

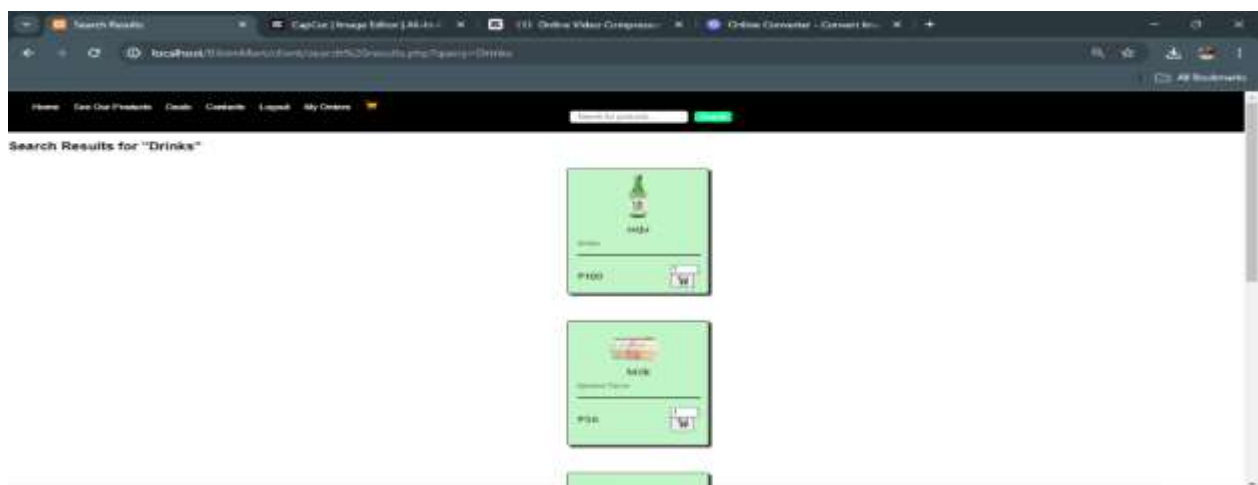
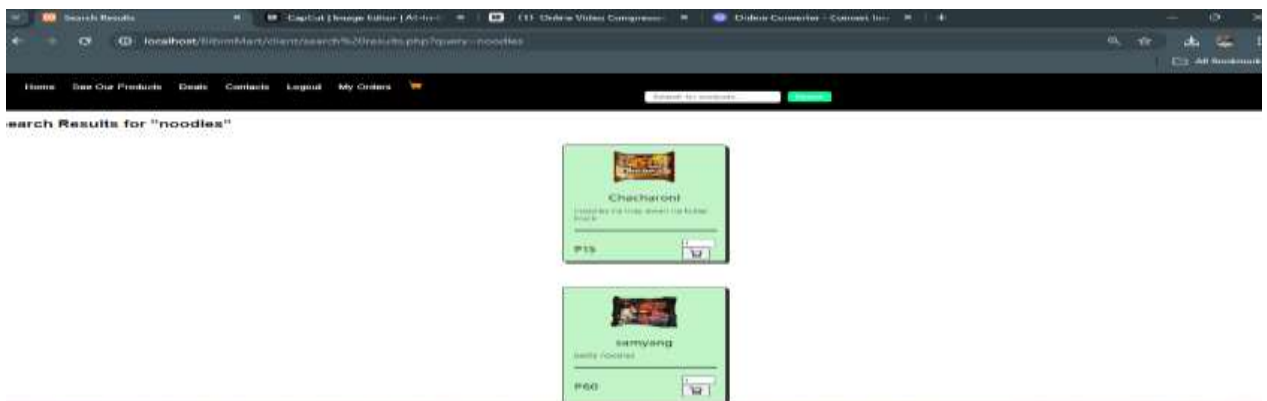
PROBLEM: Logical Debugging

1. Add an optimized Search Bar, it should be able to accept descriptive search keywords and display similar/relevant items to the search key and show order numbers and details with the relevant items being searched if any. for example, Noodles, your system should be able to display and suggest relevant items to Noodles even though your item names do not have Noodles in it.

