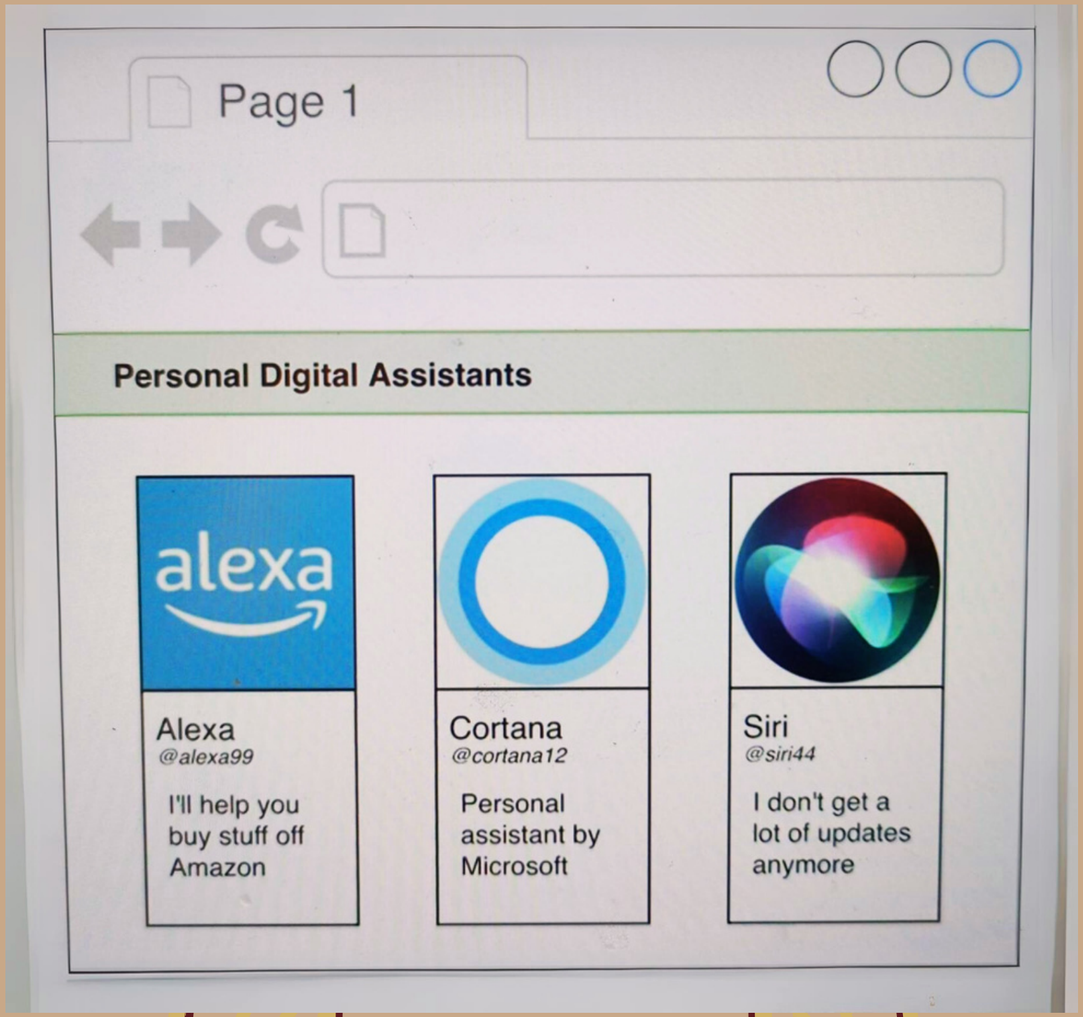
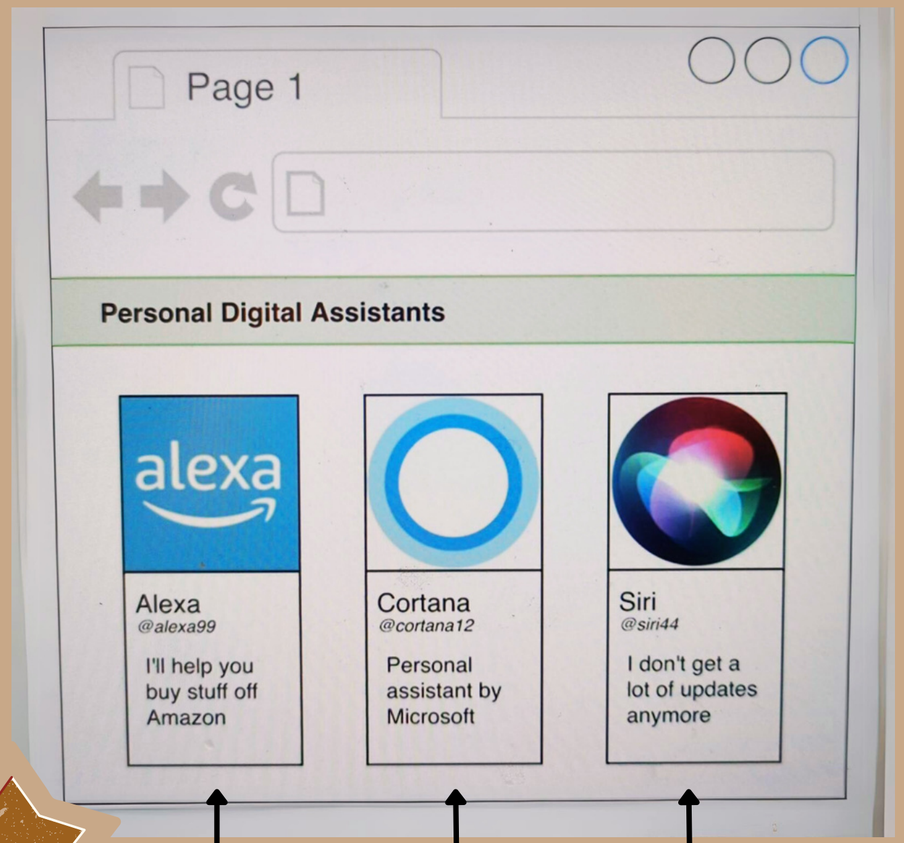
BUILDING WITH REUSABLE COMPONENTS

