React-2

**1. Explain React Components**

React components are the fundamental building blocks of a React application. They encapsulate logic, structure, and styling, allowing developers to create reusable, independent pieces of UI. Each component returns a portion of the UI using JSX and can manage its own state and lifecycle methods.

**2. Identify the Differences Between Components and JavaScript Functions**

| **Aspect** | **React Components** | **JavaScript Functions** |
| --- | --- | --- |
| Purpose | Build UI and manage state/lifecycle | Execute a task or perform an operation |
| Syntax | Uses JSX and React-specific syntax | Uses standard JavaScript syntax |
| Return Value | Returns UI elements (JSX) | Returns data values |
| Integration | Integrated within the React rendering tree | Standalone logic |
| State Management | Can maintain and update internal state | Stateless unless explicitly coded |

**3. Identify the Types of Components**

There are two main types of React components:

* **Class Components**: These are ES6 classes that extend React.Component and include lifecycle methods and a render function.
* **Function Components**: These are simpler components that use functions to return JSX. With the introduction of Hooks, they can also manage state and lifecycle logic.

**4. Explain Class Component**

A **class component** is a JavaScript ES6 class that extends from React.Component. It must include a render() method that returns JSX. Class components can have constructors and maintain internal state and lifecycle methods.

Example:

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Welcome to React</h1>;

}

}

export default Welcome;

**5. Explain Function Component**

A **function component** is a JavaScript function that returns JSX. It is a simpler and cleaner way to create components, especially with the support of React Hooks for state and lifecycle management.

Example:

import React from 'react';

function Greeting() {

return <h1>Hello, User</h1>;

}

export default Greeting;

**6. Define Component Constructor**

The **constructor** is a special function in class components used to initialize state and bind methods. It is called before the component is mounted.

Example:

constructor(props) {

super(props);

this.state = { message: "Hello" };

}

**7. Define render() Function**

The render() function is mandatory in class components. It returns JSX, which is then rendered to the DOM. It is called during the component mounting and updating phases.

**Hands-on Lab Instructions Summary:**

To create and run a React application for a Student Management Portal:

1. **Create a React project:**
2. npx create-react-app StudentApp
3. **Create a folder named Components inside src directory.**
4. **Create a file Home.js and write the following:**
5. import React from 'react';
6. function Home() {
7. return <h2>Welcome to the Home page of Student Management Portal</h2>;
8. }
9. export default Home;
10. **Create a file About.js under src/Components:**
11. import React from 'react';
12. function About() {
13. return <h2>Welcome to the About page of the Student Management Portal</h2>;
14. }
15. export default About;
16. **Create a file Contact.js under src/Components:**
17. import React from 'react';
18. function Contact() {
19. return <h2>Welcome to the Contact page of the Student Management Portal</h2>;
20. }
21. export default Contact;
22. **Edit App.js to include and render these components:**
23. import React from 'react';
24. import Home from './Components/Home';
25. import About from './Components/About';
26. import Contact from './Components/Contact';
27. function App() {
28. return (
29. <div>
30. <Home />
31. <About />
32. <Contact />
33. </div>
34. );
35. }
36. export default App;
37. **Run the project:**
38. npm start
39. **In your browser, open** http://localhost:3000.