Key Accomplishments

Using React Query

React Query was immensely useful in fetching data efficiently from the backend and making changes to the backend. In addition, it also provided an easy way to implement notifications to inform the user the current state of their actions, the outcomes and the possible additional actions to be taken.

Working on this assignment gave me the opportunity to explore the various features of this package in-depth and gave me an alternative to using Redux to manage data in the frontend.

Building a backend using Go

Having the option to build the backend with Go gave me the chance to pick up its syntax and explore the various packages developed within its ecosystem that I could use to implement backend-related functions such as parsing JSON data, interfacing with PostgreSQL and returning sensible error messages.

Learning basic SQL commands

Prior to this assignment, the only experience I had with databases was through Django, which abstracted away the complexity of the management of the underlying database. However, a lot of times the functions felt like magic and it was not exactly clear what actually happened.

In this assignment, I took some time to learn some basic SQL commands and wrote raw SQL commands in my Go backend to query/modify the database, which gave me a clearer picture of what was happening.

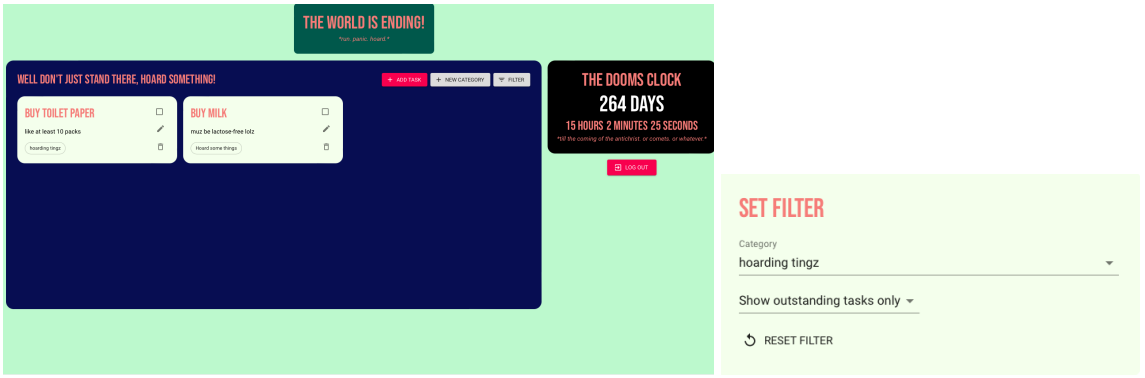
Using TypeScript

This assignment served as sufficient motivation for me to learn how to use TypeScript to write better frontend applications.

Having mainly used JavaScript during Orbital, I experienced first-hand the frustration that came with not being able to catch errors during compile-time, which cost my partner and I many hours as we went around debugging code. Hence, this time around I decided to start using TypeScript from the outset and learn how to use it as I go along.

User Manual

Home Page



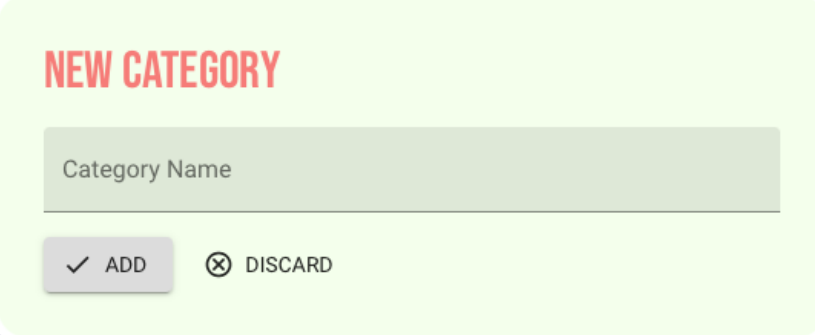
After logging in, users will be brought to the home page, which allows them to view and edit existing tasks, and also add new ones.

Users can also choose to see only tasks that belong to a certain category, outstanding/completed tasks, or some combination of the two criteria by setting the filter via the dialog-box that appears after clicking on the “Filter” button.

Adding a new Task

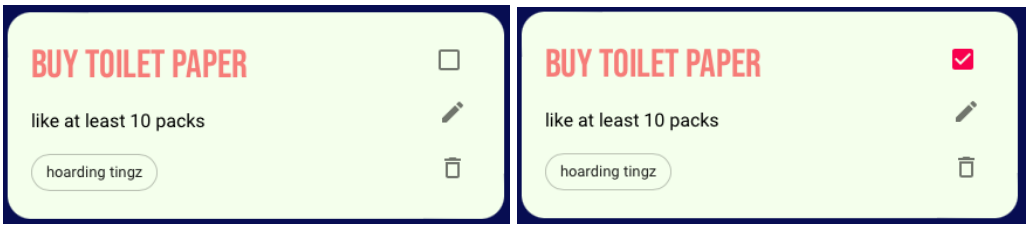
To add a new task, users can click on the “Add Task” button coloured in red, after which a dialog box will appear. When creating a new task, the user can specify additional details about it, as well as tag it with an existing category, which would come in handy for organisation (more on this in the Filter section).

Adding a new Category



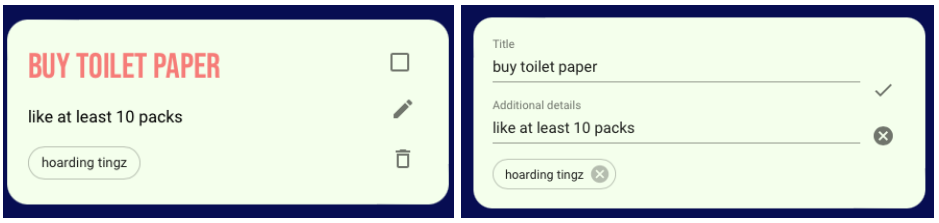
If users wish to tag the task to a new category, they can create one by clicking on the “New Category” button on the home page, after which the new category would be available for tagging (see immediate section).

Marking Tasks as Done/Incomplete



Users can mark tasks as complete or undo previously-completed tasks by clicking on the checkbox.

Editing and Deleting a Task



To edit a task, the user can click on the pencil icon, which would alter its appearance and allow for changes to its title, details and tagged category to be made. After making the changes, users can confirm or discard them by clicking on the check and cross buttons respectively.

To delete a task, the user can click on the bin icon, after which the task would be permanently-deleted.