**2. Basics of React. js**

**a. Write a React program to implement a counter button using react class components**

**CounterApp.js**

import React, { Component } from 'react';

class CounterApp extends Component {

constructor() {

super();

this.state = { count: 0 };

}

incrementCount = () => {

this.setState({ count: this.state.count + 1 });

};

inDecrement = () => {

this.setState({ count:this.state.count - 1});

};

render() {

return (

<div className="counter-container">

<h1 className="counter-title">Counter: {this.state.count}</h1>

<button

className="counter-btn"

onClick={this.incrementCount}

disabled={this.state.count === 10}

>

+

</button>

<button

className="counter-btn"

onClick={this.decrementCount}

disabled={this.state.count === 0}

>

-

</button>

</div>

);

}

}

export default CounterApp;

**APP.css**

.counter-container {

text-align: center;

margin-top: 50px;

padding: 40px 0;

background: #f6f9fc;

border-radius: 16px;

width: 340px;

margin-left: auto;

margin-right: auto;

box-shadow: 0 6px 18px rgba(25,50,100,0.13);

}

.counter-title {

color: #2b388f;

margin-bottom: 20px;

font-size: 2rem;

font-family: 'Segoe UI', sans-serif;

}

.counter-btn {

margin: 0 12px;

padding: 10px 28px;

font-size: 1.4rem;

border: none;

border-radius: 5px;

background: #217af7;

color: #fff;

cursor: pointer;

font-family: inherit;

transition: background 0.18s;

}

.counter-btn:disabled {

background: #b2c7ec;

color: #f7f7f7;

cursor: not-allowed;

}

.counter-btn:hover:not(:disabled) {

filter: brightness(0.9);

}

**b.Write a React program to implement a counter button using react functional components**

**App.js**

function App() {

let count=0;

function increaseCounter(){

count++;

document.getElementById("countervalue").innerText=count;

document.getElementById("increasebtn").disabled= count === 10;

}

function decreaseCounter(){

count--;

document.getElementById("countervalue").innerText=count;

document.getElementById("decreasebtn").disabled = count === 0;

}

function resetCounter(){

count=0;

document.getElementById("countervalue").innerText=count;

document.getElementById("increasebtn").disabled = false;

document.getElementById("decreasebtn").disabled = false;

}

return(

<div class="demo" style={{textAlign:"center",marginTop:"50px"}}>

<h2> counter without hooks</h2>

<p id="countervalue">{count}</p>

<button id="increasebtn" onClick={increaseCounter} class="btn1">increase Counter</button>

<button id="decreasebtn" class="btn2" onClick={decreaseCounter}>decrease Counter</button>

<button id="resetbtn" class="btn3" onClick={resetCounter}>reset Counter</button>

</div>

);

}

export default App;

**App.css**

.demo {

text-align: center;

margin-top: 50px;

background: #f9f9f9;

padding: 30px 0 30px 0;

border-radius: 10px;

width: 320px;

margin-left: auto;

margin-right: auto;

box-shadow: 0 8px 24px rgba(0,0,0,0.1);

}

h2 {

color: #3467eb;

letter-spacing: 1px;

margin-bottom: 30px;

font-family: 'Segoe UI', sans-serif;

}

#countervalue {

font-size: 2.5rem;

margin: 30px 0 20px 0;

color: #24292f;

font-weight: bold;

}

button {

margin: 10px 10px 0 10px;

padding: 12px 24px;

font-size: 1rem;

border: none;

border-radius: 4px;

outline: none;

cursor: pointer;

font-family: inherit;

transition: background 0.2s, color 0.2s;

}

.btn1 {

background: #3467eb;

color: #fff;

}

.btn1:disabled {

background: #bbcbf5;

color: #fff;

cursor: not-allowed;

}

.btn2 {

background: #f34b3f;

color: #fff;

}

.btn2:disabled {

background: #facbc9;

color: #fff;

cursor: not-allowed;

}

.btn3 {

background: #f9d923;

color: #212529;

font-weight: 600;

