



IE4717 Web Application Design - Project Report

Design project group number: F33-DG13

Team members: CHEN HUNG-JUI, MUHAMMAD AMEERUL BIN AZMAN

Project Title: Henry's Burgers

Summary of Project:

This project introduces an innovative online food ordering system via a user-friendly website. It allows customers to browse through a diverse menu, selecting and adding items to their cart with ease. A significant aspect of this platform is the user account feature. By creating an account, customers can save their address and preferences, which simplifies the ordering process for future visits, eliminating the need to repeatedly enter the same information.

The cart functionality is versatile, offering users the ability to modify the quantities of chosen items or remove them entirely, providing a customizable ordering experience prior to checkout. Furthermore, the website is equipped with administrative capabilities for operators. They can log into an admin account to alter menu item prices, adapting to market trends or promotional needs. Additionally, the system enables the generation of daily sales reports, a crucial tool for monitoring business performance and making data-driven decisions.

Designed with both customer convenience and operational efficiency in mind, this online ordering platform streamlines the food purchasing process, enhancing the overall experience for users and providing valuable management tools for operators.

Content Page

1. Application Requirements and Specifications.....	3
2. Functional Requirements and Specifications.....	3
3. Design of the web application.....	4
a. Site Map.....	4
b. Story Board.....	4
c. Wireframes.....	5
d. Database.....	6
e. Web application testing plan.....	7
4. Implementation.....	7
5. Testing of web application.....	8
6. Conclusion.....	10
7. Appendices.....	11
Contribution to the project by each team member.....	11
Appendix : Source Code.....	11

1. Application Requirements and Specifications

R1) Users will be able to see a front page with general details.

R2) Users will be able to browse the food available from the menu, in the menu page there will be categories for easier viewing.

R3) There will be a “Add to cart” button for each item which would then be added to the cart. The cart page can be viewed to check what has been added for users to modify the quantity or even remove them. This is also where the user can check out for their order.

R4) An account page will be there for new users to register and for existing users to login, with their account there will be their basic particulars especially address for the delivery.

R5) Users can also view the shop’s details from the ‘About Us’ page where there will be information about us, our mission and values and how to contact us.

R6) Staff members can access the admin account to adjust item prices and create daily sales reports.

2. Functional Requirements and Specifications

F1) A Home Page for general information, news and offers.

F2) A Menu Page for users to browse our products with categories like Set Meals, Burgers, Sides and Drinks

F3) A Cart Page for when items are added from the menu, from this page users can view what has been added and modify them.

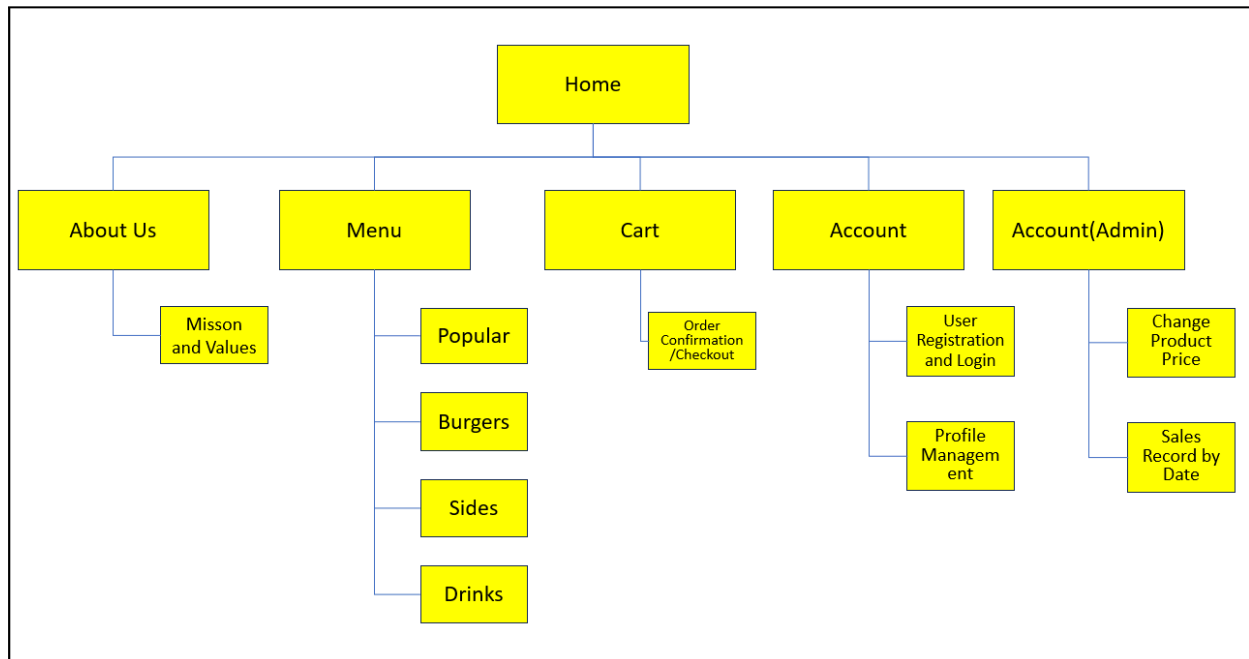
F4) An Account Page where users can login or create their account for basic particulars like their address.

F5) An About Us Page where users can view more information about our shop’s mission and values and how to contact us.

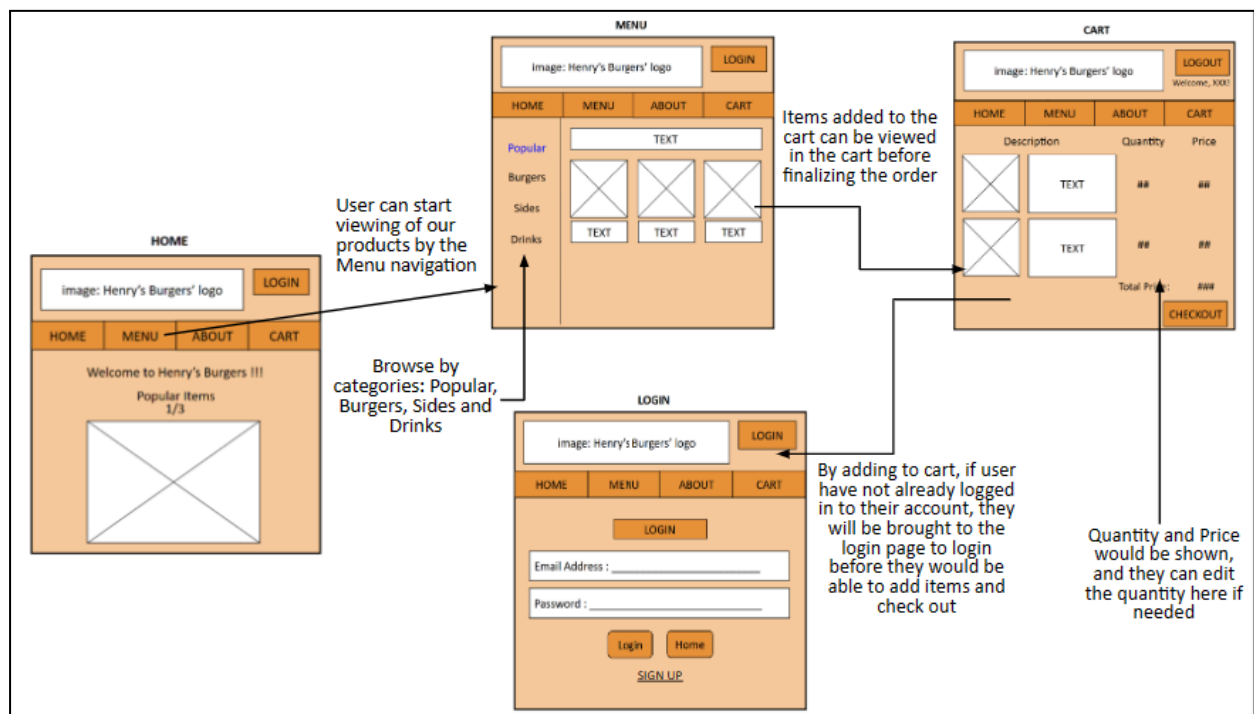
F6) An Administration Page that allows staff to change item prices and produce sales reports for selected dates.

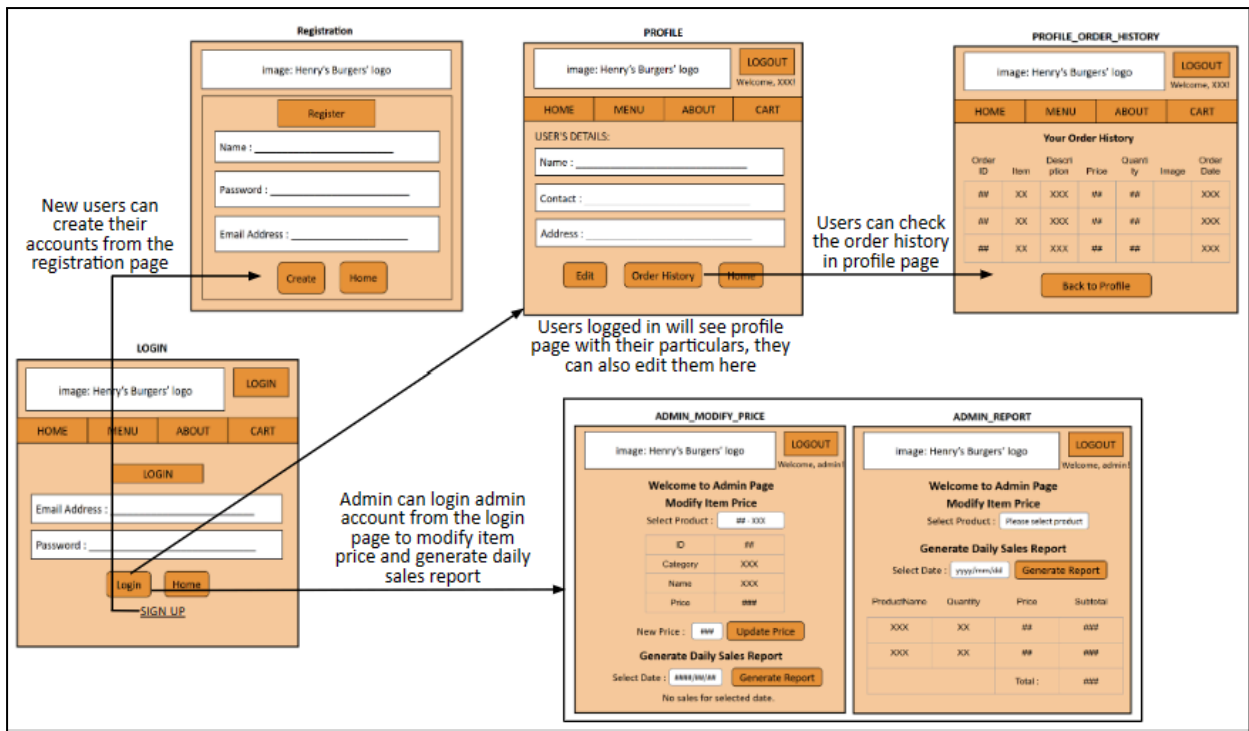
3. Design of the web application

a. Site Map

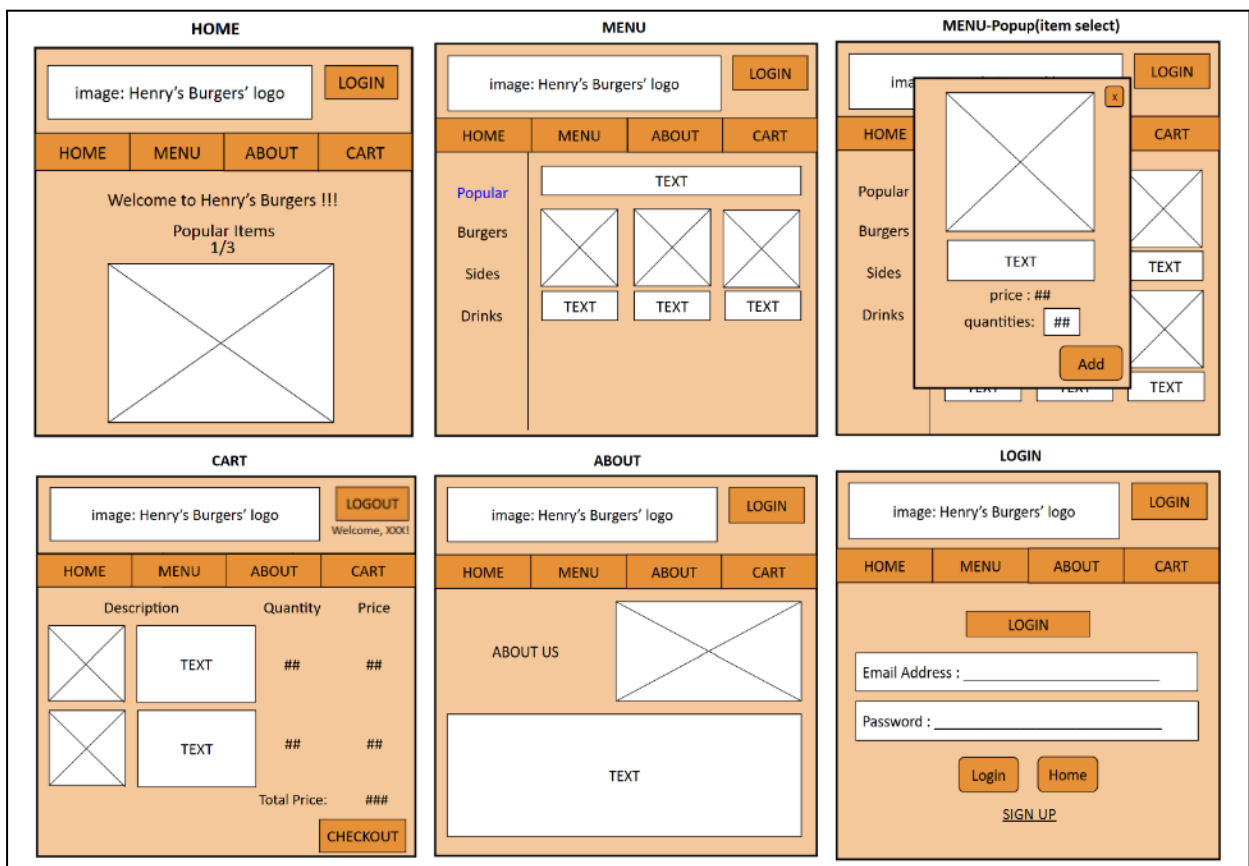


b. Story Board





c. Wireframes



Registration

image: Henry's Burgers' logo

Register

Name :

Password :

Email Address :

CreateHome

PROFILE

image: Henry's Burgers' logo

LOGOUT

Welcome, XXX!

