1. What is React Hooks?

React Hooks are the built-in functions that permit developers for using the state and lifecycle methods within React components. These are newly added features made available in React 16.8 version. Each lifecycle of a component is having 3 phases which include mount, unmount, and update. Along with that, components have properties and states. Hooks will allow using these methods by developers for improving the reuse of code with higher flexibility navigating the component tree.

Using Hook, all features of React can be used without writing class components. For example, before React version 16.8, it required a class component for managing the state of a component. But now using the useState hook, we can keep the state in a functional component.

1. What are props in React?

The props in React are the inputs to a component of React. They can be single-valued or objects having a set of values that will be passed to components of React during creation by using a naming convention that almost looks similar to HTML-tag attributes. We can say that props are the data passed from a parent component into a child component.

1. Explain about types of side effects in React component.

There are two types of side effects in React component. They are:

Effects without Cleanup: This side effect will be used in useEffect which does not restrict the browser from screen update. It also improves the responsiveness of an application. A few common examples are network requests, Logging, manual DOM mutations, etc.

4. What is prop drilling in React?

Sometimes while developing React applications, there is a need to pass data from a component that is higher in the hierarchy to a component that is deeply nested. To pass data between such components, we pass props from a source component and keep passing the prop to the next component in the hierarchy till we reach the deeply nested component. The disadvantage of using prop drilling is that the components that should otherwise be not aware of the data have access to the data.

5.What are error boundaries?

Introduced in version 16 of React, Error boundaries provide a way for us to catch errors that occur in the render phase.

What is an error boundary?

Any component which uses one of the following lifecycle methods is considered an error boundary. In what places can an error boundary detect an error?

1. Render phase

2. Inside a lifecycle method

3. Inside the constructor

6. What are the rules that must be followed while using React Hooks?

There are 2 rules which must be followed while you code with Hooks: React Hooks must be called only at the top level. It is not allowed to call them inside the nested functions, loops, or conditions. It is allowed to call the Hooks only from the React Function Components.

7. Why do React Hooks make use of refs?

Earlier, refs were only limited to class components but now it can also be accessible in function components through the useRef Hook in React. The refs are used for:

Managing focus, media playback, or text selection.

Integrating with DOM libraries by third-party.

Triggering the imperative animations.