

Implemented (Basic Features)

1. Filtering (Implemented partially)
 - Filtering by date and time not implemented.
2. Delete task (Implemented differently)
 - Auto Deletion feature no longer planned.
 - New highlighting system. Task with deadline falling today will be highlighted yellow. Task past deadline will be highlighted red.

Other Features not listed has been implemented as envisioned in the mid-assignment writeup.

Implemented (Future Features)

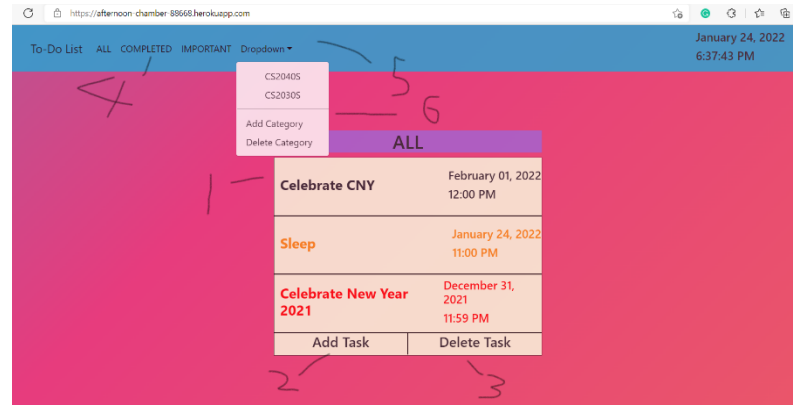
1. “Completed” Tag (Not Implemented)
 - As of now, there is a “Completed” Category
2. “Incomplete” Tag (Not Implemented)
3. Rails implementation (No longer planning to)
4. Notification (Not Implemented)
 - Users will be notified if a task is near its deadline or passed its deadline and is not completed through email.
 - This is implemented with Cron.
5. Login information (Currently in work)
 - The app will ask the user for their login information (username & Password) then the app will redirect to the user’s task list.
6. Sorting (Currently in work)
 - Users will be able to sort tasks by due date.

Implemented (New Features)

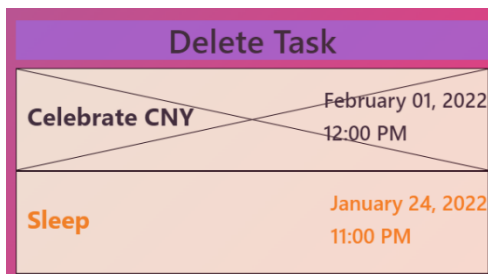
This section encompasses new feature not mentioned in the mid-assignment writeup.

1. Loading Screen
 - A loading screen overlay when the app is getting data or modifying the list.
2. Dynamic Category
 - User can make their own category instead of only pre-set category.

User Manual



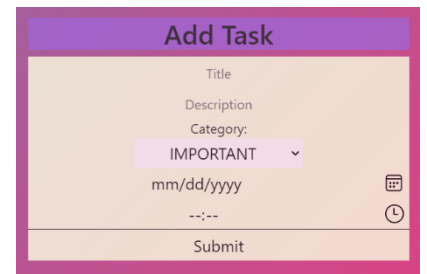
1. A task box
 - Clicking on the task item will initiate edit mode and highlight task.
 - Submitting the edit form will update the highlighted task.
 - Clicking on it while on delete mode will delete the task.
2. Add task button
 - Clicking it will initiate add mode.
 - A form will show up below the task list.
 - Submitting the form will put the new task into the list.
3. Delete task button
 - Clicking it will initiate delete mode.
 - Hovering on a task box will cross the task box.
4. Permanent categories
 - Clicking on the category will filter the task by category and the header will change from showing "ALL" to the current category.
5. Dropdown button
 - Store customizable categories.
6. Dropdown bar
 - Clicking on the category will filter the task by category and the header will change from showing "ALL" to the current category.
 - Add category by clicking the "Add Category" button and submitting a string.
 - Delete category by clicking the "Delete Category" button and clicking on a customizable category. Hovering on the category will highlight it in red.



..1 Crossed Box when hovered in delete mode



2 Highlighted box when clicked showing item to be edited



3 Add Task form. Edit Task form is similar with header reading "Edit Task" and title and description prefilled

Reflection

Having nearly zero experience in Web Development when I first applied for CVWO, I must start learning from the ground up. I was fortunate to have a senior recommend an Udemy course on starting web development and immediately after my last exam, I spent 3 weeks to learn the basic of web development from CSS/HTML/JS until React and Node.js. After completing the course, I read up on the other CVWO resources that wasn't covered by the course such as PostgreSQL, Ruby on Rails and Go. By the time I start on the final App it was already the last week of December.

Before I start on the final app, I laid out plans on what features I am confident I can implement with my current knowledge and given time constraint then I make plans for future features. Following my plan, I was able to roughly chart the progress of the app so that I am less anxious about not finishing the app on time and was able to finish the basic version before sem 2 starts and has focused on bug fixing and improving code before starting to work on other feature.

The experience was not as smooth as I have hoped. There were some planned feature that was not user-friendly, counter-intuitive or needlessly complicated to implement (Clicking on task used to automatically remove it, there was only pre-set categories, there was no completed category and task auto-delete after a while). But in the end, I was able to improvise and come up with better implemented feature.

The Backend was simple as I only need to implement simple CRUD operation. Most of my difficulty is in the Frontend, as I have trouble achieving the desired design using CSS and React and on how to format and present the information, especially date and time. Searching online for solution was also not as easy as I thought especially with recent languages, some of the solution for my question used older version of React which have not implemented Hooks. But in the end, the satisfaction from fixing bug and finishing a feature overwhelm the suffering.

As of the time of writing this, I have not finished implementing a lot of the future features. Currently I am learning how to implement JWT Authentication in my app.

Overall, I am pretty satisfied with the current state of the app, especially considering that 2 months ago I don't even know what CSS or a div is. The experience I gained within the last 2 months has instilled confidence in my ability to learn and practice web development.