

BULACAN STATE UNIVERSITY
College of Engineering
Malolos City



Republic of the Philippines
Bulacan State University
College of Engineering
Computer Engineering Department

**Integrated Healthcare Management System: Enhance Monitoring, Analysis, and
Implementation for Improved Wellness**

CPE 209/209L – BSCPE 2A
Database Management System
1730-1930 T

Submitted by:
Mendoza, Ron Cristian T.
Reyes, John Clarence V.
Sta. Ana, Jan Jonas G.
San Juan, Jimuel A.

Submitted to:
Engr. Bernard G. Yasay

Date Submitted: 05/23/2024

I. Introduction

The ability to continuously track and monitor health data is essential for maintaining optimal patient outcomes, particularly for individuals managing certain conditions. However, some existing systems are fragmented and lack seamless integration of crucial health information, leading to delays in care, inefficient treatment plans, and suboptimal patient experiences.

The proposed health monitoring system addresses these challenges by providing an advanced, integrated platform for continuous health tracking. This system collects and manages comprehensive health data, including patient information, health history, appointments, and various health metrics. By combining data from different sources, healthcare providers can gain a more holistic view of each patient's health status.

Key features of the system include personalized health metric tracking, real-time updates on patient health, and automated alerts for healthcare providers when potential issues arise. The system also facilitates streamlined communication between patients and providers, allowing for efficient scheduling and consultation.

Implementing this system can enhance patient outcomes, improve care coordination, and increase patient engagement in health management. This innovative approach to health monitoring has the potential to revolutionize patient care and set new standards for healthcare efficiency.

The following sections will outline the proposed health monitoring system's objectives, procedures, and anticipated outcomes.

Background of the Project

The project aims to address the existing challenges in the healthcare field by developing a comprehensive system. The proposed system is designed to enhance data management and analysis within healthcare, emphasizing the importance of efficient and secure data handling. This encompasses the types of data collected and the associated challenges of managing large volumes of health-related information.

By integrating programming and databases into the system, the project will optimize data storage, processing, and accessibility. This comprehensive approach can improve healthcare delivery and patient outcomes by facilitating more effective data utilization. The project will also focus on creating user-friendly interfaces for medical professionals, allowing them to access and interact with data more efficiently.

The proposed system aims to create a more connected and efficient healthcare ecosystem by fostering collaboration among healthcare professionals, researchers, and patients.

Objectives

- To provide medical personnel the capacity to systematically document the health profiles of each of their patients.
- To assist medical personnel in enhancing their ability in providing comprehensive medical care.
- To aid medical personnel in mitigating the risk of prescribing incorrect medication to their patients.

Scope and Limitations of the Study

This software is focused on optimizing data storing, processing, and accessing by creating multiple tables that can be easily navigated with the use of database management. By doing so, they will be able to find, track, and make sure that the patients are going through the right path such as proper time-to-time check-ups, taking maintenance, and to make sure that they are always updated to what their situation is. This software will also help and guide the doctors to the current situation of their patients, they can track their patients through the data that they gathered.

However, this study has several limitations. Firstly, it will not address the integration of the software with existing electronic health record systems or other healthcare databases. Secondly, the scalability of the software for use in large healthcare settings will not be evaluated. Lastly, the study will not consider the regulatory requirements or legal implications of implementing such a system. The scope is limited to the development and testing of a prototype in a controlled healthcare environment.

II. Project Details

Functionality

PHP Session Handling

- This section connects all the pages, enabling seamless navigation between them. Each PHP file includes its own session handling to manage the registration and login processes. If users enter incorrect credentials, the session acts as a barrier, preventing them from proceeding until the correct credentials are provided.

Navbar

- The navigation bar contains an image and placeholders for additional links. It includes buttons necessary for navigating all the pages.

Login Container

- This container includes buttons for choosing between patient and admin login options. It is part of the navbar and links to pages specific to patients and admins.

Register Container

- This container includes a button that allows users to create a new account. It is also part of the navbar.

Background and Main Text Sections

- This section features an image and text describing the healthcare services, serving as the welcome page that users first see when they visit the website.

