**University of Wolverhampton**

**School of Engineering, Computational and Mathematical Sciences**

**5CS020 Human-Computer Interaction**

**Workshop 7 – Creating a simple shopping cart interface**

In this workshop you will be creating simple mock shopping cart in HTML CSS and JavaScript.

**Create a simple mock shopping web app**

Creating a simple mock shopping web app with 5 products in HTML, CSS, and JavaScript, and use localStorage for all data.

Set up the HTML file: Create a new file called "index.html" and set up the basic structure of the HTML document. Add a title and include links to your CSS and JavaScript files.

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>My Mock Shopping Website</title>**

**<link rel="stylesheet" href="style.css">**

**</head>**

**<body>**

**<header>**

**<h1>My Mock Shopping Website</h1>**

**</header>**

**<main>**

**<section>**

**<h2>Products</h2>**

**<ul id="product-list">**

**<!-- Product list will be dynamically generated here -->**

**</ul>**

**</section>**

**<section>**

**<h2>Shopping Basket</h2>**

**<ul id="basket-list">**

**<!-- Basket list will be dynamically generated here -->**

**</ul>**

**<p>Total: <span id="total"></span></p>**

**<button id="checkout-btn">Checkout</button>**

**</section>**

**</main>**

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

**</body>**

**</html>**

Create the CSS file: Create a new file called "style.css" and add some basic styles to your HTML elements.

**header {**

**background-color: #333;**

**color: white;**

**padding: 20px;**

**}**

**h1, h2 {**

**margin: 0;**

**}**

**#product-list, #basket-list {**

**list-style: none;**

**padding: 0;**

**}**

**#product-list li {**

**border: 1px solid #ccc;**

**margin-bottom: 10px;**

**padding: 10px;**

**}**

**#basket-list li {**

**border: 1px solid #ccc;**

**margin-bottom: 10px;**

**padding: 10px;**

**}**

**#total {**

**font-weight: bold;**

**}**

Set up the JavaScript file: Create a new file called "script.js" and add the following code to it.

