritize fixes based on severity and impact.

- **Routine Patching**: Apply patches and updates to address any identified bugs and improve system stability.

5.3.2. Feature Updates

- **User Feedback**: Collect feedback from patients, doctors, and administrators to understand which features require improvement or enhancement.
- **Periodic Feature Releases**: Plan periodic updates to introduce new features, optimize performance, and enhance usability.
- **Testing New Features**: Conduct thorough testing of new functionalities before they go live, including unit testing, integration testing, and user acceptance testing.

5.3.3. Data Security and Backup

- **Data Encryption**: Ensure that all sensitive data (e.g., patient information, login credentials) is encrypted to maintain confidentiality.
- **Regular Backups**: Schedule daily or weekly database backups to prevent data loss and allow for data recovery in case of emergencies.
- **Security Audits**: Perform regular security audits to detect vulnerabilities, ensure compliance with data privacy standards, and apply security best practices.

5.3.4. User Support and Training

- **Help Desk**: Provide a support system or help desk for users experiencing issues or requiring guidance.
- **Ongoing Training**: Offer refresher courses for staff as new features are added to the system, ensuring all users are proficient with the latest updates.

6. Project Legacy

6.1. Outcomes

The Doctor’s Appointment System successfully automated the booking and scheduling process for healthcare facilities, creating a more efficient and user-friendly environment. The project met its primary goals by:

- **Streamlining Appointment Scheduling**: The system allowed patients to view available slots and book appointments in real-time, significantly reducing wait times and minimizing scheduling conflicts.
- **Reducing Administrative Burden**: Doctors and administrative staff now manage appointments digitally, freeing up time for other tasks and reducing paperwork.
- **Improving Patient Satisfaction**: The system’s easy-to-navigate interface and instant booking confirmation enhanced the patient experience, fostering greater trust and reliability.
- **Enhanced Data Management**: The use of MySQL for data storage ensured accurate record-keeping, data integrity, and quick retrieval of information.

6.2. Challenges

Several challenges emerged during the development and deployment of the project:

- **User Adoption**: Transitioning from a manual system to a digital platform required significant user adaptation, especially for older patients and staff unfamiliar with technology. Training and support were crucial for overcoming this challenge.
- **Data Security**: Ensuring the confidentiality of patient data was a critical concern. Implementing robust encryption, secure authentication mechanisms, and data backup strategies required careful planning and testing.
- **System Scalability**: Building a system that could scale with increasing numbers of users and appointments was challenging. Optimizing database queries and server performance was necessary to maintain speed and reliability as the user base grew.
- **Customization Needs**: Different healthcare providers had unique scheduling requirements. Balancing a flexible system architecture with specific demands required iterative adjustments to the project’s design.

6.3. Lessons Learned

The project provided several key lessons:

- **Early User Involvement is Crucial**: Involving end-users (patients, doctors, and administrative staff) during the planning and development phase ensured that the final system met their needs and expectations. Feedback from pilot testing was invaluable in refining the user interface and core functionalities.
- **Security Cannot be an Afterthought**: Prioritizing security during the initial development phase reduced potential vulnerabilities and simplified later maintenance. Early implementation of encryption and secure protocols minimized risks.
- **Modular Design is Beneficial**: Using a modular approach for coding—where each component (login, booking, schedule management) was developed and tested independently—allowed for easier troubleshooting and future expansion.
- **Documentation Saves Time**: Comprehensive documentation for each phase (from development to testing and deployment) helped streamline communication among team members and facilitated easier maintenance and updates.

6.4. Future Improvements

The current version of the Doctor's Appointment System serves its purpose effectively, but there are opportunities for enhancement:

6.4.1. Email and SMS Notifications

- **Appointment Reminders**: Implementing an automatic email and SMS notification system to remind patients of their scheduled appointments can reduce no-show rates and improve communication.
- **Cancellation and Rescheduling Alerts**: Notify patients and doctors instantly if an appointment is canceled or rescheduled, ensuring that everyone stays informed.

6.4.2. Enhanced User Interface

- **User Experience Redesign**: Updating the frontend with a more modern design, using responsive frameworks like Bootstrap or Vue.js, would make the system more visually appealing and accessible on mobile devices.
- **Accessibility Improvements**: Enhancing accessibility by adhering to Web Content Accessibility Guidelines (WCAG), making the system more user-friendly for people with disabilities.

6.4.3. Integration with Electronic Health Records (EHR)

- **Patient History Integration**: Linking the appointment system to Electronic Health Records would allow doctors to access patient history during booking, leading to more informed scheduling decisions.
- **Data Synchronization**: Allow data synchronization between the appointment system and EHR for seamless communication and patient management.

6.4.4. Advanced Analytics and Reporting

- **Usage Analytics**: Adding analytics tools to track appointment trends, user activity, and system performance can provide valuable insights for administrators to optimize operations.
- **Reports Generation**: Enable administrators to generate custom reports on patient appointments, doctor schedules, and system usage for better decision-making.

6.4.5. Mobile Application Development

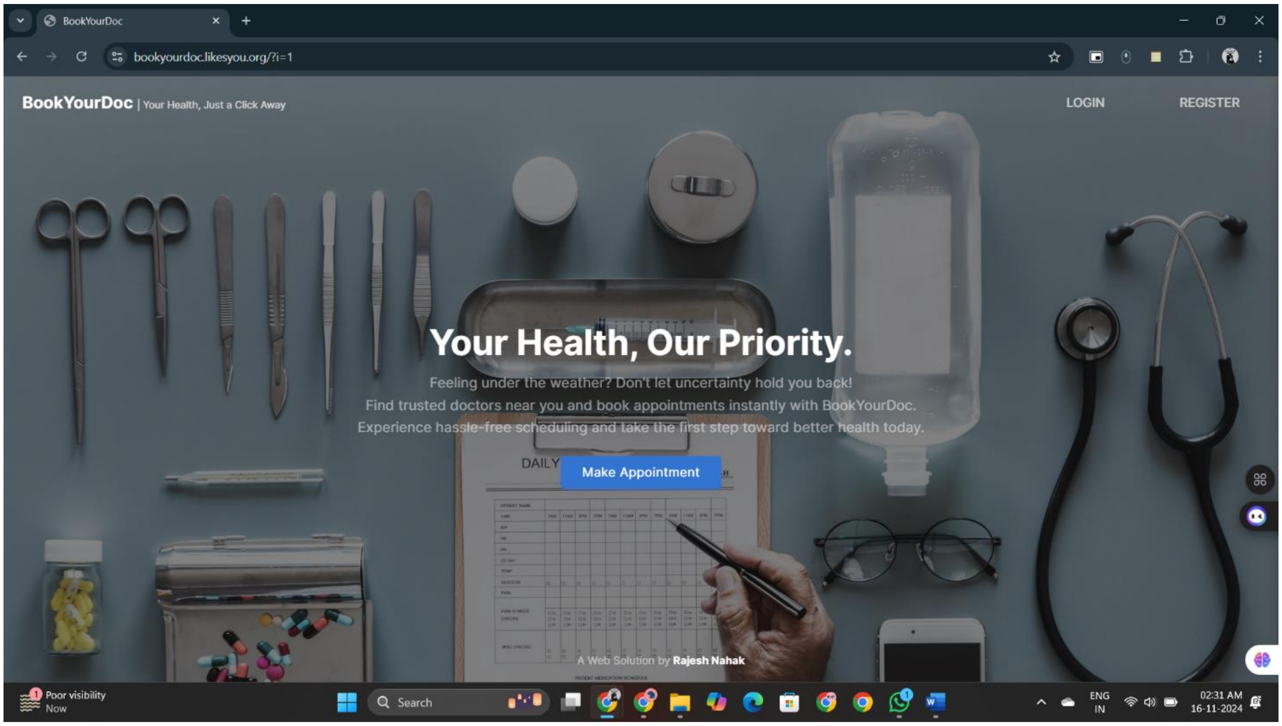
- **Native Mobile App**: Developing a dedicated mobile application for both iOS and Android platforms would provide a more convenient way for patients and doctors to access the system on the go.
- **Push Notifications**: Mobile apps can utilize push notifications for reminders and updates, enhancing user engagement and communication.

