A blue text on a black background

Description automatically generatedA pink text on a black background

Description automatically generated

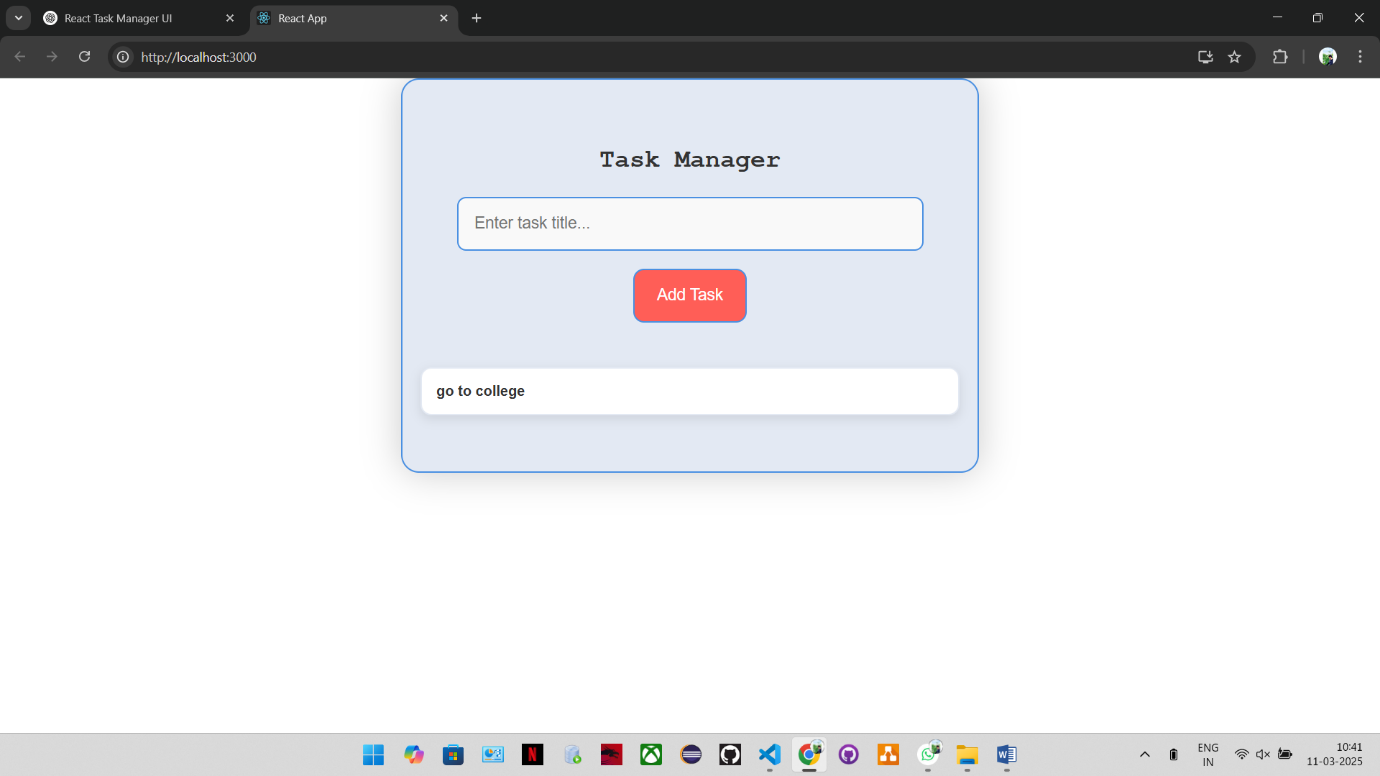
**React Hooks Assignment**

Name: - Pruthvi Gadhiya

Id no.:-500235241

Git-hub link:- <https://github.com/Pruthvi1881/React-Hooks>

Step-1:- useState



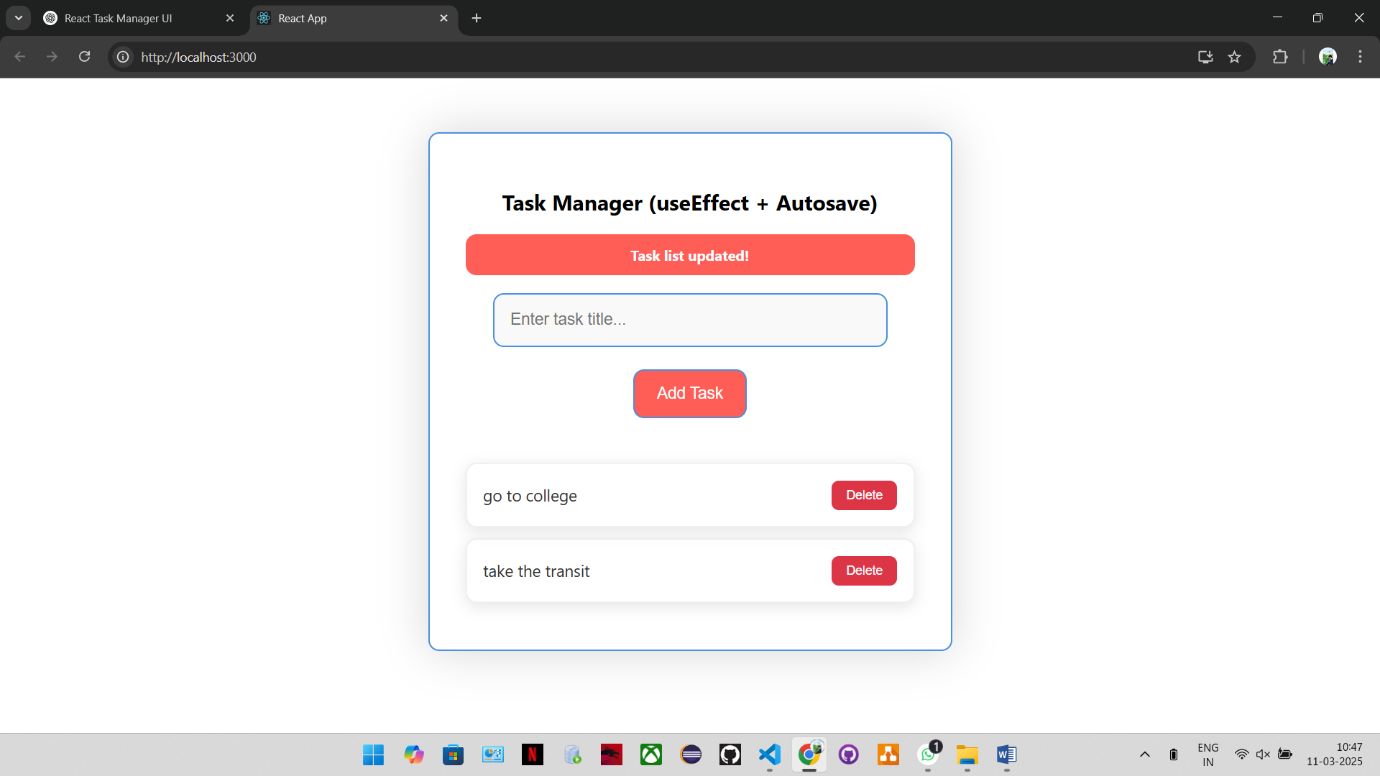
**Explanation:**

We need to create a form where users can enter a task with a title and description. We'll use useState to store the list of tasks.