WeStride

Presented by: Pakkaphong Akkabut


Page 1
Personal Digital Assistants
alexa
Alexa
@alexa99
I'll help you buy stuff off Amazon
Cortana
@cortana12
Personal assistant by Microsoft
Siri
@siri44
I don't get a lot of updates anymore



Each card looks similar
(except for the text and image)

App component

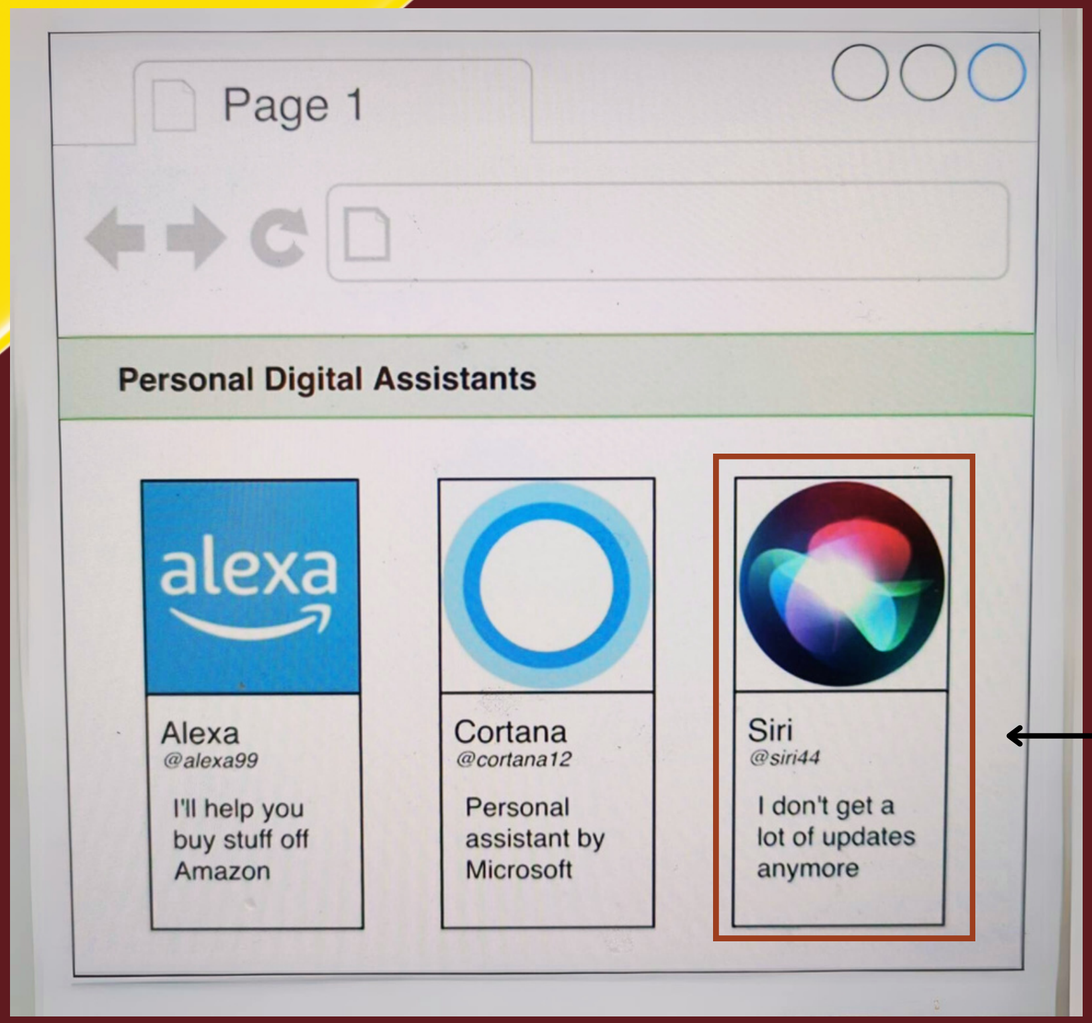
```
function App() {
    return (
      <div>
        <hl>Digital Assistants</h1>

        <div>
            <img />
            <h3>Alexa</h3>
        </div>

        <div>
            <img />
            <h3>Cortana</h3>
        </div>

        <div>
            <img />
            <h3>Siri</h3>
        </div>

      </div>
    );
}
```

