**WEEK-7\_ReactJs\_HOL**

**SUPERSET ID : 6393674**

**Objectives**

**1.Explain various ways of conditional rendering**

Conditional rendering in React allows rendering components or elements based on conditions:

* **Using if statements** – Basic control flow inside component functions.
* **Ternary Operator (? :)** – Useful in JSX for inline conditions.
* **Logical AND (&&)** – Renders the component if the condition is true.
* **Switch statements** – For multiple conditions to choose from.  
  Each method helps in customizing UI based on state or props.

**2. Explain how to render multiple components**

Multiple components can be rendered together using:

* **JSX fragments (<>...</>)** or <React.Fragment> to avoid extra DOM nodes.
* Grouping components inside a div.
* **Array rendering** using .map() to render lists of components dynamically.

**Example:**

<>

<Header />

<Content />

<Footer />

</>

**3. Define list component**

A **List component** in React is a reusable component that renders a list of items dynamically using the .map() function.  
It accepts an array (from props or state) and displays each item, often as a set of child components.  
Helps in separating UI logic from data.

**4. Explain about keys in React applications**

**Keys** are unique identifiers used by React when rendering lists.  
They help React identify which items have changed, added, or removed.  
A key must be unique among siblings and should not change between renders.  
Example: key={item.id} in a list.

**5. Explain how to extract components with keys**

When mapping a list into components, each extracted child must have a key prop.  
This allows React to manage DOM updates efficiently.  
Example:

const items = ['A', 'B', 'C'];

const listItems = items.map((item, index) => (

<ListItem key={index} value={item} />

));

**6. Explain React Map, map() function**

In React, the .map() function is used to loop over an array and return JSX for each item.  
It’s commonly used to render lists dynamically.  
Each item should have a unique key to help React with efficient rendering.  
Example:

{items.map(item => <div key={item.id}>{item.name}</div>)}

Create a React App named “bloggerapp” in with 3 components.

1. Book Details
2. Blog Details
3. Course Details

Implement this with as many ways possible of Conditional Rendering.

**App.js**

import React from 'react';

import CourseDetails from './components/CourseDetails';

import BookDetails from './components/BookDetails';

import BlogDetails from './components/BlogDetails';

import './App.css';

function App() {

return (

<div className="app-container">

<CourseDetails />

<BookDetails />

<BlogDetails />

</div>

);

}

export default App;

**OUTPUT:**