Modal Dialogs

- This section includes the Patient Modal, Patient Registration Modal, Patient Login Modal, and Admin Modal. These modals allow users to navigate to their respective pages, edit their profiles, and schedule appointments once they log in.

JavaScript for Modal Handling

- This section contains the functions that manage the display and hiding of modals. Each modal has specific functions that enable users to interact with them.

Database Connection

- A new MySQLi connection object is created to connect to the MySQL database using the provided credentials (localhost, root, root, dc1, 3308). If the connection fails, the script terminates and outputs an error message.

Sanitize Input

- This function sanitizes user input to prevent security vulnerabilities like XSS (Cross-Site Scripting) and ensures clean input data.

Handling Form Submission

- The script checks if the request method is POST, indicating that the form has been submitted. If the request method is POST, the script processes the form entries.

Fetching User's Name

- Retrieves the user's name from the database using the email stored in the session. This function ensures that the credentials are correct and exist in the database, allowing the session to continue and the user to proceed to the next page.

Sign-Out

- Checks if the sign-out parameter is set in the URL. If so, it calls the signOut function, which signs out the user and redirects them to the home page.

Billing Record Check

- Checks if a billing record already exists for the patient, allowing the admin to manage the billing records by changing the billing status.

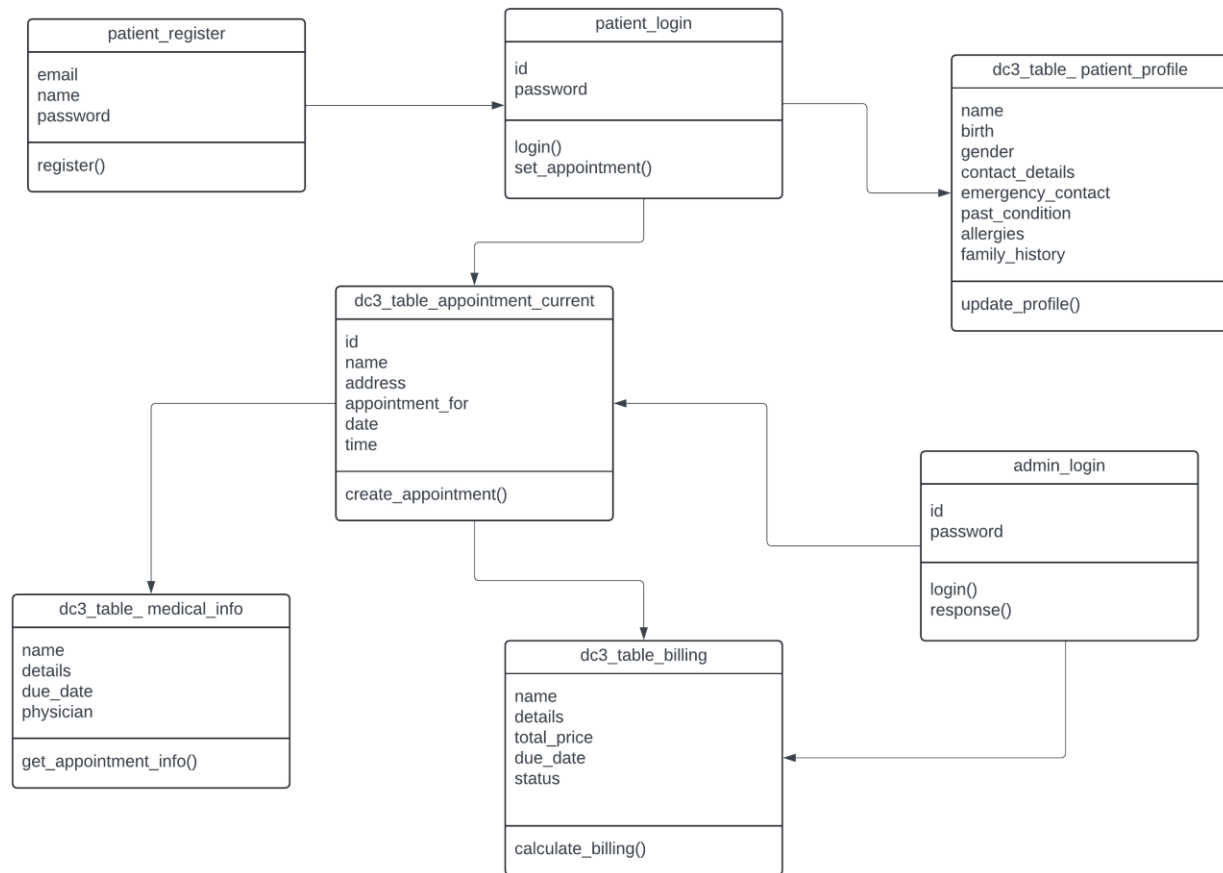
Insert Billing Record

- Inserts the new billing record if it does not already exist and provides feedback to the user.

HTML Structure and CSS

- The HTML structure and CSS handle the design and layout of each page. They manage the functions, sessions, databases, and PHP code to ensure the website operates correctly.

Flowchart/Algorithm



Codes

```

<?php
session_start(); // Start the session if not already started

// Check if the user is already logged in as a patient
if(isset($_SESSION['email'])) {
    // Redirect to the patient page
    header("Location: patient_page.php");
    exit();
}

// Check if the user is already logged in as an admin
elseif(isset($_SESSION['id'])) {
    // Redirect to the admin page
    header("Location: admin_page.php");
    exit(); // Exit to prevent further execution of code
}

// Check if there's a registration or login success message stored in the session
if (isset($_SESSION['success_message'])) {