HOME

MENU

ABOUT

CART

USER'S DETAILS:

Name :

Contact :

Address :

EditOrder HistoryHome

PROFILE_ORDER_HISTORY

image: Henry's Burgers' logo

LOGOUT

Welcome, XXX!

HOME

MENU

ABOUT

CART

Your Order History

Order ID	Item	Description	Price	Quantity	Image	Order Date
##	XX	XXX	##	##		XXX
##	XX	XXX	##	##		XXX
##	XX	XXX	##	##		XXX

Back to Profile

ADMIN_MODIFY_PRICE

image: Henry's Burgers' logo

LOGOUT

Welcome, admin!

Welcome to Admin Page

Modify Item Price

Select Product :

ID	##
Category	XXX
Name	XXX
Price	###

New Price :

Update Price

Generate Daily Sales Report

Select Date :

Generate Report

No sales for selected date.

ADMIN_REPORT

image: Henry's Burgers' logo

LOGOUT

Welcome, admin!

Welcome to Admin Page

Modify Item Price

Select Product :

Generate Daily Sales Report

Select Date :

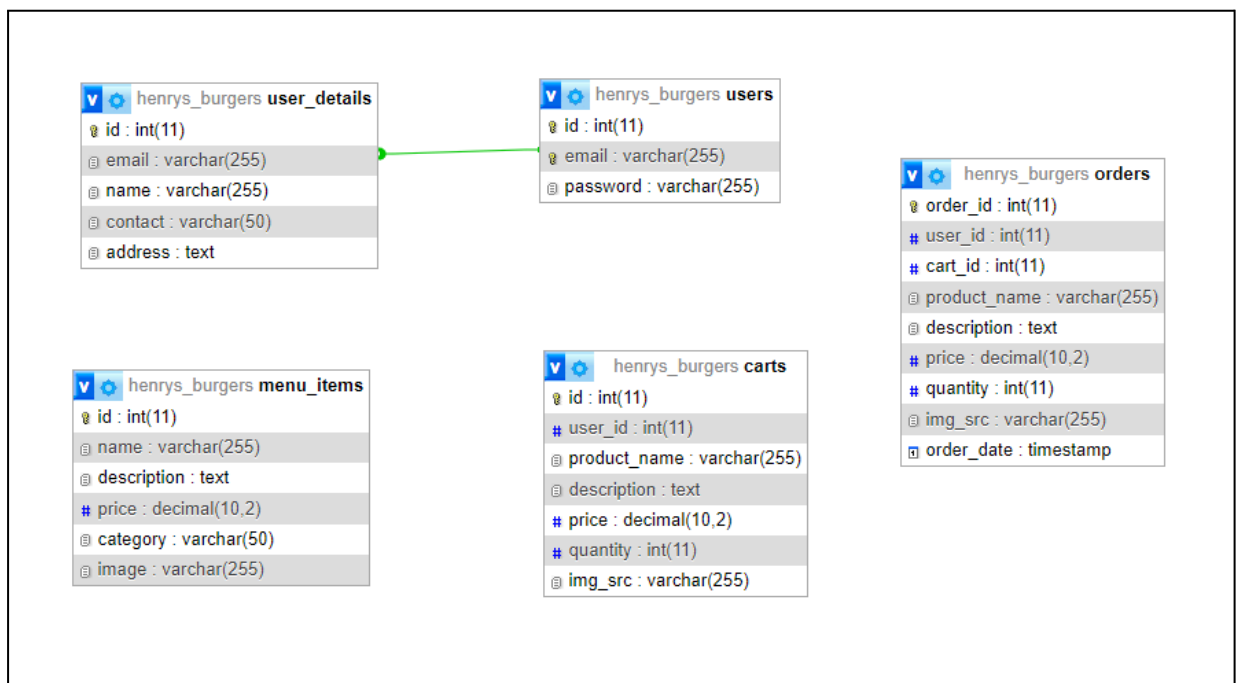
Generate Report

ProductName	Quantity	Price	Subtotal
XXX	XX	##	###
XXX	XX	##	###
Total :			###

← numbers

← image

d. Database



e. Web application testing plan

- i. Basic pages test. Ensure all the main pages are working and that the navigation bar leads to the intended pages
- ii. Menu Page. For each category ("Popular," "Burgers," "Sides," "Drinks"), ensure that the items are listed correctly with accurate descriptions and prices. Test the functionality to add items to the cart from the menu.
- iii. Cart Functionality. Test the addition and removal of items, updating quantities, and price calculations. Check the checkout process, including the order confirmation screen and any redirections post-purchase.
- iv. Account Registration and Login. Test the user registration process for validation, storage of user data, and confirmation messages. Test the login process for security, session management, and error handling with incorrect inputs.
- v. Profile Management. Ensure that users can view and edit their profiles, and changes are saved and reflected correctly. Validate form fields for correct data entry and constraints.
- vi. Admin Account Functions. Check if price changes are reflected immediately and correctly across the platform. Test the sales record retrieval by date, ensuring accurate data display and sorting.
- vii. Navigation Ease. Assess how easily a new user can navigate through the site and perform essential tasks like ordering, registering, and accessing account information.
- viii. Content Clarity. Verify that content across all pages is clear and understandable.

4. Implementation

- a. Coded the web pages and making sure that each page and CSS is how we want it to be

- b. Start on the menu page first as it will contain the items to be added to cart. Coded the cart page/ function next but realized mid way that we needed the account and sessions to be implemented first for a flawless experience when browsing the web and adding items to cart. After implementing accounts and sessions, we proceeded to finish up the cart page and added its functionalities.
- c. We coded/added functions as we progressed with each page. As each page has different needs and buttons to be coded, we worked on functions that are important for the basic structure first before adding or polishing them. After the main functions are done, we proceed to tackle the smaller functions of other pages to make the webpage more dynamic like a slideshow on the Home page.

5. Testing of web application

a. Basic Pages Test:

- Test Performed: Each link in the navigation bar was clicked to verify its functionality and correct redirection.
- Results: All main pages, including Home, Menu, Contact, and About Us, were working correctly. The links in the navigation bar accurately led to the intended pages. No broken links or misdirections were found.

b. Menu Page:

- Test Performed: Checked each menu category for correct item listings, accurate descriptions, and prices. Tested the functionality of adding items to the cart from the menu.
- Results: All menu items under "Popular," "Burgers," "Sides," and "Drinks" were listed correctly with precise descriptions and prices. The "add to cart" function worked seamlessly for various items.

c. Cart Functionality:

- Test Performed: Items were added and removed from the cart, quantities updated, and price calculations checked. The checkout process, including order confirmation and post-purchase redirections, was tested.
- Results: Cart functionality performed well with accurate price calculations and smooth updating of quantities. The checkout process was efficient, with a clear order confirmation screen. Post-purchase, the user was correctly redirected to a thank-you page.

d. Account Registration and Login:

- Test Performed: New accounts were registered with various inputs, including incorrect ones, to test validation and error handling. Login functionality was tested for security and session management.
- Results: Registration process successfully validated and stored user data with appropriate confirmation messages. The login process was secure, with robust session management. Error messages were clear and informative when incorrect inputs were entered.

e. Profile Management:

- Test Performed: Changes were made to user profiles to test if they were saved and reflected accurately. Form fields were tested for data entry constraints.
- Results: Profile edits were accurately saved and reflected in the user interface. Form fields demonstrated proper validation for data types and constraints.

f. Admin Account Functions:

- Test Performed: Admins made price changes, and the immediate reflection of these changes was checked. Sales records were retrieved by date to ensure accuracy and sorting.

- Results: Price updates by admins were immediately reflected across the platform without issues. Sales records were accurately displayed and sorted by date.

g. Navigation Ease:

- Test Performed: User testing was conducted with participants unfamiliar with the site to gather feedback on navigation ease.
- Results: New users found the site easy to navigate. They could efficiently perform tasks like ordering, registering, and accessing account information. A few users suggested minor improvements in the menu layout.

h. Content Clarity:

- Test Performed: The content across all pages was reviewed for clarity.
- Results: The content was clear and easy to understand. A copy editor suggested minor revisions to the About Us page to enhance clarity.

Overall, the website performed excellently in all tested areas, demonstrating good functionality, usability, and clarity. The few suggestions for improvement were minor and could be easily implemented for an even better user experience.

6. Conclusion

The core structure of the website has been made with a clear navigation path from the homepage to other main pages such as Menu, Cart, Account and About. A comprehensive menu page was implemented that categorizes items into Popular, Burgers, Sides and Drinks with each item having their name, description and price on it. Items are fetched from the database and Popular items are taken from the top 3 sellers amongst all the items. A Cart function/page has been also implemented that allows users to add and remove items, update quantities and proceed to checkout items which then users will receive an email confirming their orders and details regarding their orders. An account function has also been added for users to register, log in

and manage their profile. Users can also view their order history from the profile page. Lastly, an admin page that is also accessible to an admin account has been added for updating product prices and viewing of the sales records.

7. Appendices

Contribution to the project by each team member

Pages	Contributor
Home	CHEN HUNG-JUI
Menu	CHEN HUNG-JUI
About	CHEN HUNG-JUI
Cart	CHEN HUNG-JUI
Login	MUHAMMAD AMEERUL BIN AZMAN
Register	MUHAMMAD AMEERUL BIN AZMAN
Profile	MUHAMMAD AMEERUL BIN AZMAN
Order History	MUHAMMAD AMEERUL BIN AZMAN
Admin	CHEN HUNG-JUI

Appendix : Source Code

#home.php

```
<?php
// Start the session
session_start();
include 'php/dbconn.php';

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {
    header('Location: admin.php');
}

$popularItems = [];

try {
    // Create the query
    $stmt = $conn->prepare("SELECT product_name, description, price, img_src, SUM(quantity) as total_quantity FROM orders GROUP BY
product_name, description, price, img_src ORDER BY total_quantity DESC LIMIT 3");

    // Execute the query
    $stmt->execute();
```

```

// Set the resulting array to associative
$result = $stmt->setFetchMode(PDO::FETCH_ASSOC);
$popularItems = $stmt->fetchAll();
} catch(PDOException $e) {
    echo "Error: " . $e->getMessage();
}
$conn = null; // Close the database connection

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>
    <nav>
        <a href="home.php">HOME</a>
        <a href="menu.php">MENU</a>
        <a href="about.php">ABOUT</a>
        <a href="cart.php">CART</a>
    </nav>
    <main class="main-home">
        <h1>Welcome to Henry's Burgers !!!</h1>
        <div id="slideshow" class="slideshow-container">
            <h2>Popular Items!</h2>
            <!-- Full-width images with number and caption text -->
            <?php foreach($popularItems as $index => $item): ?>
                <div class="mySlides fade">
                    <div class="numbertext"><?php echo $index + 1; ?> / 3</div>
                    
                    <div class="numbertext"><?php echo htmlspecialchars($item['product_name']); ?></div>
                </div>
            <?php endforeach; ?>
            <!-- Next and previous buttons -->
            <a class="prev" onclick="plusSlides(-1)">&#10094;</a>
            <a class="next" onclick="plusSlides(1)">&#10095;</a>
        </div>
    </main>
    <footer>
        <div class="footer-container">
            <div class="footer-section">
                <h4>Menu</h4>
                <ul>
                    <li><a href="menu.php">Popular</a></li>
                    <li><a href="menu.php?category=burgers">Burgers</a></li>
                    <li><a href="menu.php?category=sides">Sides</a></li>
                    <li><a href="menu.php?category=drinks">Drinks</a></li>
                </ul>
            </div>
        </div>
    </footer>

```

```

        <div></div>
        <div class="footer-section">
            <h4>Contact Us</h4>
            <ul>
                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
henrys@burgers.com</a></li>
                <li><a href="about.php">About Us</a></li>
            </ul>
        </div>

        <div>
            <br>Copyright &copy; 2023 Henry's Burgers<br>
        </div>
    </footer>
    <script src="js/homepage.js"></script>
</body>
</html>