7. Source Code or System Snapshots

-> Github Link:- https://github.com/rjsnhk/BookYourDoctor.PHP

-> Live Link :- https://bookyourdoc.likesyou.org

System Snapshots with some code :-

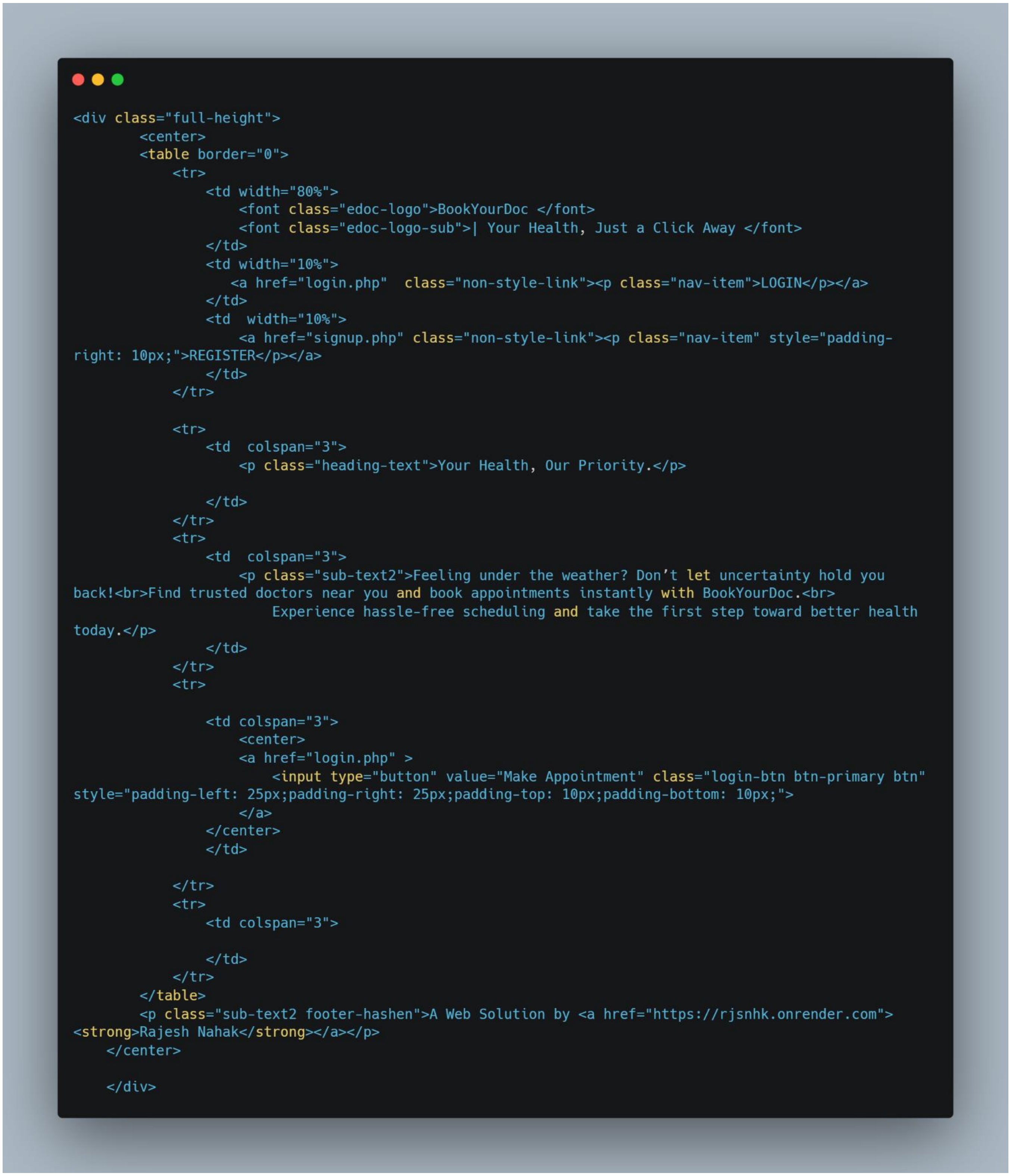


Your Health, Our Priority.

Feeling under the weather? Don't let uncertainty hold you back!
Find trusted doctors near you and book appointments instantly with BookYourDoc.
Experience hassle-free scheduling and take the first step toward better health today.

