

Andrea Zhu  
PUI Section A  
11/7/2021

## Homework 6: Fluff Stuff

[This reflection contains both 6A and 8B]

Link to the Main Page:

[http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Fluff Stuff.html](http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Fluff%20Stuff.html)

Link to the Product Detail Page:

[http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Product Detail.html](http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Product%20Detail.html)

Link to Shopping Cart Page:

[http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Shopping Cart.html](http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/blob/5522ad30f6cda120de5d8410bc92f7a04460bb03/hw6b/Shopping%20Cart.html)

Link to the source code (git repo):

<http://htmlpreview.github.io/?https://github.com/Andreajuz/pui-test-app/commit/5522ad30f6cda120de5d8410bc92f7a04460bb03>

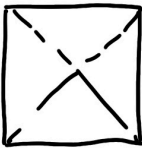
>>>

Low-Fidelity Prototype:

☰ Fluff Stuff

➔ Back ☐ Cart

Shopping Cart



Antisan Romet Rthow  
Color: \_\_\_\_\_  
Material: \_\_\_\_\_  
Silver  
Gold

Price	Quantity	Subtotal	
\$	<input type="text"/>	\$	X

Shipping

Free shipping

Total

\$69.00

Proceed to Checkout

## High-Fidelity Prototype:

≡ Fluff Stuff

↶ Back  My Cart

### Shopping Cart



#### Artisan Round Pillow

**Color** Morning Haze

**Material** Duck Down

**Size** One Size

[Edit](#)

Price **\$69.00**

Quantity

Subtotal **\$69.00** ✕

Shipping Free Shipping

Total **\$69.00**

Proceed to Checkout

>>>

## Updates from 6A:

1. Sketched low-fidelity prototype for shopping cart page
2. Designed high-fidelity prototype for shopping cart page
3. Created shopping cart page with HTML + CSS
4. Fixed the header. Every element on the header will have fixed position, same standard, and will not move along with the scroll.
5. Linked the button on the bottom of the main page to product detail page.
6. Added citations
7. Changed color selection with JS
8. Enabled insert material selection with JS
9. Created quantity change function with JS
10. Created functions to change product display pictures
11. Linked the “add to cart” button to shopping cart
12. Linked all the “back” buttons to previous pages
13. Added red dots with numbers on the cart icon of all the pages, so that after clicking “add to cart”, all the pages have indication of number of items in the cart

## Updates from 6B:

1. Adjusted visual details of product detail page (to emphasize selections)
2. Fixed bug in item number in header-shopping cart icon so every page is consistent
3. Adjusted the initial status of the shopping cart to 0 items (remove all table rows from last iteration)
4. Enabled quantity change on shopping cart page so users could change the number of items
5. Enabled the number of subtotal price to reflect the amount of items times price per item
6. Enabled "X" to remove item from shopping cart
7. Enabled storage of all selected info (color, material, quantity) and show in shopping cart
8. Enabled the total number above checkout button to reflect the sum of all subtotal prices
9. Linked "Fluff Stuff" logo on all pages to the home page
10. Added a sidebar as an additional JS function (**Bonus Point**)

>>>

## Reflection

After working on assignment 6 for the past month, I have enough reasons to think this assignment is much more challenging than the previous ones. Not only did come up workable solution became harder, but also debugging became more challenging(as the code gets longer and more complicated). One of the unexpected bug I encountered over this weekend was with event handler. There was no response when I change the quantity number of the item in my shopping cart. Nor did removing item show any response. The subtotal as well as the total number had no changes as well. It was very frustrating because after I checked multiple times, every function should be able to run but the web gave absolute zero response. The next day, with that question in mind, I reached out to my friend who has some coding background. After I explained and demonstrated my code, he checked my code and used print to find out the sequence of function's outcomes. Then we found the cause of this bug. The reason was that I need to reload the page first to add HTML elements such as changing rows and cells to reflect user's selection, and then add event handler to have browser actions. After adjusting that as well as making some other modifications, the webpage could run the way as I intend it to be. Although debugging in general required patience and attention to details, this specific bug was very hard to diagnose. I learned to use print as a method to see what each step could come up with. And it turned out to be a good way to find such bugs. Also something on the side I learned, although not totally related, was not to hesitate to reach out for help. TA, classmates in PUI, as well as other friends are all good go-to sources. They might have different methods to help with debugging. Also they could provide suggestions to improve code quality in general.

>>>

## **Programming Concepts**

### **Concept 1: Local storage**

I learned how to store and access data through local storage property. When users make selections on the product detail page (color, material, quantity), the input information (id: key + value) is stored to local storage. Then when users enter shopping cart page, the properties of the items are loaded and shown correctly in the table.

### **Concept 2: Add event handler**

I used event handlers to handle and to verify user input, user actions as well as browser actions to react to those events. Examples in my code would be button to increase or to decrease item number in shopping cart page and “X” to remove item in shopping cart page.

### **Concept 3: Modify table**

I learned how to update contents in a table or the table itself dynamically using JavaScript. An example would be when users add a new item into the shopping cart, a row will be inserted to the table in shopping cart page. Cells with information will also be inserted into that row to reflect the item.

### **Concept 4: window.onload**

I used window.onload to update information when refreshing the webpage, and to set up a sequence of functions. It allows me to indicate what actions the page is going to take. For instance, when changing the item quantity in my shopping cart, not only the product number in the header on that page, but the product number in headers of other pages should also respond to that change.

### **Concept 5: change element style**

I learned how to use JavaScript to change element styles. No matter it's adding or removing frames when mouse is hovering on top of it, or it's changing the large image being displayed (to reflect the small image when hovering over), these element styles and actions could all be done by using JavaScript.