Program-1 : Hello World Without Create-react-app

**********

```
<html>
<head>
  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-
dom.development.js"></script>
</head>
<body>at
  <div id="example"></div>
  <script>
    const reactEle = React.createElement('h1', {}, 'My First React App');
                    const domEle = document.getElementById('example');
    const root = ReactDOM.createRoot(domEle);
    root.render(reactEle);
  </script>
</body>
</html>
```

Note: React object is available(can be used in console like other objects)

Program-2 : Nested Element Without Create-react-app

**********

```
<html>
<head>
  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-
dom.development.js"></script>
</head>
<body>
  <div id="example"></div>
  <script>
    const title = React.createElement('h1', {}, 'My First React Code');
    const paragraph = React.createElement('p', {}, 'Writing some more HTML');
    const containerDiv = React.createElement('div', {id:myDiv1}, [title, paragraph]);

    const domEle = document.getElementById('example');
    const root = ReactDOM.createRoot(domEle);
    root.render(containerDiv);
  </script>
</body>
</html>
```

Program-3 : Nested Element using JSX (Babel)

**********

```
<html>
<head>
  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-
dom.development.js"></script>
        <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
</head>
<body>
  <div id="example"></div>
  <script type="text/babel">
    const reactEle = <div>
```

```
        <h1>Hello World</h1>
        <p>This is a paragraph</p>
        <div>This is a div</div>
       </div>
      const domEle = document.getElementById('example');

      const root = ReactDOM.createRoot(domEle);
      root.render(reactEle)
  </script>
</body>
</html>
```

Program-4: use Bootstrap
**********

1. install bootstrap
   npm i bootstrap
2. add below 2 lines to index.js file
   import 'bootstrap/dist/css/bootstrap.min.css';
   import "bootstrap/dist/js/bootstrap.bundle.min.js";

Program-5: use Bootstrap-icons
**********

1. install bootstrap icons library
   npm i bootstrap-icons
2. import bootstrap-icons.css file in to index.js
   import 'bootstrap-icons/font/bootstrap-icons.css';
3. Use class names

   <i className="bi-airplane" style={{ fontSize: "2rem", color: "cornflowerblue" }}>

Program-6 : How to use React-icons
**********

1. install react-icons
   npm install react-icons
2. import react-icons in component
   import { FaEdit, FaTrash } from 'react-icons/fa';
   import { BsFillCalendarDateFill,BsFillClockFill } from 'react-icons/bs';
3. use the Icons
   <FaEdit />
   <FaTrash color='red' />

Program-7: React Rendering Elements
**********

```
export default function tick() {
 const element = (
   <div>
     <h1>Hello, world!</h1>
```

```
    <h2>Now The Time is {new Date().toLocaleTimeString()}.</h2>
   </div>
  );
  root.render(element);
}
setInterval(tick, 1000);
```

Program-8:  Fragments

**********

```
import React from 'react';
class Header extends React.Component {
   render() {
      return (
         <React.Fragment>
            <div>I am an element!</div>
            <button>I am another element</button>
         </React.Fragment>
      );
   }
}
export default Header;
```

program-9: Data Binding

**********

```
import React from 'react';
class App extends React.Component {
  render() {
                        function formatName(user) {
         return user.firstName + ' ' + user.lastName;
      }
      let user = {
         firstName: 'sanjay',
         lastName: 'samantra'
      };
     return (
        <div>
                                          <h2>React Version is {React.version}</h2>
            <h2>{1 + 1}</h2>
            <h2>my name is- {user.firstName}</h2>
            <h2>my full name is- {formatName(user)}</h2>
         </div>
     );
  }
}
export default App;
```

program-10 display Image

**********

1. place the image inside 'public' folder
2. use it in HTML file
  <img src='images/sachin.jpg' />
                (OR)
1. import sachin from './sachin.jpg'; (OR) const sachin = require('./sachin.jpg')
2. <img src={sachin} />

program-11 : CSS
**********
```
import React from 'react';
class App extends React.Component {
  render() {
            var myStyle={
                              color:'blue',
                              backgroundColor:'yellow',
                              border:'5px solid green'
            }
    return (
      <div>
        Hello World!!!
                              <p style={{color:'red',fontSize:'42px'}}>this is paragraph</p>
                              <div style={myStyle}>this is a div</div>
      </div>
    );
  }
}
export default App;
```

program-12 CSS Modules
**********
DashedBox.module.css
---
```
.container {
  margin: 40px;
  border: 5px dashed pink;
}
.content {
  font-size: 15px;
  text-align: center;
}
.error {
  color: red;
}
```

---
DashedBox2.css
---
```
.error{
            background-color:yellow;
}
```

---
```
import React from 'react';
import styles from './DashedBox.module.css';
import './DashedBox2.css';

const DashedBox = () => (
            console.log(styles)
   <div className={styles.container}>
    <p className={styles.content}>Get started with CSS Modules style</p>
            <div className={styles.error}>this is a div</div>
   </div>
);
export default DashedBox;
```

Program-13 conditional CSS

**********

```
import React from 'react';
export default function Demo1() {
   let num = 5;
   let style1 = { color: 'blue' }
   let style2 = { color: 'red' }
   return (
      <>
                        <div style={num % 2 === 0 ? { color: "green" } : { color: "red" }}>
                                 Number {num} is {num % 2 === 0 ? "even" : "odd"}
                        </div>
                        <div style={num % 2 === 0 ? style1 : style2}>
                                 Number {num} is {num % 2 === 0 ? "even" : "odd"}
                        </div>
                        <button className={`btn btn-lg ${num % 2 === 0 ? "btn-success" : "btn-
danger"}`}>
                                 add a class dynamically
                        </button>
                       </>
   )
}
```

program-14 : Conditional Operator

**********

```
import React from 'react';
class Conditional extends React.Component {
   render() {
      let x = 4;
      return (
         <h1>{x % 2 == 0 ? `${x} is Even` : `${x} is odd`}</h1>
      )
   }
}
export default Conditional;
```

Program-15 : if else

**********

```
import React from 'react';
class Conditional extends React.Component {
   render() {
      let x = prompt("enter a number")
      if(x%2==0){
         return ( <h2>{x} is even</h2> )
      }
      else{
         return ( <h2>{x} is odd</h2> )
      }
   }
}
export default Conditional;
```

Program-16 Logical &&

**********

```
import React, { Component } from 'react'
export default class Conditional extends Component {
   render() {
     <div>
                        <h1>Hello!</h1>
                        {unreadMessages.length > 0 &&
                                 <h2>
                                  You have {unreadMessages.length} unread messages.
                                 </h2>
                        }
    </div>
   }
}
```

program-17 switch case

**********

```
import React from "react";

export default function ConditionalDemo2() {
 const dayNumber = new Date().getDay();
 switch (dayNumber) {
   case 1:
    return <h3>Today is Monday-{dayNumber}</h3>;
   case 2:
    return <h3>Today is Tuesday-{dayNumber}</h3>;
   case 3:
    return <h3>Today is Wednesday-{dayNumber}</h3>;
   case 4:
    return <h3>Today is Thursday-{dayNumber}</h3>;
   case 5:
```

```
    return <h3>Today is Friday-{dayNumber}</h3>;
  case 6:
    return <h3>Today is Saturday-{dayNumber}</h3>;
  case 7:
    return <h3>Today is Sunday-{dayNumber}</h3>;
  default:
    return <h3>Not a Valid number</h3>;
 }
}
```

program-18 List of items

**********

```
import React from 'react';
class List extends React.Component {
  render() {
    let cars = ['tata', 'honda', 'maruti', 'hundai', 'toyota']
    return (
      cars.map((car, ind) => {
        return <li>{ind} - {car}</li>
      })
    )
  }
}
export default List;
```

(OR)

```
import React from 'react';
class List extends React.Component {
  render() {
    let cars = ['tata', 'honda', 'maruti', 'hundai', 'toyota'];
                let carList = cars.map(car => <h2>{car}</h2>)
    return <div>{carList}</div>
  }
}
export default List;
```

Program:19 Countries DropDown & Datalist

**********

```
export default function ListDemo2() {
        const countries = ["Afghanistan","Albania","Algeria"]
 return <>
  <h2>Countries Dropdown</h2>
  Dropdown:
  <select>
    {countries.map((country,ind)=>{
      return <option key={ind}>{country}</option>
    })}
  </select>
  <hr/>
  datalist:
  <input list="countries" name="country" id="country" />
  <datalist id="countries">
    {countries.map((country,ind)=>{
```

```
        return <option key={ind} value={country} />
      })}
   </datalist>
 </>
}
```

program-20 List of Objects

**********

```
import React from 'react';
export default function EmployeeList {
      let employees = [
         { eId: 101, name: "sanjay", sal: 5000 },
         { eId: 104, name: "deepak", sal: 8000 },
         { eId: 103, name: "ranjan", sal: 7000 },
         { eId: 102, name: "manoj", sal: 9000 }
      ]
      return
         <table border='1' align='center' width='50%'>
            {employees.map(emp => (
               <tr key={emp.eId}>
                  <td>{emp.eId}</td>
                  <td>{emp.name}</td>
                  <td>{emp.sal}</td>
               </tr>
            ))}
         </table>
}
```

----------(OR)----------

```
import React, { Component } from 'react';
export default class Demo extends Component {
   render() {
      let employees = [
         { eId: 101, name: "sanjay", sal: 5000 },
         { eId: 104, name: "deepak", sal: 8000 },
         { eId: 103, name: "ranjan", sal: 7000 },
         { eId: 102, name: "manoj", sal: 9000 }
      ]
      const empRows = [];
      for (let emp of employees) {
         const row = (
            <tr key={emp.eId}>
               <td>{emp.eId}</td>
               <td>{emp.name}</td>
               <td>{emp.sal}</td>
            </tr>
         );
         empRows.push(row);
      }
      return (
         <table className='table table-bordered table-striped'>
            <thead>
```

```
            <tr>
              <th>Emp Id</th>
              <th>Name</th>
              <th>Salary</th>
            </tr>
          </thead>
          <tbody>
            {empRows}
          </tbody>
        </table>
      );
    }
}
```

Program-21 Iterate objects

**********

```
import React, { Component } from 'react'

export default class Demo1 extends Component {
  render() {
    let employees = [
      { eId: 101, name: "sanjay", sal: 5000 },
      { eId: 104, name: "deepak", sal: 8000 },
      { eId: 103, name: "ranjan", sal: 7000 },
      { eId: 102, name: "manoj", sal: 9000 }
    ]
    return <div className="container">
      <h2 className='text-center'>Employees Table</h2>
      <table className="table table-bordered">
        <thead>
          <tr>
            <th>id</th>
            <th>name</th>
            <th>salary</th>
          </tr>
        </thead>
        <tbody>
          {
            employees.map((emp, ind) => (
              <tr key={ind}>
                {Object.values(emp).map((val, ind) => (
                  <td key={ind}>{val}</td>
                ))}
              </tr>
            ))
          }
        </tbody>
      </table>
    </div>
  }
}
```

Program-22 productList & Product

**********

```
import React, { Component } from 'react'
import Product from './product'
export default class Productlist extends Component {
   productList = [
                        { pId: 101, name: "iphone", price: 5000 },
                        { pId: 102, name: "samsung", price: 4000 },
                        { pId: 103, name: "oppo", price: 3000 },
            ];
   render() {
      return (
         <div>
            <h1>This is Product-List Component</h1>
            {this.productList.map(item => (
               <Product product={item} />
            ))}
         </div>
      )
   }
}
```

---

```
import React, { Component } from 'react';
import './product.css';
export default class Product extends Component {
   render() {
      return (
         <div class='product'>
            {this.props.product.id} <br />
            {this.props.product.name} <br />
            {this.props.product.price} <br />
            <button class='btn btn-success'>Buy Now</button>
         </div>
      )
   }
}
```

program-23 problem with Local Variables declared inside render()

**********

```
import React, { Component } from 'react'
export default class Test extends Component {
   render() {
      let count = 0;
      function increment() {
         count = count + 1;
         alert(count);
      };
      return (
         <div>
            <p>The count is: {count}</p>
            <button onClick={() => increment()}>Add one</button>
         </div>
```

```
      );
   }
}
// render() method gets invoked only if there is a change in props/state
```

program-24  Force Update

**********

```
import React, { Component } from 'react'
export default class Conditional extends Component {
   count = 0;
   increment = () => {
      this.count = this.count + 1;
      this.forceUpdate();
   };
   render() {
      return (
         <div>
            <p>The count is: {this.count}</p>
            <button onClick={this.increment}>Add one</button>
         </div>
      );
   }
}
```

program-25: Props

**********

```
import React from 'react';
import Child from '../child/child';
class Parent extends React.Component {
   render() {
      let user = { name: 'sachin', age: 35 }
      return (
         <div>
            This is Parent Component
            <Child name={user.name} age={user.age} />
         </div>
      );
   }
}
export default Parent;
```

---

```
import React from 'react';
class Child extends React.Component {
   render() {
      return (
         <div>
            <h2>This is child component</h2>
            <h2>{this.props.name}--{this.props.age}</h2>
         </div>
      );
   }
}
```

```
export default Child;
```

Program-26  Props De-structuring
**********
```
import React, { Component } from 'react'

export default class Product extends Component {
   render() {
      let { pId, name, price } = this.props.prodObj;
      return (
         <div style={{ border: '2px solid red' }}>
            {pId} <br />
            {name} <br />
            {price} <br />
            <button class='btn btn-success'>Buy Now</button>
            <h2>the text from my parent comp is {this.props.children}</h2>
         </div>
      )
   }
}
```

