**Day-20 Morning Assessment**

**Props**

1. const UserCard = ({ name, age }) => (  
<div>  
<h2>{name}</h2>  
<p>Age: {age}</p>  
</div>  
);  
  
2. const HobbiesList = ({ hobbies }) => (  
<ul>  
   {hobbies.map((hobby, i) => <li key={i}>{hobby}</li>)}  
</ul>  
);

3.const Button = ({ label, color }) => (  
<button style={{ backgroundColor: color }}>{label}</button>  
);  
  
4. const Profile = ({ user }) => (  
<div>  
<h3>{user.username}</h3>  
<p>{user.email}</p>  
</div>  
);  
  
5. const ClickButton = ({ onClick }) => (  
<button onClick={onClick}>Click Me</button>  
);  
  
6. const Greeting = ({ time }) => (  
<h2>{time < 12 ? "Good Morning" : "Good Evening"}</h2>  
);  
  
7. const Counter = ({ start }) => {  
 const [count, setCount] = React.useState(start);  
 return (  
<div>  
<p>{count}</p>  
<button onClick={() => setCount(count + 1)}>+</button>  
</div>  
 );  
};  
  
8. const Avatar = ({ url }) => <img src={url} alt="Avatar" width="100" />;  
  
9. const Card = ({ title, children }) => (  
<div style={{ border: "1px solid #ccc", padding: "10px" }}>  
<h3>{title}</h3>  
   {children}  
</div>  
);  
  
10. const Product = ({ price, discount }) => (  
<p>Discounted Price: ₹{price - (price \* discount) / 100}</p>  
);  
  
**Hooks**

1. const Timer = () => {  
 const [seconds, setSeconds] = React.useState(0);  
 React.useEffect(() => {  
   const timer = setInterval(() => setSeconds(s => s + 1), 1000);  
   return () => clearInterval(timer);  
 }, []);  
 return <p>Seconds: {seconds}</p>;  
};  
  
2. const MouseTracker = () => {  
 const [pos, setPos] = React.useState({ x: 0, y: 0 });  
 React.useEffect(() => {  
   const handleMove = e => setPos({ x: e.clientX, y: e.clientY });  
   window.addEventListener("mousemove", handleMove);  
   return () => window.removeEventListener("mousemove", handleMove);  
 }, []);  
 return <p>X: {pos.x}, Y: {pos.y}</p>;  
};  
  
3. const FocusInput = () => {  
 const inputRef = React.useRef();  
 return (  
<div>  
<input ref={inputRef} />  
<button onClick={() => inputRef.current.focus()}>Focus</button>  
</div>  
 );  
};  
  
4.const RealTimeForm = () => {  
 const [input, setInput] = React.useState("");  
 return (  
<>  
<input value={input} onChange={e => setInput(e.target.value)} />  
<p>{input}</p>  
</>  
 );  
};  
  
5. const ThemeSwitcher = () => {  
 const [theme, setTheme] = React.useState(localStorage.getItem("theme") || "light");  
 React.useEffect(() => {  
   document.body.style.background = theme === "light" ? "#fff" : "#333";  
   localStorage.setItem("theme", theme);  
 }, [theme]);  
 return (  
<button onClick={() => setTheme(theme === "light" ? "dark" : "light")}>  
     Switch to {theme === "light" ? "Dark" : "Light"} Mode  
</button>  
 );  
};  
  
6. const reducer = (state, action) => {  
 switch (action.type) {  
   case "inc": return state + 1;  
   case "dec": return state - 1;  
   case "reset": return 0;  
   default: return state;  
 }  
};  
const ReducerCounter = () => {  
 const [count, dispatch] = React.useReducer(reducer, 0);  
 return (  
<>  
<p>{count}</p>  
<button onClick={() => dispatch({ type: "inc" })}>+</button>  
<button onClick={() => dispatch({ type: "dec" })}>-</button>  
<button onClick={() => dispatch({ type: "reset" })}>Reset</button>  
</>  
 );  
};  
  
7. const PrimeList = ({ limit }) => {  
 const primes = React.useMemo(() => {  
   const nums = [];  
   for (let i = 2; i <= limit; i++) {  
     if (nums.every(n => i % n !== 0)) nums.push(i);  
   }  
   return nums;  
 }, [limit]);  
 return <p>{primes.join(", ")}</p>;  
};  
  
