

*ReactJS*

## Lab Assignment

|                |                      |
|----------------|----------------------|
| Document Code  | 25e-BM/HR/HDCV/FSOFT |
| Version        | 1.0                  |
| Effective Date | 19/10/2021           |

**RECORD OF CHANGES**

| Full | Effective Date | Change Description    | Reason | Reviewer | Approver |
|------|----------------|-----------------------|--------|----------|----------|
| 1    | 26/Oct/2021    | Create new assignment | Create | ThuanNH3 | VinhNV   |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |
|      |                |                       |        |          |          |

## Contents

|                               |   |
|-------------------------------|---|
| I. Objectives .....           | 4 |
| II. Overview of the app ..... | 4 |
| 1. Features .....             | 4 |
| 2. Interface .....            | 4 |
| III. Details .....            | 5 |
| 1. Enter a task name .....    | 5 |
| 2. Task .....                 | 5 |
| 3. Note: .....                | 6 |



|           |                        |
|-----------|------------------------|
| CODE:     | ASM1- ReactJS-todo-app |
| TYPE:     | N/A                    |
| LOC:      | N/A                    |
| DURATION: | N/A                    |

## I. Objectives

- Able to initialize a React project using create-react-app CLI.
- Understand structure of the React project.
- Understand how State, Props and Events work in React.
- Dividing simple components.

## II. Overview of the app

### 1. Features:

- Add and delete tasks.
- Users can change status to completed or incompleted.
- There are filters to classify completed or incompleted work.

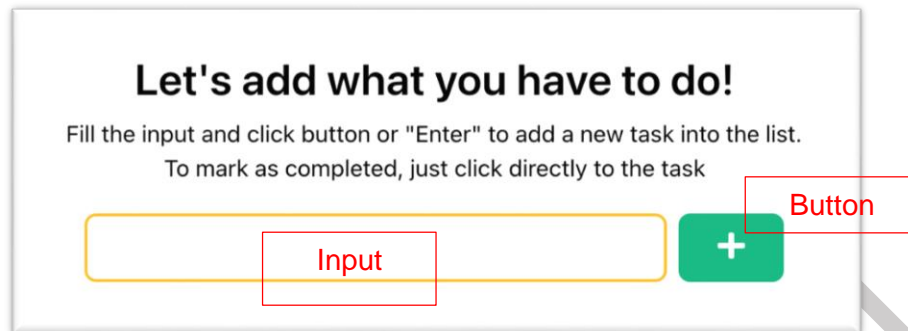
### 2. Interface:

- The app's interface can be found at the following link:  
<https://fa-react-todo-app.vercel.app>

A screenshot of a web application interface for a todo list. At the top, it says "Let's add what you have to do!". Below this, there's a text input field and a green button with a white "+" sign. Instructions below the input say: "Fill the input and click button or 'Enter' to add a new task into the list. To mark as completed, just click directly to the task". Below the input is a yellow box containing a list of tasks. The list is titled "List:" and has a dropdown menu set to "All". The tasks are: "1. Hoc react", "2. Hoc java", "3. Hoc angular", and "4. Hoc .Net". The first task is crossed out with a blue line.

### III. Details

#### 1. Enter a task name:



The mockup shows a white box with a light gray border. Inside, the text "Let's add what you have to do!" is centered in bold. Below it, two lines of smaller text read: "Fill the input and click button or 'Enter' to add a new task into the list." and "To mark as completed, just click directly to the task". At the bottom, there is a yellow-outlined input field with a red box labeled "Input" inside it, and a green button with a white "+" sign and a red box labeled "Button" next to it.

Requirements:

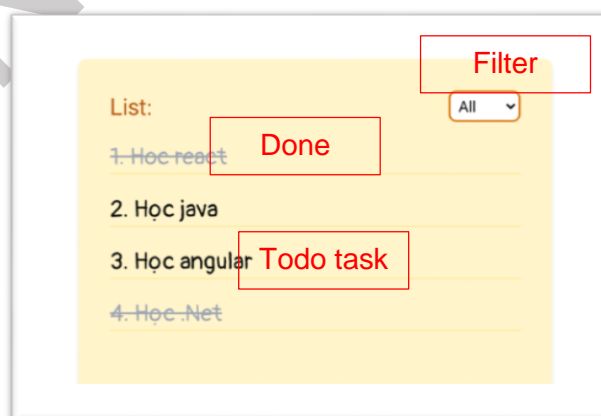
##### a) Interface:

- An input box to enter a task name.
- A "+" button to add the task.
- Color codes:
  - Input border color: #FBBF24
  - Button Color: #10B981

##### b) Function:

- Enter a task name in the input box
- When you click on "+" button, a new task is added to the list and the input is cleared.
- **Bonus:** Instead of clicking the "+" button, users can press the Enter key for the same ability.

#### 2. Task list:



The mockup shows a white box with a light gray border. Inside, there is a yellow background area. At the top left of this area, the word "List:" is written. Below it, a list of tasks is shown: "1. Hoc react", "2. Hoc java", "3. Hoc angular", and "4. Hoc Net". The first task "1. Hoc react" is crossed out with a light blue line. To the right of the list, there is a "Filter" dropdown menu with "All" selected. A red box labeled "Done" is placed over the first task, and a red box labeled "Todo task" is placed over the third task.

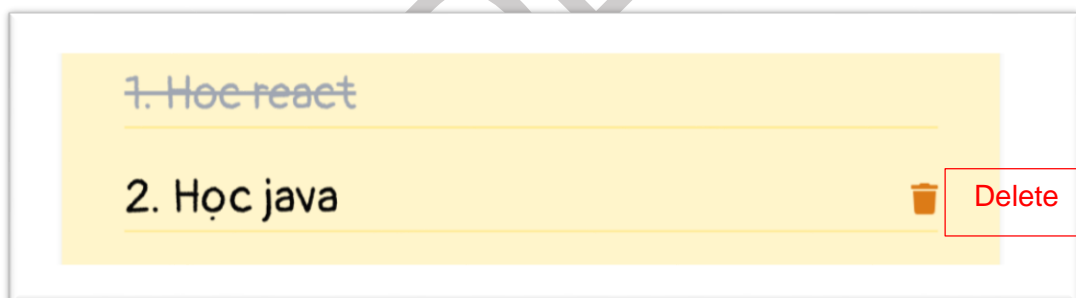
## Requirements:

## a) Interface:

- The interface is as design as possible.
- Each job is displayed on a row with order number.
- The task which marked as incompleted will be black.
- The completed task will be marked as gray and a line through.
- The filter can be filtered with three options:
  - All: Show all the task.
  - To do: Only show incompleted task.
  - Done: Only show the completed task.
- Color codes:
  - Background: #FEF3C7
  - List text/delete icon: #B45309
  - Todo text: #000
  - Done text: #9CA3AF

## b) Function:

- When clicking on a task, change the status of "Incompleted" to "Completed" and vice versa.
- When hovering a task, the delete icon will appear
- **Bonus:** When deleting or filtering, the order number of each task will be updated accordingly. For example, there are 4 tasks numbered from 1 to 4. When the second task is deleted, the order numbers will become from 1 to 3.



## 3. Note:

- Only use React Class Components.
- Do not use any frameworks or libraries such as Bootstrap, NPM packages....

----- (END && GOODLUCK) -----