Todo List

The practice task is a simple Todo List application written without using libraries with pure JS only. To simplify the development, it is proposed to use a ready-made assembly based on Webpack with the ability to hot reload and compile scripts and styles into a single bundle.

* *You should use a prebuilt assembly to start working on the application:* *https://github.com/IgorKonovalov/HTML-JS_boilerplate*

Working with a development environment

To work with a environment, you should:

1. Check that you have the latest version of Node.js
2. Clone the repository
3. Run ‘*npm install’* command from the project folder
4. For a brief description of commands to run, see the repository's README file
5. The project supports hot reload of styles, html and js code out of the box

Basic requirement

1. Interface to view/add/edit/delete Todo list items
2. The application consists of two lists: **Open** and **Done**
3. Each entry in the lists must have two statuses: completed or open
4. The application provides the ability to search by lists
5. The state of the application should be saved after a page is reloaded

Technical details

Application

1. The application must have a search field for all records. When a user starts typing a query in the search field, each time a key is pressed, if the field is not empty, both **Open** and **Done** lists are filtered relative to the entered query. When the field is cleared, the lists are displayed in their original state.
2. A data should be stored in localStorage and displayed after a page is reloaded. Saving should occur at every key action
    a. add/delete/edit records
    b. sorting lists

Lists

1. To add a new entry to the **Open** list, a user enters text in the *“New task"* field and presses the *"ADD"* button or the *Enter* key. The *"New task"* field is cleared and a new entry appears at the top of the **Open** list.
2. Under each list there is a link with which you can clear the entire list.
3. The **Open** list should be sorted by:
    a. records (asc/desc)
    b. creation date (asc/desc)
4. The **Done** list should be sorted by:
    a. records (asc/desc)
    b. due date (asc/desc)

Records

1. Each record must have the following fields:
    a. text
    b. creation date
    c. status
    d. due date (after changing the status to completed)
2. When a user clicks on the checkbox of an open record, it is marked as completed and will be placed in the **Done** list
3. When a user clicks on the checkbox of a completed record, it is marked as opened and will be placed in the **Open** list
4. When a user hovers over a record, the delete icon appears

5. When a user clicks on the delete icon the entry is deleted
6. To edit a record a user needs to double click on its text
7. When a user double-clicks, the text is replaced with an input field where you can correct the current value. When a user presses *Enter* key, the value is saved and the field is hidden, showing the corrected text. When a user presses *Esc* key, the field is hidden, showing the previous value.

