Brandon Huang
Chiraag Prafullchandra
Daniel Ku
David Lee
18 February 2017
CSE 134B

Homework 3 CocktailDex Analysis

In developing our website using both vanilla HTML/CSS and a version with Twitter Bootstrap, we found that using a framework increased our productivity, but decreased our overall load performance. Vanilla js provided us a blank slate on which our ideas could flourish and small file sizes as only the dependencies we need were included. The framework provided us a large library of assets that we could utilize as well as a well defined design language/aesthetic that we could piggyback off of.

When developing using vanilla html/css we found that often times we were focused much so on the implementation of features that were commonplace in other applications as well as simple if we could leverage another person's work. When developing the navigation bar on top, we struggled to create a navbar which would which condense depending on the user's device or screen width. The navbar would collapse into a dropdown when viewed on a mobile device.This is common on so many other applications that we wondered how we could do it with ourselves. We were able to get the nav bar to collapse into a hamburger using a media query, but could not get the drop down functioning with a click of the hamburger as homework two barred us from using javascript. A collapsible navbar is a standard feature in twitter bootstrap which can be implemented in html exclusively if the framework was included.

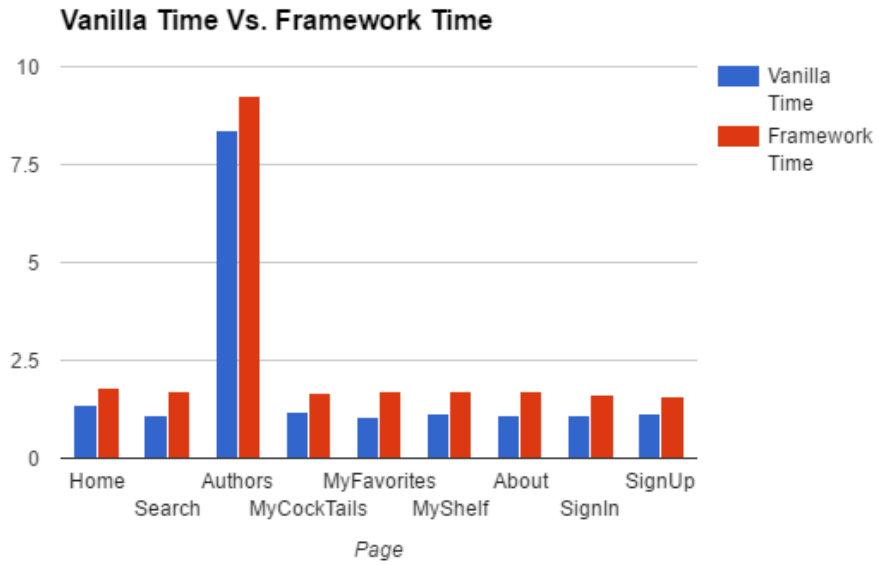
Moreover, we also found that the search results page was difficult to align all three columns using vanilla as we had to account for both desktop as well as mobile views. Plain css makes it difficult provide a responsive page. In bootstrap, less time was spent reinventing the wheel and more time was spent creating a strong user experience.

In our development, we found that the vanilla took around 26 man hours while the framework version took around 15 man hours. Although we can't say for certain due to a sample size of one team on a specific website, our developers reported that a majority of the time spent in vanilla developing features were saved when developing it again in twitter bootstrap. However, time spent not developing these features were instead spent on understanding the bootstrap framework. The bootstrap library is both complex and feature heavy. The framework

integrated neatly with the existing code, augmenting the functionality as well as providing a minor aesthetic boost to elements such as buttons and lists. However, concepts such as the grid template took a while to understand as the developers had to read into the design language that bootstrap invokes. Although it provided strong functionality in terms of responsive design, the grid system was filled with containers, rows, and columns.

One thing that we struggled with while using bootstrap was finding our own voice in the bootstrap library. By default, bootstrap overrides the look and feel of a lot of common elements in HTML. This replaces what is typically a neutral look with something more stylized and comforted to the style guide set forth by twitter. In such we found that we had to override much of the css if we, as many other developers would, want to keep the responsive design without being overpowered by the Twitter aesthetic.

Another downside to using Bootstrap is that it decreased the performance of our web application. As we had to include the brunt of the massive bootstrap library, we found that our load times as well as our page size increased by a significant amount. We found that a user on an android phone with average 3G connection would take an average of 44% longer to load the page and have to load around 55 more kilobytes of data. This may negatively impact the user's experience as longer load times may frustrate the user and larger page sizes would may cause data overages from the user's cell carrier.



In all, a developer may find it useful to use vanilla HTML/CSS if they want to keep their application simple and low drag. An example of this would be a developer trying to create a website for users who have poor network access or have outdated hardware. Users with poor

network access, such as users on mobile phones and those in disadvantaged countries. With these users, latency times plays a much larger factor in users experience However, they may be limited in the features they can develop in a timely matter. Developers who use framework would want to use frameworks if they want to build complex web applications with high functionality as well as leverage the resources of previous developers. This allows them to increase their productivity by focusing less on low value generic features and more on providing a thoughtful user experience. Developers are like any other workers, they have goals and deadlines. They can try to tailor their code as best they can to the application, but in the end they must strive to work as efficiently as possible. A framework is just a tool that can help with that.