webqem JavaScript dev test

This test helps us gain an understanding of how you work with a spec and your approach to solving a problem. The solution should be implemented as a single page app using a UI library of your choosing and should be well documented so that another developer can build and run the project from the resource in the repository.

The Spec

The ANU Quantum Random Numbers Server provides an API that will return a JSON response with an array of quantum random numbers . Requests are made using the following format:

```
https://qrng.anu.edu.au/API/jsonI.php?length=[array length]&type=[data type]
```

Examples

Requesting 10 random numbers between 0–255

```
https://qrng.anu.edu.au/API/jsonI.php?length=10&type=uint8
```

Requesting 5 random numbers between 0–65535

```
https://qrng.anu.edu.au/API/jsonI.php?length=5&type=uint16
```

Implementation

- You are to use the QRNG API to implement a page that takes the number and type of random numbers to request from the server. It is estimated this will take between 1 and 2.5 hrs to complete.
- Before starting with the implementation, estimate how much time this project should take for you to build and document this time. Keep track of how much time it took to complete this process and write a short sentence on the actual time vs your estimates to be included in the README for your project.
- The app will implement a simple form that has a **number input** and a selection for the data type **uint8** (values 0-255) or **uint16** (v alues 0-65535).
- A loading indicator should be shown when a new set of numbers is requested from the server.
- The results are to be plotted as a chart, we recommend a bar chart but you are welcome to use another if it makes more sense for you. You are free to use any library you need for displaying the charts such as Chart.js or Highcharts. Each element on your chart should have a unique colour.
- The random number data should also be presented as a list or a table which shows the array index and the value of the random number.
- The implementation should use a JavaScript library such as React, VueJS or Angular.
- You are free to use any CSS framework you are familiar with to style this app. This test is not for your design skills, but will be used to evaluate how you organise your Front end resources.
- You are free to choose a build tool that you are comfortable with such as grunt, gulp, or webpack. If you use any boilerplate project as a starting point, please document this in your README and explain why it was chosen.
- Your README should also include instructions for another developer to run your project from scratch.
- Once complete, upload the project assets to a public repository on GitHub or Bitbucket and share the URL via email.