CVWO Final-Assignment Report

Name: Tan Jin
Matriculation Number: A0199565H
Application Link: https://todo-manager.tjtanjin.com
User Guide: https://github.com/tjtanjin/todo_website/wiki/User-Guide

Summary:
Having no prior experience with projects done in React and Ruby on Rails, embarking on this Todo Manager project proved to be both exciting and challenging. The scale of this project is also one of the largest that I have attempted so far, which makes it extremely rewarding to be able to deliver a final working application. For my Todo Manager, I have decided to approach it by constructing smaller projects to have their codes and logic cleanly separated. This makes my project manageable even as the application gets increasingly complex. As of this final submission, the Todo Manager is composed of 3 smaller projects (Ruby on Rails API server, React website and Python telegram bot) whose details can be found at their respective repositories.

The journey of creating the application was rife with problems. Starting with the frontend (website), a lot of React tutorials out there were done with class components. As I adopted a functional component approach, extra effort had to be taken to adapt my code to their solutions. This, coupled with my lack of experience with React meant there was a lot of googling and flipping of keyboards before certain functionalities could be successfully added. Furthermore, as I worked on my application and began to better understand React, it dawned on me that there were parts of my code that could be more cleanly implemented (such as using useReducer instead of useState). While I have yet to refactor my code, these learning points will go a long way in improving my future applications. Even now, React remains the hardest part of the project for me.

On the other hand, the backend project (API server done with Ruby on Rails) was much more manageable as most of the heavy lifting has been done for us. That said, there were specific features such as password resets, email verifications and email reminders that still took a significant amount of coding, configuration and testing for them to work. My biggest personal take away from the backend project was the implementation and handling of user authentication with JSON Web Tokens. I have never dealt with user authentication before and picking up such a skill opens up a whole new world of possibilities in terms of embarking on new interesting projects. In addition, I have also come to appreciate the rails magic which allows for the simple yet fast delivery of desired outcomes.

Todo bot (python telegram bot) is the third small project that completes the existing Todo Manager project. Initially, Todo bot was not part of the execution plan. However, this project was added after numerous feedback on how having telegram reminders would greatly enhance the user experience. As I have prior experience with python telegram bots, this project was completed rather quickly though there was a need to update the backend server to create new endpoints. However, this project was not without its challenges. The linking of Todo Manager and telegram accounts, as well as the formatting of reminder messages, proved very challenging. In addition, differing time zone was also an extremely tricky issue but had to be dealt with in order for my reminder messages to reflect the correct information (days). Although not the main part of the project, creating the Todo bot still greatly enriched my learning experience.

Area for improvements:
Firstly, I feel that my coding style and the structure of my project could have been a lot better. This is especially true for my frontend project (React), where although I tried to use different folders and files to provide code clarity, there were still shared functions within the files that could have been abstracted (such as API calls).

Secondly, certain parts of my application do not appear well on the mobile view. I have tried to fix this but strangely, despite the mobile view on developer tools appearing fine when tested locally, the mobile view appears different on production. This is an issue I am still looking into and hope to solve since mobile devices receive more screen time than computers these days.

Lastly, the telegram bot currently only serves as a platform for receiving notifications and reminders. However, it has a lot of potentials and should features such as allowing CRUD operations on tasks through the telegram interface are implemented, I believe the convenience it can bring will further enhance the experience for users. With the backend project already well set-up, I hope to implement such a feature on the telegram bot in the future.

Conclusion:
Looking back, I am really grateful for the learning opportunities that I came across throughout this project. Despite facing countless obstacles along the way, the sense of satisfaction from getting pass each of them made all the effort worthwhile. There were times I experienced downright frustrations and times when I was overjoyed, but what was true all the time was that there was always a takeaway. While I may have wrapped up on this project, I am still very eager for new opportunities to enhance my knowledge and pick up new skills/technologies. For the upcoming summer, I would greatly appreciate an opportunity with CVWO and will strive to make the best out of my learning experience while making a meaningful contribution back to society.

Notes:

- The link to the user manual can be found at the top of this writeup - I have included a link because my user manual includes screenshots that will exceed the page count if included here.
- The technologies used for this project can be found under the project README.