# SUPERSET ID-6412891

# Week 6 - ReactJS Hands-On Lab Solutions

## 1. Create a React App

# Command Line  
npx create-react-app my-app  
cd my-app  
npm start

## 2. Create a Simple Component

// Greeting.js  
function Greeting() {  
 return <h1>Hello, welcome to React!</h1>;  
}  
export default Greeting;

## 3. Use Props in a Component

// Welcome.js  
function Welcome(props) {  
 return <h2>Hello, {props.name}!</h2>;  
}  
  
// In App.js  
<Welcome name="Sabaritha" />

## 4. Use State in a Component

import React, { useState } from 'react';  
  
function Counter() {  
 const [count, setCount] = useState(0);  
 return (  
 <div>  
 <p>Count: {count}</p>  
 <button onClick={() => setCount(count + 1)}>Increment</button>  
 </div>  
 );  
}

## 5. Handle Events in React

function ClickButton() {  
 const handleClick = () => {  
 alert("Button clicked!");  
 };  
  
 return <button onClick={handleClick}>Click Me</button>;  
}

## 6. Fetch Data using useEffect

import React, { useState, useEffect } from 'react';  
  
function UserList() {  
 const [users, setUsers] = useState([]);  
  
 useEffect(() => {  
 fetch("https://jsonplaceholder.typicode.com/users")  
 .then(res => res.json())  
 .then(data => setUsers(data));  
 }, []);  
  
 return (  
 <ul>  
 {users.map(user => <li key={user.id}>{user.name}</li>)}  
 </ul>  
 );  
}

## 7. Conditional Rendering

function LoginStatus({ isLoggedIn }) {  
 return (  
 <div>  
 {isLoggedIn ? <p>Welcome Back!</p> : <p>Please log in.</p>}  
 </div>  
 );  
}

## 8. Render List using map()

function FruitList() {  
 const fruits = ['Apple', 'Banana', 'Orange'];  
  
 return (  
 <ul>  
 {fruits.map((fruit, index) => (  
 <li key={index}>{fruit}</li>  
 ))}  
 </ul>  
 );  
}