Option 1
All JSX written out in the App component
(Probably not good)

JSX to show Alexa Card

JSX to show Cortana Card

JSX to show Siri Card



Option 2
Make a reusable component that shows a single card

ProfileCard Component

```
function ProfileCard() {
    return (
      <div>
        <img />
        <h3>....</h3>
      </div>
    );
}
```

Option 2
Make a reusable component that shows a single card

App component

```
function App() {
    return (
      <div>
        <ProfileCard />
        <ProfileCard />
        <ProfileCard />
      </div>
    );
}
```

ProfileCard Component

```
function ProfileCard() {
    return (
      <div>
        <img />
        <h3>....</h3>
      </div>
    );
}
```

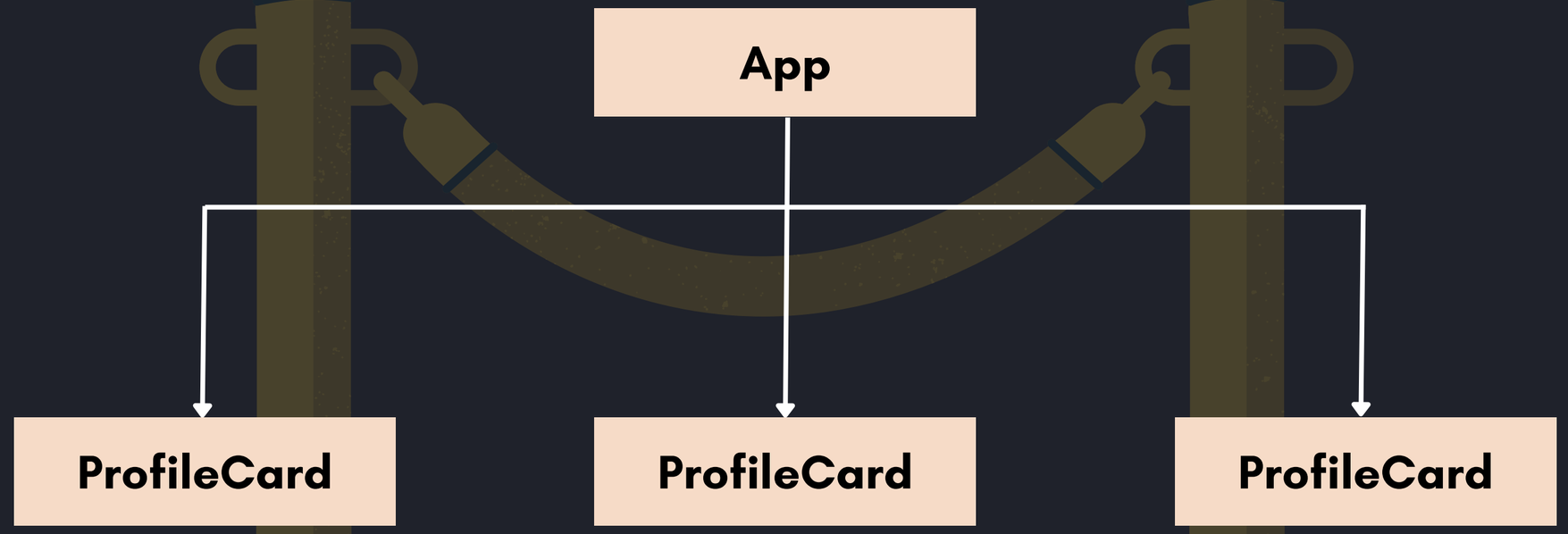
App component

```
function App() {
    return (
      <div>
        <ProfileCard />
        <ProfileCard />
        <ProfileCard />
      </div>
    );
}
```

