

Final Assignment Submission

Details:

Name: Ezekiel Toh Fun Kai

Matriculation No: A0218061N

Git-hub page: <https://github.com/takufunkai/todo-application>

Heroku (Client): <https://just-another-todo-app.herokuapp.com/>

Heroku (API): <https://mysterious-gorge-55099.herokuapp.com/>

⇒ /api/v1/todos to view todos

⇒ /logged_in to view user

Accomplishments

1. Overcome most hurdles

Along the journey I faced endless hurdles, and when faced with all of them I always felt like giving up. I always thought that it was impossible to do, or that I would be fine settling for less. However, every single one of them I told myself to give my best, and eventually succeeded in overcoming almost all of the times I wanted to give up (except Typescript. What???)

2. Knowing myself better

There were days where I spent an entire week non-stop working on the project, from the moment I woke up till when I slept. It made me realize that actually I could accomplish much more if I knew how to pace and not obsess over the work.

3. Learning how to tinker with my code more

At the beginning, I spent a lot of time following guides word-by-word, but towards the middle and the end I spent a lot of time looking up the official documentation of any particular feature I was looking to implement, such as Hooks and Redux. One thing I realized is that having the ability to work interactively with my code paved a lot of ground for improvement. As long as I took the time to digest and understand the content, and played around with the code, there was no need for any guide or tutorial.

User Manual

- Login (user management)
 1. New users have to register with their email and their password (length ≥ 6) [sends a POST request to the API]
 2. Upon registration, users will be automatically logged in to view their clean inbox [sends a POST request to the API]
 3. Existing users will have to login to view their inbox [sends a POST request to the API]
- Overview
 1. At the top, their email address
 2. In the middle, the form area
 3. Below the form area, their tasks lists
- Creating a new task
 1. A task has 3 fields: [sends a POST request to the API]
 1. Task name (required)
 2. Tag
 3. Deadline (required)
- Viewing tasks
 1. Tags
 1. Upon creating a new task with the desired tag, the new tag will automatically be added to the current list of tags. [Stored in the todosSlice]
 2. Users will be able to view their tasks by tags, or as a whole (disregards tags) [Filtered through the todosSlice]
 2. Deadline
 1. Users are able to sort their tasks by their assigned deadlines by ascending (most urgent) or by descending (least urgent) [Filtered through the todosSlice]
 3. Priority

1. Users are able to sort their tasks by priority flags (deadline sort will still be retained within priority sort) [sends a PATCH request to the API, filtered through the todosSlice]
4. Search
 1. Users are able to search through their tasks' title. [Filtered through the todosSlice]
5. Completion Status
 1. Users are presented with 2 lists: a completed list, and an incomplete list.
 2. Incomplete/current tasks are situated at the top of the container.
 3. Completed tasks can be hidden. It is situated at the bottom. Completed tasks have an active checkbox beside their task name and a line-through text decoration.
 4. Users are also able to see how many tasks they have pending and completed. It will adjust as they change the tags.
- Handling tasks
 1. Users can use the checkbox to label their tasks as complete. [sends a POST request to the API]
 2. Users can edit their tasks on the right side of their tasks. [sends a PATCH request to the API]
 3. Users can delete their tasks. [sends a PATCH request to the API]
 4. Users are able to flag their tasks as priority [sends a PATCH request to the API]
 - Theme
 1. At the header, the theme can be switched between light or dark mode by using the switch [handled in the todosSlice]
 - Logout
 1. Users are able to logout using the logout button at the top. [sends a DESTROY request to the API]