8/13/25

Architecture Style

The chosen architecture style I chose for this project is the Model View Controller(MVC) architecture. I decided to take this route for this project for many reasons. Some reasons will be simple such as, I'm most familiar with MVC when working with web applications, while others will be more aligned with the scope of the project. There will also be things that don't fit fully with the idea of this project workspace.

When I considered the requirements of the project, I noticed that each of the requirements are slightly coupled in what they are trying to achieve and could be expanded upon more if desired.This reminded me of the MVC style of my programming in how I like to separate my code into different sections to be prepared to be used again in another section. This can be seen in my Controller.js functions where I tried to implement the availability section of the code separated from the Get functions and this has allowed me to develop different parts of the functionality at different intervals while testing their functionality. Another reason as to why I used MVC is because I was going to be working with a non-relational database. The controllers would not be directly accessing the data and instead would be passed to and worked with inside the controller. This will lead to some of the shortcomings I had for this project.

At the start of this I called it Model View Controller architecture design, but with the layout of this project, I only was required to develop the Model and Controller for this API. The view became what the JSON messages were being sent out. During the process of creating the controllers I also noticed that if the controllers were not spread out correctly, they could lead to large complexity and readability for what is happening inside them.

I have enjoyed the process of following through with MVC architecture. It had its uses to help keep things separated so that I could focus on one functionality and come back to the other functionality when ready. It provides a nice way to break up the work and allow the program to be flexible without relying too much on one function.

Christian Rosales