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

(ans) hooks unable you to extract stateful logic from a functional component, allowing it to be reused and tested independently. Usestate and useeffect are two of react’s most useful hooks. The usestate hook in functional components serves the same purpose as this.state in the class component. We can use useeffect hook to perform side effect in function component.

1. What problems did hooks solve in React development? Why are hooks considered an important addition to React?

(ans) managing state and lifecycle methods in class components can sometimes lead to complex and hard to understand code. Hooks provide a more straight forward way to manage this aspects without the need for classes.

1. What is useReducer ? How we use in react app?

(ans) useReducer() gives access to data and actions defined in a react reducer. It exposes the state for a component to consume, and a dispatch() function to invoke actions that alters the state.

1. What is the purpose of useCallback & useMemo Hooks?

(ans) the useCallback & useMemo hooks in react are used to optimize application performance by values and functions

Usememo: caches the result of the function, so it only needs to be recalculated when the dependency change. This can help with expensive calculation or data transformation.

Usecallback: caches a function , so it does not need to be recreated on every re render. This can help reduce unnecessary re recreation of functions and optimize component re renders.

1. What’s the Difference between the useCallback & useMemo Hooks?

(ans) Differences between the useCallback & useMemo:

Usememo caches the return value of a function. Usecallback caches the function definition itself. Usememo is used when you have an expensive calculation you want to avoid on every render. Usecallback is used to cache a function to avoid re-creating it on every re render.

1. What is useRef ? How to work in react app?

(ans) useRef is a hook in react, a javascript library for building user interfaces. It returns a mutable reb object whose. Current property is initially set to the passed in argument. The returned ref object persists across re renders of the component.

Routing in React (React Router)

1. What is React Router? How does it handle routing in single-page applications?

(ans) React Router is a library for handling routing and navigation in react js applications. It allows you to create dynamic routes providing a seamless user experience by mapping various urls to components. It enables navigation in single page application without refreshing the entire page.

1. Explain the difference between BrowserRouter, Route, Link, and Switch components in React Router.

(ans) <BrowserRouter> : BrowserRouter is a parent component in react-router-dom that stores all the other route components. Allowing the declaration of individual routes is the main functionality of using BrowserRouter in the application.

<Route>: a child component that renders a specific ui component when the url matches the specified path.

<Link>: allows a user to navigate to another page by clicking on it.

<switch>: displays the first route whose path matches the current url.