Most importantly you will also need to provide a write up that discusses your findings with the two methods employed. You should provide numbers in terms of work effort (lines of code, hours taken, etc.) as well as user impact (load times, byte count, etc.). Your write-up should also provide a reasoned discussion of when you would want to employ each approach. Try not to declare one way the winner, instead present thoughts on how and why each approach you take would be valid for one situation or another. The write-up should be no longer than 3 pages and be succinct and to the point.

When we started this project, we assumed that using frameworks would make coding and designing our website a lot easier. Below we will list the pros and cons of each method, as well as provide some metrics as to how the work load differed.

Framework Pros	Vanilla Pros
• Using a framework in many ways is very efficient – instead of spending time styling elements from scratch and writing functions, frameworks allow us to access all these features with a few lines of code • Most frameworks are free • Responsiveness is faster and easier to implement with frameworks	• Writing CSS from scratch gives total control over how much code is used • Smaller file sizes leading to faster and more optimized load times for the app • Code is more reliable because we control all of it. This is different form using a framework because with this we are not reliant on dependencies or security flaws in a framework's code.

While coding using frameworks one thing that we did not expect was that any time we wanted to do anything that deviated from how the framework implemented something, like a UI element or a grid – we encountered a lot of problems overriding the CSS. In other places also without realizing we would have styling problems and only later realized this was because the framework already had CSS in place that was affecting elements on our page. Overriding these was quite a hassle and ultimately ended up using up a lot of the time.

What was frustrating about writing CSS from scratch was the monotony and tediousness of implementing the most basic things. Just making the site responsive is what took most of our time, and as a result we were able to spend a lot less time making the site less polished.

Below is the information on the different file sizes and numbers of lines of code:

	Lines of HTML	Lines of CSS	Project size
Framework	312	134	2.2 mb
Vanilla	161	374	1.2mb

From this comparison we can see that the project size of our code using frameworks is almost double that of the vanilla version. We did not notice any drastic differences in load times for

the user when we did our testing however we think this is because our overall project size is still relatively small. If we were to scale this up to a larger enterprise scale application, there definitely would be dramatic slow downs.

What we found interesting was how little CSS we needed in our user written files for the framework side. Using the framework took a lot of load off of us to write CSS, however it meant we had to structure our HTML in a very specific way which made us use a lot more HTML.