**CVWO Assignment Final Write Up**

Name: Justin Siaw

Matriculation Number: A0229940Y

Email: e0686139@u.nus.edu

**Accomplishments and Setbacks**

This assignment was one filled with many setbacks but despite all, it was a fulfilling experience with many lessons. Being completely new to web development, I thought that I had made good progress in understanding the field of web development but at the same time, there will still be a lot more to learn.

One of the biggest lessons from this assignment was learning how to learn. The different languages and frameworks required to make a simple application were far from my initial expectations. Starting from Youtube videos from freeCodeCamp to looking for different solutions in StackOverflow, the biggest frustration and learning point was that not all solutions will resolve the same issues. Most solutions or help from online sources required some form of tweaking to fix our problems or needs. In my implementation of a Search Bar for my task management application, I found several online guides on various implementations however most of such guides functioned differently from the way I envisioned my Search Bar to work. In the end, I found that a combination from a couple of online guides as well as putting into practice what I have learnt about react hook from the implementation of a Sort function allows my Search Bar to work the way I wanted to.

Other accomplishments included the various technical skills that I have picked up along the way. It was interesting to learn about how the frontend and the backend of an application communicate with each other. Having to squeeze the learning of the REST API and the Model-Controller-View frameworks as well as the other programming languages into just a couple of weeks was tough but rewarding. Learning is a continuous process in this field, to remain relevant, it will take constant effort to learn and master the various tools available.

Throughout this assignment, there were many setbacks. In particular, I have found it rather challenging to implement a has\_and\_belongs\_to\_many associations. In my initial plan, I wanted to create an association between the users/members and the tasks. Allowing multiple users to work on a single task and allowing a user to have many tasks. I assumed that such implementation would be simple however as I progressed, I faced several obstacles. Some of the technical issues stemmed from my lack of understanding of the tools I were using. Most of the online resources I found implemented a Rails Frontend has\_many\_through association. Having to link it to a React Frontend made things a lot more challenging. In the midst of implementing such an association, I was faced with another critical design flaw. I realised that this implementation did not make full sense as I did not have a login system to allow different users to have different permission to view or edit the status of the task. My implementation also resulted in poor User Experience requiring some overhaul of my code for this implementation to work smoothly. As this realization occurred rather close to the dateline, I decide to scrap this implementation and instead spent the last few days adding codes to improve the User Experience.

Some takeaways from this assignment would first be to improve on the technical skills I have learned from this assignment. I still have much to learn about to fully understand the REST API as well as the various React components that I can make use of to improve my current implementation. Outside of these tools, there are of course a lot more tools for various parts of the development and deployment of a web application that I have yet to explore. Aside from these technical aspects, I have also discovered the importance of self-directed learning especially in the field of computer science.

While I did not complete all the standard practice for a full development life cycle, this assignment had certainly improved my understanding of some of these stages. With the end of this assignment, I would hope to continue the development of my current task management application to include more functionality - including the aforementioned points that I removed.

**User Manual**

1. **A README file containing the instructions of the cloning of the code is available on** [GitHub](https://github.com/JustinS00/Task-Management-App). **Alternatively, to use an online version of the app, head to** <https://my-simple-task-management-app.herokuapp.com/> **(skip to step 3)**
2. **Once the code has been cloned on the terminal run $rails s and go to** <http://localhost:3000/>
3. **You will see the homepage.**

Graphical user interface, text, application, email

Description automatically generated

1. Click on the “Get Started” button to start using the application. Start by creating a new task.

Diagram

Description automatically generated with medium confidence

1. Fill up the details of the new task in the form page. Click on the “Save Task” button once you have filled up the details and you will be redirected back to the Tasks page.

Graphical user interface, application

Description automatically generated

1. Back on the Tasks page, you will be able to Sort or Search using the “Sort” button or the “Search Bar” respectively.

Graphical user interface, application

Description automatically generated

1. To view your task in detail, click on the “View Task” button at the bottom of the task you want to view.

Graphical user interface, text, application

Description automatically generated

1. You will be able to update your task or delete your task using the “Edit” or the “Delete” button respectively. Note that once you delete your task it will be gone forever.