Module 11 React -Advance React- Styling , Routing

Question 1: What are components in React? Explain the difference between functional components and class components.

Answer:
Components in React are the building blocks of a React application. They allow you to split the UI into reusable, independent pieces that can be managed and maintained separately.

- **Functional Components:**
  - Written as JavaScript functions.
  - Use React hooks (like `useState` and `useEffect`) to manage state and lifecycle.
  - Example:

```
jsx
Copy code
const MyComponent = () => {
  return <h1>Hello, World!</h1>;
};
```

- **Class Components:**
  - Written as ES6 classes that extend `React.Component`.
  - Use `this.state` for state management and `this.setState()` for updates.
  - Example:

```
jsx
Copy code
class MyComponent extends React.Component {
  render() {
    return <h1>Hello, World!</h1>;
  }
}
```

Question 2: How do you pass data to a component using props?

Answer:
You pass data to a component by adding attributes to the component's JSX tag. These attributes are accessed within the component using `props`.

Example:

```
jsx
Copy code
const Greeting = (props) => {
  return <h1>Hello, {props.name}!</h1>;
};

// Usage
<Greeting name="John" />
```

Question 3: What is the role of `render()` in class components?

Answer:
The `render()` method in class components is responsible for describing what should be displayed on the screen. It returns React elements (JSX) that make up the UI.

LAB EXERCISE

Task 1: Functional Component - Greeting

```
jsx
Copy code
import React from "react";

const Greeting = ({ name }) => {
  return <h1>Hello, {name}!</h1>;
};

export default Greeting;

// Usage
// <Greeting name="Milan" />
```

Task 2: Class Component - WelcomeMessage

```
jsx
Copy code
import React, { Component } from "react";

class WelcomeMessage extends Component {
  render() {
    return <h1>Welcome to React!</h1>;
  }
}

export default WelcomeMessage;
```

THEORY EXERCISE

Question 1: What are props in React.js? How are props different from state?

Answer:
Props (short for "properties") are used to pass data from parent to child components. They are immutable and cannot be modified by the receiving component.

Difference between props and state:

- **Props:** Passed from parent to child; read-only.

- **State:** Managed within the component; can be updated using `setState` or hooks.

Question 2: Explain the concept of state in React and how it is used to manage component data.

Answer:
State is an object managed within a component that holds dynamic data. It allows React components to create and manage their own data that can change over time.

Question 3: Why is `this.setState()` used in class components, and how does it work?

Answer:
`this.setState()` is used to update the state in class components. It triggers a re-render, ensuring the UI reflects the updated state. It works asynchronously, batching multiple updates for performance.

LAB EXERCISE

Task 1: UserCard Component

```
jsx
Copy code
import React from "react";

const UserCard = ({ name, age, location }) => {
  return (
    <div style={{ border: "1px solid black", padding: "10px", margin:
"10px" }}>
      <h2>{name}</h2>
      <p>Age: {age}</p>
      <p>Location: {location}</p>
    </div>
  );
};

export default UserCard;

// Usage
// <UserCard name="Milan" age={20} location="Surat" />
```

Task 2: Counter Component

```
jsx
Copy code
import React, { useState } from "react";

const Counter = () => {
```

```
  const [count, setCount] = useState(0);

  return (
    <div>
      <h1>Count: {count}</h1>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
};

export default Counter;
```