**LIST**

**Question 1: How do you render a list of items in React? Why is it important to use keys when rendering lists?**

**Question 2: What are keys in React, and what happens if you do not provide a unique key?**

**Hook**

**1: What are React hooks? How do useState() and useEffect() hooks work in functional components?**

=>React Hooks are special functions that let you use React features (like state and lifecycle methods) inside functional components. They make your code cleaner and easier to read by avoiding class components.

**useState()**

* **Purpose:** Manages state in functional components.
* **How it works:** It declares a state variable and a function to update it.
* **useEffect()**
* **Purpose:** Handles side effects like data fetching, subscriptions, or manually changing the DOM.
* **How it works:** Runs a function after rendering. You can control when it runs using a dependency array.

**2: What problems did hooks solve in React development? Why are hooks considered an important addition to React?**

 **Complex State Management** → Simplified with useState() and useReducer().

 **Reusability of Logic** → Achieved using custom hooks.

 **Lifecycle Confusion** → Replaced with clear useEffect() usage.

 **this Keyword Confusion** → Removed entirely in functional components.

 **Large Components** → Made modular and easier to manage.

**Before Hooks:** Class components required the use of this, leading to confusion and bugs if not bound correctly.

**With Hooks:** Functional components with hooks don’t use this, making the code cleaner and easier to maintain.

**3: What is useReducer ? How we use in react app?**

useReducer() is a react hook use to manage complex state logic in functional component.

HOW WE USE

First we import useReducer()

Second we define it

Third we define initial state outside the component

**4: What is the purpose of useCallback & useMemo Hooks?**

useCallback() → Caches functions to prevent re-creation on every render.

useMemo() → Caches calculated values to optimize performance.

**5: What’s the Difference between the useCallback & useMemo Hooks?**

| **Aspect** | **useCallback()** | **useMemo()** |
| --- | --- | --- |
| **Purpose** | Memoizes a function | Memoizes a calculated value |
| **Returns** | A **memoized function** | A **memoized value** |
| **Use Case** | When passing functions as props to child components to prevent re-renders | When performing expensive calculations that shouldn't run on every render |
| **Example** | const memoizedFn = useCallback(() => {...}, [deps]); | const memoizedValue = useMemo(() => {...}, [deps]); |
| **Performance** | Prevents function re-creation | Avoids re-calculating values |

**6 : What is useRef ? How to work in react app?**

**useRef** is a React Hook that lets you reference a value that’s not needed for rendering.

 **For DOM Manipulation:** Attach ref to a DOM element and access it using .current.

 **For Storing Values:** Store any value in .current without triggering re-renders.

* **API**

Question 1: What is the Context API in React? How is it used to manage global state across multiple components?

Question 2: Explain how createContext() and useContext() are used in React for sharing state.