ProfileCard Component

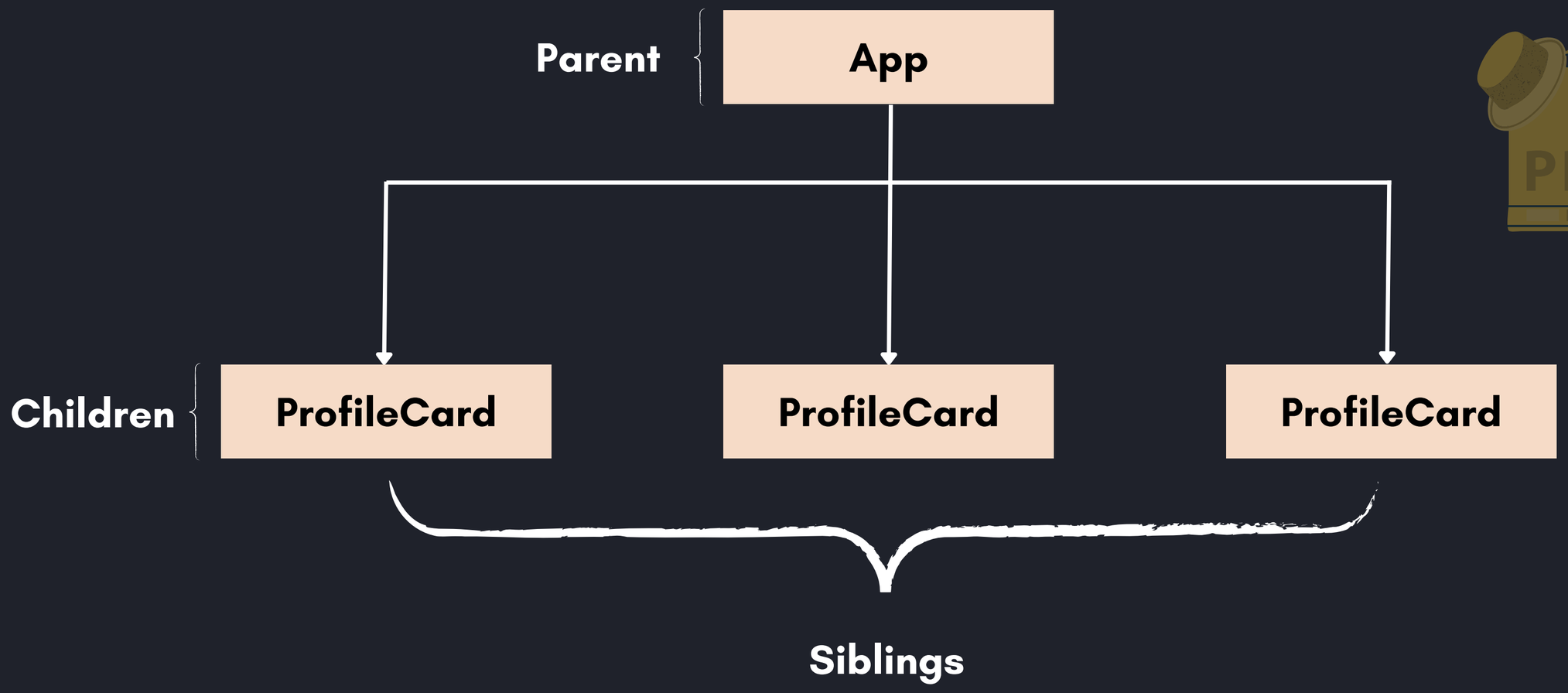
```
function ProfileCard() {
    return (
      <div>
        <img />
        <h3>....</h3>
      </div>
    );
}
```