Program-27 propTypes
**********
```
import React from 'react';
import { PropTypes } from 'prop-types';
class Student extends React.Component {

 render() {
   return (
     <div>
       <p>Student Name: {this.props.name}</p>
       <p>Age: {this.props.age}</p>
     </div>
   );
 }
}
Student.propTypes = {
 name: PropTypes.string.isRequired,
 age: PropTypes.number
};
export default Student;
```

Program-28 Requiring Single Child
**********
```
import React from 'react';
```

```
import { PropTypes } from 'prop-types';
class Container extends React.Component {
render() {
   return (
     <div>
       {this.props.children}
     </div>
   );
  }
}
Container.propTypes = {
  children: PropTypes.element
};
export default Container;
---------------------------------
import React from 'react';
import Container from './Container';
import Student from './Student';
class App extends React.Component {
render() {
   return (
     <div>
       <Container>
         <Student name="Mark" age="24" />
         <Student name="Peter" age="25" />
       </Container>
     </div>
   );
  }
}
export default App;
```

Program-29: props.children

**********

```
import React from 'react';
import Greet from '../greet/greet';
class Content extends React.Component {
   render() {
      return (
         <div>
            <Greet>
               <p>I am child-1</p>
               <p>I am child-2</p>
            </Greet>
            <Greet>
               <p>I am child-1</p>
               <p>I am child-2</p>
            </Greet>
         </div>
      );
   }
}
export default Content;
----------------------------
import React from 'react';
```

```
class Greet extends React.Component {
   render() {
      return (
         <h2>
            This is Greet Component
            {this.props.children}
         </h2>

      );
   }
}
export default Greet;
```

Program-30: Methods as props

**********

```
import React, { Component } from 'react';
import Child from '../child/child';

export default class Parent extends Component {
   constructor(props) {
      super(props);
      this.state = {
         parentName: 'Parentttt'
      };
      this.greetParent = this.greetParent.bind(this);
      this.greetParentWithParam = this.greetParentWithParam.bind(this);
   }
   greetParent() {
      alert(`Hello ${this.state.parentName}`)
   }
   greetParentWithParam(data) {
      alert(`Hello parent with parameter from ${data}`)
   }
   render() {
      return (
         <div>
            <Child greetHandler={this.greetParent} greetHandler2={this.greetParentWithParam}/>
         </div>
      )
   }
}
```

---

```
import React from 'react';
class Child extends React.Component {
   render() {
      return (
         <div>
            <div>I am child component</div>
            <button onClick={this.props.greetHandler}>Invoke parent Method</button>
            <button onClick={() => this.props.greetHandler2('childparameter')}>Invoke parent
Method</button>
         </div>
      );
   }}
```

```
export default Child;
```

Program-31: States

**********

```
class Counter extends React.Component {
  constructor(props) {
    super(props);
    this.state = {count: 0 };
  }
  increment = () => {
    this.setState({count: this.state.count + 1});
  }
  render() {
    return (
      <div>
        <div>count:{this.state.count}</div>
        <button onClick={this.increment}>click!</button>
      </div>
    );
  }
};
```

Program-32  asynchronous state

**********

```
import React, { Component } from "react";
export default class Counter extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0,
    };
  }
  incrementCountFiveTimes() {
    this.increment();
    this.increment();
    this.increment();
    this.increment();
    this.increment();
  }
  increment() {
    this.setState({ count: this.state.count + 1 });
    console.log(this.state.count);
  }
  render() {
    return (
      <h2>
        counter works
        <div>Count - {this.state.count}</div>
        <button className="btn btn-primary m-2" onClick={this.increment}>
          Increment
```

```
      </button>
      <button className="btn btn-primary m-2" onClick={this.incrementCountFiveTimes}>
        Increment
      </button>
     </h2>
   );
  }
}
```

Program-33 setState() in Detail-2
**********

```
//React may group multiple setState calls in to a single update for better performance
//when we want to update the state based on the previous State,
//we need to pass a function as an arguement to setState() instead passing an object

import React from 'react';
class Counter extends React.Component {
   constructor(props) {
      super(props);
      this.state = {
         count: 0
      }
   }
   increment() {
      this.setState((prevState) => ({
         count: prevState.count + 1
      }))
   }
   incrementFive() {
      this.increment();
      this.increment();
      this.increment();
      this.increment();
      this.increment();
   }
   render() {
      return (
         <h2>
            counter works
            <div>Count - {this.state.count}</div>
            <button onClick={() => this.incrementFive()}>Increment</button>
         </h2>
      );
   }
}
export default Counter;
```

Program -34 Batch update
**********

```
import React, { useState } from "react";

export default function Demo1() {
 const [count, setCount] = useState(0);
 const increment = () => {
  setCount((prevState) => prevState + 1);
  // setCount(count + 1);
 };
 const increment5 = () => {
  increment();
  increment();
  increment();
  increment();
  increment();
 };
 return (
  <div>
   <div>{count}</div>
   <button onClick={increment5}>Increment By 5</button>
  </div>
 );
}
```

Program-35 Show/Hide

**********

```
import React from 'react';
class MyClass extends React.Component {
   constructor(props) {
      super(props);
      this.state = { flag: true };
   }
   toggleFlag = () => {
      this.setState({ flag: !this.state.flag });
   }
   render() {
      return (
         <div>
            {this.state.flag ? <div>Hellllllo</div> : null}
            <button onClick={this.toggleFlag}>
               {this.state.flag ? 'HIDE' : 'SHOW'}
            </button>
         </div>
      );
   }
}
export default MyClass;
```

---

```
import React, { useState } from 'react';
export default function Demo() {
   const [flag, toggleFlag] = useState(true);
   const toggle = function () {
      toggleFlag(!flag)
   }
   return (
```

```
      <div>
        {flag ? <div>Hellllllo</div> : null}
        <button onClick={toggle}>
          {flag ? 'HIDE' : 'SHOW'}
        </button>
      </div>
    )
}
```

Program-36 Show/Hide Password

**********

```
import React, { Component } from 'react';
export default class Test extends Component {
   constructor(props) {
      super(props)
      this.state = {
         flag: true,
      }
   }
   togglePassword = () => {
      this.setState({
         flag: !this.state.flag
      })
   }
   render() {

      return (
         <div>
            <input type={this.state.flag ? 'password' : 'text'} />
            <button onClick={this.togglePassword}>{this.state.flag ? 'Show password' : 'Hide
password'}</button>
         </div>
      )
   }
}
```

---

```
import React, { useState } from "react";

export default function Demo() {
  const [flag, toggleFlag] = useState(true);
  const toggle = function () {
    toggleFlag(!flag);
  };
  return (
    <div>
      <input type={flag ? "password" : "text"} />
      <button onClick={toggle}>
        {flag ? "Show password" : "Hide password"}
      </button>
    </div>
  );
}
```

Program:37 Remaining Character

```
*****************************
export default function RemainingChar() {
 const [remaining, setRemaining] = useState(100);
 const keyUpHandler = (e) => {
  const msg = e.target.value;
  setRemaining(100-msg.length)
 };
 return (
  <>
   <textarea onKeyUp={keyUpHandler} maxLength={100} className="m-1"></textarea>
   <p>remaining char: {remaining}</p>
  </>
 );
}
```

Program:38 Selected value from dropdown

```
**************************************
import React, { useState } from 'react'
export default function StateDropDown() {
   const stateNames = ['Karnataka', 'Odisha', 'Tamilnadu', 'Kerala', 'Bihar', 'UP'];
   const [selectedState, setSelectedState] = useState(stateNames[0]);
   const stateChanged = (e) => {
      setSelectedState(e.target.value)
   }
   return <>
      <select onChange={stateChanged}>
         {stateNames.map((state, ind) => {
            return <option key={ind}>{state}</option>
         })}
      </select>
      <h2>You have Selected -  {selectedState}</h2>
   </>
}
```

Program:39 Dark/Light Mode

```
*************************
import React, { useEffect, useState } from 'react';
import './ThemeDemo.css';
export default function ThemeDemo() {
   const [isDark, setIsDark] = useState(false);
   const changeMode = () => {
      setIsDark(!isDark)
   }
   useEffect(() => {
      if (isDark) {
         document.body.classList.add('dark');
```

```
      } else {
         document.body.classList.remove('dark');
      }
   }, [isDark])

   return <>
      <h2>Toggle Dark/Light Mode</h2>
      <div>This page is in <b>{isDark ? 'Dark' : 'Light'}</b> Mode</div>
      <button onClick={changeMode}>Go To {!isDark ? 'Dark' : 'Light'} Mode</button>
   </>
}
======
.dark{
   background-color:black;
   color:white
}
```

Program-39 Addition

**********

```
import React from 'react';

class Addition extends React.Component {
   state = {
      a: 0,
      b: 0,
      total: 0
   }
   changeFirst = (e) => {
      this.setState({ a: parseInt(e.target.value) })
   }
   changeSecond = (e) => {
      this.setState({ b: parseInt(e.target.value) })
   }
   addition = () => {
      this.setState({ total: this.state.a + this.state.b })
   }
   render() {
      return (
         <div>
            Addition Works!!!
            <input onKeyUp={this.changeFirst}></input>
            <input onKeyUp={this.changeSecond}></input>

            <h2>Addition is {this.state.total}</h2>
            <button onClick={this.addition}>Addddddd</button>
         </div>
      );
   }
}
export default Addition;
----------------------------------------------------
import React, { useState } from 'react'
export default function Demo() {
```

```
  const [num1, setnum1] = useState(0);
  const [num2, setnum2] = useState(0);
  const [total, settotal] = useState(0);

  return (
    <div>
      <div>
        <input type="number" onKeyUp={(e) => {
          setnum1(+e.target.value);
        }} />
        <input type="number" onKeyUp={(e) => {
          setnum2(+e.target.value);
        }} />

        Addition is: {total}

        <button onClick={(e) => {
          settotal(num1 + num2);
        }}>Get Totaaal</button>
      </div>
    </div>
  );
}
```

Program-40 : Folder Explorer

**********

```
import React, { useState } from "react";
export default function Folder({ folderInfo }) {
  const [expand, setExpand] = useState(false);

  if (folderInfo.isFolder) { // if folder
    return <div>
      <span onClick={() => setExpand(!expand)} style={{ cursor: 'pointer' }}>

        {folderInfo.name}
        <br />
      </span>
      <div
        style={{ display: expand ? "block" : "none", paddingLeft: "20px" }}
      >
        {folderInfo.items.map((subfolderInfo, ind) => (
          <Folder folderInfo={subfolderInfo} key={ind} />
        ))}
      </div>
    </div>
  } else { // if file
    return <span>

      {folderInfo.name} <br />
    </span>

  }
}
```

```
import React, { useEffect, useState } from 'react';
import './TrafficLight.css';

const config = {
   red: { backgroundColor: 'red', duration: 4000, next: 'green' },
   yellow: { backgroundColor: 'yellow', duration: 1000, next: 'red' },
   green: { backgroundColor: 'green', duration: 3000, next: 'yellow' }
};
export default function TrafficLight() {
   const [currentColor, setCurrentColor] = useState('green');

   useEffect(() => {
      const { duration, next } = config[currentColor];
      const timerId = setTimeout(() => {
         setCurrentColor(next);
      }, duration);
      return () => {
         clearTimeout(timerId);
      };
   }, [currentColor]);

   return <div className="wrapper">
      <div className='traffic-light-container traffic-light-container--vertical'>
         {Object.keys(config).map((color) => (
            <div key={color} className="traffic-light"
               style={{
                  backgroundColor: color === currentColor
                     ? config[color].backgroundColor
                     : ''
               }}
            />
         ))}
      </div>
   </div>
}
-------------------------
.wrapper {
   align-items: center;
   display: flex;
   flex-direction: column;
   gap: 16px;
   justify-content: center;
}
.traffic-light-container {
   background-color: #000;
   border-radius: 8px;
   display: flex;
   padding: 8px;
   gap: 8px;
}
.traffic-light-container--vertical {
   flex-direction: column;
}
.traffic-light {
   --size: 50px;
   background-color: #555;
```

```
  border-radius: var(--size);
  height: var(--size);
  width: var(--size);
}
```

Program: 41 Search

**********

```
import React, { useState } from "react";
import SearchBar from "react-js-search";

export default function EmployeeList() {
 const initialEmployees = [
  { number: 12, name: "Buffon", position: "ST", success: true },
  { number: 21, name: "Pirlo", position: "MC", success: false },
  { number: 10, name: "Ruiz", position: "MDI" },
  { number: 7, name: "Nesta", position: "RB", success: true },
  { number: 4, name: "Cannavaro", position: "CB" },
  { number: 2, name: "Puyol", position: "CB", success: false },
  { number: 15, name: "Abate", position: "LB" },
  { number: 16, name: "Locatelli", position: "MDI" },
  { number: 1, name: "Buffon", position: "GK" },
  { number: 21, name: "Pirlo", position: "MC" },
  { number: 10, name: "Ruiz", position: "MDI" },
  { number: 7, name: "Nesta", position: "RB" },
 ];

 const [employees, setEmployees] = useState(initialEmployees);
 const [filteredEmployees, setFilteredEmployees] = useState(initialEmployees);

 const onSearchClick = (searchText) => {
  setFilteredEmployees(
   employees.filter((emp) => JSON.stringify(emp).includes(searchText))
  );
 };
 const onSearchTextChange = (searchText, filteredData) => {
  setFilteredEmployees(filteredData);
 };
 const sortAsc = () => {
  const sortedData = filteredEmployees.sort((e1, e2) => e1.name.localeCompare(e2.name))
  setFilteredEmployees([...sortedData]);
 };
 const sortDesc = () => {
  const sortedData = filteredEmployees.sort((e1, e2) => e2.name.localeCompare(e1.name))
  setFilteredEmployees([...sortedData]);
 };

 return (
  <>
   <SearchBar
    // onSearchTextChange={onSearchTextChange}
    onSearchButtonClick={onSearchClick}
    placeHolderText={"Search here..."}
    data={employees}
   />
```

```
    <button onClick={sortAsc}>Asc</button>
    <button onClick={sortDesc} className='ms-1'>Desc</button>
    <hr />
    {
     <table className="table table-bordered">
       <tbody>
        {filteredEmployees.map((emp, ind) => (
          <tr key={ind}>
            <td>{emp.number}</td>
            <td>{emp.name}</td>
            <td>{emp.position}</td>
          </tr>
        ))}
       </tbody>
     </table>
    }
   </>
 );
}
```

