React js

React js Boiler plate:-
![[Pasted image 20250326212746.png]]

LAB-1

1)
Create a new React project using:
npx create-react-app my-app •
Run the project using: npm start

3)TIme and date
// App.js

```
import React from 'react';
function App() {
const name = "Alice";
const hour = new Date().getHours();
const greeting = hour < 12 ? "Good Morning" : hour < 18 ? "Good Afternoon" : "Good Evening";
return <h1>{greeting}, {name}!</h1>;

}
export default App;
```

4)for both in one

```
import React from "react";
function Header() {
const name = "Ashish";
return <h1>Hello {name}!</h1>

}
function Footer() {
return <p>@Made by me</p>

}

const App = () => (

  <>
```

```
    <Header />

    <Footer />

  </>

);
export default App;
```

5)LIST

```
import React from "react";
function App() {
const fruits = ["apple", "banana", "papaya"];
return (
<ul>

      {fruits.map((fruit, index) => (

        <li key={index}>{fruit}</li>

      ))}

    </ul>

  );

}
export default App;
```

6)Conditional Rendering in JSX

```
import React from "react";

function App(){

  const isL=false;

  return <h1>{isL? "yes it is logged in":"please log in"}</h1>;

}

export default App;
```

7)Using Inline and Internal CSS in JSX

```
import React from "react";
function App(){
const mStyle = {
color: "blue",
fontSize: "20px"
};
return <h1 style={mStyle}>Heyyy</h1>
}

export default App;
```

8)Creating a Button with an onClick Event

```
import React from "react";

function ClickMe(){

  function handleclick(){

    alert("Button CLicked");

  }

  return <button onClick={handleclick}>Click me</button>

}

export default ClickMe;
```

9)Displaying the Current Date and Time

```
// App.js

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

function CurrentTime() {

 const [time, setTime] = useState('');

 useEffect(() => {

  // Time ko set karna har second

  const timer = setInterval(() => {

   setTime(new Date().toLocaleTimeString());

  }, 1000);

  // Component cleanup, interval band karne ke liye

  return () => clearInterval(timer);

 }, []); // Empty array means effect runs only once
 return (

  <div>

   <h2>Time: {time}</h2>

  </div>

 );

}

export default CurrentTime;
```

10)image

```
import React from "react";

function App() {

 return <img src="google.com" alt="jhhjcfbsjb" />;

}

export default App;
```

10)Rendering a List of Users from an Array

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

function App(){

  const arr=["hi","hello","heiii"];

  return(

    <ul>

      {arr.map((user,index)=>(

        <li key={index}>{user}</li>

      ))}

    </ul>

  );

}
```

```
export default App;
```

11)Rendering a List of Users from an Array

```
import React from "react";

function App() {

 const users = ["Alice", "Bob", "can"];

 return (

  <ul>

   {users.map((user, index) => (

    <li key={index}>{user}</li>

   ))}

  </ul>

 );

}

export default App;
```

12)Creating a Greeting Component with Props

```
import React from "react";

function Greetings(props){

 return <h1>Hello,{props.name}!</h1>;

}

function App() {

 return <Greetings name="Ashish" />

}

export default App;
```

13)Using React Fragments to Return Multiple Elements

```
import React from "react";

function App(){
 return (

  <>

  <h1>hii</h1>

  <p>this is me</p>

  </>

 );

}

export default App;
```

14)SImple counter App

```
import React from "react";

 import { useState } from "react";

 function App(){

  const [count,setcount]=useState(0);

  return (

   <div>

   <h1>count {count}</h1>

   <button onClick={()=>setcount(count+1)}>Increment</button>

   <button onClick={()=>setcount(count-1)}>decrement</button>

   </div>

  );

 }

 export default App;
```

15)adult or not

```
import React from "react";

 import { useState } from "react";
 function App(){
  const [age,setage]=useState("");
  return (
   <div>
   <input type="number" onChange={(e)=>setage(e.target.value)}/>
   <p>{age>18 ? "you are an adult":"you are not adult"}</p>
   </div>
  );
 }
 export default App;
```

LAB-2

1)Creating and Using Class Components with Constructors