Challenge - need to figure out how to get each ProfileCard to show different image and text!



Component Hierarchy

Helps us understand the relationship between our components


Parent
App
Children
ProfileCard
ProfileCard
ProfileCard
Siblings

src Folder

ProfileCard.js

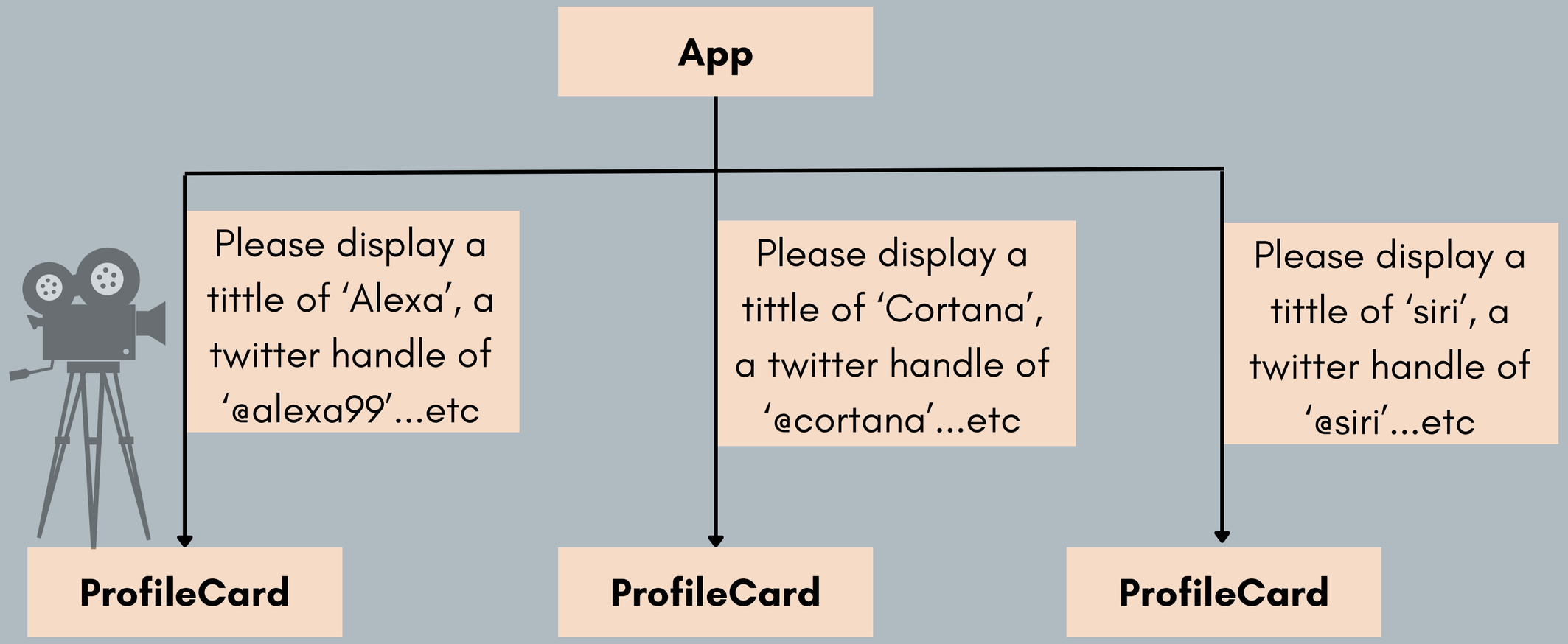
Creates a component that will show a single card

