

## CVWO Assignment Write-up

### **1. Introduction**

- 1.1. The task management app will be meant for a group of users to keep track of a common set of tasks. Users will be able to perform CRUD operations on their tasks and tag categories to each task.

### **2. User Manual**

- 2.1. The application is hosted at <https://tasks-zsig.herokuapp.com/>. Because Heroku shuts down the dyno every 30 minutes to save resources, the application could take a while to load as the dyno restarts.
- 2.2. The application will stay on a single route with all buttons opening a modal. The updates to your changes would take a while, depending on the Heroku servers.
- 2.3. The Sign Out button on the navbar has no function.
- 2.4. Functions
  - 2.4.1. Click on New Task to add a task. These will appear as cards on the homepage. You can read, edit, delete these tasks via the buttons on the card.
  - 2.4.2. Tag categories to your new or existing tasks after adding them via the Categories button.
  - 2.4.3. Choose to include or remove a deadline in your new or existing task by toggling the switch as required. z

### **3. Tools used**

- 3.1. Overview
  - 3.1.1. This is a React on Rails application using Webpacker, with a PostgreSQL database.
- 3.2. List of tools/libraries used
  - 3.2.1. React
  - 3.2.2. Rails (v6.1.4.4, the latest version of Rails to still use Webpacker by default)
  - 3.2.3. PostgreSQL
  - 3.2.4. React Rails (gem)
  - 3.2.5. Bootstrap
  - 3.2.6. Bootstrap Icons

### **4. Possible additional features**

- 4.1. Login for multiple users
  - 4.1.1. Individual users could have their individual set of tasks enabled by the requirement of login before they can access the application.
  - 4.1.2. This could be accomplished with a login page and the BCrypt gem, and the addition of user ids into the database.
- 4.2. Sorting system based on categories
  - 4.2.1. This would allow the users to filter their tasks based on the categories.
- 4.3. Push notifications
  - 4.3.1. Users could receive reminders for tasks after allowing notifications on their browser, enabled by the webpush and serviceworker-rails Ruby gems.
  - 4.3.2. This could remind users at set times before the deadline to complete the task, even if they are not actively using the task management app.

#### 4.4. Calendar view

- 4.4.1. In addition to the list view of the tasks, they can also be displayed in a calendar, where tasks, assignments, tests for the selected month are displayed according to their deadlines. Those with no deadlines will be displayed in another section of the page.

### **5. Reflection**

I feel that I have accomplished a lot throughout these 2 months of learning and debugging. The application could admittedly have been a lot better – a login feature would have made it more realistic, and the style could have been improved from the default Bootstrap library. However, I am proud that I managed to learn my first tech stack and deploy an actual working application in a short span of time.