Please answer the questions below.

Where we ask for you to provide code, this code can just be ‘pseudo code’ and does not have to be executable.

The idea of these questions are basically just to get an idea of your understanding. You are allowed to skip questions if you cannot answer but of course the more you can answer the better.

Completing all the tasks should not take more than an hour to an hour and a half.

Please create a repo on github to host your answers and test code. You can either do this on your own public repo or share ‘kkroese’ and ‘devblazer’ on your private github repo. Please do not include any OS specific files like microsoft word documents. Also do not commit binary files like zip, tar or gz files.

Please send the link to your repository to Kevin.

Javascript:

1. What is your favourite new javascript feature and why?  
   String.replaceAll(), because I can replace some words in a string simpler
2. Explain an interesting way in which you have used this javascript feature.  
   For example, there is a string variable that contains a long paragraph. I don't need to trim and loop to replace some text. I just use replaceAll() which is simpler.
3. Is there any difference between regular function syntax and the shorter arrow function syntax? (Write the answer in your own words)  
   Yes, the syntax writing. I think the shorter arrow function is simpler than the regular function. And programmer can create many functions simpler with the arrow function than a regular function.
4. What is the difference between ‘myFunctionCall(++foo)’ and ‘myFunctionCall(foo++)’  
   myFunctionCall(++foo) means the parameter is already increased, while myFunctionCall(foo++) means the parameter is not increased
5. In your own words, explain what a javascript ‘class’ is and how it differs from a function.  
   Javascript class is a container that is used in an object-oriented structure. The class can contain variables, functions, and constructors. The difference with function is you can create variables and functions in one container and follow the object-oriented concept.

Css:

1. In your own words, explain css specificity.  
   CSS describes how HTML elements are to be displayed on the screen, paper, or in other media and CSS controls how the view on screen and other media.
2. In your own words, explain, what is ‘!important’ in css. Also how does it work? Are there any special circumstances when using it, where it’s behaviour might not be what you expect?  
   ‘!important’ means the selected style will be prioritized first. The special circumstances are when we use some theme or template, we will modify some view, some view can’t be modified because it uses existing css, the ‘!important’ allow us to modify the style.
3. What is your prefered layout system: inline-block, floating + clearing, flex, grid, other? And why?  
   Flex, because it’s easy to organize element sand it makes easy to solve many cases like centering an element horizontally or vertically.
4. Are negative margins legal and what do they do (margin: -20px)?  
   Legal, it makes element position is out of its box.
5. If a <div/> has no margin or other styling and a <p/> tag inside of it has a margin top of some kind, the margin from the <p/> tag will show up on the div instead (the margin will show above the div not inside of it), why is this? What are the different things that can be done to prevent it?

Because <p> is not actually inside <div/>. This can be prevented by adding ‘overflow:hidden’ in <div/>

Unit tests:

1. What technologies do you use to unit test your react components?  
   react testing libraries, enzyme, and jest
2. Are there any pitfalls associated with this technology that have caused you difficulty in the past?  
   No
3. How do you test in your unit tests to see if the correct properties are being passed to child components.

Using .state to check the properties and check its value equal with the desired value

React:

1. React test step1:  
     
   Create a react component that has a <div/> with a border.  
   Inside this <div/> should be a <span/> that displays the ‘live’ width of the browser window at all times. Keep in mind that the size of the window could easily be changed by the user and you should reflect this.
2. React test step2:  
     
   Inside the <div/> you created in the previous step, add a text input that, as a number is entered into it, uses that number to set the height of the div itself in pixels, live as you update the text field (keypress not change event).
3. React test step3:  
     
   Add the following code to your project root (same project as in step 2, but add the code in the global / window space):   
     
    Let divHeight;  
    window.setDivHeight = (height) => divHeight = height;  
     
   Add a HOC for your div component that allows you to set the height of your <div/> component from the previous steps by calling that external function.  
     
   If you do not know what a HOC is or how to create one, that is also fine, just mention that in your answer and instead create a parent component that can still do this (allow you to call that function ‘setDivHeight’ in order to set the height of the div manually.  
     
   Bare in mind that when the height of the div is forcefully set like this, the text fields value should also update to reflect this and should still carry on working as normal (user can continue to modify its value).