App.js

Creates a component that shows 3 copies of ProfileCard

index.js

Shows the App component on the screen


App
Please display a tittle of 'Alexa', a twitter handle of '@alexa99'...etc
Please display a tittle of 'Cortana', a twitter handle of '@cortana'...etc
Please display a tittle of 'siri', a twitter handle of '@siri'...etc
ProfileCard
ProfileCard
ProfileCard

Parent Component

↓ **props**

Child Component

Props System

- Pass data from a parent to a child
- Allows a parent to configure each child differently
  (show different text, different styles, etc)
- One way flow of data. Child can't push props back up.
- Like 25% of understanding React

Communication with Props

Parent Component

```
function App() {
    return <Child color="red"
}>
```

Props Object

{ color: ‘red’}

Child Component

```
function Child(props) {
    return <div>{props.color}
}/div>
```

1. Add attributes to JSX element
2. React collects the attributes and puts them in an object
3. React collects the attributes and puts them in an object
4. We use the props however we wish

Parent Component

```
function App() {
    return (
      <div>
          <ProfileCard title = "Alexa" handle = @alexa99
          <ProfileCard title = Cortana handle = @cortana99
          <ProfileCard title = Siri handle = @siri01
      </div>
    );
}
```

title	Alexa	Cortana	Siri
handle	@alexa99	@cortana99	@siri01

Communication with Props

1. Add attributes to JSX element
2. React collects the attributes and puts them in an object
3. { title: "...", handle: "..." }
4. We use the props however we wish

Props Object

Key	values
title	Alexa
handle	@alexa99

ProfileCard Component

```
function ProfileCard(props) {
    return (
      <div>
          <div>Title is {props.title}</div>
          <div>Handle is {props.handle}
</div></div>
    );
}
```

Props Object

Key	values
title	Cortana
handle	@cortana99

Pro[illegible] Component

```
function ProfileCard(props) {
    return (
      <div>
          <div>Title is {props.title}</div>
          <div>Handle is {props.handle}
</div></div>
    );
}
```

Props Object

Key	values
title	Siri
handle	@siri01

ProfileCard Component

```
function ProfileCard(props) {
    return (
      <div>
          <div>Title is {props.title}</div>
          <div>Handle is {props.handle}
</div></div>
    );
}
```

Parent Component

```
function App() {
    return (
      <div>
          <ProfileCard                    />
          <ProfileCard                    />
          <ProfileCard                    />
      </div>
    );
}
```

