

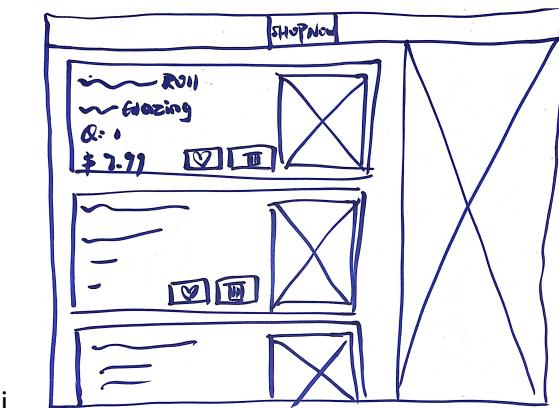
Assignment 6

Xiaofei Zhou

Andrew ID: xiaofei2

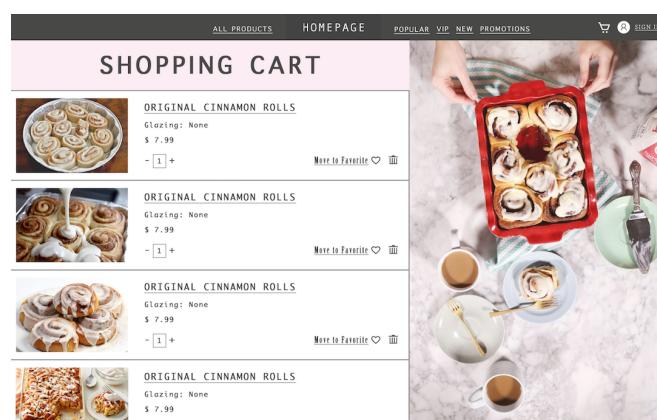
1. Low & High-Fidelity Prototypes

- Low-Fidelity prototypes:



- A hand-drawn low-fidelity prototype of a shopping cart page. The header says 'SHOPNow'. The main content area shows a grid of items, with one item highlighted in blue. The item details are: 'ROLL', 'Glazing', 'Q: 0', '\$ 7.99'. There are three small icons below the price. To the right of the grid, there are two large 'X' marks indicating sections that have been removed or are placeholders.
- The shopping cart page is basically split into two parts and the two information flows are clearly organized. One is for items added in the shopping cart and the other one is for other information display, including promotions, VIP and new comings. And the information for the items added in the shopping cart is concise and minimized. Besides, the header is consistent with all the other headers in the rest pages.

- High-Fidelity prototype



- A high-fidelity prototype of a shopping cart page. The header includes 'ALL PRODUCTS', 'HOMEPAGE', 'POPULAR', 'VIP', 'NEW', 'PROMOTIONS', a search icon, and a sign-in button. The main section is titled 'SHOPPING CART' and lists four items: 'ORIGINAL CINNAMON ROLLS' with 'Glazing: None', '\$ 7.99', quantity '1', and a 'Move to Favorite' button. To the right of the list is a photograph of a person holding a red tray with cinnamon rolls over a marble table with cups and saucers.
- High-Fidelity prototype is pretty much the same design as the low-fidelity prototype, with more design detailed in color patterns, fonts, font size, icons, images and so on. The quantity of items in the shopping cart can be changed in this page, for the convenience of customers. And customers can move the items that they don't want to buy immediately into their wish list as a reminder of shopping list. And more detailed design of

tabs is shown in the header, which is a quick navigation to those most important web pages for this website, including browsing all the products, most popular products, products for VIP, new comings, products promotions.

2. Web Prototype w/JavaScript:

a. Function Description

- i. Add items into shopping cart directly from browsing page.

The screenshot shows the homepage of the BUN BUN BAKE website. At the top, there's a dark header bar with the brand name "BUN BUN BAKE" on the left, a "SHOP NOW" button, and a sign-in link on the right. Below the header, there are two product cards. The first card features an image of three cinnamon rolls on a plate, labeled "ORIGINAL CINNAMON ROLLS" with a price of "\$4.99". The second card features an image of cinnamon rolls with blackberries, labeled "BLACKBERRY CINNAMON ROLLS" with a price of "\$7.99". Each card has a "love" icon and a "cart" icon at the bottom.