**Code Snippet:**

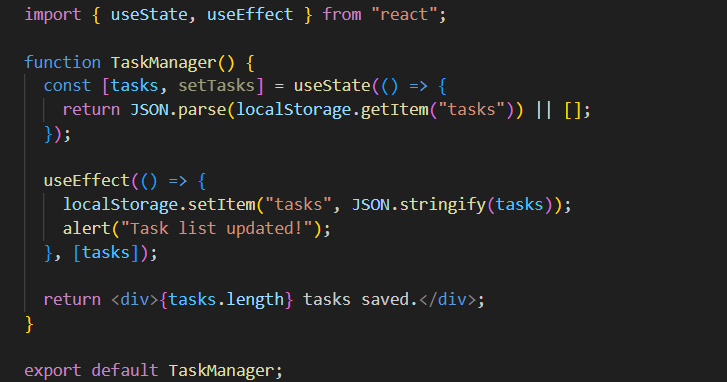


Step-2:- useEffect

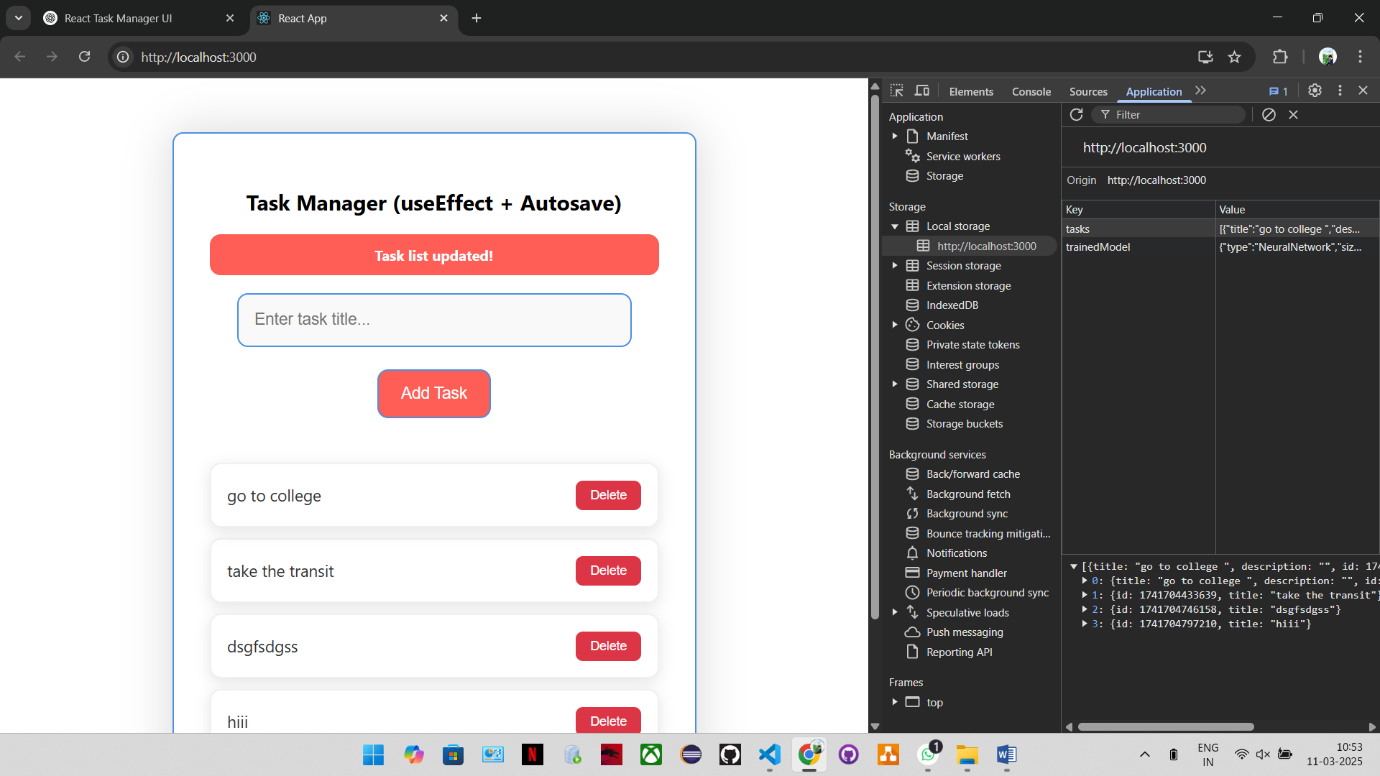


**Explanation:**

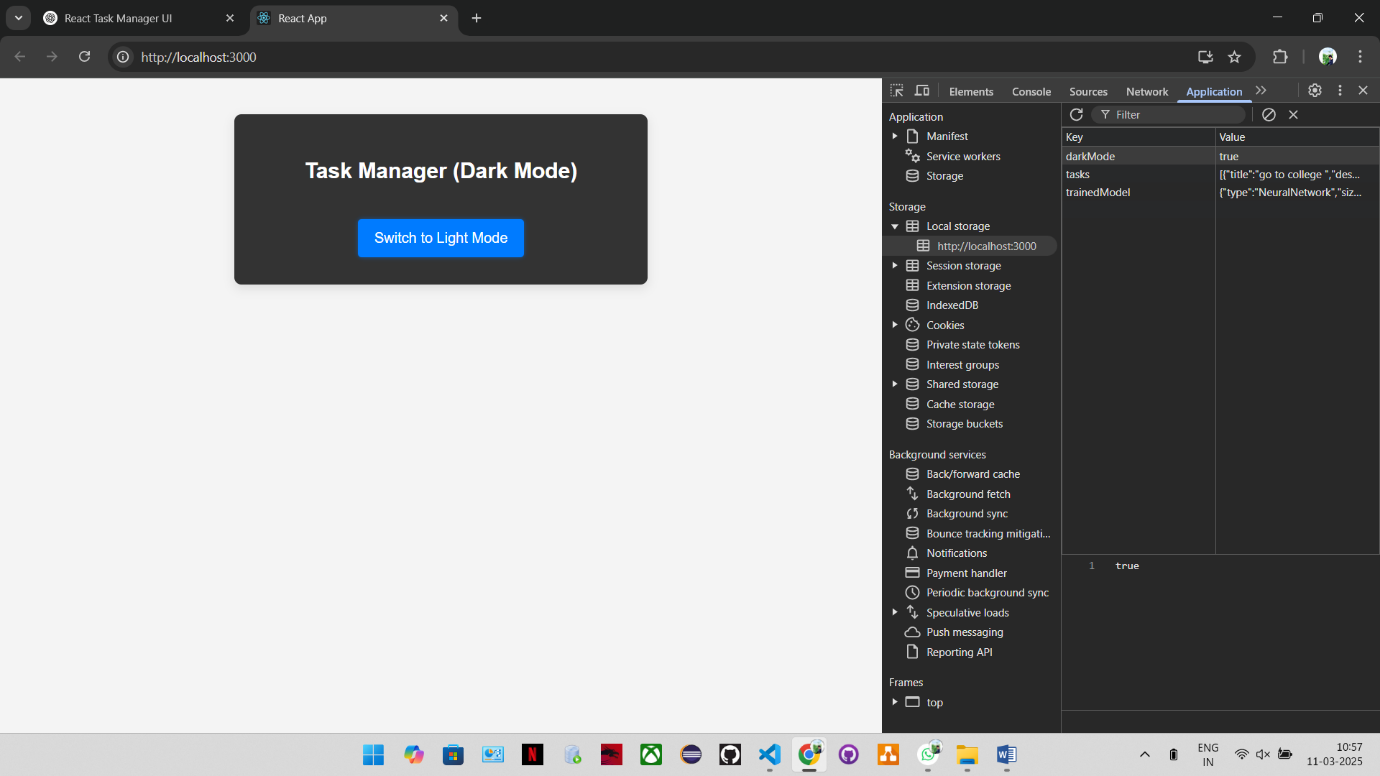
Whenever a task is added or removed, we should save the updated list to localStorage. Also, we'll show an alert whenever the task list changes.

