

Group name : TINY CODERS

**Members: AMISTOSO, CLARIES ANN B.
BATA, MARIZ S.
FETIL, JAENNIE CYRELL N.
RESENTES, SHANLEY R.
SABLAYAN, PENELOPE R.**

Course Yr. & Block: BSIS-2A

Problem Number : 1

Problem Statement: To add a search bar for orders and items in the Admin side

CODE CHANGES	INTERFACE CHANGES	Describe the functionality added or Changed
Search bar for manage items (manage_orders.php) <!-- Search Bar --> <form method="GET" style="margin-bottom: 20px;"> <input type="text" name="search" placeholder="Search by name or category" value="<?php echo isset(\$_GET['search']) ? htmlspecialchars(\$_GET['search']) : ""; ?>" style="padding: 10px; width: 300px; border: 1px solid #ddd; border-radius: 4px;"> <button type="submit" style="padding: 10px; background-color: #6b4e2e; color: #fff; border: none; border-radius: 4px; cursor: pointer;"> Search </button> </form>	-Search bars were added in the Admin Side (Manage orders and Manage Items)	- The search bar in the admin side, both for the managing orders and items allowed us to search information that we want to search, for instance, the item name can now be searched in the manage items tab as well as the user id of the customers who ordered has been filtered out in the manage orders tab. Therefore, it enabled us

```

$sql = "SELECT * FROM menu_items";
if (isset($_GET['search']) && !empty($_GET['search'])) {
    $search = htmlspecialchars($_GET['search']);
    $sql .= " WHERE item_name LIKE :search OR item_category LIKE :search";
    $stmt = $conn->prepare($sql);
    $stmt->execute([':search' => "%$search%"]);
} else {
    $stmt = $conn->query($sql);
}

```

Search bar in manage orders with filter for results (user_id only) (manage_items.php)

```

<!-- Search Bar -->
<div class="search-bar">
    <form method="GET">
        <input
            type="text"
            name="search"
            placeholder="Search by User ID"
            value="<?php echo isset($_GET['search']) ?
htmlspecialchars($_GET['search']) : ''; ?>">
        <button type="submit">Search</button>
    </form>
</div>

```

```

//Fetch orders from the database based only on user_id search
$sql = "SELECT * FROM orders";
if (isset($_GET['search']) && !empty($_GET['search'])) {
    $search = htmlspecialchars($_GET['search']);
    // Filter only by user_id
    $sql .= " WHERE user_id LIKE :search";
    $stmt = $conn->prepare($sql);
    $stmt->execute([':search' => "%$search%"]);
} else {
    $stmt = $conn->query($sql);
}

```

to easily and efficiently
search details that we
needed to find.

Problem Number : 2

Problem Statement: To modify the categories into Drop down select

CODE CHANGES	INTERFACE CHANGES	Describe the functionality added or Changed
<p>Drop down select for item category(manage_order.php)</p> <pre><!-- Add New Item Form --> <h3 id="haha">Add New Item</h3> <form method="post" enctype="multipart/form-data"> <input type="text" name="item_name" placeholder="Item Name" required> <select name="item_category" required> <option value="" disabled selected>Select Category</option> <option value="Hot Beverages">Hot Beverages</option> <option value="Cold Beverages">Cold Beverages</option> <option value="Snacks & Pastries">Snacks & Pastries</option> <option value="Tea & Non-Coffee Drinks">Tea & Non- Coffee Drinks</option> <option value="Specialty Drinks">Specialty Drinks</option> </select> <textarea name="item_desc" placeholder="Item Description" rows="4" required></textarea> <input type="number" name="item_price" placeholder="Item Price" required> <input type="file" name="item_image" required> <button type="submit" name="add_item">Add Item</button> </form> </div></pre>	<p>A dropdown select button for selecting the item category for adding an item in the admin dashboard (manage items tab) is added instead of the manual inputting of the item category.</p>	<p>- The dropdown select button for selecting the item category for adding an item in the admin dashboard (manage items tab) enabled accessibility and efficiency for admins especially in terms of adding a new item in the items lists.</p>

Problem Number : 3

Problem Statement: To modify the delete button to just update the item_status of any indicator for the item's current active or inactive status instead of deleting the item in the database, so that it will not affect the orders.

CODE CHANGES	INTERFACE CHANGES	Describe the functionality added or Changed
<p>Drop down select for item status (manage_orders.php)</p> <pre><!-- Status Dropdown --> <form action="" method='get'> <select name='status'> <option value='Active' ".(\$row['status'] == 'Active' ? 'selected' : ").">Active</option> <option value='Inactive' ".(\$row['status'] == 'Inactive' ? 'selected' : ").">Inactive</option> </select> <input type='hidden' name='item_id' value='{ \$row['item_id']}'> <button type='submit' class='status-btn'>Change Status</button> </form> </td> </tr>"; } } else { echo "<tr><td colspan='8'>No items found</td></tr>"; } ?> </tbody> </table></pre>	<p>A “Active” & “Inactive” button for the item status is added into the admin dashboard (Manage Items) instead of the previously added delete button.</p>	<p>- The dropdown select for the item status button allowed us to change the status of the items into “Active” or “Inactive”, enabling customers to see whether a particular item listed in the products lists is Active (available) or Inactive (unavailable). With these changes, we are able to easily manage the item availability in our products and services tab.</p>

Problem Number : 4

Problem Statement: Show the actual items that have been checked

CODE CHANGES	INTERFACE CHANGES	Describe the functionality added or Changed
<p>Code for adding the items that the customer ordered in order_items table where the menu_items and orders are combined. (submit_order.php)</p> <pre>// Get the last inserted order ID \$order_id = \$conn->lastInsertId(); // Insert items into the order_items table \$order_item_stmt = \$conn->prepare(" INSERT INTO order_items (order_id, item_id, quantity, price) VALUES (:order_id, :item_id, :quantity, :price) "); foreach (\$_SESSION['cart'] as \$cart_item) { \$order_item_stmt->bindParam(':order_id', \$order_id, PDO::PARAM_INT); \$order_item_stmt->bindParam(':item_id', \$cart_item['item_id'], PDO::PARAM_INT); \$order_item_stmt->bindParam(':quantity', \$cart_item['item_quantity'], PDO::PARAM_INT); \$order_item_stmt->bindParam(':price', \$cart_item['item_price']); \$order_item_stmt->execute(); }</pre>	<p>The item name and the customer's name are displayed in the My Orders tab of the customer.</p>	<p>Displaying the item(s) name in my orders tab of the customer enables easy identification of the items/products they have ordered. In addition, we have also displayed the customer's name along with the other order details.</p>

Role: Tester

I was the one to test the website, my main task was checking the website to be smooth enough, free of bugs, and with no problems at all and ensuring that the website serves its users in the simplest and most intuitive way it could. I have personally tested all the features there are, from navigation down to forms, responsiveness on various browsers, and general user interactions. I also looked at each functionality of the website through manual testing, simulating real life situations, in order to determine possible issues. For instance, I test the functionality, whether the orders placed by the customer successfully enters the database, verifying that the system accurately stores order details. Whenever there were bugs or inconsistencies in the website, I reported the bugs to the development team, and worked closely with them to make sure the bugs were resolved. All problems were fixed, which made sure the website followed the quality standards, proved to be reliable, and was an intuitive and easy-to-use platform for any visitor.