In the browsing page you can add one cinnamon roll for a certain flavor with the most popular glazing for that flavor into your shopping cart, by clicking the shopping cart button. Then number of items in your shopping cart is shown in the right side of the header.

- ii. Add rolls with different flavor and different glazing into the shopping cart in the product details page. Different product images and prices will show after selecting different glazing and quantities for a certain flavor.

The screenshot shows the product details page for "GLUTEN-FREE CINNAMON ROLLS". The page includes a large image of the cinnamon rolls in a pan next to a bowl of glazing. The product title "GLUTEN-FREE CINNAMON ROLLS" is displayed prominently. Below the title, there are sections for "GLAZING" (with options like "NONE", "SUGAR-MILK", "VANILLA-MILK", and "DOUBLE-CHOCOLATE"), "QUANTITY" (set to 1), and a price of "\$27.99". At the bottom, there are "WISHLIST" and "ADD TO CART" buttons.

BUN BUN BAKE

SHOP NOW

POPULAT VIP NEW PROMOTIONS WISHLIST

15 SIGN IN

GLUTEN-FREE CINNAMON ROLLS

GLAZING: [NONE](#) [SUGAR-MILK](#) [VANILLA-MILK](#) **DOUBLE-CHOCOLATE**

QUANTITY: [1](#) [2](#) [6](#) **12**

\$335.88

[WISHLIST](#) [ADD TO CART](#)

- iii. Click and go into the shopping cart, all the items that have been added into the shopping cart are in the list, including the information of flavor, glazing, quantity, total price and corresponding product image.

BUN BUN BAKE

SHOP NOW

POPULAT VIP NEW PROMOTIONS WISHLIST

15 SIGN IN

SHOPPING CART

Product	Glazing	Quantity	Total Price
ORIGINAL CINNAMON ROLLS	SUGAR-MILK	1	\$4.99
ORIGINAL CINNAMON ROLLS	SUGAR-MILK	1	\$4.99
ORIGINAL CINNAMON ROLLS	SUGAR-MILK	1	\$4.99

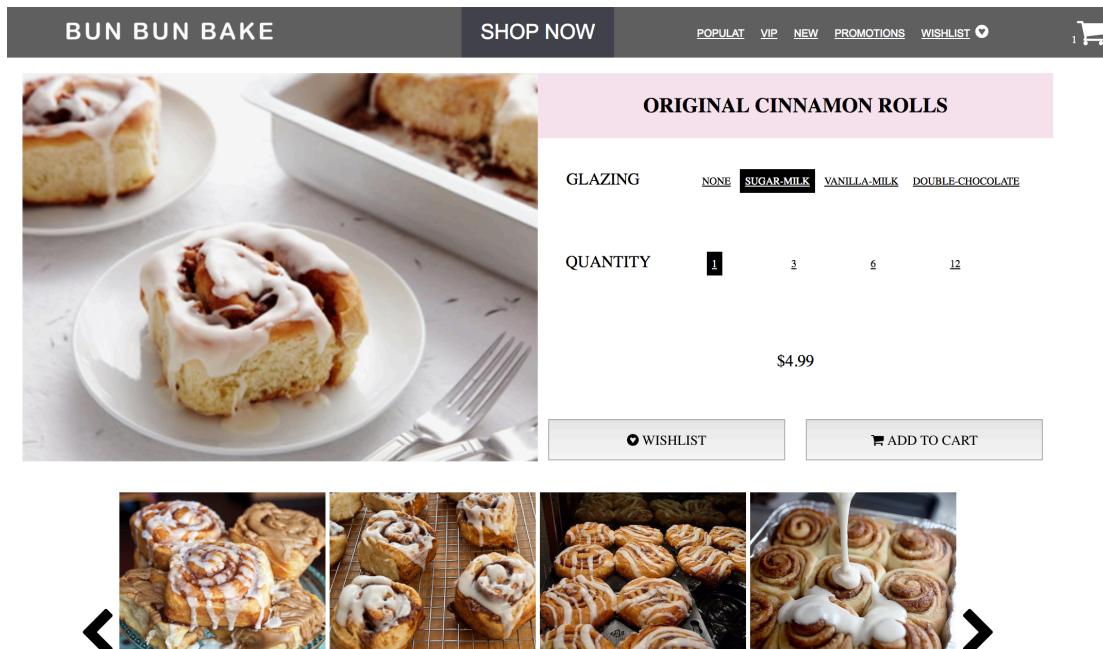
- iv. Delete the item from the shopping cart by clicking the delete button, and the number in the header will change together.