🔧 In the debugging process of optimizing the Search bar, I updated the code in search_results.php. I added an SQL query that fetches the products based on the search query. Instead of only fetching the product names from the database, I also included the product category names. This means that if you search for categories, all products under those categories will be displayed. Additionally, I included product descriptions in the search criteria, so products with the same or relevant descriptions can still be searched and displayed.

```
search_results.php X
client > search_results.php
12 $search_query = "";
13 if (isset($_GET['query'])) {
14     $search_query = $_GET['query'];
15 }
16
17 $sql = "SELECT p.*, c.product_cat_name
18         FROM products p
19         INNER JOIN product_cat c ON p.product_cat_id = c.product_cat_id
20         WHERE p.product_status = 1
21         AND (p.product_name LIKE ? OR p.product_desc LIKE ? OR c.product_cat_name LIKE ?)";
22 $stmt = $conn->prepare($sql);
23 if (!$stmt) {
24     die("Error preparing statement: " . $conn->error);
25 }
26
27 $search_term = "%" . $search_query . "%";
28 $stmt->bind_param("sss", $search_term, $search_term, $search_term);
29
30 if (!$stmt->execute()) {
31     die("Error executing statement: " . $stmt->error);
32 }
33
34 $result = $stmt->get_result();
35 if (!$result) {
36     die("Error getting result: " . $stmt->error);
37 }
38 >>
```

Results:



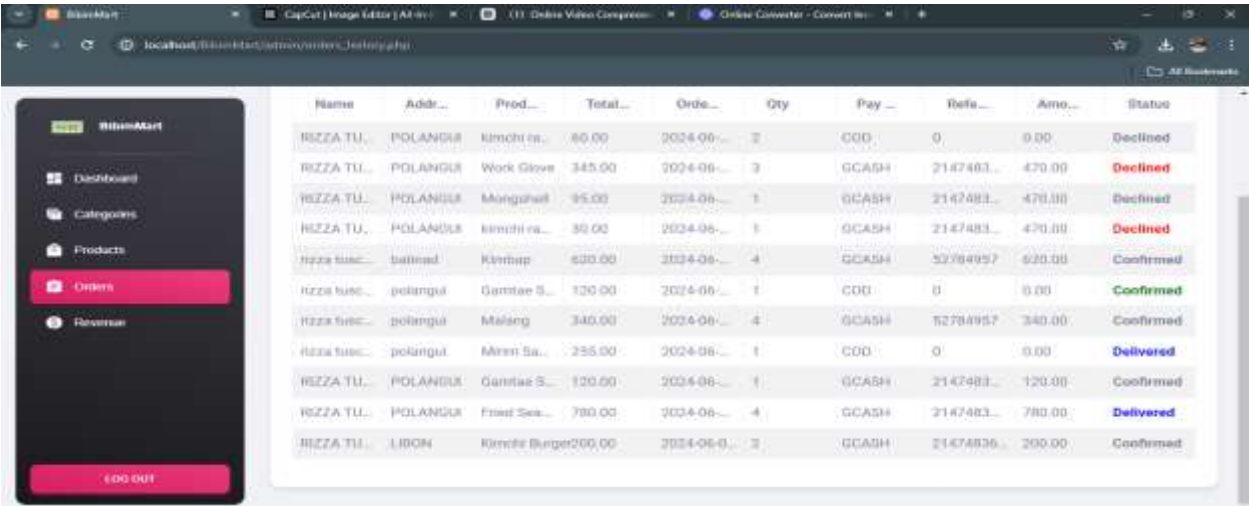
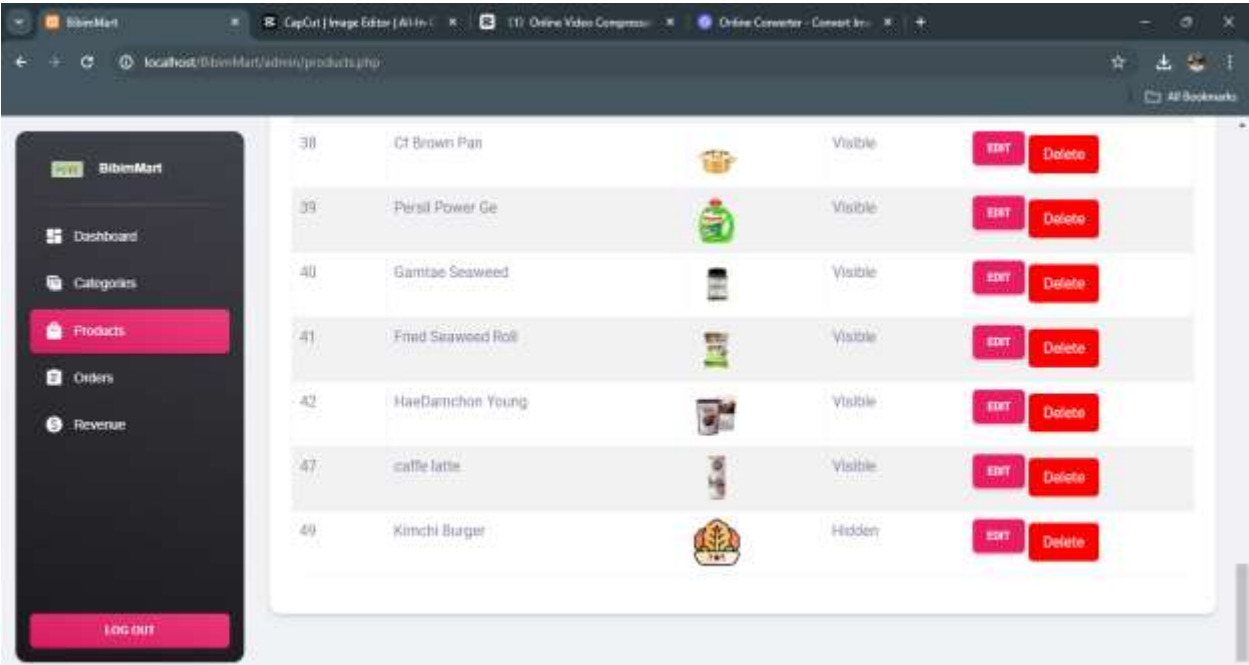
2. The completed orders must not be affected whenever an item is discontinued or deleted or deactivated.

