Group name: **Team Alvarado**

Members: **Zaijan Alvarado, Andrei Sinfuego, Jaiden Fermante, Symon Cristoffer Cano, Jon Matthew Mella**

Course Yr. & Block: **BSIS – 2B**

|  |
| --- |
| Problem Number 1: Order Tracker for Client |
| Problem Statement: Our previous code doesn’t include an order tracker for the clients which made us incomplete. |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or Changed |
| <?php  // Start session to track login status  session\_start();  // Check if the user is logged in  if (!isset($\_SESSION['username'])) {  // Redirect to login page if the user is not logged in  header("Location: login.php");  exit();  }  // Database connection  $host = 'localhost';  $db = 'zaikicks';  $user = 'root';  $pass = '';  $conn = new mysqli($host, $user, $pass, $db);  // Check if the connection was successful  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  // Get the username from the session  $username = $\_SESSION['username'];  // Fetch the orders for the logged-in user, grouped by order status  $sql = "SELECT checkout.order\_id, checkout.status, checkout.total\_price, checkout.created\_at  FROM checkout  WHERE checkout.username = ?";  $stmt = $conn->prepare($sql);  $stmt->bind\_param("s", $username); // Bind username as a string  $stmt->execute();  $result = $stmt->get\_result();  // Organize orders by status  $orders\_by\_status = [  'pending' => [],  'shipped' => [],  'delivered' => []  ];  while ($row = $result->fetch\_assoc()) {  $orders\_by\_status[$row['status']][] = $row;  }  $stmt->close();  $conn->close();  ?> | Designed a new order tracking page to provide clients with a convenient way to monitor their orders in real-time. This page features a search bar where users can enter their order ID to fetch specific details related to their purchase. The order status is displayed in a dedicated area, showing progress through various stages such as pending, shipped, or delivered. To enhance the user experience, | At present, the system includes an order tracker that enables clients to effortlessly monitor the status of their orders. Once they have confirmed their selections and completed the checkout process, they can easily track the progress of their orders in real-time. This feature provides added convenience and transparency, allowing customers to stay informed about the status of their purchases every step of the way. |

|  |
| --- |
| Problem Number 2: Checkout information for Clients |
| Problem Statement: Our previous code doesn’t include a checkout information for the clients which made us incomplete. |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or Changed |
| <?php  session\_start();  if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['confirm\_order'])) {  // Database connection  $host = 'localhost';  $db = 'zaikicks';  $user = 'root';  $pass = '';  $conn = new mysqli($host, $user, $pass, $db);  if ($conn->connect\_error) {  die("Database connection failed: " . $conn->connect\_error);  }  // Check if cart exists and is not empty  if (empty($\_SESSION['cart']) || !is\_array($\_SESSION['cart'])) {  echo "Your cart is empty. Please add items before placing an order.";  exit;  }  // Get the user ID, default to 0 for guest users  $user\_id = $\_SESSION['user\_id'] ?? 0;  $total\_price = 0;  // Calculate total price from cart items  foreach ($\_SESSION['cart'] as $cart\_item) {  $total\_price += $cart\_item['quantity'] \* $cart\_item['item\_price'];  }  // Insert the order into the `checkout` table  $stmt = $conn->prepare("INSERT INTO checkout (user\_id, total\_price) VALUES (?, ?)");  $stmt->bind\_param("id", $user\_id, $total\_price);  if ($stmt->execute()) {  $order\_id = $stmt->insert\_id; // Get the ID of the newly created order  // Insert each cart item into the `order\_items` table  $stmt = $conn->prepare("INSERT INTO order\_items (order\_id, item\_id, quantity, price) VALUES (?, ?, ?, ?)");  foreach ($\_SESSION['cart'] as $cart\_item) {  $item\_id = $cart\_item['item\_id'];  $quantity = $cart\_item['quantity'];  $price = $cart\_item['item\_price'];  $stmt->bind\_param("iiid", $order\_id, $item\_id, $quantity, $price);  $stmt->execute();  }  // Clear the cart after successful order placement  $\_SESSION['cart'] = [];  // Redirect to a success page  header("Location: success.php");  exit;  } else {  // Display error if the order insertion fails  echo "Error: " . $stmt->error;  }  // Close the database connection  $stmt->close();  $conn->close();  } else {  echo "Invalid request.";  }  ?> | * Created a new checkout information page with a clear itemized list of products. * Included fields for inputting and validating shipping details. * Added a summary section for subtotal, shipping, and total costs. | At this point, we have incorporated a checkout information section for clients, allowing them to easily review the details of the items they are about to purchase. This feature provides a clear and organized view of their selected items, helping them verify product information, quantities, and prices before completing the checkout process. It enhances the overall shopping experience by ensuring that customers have all the necessary details at their fingertips prior to finalizing their order. |

