Props – Practice Tasks

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

```
function UserCard({ name, age }) {
  return <p>{name} is {age} years old</p>;
}
<UserCard name="Raksha" age={22} />
```

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

```
function HobbiesList({ hobbies }) {
  return (
    <ul>
      {hobbies.map((h, i) => <li key={i}>{h}</li>)}
    </ul>
  );
}
<HobbiesList hobbies={["Reading", "Dancing", "Cooking"]} />
```

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

```
function Button({ label, color }) {
  return <button style={{ background: color }}>{label}</button>;
}
<Button label="Click Me" color="lightblue" />
```

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

```
function Profile({ user }) {
  return <p>{user.username} – {user.email}</p>;
}
<Profile user={{ username: "raksha123", email: "raksha@example.com" }} />
```

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

```
function ClickButton({ onClick }) {
  return <button onClick={onClick}>Click Me</button>;
}
<ClickButton onClick={() => console.log("Button clicked!")} />
```

6. Build a Greeting component that displays “Good Morning” or “Good Evening” based on a time prop.

```
function TimeGreet({ time }) {
  return <h2>{time < 12 ? "Good Morning" : "Good Evening"}</h2>;
}
<TimeGreet time={10} />
<TimeGreet time={15} />
```

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

```
import React,{useState,useEffect} from 'react';
function Welcome(props){
 const [count,setCount] = useState(0);
 useEffect( () =>{
 console.log(`welcome component for ${props.name} mounted`);
  } ,[ ]);
  useEffect( () =>{
 console.log(`counter for ${props.name} updated to ${count}`);
  } ,[count]);
  useEffect( () =>{
  return () =>{
 console.log(`welcome component for ${props.name} unmounted`);
 };
```

```
 } ,[ ]);
 return(
  <div style={{ textAlign:"center",marginTop:"50px"}}>
  <h1>Welcome,{props.name}</h1>
  <h2>counter:{count}</h2>
  <p> it is reusable component.</p>
  <button onClick ={()=>setCount(count+1)}>Increase</button>
  <button onClick ={()=>setCount(count-1)}>Decrease</button>
  </div>
);
}
export default Welcome;
```

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

```
function Avatar({ url }) {
  return <img src={url} alt="avatar" width="100" />;
}
<Avatar url="https://cdn.pixabay.com/photo/2023/07/01/22/08/avatar-8100963_640.jpg" />
```

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

```
function Card({ title, children }) {
  return (
    <div style={{ border: "1px solid #ccc", padding: "10px" }}>
      <h3>{title}</h3>
      {children}
    </div>
  );
}
```

<Card title="My Card">This is inside the card</Card>

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

```
function Product({ price, discount }) {
  const discounted = price - (price * discount) / 100;
  return <p>Discounted Price: ₹{discounted}</p>;
}
<Product price={1000} discount={10} />
```

Hooks – Practice Tasks

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

```
import React, { useState, useEffect } from 'react';
function Timer() {
  const [seconds, setSeconds] = useState(0);

  useEffect(() => {
    const timer = setInterval(() => {
      setSeconds(prev => prev + 1);
    }, 1000);
    return () => clearInterval(timer);
  }, [ ]);
  return <h1>Timer: {seconds} seconds</h1>;
}
export default Timer;
```

2. Build a MouseTracker component that shows the current mouse position using useState and useEffect.

```
import React, { useState, useEffect } from 'react';
function MouseTracker() {
  const [pos, setPos] = useState({ x: 0, y: 0 });
  useEffect(() => {
    const handleMove = e => {
      setPos({ x: e.clientX, y: e.clientY });
    };
    window.addEventListener('mousemove', handleMove);
    return () => window.removeEventListener('mousemove', handleMove);
  }, [ ]);
  return <h2>Mouse Position: X = {pos.x}, Y = {pos.y}</h2>;
}
export default MouseTracker;
```

