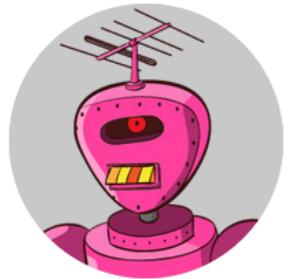


Hands-on Exercise: Building a Profile Viewer

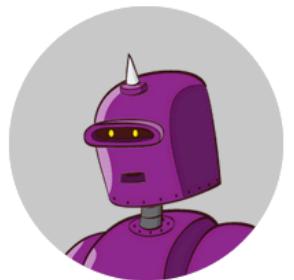
Lek Hsiang Hui

Exercise: Building A Profile Viewer



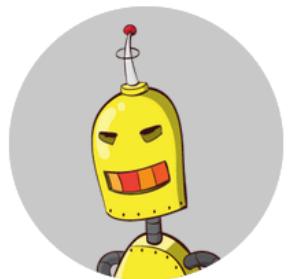
Coletta Spinka
coletta.spinka@email.com
+351 1-215-083-8787 x1674
1982-04-19

Product Design Officer - Teamwork



Jimmie D'Amore
jimmie.d'amore@email.com
+249 (156) 785-4858 x9393
1980-12-21

Healthcare Developer - Self-motivated

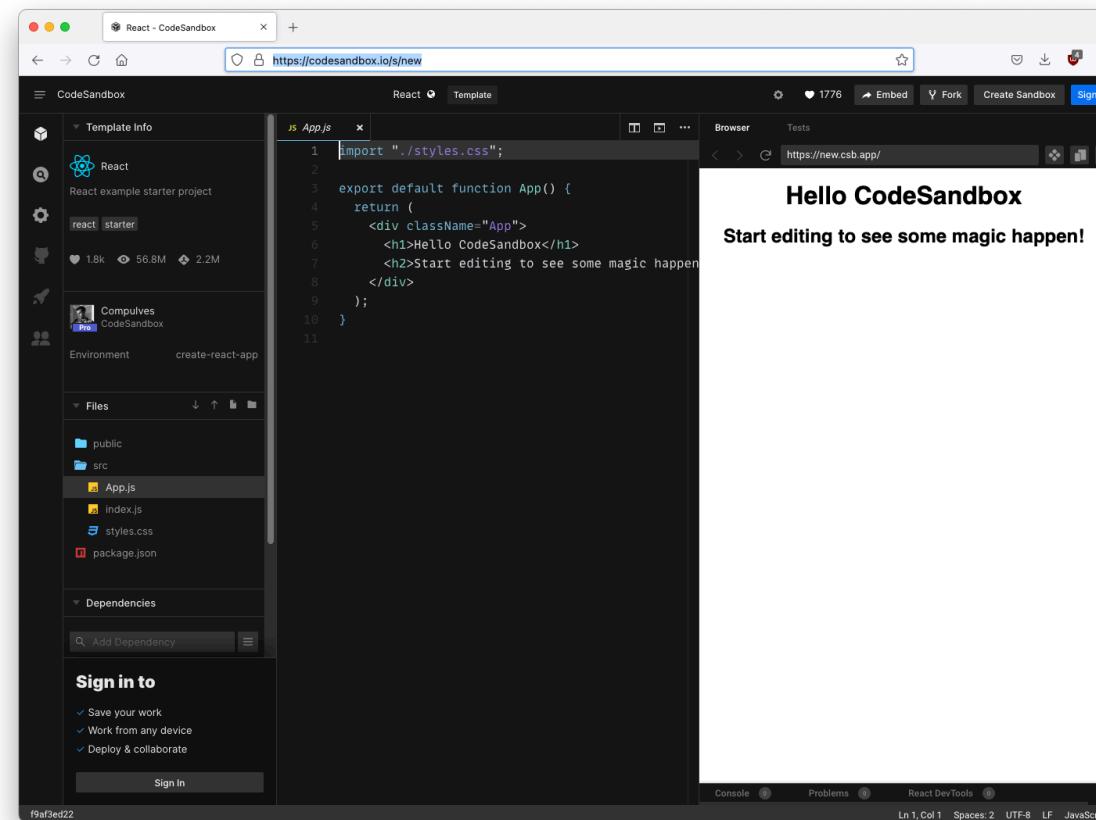


Jackeline Kautzer
jackeline.kautzer@email.com
+506 307.291.8181 x6275
1991-06-05

Product Government Architect - Networking skills

1. Create New React Sandbox

<https://codesandbox.io/s/new>



2. Working With Dummy Data

<https://random-data-api.com/api/v2/users?size=3>

```
//taken from https://random-data-api.com/api/v2/users?size=3
const people = [
  {
    id: 1193,
    first_name: "Coletta",
    last_name: "Spinka",
    email: "coletta.spinka@email.com",
    avatar: "https://robohash.org/incipitqueet.png?size=300x300&set=set1",
    phone_number: "+351 1-215-083-8787 x1674",
    date_of_birth: "1982-04-19",
    employment: { title: "Product Design Officer", key_skill: "Teamwork" }
  },
  ...
]
```

