CVWO Assignment: Riding on Rails

Cai Kai'an (A0222378A)

Unfortunately, as I only found out about this assignment days before this submission, there is currently no source code available for viewing. Instead, I will be discussing about my plan with this assignment and what I hope to achieve.

Personal Goals. Before I found out about this assignment, I was learning React on my own during the winter holidays in preparation for a module I'm taking next semester, CP3108, where I will be working on the frontend of Source Academy (https://github.com/source-academy/cadet-frontend). Aside from that, I am also very interested in picking up web development skills, hence I see this assignment as a godsend that can really aid me in learning full stack development. Quoting from the assignment overview, I hope to "build up a good web development foundation through this assignment", and hopefully deploy my skills through CVWO or for my own personal projects.

Goal for the App. I hope to deliver a performant, minimalistic to-do manager that is able to perform all of the uses cases I will describe below. I hope to pick up and follow good coding practices so as to produce clean code. Due to the upcoming academic semester where I will be overloading, I foresee myself pressed for time to work on this project. Hence, I will scale back my expectations in terms of number of features and focus instead on improving and polishing the core features.

Use Cases. As stated above, I won't be pursuing too many features but instead will be focusing on these core use cases:

1. User Creation/Authentication: users will be able to sign up and log in across devices so they can keep track of their individual to-do lists
2. To-do Manager: in addition to basic CRUD operations, user will be able to tag to-dos into categories/projects and also set priorities on their to-dos
3. List View: I will be implementing the traditional list view where to-dos will be sorted by deadlines. Each to-do will show a simple title, its deadline, and also any tags/categories set by the user. In addition, users will be able to select certain tags in a navigation pane and be shown only the to-dos tagged. If time allows, I will also look into implementing a calendar view
4. Search: users will be able to search for to-dos in a search bar

This is not an exhaustive list; if I am satisfied with how these turn out, I may look into implementing more features.

Execution Plan Looking through the requirements and optional tasks, I have decided to use Rails to build the backend in the form of a RESTful API. The frontend will be built with React with Redux and Typescript. By final submission, I hope to host the app on Heroku.

1. User Creation and Authentication
    a. User can create account in website homepage
    b. User can log into their account on website homepage
2. Basic CRUD operations for to-dos
3. List View for to-dos
    a. List of to-dos with title and deadline
    b. Sorting functionality
4. Categorising/Tagging Feature
    a. Users can add and delete tags on to-dos
    b. Add displaying and sorting by tags to list view
    c. Users can select certain tags to see all to-dos under that tag
5. Search Feature: search to-dos by title
6. Styling and polishing up the app
7. Deployment to Heroku

If time allows:

- Cron Jobs
- Docker

Challenges Being the first web application I will be developing, I don't expect this assignment to be without its challenges. On the backend side, this is my first-time using Ruby on Rails. For others such as React and Git, while I have briefly learned about them before, I am by no means experienced and will undoubtedly have to pick up and learn new things along the way. But this, I feel, is what makes this assignment interesting and exciting for me: the prospect of being able to slowly get my hands dirty with web development. I look forward to what I can achieve and learn in the month ahead.