Assignment 6B for PUI by Ritu Roychoudhury

NOTE: This is to bring into notice that I have submitted the assignment late on Sunday, November 3. In canvas, I had already made a submission of my GitHub Repo because I realized I needed more time and the assignment was closing on Thursday. Not knowing how to submit it after Thursday, I had made an empty submission. Please consider the GitHub push made on November 3 for this assignment. I am extremely sorry for my late submission. This assignment took me more than 1.5 weeks to submit but I have put the effort to ensure the website features work properly.

Work + Reflection

GitHub Repo: https://github.com/rituroy19/rituroy19.github.io/tree/master/assgn6b
Website: https://rituroy19.github.io/assgn6b/home.html

1. The following pages have the requirements for 6B:
    a. Home Page
    b. Rolls Page (Product Browsing Page)
    c. Walnut Cinnamon Roll Page (Product Detail Page)
    d. Shopping Cart Page
    e. There are other HTML pages such as the About Page and other Product Detail Pages for every product sold, but they do not have javascript functionality. ONLY Walnut Cinnamon Roll Page has javascript functionality satisfying the requirement for this assignment.
2. The following javascript functionalities have been worked on:
    a. Add to Cart feature
    b. Cart number updates on each of the above pages based on actual selection/removal of items by the user.
    c. On Shopping Cart Page, you can click on remove, next to the item you have added, to remove it from the shopping cart.
    d. All adding of items and removal from shopping cart updates the storage accordingly as well.
    e. As you remove items, or add items to the shopping cart, the total price also updates accordingly.
    f. On the Walnut Cinnamon Roll page, there is a carousal that shows you additional products and it is linked to those pages as well. (Extra Credit work)

Reflection

These were the major challenges I faced while working on this assignment:

1) **Printing the shopping cart into HTMl from the storage**: My shopping cart's HTML had 3 different divs as columns for describing Quantity, Glaze and Prize of the selected item. In my storage, I was using an array of arrays, and under each sub-array I had stored the entire selection (item, quantity, glaze and price) as one single object. I was having a

difficult time to get the entire object appear in HTML under the different headings the divs represented. I wasn't able to figure out how to do it at that time.

a) I decided to change my HTML divs in such a way that I only need to call one div element by ID name and print the entire object containing the name of item selected, with their quantity, glaze and price together in a table. Hence, for each selection made and added to cart, on the shopping cart page, it will add a new row to the table and write the entire object's properties cell by cell in it. This also ensured I could create a column where a 'remove button' could be added along with each row. I made CSS adjustments with padding to align items properly.
b) I learned that I need to plan the process of how the shopping cart logic will work ahead of writing my HTML, CSS and Javascript files. A good plan helps with structuring the work and designing each web page effectively and applying the functionality with Javascript. I also learned there were different ways of getting around this problem. One way was the way I described - have one single div element and create a table row for each object, sourcing the object's properties one by one and adding them to each cell of the row one by one. Another process was to make different div elements for columns as I had originally done and call them individually and to print each object's property.
c) In the future, I will plan it ahead based on the layout I desire for the web page and modify my storage objects/div elements before so that I do not have to make major changes.

2) **The way I was storing my selection into the local storage**: While trying to organize all of my selections for a product, I ended up making a single object of all of them and then stored them in an array. Hence, my local storage became an array of objects. This worked perfectly fine when I had to write the entire object in a table on my shopping cart page (HTML) but it became a huge problem to remove these objects from the array when a person wanted to remove items from shopping cart. I could not use splice as I could not delete an object but only remove object's property. I needed to get rid of the entire object that the user would have selected.
    a) I learned that I cannot delete objects because the only way to do that is to remove all the references to the object and I was not able to do that with my limited knowledge of javascript. I also learned that I could use splice function to cut out arrays but this required me to somehow make my array of objects to an array of arrays.
    b) Hence, I changed the way I stored the selection. I still captured the selection in a single object but passed it as an array of objects and then each of those arrays were pushed to a bigger array. Hence, I had an array of arrays and each of those sub-arrays contained individual objects holding properties of the selections made by the user. This ensured I could splice the array when a user wanted to remove it from the cart and I could also print the object in the table of the shopping cart page.

    c) In the future, I will think of how exactly I should store my storage based on the functions and features I require that edits my storage accordingly so that I can achieve all the functionalities and not just some of them.
3) **Sourcing HTML elements and Storage items**: I struggled a lot to figure out how to use individual HTML elements' and Storage items' index to run an operation based on the selection. This was particularly important when I needed to design how the remove from cart feature will work. I had to source the index of the row in HTML that was deleted to trace which product was deleted and then use that index to locate the index for the storage element. I did not know how to do this at all with my existing knowledge.
    a) I learned by searching on google that I can use index to locate and perform the operation I wanted to run.
    b) To source HTML element, I used event.target to source the click and then used parentElement.parentElement.rowIndex to get the row deleted from HTML. I used a logic to locate the index of the storage items so that I can remove the exact one that the user wanted. I run a loop to locate that index and then spliced it and set storage again to override the cart storage so that it reflects the current and correct items in the cart.
    c) In the future, I will use this method to design my web page in a more powerful manner as I can create features with a lot of flexibility.

Programming Concepts

- **Variables**: I learned how to declare variables in different types and how to parse them to another type so I can run operations on them with other variables of the same type. I use parseInt to convert a string to an integer to add and calculate my cart items, etc.
- **Control Structures**: I learned how to use a for loop to iterate through an array's index one by one and select a certain one with a particular logic to perform a specific operation. I use for loop to iterate through an array and select the array within. I then use splice function to remove that array containing the object I want to remove.
- **Data Structures**: I learned what an object is and how I can store it in a storage in different ways. I learned I can create multiple objects for different selections or create one object with properties containing those selections of users. I also learned that I could store these objects in arrays and I can make an array or array if required for certain functionalities.
- **Syntax**: I learned the different syntax used for different things such as how to declare a variable, objects, functions, etc. eg: in order to define a function, I have to do: *function addToCart(){* <logic/process> } and more. I learned the various rules for combining symbols that help me achieve different functionalities in javascript.
- **Tools**: I used the inspect element tool to evaluate my HTML and CSS elements by visually understanding each CSS styling on every HTML element and so on. I also used the console to understand how the javascript is running and print values using console.log() to test and understand how the function is working.