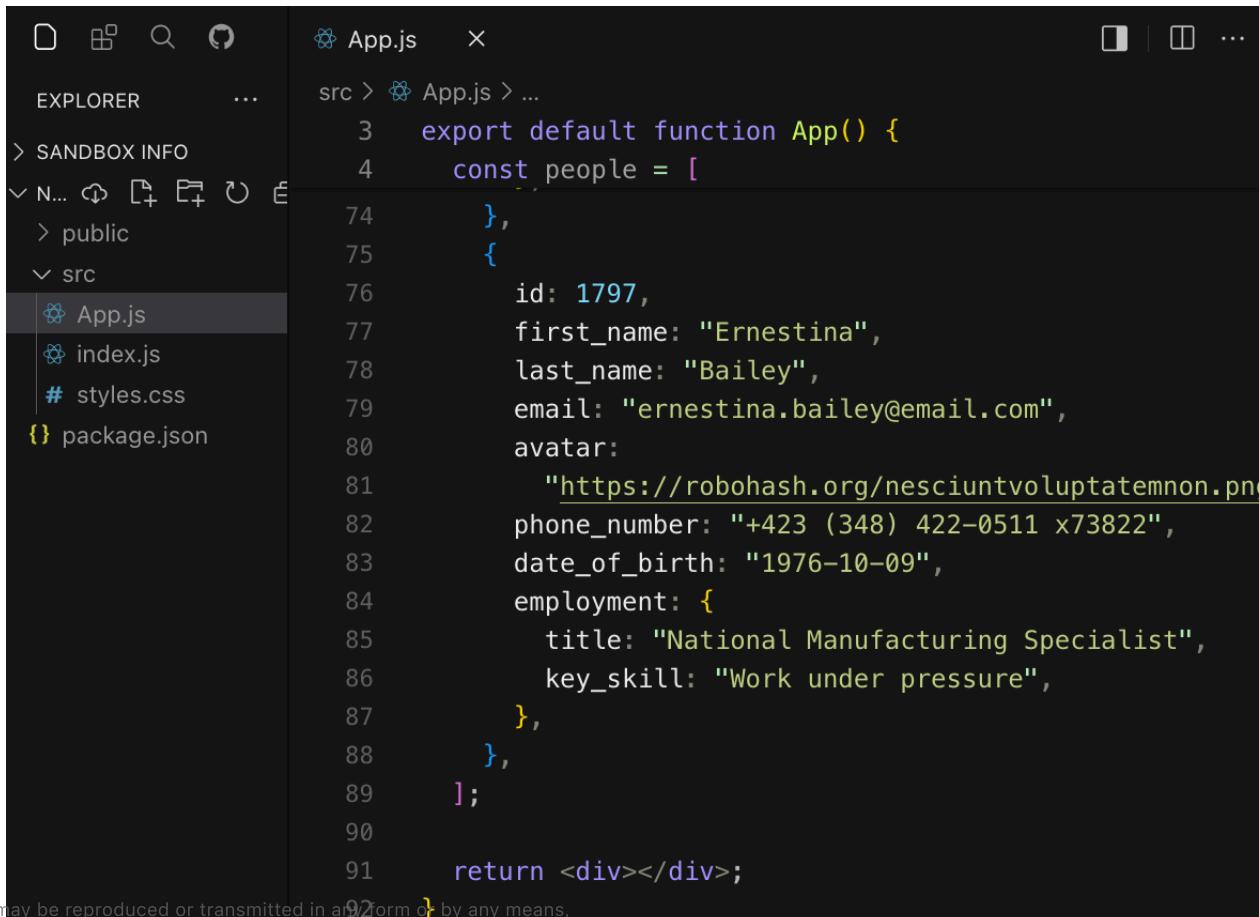
3. Copy Data to App.js

- Copy the **people array** to **src/App.js**
- Remove away other unnecessary codes

```
export default function App() {  
  const people = [ ... ];  
  
  return <div></div>;  
}
```

3. Copy Data to App.js

- It should look something like this



```
App.js
src > App.js > ...
3  export default function App() {
4    const people = [
5      {
6        id: 1797,
7        first_name: "Ernestina",
8        last_name: "Bailey",
9        email: "ernestina.bailey@email.com",
10       avatar:
11         "https://robohash.org/nesciuntvoluptatemnon.png",
12       phone_number: "+423 (348) 422-0511 x73822",
13       date_of_birth: "1976-10-09",
14       employment: {
15         title: "National Manufacturing Specialist",
16         key_skill: "Work under pressure",
17       },
18     },
19   ];
20
21   return <div></div>;
22 }
```

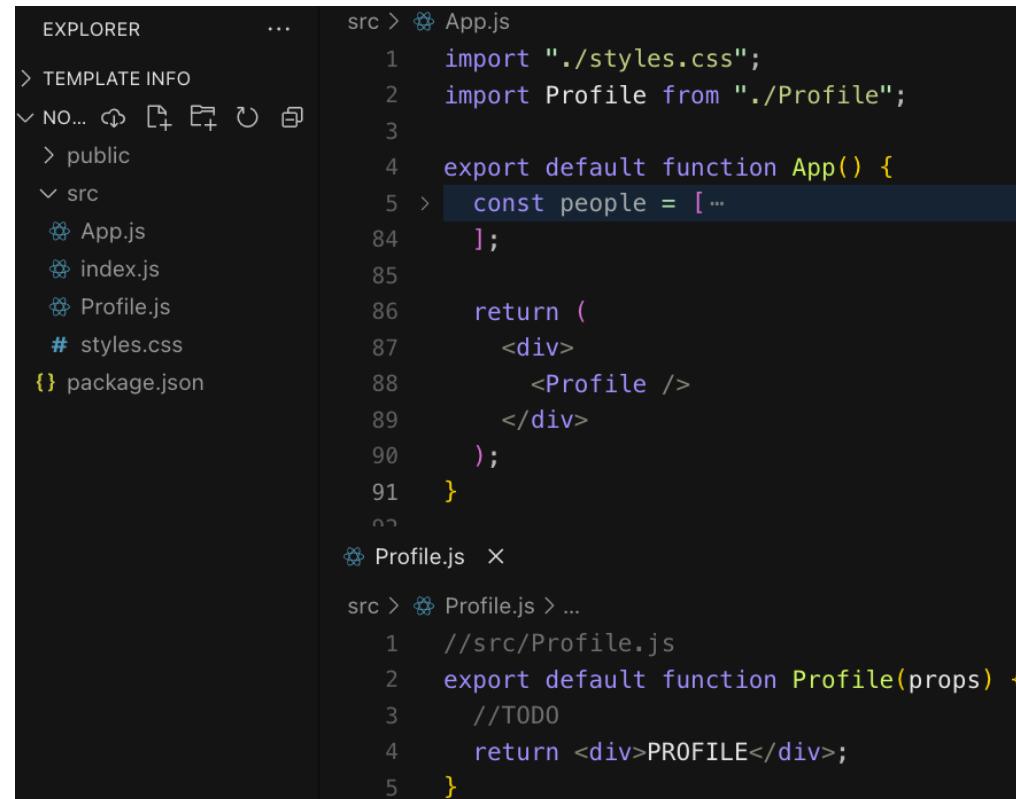
4. Create Profile.js

- Create a new File (**src/Profile.js**)
- Put some placeholder codes

```
//src/Profile.js
export default function Profile(props) {
  //TODO
  return <div>PROFILE</div>;
}
```