b. Bonus Part

- Wish list.
 - You can add a product into your wish list in the products browsing page, product details page and also the shopping cart page.
 - By clicking the button "MOVE TO WISHLIST", the corresponding item will be moved from the shopping cart into the wish list for the future

reference or reminder.

- The wish list page can be accessed from the header of every web page.
- ii. Carousel for showing similar products.
- A carousel is implemented for the original cinnamon rolls to show more images for customers to browse more similar product.



3. Reflection

- a. What challenges or bugs did you encounter?

I have met so many challenges and bugs. And the major ones are:

- How to know which product the button clicked belongs to?
- How to change the different paths of images shown for different glazing of a certain flavor cinnamon rolls?
- How to provide visual feedback for customers after each step of operation?
- How to maintain a data structure to store the list of items in the shopping cart or the list of items in the wish list.
- How to delete any of the items in the list for shopping cart or wish list, without any pre-defined order?
- How to generate a dynamic version for shopping cart, based on the changing of the item list? And those changes can happen in the browsing page, the product detail page, the shopping cart page and so on.

- b. How did you overcome these challenges?

- i. How to know which product the button clicked belongs to?

- I am not very familiar with CSS selectors before this assignment. And now I know how to trace back to any information you want to know from the website. Actually, I found that with appropriate use of selector, we can obtain every piece of information shown on the web page.
- ii. How to change the different paths of images shown for different glazing of a certain flavor cinnamon rolls?
- I modified the rule to name product images to make it easy to use different variables to discompose the path of the current image shown and also form the path to get the new image to be shown in the web page.

```
var image_path = $("#roll-image")[0].src;
var image_folder = image_path.substring(0, image_path.lastIndexOf("/));
```

```
var glazing = $("#glazing").children(".active").text();
$("#glazing").children(".option").click(function() {
    glazing = $(this).text();
    image_path = image_folder + "/" + glazing + ".png";
    $("#roll-image").attr("src", image_path);
})
```

- iii. How to provide visual feedback for customers after each step of operation?
- I identified which operations from customers need to get visual feedback, including deleting items, adding items, selecting from options and so on. And I use color changing, picture changing, alert popping out as the way to give visual feedback.
- iv. How to maintain a data structure to store the list of items in the shopping cart or the list of items in the wish list.
- At first I only thought about the object to store each individual item added. But I realized a list is needed for dynamic information display for the shopping cart page after each time of refreshing the web page. So I used the data structure array to store the object created every time. For example:

```
var item_list = JSON.parse(localStorage.getItem("item_list"));
if (item_list == null) {
    item_list = new Array();
}
```

```
var cart_num = parseInt($("#cart-quantity").text());
var cart_update = cart_num + quantity;
$("#cart-quantity").text(cart_update);
localStorage.setItem("cart_num", JSON.stringify(cart_update));
var flavor = $("#flavor").text();
var item = new Item(flavor, glazing, quantity, single, total, image_path);
item_list.push(item);
localStorage.setItem("item_list", JSON.stringify(item_list));
```

- v. How to delete any of the items in the list for shopping cart or wish list, without any pre-defined order?

- The information in the shopping cart page just messed up after I finished the first version of codes to make the delete button functional. Then I used the method of “`console.log()`” for debugging and tried different ways to access a certain object in the array. After hours and hours of working hard, I finally solved it. And I found that some of the errors I worked on for hours was just because of I typed the wrong name of variables.
- vi. How to generate a dynamic version for shopping cart, based on the changing of the item list? And those changes can happen in the browsing page, the product detail page, the shopping cart page and so on.
- I searched online for potential solutions to solve this dynamic HTML generation. I used the final solution I figured out for my shopping cart page and wish list page.

OTHER

I think that `console.log()` is really a good way for debugging in web programming.