```
import React,{Component} from "react";
class Welcome extends Component{
 constructor(props){
  super(props);
  this.state={name:"alice"};
 }
 render(){
  return <h1>Welcome,{this.state.name}</h1>;
 }
}
export default Welcome;
```

2)Implementing Component Life Cycle Methods

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

```
class DataFetcher extends Component{
constructor(){
super();
this.state={data:'loading....'};
}
componentDidMount(){
setTimeout(() => {this.setState({data:'Api data loaded'});
}, 2000);
}
componentDidUpdate(){
console.log("component updated");
}
componentWillUnmount(){
console.log("component unmounted");
}
render(){
return <h1>Welcome,{this.state.data}</h1>;
}
}
export default DataFetcher;
```

3)Using React Component API: forceUpdate and shouldComponentUpdate

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

class ForceUpdateExample extends Component {
shouldComponentUpdate() {
return true;
 }
 render() {
  return (
   <div>
    <h1>Current Time: {new Date().toLocaleTimeString()}</h1>
    <button onClick={() => this.forceUpdate()}>Update Time</button>
   </div>
  );
 }
}

export default ForceUpdateExample;
```

6)Creating a Parent-Child Component Structure

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

function Child(props){

 return <h1>child Received: {props.message}</h1>

}

function Parent(){

 return <Child message="Hello from parents"/>;

}

export default Parent;
```

7)Managing State and Lifecycle with Hooks (useEffect)

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

function Timer() {

 const [time, setTime] = useState(new Date().toLocaleTimeString());

 useEffect(() => {

  const interval = setInterval(() => {

   setTime(new Date().toLocaleTimeString());

  }, 1000);

  return () => clearInterval(interval);

 }, []);

 return <h1>Current Time: {time}</h1>;

}

export default Timer;
```

8)Implementing Component Composition with Multiple Components

```
import React from "react";
function Header(){
```

```
return <h1>THis is header</h1>;
}
function Content(){

 return <p>This is the content portion</p>
}
function Footer(){

 return <p>@This is footer</p>

}

function App(){

 return (

  <div>

   <Header/>

   <Content/>

   <Footer/>

  </div>

 )

}

export default App;

9)**Simulating an API Call and Displaying Data**

import { useState, useEffect } from "react";

function UserList() {
const [users, setUsers] = useState([]);

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

```
return (
<ul>
{users.map((user) => (
<li key={user.id}>{user.name}</li>
))}
</ul>
);
}

export default UserList;
```

10)Creating a Component with Controlled Inputs

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

function NameForm() {

 const [name, setName] = useState("");

 return (

  <div>

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

   <p>Hello, {name}!</p>

  </div>

 );

}

export default NameForm;
```

LAB-3

1)Create a React Component using JSX that Displays a Greeting Message

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

function Greeting({name}){

 return <h1>Hello {name}</h1>

}

function App() {
 return <div>
 <Greeting name="ash"/>
 <Greeting name="rik"/>
 </div>
}
export default App;
```

2)Build a Counter Component using a Constructor in a Class Component

```
import React,{ useState } from "react";
class Counter extends React.Component {
 constructor(props){
  super(props);
  this.state={count :0};
 }
 increment=()=>{
  this.setState({count:this.state.count+1});
 };
 decrement=()=>{
  this.setState({count: this.state.count-1});
 };
 render(){
  return(
   <div>
    <h1>
     count:{this.state.count}
    </h1>
    <button onClick={this.increment}>+</button>
    <button onClick={this.decrement}>-</button>
   </div>
  )
 }
}
export default Counter;
```

3)Convert a Class Component into a Functional Component using Hooks

```
import React,{ useState } from "react";
function App(){
 const [count,setCount]=useState(0);
 return (
  <div>
   <h1>count:{count}</h1>
   <button onClick={()=>setCount(count+1)}>+</button>
   <button onClick={()=>setCount(count-1)}>-</button>
  </div>  )
}
export default App;
```

4)Create a Dynamic List Rendering Component using .map()