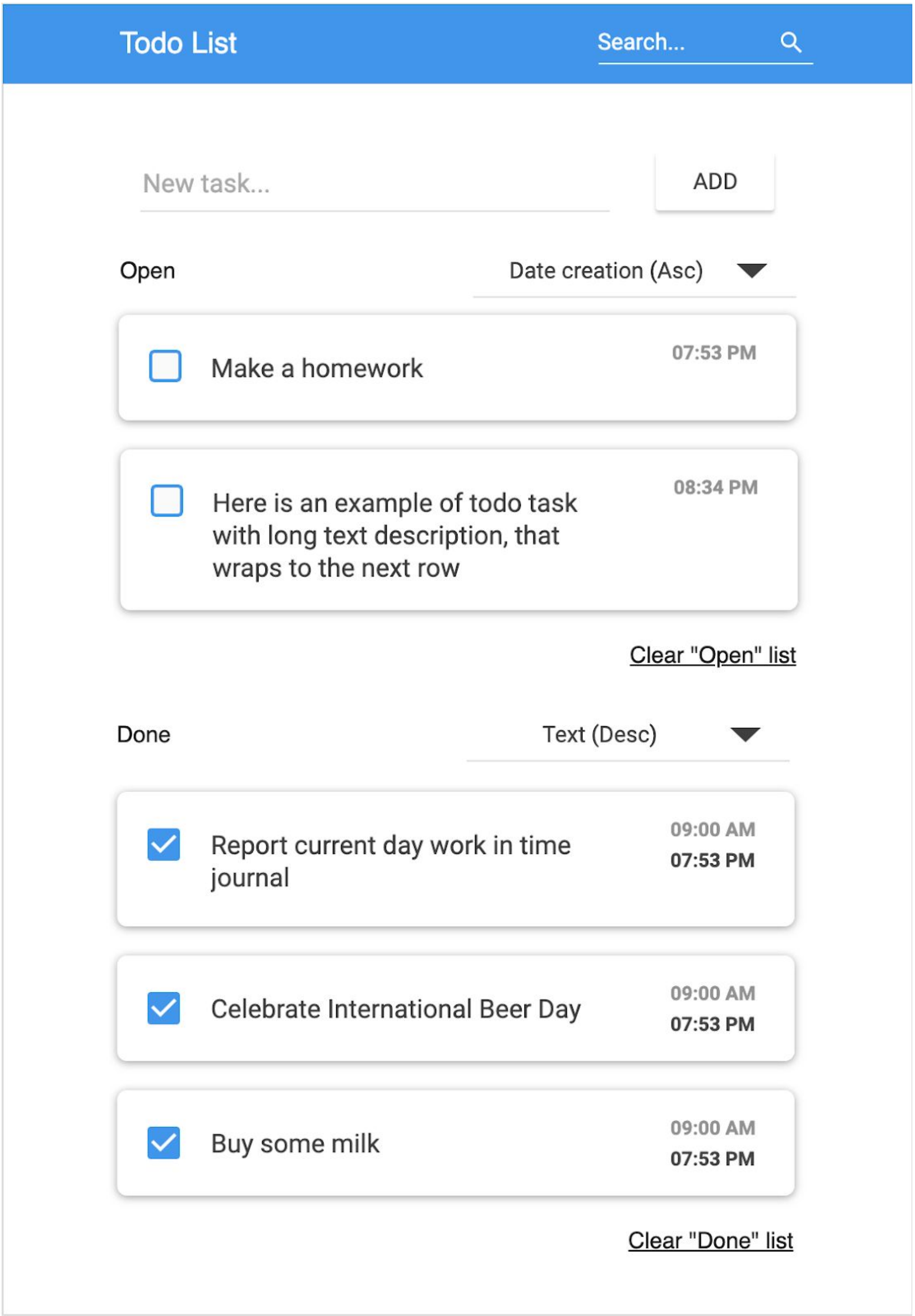
Design

The design of the application is presented in the form of layouts below in this document. When working with layouts, you should pay attention primarily to the location of elements and their behavior. The style of the elements is optional, but it is recommended to use the one proposed. Precision indentation between elements can also be neglected, as it is presented solely for the sake of clarity.

Check list

1. Overall interface of the application
2. Display two task lists
3. Add, edit, and delete tasks
4. Clearing of task lists
5. Sorting of task lists
6. Search by task lists
7. Save application state after page reload

Layouts



Open — Date creation (Asc)

□ Make a homework — 07:53 PM

□ Here is an example of todo task with long text description, that wraps to the next row — 08:34 PM

Clear "Open" list

Done — Text (Desc)

