**Front-End Developer Challenge and Questions**

We realize there are often multiple solutions or answers to the same questions and to the challenge in particular. We do not necessarily have “right or wrong” in mind, we are simply looking to learn a bit about you, and your approach to development.

**Questions:**

What front-end component libraries or frameworks have you used besides bootstrap?

* *Reactjs, Polymer 3, Lit-html*
* *Redux*
* *Typescript*
* *Sass*
* *Web components*
* *Jest, Mocha*
* *Webdriver.io*

What tools would you implement to enforce code quality rules and standards?

*To enforce the code quality, I could use some sort of typing library like Typescript, this in the beginning can looks like more work but in the long run represent a huge gain decreasing the bug density up to 20% among other benefits.*

*It is critical to have and strong definition of “Work done”, to me a code is ready to go when it is readable, maintainable, testable, clear and supported in all the mayor web browsers; the code must to have a decent if not 100% of unit test coverture and E2E or/and regression tests.*

*Another important approach to increase code quality is the CI/CD implementation which should be backed up with an strong instrumentation of login systems such as Splunk and New Relic.*

*The PR’s review process is other key piece in the process of release high quality code while the use of feature toggles enables teams to code new features with confidence.*

*We cannot let out of this list linting and code formatting tools, they catch a great deal of common errors and bad practices at the time that enforce and standard code base formatting.*

*Finally, the Test-Driven Development (TDD) is a great work flow which according with IBM studies can decrease bug density up to 80%.*

What HTML templating languages or engines have you used?

*Mustache and Handlebars are the JS templating engines I use to work with.*

Which code related blogs or sites do you read regularly?

*I learn and try more than regularly new technologies, trends and skills reading blogs or tutorials and watching videos from different sources. Here I list some of them:*

*Online sources:*

* [*https://web.dev*](https://web.dev)
* *https://developer.mozilla.org*
* [*http://caniuse.com/*](http://caniuse.com/)
* [*https://stackoverflow.com/*](https://stackoverflow.com/)

*Courses/Tutorials:*

* *Udacity*
* *Pluralsight*
* *Udemy*
* *Google Code Labs*

*Youtube videos channels:*

* *Chrome developers*
* *ReactNext*
* *Facebook Developers*
* *Firebase*

*RSS channels:*

* *SitePen Blog*
* *Mozilla Hacks*
* *Planet Webkit*
* *Brad Frost*

In Sass, what is the significance of using an underscore at the beginning of a sass file name?

*The underscore indicate that the file is a partial sass file which will be import in a main sass file and buddle as one. This enable a modularized stylesheets architecture will all its benefits.*

Name a few (npm) node modules that you think are most useful?

* *Babel-core*
* *Webpack*
* *Typescript*
* *Tslint*
* *Moment*
* *Lodash*

Given the option to choose Angular, React or Vue.js for a new project which would you choose and why? What type of requirements might affect your decision to use one over the other?

*The right answer here is: It depends.*

*For a more robust project I would pick Angular while my best option for a faster pace coding is React.js and finally for a greatest performance from the core I will use Vue.*

*Other considerations:*

* *Framework maturity*
* *Community size*
* *Learning curve / framework complexity*
* *Typescript*
* *Cli tooling*
* *The type of web application(PWA, Heavy Graphs, Advertisement… )*

If you were referred to an unfamiliar node project repo, what file or files would you first look to, to determine the commands to run, build and test the project?

*The place to look first on a new repo is the Readme.md file, is the searched information is not listed there I will open the package.json file and checkout the scripts section in order to find all the available commands on such project.*

What is the benefit of using the HTML ‘nav’ element vs. a div with the class of .nav?

*While we can make almost any html element to look and behave like other the good practice is to find and use the existing html element for the required task. Doing that we gain the semantic meaning, readability and accessibilities features which needs to be manually done is we want for any valid reason not use the proper element.*

What code related framework, library or language do you want to learn next?

*My next must to learn item is WebAssambly. I think it getting stable enough to jump into it and is a key technology to unlock never seen web capabilities. I also plant to formalize in a whitepaper a Micro UI/Frontend solution on which I have being working on for a while now.*

**Challenge:**

Utilizing React and Bootstrap, create a single, web page that collapses responsively from desktop to mobile, where the desktop and mobile breakpoints match the wireframe images provided.

The carousel at the top of the page should contain three slides – which can all display the same content, but the carousel itself should be functional in that clicking left or right will animate to the previous or next slide.

Additionally, in the mobile view, the “hamburger” (three lines icon) menu button should collapse and expand the navigation.

All other components are only present for layout purposes and do not need to function ie. navigation links do not need to lead anywhere, search does not need to function.

Utilize React for all functionality - do not include jQuery.

Bootstrap must be imported via Sass and compiled to CSS (Do not use CDN links for styles). Take care to demonstrate the use of modern, semantic HTML elements and CSS classes, and Sass style nesting in any custom CSS.

**Challenge Setup**:

Use Create React App to create your project and generate the output to a /dist folder using the build script.

Combine all of your challenge directory/files as well as this document with completed answers in a single .zip file and deliver back to us as soon as you have completed them.

Please provide your entire directory structure (with the exception of node\_modules) to keep total delivery size small.

Feel free to contact us with any questions and thank you for taking the time to complete these items!