**E COMM FINAL PROJECT**

**Step-by-Step Lab Manual: Amazon-Like Cart Page Implementation**

**Objective**

The objective of this lab is to create an interactive Amazon-like cart page using HTML, CSS, and JavaScript. This includes:

- Displaying products.
- Adding and removing products to/from the cart.
- Viewing the cart with dynamic updates.

---

**Prerequisites**

Before beginning this lab, ensure you have:

1. A code editor (e.g., VSCode, Sublime Text).
2. A basic understanding of HTML, CSS, and JavaScript.
3. A modern web browser for testing.

---

**Steps**

**Step 1: Create the Base HTML Structure**

1. Open your code editor and create a new file named cart.html.
2. Add the following HTML boilerplate:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Amazon-Like Cart</title>
</head>
<body>
</body>
</html>
```

**Step 2: Add the Header Section**

1. Inside the <body> tag, add a header section for the logo, search bar, and cart button.
2. Use the following code:

```
<div class="header">
    <div class="logo">ShopNow</div>
    <div class="search-container">
        <input type="text" placeholder="Search for products..." />
        <button>Search</button>
    </div>
    <button class="cart-btn" onclick="toggleCart()">
        <span class="cart-icon"><i class="fas fa-shopping-cart"></i></span>
        View Cart
        <span class="cart-count" id="cart-count">0</span>
    </button>
</div>
```

**Step 3: Style the Header Using CSS**

1. Add the <style> tag inside the <head> section.
2. Define the following CSS rules for the header:

```
.header {
    background-color: #232f3e;
    color: white;
    padding: 15px 30px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}
.header .logo {
    font-size: 24px;
    font-weight: bold;
    color: #ff9900;
    cursor: pointer;
}
```

### Step 4: Add the Product Container

1. Below the header, add a product container to display products.
2. Use the following code:

```
<div class="product-container">
    <div class="product" id="product-1">
        <img src="https://cdn.pixabay.com/photo/2012/11/07/02/07/jugs-64975_640.jpg" alt="Product 1">
        <h3>Product 1</h3>
        <button onclick="addToCart('Product 1', 'https://cdn.pixabay.com/photo/2012/11/07/02/07/jugs-64975_640.jpg')">Add to Cart</button>
    </div>
    <!-- Add more products as needed -->
</div>
```

### Step 5: Style the Product Section

1. Add the following CSS rules for the product section:

```css
.product-container {
    display: flex;
    justify-content: center;
    margin-top: 20px;
    flex-wrap: wrap;
}
.product {
    background-color: white;
    margin: 15px;
    padding: 20px;
    border: 1px solid #ddd;
    width: 200px;
    text-align: center;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
}
.product img {
    width: 100%;
    height: 150px;
    object-fit: cover;
    border-radius: 8px;
}
.product button {
    background-color: #ff9900;
    border: none;
    color: white;
    padding: 10px;
    cursor: pointer;
    margin-top: 10px;
    border-radius: 5px;
}
```

**Step 6: Add the Cart Section**

1. Below the product container, add a cart section to display items added to the cart.
2. Use the following code:

```html
<div id="cart" class="cart">
    <h2>My Cart</h2>
    <!-- Dynamically added cart items will appear here -->
</div>
```

3. Add the following CSS for the cart:

```css
.cart {
    margin-top: 30px;
    display: none;
    justify-content: center;
    flex-wrap: wrap;
    padding: 20px;
    max-height: 400px;
    overflow-y: auto;
    border-top: 2px solid #ddd;
}
```

**Step 7: Add JavaScript for Interactivity**

Step 7.1: Initialize the Cart Data

1. Add a <script> tag inside the <body> tag.
2. Initialize an empty object to store cart items:

```javascript
let cartItems = {};
```

Step 7.2: Add Function to Add Items to Cart

Write a function to add items to the cart:

```javascript
function addToCart(product, imageUrl) {
    if (cartItems[product]) {
        cartItems[product].quantity++;
    } else {
        cartItems[product] = { name: product, quantity: 1, imageUrl: imageUrl };
    }
    updateCart();
    updateCartButton();
}
```

Step 7.3: Add Function to Remove Items from Cart

Create a function to remove items from the cart:

```
function removeFromCart(product) {
    if (cartItems[product]) {
        cartItems[product].quantity--;

        if (cartItems[product].quantity === 0) {
            delete cartItems[product];
        }
    }
    updateCart();
    updateCartButton();
}
```

Step 7.4: Update the Cart Display

Write a function to update the cart section dynamically:

```
function updateCart() {
    const cart = document.getElementById('cart');
    cart.innerHTML = '';

    for (const product in cartItems) {
        const cartItem = cartItems[product];
        const cartItemDiv = document.createElement('div');
        cartItemDiv.classList.add('cart-item');
        cartItemDiv.id = `cart-item-${cartItem.name}`;
        cartItemDiv.innerHTML = `
            <h3>${cartItem.name}</h3>
            <img src="${cartItem.imageUrl}" alt="${cartItem.name}" style="width: 100px; height: 100px;
            object-fit: cover; border-radius: 8px; margin: 10px auto;">
            <p>Quantity: ${cartItem.quantity}</p>
            <button onclick="removeFromCart('${cartItem.name}')">Remove</button>
        `;
        cart.appendChild(cartItemDiv);
    }

    if (Object.keys(cartItems).length === 0) {
        cart.style.display = 'none';
    } else {
        cart.style.display = 'flex';
    }
}
```

Step 7.5: Update the Cart Button

Add a function to update the cart button to display the total number of items:

```
function updateCartButton() {
    const cartCount = document.getElementById('cart-count');
    const totalItems = Object.values(cartItems)
            .reduce((acc, item) => acc + item.quantity, 0);
    cartCount.textContent = totalItems;
}
```

Step 7.6: Toggle Cart Visibility

Create a function to show or hide the cart:

```
function toggleCart() {
    const cart = document.getElementById('cart');
    if (cart.style.display === 'none' || cart.style.display === '') {
        cart.style.display = 'flex';
    } else {
        cart.style.display = 'none';
    }
}
```