WEEK 8

Challenge Yourself

reactapp > src > Components > CropList.jsx > ...

```jsx
// WEEK 8 - CHALLENGE YOURSELF - Components > CropList.jsx

import React from 'react';

export default function List({crops}) {
    return (
        <ul>
            {crops.map(n => ( <li key={n}> {n} </li> ))}
        </ul>
    );
}

```

reactapp > src > Components > CropDashboard.jsx > Dashboard

```jsx
// WEEK 8 - CHALLENGE YOURSELF - Components > CropDashboard.jsx

import React from 'react';
import List from './CropList';

export default function Dashboard({c,s,setS}) {
    const f = c.filter(cr => cr.toLowerCase().includes(s.toLowerCase()));
    return(
        <div>
            <h1>Crop Management System</h1>
            <h4>Search by Crop Name:</h4>
            <input placeholder='Search crops...' value={s} onChange={e => setS(e.target.value)} />
            <List crops={f}/>
        </div>
    );
}
```

reactapp > src > App.js > ...

```jsx
// WEEK 8 - CHALLENGE YOURSELF - App.js

import React, {useState} from 'react';
import Dashboard from './Components/CropDashboard';

const crops = [
  "Tomato", "Apple", "Banana", "Carrot", "Wheat"
];

export default function App() {
  const [s,setS] = useState("");
  return(<Dashboard c={crops} s={s} setS={setS}/>);
}

```

Fix the Code

App.js | App.css | Stopwatch.jsx M

reactapp > src > App.js > App

```
import './App.css';
import StopWatch from './components/Stopwatch';

function App() {
  return (
    <div className="App">
      <StopWatch/>
    </div>
  );
}

export default App;

```

reactapp > src > components > Stopwatch.jsx > ...

```
// WEEK 8 - FIX THE CODE - Stopwatch.js
import React, {useState, useRef} from 'react';

function StopWatch() {
    const [t, setT] = useState(0);
    const [r, setR] = useState(false);
    const [l, setL] = useState([]);
    const intRef = useRef(null);

    const FT = (ms) => {
        const total = Math.floor(ms/1000);
        const m = String(Math.floor(total/60)).padStart(2,'0');
        const s = String(Math.floor(total%60)).padStart(2,'0');
        const cs = String(Math.floor((ms%1000)/10)).padStart(2,'0');
        return `${m}:${s}.${cs}`;
    }
    const start = () => { if(!r) { setR(true); intRef.current = setInterval(()=> {setT((ti) => ti+10)},10); }};
    const stop = () =>  { if(r)  { clearInterval(intRef.current); setR(false); }};
    const reset = () => { clearInterval(intRef.current); setR(false); setT(0); setL([]); };
    const addLap = () => { if(r) { setL((p) => [...p,t]); }};

    return (
        <div>
            <h1>Stopwatch</h1>
            <div>{FT(t)}</div>
            <button onClick={start} disabled={r}>Start</button>
            <button onClick={stop} disabled={!r}>Stop</button>
            <button onClick={reset}>Reset</button>
            <button onClick={addLap} disabled={!r}>Add Lap</button>
            <ul>
                {l.map((lap, ind) => (<li key={ind}>{FT(lap)}</li> ))}
            </ul>
        </div>
    );
}
export default StopWatch;
```

Ln 1, Col 40

Practice At Home

reactapp > src > JS App.js > ...

```
//WEEK 8 - PRACTICE AT HOME - App.js

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

function App() {
  const [l, setL] = useState(true);
  const [t, setT] = useState([
    {id:1, t:'Complete React Assignment', p:'High', c: false},
    {id:2, t:'Review Bootstrap Documentation', p:'Medium', c: true},
    {id:3, t:'Debug Lifecycle Methods', p:'High', c: false},
    {id:4, t:'Setup React Dev Tools', p:'Low', c: false}
  ]);
  const [f, setF] = useState('All');
  const [nt, setNT] = useState('');
  const [np, setNP] = useState('High');
  const [sm, setSM] = useState(false);

  useEffect(() => {
    const task = setTimeout(() => setL(false),2000);
    return () => clearTimeout(task);
  }, []);

  if(l) return <div className='spinner-border'></div>;

  let fd = t;
  if(f!=='All') {
    if(f==='Completed') fd = t.filter(task => task.c);
    else fd = t.filter(task => task.p === f);
  }

  const addTask = () => {
    if(!nt) return;
    setT([...t,{id: Date.now(), t: nt, p: np, c: false}]);
    setNT('');
    setNP('High');
    setSM(false);
  };

  const deleteTask = id => setT(t.filter(task => task.id !== id));
  const toggleComplete = id => setT(t.map(task => (task.id === id ? {...task, c: !task.c}: t)));

```

```
41
42  return (
43    <div>
44    <h1>Task Manager</h1>
45    <nav className="navbar">Navbar</nav>
46    <div>
47        {['All', 'High', 'Medium', 'Low', 'Completed'].map(fr => (
48            <button key={fr} onClick={() => setF(fr)}>{fr}</button>
49        ))}
50    </div>
51    <button onClick={() => setSM(true)}>Add New Task</button>
52    {sm && (
53        <div>
54            <input placeholder="Enter task title..." value={nt} onChange={ev => setNT(ev.target.value)}/>
55            <label>Priority
56                <select value={np} onChange={ev => setNP(ev.target.value)}>
57                    <option>High</option>
58                    <option>Medium</option>
59                    <option>Low</option>
60                </select>
61                </label>
62            <button onClick={addTask}>Add Task</button>
63        </div>
64    )}
65
66    <ul>
67        {fd.map(task => (
68            <li key={task.id}>
69                <input type="checkbox" aria-label={task.t} checked={task.c} onChange={() => toggleComplete(task.id)}/>
70                {task.t}
71                <span className={`badge ${task.p==='High'?'bg-primary':task.p==='Medium'?'bg-warning':'bg-secondary'}`}>{task.p}</span>
72                <button onClick={() => deleteTask(task.id)}>Delete</button>
73            </li>
74        ))}
75    </ul>
76  </div>
77    );
78  }
79
80  export default App;
81
```

Type the Code

reactapp > src > JS App.js > ...

```
1   // WEEK 8 - TYPE THE CODE - App.js
2
3   import React, {useState} from 'react';
4
5   const e = [
6     "John Doe", "Jane Smith", "Mike Johnson", "James Brown"
7   ];
8
9   function App() {
10    const [s, setS] = useState("");
11    const f = e.filter(em => em.toLowerCase().includes(s.toLowerCase()));
12
13    return (
14      <div>
15        <h1>Employee Directory</h1>
16        <input placeholder="Search employees..." value={s} onChange={ev => setS(ev.target.value)}/>
17        <ul>
18          {f.map(n => (
19            <li key={n}>{n}</li>
20          ))}
21        </ul>
22      </div>
23    );
24  }
25
26  export default App;
27
```