Computing for Voluntary Welfare Organisations, AY2020/21 – Assignment

Name: Gan Hong Yao
Matriculation Number: A0217912H

Mid-assignment Report

Use Cases and Current Progress

Basic

1. Create tasks

Users should be able to create tasks. Each task would have the following fields: title, due date, description, tags and friends involved. Currently, the first three fields have been implemented and users are able to see these fields. The latter two are work in progress.

2. View tasks

Users should be able to view the tasks that they have created in different views. Upon login, users will be brought to the home page, which shows their current outstanding tasks. Currently, only the basic table view has been created. Additional features to the table view such as sorting as well as more different views would be implemented in the near future.

3. Update tasks

Users should be able to update and edit the created tasks' fields. Upon completion, users can check off tasks from their outstanding task list. This has already been implemented.

4. Delete tasks

Users should be able to delete tasks that they have created and remove it from all the views. This has already been implemented.

Intermediate

1. Search tasks

As tasks can pile up quickly, users should be able to search through their tasks.

2. Authentication

Users should be able to create personal accounts and login to them so that only they can view their own tasks.

3. Sort tasks based on priority

As different tasks would be of varying importance and urgency, users should be able to rank their tasks based on priority.

Advanced

1. Social feature

If possible, users can 'add' their friends on the application and use social media features such as tagging, commenting and chatting. Often, our tasks to be completed do not involve us alone. To include social features where we can see others' progress on a common shared task would be a boost to our productivity.

2. Projects

Some tasks may be part of a larger common goal that the user is towards. It would be good if users could group tasks under 'projects' so that they can see their overall project progress.

3. Timer

A timer feature would help users to keep track of time while they are at a certain task. By having a timer visible as they are working on the task, users would be more productive and focused.

The intermediate and advanced use cases proposed above have not yet been implemented as of time of writing.

Execution Plan

I have ranked the features to be implemented by their ease of implementation and level of importance. I will be prioritising the features that are of higher importance and greater ease of implementation.

Feature	Ease of implementation	Importance	To be completed by
Categorise tasks by adding tags			31 Dec
Search through tasks			7 Jan
Sort tasks based on different criteria			7 Jan
Authentication			7 Jan
Additional views			14 Jan
Project feature			21 Jan
Social features			21 Jan
Timer			21 Jan

Issues faced

As someone who never delved deep into web development, there were many areas that were unfamiliar to me. Though I have learned the basics of HTML and CSS before, I have never implemented them to build a CRUD application.

Since I lacked experience, I did not know what were the 'best practices' to follow while building my application. I was aware that some decisions would be extremely difficult to reverse. Because of this, I found myself constantly searching up the Internet or asking on public forums to understand what other developers recommended as they would have more experience. However, because there were so many different ways to solve the same problem, there is almost never going to be a one-size-fits-all solution. I had to learn some things the hard way, which was to go with one solution but then realising that the alternative would have been a lot better and thus changing my approach. For example, class components vs functional components in React. With the introduction of hooks, functional components were a lot more versatile and could serve as an alternative to class components. As I thought that it could streamline my code, I set out to set up my frontend with only functional components. After a while, I realised that state management became really messy and lifecycle methods were difficult to implement, hence having to change a few components to be class components instead.

Additionally, since I am working on this application independently, there are no strict guidelines or SOPs that I have to follow. Although I may be able to exercise my own preferences when making decisions for now, it can actually be a problem when I start working in a group context such as CVWO (if I am fortunate enough to be selected). One suggestion I have for the CVWO training team is to include some of the SOPs they have when writing code for their beneficiaries in the assignment requirements. Though my application has only the basic features as of now, the code is already starting to look messy. In a collaborative project, this may become an issue.

Final Reflections

In the past month or so, I have truly learnt a lot and come a long way in picking up web development skills. Without this assignment, I would probably find myself being a lot more lost. In the next month before the final submission, I will be polishing my application and hopefully host the final working application on Heroku for everyone to use.