8.const WindowSize = () => {  
 const [size, setSize] = React.useState({ w: window.innerWidth, h: window.innerHeight });  
 React.useEffect(() => {  
   const handleResize = () => setSize({ w: window.innerWidth, h: window.innerHeight });  
   window.addEventListener("resize", handleResize);  
   return () => window.removeEventListener("resize", handleResize);  
 }, []);  
 return <p>{size.w} x {size.h}</p>;  
};  
  
9. const Stopwatch = () => {  
 const [time, setTime] = React.useState(0);  
 const [running, setRunning] = React.useState(false);  
 React.useEffect(() => {  
   let timer;  
   if (running) timer = setInterval(() => setTime(t => t + 1), 1000);  
   return () => clearInterval(timer);  
 }, [running]);  
 return (  
<>  
<p>{time}s</p>  
<button onClick={() => setRunning(true)}>Start</button>  
<button onClick={() => setRunning(false)}>Stop</button>  
<button onClick={() => setTime(0)}>Reset</button>  
</>  
 );  
};  
  
10. const DarkModeToggle = () => {  
 const [dark, setDark] = React.useState(false);  
 React.useEffect(() => {  
   document.body.style.background = dark ? "#000" : "#fff";  
   document.body.style.color = dark ? "#fff" : "#000";  
 }, [dark]);  
 return <button onClick={() => setDark(!dark)}>Toggle Dark Mode</button>;  
};  
  
**State**

1. const SimpleCounter = () => {  
 const [count, setCount] = React.useState(0);  
 return (  
<>  
<p>{count}</p>  
<button onClick={() => setCount(count + 1)}>+</button>  
<button onClick={() => setCount(count - 1)}>-</button>  
</>  
 );  
};  
  
2. const ToggleSwitch = () => {  
 const [on, setOn] = React.useState(false);  
 return <button onClick={() => setOn(!on)}>{on ? "ON" : "OFF"}</button>;  
};  
  
3. const FormState = () => {  
 const [form, setForm] = React.useState({ name: "", email: "", age: "" });  
 return (  
<>  
<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>  
</>  
 );  
};  
  
4. const QuoteGenerator = () => {  
 const quotes = ["Quote 1", "Quote 2", "Quote 3"];  
 const [quote, setQuote] = React.useState(quotes[0]);  
 return (  
<>  
<p>{quote}</p>  
<button onClick={() => setQuote(quotes[Math.floor(Math.random() \* quotes.length)])}>  
       New Quote  
</button>  
</>  
 );  
};  
  
5. const TodoList = () => {  
 const [todos, setTodos] = React.useState([]);  
 const [task, setTask] = React.useState("");  
 return (  
<>  
<input value={task} onChange={e => setTask(e.target.value)} />  
<button onClick={() => { setTodos([...todos, task]); setTask(""); }}>Add</button>  
<ul>{todos.map((t, i) => <li key={i}>{t}</li>)}</ul>  
</>  
 );  
};  
  
6. const UppercaseInput = () => {  
 const [text, setText] = React.useState("");  
 return (  
<>  
<input value={text} onChange={e => setText(e.target.value.toUpperCase())} />  
<p>{text}</p>  
</>  
 );  
};  
  
7. const LikeButton = () => {  
 const [likes, setLikes] = React.useState(0);  
 return <button onClick={() => setLikes(likes + 1)}>👍 {likes}</button>;  
};  
  
8. const ColorPicker = () => {  
 const [color, setColor] = React.useState("#ffffff");  
 React.useEffect(() => {  
   document.body.style.background = color;  
 }, [color]);  
 return <input type="color" value={color} onChange={e => setColor(e.target.value)} />;  
};  
  
9. const ImageCarousel = () => {  
 const images = ["img1.jpg", "img2.jpg", "img3.jpg"];  
 const [index, setIndex] = React.useState(0);  
 return (  
<>  
<img src={images[index]} alt="carousel" width="200" />  
<button onClick={() => setIndex((index - 1 + images.length) % images.length)}>Prev</button>  
<button onClick={() => setIndex((index + 1) % images.length)}>Next</button>  
</>  
 );  
};  
  
10. const CharCounter = () => {  
 const [text, setText] = React.useState("");  
 return (  
<>  
<textarea value={text} onChange={e => setText(e.target.value)} />  
<p>Characters: {text.length}</p>  
</>  
 );  
};