**// We define an array of products that includes 5 objects, each representing**

**// a product. Each product object has an id, name, and price property.**

**const products = [{**

**id: 1,**

**name: 'Product 1',**

**price: 10.00,**

**},**

**{**

**id: 2,**

**name: 'Product 2',**

**price: 20.00,**

**},**

**{**

**id: 3,**

**name: 'Product 3',**

**price: 30.00,**

**},**

**{**

**id: 4,**

**name: 'Product 4',**

**price: 40.00,**

**},**

**{**

**id: 5,**

**name: 'Product 5',**

**price: 50.00,**

**},**

**];**

**// We check if there is any data in the basket key of localStorage.**

**// If there is, we parse it as a JSON string and assign it to the basket variable.**

**// If not, we set the basket variable to an empty array.**

**let basket = JSON.parse(localStorage.getItem('basket')) || [];**

**// We use the document.getElementById() method to get references to several**

**// HTML elements on the page: the productList element, the basketList element,**

**// the total element, and the checkoutBtn element.**

**const productList = document.getElementById('product-list');**

**const basketList = document.getElementById('basket-list');**

**const total = document.getElementById('total');**

**const checkoutBtn = document.getElementById('checkout-btn');**

**// We define a function called displayProducts(), which is responsible**

**// for generating the list of products and displaying it on the page.**

**function displayProducts() {**

**// We first clear the existing product list by setting the innerHTML**

**// property of the productList element to an empty string.**

**productList.innerHTML = '';**

**// Iterate over each product in the products array. For each product,**

**// we create HTML elements to display its name, price, and an**

**// "Add to Basket" button.**

**for (let i = 0; i < products.length; i++) {**

**const product = products[i];**

**// Create HTML elements**

**const li = document.createElement('li');**

**const h3 = document.createElement('h3');**

**const price = document.createElement('span');**

**const addToBasketBtn = document.createElement('button');**

**// Add product data to HTML elements**

**h3.textContent = product.name;**

**price.textContent = product.price.toFixed(2);**

**addToBasketBtn.textContent = 'Add to Basket';**

**addToBasketBtn.setAttribute('data-id', product.id);**

**// add event listeners to the "Add to Basket" buttons using the**

**// addEventListener() method, so that when the user clicks the button,**

**// the addToBasket() function is called.**

**addToBasketBtn.addEventListener('click', addToBasket);**

**// Append the HTML elements to the productList element,**

**// which adds them to the page**

**li.appendChild(h3);**

**li.appendChild(price);**

**li.appendChild(addToBasketBtn);**

**productList.appendChild(li);**

**}**

**}**

**// This function called when the user clicks an "Add to Basket" button.**

**// This function retrieves the product ID from the data-id attribute of the button,**

**// checks if the product is already in the basket, and either increments the**

**// quantity of the existing product or adds a new product to the basket.**

**function addToBasket(event) {**

**// Get product ID from button attribute**

**const productId = parseInt(event.target.getAttribute('data-id'));**

**// Check if product already in basket**

**let productIndex = -1;**

**for (let i = 0; i < basket.length; i++) {**

**if (basket[i].id === productId) {**

**productIndex = i;**

**break;**

**}**

**}**

**if (productIndex !== -1) {**

**// Increment quantity if product already in basket**

**basket[productIndex].quantity++;**

**} else {**

**// Add new product to basket**

**const product = products.find((product) => product.id === productId);**

**product.quantity = 1;**

**basket.push(product);**

**}**

**// Update basket in local storage**

**localStorage.setItem('basket', JSON.stringify(basket));**

**// Refresh basket list and total**

**displayBasket();**

**displayTotal();**

**}**

**// Remove product from basket**

**function removeProduct(event) {**

**// Get product ID from button attribute**

**const productId = parseInt(event.target.getAttribute('data-id'));**

**// Remove product from basket**

**basket = basket.filter((product) => product.id !== productId);**

**// Update basket in local storage**

**localStorage.setItem('basket', JSON.stringify(basket));**

**// Refresh basket list and total**

**displayBasket();**

**displayTotal();**

**}**

**// Display basket on the page**

**function displayBasket() {**

**// Clear existing basket list by setting the innerHTML property of the**

**// basketList element to an empty string.**

**basketList.innerHTML = '';**

**// Iterate over each product in the basket array. For each product,**

**// we create HTML elements to display its name, price, quantity,**

**// and a "Remove" button.**

**for (let i = 0; i < basket.length; i++) {**

**const product = basket[i];**

**// Create HTML elements**

**const li = document.createElement('li');**

**const h3 = document.createElement('h3');**

**const price = document.createElement('span');**

**const quantity = document.createElement('span');**

**const removeBtn = document.createElement('button');**

**// Add product data to HTML elements to display its name, price,**

**// quantity, and a "Remove" button.**

**h3.textContent = product.name;**

**price.textContent = product.price.toFixed(2);**

**quantity.textContent = `x${product.quantity}`;**

**removeBtn.textContent = 'Remove';**

**removeBtn.setAttribute('data-id', product.id);**

**// Add event listener to button so that when the user clicks the button,**

**// the removeProduct() function is called.**

**removeBtn.addEventListener('click', removeProduct);**

**// Append HTML elements to basket list**

**li.appendChild(h3);**

**li.appendChild(price);**

**li.appendChild(quantity);**

**li.appendChild(removeBtn);**

**basketList.appendChild(li);**

**}**

**}**

**// Responsible for calculating the total price of the items in the basket**

**// and displaying it on the page.**

**function displayTotal() {**

**// iterate over each product in the basket array and calculate the total**

**// price by multiplying each product's price by its quantity.**

**let totalPrice = 0;**

**for (let i = 0; i < basket.length; i++) {**

**const product = basket[i];**

**totalPrice += product.price \* product.quantity;**

**}**

**// Display total price on the page by setting the textContent property**

**// of the total element.**

**total.textContent = totalPrice.toFixed(2);**

**}**

**// Initialize page with the product and basket displays.**

**displayProducts();**

**displayBasket();**

**displayTotal();**

**// Add event listener to checkout button so that when the user clicks the button,**

**// the onCheckoutClicked() function is called.**

**checkoutBtn.addEventListener('click', onCheckoutClicked);**

**// Handle checkout button click event for simulating the checkout process.**

**// This function calculates the total price of the items in the basket,**

**// displays an alert message to the user indicating the total price, clears**

**// the basket data in localStorage, and refreshes the basket and total displays**

**// on the page by calling the displayBasket() and displayTotal() functions.**

**function onCheckoutClicked() {**

**// Calculate total price**

**let totalPrice = 0;**

**for (let i = 0; i < basket.length; i++) {**

**const product = basket[i];**

**totalPrice += product.price \* product.quantity;**

**}**

**// Simulate checkout process**

**alert(`You have been charged ${totalPrice.toFixed(2)} for your purchase.`);**

**// Clear basket and update page**

**basket = [];**

**localStorage.removeItem('basket');**

**displayBasket();**

**displayTotal();**

**}**