Freshworks Studio Selection Test

1) My approach to the problem

My first step was to break down the requirements of the task. The short paragraphs before and after the list of bullet points contained a lot of information regarding the requirements as well. I wanted to make sure I had the list of requirements written down so that I could quickly refer back to them. I additionally drew out a rough application architecture diagram so that I had the big picture cemented in my head.

I first began work on the front-end as that's my usual approach. The data entry form was the most complex component of the project in my eyes so I built out a rough form to get started. After that I quickly stubbed out an endpoint on the backend so that I could test the form submit functionality. The rest of the backend functionality, which included persisting the data to the database, was implemented last. I followed the same process for building out a simple view with which all of the entries could be viewed.

My first priority was to have a running and functional project. Once the functionality of the application had been implemented I went back to focus on the user interface and tidy it up. I also made sure to include form validation and a simple test for both the client and server to round out the assignment.

2) Technologies chosen

As a personal preference, I chose React.js to build out the front-end portion of the assignment. React.js allows a developer to quickly build out the user interface. Its focus on components and modularity makes it incredibly flexible. I included Typescript for this assignment as well. Typescript's type checking capabilities can help catch errors very quickly. It also provides structure to the project, unlike plain Javascript, which makes the code much for readable.

I chose to use Node.js for the back-end because I have prior working experience with it and it complements the Javascript frontend. I included Typescript with the backend as well for the same reasons mentioned above. I wanted to design and build a rest api to manage the data entries so I included the Express framework on the backend as well.

Additionally, I included the following libraries in the front-end code base:

- create-react-app: This toolchain does away with a lot of boiler plate code and configuration so that the developer can focus on building out the actual application.
- Material-UI: A user interface component library. This library provided me with an out-of-the-box solution for a dialog box.

3) Number of hours spent on test project

I put approximately 9 1/2 hours towards the selection test assignment.