    $message = $_SESSION['success_message'];
    unset($_SESSION['success_message']); // Remove the message from the session
}

// Check if there's an error message stored in the session
if (isset($_SESSION['error_message'])) {

```

```

$error_message = $_SESSION['error_message'];
unset($_SESSION['error_message']); // Remove the message from the session
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>
  <link rel="stylesheet" href="home_page_stylesss.css">
</head>
<body>
<div id="floating-message" <?php if(isset($message)) echo 'style="display: block;"; ?><?php if(isset($message)) echo
$message; ?></div>

<div id="error-message" <?php if(isset($error_message)) echo 'style="display: block;"; ?><?php if(isset($error_message))
echo $error_message; ?></div>
  <nav class="navbar">
    <ul>
      <li></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a href="#"></a></li>
      <li><a></a></li>
      <li><a></a></li>
      <li><a></a></li>
      <li><a></a></li>
    </ul>
  </nav>
  <div class="login-container">
    <div class="role-buttons">
      <button class="role-btn role-transform" onclick="showPatientOptions()">Patient</button>
      <button class="role-btn role-transform" onclick="showAdminOptions()">Admin</button>
    </div>
    <!-- Trust image -->
  </div>
  <div class="background">
    
    <!-- All between the box -->
    <div id="field">
      <div id="maintext">
        <div id="maintext-first">
          <span id="ms0img">
            
          </span>
          <h6>Health and medical<br>theme</h6>
          <p>Exploring the intricate interplay<br>between biology, technology,<br>and human resilience, the<br>health and
medical theme delves<br>into the frontline advancements<br>shaping the future of well-being</p>
        </div>
        <div id="maintext-second">
          <span id="ms1img">
            
          </span>
          <h6>World Class Doctors</h6>
          <p>"World-class doctors possess<br>unparalleled expertise, empathy,<br>and dedication,
revolutionizing<br>healthcare"</p>
        </div>
        <div id="ms2under">
          <span id="ms2img">
            
          </span>
          <h6>Caring Doctors</h6>
          <p>We will treat you with<br>utmost care that no other<br>hospital will</p>
        </div>
      </div>
      <div id="maintext-third">
        <span id="ms3img">
          
        </span>
        <h6>Medical Counseling</h6>
        <p>We will help you diagnose your<br>sickness with capable doctors</p>
        <div id="ms3under">
          <span id="ms4img">
            
          </span>
        </div>
      </div>
    </div>
  </div>

