

Week -9

1.Skill builder

Welcome.jsx

```
import React from 'react';
import './Welcome.css';
function Welcome({ name }) {
  return (
    <div className='welcome'>
      <h2>{name}</h2>
    </div>
  );
}
export default Welcome;
```

Welcome.css

```
.welcome h2 {
  margin: 5px 0;
  color: #333;
}
```

App.js

```
import './App.css';
import Welcome from './components/Welcome';
function App() {
  return (
    <div className="App">
      <h1>Neo Players</h1>
      <h3>Top Students List</h3>
      <Welcome name="Sara" />
      <Welcome name="Chahal" />
      <Welcome name="Phillips" />
    </div>
  );
}
export default App;
```

```
</div>
);
}

export default App;
```

2.Challenge yourself

LifecycleLogs.js

```
import React from 'react';

const LifecycleLogs = ({ logs, CL }) => {
  return (
    <div style={{ marginTop: '30px' }}>
      <h2>Lifecycle Logs</h2>
      <button onClick={CL}>Clear Log</button>
      <div
        style={{
          border: '1px solid #ccc',
          marginTop: '10px',
          padding: '10px',
          minHeight: '100px',
        }}
      >
        {logs.length === 0 ? (
          <p>No logs yet</p>
        ) : (
          logs.map((l, ind) => <p key={ind}>{l}</p>)
        )}
      </div>
    </div>
  );
}
```

```
export default LifecycleLogs;
```

CityFilter.js

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

const CityFilter = ({ currentFilter, onFilterChange, addLog, renderCount }) => {

useEffect(() => {
  addLog('CityFilter Mounted');
}, [addLog]);

return (
<div style={{ marginBottom: '20px' }}>
<h2>Filter by Condition</h2>
<select
  role="combobox"
  value={currentFilter}
  onChange={(e) => onFilterChange(e.target.value)}
>
<option value="all">All Conditions</option>
<option value="Sunny">Sunny</option>
<option value="Cloudy">Cloudy</option>
<option value="Rainy">Rainy</option>
<option value="Snowy">Snowy</option>
</select>
<p>Render Count: {renderCount}</p>
</div>
);
};

export default CityFilter;
```

WeatherCard.js

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

const WeatherCard = ({ city, onToggle, addLog }) => {
  useEffect(() => {
    addLog(`WeatherCard Mounted: ${city.name}`)
  }, [addLog, city]);
  return (
    <div style={{
      border: '1px solid #ddd',
      marginBottom: '10px',
      padding: '10px',
      borderRadius: '8px',
      backgroundColor: '#f9f9f9'
    }}>
      <h3>{city.name}</h3>
      <p>{city.temp}°C</p>
      <span style={{
        background: '#ddd',
        padding: '4px 8px',
        borderRadius: '6px'
      }}>
        {city.condition}
      </span>
      <div style={{ marginTop: '10px' }}>
        <button onClick={() => onToggle(city.id)}>
          {city.active === true ? 'Remove City' : 'Add City'}
        </button>
      </div>
      <div>
        <p>{city.condition}</p>
      
```

```

<p>{city.condition}</p>
</div>
</div>
);
};

export default WeatherCard;

```

App.js

```

import React, { useState, useEffect, useCallback } from 'react';
import CityFilter from './components/CityFilter';
import WeatherCard from './components/WeatherCard';
import LifecycleLogs from './components/LifecycleLogs';
function App() {
  const [c, setC] = useState([
    { id: 1, name: 'New York', condition: 'Sunny', temp: 22, active: true },
    { id: 2, name: 'London', condition: 'Cloudy', temp: 15, active: true },
    { id: 3, name: 'Tokyo', condition: 'Rainy', temp: 18, active: true },
    { id: 4, name: 'Paris', condition: 'Snowy', temp: 20, active: true }
  ]);
  const [l, setL] = useState([]);
  const [f, setF] = useState('all');
  const [rc, setRC] = useState(1);
  const AL = useCallback((m) => {
    setL((p) => [...p, `${new Date().toLocaleTimeString()} - ${m}`])
  }, []);
  const FC = (v) => {
    setF(v);
    AL(`Filter changed to: ${v}`);
  };
  const TC = (id) => {
    setC((p) => p.map((ct) => ct.id === id ? { ...ct, active: !ct.active } : ct));
  };
}

```

```
AL(`City toggled: ${id}`);
};

const CL = () => setL(['Logs cleared']);

useEffect(() => { AL('App Mounted'); }, [AL]);

useEffect(() => { setRC((cnt) => cnt + 1); }, [!]);

const fc = f === 'all' ? c.filter((ct) => ct.active) : c.filter((ct) => ct.active && ct.condition === f);

return (
<div style={{ padding: '20px', fontFamily: 'Arial' }}>
  <h1>Weather Dashboard</h1>
  <CityFilter currentFilter={f} onFilterChange={FC} addLog={AL} renderCount={rc} />
  <h2>Cities</h2>
  {
    fc.length > 0 ? (
      fc.map((ct) => (
        <WeatherCard key={ct.id} city={ct} onToggle={TC} addLog={AL} />
      ))
    ) : (
      <p>No cities to display</p>
    )
  }
  <LifecycleLogs logs={l} CL={CL} />
</div>
);
}

export default App;
```