**WEEK 6**

**3.ReactJS-HOL**

**OBJECTIVES:**

**1. Explain React Components**

A **React Component** is a reusable and independent block of code that defines part of the UI. Components let you split the UI into small, manageable parts. React components can be rendered with custom data using **props** and can manage internal data using **state**.

🔹 Each component:

* Returns JSX (HTML-like syntax in JavaScript)
* Can be reused multiple times
* Can be a class or function

**Example:**

function Welcome() {

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

}

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

| **Feature** | **JavaScript Function** | **React Component** |
| --- | --- | --- |
| Return Type | Any JS data (number, string, etc.) | JSX (HTML-like structure) |
| Naming | Can use lowercase | Must use PascalCase |
| Use | Called in code (e.g., func()) | Used in JSX (e.g., <Component />) |
| Side Effects | General logic | Handles UI rendering and state |

**3. Identify the Types of Components**

There are **two types** of components in React:

**🔸 1. Function Components**

* Defined using a JavaScript function.
* Lightweight and easier to read.
* Can use **React Hooks** for state and effects.

**Example:**

function Greet() {

return <h1>Hello</h1>;

}

**2. Class Components**

* Uses ES6 classes and extends React.Component.
* Used more often before React Hooks.
* Includes constructor, render(), lifecycle methods.

**Example:**

class Greet extends React.Component {

render() {

return <h1>Hello</h1>;

}

}

**4. Explain Class Component**

A class component is a JavaScript ES6 class that extends React.Component. It can hold state and use lifecycle methods.

**Example:**

import React, { Component } from 'react';

class Hello extends Component {

render() {

return <h2>Hello from Class Component</h2>;

}

}

**5. Explain Function Component**

A **function component** is a simpler, modern way to write components. It’s just a function that returns JSX. With the introduction of **Hooks**, function components can now also handle state and side effects.

**Example:**

function Hello() {

return <h2>Hello from Function Component</h2>;

}

**6. Define Component Constructor**

The **constructor** method is used only in **class components** to:

* Initialize **state**
* Bind **event handlers**

You must call super(props) before accessing this.

**Example:**

class Example extends React.Component {

constructor(props) {

super(props);

this.state = { count: 0 };

}

}

**7. Define render() Function**

The render() function is a **mandatory method** in every class component. It returns JSX that describes what should appear on the screen.

**Example:**

class Example extends React.Component {

render() {

return <p>This is rendered on screen.</p>;

}

}

⚠️ render() must return **only one JSX element** (you can wrap multiple elements in a <div> or <> fragment).