**CSE 134B Homework 3**

**Building Out Clickable Templates in HTML & CSS**

By Sherman Cheung and Trent Miller

**Analysis**

Vanilla (Pure CSS)

**Productivity** – Overall, we spent more time using the vanilla method to implement our design. We estimate that we spent about 20 hours in total to implement the CSS for our site. We first focused on the desktop implementation for the site before moving on to mobile. This split focus increased the development time overall.

However, because we had full control and did not use any frameworks for this method, we were able to more precisely implement the design we envisioned, and did not have to deal with the constraint of working around another developer’s template. We believe that this will result in a superior user experience because we were able to fully customize the implementation to match our specific design and use case.

Moreover, this method resulted in much clearer and cleaner HTML code as it was more loosely coupled with our HTML than the Bootstrap implementation. As a result, the HTML is more readable and has more appropriate and logical class names. This followed the philosophy that HTML is structural that was laid out in Homework 2.

On the other hand, we needed to write much more CSS code for the vanilla implementation. Because we did not borrow CSS that came from a framework, we wrote it all ourselves and this resulted in much more lines of style code.

**Speed** – The results of our speed analysis are shown below. For our testing we used a Android phone and throttled the network speed to “Regular 3G”. When we first were testing speeds, we got pretty bad results due to our images not being compressed. After compressing all of our images the load times significantly decreased.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Page | DomContent | Transferred | Load | Finish |
| session\_tracker.html | 241 ms | 1.1 KB | .531 s | .523 s |
| login.html | 297 ms | 134 KB | 1.83 s | 1.83 s |
| register.html | 387 ms | 134 KB | 1.93 s | 1.93 s |
| index.html | 324 ms | 258 KB | 2.92 s | 3.53 s |
| heropicker.html | 382 ms | 335 KB | 4.43 s | 4.43 s |

Non-Vanilla (Bootstrap)

**Productivity** – The development time for this method was shorter overall. We estimate that we spent around 12 hours developing the site using Bootstrap. Because of the many existing templates for Bootstrap, we were able to implement certain parts of our site almost immediately. Furthermore, Bootstrap’s mobile-first philosophy and grid system allowed for side-by-side implementations of both the desktop and mobile versions of the site. By taking advantage of the grid system, we did not need to tinker with media queries like we did with the vanilla implementation.

However, we ran into a few downsides with Bootstrap as well. Namely, neither of us had used this framework before so we spent a lot of time early on just ramping up. This added to our development overhead. Furthermore, there were no templates for some of the more complicated aspects of our design and we had to spend a considerable amount of time tinkering with Bootstrap’s existing tools to fit our specific needs. For example, our design included a cardview for the main screen but Bootstrap 3 does not have a component for a cardview. Consequently, we had to implement our card view using Bootstrap’s existing tools. This felt extremely sloppy and “hacky”. This is one of the shortcomings of utilizing frameworks. They try to act as a “one size fits all” solution, but this is far too idealistic in reality.

In terms of actual code, utilizing bootstrap closely coupled our HTML and our stylesheets. While we tried to make our HTML mostly structural in Homework 2, Bootstrap uses class attributes to specify placements using a grid system. For example, an element could be placed using the class “col-md-2” meaning it takes up 2 columns/gridlines of space. This is a purely presentational concept which conflicts with the logical philosophy we had for HTML in Homework 2.

On the other hand, utilizing Bootstrap’s tools allowed for a much shorter CSS file as we did not have to write everything from scratch like in the vanilla approach. After ramping up, we were able to develop faster by using Bootstrap’s abstractions for layout. We did not need to think in terms of raw CSS, but instead used Bootstrap’s grid system.

**Speed** - The load times using Bootstrap were significantly worse. Bootstrap significantly increased both the time it took to create the DOM Tree, and the overall bytecount on the wire.

The “div-itus” nature of Bootstrap significantly affects the time it takes to create the DOM Tree. By introducing so many div tags into the existing html code base Bootstrap creates a lot of unnecessary new elements in the DOM Tree and results in more code that needs to be parsed. When dealing with a device with limited computing power, the effect of the bad DOM Tree load time becomes even worse. A Good example of this was with our index.html page. Index.html DOM creation time with Bootstrap was 9X worse. This page relied heavily on Bootstrap card divs.

Another area where Bootstrap hurt us was the total number of bytes sent on the wire. Bootstrap introduces some overhead because we have to include all of their CSS and JS files.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Page | DomContent | Transferred | Load | Finish |
| session\_tracker.html | 486 ms | 20.4 KB | .870 s | .918 s |
| login.html | 1.65 s | 187 KB | 2.43 s | 2.43 s |
| register.html | 1.69 s | 186 KB | 2.60 s | 2.62 s |
| index.html | 3.03 s | 301 KB | 3.96 s | 3.98 s |
| heropicker.html | 428 ms | 354 KB | 4.57 s | 4.66 s |

**Conclusion**

Overall, in comparison to the vanilla method, Bootstrap allows for much faster development time because it allows for developing for desktop and mobile platforms simultaneously. Seasoned developers who are already experienced with the framework would not need to spend time ramping up like we did and could get started right away. However, these tools come at a cost. Loading Bootstrap into the user’s web page increases load times by quite a bit compared to the vanilla method. Bootstrap also coupled the HTML and CSS much more tightly due to how it uses classes and divs in its grid system.

The vanilla method on the other hand offer much more customization for specific use cases and much faster load times as no additional libraries need to be loaded. However, using vanilla HTML and CSS to develop for multiple platforms proves to be a more challenging task than using Bootstrap’s grid system.

We would recommend using a framework like Bootstrap when the developer needs to get implement a generic design quickly for multiple platforms at the cost of speed and flexibility. When a developer has a very specific vision of their site design and cares a great deal about load times, they may opt for the vanilla approach.