Completing the Coding Challenge

In order to complete the coding challenge the best way possible given the allotted time, I handled the challenge as I do a usual sprint at work. I broke up the challenge into three major features: database configuration, API to send in data, and API to calculate the data. I took those features and wrote down the list of tasks that would be needed to complete those features. My first mvp, which can be found at github.com/britneyepps/stockxchallenge (private repository) was just about learning the go language. I worked on database configuration, how to set up APIs, and receiving the data from the repositories. I didn’t focus on the layout of the program so everything was in one file. The second mvp, I structured the code as if it was a springboot or any other regular MVC project. I split the repository away from the API calls, created structs as models to pass information back and forth, and made sure everything from mvp one still worked. Once the structure was set, I was able to focus on the front end to connect everything so it wouldn’t have to be only tested in postman. Finally, I focused on packaging everything up to be passed to another developer, minus the code comments.

I ran into multiple issues once packaging the data. I was able to successfully create flyway scripts in order for the database needed entities to be there such as the schema, tables, and some test data to start. However, once trying to connect Postgres, Flyway, and the App using docker-compose, it quickly ballooned into a larger than thought task. Docker-compose had computer specs that most of the computers in the house did not meet. Once I realized I would most likely need outside help and more time to get up properly, I decided to package it the best possible way. I left the Dockerfile and YML files to show how close I may have been using the containers for easy environment setup.

Overall, learning the language turned out to be relatively easy once I found the best tutorials to walk through and understand the syntax. I spent quite a bit of time on the front end wiring and the containerization that ultimately did not work out.