**E-Commerce\_Application\_on\_IBM\_Cloud\_Foundry\_**

**HTML:**

* This HTML code represents the structure of an e-commerce website called "Artisanal Marketplace." Here's an explanation of the code in 200 words:
* The code starts with the declaration of an HTML5 document type and contains various elements that make up a web page. It includes a title, which identifies the webpage as an "E-commerce website." It links to an external stylesheet called "styles.css" for applying styles.
* The page is divided into sections with header tags that define the website's name as "Artisanal Marketplace" and a navigation bar with links to "Home," "Products," and "Cart."
* The first section introduces the site, inviting users to "Discover Unique Handcrafted Items."
* The "Products" section displays featured products. Two product entries are included with images, titles (e.g., "Artistic Jewelry"), prices, and "Add to Cart" buttons. More products can be added similarly.
* A "Shopping Cart" section is outlined but currently empty, designed to display cart items and a total price.
* Further down, there's a JavaScript form for the checkout page, including fields for name, email, address, and credit card information. A "Place Order" button submits the form, triggering client-side validation.
* The JavaScript function "validateForm" checks if all fields are filled, alerts the user if any are missing, and confirms a successful order placement.
* Finally, there's a footer that states the copyright information for "Artisanal Marketplace."
* This code serves as the foundation for an e-commerce website, providing a user interface for browsing products, adding them to a cart, and proceeding to checkout while ensuring data input validation.

**My Website code:**

<!DOCTYPE html>

<html>

<head><title>E-commerce website</title>

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

<header>

<h1>Welcome to Artisanal Marketplace</h1>

<nav>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#products">Products</a></li>

<li><a href="#cart">Cart</a></li>

</ul>

</nav>

</header>

<section id="home">

<h2>Discover Unique Handcrafted Items</h2>

<p>Explore our collection of artisan-made products.</p>

</section>

<section id="products">

<h2>Featured Products</h2>

<div class="product-gallery">

<!-- Display product images, names, and prices -->

<div class="product">

<img src="image\_1.jpg" alt="Product 1">

<h3>Artistic Jewelry</h3>

<p>$50.00</p>

<button>Add to Cart</button>

</div>

<!-- Add more products -->

</div>

</section>

<section id="products">

<h2>Featured Products</h2>

<div class="product-gallery">

<!-- Display product images, names, and prices -->

<div class="product">

<img src="image\_2.jpg" alt="Product 2">

<h3>Artistic Jewelry</h3>

<p>$75.00</p>

<button>Add to Cart</button>

</div>

<!-- Add more products -->

</div>

</section>

<section id="cart">

<h2>Your Shopping Cart</h2>

<!-- Display cart items and total -->

<div class="cart-items">

<!-- Display cart items -->

</div>

<p>Total: $[Total Amount]</p>

<button>Proceed to Checkout</button>

</section>

<script src="script.js"></script>

<h1>Checkout</h1>

<form id="checkout-form" onsubmit="return validateForm()">

<label for="name">Name:</label>

<input type="text" id="name" required><br><br>

<label for="email">Email:</label>

<input type="email" id="email" required><br><br>

<label for="address">Address:</label>

<input type="text" id="address" required><br><br>

<label for="credit-card">Credit Card:</label>

<input type="text" id="credit-card" required><br><br>

<button type="submit">Place Order</button>

</form>

<script>

function validateForm() {

var name = document.getElementById("name").value;

var email = document.getElementById("email").value;

var address = document.getElementById("address").value;

var creditCard = document.getElementById("credit-card").value;

if (name === "" || email === "" || address === "" || creditCard === "") {

alert("All fields are required");

return false;

}

// You can add more validation for the credit card, email format, etc.

alert("Order placed successfully!");

return true;

}

</script>

<footer>

<p>&copy; 2023 Artisanal Marketplace. All rights reserved.</p>

</footer>

</body>

   </html>

**CSS:**

* This CSS code provides styling instructions for the HTML structure of the e-commerce website. It resets default styles for body, headings, and paragraphs. It applies a light gray background and specifies the Arial font for the entire page.
* The header gets a dark background, white text, centered alignment, and padding. Headings are styled with specific font sizes. The navigation menu items are displayed inline with some spacing.
* Sections receive padding and heading styles, and the product gallery is formatted as a flex container with individual product cards. These cards have a white background, borders, margin, padding, and are centered. Images are made responsive.
* The cart section has a border and padding, and the footer has a dark background, white text, centered alignment, padding, and reduced font size.
* In summary, this CSS code defines the visual design of the e-commerce website, from its overall layout to specific styling for headers, navigation, product display, shopping cart, and footer elements.

