

Group name: Redvelvet

Members: Christine Joy Bande, Rose Angela Marie Cañon, Magdaline Infante, and Christine Jade Ondis

Course Yr. & Block: BSIS-2A

Problem Number: 1

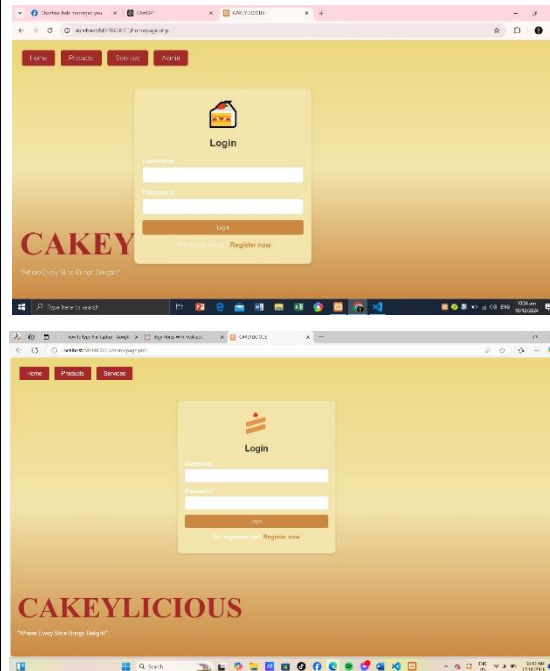
Problem Statement

admin creation must be done directly in the database or must be a default account in your system.

#### CODE CHANGES

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    if (isset($_POST['login'])) {  
        session_start(); // Start session here before any  
        output  
  
        $login_username =  
        mysqli_real_escape_string($conn,  
        $_POST['login_username']);  
        $login_password =  
        mysqli_real_escape_string($conn,  
        $_POST['login_password']);  
  
        // Check in admin_user table  
        $stmt = $conn->prepare("SELECT password FROM  
        admin_user WHERE username = ?");  
        $stmt->bind_param("s", $login_username);  
        $stmt->execute();  
        $stmt->store_result();  
  
        if ($stmt->num_rows > 0) {  
            $stmt->bind_result($stored_password);  
            $stmt->fetch();  
  