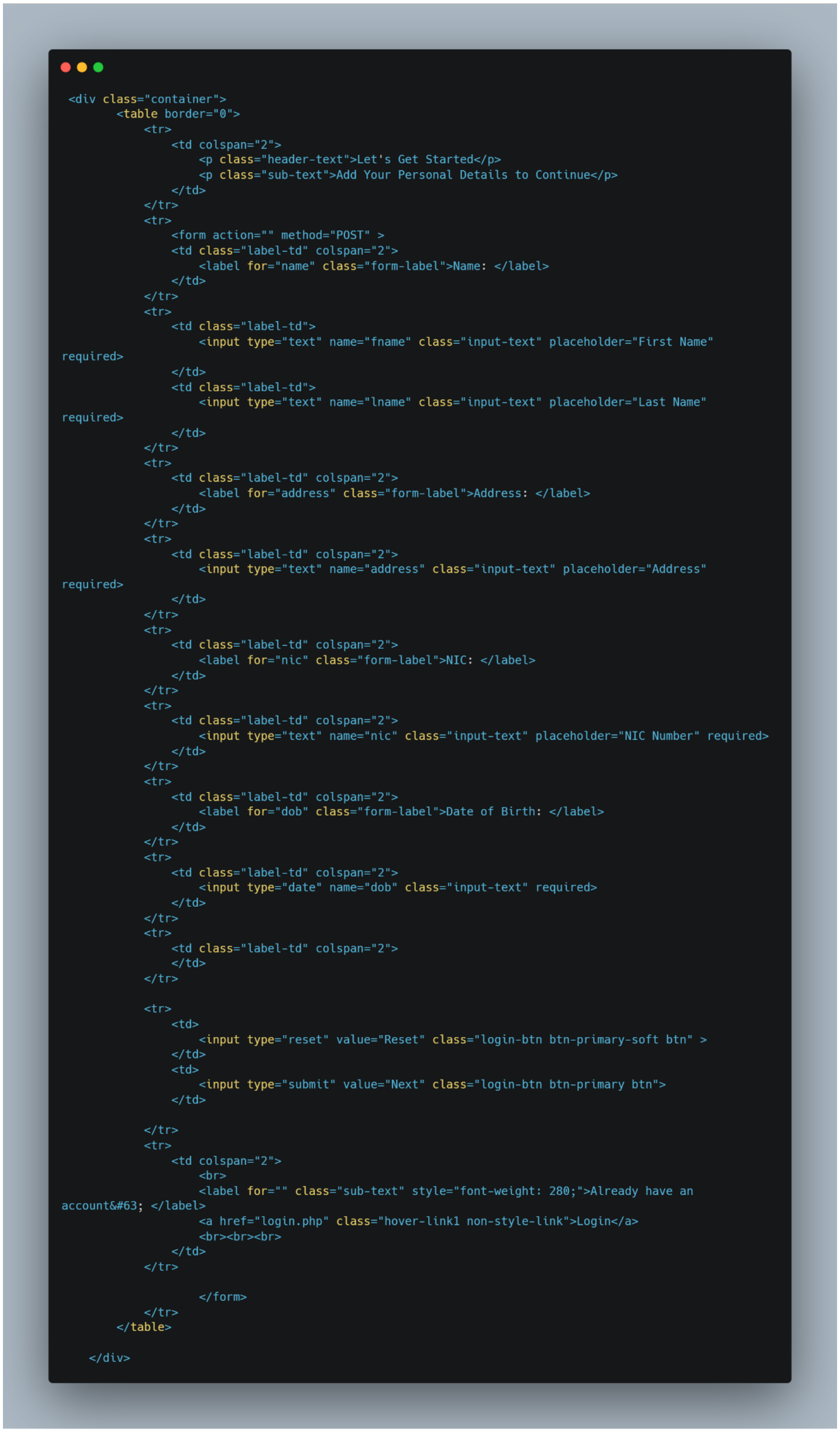
A Web Solution by **Rajesh Nahak**

Login.php-



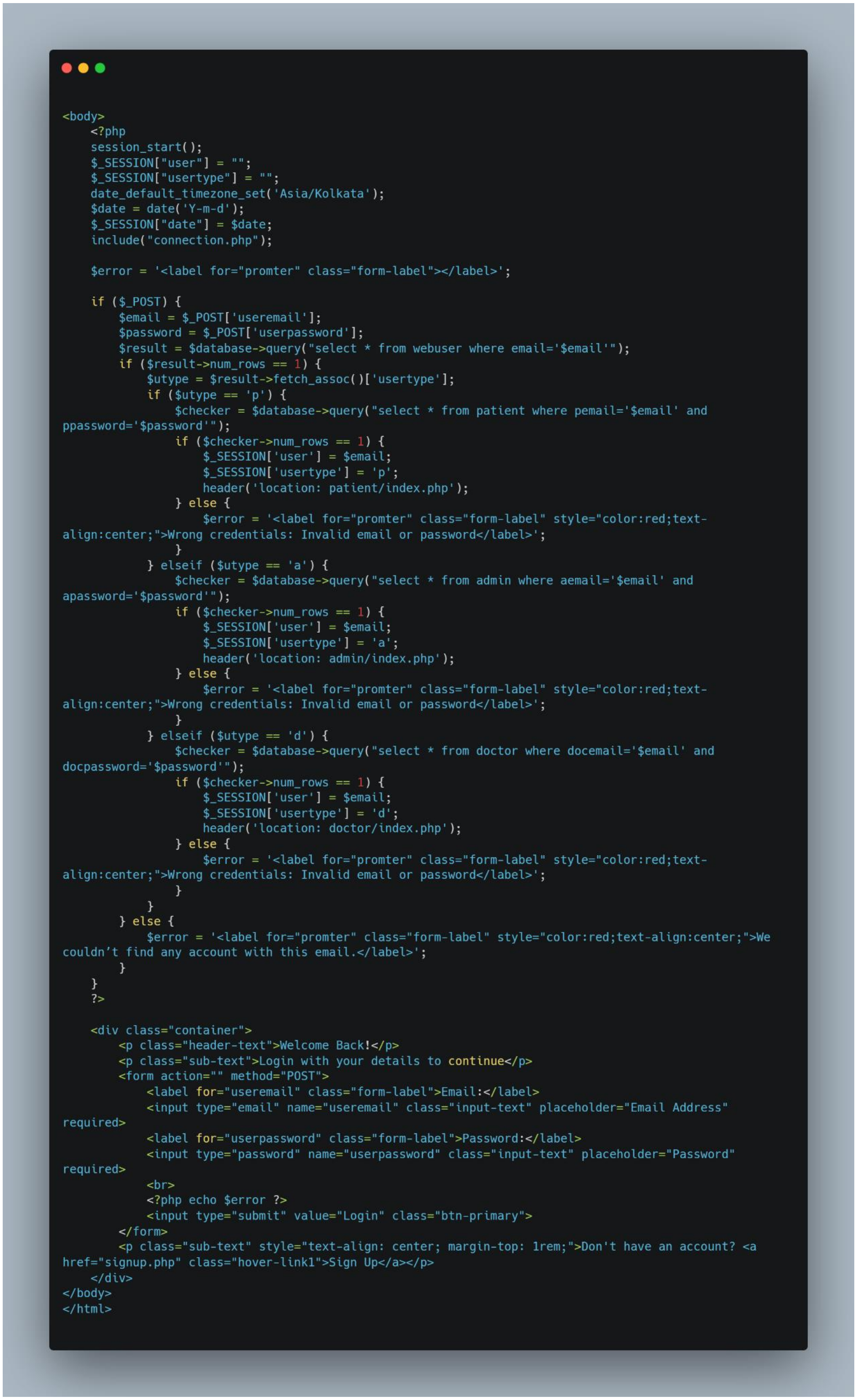
```
<div class="full-height">
        <center>
        <table border="0">
            <tr>
                <td width="80%">
                    <font class="edoc-logo">BookYourDoc </font>
                    <font class="edoc-logo-sub">| Your Health, Just a Click Away </font>
                </td>
                <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">Your Health, Our Priority.</p>

                </td>
            </tr>
            <tr>
                <td  colspan="3">
                    <p class="sub-text2">Feeling under the weather? Don’t let uncertainty hold you
back!<br>Find trusted doctors near you and book appointments instantly with BookYourDoc.<br>
                        Experience hassle-free scheduling and take the first step toward better health
today.</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">A Web Solution by <a href="https://rjsnhk.onrender.com">
<strong>Rajesh Nahak</strong></a></p>
    </center>

    </div>
```