```
function UserList({ users }) {
return (
<ul>
   {users.map((user, index) => (
    <li key={index}>{user}</li>
   ))}
  </ul>
```

```
 );

}
export default function App() {
const users = ["John", "Jane", "Alice", "Bob"];
return <UserList users={users} />;
}
```

5)Implement a Simple Theme Switcher using useState

```
import React, { useState } from 'react';
function Themeswitcher() {
const [theme, setTheme] = useState('light');
const toggleTheme = () => {
setTheme(theme === 'light' ? 'dark' : 'light');
};
return (
<div style={{ background: theme === 'light' ? '#fff' : '#333', color: theme ===
'light' ? '#000' : '#fff' }}>
<h1>{theme} Mode</h1>
<button onClick={toggleTheme}>Toggle Theme</button>
</div>
);

}

export default Themeswitcher;
```

6)Use useEffect to Fetch and Display Data from an API

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

function Users(){

 const [users,setUsers]=useState([]);

 useEffect(()=>

 {

  fetch('https://jsonplaceholder.typicode.com/users')

  .then((Response)=>Response.json())

  .then((data)=>setUsers(data));

 },[])
 return (
  <div>
   {users.map((user) => (
    <div key={user.id}>
     <h2>{user.name}</h2>
     <p>Email: {user.email}</p>
     <p>Website: {user.website}</p>
    </div>
   ))}
  </div>
 )
}

export default Users;
```

7)Implement a Simple Form Handling Component using useState

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

function SignupForm() {
const [name, setName] = useState('');
const [email, setEmail] = useState('');

const handleSubmit = (e) => {
e.preventDefault();
alert(Name: ${name}, Email: ${email});
};

return (
<form onSubmit={handleSubmit}>
<input type="text" value={name} onChange={(e) => setName(e.target.value)}
placeholder="Name" />
<input type="email" value={email} onChange={(e) => setEmail(e.target.value)}
placeholder="Email" />
<button type="submit">Submit</button>
</form>
);
}

export default SignupForm;
```

8)Create a Component that Uses useEffect to Track Window Resize

```
import React, { useState, useEffect } from 'react';
function WindowResize() {
const [width, setWidth] = useState(window.innerWidth);
useEffect(() => {
const handleResize = () => setWidth(window.innerWidth);
window.addEventListener('resize', handleResize);
return () => {
window.removeEventListener('resize', handleResize);
};
}, []);
return <h1>Window width: {width}px</h1>;
}

export default WindowResize;
```

9)Create a Parent-Child Component Communication using Props and Callbacks

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

function Parent() {

 const [message, setMessage] = useState("Hello");
return (
<div>
<h1>{message}</h1>
<Child updateMessage={() => setMessage("Updated!")}/>
</div>
);
}
function Child({ updateMessage }) {
return <button onClick={updateMessage}>Change Parent Message</button>;
}
export default Parent;
```

10)Implement a StopWatch Using useState and useEffect

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

function StopWatch() {
// State variables
```

```
const [time, setTime] = useState(0); // Total time in seconds
const [isRunning, setIsRunning] = useState(false); // To check if stopwatch is
running

// useEffect to handle the timer
useEffect(() => {
let interval = null;
if (isRunning) {
// Start the interval when the stopwatch is running
interval = setInterval(() => {
setTime((prevTime) => prevTime + 1); // Increment time every second
}, 1000);
} else if (!isRunning) {
// Clear the interval when the stopwatch is stopped
clearInterval(interval);
}
return () => clearInterval(interval); // Clean up the interval on unmount
}, [isRunning]); // Dependency: isRunning

// Helper function to format time (minutes : seconds)
const formatTime = () => {
const minutes = Math.floor(time / 60);
const seconds = time % 60;
return String(minutes).padStart(2,″0″) :{String(seconds).padStart(2, "0")} ;
};

