Chan Kok Jing, Daryl
A0183998B
29 January 2019

CVWO AY18/19 Final Write-up

# 1   Review

## 1.1   Basic Features

The process of building this application was a very enjoyable and interesting one. I was able to use this application as a platform to try out some libraries and tools that I've been wanting to try for a while now such as `react-rails` and Bulma.

Starting with the back-end, Rails wasn't new to me but it's been awhile since I've worked with it. As always, it was pleasure working with Rails and its gems, and really nice to be able to build a back-end with full test coverage. It was also nice to work with Circle CI again for continuous integration.
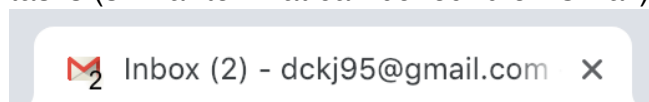
For the front-end, I'm not new to React but it was my first time using the `react-rails` gem. Certain things were different from a standard React setup so it was definitely a good learning experience. I didn't have time to work on writing tests for the components so that's something that I still have to learn in the future. It was also my first time working with Bulma, a flex-based CSS framework. The experience was overall enjoyable and I managed to use some of Bulma's components for parts of the UI. I feel like the overall UI does provide a fuss-free way for the user to access all of the application's features but I will admit that it looks a bit bland – another thing to work on in the future.

Deployment on Heroku also went surprisingly smoothly. Heroku was able to detect the environments that the application required – Ruby and Node, and set them up automatically without any sort of configuration on my part. I also set up automatic deploys on Heroku such that the latest version of the `master` branch would be deployed after all tests on Circle CI pass.

## 1.2   Additional Features

In my mid-assignment write-up, I listed the following as features that I would like to implement if time permitted.

- Inline creation / modification of tasks instead of using a modal
- Split complete and incomplete tasks into separate lists
- Dynamic modification of `<title>` and favicon to reflect number of incomplete tasks (similar to what can be found on Gmail)

I'm glad to say that I had the time to look into every one of these features, and did some exploration into what was the best way to implement them. The final version of the features can be found in the version that has been deployed.
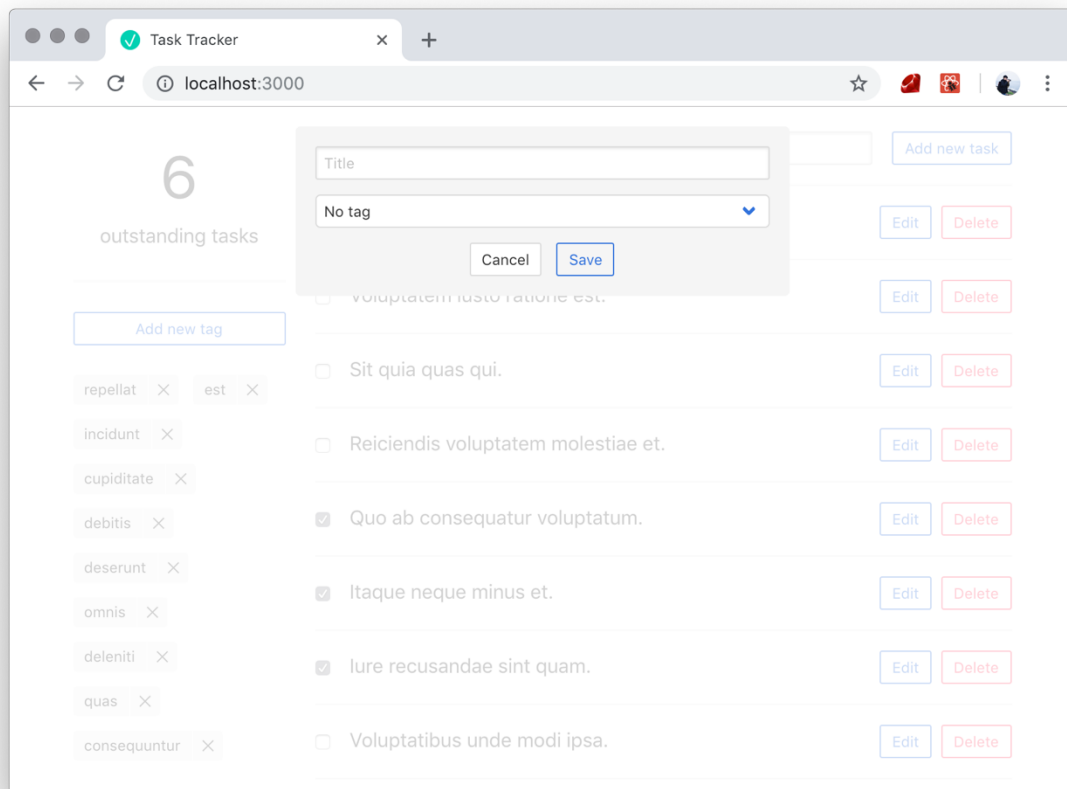


Figure 1: Modal for adding a new task

For the first feature, I replaced the modal found in earlier versions (Figure 1) with an inline form that can be used to both add and edit tasks. This allows for a more seamless user experience especially on mobile devices where modals might be difficult to navigate.