Signup.php

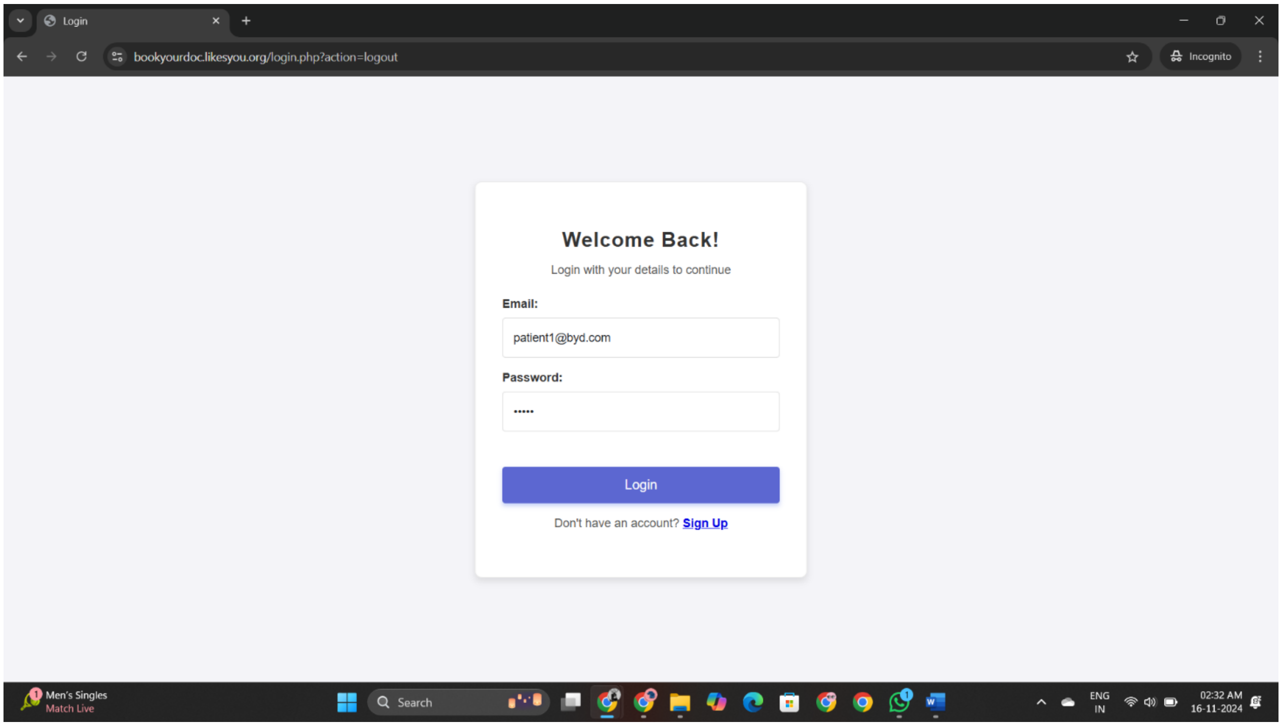


```
 <div class="container">
        <table border="0">
            <tr>
                <td colspan="2">
                    <p class="header-text">Let's Get Started</p>
                    <p class="sub-text">Add Your Personal Details to Continue</p>
                </td>
            </tr>
            <tr>
                <form action="" method="POST" >
                <td class="label-td" colspan="2">
                    <label for="name" class="form-label">Name: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td">
                    <input type="text" name="fname" class="input-text" placeholder="First Name"
required>
                </td>
                <td class="label-td">
                    <input type="text" name="lname" class="input-text" placeholder="Last Name"
required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <label for="address" class="form-label">Address: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="text" name="address" class="input-text" placeholder="Address"
required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <label for="nic" class="form-label">NIC: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="text" name="nic" class="input-text" placeholder="NIC Number" required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <label for="dob" class="form-label">Date of Birth: </label>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                    <input type="date" name="dob" class="input-text" required>
                </td>
            </tr>
            <tr>
                <td class="label-td" colspan="2">
                </td>
            </tr>

            <tr>
                <td>
                    <input type="reset" value="Reset" class="login-btn btn-primary-soft btn" >
                </td>
                <td>
                    <input type="submit" value="Next" class="login-btn btn-primary btn">
                </td>

            </tr>
            <tr>
                <td colspan="2">
                    <br>
                    <label for="" class="sub-text" style="font-weight: 280;">Already have an
account&#63; </label>
                    <a href="login.php" class="hover-link1 non-style-link">Login</a>
                    <br><br><br>
                </td>
            </tr>

                    </form>
            </tr>
        </table>

    </div>
```

index.html



```
<body>
    <?php
    session_start();
    $_SESSION["user"] = "";
    $_SESSION["usertype"] = "";
    date_default_timezone_set('Asia/Kolkata');
    $date = date('Y-m-d');
    $_SESSION["date"] = $date;
    include("connection.php");

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

    if ($_POST) {
        $email = $_POST['useremail'];
        $password = $_POST['userpassword'];
        $result = $database->query("select * from webuser where email='$email'");
        if ($result->num_rows == 1) {
            $utype = $result->fetch_assoc()['usertype'];
            if ($utype == 'p') {
                $checker = $database->query("select * from patient where pemail='$email' and
ppassword='$password'");
                if ($checker->num_rows == 1) {
                    $_SESSION['user'] = $email;
                    $_SESSION['usertype'] = 'p';
                    header('location: patient/index.php');
                } else {
                    $error = '<label for="promter" class="form-label" style="color:red;text-
align:center;">Wrong credentials: Invalid email or password</label>';
                }
            } elseif ($utype == 'a') {
                $checker = $database->query("select * from admin where aemail='$email' and
apassword='$password'");
                if ($checker->num_rows == 1) {
                    $_SESSION['user'] = $email;
                    $_SESSION['usertype'] = 'a';
                    header('location: admin/index.php');
                } else {
                    $error = '<label for="promter" class="form-label" style="color:red;text-
align:center;">Wrong credentials: Invalid email or password</label>';
                }
            } elseif ($utype == 'd') {
                $checker = $database->query("select * from doctor where docemail='$email' and
docpassword='$password'");
                if ($checker->num_rows == 1) {
                    $_SESSION['user'] = $email;
                    $_SESSION['usertype'] = 'd';
                    header('location: doctor/index.php');
                } else {
                    $error = '<label for="promter" class="form-label" style="color:red;text-
align:center;">Wrong credentials: Invalid email or password</label>';
                }
            }
        } else {
            $error = '<label for="promter" class="form-label" style="color:red;text-align:center;">We
couldn’t find any account with this email.</label>';
        }
    }
    ?>

    <div class="container">
        <p class="header-text">Welcome Back!</p>
        <p class="sub-text">Login with your details to continue</p>
        <form action="" method="POST">
            <label for="useremail" class="form-label">Email:</label>
            <input type="email" name="useremail" class="input-text" placeholder="Email Address"
required>
            <label for="userpassword" class="form-label">Password:</label>
            <input type="password" name="userpassword" class="input-text" placeholder="Password"
required>
            <br>
            <?php echo $error ?>
            <input type="submit" value="Login" class="btn-primary">
        </form>
        <p class="sub-text" style="text-align: center; margin-top: 1rem;">Don't have an account? <a
href="signup.php" class="hover-link1">Sign Up</a></p>
    </div>
</body>
</html>
```



