Software Design and Engineering

Lab Document

High Level Purpose Statement:	The goal of this lab is to explore a web framework. I chose to use Express, due to the increasing demand in industry and for its reputation. I also used Express for routing purposes. When a REST call is made, "server.js" (implementing Express framework) routes the request to the Spring Boot API.
Experimental Design:	This lab will be divided among the following components: • Spring Boot Framework configuration (REST Controller, Service classes, DAO classes) • **Backend Configuration** • Frontend Configuration • Compatibility (ensuring each component can communicate with each other and work together efficiently) *Implementing Express is part of the Backend Configuration. This project has been used in the past for the Spring Boot lab, so the experimental design has not changed.*
Resources Available:	**Websites (Tutorials/Documentation):** https://expressjs.com/en/guide/routing.html I also plan on utilizing Copilot to assist with JavaScript. JavaScript is pretty scary.
Time Estimate:	I am estimating this project to take between 12-20 hours. *(I was soooooo wrong. 45-50 hours later, here I stand.*) **Addendum (Spring Boot Pt II):** Because I did *so* much of the work for Part I, I am only expecting to spend 3-6 hours on this lab. **Addendum (Express Lab):** I estimated the configuration to take between 2-4 hours, including reading/watching videos.
Experiment Notes:	**Backend Configuration** To configure the backend, I implemented a Node Express framework to assist with routing REST calls. This was not *too* difficult to learn and implement. The main challenge was configuring CORs *and* knowing it was an issue. All of my HTTP requests were being blocked and returning errors. I probably spent about 2-3 hours banging my head against a wall because of CORs. It had a simple fix, but I did not realize how important it is to manage its configuration.

	NOTE: I "completed" (the final product is not completed) this lab weeks ago in order to allow myself time in the future, as I am getting surgery this week and tackling many final projects/personal obligations this upcoming week. I have spent roughly 55-60 hours on this project so far.
Results:	The current features are: • Individually listed categories for degree program • Remaining required credit hours in each category • Interactive courses from which users can toggle completion Although I originally wanted to generate at least a *basic* schedule by the end of this project, I think it's for the best that I stop at this point. Building the full-stack web app was much more challenging than I anticipated, so I did not have as much time to add more features. I will say that I am very disappointed in the deployment of this web app. I originally wanted to use Docker to containerize the app, but I was having issues with connecting to the database. I scratched that idea, and then once I thought the project was finally finished, I submitted it to a guinea pig, and the web app would not even build on his system. So, I tried to utilize Docker again… five hours later, I was still having the same issue with web app connecting to the JDBC driver via a Docker container. After giving up on Docker, I cloned the repo to a separate device and used it to test new modifications to the web app. I hope that I can figure out how to use Docker correctly eventually… **Addendum (Spring Boot II):** The current features are: • Individually listed categories for degree program • Remaining required credit hours in each category • Interactive courses from which users can toggle completion • Prompt for user to select of semesters to make schedules for • Prompt for user to enter time constraints The mission to employ Docker failed again… although I'm very disappointed (again), I know that I should probably explore other options. Maybe I should try uninstalling and re-installing Postgres and Docker. **Addendum (Express):**

	Express was relatively straightforward to use! It works very well, once CORs is configured (that was sooo frustrating).
Consequences for the Future:	I think that next time I build a full-stack web app, focusing on the deployment should be an early step. Instead of spending so much time trying to add features, I should have ensured everything was compatible on someone else's computer first. Though, I could argue that I would have probably wasted a lot of time trying to containerize the app (and failed) and run out of time on adding features. Interesting thought. **Addendum (Part II):** I followed the advice I gave myself from the last lab to explore deployability at the early stage. I think that was an excellent decision, although I did not get the results I was hoping for. I did most of the configuration in the last deadline, so I believe that is my biggest takeaway as far as Part II goes. **Addendum (Express):** If I find myself creating another full-stack web application in the future, I will definitely use Express again. It does an excellent job at routing REST calls and is easy to set up.