|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subject code & Name | : | **CSE304 &**  **Full Stack Development** | Practical | : | **6** | Academic Year | : | **2024-25** |

|  |
| --- |
| **Task - 6** |
| **Aim** |
| Build a To-do List using React Hooks involves creating a user interface that allows users to add, remove, and delete tasks. This can be achieved using React's state management capabilities through hooks like useState and event handling for managing tasks. |
| **Code** |
| Practical\_6.js  import React, { useState } from 'react';  const TodoList = () => {  const [tasks, setTasks] = useState([]);  const [newTask, setNewTask] = useState('');  const addTask = () => {  if (newTask.trim() !== '') {  setTasks([...tasks, newTask]);  setNewTask('');  }  };  const deleteTask = (index) => {  const newTasks = tasks.filter((\_, i) => i !== index);  setTasks(newTasks);  };  const editTask = (index, newTaskValue) => {  const newTasks = tasks.map((task, i) => (i === index ? newTaskValue : task));  setTasks(newTasks);  };  return (  <div className="todo-list">  <h1>Get Things Done!</h1>  <div className="input-container">  <input  type="text"  value={newTask}  onChange={(e) => setNewTask(e.target.value)}  placeholder="What is the task today?"  />  <button onClick={addTask}>Add Task</button>  </div>  <div className="tasks-container">  {tasks.map((task, index) => (  <div key={index} className="task">  <input  type="text"  value={task}  onChange={(e) => editTask(index, e.target.value)}  />  <button onClick={() => deleteTask(index)}>Delete</button>  </div>  ))}  </div>  </div>  );  };  export default TodoList;  App.js  import './App.css';  import TodoList from './components/Practical\_6';  function App() {  return (  <div className="App">  <TodoList/>  </div>  );  }  export default App; |
| **Output** |
|  |

|  |
| --- |
| **Signature :** |