Welcome Back!

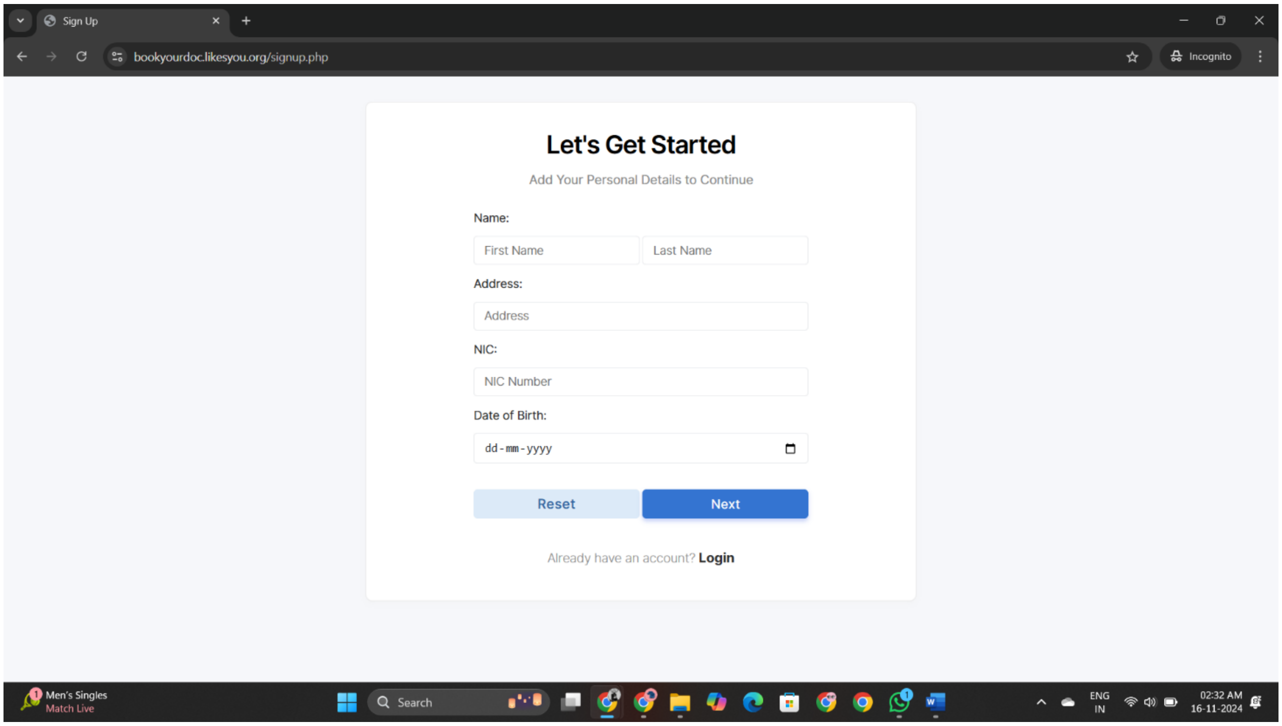
Login with your details to continue

Email: patient1@byd.com

Password: •••••

Login

Don't have an account? **Sign Up**



Let's Get Started

Add Your Personal Details to Continue

Name: First Name Last Name

Address: Address

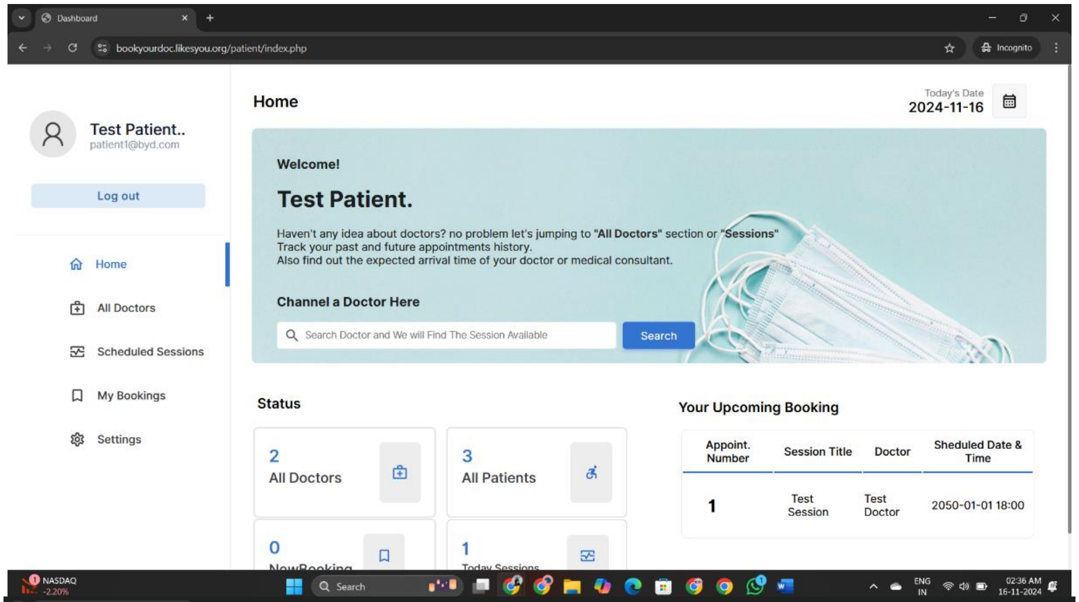
NIC: NIC Number

Date of Birth: dd-mm-yyyy

Reset Next

Already have an account? **Login**

Patient view :-


Dashboard
bookyourdoc.likesyou.org/patient/index.php
Incognito

Test Patient..
patient1@byd.com

Log out
Home
All Doctors
Scheduled Sessions
My Bookings
Settings

Home

Today's Date
2024-11-16

Welcome!