```

```

        </span>
        <h6>Emergency Services</h6>
        <p>Accomodate bad conditions<br>with fast to act staffs</p>
    </div>
</div>
</div>
</div>
<!-- Patient Login Modal -->
<div id="patientModal" class="modal">
    <div class="modal-content">
        <span class="close" onclick="closeModal()">&times;</span>
        <h2>Login as Patient</h2>
        <button class="login-btn" onclick="showPatientLogin()">Login</button>
        <button class="register-btn" onclick="showPatientRegistration()">Register</button>
    </div>
</div>
<!-- Patient register Modal -->
<div id="patientReg" class="modal">
    <div class="reg-content">
        <h2>Registration</h2>
        <form name = "register" method="post" action="patient_register.php">
            <label for="name">Name</label>
            <input type="text" id="name" name="name" required><br><br>
            <label for="email">Email</label>
            <input type="email" id="email" name="email" required><br><br>
            <label for="password">Password</label>
            <input type="password" id="password" name="password" required><br><br>
            <button class="register-btn" onclick="registerPatient()">Register</button>
        </form>
    </div>
</div>
<!-- Patient Login Modal -->
<div id="patientLog" class="modal">
    <div class="login-content">
        <h2>Login</h2>
        <form name="login" method="post" action="patient_login.php">
            <label for="email">Email</label>
            <input type="email" id="email" name="email" required><br><br>
            <label for="password">Password</label>
            <input type="password" id="password" name="password" required><br><br>
            <button class="login-btn" onclick="loginPatient()">Login</button>
        </form>
    </div>
</div>
<!-- Admin Login Modal -->
<div id="adminModal" class="modal">
    <div class="modal-content">
        <span class="close" onclick="closeModal()">&times;</span>
        <h2>Login as Admin</h2>
        <button class="login-btn" onclick="showAdminLogin()">Login</button>
    </div>
</div>
<!-- Admin Login Modal -->
<div id="adminLog" class="modal">
    <div class="login-content">
        <h2>Admin Login</h2>
        <form name="adminLogin" method="post" action="admin_login.php">
            <label for="adminID">ID Number</label>
            <input type="text" id="adminID" name="adminID" required><br><br>
            <label for="adminPassword">Password</label>
            <input type="password" id="adminPassword" name="adminPassword" required><br><br>
            <button class="login-btn" onclick="loginAdmin()">Login</button>
        </form>
    </div>
</div>
<!-- baba ng medical services-->
<script>
function showPatientOptions() {
    document.getElementById("patientModal").style.display = "block";
}

function showPatientRegistration() {
    document.getElementById("patientReg").style.display = "block";
}

```

```

}

function showPatientLogin() {
    document.getElementById("patientLog").style.display = "block";
}

function showAdminOptions() {
    document.getElementById("adminModal").style.display = "block";
}

function showAdminLogin() {
    document.getElementById("adminLog").style.display = "block";
}

function closeModal() {
    document.getElementById("patientModal").style.display = "none";
    document.getElementById("adminModal").style.display = "none";
    document.getElementById("patientReg").style.display = "none";
    document.getElementById("patientLog").style.display = "none";
    document.getElementById("adminLog").style.display = "none";
    document.getElementById("name").value = "";
    document.getElementById("age").value = "";
    document.getElementById("sex").value = "";
    document.getElementById("phone").value = "";
    document.getElementById("email").value = "";
    document.getElementById("password").value = "";
}

window.onclick = function(event) {
    if (event.target == document.getElementById("patientModal") ||
        event.target == document.getElementById("adminModal") ||
        event.target == document.getElementById("adminLog") ||
        event.target == document.getElementById("patientReg") ||
        event.target == document.getElementById("patientLog")) {
        closeModal();
    }
}

document.addEventListener('DOMContentLoaded', function() {
    var floatingMessage = document.getElementById('floating-message');
    var errorMessage = document.getElementById('error-message');
    // Check if the message elements are present
    if (floatingMessage && errorMessage) {
        // Check if the messages are initially shown
        if (floatingMessage.style.display === 'block' || errorMessage.style.display === 'block') {
            // Set a timeout to hide the messages after 3 seconds
            setTimeout(function() {
                floatingMessage.style.display = 'none';
                errorMessage.style.display = 'none';
            }, 3000);
        }
    }
});
</script>
</body>
</html>
<?php
// Start the session
session_start();
// Establish a database connection
$conn = new mysqli('localhost', 'root', 'root', 'dc1', 3308);
// Check the connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Validate the session
if (!isset($_SESSION['email'])) {
    header("Location: login.php"); // Redirect to login page
    exit;
}
// Fetch user's name from the database
$stmt = $conn->prepare("SELECT name FROM dc1_table WHERE email = ?");
$stmt->bind_param("s", $_SESSION['email']);
$stmt->execute();
$result = $stmt->get_result();