```

#menu.php

```

<?php
// Start the session
session_start();
include 'php/dbconn.php';

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {
    header('Location: admin.php');
}

// Get the category from the URL parameter
$category = isset($_GET['category']) ? $_GET['category'] : null;

// Fetch the items from the database based on the category
$items = [];
if ($conn) {
    if ($category) {
        // If a category is specified, fetch items from that category
        $stmt = $conn->prepare("SELECT * FROM menu_items WHERE category = :category");
        $stmt->bindParam(':category', $category, PDO::PARAM_STR);
    } else {
        // If no category is specified, fetch popular items
        // $stmt = $conn->prepare("SELECT product_name, description, price, img_src, SUM(quantity) as total_quantity FROM orders GROUP BY
product_name, description, price, img_src ORDER BY total_quantity DESC LIMIT 3");
        $stmt = $conn->prepare("SELECT o.product_name, o.description, mi.price, o.img_src, SUM(o.quantity) as total_quantity FROM
orders o INNER JOIN menu_items mi ON o.product_name = mi.name GROUP BY o.product_name, o.description, mi.price, o.img_src ORDER BY
total_quantity DESC LIMIT 3");
    }

    // Execute the statement
    $stmt->execute();

    // Fetch the results
    if ($category) {
        $items = $stmt->fetchAll(PDO::FETCH_ASSOC);
    } else {
        $popularItems = $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
} else {
    // Handle the error if the connection failed
    die("Connection failed: " . $conn->errorInfo());
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Menu - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>

```

```

<div class="logo-container">
  
</div>
<div class="session-container">
  <div class="login-container">
    <?php if(isset($_SESSION['user_id'])): ?>
      <a href="logout.php">LOGOUT</a>
    <?php else: ?>
      <a href="loginpage.php">LOGIN</a>
    <?php endif; ?>
  </div>
  <div class="user-name-container">
    <?php if(isset($_SESSION['user_name'])): ?>
      <?php echo 'Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'</a>'; ?>
    <?php endif; ?>
  </div>
</div>
</header>
<nav>
  <a href="home.php">HOME</a>
  <a href="menu.php">MENU</a>
  <a href="about.php">ABOUT</a>
  <a href="cart.php">CART</a>
</nav>
<main class="menu-content">
  <aside>
    <ul>
      <li><a href="menu.php">Popular</a></li>
      <li><a href="menu.php?category=burgers">Burgers</a></li>
      <li><a href="menu.php?category=sides">Sides</a></li>
      <li><a href="menu.php?category=drinks">Drinks</a></li>
    </ul>
  </aside>
  <section class="menu">
    <?php if ($category): ?>
      <?php foreach($items as $item): ?>
        <div class="menu-item" data-name="<?php echo htmlspecialchars($item['name']); ?>"
data-description="<?php echo htmlspecialchars($item['description']); ?>" data-price="<?php echo htmlspecialchars($item['price']); ?>"
class="menu-image">
          <?php echo htmlspecialchars($item['name']); ?></p>
        </div>
      <?php endforeach; ?>
    <?php else: ?>
      <h2>Most Popular Items at Henry's Burgers</h2>
      <?php foreach ($popularItems as $item): ?>
        <div class="menu-item" data-name="<?php echo htmlspecialchars($item['product_name']); ?>"
data-description="<?php echo htmlspecialchars($item['description']); ?>" data-price="<?php echo htmlspecialchars($item['price']); ?>"
class="menu-image">
          <?php echo htmlspecialchars($item['product_name']); ?></p>
        </div>
      <?php endforeach; ?>
    <?php endif; ?>
  </section>
  <div id="toast" class="toast">Item(s) added to cart!</div>
</main>
<!-- The Modal -->
<div id="myModal" class="modal">
  <div class="modal-content">
    <span class="close">&times;</span>
    <img id="modal-image" src="" alt="Menu Item Image">
    <h2>Menu Item Name</h2>
    <p>Description of the item...</p>
    <p id=priceText>Price: $xx.xx</p>
    <!-- Quantity Input Field -->
    <div class="quantity-input">
      <label for="item-quantity">Quantity:</label>
      <input type="number" id="item-quantity" value="1" min="1">
    </div>
  </div>
</div>

```

```

        <button class="add-to-cart">Add to Cart</button>
    </div>
</div>
<footer>
    <div class="footer-container">
        <div class="footer-section">
            <h4>Menu</h4>
            <ul>
                <li><a href="menu.php">Popular</a></li>
                <li><a href="menu.php?category=burgers">Burgers</a></li>
                <li><a href="menu.php?category=sides">Sides</a></li>
                <li><a href="menu.php?category=drinks">Drinks</a></li>
            </ul>
        </div>
        <div></div>
        <div class="footer-section">
            <h4>Contact Us</h4>
            <ul>
                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
henrys@burgers.com</a></li>
                <li><a href="about.php">About Us</a></li>
            </ul>
        </div>
    </div>
    <br>Copyright &copy; 2023 Henry's Burgers<br>
</footer>

<script src="js/menu.js"></script>
</body>
</html>

```

#about.php

```

<?php
// Start the session
session_start();

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {
    header('Location: admin.php');
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>About - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>
    <nav>
        <a href="home.php">HOME</a>
        <a href="menu.php">MENU</a>
    </nav>

```

```

    <a href="about.php">ABOUT</a>
    <a href="cart.php">CART</a>
</nav>
<main class = "main-about">
    <div class="about-header">
        <div class="text-header">
            <h1>ABOUT US</h1>
            <h2>🍔 Henry's Burgers 🍔</h2>
        </div>
        
    </div>
    <div class="about-home">
        <p class="about-text">
            <b>Where Legacy Meets Culinary Innovation</b> <br><br>
            <b>Our Origin Story:</b><br><br>
            In the heart of Brooklyn in 1957, a passionate young chef named Henry envisioned a place where people could come together, not just to eat, but to connect over food that tells a story. Henry's wasn't just any diner; it was where time-honored recipes were cherished and new ones were born.
            <br><br>
            <b>The Journey to Nationwide Sensation:</b><br><br>
            Henry's relentless pursuit of perfection and his flair for combining tradition with modernity turned one small diner into a household name across the nation. Each branch, while retaining the core essence of Henry's, brings with it a touch of local flavor, celebrating the diversity and unity of our great nation.
            <br><br>
            <b>Why Henry's Stands Out:</b><br><br>
            Every burger at Henry's isn't just a meal; it's a narrative. Our ingredients are meticulously sourced, ensuring the freshest and most ethically-produced elements come together on your plate. Our signature blend of traditional and innovative flavors means every visit is both a walk down memory lane and a new culinary adventure.
            <br><br>
            <b>Our Commitment to the Future:</b><br><br>
            While we take pride in our rich history, we're not just looking backward. At Henry's, we're committed to sustainability, ethical sourcing, and constantly evolving our menu to suit the dynamic tastes of our patrons.
            <br><br>
            🌟 #RelishTheLegacy at Henry's Burgers 🌟
        </p>
    </div>
</main>
<footer>
    <div class="footer-container">
        <div class="footer-section">
            <h4>Menu</h4>
            <ul>
                <li><a href="menu.php">Popular</a></li>
                <li><a href="menu.php?category=burgers">Burgers</a></li>
                <li><a href="menu.php?category=sides">Sides</a></li>
                <li><a href="menu.php?category=drinks">Drinks</a></li>
            </ul>
        </div>
        <div class="footer-section">
            <h4>Contact Us</h4>
            <ul>
                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
                    henrys@burgers.com</a></li>
                <li><a href="about.php">About Us</a></li>
            </ul>
        </div>
    </div>
    <br>Copyright &copy; 2023 Henry's Burgers<br>
</footer>
<script src="scripts.js"></script>
</body>
</html>

```

#cart.php

```

<?php
// Start the session
session_start();
include 'php/dbconn.php';

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {

```



```

        header('Location: admin.php');
    }

    // Check if the user is logged in
    if(!isset($_SESSION['user_id'])) {
        echo "<script>alert('Please log in to see the cart'); window.location.href='loginpage.php';</script>";
        exit; // End the execution of the script
    }
    $totalPrice = 0;
    $user_id = $_SESSION['user_id'];
    $stmt = $conn->prepare("SELECT * FROM carts WHERE user_id = ?");
    $stmt->execute([$user_id]);
    $cartItems = $stmt->fetchAll();
    ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Cart - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
    <style>

</style>
</head>
<body>
    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'!</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>
    <nav>
        <a href="home.php">HOME</a>
        <a href="menu.php">MENU</a>
        <a href="about.php">ABOUT</a>
        <a href="cart.php">CART</a>
    </nav>
    <div class="main-cart">
        <table class="cart-table">
            <thead>
                <tr>
                    <th>Description</th>
                    <th>Quantity</th>
                    <th>Price</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach($cartItems as $item): ?>
                    <tr > <!-- Add an ID to the row -->
                        <td class="product-desc">
                            
                            <?php
                                echo htmlspecialchars($item['product_name']) . "<br><br>" .
                                htmlspecialchars($item['description']);
                            ?>
                        </td>
                        <td>
                            <form method="POST" action="php/update_quantity.php">

```

```

$item['id']; ?>" />

onclick="return confirmDecrease(this.form, this);" />

?></span>

onclick="return confirmIncrease(this.form, this);" />

</td>
</td>

<?php
$itemTotal = $item['price'] * $item['quantity'];
echo "$" . number_format($itemTotal, 2);
$totalPrice += $itemTotal; // Add to the total price
?>

</td>
</td>
<form method="POST" action="php/remove_item.php">
Total Price : </td>
 <?php echo number_format($totalPrice, 2); ?></td> </tr> </tfoot> </table> <form action="php/checkout.php" method="post"> <button type="submit" class="checkout-button">CHECKOUT</button> </form> </div>  <footer> <div class="footer-container"> <div class="footer-section"> <h4>Menu</h4> <ul> <li><a href="menu.php">Popular</a></li> <li><a href="menu.php?category=burgers">Burgers</a></li> <li><a href="menu.php?category=sides">Sides</a></li> <li><a href="menu.php?category=drinks">Drinks</a></li> </ul> </div> <div></div> <div class="footer-section"> <h4>Contact Us</h4> <ul> <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br> henrys@burgers.com</a></li> <li><a href="about.php">About Us</a></li> </ul> </div> </div> <br>Copyright &copy; 2023 Henry's Burgers<br> </footer> <script src="js/cart.js"></script> </body> </html> |
```

#loginpage.php

```

<?php
// Start the session
session_start();

?>
<!DOCTYPE html>

```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Home - Henry's Burgers</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <div class="logo-container">
      
    </div>
    <div class="login-container">
      <a href="loginpage.php">LOGIN</a>
    </div>
  </header>
  <nav>
    <a href="home.php">HOME</a>
    <a href="menu.php">MENU</a>
    <a href="about.php">ABOUT</a>
    <a href="cart.php">CART</a>
  </nav>
  <div class="login-page">
    <h2>LOGIN</h2>
    <form action="php/login.php" method="post">
      <label for="email">Email Address:</label>
      <input type="email" id="email" name="email" required>

      <label for="password">Password:</label>
      <input type="password" id="password" name="password" required>
      <!--Error message for login credentials being wrong-->
      <?php
      if (isset($_GET['error'])) {
        echo "<div class='error-message'>" . htmlspecialchars($_GET['error']) . "</div>";
      }
      ?>
      <div class="buttons">
        <button type="submit">Login</button>
        <a href="home.html" class="home-button">Home</a>
      </div>
    </form>
    <a href="sign.php">New user ? Create your account here! </a>
  </div>
  <footer>
    <div class="footer-container">
      <div class="footer-section">
        <h4>Menu</h4>
        <ul>
          <li><a href="menu.php">Popular</a></li>
          <li><a href="menu.php?category=burgers">Burgers</a></li>
          <li><a href="menu.php?category=sides">Sides</a></li>
          <li><a href="menu.php?category=drinks">Drinks</a></li>
        </ul>
      </div>
      <div></div>
      <div class="footer-section">
        <h4>Contact Us</h4>
        <ul>
          <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
            henrys@burgers.com</a></li>
          <li><a href="about.php">About Us</a></li>
        </ul>
      </div>
    </div>
    <br>Copyright &copy; 2023 Henry's Burgers<br>
  </footer>

  <script></script>
</body>
</html>

```

#logout.php

```

<?php

```

```

session_start();

// Unset all session variables
session_unset();

// Destroy the session
session_destroy();

// Redirect to homepage with a logged_out parameter
header('Location: home.php');
exit();
?>

```

#profile.php

```

<?php
session_start();

include 'php/dbconn.php';

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {
    header('Location: admin.php');
}

// Fetch user details using user ID
$user_id = $_SESSION['user_id'];

// Handle form submission to edit user details
if ($_SERVER['REQUEST_METHOD'] == "POST") {
    $updatedName = $_POST['Name'];
    $updatedContact = $_POST['Contact'];
    $updatedAddress = $_POST['Address'];

    // Prepare SQL to update user details
    $sql = "UPDATE user_details SET name = :name, contact = :contact, address = :address WHERE id = :user_id";
    $stmt = $conn->prepare($sql);

    // Bind values
    $stmt->bindParam(':name', $updatedName);
    $stmt->bindParam(':contact', $updatedContact);
    $stmt->bindParam(':address', $updatedAddress);
    $stmt->bindParam(':user_id', $user_id);

    // Execute and check for success
    if ($stmt->execute()) {
        echo "<script>alert('Details Updated Successfully!');</script>";
    } else {
        echo "<script>alert('Error updating details. Please try again later.');