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

```
import React, { useRef } from 'react';
function FocusInput() {
  const inputRef = useRef(null);
  const handleFocus = () => {
    inputRef.current.focus();
  };
  return (
    <div>
      <input ref={inputRef} placeholder="Type here" />
      <button onClick={handleFocus}>Focus Input</button>
    </div>
```

```
  );
}
export default FocusInput;
```

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

```
import React, { useState } from 'react';
function RealForm() {
  const [text, setText] = useState('');
  return (
    <div>
      <input value={text} onChange={e => setText(e.target.value)} placeholder="Type something" />
      <p>You typed: {text}</p>
    </div>
  );
}
export default RealForm;
```

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

```
import React, { useState, useEffect } from 'react';
function ThemeSwitcher() {
  const [theme, setTheme] = useState(localStorage.getItem('theme') || 'light');
  useEffect(() => {
    localStorage.setItem('theme', theme);
    document.body.style.background = theme === 'light' ? '#fff' : '#333';
    document.body.style.color = theme === 'light' ? '#000' : '#fff';
  }, [theme]);
  return (
```

```
    <button onClick={() => setTheme(theme === 'light' ? 'dark' : 'light')}>
      Switch to {theme === 'light' ? 'Dark' : 'Light'} Mode
    </button>
  );
}
export default ThemeSwitcher;
```

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

```
import React, { useReducer } from 'react';
const reducer = (state, action) => {
  switch (action) {
    case 'inc': return state + 1;
    case 'dec': return state - 1;
    case 'reset': return 0;
    default: return state;
  }
};
function CounterReducer() {
  const [count, dispatch] = useReducer(reducer, 0);
  return (
    <div>
      <h1>{count}</h1>
      <button onClick={() => dispatch('inc')}>increment</button>
      <button onClick={() => dispatch('dec')}>decrement</button>
      <button onClick={() => dispatch('reset')}>Reset</button>
    </div>
  );
```

```
}
export default CounterReducer;
```

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

```
import React, { useState, useMemo } from 'react';
function PrimeList() {
  const [num, setNum] = useState(10);
  const primes = useMemo(() => {
    const arr = [ ];
    for (let i = 2; i <= num; i++) {
      if (arr.every(p => i % p !== 0)) arr.push(i);
    }
    return arr;
  }, [num]);
  return (
    <div>
      <input type="number" value={num} onChange={e => setNum(+e.target.value)} />
      <p>Primes: {primes.join(', ')}</p>
    </div>
  );
}
export default PrimeList;
```

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

```
import React, { useState, useEffect } from 'react';
function WindowSize( ) {
  const [size, setSize] = useState({ w: window.innerWidth, h: window.innerHeight });
  useEffect(( ) => {
```

```
    const resizeHandler = ( ) => setSize({ w: window.innerWidth, h: window.innerHeight });
    window.addEventListener('resize', resizeHandler);
    return ( ) => window.removeEventListener('resize', resizeHandler);
  }, [ ]);
  return <h2>Width: {size.w}, Height: {size.h}</h2>;
}
export default WindowSize;
```

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

```
import React, { useState, useEffect } from 'react';
function Stopwatch() {
  const [time, setTime] = useState(0);
  const [running, setRunning] = useState(false);
  useEffect(() => {
    let interval;
    if (running) {
      interval = setInterval(() => setTime(t => t + 1), 1000);
    }
    return () => clearInterval(interval);
  }, [running]);
  return (
    <div>
      <h1>{time}s</h1>
      <button onClick={() => setRunning(true)}>Start</button>
      <button onClick={() => setRunning(false)}>Stop</button>
      <button onClick={() => setTime(0)}>Reset</button>
    </div>
```

```
  );
}
export default Stopwatch;
```

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

```
import React, { useState, useEffect } from 'react';
function DarkModeToggle() {
  const [dark, setDark] = useState(false);
  useEffect(() => {
    document.body.style.background = dark ? '#000' : '#fff';
    document.body.style.color = dark ? '#fff' : '#000';
  }, [dark]);
  return (
    <button onClick={() => setDark(!dark)}>
      {dark ? 'Light Mode' : 'Dark Mode'}
    </button>
  );
}
export default DarkModeToggle;
```

