Team Name: Code Ninjas

Github User ID of submitted assignment: oliviercm

Members:

Name	**SFSU ID**	**Github User ID**
Olivier Chan Sion Moy	913202698	oliviercm
Jasneil Nat	921273467	sikorskyhawk
Jinal Shah	913318697	Jinal-Shah-Design
Jingxing Luo	921266941	JingxingLuo

Description of work:

Homepage Products (Jinal):
I worked on displaying five product pages dynamically. The data was populated using the JSON file. The HTML tag template was utilized to create and display all 16 products. Products under $20 were displayed by creating a filtered array and then a random product was displayed. Popular products display a random product and highest rated products display the first high rated product. Lastly, on sale products were filtered by discounted price and then a random product would be displayed. I had a difficult time with syntax and chaining things together. I'm a JavaScript beginner so coding this took me over 12 hours. However, at the end I felt the process was very rewarding and I learned a lot!

Checkout Page Styling (Jinal):
Additional styling was added to the checkout page such as icons, proper labeling and other css.

Product Page (Jingxing/Olivier):
The Product page used to have 16 individual html pages. The product page now could display 16 products with loading different data stored in JSON file. In addition, the "Other Product" can dynamically display 4 other products behind and ahead of the current product. The "Add to Cart" button was updated with a feature so it could store the data in the sessionStorage.

Cart Page (Olivier/Jasneil):
The cart page was updated to dynamically display the products in the cart.
Functionality was added to the various controls on the cart page including the checkbox (selected/not selected for checkout), delete from cart, change quantity, uncheck all products, automatic calculation and display of subtotal.
The "proceed to checkout" button was updated to be disabled when checking out 0 items.

Checkout Page (Jingxing/Olivier):
The checkout page was updated with some corrections. As a result, some fields in the input form could only be typed in numbers.
The subtotal, tax, and total were updated to be dynamically calculated and displayed.

Login, Register, User Preferences (Olivier):
Custom validation and custom error displays were added to the forms on these pages. The styling on input focus and input invalid states was also updated. Additionally, the appearance of the input field was unified between Chrome and Firefox (previously, Firefox was overriding the input field's styling).

Problems Encountered and Solutions:

- Everyone in the group has different levels of comfort with JavaScript, which made it difficult for those who were not familiar.
- Some members also had difficulties with creating a server and understanding how to constantly test through console.log
- Since the content was dynamically displayed with JavaScript, the products could not be seen when viewing the HTML using the file protocol due to a CORS error, making it inconvenient to debug quickly. To solve this issue, we used Node.js + http-server to quickly host servers on our development machines instead of having to use the virtual system.
- Input fields were not being styled the same way on Firefox, causing an inconsistency. Research and experimentation eventually led to a solution which looks similar on both Chrome and Firefox.