CSE 134B Homework 3 Analysis

Jason Pon
A10871437

Initial lines of code from Homework 2:

login.html: 30
signup.html: 35
forgotpassword.html: 23
home.html: 41
mydex.html: 42

Version 1: Vanilla CSS

Lines of added code:

login.html: 30 -> 30 (+0)
signup.html: 35 -> 31 (-4)
forgotpassword.html: 23 -> 28 (+5)
home.html: 41 -> 52 (+11)
mydex.html: 42 -> 52 (+10)
stylesheet.css: (+29)

Overall, 0 – 4 + 5 + 11 + 10 = 22 lines of code added.

Hours taken: 14+

User Impact (on webpagetest -> Nexus 5 -> 3G Slow)

login.html: 1.69s – 1.99s | 1KB – 6KB
signup.html: 1.82s – 2.19s | 1KB – 5KB
forgotpassword.html: 1.59s – 1.89s | 1KB – 6KB
home.html: 1.86s – 2.14s | 15KB – 20KB
mydex.html: 1.89s – 2.23s | 15KB – 20KB

Why Employ this approach?

+Less complex organizationally. Unlike framework CSS, containers and tags can be can be used to signify something, rather than using tags like div, which signify nothing. The number of lines of code for the developer can be reduced, promoting understandability and maintainability.
+Faster page speeds and lower byte sizes as discussed in analysis 3.

Version 2: Framework CSS (Bootstrap utilized)

Lines of added code:

login.html: 30 -> 37 (+7)
signup.html: 35 -> 41 (+6)
forgotpassword.html: 23 -> 36 (+13)
home.html: 41 -> 58 (+17)
mydex.html: 42 -> 56 (+14)

Overall, 7 + 6 + 13 + 17 + 14= 57 lines of code added.

Hours taken: 4+

User Impact (on webpagetest -> Nexus 5 -> 3G Slow)
login.html: 2.00s – 2.39s | 20KB – 24KB
signup.html: 2.11s – 2.50s | 20KB – 24KB
forgotpassword.html: 1.88s – 2.19s | 20KB – 24KB
home.html: 2.43s – 2.75s | 49KB – 54 KB
mydex.html: 2.40s – 2.76s | 49KB – 54KB

Why Employ this approach?

+More specific/modern features. For example, there is an email field in bootstrap css which doesn't exist in vanilla css. For vanilla css, you would have to write regular expression to check for a valid email address, which would take more time and effort to implement.
+Different way of programming. Compared to vanilla css, Bootstrap relies on classes more, adding style attributes can cause a conflict, and there is a grid system for laying out elements. These approach allows for a different way to think about creating and implementing elements on a page.

ANALYSIS 1: In terms of the lines of code written for each version, Framework CSS using Bootstrap took more than double the amount of code. By looking directly at the code however, this can be attributed to the reliance of Bootstrap on containers like div.

ANALYSIS 2: In terms of the number of time taken to implement webpages styled with similar CSS, the framework CSS used (Bootstrap) took about 1/3rd of the time to implement. This appears to be beneficial for rapidly implementing functionality, much like what is desired for in topics like prototyping. But tradeoffs in other areas like site speed, security, etc. may need to be looked at and weighed against.

ANALYSIS 3: For load times and byte counts, pages using framework CSS (bootstrap) tend to load at most upper bounded by a little over half a second more than vanilla css. This trade of may not be too bad in exchange for a more modern visual style and feature set than vanilla css. The reason for the longer load time might be due to the continual need to reference external stylesheets. On the byte count side however, framework css pages consume three to four times more bytes than vanilla css pages. This makes sense because we are importing bootstrap stylesheets (library). On the users though, this would be affecting on those who either have slow bandwidth (making pages take longer to load) or data caps, the later which can be pervasive since we currently live in a cellular / mobile-centric world.