5. Use Profile component in App.js

- Import the Profile module
- Use the Profile component



The screenshot shows a code editor with two files open: App.js and Profile.js. The left pane displays the file structure:

- EXPLORER
- TEMPLATE INFO
- NO... ↕ ⌂ ⌂ ⌂ ⌂
- public
- src
 - App.js
 - index.js
 - Profile.js
 - styles.css
- package.json

The right pane shows the content of App.js:

```
src > App.js
1 import "./styles.css";
2 import Profile from "./Profile";
3
4 export default function App() {
5   const people = [
6     {
7       name: "Lek Hsiang Hui",
8       age: 30,
9       job: "Software Engineer"
10    }
11  ];
12
13  return (
14    

15      
16

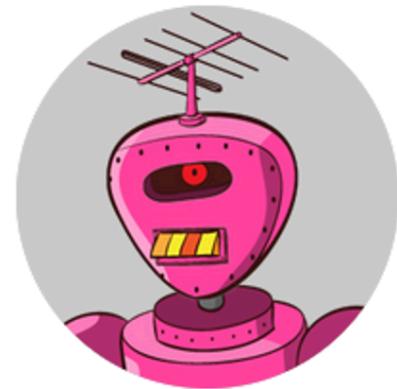

17  );
18}
```

Below App.js, the content of Profile.js is shown:

```
src > Profile.js > ...
1 //src/Profile.js
2 export default function Profile(props) {
3   //TODO
4   return <div>PROFILE</div>;
5 }
```

Profile Component

- The `<Profile />` should take in one user record through the props and display the user profile



Coletta Spinka
coletta.spinka@email.com
+351 1-215-083-8787 x1674
1982-04-19

Product Design Officer - Teamwork

6. Modify App.js to use the `map()` function

- Modify `src/App.js` to use the `map()` function on the `people` array
- Do not forget to include the `key` prop

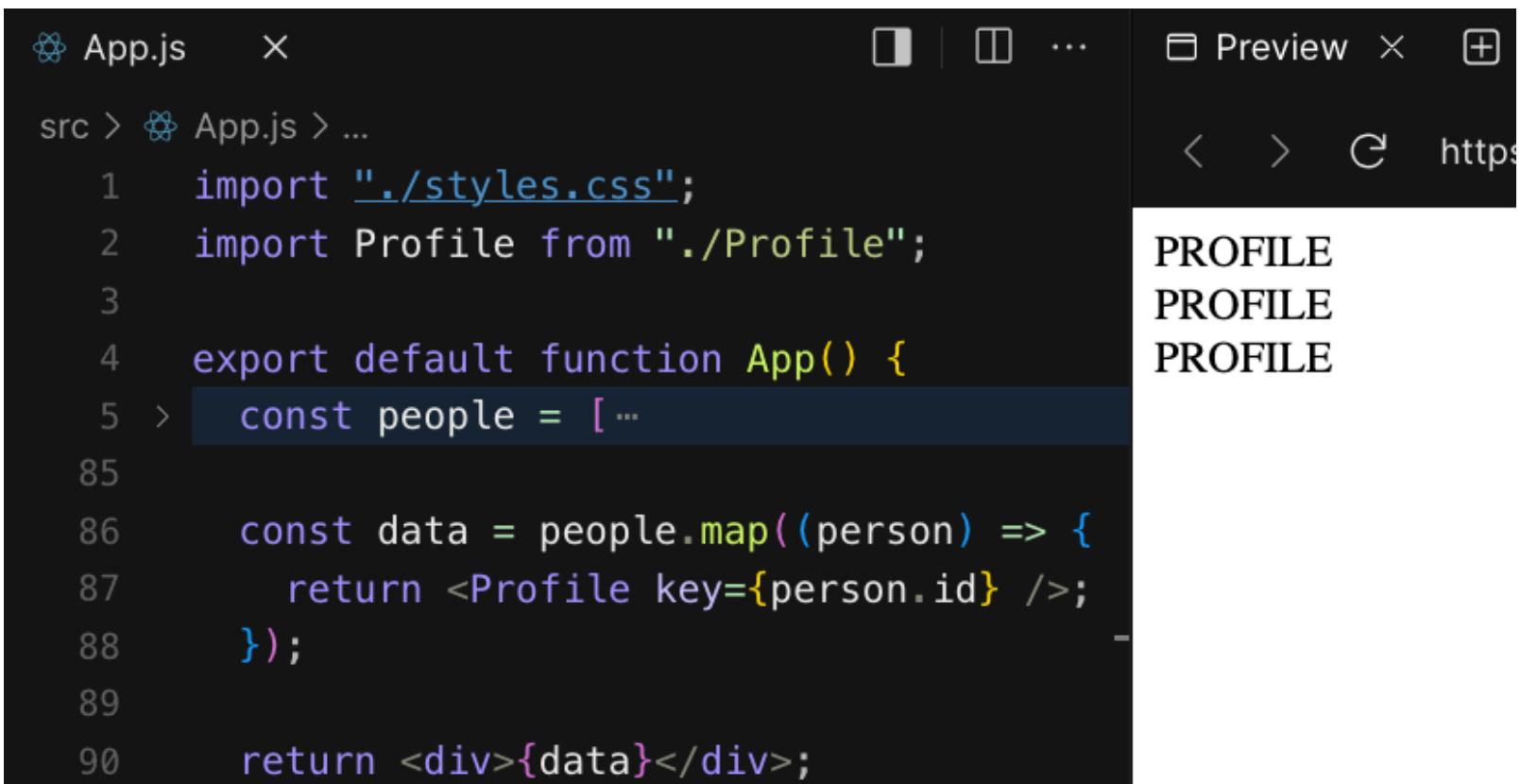
```
...
export default function App() {
  const people = [ ... ];

  const data = people.map((person) => {
    return <Profile key={person.id} />;
  });

  return <div>{data}</div>;
}
```