Test Patient.

Haven't any idea about doctors? no problem let's jumping to **"All Doctors"** section or **"Sessions"**
Track your past and future appointments history.
Also find out the expected arrival time of your doctor or medical consultant.

Channel a Doctor Here

Search Doctor and We will Find The Session Available
Search

Status

2 All Doctors

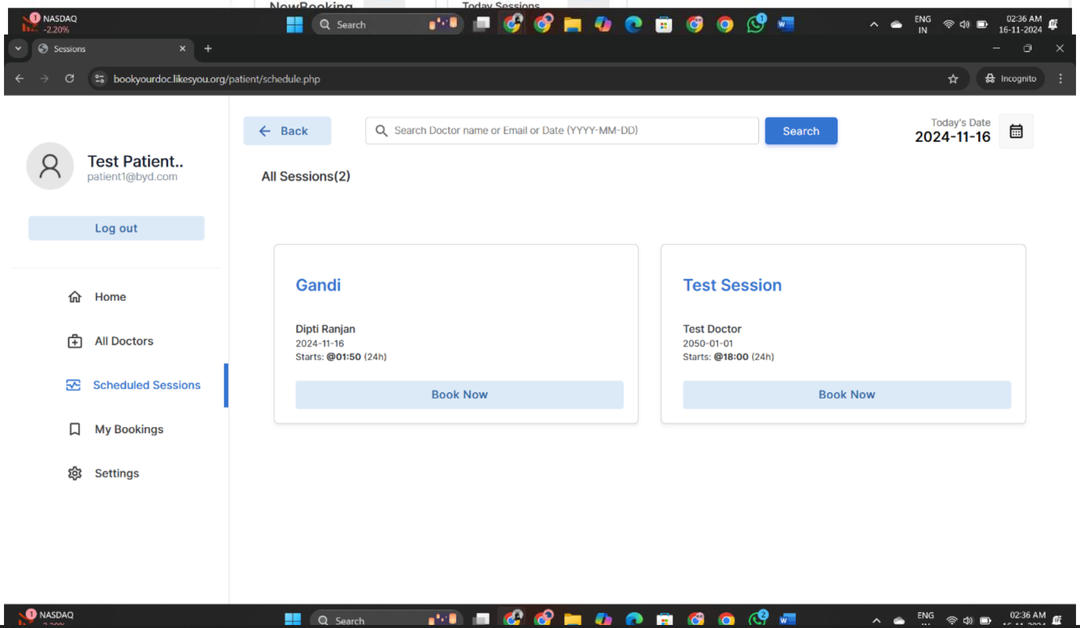
3 All Patients

0 NewBooking

1 Today Sessions

Your Upcoming Booking

Appoint. Number	Session Title	Doctor	Sheduled Date & Time
1	Test Session	Test Doctor	2050-01-01 18:00


Sessions
bookyourdoc.likesyou.org/patient/schedule.php
Incognito
Back
Search Doctor name or Email or Date (YYYY-MM-DD)
Search

Today's Date
2024-11-16

Test Patient..
patient1@byd.com

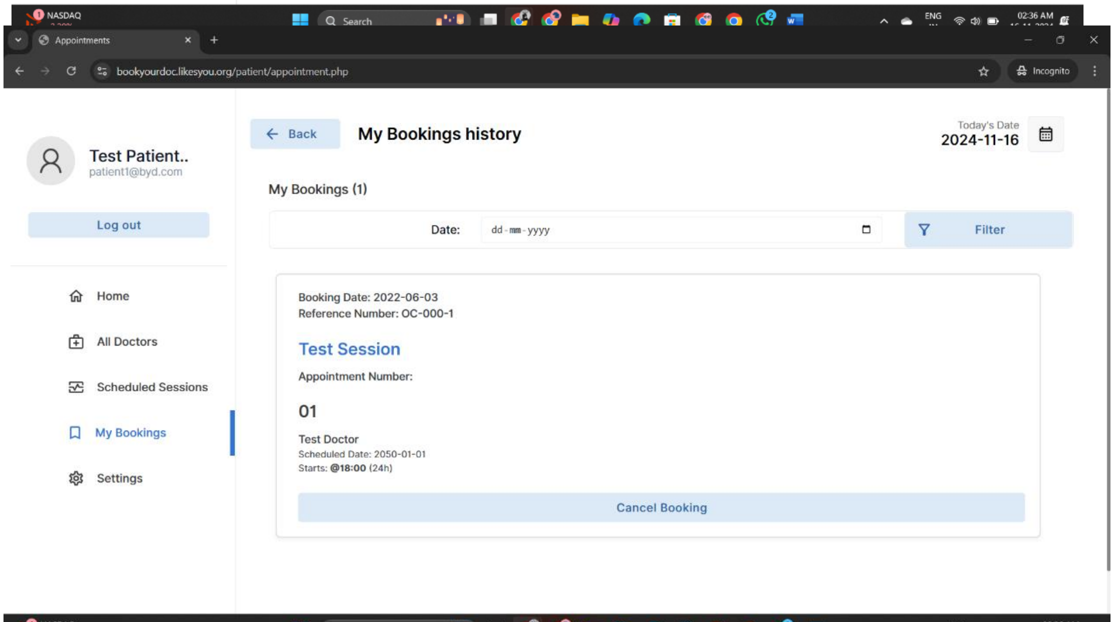
All Sessions(2)

Gandi

Dipti Ranjan
2024-11-16
Starts: **@01:50** (24h)

Test Session

Test Doctor
2050-01-01
Starts: **@18:00** (24h)



My Bookings history

Today's Date
2024-11-16

Test Patient..
patient1@byd.com

My Bookings (1)

