

## Basic Use Cases

There will be a home page that displays the tasks that are due today. Another that displays the upcoming tasks that are not due today and another that displays all the current active tasks.

## CRUD Functionalities

- **Create**
  - There will be a button that allows users to create a new task. There are several properties that a task has. There will be a dropdown calendar for users to select the due date. A tag that allows for easy searching. Priority that determines how urgent the task is. Notification settings that can be ticked if users want to be reminder through email at a specific time. Comments that describe the task.
- **Read**
  - Once a task is created, the page is updated and users can click on the task to review it. Users should also be able to search for a task by its tag easily.
- **Update**
  - There is an edit button on the task button that can be used to edit the properties of the task, namely the name of the task, due date, tag, priority, notification settings and comments.
- **Destroy**
  - There is also a delete button on the task button that allows users to delete the task that was created.

Apart from the basic CRUD operations, there are several functionalities that I want to implement.

- **Comment**
  - Users will be able to leave comments on a particular task.
- **Sorting**
  - Users will be able to sort the task alphabetically, by due date or by priority.
- **Notification**
  - Users will be notified through email at a specific time set by the user.

## **Execution Plan**

### **Phase 1 – Basic Functionalities**

I will first work on implementing the create task functionality. I will need to implement a dropdown calendar to select date for the due date and the notification date and time. Tagging system needs to be implemented where a task is associated with a tag that can be searched. I also need to work on comment after that.

Once tasks can be created, a button that can be clicked to view each task will be implemented. A sort button to sort the tasks according to different parameters will also need to be implemented here.

Then, I will add an edit button on the task button that allows users to edit the various properties of the task.

Next to the edit button, I will add a delete button that allows users to delete the task created.

### **Phase 2 – Frontend**

I will use React to make the frontend of the todo manager.

### **Phase 3 – Optional Tasks**

If there is enough time, I will work on the optional tasks such as Cron, TypeScript, RESTful APIs, Hosting, Redux and Docker.