```



```

if ($result && $result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $user_name = $row['name'];
} else {
    $user_name = "User"; // Default to "User" if the name couldn't be fetched
}

// Handle sign-out logic
if (isset($_GET['signout'])) {
    signOut();
}

// Handle form submission
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $user_name; // Use the session user's name directly
    $address = $_POST['address'];
    $appointment_for = $_POST['appointment_for'];
    $date = $_POST['date'];
    $time = $_POST['time'];

    // Check if the user already has an appointment
    $stmt = $conn->prepare("SELECT COUNT(*) as num_appointments FROM dc3_table_appointment_current WHERE name = ?");
    $stmt->bind_param("s", $name);
    $stmt->execute();
    $result = $stmt->get_result();
    $row = $result->fetch_assoc();
    $num_appointments = $row['num_appointments'];
    if ($num_appointments < 1) {
        // Generate a random ID for the appointment
        $random_id = rand(1, 1000);
        // Check if the random ID already exists in the database
        do {
            $stmt = $conn->prepare("SELECT COUNT(*) as num_appointments FROM dc3_table_appointment_current WHERE id = ?");
            $stmt->bind_param("i", $random_id);
            $stmt->execute();
            $result = $stmt->get_result();
            $row = $result->fetch_assoc();
            $num_appointments = $row['num_appointments'];
            if ($num_appointments > 0) {
                $random_id = rand(1, 1000);
            }
        } while ($num_appointments > 0);
        // Insert appointment data into the database
        $stmt = $conn->prepare("INSERT INTO dc3_table_appointment_current (id, name, address, appointment_for, date, time)
VALUES (?, ?, ?, ?, ?, ?)");
        $stmt->bind_param("isssss", $random_id, $name, $address, $appointment_for, $date, $time);
        if ($stmt->execute()) {
            $_SESSION['booking_message'] = "Booked successfully!";
            header("Location: " . $_SERVER['PHP_SELF']);
            exit;
        } else {
            $message = "Error: " . $stmt->error;
        }
    } else {
        $message = "You have reached the maximum number of appointments.";
    }
}

// Check for booking message session variable
if (isset($_SESSION['booking_message'])) {
    $message = $_SESSION['booking_message'];
    unset($_SESSION['booking_message']);
}

// Close the database connection
$conn->close();

// Sign-out function
function signOut() {
    // Unset all session variables
    $_SESSION = array();

    // Destroy the session
    session_destroy();

    // Redirect to the login page or any other page after signing out
    header("Location: home_page.php"); // Change "login.php" to your desired page
    exit;
}

```

```

?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Patient Page</title>
  <link rel="stylesheet" href="patient_page_styles.css">
</head>
<body>

<nav class="navbar">
  <ul>
    <li></li>
    <li><a href="?signout=1">Sign Out</a></li>
  </ul>
</nav>
<div class="background">
<div class="welcome-message">
  <?php echo htmlspecialchars($user_name); ?>
</div>
<div id="background">
  <a href="patient_billing.php" class="image-link"></a>
  <a href="patient_info.php" class="image-link"></a>
  <div class="appointment-form">
    <h2>Book an Appointment</h2>
    <form action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']); ?>" method="POST">
      <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" value="<?php echo htmlspecialchars($user_name); ?>" readonly>
      </div>
      <div class="form-group">
        <label for="address">Address:</label>
        <input type="text" id="address" name="address" required>
      </div>
      <div class="form-group">
        <label for="appointment_for">Appointment for:</label>
        <textarea id="appointment_for" name="appointment_for" rows="4" required></textarea>
      </div>
      <div class="form-group">
        <label for="date">Date:</label>
        <input type="date" id="date" name="date" required>
      </div>
      <div class="form-group">
        <label for="time">Time:</label>
        <input type="time" id="time" name="time" required>
      </div>
      <button type="submit">Submit</button>
    </form>
  </div>
</div>
<div id="floating-message"></div>
<script>
document.addEventListener("DOMContentLoaded", function() {
  const floatingMessage = document.getElementById("floating-message");

  <?php if (!empty($message)) : ?>
    floatingMessage.textContent = "<?php echo htmlspecialchars($message); ?>";
    floatingMessage.style.display = "block";
    setTimeout(() => {
      floatingMessage.style.display = "none";
    }, 3000);
  <?php endif; ?>
});
</script>
</body>
</html>
<?php
  // Start session
  session_start();

  // Handle sign-out logic
  if (isset($_GET['signout'])) {
    // Unset all session variables