**Code Snippet:**

Local storage:-



Step-3:- useContext



Explanation:

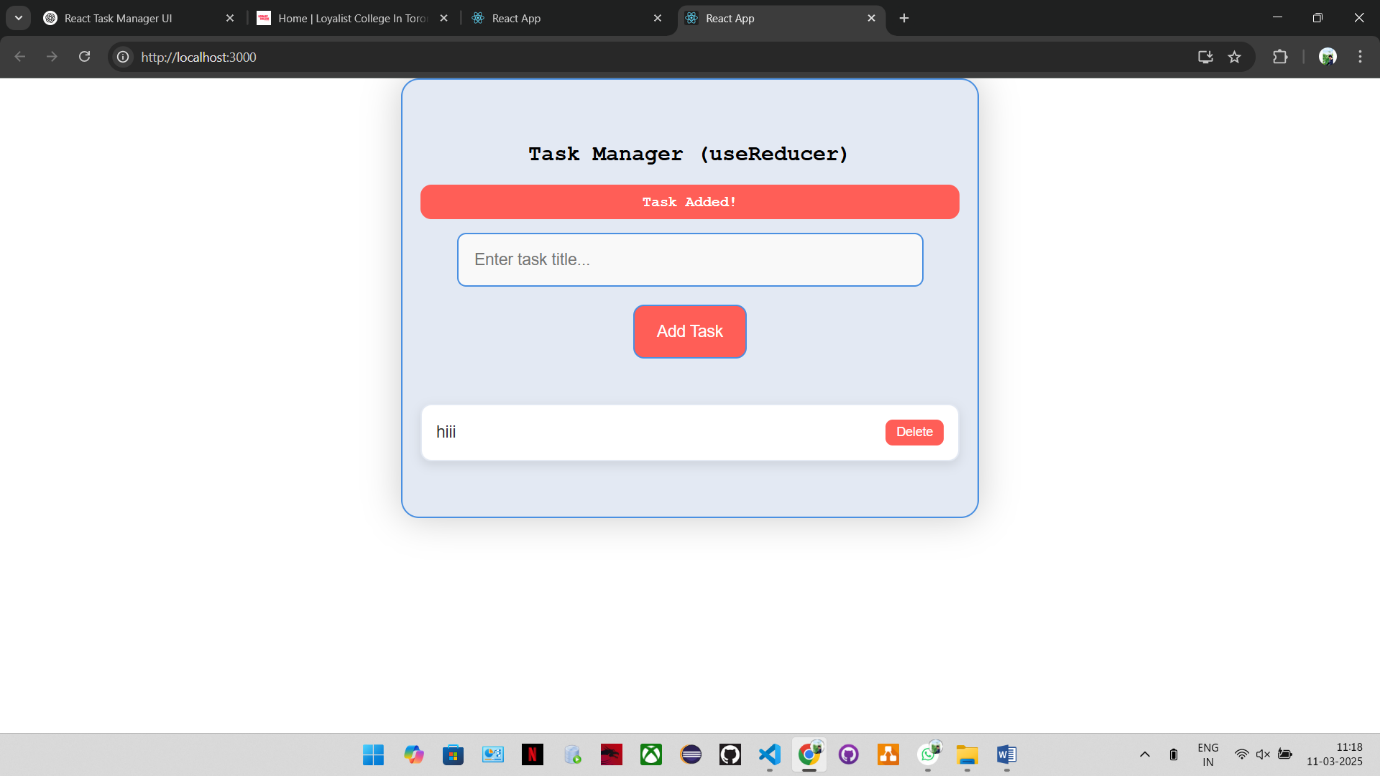
We'll create a ThemeContext to switch between dark and light mode. The theme should be remembered even after the page reloads.

