

TYPE THE CODE

```
// src/App.js

import React from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    const employees = [
      'John Doe',
      'Jane Smith',
      'Mike Johnson',
      'Emily Davis',
      'James Brown',
      'Jennifer Wilson',
      'Paul Garcia',
      'Laura Martinez',
    ];
    this.state = {
      query: '',
      employees,
      filteredEmployees: employees,
    };
  }
  componentDidUpdate(prevProps, prevState) {
    if (prevState.query !== this.state.query) {
      const q = this.state.query.toLowerCase();
      const filtered = this.state.employees.filter((n) =>
        n.toLowerCase().includes(q)
      );
      this.setState({ filteredEmployees: filtered });
    }
  }
  handleSearch = (e) => {
    this.setState({ query: e.target.value });
  };
  render() {
    return (
      <div className="container py-5">
        <h1 className="text-center mb-4">Employee Directory</h1>
        <div className="d-flex justify-content-center mb-3">
          <input
            type="text"
            className="form-control w-50"
            placeholder="Search employees..."
            value={this.state.query}
            onChange={this.handleSearch}
          />
        </div>
        <ul className="list-unstyled d-flex flex-column align-items-center gap-2">
          {this.state.filteredEmployees.map((name) => (
            <li
              key={name}
              className="w-50 text-center py-2 border rounded bg-light"
            >
              {name}
            </li>
          ))}
        </ul>
      </div>
    );
  }
}
```

```
export default App;
```

FIX THE CODE

```
// src/components/Stopwatch.jsx

import React from 'react';
export default class Stopwatch extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      elapsedMs: 0,
      isRunning: false,
      laps: [],
    };
    this.intervalId = null;
  }
  componentWillUnmount() {
    if (this.intervalId) clearInterval(this.intervalId);
  }
  // mm:ss.cs where cs = centiseconds
  formatTime = (ms) => {
    const totalCs = Math.floor(ms / 10);
    const minutes = Math.floor(totalCs / 600);
    const seconds = Math.floor((totalCs % 600) / 100);
    const cs = totalCs % 100;
    const mm = String(minutes).padStart(2, '0');
    const ss = String(seconds).padStart(2, '0');
    const cc = String(cs).padStart(2, '0');
    return `${mm}:${ss}.${cc}`;
  };
  start = () => {
    if (this.state.isRunning) return;
    this.setState({ isRunning: true });
    const startAt = performance.now() - this.state.elapsedMs;
    this.intervalId = setInterval(() => {
      const now = performance.now();
      this.setState({ elapsedMs: now - startAt });
    }, 10);
  };
  stop = () => {
    if (!this.state.isRunning) return;
    this.setState({ isRunning: false });
    if (this.intervalId) {
      clearInterval(this.intervalId);
      this.intervalId = null;
    }
  };
  reset = () => {
    this.stop();
    this.setState({ elapsedMs: 0, laps: [] });
  };
  addLap = () => {
    if (!this.state.isRunning) return;
    this.setState((prev) => ({ laps: [...prev.laps, prev.elapsedMs] }));
  };
  render() {
    const { elapsedMs, isRunning, laps } = this.state;
    return (
      <div className="d-flex justify-content-center align-items-start py-5">
        <div className="border rounded p-4 text-center" style={{ width: 320 }}>
          <h3 className="mb-3">Stopwatch</h3>
          <div className="display-6 mb-3" aria-label="time">
```

```

        {this.formatTime(elapsedMs)}
    </div>
    <div className="d-flex gap-2 justify-content-center mb-3">
        <button
            className="btn btn-success"
            onClick={this.start}
            disabled={isRunning}
        >
            Start
        </button>
        <button
            className="btn btn-danger"
            onClick={this.stop}
            disabled={!isRunning}
        >
            Stop
        </button>
        <button
            className="btn btn-info text-white"
            onClick={this.reset}
            disabled={isRunning}
        >
            Reset
        </button>
    </div>
    <button
        className="btn btn-warning fw-semibold mb-3"
        onClick={this.addLap}
        disabled={!isRunning}
    >
        Add Lap
    </button>
    <ul className="list-unstyled m-0">
        {laps.map((ms, idx) => (
            <li key={idx} className="text-muted small">
                {this.formatTime(ms)}
            </li>
        )))
    </ul>
</div>
</div>
);
}
}

// src/App.js

import React from 'react';
import Stopwatch from './components/Stopwatch';
export default function App() {
    return <Stopwatch />;
}

```

CHALLENGE YOURSELF

```

// src/App.js

import React from 'react';
import CropDashboard from './Components/CropDashboard';
import './App.css';
const initialCrops = [
    { name: 'Tomato', type: 'Vegetables', status: 'Growing' },
    { name: 'Apple', type: 'Fruits', status: 'Planted' },

```

```

    { name: 'Wheat', type: 'Grains', status: 'Harvested' },
    { name: 'Carrot', type: 'Vegetables', status: 'Growing' },
    { name: 'Banana', type: 'Fruits', status: 'Growing' },
];
export default function App() {
  return <CropDashboard crops={initialCrops} />;
}

// src/Components/CropDashboard.jsx

import React, { useState, useMemo } from 'react';
import CropList from './CropList';
export default function CropDashboard({ crops = [] }) {
  const [query, setQuery] = useState('');
  const filtered = useMemo(() => {
    const q = query.trim().toLowerCase();
    if (!q) return crops;
    return crops.filter((c) => c.name.toLowerCase().includes(q));
  }, [query, crops]);
  return (
    <div className="container py-5 d-flex justify-content-center">
      <div className="card shadow-sm p-4" style={{ maxWidth: 720, width: '100%' }}>
        <h2 className="fw-bold text-uppercase mb-2">Crop Management System</h2>
        <hr className="border-2 border-success opacity-100 mt-0" />
        <div className="mb-3">
          <label className="form-label fw-semibold">Search by Crop Name</label>
          <input
            type="text"
            className="form-control border-success"
            placeholder="Search crops..."
            value={query}
            onChange={(e) => setQuery(e.target.value)}
          />
        </div>
        <CropList crops={filtered} />
      </div>
    </div>
  );
}

// src/Components/CropList.jsx

import React from 'react';
export default function CropList({ crops = [] }) {
  if (crops.length === 0) {
    return <div className="text-muted">No crops found.</div>;
  }
  return (
    <ul className="list-unstyled m-0">
      {crops.map((c) => (
        <li key={c.name} className="d-flex justify-content-between py-2 border-bottom">
          <div className="d-flex align-items-center">
            <span className="text-success me-2" aria-hidden="true">{'{'}</span>
            <span className="fw-semibold">{c.name}</span>
          </div>
          <div className="text-muted small">
            - Type: {c.type} - Status: {c.status}
          </div>
        </li>
      )))
    </ul>
  );
}

```

}