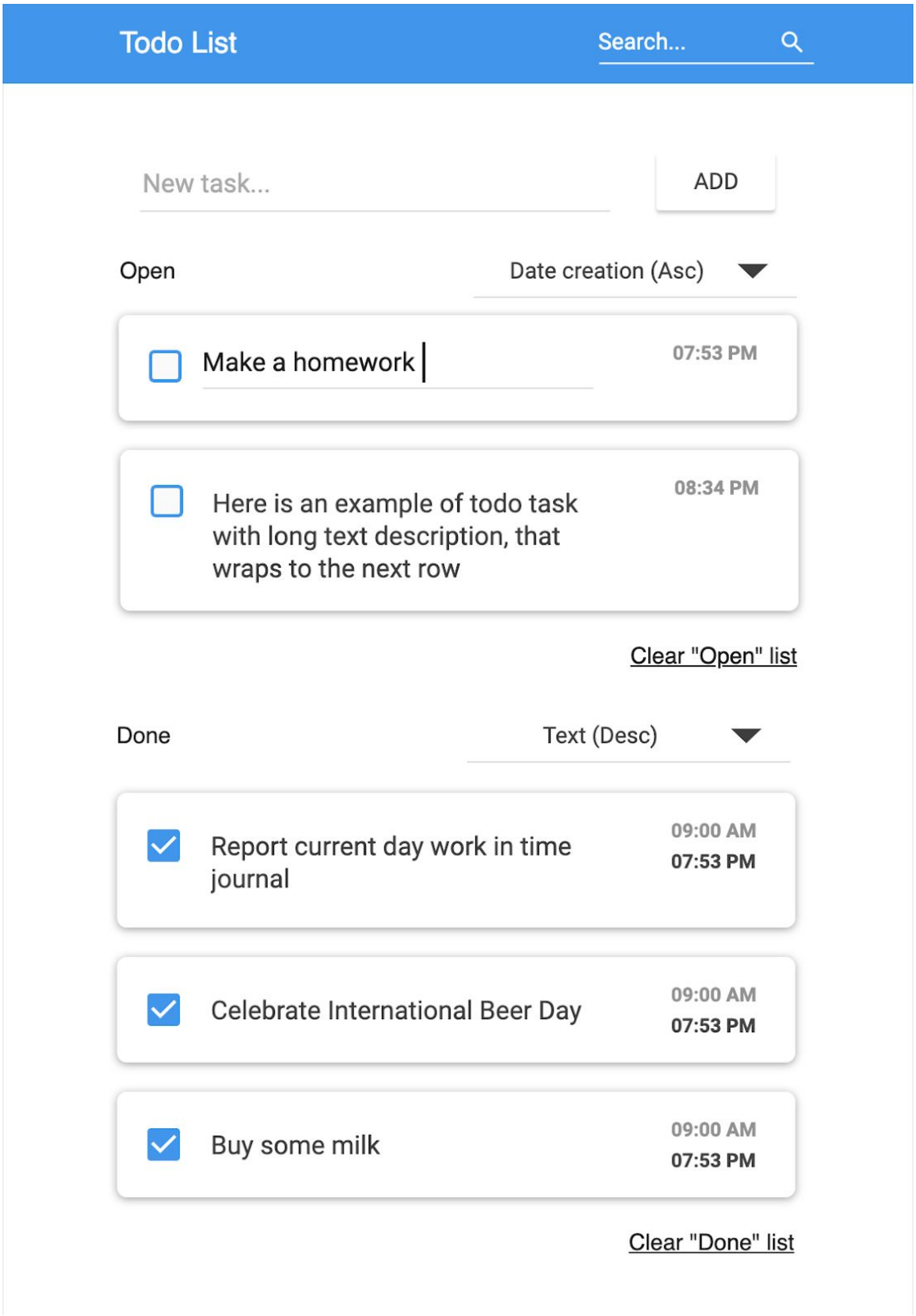
☑ Report current day work in time journal — 09:00 AM **07:53 PM**

☑ Celebrate International Beer Day — 09:00 AM **07:53 PM**

☑ Buy some milk — 09:00 AM **07:53 PM**

Clear "Done" list

Main application window



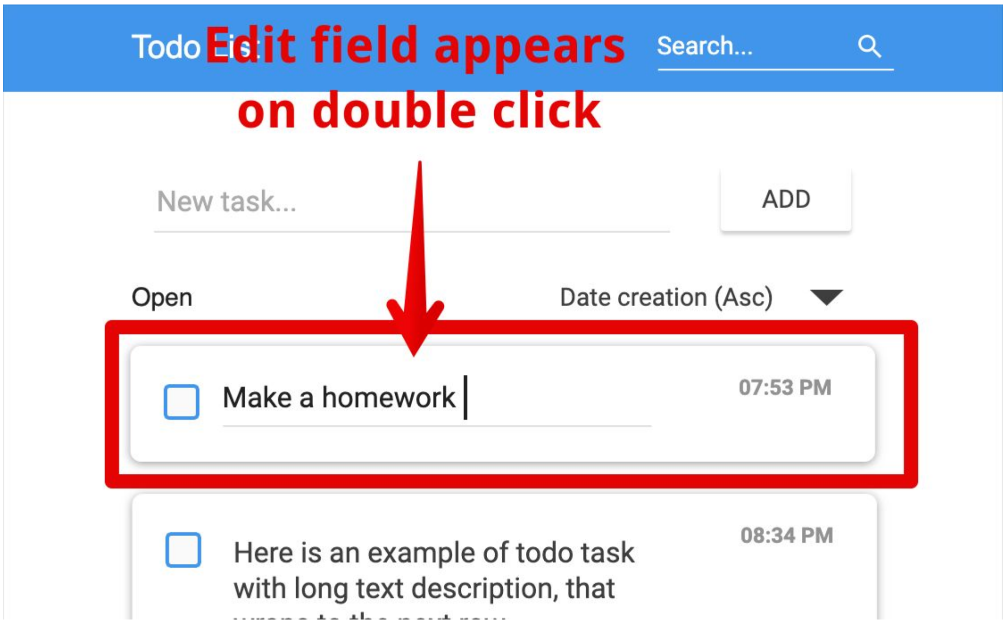
Open — Date creation (Asc)

