## 3 – Creating New Components

### User Component

To display our user information, we’ll need a few new components. Firstly, lets create a new component that’ll act as page to show the user information. If we were doing routing in this exercise, this page would be the component we’d navigate to.

To create a new component, you’ll need to create a new **.js** file. Create a new file called **User.js** in a folder called **User/.** Your folder structure should look something like this:

src/

User/

User.js

Add a simple component to the new file:

import React, { Component } from 'react';

class User extends Component {

render() {

return (<p>user page works!</p>);

}

}

export default User;

### User Detail Component

This component will be what actually displays the user information. It will sit inside the user component.

Repeating what we did previously, create a user detail component in the **User/** folder. Call it **UserDetail.js.** Add a simple component class like before.

## 4 – Nesting Components

Right now, we can’t see either of the components we’ve made; we need to fix that. By default our React app has a structure like this:

index.js > App

What we want is a structure like this:

index.js > App > User > UserDetail

To do this, first remove the default content of App.render() and replace it with our own:

class App extends Component {

render() {

return (

<User />

);

}

}

We can do the same thing with the User component, nesting the UserDetail component:

render() {

return (

<div>

<h1>User</h1>

<UserDetail />

</div>

);

}

Now in the browser you should see the content of the UserDetail component.