6. Modify App.js to use the `map()` function

- It should look something like this



The screenshot shows a code editor and a browser preview side-by-side. The code editor on the left displays the file `App.js` with the following content:

```
App.js    ×
src > App.js > ...
1 import "./styles.css";
2 import Profile from "./Profile";
3
4 export default function App() {
5 >   const people = [
85
86     const data = people.map(person) => {
87       return <Profile key={person.id} />;
88     );
89
90   return <div>{data}</div>;
}
```

The browser preview on the right shows three identical "PROFILE" components displayed vertically.

Styling Design Of Profile Component

- This is the styling design of the Profile component
- We will define 4 styles as normal Javascript objects:
 - `containerStyle` , `imageContainerStyle` , `imageStyle` ,
`descriptionContainerStyle`



7. containerStyle

- Now, heading to **Profile.js**, we will define 4 JS objects for each of the style

```
//src/Profile.js
export default function Profile(props) {
  const containerStyle = {
    border: "1px solid black",
    width: "600px",
    height: "150px",
    padding: "10px",
    borderRadius: "5px",
  };
  ...
}
```

7. imageContainerStyle

- Now, heading to **Profile.js**, we will define 4 JS objects for each of the style

```
//src/Profile.js
export default function Profile(props) {
  ...
  const imageContainerStyle = {
    display: "absolute",
    float: "left",
  };
}
```

7. imageStyle

- Now, heading to **Profile.js**, we will define 4 JS objects for each of the style

```
//src/Profile.js
export default function Profile(props) {
  ...
  const imageStyle = {
    width: "150px",
    borderRadius: "50%",
    background: "#ccc",
  };
}
```

7. descriptionContainerStyle

- Now, heading to **Profile.js**, we will define 4 JS objects for each of the style

```
//src/Profile.js
export default function Profile(props) {
  ...
  const descriptionContainerStyle = {
    float: "left",
    width: "400px",
    fontFamily: "Arial, Helvetica, sans-serif",
    padding: "20px 10px",
  };
}
```

7. After Styling to Profile.js

- **src/Profile.js** should look like this

```
//src/Profile.js
export default function Profile(props) {
  const containerStyle = { ... };
  const imageContainerStyle = { ... };
  const imageStyle = { ... };
  const descriptionContainerStyle = { ... };
  ...
}
```

