Learner Assignment Submission Format

Learner Details

- **Name:H O Akash**
- **Enrollment Number:SU625MR015**
- **Batch / Class:MERN Stack**
- **Assignment: React JS**
- **Date of Submission:31/07/2025**

Problem Solving Activity 1.1

1.TO DO LIST

---

2. Algorithm

1. Initialize tasks array with initial tasks.
2. Initialize newTask string.
3. When input field is changed, update newTask string with input field value.
4. When "Add" button is clicked, add newTask to tasks array and reset newTask string to empty.
5. When "Delete" button is clicked, remove task from tasks array at specified index.
6. Display tasks array as ordered list.

---

3. Pseudocode

```
BEGIN
  INITIALIZE tasks array with initial tasks
  INITIALIZE newTask string
  WHEN input field is changed
    UPDATE newTask string with input field value
  END WHEN
```

WHEN "Add" button is clicked

ADD newTask to tasks array

RESET newTask string to empty

END WHEN

WHEN "Delete" button is clicked

REMOVE task from tasks array at specified index

END WHEN

DISPLAY tasks array as ordered list

END

---

4. Program Code(jsx)

```
import React, { useState } from 'react'
import './ToDoList.css';


function ToDoList() {
    const [tasks, setTasks] = useState(["Get up","Take a bath"," Tiffin","
Go for Work"]);
    const [newTask, setNewTask]=useState("");


    function handleInput(e){
    setNewTask(e.target.value);
    }
    function addTask(){

        setTasks(t=>[...t, newTask])
        setNewTask("");
```

```
    }
    function deleteTask(index){
        const UpdatedTasks = tasks.filter((_,i) => i!== index);
        setTasks(UpdatedTasks);
    }



return (
    <div className='a'>
      <h1>My To Do List</h1>
      <div>
        <input
          type="text"
          placeholder='Enter a task'
          value={newTask}
          onChange={handleInput}
        />
        <button className='add' onClick={addTask}>Add</button>
      </div>
      <ol>
        {tasks.map((task, index) => (
          <li key={index}>
            <span className='text'>{task}</span>
            <button className='delete' onClick={() =>
deleteTask(index)}>Delete</button>


          </li>
```

```
        ))}
      </ol>
    </div>
  );
}

export default ToDoList
```

---

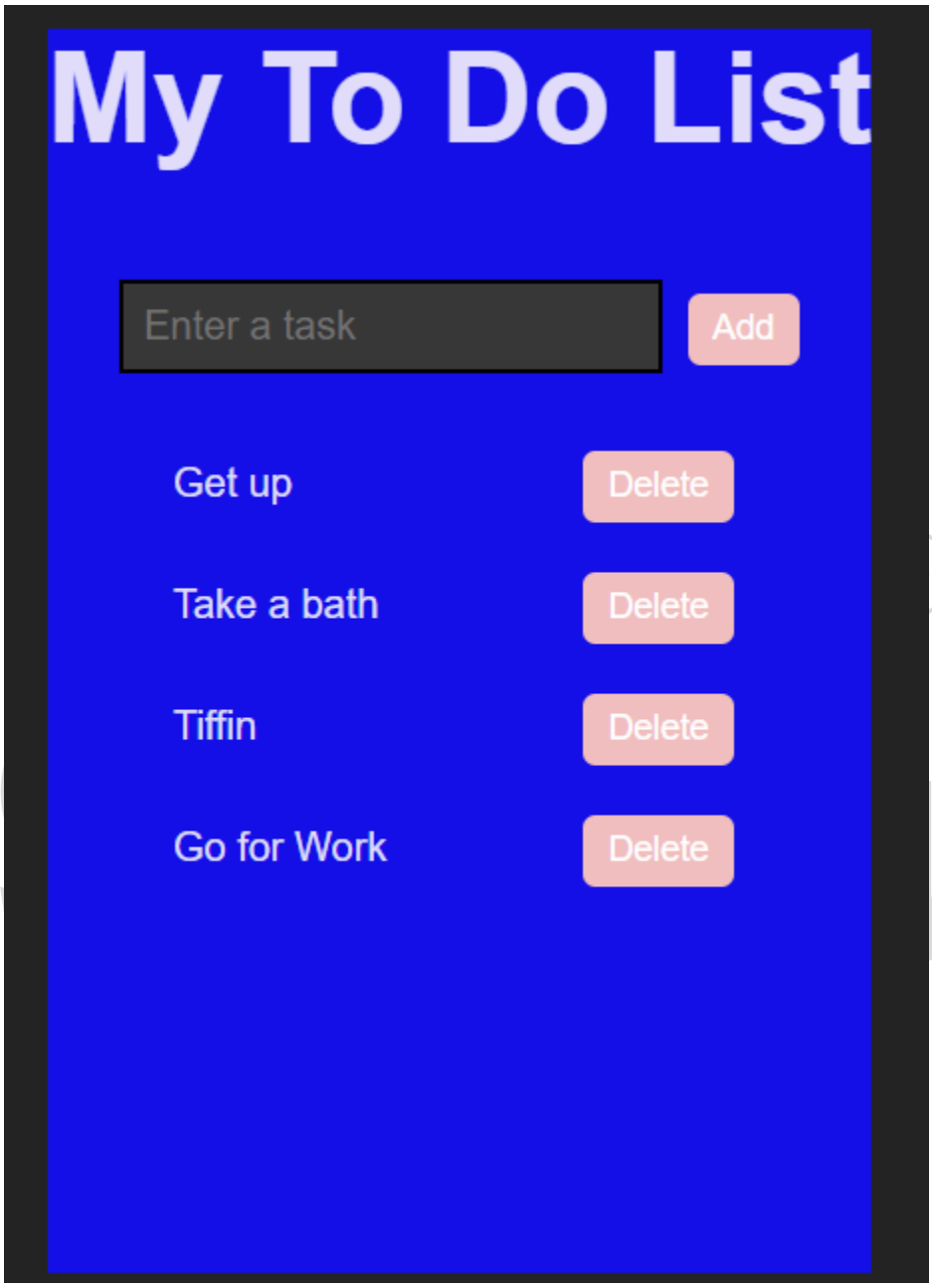
4. Program Code(CSS)

```
.a {
    max-width: 600px;
    margin: auto;
    text-align: center;
    font-family: sans-serif;
    background-color: #1713ea;
    height: 500px;
}

input[type="text"] {
    padding: 8px;
    margin: 10px 5px;
    width: 60%;
    font-size: 16px;
    border: 2px solid black;
}
```

```
button {
    padding: 6px 10px;
    margin: 5px;
    font-size: 14px;
    cursor: pointer;
    border-radius: 5px;
    border: none;
    background-color: #f2c0c0;
    transition: background-color 0.2s;
}

button:hover {
    background-color: #ccc;
}

ol {
    padding-left: 0;
}

li {
    list-style: none;
    margin: 10px 0;
}

.text {
```

```
    display: inline-block;

    width: 150px;

    text-align: left;

    margin-right: 10px;

}
```

6. Screenshots of Output



7. Observation / Reflection

The given code is a simple To-Do List application built using React. The application allows users to add new tasks and delete existing tasks.

One area for improvement is the lack of error handling. For example, if the user tries to add an empty task, the application will add an empty string to the tasks array. To improve this, we could add a check to ensure that the newTask string is not empty before adding it to the tasks array.

Problem Solving Activity 1.2

Follow the same Structure as problem Solving Activity 1.1.