## **WEEK 6**

## **2.ReactJS-HOL**

## **OBJECTIVES:**

**1. Explain React Components**

**React Component** is the fundamental building block of a React application. Components are reusable, independent pieces of UI logic that render specific parts of the web page.

* Think of components like functions that return HTML.
* Each component has its own structure, logic, and styling.

**Example:**

function Welcome() {

return <h1>Hello, React!</h1>;

}

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

| **Feature** | **JavaScript Function** | **React Component** |
| --- | --- | --- |
| Purpose | Performs calculations or logic | Returns UI (JSX/HTML-like structure) |
| Return Type | Data (string, number, object, etc.) | JSX (UI elements) |
| Naming Convention | Can start with lowercase | Starts with uppercase |
| Usage | Called like myFunction() | Used like <MyComponent /> |

**3. Identify the Types of Components**

React provides **two main types** of components:

1. **Function Components**
   * Simple, stateless components.
   * Use Hooks to manage state and lifecycle (from React 16.8+).
   * Lightweight and commonly used now.
2. **Class Components**
   * Stateful and more feature-rich.
   * Use this.state, lifecycle methods like componentDidMount().
   * Older way (still used in some projects).

### 4. **Explain Class Component**

A **Class Component** is a JavaScript ES6 class that extends from React.Component and has a render() method that returns JSX.

**Example:**

import React, { Component } from 'react';

class Welcome extends Component {

render() {

return <h1>Hello from Class Component</h1>;

}

}

* Has access to this.props and this.state.

**5. Explain Function Component**

A **Function Component** is a plain JavaScript function that accepts props as arguments and returns JSX.

**Example:**

function Welcome(props) {

return <h1>Hello, {props.name}</h1>;

}

* If needed, useState, useEffect hooks can be used for managing state and side effects.

**6. Define Component Constructor**

In class components, the **constructor** is a special method used for:

* Initializing state
* Binding methods to this

**Example:**

class Welcome extends React.Component {

constructor(props) {

super(props);

this.state = { message: 'Hello!' };

}

render() {

return <h1>{this.state.message}</h1>;

}

}

* Always call super(props) before using this.

**7. Define render() Function**

The render() method is **mandatory in class components**. It defines what will be displayed on the screen.

* Must return **a single JSX element** (can wrap multiple elements in a <div> or fragment <>).

**Example:**

render() {

return (

<div>

<h1>Student Portal</h1>

</div>

);

}

* React automatically calls render() when the component needs to update the UI.