webPortal Development Log

**Technologies Used**

Frontend:

React 18.3.1

Backend:

NodeJS v21.3.0 / ExpressJS v4.21.0

Database:

MySQL Server 8.0.36

MySQL Workbench 8.0.36

IDE:

Visual Studio Code 1.93.1

OS:

Windows 11 Home Version 23H2 Build 22631.4169

**Application features**

* Enhanced logging (morgan)
* Browser alerts (express-flash)
* Session management via MySQL (express-MySQL-session)
* Content Security Policy (CSP) security via added headers (helmet)
* Cross-Origin Resource Sharing (CORS) security enabled (cors)
* Data sanitization and validation (express-validator)
* State management (Redux)

**Features NOT included in this current version**

* Password encryption (bcrypt)
* Middleware, such as route protection to ensure a user must be logged in and be the correct user
* Unit testing (Jest)
* User interface and visual experience improvements

**Default login credentials**

Admin account:

Username = user

Password = pass

Basic user:

Username = user2

Password = pass2

**Basic Project Directory Structure**

Frontend:

*Conforms to standard React directory practices*

---------- webclientportal/

|---------- public/ (stores all static assets)

|---------- src/ (stores all JS and CSS files)

Backend:

*Conforms to standard NodeJS directory practices*

---------- app.js (entry point of the server)

---------- .env (environment variables)

---------- package.json (configurations)

---------- package-lock.json

---------- README.md

---------- src/

|---------- routes/

|---------- middleware/

|---------- utilities/ (database and Redux store JS files)

**All commands used in building the application**

Frontend:

npx create-react-app webportalclient

npm install react-router-dom

npm install @reduxjs/toolkit react-redux

Backend:

npm init

npm install express

npm install nodemon –save-dev

npm install ejs

npm install dotenv

npm install bcrypt

npm install morgan

npm install http-errors

npm install cookie-parser

npm install express-session

npm install express-mysql-session

npm install mysql2

npm install helmet

npm install express-flash

npm install cors

npm install express-validator

**Reflection from the Author (Joshua Lor)**

This project took a total of two days to complete. Juggling between computer science homework done in Java and learning the Python programming language, this project was a refresher to the fundamentals in web application development. I enjoyed the process of building this app and felt like it definitely sharpened and honed in some key skills.

There were some challenges and obstacles I faced while building this app:

Review of JavaScript syntax: While a lot of it was review from college course web projects, it took some undertaking to review and remember proper arrow function syntax, mapping, hooks, fetch API formatting, etc. Once I got used to it again, productivity went faster. It definitely helps after you do produce correct code, you can mostly copy and paste that code and alter it for alternative uses.

Adjustment from React-Native to React: While I did have prior experience with React Native, I did have to make some adjustments to developing the app in pure React for mostly syntactical reasons. Routing in React is slightly different from React Native (React Native’s NavigationContainer v.s. React’s usage of the Router, “navigation.navigate()” compared to “navigate()”, etc.). But once again, after making adjustments to the syntax I was already familiar with, I was able to breeze through and create the API endpoints that I needed.

Learning Redux: I also picked up the basics of Redux, from concepts of the Redux store, to reducers, to dispatchers and selectors. State management is an absolute must for large-scale applications, and I intend to keep using it in further applications and building upon my skills with it.

Understanding more security via CORS and CSP: Security is essential in all software development, and I decided to challenge myself to learn and implement these two things to make the application more secure. CORS did give me a headache, making me wonder several times why my HTTP requests weren’t getting through to the server, but properly learning its options to set it up was key. Helmet is a powerful security enhancement that was actually rather easy to implement.

Overall, it was a pleasure building this simple app, and I hope you find something to take away from this if you are learning web application deveopment!