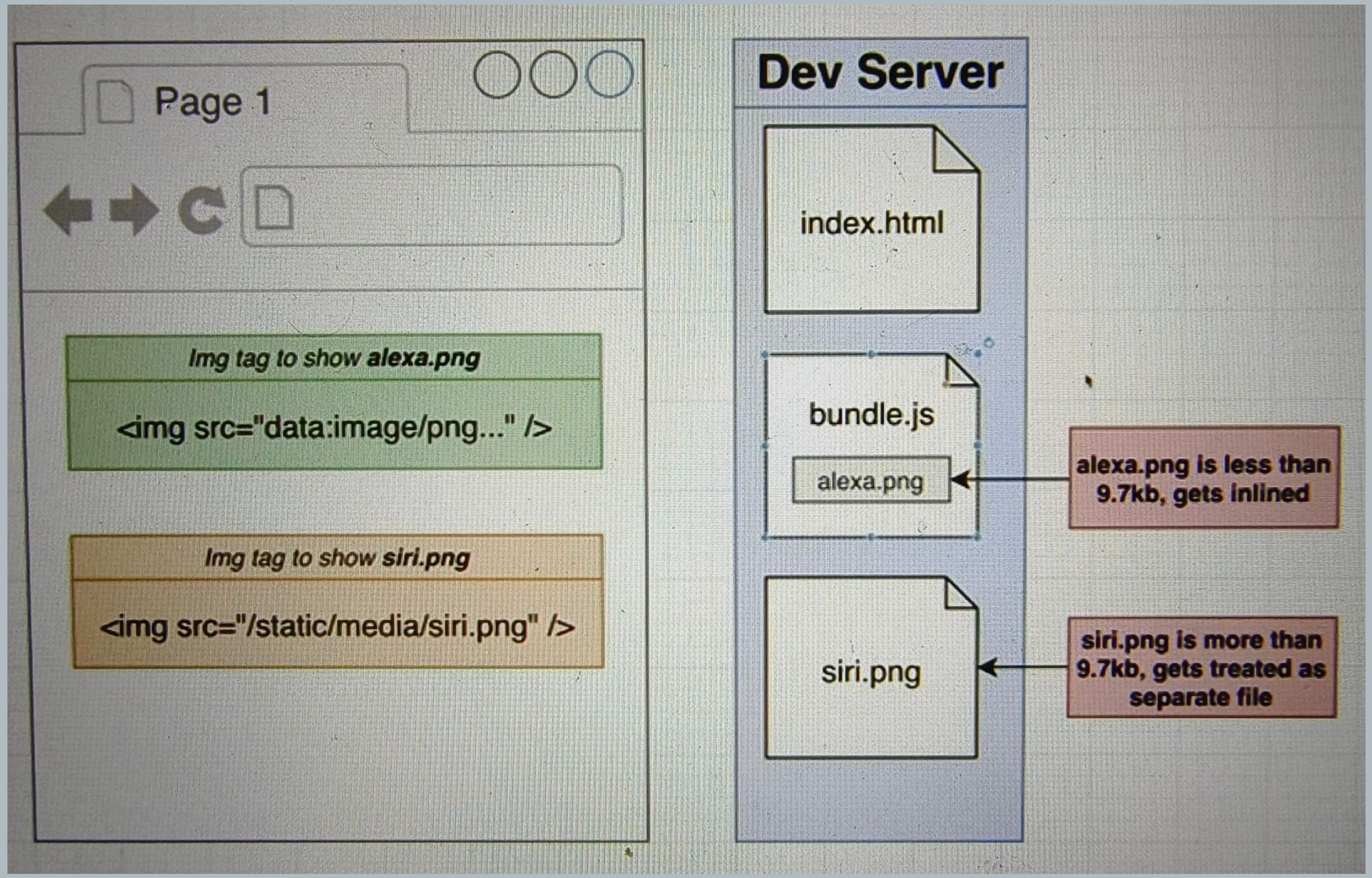
title	Cortana
handle	@cortana
Alexa	@alexa99
Siri	@siri

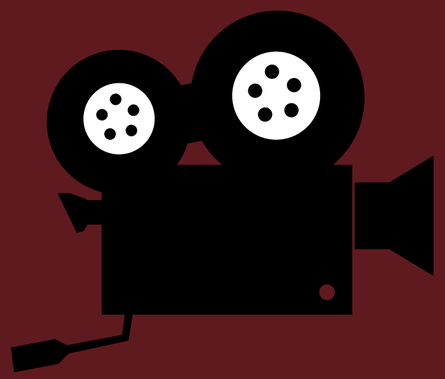
Props Object

Key	values

ProfileCard Component

```
function ProfileCard(props) {
    return (
      <div>
          <div>Title is {props.title}</div>
          <div>Handle is {props.handle}
</div></div>
    );
}
```


Page 1
Img tag to show alexa.png
<img src="data:image/png..." />
Img tag to show siri.png
<img src="/static/media/siri.png" />
Dev Server
index.html
bundle.js
alexa.png
siri.png
alexa.png is less than 9.7kb, gets inlined
siri.png is more than 9.7kb, gets treated as separate file



Are you trying to write custom CSS <u>or</u> use a CSS library?

CSS Library

- Download the CSS file, add to the 'public' folder, add link tag to HTML file
- Download the CSS file, add to src folder, import it.
- Add link tag to CSS file hosted on a CDN
- Install the CSS library using NPM, import the CSS file.

Write Custom CSS

- Write CSS into a CSS file, place in 'public' folder, add link tag to index.html
- Write CSS into a CSS file, place in 'src' folder, import the css file
- Install a CSS-in-JS library. Write CSS in your JS file
- Write css directly into index.html file
- Write SASS, set up CRA to handle your SASS files, import SASS files
- Add style tags to appropriate JSX elements

CSS in Bulma

```
.card {
  background-color: #fff;
  border-radius: 0.25rem;
}
```

HTML Produced by React App

```
<div className="card">
  <div className="card-image">
    ....
  </div>
</div>
```

Find all the elements with a className of 'card' and apply these rules to them

CSS in Bulma

```
.card {
  background-color: #fff;
  border-radius: 0.25rem;
}
```

HTML Produced by React App

```
<div className="card">
  <div className="card-image">
    ....
  </div>
</div>
```

DRAMA

- Personalize your presentation by customizing this slide as much as you want.
- Add a personal touch by customizing this layout as much as you want.

