**REACT**

**WEEK-7 HANDSON-13 EXPLANATION**

**1. All the variations of conditional rendering in React**

Conditional rendering is used when you want to render components based on certain conditions. The popular ways are:

if-else Statement

if (isLoggedIn) {

return <Dashboard />;

} else {

return <Login />;

}

Ternary Operator

{isLoggedIn ? <Dashboard /> : <Login />}

Logical AND (&&) Operator

{isLoggedIn && <Dashboard />}

Switch Case (via function)

function renderPage(page) {

switch(page) {

case 'home': return <Home />;

case 'about': return <About />;

default: return <NotFound />;

}

IIFE (Immediately Invoked Function Expression) inside JSX

{(() => {

if (isLoggedIn) return <Dashboard />;

return <Login />;

})()}

**2. How To Render Multiple Components**

Rendering multiple components means rendering more than one component from a parent component.

Example:

function App() {

return (

<div>

<Header />

<MainContent />

<Footer />

</div>

);

}

Members of list also:

const items = [<Item1 />, <Item2 />, <Item3 />];

return <div>{items}</div>;

**3. Define List Component**

A List Component is a component that outputs data in a list using, generally, .map().

Example:

function ItemList({ items }) {

return (

<ul>

{items.map((item, index) => <li key={index}>{item}</li>)}

</ul>

);

}

Usage:

<ItemList items={['React', 'Node', 'MongoDB']} />

**4. Keys in React Applications**

Keys are unique identifiers given to elements of a list so that React recognizes which items actually changed, are added, or removed.

Example:

{items.map(item => (

<li key={item.id}>{item.name}</li>

))}

The keys enhance performance and debugging rendering issues.

**5. Extract Components with Keys**

You can extract a child component when mapping over data, passing the key prop.

Example:

function Item({ item }) {

return <li>{item.name}</li>;

}

function ItemList({ items }) {

return (

<ul>

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

</ul>

);

}

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

The map() function in React is used to take arrays of data and convert them into arrays of elements.

Example:

const numbers = [1,2,3,4];

const listItems = numbers.map(number => <li key={number}>{number}</li>);

return <ul>{listItems}</ul>;

.map() creates a new array through some operation on every element of array – an ideal candidate for being used to dynamically render lists.