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Home - Henry's Burgers</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
<div class="logo-container">

</div>
<div class="session-container">
<div class="login-container">
<?php if(isset($_SESSION['user_id'])): ?>
<a href="logout.php">LOGOUT</a>
<?php else: ?>
<a href="loginpage.php">LOGIN</a>
<?php endif; ?>
</div>
<div class="user-name-container">
<?php if(isset($_SESSION['user_name'])): ?>
<?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).!</a></p>'; ?>
<?php endif; ?>
</div>
</div>
</header>
<nav>
<a href="home.php">HOME</a>
<a href="menu.php">MENU</a>
<a href="about.php">ABOUT</a>
<a href="cart.php">CART</a>
</nav>
<div class="profile-container">
<h2>USER'S DETAILS</h2>
<form action="profile.php" method="post" onsubmit="return validateForm()" class="profile-form">
<div class="input-group">
<label for="Name">Name :</label>
<input type="text" id="UserName" name="Name" value="<?php echo htmlspecialchars($userName); ?>"
required readonly>
<span id="nameError" class="error-text"></span>
</div>
<div class="input-group">
<label for="Contact">Contact :</label>
<input type="text" id="UserContact" name="Contact" value="<?php echo
htmlspecialchars($userContact); ?>" required readonly>
<span id="contactError" class="error-text"></span>
</div>
<div class="input-group">
<label for="Address">Address :</label>
<input type="text" id="UserAddress" name="Address" value="<?php echo
htmlspecialchars($userAddress); ?>" required readonly>
<span id="addressError" class="error-text"></span>
</div>
<div class="buttons">
<button type="button" id="editButton">Edit</button>
<button type="submit" id="submitButton" style="display:none;">Save Changes</button>
<button type="button" id="cancelButton" style="display:none;">Cancel</button>
<a href="orderhistory.php" class="order-history-button">Order History</a>
<a href="home.php" class="home-button">Home</a>
</div>
</form>
</div>
<footer>
<div class="footer-container">
<div class="footer-section">
<h4>Menu</h4>
<ul>
<li><a href="menu.php">Popular</a></li>
<li><a href="menu.php?category=burgers">Burgers</a></li>
<li><a href="menu.php?category=sides">Sides</a></li>
<li><a href="menu.php?category=drinks">Drinks</a></li>
</ul>

```

```

        </div>
    </div></div>
    <div class="footer-section">
        <h4>Contact Us</h4>
        <ul>
            <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
henrys@burgers.com</a></li>
            <li><a href="about.php">About Us</a></li>
        </ul>
    </div>
</div>
<br>Copyright &copy; 2023 Henry's Burgers<br>
</footer>

<script src="js/profilevalid.js"></script>
<script src="js/editprofile.js"></script>
</body>
</html>

```

#sign.php

```

<?php
// Start the session
session_start();

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>
    <main>
        <div class="registration-form">
            <h2>Account Registration</h2>
            <form action="php/register.php" method="post" onsubmit="return validateForm();">

                <!-- Name input -->

                <label for="name">Name :</label>
                <input type="text" id="name" name="name" required>
                <span id="nameError" class="error-text"></span>

                <!-- Email input -->
                <label for="email">Email Address :</label>
                <input type="email" id="email" name="email" required>
                <span id="emailError" class="error-text"></span>

                <!-- Password input -->
                <label for="password">Password :</label>
                <input type="password" id="password" name="password" required>
                <span id="passwordError" class="error-text"></span>
            </form>
        </div>
    </main>

```

```

        <!-- Re-enter Password input -->
        <label for="confirm_password">Re-enter Password:</label>
        <input type="password" id="confirm_password" name="confirm_password" required>
        <span id="confirmPasswordError" class="error-text"></span>

        <!-- Contact input -->
        <label for="contact">Contact Number :</label>
        <input type="tel" id="contact" name="contact" required>
        <span id="contactError" class="error-text"></span>

        <!-- Address input -->
        <label for="address">Address :</label>
        <input type="text" id="address" name="address" required>
        <span id="addressError" class="error-text"></span>

        <div class="buttons">
            <button type="submit">Create</button>
            <a href="home.php" class="home-button">Home</a>
        </div>
    </form>
</div>
</main>
<footer>
    <div class="footer-container">
        <div class="footer-section">
            <h4>Menu</h4>
            <ul>
                <li><a href="menu.php">Popular</a></li>
                <li><a href="menu.php?category=burgers">Burgers</a></li>
                <li><a href="menu.php?category=sides">Sides</a></li>
                <li><a href="menu.php?category=drinks">Drinks</a></li>
            </ul>
        </div>
        <div></div>
        <div class="footer-section">
            <h4>Contact Us</h4>
            <ul>
                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
                henrys@burgers.com</a></li>
                <li><a href="about.php">About Us</a></li>
            </ul>
        </div>
    </div>
    <br>Copyright &copy; 2023 Henry's Burgers<br>
</footer>
<script src="js/registervailid.js"></script>
</body>
</html>

```

#orderhistory.php

```

<?php
session_start();

include 'php/dbconn.php';

//navigate to admin page for admin account
if (isset($_SESSION['user_email']) && $_SESSION['user_email'] === 'admin@admin.com') {
    header('Location: admin.php');
}

// Check if user is logged in
if(!isset($_SESSION['user_id'])) {
    header("Location: loginpage.php");
    exit();
}

$user_id = $_SESSION['user_id'];

// Fetch order history
$query = "SELECT cart_id, GROUP_CONCAT(product_name) as product_names,
        GROUP_CONCAT(description) as descriptions,
        GROUP_CONCAT(price) as prices,

```

```

        GROUP_CONCAT(quantity) as quantities,
        GROUP_CONCAT(img_src) as img_srcs,
        MAX(order_date) as order_date
    FROM orders
    WHERE user_id = :user_id
    GROUP BY cart_id
    ORDER BY order_date ASC";
$stmt = $conn->prepare($query);
$stmt->execute([':user_id' => $user_id]);

$orders = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
    <title>Order History</title>
</head>
<body>
    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo ' <p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'!</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>
    <nav>
        <a href="home.php">HOME</a>
        <a href="menu.php">MENU</a>
        <a href="about.php">ABOUT</a>
        <a href="cart.php">CART</a>
    </nav>
    <div class="orderhistory-cart">
        <h2>Your Order History</h2>
        <table class="orderhistory-table">
            <thead>
                <tr>
                    <th>Order ID</th>
                    <th>Product Name</th>
                    <th>Description</th>
                    <th>Price</th>
                    <th>Quantity</th>
                    <th>Image</th>
                    <th>Order Date</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($orders as $order): ?>
                    <tr>
                        <td><?php echo htmlspecialchars($order['cart_id']); ?></td>
                        <td>
                            <?php
                                $products = explode(',', $order['product_names']);
                                foreach($products as $product) {
                                    echo htmlspecialchars($product) . '<br>';
                                }
                            <?>
                        </td>
                    </tr>
                </tbody>
            </table>
        </div>
    </body>
</html>

```



```

        <td>
            <?php
                $descriptions = explode(',', $order['descriptions']);
                foreach($descriptions as $description) {
                    echo htmlspecialchars($description) . '<br>';
                }
            ?>
        </td>
        <td>
            <?php
                $prices = explode(',', $order['prices']);
                foreach($prices as $price) {
                    echo htmlspecialchars($price) . '<br>';
                }
            ?>
        </td>
        <td>
            <?php
                $quantities = explode(',', $order['quantities']);
                foreach($quantities as $quantity) {
                    echo htmlspecialchars($quantity) . '<br>';
                }
            ?>
        </td>
        <td>
            <?php
                $imgs = explode(',', $order['img_srcs']);
                foreach($imgs as $img) {
                    echo '<br>';
                }
            ?>
        </td>
        <td><?php echo htmlspecialchars($order['order_date']); ?></td>
    </tr>
    <?php endforeach; ?>
</tbody>
</table>

<a href="profile.php" class="home-button">Back to Profile</a>
</div>
<footer>
    <div class="footer-container">
        <div class="footer-section">
            <h4>Menu</h4>
            <ul>
                <li><a href="menu.php">Popular</a></li>
                <li><a href="menu.php?category=burgers">Burgers</a></li>
                <li><a href="menu.php?category=sides">Sides</a></li>
                <li><a href="menu.php?category=drinks">Drinks</a></li>
            </ul>
        </div>
        <div></div>
        <div class="footer-section">
            <h4>Contact Us</h4>
            <ul>
                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
henrys@burgers.com</a></li>
                <li><a href="about.php">About Us</a></li>
            </ul>
        </div>
    </div>
    <br>Copyright &copy; 2023 Henry's Burgers<br>
</footer>
</body>
</html>

```

#add_to_cart.php

```

<?php
session_start();

// Check if the user is logged in
if (!isset($_SESSION['user_id'])) {

```

```

    // User is not logged in
    echo "Please log in to add items to the cart.";
    exit; // End the script here
}

$host = 'localhost'; // Database host
$db = 'henrys_burgers'; // Database name
$user = 'root'; // Database username
$pass = ''; // Database password
$charset = 'utf8mb4';

$dsn = "mysql:host=$host;dbname=$db;charset=$charset";
$options = [
    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES => false,
];
$pdo = new PDO($dsn, $user, $pass, $options);

// Assuming $_POST contains 'itemName', 'itemPrice', 'itemQuantity', 'itemDesc', and 'itemImgSrc'
$itemName = $_POST['name'];
$itemPrice = $_POST['price'];
$itemQuantity = $_POST['quantity'];
$itemDesc = $_POST['desc'];
$itemImgSrc = $_POST['imgSrc'];
$user_id = $_SESSION['user_id'];

// Check if the item already exists in the cart for the user
$stmt = $pdo->prepare("SELECT * FROM carts WHERE user_id = ? AND product_name = ?");
$stmt->execute([$user_id, $itemName]);
$existingItem = $stmt->fetch();

if ($existingItem) {
    // Item already exists, so update the quantity and total
    /*$newQuantity = $existingItem['quantity'] + $itemQuantity;
    $newTotal = $newQuantity * $itemPrice;*/
    $newQuantity = $existingItem['quantity'] + $itemQuantity;
    $stmt = $pdo->prepare("UPDATE carts SET quantity = ? WHERE user_id = ? AND product_name = ?");
    $stmt->execute([$newQuantity, $user_id, $itemName]);
} else {
    // Item doesn't exist, so insert a new row
    $total = $itemPrice * $itemQuantity;
    $stmt = $pdo->prepare("INSERT INTO carts (user_id, product_name, description, price, quantity, img_src) VALUES (?, ?, ?, ?, ?, ?)");
    $stmt->execute([$user_id, $itemName, $itemDesc, $itemPrice, $itemQuantity, $itemImgSrc]);
}
echo "Item Added!!!";
?>

```

#admin.php

```

<?php
// admin.php
session_start();

// Redirect users who are not admin.
/*if (!isset($_SESSION['user_email']) || $_SESSION['user_email'] != 'admin@admin.com') {
    header('Location: loginpage.php');
    exit();
}*/

include 'php/dbconn.php';

$menuItems = [];

try {
    // Fetch all menu items
    $stmt = $conn->prepare("SELECT id, name, price, category FROM menu_items");
    $stmt->execute();
    $menuItems = $stmt->fetchAll(PDO::FETCH_ASSOC);
} catch (PDOException $e) {
    $error = "Error: " . $e->getMessage();
}

// Check if the modify price form has been submitted

```

```

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['modify_price'])) {
    $newPrice = $_POST['new_price'];
    $productId = $_POST['product_id'];

    // Update the item's price in the database
    $updateQuery = "UPDATE menu_items SET price = ? WHERE id = ?";
    $stmt = $conn->prepare($updateQuery);
    if ($stmt->execute([$newPrice, $productId])) {
        echo "Price updated successfully!";
    } else {
        echo "Error updating price.";
    }
}

// Generate daily sales report
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['generate_report'])) {
    $selectedDate = $_POST['report_date']; // Fetch the date input from the form

    // Generate daily sales report
    if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['generate_report'])) {
        $selectedDate = $_POST['report_date']; // Fetch the date input from the form

        // Define your query to get the sales report for the selected date, grouped by product name and price
        $reportQuery = "
        SELECT product_name, price, SUM(quantity) AS total_quantity, (price * SUM(quantity)) AS total_price_per_item
        FROM orders
        WHERE DATE(order_date) = ?
        GROUP BY product_name, price";
        $stmt = $conn->prepare($reportQuery);
        $stmt->execute([$selectedDate]);
        $dailySalesReport = $stmt->fetchAll(PDO::FETCH_ASSOC);

        // Calculate the grand total price
        $totalPriceQuery = "SELECT SUM(price * quantity) AS grand_total_price FROM orders WHERE DATE(order_date) = ?";
        $totalPriceStmt = $conn->prepare($totalPriceQuery);
        $totalPriceStmt->execute([$selectedDate]);
        $totalPriceResult = $totalPriceStmt->fetch(PDO::FETCH_ASSOC);
        $grandTotalPrice = $totalPriceResult['grand_total_price'] ?? 0; // Coalesce to 0 if null
    }
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Dashboard - Henry's Burgers</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>