            // Compare the entered password with the stored  
            plain password  
            if ($login_password === $stored_password) {  
                $_SESSION['username'] = $login_username;
```

#### INTERFACE CHANGES



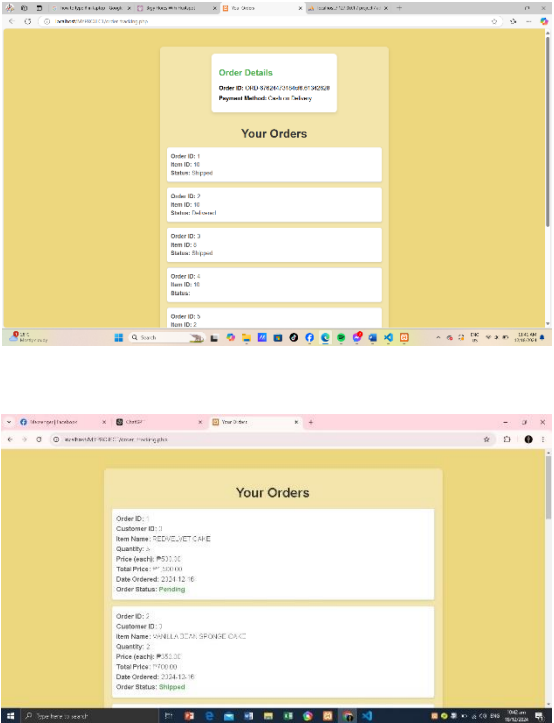
Describe the functionality added or Changed

Some parts of the code were modified so that when a user logs in, only accounts that exist in the admin\_user table in the database can access the admin page, which displays the dashboard, manage orders, and item management. However, if a customer logs in, they are only directed to the items page since the customer\_user and admin\_user tables in the database are separate.

<pre> // Redirect to admin page header("Location: adminpage.php"); exit(); } else {     \$message = "&lt;p class='error-message'&gt;Invalid username or password.&lt;/p&gt;"; } } else {     // Check in user_account table     \$stmt-&gt;close();      \$stmt = \$conn-&gt;prepare("SELECT password FROM user_account WHERE username = ?");     \$stmt-&gt;bind_param("s", \$login_username);     \$stmt-&gt;execute();     \$stmt-&gt;store_result();      if (\$stmt-&gt;num_rows &gt; 0) {         \$stmt-&gt;bind_result(\$stored_password);         \$stmt-&gt;fetch();          // Compare the entered password with the         stored plain password         if (\$login_password === \$stored_password) {             \$_SESSION['username'] = \$login_username;              // Redirect to items page             header("Location: items.php");             exit();         } else {             \$message = "&lt;p class='error- message'&gt;Invalid username or password.&lt;/p&gt;";         }     } else {         \$message = "&lt;p class='error-message'&gt;Invalid username or password.&lt;/p&gt;";     } } \$stmt-&gt;close(); </pre>		
---	--	--

Problem Number: 2
Problem Statement

Order History, must be able to list the item name instead of item\_id. also include the Quantity ordered as well as the Amount to Pay.

CODE CHANGES	INTERFACE CHANGES	Describe the functionality added or Changed
<pre> \$sql = "     SELECT         o.order_id,         o.item_qty,         o.date_ordered,         o.status, -- Added order status         o.cus_id, -- Added customer name from orders         i.item_name,         i.item_price,         (o.item_qty * i.item_price) AS total_price     FROM         orders o     LEFT JOIN         items i ON o.item_id = i.item_id -- Use LEFT JOIN to include orders without matching items ";  \$stmt = \$conn-&gt;prepare(\$sql); \$stmt-&gt;execute(); \$result = \$stmt-&gt;get_result();  // Check if any orders were found \$orders = []; \$total_amount = 0;  while (\$row = \$result-&gt;fetch_assoc()) {     \$orders[] = \$row;     \$total_amount += \$row['total_price']; } </pre>		<p>The changes to the interface now display the item name, item quantity, customer ID, order ID, total price, order date, and order status. The changes to the code include adding more to the SQL query for fetching data and modifying the echo statements.</p>

Problem Number: 3

Problem Statement

do the bullet #2 as well in the Admin where it changes the Order Status, (must be able to see the items, qty and amount to Pay as well as the customer Info)

CODE CHANGES

```
$sql = "
SELECT
    o.order_id,
    o.item_qty,
    o.date_ordered,
    i.item_name,
    i.item_price,
    (o.item_qty * i.item_price) AS total_price,
    o.status,
    u.user_id,
    u.name,
    u.contact_no,
    u.address
FROM
    orders o
LEFT JOIN
    items i ON o.item_id = i.item_id
LEFT JOIN
    users u ON u.user_id = u.user_id //
Query to fetch orders and related user
information
```

INTERFACE CHANGES

The top screenshot shows the 'Order Management' form with three order entries. Each entry has a 'Current Status' and a 'Change Status' dropdown menu. The bottom screenshot shows a table of orders with columns for Order ID, User ID, Customer Name, Contact No, Address, Item Name, Quantity, Price, Date, Current Status, and Change Status.

Order ID	User ID	Customer Name	Contact No	Address	Item Name	Quantity	Price	Date	Current Status	Change Status
1	1	John Doe	090-1234567	12345	RED VELVET CAKE	2	\$500.00	2021-12-10	Pending	Pending
2	2	Jane Smith	090-7654321	67890	VANILLA BEAN SPONGE CAKE	2	\$500.00	2021-12-10	Shipped	Shipped
3	3	John Doe	090-1234567	12345	COCONUT CREAM CAKE	2	\$500.00	2021-12-10	Delivered	Delivered
4	4	Jane Smith	090-7654321	67890	STRAWBERRY SPONGE CAKE	2	\$500.00	2021-12-10	Pending	Pending

Describe the functionality added or Changed

In the Manage Orders section, updates were made to the SQL query to fetch more comprehensive details. These changes allow the admin to view important information about each order, including the name of the customer who placed the order, the specific items ordered, the quantity of each item, the date and time the order was placed, the total amount of the order, and its current status. Additionally, the interface was enhanced to provide functionality for the admin to update the order status, making it easier to track and manage the progress of each order of the customers.

Contribution to the group project:

(Christine Jade Ondis)

Part of the code

- (items(add to cart, insert images), checkout, order\_success)

- database

- edit documentation changes

Role

- designer, testing, support