**WEEK 6**

**7.ReactJS-HOL**

**OBJECTIVES:**

**1. Define Props**

**Props** (short for "properties") are read-only attributes used to pass data from one component to another in React, typically from a **parent component to a child component**.

* Props are passed like attributes in HTML tags.
* They allow **component reusability** by customizing components with different data.
* Props are **immutable** inside the component receiving them.

**🔹 Example:**

**function Welcome(props) {**

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

**}**

**<Welcome name="Michael" />**

**Output:** Hello, Michael

**2. Explain Default Props**

**Default Props** are values assigned to props **if no value is passed** from the parent component.

* Used to prevent undefined errors.
* Can be set using Component.defaultProps.

**🔹 Example:**

function Welcome(props) {

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

}

Welcome.defaultProps = {

name: 'Guest'

};

If you call <Welcome /> without a name, it will display:

Hello, Guest

**3. Identify the Differences between State and Props**

| **Feature** | **Props** | **State** |
| --- | --- | --- |
| Definition | Data passed to components | Data managed **within** a component |
| Mutability | Immutable | Mutable |
| Usage | Used to **configure** component | Used to **control behavior** |
| Access | Received via props parameter | Managed using useState hook or this.state in class |
| Modification | Cannot be changed inside component | Can be changed with setState() or useState() |
| Data Flow | Parent → Child (unidirectional) | Local to component |

**4. Explain ReactDOM.render()**

The ReactDOM.render() function is used to **render a React element into the DOM** (i.e., actual HTML page).

It takes two arguments:

1. The React element/component to render
2. The DOM node (HTML element) where it should render

**🔹 Syntax:**

ReactDOM.render(<App />, document.getElementById('root'));

* <App /> is the root component.
* document.getElementById('root') points to the HTML element in public/index.html.

**From React 18 onward, ReactDOM.createRoot() is used instead:**

import { createRoot } from 'react-dom/client';

const root = createRoot(document.getElementById('root'));

root.render(<App />);