Program-42 Pagination

**********

```
import React, { useState } from "react";
import productsArr from "./Products.json";
import ReactPaginate from "react-paginate";
import "./ProductsPagination.css";

export default function ProductsPagination() {
 const [itemOffset, setItemOffset] = useState(0);
 const itemsPerPage = 4;

 const endOffset = itemOffset + itemsPerPage;
 console.log(`Loading items from ${itemOffset} to ${endOffset}`);
 const filteredProducts = productsArr.slice(itemOffset, endOffset);
 const pageCount = Math.ceil(productsArr.length / itemsPerPage);

 const handlePageClick = (event) => {
  const newOffset = (event.selected * itemsPerPage) % productsArr.length;
  console.log(
   `User requested page number ${event.selected}, which is offset ${newOffset}`
  );
  setItemOffset(newOffset);
 };

 return (
  <>
   <h1 className="text-center">Products With Pagination</h1>

   <div className="container">
    <div className="row">
     {filteredProducts.map((product) => (
      <div className="col-sm-3" key={product.id}>
       <div className="card text-center">
        <img
         src={product.image}
         className="card-img-top"
```

```
            alt="..."
            style={{ height: "200px" }}
           />
           <div className="card-body">
            <h5 className="card-title">{product.category}</h5>
            <p className="card-text">{product.title}</p>
            <p
             className="card-text"
             style={{ height: "200px", overflow: "scroll" }}
            >
             {product.description}
            </p>
            <p className="card-text"> {product.price}</p>
            <a href="#" className="btn btn-primary">
             BUY NOW
            </a>
           </div>
          </div>
         </div>
        ))}
      </div>
    </div>

    <ReactPaginate
      breakLabel="..."
      nextLabel="next >"
      onPageChange={handlePageClick}
      pageRangeDisplayed={5}
      pageCount={pageCount}
      previousLabel="< previous"
      renderOnZeroPageCount={null}
      containerClassName="pagination"
      pageLinkClassName="page-num"
      previousLinkClassName="page-num"
      nextLinkClassName="page-num"
      activeLinkClassName="active"
    />
   </>
  );
}
```

ProductsPagination.css
==========

```
.pagination {
  list-style: none;
  display: flex;
  justify-content: center;
  align-items: center;
  margin: 10px;
  font-size: 16px;
  gap: 5px;
}
.pagination .page-num {
  padding: 10px 15px;
  cursor: pointer;
  border-radius: 5px;
  font-weight: bold;
```

```
  text-decoration: none;
}
.pagination .page-num:hover {
  background-color: aqua;
  color: white;
}
.pagination .active {
  background-color: aqua;
}
```

Program- 43  ToDo CRUD

**********

```
export default function ToDoCRUD() {
    let initialtodos = [
        { id: 1, text: 'To Do 1', completed: true },
        { id: 2, text: 'To Do 2', completed: false },
        { id: 3, text: 'To Do 3', completed: true }
    ]
    let [todoArr, setToDoArr] = useState(initialtodos);

    let deleteToDo = (id) => {
        let filteredToDos = todoArr.filter(todo => todo.id !== id);
        setToDoArr([...filteredToDos]);
    }
    let toggleToDo = (id) => {
        let updatedToDos = todoArr.map(ele => {
            return ele.id === id ? { ...ele, completed: !ele.completed } : ele;
        })
        setToDoArr([...updatedToDos]);
    }
    let addToDo = (text) => {
        let newToDo = { id: todoArr.length + 1, text: text, completed: false };
        todoArr.push(newToDo);
        setToDoArr([...todoArr])
    }
    return <>
        <h2 className='text-center'>This is ToDo CRUD Example</h2>
        <div className='container'>
            <div className='row'>
                <div className='col-sm-8'>
                    <h3 className='text-center'>ToDo List</h3>
                    <ToDoList todoArr={todoArr} deleteToDo={deleteToDo} toggleToDo={toggleToDo} />
                </div>
                <div className='col-sm-4'>
                    <h3 className='text-center'>Add To Do</h3>
                    <AddToDo addToDo={addToDo} />
                </div>
            </div>
        </div>
    </>
}
```

Program-44 Event with parameter
**********
```
import React, { Component } from 'react';
export default class EventDemo1 extends Component {
  f1 = () => {
    console.log('I am F1...')
  }
  add = (a, b) => {
    console.log(`Addition of ${a} & ${b} is ${a + b}`)
  }
  render() {
    return <>
      <button onClick={this.f1} className='btn btn-primary'>btn-1</button>
      <button onClick={() => { this.add(2, 3) }} className='btn btn-secondary m-1'>Add</button>
    </>
  }
}
```

Assignment
==========
-add 2 input boxes (num1,num2)
-add a button, onClick of button find the addition Result

Program-45  Modal
**********
1. npm i react-modal
2. add the below code

```
import React, { useState } from 'react';
import Modal from 'react-modal';
export default function Test() {
  const [modalIsOpen, setIsOpen] = React.useState(false);

  function openModal() {
    setIsOpen(true);
  }
  function closeModal() {
    setIsOpen(false);
  }
  return (
    <div>
      <button onClick={openModal}>Open Modal</button>
      <Modal
        isOpen={modalIsOpen}
        onRequestClose={closeModal}
        contentLabel="Example Modal"
        shouldCloseOnOverlayClick={false}
      >
```

```
          <button onClick={closeModal}>close</button>
          <div>I am a modal</div>
          <div>this is a div</div>
          <input />
        </Modal>
      </div>
    )
  }
```

Program-46  Employee CRUD

**********

```
import React from 'react';

class Crud extends React.Component {
  constructor() {
    super();
    this.state = {
      employees: [
        { eId: 101, name: "sanjay", sal: 5000 },
        { eId: 104, name: "deepak", sal: 8000 },
        { eId: 103, name: "ranjan", sal: 7000 },
        { eId: 102, name: "manoj", sal: 9000 }
      ],
      newEmp: { eId: '', name: '', sal: 0 }
    }
  }
  deleteRow = (ind) => {
    this.state.employees.splice(ind, 1);
    this.setState({ employees: this.state.employees })
  }
  viewRow = (emp) => {
    alert(`${emp.eId} ${emp.name}  ${emp.sal}`)
  }
  updateId = (e) => {
    //this.state.newEmp.eId = e.target.value;
    this.setState({ newEmp: this.state.newEmp.id = e.target.value })
    this.setState({ newEmp: this.state.newEmp })
  }
  updateName = (e) => {
    this.state.newEmp.name = e.target.value;
    this.setState({ newEmp: this.state.newEmp })
  }
  updateSal = (e) => {
    this.state.newEmp.sal = e.target.value;
    this.setState({ newEmp: this.state.newEmp })
  }
  addRow = () => {
    this.state.employees.push(this.state.newEmp);
    this.setState({ employees: this.state.employees })
    //this.setState({ newEmp: { eId: '', name: '', sal: 0 } })
  }

  render() {
    return (
      <div>
```

```
          <table border='1' align='center' width='50%'>
            <tr>
              <th>Sl No.</th>
              <th>Emp Id</th>
              <th>Name</th>
              <th>Salary</th>
              <th>Action</th>
            </tr>
            {this.state.employees.map((emp, ind) => (
              <tr key={emp.eId}>
                <td>{ind}</td>
                <td>{emp.eId}</td>
                <td>{emp.name}</td>
                <td>{emp.sal}</td>
                <td>
                  <button onClick={() => this.deleteRow(ind)}>Delete</button>
                  <button onClick={() => this.viewRow(emp)}>View</button>
                </td>
              </tr>
            ))}
          </table>

          Emp Id:
            <input onKeyUp={this.updateId}></input> <br></br><br></br>
          Name:
            <input onKeyUp={this.updateName}></input> <br></br><br></br>
          Salary:
            <input onKeyUp={this.updateSal}></input> <br></br><br></br>

          <button onClick={this.addRow}>Add a new employee</button>

        </div>
      );
    }
}
export default Crud;
```

Program-47 CRUD (functional component)

**********

```
import React, { useState } from 'react'
import EmployeeList from './EmployeeList'
import AddEmployee from './AddEmployee'
import EditEmployee from './EditEmployee';

export default function EmployeeCrud() {
  const initialEmployees = [
    { eId: 101, name: "sanjay", sal: 5000 },
    { eId: 104, name: "deepak", sal: 8000 },
    { eId: 103, name: "ranjan", sal: 7000 },
    { eId: 102, name: "manoj", sal: 9000 },
  ];
  const [employees, setEmployees] = useState(initialEmployees);
  const [isEdit, setIsEdit] = useState(false);
  const initialEmp = { eId: '', name: '', sal: 0 }
  const [selectedEmployee, setSelectedEmployee] = useState(initialEmp);
```

```
  const deleteEmployee = (idToDelete) => {
    let filteredEmployees = employees.filter(emp => emp.eId !== idToDelete);
    setEmployees([...filteredEmployees])
  }
  const addEmployee = (emp) => {
    employees.push(emp);
    setEmployees([...employees]);
  }
  const editEmployee = (emp) => {
    setSelectedEmployee(emp);
    setIsEdit(true);
  }
  const saveEditedEmployee = (editedEmp) => {
    let updatedEmployees = employees.map(emp => {
      if (emp.eId === editedEmp.eId) {
        return editedEmp;
      } else {
        return emp;
      }
    })
    setEmployees([...updatedEmployees]);
    setIsEdit(false);
  }
  const cancelSave = () => {
    setSelectedEmployee(initialEmp);
    setIsEdit(false);
  }

  return <>
    <h2 className='text-center'>This is EmployeeCRUD Component</h2>
    <div className='container'>
      <div className='row'>
        <div className='col-sm-8'>
          <h3>Employee List</h3>
          <EmployeeList
            employees={employees}
            deleteEmployee={deleteEmployee}
            editEmployee={editEmployee}
          />
        </div>
        <div className='col-sm-4'>
          {
            isEdit ?
              <EditEmployee
                selectedEmployee={selectedEmployee}
                saveEditedEmployee={saveEditedEmployee}
                cancelSave={cancelSave}
                />
              :
              <AddEmployee addEmployee={addEmployee} />
          }
        </div>
      </div>
    </div>
  </>
}
```

```
-------------------------------------------------
import React from 'react'
import { useState } from 'react';

export default function EditEmployee({ selectedEmployee, saveEditedEmployee, cancelSave }) {
   const [emp, setEmp] = useState(selectedEmployee);
   const changeHandler = (e) => {
      const { name, value } = e.target;
      setEmp({ ...emp, [name]: value })
   }
   return <>
      <h3>Edit Employee</h3>
      <p>eId : <input value={emp.eId} name='eId' onChange={(e) => changeHandler(e)} /></p>
      <p>Name : <input value={emp.name} name='name' onChange={(e) => changeHandler(e)} /></p>
      <p>Salary : <input value={emp.sal} name='sal' onChange={(e) => changeHandler(e)} /></p>
      <div className='text-center'>
         <button onClick={cancelSave} className='btn btn-secondary mx-1'>Cancel</button>
         <button onClick={() => saveEditedEmployee(emp)} className='btn btn-
primary'>Save</button>
      </div>
   </>
}

-------------------------------------------------
import React, { useState } from 'react'

export default function AddEmployee({ addEmployee }) {
   const initialEmp = { eId: '', name: '', sal: 0 }
   const [emp, setEmp] = useState(initialEmp);
   const changeHandler = (e) => {
      const { name, value } = e.target;
      setEmp({ ...emp, [name]: value })
   }
   const addMyEmployee = (emp) => {
      addEmployee(emp);
      setEmp(initialEmp);
   }
   return <>
      <p>eId : <input value={emp.eId} name='eId' onChange={(e) => changeHandler(e)} /></p>
      <p>Name : <input value={emp.name} name='name' onChange={(e) => changeHandler(e)} /></p>
      <p>Salary : <input value={emp.sal} name='sal' onChange={(e) => changeHandler(e)} /></p>
      <div className='text-center'>
         <button onClick={() => addMyEmployee(emp)} className='btn btn-primary'>Add
Employee</button>
      </div>
   </>
}
```

Program-48  Pure Component
**********

```
import React, { Component } from 'react'
import { PureComponent } from 'react';
export default class Demo1 extends PureComponent {
   state = {
      name: 'sachin'
   }
   render() {
      console.log('Demo1 render called....');
      setInterval(() => {
         this.setState({name:'sachin1'})
      }, 5000)
      return (
         <div>
            This is Demo1 Component {this.state.name}
         </div>
      )
   }
}
```

Program:49 react memo

**********

```
import React, { useState } from "react";
import Demo2 from "../demo2/Demo2";

export default function Demo() {
 const [counter, setCounter] = useState(0);
 const [cars, setCars] = useState(["tata", "honda", "maruti"]);
 return (

function Demo2(props) {
 console.log("demo-2 render called");
 return (
  <div>
   Demo2
   {props.cars.map((car) => {
    return <li>{car}</li>;
   })}
  </div>
 );
}
export default React.memo(Demo2);
```

