Internship Task Report: React.js Developer

SUBMITTED BY:

- **Intern Name:** Gopinath U
- **Intern ID:** BS/REG/115245
- **Date of Submission:** December 31, 2025
- **Task Number:** 03
- **Task Title:** Todo App with Local Storage

---

1. Project Overview

- **Project Title:** Full-Featured Todo Application
- **Domain:** React.js Development
- **Objective:** To build a functional Todo application that allows users to perform CRUD (Create, Read, Update, Delete) operations with data persistence using the browser's Local Storage.

2. Technologies Used

- **Frontend Library:** React.js (Functional Components)
- **State Management:** React Hooks (useState, useEffect)
- **Styling:** CSS Modules
- **Persistence:** Window LocalStorage API

3. Key Features Implemented

- **Add Task:** Users can input a task and add it to the list.
- **Read/Display:** A dynamic list that renders all saved tasks.
- **Delete Task:** Users can remove specific tasks from the list.
- **Data Persistence:** Integrated useEffect hook to sync the state with localStorage, ensuring data remains available after page refresh.

4. Code Snippets (Core Logic)

ToDoList.jsx

```jsx
import React, { useState } from "react"

function ToDoList() {
    const [tasks, setTask] = useState([]);
    const [newTask, setNewtask] = useState();

    const handleNewTask = (event) => {
        setNewtask(event.target.value);

    }

    const handleAddTask = () => {
        if (newTask.trim()) {
            setTask([...tasks, newTask]);
        }
        setNewtask("");

    }

    const handleDeleteTask = (index) => {
        setTask(tasks.filter((_, i) => i != index));
    }

    const handleMoveUp = (index) => {
        if (index > 0) {
            const newTasks = [...tasks];
            [newTasks[index], newTasks[index - 1]] = [newTasks[index - 1], newTasks[index]];
            setTask(newTasks);
        }
    };

    const handleMoveDown = (index) => {
        if (index < tasks.length - 1) {
            const newTasks = [...tasks];
            [newTasks[index], newTasks[index + 1]] = [newTasks[index + 1], newTasks[index]];
            setTask(newTasks);
        }
```

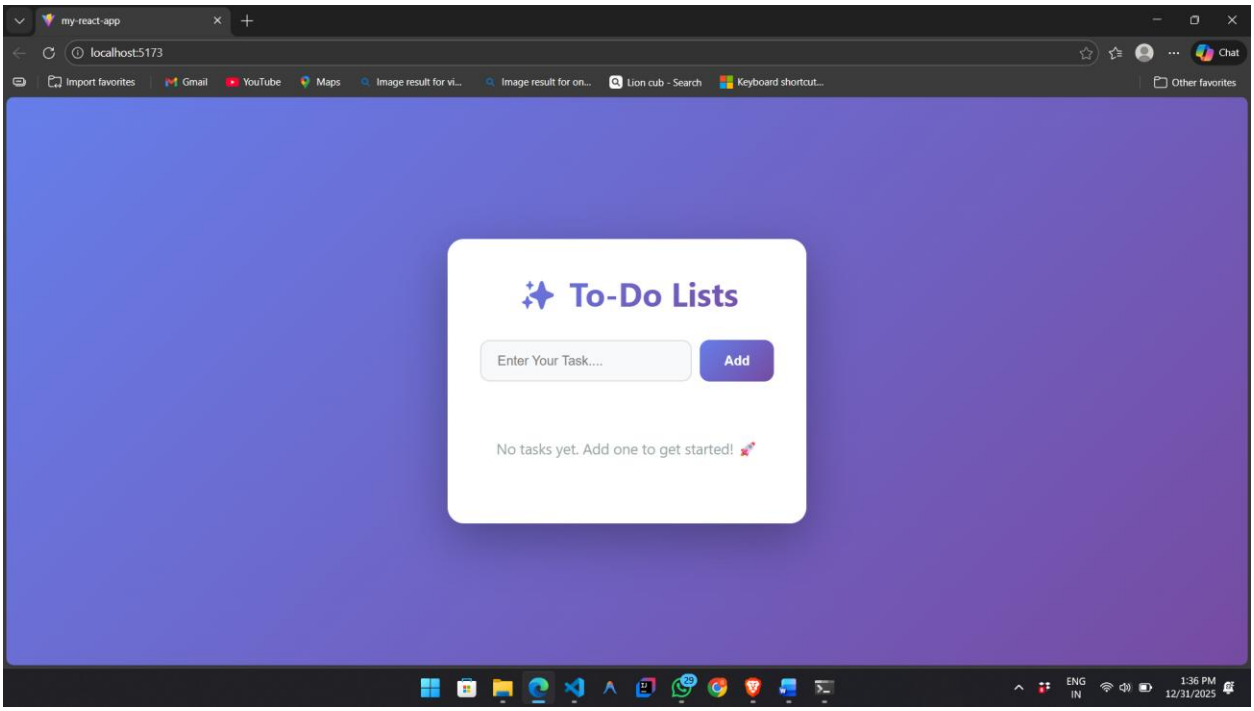
Html (Return Module From React Function)