// Handlers for Start, Stop, and Reset buttons
const handleStart = () => {
setIsRunning(true); // Set stopwatch running
};

const handleStop = () => {
setIsRunning(false); // Stop the stopwatch
};

const handleReset = () => {
setIsRunning(false); // Stop and reset the time
setTime(0);
};

return (
<div>
<h1>StopWatch</h1>
<h2>{formatTime()}</h2>
<div>
<button onClick={handleStart}>Start</button>
<button onClick={handleStop}>Stop</button>
<button onClick={handleReset}>Reset</button>
</div>
</div>
);
}
```

```
export default StopWatch;
```

```
===============================================================
```

```
==LAB-5==
1)
import React, { useState, useEffect } from 'react';
import PropTypes from 'prop-types';
import './ContactList.css'; // External CSS for styling
// Contact Item Component
function ContactItem({ contact, onDelete }) {
return (
<div className="contact-item">
<p>{contact.name}</p>
<p>{contact.phone}</p>
<button onClick={() => onDelete(contact.id)}>Delete</button>
</div>
);
}
// Prop validation
ContactItem.propTypes = {
contact: PropTypes.shape({
id: PropTypes.number,
name: PropTypes.string,
phone: PropTypes.string,
}).isRequired,
onDelete: PropTypes.func.isRequired,
};
// Contact List Component
function ContactList() {
```

```
const [contacts, setContacts] = useState([]);
const [name, setName] = useState('');
const [phone, setPhone] = useState('');
const [error, setError] = useState('');
// Load contacts from localStorage on mount
useEffect(() => {
const savedContacts = JSON.parse(localStorage.getItem('contacts')) || [];
setContacts(savedContacts);
}, []);
// Save contacts to localStorage whenever contacts state changes
useEffect(() => {
localStorage.setItem('contacts', JSON.stringify(contacts));
}, [contacts]);
// Add contact
const addContact = () => {
if (!name || !phone) {
setError('Both name and phone number are required.');
return;
}
const newContact = { id: Date.now(), name, phone };
setContacts([...contacts, newContact]);
setName('');
setPhone('');
setError(''); // Clear error message
};
const deleteContact = (id) => {
setContacts(contacts.filter(contact => contact.id !== id));
};
return (
<div>
<h2>Contact List</h2>
{error && <p className="error-message">{error}</p>}
<input
type="text"
placeholder="Name"
value={name}
onChange={(e) => setName(e.target.value)}
/>
<input
type="text"
placeholder="Phone"
value={phone}
onChange={(e) => setPhone(e.target.value)}
/>
<button onClick={addContact}>Add Contact</button>
<div>
{contacts.map((contact) => (
<ContactItem key={contact.id} contact={contact} onDelete={deleteContact} />
))}
</div>
</div>
);
}
export default ContactList;
```

```
------------
ConstactList.css
/* Contact List Styles */

body {

  font-family: Arial, sans-serif;

  background-color: #f4f4f9;

  margin: 0;

  padding: 20px;

}

h2 {

  text-align: center;

  color: #333;

}

.contact-item {

  background-color: #fff;

  border: 1px solid #ddd;
```

```css
  border-radius: 5px;

  padding: 10px;

  margin: 10px 0;

  display: flex;

  justify-content: space-between;

  align-items: center;

  box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

  transition: background-color 0.3s;

}

.contact-item:hover {

  background-color: #f0f8ff;

}

.contact-item p {

  margin: 0;

  color: #555;

}

button {

  background-color: #ff4d4d;

  color: white;

  border: none;

  border-radius: 5px;

  padding: 8px 12px;

  cursor: pointer;

  transition: background-color 0.3s;
```

```
}
button:hover {
  background-color: #ff1a1a;
}
input[type="text"] {
  padding: 10px;
  margin: 5px 0;
  border: 1px solid #ddd;
  border-radius: 5px;
  width: calc(100% - 22px); /* Full width minus padding */
}
.error-message {
  color: #ff4d4d;
  text-align: center;
  margin: 10px 0;
}
```

2)Product Catalog with Inline Styling:

```
import React from 'react';
import PropTypes from 'prop-types';