Program-50 constructor & getDerivedStateFromProps

**********

```
class Header extends React.Component {
 constructor(props) {
  super(props);
  this.state = {favoritecolor: "red"};
 }
 static getDerivedStateFromProps(props, state) {
  return {favoritecolor: props.favcol };
```

```
  }
  render() {
    return (
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    );
  }
}
```

program-51  componentDidMount

**********

```
class Header extends React.Component {
  constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  componentDidMount() {
    setTimeout(() => {
      this.setState({favoritecolor: "yellow"})
    }, 1000)
  }
  render() {
    return (
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    );
  }
}
```

Program-52   shouldComponentUpdate

**********

```
class Header extends React.Component {
  constructor(props) {
    super(props);
    this.state = {favoritecolor: "red"};
  }
  shouldComponentUpdate() {
    return false;
  }
  changeColor = () => {
    this.setState({favoritecolor: "blue"});
  }
  render() {
    return (
      <div>
      <h1>My Favorite Color is {this.state.favoritecolor}</h1>
      <button type="button" onClick={this.changeColor}>Change color</button>
      </div>
    );
  }
}
```

Program-53   getSnapshotBeforeUpdate

```
***********************************
class Header extends React.Component {
 constructor(props) {
  super(props);
  this.state = {favoritecolor: "red"};
 }
 componentDidMount() {
  setTimeout(() => {
   this.setState({favoritecolor: "yellow"})
  }, 1000)
 }
 getSnapshotBeforeUpdate(prevProps, prevState) {
  document.getElementById("div1").innerHTML =
  "Before the update, the favorite was " + prevState.favoritecolor;
 }
 componentDidUpdate() {
  document.getElementById("div2").innerHTML =
  "The updated favorite is " + this.state.favoritecolor;
 }
 render() {
  return (
   <div>
    <h1>My Favorite Color is {this.state.favoritecolor}</h1>
    <div id="div1"></div>
    <div id="div2"></div>
   </div>
  );
 }
}
```

program-54 componentDidUpdate

```
*********************************
class Header extends React.Component {
 constructor(props) {
  super(props);
  this.state = {favoritecolor: "red"};
 }
 componentDidMount() {
  setTimeout(() => {
   this.setState({favoritecolor: "yellow"})
  }, 1000)
 }
 componentDidUpdate() {
  document.getElementById("mydiv").innerHTML =
  "The updated favorite is " + this.state.favoritecolor;
 }
 render() {
  return (
   <div>
   <h1>My Favorite Color is {this.state.favoritecolor}</h1>
   <div id="mydiv"></div>
   </div>
  );
 }
```

```
}
```

program-60 (componentWillUnmount)

**********

```
class Container extends React.Component {
  constructor(props) {
    super(props);
    this.state = {show: true};
  }
  delHeader = () => {
    this.setState({show: false});
  }
  render() {
    let myheader;
    if (this.state.show) {
      myheader = <Child />;
    };
    return (
      <div>
      {myheader}
      <button type="button" onClick={this.delHeader}>Delete Header</button>
      </div>
    );
  }
}

class Child extends React.Component {
  componentWillUnmount() {
    alert("The component named Header is about to be unmounted.");
  }
  render() {
    return (
      <h1>Hello World!</h1>
    );
  }
}
```

program: 61  useEffect hook

**********

```
import React, { useEffect, useState } from "react";
export default function Demo() {
  const [a, setA] = useState(10);
  const [b, setB] = useState(20);
  const [c, setC] = useState(30);

  useEffect(() => {
    console.log("use effect called");
  }, [a,b]);
  return (
    <>
      <div>Demo</div>
```

```
    <div>
      {a} {b} {c}
    </div>
    <button onClick={() => setA(15)}>
      update A
    </button>
    <button onClick={() => setB(25)}>
      update B
    </button>
    <button onClick={() => setC(35)}>
      update C
    </button>
   </>
  );
}
```

program: 62  componentWillUnmount in functional component

```
**************************************************
import React, { useEffect, useState } from "react";

export default function Demo3() {
  const [color, setColor] = useState("red");
  useEffect(() => {
    console.log("use effect called while mounting/updating");
    return () => {
      console.log("use effect called while unmounting");
    };
  }, []);
  return (
    <div>
      <p id="para1">This is a paragraph</p>
      <button
        onClick={() => {
          setColor("red");
        }}
      >
        click me
      </button>
      <h2>color value is {color}</h2>
    </div>
  );
}
```

Program:63 How to get prev state and props

```
****************************************
import { useEffect, useRef, useState } from "react";
export default function Counter() {
  const [count, setCount] = useState(0);
    const myRef = useRef();
    useEffect(() => {
      myRef.val = count;
    });
```

```
  return (
    <h1>
      Now: {count}, before: {myRef.val}
      <button onClick={()=>{setCount(count+1)}}>Increment</button>
    </h1>
  );
}
```

program-64  Addition using refs

**********

```
import React, { Component } from 'react';
import PropTypes from 'prop-types';

export default class Test extends Component {
  constructor(props) {
    super(props)
    this.myRef1 = React.createRef();
    this.myRef2 = React.createRef();
  }
  add = () => {
    let value_1 = parseInt(this.myRef1.current.value);
    let value_2 = parseInt(this.myRef2.current.value);
    alert(value_1 + value_2);
  }
  render() {
    return (
      <>
        <input name="textbox1" ref={this.myRef1} type="text" />
        <input name="textbox2" ref={this.myRef2} type="text" />
        <button onClick={this.add}>Add</button>
      </>
    )
  }

}
```

---

```
import React, { useRef } from "react";
export default function Addition() {
 const inputRef1 = useRef();
 const inputRef2 = useRef();

 const add = () => {
  const val1 = +inputRef1.current.value;
  const val2 = +inputRef2.current.value;
  alert(val1 + val2);
 };

 return (
  <div>
    num1: <input ref={inputRef1} />
    num2: <input ref={inputRef2} />
    <button onClick={add}>Add</button>
  </div>
 );
```

```
}
```

program-65 call back refs

**********

```
import React, { Component } from 'react'
export default class Test extends Component {
   add = () => {
      let val1 = parseInt(this.input1.value)
      let val2 = parseInt(this.input2.value)
      alert(val1 + val2)
   }
   render() {
      return (
         <div>
            <input type="text" ref={(input) => { this.input1 = input }} />
            <input type="text" ref={(input) => { this.input2 = input }} />
            <button type="button" onClick={this.add}>Add</button>
         </div>
      )
   }
}
```

---

```
export default function Addition() {
 let inputRef1;
 let inputRef2;

 const add = () => {
  const val1 = +inputRef1.value;
  const val2 = +inputRef2.value;
  alert(val1 + val2);
 };

 return (
  <div>
   num1: <input ref={input => { inputRef1 = input; }} />
   num2: <input ref={input => { inputRef2 = input; }} />
   <button onClick={add}>Add</button>
  </div>
 );
}
```

program- 66  Ref Forwarding

**********

```
import React, { Component } from 'react'
import FancyButton from '../FancyButton/FancyButton'

export default class Test extends Component {
   constructor(props) {
      super(props)
      this.myRef = React.createRef();
   }
```

```
  fun1 = () => {
    console.log(this.myRef.current.className)
    console.log(this.myRef.current.id)
  }
  render() {
    return (
      <div>
        <FancyButton ref={this.myRef} >
          <div>button text</div>
        </FancyButton>

        <button onClick={this.fun1}>click me</button>
      </div>
    )
  }
}
```

---

```
import React from 'react';
const FancyButton = React.forwardRef((props, ref) => (
  <button ref={ref} className="FancyButton" id='btn1'>
    {props.children}
  </button>
));
export default FancyButton;
```

Program : display previous State & Current State

**********

```
export default function CounterDemo3() {
  let [counter, updateCounter] = useState(0);
  let prevValue = useRef(0);

  let increment = () => {
    updateCounter(counter + 1);
  }
  useEffect(() => {
    prevValue.current = counter;
  }, [counter]);
  return <>
    <p>Current value - {counter} , previous Value is-{prevValue.current}</p>
    <button onClick={increment}>Increment</button>
  </>
}
```

Program : Search Filter with Deferred Value  (useState, useDeferredValue, useMemo)

==========

```
// App.js
import { useState, useDeferredValue, useMemo } from "react";
const ITEMS = Array.from({ length: 10000 }, (_, i) => `Item ${i + 1}`);

export default function SearchFilter() {
```

```
  const [query, setQuery] = useState("");
  const deferredQuery = useDeferredValue(query);

  const filtered = useMemo(() => {
    return ITEMS.filter(i =>
      i.toLowerCase().includes(deferredQuery.toLowerCase())
    );
  }, [deferredQuery]);

  return (
    <div style={{ padding: 20 }}>
      <h2>Search Filter</h2>
      <input
        value={query}
        onChange={e => setQuery(e.target.value)}
        placeholder="Type to search"
      />
      <p>Showing {filtered.length} results</p>
      <ul>
        {filtered.slice(0, 20).map(i => (
          <li key={i}>{i}</li>
        ))}
      </ul>
    </div>
  );
}
```

program-67  Form Submit

**********

```
import React, { useState } from "react";

export default function Demo() {
  const [fname, setFname] = useState("sachin");
  const [lname, setLname] = useState("tendulkar");

  const submitHandler = (e) => {
    e.preventDefault();
    alert(fname+' '+lname);
  };

  return (
    <>
      <form onSubmit={submitHandler}>
        First Name:
        <input name="fname" value={fname} onChange={(e) => setFname(e.target.value)} />
        Last Name:
        <input name="lname" value={lname} onChange={(e) => setLname(e.target.value)} />
        <br />
        <br />
        <button>submit</button>
      </form>
    </>
  );
}
```

Program-68 UnControlled Form Elements
**********

```
import React, { useState } from 'react'
import { useRef } from 'react';

export default function FormDemo2() {
  let myRef1 = useRef('')
  let myRef2 = useRef('')
  const submitHandler = (e) => {
    e.preventDefault();
    const fname = myRef1.current.value;
    const lname = myRef2.current.value;
    console.log(fname, lname)
  }
  return <form onSubmit={submitHandler}>
    First Name:
    <input name="fname" defaultValue={'Sachin'} ref={myRef1} />
    Last Name:
    <input name="lname" defaultValue={'Tendulkar'} ref={myRef2} />
    <br /><br />
    <button>submit</button>
  </form>
}
```

Program-70 Multiple input fields
**********

```
import React from 'react';

class MyForm extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      username: '',
      age: null,
    };
  }
  mySubmitHandler = (event) => {
    event.preventDefault();
    alert("You are submitting " + this.state.username + " " + this.state.age);
  }
  myChangeHandler = (event) => {
    let nam = event.target.name;
    let val = event.target.value;
    console.log(nam, val)
    this.setState({ [nam]: val });
  }
  render() {
    return (
      <form onSubmit={this.mySubmitHandler}>
        <h1>Hello {this.state.username} {this.state.age}</h1>
        <p>Enter your name:</p>
        <input
```

```
            type='text'
            name='username'
            onChange={this.myChangeHandler}
          />
          <p>Enter your age:</p>
          <input
            type='text'
            name='age'
            onChange={this.myChangeHandler}
          />
          <input type='submit' />
          <input type='reset' value='clear' />
        </form>
      );
    }
}
export default MyForm;
```

Program-71  form data in an object

```
**********
import React, { useState } from "react";

export default function Form1() {
 const [user, setUser] = useState({ fname: "sachin", lname: "tendulkar" });

 const submitHandler = (e) => {
  e.preventDefault();
  console.log(user);
 };

 return (
  <>
   <form onSubmit={submitHandler}>
    First Name:
    <input
     name="fname"
     value={user.fname}
     onChange={(e) => setUser({ ...user, fname: e.target.value })}
    />
    Last Name:
    <input
     name="lname"
     value={user.lname}
     onChange={(e) => setUser({ ...user, lname: e.target.value })}
    />
    <br />
    <br />
    state data: {user.fname + " " + user.lname}
    <button>submit</button>
   </form>
  </>
 );
}
```

program-72  Form Validation while typing

```
**************************************
import React, { useState } from "react";

export default function Demo() {
  const [fname, setFname] = useState("sachin");
  const [lname, setLname] = useState("tendulkar");
  const [fNameErrorMsg, setfNameErrorMsg] = useState("");
  const [lNameErrorMsg, setlNameErrorMsg] = useState("");

  const submitHandler = (e) => {
    e.preventDefault();
    alert(fname + " " + lname);
  };

  const changeHandler = (e) => {
    const inputField = e.target.name;
    if (inputField == "fname") {
      setFname(e.target.value);
      if (!e.target.value) {
        setfNameErrorMsg("firstname should not be empty");
      } else if (e.target.value.length < 5) {
        setfNameErrorMsg("firstname should have atleast 5 chars");
      } else {
        setfNameErrorMsg("");
      }
    } else if (inputField == "lname") {
      setLname(e.target.value);
      if (!e.target.value) {
        setlNameErrorMsg("lastname should not be empty");
      } else if (e.target.value.length < 5) {
        setlNameErrorMsg("lastname should have atleast 5 chars");
      } else {
        setlNameErrorMsg("");
      }
    } else {
    }
  };

  return (
    <>
      <form onSubmit={submitHandler}>
        First Name:
        <input name="fname" value={fname} onChange={changeHandler} />
        <span class="text-danger">{fNameErrorMsg}</span> <br/>
        Last Name:
        <input name="lname" value={lname} onChange={changeHandler} />
        <span class="text-danger">{lNameErrorMsg}</span>
        <br /><br />
        <button>submit</button>
      </form>
    </>
  );
}
```

