React Hook Exercises

This document provides exercises to practice important React hooks: useRef, useMemo, useCallback, and custom hooks. Each exercise is designed to help you understand the purpose and usage of these hooks in real-world scenarios.

# 1. useRef

- Build a form with an <input> for typing a name.  
- Add a 'Focus Input' button. When clicked, the input should get focus without re-rendering the component.  
- Bonus: Count how many times the component re-renders using another useRef.  
  
Goal: learn how useRef stores DOM nodes and persistent values across renders.

# 2. useMemo

- Create a component with a large array of numbers.  
- Add a filter input (<input type='number'>) to display numbers greater than the input value.  
- Use useMemo to memoize the filtered result so it doesn’t recompute on every keystroke unrelated to filtering.  
- Add a counter that updates with a button (to show unrelated re-renders don’t recalc the filter).  
  
Goal: prevent expensive recalculations.

# 3. useCallback

- Build a parent component with a list of items and a child component <Item> that takes onSelect(item) as a prop.  
- Without useCallback, the child re-renders unnecessarily whenever the parent re-renders.  
- Use useCallback to memoize the function and prevent unnecessary re-renders.  
- Verify using React.memo on the child.  
  
Goal: stabilize function references passed to children.

# 4. Custom Hook

- Create a custom hook useLocalStorage(key, initialValue) to store and retrieve data in localStorage.  
- Use it in a component to:  
 • Save a theme preference (dark/light).  
 • Persist the preference even after reload.  
- Bonus: Combine it with useEffect and useState.  
  
Goal: learn to encapsulate reusable logic into hooks.

# 5. Final Combo Exercise

Make a Todo List App:  
- Add todos (useState).  
- Store todos in localStorage using your useLocalStorage custom hook.  
- Use useCallback for handlers like addTodo, deleteTodo.  
- Use useMemo to show counts (e.g., number of completed vs pending tasks).  
- Use useRef to:  
 • Focus input on mount.  
 • Track how many times a todo has been edited (without re-render).  
  
Goal: combine all 4 concepts in a real-world mini app.