Date: dd-mm-yyyy

Booking Date: 2022-06-03
Reference Number: OC-000-1

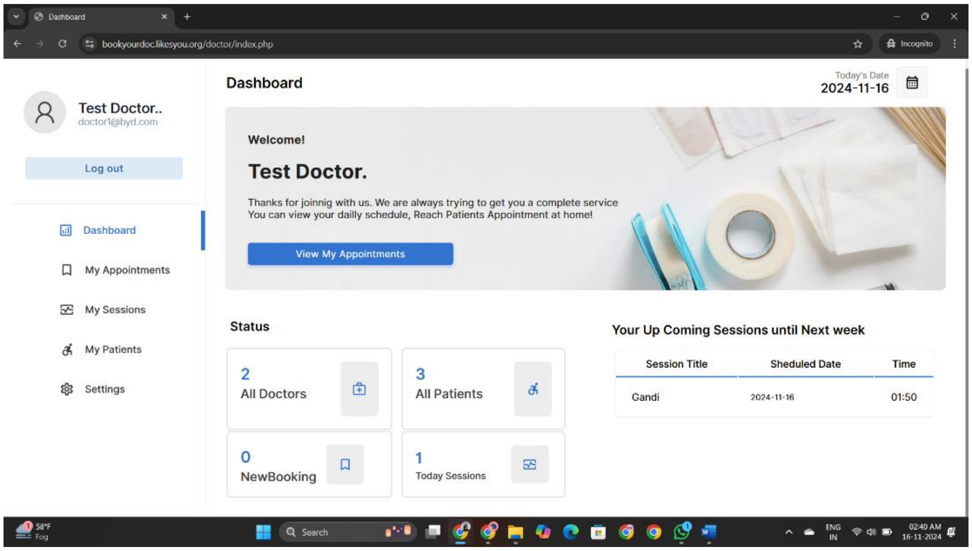
Test Session

Appointment Number:

01

Test Doctor
Scheduled Date: 2050-01-01
Starts: **@18:00** (24h)

Doctor view :-



Dashboard

Today's Date
2024-11-16

Welcome!

Test Doctor.

Thanks for joinnig with us. We are always trying to get you a complete service
You can view your dailly schedule, Reach Patients Appointment at home!

View My Appointments

Status

2 All Doctors

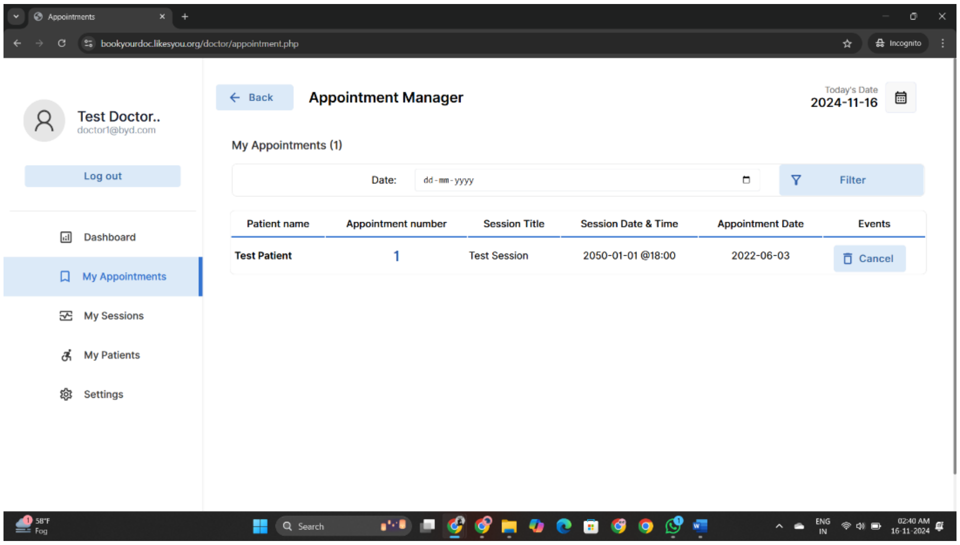
3 All Patients

0 NewBooking

1 Today Sessions

Your Up Coming Sessions until Next week

Session Title	Sheduled Date	Time
Gandi	2024-11-16	01:50



Back

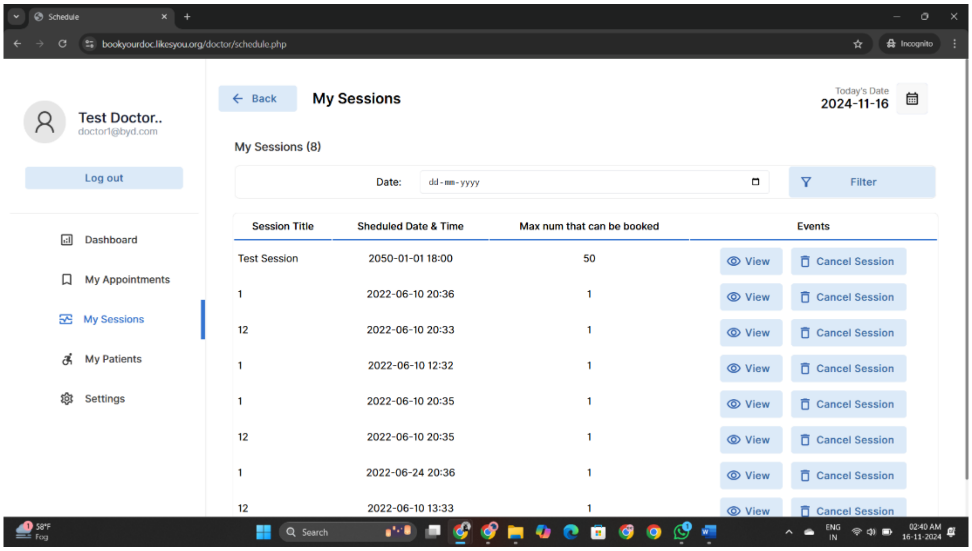
Appointment Manager

Today's Date
2024-11-16

My Appointments (1)

Date: dd-mm-yyyy Filter

Patient name	Appointment number	Session Title	Session Date & Time	Appointment Date	Events
Test Patient	1	Test Session	2050-01-01 @18:00	2022-06-03	Cancel



Back

My Sessions

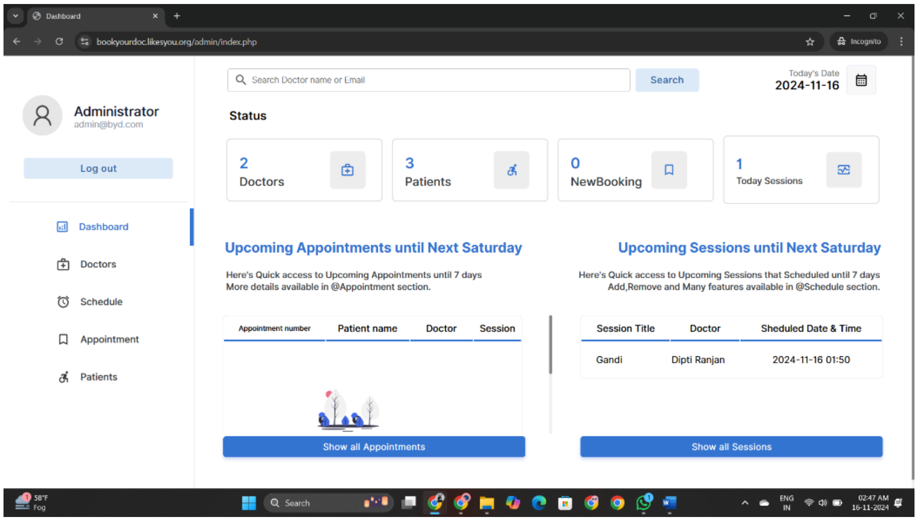
Today's Date
2024-11-16

My Sessions (8)

