CVWO Mid-Assignment Report

Name: Tan Jin
Matriculation Number: A0199565H
Heroku Link: https://project-todo-manager.herokuapp.com

Basic Use Cases (Users):
- Sign up [C]:
  - User navigates to https://project-todo-manager.herokuapp.com/signup.
  - User enters details and upon successful validation, a new account will be created and users will receive a welcome email.
- Login:
  - User navigates to https://project-todo-manager.herokuapp.com/login.
  - User enters details and upon successful authentication, will be logged into the account.
- View Account Details [R]:
  - User navigates to https://project-todo-manager.herokuapp.com/profile.
- Update Profile [U]:
  - User navigates to https://project-todo-manager.herokuapp.com/profile.
  - User clicks on edit profile or change password button to update their desired information. Password has to be provided for any profile changes. Upon successful validation, their account will be updated.
- Delete Account [D]:
  - User navigates to https://project-todo-manager.herokuapp.com/profile.
  - User clicks on delete account. Upon confirmation, their account will be deleted.

Basic Use Cases (Tasks):
- Creating Task [C]:
  - User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - User clicks on the create task button and enters task details into the modal. Upon successful validation, a new task will be created.
- View Task Details [R]:
  - User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - Under actions/tools, users can click on the info button to open up a modal to view more details of their desired task.
- Editing Task [U]:
  - User navigates to https://project-todo-manager.herokuapp.com/tasks.

- ○ Under actions/tools, users can click on the wrench to open up a modal to edit their desired task. Upon successful validation, the task will be updated.
- Deleting Task [D]:
  - ○ User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - ○ Under actions/tools, users can click on the cross to open up a confirmation for deleting the specified task. Upon confirmation, the task will be deleted.
- Completing Task [U]:
  - ○ User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - ○ Under actions/tools, users can click on the tick to open up a confirmation for marking the specified task as complete. Upon confirmation, the task will be marked as completed.
- Searching for Tasks:
  - ○ User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - ○ User can choose to search by task name, category or priority via a dropdown beside the searchbar and the relevant results will be shown.
- View by status of Tasks:
  - ○ User navigates to https://project-todo-manager.herokuapp.com/tasks.
  - ○ User can choose to show tasks in-progress, completed tasks or all tasks from the dropdown on the top right hand side of the table.

Additional Use Cases (Admin):
- View All Users [R]:
  - ○ Admin navigates to https://project-todo-manager.herokuapp.com/users.
  - ○ Admins can view all information of users, including their account details (excluding password) and tasks.
- Update User [U]:
  - ○ Admin navigates to https://project-todo-manager.herokuapp.com/users.
  - ○ Admins can edit user account details/tasks (excluding password).
- Delete User [D]:
  - ○ Admin navigates to https://project-todo-manager.herokuapp.com/users.
  - ○ Admins can delete user accounts.

Execution Plan:
- Rails:
  - ○ Setup rails as an API in the backend [Completed].
  - ○ Create user model and task model and ensure CRUD operations are implemented in their controllers [Completed].

- ○ Use Json Web Tokens for authentication and ensure permissions are setup properly [Completed].
  - ○ Add welcome email for new accounts [Completed].
  - ○ Add farewell email for deleted accounts [In-progress].
  - ○ Add validation for user account details [Completed].
  - ○ Add validation for task details [In-progress].
  - ○ Add admin role and account [Completed].
  - ○ Implement forget password [In-progress].
  - ○ Deploy to Heroku [Completed].
- ● React:
  - ○ Setup react as the frontend that will interact with the rails API [Completed].
  - ○ Create signup page, login page, tasks page, profile page and users page and link up all the necessary operations with the rails API [Completed].
  - ○ Add search functionality for users and tasks page [Completed].
  - ○ Create a logo for the project [Completed].
  - ○ Improve the design of the website [In-progress].
  - ○ Add more functionalities - flexible [In-progress].
  - ○ Deploy to Heroku [Completed].

Problems Faced:
- ● Version incompatibility when running rails API on Ubuntu:
  - ○ The stable release of rails on Ubuntu is on 2.5 compared to 2.6 on a Mac so deployment on Ubuntu proved challenging (although possible).
  - ○ Opted to deploy on heroku instead to avoid the issue with needing to maintain SSL certs but heroku does not allow the use of sqlite as the database which leads to the next issue.
- ● Heroku does not support sqlite:
  - ○ While deploying the app on heroku, an error was thrown as heroku does not support sqlite. A lot of time was spent on the conversion to use postgresql due to following a bunch of tutorials at once.
- ● Limited react tutorials for some of the features I wish to implement:
  - ○ A lot of the answers/tutorials for react are done using class components. As I started off with functional components, there was a lot of extra work in adapting solutions to my app.
- ● Welcome emails are more often than not tagged as spam/promotions.
  - ○ In a lot of instances, welcome emails for new accounts are sent to the spam/promotions folder instead of the inbox. There are a few success cases but it does not happen often. Unfortunately, I have yet to find a solution to tackle this.