**My website code:**

/\* Reset some default styles \*/

body, h1, h2, p {

margin: 0;

padding: 0;

}

/\* Apply a background color to the body \*/

body {

background-color: #f0f0f0;

font-family: Arial, sans-serif;

}

/\* Style the header \*/

header {

background-color: #333;

color: #fff;

text-align: center;

padding: 20px;

}

h1 {

font-size: 24px;

}

nav ul {

list-style: none;

}

nav ul li {

display: inline;

margin: 0 15px;

}

nav a {

text-decoration: none;

color: #fff;

}

/\* Style the sections \*/

section {

padding: 20px;

}

h2 {

font-size: 20px;

}

/\* Style the product gallery \*/

.product-gallery {

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

.product {

background-color: #fff;

border: 1px solid #ddd;

margin: 10px;

padding: 10px;

text-align: center;

width: 250px;

}

.product img {

max-width: 100%;

height: auto;

}

/\* Style the cart section \*/

.cart-items {

border: 1px solid #ddd;

padding: 10px;

}

/\* Style the footer \*/

footer {

background-color: #333;

color: #fff;

text-align: center;

padding: 10px;

font-size: 12px;

}

**JAVASCRIPT:**

* This JavaScript code handles cart and checkout functionality for an e-commerce website. It begins by selecting all buttons with the class "product" and setting up event listeners to add products to the cart. When a button is clicked, it extracts the product name and price, creates a cart item element, appends it to the cart, and updates the total price.
* A separate event listener is established on the cart container to remove items when the "Remove" button is clicked. It deducts the item's price from the total, updates the total display, and removes the item from the cart.
* The code maintains a running total using the `currentTotal` variable and ensures that the cart's content and total price are updated dynamically as products are added or removed.

**My Website code:**

// JavaScript for cart and checkout functionality

const addToCartButtons = document.querySelectorAll('.product button');

const cartItemsContainer = document.querySelector('.cart-items');

const cartTotal = document.querySelector('#cart p'); // Moved this line here

let currentTotal = 0; // Initialize currentTotal outside the event listeners

addToCartButtons.forEach(button => {

button.addEventListener('click', () => {

const product = button.parentElement;

const productName = product.querySelector('h3').textContent;

const productPrice = parseFloat(product.querySelector('p').textContent.slice(1)); // Remove the extra opening parenthesis

const cartItem = document.createElement('div');

cartItem.classList.add('cart-item');

cartItem.innerHTML = `

<p>${productName}</p>

<p>$${productPrice.toFixed(2)}</p>

<button class="remove-item">Remove</button>

`;

cartItemsContainer.appendChild(cartItem);

// Calculate and display updated total

const newTotal = currentTotal + productPrice;

currentTotal = newTotal; // Update currentTotal

cartTotal.textContent = 'Total: $${newTotal.toFixed(2)}';

});

});

// Remove item from cart

cartItemsContainer.addEventListener('click', event => {

if (event.target.classList.contains('remove-item')) {

const cartItem = event.target.parentElement;

const itemPrice = parseFloat(cartItem.querySelector('p:nth-child(2)').textContent.slice(1)); // Remove the extra opening parenthesis

const newTotal = currentTotal - itemPrice;

currentTotal = newTotal; // Update currentTotal

cartTotal.textContent = 'Total: $${newTotal.toFixed(2)}';

cartItem.remove();

  }

});

**CONCLUSION:**

* HTML, CSS, and JavaScript are fundamental technologies for web development. HTML structures content, CSS styles it, and JavaScript adds interactivity.
* They work in harmony to create dynamic and visually appealing websites. Mastering these languages empowers developers to build responsive, user-friendly web applications that drive the modern internet.