Controlled Form Example

**********

```
import React, { useState } from 'react'
export default function FormDemo2() {
  const defaultFormValues = {
    name: '',
    age: '',
    gender: '',
    dob: '',
    interests: [],
    country: '',
    bio: ''
  }
  const [formData, setFormData] = useState(defaultFormValues);
  const handleSubmit = (event) => {
    event.preventDefault();
    console.log(formData);
  }
  const handleReset = () => {
    setFormData(defaultFormValues);
  }
  const isFormValid = formData.name !== '' && formData.age !== '';
  const handleChange = (event) => {
    const { name, value, type, checked } = event.target;

    if (type === 'checkbox') {
      if (checked) {
        setFormData({ ...formData, interests: [...formData.interests, value] });
      } else {
        setFormData({
          ...formData,
          interests: [...formData.interests.filter(interest => interest !== value)]
        });
      }
    } else {
      setFormData({ ...formData, [name]: value });
    }
  }
  return <>
    <h3 className='text-center'>React Controlled Form</h3>
    <div className='container'>
      <div className='row'>
        <div className='col-sm-6 offset-sm-3'>
          <div className='border border-3 rounded-3 p-2 m-2'>
            <form onSubmit={handleSubmit}>
              <p>
                <label>Name:</label>
                <input name='name' value={formData.name} onChange={handleChange}
required></input>
              </p>
              <p>
                <label>Age:</label>
                <input name='age' value={formData.age} onChange={handleChange}></input>
              </p>
              <p>
                <label>Gender:</label>
```

```
                    <input name='gender' type='radio'
                      value='male' checked={formData.gender === 'male'}
                      onChange={handleChange} /> Male
                    <input name='gender' type='radio'
                      value='female' checked={formData.gender === 'female'}
                      onChange={handleChange} /> female
                  </p>
                  <p>
                    <label>DOB:</label>
                    <input name='dob' type='date'
                      value={formData.dob} onChange={handleChange} />
                  </p>
                  <p>
                    <label>Interests:</label>
                    <input name='interests' type='checkbox'
                      value='reading' checked={formData.interests.includes('reading')}
                      onChange={handleChange} />Reading

                    <input name='interests' type='checkbox'
                      value='swimimg' checked={formData.interests.includes('swimimg')}
                      onChange={handleChange} />Swimimg
                  </p>
                  <p>
                    <label>Country:</label>
                    <select name='country' value={formData.country} onChange={handleChange}>
                      <option value=''>Select a Country</option>
                      <option value='india'>India</option>
                      <option value='usa'>USA</option>
                      <option value='srilanka'>Srilanka</option>
                      <option value='canada'>Canada</option>
                      <option value='mexico'>Mexico</option>
                    </select>
                  </p>
                  <p>
                    <label>Bio:</label>
                    <textarea rows='2' cols='30' name='bio'
                      value={formData.bio} onChange={handleChange}>

                    </textarea>
                  </p>
                  <input type='reset' value='Clear' className='mx-2 btn btn-danger'
                    onClick={handleReset} />
                  <input type='submit' value='Submit' className='btn btn-primary'
disabled={!isFormValid} />
                </form>

                <pre>
                  {JSON.stringify(formData,null,4)}
                </pre>
              </div>
            </div>
          </div>
        </div>
      </>
}
```

Formik & Yup
==========

```
import React, { useState } from 'react';
import { Formik, Form, Field, ErrorMessage } from "formik";
import * as Yup from "yup";

const initialFormValue = {
   name: "", email: "", password: "", confirmPassword: "",
   gender: "", role: "", skills: [], acceptTerms: false
}
const skillsList = ["JavaScript", "React", "Node.js", "Python"];
const validationSchema = Yup.object({
   name: Yup.string().required("Name is required"),
   email: Yup.string()
      .email("Invalid email address")
      .required("Email is required"),
   password: Yup.string()
      .min(6, "Password must be at least 6 characters")
      .required("Password is required"),
   confirmPassword: Yup.string()
      .oneOf([Yup.ref("password"), null], "Passwords must match")
      .required("Confirm password is required"),
   gender: Yup.string().required("Gender is required"),
   role: Yup.string().required("Role is required"),
   skills: Yup.array().min(1, "Select at least one skill"),
   acceptTerms: Yup.boolean().oneOf([true], "You must accept the terms"),
});

export default function FormDemo3() {
   const [formData, setFormData] = useState(null);

   return <>
      <h3 className='text-center'>React Form Using Formik & Yup</h3>
      <div className='col-sm-6 offset-sm-3'>
         <div className='border border-3 rounded-3 p-2 m-2'>
            <Formik
               initialValues={initialFormValue}
               validationSchema={validationSchema}
               onSubmit={(values, { resetForm }) => {
                  setFormData(values);
                  resetForm();
               }}
            >
               {({ values, isSubmitting }) => (
                  <Form>
                     <div className='my-1'>
                        <label>Name:</label>
                        <Field name="name" type="text" />
                        <div className="text-danger"><ErrorMessage name="name" /></div>
                     </div>
                     <div className='my-1'>
                        <label>Email:</label>
                        <Field name="email" type="email" />
                        <div className="text-danger"><ErrorMessage name="email" /></div>
                     </div>
```

```
      <div className='my-1'>
        <label>Password:</label>
        <Field name="password" type="password" />
        <div className="text-danger"><ErrorMessage name="password" /></div>
      </div>
      <div className='my-1'>
        <label>Confirm Password:</label>
        <Field name="confirmPassword" type="password" />
        <div className="text-danger"><ErrorMessage name="confirmPassword" /></div>
      </div>
      <div className='my-1'>
        <label>Gender:</label>
        <Field as="select" name="gender">
          <option value="">Select</option>
          <option value="male">Male</option>
          <option value="female">Female</option>
        </Field>
        <div className="text-danger"><ErrorMessage name="gender" /></div>
      </div>
      <div className='my-1'>
        <label>Role:</label>
        <label>
          <Field type="radio" name="role" value="developer" />
          Developer
        </label>
        <label>
          <Field type="radio" name="role" value="designer" />
          Designer
        </label>
        <div className="text-danger"><ErrorMessage name="role" /></div>
      </div>
      <div className='my-1'>
        <label>Skills:</label>
        {skillsList.map((skill) => (
          <label key={skill} style={{ display: "block" }}>
            <Field
              type="checkbox"
              name="skills"
              value={skill}
            />
            {skill}
          </label>
        ))}
        <div className="text-danger"><ErrorMessage name="skills" /></div>
      </div>
      <div className='my-1'>
        <label>
          <Field type="checkbox" name="acceptTerms" />
          I accept the terms and conditions
        </label>
        <div className="text-danger"><ErrorMessage name="acceptTerms" /></div>
      </div>
      <button type="submit" disabled={isSubmitting} style={{ marginTop: "10px" }}>
        Submit
      </button>
    </Form>
  )}
```

```
          </Formik>
          {formData && (
            <div style={{ marginTop: "20px", background: "#f9f9f9", padding: "15px" }}>
              <h3>Submitted Data:</h3>
              <pre>{JSON.stringify(formData, null, 2)}</pre>
            </div>
          )}
        </div>
      </div>
    </>
}
```

program-73  Form Validation while submitting

```
**********************************************
import React from 'react';
class MyForm extends React.Component {
 constructor(props) {
  super(props);
  this.state = {
   username: '',
   age: null,
  };
 }
 mySubmitHandler = (event) => {
  event.preventDefault();
  let age = this.state.age;
  if (!Number(age)) {
   alert("Your age must be a number");
  }
 }
 myChangeHandler = (event) => {
  let nam = event.target.name;
  let val = event.target.value;
  this.setState({[nam]: val});
 }
 render() {
  return (
   <form onSubmit={this.mySubmitHandler}>
   <h1>Hello {this.state.username} {this.state.age}</h1>
   <p>Enter your name:</p>
   <input
    type='text'
    name='username'
    onChange={this.myChangeHandler}
   />
   <p>Enter your age:</p>
   <input
    type='text'
    name='age'
    onChange={this.myChangeHandler}
   />
   <br/>
   <br/>
   <input type='submit' />
```

```
    </form>
   );
  }
}
export default MyForm;
```

Program-74 Error Message While Typing

**********

```
import React from 'react';
class MyForm extends React.Component {
   constructor(props) {
      super(props);
      this.state = {
         username: '',
         age: null,
         nameErrorMsg: '',
         ageErrorMsg: '',
      };
   }
   myChangeHandler = (event) => {
      let nam = event.target.name;
      let val = event.target.value;
      let err1= '';
      let err2 = '';
      if (nam === "username") {
         if (val.length < 5) {
            err1 = <strong>Your name must contain 5 chars</strong>;
         }
      }
      if (nam === "age") {
         if (val !== "" && !Number(val)) {
            err2 = <strong>Your age must be a number</strong>;
         }
      }
      this.setState({ nameErrorMsg: err1 });
      this.setState({ ageErrorMsg: err2 });
      this.setState({ [nam]: val });
   }
   render() {
      return (
         <form>
            <h1>Hello {this.state.username} {this.state.age}</h1>
            <p>Enter your name:</p>
            <input
               type='text'
               name='username'
               onChange={this.myChangeHandler}
            />
            {this.state.nameErrorMsg}

            <p>Enter your age:</p>
            <input
               type='text'
               name='age'
               onChange={this.myChangeHandler}
```

```
            />
            {this.state.ageErrorMsg}
          </form>
      );
    }
}

export default MyForm;
```

Program-75  Error Message onSubmit

**********

```
import React from 'react';

class MyForm extends React.Component {
   constructor(props) {
      super(props);
      this.state = {
         username: '',
         age: null,
         nameErrorMsg: '',
         ageErrorMsg: '',
      };
   }
   myChangeHandler = (event) => {
      let nam = event.target.name;
      let val = event.target.value;
      this.setState({ [nam]: val });
   }
   mySubmitHandler = (event) => {
      event.preventDefault();

      let err1 = '';
      let err2 = '';

      if (this.state.username.length < 5) {
         err1 = <strong>Your name mustcontain 5 chars</strong>;
      }
      if (this.state.age !== "" && !Number(this.state.age)) {
         err2 = <strong>Your age must be a number</strong>;
      }
      this.setState({ nameErrorMsg: err1, ageErrorMsg: err2 });
      //alert("You are submitting " + this.state.username + " " + this.state.age);
   }
   render() {
      return (
         <form onSubmit={this.mySubmitHandler}>
            <h1>Hello {this.state.username} {this.state.age}</h1>
            <p>Enter your name:</p>
            <input
               type='text'
               name='username'
               onChange={this.myChangeHandler}
            />
            {this.state.nameErrorMsg}
```

```
            <p>Enter your age:</p>
            <input
               type='text'
               name='age'
               onChange={this.myChangeHandler}
            />
            {this.state.ageErrorMsg}

            <input type="submit" />
         </form>
      );
   }
}
export default MyForm;
```

program-76  Load a Form With Default Values

**********

```
import React from 'react';

class MyForm extends React.Component {
   constructor(props) {
      super(props);
      this.state = {
         username: 'sachin',
         age: 35,
         address: 'bang'
      };
   }
   myChangeHandler = (event) => {
      let nam = event.target.name;
      let val = event.target.value;
      this.setState({ [nam]: val });
   }
   mySubmitHandler = (event) => {
      event.preventDefault();
      alert(this.state.username + " " + this.state.age + " " + this.state.address);
   }
   render() {
      return (
         <form onSubmit={this.mySubmitHandler}>
            <h2>Hello {this.state.username} age is {this.state.age}
               address is {this.state.address} </h2>
            <p>Enter your name:</p>
            <input
               type='text'
               name='username'
               onChange={this.myChangeHandler}
               value={this.state.username}
            />
            <p>Enter your age:</p>
            <input
               type='text'
               name='age'
               onChange={this.myChangeHandler}
               value={this.state.age}
```

```
            />
            <p>Enter your Address:</p>
            <textarea
               name='address'
               onChange={this.myChangeHandler}
               value={this.state.address} />

            <br></br>
            <input type="submit" />
         </form>
      );
   }
}
export default MyForm;
```

Program-77  DropDown

**********

```
class MyForm extends React.Component {
 import React, { Component } from 'react'

export default class Test extends Component {
   constructor(props) {
      super(props);
      this.state = {
         mycar: 'Volvo'
      };
   }
   changeHandler = (event) => {
      this.setState({ mycar: event.target.value })
   }
   render() {
      return (
         <form>
            <select value={this.state.mycar} onChange={this.changeHandler}>
               <option value="Ford">Ford</option>
               <option value="Volvo">Volvo</option>
               <option value="Fiat">Fiat</option>
               <option value="Maruti">Maruti</option>
               <option value="Tata">Tata</option>
            </select>
            <div>Selected Car: {this.state.mycar}</div>
         </form>
      );
   }
}

}
```

