MediKeeper Frontend Skills Assessment –

* **(CSS) Explain what specificity is in CSS, and list the various ways to increase specificity (Try and resist the urge to use google…. At least for the first five minutes).**Specficity determines which CSS rule is applied in the browsers. Sometimes specificity is the reason that the CSS rules don't work to some elements. If there are 2 selectors that are being applied to the same element, specificity chooses the one that is more accurate. If the selectors are exactly the same it was applied multiple times, the browser usually chooses the latest rule that is defined.  **Ways to increase specificity would be the following..**

a { color: black }

.modal a { color: green } <<< specificity would choose this, it's more specific

------------------------------------------------------------------

div #spancolor span { color: green } <<< this rule would be selected

div span { color: orange }

span { color: black }

The text of the span would be green because that rule is the most specific. The orange would also overrule the black because the div span is more specific than the span by itself.

------------------------------------------------------------------

html body li.favorite { color: red;

font-size: 20px; }

.favorite { color: green !important;

font-size: 15px !important; }

We would of have a conflict if the !important wasn't there, but because !important is set for .favorite, now the browser will choose the selector with the !important due to the increased specificity.

------------------------------------------------------------------

* **(CSS) Write the necessary CSS, in normal CSS or in a preprocessed syntax (SASS or LESS) to achieve the following layout:  
    
    
  On Screen Sizes 1000px or larger**

**On screen sizes less than 1000px**

* **(JS) Tell us what your front-end framework of choice is and why it’s the best framework to use.**  
  Angular- The reason I choose Angular is because Angular lets you make your page more dynamic and responsive. It allows you to scale and build complex logic without the need to think too much about it. Data binding is another powerful feature that can be used with Angular to help reduce re-writing code. When a web application is built, it may contain numerous amount of code base, but with the use of data binding, it would make the code disappear. Directives in Angular is a great feature as well, it allows you to re-use components, and avoid re-writing code. The ViewModel in Angular also provides specific data and methods that can maintain the specific views. Dependency Injection in Angular is a great feature to use as well, it allows you to just ask for the dependency rather than having to create it yourself. There are always debates over what the best framework is in the technology industry for front end developing, but in my opinion Angular seems to be the more popular. Angular is easier to use for programmers, it is also more comprehensive, and not to forget, GOOGLE created Angular :D

* **(JS) Write the necessary event handlers to capture the value of the button in the following markup.**The ajax-load.html markup will be loaded for you via a jQuery.load() call that takes place after the DOM has been loaded. The desired output is the value of the data attribute **data-button-value** on the dynamically loaded anchor. (Just output to console).<div class="outer">

<!--- EMPTY INITIALLY -->

</div>After jQuery Load Call (you do not need to implement the load call, this is done for you)<div class="outer">

<a href="#" id="loadedButton" data-button-value="gotcha">Button</a>

</div>  
  
 ANSWER:

$("#loadedButton").click(function(){

console.log($("#loadedButton").attr("data-button-value"));

});

(ALSO TESTED IN INDEX.HTML FILE)

* **(JS) Perform the following jQuery AJAX call using jQuery promise chaining:**The following markup exists on the page:  
    
  <div class="component">

<a href="#" id="ajaxCaller" class="button" data-button-value="123"> Button</a>

</div>  
  
  
There is a generic handler called remote-data.ashx that returns a JSON object. That object contains an array of key-value pairs.   
  
  
Create a POST ajax call when the button above is clicked that calls remote-data.ashx, passing the value in the button’s data-attribute **data-button-value** in a parameter called passedValue. Process the returned JSON and output the value with the key **specialKey** to the console via a basic console.log() call. In the event of an error output **“The request has failed”** to the console. No matter what when the request is done, regardless of success, output **“The request has completed”** to the console. **ANSWER:**

$.ajax({

type: "POST",

url: "remote-data.ashx",

data: JSON.stringify({ Markers: markers }),

contentType: "application/json; charset=utf-8",

dataType: "json",

success: function(data)

alert( data.specialKey );

{console.log("The request has completed");

},

failure: function(errMsg) {

console.log("The request has failed");

}

});

* **(Optimization) Explain the benefits to using a CSS Preprocessing language like SASS or LESS when working with large scale projects.**  
  - It keeps the responsive design projects more organized

- It allows you to create variables so we can avoid re-writing code

- Less code will be written therefore you can write your CSS quicker.

- With the use of @import, it helps reduce HTTP requests

- Nesting in Saas allows you to have a cleaner method of targeting DOM elements (example: div, html, body, etc) and it prevents the need to re-write selectors multiple times.

* **(Optimization) Explain some techniques to optimize page load with a large scale project that uses multiple internal and external resources (JavaScript, CSS, images etc.)**

- Overall it is better to use external over internal because of performance and maintenance. It is better to use external files (for js, css, images, etc) because it can be easier cached by the browsers.

- Minimizing the number of files referenced in a web page also lowers the number of HTTP connections that are required to download a page.

- Compressing your javascript file smaller by getting rid of whitespace and comments

- Placing the javascript files at the end of your HTML file, this will allow the images, tables, text, etc to be loaded and rendered first.

- Specifying the size for images and tables, this allows to display a webpage without having to reflow the content. It's good practice to specifiy the height and weight for images so that it prevents the web page to change in a page's layout when loading.

- Applying CSS in the head of the html page so that way the CSS styles load ahead of time.