Assignment 6: Adding Functionality to a Website with JS

Jessica Nip / Section A

Low and High-Fidelity Prototypes: Design Choices

The low-fidelity prototype was kept the same from the previous assignment, in order to:

1. Create consistency among pages
   The navigation bar, header style, body style and button design were kept identical.

2. Follow the heuristic of aesthetics and minimalistic design
   By limiting the UI elements to the bare essentials — text with varying hierarchy and a table.

Two high-fidelity prototypes were created in the assignment. V1 referenced the design of the low-fidelity mock up, and V2 was an iteration made during the implementation phase. I learned that it was better to create a separate column for “glazing” during implementation, in order to clearly differentiate the product categories. This also improved the design as the product name and glazing were no longer chunked together, which were difficult to read at a glance.

What challenges or bugs did you encounter and how did you overcome the challenges?

Challenges I encountered and ways I overcame them:

1. Creating a live shopping cart item count (across all pages)
   I created the function cartName() to update the cart item count in the navigation bar across all pages. What I found was that after adding a new item to cart, the item count would only update when I refreshed the page. The challenge was to simultaneously update the item count when the user selects the “Add to Cart” button.

   I attempted to create a separate function that reloads the function without reloading the entire page, which was unsuccessful. In the end, I called cartName() within the addCart(flavor) function, which easily allowed the update to happen upon “onclick” of the “Add to Cart” button.

2. Disabling previous selection when new selections were made (in product detail page)
   When the user selects quantity and glazing in the product detail page, the selected buttons changed in color (from white to grey). However, I struggled to disable previous selections in the case where users change their selection before adding to cart.

   To overcome this challenge, I attempted to use indexing to identify which button was clicked, then create constraints to revert color for the ones unselected. However, I found that running a style update of “button.style.background=’none’;" for all the buttons at the end of each mouse click was less complex and more effective.