8. Passing Props from App to Profile

- The next step is to spread the properties of `person` to the `<Profile />` component

```
//src/App.js
...
export default function App() {
  const people = [ ... ];

  const data = people.map((person) => {
    return <Profile {...person} key={person.id} />;
  });

  return <div>{data}</div>;
}
```

8. Destructure Props in Profile

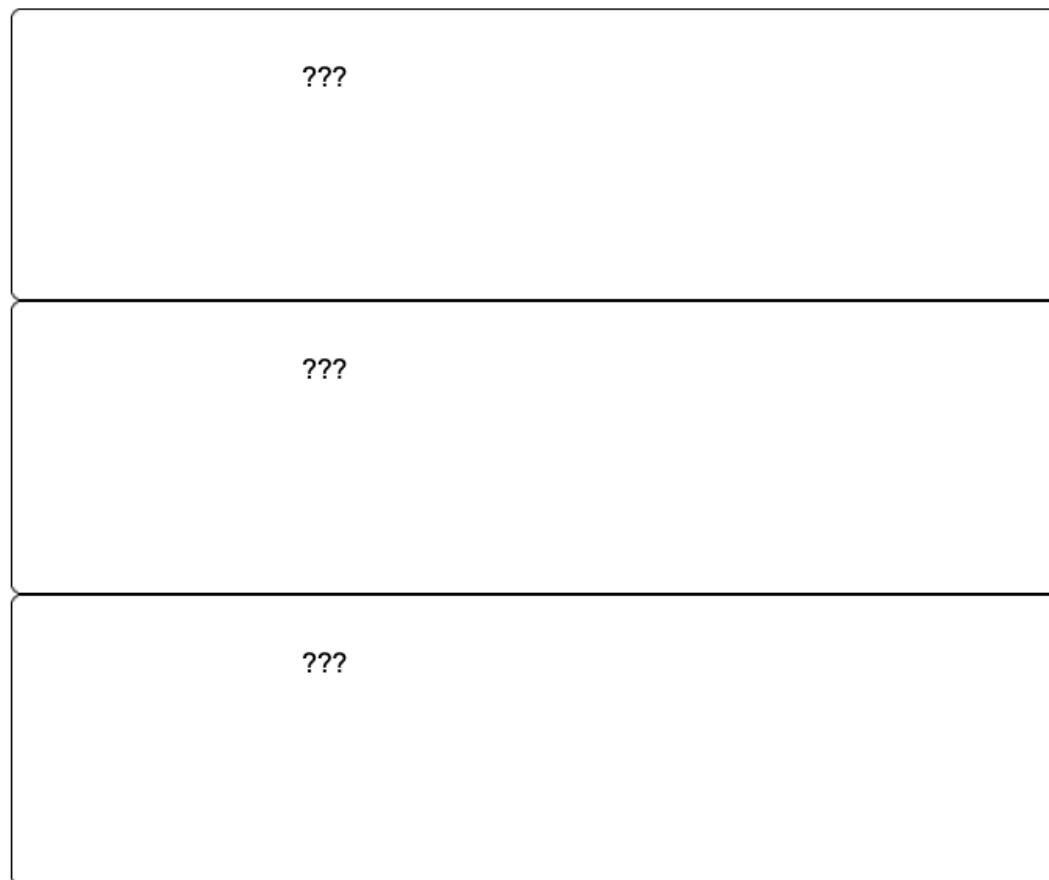
```
//src/Profile.js
export default function Profile(props) {
  ...
  //destructure the individual fields from the props
  const {
    first_name,
    last_name,
    avatar,
    email,
    phone_number,
    date_of_birth,
    employment,
  } = props;
  ...
}
```

9. Incorporate The Styling In Profile

```
//src/Profile.js
export default function Profile(props) {
  ...
  return (
    <div style={containerStyle}>
      <div style={imageContainerStyle}>
        <img style={imageStyle} />
      </div>
      <div style={descriptionContainerStyle}>
        ???
      </div>
    </div>
  );
}
```