// Product Item Component with inline styling
function ProductItem({ product }) {
const cardStyle = {
backgroundColor: product.available ? 'lightgreen' : 'lightcoral',
padding: '10px',
margin: '10px',
border: '1px solid black',
borderRadius: '5px',
};
return (
<div style={cardStyle}>
<h3>{product.name}</h3>
<p>Price: ${product.price}</p>
<p>{product.available ? 'In Stock' : 'Out of Stock'}</p>
</div>
);
}

// Prop validation
ProductItem.propTypes = {
product: PropTypes.shape({
name: PropTypes.string,
price: PropTypes.number,
available: PropTypes.bool,
}).isRequired,
};

// Product Catalog Component
function ProductCatalog() {
const products = [
{ name: 'Laptop', price: 999, available: true },
{ name: 'Phone', price: 599, available: false },
{ name: 'Headphones', price: 199, available: true },
];

return (
<div>
<h2>Product Catalog</h2>
{products.map((product, index) => (
<ProductItem key={index} product={product} />
))}
</div>
);
}

export default ProductCatalog;
```

`3)Theme Switcher using CSS Modules:`

```
import React, { useState } from 'react';
import styles from './ThemeSwitcher.module.css'; // CSS Module
function ThemeSwitcher() {
```

```
const [isDarkMode, setIsDarkMode] = useState(false);
const toggleTheme = () => {
setIsDarkMode(!isDarkMode);
};
return (
<div className={isDarkMode ? styles.dark : styles.light}>
<h1>Theme Switcher</h1>
<button onClick={toggleTheme}>
Switch to {isDarkMode ? 'Light' : 'Dark'} Mode
</button>
</div>
);
}
export default ThemeSwitcher;

--------------
ThemeSwitcher.module.css========

.light {

  background-color: white;

  color: black;

 }

 .dark {

  background-color: black;

  color: white;

 }
```

4)Styled Profile Card:

```
App.js-------
import ProfileCard from "./ProfileCard";

function App() {

 return (

  <div>

   <ProfileCard name="John Doe" age={28} location="New York" />
   <ProfileCard name="Jane Smith" age={35} location="California" />
  </div>
 );
}
export default App;
```

ProfileCard.js------------

```
import React from 'react';
import PropTypes from 'prop-types';
import styles from './ProfileCard.module.css'; // CSS Module import karenge
function ProfileCard({ name, age, location }) {
 // Inline style for dynamic elements
 const inlineStyle = {
  border: '2px solid #333',
  padding: '10px',
  margin: '10px',
  backgroundColor: age > 30 ? '#f0f0f0' : '#dff0d8',  // Conditional color change
 };
 return (
  <div style={inlineStyle} className={styles.profileCard}>
   <h2>{name}</h2>
   <p>Age: {age}</p>
   <p>Location: {location}</p>
  </div>
 );
}
```

```
// Prop-types validation for props

ProfileCard.propTypes = {

 name: PropTypes.string.isRequired,

 age: PropTypes.number.isRequired, // Age ko number hone ka validation

 location: PropTypes.string.isRequired,

};

export default ProfileCard;

ProfileCard.module.css---------------
.profileCard {

 border-radius: 10px;

 box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);

 max-width: 250px;

 text-align: center;

 font-family: 'Arial, sans-serif';

}

.profileCard h2 {

 color: #333;

 font-size: 24px;

}
.profileCard p {
font-size: 16px;
color: #555;
}
5)Task Tracker with State and Props:
App.js----
import React, { useState } from "react";
import PropTypes from "prop-types";
import "./TaskTracker.css";
const Task = ({ task, toggleComplete, deleteTask }) => {
```

```
return (
<div className="task">
<span
style={{
textDecoration: task.completed ? "line-through" : "none",
}}
>
{task.text}

   </span>

   <button onClick={() => toggleComplete(task.id)}>

    {task.completed ? "Undo" : "Complete"}

   </button>

   <button onClick={() => deleteTask(task.id)}>Delete</button>

  </div>

 );

};

