Assignment 6B – Interactive Shopping Cart (JavaScript)

Assignment Links

Website Link: https://rkavari.github.io/root/cart.html
GitHub Repo Link: https://github.com/rkavari/homework_6b
Source Code Link:
https://github.com/rkavari/homework_6b/tree/main/homework_6b

Resources

- **Images**: I used photos sourced from Unsplash. My understanding is that the images are free-to-use. However, just in case, here are the names of the photographers whose work I used: Jamie Street, Ana Margarida Almeida, Dominik Kempf, Ali Kazal, Ira Pavlyukovich, James Lacy, 玲红 贺, Damir Spanic, and Omid Armin.
- **Troubleshooting/Documentation Resources**: W3Schools and StackOverflow were heavily referenced during the completion of this assignment. Whenever necessary, I followed the examples provided to write my own code for my website.

Reflection

To make my work for Assignment 6B more efficient, I had to clean up much of the code I wrote for Assignment 6A. For example, I had separate functions for the selection of every size and color. While the code worked and fulfilled its requirements, using it, as is, would make adding interactivity to the shopping cart much more difficult. Therefore, to make the code clearer and easier to work with, I condensed the functions into one for selecting any size and another for color. Accordingly, I eliminated any unnecessary styling in the CSS. By doing this, it was much simpler to implement an interactive shopping cart.

However, that does not mean I did not experience any challenges or bugs. The two most common types of bugs I encountered were due to syntax and runtime errors. When updating my code, I made careless errors while typing that resulted in syntax errors. Since there was also so much code to handle, I occasionally would accidentally reference elements by the wrong name. To overcome those minor but burdensome errors, I used the Developer Tools panel to understand what was going wrong with my code. This strategy was effective, and I used it to overcome some of my more significant challenges.

One of the first major challenges I faced was properly displaying the items in the cart. I initially thought that only one item appeared in the cart, despite the count stating that there should be seven. After inspecting the page, I realized that there were actually seven items appearing in the cart. They were just at the same place, thus looking like there was only one item there. I ended up editing the CSS so that the items' positions were relative, based on their div.

Another challenge I encountered was setting the size and color drop downs on the cart page to select the actual size and color of the item. The code that I initially wrote used the selected attribute on each line and would equate to true/false after checking the item's properties in local storage. The console would properly indicate true or false, but it would still select the incorrect size and color on the dropdowns. I added more items to understand why this happened and eventually realized that the bug was occurring because the selected attribute cannot be used on more than one option in a dropdown. I ended up using if/else statements to check if the item was a certain size or color, and if it were, I would use the line with the selected attribute. If not, the line of code without the selected attribute would be used.

The last challenge I encountered while completing this assignment was in regards to removing an item from the cart. I assigned each added item a productid and needed to retrieve the relevant productid so that the item could be deleted. I tried many different approaches that did not work. Eventually, I referenced StackOverflow and learned how I could use event handlers to locate the item.

Programming Concepts

This portion of the assignment references some of the concepts given as examples in the slide deck shared by the TA's on the topic.

1. Representing Shopping Cart with Array of Objects: I created an empty array in my code that would store items (objects) created after the user clicks on the 'Add to Cart' button. Clicking on the button would trigger a function that would add the item, along with its properties, to the cartItems array.
2. Representing Cart Items with Objects: As discussed in the previous concept, I represented items added to cart with objects. Once the user clicks 'Add to Cart', the properties of the item, such as its name, selected color, and selected size, are pushed into the cartItems array.
3. Using Conditional Statements to Check Selected Property: I referenced this concept in my reflection. I used if/else statements to check if options in the size and color dropdowns matched the item's size and color. If there was a match, then the line of code that used the selected attribute would be executed.
4. Using Event Handlers to Locate Selected Items: As discussed in the reflection, I used event handlers in order to locate items such as the productid. By using the event to locate the productid, I was able to remove the corresponding item from the cart.
5. Using Filter Function to remove an Item and Display Remaining Items: I finally used the productid that I retrieved in the filter function. By doing so, I was able to display all items except the item that had been removed.