Megan Lin

Reflection + Programming Concepts

One bug I encountered is adding and retrieving the array in sessionStorage to show all the added items in the cart page. To debug, I added alert statements to see why the items are not showing up on the cart page. I saw that the statements print out “undefined” and I immediately knew that I wasn’t adding the items into the session storage correctly. Therefore, I went through my onload function for the product details page and found out that I did not push the new cinnamon object correctly to the array, as I was using “append” instead of “push”. After fixing that, I realize that I would also need to directly push that array to the sesssionStorage instead of doing it in the addToCart function or else all the information about the array would be lost (as show by the “undefined” alert messages). After fixing these issues, I was able to successfully add and retrieve the cart items.

Another bug I faced is implementing the delete button. I was having trouble finding a way to identify the exact div element I need to delete. After inserting alert statements and getting “null” for the div elements I tried to search, I realized that I needed to give each div element an ID that would match that of the one passed into the parameter from the delete function. To do so, I decided to add an num attribute to the div element. Then, I would search the element by class names and then loop through the elements to find the one whose ID matches that of the value passed in from the parameter. After doing so, I was able to see the div element removed on the cart page.

I have used localStorage to store the product details information as shown in the onload function. I used it because information stored in the localStorage will persist after closing the browsers. This means that users who return to the website later can still see the cinnamon item they clicked on from last session. I used sessionStorage to store the cart information as shown in the onLoad and showCart functions. I used it because information stored in there will disapeear when the session ends. This is convenient because the users will not be stuck with the cart items they have already bought from the last session and will be able to start anew each time. I used JSON.stringify to store an array in the sessionStorage because sessionStorage is not able to directly store the array. I used JSON.parse to store parse the items in the sessionStorage so I can access these items as arrays. Lastly, I used to constructors to build each cinnamon object as shown in the first three functions of the JS file. I used them so that I wouldn’t have to set some of the default attribute values (such as description and name) each time that I am creating a new object.