```

```

$_SESSION = array();
// Destroy the session
session_destroy();

// Redirect to the login page or any other page after signing out
header("Location: home_page.php"); // Change "home_page.php" to your desired page
exit;
}
// Database connection
$conn = new mysqli('localhost', 'root', 'root', 'dc1', 3308);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Initialize the message variable
$message = "";
// Check if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Escape user inputs for security
    $name = $conn->real_escape_string($_POST['patient']);
    $detail = $conn->real_escape_string($_POST['detail']);
    $total_price = $conn->real_escape_string($_POST['total_price']);
    $due_date = $conn->real_escape_string($_POST['due_date']);
    $status = $conn->real_escape_string($_POST['status']);
    // Check if a billing record already exists for the given patient name
    $sql_check = "SELECT * FROM dc3_table_billing WHERE name = '$name'";
    $result_check = $conn->query($sql_check);

    if ($result_check->num_rows > 0) {
        $message = "Billing record already exists for $name";
    } else {
        // Prepare an insert statement
        $sql = "INSERT INTO dc3_table_billing (name, details, total_price, due_date, status) VALUES ('$name', '$detail', '$total_price', '$due_date', '$status')";
        if ($conn->query($sql) === TRUE) {
            $message = "Saved successfully!";
        } else {
            $message = "Error: " . $sql . "<br>" . $conn->error;
        }
    }
}
// Close database connection
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Page</title>
    <link rel="stylesheet" href="admin_styles.css">
</head>
<body>
<nav class="navbar">
    <ul>
        <li></li>
        <li><a href="?signout=1">Sign Out</a></li> <!-- Added sign-out button -->
    </ul>
</nav>
<div class="background">
    <div class="image-container">
        <a href="admin_appointment.php"></a>
        <a href="admin_patient_info.php" id="tago"></a>
        <a href="admin_patient_billing.php"></a>
    </div>
    <!-- Display patient list in a scrollable table -->
    <div class="scrollable-table">
        <table>
            <thead>
                <tr>
                    <th colspan="3" class="patients-label">Patients List</th>
                </tr>
            </thead>
            <tbody>

```

```

<?php
// Database connection
$conn = new mysqli('localhost', 'root', 'root', 'dc1', 3308);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// Fetch patient data from the database
$sql = "SELECT name FROM dc1_table";
$result = $conn->query($sql);

// Check if there are any patients in the result set
if ($result->num_rows > 0) {
    // Output data of each row
    while ($row = $result->fetch_assoc()) {
        echo "<tr>";
        echo "<td>" . $row["name"] . "</td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='3'>No patients found</td></tr>";
}
// Close database connection
$conn->close();
?>
</tbody>
</table>
</div>
</div>
<!-- Add/Edit Billing Section -->
<div class="billing-section">
    <h2>Billing Information</h2>
    <form method="post">
        <div class="billing-details">

            <label for="detail">Name:</label>
            <input type="text" id="patient" name="patient" placeholder="patient" required>

            <label for="detail">Detail:</label>
            <input type="text" id="detail" name="detail" placeholder="Billing Detail" required>

            <label for="total_price">Total Price:</label>
            <input type="number" id="total_price" name="total_price" placeholder="Total Price" required>

            <label for="due_date">Due Date:</label>
            <input type="date" id="due_date" name="due_date" required>

            <label for="status">Status:</label>
            <select id="status" name="status" required>
                <option value="Pending">Pending</option>
            </select>
        </div>
        <button id="save_button" type="submit">Save</button>
    </form>
    <?php if (!empty($message)) : ?>
        <div class="success-message"><?php echo $message; ?></div>
    <?php endif; ?>
</div>
<script>
    document.addEventListener("DOMContentLoaded", function() {
        const rows = document.querySelectorAll(".scrollable-table tbody tr");

        rows.forEach(row => {
            row.addEventListener("click", function() {
                const name = this.querySelector("td:first-child").innerText;
                // Populate patient name in the billing section
                document.getElementById("patient").value = name;
                fetchBillingDetails(name);
            });
        });

        function fetchBillingDetails(name) {
            // AJAX call to fetch billing details for the selected name
            // Replace 'fetch_billing_details.php' with your PHP script to fetch billing details

