I thought the CVWO assignment was very interesting because it would lead to creating a working product that could be accessed from the web. This was quite different from all the assignments or schoolwork that I have seen so far and I liked the idea of creating my own website, albeit a very simple one. I had never done web development before and thus, the learning curve was quite steep. However, I think I have come quite far since reading my first Getting Started with Rails tutorial.

I tried to implement an app that uses React for the frontend and Rails as the backend with a RESTful API design. This ended up being one of my main challenges. I found it difficult to integrate React with Rails serving as only the backend. I had to learn how to send requests from the React frontend to the Rails backend via fetch and HTTP requests in order to retrieve information or manipulate data in my databases. At first, I was extremely confused and had to look up many tutorials to even implement simple CRUD functions. However, as I got more used to working with the frontend and backend separated, I found it becoming more and more natural. After a while, I found that I could envision what I wanted to do, write my own code and have it work as expected. I also had to learn how to route from React to Rails. Sometimes, the small bugs were infuriating as I would have extra letters like "s" in either my specified routes or the urls I was trying to fetch from. Working with Rails was another difficulty. I decided to implemented the Search function myself through a has_and_belongs_to_many relation between that Tags and Tasks models. I let the users select tags, sent the selected tags' ids in an array to the backend. In the Tasks controller, I had a search function that would iterate through the tag ids, find each tag, and return all the tasks associated with each tag in the join table. However, I did not realise that the tasks were being returned in an array. So instead of returning an array of tasks to be displayed in the frontend, I was returning an array of arrays. This led to hours of debugging in the inspector and console since the results seemed to be correct sometimes and incorrect other times.

While there is still a lot of room for improvement in my app, I believe I have fulfilled the basic CRUD functionality, Search function by tags, as well as hosting on Heroku. I would have liked to add more styling. However, some parts of the styling don't seem to work on Heroku and only show up on my local server. Given more time, I would have liked to implement user authentication, as well as enable the user to sort the tasks in the list by a certain feature (perhaps due date or by tag). However, I wasted too much time debugging very small errors. I still hope to update and improve my app in the future, since it is my very first webpage. Overall, I am happy that I completed the assignment. I feel that although I am still very inexperienced, I have learnt a lot about development with Ruby on Rails, including the MVC framework, routing, and unique Rails syntax to access tables/models in a many-to-many relationship. I also learnt about RESTful API and how to separate the frontend and backend of an app.

User Manual

The application can be found at: https://njx-cvwo-my-list.herokuapp.com/

Home Page: Shows name of application (To-Do List) and a button labelled View Tasks. On clicking the button, the user is brought to the page that lists all the tasks. This page allows users to add new tasks, add new tags, view, edit or delete existing tasks. The page also includes a Search by Tags button as well as a View All Tags button.

Create New Task: Users can set the title, details and tags of a task. The title will be shown on the list page where all the tags are displayed. Details and Tags can only be seen when clicking View Task. Only Tags that have already been created can be added to a new Task.

Search by Tags: On this page, all the tags created are displayed and users can select each tag by either clicking the tag name or the checkbox beside its name. After clicking the Search button, all tasks that have one or more of the selected tags will be listed on the same page. Clicking the Clear button resets the page so that there will be no tags selected (and no tasks displayed).

View All Tags: On clicking this button, the user is brought to the page listing all the tags. Buttons allow the user to edit or delete existing tags and create a new tag.