

Group name: **Team Alvarado**

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

Course Yr. & Block: **BSIS – 2B**

Role:

Admin- Create a code for admin from login, item management, order management, admin dashboard, logout.

Client- Create a code Orders, order Confirmation, Order tracking, logout.

Visitor- Create a code for shopping list of visitor.

I was the one who present the system during the defense.

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
<pre><?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';</pre>	<p>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,</p>	<p>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.</p>

```
$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']][] =
```

<pre>\$row; } \$stmt->close(); \$conn->close(); ?></pre>		
--	--	--

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
<pre><?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: " .</pre>	<ul style="list-style-type: none"> 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. 	<p>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.</p>

<pre>\$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</pre>		
--	--	--

```
`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.";
}
?>
```

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
<pre>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</pre>	<p>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.</p>	<p>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.</p>

<pre>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();</pre>		
--	--	--

```
// 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_descr
ip']);
    $price = floatval($_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]);
    }
```

<pre> } else { http_response_code(500); echo json_encode(['error' => \$conn- >error]); } exit; } \$items = fetchItems(\$conn); \$conn->close(); ?> </pre>		
--	--	--

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
<pre> <?php session_start(); // Database connection \$host = 'localhost'; \$db = 'zaikicks'; \$user = 'root'; \$pass = ''; \$conn = new mysqli(\$host, \$user, \$pass, \$db); </pre>	<ul style="list-style-type: none"> Updated the admin dashboard to include a new section for detailed order information. Added the complete information of the order info of the client 	<p>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</p>

```

// 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

```

better control and organization of the order management system.

<pre>order \$sql_items = "SELECT * FROM order_items WHERE order_id = ?"; \$stmt_items = \$conn- >prepare(\$sql_items); ?></pre>		
---	--	--