Web Development – Mr. Turner

Project – Product Page

Project Overview

Companies have different ways of displaying their individual products once selected from the catalogue. Make a complete product page that is eye catching and easy to negotiate.

The Page

Your product page will be dynamic, meaning that it can support any product in your catalogue. ***You are not designing a catalogue, just the product page.***

The elements of your page must include:

- Your company banner.
- The title of the product.
- An image of the product.
- A description of the product.
- The product's price.
- A purchase area.
- A review section.
- A related products section.

Your Company Banner

Create a company for yourself and design a banner that goes along the top of the page. Your company should have a logo font (try 1001 free fonts). Remember to create your logo using the font in photoshop or paint and save it as an image. Your font will not show up on someone else's machine unless you force a download (which is not recommended).

Title, image, description, price

Place the title, image, description, and price of the item on your page. Try to arrange them so that they are eye catching and easy to read. Don't overwhelm your user with crazy backgrounds. Remember that the spotlight is on the product. Use CSS.

The Purchase Area

This is where the user will commit to buying the product. If your page scrolls, this area should appear somewhere near the top so that the user doesn't have to scroll to find it. Never force your user to have to work to buy your product.

Your purchase area must include an "Add To Cart" Button. When the user clicks this button, the item (and all relevant information related to the item) will be stored in the cart. You will keep track of the cart in ***Local Storage*** (described later in this document and found at https://www.w3schools.com/html/html5_webstorage.asp).

Before a user adds the product to his or her cart, there are options. You must include a way for the user to choose a quantity. This ***may not*** be something into which the user types. The quantity will be selected from a series of choices.

Your product should contain at least one other type of choice, also selected from a series of choices. For example, if your product is a piece of clothing, the user should have to choose a size and color. If the product is a game, the user may have to choose a console.

Note that the Purchase Area uses form controls but is not a form. The information is saved to local storage, but does not load a new page.

The last control in the purchase area is the *Purchase* button. For our purposes, this button will just empty the user's cart.

The Review Section

Give your user an opportunity to give your product a rating and write a text review. This section should have a list of previous reviews (make them up) and a space at the bottom for a new review.

Your product rating should be out of 5. It can be numbers or stars or porcupines or ice cream cones. Whatever makes sense for your company. Create a control for the user to select the rating number.

Create a text area so that your user can write an in-depth product review.

Create a submit button.

You do not need to save reviews. When a user creates a new review, update your page so that it shows the new one (and any existing ones from that session). Upon reloading the page, it will reset to the default made-up reviews that you included.

Enhancements

The Related Products Section

Create a javascript data file with a series of related products (at least 4, but no more than 7). When your page initially loads, it will pick one of these products and populate your page with its information (as detailed above). At the bottom of the page, list the remainder of your products, including a title, a picture, and the price. The user may click on any one of these, which will load your product page and populate it with the information for the selected product.

You can put all of this into a form, which will load a new version of the same page, but pass in the information, or use the location object's assign function.

Either way, if you are bringing in data from the query string in the URL, you must take into account the fact that that information will not be there during the first loading of the page. In initialize(), you will need to check and see whether or not there is a query string and proceed from there.

Local Storage (very similar to sessionStorage)

A ***cookie*** is a way for a web page to store information about itself (and you) on your computer. Cookies are frequently used so that users don't have to consistently log in to websites over and over again. Usually, readable information is not stored in cookies. Cookies usually have codes which will be used to access your information on a secure database on the website's server.

Local Storage has created an alternative to cookies. It gives a web page the ability to store information on the user's machine without any server interaction at all. The access is severely restricted to a tiny space on your hard drive that is reserved for web information. Programmers cannot access your root directory or any of your private folders with local storage.

Local storage belongs to the ***window*** (as localStorage) so you can omit the object when using it.

localStorage is an object and is manipulated through the creation of properties.

```
localStorage.property = someValue;
```

These properties can be written and changed on the fly as you, the programmer, see fit. If you want to eliminate a property completely, use the function:

```
localStorage.removeItem(property);          //property is passed as a string
```

Remember that all of the values in localStorage are strings. That means that you cannot store objects (although you can break up an object into a string using JSON) and any numbers must be converted back using parseInt, parseFloat, or the Number object.