**Part 2: Changes to your application**

1. You have a new requirement to implement for your application: its logic should stay exactly the same but it will need to have a different user interface (e.g. if you wrote a web app, a different UI may be a REPL).

Please describe how you would go about implementing this new UI in your application? Would you need to restructure your solution in any way?

* I would improve the current UI from a REPL application to a web-based user interface. For this, I would create it as a form with two user input fields that accepts integers for time interval and number. Similarly, I would create the halt, resume and quit labels as buttons.
* For this new UI to be properly implemented, I would need event listener for the buttons to be implemented. Similarly, the input fields would need to be queried based on the text field ids (document.getElementById(“id”).value) or create an event listener and use event.target.value in react.
* Similarly, the timely display of the frequency can be made as an alert that pops up every desired second.
* The overall functionality would remain the same and only some changes to adapt to jQuery/react.

1. You now need to make your application “production ready”, and deploy it so that it can be used by customers. Please describe the steps you’d need to take for this to happen.

* The first step would be to create wireframes and prototypes to determine the functionalities and requirements of the application.
* Then create a version control system like Git or Bitbucket to track the progress of the application and for change management.
* During the development phase, the tasks can be categorised based on priority and converted into several tasks while testing for the functionalities.
* After the app is fully developed, it needs to be sent for unit, QA and content testing to ensure that all the functionalities are working properly, the requirements are met and the content is checked for any copyrights.
* Once the QA is complete, it should be ready for deployment which can then be hosted using cloud computing service provider like AWS.

1. What did you think about this coding test - is there anything you’d suggest in order to improve it?

* The test was very interesting to code and apply logic to it. However, the timer functionality was a bit challenging as the results were being displayed only at the end of the program. In a real-case scenario, would probably need a little guidance from a supervisor, just to see the solution from a different perspective.