React

Lab Guide

1. Design the component tree for a todo list application. This application maintains a list of things to do (task list), with indication if a task is completed or not. Each task has a checkbox against it to mark if it has been completed or not, and a button to delete the task from the list. On top of the task list is an input to accept the name of a new task to be added to the list. Below the input is a button that the user clicks after entering the task name in the input. Once the user clicks the button, the task is added to the task list.

2. Create a div React element. It should have as children the following 3 elements
a. A span with id 'counter' and the text '0' as child
b. A button with text '+'
c. A button with text '-'
Also, render the React element inside the root div.

3. Redo the above exercise by creating the element using JSX.

4. Render 2 div elements with the same structure as in the previous exercise. Try each of the following 3 ways to do it.
a. Render the 2 divs as children of another div
b. Add the 2 divs to an array and render the array of React elements
c. Enclose the 2 divs in a React.Fragment and render

5. Define a function component called TaskInput that shall take user input in your todo list app. The TaskInput component shows a text input, and a button. Enclose the 2 inside a form element and return the form element. Render an TaskInput element.

6. Redo the above exercise using a class component.

7. Create a class component that shows a number within a span. This number should initially be 1, and should double every second. For example, after a second it becomes 2, then after another second 4, etc.

8. Create a function component called BusinessCard. It takes as input props the following

a. Full name

b. Designation

c. Company

d. Contacts details in an object with tel and emailid properties

e. Image path (eg. if the image is in the current folder you may pass './name-of-image.jpg')

9. Redo the above exercise by defining a new component called Contact. This takes the contact object as a prop and shows the contact details. Use it in the BusinessCard component.

10. Define a TaskList component (class component) that takes an array of tasks like below.

```
const tasks = [
    { name: 'Buy milk', completed: true },
    { name: 'Learn React', completed: false },
]
```

The TaskList renders the tasks within an ordered list (ol, li). Each li should have a checkbox, name of the task and a delete button.
Define a App component that has a TaskInput element and TaskList element as children. The app takes the tasks array as prop and passes it down to TaskList component. Render the App component element.

You can also define a TaskListItem component if you like. It renders a single task (checkbox, name of the task and a delete button).

11. Add styles to the tasks in the task list. If a task is completed it appears struck out and in olive color. If not it appears normally.

12. Change the Counter component by adding an input text box where user can enter a number. If '+' is clicked, the counter value increases by the number specified in the text box. Similarly for click on '-'.

13. Add state to the TaskList component and store tasks array in it. Implement add task, delete task, and marking task as complete/not complete functionality in the todo list app.

14. We shall add filter functionality in the todo list. Add a dropdown with options - 'All', 'Completed', 'Not completed'. 'All' should be selected by default, and all tasks are shown. When 'Completed' is selected, only completed tasks are shown. When 'Not completed' is selected, only incomplete tasks are shown.

15. Add PropTypes and defaultProps for all components in the todo list app.