Task.propTypes = {

 task: PropTypes.shape({

  id: PropTypes.number.isRequired,

  text: PropTypes.string.isRequired,

  completed: PropTypes.bool.isRequired,

 }),

 toggleComplete: PropTypes.func.isRequired,

 deleteTask: PropTypes.func.isRequired,

};

const TaskTracker = () => {

 const [tasks, setTasks] = useState([]);
 const [taskText, setTaskText] = useState("");

 const addTask = () => {

  if (taskText.trim() === "") return;

  setTasks([...tasks, { id: Date.now(), text: taskText, completed: false }]);

  setTaskText("");

 };

 const toggleComplete = (id) => {

  setTasks(

   tasks.map((task) =>

    task.id === id ? { ...task, completed: !task.completed } : task

   )

  );

 };

 const deleteTask = (id) => {

  setTasks(tasks.filter((task) => task.id !== id));

 };

 const completedTasks = tasks.filter((task) => task.completed).length;

 const pendingTasks = tasks.length - completedTasks;

 return (

  <div className="task-tracker">

   <h1>Task Tracker</h1>

   <input

    type="text"

    value={taskText}
```

```
    onChange={(e) => setTaskText(e.target.value)}

    placeholder="Add a task..."

   />

   <button onClick={addTask}>Add Task</button>

   <div className="tasks-list">

    {tasks.map((task) => (

     <Task

      key={task.id}

      task={task}

      toggleComplete={toggleComplete}

      deleteTask={deleteTask}

     />

    ))}

   </div>

   <div className="task-count">

    <p>Pending Tasks: {pendingTasks}</p>

    <p>Completed Tasks: {completedTasks}</p>

   </div>

  </div>

 );

};

export default TaskTracker;

TaskTracker.css-----------------
.task {

  display: flex;
  justify-content: space-between;
  margin: 10px 0;
 }
 .task-count {
  margin-top: 20px;
 }
 input {
  margin-right: 10px;
 }
 button {
  margin-left: 5px;
 }
```

6)stylish calculator

```
import React, { useState } from "react";
import PropTypes from "prop-types";
const CalculatorButton = ({ value, onClick }) => {
 return (
  <button style={{ padding: "10px", margin: "5px" }} onClick={() => onClick(value)}>
   {value}
  </button>
 );
};
CalculatorButton.propTypes = {
 value: PropTypes.oneOfType([PropTypes.string, PropTypes.number]).isRequired,
 onClick: PropTypes.func.isRequired,
};
const StylishCalculator = () => {
 const [input, setInput] = useState("");
 const handleButtonClick = (value) => {
  if (value === "=") {
   try {
    setInput(eval(input).toString()); // Evaluate the expression
   } catch {
    setInput("Error");
   }
  } else if (value === "C") {
   setInput(""); // Clear input
  } else {
   setInput(input + value); // Add to input
  }
 };
 return (
  <div>
   <h1>Stylish Calculator</h1>
   <div style={{ border: "1px solid black", padding: "10px", width: "200px" }}>
    <input type="text" value={input} readOnly style={{ width: "100%", marginBottom: "10px" }} />
    <div>
     {[7, 8, 9, "/"].map((item) => (
      <CalculatorButton key={item} value={item} onClick={handleButtonClick} />
     ))}
    </div>
    <div>
     {[4, 5, 6, "*"].map((item) => (
      <CalculatorButton key={item} value={item} onClick={handleButtonClick} />
     ))}
    </div>
    <div>
     {[1, 2, 3, "-"].map((item) => (
      <CalculatorButton key={item} value={item} onClick={handleButtonClick} />
     ))}
    </div>
    <div>
     {[0, ".", "=", "+"].map((item) => (
      <CalculatorButton key={item} value={item} onClick={handleButtonClick} />
     ))}
    </div>
    <button onClick={() => handleButtonClick("C")}>Clear</button>
   </div>
  </div>
 );
};
export default StylishCalculator;
```

