Few notes for the test code.

I have uses jQuery, Less and Grunt as expected and no other framework or big library. Plus I added a js bundler (webpack) plus Babel to transpire ES6 code. The bundled project (NOT BUILT) is in “public” folder, just open /public/index.html to visualise the test on the browser (I worked with Chrome and I did not test any other browser). Obviously “dev” is the development folder (where the grunt “watcher” runs).

If you want to do js/LESS changes:

1. go in the main root
2. Type in the terminal: npm install
3. After installation, type: grunt (and it will start watching for js/LESS changes)
4. Open /public/index.html

---

1. Bootstrap / LESS

1.1 Q: What are your thoughts when considering a frontend framework like Bootstrap for a project?

> In general I love frameworks but I try not to abuse them. For fast-delivery projects probably bootstrap is really handy but for in-house custom projects I would completely go for a custom solution, with a custom UI library based on company needs. I would probably just inherit grid systems from such tools like bootstrap.

1.2 Implement the form as shown in the image, applying reponsive layout down to mobile phone layout using bootstrap 3.3 DONE

1.3 Q: When does it (normally) make sense to use CSS preprocessor like LESS/SASS and what are the biggest benefits you see?

> I always use CSS preprocessor and for the current frontend development situation and its complexity, it is a must. The most important benefits are variables and functions that allow you to easily edit your UI across many pages through few vars changes. And also, how nice it is to organze your project CSS-wise and component based.

1.4 Setup a simple Grunt file to compile custom LESS stylesheet(s) to CSS DONE

1.5 Include the custom CSS file into the page and extend(override) parts of the styling of the bootstrap form DONE

2. jQuery / JS

2.0 Q: How do you structure / encapsulate your JS code if your NOT building on top of a JS lib/framework like AngularJS, ReactJS etc.?

> It depends on the project but generally I try to work component-based. My first webpage analysis is to find out common components and create something which is configurable to be reused in different scenarios (I am a big fun of reusing code). I think of not one big webpage but instead, many small components bundled together).

2.1 Q: When does it make sense to use a jQuery plugin script vs. implementing a functionality yourself?

I think it is not worth to reinvent the wheel every time a new functionality needs to be implemented. So, if I find a plugin that fits my needs 100% then I just use it. But on the other hand, for custom features which do not fit so well such a plugin, I'd rather choose implementing it myself (and the final solution might become a "company" plugin for every developer to use).

2.2 Add a form field validation(=not empty check) with jQuery for certain form fields(first/ last na me, gender), use jQuery - plugins as you please DONE

2.3 Show an error message (if validation fails) using the jQuery Colorbox (http:// www.jacklmoore.com/colorbox/) plugin and visually indicate which fields are invalid DONE

2.4 Q: What are some (personal) principles y ou follow when working with JS DOM event listeners to avoid bad page performance, 'memory leaks' etc.?

First of all, I avoid global variables. Secondo of all, I try to reduce as much as I can the amount of DOM changes (based on the user interaction) as it is really heavy for the browser to process. Another optimization might be about pictures: in case of a medium/big amount of images in a certain page, I try to use lazy loading.

3. Ajax / HTTP

3.1 How would you submit the form via AJAX using jQuery? (just JS code example, no need mock any submit endpoint) DONE

3.2 Q: When does it make sense to submit a form as a non - AJAX (regular) request?

I would say when it comes a redirect to a different page after the server response. But generally I personally think that AJAX requests are always a better solution for web application as it improves a lot the user experience, obviously is much faster and the page more reactive.

3.3 Q: What relevance does it have for the frontend JS - code that the server endpoints repsond with different HTTP codes? Which HTTP codes do you consider most important?

Most importantly, via HTTPS code response, the developer can handle way better the final UI response on the browser. Let's say that you get back a 404 not found code as response and, based on that, you can create a custom UI solution for such error, that would increase a lot the user experience. The most relevant to me are: 200 (succes), 400 (client error), 500 (server error)