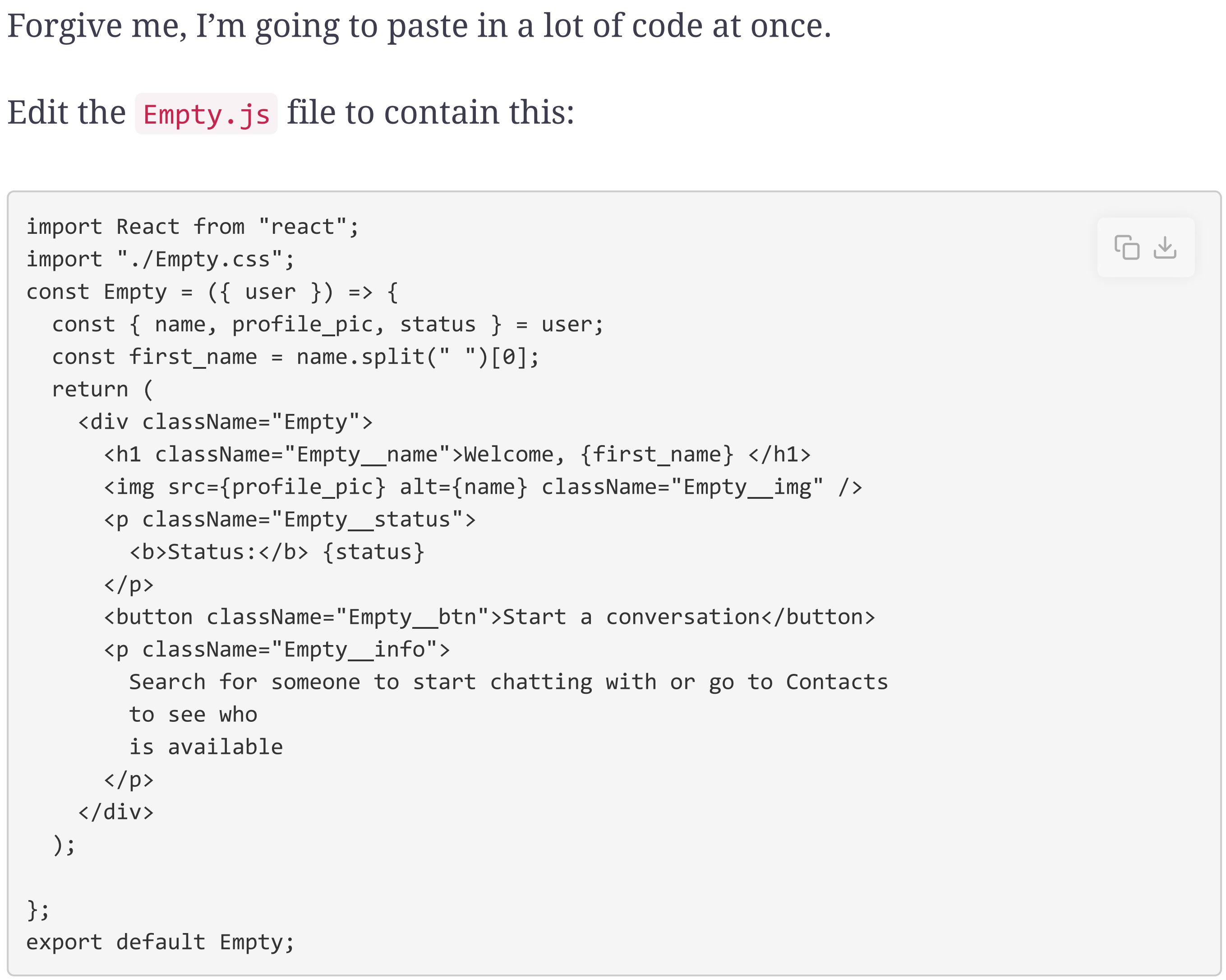
Javascript Implementation

We flesh out the complete functionality of the Empty screen, which shows all the active user attributes. For this we need to edit the Empty.js file.



