Sean Mandable

smandable@gmail.com | 203.590.0767 | https://github.com/smandable | seanmandable.me

What was one of the best projects you worked on?

DiscoverOrg's new platform. I was hired to literally replace their entire front end codebase, which consisted of a number of different stylesheets that often overrode each other multiple times and HTML that was either generated by stored procedures or scattered about multiple directories. The backend was a mix of procedural PERL, some rather-poorly implemented PHP, and a MySQL database.

So I had a number of choices. The CTO and I eventually decided on an Angular/Bootstrap-sass front end, and Java Spring Boot on the backend. He was a Java developer, so eventually hired another back end Java dev (I am much more familiar with PHP), but I got to learn Angular, improve my SASS, and learn how to interact with Java APIs.

What was the worst project you worked on, and how did you turn it around?

I guess one of the worst projects was at another company a few years ago. They used a lot of simple, generic templates on a .NET backend. I was able to get those updated, with some newer-looking designs as well as consolidate a bunch of CSS and fix some JQuery, but the biggest issue I had at that company were the multiple layers of middle management that necessitated an awful lot of meetings. Many of those were somewhat redundant, and it was often difficult to get actual work done! I didn't really turn much around there; the company was bought and taken private, and apparently went out of business a couple of years ago.

What are some of your pet projects?

I'm working on a web application to categorize and record video files that I've downloaded. Given a directory, it first normalizes all the filenames- converting them to title case, removing extraneous characters, etc. Then, it adds the titles to a database, after making certain that they don't already exist. If they do, it moves the new files to a "duplicates" directory that it creates, gives me info like length, dimensions, file size, date added to the database, etc.

I am also working on a small business venture with a friend. It's early stages, but we have a minimum viable product.

What is your favorite development stack and which tools do you prefer?

I do like the MAMP stack: my iMac, built-in Apache, PHP, and MySQL/MariaDB. I use Atom as my preferred text editor, and iTerm2 as a terminal replacement. I am using Affinity Photo for image editing. It is awesome so far, and I don't have to use the GIMP!

Describe your experience with SASS or LESS?

I started using SASS in 2012, when I was using Mobify.js to create mobile version of some existing sites. They used SASS in their application, so I needed to learn it fairly quickly once I was given the project. After that, I just started using it in different projects, and when Bootstrap-SASS came out, I moved on to using it.

What is your experience supporting retina graphics and what techniques did you use?

Well, I will kind of "eyeball" things to make sure they look good at high resolutions. I have a 5K Retina iMac here, so I generally see pretty quickly if things don't look good for whatever reason. That being said, if I use things like font-awesome I don't have to worry about scaling. I try to use higher-resolution images where practical; If I can't, or that particular page's weight is over what I am comfortable with given the audience, I've used CSS image replacement at varying breakpoints.

Describe what you like and/or dislike about MVC frameworks. Which ones have you used? Which ones would you not use again?

I like the fact that they take a lot of the set-up and grunt work out by not reinventing the wheel. They also force you to maintain an MVC pattern during

development. I am sometimes tempted to throw a bunch of files in the same directory for testing and such, but something like Angular makes that harder!
I've been using Angular since around September, 2014. Recently, I've begun transitioning to Angular 7/Angular-CLI and Typescript. I also have some experience with Knockout.
I guess I wouldn't use Knockout again. It requires the full JQuery (which makes it big, comparatively).

Do you have experience with JSON?

I do- If I am not reading or writing to a database, I generally store data in JSON. Especially when dealing with the front end. It can get a little hard to read, but there are so many formatters (and validators!) out there that it isn't usually a problem. And it certainly beats having to deal with XML!

Do you have experience with Javascript templates? If so, which ones have you used and why would you prefer one over another?

I have some experience with Liquid while supporting a LiveChat chat system, as well as Moustache. I'd prefer Moustache over, say, Handlebars, as I don't have to deal with compiling it or having its logic in templates.

What are data- attributes good for?

To store extra attributes on HTML attributes. These can then be accessed in JS or CSS.

What version/source control systems have you used and which do you prefer?

Git. Hands down. I've had to use Subversion in the past- but having files and directories locked is just crazy to me. Especially now that there is Git! I am also familiar with both Github and Gitlab.

What is the difference between responsive and adaptive design?

Adaptive design adapts to different screen sizes based on predetermined layouts for those sizes. Responsive design is far more flexible in that it is by nature fluid (and flexible), and will adapt its layout as needed to a given screen resolution. Of course, this approach generally uses (predefined) CSS breakpoints, but at least Bootstrap assumes that it is being used on a mobile device, and uses an "at least this size" perspective to change layouts.

What do you like best / dislike most about working on teams?

I like having other developers that I can ask questions! Or just ask to take a look at something. Sometimes I will stare at code for an hour and not see obvious errors that a fresh set of eyes can spot immediately.

You and another engineer completely disagree on a technical approach. How do you resolve this conflict?

I look for evidence to back up my claim. And ask them for the same. I have certainly been wrong before!

What sites or methods do you use to keep up on the latest advances in front-end development and user experience?

Well, I learn by doing. So I generally start with a Google search for whatever I am looking for or whatever problem I am having. This almost invariably leads me to Stack Overflow, where I can choose between a number of possible solutions or find more info. And by doing this, I learn more about the subject at hand.

I check sites like smashingmagazine.com, alistapart.com, csstricks.com, and others I'm probably not thinking of right now. I also get a bunch of email newsletters, from Site Point, Web Tools Weekly, the Smashing Newsletter, etc. I generally skim these, but occasionally something catches my eye.