CSE 134 Homework 3 Write Up

CSS Vanilla Approach

Lines of code: 2,004 lines total (1,648 lines of HTML; 356 lines of CSS)
Hours taken: ~ 7 hours
Load times and byte counts (not counting game images):

- Logged-out homepage: 179 ms; 2.887 KB
- Login page: 162 ms; 2.9 KB
- Register page: 252 ms; 2.473 KB
- Logged-in homepage: 138 ms; 3.1 KB
- My Played Games page: 149 ms; 3 KB
- My Planned Games page: 143 ms; 3 KB
- My Collection page: 148 ms; 3 KB
- My Wishlist page: 142 ms; 3.1 KB
- My Invitations page: 189 ms; 2.875 KB
- View Game page: 141 ms; 3.3 KB
- Game Recommendations page: 186 ms; 3.4 KB
- Create Invitation page: 175 ms; 3.1 KB
- Add / Edit Game page: 282 ms; 2.857 KB
- View Invitation page: 312 ms; 3.2 KB

When to employ this approach:

- When we want to focus on performance: the vanilla CSS approach tends to lead to faster page loads as total size tends to be smaller (you only write, and thus the user only downloads, the code you need for the specific site).
- When we want to make our site distinctive from other websites, with its own unique look and feel, the vanilla CSS approach makes customization easier and gives more design flexibility.

Advantages:

- Using one CSS file that has all our styles defined there allows us to easily change how our application looks like instead of individually editing structural/framework classes in each HTML file/template and keeping track of which ones have been changed.
- Pages tend to load faster, as the user is only downloading site-specific code.
- Easier to keep content more separate from presentation, allowing for cleaner, more maintainable, more semantic HTML markup.
- Reduces dependencies on others: no new version of a framework to upgrade to that requires massive reworking of your code.

Disadvantages:

- A huge time sink if you’re not very familiar with CSS.
- Developers have to check for compatibility across different browsers, since what works in some browser does not mean that it will work in other browsers.
  (Note: developers should still do compatibility testing even when using Bootstrap, but Bootstrap provides an already well-tested framework to build on, likely reducing compatibility engineering/debugging time.)

Bootstrap Approach

Lines of code: 1,478 lines of HTML (not including Bootstrap itself, of course)
Hours taken: ~ 2 hours
Load times and byte counts (not counting game images):

- Logged-out homepage: 923 ms; 24 KB
- Login page: 597 ms; 23.619 KB
- Register page: 599 ms; 23.578 KB
- Logged-in homepage: 834 ms; 24.2 KB
- My Played Games page: 829 ms; 24.2 KB
- My Planned Games page: 827 ms; 24.2 KB
- My Collection page: 836 ms; 24.2 KB
- My Wishlist page: 837 ms; 24.2 KB
- My Invitations page: 1.07 s; 24.1 KB
- View Game page: 840 ms; 24.4 KB
- Game Recommendations page: 1.53 s; 24 KB
- Create Invitation page: 843 ms; 24 KB
- Add / Edit Game page: 596 ms; 24 KB
- View Invitation page: 837 ms; 24.2 KB

When to employ this approach: This approach would be good for saving time by not having to write our own CSS style rules and if we don't want to spend a large amount of time in designing the look for the application. It's good for developers that are not so good at design to easily make a nice-looking, responsive web page that will work across several browsers and mobile devices. The framework-based approach is also excellent for *internal* pages, eg. admin pages that only you / staff will ever see. In this case having a distinctive design doesn't really matter, and you don't have to worry about performance across the highly heterogenous device / connection combinations used to access publicly-facing websites.

Advantages:

- Has defined classes for us to use so we are able to quickly implement it into our wireframes instead of starting from nothing. This means we can save time since we don't have to write our own style rules
- Bootstrap provides an already well-tested framework to build on, likely time spent testing/engineering/debugging for browser compatibility.
- Allows for a more uniform and consistent look across many different browsers.
- Large support group if we need help in wanting blocks of code to be styled in a certain way. There's also a lot of Bootstrap documentation which means there's not a high learning curve for someone who is new to HTML/CSS.

Disadvantages:

- Customization is more difficult; Bootstrap (and similar frameworks) impose a lot of design decisions on you by default. The website might not look as unique or professional since a lot of web developers are using Bootstrap.
- The presentation not being separated from the content may make the HTML less semantic, can become tangled and hard to maintain or keep consistent especially across files.
- Bootstrap and similar frameworks tend to be distributed as large blobs of code, so users with slow internet connections may become frustrated with loading times.