Got it — no **counter**, no **list rendering**, and now no **form/input** either.

You're aiming for a clean assignment that **only** exercises:

* Functional Components
* Props (as data, method, and children)
* useState with a **number-type**, but **not** in the context of a counter or form

**📝 React Assignment: Light Control Panel**

**Concept**: Simulate a simple light brightness control system using buttons and state — no input, no list, no counter-style example.

**🧩 Assignment Requirements**

**1. App.jsx**

* Renders the LightControl component inside a Panel component (to demonstrate props as children).
* Passes a method to LightControl called logBrightnessChange.

**2. LightControl.jsx**

* Functional component.
* Uses useState to manage a **brightness** state (number, default: 50).
* Has two buttons:
  + **Increase Brightness** (adds 10, max 100)
  + **Decrease Brightness** (subtracts 10, min 0)
* Displays the current brightness level as a percentage.
* Calls the onBrightnessChange prop (a function) every time brightness changes — to demonstrate **props as method**.

**3. Panel.jsx**

* A simple wrapper component.
* Accepts children and renders them inside a styled box.
* Demonstrates **props as children**.

**4. LogButton.jsx**

* A button that logs a predefined message when clicked.
* Accepts a function prop logMessage and calls it when clicked — another **props-as-method** example.

**✅ Concepts Covered**

| **Concept** | **Covered In** |
| --- | --- |
| Functional Components | All components |
| Props | LightControl, Panel, LogButton |
| Props as method | LightControl → onBrightnessChange, LogButton → logMessage |
| Props as children | Panel wraps children |
| useState with number | brightness in LightControl |

**💡 Bonus Ideas**

* Change background color intensity based on brightness level.
* Show an emoji 🌞 or 🌚 depending on brightness level.
* Disable buttons at min/max values (e.g., disable "Increase" if brightness = 100).

Would you like the starter code for this version?