    <header>
        <div class="logo-container">
            
        </div>
        <div class="session-container">
            <div class="login-container">
                <?php if(isset($_SESSION['user_id'])): ?>
                    <a href="logout.php">LOGOUT</a>
                <?php else: ?>
                    <a href="loginpage.php">LOGIN</a>
                <?php endif; ?>
            </div>
            <div class="user-name-container">
                <?php if(isset($_SESSION['user_name'])): ?>
                    <?php echo '<p> Welcome, <a
href="profile.php">'.htmlspecialchars($_SESSION['user_name']).'</a></p>'; ?>
                <?php endif; ?>
            </div>
        </div>
    </header>

```

```

<main class="main-cart">
    <h1>Welcome to Admin Page</h1>
    <section class="modify-price">
        <h2>Modify Item Price</h2>
        <form id="modify-price-form" action="admin.php" method="post">
            <label for="product_selector">Select Product:</label>
            <select id="product_selector" name="product_id" onchange="this.form.submit()" required>
                <option value="">Please select an item</option>
                <?php foreach ($menuItems as $item): ?>
                    <option value="<?php echo htmlspecialchars($item['id']); ?>" <?php if
(isset($_POST['product_id']) && $_POST['product_id'] == $item['id']) echo 'selected'; ?>
                    <?php echo htmlspecialchars($item['id']) . ' ' .
htmlspecialchars($item['name']); ?>
                </option>
                <?php endforeach; ?>
            </select>
        </form>

        <?php
        if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['product_id'])) {
            // Fetch the item details from the database
            $itemId = $_POST['product_id'];
            $query = "SELECT * FROM menu_items WHERE id = ?";
            $stmt = $conn->prepare($query);
            $stmt->execute([$itemId]);
            $itemDetails = $stmt->fetch(PDO::FETCH_ASSOC);

            if ($itemDetails) {
                // Display the details in a table
                echo '<table class="modify-table">';
                echo '<tr><td>ID</td><td>' . htmlspecialchars($itemDetails['id']) . '</td></tr>';
                echo '<tr><td>Category</td><td>' . htmlspecialchars($itemDetails['category']) . '</td></tr>';
                echo '<tr><td>Name</td><td>' . htmlspecialchars($itemDetails['name']) . '</td></tr>';
                echo '<tr><td>Price</td><td>' . htmlspecialchars($itemDetails['price']) . '</td></tr>';
                // ... other details
                echo '</table>';

                // Display the form to update the price
                echo '<form action="admin.php" method="post">';
                echo '<input type="hidden" name="product_id"
value="" . htmlspecialchars($itemDetails['id']) . ">';

                echo '<label for="new_price">New Price : </label>';
                echo '<input type="text" id="new_price" name="new_price" required>';
                echo '<button type="submit" name="modify_price">Update Price</button>';
                echo '</form>';
            }
        }
    </section>

    <section class="sales-report">
        <h2>Generate Daily Sales Report</h2>
        <form action="admin.php" method="post">
            <label for="report_date">Select Date:</label>
            <input type="date" id="report_date" name="report_date" value="<?php echo
isset($_POST['report_date']) ? htmlspecialchars($_POST['report_date']) : ">"; ?>" required>
            <button type="submit" name="generate_report">Generate Report</button>
        </form>

        <?php if (isset($dailySalesReport) && count($dailySalesReport) > 0):
            echo '<table class="report-table">
                <tr>
                    <th>Product Name</th>
                    <th>Quantity</th>
                    <th>Price</th>
                    <th>SubTotal</th>
                </tr>';
            $totalQuan = 0;
            foreach ($dailySalesReport as $reportItem):
                echo '<tr>';
                echo '<td>' . htmlspecialchars($reportItem['product_name']) . '</td>';
                echo '<td>' . htmlspecialchars($reportItem['total_quantity']) . '</td>';
                echo '<td>$' . htmlspecialchars($reportItem['price']) . '</td>';
                echo '<td>$' . htmlspecialchars($reportItem['total_price_per_item']) .
'</td>';

```

```

                                echo '</tr>';
                            endforeach;
                            echo '<tr>';
                                echo '<td colspan="3" style="text-align: right;"><strong>Grand
                                echo '<td><strong>$' . htmlspecialchars(number_format((float)$grandTotalPrice,
                                echo '</tr>';
                                echo '</table>';
                            else:
                                echo '<p>No sales for selected date.</p>';
                        endif; ?>
                    </section>
                </main>
                <footer>
                    <div class="footer-container">
                        <div class="footer-section">
                            <h4>Menu</h4>
                            <ul>
                                <li><a href="menu.php">Popular</a></li>
                                <li><a href="menu.php?category=burgers">Burgers</a></li>
                                <li><a href="menu.php?category=sides">Sides</a></li>
                                <li><a href="menu.php?category=drinks">Drinks</a></li>
                            </ul>
                        </div>
                        <div></div>
                        <div class="footer-section">
                            <h4>Contact Us</h4>
                            <ul>
                                <li><a href="mailto:henrys@burgers.com">More enquiries?<br> Drop us an email!<br>
                                <li><a href="about.php">About Us</a></li>
                            </ul>
                        </div>
                    </div>
                    <br>Copyright &copy; 2023 Henry's Burgers<br>
                </footer>
            </body>
        </html>

```

#cart.js

```

document.querySelector('.checkout-button').addEventListener('click', function(e) {
    if (!confirm('Are you sure you want to checkout?')) {
        e.preventDefault();
    } else {
        alert("Checkout successfully!!");
    }
});

function confirmDecrease(formElement, buttonElement) {
    var currentQuantity = parseInt(formElement.querySelector('span').textContent);
    if(currentQuantity <= 1) {
        return confirm("Are you sure you want to remove this item from your cart?");
    }
    return true;
}

function confirmIncrease(formElement, buttonElement) {
    return true; // Always return true since no confirmation is needed to increase quantity
}

window.onload = function() {
    // Check if we have a saved position in sessionStorage
    const savedPosition = sessionStorage.getItem('scrollPosition');
    if (savedPosition) {
        // Use setTimeout to put at end of call stack to ensure layout has stabilized
        setTimeout(() => {
            window.scrollTo(0, parseInt(savedPosition, 10));
            // Clear the saved position from sessionStorage
            sessionStorage.removeItem('scrollPosition');
        }, .1000);
    }
}

```

```

// Attach event listeners to form submissions to save the scroll position
document.querySelectorAll('form').forEach(form => {
  form.addEventListener('submit', function(event) {
    // Save the current scroll position before submitting the form
    sessionStorage.setItem('scrollTop', window.pageY.toString());
  });
});
};

```

#editprofile.js

```

// Store the initial values
var initialUserName = document.getElementById('UserName').value;
var initialUserContact = document.getElementById('UserContact').value;
var initialUserAddress = document.getElementById('UserAddress').value;

document.getElementById('editButton').addEventListener('click', function() {
  // Toggle the readonly attribute on input fields
  document.getElementById('UserName').removeAttribute('readonly');
  document.getElementById('UserContact').removeAttribute('readonly');
  document.getElementById('UserAddress').removeAttribute('readonly');

  // Hide the edit button and show the save changes and cancel buttons
  document.getElementById('editButton').style.display = 'none';
  document.getElementById('submitButton').style.display = 'block';
  document.getElementById('cancelButton').style.display = 'block';

  // Hide the home and order history buttons using their class names
  document.querySelector('.home-button').style.display = 'none';
  document.querySelector('.order-history-button').style.display = 'none';
});

document.getElementById('cancelButton').addEventListener('click', function() {
  // Reset the fields to their original values using JavaScript variables
  document.getElementById('UserName').value = initialUserName;
  document.getElementById('UserContact').value = initialUserContact;
  document.getElementById('UserAddress').value = initialUserAddress;

  // Set the input fields back to readonly
  document.getElementById('UserName').setAttribute('readonly', true);
  document.getElementById('UserContact').setAttribute('readonly', true);
  document.getElementById('UserAddress').setAttribute('readonly', true);

  // Hide the save changes and cancel buttons and show the edit button
  document.getElementById('editButton').style.display = 'block';
  document.getElementById('submitButton').style.display = 'none';
  document.getElementById('cancelButton').style.display = 'none';
  document.querySelector('.home-button').style.display = 'block';
  document.querySelector('.order-history-button').style.display = 'block';
});

```

#homepage.js

```

var slideIndex = 1;
var slides = document.getElementsByClassName("mySlides");
showSlides(slideIndex); // Show the initial slide

// Next/previous controls
function plusSlides(n) {
  clearInterval(slideInterval); // Clear the existing slide interval
  slideIndex += n;
  if (slideIndex > slides.length) { slideIndex = 1; } // Wrap back to the first slide if the end is passed
  else if (slideIndex < 1) { slideIndex = slides.length; } // Wrap to the last slide if the beginning is passed
  showSlides(slideIndex);
  slideInterval = setInterval(automaticSlide, 5000); // Restart the slide interval
}

function setCurrentSlide(n) {
  clearInterval(slideInterval); // Clear the existing slide interval
  slideIndex = n; // Directly set the slideIndex to the desired slide
  if (slideIndex > slides.length) { slideIndex = 1; } // Wrap back to the first slide if the end is passed
  else if (slideIndex < 1) { slideIndex = slides.length; } // Wrap to the last slide if the beginning is passed
  showSlides(slideIndex);
  slideInterval = setInterval(automaticSlide, 5000); // Restart the slide interval
}

```

```

}

function showSlides(n) {
  if (slides.length === 0) { return; } // Exit the function if there are no slides
  var i;
  for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
  }
  slides[n-1].style.display = "block";
}

// Call this function to go to the next slide every 8 seconds
function automaticSlide() {
  plusSlides(1);
}

// Start the automatic slideshow
var slideInterval = setInterval(automaticSlide, 5000);

```

#menu.js

```

// Get the modal
var modal = document.getElementById("myModal");

// Get the close button
var closeBtn = document.querySelector(".close");

var cart = [];

// Function to open modal
function openModal(menuItemName, description, price, imgSrc) {
  modal.querySelector('h2').textContent = menuItemName;
  modal.querySelector('p:nth-of-type(1)').textContent = description;
  modal.querySelector('p:nth-of-type(2)').textContent = 'Price: $' + price;
  modal.querySelector('#modal-image').setAttribute('src', imgSrc);
  modal.style.display = "block";
}

// When the user clicks on the close button, close the modal
closeBtn.onclick = function() {
  modal.style.display = "none";
}

// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target === modal) { // Ensure the target is specifically the modal
    modal.style.display = "none";
  }
}

// Add event listeners to all menu items
var menuItems = document.querySelectorAll(".menu-item .menu-image");
menuItems.forEach(function(item) {
  item.addEventListener('click', function() {
    var parent = this.parentElement;
    var itemName = parent.getAttribute('data-name');
    var itemDescription = parent.getAttribute('data-description');
    var itemPrice = parent.getAttribute('data-price');
    var itemImageSrc = this.getAttribute('src');

    document.getElementById('item-quantity').value = "1";
    openModal(itemName, itemDescription, itemPrice, itemImageSrc);
  });
});

document.querySelector('.add-to-cart').addEventListener('click', function() {
  var quantity = document.getElementById('item-quantity').value;

  if (quantity == 0) {
    modal.style.display = "none"; // close the modal
    return; // exit without adding to cart
  }
}

```

```

var itemName = modal.querySelector('h2').textContent;
var itemDesc = modal.querySelector('p:nth-of-type(1)').textContent;
var itemPrice = parseFloat(modal.querySelector('p:nth-of-type(2)').textContent.split('$')[1]);
var itemImageSrc = modal.querySelector('#modal-image').getAttribute('src');

var item = {
  name: itemName,
  desc: itemDesc,
  price: itemPrice,
  quantity: parseInt(quantity),
  total: itemPrice * parseInt(quantity),
  imgSrc: itemImageSrc
};

fetch('add_to_cart.php', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/x-www-form-urlencoded'
  },
  body:
`name=${encodeURIComponent(item.name)}&desc=${encodeURIComponent(item.desc)}&price=${item.price}&quantity=${item.quantity}&imgSrc=${en
codeURIComponent(item.imgSrc)}`
})
.then(response => response.text())
.then(data => {
  if (data === "Please log in to add items to the cart.") {
    alert(data); // Show the alert to the user
  } else if (data === "Item Added!!") {
    let toast = document.getElementById("toast");
    toast.className = "toast show";
    setTimeout(() => { toast.className = "toast"; }, 3000);
  } // Show the alert message of item added.
  } else {
    console.log(data); // Process the response or do other actions as needed
  }
})
.catch(error => console.error('Error:', error));

modal.style.display = "none"; // close the modal
});

```

#profilevalid.js

```

// Validation for the Name field
function validateName() {
  let name = document.getElementById("UserName").value;
  let nameError = document.getElementById("nameError");
  let namePattern = /^[a-zA-Z\s]+$/;

  if (!namePattern.test(name)) {
    nameError.innerText = "Name can only contain alphabets and spaces.";
    return false;
  } else {
    nameError.innerText = "";
    return true;
  }
}

// Validation for the Contact field
function validateContact() {
  let contact = document.getElementById("UserContact").value;
  let contactError = document.getElementById("contactError");
  let contactPattern = /^[0-9]+$/;

  if (!contactPattern.test(contact) || contact.length > 10 || contact.length < 8) {
    contactError.innerText = "Contact number should have numbers only and be at 8-10 digits.";
    return false;
  } else {
    contactError.innerText = "";
    return true;
  }
}

```



```

// Main validation function that will be called on form submission
function validateForm() {
    // Initialize a variable to track if any validation checks fail
    let valid = true;

