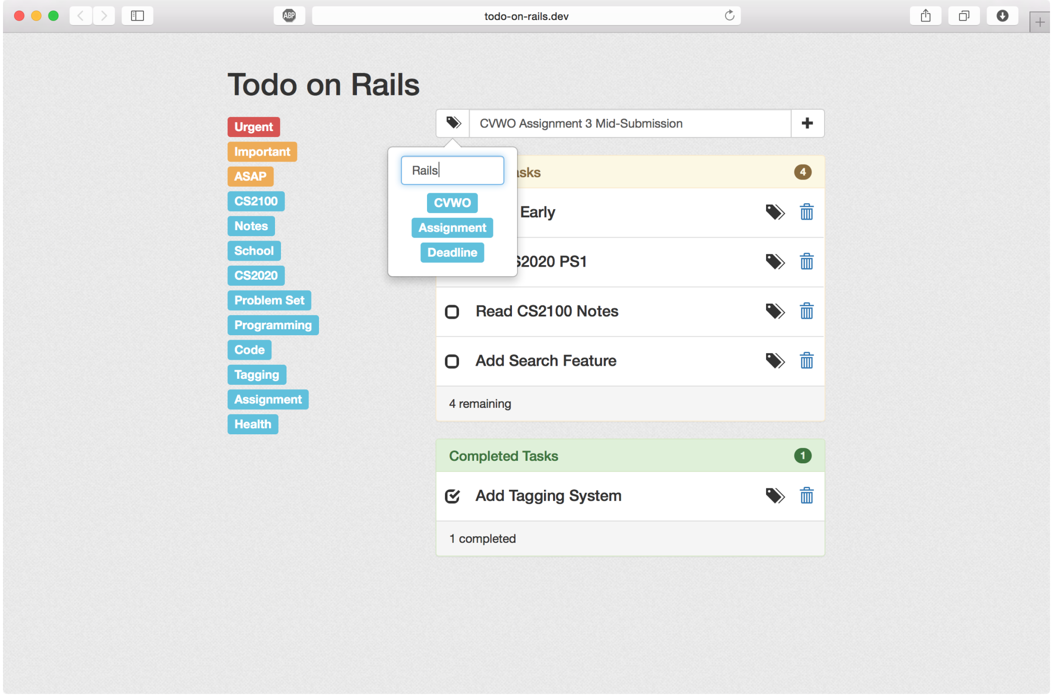
Todo on Rails

Liu Xinan

January 28, 2015

Background

This is a single page todo app made using Ruby on Rails.



todo-on-rails.dev

Todo on Rails

CVWO Assignment 3 Mid-Submission

Rails

sks 4

Early

$2020 PS1

Read CS2100 Notes

Add Search Feature

4 remaining

Completed Tasks 1

Add Tagging System

1 completed

Figure 1: Todo on Rails

1 Basic Use Case

This todo app is very easy to use. Just like another other todo apps, users can easily `Add`, `Edit`, `Remove` tasks, as well as `Mark as done` and `Mark as undone`. As shown in Figure 1, the graphical user interface is very intuitive.

Besides that, this todo app also make use of `ajax` to provide a seamless user experience. Every action happens in a single page. Whenever a action is performed, the page content updates itself without the need to reload the whole page.

1.1 Add new task

Tasks can be easily added by just entering it into the `Create new task` text field and press `Enter`. If you want to attach some tags to the task that you are creating, just click on the `Tag Button` at the left of the text field and enter them, one by one. Tags added will appear below the tag input field instantly. When you are done, just click on the `Plus Button` or switch focus to the `Create new task` text field and press `Enter`. If the request to the server is successful, the task will appear in the pending tasks instantly and the pending task counter will increment by one automatically.

1.2 Edit a todo

Click on the title of a task will turn the region into a input text field. User can edit the title of the task, and press `Enter` to save changes. Press `ESC` to discard changes.

1.3 Mark as done

Clicking on the checkbox on the left of a task marks it as done. Upon successful communication with the server, the task will automatically goes down to the list of completed tasks, and both pending counter and completed counter updates accordingly.
Similarly, clicking on the checked checkbox of a completed task will move it up to the pending tasks list and update counters accordingly.

1.4 Remove a task

Click on the trash icon on the right of a task will remove the task. A confirm popup appears before executing the action.

1.5 Tagging

Click on the tag icon to add tags when adding a task. Select any tag from the tag list on the left to filter by that tag. Hover on any task to highlight its tags on the left.

2 Technical Details

- Used `best_in_place` ruby gem for the in-place editing feature.
- Used `Bootstrap` for simple styling.
- Used `autoprefixer` for the stupid vender prefixes that I may need in the future.
- Used `act_as_taggable_on` ruby gem for the tagging feature.
- Wrote some `CoffeeScript` to auto update page content.

3 Reflection

I did not have much time to spend on this assignment. Although I still finished it, personally I feel a bit unsatisfied. Feature for editing the tags of a certain task is missing. It is not hard to do, but I have not come up with a good interface design. I did not have time to refactor the js, or use a front end MVC framework. Nearly all JS code are in one file. Although it does not look messy. It is not best practice. Anyway overall the app feels ok, I think.