

**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**

<b>CODE CHANGES</b>	<b>INTERFACE CHANGES</b>	<b>Describe the functionality added or Changed</b>
<b>Search bar for manage items (manage_orders.php)</b> <!-- 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><b>Drop down select for item category(manage_order.php)</b></p> <pre>&lt;!-- Add New Item Form --&gt;     &lt;h3 id="haha"&gt;Add New Item&lt;/h3&gt;     &lt;form method="post" enctype="multipart/form-data"&gt;         &lt;input type="text" name="item_name" placeholder="Item Name" required&gt;         &lt;select name="item_category" required&gt;             &lt;option value="" disabled selected&gt;Select Category&lt;/option&gt;             &lt;option value="Hot Beverages"&gt;Hot Beverages&lt;/option&gt;             &lt;option value="Cold Beverages"&gt;Cold Beverages&lt;/option&gt;             &lt;option value="Snacks &amp; Pastries"&gt;Snacks &amp; Pastries&lt;/option&gt;             &lt;option value="Tea &amp; Non-Coffee Drinks"&gt;Tea &amp; Non- Coffee Drinks&lt;/option&gt;             &lt;option value="Specialty Drinks"&gt;Specialty Drinks&lt;/option&gt;         &lt;/select&gt;         &lt;textarea name="item_desc" placeholder="Item Description" rows="4" required&gt;&lt;/textarea&gt;         &lt;input type="number" name="item_price" placeholder="Item Price" required&gt;         &lt;input type="file" name="item_image" required&gt;         &lt;button type="submit" name="add_item"&gt;Add Item&lt;/button&gt;     &lt;/form&gt; &lt;/div&gt;</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><b>Drop down select for item status (manage_orders.php)</b></p> <pre>&lt;!-- Status Dropdown --&gt;     &lt;form action="" method='get'&gt;         &lt;select name='status'&gt;             &lt;option value='Active' ".(\$row['status'] == 'Active' ? 'selected' : ")."&gt;Active&lt;/option&gt;             &lt;option value='Inactive' ".(\$row['status'] == 'Inactive' ? 'selected' : ")."&gt;Inactive&lt;/option&gt;         &lt;/select&gt;         &lt;input type='hidden' name='item_id' value='{ \$row['item_id']}'&gt;         &lt;button type='submit' class='status-btn'&gt;Change Status&lt;/button&gt;     &lt;/form&gt; &lt;/td&gt; &lt;/tr&gt;";     } } else {     echo "&lt;tr&gt;&lt;td colspan='8'&gt;No items found&lt;/td&gt;&lt;/tr&gt;"; } ?&gt; &lt;/tbody&gt; &lt;/table&gt;</pre>	<p>A “Active” &amp; “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><b>Code for adding the items that the customer ordered in order_items table where the menu_items and orders are combined. (submit_order.php)</b></p> <pre>// Get the last inserted order ID \$order_id = \$conn-&gt;lastInsertId();  // Insert items into the order_items table \$order_item_stmt = \$conn-&gt;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-&gt;bindParam(':order_id', \$order_id, PDO::PARAM_INT);     \$order_item_stmt-&gt;bindParam(':item_id', \$cart_item['item_id'], PDO::PARAM_INT);     \$order_item_stmt-&gt;bindParam(':quantity', \$cart_item['item_quantity'], PDO::PARAM_INT);     \$order_item_stmt-&gt;bindParam(':price', \$cart_item['item_price']);     \$order_item_stmt-&gt;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>

