**⚡ Day 3 – Props Reusability + Lists + Conditional Rendering**

**🎯 GOAL:**

* Render **multiple components using arrays**
* Use .map() and key
* Add **conditional rendering** (show/hide based on condition)
* 🔨 **Build: “Topper List” App** (Desi school vibes 😄)

**🧠 Recap: Props + Map**

You already did this yesterday:

jsx

CopyEdit

devs.map((dev, index) => (

<Card key={index} name={dev.name} skill={dev.skill} />

))

Now let’s go 🔥 level up.

**🔶 Step 1: Create a StudentCard.jsx**

jsx

CopyEdit

function StudentCard({ name, marks }) {

return (

<div style={{ padding: "10px", margin: "10px", border: "1px solid #ccc" }}>

<h3>{name}</h3>

<p>Marks: {marks}</p>

</div>

);

}

export default StudentCard;

**🔶 Step 2: In App.jsx**

jsx

CopyEdit

import StudentCard from "./StudentCard";

function App() {

const students = [

{ name: "Riya", marks: 85 },

{ name: "Aman", marks: 72 },

{ name: "Neha", marks: 91 },

{ name: "Ravi", marks: 60 }

];

const toppers = students.filter(student => student.marks >= 80);

return (

<div>

<h1>🏆 Topper List</h1>

{toppers.length === 0 ? (

<p>No toppers this time 😢</p>

) : (

toppers.map((student, index) => (

<StudentCard

key={index}

name={student.name}

marks={student.marks}

/>

))

)}

</div>

);

}

export default App;

**🔍 What You’re Learning Here:**

| **Concept** | **Example** |
| --- | --- |
| .filter() | Get only students with marks ≥ 80 |
| Conditional Rendering | toppers.length === 0 ? ... : ... |
| .map() inside JSX | Render dynamic cards |
| Props Flow | name, marks passed to StudentCard |

**✅ Mini Challenge:**

Build your own:

jsx

CopyEdit

📋 PlayerList app

→ { name: "Virat", runs: 80 }

→ Only show players with runs ≥ 50

Make a new PlayerCard.jsx  
Use .filter() + .map() logic in App.jsx  
Add a message if no player scored above 50.

**🧠 Smart Tip of the Day:**

Whenever you loop with .map(), always give key={index} (or unique ID)  
Conditional rendering in React = use ? :, &&, or simple if before returning JSX