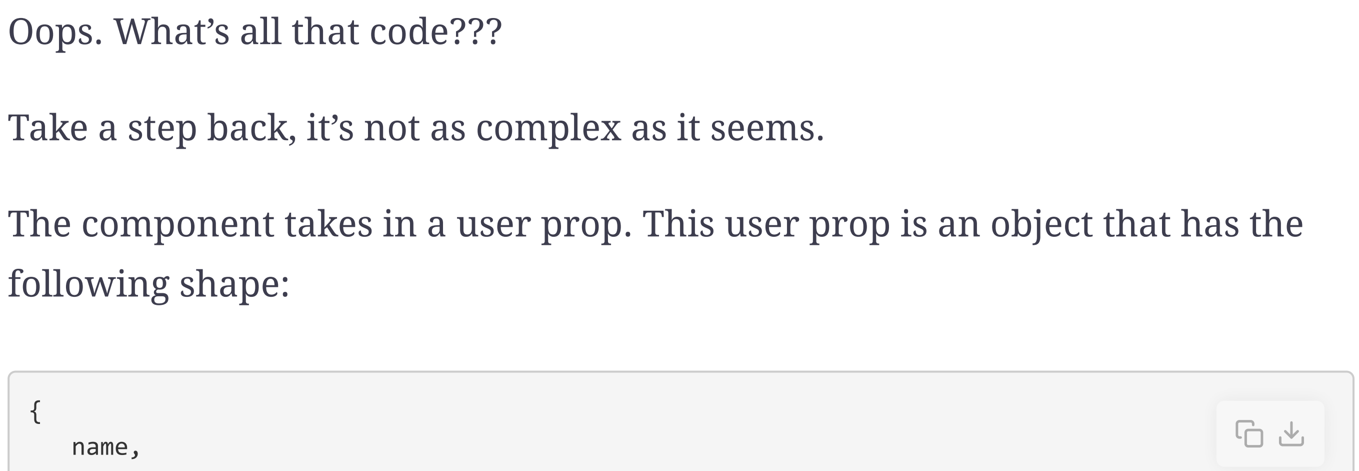
Forgive me, I’m going to paste in a lot of code at once.

Edit the `Empty.js` file to contain this:

```
import React from "react";
import "./Empty.css";
const Empty = ({ user }) => {
  const { name, profile_pic, status } = user;
  const first_name = name.split(" ")[0];
  return (
    <div className="Empty">
      <h1 className="Empty__name">Welcome, {first_name} </h1>
      <img src={profile_pic} alt={name} className="Empty__img" />
      <p className="Empty__status">
        <b>Status:</b> {status}
      </p>
      <button className="Empty__btn">Start a conversation</button>
      <p className="Empty__info">
        Search for someone to start chatting with or go to Contacts
        to see who
        is available
      </p>
    </div>
  );

};
export default Empty;
```

empty.js



Oops. What’s all that code???

Take a step back, it’s not as complex as it seems.

The component takes in a user prop. This user prop is an object that has the following shape:

```
{
   name,
    email,
    profile_pic,
    status,

    user_id:
}
```

Using the ES6 destructuring syntax, grab the `name`, `profile_pic` and `status` from the `user` object:

```
const { name, profile_pic, status } = user;
```

For most users, the name contains two words e.g. Ohans Emmanuel. Grab the first word and assign it to the variable first_name like this:

```
const first_name = name.split(" ")[0];
```

The return statement just spits out a chunk of markup. You'll see the result of this very soon.

Before we go ahead, let's not forget to create a `ChatWindow` component within the containers directory.