**Code Snippet:**

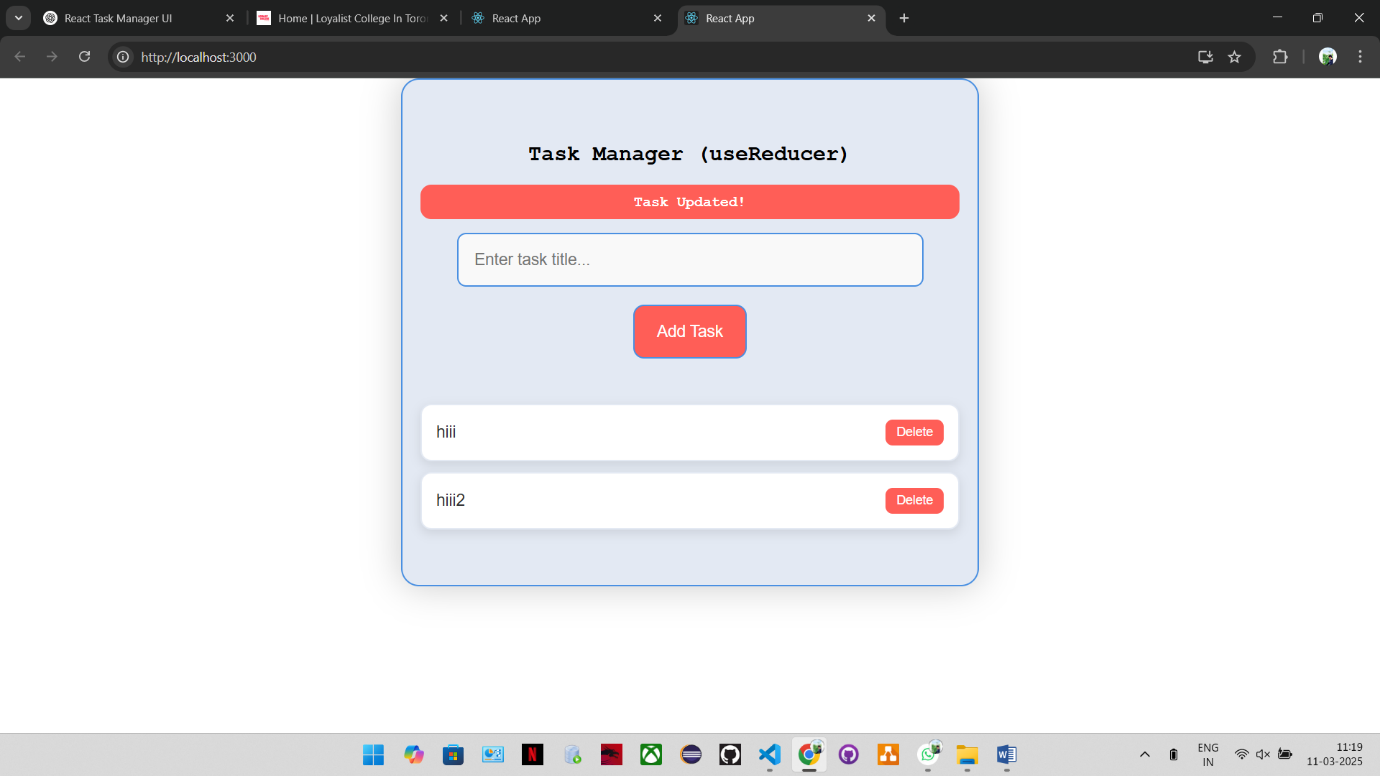


Step-4:- useReducer

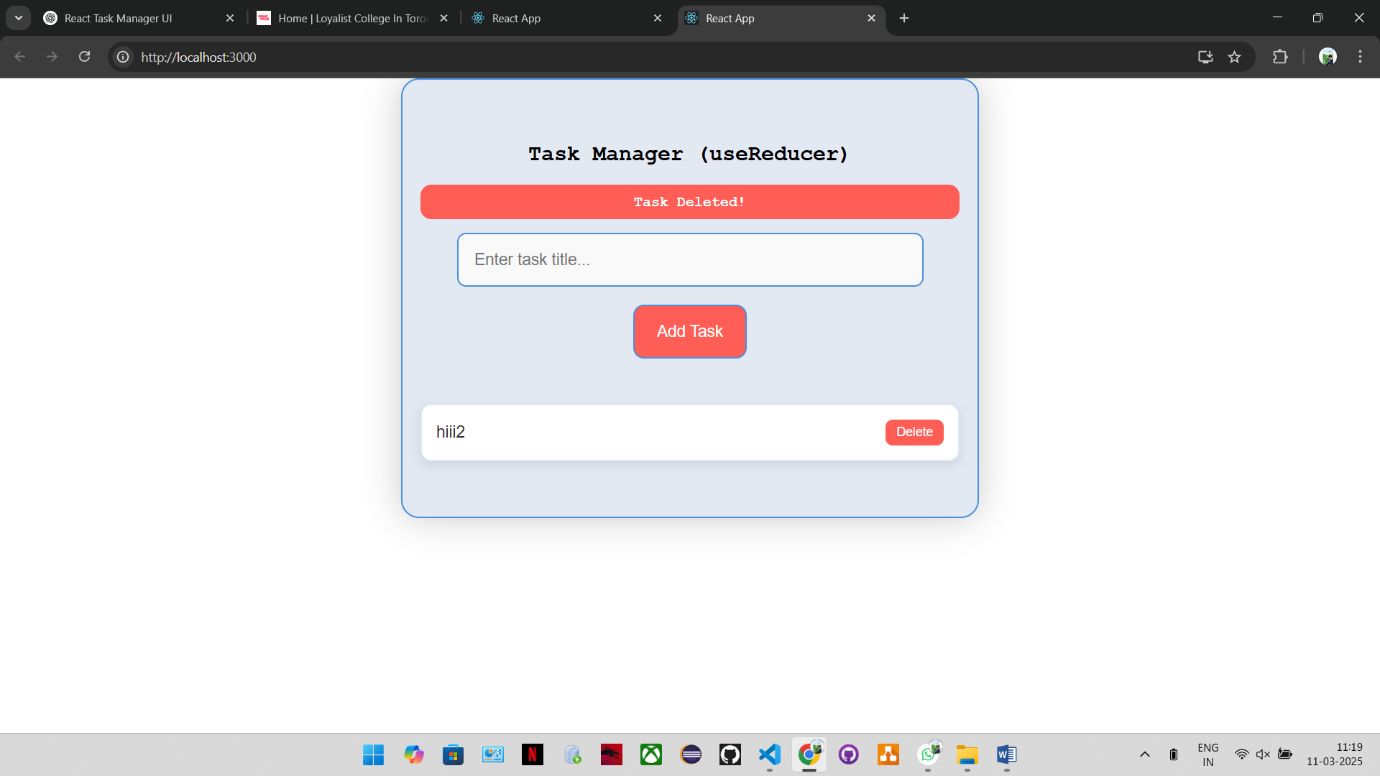
1-task\_added



2-task\_updated



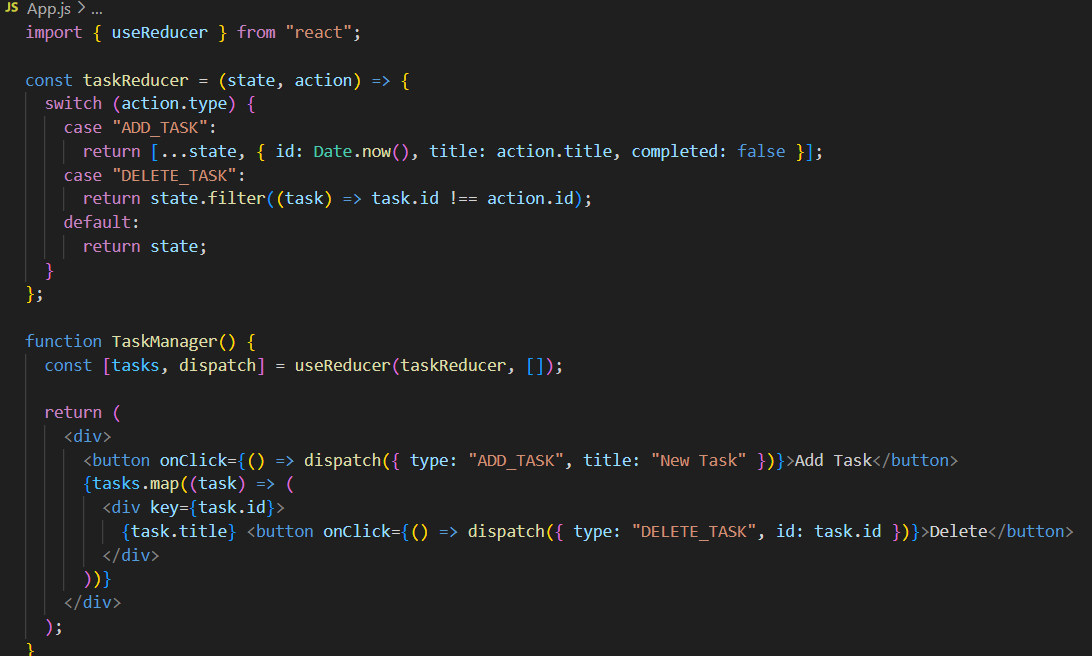
3-task\_deleted



