Bennett Huffman
Homework 6B
11/8/2020

Link
https://homework-6b.vercel.app/

GitHub Repository
https://github.com/bhuffy/homework_6b

Reflection
What challenges or bugs did you encounter? How did you overcome these challenges?
Challenges for this project for me were mostly adding JavaScript functionality across pages for the shopping cart. I haven't used React Context before and decided to try using it instead of localStorage for this project. I found a couple YouTube videos that were helpful in understanding how to set up and use contexts, since the documentation isn't entirely clear how to use setters with it.

Additionally, I had trouble updating the cart quantity due to a lack of understanding about the asynchronous aspect of updating state, which caused a delay in updating the cart details. I was able to figure this out with some advice to look into moving code that doesn't directly update a particular variable into the useEffect, which solved my issue with real-time updates.

Programming Concepts
1. DRY

While this code isn't 100% DRY, I was able to abstract most of it into a few functions. Without copying and pasting code. There are some places where I use duplicates for state management, but the HTML/CSS is abstracted such that most components are defined once and reused. This includes functions that I would pass down into the children as shown below (onRemove).

```
<CartItems>
    <h2>Cart Items ({cart.length})</h2>
    {
        cart.map((item, index) =>
            <CartItem
                key={item.id}
                id={item.id}
                src={item.src}
                alt={item.alt}
                name={item.name}
                price={item.price}
                size={item.size}
                color={item.color}
                quantity={item.quantity}
                onRemove={removeFromCart}
            />
        )
    }
</CartItems>
```

2. Componentization

Shifting to react, I tried to make components that are encapsulated such that they are both reusable and hide what's not necessary. This abstracts a lot of the logic and makes entire pages look like the screenshot below, for the "Cats" product page.

```
<Header />
<CategoryHeader title="cats" />
<main className="category grid">
    <Breadcrumb>Home / Cats</Breadcrumb>
    <CategorySidebar />
    <CategoryProducts>
    {
        products.map((product) =>
            <CategoryProduct
                key={product.id}
                src={product.src}
                alt={product.alt}
                name={product.name}
                price={product.price}
                slug={product.slug}
                colors={product.colors}
            />
        )
    }
    </CategoryProducts>
</main>
<Footer />
```

3. Using arrays and function expressions to print options

It's much easier to print items from arrays rather than repeat code (similar to DRY). Aside from using components, you can also make function expressions from variables inside the components such as printing color options as shown below.

```
// list color options
const colorOptions = ITEM.colors.map((c) =>
    <button key={c} onClick={() => setColor(c)} className={`color-box color-box--${c}`}>
        {(color == c) && <Checkmark />}
    </button>
);
```

4. Using JavaScript's native functions like filter, reduce, etc. to write short and understandable pieces of code, like aggregating numbers from an array of objects.

```
useEffect(() => {
    totalPrice = cart.reduce((acc, curr) => acc + curr.price * curr.quantity, 0);
}, [cart])
```

5. Async functionality with useState and useEffect

I learned how to handle updates with the useEffect hook, since useState may update asynchronously. This was apparent when updating the quantity of the cart items.

```
// decrease quantity input change
const handleDecreaseQuantity = () => {
    if (itemQuantity - 1 > 0) {
        setItemQuantity(prevState => prevState - 1);
    }
}

// update cart item quantity
useEffect(() => {
    let item = cart.find(x => x.id === id);
    item.quantity = itemQuantity;
    setCart([...cart.filter((i) => i.id != id), item]);

}, [itemQuantity])
```