Exercise 1: Bank Application

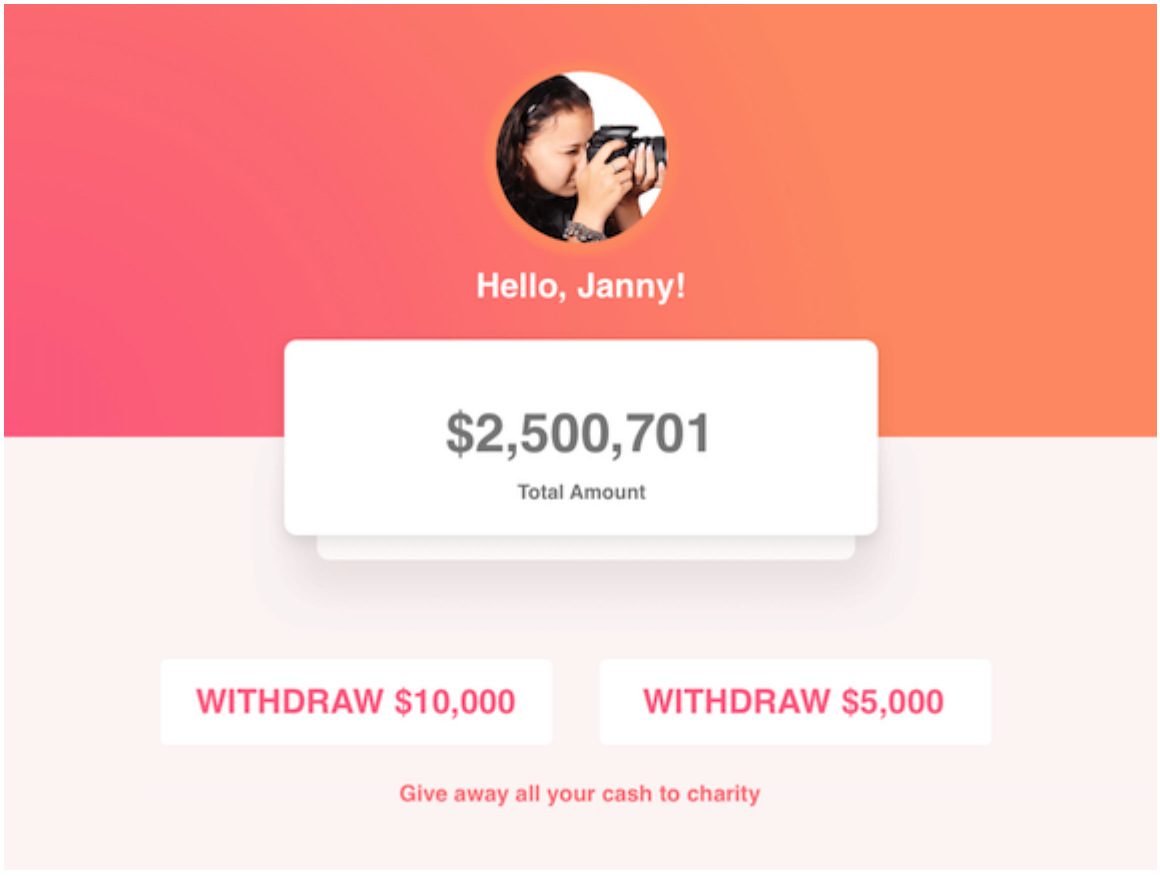
We'll make a simple bank account application where money withdrawal is possible.

Okay, now it's your time to do something cool.