State – Practice Tasks

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

```
import React, { useState } from 'react';
function SimpleCounter() {
  const [count, setCount] = useState(0);
  return (
    <div>
```

```
      <h1>{count}</h1>
      <button onClick={() => setCount(count + 1)}>+</button>
      <button onClick={() => setCount(count - 1)}>-</button>
    </div>
  );
}
export default SimpleCounter;
```

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

```
import React, { useState } from 'react';
function ToggleSwitch() {
  const [on, setOn] = useState(true);
  return (
    <button onClick={() => setOn(!on)}>
      {on ? 'ON' : 'OFF'}
    </button>
  );
}
export default ToggleSwitch;
```

3. Build a form that stores name, email, and age in a single state object.

```
import React, { useState } from 'react';
function MultiForm() {
  const [form, setForm] = useState({ name: '', email: '', age: '' });
  return (
    <div>
      <input placeholder="Name" onChange={e => setForm({ ...form, name: e.target.value })} />
```

```
      <input placeholder="Email" onChange={e => setForm({ ...form, email: e.target.value })} />
      <input placeholder="Age" onChange={e => setForm({ ...form, age: e.target.value })} />
      <p>{JSON.stringify(form)}</p>
    </div>
  );
}
export default MultiForm;
```

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

```
import React, { useState } from 'react';
function RandomQuote() {
  const quotes = ["Keep going!", "You can do it!", "React is fun!"];
  const [quote, setQuote] = useState(quotes[0]);
  return (
    <div>
      <p>{quote}</p>
      <button onClick={() => setQuote(quotes[Math.floor(Math.random() * quotes.length)])}>
        New Quote
      </button>
    </div>
  );
}
export default RandomQuote;
```

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

```
import React, { useState } from 'react';
function TodoList() {
  const [todos, setTodos] = useState([]);
  const [item, setItem] = useState('');
  return (
    <div>
      <input value={item} onChange={e => setItem(e.target.value)} />
      <button onClick={() => { setTodos([...todos, item]); setItem(''); }}>Add</button>
      <ul>
        {todos.map((t, i) => <li key={i}>{t}</li>)}
      </ul>
    </div>
  );
}
export default TodoList;
```

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

```
import React, { useState } from 'react';
function UppercaseInput() {
  const [text, setText] = useState('');
  return (
    <input value={text} onChange={e => setText(e.target.value.toUpperCase())} />
  );
}
export default UppercaseInput;
```

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

```
import React, { useState } from 'react';
function LikeButton() {
  const [likes, setLikes] = useState(0);
  return (
    <button onClick={() => setLikes(likes + 1)}>
       {likes}
    </button>
  );
}
export default LikeButton;
```

8. Build a color picker that changes the page background color.

```
import React, { useState } from 'react';
function ColorPicker() {
  const [color, setColor] = useState('#ffffff');
  return (
    <div>
      <input type="color" onChange={e => setColor(e.target.value)} />
      <div style={{ background: color, width: '100px', height: '100px' }}></div>
    </div>
  );
}
export default ColorPicker;
```

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

```
import React, { useState } from 'react';
function ImageSlider() {
  const images = ['img1.jpg', 'img2.jpg', 'img3.jpg'];
  const [index, setIndex] = useState(0);
  return (
    <div>
      <img src={images[index]} alt="" width="200" />
      <br />
      <button onClick={() => setIndex((index - 1 + images.length) % images.length)}>Prev</button>
      <button onClick={() => setIndex((index + 1) % images.length)}>Next</button>
    </div>
  );
}
export default ImageSlider;
```

10. Implement a “character counter” for a textarea

```
import React, { useState } from 'react';
function CharCounter() {
  const [text, setText] = useState('');
  return (
    <div>
      <textarea value={text} onChange={e => setText(e.target.value)} />
      <p>Characters: {text.length}</p>
    </div>
  );
}
export default CharCounter;
```