Program-79  Fetch() Example

**********

```
import React, { Component } from 'react'

export default class Test extends Component {
   constructor(props) {
      super(props)
      this.state = {
         users: []
      }
   }

   fetchUsers = () => {
      const myURL = 'https://jsonplaceholder.typicode.com/users';

      fetch(myURL)
         .then((response) => response.json())
         .then((data) => {
            console.log(data);
            this.setState({ 'users': data })
         });
   }
   componentDidMount() {
      this.fetchUsers();
   }
   render() {
      return (
         <table className='table table-bordered table-striped'>
            <thead>
               <tr>
                  <th>userID</th>
                  <th>name</th>
                  <th>email</th>
               </tr>
            </thead>
            <tbody>
               {this.state.users.map((user, ind) => {
                  return (<tr key={ind}>
                     <td>{user.id}</td>
                     <td>{user.name}</td>
                     <td>{user.email}</td>
                  </tr>)
               })}
            </tbody>
         </table>
      )
   }
}
-----------------------------------------------------
import React, { useEffect, useState } from "react";
export default function Comments() {
  const [comments, setComments] = useState([]);
  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/comments")
      .then((response) => response.json())
      .then((result) => {
        setComments(result);
      });
  });
```

```
 return (
  <>
   <h1>Comment List</h1>

   <table className="table table-bordered">
    {comments.map((comment) => (
     <tr key={comment.id}>
      <td>{comment.id}</td>
      <td>{comment.name}</td>
      <td>{comment.email}</td>
      <td>{comment.body}</td>
     </tr>
    ))}
   </table>
  </>
 );
}
```

Program-80  HTTP CRUD using Axios

**********

```
import React, { Component } from "react";
import Modal from "react-modal";
import axios from "axios";
import Swal from "sweetalert2";
import "./Curd.css";

export default class Curd extends Component {
 constructor(props) {
  super(props);

  this.state = {
   modalIsOpen: false,
   isEdit: false,
   users: [],
   id: "",
   name: "",
   email: "",
   address: "",
   phone: "",
  };
 }
 myChangeHandler = (event) => {
  let nam = event.target.name;
  let val = event.target.value;
  this.setState({ [nam]: val });
 };

 mySubmitHandler = (event) => {
  event.preventDefault();
  let output = {};
  output = {
   name: this.state.name,
   email: this.state.email,
```

```
    phone: this.state.phone,
    address: this.state.address,
  };
  if (this.state.isEdit) {
    this.updateEmployees(output, this.state.id);
  } else {
    this.addEmployee(output);
  }
};
fetchEmployees = () => {
  const myURL = "http://localhost:4000/users";
  axios.get(myURL).then((result) => {
    this.setState({ users: result.data });
  });
};

addEmployee(data) {
  const myURL = "http://localhost:4000/users";
  axios.post(myURL, data).then((response) => {
    Swal.fire("Employee got added", "", "success");
    this.closeModal();
    this.fetchEmployees();
  });
}

updateEmployees = (data, empId) => {
  axios
    .put(`http://localhost:4000/users/${empId}`, data)
    .then((response) => {
      Swal.fire("Employee got updated", "", "success");
      this.closeModal();
      this.fetchEmployees();
    })
    .then((data) => {
      console.log(data);
    })
    .catch((error) => {
      console.log(error);
    });
};
deleteEmployee = (empId) => {
  axios.delete(`http://localhost:4000/users/${empId}`).then((result) => {
    this.fetchEmployees();
  });
};
componentDidMount() {
  this.fetchEmployees();
}
openModal = () => {
  this.setState({ modalIsOpen: true, isEdit: false });
};
openEditModal = (id, name, email, address, phone) => {
  this.setState({
    id: id,
    name: name,
    email: email,
    address: address,
```

```
    phone: phone,
    isEdit: true,
    modalIsOpen: true,
   });
 };
 closeModal = () => {
   this.setState({ modalIsOpen: false });
 };
 render() {
   let nameValue;
   if (this.state.isEdit) {
     nameValue = this.state.name;
   }
   let emailValue;
   if (this.state.isEdit) {
     emailValue = this.state.email;
   }
   let addressValue;
   if (this.state.isEdit) {
     addressValue = this.state.address;
   }
   let phoneVlue;
   if (this.state.isEdit) {
     phoneVlue = this.state.phone;
   }
   return (
     <div>
       <div className="container">
         <div className="table-responsive">
           <div className="table-wrapper">
             <div className="table-title">
               <div className="row">
                 <div className="col-sm-6">
                   <h2 style={{textAlign : 'left'}}>
                     Manage <b>Employees</b>
                   </h2>
                 </div>
                 <div className="col-sm-6">
                   <a
                     className="btn btn-success"
                     onClick={this.openModal}
                     data-toggle="modal"
                   >
                     <GrAddCircle/>
                     <span>Add New Employee</span>
                   </a>
                 </div>
               </div>
             </div>
             <table className="table table-striped table-hover">
               <thead>
                 <tr>
                   <th>ID </th>
                   <th>Name</th>
                   <th>Email</th>
                   <th>Address</th>
                   <th>Phone</th>
```

```
            <th>Actions</th>
          </tr>
        </thead>
        <tbody>
          {this.state.users.map((user, ind) => {
            return (
              <tr key={user.id}>
                <td>{user.id}</td>
                <td>{user.name}</td>
                <td>{user.email}</td>
                <td>{user.address}</td>
                <td>{user.phone}</td>
                <td>
                  <a
                    className="edit"
                    onClick={() =>
                      this.openEditModal(
                        user.id,
                        user.name,
                        user.email,
                        user.address,
                        user.phone
                      )
                    }
                  >
                    <AiFillEdit />
                  </a>
                  <a
                    className="delete"
                    href="/"
                    onClick={() => this.deleteEmployee(user.id)}
                  >
                    Delete
                  </a>
                </td>
              </tr>
            );
          })}
        </tbody>
      </table>
    </div>
  </div>
</div>
<Modal
  isOpen={this.state.modalIsOpen}
  onRequestClose={this.closeModal}
  ariaHideApp={false}
  contentLabel="Example Modal"
>
  <button onClick={this.closeModal}>close</button>
  <form onSubmit={this.mySubmitHandler}>
    <div className="modal-header">
      <h4 className="modal-title">Add Employee</h4>
    </div>
    <div className="modal-body">
      <div className="form-group">
        <label>Name</label>
```

```
            <input
              type="text"
              name="name"
              onChange={this.myChangeHandler}
              className="form-control"
              value={nameValue}
              required
            />
          </div>
          <div className="form-group">
            <label>Email</label>
            <input
              type="email"
              name="email"
              onChange={this.myChangeHandler}
              className="form-control"
              value={emailValue}
              required
            />
          </div>
          <div className="form-group">
            <label>Phone</label>
            <input
              type="number"
              name="phone"
              onChange={this.myChangeHandler}
              className="form-control"
              value={phoneVlue}
              required
            />
          </div>
          <div className="form-group">
            <label>Address</label>
            <textarea
              name="address"
              onChange={this.myChangeHandler}
              className="form-control"
              value={addressValue}
              required
            ></textarea>
          </div>
        </div>
        <div className="modal-footer">
          <input
            type="button"
            className="btn btn-default"
            onClick={this.closeModal}
            data-dismiss="modal"
            value="Cancel"
          />
          <input type="submit" className="btn btn-info" value="Save" />
        </div>
      </form>
    </Modal>
  </div>
 );
}
```

```
}
```

Program:81 CRUD example Functional component

**********

```
import axios from "axios";
import React, { useState } from "react";
import Swal from "sweetalert2";

export default function Crud() {
 const [users, setUsers] = useState([]);
 const [name, setName] = useState("");
 const [email, setEmail] = useState("");
 const [phone, setPhone] = useState("");
 const [address, setAddress] = useState("");

 const fetchUsers = async function() {
  let response = await axios.get("http://localhost:4000/users");
  setUsers(response.data);
 };
 const addUser = async function() {
  const newUser = {
     "name": name,
     "email": email,
     "phone": phone,
     "address": address,
  }
  let response = await axios.post("http://localhost:4000/users", newUser);
  Swal.fire("Employee Added Successfully", "", "success");
  fetchUsers();
 };
 const deleteUser = (id) => {
  axios.delete(`http://localhost:4000/users/${id}`).then((res) => {
   Swal.fire("Employee Deleted Successfully", "", "success");
   fetchUsers();
  });
 };
 const editUser = (id) => {

 };
 const mySubmitHandler = function(event) {
  event.preventDefault();
  addUser();
 };
 return (
  <>
   <h1 className="text-center">CRUD Example</h1>
   <hr />
   <button onClick={fetchUsers}>get data</button>

   <div className="container">
    <div className="row">
     <div className="col-sm-8">
      <table className="table table-bordered table-striped table-responsive">
       <thead>
        <tr>
```

```
            <th>Name</th>
            <th>email</th>
            <th>phone</th>
            <th>address</th>
            <th>Action</th>
           </tr>
          </thead>
          <tbody>
           {users.length > 0 ? (
            users.map((user) => (
             <tr key={user.id}>
              <td>{user.name}</td>
              <td>{user.email}</td>
              <td>{user.phone}</td>
              <td>{user.address}</td>
              <td>
               <button className="btn btn-danger" onClick={() => deleteUser(user.id)}>
                DELETE
               </button>  
               <button className="btn btn-secondary" onClick={() => editUser(user.id)}>
                EDIT
               </button>
              </td>
             </tr>
            ))
           ) : (
            <tr>
             <td colSpan={5} className='text-center'>No data yet</td>
            </tr>
           )}
          </tbody>
         </table>
        </div>
        <div className="col-sm-4">
         <form onSubmit={mySubmitHandler}>
          Name: <input name="name" value={name} onChange={e=>setName(e.target.value)} /> <br
/><br />
          email: <input name="email" value={email} onChange={e=>setEmail(e.target.value)} /> <br
/><br />
          phone: <input name="phone" value={phone} onChange={e=>setPhone(e.target.value)} /> <br
/><br />
          address: <input name="address" value={address} onChange={e=>setAddress(e.target.value)}
/><br />
          <br />
          <button className="btn btn-primary">Add Employee</button>
         </form>
        </div>
       </div>
      </div>
    </>
  );
}
```

program: 82 axios & async-await

```
*****************************
import React, { Component } from 'react';
import axios from 'axios';

export default class Test extends Component {
   constructor(props) {
      super(props)
      this.state = {
         users: []
      }
   }
   async fetchUsers() {
      try {
         let response = await axios.get('https://jsonplaceholder.typicode.com/users');
         this.setState({ users: response.data })
      }
      catch (error) {
         console.log(error)
      }
   }
   componentDidMount() {
      this.fetchUsers()
   }
   render() {
      return (
         <table className='table table-bordered table-striped'>
            <thead>
               <tr>
                  <th>userID</th>
                  <th>name</th>
                  <th>email</th>
               </tr>
            </thead>
            <tbody>
               {this.state.users.map((user, ind) => {
                  return (<tr key={ind}>
                     <td>{user.id}</td>
                     <td>{user.name}</td>
                     <td>{user.email}</td>
                  </tr>)
               })}
            </tbody>
         </table>
      )
   }
}


Program-83 Request Interceptor
*****************************
axios.interceptors.request.use( req => {
 req.meta = req.meta || {}
 req.meta.requestStartedAt = new Date().getTime();
 return req;
});
```

Program-84 Response Interceptor
**********
```
axios.interceptors.response.use(res => {
      res.durationInMs = new Date().getTime() - res.config.meta.requestStartedAt
      return res;
   },
      err => {
         err.durationInMs = new Date().getTime() - err.config.meta.requestStartedAt
         throw err;
      });
```

Program:85 Request & Response Interceptor
**********
```
import axios from 'axios';
axios.interceptors.request.use( req => {
 req.meta = req.meta || {}
 req.meta.requestStartedAt = new Date().getTime();
 return req;
});

axios.interceptors.response.use(res => {
 res.durationInMs = new Date().getTime() - res.config.meta.requestStartedAt
 return res;
},
err => {
 err.durationInMs = new Date().getTime() - err.config.meta.requestStartedAt
 throw err;
});

(async () => {
 try {
   const headers = { Accept: 'application/json', 'Accept-Encoding': 'identity' };
   const githubUserName = 'abraham';

   const url = `https://api.github.com/users/${githubUserName}`;
   console.log(`Sending a GET reqeust to: ${url}`);
   const { data, durationInMs } = await axios.get(url, { headers });
   console.log(`Github username ${githubUserName} has real name: ${data?.name} and works at
${data?.company}`,);
   console.log(`Successful response took ${durationInMs} milliseconds`);

   //the below request will fail
   const nonExistinggithubUserName = 'thisUserShouldNotBeOnGithub';
   const failingUrl = `https://api.github.com/users/${nonExistinggithubUserName}`;
   console.log(`Sending a GET reqeust to: ${failingUrl}`);
   const response = await axios.get(failingUrl, headers);
   console.log(response.data); //it will never reach here as get will hit a 404, it will go to the catch part
 } catch(err) {
   console.log(`Error message : ${err.message} - `, err.code);
   console.log(`Error response took ${err.durationInMs} milliseconds`);
 }
})();
```

Program: graphQL
==========
1. install apollo-client and graphQL
  npm install @apollo/client graphql

2. Initialize ApolloClient (index.js)
```
  import { ApolloClient, InMemoryCache, ApolloProvider } from '@apollo/client';
  const client = new ApolloClient({
            uri: 'http://localhost:5000/',
            cache: new InMemoryCache()
          });
```

3. Connect your Apollo-client to React / Set Up ApolloProvider
```
 <ApolloProvider client={client}>
  <App />
 </ApolloProvider>
```

4. Fetch data with useQuery
```
  import { gql, useQuery } from '@apollo/client';

  const GET_COMMENTS_QUERY = gql`
   query comments {
    comments {
     id
     name
                    email
    }
   }
  `;
  const { loading, error, data } = useQuery(GET_COMMENTS_QUERY);
```