- □ Make a homework — 07:53 PM
- □ Here is an example of todo task with long text description, that wraps to the next row — 08:34 PM

Done — Text (Desc)

- ☑ Report current day work in time journal — 09:00 AM 07:53 PM
- ☑ Celebrate International Beer Day — 09:00 AM 07:53 PM
- ☑ Buy some milk — 09:00 AM 07:53 PM

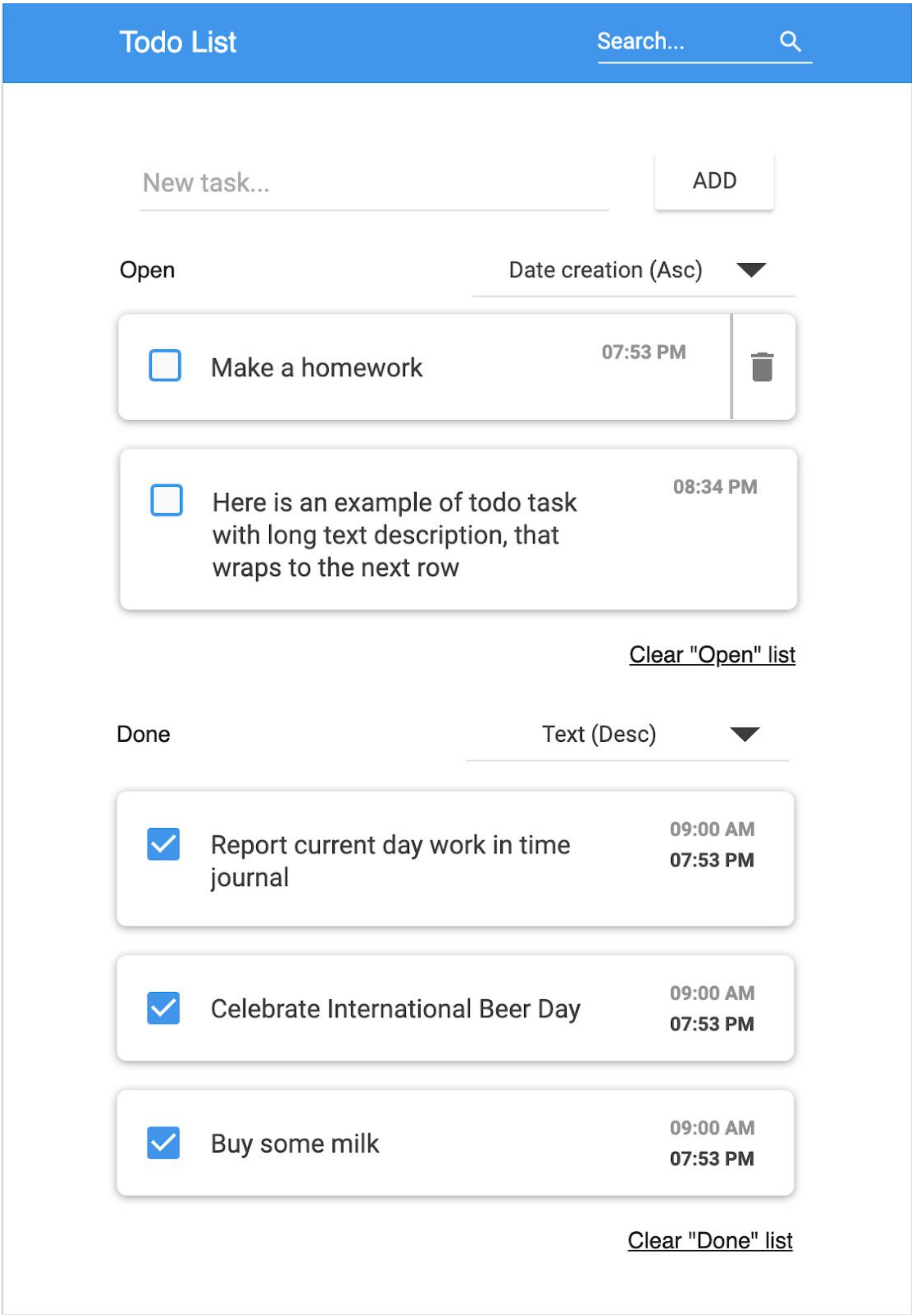
Editing a task



□ Make a homework

□ Here is an example of todo task with long text description, that

Double click on the button brings up the edit field



□ Make a homework 07:53 PM

