MERN Assessment Project Greg Holdich

Initially for this project, I pulled together a group of node modules that I knew I would need to implement a full MERN stack. This included react-scripts, express, axios, body-parser, foreman, mongoose and nodemon. However because I wanted to customise the configuration of react-scripts, I used the command “npm eject” which basically replicates itself into your project directory. From this I was able to fully alter and customise the setup to better suit the project. The alterations included changes to the folder structure, paths, webpack-dev-server protocol, and removal of node modules and files that were not relevant for the project. Express was utilised as it was a requirement and was used as a base web server to handle client requests to retrieve data from the database(mongoDB) and parse the response into JSON to send back. Additionally, express was setup to handle CRUD requests, so methods were created to Create(POST), Read(GET), Update(PUT) and Delete(DELETE). I chose axios to make http requests and to handle responses from the express API, because I felt it would simplify the complexity needed to send client requests, and also generally perform better than the standard fetch method. I chose to use body-parser to parse req.body data in post requests, to make parsing body data simple. I installed foreman to launch both the development sever and express API simultaneously. In addition, mongoose was used to create schemas and for general querying of mongoDB as a driver. Nodemon was used to make express easier to development, as nodemon watches express and if anything changes, the express API is reloaded. Similar to how webpack watches the development sever.