Chris Camano ID#921642160

Nathaniel Miller ID #922024360

Zachary Colbert Student ID #921899547

Arielle Riray SFSU ID# 917861209

Group Name: Normal Animal Merchants

CSC 317 Project JS version write up.

For this version of the e-commerce website project we decided to allocate the majority of our efforts towards developing and implementing a backend for our products using JSON, form validation, and cookie system.

JSON backend

Our product's data was systematically organized within the JSON by having each animal name declared as the key for the animal's respective information. The key for the required data is stored in a cookie, and the data is retrieved from the JSON file when needed by parsing the key from the cookie and locating the corresponding record. The data pulled from the JSON file is used to populate a single dynamically defined product page. Implementing this system reduced the overall size of the project as we were able to delete all of the redundant product pages. This process was very difficult to implement and required a lot of trial and error with consle.log statements to print in the console of the browser to track the cookies information. Overall this design change was an excellent trial in early database concepts and familiarized our team with the process of creating dynamically defined web pages. We now have the infrastructure in place to make the transition to a proper back-end and database much easier to implement.

Shopping cart system

This version of our website also implemented a cookie system that tracks the user’s item selection and cart contents. When a user clicks the Add to Cart button for a given product, that product’s key is passed to and addItemToCart function that stores the key in a cookie. When the checkout pages are loaded, the keys can be retrieved from the cookie and used to load the appropriate products from the JSON file. This allowed us to dynamically populate the product details without needing to duplicate the data.

Shop Filters

This was a small upgrade to make our shop page easier to navigate if the user wants to search for a more specific product. The filters are divided into four sections consisting of ‘all’, ‘warm-blooded’, ‘cold-blooded’, and ‘toxic’. How this feature works is by adding extra keywords into the class div names of the corresponding products. Then with the help of a “show/hide” element in CSS, a Javascript function enables or disables this CSS element according to which button was clicked by the user. In addition, the filter is set to make all elements visible on default.

Form validation

The final upgrade added to our website in the JavaScript version was the addition of proper form validation in the membership sign up screen, as well as the login page. The page now ensures that inputted information to the form follows the appropriate conventions for each entry and makes a text box red if there is a violation. This process was created using a series of javascript functions that are related to the submit button on these two forms. These functions consist of if and if-else statements, with a few comparing the inputs to regex variables which are found on w3schools.com. These regex variables help a lot in ensuring the inputs are in proper order or meeting the character requirements. Also, a counter variable is included to check if all membership fields pass its conditions before the user can navigate to the confirmation page.

Major Takeaways

JavaScript enabled our website to become more dynamic and require less space. In the realm of computer science these are innately positive attributes and it is clear how useful javascript can be for creating an environment that is customizable and capable of offering a modern user experience. Javascript has a very syntactically similar structure to other programming languages that we have used in the past and it was nice being able to use traditional coding methodologies such as iterative loops to create desired behavior. The separation of data and visualization also made it easy for us to add or alter functionality since it greatly reduced coupling between certain components and eliminated a lot of redundant files.