

Assignment 5

Due Date	@Oct 11, 2020
Student	ⓑ Bill Guo

Links:

- GitHub Repo: https://github.com/BilLogic/boyuang.pui.github.io/tree/master/homework_5
- Index Home Page: https://billogic.github.io/boyuang.pui.github.io/homework_5/index.html

Reflection:

Changes after Heuristic Evaluation

- Problem1: Lack of visibility on user's status in the navigation bar. Though I have made the navigation bar consistently floating at the top across all pages, there was a lack of indication on which page the user is currently at.
(Violates recognition rather than recall)
 - Fix: Indicate which page they are currently on by darkening the button background color.
- Problem2: The original color theme was quite error prone. I didn't take red & green deficiency in consideration.
(Violates Error Prevention)
 - Fix: Change the color theme to better delivery brand identity with a simple and minimalism style.
- Problem 3: it's unclear what features clickable. Users might need to recall what they've clicked and how they land on certain pages.
(Violates recognition rather than recall)
 - Fix: Indicate what's clickable with rotation animation when the mouse hover about them.
- Problem 4: Users cannot cancel their selection once they swiped the green process bar. (Violates user control and freedom)
 - Fix: Added more features such as cancel button to help users recovery from potential slips.

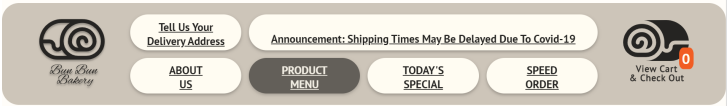


Fig-1: User Status indicated by darkened navigation button



Fig 3.1: Previous error prone color theme



Fig 3.2: New color theme

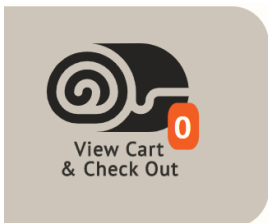


Fig2.1: Example of clickable icon

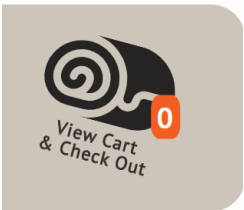


Fig2.2: Example of clickable with mouse hover above

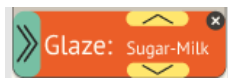


Fig 4.1: Cancel options on the right corner for users to recover from errors

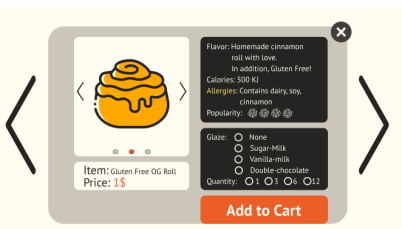


Fig 4.2: Cancel option on the right corner to get back to the main menu

What challenges or bugs did I encounter, how did I overcome them?

1. Alignment and Justify Issue in CSS

- The biggest issue I encountered was with the alignment and justification system in CSS. I was struggling with distinguishing the differences between style properties such as justify-self, justify-content, justify-items.

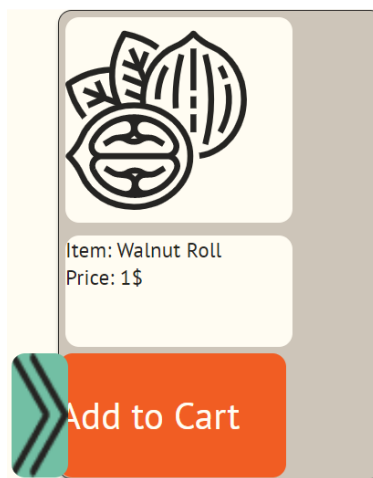


Fig 5: An example of the weird alignment I encountered.

- Lots of the alignments and gridding regulations we used in Figma turned out to be a much more complicated implementation and design process in CSS. I had to have an accurate DOM model in mind to organize and make reference to the elements.

```
display: grid;
grid-template-areas:
  'roll_1 roll_2 roll_3'
  'roll_4 roll_5 roll_6';
grid-auto-columns: 1fr 1fr 1fr;
grid-auto-rows: 1fr 1fr;
```

Fig 6: Grid system I used on the menu page

- I overcome the alignment issues with CSS's built-in grid and flex system. They helped a lot in making the grid adjust to different screen sizes automatically. For instance, I used grid-template-areas and assign inner items to a specific grid-area. The grid-template-rows and grid-template-columns commands help me maintaining the portion between different parts of the grids. As I move on, I explored with the repeat command and assigning elements to areas by grid starting lines and ending lines. (This grid line system was initially quite confusing as I was more used to zero-indexed systems.)
- Another challenge I ran into is with overriding and nesting alignment systems. In my design, some elements consistently float at a fixed position. Thanks to my TAs, I figured out how to implement nested grid areas with command display: grid, and how to fix an element to a place with command position: relative. It's quite interesting that if I manually set an outer area's position to be relative, I could limit the scope of its inner items' coordinate system and their absolute position. Throughout this process, the google developer inspecting tool also helped me a lot.

2. HTML reference within the CSS file, HTML and CSS modularity issues

- The second challenge I had was with HTML elements reference. When I started writing my CSS file, I wrote detailed decoration to almost every HTML element. This made the workflow very slow and made my stylesheet non-reusable for a different HTML page. It breaks us the modality, and I found myself going over all HTML files to update a single change in style. Even worse, I did poorly at writing comments. In hindsight, I should have started with most general style settings such as font-family, font-color, background, border-radius, elements alignment, and justifications, etc. It would make my CSS styling process much more efficient. My code would also be more reusable and easier to understand.
- I wasn't worried about this modularity issue until I started experimenting with JavaScript interactive features. When I tried to implement some universal event to my navigation bar, I need to copy and paste certain pieces of code to all my HTML and CSS files. I didn't find an excellent way to overcome this besides exploiting the "id" attribute. In some other languages such as React, I've heard that elements could work as components and be directly imported and exported. For future works, I might try to learn and program in JSX with React.

3. Version Control

- The third challenge I had was with GitHub as its learning curve was relatively high to me. I overcame it by going over a few practice online and a fully online course on LinkedIn Learning. Now I appreciate the ability to do version controls. It allows me to explore and play around with my code more freely.

How is the brand identity of my client reflected through my design choices? (What kind of look and feel did my design for them and why?)

- The brand identity I imagined for my client is to be welcoming, simple, and considerate. "Just like a good neighbor."
- To make the website more welcoming, I set the color theme to have a warm and cozy tone. I also paid attention to the border-radius, icons, and font-family selection. There is consistency in font and interactive icons that helps make the website clean and straightforward.
- In terms of the considerate characteristic, I've decided to implement some additional user-centered webpages such as "Today's Special" and "Speed Order," which can be found directly from the navigation bar. The reason behind this is to enhance users' flexibility and efficiency during their shopping process. First-time users could browse through the menu page, long-time users could find short-cuts to checkout from the speed order page, and all users could quickly get access to discounted items by one click no matter which page they are currently on. The add to cart button under all items from the menu would also help boost up user's buying process.