## **Objectives**

**Explain about conditional rendering in React**

Conditional rendering in React means showing different UI elements or components based on certain conditions. It works like regular JavaScript conditions, such as “if” statements or the ternary (?:) operator, but is applied inside JSX.

**Define element variables**

Element variables are React elements stored in variables so they can be used conditionally or reused later in the render. Instead of writing JSX directly in the return statement, you can assign it to a variable and then decide when or where to display it.

**Explain how to prevent components from rendering**

Sometimes you don’t want a component to render anything under certain conditions. In React, you can do this by returning “null” from the component’s render function. Returning “null” means the component will not produce any DOM output.

**Codes:**

**App.js:**

import React,{useState} from 'react';

function **UserGreeting**(){

  return <h1>Welcome back</h1>;

}

function **GuestGreeting**(){

  return <h1>Please sign up.</h1>

}

function **Greeting**(props){

  const isLoggedIn=props.isLoggedIn;

  if(isLoggedIn){

    return <**UserGreeting**/>;

  }

  return <**GuestGreeting**/>;

}

function **LoginButton**(props){

  return <button *onClick*={props.onClick}>Login</button>

}

function **LogoutButton**(props){

  return <button *onClick*={props.onClick}>Logout</button>;

}

function **App**() {

  const [isLoggedIn,**setIsLoggedIn**]=**useState**(false);

  const **HandleLoginClick**=()=>{**setIsLoggedIn**(true);};

  const **HandleLogoutClick**=()=>{**setIsLoggedIn**(false);};

  let button;

  if(isLoggedIn){

    button=<**LogoutButton** *onClick*={**HandleLogoutClick**}/>

  }

  else{

    button=<**LoginButton** *onClick*={**HandleLoginClick**}/>

  }

  return (

    <div *style*={{padding:'30px'}}>

      <**Greeting** *isLoggedIn*={isLoggedIn}/>

      {button}

      </div>

  );

}

export default **App**;

**Output:**



