The design of this app is inspired by todoist.com.
Use cases:
1. Login/Registration page
Registration requires unique email (with respect to model-level validation).
I have yet to implement database level restrictions on uniqueness.
After registration/login, a session is created until the browser is closed.
After lregistration/login , the user will be redirected to the homepage.
The user should see a Welcome, [user’s name] message below the Jumbotron.

2. Header
Currently, the buttons “Support” and “Notifications” are unimplemented. They will remain as decorations since I am trying to mimic the look of todoist.com.

2.1. TodoApp NavBrand
Clicking this will redirect the user to the homepage.
3.2. Sidebar toggler
Opens and closes the sidebar
When the sidebar closes, all opened lists(Projects, Tags, Labels, Filters) in the sidebar will be closed.

2.3. Home link
Redirects to the homepage

2.4. Search bar
If the current route does not begin with “/tasks”, the user will be redirected to “/tasks” with the relevant search parameters as long the search bar text field is non-empty.
The tasks will be filtered based on whether the task name contains the search term. Note that search is case-sensitive.
I may make it non-case-sensitive in future.

If the user is currently on a “/tasks…” page, with certain filters applied, then the page will filter by both the applied filter as well as the search term.

2.5. Quick Add New Task
Allows the user to quickly add a task on any page.

2.6. Logout
Logs the user out and resets the session.

3. Sidebar
3.1. Homepage (Links to the homepage)

3.2. All tasks

View all tasks without filters.

3.3. Projects

Click to open a list consisting of the current projects.
Click the corresponding projects to view the project’s details and the tasks associated with the project.

3.4. Tags

Tags are essentially permanent labels that cannot be deleted.
The tags “Work”, “Social”, “Personal” are created per-user during registration.

3.5. Labels

Labels work just like tags, however, the user can assign bootstrap colors to the label during label creation.

3.6. Filters

A filter is a collection of filter fields, which will all be applied in the app view.
The user can specify a start date and end date (at the backend it is required that start date < end date), as well as a group of tags and labels that will become part of the filter.

4. Main view

4.1 All task view

Shows all tasks ordered by date.
Under the overdue section, the user is not allowed to create tasks. (However, the user can still create an overdue task using Quick Add Task).
Click add new task to create a task without any tags or labels.
Click on each task to see its details, like subtasks and comments.
Currently, the activity tab is unimplemented.
Click on the edit button of any task to edit its parameters.
Editing of subtasks and comments is not supported.

4.2. Project View

Shows a list of tasks (not ordered by date) that are relevant to the project.
Any task created here will belong to the project.

4.3. Tag/Label View

Shows the tasks with the currently selected tag/label.
Any task created here will have the currently selected tag/label.

4.4. Filter View

Shows the tasks after filtering by the currently selected filter.
E.g. If during filter creation, the filter has the properties
startdate: 20 Dec 2020

enddate: 25 Dec 2020
tags: [“work”, “personal”]
labels: [“label_1”, “label_3”, “label_4”]
Then a task will only be shown in the filter view if and only

- the task’s deadline falls between 20 Dec 2020 and 25 Dec 2020
- the task has the tags “work”, “personal”
- the task has the labels “label_1”, “label_3”, “label_4”.

For e.g. the following task will be shown in the filter view:
task = {
deadline: 20 Dec 2020
relationship.tags: {“work”, “personal”}
relationship.labels: {“label_1”, “label_3”, “label_4”, “label_5”}
} *(This is not a legit js object, but you get the idea)*

Any task created under the filter view will have precisely the tags and labels associated with the current filter.
e.g. If during filter creation, the filter has the properties
startdate: 20 Dec 2020, enddate: 25 Dec 2020
tags: [“work”, “personal”]
labels: [“label_1”, “label_3”, “label_4”]

Then any new task created will be like this:
task = {
content: “XXX”, deadline: XXX,
relationship.tags: {“work”, “personal”}
relationship.labels: {“label_1”, “label_3”, “label_4”}
}

5. Deletion
The following items can be deleted:
Labels (Click on the cross next to the label on the sidebar to delete it)
Projects (In the project view there is a delete project button that will also delete the associated tasks)
Tasks (Click on the circled cross on the left of each task)
Subtasks and Comments (Click of the circled cross)
Yet to implement: Deletion of filters
Note:
I have outlined a lot of problems I faced during development within my code.
I would like keep them during my final submission as well, because I want to refer to some of my still-unsolved issues when I come back to this project in future.