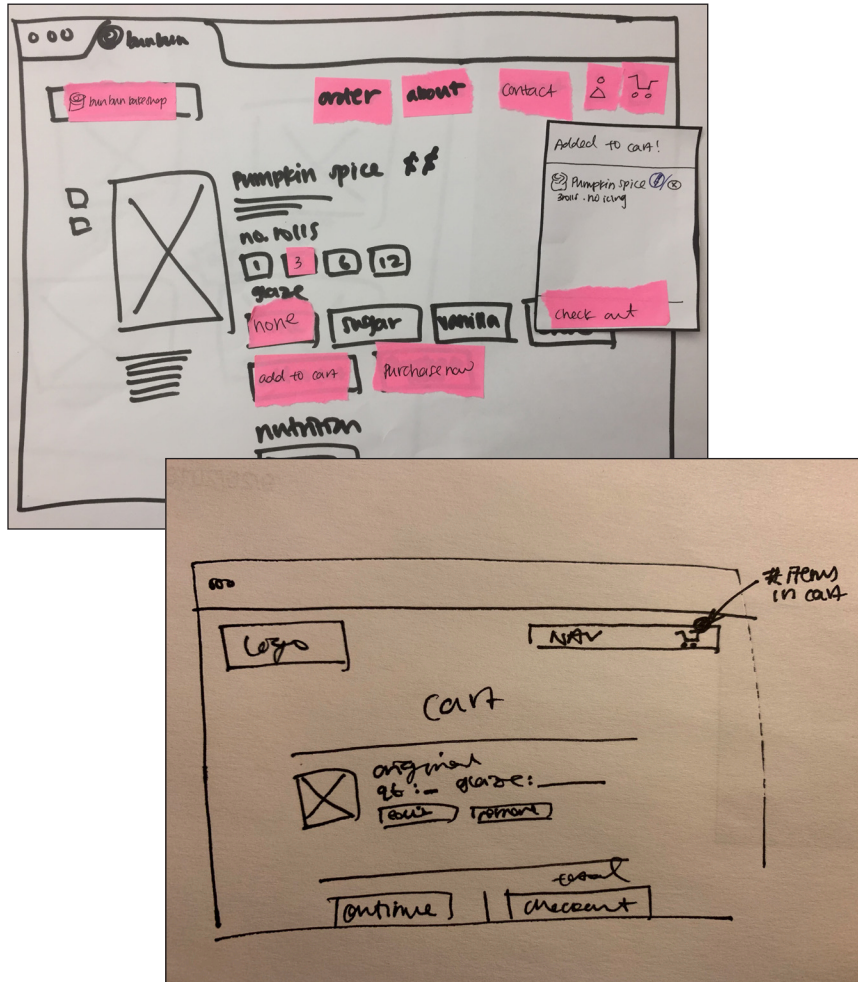
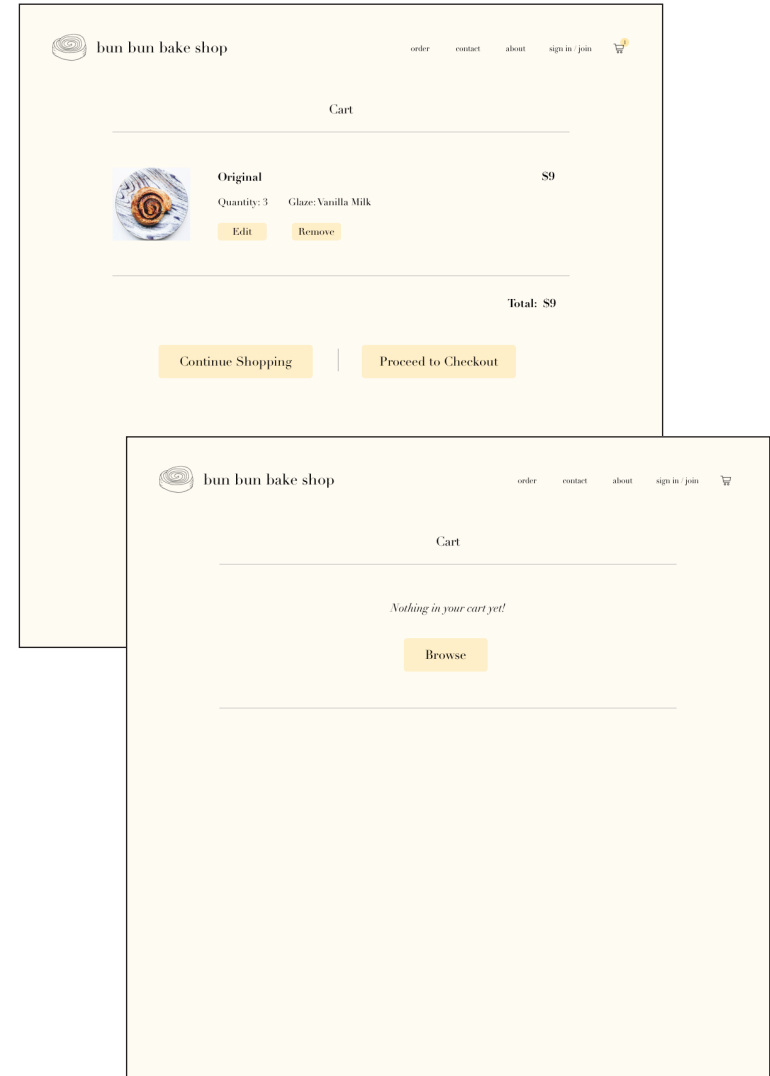