MUSICAL

- Ask questions to interact with your audience.
- You can also add other related visuals to capture the attention of your audience.

FAMOUS THEATRICAL PERFORMANCES

Personalize your presentation by customizing this slide as much as you want.

Performance #1

You can duplicate this slide as much as you want to customize your presentation.

Performance #2

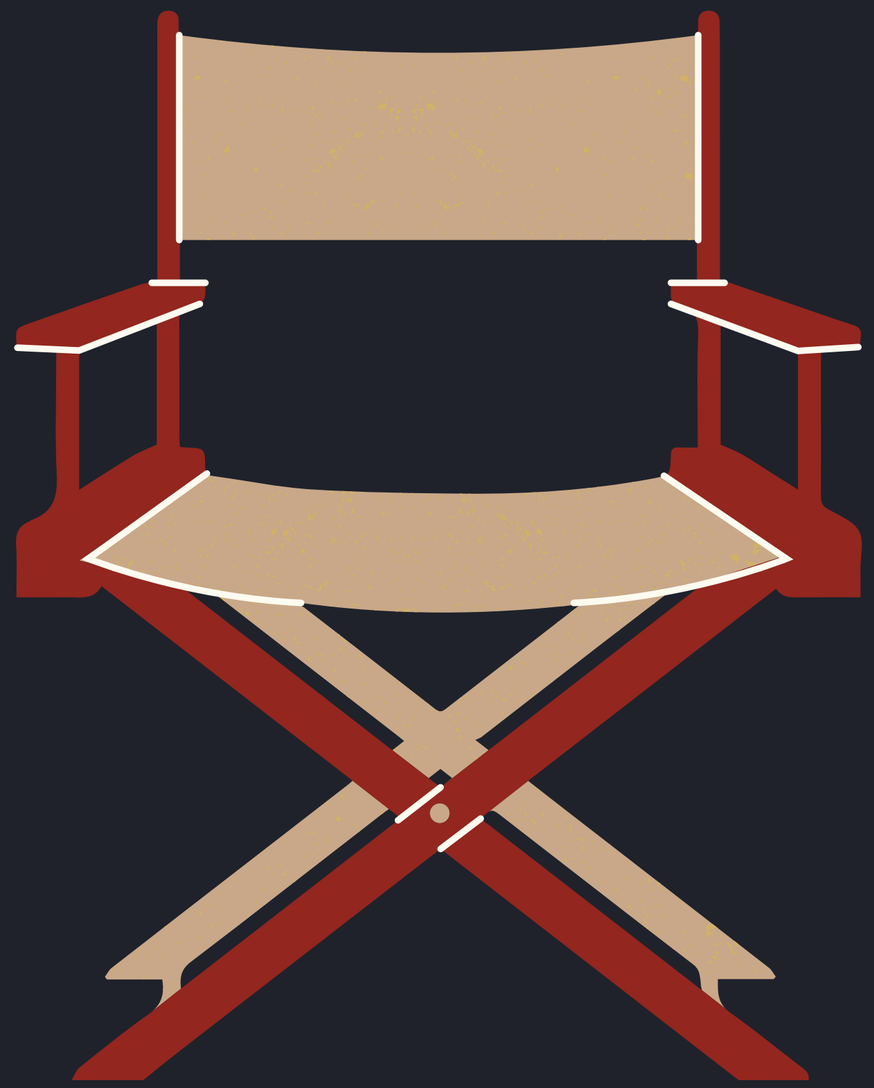
DIRECTING & SCRIPTWRITING

DIRECTING

- Then, ask questions to interact with your audience.
- Add a personal touch by customizing this layout as much as you want.
- You can also add other related visuals to capture the attention of your audience.

SCRIPTWRITING

- Then, ask questions to interact with your audience.
- Add a personal touch by customizing this layout as much as you want.
- You can also add other related visuals to capture the attention of your audience.



CONCLUSION

- Then, ask questions to interact with your audience.
- Add a personal touch by customizing this layout as much as you want.
- You can also add other related visuals to capture the attention of your audience.

THANK YOU FOR LISTENING

Don't hesitate to ask any questions!