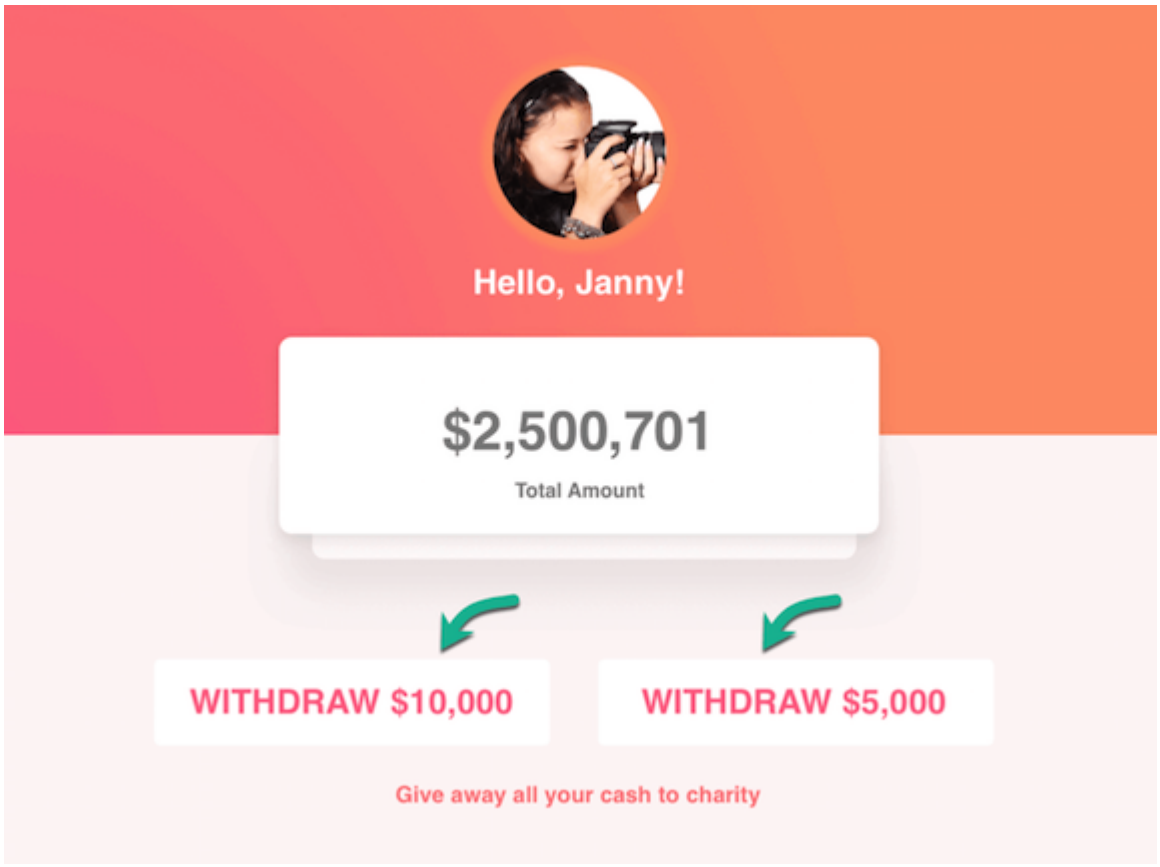
In the exercise files, I have set up a simple ReactJS application that models a user's bank application.