    // Validate the name
    if (!validateName()) {
        valid = false;
    }

    // Validate the contact
    if (!validateContact()) {
        valid = false;
    }

    // Validate the address (you can add more specific checks if needed)
    if (document.getElementById("UserAddress").value.trim() === "") {
        document.getElementById("addressError").innerText = "Address cannot be empty.";
        valid = false;
    } else {
        document.getElementById("addressError").innerText = ""; // Clear any previous error message
    }

    // Return the final validation status
    return valid;
}

// Add event listeners to validate fields on input events
document.getElementById("UserName").addEventListener("input", validateName);
document.getElementById("UserContact").addEventListener("input", validateContact);

```

#registervalid.js

```

document.addEventListener('DOMContentLoaded', function() {

    // Real-time validation for the name field
    document.getElementById('name').addEventListener('input', validateName);

    // Real-time validation for the email field
    document.getElementById('email').addEventListener('input', validateEmail);

    // Real-time validation for the password field
    document.getElementById('password').addEventListener('input', validatePassword);

    // Real-time validation for the confirm password field
    document.getElementById('confirm_password').addEventListener('input', validateConfirmPassword);

    // Real-time validation for the contact field
    document.getElementById('contact').addEventListener('input', validateContact);

    // Real-time validation for the address field
    document.getElementById('address').addEventListener('input', validateAddress);

});

function validateName() {
    var name = document.getElementById('name').value;
    var namePattern = /^[A-Za-z\s]{3,}$/;
    if(!namePattern.test(name)) {
        document.getElementById('nameError').textContent = 'Invalid name. Only alphabets are allowed and minimum 3 letters.';
    } else {
        document.getElementById('nameError').textContent = "";
    }
}

function validateEmail() {
    var email = document.getElementById('email').value;
    var emailPattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}$/;
    if(!emailPattern.test(email)) {
        document.getElementById('emailError').textContent = 'Invalid email format.';
    } else {
        document.getElementById('emailError').textContent = "";
    }
}

```

```

}

function validatePassword() {
    var password = document.getElementById('password').value;
    if(password.length < 6) {
        document.getElementById('passwordError').textContent = 'Password should be at least 6 characters long.';
    } else {
        document.getElementById('passwordError').textContent = "";
        // Trigger confirmation password validation as the password might have changed
        validateConfirmPassword();
    }
}

function validateConfirmPassword() {
    var password = document.getElementById('password').value;
    var confirmPassword = document.getElementById('confirm_password').value;
    if(password !== confirmPassword) {
        document.getElementById('confirmPasswordError').textContent = 'Passwords do not match.';
    } else {
        document.getElementById('confirmPasswordError').textContent = "";
    }
}

function validateContact() {
    var contact = document.getElementById('contact').value;
    var contactPattern = /^\d{8,10}$/;
    if(!contactPattern.test(contact)) {
        document.getElementById('contactError').textContent = 'Invalid contact number. It should be 8 to 10 digits long.';
    } else {
        document.getElementById('contactError').textContent = "";
    }
}

function validateAddress() {
    var address = document.getElementById('address').value;
    if(address.trim() === "") {
        document.getElementById('addressError').textContent = 'Address cannot be empty.';
    } else {
        document.getElementById('addressError').textContent = "";
    }
}

```

#checkout.php

```

<?php
session_start();
include 'dbconn.php';

$user_id = $_SESSION['user_id'];

// Fetch items from cart
$stmt = $conn->prepare("SELECT * FROM carts WHERE user_id = ?");
$stmt->execute([$user_id]);
$cartItems = $stmt->fetchAll();

$cart_id = getNextCartID($user_id);

// Email header for HTML email content
$headers = "MIME-Version: 1.0" . "\r\n";
$headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";
$headers .= 'From: f31ee@localhost' . "\r\n" .
    'Reply-To: f31ee@localhost' . "\r\n" .
    'X-Mailer: PHP/' . phpversion();

// Start with the email content
$emailContent = '<html><body>';
$emailContent .= '<h1>Order Confirmation - Cart ID: ' . htmlspecialchars($cart_id) . '</h1>';
$emailContent .= '<p>Thank you for your order! Here are the details:</p>';
$emailContent .= '<table border="1" style="width:100%; border-collapse: collapse;">';
$emailContent .= '<tr><th>Item</th><th>Quantity</th><th>Price</th></tr>';

$totalPrice = 0;

// Insert items into orders table and remove from cart

```

```

foreach($cartItems as $item) {
    $stmt = $conn->prepare("INSERT INTO orders ( user_id, cart_id, product_name, description, price, quantity, img_src) VALUES (?, ?, ?, ?, ?, ?, ?)");
    $stmt->execute([$user_id, $cart_id, $item['product_name'], $item['description'], $item['price'], $item['quantity'], $item['img_src']]);
    $stmt = $conn->prepare("DELETE FROM carts WHERE user_id = ? AND product_name = ?");
    $stmt->execute([$user_id, $item['product_name']]);

    //Email content
    $itemTotal = $item['price'] * $item['quantity'];
    $totalPrice += $itemTotal; // Add to the total price

    $emailContent .= '<tr>';
    $emailContent .= '<td>' . htmlspecialchars($item['product_name']) . '</td>';
    $emailContent .= '<td>' . htmlspecialchars($item['quantity']) . '</td>';
    $emailContent .= '<td>$' . number_format($itemTotal, 2) . '</td>';
    $emailContent .= '</tr>';
}

$emailContent .= '<tr><td colspan="2" style="text-align:right;">Total Price:</td>';
$emailContent .= '<td>$' . number_format($totalPrice, 2) . '</td></tr>';
$emailContent .= '</table>';
$emailContent .= '</body></html>';

function getNextCartID($user_id) {
    global $conn; // Ensure the database connection is accessible within this function

    // Fetch the highest cart_id for this user
    $stmt = $conn->prepare("SELECT MAX(cart_id) as maxCartID FROM orders WHERE user_id = ?");
    $stmt->execute([$user_id]);
    $row = $stmt->fetch(PDO::FETCH_ASSOC);

    // If no previous cart exists, return 1. Otherwise, return the next available cart_id.
    if ($row['maxCartID']) {
        return $row['maxCartID'] + 1;
    } else {
        return 1;
    }
}

$emailContent .= '<tr><td colspan="2" style="text-align:right;">Total Price:</td>';
$emailContent .= '<td>$' . number_format($totalPrice, 2) . '</td></tr>';
$emailContent .= '</table>';
$emailContent .= '</body></html>';

// Recipient email
$to = 'f32ee@localhost';
$subject = 'Order Confirmation';

// Send the email
if(mail($to, $subject, $emailContent, $headers)) {
    echo "Email sent successfully";
} else {
    echo "Email sending failed";
}

// Optionally, redirect to a confirmation page or the homepage after checkout
header("Location: ../home.php");
exit;
?>

```

#dbconn.php

```

<?php
$host = 'localhost';
$db_name = 'henrys_burgers';
$username = 'root'; // MySQL username
$password = ''; // MySQL password

try {
    $conn = new PDO("mysql:host=$host;dbname=$db_name", $username, $password);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}

```

?>

#login.php

```
<?php
session_start();
include 'dbconn.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];

    $sql = "SELECT * FROM users WHERE email = ?";
    $stmt = $conn->prepare($sql);
    $stmt->execute([$email]);
    $user = $stmt->fetch();

    $sql = "SELECT * FROM user_details WHERE email = ?";
    $stmt = $conn->prepare($sql);
    $stmt->execute([$email]);
    $user_details = $stmt->fetch();

    if ($user && password_verify($password, $user["password"])) {
        $_SESSION["user_id"] = $user["id"];
        $_SESSION["user_name"] = $user_details["name"];
        $_SESSION["user_email"] = $user_details["email"];
        // Check if the logged-in user is an admin
        if ($email === 'admin@admin.com') {
            header('Location: ../admin.php');
        } else {
            header('Location: ../profile.php');
        }
        exit(); // Always exit after a redirect to stop further script execution.
    } else {
        // Redirect back to the login page with an error message.
        header('Location: ../loginpage.php?error=Incorrect email or password!');
        exit(); // Always exit after a redirect to stop further script execution.
    }
}
?>
```

#register.php

```
<?php
include 'dbconn.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $email = $_POST["email"];
    $password = password_hash($_POST["password"], PASSWORD_DEFAULT);
    $contact = $_POST["contact"];
    $address = $_POST["address"];

    // Check if email already exists
    $sql_email_check = "SELECT * FROM users WHERE email = ?";
    $stmt_email_check = $conn->prepare($sql_email_check);
    $stmt_email_check->execute([$email]);
    $email_exist = $stmt_email_check->fetchColumn();

    if ($email_exist) {
        // Email already exists in the database
        echo "<script>alert('Email already exists. Please use a different email.');" . window.history.go(-1) . "</script>";
    } else {
        // Email doesn't exist, proceed with registration
        // Insert user credentials into users table
        $sql_users = "INSERT INTO users (email, password) VALUES (?, ?)";
        $stmt_users = $conn->prepare($sql_users);
        $stmt_users->execute([$email, $password]);

        // Insert user details into user_details table
        $sql_details = "INSERT INTO user_details (email, name, contact, address) VALUES (?, ?, ?, ?)";
```

```

$stmt_details = $conn->prepare($sql_details);
$stmt_details->execute([$email, $name, $contact, $address]);

// Redirect to login page with success message
echo "<script>alert('Account created successfully!'); window.location.href='../loginpage.php';</script>";
//header('Location: ../loginpage.php');
exit();
}
}
?>

```

#remove_item.php

```

<?php
session_start();
include 'dbconn.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $item_id = $_POST['item_id'];

    if (isset($_SESSION['user_id'])) {
        $user_id = $_SESSION['user_id'];
        $stmt = $conn->prepare("DELETE FROM carts WHERE id = ? AND user_id = ?");
        $stmt->execute([$item_id, $user_id]);
    }

    header('Location: ../cart.php');
    exit();
}
?>

```

#update_quantity.php

```

<?php
session_start();
include 'dbconn.php';

// Check if the user is logged in
if(isset($_SESSION['user_id']) && isset($_POST['item_id'])) {
    $userId = $_SESSION['user_id'];
    $itemId = $_POST['item_id'];

    // Start a transaction
    $conn->beginTransaction();

    try {
        if(isset($_POST['increase'])) {
            $sql = "UPDATE carts SET quantity = quantity + 1 WHERE id = ? AND user_id = ?";
            $stmt = $conn->prepare($sql);
            $stmt->execute([$itemId, $userId]);
        } elseif(isset($_POST['decrease'])) {
            // First, get the current quantity to check if it's 1
            $sql = "SELECT quantity FROM carts WHERE id = ? AND user_id = ?";
            $stmt = $conn->prepare($sql);
            $stmt->execute([$itemId, $userId]);
            $currentQuantity = $stmt->fetchColumn();

            if ($currentQuantity <= 1) {
                // If the quantity is 1, delete the item
                $sql = "DELETE FROM carts WHERE id = ? AND user_id = ?";
                $stmt = $conn->prepare($sql);
                $stmt->execute([$itemId, $userId]);
            } else {
                // If the quantity is more than 1, decrease it
                $sql = "UPDATE carts SET quantity = quantity - 1 WHERE id = ? AND user_id = ?";
                $stmt = $conn->prepare($sql);
                $stmt->execute([$itemId, $userId]);
            }
        } else {
            // Redirect back with an error message if action is not defined
            header('Location: ../cart.php?error=invalid_action');
            exit();
        }
    }
}

```

```

// Commit the transaction
$conn->commit();

// Redirect back to the cart page after updating or deleting
$itemIndex = $_POST['item_index'] ?? 'top'; // Fallback to 'top' if not set
header('Location: ../cart.php');
exit;
} catch (Exception $e) {
    // An exception has been caught, rollback the transaction
    $conn->rollback();
    // Redirect with an error message
    header('Location: ../cart.php?error=exception');
    exit;
}
} else {
    // Redirect back with an error message if user_id or item_id is not set
    header('Location: ../cart.php?error=invalid_request');
    exit;
}
?>

```

#henrys_burgers.sql

```

-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Nov 07, 2023 at 04:55 AM
-- Server version: 10.4.24-MariaDB
-- PHP Version: 8.1.6

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `henrys_burgers`
--

--
-- Table structure for table `carts`
--

CREATE TABLE `carts` (
  `id` int(11) NOT NULL,
  `user_id` int(11) NOT NULL,
  `product_name` varchar(255) DEFAULT NULL,
  `description` text DEFAULT NULL,
  `price` decimal(10,2) DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL,
  `img_src` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Table structure for table `menu_items`
--

CREATE TABLE `menu_items` (
  `id` int(11) NOT NULL,
  `name` varchar(255) NOT NULL,
  `description` text NOT NULL,
  `price` decimal(10,2) NOT NULL,
  `category` varchar(50) NOT NULL,

```

```
`image` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Dumping data for table `menu_items`
--
```

