1. Create a UserCard component that takes name and age as props and displays them.

export const UserCard = ({ name, age }) => (

<div>

<h3>{name}</h3>

<p>Age: {age}</p>

</div>

);

function App() {

return (

<div>

<h1>User Information</h1>

{/\* Passing props to UserCard \*/}

<UserCard name="John Doe" age={28} />

<UserCard name="Jane Smith" age={32} />

</div>

);

}

export default App;

2. Pass a list of hobbies as a prop to a HobbiesList component and render them as <li> items.

import React from "react";

export const HobbiesList = ({ hobbies }) => (

<ul>

{hobbies.map((hobby, index) => (

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

))}

</ul>

);

function App() {

const hobbies = ["Reading", "Cycling", "Gaming"];

return (

<div>

<h1>My Hobbies</h1>

<HobbiesList hobbies={hobbies} />

</div>

);

}

export default App;

3. Build a Button component that takes label and color as props and styles itself.

import React from "react";

export const CustomButton = ({ label, color }) => (

<button style={{ backgroundColor: color, padding: "10px", color: "white" }}>

{label}

</button>

);

function App() {

return (

<div>

<CustomButton label="Click Me" color="blue" />

<CustomButton label="Delete" color="red" />

</div>

);

}

export default App;

4. Create a Profile component that takes a user object prop and displays the username and email.

import React from "react";

export const Profile = ({ user }) => (

<div>

<h3>{user.username}</h3>

<p>{user.email}</p>

</div>

);

function App() {

const userData = { username: "john\_doe", email: "john@example.com" };

return (

<div>

<Profile user={userData} />

</div>

);

}

export default App;

5. Pass a function as a prop to a ClickButton component that logs “Button clicked!” when pressed.

import React from "react";

export const ClickButton = ({ onClick }) => (

<button onClick={onClick}>Click Me</button>

);

function App() {

const handleClick = () => {

console.log("Button clicked!");

};

return (

<div>

<ClickButton onClick={handleClick} />

</div>

);

}

export default App;6. Build a Greeting component that displays “Good Morning” or “Good Evening” based on a time prop. \*/

export const Greeting = ({ time }) => (

<h3>{time < 12 ? "Good Morning" : "Good Evening"}</h3>

);

7. Create a Counter component where the starting value is passed as a prop. \*/

import React, { useState } from "react";

export const Counter = ({ start }) => {

const [count, setCount] = useState(start);

return (

<div>

<p>Count: {count}</p>

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

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

</div>

);

};

function App() {

return (

<div>

<Counter start={5} />

</div>

);

}

export default App;

8. Pass an image URL as a prop to an Avatar component and render it in <img>. \*/

import React from "react";

export const Avatar = ({ src }) => (

<img src={src} alt="avatar" style={{ width: "100px", borderRadius: "50%" }} />

);

function App() {

return (

<div>

<Avatar src="https://via.placeholder.com/100" />

</div>

);

}

export default App;

9. Create a Card component that takes title and children as props and displays them in a styled card.

import React from "react";

export const Card = ({ title, children }) => (

<div style={{ border: "1px solid gray", padding: "10px", borderRadius: "8px" }}>

<h3>{title}</h3>

<div>{children}</div>

</div>

);

function App() {

return (

<div>

<Card title="About Me">

<p>Hello, I’m a React Developer!</p>

</Card>

</div>

);

}

export default App;

10. Build a Product component that receives price and discount props and displays the discounted price.

import React from "react";

export const Product = ({ price, discount }) => {

const discountedPrice = price - (price \* discount) / 100;

return (

<div>

<p>Original Price: ${price}</p>

<p>Discount: {discount}%</p>

<p>Discounted Price: ${discountedPrice}</p>

</div>

);

};

function App() {

return (

<div>

<Product price={100} discount={20} />

</div>

);

}

export default App;

2. HOOKS – Practice Tasks

1. Create a Timer component using useState and useEffect to count seconds.

import React, { useState } from "react";

function App() {

const [text, setText] = useState("Hello");

return (

<div>

<p>{text}</p>

<button onClick={() => setText(text === "Hello" ? "Goodbye" : "Hello")}>

Toggle Text

</button>

</div>

);

