# DAY-20 MORNING ASSESSMENT

Props practice tasks

1. UserCard  
function UserCard({ name, age }) {  
 return <p>{name} - {age} years old</p>;  
}  
  
2. HobbiesList  
function HobbiesList({ hobbies }) {  
 return (  
<ul>  
     {hobbies.map((hobby, i) => <li key={i}>{hobby}</li>)}  
</ul>  
 );  
}  
  
 3. Button  
function Button({ label, color }) {  
 return <button style={{ backgroundColor: color }}>{label}</button>;  
}  
  
 4. Profile  
function Profile({ user }) {  
 return <p>{user.username} - {user.email}</p>;  
}  
  
 5. ClickButton  
function ClickButton({ onClick }) {  
 return <button onClick={onClick}>Click Me</button>;  
}  
  
6. Greeting  
function Greeting({ time }) {  
 return <h2>{time < 12 ? "Good Morning" : "Good Evening"}</h2>;  
}  
  
7. Counter  
function Counter({ start }) {  
 const [count, setCount] = React.useState(start);  
 return (  
<>  
<p>{count}</p>  
<button onClick={() => setCount(count + 1)}>+</button>  
</>  
 );  
}  
  
8. Avatar  
function Avatar({ url }) {  
 return <img src={url} alt="Avatar" width="100" />;  
}  
  
9. Card  
function Card({ title, children }) {  
 return (  
<div style={{ border: "1px solid black", padding: "10px" }}>  
<h3>{title}</h3>  
     {children}  
</div>  
 );  
}  
  
10. Product  
function Product({ price, discount }) {  
 const discountedPrice = price - (price \* discount / 100);  
 return <p>Price: ${discountedPrice.toFixed(2)}</p>;  
}

HOOKS

1. Timer  
function Timer() {  
 const [seconds, setSeconds] = React.useState(0);  
 React.useEffect(() => {  
   const interval = setInterval(() => setSeconds(s => s + 1), 1000);  
   return () => clearInterval(interval);  
 }, []);  
 return <p>{seconds}s</p>;  
}  
  
2. MouseTracker  
function MouseTracker() {  
 const [pos, setPos] = React.useState({ x: 0, y: 0 });  
 React.useEffect(() => {  
   const handler = e => setPos({ x: e.clientX, y: e.clientY });  
   window.addEventListener("mousemove", handler);  
   return () => window.removeEventListener("mousemove", handler);  
 }, []);  
 return <p>X: {pos.x}, Y: {pos.y}</p>;  
}  
  
3. Focus input with useRef  
function FocusInput() {  
 const inputRef = React.useRef();  
 return (  
<>  
<input ref={inputRef} />  
<button onClick={() => inputRef.current.focus()}>Focus</button>  
</>  
 );  
}

4. Form with live display  
function LiveForm() {  
 const [value, setValue] = React.useState("");  
 return (  
<>  
<input value={value} onChange={e => setValue(e.target.value)} />  
<p>{value}</p>  
</>  
 );  
}  
  
5. Theme switcher (localStorage)  
function ThemeSwitcher() {  
 const [theme, setTheme] = React.useState(() => localStorage.getItem("theme") || "light");  
 React.useEffect(() => localStorage.setItem("theme", theme), [theme]);  
 return (  
<button onClick={() => setTheme(t => t === "light" ? "dark" : "light")}>  
     Switch to {theme === "light" ? "dark" : "light"}  
</button>  
 );  
}  
  
6. Counter with useReducer  
function ReducerCounter() {  
 function reducer(state, action) {  
   switch(action.type) {  
     case "inc": return state + 1;  
     case "dec": return state - 1;  
     case "reset": return 0;  
     default: return state;  
   }  
 }  
 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. useMemo prime numbers  
function PrimeList({ num }) {  
 const primes = React.useMemo(() => {  
   const result = [];  
   for(let i=2; i<=num; i++) {  
     if(result.every(p => i % p !== 0)) result.push(i);  
   }  
   return result;  
 }, [num]);  
 return <p>{primes.join(", ")}</p>;  
}  
  
8. WindowSize  
function WindowSize() {  
 const [size, setSize] = React.useState({ w: window.innerWidth, h: window.innerHeight });  
 React.useEffect(() => {  
   const handler = () => setSize({ w: window.innerWidth, h: window.innerHeight });  
   window.addEventListener("resize", handler);  
   return () => window.removeEventListener("resize", handler);  
 }, []);  
 return <p>{size.w} x {size.h}</p>;  
}  
  
9. Stopwatch  
function Stopwatch() {  
 const [time, setTime] = React.useState(0);  
 const [running, setRunning] = React.useState(false);  
 React.useEffect(() => {  
   if (!running) return;  
   const id = setInterval(() => setTime(t => t + 1), 1000);  
   return () => clearInterval(id);  
 }, [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. DarkModeToggle  
function DarkModeToggle() {  
 const [dark, setDark] = React.useState(false);  
 React.useEffect(() => {  
   document.body.style.backgroundColor = dark ? "black" : "white";  
   document.body.style.color = dark ? "white" : "black";  
 }, [dark]);  
 return <button onClick={() => setDark(d => !d)}>Toggle Dark Mode</button>;  
}

 STATE

1. Counter  
   function SimpleCounter() {  
    const [count, setCount] = React.useState(0);  
    return (  
   <>  
   <p>{count}</p>  
   <button onClick={() => setCount(c => c + 1)}>+</button>  
   <button onClick={() => setCount(c => c - 1)}>-</button>  
   </>  
    );  
   }  
     
   2. Toggle switch  
   function Toggle() {  
    const [on, setOn] = React.useState(false);  
    return <button onClick={() => setOn(o => !o)}>{on ? "ON" : "OFF"}</button>;  
   }

3. Form with single state object  
function Form() {  
 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. Random quote generator  
function 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. Todo list  
function 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. Uppercase input  
function UppercaseInput() {  
 const [text, setText] = React.useState("");  
 return (  
<>  
<input value={text} onChange={e => setText(e.target.value.toUpperCase())} />  
<p>{text}</p>  
</>  
 );  
}  
  
7. Like button  
function LikeButton() {  
 const [likes, setLikes] = React.useState(0);  
 return <button onClick={() => setLikes(l => l + 1)}>{likes}</button>;  
}

8. Color picker  
function ColorPicker() {  
 const [color, setColor] = React.useState("#ffffff");  
 React.useEffect(() => { document.body.style.backgroundColor = color; }, [color]);  
 return <input type="color" value={color} onChange={e => setColor(e.target.value)} />;  
}  
  
 9. Image carousel  
function ImageCarousel() {  
 const images = ["img1.jpg", "img2.jpg", "img3.jpg"];  
 const [index, setIndex] = React.useState(0);  
 return (  
<>  
<img src={images[index]} width="200" alt="carousel" />  
<button onClick={() => setIndex((index - 1 + images.length) % images.length)}>Prev</button>  
<button onClick={() => setIndex((index + 1) % images.length)}>Next</button>  
</>  
 );  
}  
  
 10. Character counter  
function CharCounter() {  
 const [text, setText] = React.useState("");  
 return (  
<>  
<textarea value={text} onChange={e => setText(e.target.value)} />  
<p>Characters: {text.length}</p>  
</>  
 );  
}