Riding on Rails - Mid-Assignment Write-Up

Computing for Voluntary Welfare Organisations (CVWO) AY2020/21

Elumalai Oviya Dharshini
A0219695M

Problem Statement

To build a todo list web application with React and Ruby on Rails. Look into progressive web applications and responsive design if possible.

Use Cases with Descriptions

The primary use cases will be to create, read, update and destroy tasks.

User <u>adds</u> task to todo list

Tentative action flow:

1. User enters task in an empty textbox
2. (optional) User clicks on the 'tag' button on the right of the textbox and adds a tag
3. User either presses enter, presses an 'add' button or clicks elsewhere on the page
4. Task is added to todo list

User <u>edits</u> task

Tentative action flow:

1. User hovers over a task or clicks on the 'edit' icon on the right side of the textbox
2. User presses enter
3. Task is updated

User <u>removes</u> task from todo list

Tentative action flow:

1. User hovers over a task
2. User clicks on the 'trash' icon on the right of the textbox
3. User presses enter
4. Task is deleted

User <u>selects all</u> tasks

Tentative action flow:

1. User clicks on the 'hand' icon on the top left side of the todo list
2. All tasks are selected
3. User may deselect tasks by reclicking on individual 'hand' icons on the left side of individual tasks

User <u>marks</u> task <u>as complete</u>

Tentative action flow:

1. User clicks on the checkbox on the left side of the todo list
2. Task is marked as completed - greyed out and moved to the bottom of the list
3. User may un-complete tasks by unchecking checkboxes so long as tasks have yet to be deleted

User <u>adds a tag</u> to a task

Tentative action flow:

1. User clicks on the 'tag' button on the right of the task and enters a label
2. User presses enter

3. Tag is added to task and a new folder with tag name is added to the side navigation menu if it is a new tag. User may click on this later to view tasks with specific tags

User removes a tag from a task

Tentative action flow:

1. User clicks on the ‘x’ icon belonging to the specific tag at the bottom of the task
2. Tag is deleted from task. If user is browsing inside a folder, the page redirects to the complete list view

User searches for task(s)

Tentative action flow:

1. User clicks on the search bar on the main page and enters a keyword or a tag
2. Matching tasks are displayed

User filters tasks

Tentative action flow:

1. User clicks on the filter button on the right side of the search bar
2. User selects appropriate filters - tags (click once to include, twice to exclude, thrice to revert back to normal), date (“from <date1> to <date2>” where either field can be left absent). User can click “clear filter” to reset filter to default.
3. Matching tasks are displayed

User adds deadline to task

Tentative action flow:

1. User clicks on the ‘tag’ button on the right of the task and chooses the ‘date’ tag
2. User chooses a date and presses enter or clicks away
3. Date is added to task as a tag (note: this does not result in the generation of a folder)

User adds importance level to task

Tentative action flow:

1. User clicks on the ‘tag’ button on the right of the task and chooses the ‘importance’ tag
2. User chooses a level - indicated by colored icons (red, yellow etc.) and presses enter or clicks away
3. Importance level is added to task as a tag (note: this does not result in the generation of a folder) and task gets color-coded to reflect this status

Quality-of-life features that I am looking into implementing:

User switches to Dark mode

Tentative action flow:

1. User toggles the “theme” option on the top right of the page
2. Web application changes colours to reflect a darker theme

User enables push notifications

Tentative action flow:

1. User toggles the “enable notifications” option on the top right of the page.
2. User will receive push notifications when this option is enabled

Push notifications will be served by using Service Workers on Rails.

User switches to Calendar view

Tentative action flow:

3. User toggles the “view” option on the top right of the page.
4. User will view tasks (with deadlines) in a calendar

Possible integration with google calender

User adds ‘in progress’ status to task

Tentative action flow:

1. User clicks on the greyed out ‘in progress’ tag at the bottom of the task
2. Task gets labelled with ‘in progress’

This is essential for kanban board view

User switches to Kanban Board view

Tentative action flow:

1. User toggles the “view” option on the top right of the page
2. User will view tasks (with deadlines) in a Kanban Board format

Other features/tasks I am looking into:

- Hosting the app on Heroku
- Email notifications - on a task by task basis
- Sign in with email/facebook so as to sync todo list across devices. This
- Docker
- Sharing features

Execution Plan

I will be attempting to build a working application in an iterative way, first with basic CRUD functionality, then with added quality of life features and other tasks, with UI improvements along the way with Bootstrap etc.

1. Basic CRUD functionality + set up database + dark mode (by 02/01/2021)
2. All of the above + push notifications (tentatively, 08/01/2021)
3. All of the above + kanban board view + calendar view + more features with more user testing
4. Deployment to Heroku + more user testing