

User Manual

Sign Up

For new users, you can register an account with your email and password. Each email will be tied to an account. Upon, signing up you will be redirected to the homepage.

Login & Logout

For users that has been registered, you can login with your email and password. You can logout using the logout button on the top right of the navigation bar.

Add Todo

A new todo can be created by entering a title and clicking the Add Item button.

Edit Todo

A todo item can be edited by clicking on the green pencil icon. A popup will be displayed and after editing of the title, user can press the save button.

Deleting Todo

A existing todo item can be deleted from the todo list by clicking the trash icon.

Categorizing Todo

A new todo item will automatically be inserted into the To do column. Users are able to drag and drop the selected todo item into other columns [In Progress, Done]. This serves as a filter for users to segregate their items.

Learning Points

Ruby on Rails with React & Redux

It was difficult in the beginning to incorporate react with rails. After countless number of YouTube tutorials, I managed to start the server with react as the frontend. Implementing redux with react was a huge challenge and confusing. However, after reviewing videos on the web I managed to implement it. I also learnt the usefulness of using redux instead of the state provided by React even though my application is small.

Security & Authentication

Initially, I was storing plain text for the password in the database after registration. However, after reading on the best practices around the web, I implemented Bcrypt to hash the user password before storing in the database to protect the security of the user. To manage the user state, I implemented the session authentication provided in the rails.

Third Party Libraries

In my application, I utilized a numerous third party packages such as react-beautiful-dnd where user can drag and drop items. This helps to serve as a category for users to drag and drop across. I had to learn how to utilize this library and persist the data across my components and in the database which was a challenge.

Improvements

Modularity & Clean Code

As a newbie learning rails with react, it is very hard to write good and clean code for the functionality as I am still in the learning phase. Furthermore, I believe that my current code can be better abstracted into smaller modules.

Test Cases

I can write better code by writing test cases to test my functionality so that I do not have to test my application manually.