Dennis Cao
Deryl Lam
Jeffrey Trang

As the instructions pointed out, we created CSS styling and Bootstrap styling and measured our productivity by keeping track of how long it took us to create a page, file size, and how long it takes to render on an Android device using 3G network.

In regards to the time it took to code the pages in vanilla CSS vs. Bootstrap, it was not even close. Vanilla CSS took our team a collective 15 hours to finish compared to only 3-4 hours for bootstrap. This discrepancy in time has to do in large part with most of the planning taking place in the CSS portion of the coding. Because the entire team worked on Vanilla CSS first, by the time we got to coding the Bootstrap version, we all already knew where to place all the elements and had shaken off the rust.

Something we noticed was that Vanilla CSS gave us a lot more control of the elements in general. While it would take longer to code and involved numerous google/stack-overflow lookups, coding everything manually with CSS would in the end allow us to design our site with more degrees of freedom than if we were constrained with bootstrap. An advantage of bootstrap, however, was that the prebuilt elements all look slick and clean. Nobody on this team claims to have any artistic ability of any sort, so the ready-to-go designs that came with bootstrap provided us great relief.

In the end, we decided that it would serve our team best if we continued using bootstrap due to our collective lack of artistic talent. Much time was wasted in the Vanilla CSS element trying to figure out why the page was so damn ugly, time that would be saved if we used bootstrap from the beginning. For teams that have some sort of graphic designer, we would suggest that they try designing everything in vanilla CSS. By avoiding bootstrap, such a team would save a lot of space in terms of the size of their website, and thus save loading time for the users of their site.

Speaking of loading time, the following tables represent the loading times of our website pages under three different conditions: Desktop, Unthrottled Mobile, and Throttled Mobile. We measured the mobile data by connecting the phone to Chrome Dev Tools and throttling/unthrottling the connection. The data is below.

Vanilla

Page	Desktop: Unthrottled	Mobile: Unthrottled	Mobile: Throttled	Size
Login	191ms	190ms	131ms	1.67KB
Home	449ms	382ms	335ms	5.84KB
Remove Players From Watchlist	51ms	203ms	913ms	9.25KB
Choose Team	312ms	832ms	351ms	8.23KB

Dennis Cao
Deryl Lam
Jeffrey Trang

Players by team	273ms	435ms	625ms	5.08KB
Choose Position	52ms	117ms	241ms	2.67KB
Players by position	65ms	134ms	331ms	9.53KB
Search Results	280ms	213ms	28s	6.49KB
Player Stats	162ms	174ms	5.5s	3.37KB
Registration page	49ms	104ms	119ms	1.27KB

Bootstrap:

Page	Desktop: Unthrottled	Mobile: Unthrottled	Mobile: Throttled	Size
Login	437ms	390ms	720ms	2.33KB
Home	309ms	512ms	640ms	6.84KB
Remove Players From Watchlist	100ms	1.26s	578ms	9.25KB
Choose Team	149ms	2.49s	1.51s	10.7KB
Players by team	135ms	540ms	845ms	7.48KB
Choose Position	95ms	1.55s	497ms	4.07KB
Players by position	101ms	379ms	357ms	10.5KB
Search Results	114ms	455ms	606ms	6.56KB
Player Stats	110ms	430ms	1.65s	4.39KB

Dennis Cao
Deryl Lam
Jeffrey Trang

Registration page	107ms	284ms	10s	2.08KB

Based off our data, throttled connection for vanilla CSS usually is slower but not by too much and is unnoticeable for some of our pages. For example, our login page was took about the same amount of time to load for throttled connection and a wifi connection. All but one page loaded faster on wifi speed connection. The outlier was the Choose Team page and this is the 3rd biggest page on our website which makes sense why it took so long but there may be underlying reasons that we do not know of that's causing it to load so slow.

At first, our pages took way too long to load for the throttled connection on mobile devices with vanilla CSS. We figured out that our images were too big and we could tell by looking at the waterfall on the developer tools and the images themselves were taking at least 5 seconds to load and some of them took 10 seconds or more! To solve this issue, we resized and compressed all the images to fit our needs and most of the images were cut down by at least 50%. After optimizing our performance, there was only one page that took too long to load but the others were under 6 seconds which is acceptable. Search Results took 28 seconds to load on a throttle connection and this was alarming. We tried resizing and recompressing the images in that page hoping it would improve our time but the changes were not significant enough to improve our performance. The waterfall was saying that our html page which is about 6KB was taking about the same time as the image which is about 900KB (this was the smallest the image would go after compressing) to load. We think the developer tool might've had a hiccup which would explain this strange behavior.

When we measured performance for the bootstrap pages all the files were bigger than the vanilla css and after logging the data, the waterfall told us that the bootstrap dependencies took at least a few hundred ms to load. The throttled speeds were obviously slower but when we compared the vanilla css with bootstrap on unthrottled speeds, the vanilla css ran a lot quicker and some of the bootstrap pages took a few seconds to load while the vanilla css took at most one second to load (except that outlier).

When we compare Desktop speeds to Mobile speeds, it's obvious that Desktop was faster in every category or the difference is so small it's unnoticeable. Throttled speeds on Desktop were obviously slower than unthrottled which is why we didn't include the Desktop data. We conclude that Bootstrap is easier to use and can improve productivity when coding but at the cost of performance.