Professional Site Reflection Paper

by: Bryce DeAlessio  
for: CS 110 Intro to Internet Programming and Computer Applications

Background

I am currently a Draftsman at a major aerospace company in Connecticut. I am responsible for creating 3d models and drawings for some mechanical parts that require a lot of data and information to properly define them. The large amount of data I have to manage in my work led me to teach myself the Python programming language for which there is a rich ecosystem of open source data analysis tools available. I have developed a number of tools in Python that need a user interface. While i could use some of the desktop based User Interface kits for building User Interfaces, having local tools running on multiple desktop machines is difficult to manage. Creating web-based applications is becoming more common place within our enterprise due to centralized management aspects of web-apps. Therefore as an elective in my current degree program i decided to take CS 110 to help me hone my web development skills.

Python vs. JavaScript

So far I can say my experience with JavaScript has been somewhat frustrating. There seem to be so many more gotchas with JavaScript than Python. One of my biggest frustrations was that of iteration through an array. In Python when you want to iterate through the values of an array (Python calls it a list) you do the following:

flavors = ["strawberry", "vanilla", "chocolate"]

for flavor in flavors:

print(flavors)

In JavaScript it's:

var flavors = ["strawberry", "vanilla", "chocolate"]

for (flavor\_index in flavors) {

console.log(flavors[flavor\_index])

}

It almost seems pointless to have the "for in" in JavaScript since the temporary variable given during the iteration is the value's index of which you then have to use against the original array to get the corresponding value.

Professional Site Intro

My Wife is dedicated to being a stay at home mom while our two boys are still young. She is also an avid knitter and fiber artist. While her primary focus is giving our two boys the best head-start they can get in life she dreams one day of having her own business based on knitting and other fiber arts. When I told her the main assignment for this class was building a professional web-site she pretty much decided for me what i was going to be building... a future business/portfolio site for her!

The basic features we agreed upon were:

* Home Page
* Blog page
* Portfolio page
* Store page

Out of the many aspects of this project the most challenging aspect to me was not so much the coding but coming up with the content and getting a visual design I was happy with.

Home Page & overall style

The Home Page was pretty straight forward. It's just an intro to the site and an overview of its features. Most important to her was that the site share the same header and navigation look over all the pages. I lucked out in the Logo Department because about a year ago My wife had an artist friend of hers design the logo and title image for her business cards in exchange for what else? A hand knit sweater of course! So I will state for the record that I did not create the title image or the "Knitty Girl" logo they were created by Janet Galasso of [Janet Galasso Design](http://www.janetgalasso.com/) [3].

For styling the page I wanted to use colors that complemented the logo, fonts that conveyed a handmade and crafty feel and a layout that was simple yet responsive and interactive.

To pick the colors I found a really nice site called [coolors.co](https://coolors.co/). It's a web-app that helps generate a color palate to reference as you design a website. It also allows you to upload a photo and it will create a palate for you based on the colors in the photo. I used the logo picture in the top right of the site to help me pick out a palate I liked.

For the site headers h1, h2, etc. I used a google font [1] called "Crafty Girls". For the main text I used a font called "Muli". "Crafty girls" definitely had a handmade feel to it and "Muli" struck a good balance between "hand-made" and readable. The google fonts are included in the site as external links in the "<head>" tag and can then be referenced in your CSS sheet as a font-family just like any of the built-in browser fonts. I also changed the default font line spacing to 1.5 times and upped the font size itself to 14pts.

Most of the layout of the site is done using the css3 ["Flexible Box Model"](https://www.w3schools.com/css/css3_flexbox.asp) [2] which I figured out how to use doing the personal site and expanded upon for this one. The most notable use of the flex box model is in the Store and Portfolio page items for displaying the set of photos as a main photo above a row of smaller photo's. Also the navigation bar uses the Flexible Box Model which allows the items to automatically stretch to the parent container. When done properly The Flexible Box Model allows page elements to scale in size as well as gracefully overflow from one line to the next depending on the browser window size.

In order to make sure the many different pages of the site all shared the same code for the Header and Navigation bar I wrote a python script that would merge the content area of each page with the header and navigation area of each page. This made it so I would not have to keep copying and pasting changes I made in those areas to every page.

Portfolio & Store

The portfolio and store pages have a similar layout listing the different knitting projects my Wife wanted to display or sell. The items in the Portfolio were ones she wouldn't part with and the items in the store are ones she could. The other difference with the store is a pop-up modal checkout form. A modal is similar to a pop-up except it creates an overlay on an existing page instead of opening up a separate browser window. I did this as most browsers block standard pop-up dialogs.

I have a JavaScript object named “items\_db” that is loaded on the store page that mimics what a database might send to the page on loading. A JavaScript event is attached to the purchase button and upon clicking it the entry in the JavaScript object associated with that sale item is loaded into the item summary table in the order-form. The necessary <td> cells have an <input> field that is disabled so that the data shows up in the html encoded form data but can’t be modified by the user..

Blog

The Blog is pretty straight forward but my wife and I really only had enough time to write one blog post so I didn't incorporate some of the fancy features yet that were on my personal site. As I examined what it takes to truly organize and categorize a blog site I decided it was best to keep this one simple. To do a blog properly one needs to be able to track and categorize your posts & keep the tree listing in the sidebar up to date. This is possible with a static site but doing it all by hand would be very tedious. If I were doing this site for real I would probably use a database backed web framework like [django](https://www.djangoproject.com/)[4] which can generate pages dynamically or a static site generator like [Pelican](http://docs.getpelican.com/en/latest/index.html)[5] which scans a directory structure of static pages and builds the blog post and index entries in one shot.

Conclusion

This Project was a test of my planning and design capabilities as much as it was my coding. I had a lot of fun with it however, especially seeing my wife's excitement as her website took shape.

At work I had already started doing a little bit of web development but productivity is king and I was using a lot of helper libraries like [Jquery](https://jquery.com/) [6] for DOM Manipulation, [Bootstrap](http://getbootstrap.com/)[7] for styling and layout and Python web-frameworks for server-side actions. Having to do this project without those tools really makes me appreciate having them but also helps me understand what they are doing behind the scenes.

Citations

[1] **“Google Fonts,” *Google Fonts*. [Online]. Available: https://fonts.google.com/. [Accessed: 05-Aug-2017].**

[2] **J. Galasso, “108 Intentions Project,” *Janet Galasso Design*. [Online]. Available: http://www.janetgalasso.com/. [Accessed: 06-Aug-2017].**

[3] **“CSS3 Flexible Box,” *CSS3 Flexible Box*. [Online]. Available: https://www.w3schools.com/css/css3\_flexbox.asp. [Accessed: 06-Aug-2017].**

[4] **Mark Otto, Jacob Thornton, and Bootstrap contributors, “Bootstrap · The world's most popular mobile-first and responsive front-end framework.,” *Bootstrap · The world's most popular mobile-first and responsive front-end framework.*[Online]. Available: http://getbootstrap.com/. [Accessed: 06-Aug-2017].**

[5] **jQ. F.- jquery.org, “jQuery,” *jQuery*. [Online]. Available: https://jquery.com/. [Accessed: 06-Aug-2017].**

[6] **“Django,” *The Web framework for perfectionists with deadlines | Django*. [Online]. Available: https://www.djangoproject.com/. [Accessed: 06-Aug-2017].**

[7] **“Pelican 3.7.1¶,” *Pelican 3.7.1 — Pelican 3.7.1 documentation*. [Online]. Available: http://docs.getpelican.com/en/stable/. [Accessed: 06-Aug-2017].**