

# CVWO Riding on Rails

## Final Writeup

Ryo Hilmawan (A0245871X)

---

I've had a lot of fun with this assignment. It was the perfect chance for me to learn some of the tools I've been constantly hearing about: Rails, React, and TypeScript. I've had experience with manipulating user interfaces using JavaScript, but Rails and React have introduced me to a totally new way to create interactive interfaces. Although, I've had to stray away from the guides provided as I wanted to use the newer version of Rails, which took a while to figure out.

TypeScript was a very interesting variation of JavaScript, although it did give me a lot of headaches at first. I do, however, appreciate that it has helped me in avoiding many errors relating to type mismatches, especially when passing around React props and states.

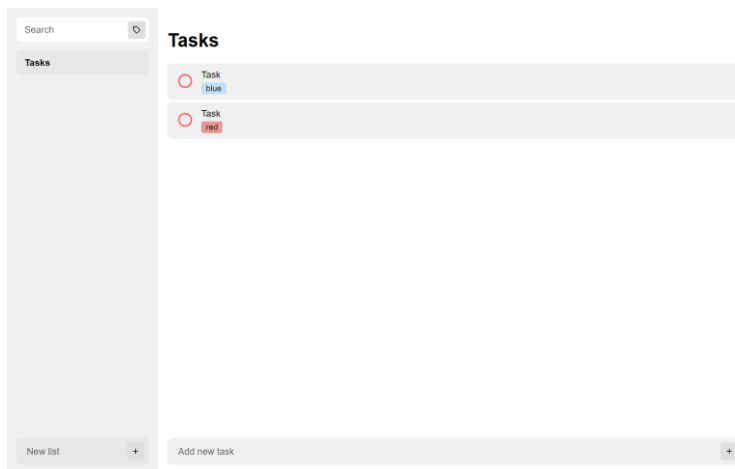
I've tried my best to create an intuitive experience with the time I have - especially among the many obstacles I've had to face, such as attending a relative's funeral and flying to Singapore to live on my own for the first time. There are many little details that I've missed, such as using forms for the inputs and adding context menus for various elements that could make the app more intuitive.

As this is the first time I've used React, Rails, and TypeScript, the code I've written ended up pretty messy and lacks proper commenting. I also didn't have many opportunities to sit down and think things through and I've had to dismiss many of the features I had planned. A couple features I really wanted to add were list groups (which I had originally planned as one of the core features) and user account management. Right now, the task list in the deployed app is shared globally, which makes it pretty much unusable in any real-life scenario.

Still, even though I couldn't implement many of my planned features, I'm very happy with the result. I've mostly been developing the app on-and-off whenever I get the chance to, and it can definitely use some polishing, but I've learned a ton in the process and the app still features a working interface that performs pretty smoothly.

# User Manual

The app interface is split into two major sections: the **navbar** and **tasks**



## Navbar

The **navbar** consists of a search bar, a list of all task lists, and an input bar.

### Search bar

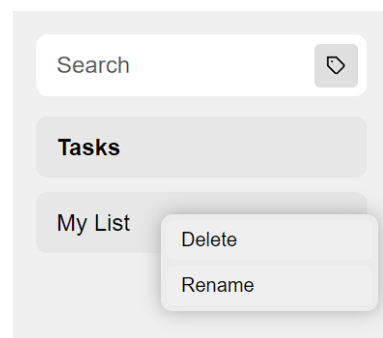
The search bar allows the user to filter all tasks by their title and tags.

### Task Lists

Clicking on a task list will set that list as the current active task list. The current active task list will be bolded.

A context menu can be accessed by right clicking on a task list, which gives the user the option to rename or delete the list.

The default task list “**Tasks**” cannot be deleted or renamed.



### Input bar

The input bar allows the user to create a new task list

## Tasks

The **tasks** section consists of a list of tasks and an input bar.

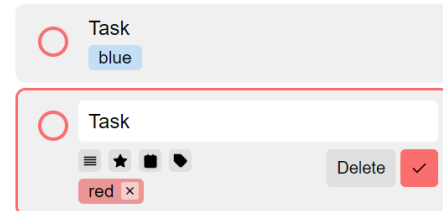
### Task list

The list of tasks will either contain

- all tasks within the current active task list
- all tasks that match the search query (if the search bar is populated)

A task's completion can be toggled by clicking the circle on the left of the task.

## Tasks



A task can be edited by clicking on them, in which its interface will be replaced by the **task edit** interface. Any changes performed by the user will update the task immediately.

### Input bar

The input bar allows the user to add a new task to the currently active task list.

## Tags

Each task can be assigned multiple tags. The user can then filter tasks by their tags through the search bar.

Tags can be created and modified through the **task edit** interface.

Each tag can be assigned a name and color. At this time, a tag cannot be renamed after its creation.

## Tasks