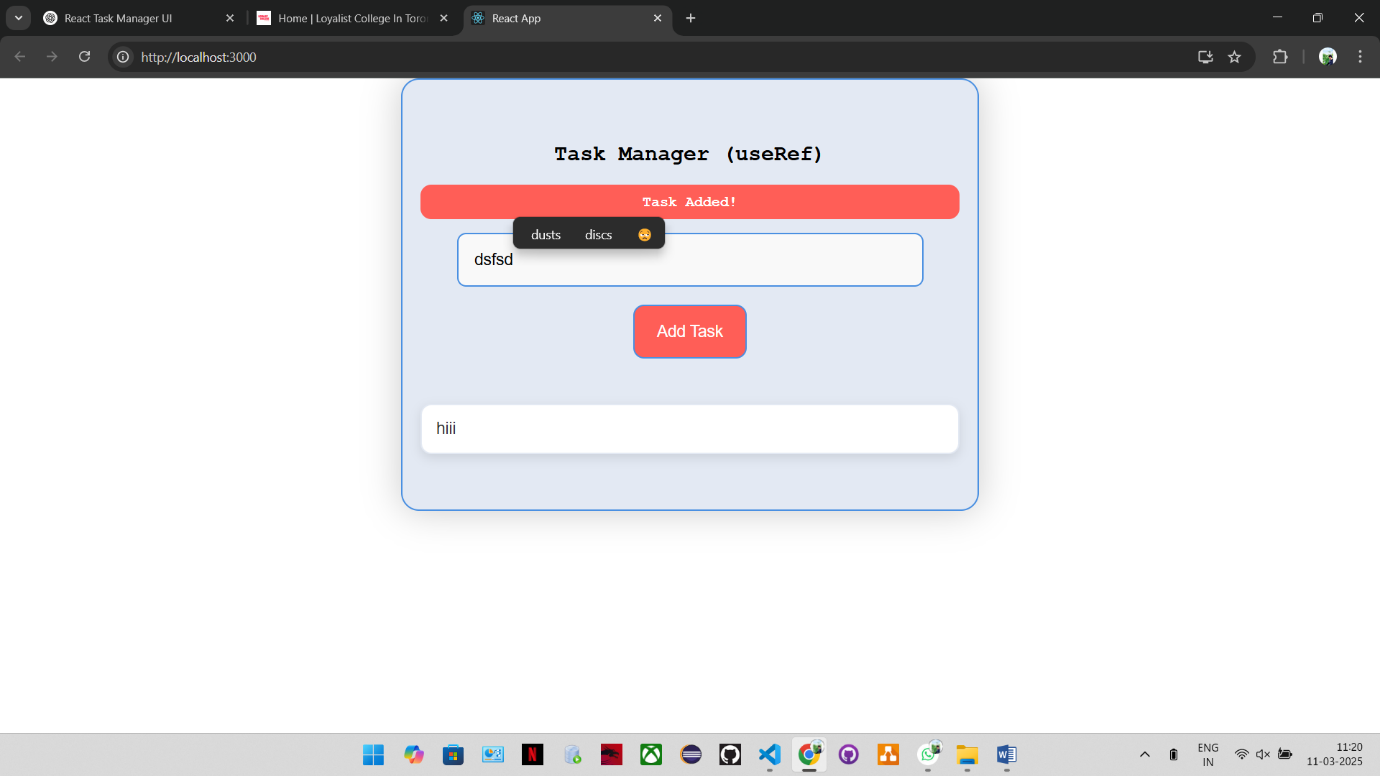
Explanation:

Instead of useState, we'll use useReducer to manage adding, updating, and deleting tasks in a structured way.

**Code Snippet:**



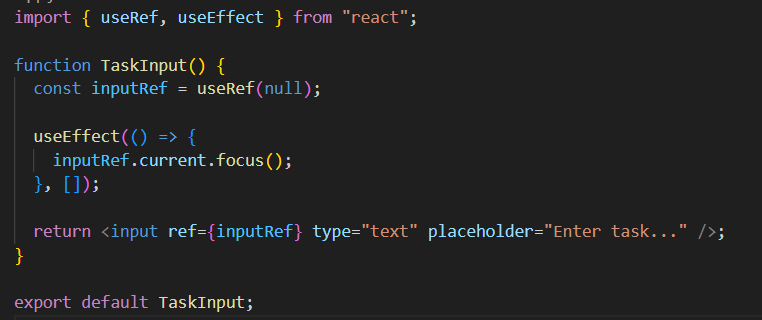
