**CVWO Final Write-Up**

Aw Hang Bin (A0252492A)

Reflections

This assignment has given me an opportunity to expose myself to React and Ruby frameworks, which I have never been exposed to before this. Though I had some previous experience was with HTML, CSS, Javascript basics, with Django/Flask exposure from the popular CS50 course, the structure of a React-Rails framework greatly differed from the Django framework I had been used to.

In trying to get familiarised with both frameworks, I sought to find resources on the net in order to learn. I did learn some basics, but in retrospect I should have started building the app earlier and learned as I did it. The greatest learning comes during practice, after all.

Another thing I felt I learned from this experience was the need for independent learning. In my previous developing experiences, I was used to video lessons and did not really like reading the docs for syntax explanations. The scale that this app demanded, however, made it necessary to explore and understand more about the framework and functionality. And this turned out to be a great habit to start building, because reading documentation is something that all developers have to get used to.

A third important aspect I picked up from doing this project was the need to plan ahead, making progress in incremental steps. Unfortunately, I learned this the hard way; often I would work on code for hours only to realise something went wrong, and it was an insane headache to backtrack and scour through everything just to locate the source of the problem. Incremental work using git for version control was definitely the way to go to address this issue.

**Frontend**

React, Redux and Typescript were all completely foreign to me. The only reasonable exposure I had to Javascript was through CS1101S, which wasn’t really practical.

I did have some fun when I figured out how things clicked, however. Part of how I learned was through the docs, but I thought Youtube videos definitely bridged the gap for me to understand how these frameworks would be set up in various kinds of web app examples to accomplish various kinds of tasks.

**Backend**

I chose Ruby with the mistaken impression that it would be easy to learn. Sure, the syntax is easy enough, but the infrastructure of a rails app requires quite a bit of exploring to understand. Another issue I kept being frustrated by is the lack of comprehensive error-checking in the rails app. Some errors are simply not detected by the rails controller until during runtime, and the errors aren’t thrown in the same way that they are thrown in React, hence problems that cropped up took

Another issue I had was blindly following one of the CVWO-recommended guides that advised to implement the react frontend under the app/javascript folder of the rails app. I kept trying to force that structure thinking it was the way to go, but the typical react debugging capabilities were blunted and it was frustrating that when there was a bug, the page would either turn out a whitespace or reload a previous working version of the site. I have wasted immense amounts of time backtracking because I wasn’t aware when the site stopped working normally, or what exactly the part of my app that caused the error was. I ended up switching to using two separate repositories for the frontend and backend, but the time wasted was a harsh lesson that in app development, debugging practicalities are an extremely important issue to consider.

**Final thoughts**

Unfortunately, I am still rather inexperienced in planning ahead for completing such a project so I wasn’t able to see it through from start to end given the time I had. Though the experience was no doubt challenging, it was also fulfilling in a way that made me understand and appreciate web development beyond a beginner level overview. I don’t think my finished product is exactly functional as I’d hoped it would have been, but I think that regardless I still did learn quite a lot about the frameworks behind the project while working on the it. I will continue refining the project after this has been graded, and I do think that this experience will be a springboard to me improving as a developer for future projects, under CVWO or otherwise.

User Manual

Graphical user interface, application

Description automatically generated

(Login screen, with menu bar on top)

Graphical user interface, application

Description automatically generated

(Sign up bar for users)

Text

Description automatically generated with medium confidence

(thread view with search bar functionality: toggles between search by title and category)  
  
Graphical user interface, application

Description automatically generated

(view each thread and the associated comments)

Graphical user interface, text, application, website

Description automatically generated

(user info page)