



# Lab Report 7

Shopping Cart Implementation

**Course:** CSC381 - E-Commerce

**Submitted by:**

Parakram Kharel  
Roll No: 24

*Kathford International College of Engineering and Management*  
Affiliated to Tribhuvan University

August 24, 2025

## 1. Objective

To implement a shopping cart in PHP that allows users to add, remove, and update items dynamically using sessions.

## 2. Tools and Technologies Used

Technology	Purpose
HTML5	Cart pages and form structure
CSS3	Layout and visual styling
JavaScript	Inline confirmations and UI behavior
PHP	Server-side logic, CRUD, Sessions
MySQL	Persistent storage for catalog and orders
Caddy	Web server and local hosting
Bootstrap 5	Admin UI, tables, forms, responsive grid
VS Code	Development environment
Browser DevTools	Testing and debugging

## 3. Theory / Background

A shopping cart is a temporary online storage area that keeps track of products selected by the user before checkout. In PHP-based systems, the cart is usually implemented using sessions, which store user data temporarily on the server. When a user adds an item to the cart:

- PHP creates a session for that user.
- Product details (name, price, quantity, etc.) are stored in the session array.
- The cart data persists across multiple pages until the user checks out or logs out.

## 4. Page Layout Design

### 4.1 Cart Format

- Product image, name, price
- Quantity input field
- "Remove" and "Update" buttons

## 5. Code Snippets

### 5.1 Database Connection

```
1 <?php
2 $host = "localhost";
3 $user = "root";
4 $pass = "root";
5 $dbname = "ecom";
6
7 $conn = mysqli_connect($host, $user, $pass, $dbname);
8
9 if (!$conn) {
10     die("Connection failed: " . mysqli_connect_error());
11 }
12 ?>
13
```

### 5.2 Cart Add

```
1 <?php
2 session_start();
3
4 if (isset($_POST['add_to_cart'])) {
5     $product_id = $_POST['product_id'];
6     $product_name = $_POST['product_name'];
7     $product_price = $_POST['product_price'];
8     $quantity = $_POST['quantity'];
9
10    $cart_item = array(
11        'id' => $product_id,
12        'name' => $product_name,
13        'price' => $product_price,
14        'quantity' => $quantity
15    );
16
17    if (!isset($_SESSION['cart'])) {
18        $_SESSION['cart'] = array();
19    }
20
21    $_SESSION['cart'][$product_id] = $cart_item;
22    header("Location: cart.php");
23    exit();
24}
```

```
24 }  
25 ?>
```

### 5.3 Update / Clear Cart

```
1   <?php  
2 // Remove item  
3 if (isset($_GET['remove'])) {  
4     $id = $_GET['remove'];  
5     unset($_SESSION['cart'][$id]);  
6     header("Location: cart.php");  
7 }  
8  
9 // Update quantity  
10 if (isset($_POST['update_cart'])) {  
11   foreach ($_POST['quantity'] as $id => $qty) {  
12     $_SESSION['cart'][$id]['quantity'] = $qty;  
13   }  
14   header("Location: cart.php");  
15 }  
16 ?>
```

### 5.4 PHP: Display Cart Items

```
1   <?php  
2     session_start();  
3     $total = 0;  
4   ?>  
5  
6 <h2>Your Shopping Cart</h2>  
7 <table border="1" cellpadding="10">  
8 <tr>  
9   <th>Product</th>  
10  <th>Price</th>  
11  <th>Quantity</th>  
12  <th>Total</th>  
13  <th>Action</th>  
14 </tr>  
15  
16 <?php  
17 if (!empty($_SESSION['cart'])) {  
18   foreach ($_SESSION['cart'] as $item) {
```

```

19     $subtotal = $item['price'] * $item['quantity'];
20     $total += $subtotal;
21     echo "
22     <tr>
23       <td>{$item['name']}</td>
24       <td>\${$item['price']}</td>
25       <td>{$item['quantity']}</td>
26       <td>\${$subtotal}</td>
27       <td><a href='cart.php?remove={$item['id']}>Remove</a></td>
28     </tr>";
29   }
30   echo "<tr><td colspan='3'><b>Grand
31   Total</b></td><td>\${$total}</td><td></td></tr>";
32 } else {
33   echo "<tr><td colspan='5'>Your cart is empty!</td></tr>";
34 }
35 ?>
36 </table>

