

Plan vs End Product

The initial plan was to create a unique task management system that can be used optimally with only the keyboard. However, I chose to strive for learning the optional additional tools defined for the assignment (such as learning TypeScript with Redux and using Docker for deployment) instead.

Prior to this project, I had experience with other MVC web frameworks (such as Django and Laravel) and Vuejs. However, web development with Ruby on Rails and Reactjs was a tough experience for me.

Ruby is a very different language compared to the previous languages that I had experience with, so it proved to be slightly challenging to pick up. It also took me more than two weeks to learn Redux and Typescript as there is a lack of understandable Redux tutorials implemented in Typescript, and I had to resort to mostly trial and error.

Due to the long learning period, I had limited time to even implement the basic functionality of the task management system. Therefore, I had to forgo the extra features such as the 'command line' interface, the ability to assign users to tasks as a tradeoff due to time constraints. I also did not write unit tests as I am not confident of doing so.

Despite the shortcomings, I still feel satisfied about the submitted product, though I aim to improve it further in the future (after the end recess week as mentioned in the chat).

Quick Project Rundown

Deployment instructions are mentioned within the README files of the respective repos. Note that Docker deployment is planned as one of the improvements I aim to make in the future.

Demo is available at <https://cvwo.ayzh.website>. Both the frontend and the backend are deployed on the same DigitalOcean droplet, with an nginx server acting as an SSL proxy and a reverse proxy to differentiate requests made to <https://cvwo.ayzh.website> and <https://cvwo-api.ayzh.website>. As the website is hosted in the public domain, and this document is also in a public repository, credentials will not be provided here, but instead will only be given upon contacting @polygonal (zhiheng) on Telegram.

Short user Guide

After the user has logged in, the user will be able to view all the tasks that are in the database. To filter tasks by tags, the user can click on any of the tag buttons beside the 'Filter by Tag' text at the top left.

To create a new task, the user may click on the top left button as pictured at the side in Diagram 1. The user can then fill up the form and select the appropriate tags before clicking on the 'ADD TASK' button to submit the task.



Filter by Tag:

Diagram 1

To view or manage tags, the user may click on the button beside the new task button as seen at the side in diagram 2 to bring up the tag menu. In the menu, the user can view and add new tags as a user. The user must be an admin to delete tags.



Filter by Tag:

Diagram 2

As an admin user, it is possible to create new users. This menu is within the accounts menu which is accessed by the button at the top right as pictured in Diagram 3.

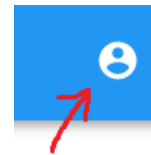


Diagram 3

As any logged in user, it is possible to logout with the same menu as well.

When clicking on any task, any user can bring up the modal to view the details of a task as seen in Diagram 4 below.

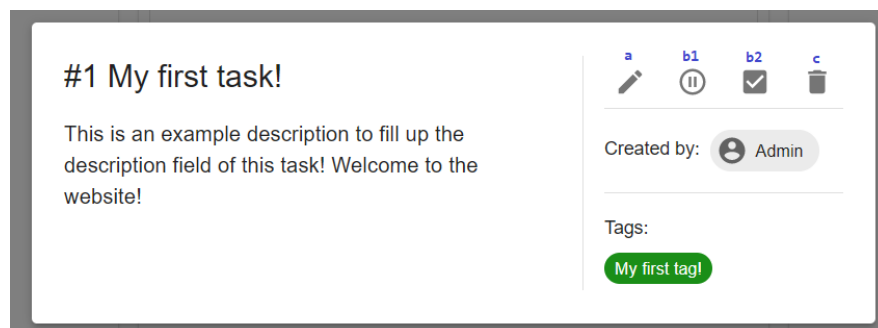


Diagram 4

Buttons labelled 'a', 'b1', and 'b2' will only be seen by the user who created the task. The button labelled 'a' allows you to edit the task. On the other hand, buttons 'b1' and 'b2' varies depending on the status of the selected task and will change the current status of the selected task.

Button 'c' can only be viewed by admin users, which allows an administrator to delete a task.