Styling Of The Profile Component

- If everything works out, the App should look like this



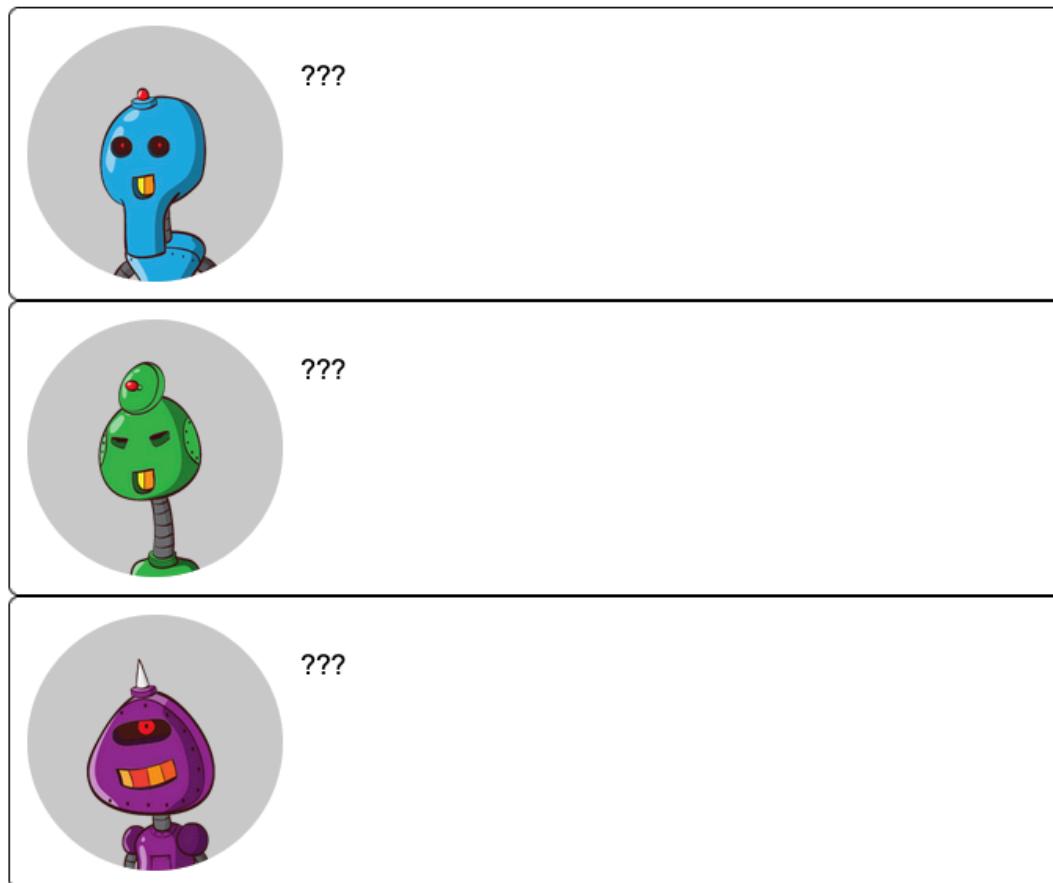
10. Adding Avatar To Profile

- Add `src={avatar}` to ``
- Note that the avatar field has been destructured from props earlier

```
//src/Profile.js
export default function Profile(props) {
  ...
  return (
    <div style={containerStyle}>
      <div style={imageContainerStyle}>
        <img style={imageStyle} src={avatar} />
      </div>
      ...
    </div>
  );
}
```

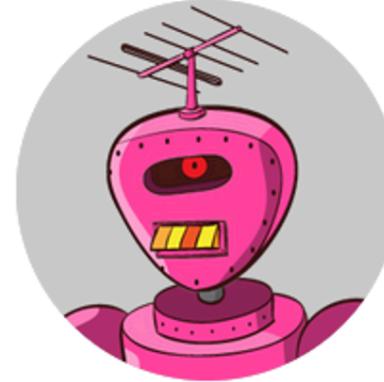
10. Adding Avatar To Profile

- If everything works out, the App should look like this



TODO: Adding The Remaining Fields

- Similar to `avatar`, modify the JSX to complete the remaining information
- *Hint: For the `employment` prop, you need to further destructure out the `title` and `key_skill` fields*



Coletta Spinka
coletta.spinka@email.com
+351 1-215-083-8787 x1674
1982-04-19

Product Design Officer - Teamwork