```

## 6. Output / Screenshots

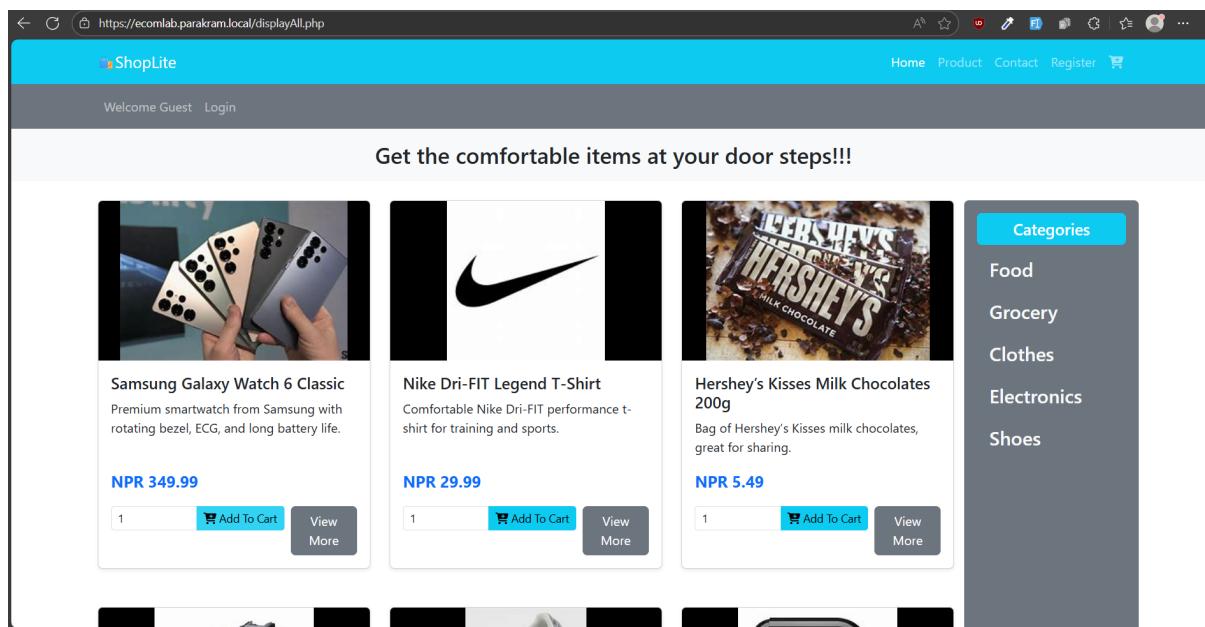


Figure 1: Products with Add to Cart button.

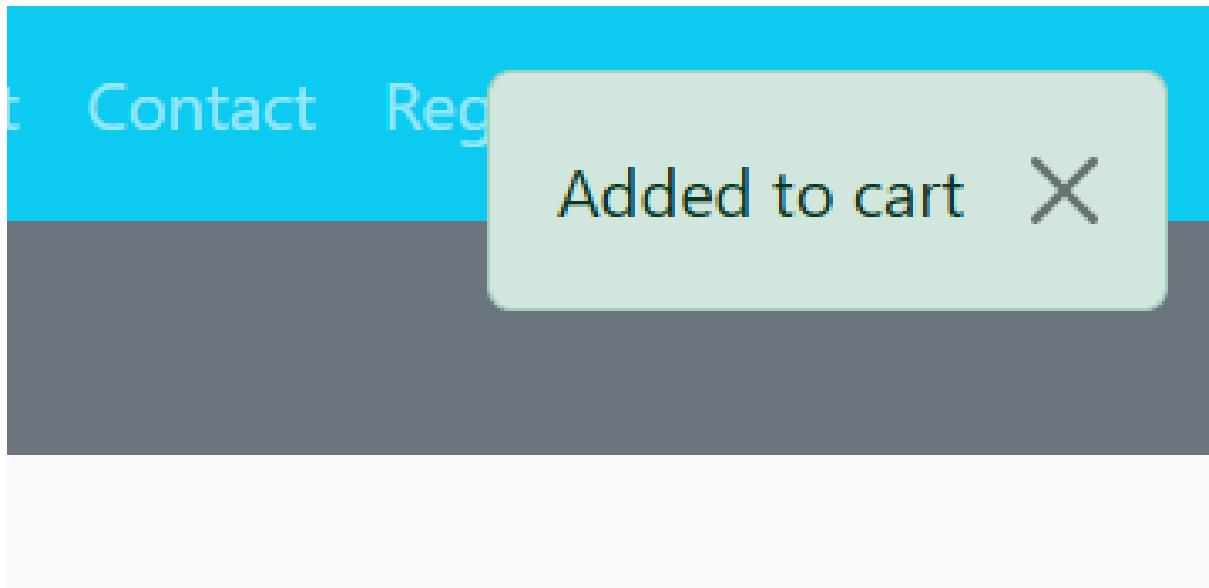


Figure 2: Add to Cart Toast.

