Experiment-4

Develop a To-Do List Application using React functional components that demonstrates the use of the useState hook for state management. Create a functional component named ToDoFunction to manage and display the to do list. Maintain a list of tasks using state. Provide an input field for users to add new tasks. Dynamically render the list of tasks below the input field. Ensure each task is displayed in a user-friendly manner. Allow users to delete tasks from the list. Mark tasks as completed or pending, and visually differentiate them.

App.js

```
import React, { useState } from 'react';
import './App.css';
function ToDoFunction() {
// State to hold the list of tasks
const [tasks, setTasks] = useState([]);
// State to hold the current input value for adding new tasks
const [newTask, setNewTask] = useState('');
// Function to handle adding a new task
const addTask = () => {
if (newTask.trim() === '') return; // Don't add empty tasks
const task = {
    id: Date.now(),
    text: newTask,
    completed: false,
     };
  setTasks([...tasks, task]); // Add the new task to the list
  setNewTask(''); // Clear the input field
 };
// Function to handle task completion toggle
```

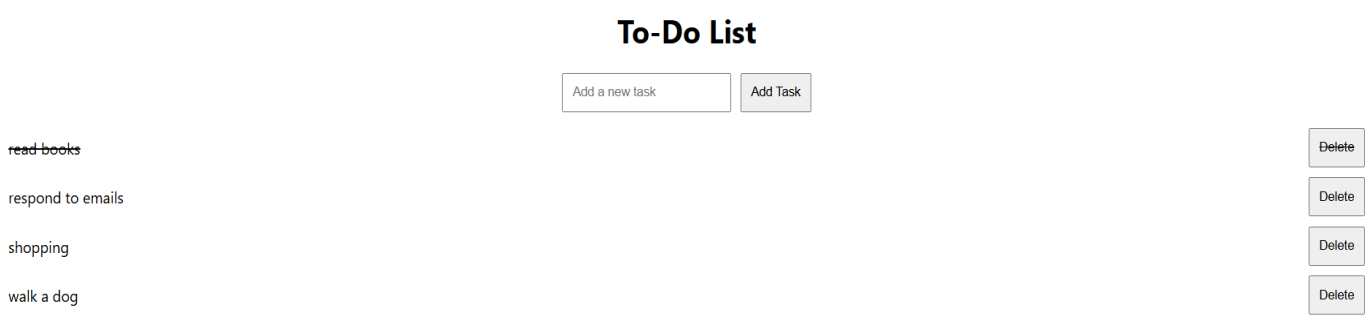
```
 const toggleCompletion = (taskId) => {
    setTasks(
      tasks.map((task) =>
 task.id === taskId ? { ...task, completed: !task.completed } : task
      )
    );
  };
 // Function to handle deleting a task
const deleteTask = (taskId) => {
    setTasks(tasks.filter((task) => task.id !== taskId));
  };
   return (
    <div className="todo-container">
      <h1>To-Do List</h1>
      <div>
        <input
          type="text"
          value={newTask}
          onChange={(e) => setNewTask(e.target.value)}
          placeholder="Add a new task"
        />
        <button onClick={addTask}>Add Task</button>
      </div>
      <ul>
        {tasks.map((task) => (
          <li key={task.id} className={task.completed ? 'completed' : ''}>
```

```
          <span
            style={{
              textDecoration: task.completed ? 'line-through' : 'none',
            }}
            onClick={() => toggleCompletion(task.id)}
          >
            {task.text}
          </span>
          <button onClick={() => deleteTask(task.id)}>Delete</button>
        </li>
      ))}
    </ul>
  </div>
 );
}
function App() {
 return (
  <div className="App">
    <ToDoFunction />
  </div>
 );
}
export default App;
```

App.css

```
.todo-container {
text-align: center;
padding: 20px;
}
input {
padding: 10px;
margin-right: 10px;
}
button {
padding: 10px;
cursor: pointer;
}
ul {
list-style-type: none;
padding: 0;
}
li {
display: flex;
justify-content: space-between;
align-items: center;
margin: 10px 0;
}
.completed {
text-decoration: line-through;
}
```

Output


To-Do List
Add a new task
Add Task
read books
Delete
respond to emails
Delete
shopping
Delete
walk a dog
Delete