1. How are events handled in React compared to vanilla JavaScript? Explain the concept of synthetic events.

(ans) the react event handling system is known as synthetic events. The synthetic event is a cross browser wrapper of the browsers native event. React events are named as camel case instead of lower case. With jsx a function is passed as the event handler instead of a string.

1. What are some common event handlers in React.js? Provide examples of onClick, onChange, and onSubmit.

(ans) commonly used react event handlers onclick, onchange, and onsubmit

Onclick used for handling click events on elements like buttons, links, etc

Onchange used for handling input changes such as typed text or selected option values.

Onsubmit used for handling form submissions

1. Why do you need to bind event handlers in class components?

(ans) you need to bind event handlers in class components in react to ensure that the this keyword within the event handler function refers to the component instance. Without binding, the event handler method loses its context. When the event occurs, the this value is set to undefined.

Conditional Rendering

1. What is conditional rendering in React? How can you conditionally render elements in a React component?

(ans) conditional rendering in react works the same way conditions work in javascript. Use javascript operators like if or the conditional operator to create elements representing the current state, and let react update the UI to match them. This example renders a different greeting.

1. Explain how if-else, ternary operators, and && (logical AND) are used in JSX for conditional rendering.

(ans) you can use if-else, ternary operators, and the logical AND (&&) operator for conditional rendering to create dynamic user interface.

Ternary operator : similar to the if-else operator, the ternary operator is used to render components based on a condition.

Logical AND (&&) operator : used to render jsx when a condition is true, or nothing otherwise.

If-else : can be used in an inline if else with the conditional operator.

Lists and Keys

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

(ans) key should be given to the elements inside the array to give the elements a stable identity

Const number =[1,2,3,4,5]; const listitems = numbers.map ((number)=><li key={number.tostring()}>{number}</li>) the best way to pick a key is to use a string that uniquely identifies a list items among its siblings.

1. What are keys in React, and what happens if you do not provide a unique key?

(ans) react uses the unique key value of each list item to keep track of it. If you neglect to add them, react uses the index value of each item as a default. Although this sounds like logical solution, the react docs explain that this can lead to performance issues.

Forms in React

1. How do you handle forms in React? Explain the concept of controlled components.

(ans) controlled components are those in which forms data is handled by the components state. It takes its current value through props and makes changes through callbacks like onclick, onchange, etc. a parent component manages its own state and passes the new values as props to the controlled component.

1. What is the difference between controlled and uncontrolled components in React?

(ans) controlled components are tightly coupled with react’s state management. Any changes to the component’s state are reflected immediately in the component’s UI. Uncontrolled components maintain their state independently of react’s state management .

Lifecycle Methods (Class Components)

1. What are lifecycle methods in React class components? Describe the phases of a component’s lifecycle.

(ans) a react component has three different phases in its lifecycle, including mounting, updating, and unmounting. Each phase has its own methods which are responsible for a particular stage in a component’s lifecycle. These methods are for class-based components, functional component has their own life cycle methods.

1. Explain the purpose of componentDidMount(), componentDidUpdate(), and componentWillUnmount().

(ans) lifecycle methods like componentDidMount, componentDidUpdate(), and componentWillUnmount() are commonly used in class – based components for managing the component’s lifecycle and performing side effects.