CVWO Final Submission
Name: Ong Yin Ming Jonas
Matriculation Number: A0155237E

Achievements

In this assignment, I have developed a single page web application using React and Rails. This is with the aid of the front end styling framework Material-UI.

I gained a better appreciation for how Redux works and attempted to use React Hooks in place of lifecycle methods.

I have put in place a workable database schema that suites how I plan to approach the design of the application. In the process I have expanded on what I know about databases and learnt about the flexibility of using migrations which is a novel concept to me.

I have gained a better appreciation on how Rails helps facilitate rapid development of an application such as generating routes, database models and controllers.

In the process of attempting to deploy, I have learnt how to use Docker to prevent my application from being dependent on packages on my development machine and why it is important to do so. However I still faced troubles and could not deploy my React application with Rails in production onto Heroku but the process was an invaluable learning experience.

I have also realized how little I know about certain things regarding web development such as stylings, the different ways of structuring webapplications and handling HTTP requests, webpacker, how different production environment is from development environment and seek to learn more of it in the future.

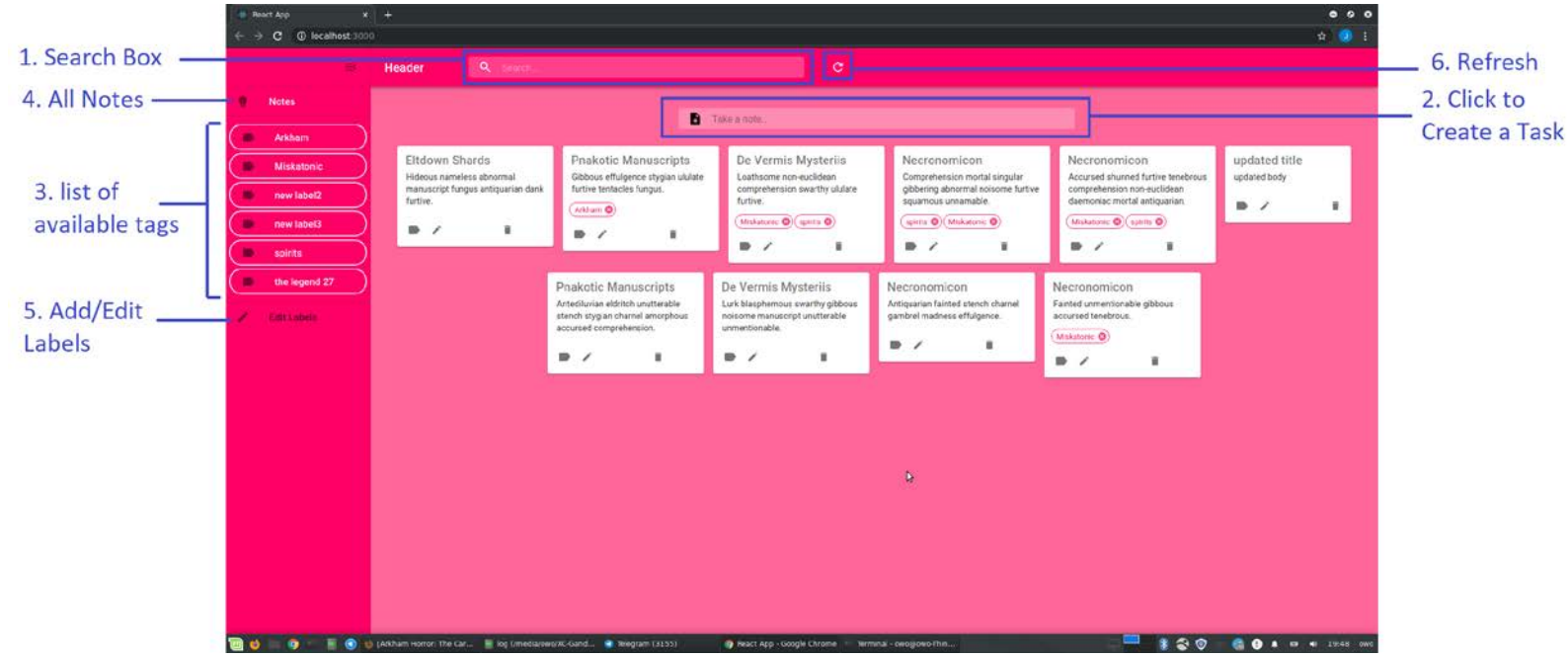
User Manual

Managing Tasks: To keep track of all the tasks I have to do, my todo manager allows creating (2)[1], updating (13), deleting (12) of tasks in the form of Notes[2].

Categorizing Tasks: To organize tasks, we can tag Notes with Labels (10). This Labels allows you to quickly search all Notes by the tags they are tagged with (3). A list of Labels (3) available are tracked by the ToDo Manager. To tag a Note, you first have to create it (5) and then tag the Note with the Label (10).

[1] These numbers refer to numbered sections in the diagrams on the following pages. They are further described in detail there. E.g (2) refers to "2. Add Note" paragraph on the next page.
[2] Refer to diagram "Anatomy of a Note" for how a task is displayed as a Note and how tags used to categorize and search for Notes are displayed as Labels

As we are all busy people, such a process ensures that all Labels are tracked for you and does not require you, the user, to keep track of what tags exist.



1. Search for notes:
   Enter text to set to filter for notes that contain the provided text in title or content. Clear this field to display all notes.

2. Add note:
   Click this bar to be provided with a form to enter in details for a new task, content is a compulsory field. Clicking “close” on the form will cause it to add a new note if valid.

3. Filter by Labels:
   all labels you can use to tag a task/note can be found here. Click on the respective label to display only notes tagged with the label

4. Show all Notes:
   click to display all notes regardless of tags, note may not display all notes if search bar (1) is not empty

5. Add/Edit/Delete Label:
   Before you tag a note you need to add a label first, click this to add a new label or rename an existing label (all notes tagged with the old name will be updated accordingly). Deleting a label would delete the specified label from all notes.

6. Refresh application:
   Forces application to sync Notes and Labels with backend

Anatomy of a Note/Task

7. Title of Note: A note/task may have a title, this field is optional

8. Content of Note: A note must have a content field, clicking this will allow you to edit the note. Closing the form applies the edit.

9. Labels the Note is tagged with: This note is tagged with the respective labels, click to filter all notes by that label

10. Tag a Note with a Label: To facilitate organization of notes/tasks, click this to tag the note with an existing label. If no label you want exists, add a label with (5)

11. Untag Label from Note: To remove a Label from the Note, click the X, Label will still exist for other tagged notes, to delete a label from all notes do it in (5)

12. Delete Note: Note will be deleted with no further confirmation

13. Edit Note: edit the note, closing the form applies the edit.

Proof of Working Application

1. Requires: git, docker and docker-compose to be installed
2. `$ git clone https://github.com/e0031374/rails_taskassignment.git`
3. In rails_taskassignment directory run: `$ sudo docker-compose up`
4. Wait for containers to finish building and startup
5. Goto localhost:3000 and wait for application to load.
6. Comment out `rails db:seed` in `/backend/bin/entry` before running step 3 on subsequent startup. Application cannot seed database twice.

Database: `/backend/db/seeds.rb` file seeds the database on startup with dummy data. Do remember to follow step 6 for subsequent use. While database dump: `pg_database_dump` is provided, it is not required to be used.

Troubleshooting

1. If prompted, you may need to add a folder (directory) `/backend/tmp/pids`
2. If step 3 does not work try `$ sudo docker-compose up --build`
3. If there are errors regarding duplicate entry, in project root directory delete folder `tmp` to delete all postgres data ($ sudo rm –r tmp) from last session