```
INSERT INTO `menu_items` (`id`, `name`, `description`, `price`, `category`, `image`) VALUES
(1, 'Beef Burger', 'Just a normal beef burger', '7.00', 'burgers', 'beefburger.jpg'),
(2, 'Chicken Burger', 'Just a normal chicken burger', '6.99', 'burgers', 'chickenburger.jpg'),
(3, 'Lamb Burger', 'Just a normal lamb burger', '7.99', 'burgers', 'lambburger.jpg'),
(4, 'Fish Burger', 'Just a normal fish burger', '5.99', 'burgers', 'fishburger.jpg'),
(5, 'Croquette Burger', 'Just a normal croquette burger', '5.99', 'burgers', 'croquetteburger.jpg'),
(6, 'Chicken Katsu Burger', 'Chicken katsu with cabbages dressed in our signature sauce', '4.99', 'burgers', 'katsuburger.jpg'),
(7, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 'sides', 'fries.jpg'),
(8, 'Croquette Pie', 'Hot and crunchy croquette', '3.99', 'sides', 'pie.jpg'),
(9, 'Clam Chowder', 'Creamy goodness', '4.99', 'sides', 'clamchowder.jpg'),
(10, 'Green Salad', 'Going GREEN', '2.99', 'sides', 'salad.jpg'),
(11, 'Cola', 'Refreshing cola to beat the heat', '2.99', 'drinks', 'cola.jpg'),
(12, 'Fanta Grape', 'Grape flavored soft drink with a crisp taste', '2.99', 'drinks', 'fantagrape.jpg'),
(13, 'Ice Lemon Tea', 'Ice-cold tea with lemons', '2.99', 'drinks', 'icelemontea.jpg'),
(14, 'Bottled Water', 'Nothing beats H2O', '1.00', 'drinks', 'water.jpg');
```

```
--
-- Table structure for table `orders`
--
```

```
CREATE TABLE `orders` (
  `order_id` int(11) NOT NULL,
  `user_id` int(11) DEFAULT NULL,
  `cart_id` int(11) DEFAULT NULL,
  `product_name` varchar(255) DEFAULT NULL,
  `description` text DEFAULT NULL,
  `price` decimal(10,2) DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL,
  `img_src` varchar(255) DEFAULT NULL,
  `order_date` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Dumping data for table `orders`
--
```

```
INSERT INTO `orders` (`order_id`, `user_id`, `cart_id`, `product_name`, `description`, `price`, `quantity`, `img_src`, `order_date`) VALUES
(1, 1, 1, 'Beef Burger', 'Just a normal beef burger', '7.99', 5, 'pics/beefburger.jpg', '2023-11-05 14:09:14'),
(2, 1, 1, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 5, 'pics/fries.jpg', '2023-11-05 14:09:14'),
(3, 1, 1, 'Cola', 'Refreshing cola to beat the heat', '2.99', 5, 'pics/cola.jpg', '2023-11-05 14:09:14'),
(4, 3, 1, 'Beef Burger', 'Just a normal beef burger', '7.99', 2, 'pics/beefburger.jpg', '2023-11-06 05:00:59'),
(5, 3, 2, 'Beef Burger', 'Just a normal beef burger', '7.00', 2, 'pics/beefburger.jpg', '2023-11-06 05:02:05'),
(6, 3, 3, 'Beef Burger', 'Just a normal beef burger', '7.00', 1, 'pics/beefburger.jpg', '2023-11-06 05:04:32'),
(7, 3, 4, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 2, 'pics/fries.jpg', '2023-11-06 05:41:28'),
(8, 4, 1, 'Chicken Burger', 'Just a normal chicken burger', '6.99', 1, 'pics/chickenburger.jpg', '2023-11-06 12:41:39'),
(9, 4, 1, 'Fish Burger', 'Just a normal fish burger', '5.99', 1, 'pics/fishburger.jpg', '2023-11-06 12:41:39'),
(10, 4, 2, 'Chicken Burger', 'Just a normal chicken burger', '6.99', 6, 'pics/chickenburger.jpg', '2023-11-06 13:30:34'),
(11, 4, 2, 'Fish Burger', 'Just a normal fish burger', '5.99', 4, 'pics/fishburger.jpg', '2023-11-06 13:30:34'),
(12, 4, 2, 'Croquette Burger', 'Just a normal croquette burger', '5.99', 3, 'pics/croquetteburger.jpg', '2023-11-06 13:30:34'),
(13, 4, 3, 'Chicken Burger', 'Just a normal chicken burger', '6.99', 2, 'pics/chickenburger.jpg', '2023-11-06 13:36:20'),
(14, 4, 3, 'Cola', 'Refreshing cola to beat the heat', '2.99', 1, 'pics/cola.jpg', '2023-11-06 13:36:20'),
(15, 4, 4, 'Croquette Pie', 'Hot and crunchy croquette', '3.99', 1, 'pics/pie.jpg', '2023-11-06 13:41:36'),
(16, 4, 4, 'Green Salad', 'Going GREEN', '2.99', 1, 'pics/salad.jpg', '2023-11-06 13:41:36'),
(17, 4, 5, 'Clam Chowder', 'Creamy goodness', '4.99', 1, 'pics/clamchowder.jpg', '2023-11-06 13:44:52'),
(18, 4, 6, 'Bottled Water', 'Nothing beats H2O', '1.00', 1, 'pics/water.jpg', '2023-11-06 13:46:12'),
(19, 4, 7, 'Beef Burger', 'Just a normal beef burger', '7.99', 1, 'pics/beefburger.jpg', '2023-11-06 13:47:41'),
(20, 4, 7, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 1, 'pics/fries.jpg', '2023-11-06 13:47:41'),
(21, 4, 7, 'Croquette Pie', 'Hot and crunchy croquette', '3.99', 1, 'pics/pie.jpg', '2023-11-06 13:47:41'),
(22, 4, 8, 'Croquette Pie', 'Hot and crunchy croquette', '3.99', 1, 'pics/pie.jpg', '2023-11-06 13:54:33'),
(23, 4, 8, 'Cola', 'Refreshing cola to beat the heat', '2.99', 1, 'pics/cola.jpg', '2023-11-06 13:54:33'),
(24, 4, 8, 'Lamb Burger', 'Just a normal lamb burger', '7.99', 1, 'pics/lambburger.jpg', '2023-11-06 13:54:33'),
(25, 4, 9, 'Chicken Katsu Burger', 'Chicken katsu with cabbages dressed in our signature sauce', '4.99', 9, 'pics/katsuburger.jpg', '2023-11-06 14:19:40'),
(26, 4, 10, 'Fish Burger', 'Just a normal fish burger', '5.99', 1, 'pics/fishburger.jpg', '2023-11-06 15:13:58'),
(27, 4, 10, 'Croquette Burger', 'Just a normal croquette burger', '5.99', 1, 'pics/croquetteburger.jpg', '2023-11-06 15:13:58'),
(28, 4, 10, 'Chicken Burger', 'Just a normal chicken burger', '6.99', 1, 'pics/chickenburger.jpg', '2023-11-06 15:13:58'),
(29, 4, 10, 'Beef Burger', 'Just a normal beef burger', '7.99', 1, 'pics/beefburger.jpg', '2023-11-06 15:13:58');
```

```
(30, 4, 10, 'Chicken Katsu Burger', 'Chicken katsu with cabbages dressed in our signature sauce', '4.99', 1, 'pics/katsuburger.jpg', '2023-11-06 15:13:58'),
(31, 4, 10, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 5, 'pics/fries.jpg', '2023-11-06 15:13:58'),
(32, 4, 10, 'Fanta Grape', 'Grape flavored soft drink with a crisp taste', '2.99', 3, 'pics/fantagrape.jpg', '2023-11-06 15:13:58'),
(33, 4, 10, 'Ice Lemon Tea', 'Ice-cold tea with lemons', '2.99', 3, 'pics/icelemontea.jpg', '2023-11-06 15:13:58'),
(34, 3, 5, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 1, 'pics/fries.jpg', '2023-11-07 01:39:57'),
(35, 3, 5, 'Beef Burger', 'Just a normal beef burger', '7.99', 1, 'pics/beefburger.jpg', '2023-11-07 01:39:57'),
(36, 8, 1, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 1, 'pics/fries.jpg', '2023-11-07 02:05:39'),
(37, 3, 6, 'French Fries', 'Freshly fried fries you will never forget', '3.99', 2, 'pics/fries.jpg', '2023-11-07 03:46:49');
```

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

```
--
-- Table structure for table `users`
--
```

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Dumping data for table `users`
--
```

```
INSERT INTO `users` (`id`, `email`, `password`) VALUES
(1, 'admin@admin.com', '$2y$10$1.hbwleouuBC8wsEOAntfOhvJZHhjWWkIQl8m6TzCCUGG/J7dZgNm'),
(2, 'wang@gmail.com', '$2y$10$BVuc/8/8D8fOFbHuBgd4yuS5tOjO865OZQigFIDwUsQex2MjM2HM.'),
(3, 'alex@alex.com', '$2y$10$ZhtMUIKigHCtCYoOjF6PHuw7oThgyX04y2QmpbXOKBPjmZyqVUe6u'),
(4, 'maba@gmail.com', '$2y$10$RmKcS8G4UQOxQDknM8qT.zfgEEVjAfAsvKVImLBsGYTWerrYzZC'),
(5, 'ben@gmail.co.uk', '$2y$10$2W9fvx.zVoosKRmQrAuzOeqWUI0edwtmDkl2ykdDxbW.v27JimHbS'),
(6, 'we@gmail', '$2y$10$qlpeEw1a3KtxEq8P.yzfQoO2zy8hE4tqi3MckhLeNaYLDNROGnE46'),
(7, 'test@gmail.co.uk.sg', '$2y$10$rkAwGqt6RRzITCjfbV2LbOYSpuIAI1sCXqPJGnbnAvTczJu8O5Dg2'),
(8, 'david@david.com', '$2y$10$WmbcLgcqHUH.ELerBSng.ubCrvzO0UqlfwEypMF2pODWYwWiKqSH.'),
(9, 'john@gmail.com', '$2y$10$e2gPRBV.9tEHba4wpxCi..yva.19CfYG0Ky0Nr9rMXlkadobAPALS');
```

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

```
--
-- Table structure for table `user_details`
--
```

```
CREATE TABLE `user_details` (
  `id` int(11) NOT NULL,
  `email` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  `contact` varchar(50) NOT NULL,
  `address` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
-- Dumping data for table `user_details`
--
```

```
INSERT INTO `user_details` (`id`, `email`, `name`, `contact`, `address`) VALUES
(1, 'admin@admin.com', 'admin', '00000000', 'admin'),
(2, 'wang@gmail.com', 'Henry Chen', '97150650', 'ewewewewe'),
(3, 'alex@alex.com', 'Alex', '11111111', '11111111'),
(4, 'maba@gmail.com', 'Ameerul', '98765432', '123 Nanyang Road, #03-248'),
(5, 'ben@gmail.co.uk', 'Ben', '88889999', '312 Oxley Road'),
(6, 'we@gmail', 'Frank', '132131313', '312 Oxley Road'),
(7, 'test@gmail.co.uk.sg', 'Sam', '98765432', '567 Oxley Road'),
(8, 'david@david.com', 'David', '12345678', 'david123'),
(9, 'john@gmail.com', 'john', '12345678', 'david123');
```

```
--
-- Indexes for dumped tables
--
```

```
--
-- Indexes for table `carts`
--
```

```
ALTER TABLE `carts`
  ADD PRIMARY KEY (`id`);
```



```

--
-- Indexes for table `menu_items`
--
ALTER TABLE `menu_items`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `orders`
--
ALTER TABLE `orders`
  ADD PRIMARY KEY (`order_id`);

--
-- Indexes for table `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `email` (`email`);

--
-- Indexes for table `user_details`
--
ALTER TABLE `user_details`
  ADD PRIMARY KEY (`id`),
  ADD KEY `email` (`email`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `carts`
--
ALTER TABLE `carts`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=45;

--
-- AUTO_INCREMENT for table `menu_items`
--
ALTER TABLE `menu_items`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

--
-- AUTO_INCREMENT for table `orders`
--
ALTER TABLE `orders`
  MODIFY `order_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=38;

--
-- AUTO_INCREMENT for table `users`
--
ALTER TABLE `users`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

--
-- AUTO_INCREMENT for table `user_details`
--
ALTER TABLE `user_details`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=10;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `user_details`
--
ALTER TABLE `user_details`
  ADD CONSTRAINT `user_details_ibfk_1` FOREIGN KEY (`email`) REFERENCES `users` (`email`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

#styles.css

```
* {
  box-sizing: border-box;
}

html {
  height: auto;
  background-color: #ffa64d;
}

body {
  margin : 0;
  font-family: Arial, sans-serif;
  background-color: #eab676;
  height: auto;
  min-width: 1200px;
}

header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 20px;
  background-color: #ffa64d;
}

.logo-container {
  flex: 1;
  font-size: 24px;
  font-weight: bold;
  color: white;
}

.login-container, .user-name-container {
  flex: 1;
  text-align: right;
}

.login-container a {
  text-decoration: none;
  color: white;
  padding: 8px 16px;
  background-color: #ffa64d;
  border: 1px solid white;
  border-radius: 5px;
}

.session-container {
  display: flex;
  flex-direction: column;
  align-items: flex-end; /* To align the items to the right */
}

nav {
  display: flex;
  justify-content: space-around;
  align-items: center;
  padding: 10px 0;
  background-color: #ffa64d;
  font-size: 1.2em;
}

nav a {
  text-decoration: none;
  color: white;
  padding: 8px 16px;
  transition: background-color 0.3s;
}

nav a:hover {
  background-color: white;
  color: #ffa64d;
}
```

```

}

.main-home {
  padding: 20px;
  text-align: center;
  height: 100%;
}

.logo-image {
  width: 500px;
  height: auto;
}

.restaurant-image {
  width: 80%;
  height: 60%;
  margin-top: 20px;
}

.main-about{
  padding: 20px;
  height: 100%;
}

.about-header {
  display: flex;
  flex-direction: row; /* Items will be placed from left to right */
  align-items: center; /* Vertically center align items */
  gap: 20px; /* Space between items */
}