}

**c.Write a React program to handle the button click events in functional component**

**ButtonClickDemo.js**

import React from 'react';

function ButtonClickDemo() {

// External function handler

function handleClick() {

alert("Button clicked using external function!");

}

// Arrow function as handler

consthandleArrowClick = () => {

alert("Button clicked using arrow function!");

};  
// With parameter

consthandleWithParameter = (name) => {

alert(`Hello, ${name}!`);

};

// Using event object

consthandleEventObject = (event) => {

alert(`Button clicked! Event type: ${event.type}`);

};

return (

<div className=”container” style={{ padding: '20px', fontFamily: 'Arial' }}>

<h2>React Button Click Event Examples</h2>

{/\* 1. Inline arrow function \*/}

<button className= “btn” onClick={() => alert("Inline button clicked!")}>

Inline Function

</button>

<br /><br />

{/\* 2. External named function \*/}

<buttonclassName= “btn” onClick={handleClick}>

External Function

</button>

<br /><br />

{/\* 3. Arrow function reference \*/}

<button className= “btn” onClick={handleArrowClick}>

Arrow Function

</button>

<br /><br />

{/\* 4. Function with parameter \*/}

<button className= “btn” onClick={() =>handleWithParameter(“Sanjana")}>

Click with Parameter

</button>

<br /><br />

{/\* 5. Event object usage \*/}

<button className= “btn” onClick={handleEventObject}>

Event Object Handler

</button>

</div>

);

}

export default ButtonClickDemo;

**APP.css**.container {

padding: 20px;

font-family: Arial, sans-serif;

}

.btn {

padding: 10px 20px;

background-color: #007BFF;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

font-size: 16px;

transition: background-color 0.3s ease;

}

.btn:hover {

background-color: #0056b3;

}

**d.Write a React program to conditionally render a component in the browser**

**App.js**

Import ‘./App.css’;

function App(){

return(

<>

<Header></Header>

{

isAdmin ?</Admin> : null;

}

</>

)

}

Let loggedIn = false;

Let isAdmin = true;

function Header(){

return(

<ul>

<li> Home</li>

<li> About Us</li>

<li>Contact Us</li>

<li> Login </li>

(or)  
 {

loggedIn ? (<li>logout</li>) :(<li> login </li>)

}

</ul>

)

}

function Admin(){

return(

<h1> this is admission dash board</h1>

)

}

**App.css**

Ul {

list-style-type: none;

display: flex;

back-ground-color: blue;

padding:20px;

color: #fff;

}

**e.Write a React program to display text using String literals**

import React from "react";

function greet(name) {

return `Hello ${name}, have a great day!`; // Way 5

}

function App() {

const name = "Mounika";

const subject = "React";

const message = `Welcome ${name}, this is a stored template literal!`; // Way 2

const details = `Name: ${name}

Course: ${subject}

Topic: String Literals`; // Way 3

const a = 20, b = 10;

return (

<div ClassName=”container”>

{/\* Way 1: Inline JSX with Template Literals \*/}

<h2>{`Hello, my name is ${name}`}</h2>

{/\* Way 2: Stored Template Literal \*/}

<p>{message}</p>

{/\* Way 3: Multi-line String Literal \*/}

<pre style={{ backgroundColor: "#f0f0f0", padding: "10px", borderRadius: "8px" }}>

{details}

</pre>

{/\* Way 4: Expressions inside String Literals \*/}

<h3>{`The sum of ${a} and ${b} is ${a + b}`}</h3>

{/\* Way 5: Function Returning String Literal \*/}

<h2>{greet(name)}</h2>

{/\* Extra: Date using string literal \*/}

<p>{`Today’s date is ${new Date().toLocaleDateString()}`}</p>

</div>

);

}

export default App;

**App.css**

.container {

text-align: center;

margin-top: 30px;

font-family: Arial, sans-serif;

padding: 20px;

}

h2 {

color: #2c3e50;

}

h3 {

color: #16a085;

}

p {

font-size: 18px;

color: #444;

}

.details {

background-color: #f0f0f0;

padding: 12px;

border-radius: 8px;

font-size: 16px;

text-align: left;

display: inline-block;

margin-top: 10px;

}