

What I feel about my accomplishments

Having learnt programming from CS1101S last semester, this assignment was my first attempt at applying my knowledge in a practical setting. I am rather contented that I was able to build a functional to-do list web application in this span of time despite my lack of experience. I am also glad that I embarked on this project as it taught me not only about web development, but also vital skills and approaches for future coding projects.

I have developed a better understanding of the various aspects of web development and learnt to use relevant libraries such as ReactJS and Redux, as well as the Ruby on Rails framework. I found the frontend the most intuitive part of the process and spent most of my time focusing on improving it. While I am not entirely certain I have implemented it in the best way possible, I found Redux useful to ensure I followed the DRY principle of software engineering. Typescript also helped ensure I was passing the right output to different functions in more complicated components, so I decided to implement it.

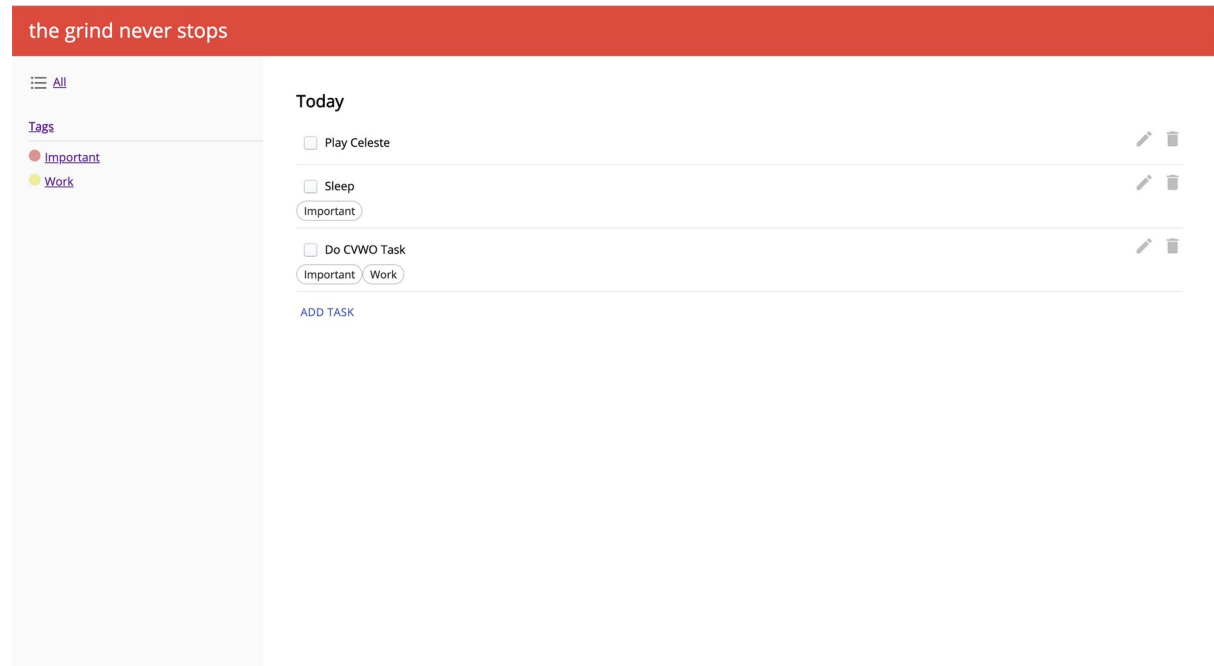
One thing I made a conscious effort to focus on for this project was coding the 'right' way, which meant keeping code simple and neat, minimizing repetition (in accordance to the DRY principle) and working with the frameworks and libraries instead of against them. At times, this felt extremely frustrating and tedious, but I hope that this has minimized bugs and problems in my code.

After this valuable experience, I have realised how much there is to web development. There are several aspects that I did not delve into for this project, such as security and speed, and I hope to learn more about these aspects in future projects.

One thing I regret about this project is playing it too safe. My original plan was to build an API-only application using Rails, with a React frontend. However, after completing the frontend, I did not feel confident enough to finish the project in time following the original plan, given my lack of knowledge especially when it came to asynchronous programming. Hence, I decided to redo the application using react-rails, which burnt a lot of time, preventing me from adding in additional functionality such as date pickers and a login system.

Nonetheless, I found this an extremely enriching experience, and hope to continue learning more about web development through future projects.

User Manual



Sidebar

To see all tasks, click "All"

To see all tags, click "Tags"

Selecting the tags listed filters the tasks in the main component to tasks containing said tag

Tasks

To mark task as completed:

Click the checkbox beside the task

To add tasks:

Click the "Add Task" button, which opens a dialog containing a form. Fill in the Task name (compulsory) and the tags (optional), and click "Confirm"

To edit tasks:

Click the edit icon on the right of the corresponding task (the pen), edit the respective fields and click "Confirm"

To delete tasks:

Click the delete icon on the right of the corresponding task (the bin), and click confirm on the popup that appears

Tags

CVWO Final Submission

To add tags:

Click the “Add Tag” button, which opens a dialog containing a form. Fill in the Tag name and color (both compulsory), and click “Confirm”

To edit tags:

Click the edit icon on the right of the corresponding tag (the pen), edit the respective fields and click “Confirm”

To delete tags:

Click the delete icon on the right of the corresponding tags (the bin), and click confirm on the popup that appears