Ethan Hanlon

|  |  |
| --- | --- |
| GitHub: OccultSlolem  Group: The Beer Men  Teammates:  Pablo Martinez  Jake Swenson  Eric Thompson | ID: 921256970  CSC317 Intro to Web Software Development  922270801  918356067  922361944 |

Group Project 3 – Storefront in JS

**Description**

The assignment was to build upon the storefront we built with html and CSS using JavaScript.

Functionality needed to be added to the searchbar, the filters, the cart, the checkout page, and the footer of the website. Only JS could be used to implement these features.

**Approach / What we did**

We chose to create a website named RoboStore to sell robotics equipment, including microcontrollers, tools, sensors, and motors. We also added a settings page where a user could change various account and profile settings.

We divided the work by having each team member be responsible for their own pages. During the CSS stage of the project, the color scheme was already decided upon, so the only consistent CSS element between pages had to be the navbar and the fonts. Apart from that, team members were free to use CSS as they pleased.

For the JS phase of the project, different functionalities were assigned to each group member. These functionalities included the searchbar, filters, login, cart, and checkout pages.

Authentication was probably the simplest JS aspect to implement. When the user presses sign in or register, whatever is in the username box is recorded to localstorage. The username is read in the settings page, and when a user presses to sign out the token is cleared.

Searching was carried out by storing all the items for sale in an array. When enter is pressed, if the index of any item in the list is greater than zero the user is redirected to that page. An improvement to be made in future versions would be to list all similar objects in a search page.

The filter was implemented with a Javascript Object – the key is the name of the product, and the value is an array containing the rating, price, manufacturer, and preview image source. When the filter button is pressed, it iterates through all entries, and if the rating, price, or manufacturer matches, the items are written to localstorage and the user is redirected to filter.html. On this page, a for loop runs that adds the product card HTML for all the items onto the page.

For the cart, local storage was used to store the items that the user selects, and then they were displayed in the cart, and then the checkout page. Three .js files were created for the products, the cart, and the checkout page. In the cart, the price updates as the user changes the quantities, and items can be removed from it.

**Issues and Resolutions**

During the JS phase of the project, there were many more difficulties than the other phases.

One of the issues we faced was adding functionality to the navbar. Changing searchbar behavior on every single page was a major nuisance, so I wrote it out into a separate script that adds the HTML into the “navbar-container” div tag. I wrote similar behavior that adds the footer to the “footer-container” div.

The filtering was an especially complex project, although it worked surprisingly quickly for a 168 SLOC segment. The hardest part was displaying the results correctly. For some reason, I could not get JavaScript to write the td tag into the HTML, so I instead applied the requisite CSS styles to the a tag contained within and playing with the margin properties until it looked the same as other pages.

Just getting the cart to work took hours, and due to lack of experience of JS caused problems. Getting the total price in the cart to change when the products were modified was a challenge to get working correctly. Also, when adding a product to the cart, working with local storage was a challenge, as the data had to be stored in a complicated way, and inserting and extracting information was tough. It was a challenge to display the products correctly.

**Screenshots**

*Home page*

*Graphical user interface, website

Description automatically generated*

*Graphical user interface, website

Description automatically generated*

*Product Page*

*Graphical user interface, website

Description automatically generated*

*Graphical user interface, website

Description automatically generated*

*Category Page*

*Graphical user interface, application, website

Description automatically generated*

*Settings Page*

*Graphical user interface

Description automatically generated*

*Login page*

*Graphical user interface, website

Description automatically generated*

*Cart page (Zoomed out to 67%)*

*Graphical user interface, application

Description automatically generated*

*Checkout page (Zoomed to 67%)*

*Graphical user interface

Description automatically generated*

***Graphical user interface, website

Description automatically generated***

*About us*

*Graphical user interface, text, website

Description automatically generated*

*FAQGraphical user interface, text

Description automatically generated*

*Filter (Manufacturer: Raspberry Pi)*

*Graphical user interface, website

Description automatically generated*