Assignment 6: Implementing the Shopping Cart

Paper Prototype



Sketch Prototype



I originally planned for some form of a shopping cart during the paper prototyping stage, but ended up making drastic design changes prior to the Sketch prototyping stage. Therefore for this assignment, I implemented the shopping cart as I had designed in Sketch. The two paper prototypes are from before the Sketch prototype assignment, and the Sketch prototype pages are in my previous assignment.

Shopping Cart Design Changes + Implementation

As the first paper prototype only had a pop-up window for a shopping cart, I decided to get rid of the pop-up and design a separate cart page. Instead of a pop-up, there would be a small number indicating the number of items currently in your shopping cart. As for the cart page, I wanted to list all the selected options, the image, as well as total price. The user would have the option to "Continue Shopping" or "Checkout," allowing the user to navigate freely between pages.

I implemented the new paper prototype onto Sketch almost identically, except with the set website styles (color, font, etc).

Writing Javascript for the shopping cart and different elements of the page (such as the rotating banner images on the Homepage) was really difficult because there were a lot of variables to keep track of such as image, flavor, quantity and glaze. I also had to then separately keep track of the cart's subtotal. I eventually resorted to arrays as we had discussed in section. Having taken C++ in the past, storing everything in arrays made the most sense to me, so I took that route.

The biggest bug I encountered was deleting entire items from the cart. Getting rid of the cart item on the cart page seemed simple enough, but deleting the item from my localStorage data was an issue. How do I tell it which items in each of the arrays to delete? I eventually resorted to adding unique IDs to each cart item div, slicing and parsing the number part of the ID ("no-1" --> "1" --> 1) into an int, which was actually also its index in each of the arrays. I then used the index number to delete the items from arrays, and refreshing on each delete-button press to update localStorage and what we'd see in the cart. This is probably not the best or most efficient way to do it, but I couldn't figure out a better way.

Another bug I had was implementing changing images on only the Original flavor detail page. ***This detail page is the only page that has the ability to change images based on glaze selection*** (because of my shortage of images). However, I couldn't figure out how to tell that part of my JS to only run on the Original page. Being new to the syntax, it was just a small syntax error that was preventing it from working.

I thought populating the shopping cart page would be the biggest challenge, but because I wrote out what each cart item would look like before I started my JS file, populating wasn't too big of a problem. I quickly figured out I just had to append my entire cart item HTML in bits and pieces at a time.

Finally, my JS was incredibly long at the end. My pages were freezing very frequently and were running very slowly. Therefore, I broke everything up into individual functions and ran only the appropriate functions on each page.

Non-creative commons image links:

<https://www.americastestkitchen.com/recipes/220-quick-cinnamon-buns-with-buttermilk-icing>

<http://thetoastedpinenut.com/pecan-cinnamon-buns/>

<https://www.chowhound.com/recipes/iced-cinnamon-rolls-31724>

<http://newagebakingcompany.com/product/caramel-walnut-cinnamon-roll/>

<https://avirtualvegan.com/one-minute-cinnamon-roll-in-a-mug/>

<http://www.cheftographer.com/vegan-gluten-free-cinnamon-rolls-with-chocolate-gourmesso-glaze/>

*All code that I based my JS on is credited within the JS/HTML/CSS files.