React Hooks Practice Sheet

This practice sheet focuses on five React hooks: useContext, useReducer, useMemo, useCallback, and Custom Hooks.

The tasks will help you understand how to use these hooks in real-world scenarios. Please complete each task and try to implement them in your projects.

# ✅ 1. useContext Practice

Objective: Avoid prop drilling by using context to pass theme or user data.

Task: Create a theme toggler with Light and Dark modes using useContext.

Steps:

- Create a ThemeContext.  
- Provide the theme value using ThemeProvider.  
- Consume it in a component and change the background color.

# ✅ 2. useReducer Practice

Objective: Manage more complex state transitions.

Task: Create a counter app with INCREMENT, DECREMENT, and RESET actions using useReducer.

Bonus: Add a second counter using the same reducer.

# ✅ 3. useMemo Practice

Objective: Optimize performance by memoizing an expensive calculation.

Task: Create a factorial calculator:

- Add an input field to enter a number.  
- Calculate the factorial of that number using a heavy recursive function.  
- Use useMemo to prevent unnecessary recalculations.

# ✅ 4. useCallback Practice

Objective: Prevent unnecessary re-renders in child components.

Task: Create a list of numbers.

- Add a button that sorts them when clicked.  
- Pass the sorting function to a child component.  
- Use useCallback to ensure the function isn’t recreated on each render.

# ✅ 5. Custom Hook Practice

Objective: Extract reusable logic into a custom hook.

Task: Build a custom hook called useWindowWidth:

- It should return the current window width.  
- Use resize event listener inside useEffect.  
- Display the width in a component.

Bonus: Create another custom hook like useFetch(url) to fetch data from an API and return loading, data, and error.