Program- HOC For Logging
==========
```
import React, { useState, useEffect, useRef } from 'react';
const LoggerHOC = (InputComponent) => {
  const EnhancedComponent = (props) => {
                  const isFirstRender = useRef(true);
    useEffect(() => {
      console.log(`${InputComponent.name} Mounted`);
      return () => {
        console.log(`${InputComponent.name} UnMounted`);
      }
    }, []);
    useEffect(() => { // To Make sure this effect gets called only on update not on mount
      if (isFirstRender.current) {
        isFirstRender.current = false;
      } else {
        console.log(`${InputComponent.name} Updated`);
      };
    });
    return <InputComponent {...props} />
  }
  return EnhancedComponent;
```

```
}
export default LoggerHOC;
```

program-78 without HOC
**********

```
import React,{ useState } from "react";
export default function ClickCounter() {
 const [count, setCount] = useState(0);

 const increment = () => {
  setCount(count + 1);
 };
 return (
  <>
   <h2>In Click Counter Component - count is {count}</h2>
   <button onClick={increment}>Click me</button>
  </>
 );
}
```

---

```
import React, { useState } from "react";

export default function HoverCounter() {
 const [count, setCount] = useState(0);
 const increment = () => {
  setCount(count + 1);
 };
 return (
  <>
   <h2>In Hover Counter Component - count is {count}</h2>
   <button onMouseOver={increment}>Hover me</button>
  </>
 );
}
```

Program-79  With HOC
**********

withcounter.js
---

```
import React, { useState } from "react";
const UpdatedComponent = (OriginalComponent) => {
 function CounterHOC() {
  const [count, setCount] = useState(0);
  const increment = () => {
   setCount(count + 1);
  };
```

```
   return <OriginalComponent count={count} increment={increment} />;
  }
  return CounterHOC;
};
export default UpdatedComponent;
```

clickCounter.js

```
---------------
import React from "react";
import CounterHOC from "./CounterHOC";
function ClickCounterWithHOC(props) {
  return (
    <button onClick={props.increment}> Clicked {props.count} Times</button>
  );
}
export default CounterHOC(ClickCounterWithHOC);
```

hovercounter.js

```
--------------
import React from "react";
import CounterHOC from "./CounterHOC";
function HoverCounterWithHOC(props) {
  return (
    <button onMouseOver={props.increment}> Clicked {props.count} Times</button>
  );
}
export default CounterHOC(HoverCounterWithHOC);
```

Program- Logging

```
==========================
const withLogger = (WrappedComponent) => {
  const WithLogger = (props) => {
    useEffect(() => {
      // Log data on component mount
      console.log(`Component ${WrappedComponent.name} mounted.`);
      return () => {
        // Log data on component unmount
        console.log(`Component ${WrappedComponent.name} unmounted.`);
      };
    }, []);

    useEffect(() => {
      // Log data on component update
      console.log(`Component ${WrappedComponent.name} updated.`);
    });

    return <WrappedComponent {...props} />;
  };

  WithLogger.displayName = `withLogger(${WrappedComponent.displayName ||
WrappedComponent.name})`;
  return WithLogger;
```

```
};

export default withLogger;
```

program:86 Routing
**********
index.js
---
```
import { BrowserRouter } from 'react-router-dom';
  <BrowserRouter>
      <App />
  </BrowserRouter>
```

center.js
---
```
import { Route, Routes } from "react-router";
<Routes>
    <Route exact path="/" element={<Home />} />
    <Route exact path="/home" element={<Home />} />
    <Route exact path="/aboutus" element={<Aboutus />} />
    <Route exact path="/careers" element={<Careers />} />
    <Route exact path="/products" element={<Products />} />
    <Route path="*" element={<NotFound />} />
   </Routes>
```

nav.js
---
```
<ul class="nav navbar-nav">
            <li class="active">
                        <Link to="/products">Product</Link>
            </li>
   <li>
                        <Link to="/greet">greet</Link>
            </li>
            <li>
                        <Link to="/http">http</Link>
            </li>
            <li>
                        <Link to="/parent">parent</Link>
            </li>
</ul>
```

Program : 87 Protected Routes
==========
```
<Route exact path="/careers" element={
        <ProtectedRoute>
         <Careers></Careers>
```

```
        </ProtectedRoute>
      } />
```

---

```
import React from "react";
import { Navigate } from "react-router-dom";
export default function ProtectedRoute({ children }) {
 const role = "student";
 if (role === "student") {
    alert('you are not allowed to go to this route')
  return <Navigate to="/home" replace />;
 }
 return children;
}
```

Lazy Loading

==========

```
const ProductList = React.lazy(() => import("./components/list/product_list"));

<Suspense fallback={<h1>Loading...</h1>}>
  <Routes>
    <Route exact path="products" element={<ProductList />} />
  </Routes>
</Suspense>
```

program-88 useState Hook

**********

```
import React, { Component } from 'react';

export default class Demo1 extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
  }
  incrementCount = () => {
    this.setState({ count: this.state.count + 1 })
  }
  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button onClick={() => this.incrementCount()}>
          Click me
      </button>
      </div>
    );
  }
}
```

---

```
import React, { useState } from 'react';
export default function Demo1() {
```

```
  // Declare a new state variable, which is "count"
  const [count, setCount] = useState(0); // initial value for count is '0'

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Program-89  useState with array data

**********

```
import React, { useState } from 'react'

export default function Test() {
  const [employees, setEmployees] = useState([
    { id: 101, name: 'sanjay', sal: 5000 },
    { id: 102, name: 'yash', sal: 7000 },
    { id: 103, name: 'suresh', sal: 6000 },
  ]);

  const [id, setId] = useState();
  const [name, setName] = useState('');
  const [sal, setSal] = useState();

  function setDefaultValues() {
    setId('');
    setName('');
    setSal('')
  }
  const deleteEmp = (ind) => {
    employees.splice(ind, 1)
    setEmployees([...employees])
  }
  const addEmployee = (event) => {
    event.preventDefault();
    let newObj = { "id": id, "name": name, "sal": sal };
    setEmployees([...employees, newObj])
    setDefaultValues();
  }

  return (
    <>
      <table className='table table-bordered table-striped table-responsive'>
        <thead>
          <tr>
            <th>eid</th>
```

```
                <th>name</th>
                <th>sal</th>
                <th>Action</th>
              </tr>
            </thead>
            <tbody>
              {employees.map((employee, ind) => {
                return (
                  <tr>
                    <td>{employee.id}</td>
                    <td>{employee.name}</td>
                    <td>{employee.sal}</td>
                    <td>
                      <button className='btn btn-danger' onClick={() =>
deleteEmp(ind)}>DELETE</button>
                    </td>
                  </tr>
                )
              })}
            </tbody>
          </table>
          <hr />
          <form onSubmit={addEmployee}>
            id : <input
              name="id"
              type="text"
              value={id}
              onChange={e => setId(e.target.value)}
            /> <br /><br />

            Name : <input
              name="name"
              type="text"
              value={name}
              onChange={e => setName(e.target.value)}
            /> <br /><br />


            Name : <input
              name="sal"
              type="text"
              value={sal}
              onChange={e => setSal(e.target.value)}
            /> <br /><br />

            <input type='submit' />
          </form>

        </>
      );
}
```

program-90 useEffect

**********

```
import React, { useState, useEffect } from 'react'
```

```
export default function Test() {
   const [users, setUsers] = useState([]);

   useEffect(() => {
      fetch("https://jsonplaceholder.typicode.com/users")
         .then(response => response.json())
         .then(result => setUsers(result));
   }, []);

   return (
      <div>
         <table className='table table-bordered table-striped table-responsive'>
            <thead>
               <tr>
                  <th>id</th>
                  <th>name</th>
                  <th>email</th>
                  <th>phone</th>
               </tr>
            </thead>
            <tbody>
               {users.map((user, ind) => {
                  return (
                     <tr>
                        <td>{user.id}</td>
                        <td>{user.name}</td>
                        <td>{user.email}</td>
                        <td>{user.phone}</td>
                     </tr>
                  )
               })}
            </tbody>
         </table>
      </div>
   );
}
```

program-91 custom Hook

**********

fetchHook.js

---

```
import React, { useEffect, useState } from "react";

export default function useFetch(url) {
  const [data, setData] = useState([]);

  useEffect(async () => {
    const response = await fetch(url);
    const data = await response.json();
    setData(data);
  }, []);

  return data;
}
```

test.js
---

```
import React, { useState, useEffect } from 'react'
import useFetch from '../customhook/fetchHook';

export default function Test() {
  let users = useFetch('https://jsonplaceholder.typicode.com/users')
  return (
    <div>
      <table className='table table-bordered table-striped table-responsive'>
        <thead>
          <tr>
            <th>id</th>
            <th>name</th>
            <th>email</th>
            <th>phone</th>
          </tr>
        </thead>
        <tbody>
          {users.map((user, ind) => {
            return (
              <tr>
                <td>{user.id}</td>
                <td>{user.name}</td>
                <td>{user.email}</td>
                <td>{user.phone}</td>
              </tr>
            )
          })}
        </tbody>
      </table>
    </div>
  );
}
```

program-92: Context Example
**********
UserContext.js
---

```
import { createContext, useState } from "react";
export const UserContext = createContext();
export const UserProvider = ({ children }) => {
  const [user, setUser] = useState({ name: 'Guest' });
  const setLoggedinUser = (name) => {
    setUser({ name })
  }
  return <UserContext.Provider value={{ user, setLoggedinUser }}>
    {children}
  </UserContext.Provider>
}
```

ThemeContext.js
---

```
import { createContext, useState } from "react";
```

```
export const ThemeContext = createContext();
export const ThemeProvider = ({ children }) => {
   const [theme, setTheme] = useState('light');
   const toggleTheme = () => {
      setTheme((prev) => {
         return prev === 'light' ? 'dark' : 'light';
      })
   }
   return <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
   </ThemeContext.Provider>
}
```

-Wrap the component with Context Providers

```
            <ThemeProvider>
      <UserProvider>
        <App />
      </UserProvider>
     </ThemeProvider>
```

-Consume the data from Context (DashBoard.js)

```
import React, { useContext } from "react";
import { ThemeContext } from "./ThemeContext";
import { UserContext } from "./UserContext";
const Dashboard = () => {
 const { theme, toggleTheme } = useContext(ThemeContext);
 const { user, setLoggedinUser } = useContext(UserContext);
 return <div style={{ padding: 20, background: theme === "light" ? "#fff" : "#333", color: theme ===
"light" ? "#000" : "#fff" }}>
         <h2>Welcome, {user.name}</h2>
         <h2>Current Theme : {theme}</h2>
         <button onClick={() => setLoggedinUser("John Doe")}>Login as John Doe</button>
         <br /><br />
         <button onClick={toggleTheme}>Toggle Theme</button>
      </div>
};
export default Dashboard;
```

Program- 94 Redux Counter

**********

1. create a new project

```
  npx create-react-app counter-app
```

2. install redux & react-redux

```
  npm install redux react-redux @reduxjs/toolkit
```

3. Create Folders & Files for actions, reducers, store and components

```
   store-->store.js
            actions-->actions.js
            reducers-->reducers.js
            components-->counter.js
```

4. Create Actions (actions.js)

```
export const increment = () => { // Action Creator function
        return { type: 'INCREMENT' } // action object
}
export const decrement = () => {
        return { type: 'DECREMENT' }
}
export const reset = () => {
        return { type: 'RESET' }
}
```

5. Add Reducer Code

```
const counterReducer = (state = 1, action) => {
 switch (action.type) {
        case "INCREMENT":
         return state + 1;
        case "DECREMENT":
         return state - 1;
        case "RESET":
         return (state = 0);
        default:
         return state;
 }
};
export default counterReducer;
```

6. create Store (store.js)

```
import { configureStore } from '@reduxjs/toolkit';
import counterReducer from '../reducers/counter_reducer';
const myStore = configureStore({
        reducer: {
                counter: counterReducer
        }
});
export default myStore;
```

7. Provide Store (index.js)

```
import { Provider } from 'react-redux';
import mystore from './store/store'

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
 <React.StrictMode>
        <Provider store={mystore}>
         <App />
        </Provider>
 </React.StrictMode>
);
```

8. Import and Dispatch the Actions(Counter.js)

```
import { useDispatch, useSelector } from 'react-redux';
import { decrement, increment, reset } from '../actions/actions'
export default function Counter() {
        const count = useSelector((state) => state.counter)
        const dispatch = useDispatch();
        return <>
```

```
                        <h3 className='text-center'>Counter Example Using REDUX</h3>
                        <div className='col-md-4 offset-md-4'>
                                <div className='m-3 p-3 border border-3 rounded-3 text-center'>
                                        <h4>Count is: {count}</h4>

                                        <button class="btn btn-secondary" onClick={() =>
dispatch(decrement())}>Decrement</button>
                                        <button class="btn btn-danger mx-2" onClick={() =>
dispatch(reset())}>Reset</button >
                                        <button class="btn btn-primary" onClick={() =>
dispatch(increment())}> Increment</button >
                                </div>
                        </div >
                </>
        }
```

Program-95: Redux ToDo example

**********

1. create a new project

```
  npx create-react-app <projectName>
```

2. install react-redux

```
  npm install @reduxjs/toolkit react-redux
```

3. Create Folders & Files for actions, reducers, store and components

```
    store-->store.js
            features/todo --> todoSlice.js
            components--> ToDoList.js , ToDo.js, AddToDo.js
```

4. Create a Redux Store

```
   import { configureStore } from '@reduxjs/toolkit'
        export const store = configureStore({
                reducer: {todoReducer:todoReducer}
        })
```

5. Provide the Redux Store to React(index.js)

```
   <Provider store={myStore}>
     <App />
   </Provider>
```

6. Create a Redux "slice" reducer with createSlice , Export the generated slice reducer and action creators

