Use Cases of To-do Manager

Use Cases	
Use Cases: to keep track of	Features needed
Tasks and their content	Implement a system to record tasks, and a small description of them
Completion of tasks	Implement checkbox that records completion
The type/nature of the tasks	Implement a tagging system Potential sub-feature: implement a filter/organization system for users to access tasks based on their tags
Volume of tasks they have	Implement a counter showing the number of uncompleted tasks
Due date of tasks	Implement a system to record due dates of tasks Potential sub-feature: implement a system that displays and filters tasks by due date

Features of To-do Manager

The main features of the to-do manager will include

- CRUD operations of the tasks
- Tagging system
- Checkbox to mark completion of tasks

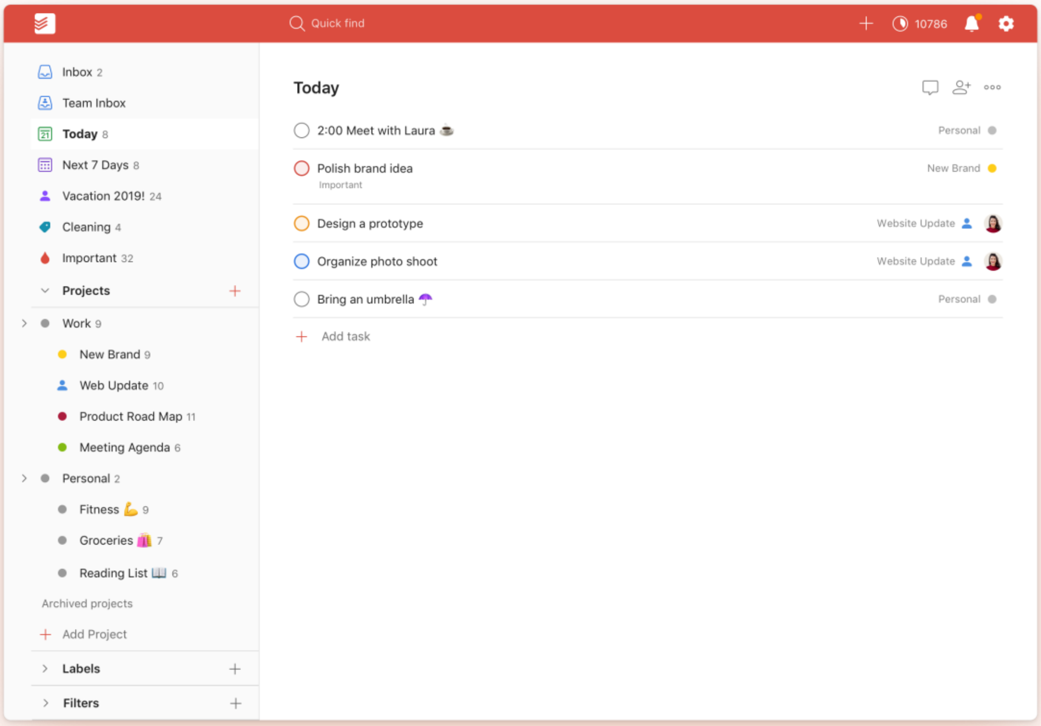
Additional features I hope to implement would include

- Counter for number of tasks
- Due date of tasks
- Different tabs to access tasks based on due date or tags
- Ability to organize tasks in a specific order based on due date or tags
- Search bar function

Execution Plan

<u>Frontend</u>

The layout of the website will be modeled loosely after Todoist, an existing to-do list application. It will have a banner, a navigation bar on the left displaying the tags and allowing the user to filter tasks based on tags or due date. The tasks will be displayed in the main display.



I have completed a similar looking layout using HTML and CSS and intend to use React with Semantic UI to complete the design.

<u>Backend</u>

Rails will be used for the backend. I have read up on the philosophy behind Rails and how it works. I am currently learning how to integrate Rails with React to write the project.