```

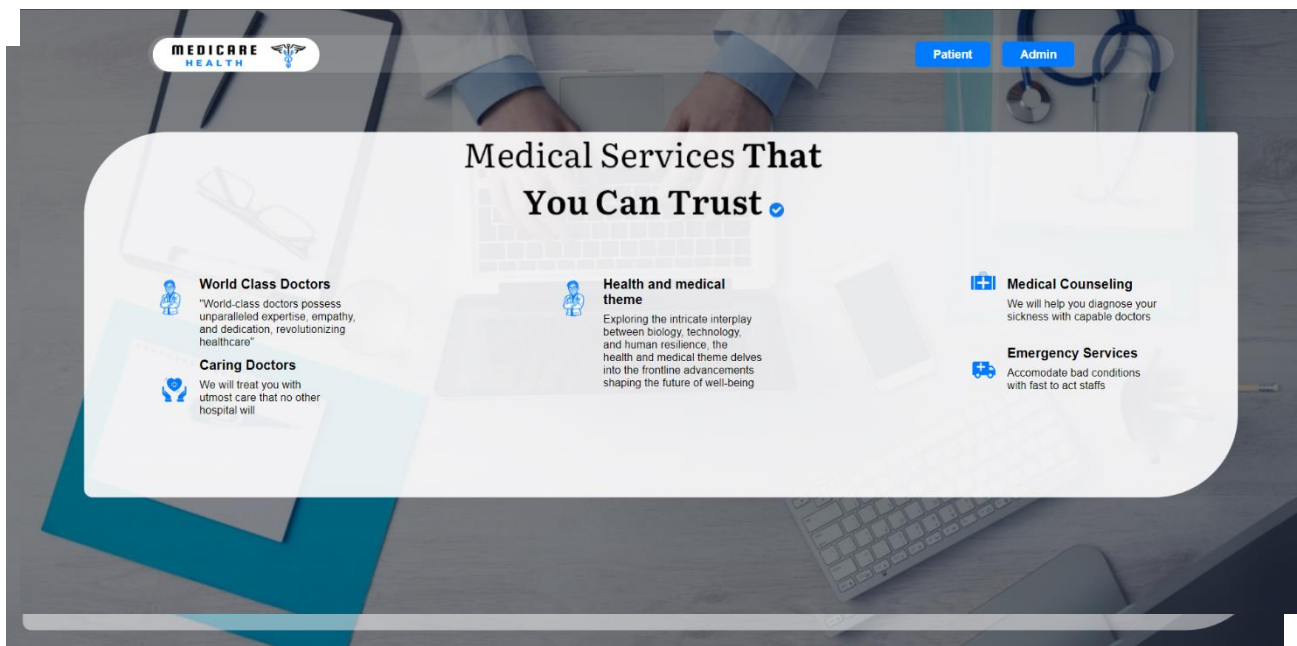
```

    fetch(`fetch_billing_details.php?name=${name}`)
    .then(response => response.json())
    .then(data => {
        // Populate billing section with fetched data
        document.getElementById("detail").value = data.details;
        document.getElementById("total_price").value = data.total_price;
        document.getElementById("due_date").value = data.due_date;
        document.getElementById("status").value = data.status;
    })
    .catch(error => console.error("Error fetching billing details:", error));
}
});
</script>
</body>
</html>

```

Testing

Trying to open and test the home page for errors and misaligned texts and designs.