```
        import { createSlice } from '@reduxjs/toolkit';
        const initialState = {
                todoList: [
                        { id: 1, text: "Learn React", isCompleted: false },
                        { id: 2, text: "Complete Java Assignments", isCompleted: true },
                ]
        }
        // slice = reducers + action_creators
        export const todoSlice = createSlice({
                name: 'todo',
                initialState,
                reducers: {
                        addToDo: {
```

```
                reducer: (state, action) => {
                        state.todoList.push(action.payload);
                },
                prepare: (text) => ({
                        payload: {
                                id: Math.random() * 20,
                                text,
                                completed: false,
                        }
                })
        },
        deleteToDo: (state, action) => {
                state.todoList = state.todoList.filter(todo => todo.id !==
action.payload);
                return state;
        },
        toggleToDo: (state, action) => {
                state.todoList = state.todoList.map((toDo) =>
                        toDo.id === action.payload ? { ...toDo,
isCompleted: !toDo.isCompleted } : toDo
                );
                return state;
        }
    }
})
// Action creators are generated for each case reducer function
export const { addToDo, deleteToDo, toggleToDo } = todoSlice.actions; // Action Creators
export default todoSlice.reducer;
```

7. Read data from the store with the useSelector hook (ToDoList.js)

```
import { useSelector } from 'react-redux'
import ToDo from './ToDo';
import AddToDo from './AddToDo';
export default function ToDoList() {
        const todoArr = useSelector((state) => state.todoReducer.todoList);
        const CompletedToDos = todoArr.filter(todo => todo.isCompleted);
        return <>
                <div className='col-sm-4 offset-4'>
                        <div className='p-3 m-3 border border-3 rounded-3'>
                                <h2 className='text-center'>ToDo List Using
REDUX</h2>
                                <hr />
                                <AddToDo />
                                <hr/>
                                {todoArr.map((todoObj, ind) => {
                                        return <ToDo todoObj={todoObj} ind={ind} />
                                })}
                                <hr />
                                {CompletedToDos.length}/{todoArr.length} are
completed
                        </div>
                </div>
        </>
}
```

9. ToDo.js

```
import { useDispatch } from 'react-redux'
import { deleteToDo, toggleToDo } from '../features/todo/todoSlice';
export default function ToDo({ todoObj }) {
	const dispatch = useDispatch();
	const deleteMyTodo = (id) => {
		const actionObj = deleteToDo(id);
		dispatch(actionObj)
	}
	const toggleMyTodo = (id) => {
		const actionObj = toggleToDo(id);
		dispatch(actionObj)
	}
	return <>
		<li>
			<span style={{ textDecoration: todoObj.isCompleted ? 'line-
through' : 'none' }}>
				{todoObj.text}
			</span>  
			<button onClick={() =>
{ deleteMyTodo(todoObj.id) }}>DELETE</button>  
			<button onClick={() =>
{ toggleMyTodo(todoObj.id) }}>TOGGLE</button>  
		</li>
	</>
}
```

10. AddToDo.js

```
import React, { useRef } from 'react'
import { useDispatch } from 'react-redux';
import { addToDo } from '../features/todo/todoSlice';
export default function AddToDo() {
	const dispatch = useDispatch();
	const myRef1 = useRef();
	const addNewToDo = () => {
		const actionObj = addToDo(myRef1.current.value);
		dispatch(actionObj);
		myRef1.current.value = '';
	};
	return <>
		<input ref={myRef1} />  
		<button onClick={addNewToDo}>Add New ToDo</button>
	</>
}
```

Program:-96 useReducer hook

**********

```
import React, { useReducer, useRef } from "react";

const initialTodos = [
  { id: 1, title: "Todo 1", complete: false },
  { id: 2, title: "Todo 2", complete: true },
];
```

```
const reducer = (state, action) => {
  switch (action.type) {
    case "ADD":
      return [...state, action.payload];
    case "DELETE":
      return state.filter((todo) => todo.id !== action.id);
    case "TOGGLE":
      return state.map((todo) => {
        if (todo.id === action.id) {
          return { ...todo, complete: !todo.complete };
        } else {
          return todo;
        }
      });
    default:
      return state;
  }
};

let nextTodoId = 3;
export default function UseReducerDemo() {
  const [todos, dispatch] = useReducer(reducer, initialTodos);
  const myref1 = useRef();

  const addNewTodo = (title) => {
    const actionObj = {
      type: "ADD",
      payload: {
        id: nextTodoId++,
        title: title,
        complete: false,
      },
    };
    dispatch(actionObj);
  };
  const handleComplete = (todo) => {
    dispatch({ type: "TOGGLE", id: todo.id });
  };
  const deleteTodo = (id) => {
    dispatch({ type: "DELETE", id: id });
  };
  return (
    <>
      <h3>Add ToDo</h3>
      <input ref={myref1} />
      <button onClick={() => addNewTodo(myref1.current.value)}>Add todo</button>
      <hr />
      {todos.map((todo) => (
        <div key={todo.id}>
          <label>
            <input
              type="checkbox"
              checked={todo.complete}
              onChange={() => handleComplete(todo)}
            />
            <span
              style={{
```

```
            textDecoration: todo.complete ? "line-through" : "none",
           }}
          >
           {todo.title}
           <button onClick={() => deleteTodo(todo.id)}>DELETE</button>
          </span>
         </label>
        </div>
     ))}
   </>
 );
}
```

Program:-97 useCallback Hook

**********

```
import { useState, useCallback } from "react";
import ReactDOM from "react-dom";
import Todos from "./Todos";

const App = () => {
 const [count, setCount] = useState(0);
 const [todos, setTodos] = useState([]);

 const increment = () => {
  setCount((c) => c + 1);
 };
 const addTodo = useCallback(() => {
  setTodos((t) => [...t, "New Todo"]);
 }, [todos]);

 return (
  <>
   <Todos todos={todos} addTodo={addTodo} />
   <hr />
   <div>
    Count: {count}
    <button onClick={increment}>+</button>
   </div>
  </>
 );
};
-----------------------------------------------
import { memo } from "react";
const Todos = ({ todos, addTodo }) => {
 console.log("child render");
 return (
  <>
   <h2>My Todos</h2>
   {todos.map((todo, index) => {
    return <p key={index}>{todo}</p>;
   })}
   <button onClick={addTodo}>Add Todo</button>
  </>
 );
```

```
};
export default memo(Todos);
```

Program :-98 usememo

**********

```
import React, { useMemo, useState } from "react";

export default function Demo() {
  const [count, setCount] = useState(0);
  const [todos, setTodos] = useState(['To Do-1', 'To Do-2']);
  const calculation = useMemo(() => expensiveCalculation(count), [count]);
  // const calculation = expensiveCalculation(count);

  const increment = () => {
    setCount((c) => c + 1);
  };
  const addTodo = () => {
    setTodos([...todos, "New Todo"]);
  };
  return (
    <div>
      <div>
        <h2>My Todos</h2>
        {todos.map((todo, index) => {
          return <p key={index}>{todo}</p>;
        })}
        <button onClick={addTodo}>Add Todo</button>
      </div>
      <hr />
      <div>
        Count: {count}
        <button onClick={increment}>+</button>
        <h2>Expensive Calculation</h2>
        {calculation}
      </div>
    </div>
  );
}
const expensiveCalculation = (num) => {
  console.log("Calculating...");
  for (let i = 0; i < 1000000000; i++) {
    num += 1;
  }
  return num;
};
```

Program-99 Error Boundary

**********

```
import React, { Component } from 'react';
export default class MyErrorBoundry extends Component {
    constructor(props) {
        super(props);
        this.state = { hasError: false };
```

```
  }
  static getDerivedStateFromError(error) {
    // Update state so the next render will show the fallback UI.
    return { hasError: true };
  }
  componentDidCatch(error, errorInfo) {
    // You can also log the error to an error reporting service
    // logErrorToMyService(error, errorInfo);
  }
  render() {
    if (this.state.hasError) {
      return <h2>Something went wrong, Component cann't be loaded </h2>
    } else {
      // Normally, just render children
      return this.props.children;
    }
  }
}


import React, { Component } from 'react'
export class BuggyCounter extends Component {
  constructor(props) {
    super(props);
    this.state = { counter: 0 };
  }
  increment = () => {
    this.setState({ counter: this.state.counter + 1 });
  }
  render() {
    if (this.state.counter === 5) {
      // Simulate a JS error
      throw new Error('I crashed!');
    }
    return <>
      <h1>Count is : {this.state.counter}</h1>
      <button onClick={this.increment}>Increment</button>
    </>
  }
}


function Test() {
 return (
  <div>
   <p>
    <b>
     This is an example of error boundaries in React 16.
     <br /><br />
     Click on the numbers to increase the counters.
     <br />
     The counter is programmed to throw when it reaches 5. This simulates a JavaScript error in a
component.
    </b>
   </p>
   <hr />
   <ErrorBoundary>
```

```
      <p>These two counters are inside the same error boundary. If one crashes, the error boundary will replace both of them.</p>
      <BuggyCounter />
      <BuggyCounter />
    </ErrorBoundary>
    <hr />
    <p>These two counters are each inside of their own error boundary. So if one crashes, the other is not affected.</p>
    <ErrorBoundary><BuggyCounter /></ErrorBoundary>
    <ErrorBoundary><BuggyCounter /></ErrorBoundary>
  </div>
 );
}
```

Program-100  Portals
**********

1. add target div in index.html

```
<div id="modal-root"></div> <!--Root Element for Portal-->
```

2. Create a Portal Component

```
import React from 'react';
        import ReactDOM from 'react-dom';
        const styles = {
                overlay: {
                        position: "fixed", top: 0, left: 0, right: 0, bottom: 0,
                        backgroundColor: "rgba(0, 0, 0, 0.5)",
                        display: "flex", justifyContent: "center", alignItems: "center"
                },
                modal: {
                        background: '#fff', padding: '20px', borderRadius: '5px'
                }
        }
        export default function MyModal({ children }) {
                return ReactDOM.createPortal(
                        <div style={styles.overlay}>
                                <div style={styles.modal}>
                                        {children}
                                </div>
                        </div>,
                        document.getElementById('modal-root')
                )
        }
```

3. Use the Portal Component in a parent component

```
 import { useState } from 'react';
        import MyModal from './components/MyModal';
        function App() {
          const [showModal, setShowModal] = useState(false);
          return <>
                <h1>Let's Learn Portals</h1>
                <button className='btn btn-primary' onClick={() => setShowModal(true)}>Open Modal</button>
                {showModal &&
                  <MyModal>
                        <h2>This is a Modal</h2>
```

```
                                <button className='btn btn-primary' onClick={() =>
setShowModal(false)}>Close Modal</button>
                            </MyModal>
                          }
                </>
              }
              export default App;
```

Jest Unit Testing
==========

```
import { render, screen, waitFor } from '@testing-library/react';
import UserList from '../components/users/UserList';
const MockUserList = [
  {
    id: 1, userName: 'sanjay_kumar', name: 'sanjay',
    email: 'sanjay@gmail.com', phone: 1234512345,
    address: { city: 'bangalore' }
  }
]
// Mock fetch()
beforeEach(() => {
  global.fetch = jest.fn(() => {
    return Promise.resolve({
      json: () => Promise.resolve(MockUserList)
    })
  });
})
afterEach(() => {
  jest.clearAllMocks();
});
describe('Test Suite For Http Demo 2 Component', () => {
  it('Lets Verify fetchUser functionality', async () => {
    render(<UserList />);
    // wait for user data to be displayed on the screen
    await waitFor(() => {
      expect(screen.getByText('sanjay')).toBeInTheDocument();
    });
    expect(global.fetch).toHaveBeenCalledWith('https://jsonplaceholder.typicode.com/users');
  })
})
```

Unit Testing Axios Code
==========

```
// UserService.js
import axios from "axios";
const API = "/api/users";
export const getUsers = () => axios.get(API);
export const createUser = (user) => axios.post(API, user);
export const updateUser = (id, user) => axios.put(`${API}/${id}`, user);
export const deleteUser = (id) => axios.delete(`${API}/${id}`);
```

---

```
import axios from "axios";
```

```
import * as UserService from "./UserService";

// Automatically mock axios
jest.mock("axios");

describe("UserService API calls", () => {
 afterEach(() => {
  jest.clearAllMocks();
 });

 test("getUsers should fetch users", async () => {
  const mockData = [{ id: 1, name: "Alice" }];
  axios.get.mockResolvedValueOnce({ data: mockData });

  const response = await UserService.getUsers();
  expect(response.data).toEqual(mockData);
  expect(axios.get).toHaveBeenCalledWith("/api/users");
 });

 test("createUser should post user data", async () => {
  const newUser = { name: "Bob" };
  axios.post.mockResolvedValueOnce({ data: { id: 2, ...newUser } });

  const response = await UserService.createUser(newUser);
  expect(response.data).toEqual({ id: 2, name: "Bob" });
  expect(axios.post).toHaveBeenCalledWith("/api/users", newUser);
 });

 test("updateUser should send updated data", async () => {
  const updatedUser = { name: "Updated Alice" };
  axios.put.mockResolvedValueOnce({ data: updatedUser });

  const response = await UserService.updateUser(1, updatedUser);
  expect(response.data).toEqual(updatedUser);
  expect(axios.put).toHaveBeenCalledWith("/api/users/1", updatedUser);
 });

 test("deleteUser should send delete request", async () => {
  axios.delete.mockResolvedValueOnce({ status: 204 });

  const response = await UserService.deleteUser(1);
  expect(response.status).toBe(204);
  expect(axios.delete).toHaveBeenCalledWith("/api/users/1");
 });

 test("getUsers should handle errors", async () => {
        axios.get.mockRejectedValueOnce(new Error("Network error"));
        await expect(UserService.getUsers()).rejects.toThrow("Network error");
 });

});
```