CVWO Writeup

During the holidays and my first few weeks of school, I spent a lot of time on this CVWO project and learnt many things along the way. I never had any web development experience beforehand and had to learn everything from scratch. However, after many weeks of hard work, I managed to build a functional to-do application with a tagging feature. I also managed to host it on Heroku. This is the link: https://shrouded-castle-05452.herokuapp.com/.

This is the link to the Github repository: https://github.com/pooty3/CVWO

Over the course of the project, I learnt how to use ruby, rails, react, javascript, postgresql, html, css, git and heroku and how to integrate all of them together to make a functional web application. One of the most valuable thing I learnt however was how to source for information alone. After many hours of watching and finding tutorials, I finally managed to have a working understanding of the whole application and managed to start building. I faced many difficulties along the way. The most notable of which was to integrate react to rails. I originally wanted to have 2 separate applications, a react frontend with a rails backend API but I was unsure on how to deploy it. In the end, I decided to use the react-rails gem. Another difficulty was to get my CSS files to work on Heroku. While the CSS modules work perfectly during development on my local machine, it doesn't seem to port over to Heroku. In the end, I had to manually copy all the CSS code directly into the stylesheet CSS file for it to work on Heroku, which can be quite unclean.

Currently, while functional, the application has quite a few performance flaws. One of which is that the application had to re-fetch the whole database every time a change occurs which may cause performance lags. Another area of improvement would be to instill a user and password system so as that different users would have their own unique to do list. Currently, the to-do list is accessible and editable to anyone who accesses the site which could be a problem. Redux could also be incorporated into the application. Currently, I use React Context to manage state which is still good enough for this scale of the project but converting to Redux may be a better alternative if the project gets bigger.

All in all, I learnt a lot of things from frontend to databases to using git to save my work and it was a great experience for me. I hope to continue learning more along the way.

The User Manual is included in the application but I shall include it here as well.

There are the homepage, the about page, the user manual page, task and tag page.

The task page is where you would add it new task, edit them or delete them. The tag page is the page where you manage your tags so that it can be used in the task page.

A task must not have an empty title for it to be saved. A tag must not be empty as well and must be unique. No duplicates are allowed.