Building my first full stack project I encountered a few things that I hadn't ever really considered before. Those things mainly revolved around how when you deliver something up from your server, your map starts in relation to the first item delivered up. My application was a scheduling application that a small doctors office could potentially use and the first file delivered was a login screen. I was extremely puzzled for a time trying to figure out why my redirects were not working the way I thought they would be, then once I realized what the correct URL redirect was it clicked that I had organized all my linking and mapping in a way that only worked on a local machine. After realizing this I was able to remap and re-link the files in the correct way to deliver up a fully functional, full-stack web-based scheduling application.

Another quirky bug I encountered when building this project occurred during an API request from the front end to the server for the list of calendar events being populated on screen. Every time the screen refreshed the events would duplicate. Also, any interaction with the calendar caused all the events on screen to duplicate. I read just about every StackOverflow similar question, as well as scoured the package documentation trying to find out information but ultimately, they were unable to provide an answer for my specific bug. I went back to my code after thinking about the problem for a little bit and decided to experiment with the method the controller file was using to deliver up the calendar events through the API endpoint. I realized that the way I had written the method was essentially the cause for the duplication, so I broke up the method into smaller methods and moved the main generation function outside of the exported modules object. This solved the problem for me. The fix changed the number of times the events were being looped through, rather than every time the endpoint was being generating a loop, I just made the loop occur upon server initialization and when a new event was added to the server. Then every time the page was refreshed the static copy of the completed loop was served to the client.