```
7)**Product Review System**

app---------------
import React from "react";

import ProductReview from "./ProductReview";

const App = () => {

 const sampleProduct = {

  name: "Sample Product",

  image: "https://via.placeholder.com/150", // Replace with actual product image URL

  description: "This is a sample product description.",

  reviews: [

   { username: "Alice", comment: "Great product!", rating: 5 },

   { username: "Bob", comment: "Not bad, could be better.", rating: 3 },

   { username: "Charlie", comment: "Terrible experience.", rating: 1 },

  ],

 };

 return (

  <div>

   <ProductReview product={sampleProduct} />

  </div>

 );

};

export default App;

ProductReview.js----------------------

import React, { useState } from "react";
import PropTypes from "prop-types";
import styles from "./ProductReview.module.css"; // Using CSS Modules
const Review = ({ review }) => (
 <div className={styles.review}>
  <strong>{review.username}:</strong> {review.comment} - <em>{review.rating} ⭐</em>
 </div>
);
Review.propTypes = {
 review: PropTypes.shape({
  username: PropTypes.string.isRequired,
  comment: PropTypes.string.isRequired,
  rating: PropTypes.number.isRequired,
 }).isRequired,
};
const ProductReview = ({ product }) => {
 // Provide a safe default for product
 const safeProduct = product || { name: "", image: "", description: "", reviews: [] };
 const [reviews] = useState(safeProduct.reviews);
 const validateReview = (review) => {
  return review.rating >= 1 && review.rating <= 5;
 };
 return (
  <div className={styles.product}>
```

```
   <h2>{safeProduct.name}</h2>
   {safeProduct.image && <img src={safeProduct.image} alt={safeProduct.name} />}
   <p>{safeProduct.description}</p>
   <h3>Reviews:</h3>
   <div className={styles.reviewSection}>
    {reviews.map((review, index) =>
     validateReview(review) ? <Review key={index} review={review} /> : null
    )}
   </div>
  </div>
 );
};
ProductReview.propTypes = {
 product: PropTypes.shape({
  name: PropTypes.string.isRequired,
  image: PropTypes.string.isRequired,
  description: PropTypes.string.isRequired,
  reviews: PropTypes.arrayOf(
   PropTypes.shape({
    username: PropTypes.string.isRequired,
    comment: PropTypes.string.isRequired,
    rating: PropTypes.number.isRequired,
   })
  ),
 }),

};

export default ProductReview;

ProductReview.module.css----------------------
.product {
border: 1px solid #ddd;
border-radius: 5px;
padding: 16px;
margin: 16px 0;
background-color: #fff;
box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.product img {
max-width: 100%;
height: auto;
border-radius: 5px;
}

.product h2 {
font-size: 24px;
margin: 0 0 8px;
}

.product p {
font-size: 16px;
color: #555;
}

.reviewSection {
margin-top: 24px;
}

.review {
border: 1px solid #e0e0e0;
border-radius: 5px;
padding: 12px;
margin-bottom: 12px;
background-color: #f9f9f9;
}
```

```
.review strong {
color: #333;
}

.review em {
color: #ff9800;
margin-left: 4px;
}

8)**Dynamic Form Styling (Signup Form)**
import React, { useState } from "react";

const SignupForm = () => {

 const [formData, setFormData] = useState({ name: "", email: "", password: "" });

 const [errors, setErrors] = useState({ name: "", email: "", password: "" });

 const handleChange = (e) => {

  const { name, value } = e.target;

  setFormData({ ...formData, [name]: value });

 };

 const validate = () => {

  let isValid = true;

  let newErrors = { name: "", email: "", password: "" };

  if (formData.name.trim() === "") {

   newErrors.name = "Name is required";

   isValid = false;

  }

  if (!/\S+@\S+\.\S+/.test(formData.email)) {

   newErrors.email = "Invalid email format";

   isValid = false;

