Assignment 6 Reflection

Melissa Shi

Section B Cole Gleason

Challenges and How to Overcome

During assignment 6, I encountered multiple challenges as this is the first time I used JavaScript, therefore, it was a trial and error process for me. The challenges that I faced falls in various scales. As basic as how to get element value from html, and some were higher-level, such as how to update my local storage when deleting a product from my shopping cart page.

The first challenge that I faced was how to get element value from my html page. I started by reviewing lab exercises and modified my code based on what the example codes. I soon realize that I could not succeed in using JavaScript and started using JQuery, which proved to be much easier.

The second major obstacle that I faced was how to store my array, consisted of multiple object, into my local storage without being overwritten whenever I added new items from a different product page. For example, after I stored an array from walnut cinnamon roll web page, my local storage will be overwritten if I stared adding new object to in my pecan cinnamon roll web page. To overcome this obstacle, I used *console.log()* extensively, and returned my array at every step. From this practice, I was able to debug because I realize it is happening because every time I load a new page, I created a new empty array to store the objects created. The solution that I came up with was to use an if/else statement and select with array to modify based on whether local storage is empty or not. This approach successfully solved the problem.

The last major hindrance came up when I attempted to update my local storage based on which item to delete from my shopping cart. Initially, I assigned each item with a unique number based on the order that they were populated into the table on shopping cart page, then I removed them from the array using index. This approach was proven to be wrong because each object's index will change when the array, which holds all the objects, was being modified. To solve this problem, I revisited my constructor and added a new key called id, which is meant to give each object a unique identifier. The unique identifier was generated with *Math.random()*. This unique identifier was then used to find the correct to be removed from the array. Lastly, the modified array gets stored and overwrites what local storage.

Learnings

Assignment 6 has been extremely challenging and insightful. I now have much in-depth understanding of MVC (model view control), as well as how to create meaningful interactions. This experience also provided me with a peek of what the job of a front-end developer like, which could help me later on with communicating and looking from their perspective. Additionally, I learned that sometimes it is necessary to make compromises because the cost of implementing a design would outweigh the aesthetic gain. For example, in my original design, I wanted to update only the quantity when a new item is being added when there exist an item with the same flavor and glazing. However,

when I started implementing the function, I realized the time cost for me to figure out how to implement would be much longer than desired.

Design Changes

During the coding process, couple changes were made as part of design improvement, some for satisfying assignment requirements, others due to my coding limitations. Some of the major changes includes removing popup screen on clicking cart icon, restructure content on shopping cart page, and minimize functionality from order summary page.

Design Improvement

One of the changes that I made to improve my design was to re-position the delete button on my shopping cart page to enhance the grouping of functions. The delete button was originally next to product image, which was later changed to be below product quantity. This will help user better navigate the delete function because both product quantity and delete are associated with product count.

Assignment Requirement

As part of the assignment requirement, there must be associate changes to product page when specifications such as quantity or glazing are being selected. To satisfy this requirement, I added a label to each product image. When a glazing option is selected, the label will automatically update to reflect which glazing option has been selected.

As part of satisfying requirements, I removed my popup screen design from Homework 3 and redesigned my shopping cart page to purely review shopping cart instead of including placing order. This is to allow user to review and confirm what they have added before placing orders.

Coding limitation

Due to my skill limitations, I was not able to update product quantity in the shopping cart when the selected flavor and glazing are the same. Instead, all products are added as new item in shopping cart.

Credits

1. W3School
2. Stack Overflow
3. Asit Parida
4. Kunal Bhuwalka