Tony Ly
Mark Santa Presca
Alvin Yan

Homework 3: Specifications

Developer Analysis for Vanilla versus Framework CSS

Using a framework CSS such as bootstrap versus implementing vanilla CSS for the webpage has both benefits as well as tradeoffs in both the user end and developer side. In terms of work hours for the developer, we found that using a framework yielded less developing time in for specifying and distributing the styling to all other pages on site. This in total took 2.5 hours while implementing vanilla CSS to create the style for the table and distributing the same styling other pages took 3 hours. Another metric we can compare the two implementation strategies is lines of code, where using the bootstrap framework outclasses vanilla CSS, as all it takes is 1 line of code to link a bootstrap framework while implementing vanilla CSS took 4 files that added to 84 lines of code. While using bootstrap is certainly a benefit in the implementation time, using a framework comes at the cost of customizability. It is much more difficult to change a particular style of a borrowed CSS framework than it is to vanilla CSS, since the former requires the developer to understand another developer's work to make sure making chances doesn't cause a cascade of unintended changes while working within vanilla CSS gives more developer the freedom to make changes because it is starting from scratch.

Impact on performance and size

The effect on using a framework or vanilla CSS is has more apparent consequences on the user side more so than the developer side. The amount of time it takes to load a webpage created using vanilla CSS has the advantage of loading only the assets the developer used to make that webpage. The overall size of the bootstrap CSS framework was in total 22.9KB while the vanilla CSS 2.6KB, approximately 10 times the size amount of the vanilla. The execution time margin between vanilla CSS and framework CSS is significant as well. Under regular 3G load (100ms latency, 750kb/s download and 250kb/s upload) with no caching optimization, the longest time it took for the vanilla CSS version to download a 40KB image was 9670ms, while the framework CSS took 9890ms at worst to download a 40KB image. With good 3G speed (40ms latency, 1.5Mb/s download and 750kb/s upload), vanilla CSS took 4590ms and framework CSS took 4810ms to download a 40KB image at worst. The vanilla CSS takes 1276 ms to establish a connection with firebase in regular 3G speeds, and in good 3G speeds, the firebase connection took 75.52 ms to establish. The framework CSS took 123.12 ms to establish a firebase connection in regular 3G as well and 93.27 ms to load a firebase connection with good 3G speeds. To load all requests, the vanilla CSS took 10.6 seconds under regular 3G and 634 ms with caching, while the framework CSS took 11.7 seconds and 1.05 seconds when we factor in caching. In good 3G, these numbers improve the vanilla CSS loading times to 4.84 seconds without caching and 598 ms with caching and for the framework CSS, total load time

reduced to 5.5 seconds without caching and 879 ms with caching optimization. The performance differences results demonstrate that while the vanilla CSS took longer to develop from a programmer's standpoint, the user's perspective will experience a faster loading time and a much more portable website that loads well on desktop and mobile devices with average network speeds. Framework CSS as shown from the results is load slower in all situations in comparison.

Analysis of methodologies

Using framework CSS as opposed to vanilla CSS comes with both benefits and tradeoffs and as such there is no clear best-fits-all method. Using a framework such as bootstrap comes with the most obvious benefit of less work for the developer. A framework comes with all the standard designs that work for a lot of general purpose websites, and these frameworks have been tested and optimized to catch common bugs and prevent developers from having to go through all the work of going through common pitfalls such as poor structure and presentation separation. On the other hand, because frameworks are made by a third party, it is often much more difficult to design a customizable, personable prototype without the developer from either having to edit most of the framework code or spend a long time learning the framework inside and out when a developer might just need to make one small customized unit. Because of their popularity, frameworks are often used for most websites and as such, can lead to a sense of genericness for a website. This comes with a benefit that this breeds familiarity for most users on the web which can improve retention rate for how to navigate through a website. Vanilla CSS on the other hand come with the more difficult developer work in terms of time spent. This requires the developer to work from scratch and make the designs all up front which takes a significant amount of time. This comes with the added bonus of letting the developer choose what they specifically want for every part of their website, leading to a much more customized look. Lines of code is another measure between the two, but the lines of code impact developers and the end user differently. While the written lines of code for vanilla CSS is much longer than using a framework, the amount of bytes to load to a website is significantly more for a framework, because using that framework means importing all the code that comes with it, even code that the developer did not use. With vanilla CSS, the code that is sent to the user is the code that the website will use, and that amount is often much lower than a framework. This comes with the impact of performance which is shown in the above data where the framework loads slower in all situations and all network speeds in comparison to the vanilla CSS.

The applications of vanilla CSS and framework CSS comes from the purposes of the developer. If the developer is trying to make a website that is trying to be more familiar to users so that they can learn the flow of the website, then using a popular framework such as bootstrap might be the best choice for the developer. Using the entirety or the majority of a framework is also a good sign to use the framework as opposed to using many frameworks but only using a small portion of each one. This would lead to sending less unused code to the user in comparison. Vanilla CSS is a legitimate approach if the user is stylizing a lightweight website that isn't going to make use of the fancier style applications. For a dex like web application such as the

amiibodex, the performance factor may be a worth the extra use of time develop the vanilla CSS, since the necessity of stylization of an application like this is relatively bare. In situations where using an application that's purely functional like a dex that doesn't need much stylization, vanilla CSS would be a valid approach considering function trumps form and function cannot work without performance. Each methodology has it's merits and consequences, but in general vanilla CSS is much more lightweight and compact if the stylization is compact, but a framework is much quicker to implement and come with a lot of functionality that can be used without having to go through the making all the functionality.