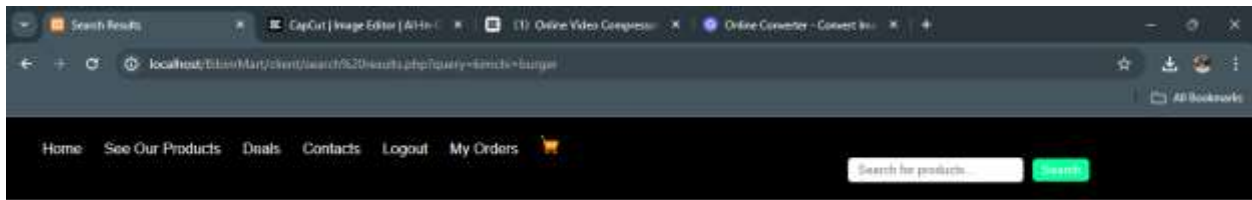
✚ In the debugging process of completed orders, it was necessary to ensure that the orders are not affected when items are deactivated or disabled. To address this issue, I made changes to the SQL query in myfunctions.php. Instead of using the query 'SELECT * FROM products WHERE product_cat_id='\$category_id';', I updated it to 'SELECT * FROM products WHERE product_cat_id='\$category_id' AND product_status = 1;'. This modification ensures that only products with a product_status of 1 are displayed in the products list. Even if a product is disabled, it will still appear in the order history once it has been ordered. This solution prevents the need to delete products, as deleting them from the database would also remove them from the order history. Instead, we disable them to ensure that customers/clients cannot see them.

```
myfunctions.php X
myfunctions.php > ...

1 reference
37 function getProductsByCategory($category_id){
38     global $conn;
39     $query = "SELECT * FROM products WHERE product_cat_id='$category_id' AND product_status = 1";
40     $query_run = mysqli_query($conn, $query);
41     if(!$query_run) {
42         die("Query Failed: " . mysqli_error($conn));
43     }
44     return $query_run;
45 }
46
47
```

Results:





Search Results for "kimchi burger"