For the second feature, I found splitting the complete and incomplete tasks into separate lists quite confusing for the user, especially when toggling between the states causes the task to "jump" around the page. I solved this by moving completed tasks to the bottom of the list and giving them a lighter appearance, allowing the user to focus on the incomplete tasks.

Lastly, I did some digging around into Gmail's source to see how they implemented the dynamic favicon. I found out that they did this by basically having lots of favicons hosted on their server, each with a different number on it.

https://ssl.gstatic.com/ui/v1/icons/mail/images/2/unreadcountfavicon/0.png
Replace 0 with any number in the following ranges – 0-19, 20+, 30+, … , 100+

From a performance / resources standpoint, this seems like a bad idea. It might make sense for an application on the scale of Gmail but for a standalone app like mine, it definitely did not make sense. As such, I only implemented the dynamic `<title>` tag to reflect the number of incomplete tasks.

## 2   User Guide



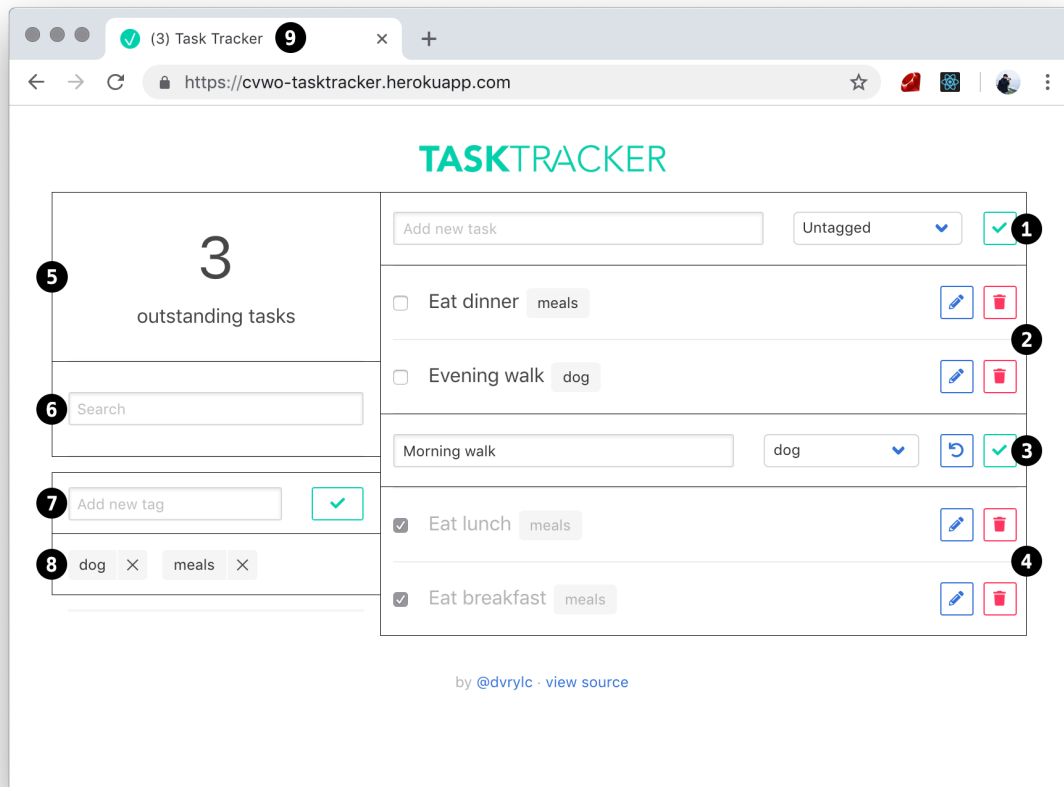Figure 2: Task Tracker

The following list corresponds to the numbered labels in Figure 2. In addition, a full REST API documentation can be found here in the README.

1. Add a new task with an optional tag
2. Incomplete tasks – Tags can be clicked to filter
3. Edit a task
4. Complete tasks – Tags can be clicked to filter
5. Overview with number of incomplete tasks
6. Search for tasks by title
7. Add a new tag
8. Tag listing
   Click on a tag to filter, click on X to delete the tag.
9. Dynamic document title with number of incomplete tasks
   Reminds the user of incomplete tasks even when the user is on another tab.