

HW3 Write Up

Github link: <https://github.com/likeabbas/cse134b-hw3>

Discussion - No Framework:

Who implemented: Abbas and Adam
Lines of code written: 432 (not compressed. readable)
Amount of time spent: 6 hours
Android phone download time for lead page: Just under 1 second

We started with not having a CSS framework just to see what we could come up with. Not having a CSS framework definitely seemed like a disadvantage in some ways, but at the same time it really allowed us to deeply understand how HTML and CSS are intertwined. With that said, writing this version took quite a while. Although the website is very simple, and uses a lot of similar components throughout, each page had it's own quirks that needed to be taken care of individually. At the same time, we felt very accomplished at the end for having built something from scratch.

Discussion - Framework:

Who implemented: Abbas and Adam
Lines of code written: (not compressed. readable)
Amount of time spent: Under 2 hours
Android phone download time for lead page: Just less than 3 seconds

We chose to use the Materialize library for our framework. After implementing the non-framework version, picking up with the framework seemed like heaven. Having almost everything ready and all you had to do was add to your class name was GREAT after spending hours figuring out the smallest annoyances. It's almost like you didn't really need to know CSS that well to build a great looking application. This could go well for several reasons, like if you need to get something up extremely fast without having to think then a framework is definitely the right choice.

Discussion - Both (Compare, contrast)

The tradeoffs of using a framework vs not using one are pretty straight forward in terms of two metrics: time to deploy, and time to download. And, of course, they are inversely proportional to each other. Without the framework, we spent 6 hours trying to get the application to look like our mockups. But, with the framework we were able to get it up in just under 2 hours. But, that time spent in development for the non framework version would potentially allow us to access more users. With an access time of under a second for the version without a framework, it's easy to see that if you're a new business trying to impress your customers the first time they reach your site then you **need** to have a loading page without a framework. That < 1 second response on an extremely old android phone falls in line with RAIL principles.

That's not to say that having a framework doesn't have a place on a production grade web site. The CSS framework, whether it's one you downloaded or created yourself, allows you to easily integrate the look of different parts of your web application easily, and also cache similar CSS components that span multiple pages into a single CSS file. It may be worth it to have your first landing page with just the bare amount of CSS that is required to load, so that user becomes invested in your website. Then, as they begin to explore, you can include a larger CSS framework file, cache that so they can go through your website without having to load a giant CSS file every time they load a new webpage.

Aside:

Perhaps we could have done the no framework version up faster if we were more practiced, but we believe there is more going on. With a framework, most of the slight adjustments like margin or padding are already there for you and are usually close enough to what you want, and we almost never ended up modifying them from their defaults. Without the framework, almost every bit of CSS had to have some slight adjustment in the margin or padding just to make everything not look so close together. We found that we were more okay with "close enough" than without the framework. To elaborate, when coding without the framework it felt much more like our own product and it made you really want to spend the time to get the little things right.