Preethi Surapaneni November 7 2021

Reflection for Assignment 6B:

This part of the project was the most fun for me because the functionality of all the backend javascript work that was put in was finally visible. This leads me to my first lesson:

During Assignment 6A a lot of the javascript code could have been set up to ensure that 6B was smooth sailing. For example storing variables (specifically my orders list) in a way that allows them to be accessed in different functions, storing variables using session as well as local storage. It was important for me to go back and modify all the storage methodologies of each variable (orders list, cart total, and attributes for each product).

This taught me to always have forethought regarding what I might need to do with the data later on, so that future deliverables become intuitive rather than needing to go through reprogramming like I had to do for this assignment.

This lesson ties into other bugs I faced. One particular bug that made me rewire a couple elements was the remove feature in the shopping cart. I had ensure I stored the cart total value locally so that it could be automatically changed when a. items were added and b. items were removed while ensuring it remained the same when the page was reloaded or if the user navigated away and came back. This then also made it possible for me to calculate the total price of the cart.

I had to ensure that if the product detail page was reloaded the dropdown options were reset to default and the page no longer displayed the feedback of the last ordered item while ensuring that they could still view the cart total amount on each page on the top right corner. This reduces confusion for the user and makes it a clean UI.

One bug that I was able to fix that massively improved the UI was being able to transfer the attributes of the product selected straight to the shopping cart page, without needing to edit it once in the cart. Initially I was unable to dynamically get the picture of the product into the shopping cart and tried to modify the css to have it reshape the same picture. I realized it was easier to be able to duplicate the picture and then modify the size based on the requirements of the cart page.The shopper completes all the selection of glaze and quantity in the product detail page so when they hit add all their information has been saved in the cart.

Another part I took a long time to figure out is how to prevent them from adding a duplicate order (because that would then effect the *remove* function). I was able to figure this out by throwing feedback through an alert to the user mentioning that they have already added this to the cart. I decided to add more functionality to solve this problem by introducing *plus* and *minus* features to the quantity part of the cart item in the shopping cart itself, however did not have enough time to do so.

Programming Concepts:

1. Storage- LocalStorage was a programming concept that I used throughout with regard to dynamically updating cart, cart total, orders and attributes of each order. An additional concept I learned a lot about while implementing localStorage was sessionStorage- distinguishing the two helped me decide what was appropriate where. For example while I only want sessionStorage for my dropdown selections because if they refresh the page they should see default options so they can start fresh. However I want localStorage when adding items to cart or calculating totals because they should be able to navigate back to that page and still find that their cart is ready to be checked out.

2. Using Arrays I was able to store most of the information about one cart item which I then accessed as needed. Setting up my arrays correctly and accessing each element correctly made most of the assignment of variables easier.

3. Events were another programming concept that I used throughout the webpage given that each click or load would result in different (or in some cases the same) output. When clicking buttons or even on load for my shopping cart page, I was able to use events to trigger various the actions I needed whether it was to move from page to page: *return to shop* or submitting orders: *add to cart, remove from cart,* or updating my cart total to be displayed on each page.

4. For loops- I used for loops in most functions that required me to display or update the users current selection of orders. this helped me be be organized and made my code readable to be able to easily access each part of what I needed to manipulate. (for removing the item, I used a loop to both remove the item as well as update other items that would change because of it: cart total and order list in the local storage.

5. The last programming concept that helped me throughout and is more so related to coding practices was code tracing through intermittent testing. I used console.log and alerts to trace my code as I was writing it so that any errors in logic could be caught right away rather than waiting till wrong code was fully integrated before I found out. This especially helped me when adding and removing items from cart and updating list to identify how the elements were being stored (as objects) and how I could view what those objects were (JSON.stringify).