

Team Name: Byteime

Team Member Names: Jay Gupta (Student ID: 921804023), Kishan Patel (922243969), Gabriel Gidey (922049034)

GitHub UserID (for Submitted Assignment): jaygupt

Class: CSC 0317-04

## Group Project #4

- 1) What was Done: We developed our website further to include JavaScript. JavaScript allowed us to further simplify the development process; using the DOM, loops, and other aspects of the language, we were able to create most of the site using much less HTML than before, and it has become more dynamic and interactive. Using localStorage, users' cart selections, checkout values, and more are stored indefinitely, even if the tab is deleted. We also used JavaScript's validation features to make input more refined, especially on the login/register page. The JavaScript files are stored in a folder entitled "js" in our project directory.
- 2) Who worked on What:
  - a) Gabriel worked on the design of the add to cart button, as well as its transition to an increment/decrement button to control quantity. He also worked on adding products to the website's JSON object, ensuring the about page was responsive, and assisting the rest of the team in finding solutions to JavaScript problems.
  - b) Kishan worked on the login/registration page, ensuring that it was only one page instead of two (before, it was login.html and register.html, now it is just login.html), adding the sliding animation when the user switches between them, and the validation for the forms. He also added admin credentials that will allow us to transition to a sign in feature in the next iteration.
  - c) Jay worked on the add to cart feature, such that when "Add to Cart" is clicked, the product, as well as it's quantity, is displayed when the cart page is shown. He also worked on having each row of the categories page be draggable. He also worked on having the value on the cart automatically update, such as the subtotal, the discount, tax, delivery, and total. He also worked on a coupon feature, which is fully functional and doesn't allow for any input after the coupon is correctly inputted. These values are stored and displayed on the review page of the checkout process with the help of local storage.
- 3) How Work was Divided: Gabriel was very new to JavaScript, and as such, we gave him simple tasks, and things that were less JavaScript inclined, like the design of our "Add to Cart" buttons, and ideas on how we should animate our login/registration pages. Kishan had a good amount of prior experience with web development, and had created a few designs in terms of login/registration, and as such, we gave him those features to work on. He had some idea about input validation, and as such, he was the person who incorporated it into our forms. Jay had an interest in local storage; therefore, we gave him tasks that involved pushing and pulling information from local storage, such as the cart information. He wanted to have some experience

Team Name: ByteMe

Team Member Names: Jay Gupta (Student ID: 921804023), Kishan Patel (922243969), Gabriel Gidey (922049034)

GitHub UserID (for Submitted Assignment): jaygupt

Class: CSC 0317-04

## Group Project #4

in the DOM, and as such, we assigned him tasks such as updating the cart prices, that heavily on DOM manipulation.

### 4) Problems we Encountered & How we Solved Them:

- a) One issue we encountered was during the creation of the “sliding” animation on each individual product page (where the user can select a mini-photo, which then becomes the main photo): although we were using appendChild on the image element in a loop running 4 times, only the last photo was being appended; the first three weren’t appearing.
  - i) The solution to this was to use the cloneNode method on the image element; since each image was the same (only rotated a certain amount of degrees in the CSS), we needed to use this method in order to append it to each of the 4 places it was needed.
- b) Another issue faced was deciding where addEventListeners would be relative to each other in the code: for instance, the “Add to Cart” button, when clicked anywhere on the site, will be replaced by a -/+ button to decrement and increment the quantity one wants of that product. Initially, the decrement and increment buttons had their addEventListener functions inside the addToCartButton’s addEventListener function; however, that was not working as intended, as we weren’t able to click the - or + buttons.
  - i) After researching more about the DOM, it was apparent that it was only through the initial event (the initial click of the “Add to Cart”) that anything inside the function was going to be triggered. It was not the case, like we initially assumed, that the addEventListeners inside it would listen indefinitely, even after the event became over. The solution, therefore, was to put the -/+ button event listeners on the same indentation level as the “Add to Cart” button; using this solution, the buttons worked as intended.
- c) One issue we were facing was with the coupon feature: whenever someone would enter the correct coupon, we didn’t want to allow any more input, and we wanted to have the discount section in the summary of charges be green. We were using localStorage, and once the correct coupon was entered, we set the “receivesDiscount” variable in localStorage to true. However, even though the variable did get set to true, input was still being taken in, and the discount was not in green.
  - i) The solution was to use location.reload() after a user clicks on the “Apply Coupon” button: this is advantageous over the solution to do everything in the click event for the “Apply Coupon” button, as you

Team Name: Byteme

Team Member Names: Jay Gupta (Student ID: 921804023), Kishan Patel (922243969), Gabriel Gidey (922049034)

GitHub UserID (for Submitted Assignment): jaygupt

Class: CSC 0317-04

## Group Project #4

would need to recalculate the summary of charges (this would be a repetition of code). Instead, with `location.reload()`, everything is done when the page initially loads, including checking if the “receivesDiscount” variable is set to true. If it is, the summary of charges, input being disabled, and other events can be appropriately configured.