.text-header {
  text-align: center;
  font-size: 1.5em;
  flex: 1;
  display: block; /* This is default, but added for clarity */
}

.about-content {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  margin-top: 20px;
}

.about-text {
  text-align: justify;
  font-size: 20px;
  padding: 10px;
}

.about-image {
  flex:2;
  width: 200px;
  height: 600px;
  /*flex-shrink: 0; /* Prevents shrinking */
  float: right;
}

.menu-content .menu-item {
  text-align: center; /* Centers the text horizontally */
}

.menu-content .menu-item .menu-image {
  display: block; /* Makes the image a block element */
  margin-left: auto; /* Centers the image if it doesn't take up full width */
  margin-right: auto;
}

/* Styles for each menu item */
.menu-item {
  width: calc(33.33% - 10px); /* Gives a little space between items */
  box-sizing: border-box;

```

```

    flex-direction: row;
    align-items: center;
    margin-bottom: 20px;
        margin-top: 20px;
        font-size: 1.5em;
}

/* Styles for each menu item's image */
.menu-image {
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
        border: 2px solid black;
        width: 100%;
    height: 70%;
        border-radius: 20px;
    object-fit: cover; /* This ensures the image covers the area without distortion */
}

/* Ensure main content uses flexbox to place aside and section side by side */
.menu-content {
    display: flex;
        flex-direction: row;
        background-color: #E4AB7C;
}

/* Style for the menu items section */
.menu {
    flex: 6; /* take up 3 fractions of the available space, */
    display: flex;
    flex-wrap: wrap; /* Allows menu items to wrap to the next line if they don't fit */
    justify-content: space-between; /* Adds some space between items */
}

/* Style for the aside (left-side menu) */
aside {
    flex: 1; /* take up 1 fraction of the available space */
    margin-right: 20px; /* Adds some space between the aside and the menu items */
    background-color: #FFFDEE;
    flex-grow: 1; /* Allow the <aside> to grow and take available space */
    flex-shrink: 0; /* Prevent the <aside> from shrinking smaller than its content */
    display: grid;
        align-content: start;
        place-items: center;
        text-align: center;
}

aside ul li {
    margin-bottom: 20px; /* Adjust value as needed */
    margin-top: 20px;
}

/* Target the <ul> within the <aside> */
aside ul {
    list-style-type: none; /* Removes the bullet points */
    padding: 0; /* Removes default padding */
    margin: 10px;
}

/* Style each list item and the links within them */
aside li {
    margin-bottom: 10px; /* Adds some space between items */
}

aside li a {
    font-size: 1.5em; /* Increases the font size to make it bigger */
    text-decoration: none; /* Removes the underline from the links */
    color: inherit; /* Uses the same color as the surrounding text */
}

aside li a:hover {
    color: #007BFF; /* Changes the color when you hover over the link */
}

```

```

aside li .active {
  color: #ff0000; /* Red text for the active link */
  font-weight: bold; /* Bold text for the active link */
  /* Additional styles as required */
}

/*toast message*/
.toast {
  visibility: hidden;
  max-width: 50%;
  margin-left: 0px;
  background-color: #333;
  color: #fff;
  text-align: center;
  border-radius: 2px;
  padding: 16px;
  position: fixed;
  z-index: 1;
  left: 50%;
  bottom: 30px;
  font-size: 17px;
  transition: visibility 0.5s, opacity 0.5s linear;
  opacity: 0;
}

.toast.show {
  visibility: visible;
  opacity: 1;
}

/*cart table*/
.cart-table {
  width: 75%;
  border-collapse: collapse;
  margin-top: 20px;
}

.cart-table th, td {
  border: 1px solid #ddd;
  padding: 8px 12px;
  text-align: center;
}

.product-desc {
  display: flex;
  align-items: center;
  text-align: left;
}

.cart-table th:nth-of-type(4) {
  border: none;
}

.cart-table td:nth-of-type(4) {
  border: none;
}

.remove-button {
  padding: 10px;
  border-radius: 5px;
  color: #fff;
  border: 1px solid #000;
  font-size: 1em;

  background-color: orange;
}

.product-image {
  width: 250px;
  height: 200px;
  margin-right: 70px;
}

.main-cart, .orderhistory-cart{

```

```

display: flex;
flex-direction: column;
    height: 100%;
    background-color: #E4AB7C;
    padding-bottom: 10px;
    align-items: center;
}

.cart-header {
display: flex;
justify-content: space-between;
width: 100%;
padding: 10px 0;
border-bottom: 1px solid black;
}

.menu h2 {
flex-basis: 100%;
text-align: center;
}

.cart-item {
display: flex;
justify-content: space-between;
width: 100%;
padding: 10px 0;
border-bottom: 1px solid black;
}

.cart-item img {
flex-basis: 33%;
max-width: 100%;
height: auto;
}

.item-text, .item-quantity, .item-price {
flex-basis: 33%;
text-align: center;
}

.checkout-button {
margin-top: 20px;
padding: 10px 20px;
background-color: orange;
cursor: pointer;
color: white;
border: 1px solid black;
align-self: flex-end;
}

.login-page, .profile-container {
display: flex;
flex-direction: column;
align-items: center;
background-color: #E4AB7C;
padding: 20px;
}

.profile-container {
height: 550px;
}

.login-page {
height: 650px;
}

.login-page h2 {
padding: 10px 20px;
border-radius: 5px;
}

.login-page label {
width: 100%;
margin-top: 20px;
text-align: left;
}

```

```

}

.login-page input {
  width: 100%;
  padding: 10px;
  margin-top: 10px;
  margin-bottom: 5px;
  border: 1px solid black;
  border-radius: 5px;
}

.buttons {
  display: flex;
  justify-content: center;
  width: 100%;
  margin-top: 20px;
}

button, .home-button, .order-history-button{
  padding: 10px 20px;
  border: 1px solid black;
  border-radius: 5px;
  background-color: orange;
  color: white;
  cursor: pointer;
  margin: 10px 10px;
  font-size: 20px;
}

.home-button, .order-history-button {
  text-decoration: none;
  display: flex;
  align-items: center;
  justify-content: center;
}

input[readonly] {
  background-color: transparent; /* make the background transparent */
  border: none; /* remove the border */
  outline: none; /* remove the focus outline */
  cursor: default; /* set the cursor to default to indicate it's not editable */
}

.sign-in {
  margin-top: 20px;
}

/* Registration Form Styles */
.registration-form {
  max-width: 500px;
  margin: 50px auto; /* Center the form */
  padding: 20px;
  border: 1px solid #000;
  background-color: #F5F5DC;
}

.registration-form h2 {
  text-align: center;
  background-color: orange;
  padding: 10px 0;
  border: 1px solid black;
  width: 50%;
  margin: 0 auto; /* This will center the bar */
}

.registration-form label {
  display: block;
  margin: 20px 10px 10px;
}

.registration-form input {
  width: 90%;
  padding: 10px;
}

```

```

border: 1px solid #000;
margin: 20px 10px 10px;
display: flex;
justify-content: center; /* Horizontally centers the contents */
}

.error-text {
color: red;
font-size: 0.8em;
margin: 10px 0px;
}

//Modal CSS
/* Modal Base Styling */
.modal {
display: none; /* Hidden by default */
position: fixed;
top: 50%;
left: 50%;
width: 50%;
height: auto;
z-index: 10000; /* Higher z-index than other styles */
transform: translate(-50%, -50%); /* Center the modal */
}

.modal-content {
position: relative;
background-color: #fefefe;
margin: auto;
padding: 20px;
width: 80%;
box-shadow: 0 4px 8px 0 rgba(0,0,0,0.2);
display: flex;
flex-direction: column;
justify-content: center;
align-items: center;
}

.modal-content h2 {
font-size: 30px; /* Adjust this value as needed */
}

.modal-content p {
font-size: 30px; /* Adjust this value as needed */
}

#priceText {
font-size: 32px; /* Adjust this value as needed */
font-weight: bold;
}

/* Buttons */
.close, .add-to-cart {
color: white;
padding: 10px 20px;
text-align: center;
text-decoration: none;
display: inline-block;
position: absolute;
}

.close {
background-color: red;
top: 10px;
right: 10px;
}

.add-to-cart {
background-color: green;
bottom: 20px;
right: 20px;
margin-top: auto;
}

#myModal {
display: none; /* Hidden by default */
position: fixed;
z-index: 10000; /* Higher z-index than other styles */
}

```



```

top: 50%;
left: 50%;
width: 50%;
transform: translate(-50%, -50%); /* Center the modal */
}
#modal-image {
width: 500px;
height: 400px;
object-fit: cover;
display: block;
margin: auto;
border-radius: 10px;
}
.notification {
background-color: #f4f4f4;
padding: 10px;
border: 1px solid #ddd;
margin: 10px 0;
color: #333;
}

.quantity-input {
display: flex;
align-items: center;
gap: 10px; /* spacing between label and input */
font-size: 1.5em;
}

.quantity-input input {
width: 45px; /* adjust width as needed */
text-align: center;
font-size: 1em;
}

.profile-form {
width: 700px;
height: 400px;
font-size: 1.2em;
display: flex;
flex-direction: column;
align-items: flex-start;
}

.input-group { /* This is our new class for each row */
display: flex;
align-items: center;
margin-bottom: 20px;
}

.profile-form label {
flex: 0 0 150px; /* Adjust the 150px to set width of the label */
text-align: right;
padding-right: 30px;
}

.profile-form input {
flex: 1; /* This makes input take the remaining space */
padding: 10px 10px 10px 20px;
border-radius: 5px;
font-size: 1.2em;
width: 600px; /* Setting a fixed width */
box-sizing: border-box; /* Ensure padding is included in the width */
overflow: visible;
}

.error-text {
margin-left: 10px;
color: red;
}

.orderhistory-table, .modify-table, .report-table {
width: 90%;
border-collapse: collapse;
margin-top: 20px;
}

```

```

.orderhistory-cart {
    background-color: transparent;
}

/*-----Slideshow-----*/
/* Slideshow container */
.slideshow-container {
    position: relative;
    margin: auto;
}

/* Set the size of the slideshow container */
.slideshow-container {
    width: 400px;
    height: 100%; /* Set a height for your container */
    position: relative; /* Used for positioning the navigation arrows */
    margin: auto; /* This will center your slideshow on the page */
}

/* Make the images fill the container without stretching */
.slideshow-container img {
    width: 100%; /* This makes the image fill the width of the container */
    height: auto; /* This maintains the aspect ratio */
    object-fit: cover; /* This makes sure that the image covers the area, cropping if necessary */
    border-radius: 5px;
}

/* Hide the images by default */
.mySlides {
    display: none;
}

/* Next & previous buttons */
.prev, .next {
    cursor: pointer;
    position: absolute;
    top: 55%; /* Center vertically */
    transform: translateY(-50%); /* Align the center of the button with the center of the container */
    width: auto;
    padding: 16px;
    color: white;
    font-weight: bold;
    font-size: 18px;
    transition: 0.6s ease;
    border-radius: 0 3px 3px 0;
    user-select: none;
    background-color: rgba(0,0,0,0.5); /* Add some transparency for aesthetics */
}

.numbertext {
    margin: 10px 0px;
}

/* Position the "next button" to the right */
.next {
    right: 0;
    border-radius: 3px 0 0 3px;
}

/* Position the "prev button" to the left */
.prev {
    left: 0;
    border-radius: 3px 0 0 3px;
}

/* On hover, add a black background color with a little bit see-through */
.prev:hover, .next:hover {
    background-color: rgba(0,0,0,0.8);
}

/* Fading animation */
.fade {
    animation-name: fade;

```

```

        animation-duration: 1.5s;
    }

    @keyframes fade {
        from {opacity: .4}
        to {opacity: 1}
    }

    /*-----End of Slideshow-----*/
    .modify-table ,.report-table {
        margin-left: auto;
        margin-right: auto;
        margin-bottom: 20px;
    }

    .modify-price, .sales-report {
        text-align: center;
        width: 50%;
    }

    .modify-price{
        margin-bottom: 30px;
    }

    footer {
        background-color: #333; /* Dark background */
        color: white; /* Text color */
        padding: 20px 0; /* Some padding */
        text-align: center;
    }

    .footer-container {
        display: flex;
        justify-content: space-around; /* Distribute space between sections */
        max-width: 1200px; /* Max width of the footer */
        margin: 0 auto; /* Center the footer */
        text-align: left;
    }

    .footer-section h4 {
        margin-bottom: 15px; /* Space below the section titles */
    }

    .footer-section ul {
        list-style: none; /* Remove default list styling */
        padding: 0; /* Remove default padding */
    }

    .footer-section ul li {
        margin-bottom: 10px; /* Space between list items */
    }

    /* Responsive adjustments */
    @media (max-width: 768px) {
        .footer-container {
            flex-direction: column; /* Stack the sections on small screens */
            align-items: center; /* Center the items */
        }

        .footer-section ul li a {
            color: white; /* Anchor text color */
            text-decoration: none; /* Remove underline from links */
            transition: color 0.3s ease; /* Smooth color transition for hover effect */
        }

        .footer-section ul li a:hover {
            color: #007BFF; /* Lighter color on hover */
        }
    }

```