Aaron Bishop
PUI Fall 2019

Assignment 5 – Web HTML and CSS Prototypes

User Interface Bugs and Fixes

By conducting a heuristic evaluation on the 7-page InVision prototype I had previously built, I discovered 10 user interface bugs. Of these bugs, 8 were fixable with HTML and CSS (the 2 remaining bugs must be fixed with JavaScript). The bugs and their fixes are categorized by heuristic and detailed below in **Table 1**.

Table 1 includes the 10 noted user interface bugs and their design corrections.

Heuristic	Bug	Correction	Language
Visibility of system status	User unable to know if cart is empty or not	Update cart icon in navigation bar to display number of items in cart	JavaScript
Consistency and standards	Inconsistent use of font	Use a limited number of fonts	CSS
	Inconsistent button coloration across pages	Apply one color for each type of button across all pages	CSS
Error prevention	Users mistaking Shop button for MyPlace button	Provide consistent mapping between buttons (i.e., MyPlace button always beneath Add-to-cart)	CSS
Recognition rather than recall	Price of item displayed on the Product Browse page but not the Product Details page	Include price of item in Product Details page	HTML
Flexibility and efficiency of use	Users unable to directly access Product Details page from Landing page	Provide link to Product Details page from the Landing page	HTML
	Users could not add an item to his/her cart directly from the Product Browse page	Include an Add-to-cart button on the Product Browse page	HTML
Aesthetic and minimalist design	Branding elements clutter multiple pages	Remove the branding headers from the Product Browse page and the Product Details page and remove unnecessary advertisement	HTML
	Too many elements clutter the product details page	Remove the quantity button and move this to the checkout page	HTML
Help users recognize, diagnose, and recover from errors	Users do not receive feedback on his/her selection of a product configuration option in the Product Details page	Highlight chosen option(s) on the Product Details page after it is selected	JavaScript

Challenges Overcome

This was the first website that I have created from scratch. It was a rewarding learning experience. It was also frustrating, especially at the beginning. Three main obstacles encountered were determining the right number of layers of HTML div elements to nest, assigning classes and IDs to HTML elements to produce concise CSS code, and structuring page layouts with CSS grid.

Aaron Bishop
PUI Fall 2019

While writing the HTML, I was unsure when to nest multiple elements within a single div parent and when to distribute elements across several parents. My confusion was compounded by uncertainty over how the CSS styling of a parent element would impact its children. As an example, I retained (but commented out) my initial coding of the Product Details page in the HTML file to show how I attempted a flat hierarchy of div elements (i.e., minimal nesting), wherein almost every item in the main body of the webpage was housed in a unique div. Of course, this did not work. To fix the problem, I researched some common template codes on w3schools.com to get a better idea of how to structure HTML code. This, combined with trial and error, allowed me to create more effective code structure.

Likewise, I struggled to correctly tag my HTML code and link to CSS selectors. While building out the HTML code, I gave nearly every element a unique class. I was operating under the mistaken principle that this specificity would be rewarded later. However, as I began connecting my myriad classes to CSS selectors, I realized I had been too specific. Having created a unique class for nearly every element, I couldn't use commonly held classes to style collections of elements. Instead, I had to create a separate CSS selector for the majority of HTML elements. The code piled up. My breakthrough arrived after applying a familiar approach: reviewing HTML and CSS code templates on w3schools.com to see the level of specificity that others had used to name and organize HTML elements. This code review reminded me that CSS selectors could be specified by following a class name with an tag type (.e.g, .main img {}). That affordance made many of my classes obsolete. I rewrote and significantly cleaned up my HTML and CSS code.

Lastly, I had a brief but torturous affair with the CSS grid. Having learned about this feature through Codecademy, I assumed it would be the quickest route to implementing page layouts. In HTML, I established a main container to house all page elements and configured that container as a grid. After some trial and error, I managed to roughly fit each child element in its proper position, but I could not figure out how to style the grandchildren. Unfortunately, I still have not figured this out, despite some online research. During PUI lab on Monday (9/30) I discussed this problem with a classmate, and she recommend that I ditch the grid and opt for *floating* elements within the main container. She also pointed out that w3schools.com had some guidance on creating product cards, which were a key element I was attempting to use the grid to create. I followed her suggestions, returning yet again to W3 schools to review the code and leverage a few ideas.

Branding

The Fluff Stuff corporation for whom I built this prototype are minimalists. I sought to reflect that character in the website by delivering a soft color palette and minimal text. There is a fair deal of negative space and nothing to distract a visitor to the site from the company's best asset: the comfortable pillows. Additionally, Fluff Stuff stands for quality above all else, so I prominently featured real customer reviews on the Landing Page of the site.

Perhaps resulting from a brief conciliation to my own artistic temperament, I also embedded a subversive cultural statement on the Landing Page. The background of the Instagram section displays an image that monotonously repeates itself to fill the space devoted to it. In this way, it mirrors the content generated by Instagram's userbase.

Aaron Bishop
PUI Fall 2019

Sources cited

- w3schools.com – used to look at code template for HTML and CSS
- fontawesome.com – used to source icons for the navigation bar of the website
- hookagency.com – used to source recommendations on color pallets
- Flickr creative commons – used to source the unlicensed pictures of pillows featured in the website