**A small writeup to explain the approach you have taken and why.**

1. **Separation of Concerns**:
   * Two separate components are created: **Loader** and **ContentLoader**.
   * **Loader** is responsible for managing the loading state and conditionally rendering either the loader animation or the loaded content.
   * **ContentLoader** is responsible for rendering the loader animation itself.
2. **Component Structure**:
   * The **Loader** component receives props like **loaded**, **children**, **customstyle**, and **position**.
   * It wraps its content with a **div** that sets **position: relative** to ensure proper positioning of the loader.
   * If **loaded** is false, it renders the **ContentLoader**, otherwise, it renders the **children**.
3. **Reusable ContentLoader**:
   * The **ContentLoader** component is a simple presentation component that receives **extraStyles** and **position** as props.
   * It renders the loader animation inside a wrapper **div** with the specified styles.
4. **Configurability**:
   * The **Loader** component is highly configurable. It allows the user to specify custom styles (**customstyle**) for the loader wrapper.
   * Users can also choose whether the loader should be positioned absolutely or relatively through the **position** prop.
5. **Dynamic Loading State**:
   * The **Loader** component dynamically switches between the loading animation and the loaded content based on the **loaded** prop.
   * When the **loaded** prop is true, the loader animation is replaced by the actual content.