Answer the following

What are components, props and useState in React? Explain different types of components with example.

Ans –

1. Components

Components are the building blocks of a React application. They let you split the UI into independent, reusable pieces, making code easier to manage and reuse. Components can be thought of as JavaScript functions that return JSX (JavaScript XML).

Types of Components:

- **Functional Components:** These are JavaScript functions that accept props as an argument and return JSX. Functional components are simpler and focus on rendering UI.

```
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

- **Class Components:** These are ES6 classes that extend the React.Component class. They include a render() method to return JSX and can also manage state and lifecycle methods.

```
class Greeting extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}!</h1>;
  }
}
```

2. Props (Properties)

Props are inputs to React components. They are passed from parent components to child components as an object and are read-only, meaning they cannot be modified by the receiving component.

Example:

```
function Welcome(props) {
  return <h1>Welcome, {props.user}!</h1>;
}

function App() {
  return <Welcome user="Alice" />;
}
```

Here, user="Alice" is the prop passed to the Welcome component.

3. useState Hook

useState is a React Hook that allows functional components to manage state. It returns an array with two values: the current state and a function to update it.

Syntax:

```
const [state, setState] = useState(initialState);
```

Example:

```
import React, { useState } from "react";

function Counter() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

Types of Components:

1. **Stateless Components:** These components do not manage their own state and rely entirely on props for data.

   ```
   function Stateless(props) {
    return <h1>{props.message}</h1>;
   }
   ```

2. **Stateful Components:** These components manage their own internal state.

   ```
   import React, { Component } from "react";


   class Stateful extends Component {
    constructor() {
     super();
     this.state = { count: 0 };
    }


    render() {
     return (
       <div>
         <p>Count: {this.state.count}</p>
         <button onClick={() => this.setState({ count: this.state.count + 1
   })}>
           Increment
         </button>
       </div>
     );
    }
   ```

```
}
```

3. **Presentational Components:** Focus on UI rendering and do not handle logic or state (similar to stateless components).

```
function Presentational() {
  return <h1>This is a presentational component</h1>;
}
```

4. **Container Components:** Handle business logic, state, and data fetching. They pass props to presentational components.

```
import React, { Component } from "react";

class Container extends Component {
  render() {
    return <Presentational />;
  }
}
```