}

export default App;

2. Build a MouseTracker component that shows the current mouse position using import React, { useState } from "react";

function App() {

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

return (

<div>

<p>Count: {count}</p>

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

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

</div>

);

}

export default App;

3. Use useRef to focus an input field when a button is clicked.

import React, { useState } from "react";

function App() {

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

return (

<div>

<input

type="text"

placeholder="Enter your name"

onChange={(e) => setName(e.target.value)}

/>

<p>Your name: {name}</p>

</div>

);

}

export default App;

4. Create a form that uses useState to store and display input values in real time.

import React, { useState } from "react";

function App() {

const [visible, setVisible] = useState(true);

return (

<div>

<button onClick={() => setVisible(!visible)}>

{visible ? "Hide" : "Show"}

</button>

{visible && <p>This is visible!</p>}

</div>

);

}

export default App;

5. Build a theme switcher using useState and useEffect to store the selected theme in localStorage.

import React, { useEffect } from "react";

function App() {

useEffect(() => {

console.log("Component mounted!");

}, []);

return <h1>Check the console!</h1>;

}

export default App;

6. Use useReducer to create a counter with increment, decrement, and reset buttons.

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

function App() {

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

useEffect(() => {

console.log("Count changed:", count);

}, [count]);

return (

<div>

<p>Count: {count}</p>

<button onClick={() => setCount(count + 1)}>Increase</button>

</div>

);

}

export default App;

7. Use useMemo to calculate and display a list of prime numbers up to a given number.

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

function App() {

const [seconds, setSeconds] = useState(0);

useEffect(() => {

const timer = setInterval(() => setSeconds((prev) => prev + 1), 1000);

return () => clearInterval(timer);

}, []);

return <p>Time: {seconds}s</p>;

}

export default App;

8. Create a WindowSize component using useEffect to track window width/height.

import React, { useState } from "react";

function App() {

const [color, setColor] = useState("white");

return (

<div style={{ backgroundColor: color, height: "100vh", padding: "20px" }}>

<button onClick={() => setColor("lightblue")}>Blue</button>

<button onClick={() => setColor("lightgreen")}>Green</button>

<button onClick={() => setColor("pink")}>Pink</button>

</div>

);

}

export default App;

9. Build a Stopwatch component that starts, stops, and resets using hooks.

import React, { useState } from "react";

function App() {

const [form, setForm] = useState({ name: "", email: "" });

return (

<div>

<input

type="text"

placeholder="Name"

onChange={(e) => setForm({ ...form, name: e.target.value })}

/>

<input

type="email"

placeholder="Email"

onChange={(e) => setForm({ ...form, email: e.target.value })}

/>

<p>

Name: {form.name} | Email: {form.email}

</p>

</div>

);

}

export default App;

10. Create a DarkModeToggle component using useState and useEffect to switch background color.

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

function App() {

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

useEffect(() => {

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

.then((res) => res.json())

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

}, []);

return (

<div>

<h1>User List</h1>

<ul>

{users.map((u) => (

<li key={u.id}>{u.name}</li>

))}

</ul>

</div>

);

}

export default App;

3. STATE – Practice Tasks

1. Create a counter using useState with + and – buttons. \*/

import React, { useState } from "react";

const UserCard = ({ name, age }) => (

<div>

<h3>{name}</h3>

<p>Age: {age}</p>

</div>

);

function App() {

const [show, setShow] = useState(true);

return (

<div>

<button onClick={() => setShow(!show)}>

{show ? "Hide User" : "Show User"}

</button>

{show && <UserCard name="John Doe" age={28} />}

</div>

);

}

export default App;

2. Make a toggle switch that changes between “ON” and “OFF” state.

import React, { useState } from "react";

const Counter = ({ title }) => {

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

return (

<div>

<h2>{title}</h2>

<p>{count}</p>

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

</div>

);

