CVWO Assignment: Final Writeup

Table of Contents

The Assignment

I came into this assignment with some prior experience in web development, which is really what got me interested to sign up in the first place. After hearing seniors' experiences in past years, I must admit I was quite surprised by the complexity of this year's assignment. The requirements for implementing user authentication, backends hooked up to relational databases were all quite interesting. These were things that I didn't have the time or courage to explore in my past web development projects.
All in all, I decided to give myself a bit of a challenge by choosing Go as my backend, and planning to implementing several optional requirements, like JWTs and Docker. I was well aware that I would have to scale down some of my plans as I went along, but ended up feeling quite satisfied at having achieved most of what I set out to do.

Mid-assignment Submission

The first time I truly sat down before a project to plan it out. Through my research, I learnt all about use cases and user stories and how big software development projects are carried out with these tools aiding in their planning. It was also my first look into the world of relational databases, where I learnt about entity relationship diagrams and the kinds of relationships inside these databases.
The submission really helped to remove some of the fog around the assignment and got me thinking about the best way to move forward.

The Code: Golang and Chi

Learning Go from scratch while trying to get a relatively less mature web framework running was a time-consuming process. It didn't help that the official Chi example for a REST API came in the form of one big main.go file. But in the process of incrementally building the web server and slowly rearranging the provided example into different parts of my Go project, I came to understand the reasoning behind the MVC framework, and its value in keeping a project logical and neat.

The Code: GORM

With DBaaS (CockroachDB) simplifying most if not all the database setup, and GORM's succinct documentation, linking my backend server to the database was quite straightforward. I could really appreciate the high-level abstraction that GORM provided, and while coding I was able to get some practical experience with the then-unfamiliar concept of pointers and references, thanks to GORM's constant use of said concept.

The Code: React and react-router

Although I had some experience with React, react-router was a brand-new experience. It was during this phase of development where I learnt a harsh lesson – always read the documentation first. Otherwise, you end up spending a lot of time refactoring your code to use new features like createBrowserRouter and RouterLoader.

The Code: Docker and Deployment

The setup was somewhat long-winded, but ultimately building an image and deploying the containerised application on GCP was a fuss-free experience. Containerisation is really useful, and it shows. Seeing the live version of your app load up is a reward in itself!

Looking Forward

This assignment took me through the whole process of full stack development, from planning all the way to deployment. I can confidently say it is by far the largest web development project I have ever completed, and I am quite proud of what I was able to learn and build during this project.
However, it is far from perfect. The backend could use better logging and more unified error handling. The frontend could benefit from more consistent form handling and better UI with mobile responsiveness. There are many optimisations to be made.
As with everything else, it's a learning process :)

User Guide

Browsing and Searching

New users are immediately able to view all threads and comments. Through the search field in the header bar, they are also able to search threads by title. More search filters can be accessed by clicking the filter button next to the search bar. Search filters allow users to further filter threads by tags and author, as well as adjust sort order.
Tag search syntax: Each tag is alphanumeric, separated by a semicolon.

Creating an Account

When not logged in, the Log In button in the header bar opens the Log In/Register menu. Once logged in, users can log out with the Log Out button in the header bar.

Creating, Editing, and Deleting Threads and Comments

Once logged in, users can create new threads using the “+ Thread” button in the header bar. Upon viewing a specific thread, they may post a comment under the thread using the text box under the thread content. If they are the author of the comments or threads that are currently being viewed, they may choose to edit or delete them with the respective buttons.