Step-5:- useRef



Explanation:

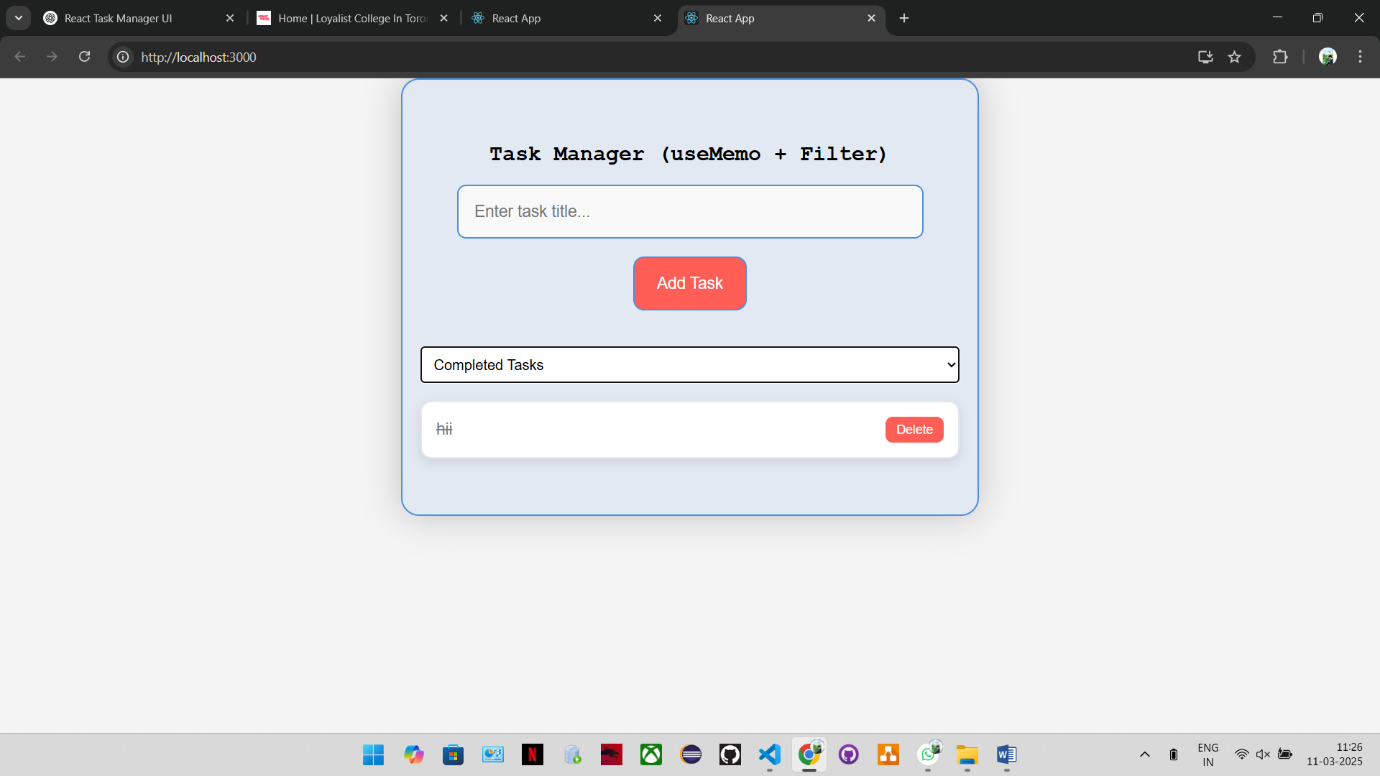
We'll use useRef to store a reference to the task input field so it gets auto-focused when the component loads.