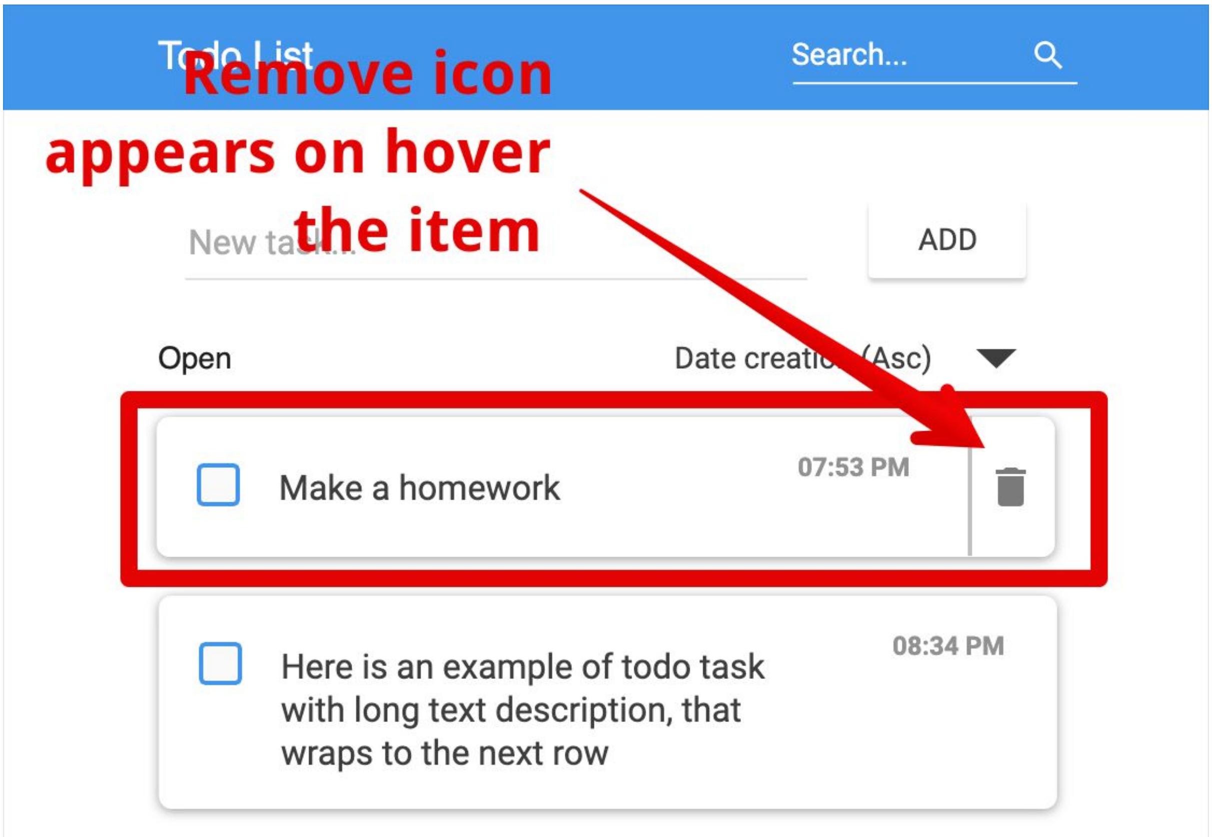
□ Here is an example of todo task with long text description, that wraps to the next row 08:34 PM

☑ Report current day work in time journal 09:00 AM 07:53 PM

☑ Celebrate International Beer Day 09:00 AM 07:53 PM

☑ Buy some milk 09:00 AM 07:53 PM

Task delete icon



Remove icon appears on hover the item

□ Make a homework 07:53 PM

□ Here is an example of todo task with long text description, that wraps to the next row 08:34 PM

Delete icon appears when user hovers over the item