|  |
| --- |
| Problem Number 3: Display of items for Client and Admin |
| Problem Statement: Our previous code doesn’t include a display of items for the client and admin which made us incomplete. |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or Changed |
| Client: <?php // Start the session to access session variables  session\_start();  // Initialize cart if it doesn't exist  if (!isset($\_SESSION['cart'])) {  $\_SESSION['cart'] = [];  }  // Database connection  $host = 'localhost';  $db = 'zaikicks';  $user = 'root';  $pass = '';  $conn = new mysqli($host, $user, $pass, $db);  // Check if the connection was successful  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  // Fetch available items from the database  $sql = "SELECT \* FROM items"; // Fetch all items including images  $result = $conn->query($sql);  // Check if the form is submitted to add items to the cart  if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {  $item\_id = $\_POST['item\_id'];  $quantity = $\_POST['quantity'];  // Add item to the cart session  $cart\_item = [  'item\_id' => $item\_id,  'quantity' => $quantity  ];  // Add the item to the session cart  $\_SESSION['cart'][] = $cart\_item;  // Redirect to cart2.php for checkout  header("Location: cart3.php");  exit;  }  $conn->close();  ?>  Admin: <?php  // Enable error reporting  ini\_set('display\_errors', 1);  ini\_set('display\_startup\_errors', 1);  error\_reporting(E\_ALL);  session\_start();  // Database connection  $servername = "localhost";  $username = "root";  $password = "";  $dbname = "zaikicks";  $conn = new mysqli($servername, $username, $password, $dbname);  // Check connection  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  // Fetch items for display  function fetchItems($conn) {  $sql = "SELECT \* FROM items";  $result = $conn->query($sql);  $items = [];  if ($result->num\_rows > 0) {  while ($row = $result->fetch\_assoc()) {  $items[] = $row;  }  }  return $items;  }  // Handle adding an item with image upload  if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['ajax']) && $\_POST['ajax'] === '1') {  $name = $conn->real\_escape\_string($\_POST['item\_name']);  $description = $conn->real\_escape\_string($\_POST['item\_descrip']);  $price = floatval($\_POST['item\_price']);  $stock = intval($\_POST['stock']);  $imagePath = 'No Image';  // Image upload handling  if (isset($\_FILES['item\_image']) && $\_FILES['item\_image']['error'] == UPLOAD\_ERR\_OK) {  $imageTmpPath = $\_FILES['item\_image']['tmp\_name'];  $imageName = basename($\_FILES['item\_image']['name']);  $imagePath = 'uploads/'. $imageName;  // Ensure uploads directory exists  if (!is\_dir('uploads')) {  mkdir('uploads', 0777, true);  }  if (!move\_uploaded\_file($imageTmpPath, $imagePath)) {  echo json\_encode(['error' => 'Failed to upload image.']);  exit;  }  }  // Insert into database  $sql = "INSERT INTO items (item\_name, item\_descrip, item\_price, stock, image\_path)  VALUES ('$name', '$description', $price, $stock, '$imagePath')";  if ($conn->query($sql) === TRUE) {  $id = $conn->insert\_id;  echo json\_encode([  'id' => $id,  'item\_name' => $name,  'item\_descrip' => $description,  'item\_price' => $price,  'stock' => $stock,  'image\_path' => $imagePath  ]);  } else {  http\_response\_code(500);  echo json\_encode(['error' => $conn->error]);  }  exit;  }  // Handle item deletion  if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && isset($\_POST['delete\_item\_id'])) {  $item\_id = intval($\_POST['delete\_item\_id']);  $sql = "DELETE FROM items WHERE item\_id = $item\_id";  if ($conn->query($sql) === TRUE) {  echo json\_encode(['success' => true]);  } else {  http\_response\_code(500);  echo json\_encode(['error' => $conn->error]);  }  exit;  }  $items = fetchItems($conn);  $conn->close();  ?> | On the client side, the product listing page now dynamically fetches items directly from the database. We implemented filter options that allow users to sort products by categories, price range, and availability, along with a search functionality for quick item lookup. Each product is displayed with a thumbnail image and essential details to create a visually engaging and user-friendly shopping experience. For admins, a comprehensive item management section was added to the dashboard. Items are displayed in a tabular format, showcasing attributes such as name, price, stock quantity, and description. Admins can upload images, update product details, or delete items through an interactive interface. | In essence, we have added a display feature for both clients and admins, allowing them to easily view all the items showcased on a dedicated page. This enhancement provides a user-friendly interface where both parties can quickly browse and access detailed information about the items available. For clients, it improves their shopping experience by offering an organized and visually appealing layout, while for admins, it simplifies item management and oversight on the platform. |

|  |
| --- |
| Problem Number 4: Order Information with the list of items by a Client into the Admin Page |
| Problem Statement: Our previous code doesn’t include an order information with the list of items by a client into an admins page which made us incomplete. |

|  |  |  |
| --- | --- | --- |
| CODE CHANGES | INTERFACE CHANGES | Describe the functionality added or Changed |
| <?php  session\_start();  // Database connection  $host = 'localhost';  $db = 'zaikicks';  $user = 'root';  $pass = '';  $conn = new mysqli($host, $user, $pass, $db);  // Check if the connection was successful  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }  // Update order status  if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['order\_id'])) {  $order\_id = $\_POST['order\_id'];  $new\_status = $\_POST['status'];  // Update status directly (delivered means final status)  $sql\_update = "UPDATE checkout SET status = ? WHERE order\_id = ?";  $stmt = $conn->prepare($sql\_update);  $stmt->bind\_param("si", $new\_status, $order\_id);  $stmt->execute();  }  // Fetch all orders with their details  $sql\_orders = "SELECT c.order\_id, c.username, c.status, c.total\_price, c.tracking\_number,  c.created\_at, c.delivery\_address, c.payment\_method  FROM checkout c  ORDER BY c.created\_at DESC";  $result\_orders = $conn->query($sql\_orders);  // Prepare to fetch order items for each order  $sql\_items = "SELECT \* FROM order\_items WHERE order\_id = ?";  $stmt\_items = $conn->prepare($sql\_items);  ?> | * Updated the admin dashboard to include a new section for detailed order information. * Added the complete information of the order info of the client | Essentially, we have introduced an order information feature on the admin page, which includes a detailed list of items purchased by each client. This addition allows admins to easily access and manage the information related to a specific order, streamlining the process of tracking and handling customer orders. Admins can now efficiently review, update, or modify order details, ensuring better control and organization of the order management system. |