**Code Snippet:**

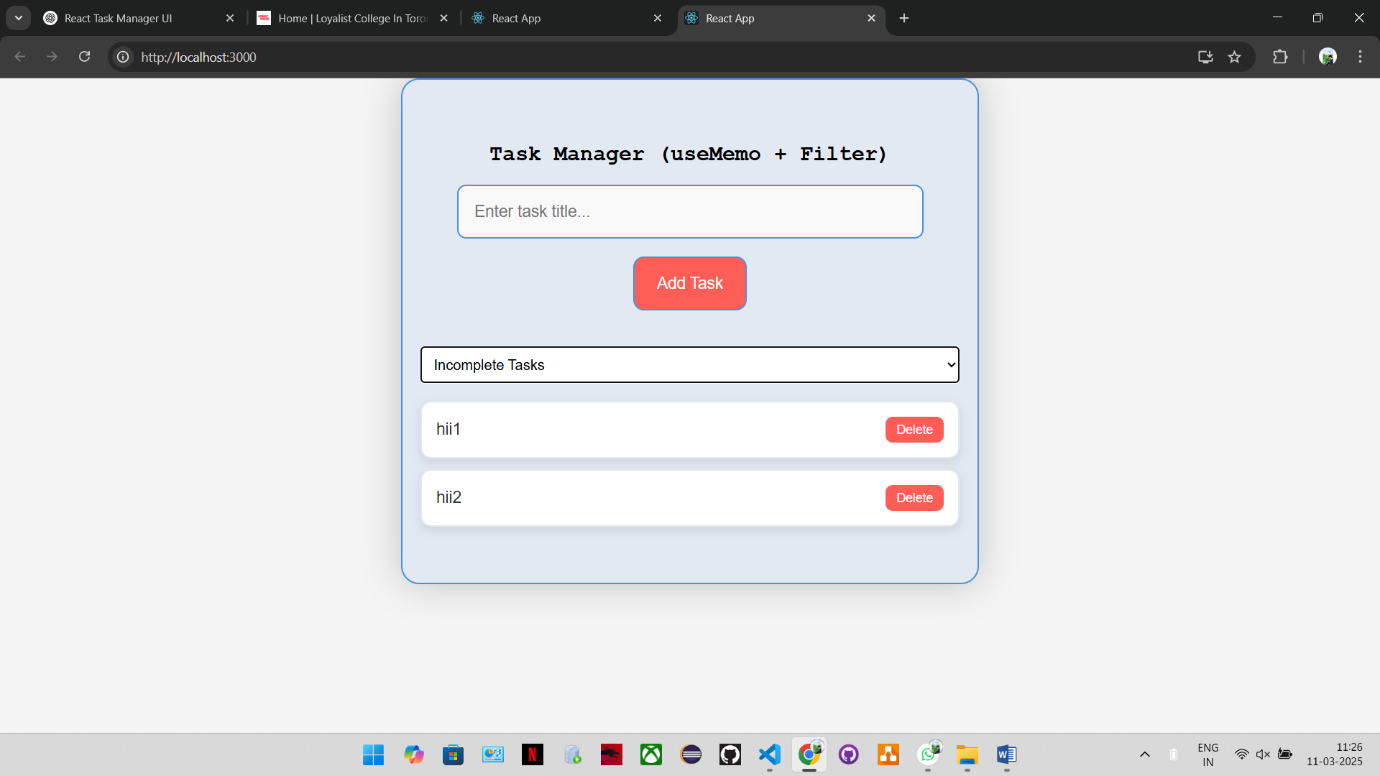


Step-6:- useMemo

1-completed\_task\_filter



2-incompleted\_task\_filter



Explanation:

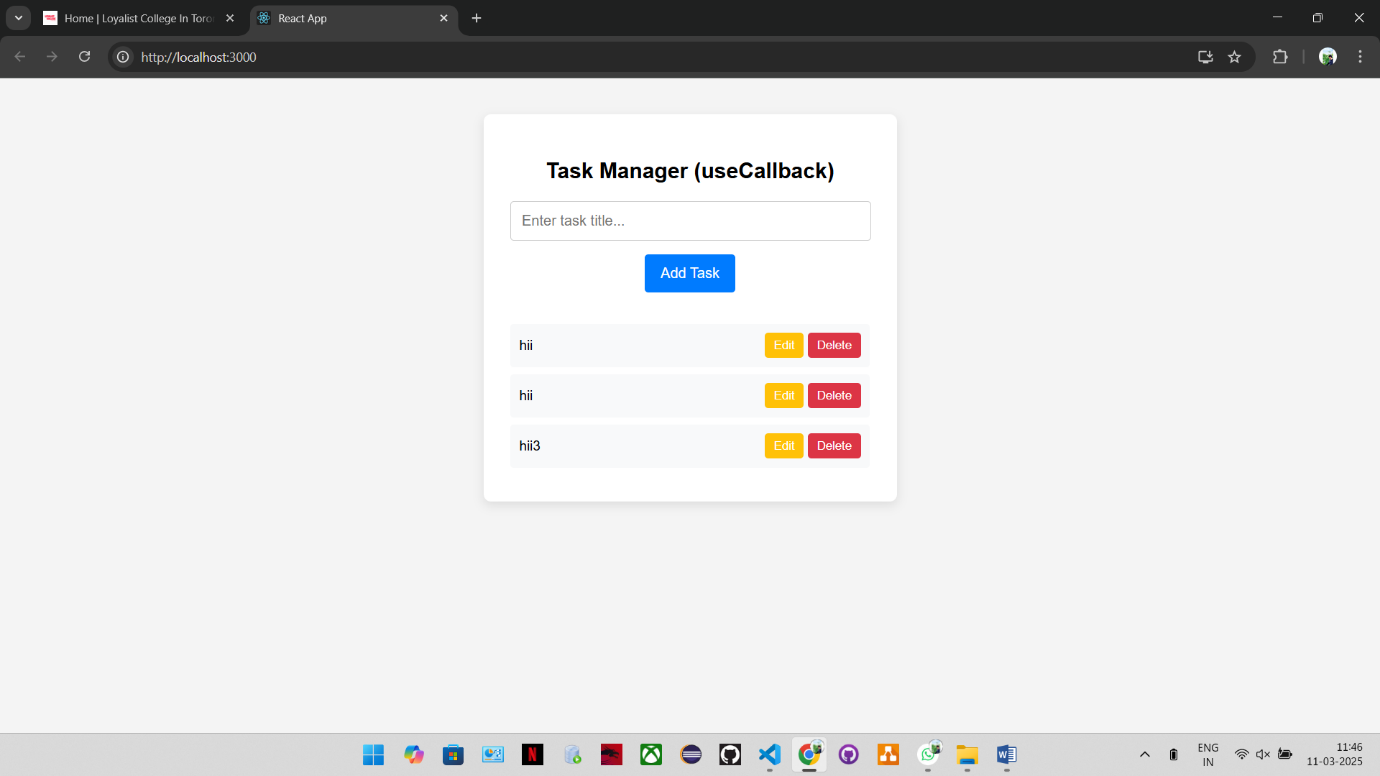
Filtering tasks can be slow if there are too many. We'll use useMemo to optimize it and prevent unnecessary calculations.

**Code Snippet:**

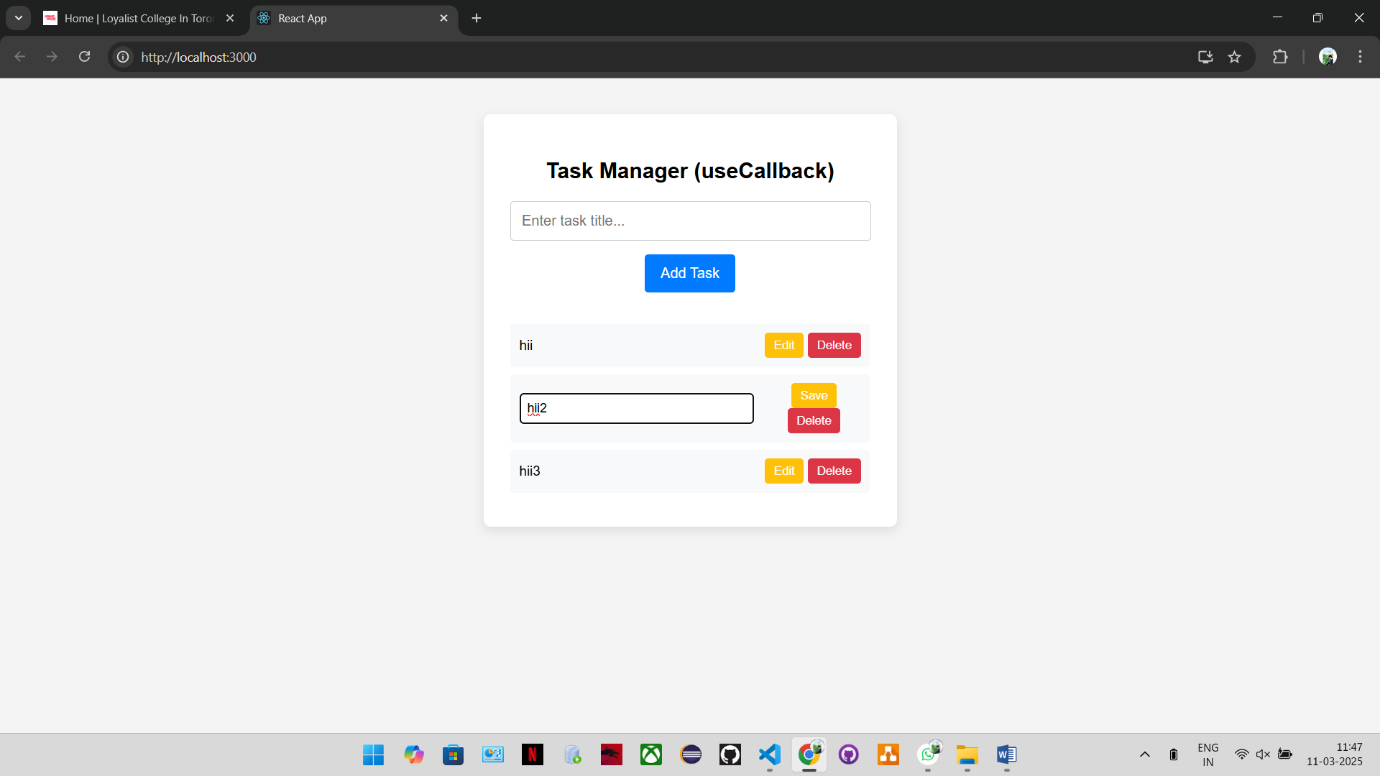


Step-7:- useCallback

1-added\_tasks



2-edited\_task



Explanation:

If a function is created every time the component re-renders, it can slow down performance. We'll use useCallback to optimize event handlers.

**Code Snippet:**

