



# React Challenge

---

## 1 Purpose

The purpose of this challenge is to assess your skill in several languages and technologies used in modern web design, such as HTML5, CSS3, TypeScript, JavaScript, and React.

## 2 Challenge

The challenge is to create a simple single page application (SPA) for managing a local “to-do list”. The application should allow the user to;

- List the items on the to-do list, optionally filtered by state and sorted.
- Add an item to the list.
- Edit an item on the list (i.e., change its description or mark it as complete).
- Remove an item from the list.

### 2.1 Requirements and Recommendations

The application should be programmed in TypeScript, or alternatively in JavaScript (ES6/7).

The use of the ‘var’ keyword is forbidden.

Use ‘async/await’ to handle asynchronous operations (e.g., retrieving data from the server).

You can use any public package in the NPM repository.

The use of utility libraries, such as **lodash** or **underscore** is forbidden.

The use of promise libraries, such as **bluebird** is forbidden.

You are encouraged to use **ESLint** and the Airbnb rules (<https://github.com/airbnb/javascript>).

The application must use the **React** library (<https://reactjs.org/>) for rendering its components.

For your convenience, we strongly recommend you scaffolded the application using the **create-react-app** tool (<https://github.com/facebook/create-react-app>).

The application must use **React Hooks** (<https://reactjs.org/docs/hooks-intro.html>).

The application must use **React Context** (<https://reactjs.org/docs/context.html>) for managing its global state (i.e., the “to-do list”).

The source code must be published on a git repository. Make sensible commits and detail the changes on the commit messages.

## 2.2 Layout

The application should have two sections. Both sections should be displayed in the same view, with a task creation form atop the “to-do list”, as shown below:

<input type="text" value="Write new task here..."/>		<input type="button" value="Create"/>
Tasks		
<input type="checkbox"/>	Learn JavaScript	Edit / Delete
<input checked="" type="checkbox"/>	Complete the React Challenge	Edit / Delete
...		
Hide completed <input type="checkbox"/>		

## 2.3 User Interaction

The following user interactions must be implemented:

- **Creating a task** – The user writes the task on an input text field and clicks the “Create” button. The task is then added to the list and marked as incomplete.
- **Marking a task as complete/incomplete** – The user clicks the checkbox on the left of the list entry, which toggles its state.
- **Editing a task** – Clicking the “Edit” link on the right side of the list entry should allow the task name to be edited. You can display an input text field in-place, or re-use the task creation form.
- **Deleting a task** – Clicking the “Delete” link on the right side of the list entry should delete the entry.
- **Filtering the list** – Toggling the “Hide completed” checkbox on the bottom of the list should filter the completed entries from the list.
- **Sorting the list** – Clicking the “Tasks” list header should sort the list alphabetically from A to Z, clicking it again should change the order from Z to A, and a third time return to the default sorting (by creation date).

## 2.4 Styling

The styling of the elements is up to you, but you should show proficiency in positioning and sizing elements, styling text, colors, and borders.

## 2.5 Bonus – Local Storage

Have the application synchronize its global state with the browser's local storage, to allow the user to close and re-open the browser without losing information.

You can investigate <https://medium.com/@akrush95/global-cached-state-in-react-using-hooks-context-and-local-storage-166eacf8ab46> if you wish to pursue this endeavor.

## 2.6 Bonus – Animations

Use animations to improve the look and feel of your application.

You can investigate <https://medium.com/hackernoon/5-ways-to-animate-a-reactjs-app-in-2019-56eb9af6e3bf> if you wish to pursue this endeavor.

## 2.7 Bonus – Authentication

Add support for user authentication.

- Registration form
- User login form
- Edit profile form
- Logout button