Han Geng Ning
A0222055U

My Accomplishments in this Assignment

This assignment increased my ability to pick up programming languages and frameworks. I learnt how to quickly identify the main features of a framework and start applying them in small ways before moving on to more complex cases. For example, basic understanding of state, props and how components render and connect to each other was sufficient to start making the UI of an app. Likewise in Ruby on Rails, basic understanding of the MVC framework could make a simple working app. I feel that this accomplishment significantly lowered my barrier to learning new technology by showing me how to gain an idea of the big picture behind a tool and how to proceed to use it.

It also increased my ability to debug. First, I learnt how to read error messages and search for the root of the problem. Next, I gained an appreciation for writing small blocks of code to achieve functionality bit by bit, so that it was easier to ensure all parts were working. I also learnt TypeScript which reminded me to be more prudent in writing code as it checked for errors before and during compilation. I think it helped me gain better programming practices such as checking the types of my variables and ensuring that I used exactly what was needed, nothing more or less. There are still more techniques to learn when it comes to handling errors and throwing exceptions that I did not manage to cover in this project and hope to gain insight on them later on.

The assignment also challenged me to think logically about the flow of events. One thing to consider was when to re-render components after the state is updated. For, example, if I should re-render all to-dos if it was marked as completed or if its view should change immediately while the request was still being processed. I also had to consider small details like adjusting the end date to match the start date. If I displayed alerts, such as an alert to confirm their deletion, I had to consider how user's responses to the alert would give different results. While I managed to implement most functionality, I could give more thought to reducing computational resources when retrieving or finding data.

It was fulfilling to be able to develop a systematic approach in creating a working application. At first, I did not even know how to integrate React and Rails frameworks. However, I realized it was helpful to develop the backend and set up the database first. Then, figure out what data I needed on the front end and write the controller methods to provide the relevant data. On the front end, I set up the UI to display and capture data before writing the methods to pass data from the front to back end. This experience helped me learn to be systematic when embarking on future projects.

I learnt a little about considering the scale of a project. When I started learning, some tutorials integrated React within Rails and embedded JavaScript in Rails views as

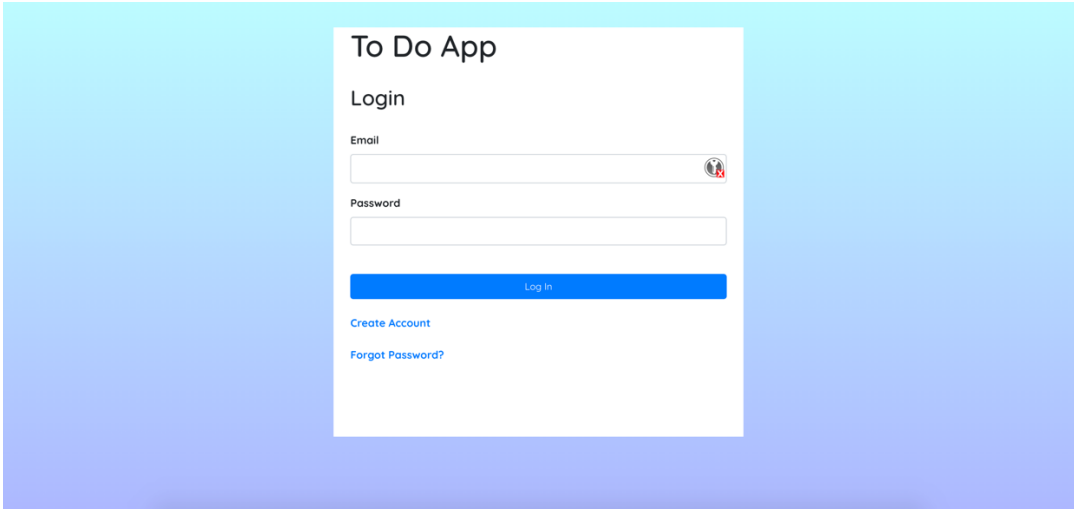
one project while others created a separate React project. I learned how having a separate front end would be easier to scale and increase clarity. When I separated the front and backend, I realized that user authentication became a problem and required passing of tokens in HTTP requests, so I had to find new ways to achieve that. As I considered the scale and learnt more about good coding practices, I had to refactor code several times. This helped me develop a patience for improving my code and to be more willing to achieve functionality in better ways.

Overall, it was a project that introduced me to many new concepts and helped me apply new skills. It was fun and fulfilling and made me interested to try new things.

User Manual

The web app url is https://zognin-todoapp.herokuapp.com/
If the backend server is inactive, it takes some time for the backend server to start when the first http request is made

Log In

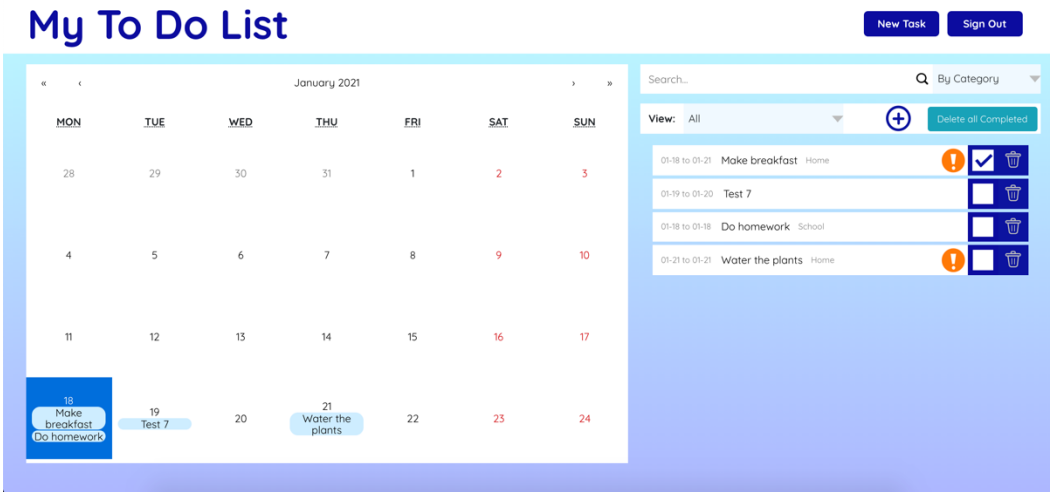


Users will see the log in page once they go to the above url.
They can create an account by clicking 'Create Account' below the log in button.
Unfortunately, I am unable to send the email for 'Forgot Password', although the process works in the console.

After creating an account, users are redirected to the log in page. When they log in, they arrive at the home page.

Home Page

Han Geng Ning
A0222055U



Add a new to-do item
To add a new to-do item, click 'New Task' or the blue plus button
For each to-do item, users can specify the task, description, category, start datetime, end datetime, whether it is a priority and whether it is completed.

Edit a to-do item
There are 2 ways to edit a to-do item.
Firstly, users can click the to-do item in the list to edit it.
Secondly, users can click the to-do item on the calendar to edit it.
Users click the checkbox to mark a to-do item as completed from the list.
Users click the trash icon to delete items from the list.

Search
Users search for to-dos by typing in the search bar. They can search by Category/ Task/ Priority/ Completed/ Start Date/ End Date

View
Users can restrict their view of to-dos to a Category or in terms of Priority/ Complete/ Incomplete/ Start Date Order/ End Date Order by choosing a field in the dropdown

Calendar
Users can see their to-dos on the calendar
They can click the to-do on the calendar to edit it

Delete all completed
Users can click the green 'Delete all Completed' button to delete all completed items in the list

Sign out
Users click the blue 'Sign out' button to sign out