Quiz

In this quiz, we will test what we have learned about strongly-typed function component props with TypeScript

Creating strongly-typed function props

1

Below is a component for a red button:

```
const RedButton = ({ children }) => (
    <button style={{ backgroundColor: "red" }}>{children}</butt
on>
);
```

What type annotation can we use to strongly type the props of the button? (There is more than one correct answer).

use strongly-typed props so that only "blue" or "red" can be passed into the color prop?

```
const Label = ({ children, color }) => (
  <label style={{ color }}>{children}</label>
);
```

We have a button component below that has a size prop to specify the height of the button. How can we allow consumers of the component not to specify the size prop and have it default to "small"?

```
type Props = { size: "small" | "large", children: React.React
Node };
const Button = ({ children, size }: Props) => (
```

4

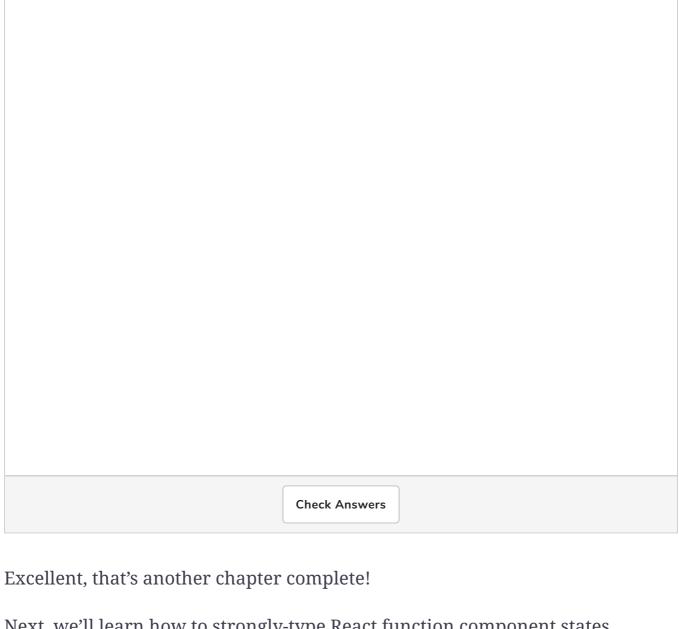
We have a card component below that shows the name and email address of a person:

How can we strongly-type the props for this component?

5

We have a List component that can be consumed as follows:

What would strongly-typed props for this component be?



Next, we'll learn how to strongly-type React function component states.