

useEffect() -> fn -> side effects -> jo kaam rendering ke dauran nhi ho sakte -> setTimeout , setInterval , fetch , axios , backend , alert etc.

useEffect( function(){state/prop} , [depdancy array] )

Whenever there was a change in state/prop -> leads to rerendering

Memo -> memoization -> prop change -> rerender ✓

useMemo() -> hook -> memoiztion of values

-> n =100

1 to n -> loop -> output => 5050 value

somehow -> state chage ->

compoent rereder -> loop firse evaluate ✓

stateChage -> repeat(loop) => loop heavy ✓ , avoid ✓

no need to do baar baar ✗ -> yaad rakhlu -> output value ✓

=> remember the **value**

Inp - > 200 => evalute

useCallback() -> hook -> memoiztion of functions

-> relies on the referencial equality -> unnecessary renders avoid ✓

dependacnyc array -> state/prop -> chage hone par firse chalo

useState , useEffect , useCallback , useMemo , memo ,

useRef , prop drilling

contextapi

React-router-dom

React-icons

Project x 2 -> only frontend , backend + front (integration)