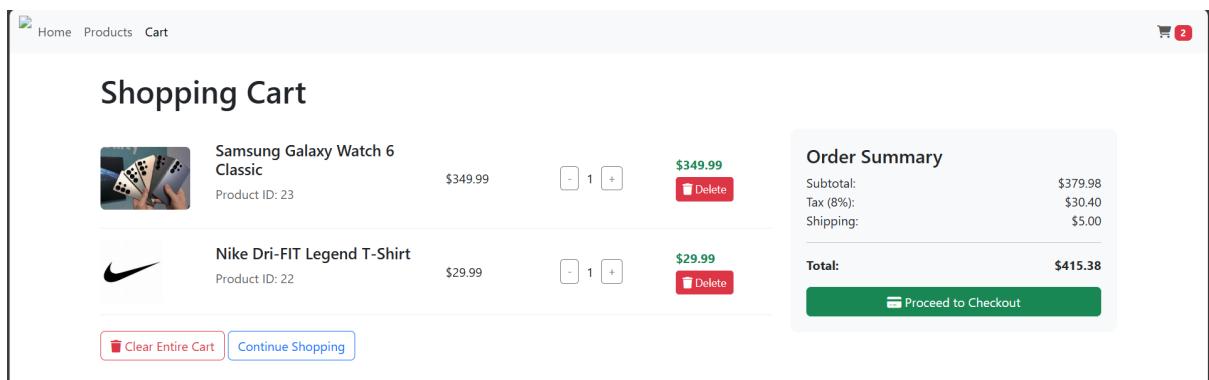


Figure 3: Shopping Cart.

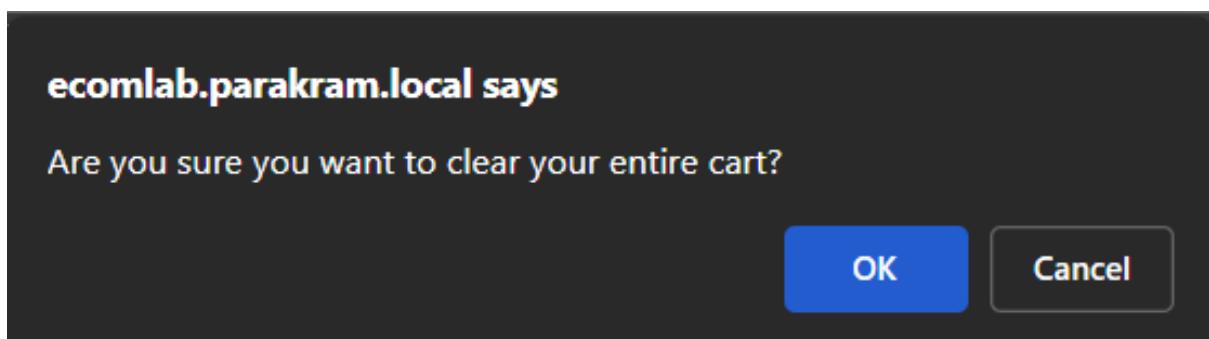


Figure 4: Clear Cart Confirmation.

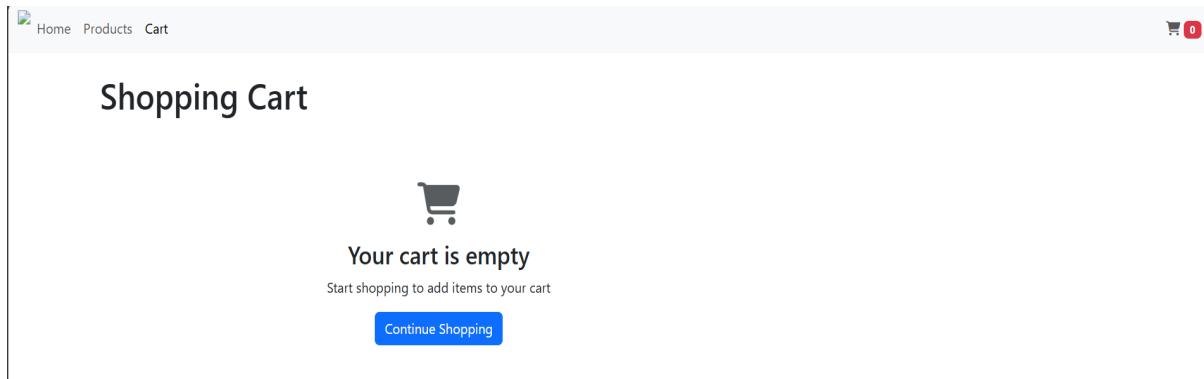


Figure 5: Empty Cart.

## 7. Result

User can add, update, or remove items from the cart. The total quantity and amount update dynamically. Cart data remains stored in the session until the user closes the browser or checks out.

## 8. Conclusion

This lab demonstrated how to use PHP sessions to build a shopping cart. It improved understanding of session management, dynamic page updates, and e-commerce functionality.

## 9. References

- PHP mysqli Prepared Statements
- PHP htmlspecialchars
- PHP Session Examples
- Bootstrap 5 Documentation
- MDN HTTP Reference
- Caddy Web Server Documentation