};

function App() {

return <Counter title="Click Counter" />;

}

export default App;

4. Create a random quote generator where each click changes the displayed quote.

import React, { useState } from "react";

const Greeting = ({ name }) => <h1>Hello, {name}!</h1>;

function App() {

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

return (

<div>

<Greeting name={name} />

<input

type="text"

placeholder="Enter your name"

onChange={(e) => setName(e.target.value)}

/>

</div>

);

}

export default App;

5. Store an array of todo items in state and render them in a list.

import React, { useState } from "react";

const Task = ({ text }) => <li>{text}</li>;

function App() {

const [tasks, setTasks] = useState(["Learn React", "Build a project"]);

const [newTask, setNewTask] = useState("");

const addTask = () => {

if (newTask.trim()) {

setTasks([...tasks, newTask]);

setNewTask("");

}

};

return (

<div>

<ul>

{tasks.map((t, index) => (

<Task key={index} text={t} />

))}

</ul>

<input

type="text"

placeholder="New task"

value={newTask}

onChange={(e) => setNewTask(e.target.value)}

/>

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

</div>

);

}

export default App;

6. Create a text input that converts text to uppercase in real time.

import React, { useState } from "react";

const Profile = ({ name, bio }) => (

<div>

<h2>{name}</h2>

<p>{bio}</p>

</div>

);

function App() {

const [show, setShow] = useState(false);

return (

<div>

<button onClick={() => setShow(!show)}>

{show ? "Hide Profile" : "Show Profile"}

</button>

{show && (

<Profile name="Jane Smith" bio="Software Developer from NY" />

)}

</div>

);

}

export default App;

7. Make a “Like” button that increments a count each time it’s clicked.

import React, { useState } from "react";

const ColorBox = ({ color }) => (

<div

style={{

backgroundColor: color,

width: "100px",

height: "100px",

margin: "10px",

}}

></div>

);

function App() {

const [color, setColor] = useState("lightblue");

return (

<div>

<ColorBox color={color} />

<button onClick={() => setColor("lightblue")}>Blue</button>

<button onClick={() => setColor("lightgreen")}>Green</button>

<button onClick={() => setColor("pink")}>Pink</button>

</div>

);

}

export default App;

8. Build a color picker that changes the page background color. \*/

import React, { useState } from "react";

const LikeButton = ({ label }) => {

const [likes, setLikes] = useState(0);

return (

<div>

<p>{label}: {likes} Likes</p>

<button onClick={() => setLikes(likes + 1)}>Like</button>

</div>

);

};

function App() {

return (

<>

<LikeButton label="Post 1" />

<LikeButton label="Post 2" />

</>

);

}

export default App;

9. Store a list of images in state and cycle through them with next/previous buttons.

import React, { useState } from "react";

const Weather = ({ city, temperature }) => (

<p>

{city}: {temperature}°C

</p>

);

function App() {

const [temp, setTemp] = useState(25);

return (

<div>

<Weather city="Delhi" temperature={temp} />

<button onClick={() => setTemp(temp + 1)}>Increase Temp</button>

<button onClick={() => setTemp(temp - 1)}>Decrease Temp</button>

</div>

);

}

export default App;

10. Implement a “character counter” for a textarea.

import React, { useState } from "react";

const CartItem = ({ name, price }) => (

<li>

{name} - ${price}

</li>

);

function App() {

const [cart, setCart] = useState([]);

const products = [

{ name: "Shoes", price: 50 },

{ name: "Shirt", price: 30 },

{ name: "Hat", price: 20 },

];

const addToCart = (product) => {

setCart([...cart, product]);

};

return (

<div>

<h2>Products</h2>

{products.map((p, index) => (

<div key={index}>

{p.name} - ${p.price}{" "}

<button onClick={() => addToCart(p)}>Add to Cart</button>

</div>

))}

<h2>Cart</h2>

<ul>

{cart.map((item, index) => (

<CartItem key={index} name={item.name} price={item.price} />

))}

</ul>

</div>

);

}

export default App;