

Repo Link: <https://github.com/EvelynLi06/evelynli06.github.io/tree/main/HW6B>

Website Link: <https://evelynli06.github.io/HW6B/>

I think a lot of the bugs that I encountered this time also involves alignment. Initially I tried to go straight into the JavaScript DOM part for adding an item to the shopping cart page, and then the texts and images went misaligned everywhere. I think what was helpful was to first add a dummy item to the page, and get all the alignment issues figured out, and then copy that as a template into the JavaScript for adding items so that all class names matches, and then I don't have to worry about styles. Session storage is definitely a new concept for me as well, so I encountered a few bugs for adding, removing, and resetting the items. But I've worked with Django Sqlite database before, so picking up session storage and debugging for this part was not overwhelming. I think YouTube tutorial along with seeing other examples helped me a lot with pinning down what I wanted to do and how to achieve that. Overall, I think seeing examples online has been a super helpful way for me to debug and improve my code.

Programming concepts:

1. JSON.stringify

- I found JSON.stringify to be very helpful when I was adding an item into the session storage. I can just pass in a JavaScript object to JSON.stringify and store this in the storage.

2. sessionStorage.setItem

- I used setItem to add items to the session storage when user click 'add to cart'
- Example:

```
let item = {"product": product, "glaze": glaze.innerText,  
            "qt": qt.innerText, "price": price.value,  
            "save": save.value};  
sessionStorage.setItem(key, JSON.stringify(item));
```

3. sessionStorage.removeItem

- I used sessionStorage.removeItem to support deleting items from the shopping cart

```
function deleteItem(i){
  const key = sessionStorage.key(i);
  sessionStorage.removeItem(key);
  loadCartHeader();
  loadCart();
}
```

4. sessionStorage.key

- I used sessionStorage.key to loop through the length of the session storage and get the item from each key

5. sessionStorage.getItem

- I used sessionStorage.getItem to get items from session storage to load the cart page

6. JSON.parse

- I used JSON.parse after getting an item from session storage to 'restore' the JavaScript Object format of an item from the storage

```
for (var i=0; i < sessionStorage.length; i++) {
  const key = sessionStorage.key(i);
  let item = JSON.parse(sessionStorage.getItem(key));
  let name = item.product;
  let glaze = item.glaze;
  let qt = item.qt;
  let price = item.price;
  let save = item.save;
```