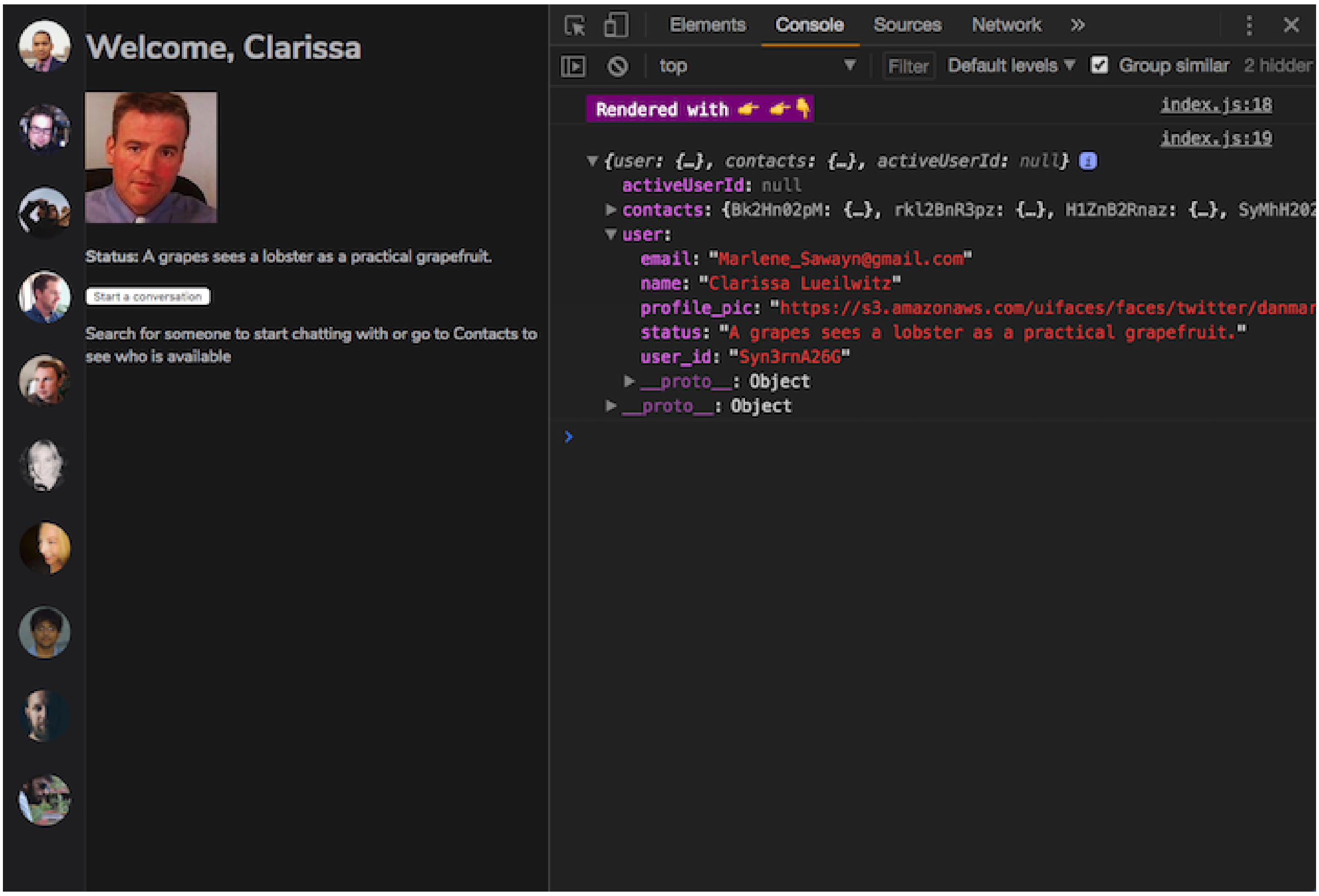
ChatWindow will be responsible for displaying the conversations for an active user contact, and it's going to do a lot of direct talking to Redux!

In ChatWindow.js write the following:

```
import React from "react";
const ChatWindow = ({ activeUserId }) => {
  return (
    <div className="ChatWindow">Conversation for user id:
      {activeUserId}</div>
  );
};
export default ChatWindow;
```

We will come back to flesh this out. Right now, this is good enough.

Save all the changes we've made so far, and here's what I've got!



Welcome, Clarissa

Status: A grapes sees a lobster as a practical grapefruit.

Search for someone to start chatting with or go to Contacts to see who is available

```
Rendered with 👉 👉👇                                      index.js:18
                                                          index.js:19
▼{user: {…}, contacts: {…}, activeUserId: null} ℹ
   activeUserId: null
 ▶ contacts: {Bk2Hn02pM: {…}, rkl2BnR3pz: {…}, H1ZnB2Rnaz: {…}, SyMhH202
 ▼ user:
     email: "Marlene_Sawayn@gmail.com"
     name: "Clarissa Lueilwitz"
     profile_pic: "https://s3.amazonaws.com/uifaces/faces/twitter/danmar
     status: "A grapes sees a lobster as a practical grapefruit."
     user_id: "Syn3rnA26G"
   ▶ __proto__: Object
 ▶ __proto__: Object
›
```

You should have something very similar too.

In the next lesson, we'll refine the empty screen's appearance and finish this component!