Testing the registration form to see if the user's credentials are can be saved to the database.

The screenshot shows a web application for Medicare Health. At the top, there is a logo for "MEDICARE HEALTH" and two buttons labeled "Patient" and "Admin". The main content area features a central "Registration" form. The form has three input fields: "Name" with the value "testrun", "Email" with the value "testrun@gmail.com", and "Password" with the value "*****". To the left of the form, there are two sections: "World Class Doctors" with a quote about unparalleled expertise, and "Caring Doctors" with a quote about utmost care. To the right of the form, there are two sections: "Medical Counseling" with a quote about diagnosing sickness, and "Emergency Services" with a quote about accommodating bad conditions. The background of the page is a dark, blurred image of a medical professional's hands and a stethoscope.

Testing the login form to see if the user's credentials are can be called from the database.

The screenshot shows the same web application for Medicare Health. The main content area features a central "Login" form. The form has two input fields: "Email" with the value "testrun@gmail.com" and "Password" with the value "*****". Below the password field is a "Login" button. The background of the page is a dark, blurred image of a medical professional's hands and a stethoscope. The "Patient" and "Admin" buttons are visible at the top right.

Trying to book an appointment. Testing if the booked appointment can be seen in admin's page.

The screenshot shows the Medicare Health app interface. At the top left is the 'MEDICARE HEALTH' logo. At the top right is a 'Sign Out' button. Below the logo, the username 'testrun' is displayed. Below the username are two icons: 'BILLING' (a hand holding a coin) and 'Profile' (a person icon). The 'Book an Appointment' form is open, containing the following fields: 'Name:' (testrun), 'Address:' (testrun), 'Appointment for:' (testrun), 'Date:' (23/05/2024), and 'Time:' (08:39 am). A 'Submit' button is at the bottom of the form.

Trying to update user's profile details. Testing if the updated details are can be seen in the database.

The screenshot shows the Medicare Health app interface. At the top left is the 'MEDICARE HEALTH' logo. At the top right is a 'Back' button. The 'Profile Details' form is open, containing the following fields: 'Name:' (testrun), 'Birth Date:' (10/05/2024), 'Gender:' (Male), 'Contact Details:' (testrun), and 'Emergency Contact:' (testrun).

Testing the admin login form if the credentials can be retrieved from the database and be able to login.

The image shows an 'Admin Login' form overlay on a website background. The form has a title 'Admin Login' and three input fields: 'ID Number' with the value '123', 'Password' with the value '...', and a 'Login' button. The background features the 'MEDICARE HEALTH' logo, a 'Patient' button, an 'Admin' button, and several service descriptions: 'World Class Doctors', 'Caring Doctors', 'Medical Counseling', and 'Emergency Services'.

Testing the Admin page if the patient list are updated and can see their billing information.

The image shows an 'Admin' page with a 'Billing Information' form and a 'PATIENTS LIST' table. The 'Billing Information' form has fields for 'Name' (patient), 'Detail' (Billing Detail), 'Total Price' (Total Price), 'Due Date' (dd/mm/yyyy), and 'Status' (Pending), with a 'Save' button. The 'PATIENTS LIST' table has a header 'PATIENTS LIST' and a body with five rows of patient data.

PATIENTS LIST
123
jimuel
jonas
testrun
yasay

Checking if the booked appointments are on the list and can be marked as done and remove from the list.

MEDICARE
HEALTH

BackSign Out

Appointment List

Name: jonas	Address: #30 iba, hagonoy, bulacan	Appointment for: check up	Date: 2024-05-22	Time: 08:30:00	Done
Name: testrun	Address: testrun	Appointment for: testrun	Date: 2024-05-23	Time: 08:39:00	Done
Name: yasay	Address: 123	Appointment for: 123	Date: 2024-05-22	Time: 08:30:00	Done