  }
  if (formData.password.length < 6) {
   newErrors.password = "Password must be at least 6 characters";
   isValid = false;
  }
  setErrors(newErrors);
  return isValid;
 };
 const handleSubmit = (e) => {
  e.preventDefault();
  if (validate()) {
   alert("Form submitted successfully!");
   // Submit logic goes here
  }
 };
 return (
  <form onSubmit={handleSubmit}>
   <div>
    <label>Name:</label>
    <input type="text" name="name" value={formData.name} onChange={handleChange} />
    {errors.name && <span style={{ color: "red" }}>{errors.name}</span>}
   </div>
   <div>
    <label>Email:</label>
    <input type="email" name="email" value={formData.email} onChange={handleChange} />
    {errors.email && <span style={{ color: "red" }}>{errors.email}</span>}
   </div>
   <div>
    <label>Password:</label>
    <input type="password" name="password" value={formData.password} onChange={handleChange} />
    {errors.password && <span style={{ color: "red" }}>{errors.password}</span>}
   </div>
   <button type="submit">Signup</button>
  </form>
 );
};
export default SignupForm;
```

9)E-commerce Product Filter:

```
App.js--------------------------------------------------------
import React from "react";
import ProductFilter from "./ProductFilter";
const App = () => {
 const products = [
  { name: "Smartphone", category: "Electronics" },
  { name: "Jeans", category: "Clothing" },
  { name: "Wall Art", category: "Home Decor" }
 ];
 return (
  <div>
   <ProductFilter products={products} />
  </div>
 );
};
export default App;
```

```
ProductFilter.module.css
.container {
  text-align: center;
 }
 .productList {
  margin-top: 20px;
 }
 .product {
  border: 1px solid #ccc;
  padding: 10px;
  margin-bottom: 10px;
 }
```

```
ProductFilter.js----------------------
```

```
import React, { useState } from "react";
import PropTypes from "prop-types";
import styles from "./ProductFilter.module.css";
const Product = ({ product }) => (
 <div className={styles.product}>
  <h3>{product.name}</h3>
  <p>Category: {product.category}</p>
 </div>
);
Product.propTypes = {
 product: PropTypes.shape({
  name: PropTypes.string.isRequired,
  category: PropTypes.string.isRequired,
 }),
};
const ProductFilter = ({ products }) => {
 const [category, setCategory] = useState("All");
 const handleFilterChange = (e) => {
  setCategory(e.target.value);
 };
 const filteredProducts = category === "All"
  ? products
  : products.filter((product) => product.category === category);
 return (
  <div className={styles.container}>
   <h1>Product Filter</h1>
   <select onChange={handleFilterChange}>
    <option value="All">All</option>
    <option value="Electronics">Electronics</option>
    <option value="Clothing">Clothing</option>
    <option value="Home Decor">Home Decor</option>
   </select>
   <div className={styles.productList}>
    {filteredProducts.map((product) => (
     <Product key={product.name} product={product} />
    ))}
   </div>
  </div>
 );
};
ProductFilter.propTypes = {
 products: PropTypes.arrayOf(
  PropTypes.shape({
   name: PropTypes.string.isRequired,
   category: PropTypes.string.isRequired,
  })
 ).isRequired,
};
export default ProductFilter;
```

10)News feed

```
import React, { useState, useEffect } from "react";
const NewsFeed = () => {
 const [articles, setArticles] = useState([]);
```

```
useEffect(() => {
 // Fetch news articles (using placeholder API)
 const fetchArticles = async () => {
  const response = await fetch("https://jsonplaceholder.typicode.com/posts");
  const data = await response.json();
  setArticles(data.slice(0, 5)); // Simulate a small news feed
 };
 fetchArticles();
}, []);
return (
 <div>
  <h1>News Feed</h1>
  <div>
   {articles.map((article, index) => (
    <div
     key={article.id}
     style={{
      border: "1px solid #ccc",
      padding: "10px",
      marginBottom: "10px",
      backgroundColor: index === 0 ? "lightyellow" : "white",
     }}
    >
     <h3>{article.title}</h3>
```

```
      <p>{article.body}</p>
      <em>Author: {article.userId}</em>
     </div>
    ))}
   </div>
  </div>
 );
};
export default NewsFeed;
```