ToDoList.jsx

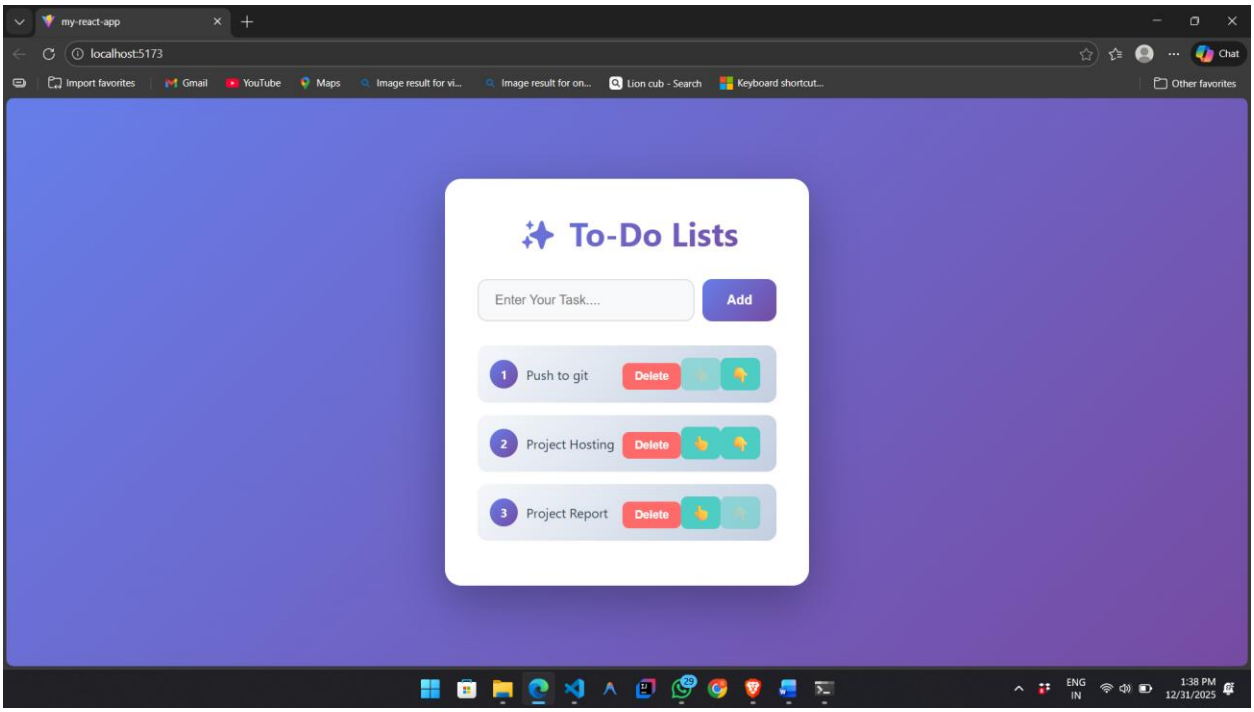
```jsx
function ToDoList() {
            <div className="listContainer">
                <h1>✨ To-Do Lists</h1>

                <div className="input-section">
                    <input
                        type="text"
                        value={newTask}
                        id="inp"
                        placeholder="Enter Your Task...."
                        onChange={handleNewTask}
                        onKeyPress={(e) => e.key === 'Enter' && handleAddTask()}
                    />

                    <button className="add-btn" onClick={handleAddTask}>Add</button>
                </div>

                <div className="tasks">
                    {tasks.length === 0 ? (
                        <div className="empty-state">No tasks yet. Add one to get started! 🚀</div>
                    ) : (
                        <ol>
                            {tasks.map((ele, index) => (
                                <li key={index}>
                                    <span>{ele}</span>
                                    <div>
                                        <button
                                            className="task-btn delete-btn"
                                            onClick={() => handleDeleteTask(index)}
                                        >
                                            Delete
                                        </button>
                                        <button
                                            className="task-btn move-btn"
                                            onClick={() => handleMoveUp(index)}
                                            disabled={index === 0}
                                        >
```

Ln 16, Col 24 Spaces: 4 UTF-8 CRLF { } JavaScript JSX Go Live

5. Screenshots



1. **Initial UI:** Screenshot of the empty list



2. **Active Tasks:** Screenshot with multiple todos added

6. Project Links

- **GitHub Repository:** https://github.com/gopinath2207/todoUsingReact
- **Live Hosted Link:** https://commit-rank-tasks.netlify.app/

7. Conclusion

In this project, I successfully implemented a React application that handles complex state updates and persistent data. This task strengthened my understanding of the React lifecycle and how to build user-friendly, data-driven interfaces.

Declaration:
I, Gopinath U, confirm that this project was developed by me as part of the React.js Developer Internship at Alfido Tech.