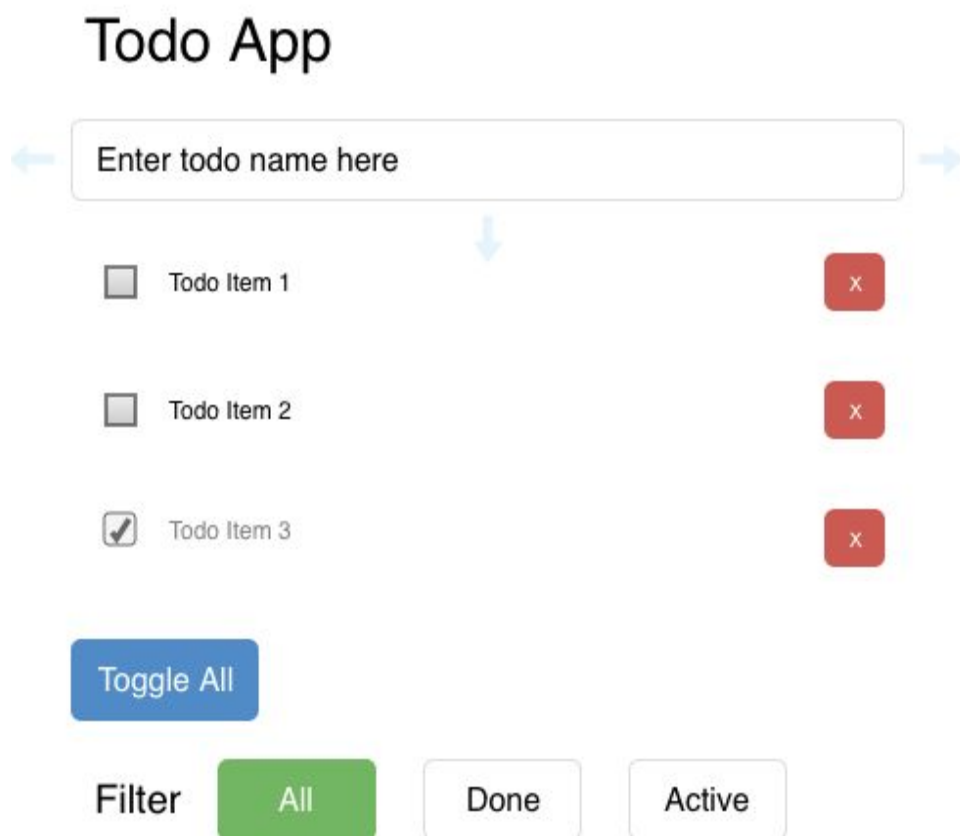


ReactJS - A SIMPLE TODO APP

1.UI/UX



2. Technical requirements

- Use codesandbox to do the test
- Use React to create the application
- Use the new context API to manage the state
- Do not use redux and other state management packages
- Do not use the interface package as: bootstrap, antd ...

3. Features

- "Enter todo name here" is a placeholder for the todoName textbox

- b. When pressing todoName, press enter to add a new todo item, delete textbox content.
- c. The Todo item is represented by checkbox and name
- d. By clicking on the checkbox of the todo item, it will be converted to DONE
- e. There are 2 statuses for a todo item: DONE / ACTIVE
- f. When clicking on Toggle All, the entire state of the todo items is displayed (do not change the status of the todo not displayed) DONE ⇔ ACTIVE
- g. Clicking on All will display all the todo items: DONE and ACTIVE
- h. Clicking on Done will display the DONE status of the todo item
- i. Clicking on Active will show the item to have ACTIVE status
- j. Clicking on the red X button will delete the todo item in that row

4. Evaluation Factors

- a. Organization of folders / files for components
- b. How to split components
- c. How to use HOC
- d. How to use the new context API
- e. How to organize your app state
- f. Separation of logic and view
- g. Optimal re-render
- h. Logic code is concise, effective
- i. How to use immutable objects

5. Extra points

- a. Create beautiful UI
- b. Animation effect
- c. Use react hooks
- d. There are additional utility features that are beyond the requirements
- e. Use firebase / firestore to store todo lists