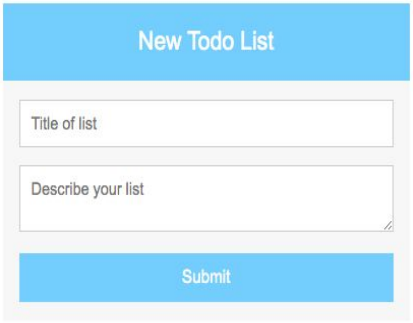
User Manual:

Root page is the Index Page for all Todo Lists:

The index page will show a list of Todo Lists that have been created.

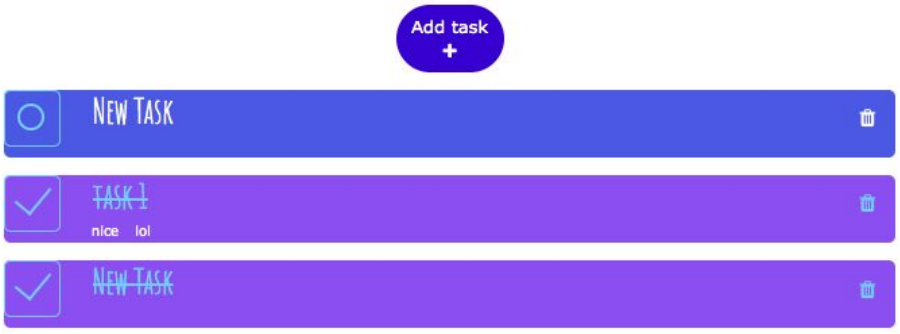
You can add a new Todo List by clicking the button show on the left, which will bring you to the page shown on the right. After filling in the details of the todo list, click the submit button which will bring you back to the main page consisting of all todo lists(if it doesn't, click the 'home' button at the top). You will see the new list at the top of the page.



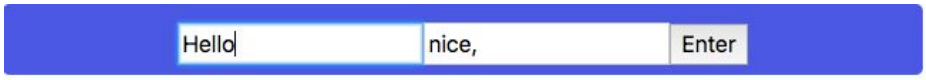
Editing/Viewing/Deleting Todo List:

For each todo list, there will be 3 buttons below the title and description. Clicking View will bring you to the display page of the todo list. Edit will bring you to the form shown in the pic above, containing the details of the todo list, which you can then edit. Delete will remove the todo list from the page. This action is irrevocable.

Show page for Todo List, consisting of respective todo items:



Click the button Add Task to add a task to the list, which you will see at the top of the list. If the page is refreshed, the tasks are ordered in terms of date added.



You can edit todo items by directly

clicking on the 'New Task'(if new) or text of the todo items which will show you a box like the one above. You can edit the content or tags of the todo item and once done, click the button, 'enter'. The todo list will then be refreshed with the updated information. *Note: You can only edit incomplete tasks(to prevent completed tasks from getting edited by accident)*

You can add tags to the todo item by editing the todo item. *Note: To add tags, you must separate each tag with a ',' and all characters of the tags will be updated to lowercase for ease of searching tags.*

An incomplete task will be placed in a blue box while a completed task will be placed in a purple box, with its text having the property 'line-through'. You can change the state of the task as complete or incomplete by clicking on the checkbox on the left-most side of the todo box.

You can delete tasks by clicking on the trashcan icon on the right side of the todo box.

You can delete tags by removing the name of tag when editing the todo item.

Searching for todo items with tags:

You can search for todo items with their tags by clicking on the search button found in the middle of the navigation bar(shown on the left) which will open up to show the one on the right. Type the name of the tag that you would like to search and press enter. *Note: clicking cross will only close the search bar.*

If the tag exists, you would see the page shown on the left. You can view particular todo items by clicking on 'View Todo List' button which will redirect you to the todo list page containing that particular todo item.

You would find a home button on the top right of the navigation bar. Clicking it will bring you to the root page which is the index page for all todo lists.

Reflections:

These past two months were the most exhausting yet most rewarding months for me. I can't believe how this app used to have a simple 'Yay, you are on rails!' page just over a month ago. I can still remember the times I was so frustrated over the bugs I found in my code and those times when I squealed in joy when I finally got something to work or discovered how to solve the bug. Overall, it was a great learning experience for me.

Areas I could improve on/Problems I faced and how I overcame them:

- ReactJS. Should have spent more time learning react and should have started on front-end first before working on back-end. I had a hard time trying to connect both together as the front end had to get data from the backend before it can display the todo items on the page. I had created a separate html.erb front-end for the backend initially, which made things even more complicated for me when I had to integrate react.
- Should have used Material-UI or Semantic-UI instead of hardcoding everything in CSS.
- Should have used AJAX instead of Axios. I had a hard time trying to patch and post data to the api and the response returns some of the object attributes as undefined, which caused a lot of problems on the front end as most of my JS functions were expecting arrays/objects not undefined. The data only got updated when the page was refreshed. I had to work around that by updating the current state manually so that new todo items can be created without having to refresh the page and likewise for completed todo items.
- Heroku deployment: Spent three days trying to figure out why my slug size was too huge and hence the app couldn't be deployed to Heroku. Added a clean-webpack-plugin to speed up the process and managed to deploy to Heroku.
- Should have used React-Router. I didn't want to spend too much time learning it so I avoided it all together, but thanks to that, I had to manually add "a href" links to buttons to navigate around the app which made some of my code unnecessarily long.