Date: dd-mm-yyyy Filter

Session Title	Sheduled Date & Time	Max num that can be booked	Events
Test Session	2050-01-01 18:00	50	View Cancel Session
1	2022-06-10 20:36	1	View Cancel Session
12	2022-06-10 20:33	1	View Cancel Session
1	2022-06-10 12:32	1	View Cancel Session
1	2022-06-10 20:35	1	View Cancel Session
12	2022-06-10 20:35	1	View Cancel Session
1	2022-06-24 20:36	1	View Cancel Session
12	2022-06-10 13:33	1	View Cancel Session

Admin View:-



Status

2 Doctors | 3 Patients | 0 NewBooking | 1 Today Sessions

Upcoming Appointments until Next Saturday

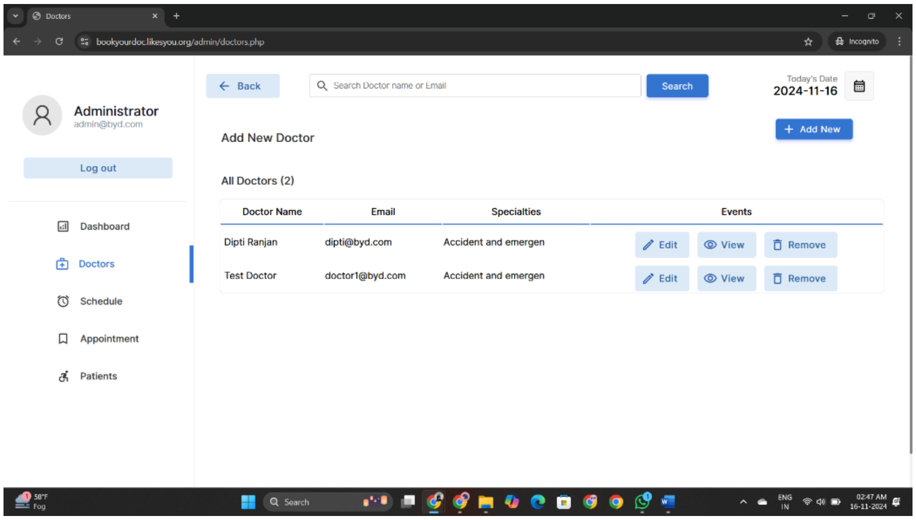
Here's Quick access to Upcoming Appointments until 7 days
More details available in @Appointment section.

Appointment number	Patient name	Doctor	Session

Upcoming Sessions until Next Saturday

Here's Quick access to Upcoming Sessions that Scheduled until 7 days
Add,Remove and Many features available in @Schedule section.

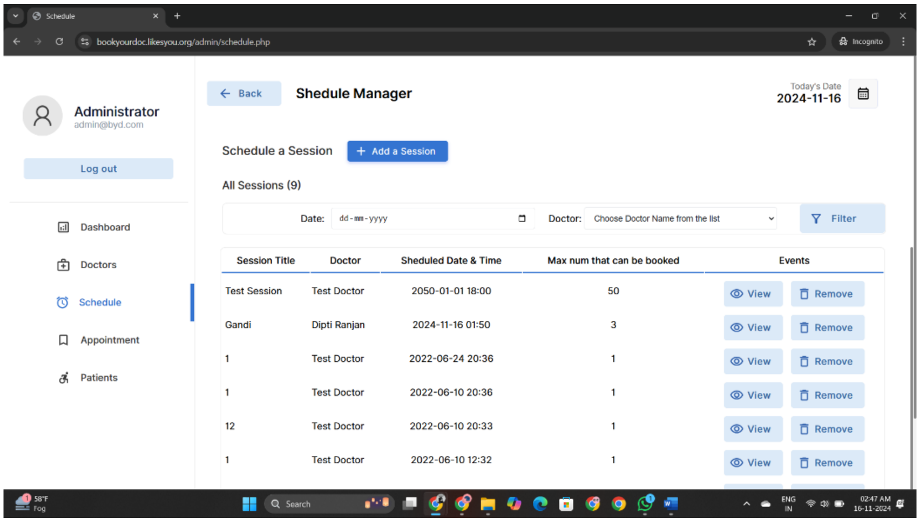
Session Title	Doctor	Sheduled Date & Time
Gandi	Dipti Ranjan	2024-11-16 01:50



Add New Doctor

All Doctors (2)

Doctor Name	Email	Specialties	Events
Dipti Ranjan	dipti@byd.com	Accident and emergen	Edit View Remove
Test Doctor	doctor1@byd.com	Accident and emergen	Edit View Remove



Shedule Manager

Schedule a Session

All Sessions (9)

Date: dd-mm-yyyy Doctor: Choose Doctor Name from the list

Session Title	Doctor	Sheduled Date & Time	Max num that can be booked	Events
Test Session	Test Doctor	2050-01-01 18:00	50	View Remove
Gandi	Dipti Ranjan	2024-11-16 01:50	3	View Remove
1	Test Doctor	2022-06-24 20:36	1	View Remove
1	Test Doctor	2022-06-10 20:36	1	View Remove
12	Test Doctor	2022-06-10 20:33	1	View Remove
1	Test Doctor	2022-06-10 12:32	1	View Remove

8.Conclusion

The Doctor Appointment System project successfully demonstrated how technology can improve the efficiency and accessibility of healthcare services. By transitioning from a manual or fragmented appointment scheduling system to a centralized digital platform, the project addressed key challenges such as long wait times, scheduling conflicts, and administrative burdens. The system enhanced user satisfaction by providing a convenient, easy-to-use interface that allows patients to book, modify, and manage their appointments online. Doctors and administrators also benefited from streamlined workflows, real-time schedule management, and secure access to patient information.

Through the development process, this project highlighted the importance of user-centered design, robust security protocols, and flexibility to accommodate various healthcare environments. Challenges such as user adaptation, data security, and system scalability were effectively tackled, leading to a resilient and scalable solution.

Looking forward, the Doctor Appointment System has the potential to expand its capabilities, incorporating features like email and SMS notifications, improved analytics, and EHR integration to further enhance its value. The lessons learned from this project provide a solid foundation for future enhancements, aiming to create a more comprehensive and sophisticated healthcare management platform.

Overall, the project not only met its objectives but also demonstrated the impact of digital solutions in optimizing healthcare operations and improving patient care. The system's success serves as a testament to how well-designed software can bridge gaps in traditional processes, fostering a more efficient and patient-friendly healthcare environment.