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

it('renders without crashing', () => {
  const div = document.createElement('div');
  ReactDOM.render(<App />, div);
  ReactDOM.unmountComponentAtNode(div);
});
```



Have a good look at the mockup above. In addition to the the user being able to view their total balance, they can also perform withdrawal actions.



The name and balance of the user are stored in the application state.

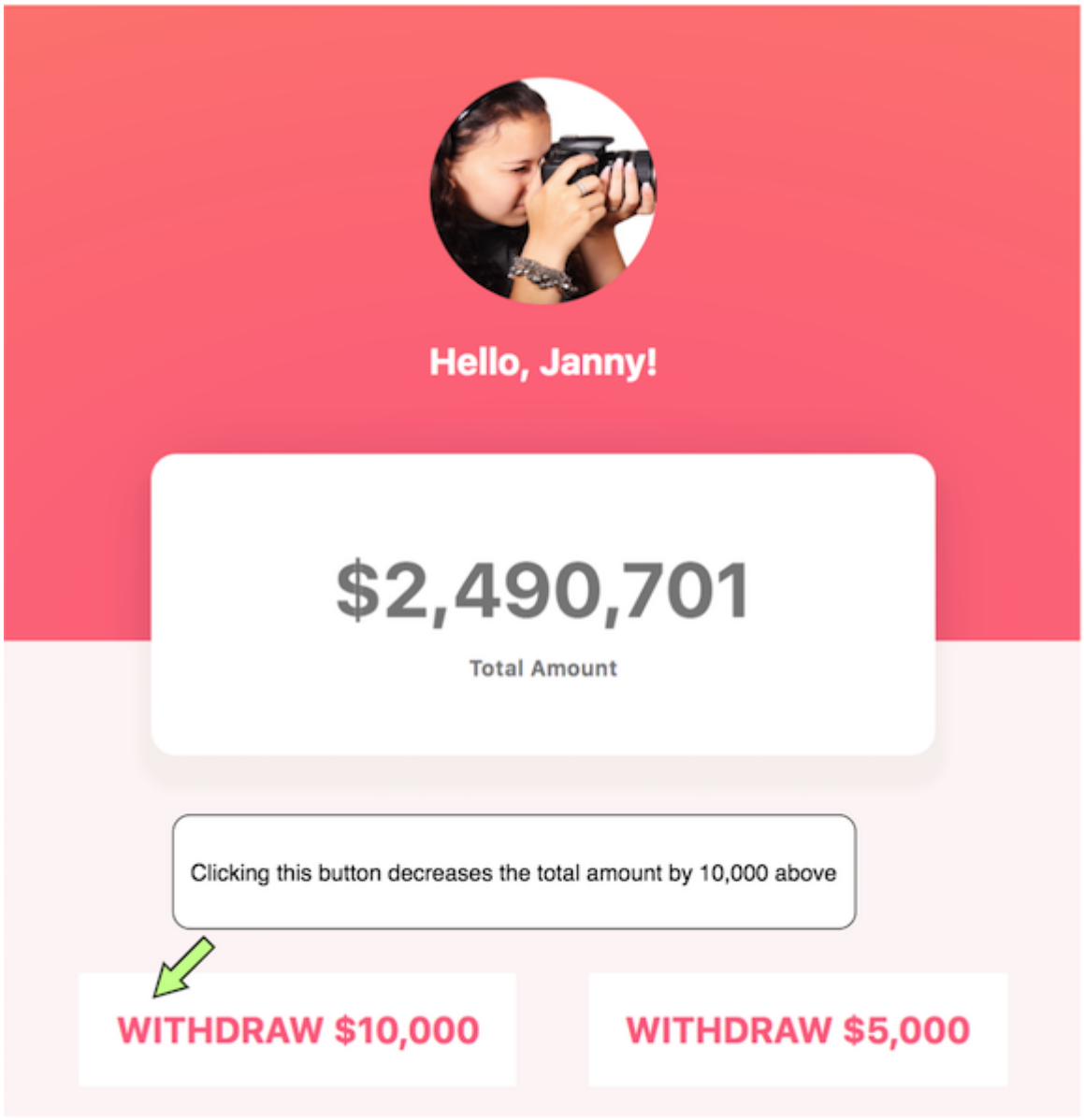
```
{
    name: "Ohans Emmanuel",
    balance: 1559.30
}
```

There are two things you need to do.

(i) Refactor the App's state to be managed solely by Redux.

(ii) Handle the withdrawal actions to actually deplete the user's balance i.e on clicking the buttons, the balance reduces.

NB: you must do this via Redux only.



> **Hint**: Remember to make a separate directory for the three different Redux actors.

An index.js file is our entry point for each component.

In the next lesson, we'll take look at the solution.