

# Ocado Frontend Pair Programming Exercise

## Introduction

We have a tournament next week and a few engineers have been working on a simple app that will display our players and help us in the creation of teams.

However, the engineers who were working on this app have gone on annual leave and have left us to implement the rest of the design and functionality.

The application is still very much a work in progress and there are lots of improvements that could be made. As we go through, feel free to talk about/make any changes that would help improve the code.

The application is a React app created using Create React App (<https://github.com/facebook/create-react-app>) and so includes source maps & hot reloading for your convenience. It has some basic styles using SCSS.

## Scripts

Tests: ``yarn test`` or ``npm run test``

App: ``yarn start`` or ``npm run start``

## Setup

In **src/components/App.js** please replace the placeholder with the code found below.

```
<div className="heading1">Squad</div>
<div className="heading2">
  <span>Selected players: </span>
  <span>0</span>
</div>
<div>
  <div className="button button--link">
    Select 10 random players
  </div>
  <div className="button button--link">Select all players</div>
  <div className="button button--link">Unselect all players</div>
</div>
<button
  onClick={() => setModalOpen(true)}
  className="button button--primary"
>
  Generate teams
</button>
```

# Exercises

## Part 1: Design (45 min)

We've been given some designs for both mobile & desktop. These can be found below in this document.

The app doesn't currently look like the designs we've been given. So for this first section, we will work on implementing these designs.

Styles can be found in **src/styles**.

We've moved all styles you're unlikely to need into the **utility-style** folder.

You'll mostly only need to touch the **\_player.scss** file, but you are welcome to change anything you like.

We have provided some variables in **\_settings.scss** that you are welcome to use.



























Desktop Design ( $\geq 576\text{px}$ )

## Squad

Selected players: 0

[Select 10 random players](#) [Select all players](#) [Unselect all players](#)

GENERATE TEAMS

 Alice <input type="checkbox"/> SELECT	 Bob <input type="checkbox"/> SELECT	 Carol <input type="checkbox"/> SELECT	 Dave <input type="checkbox"/> SELECT	 Eve <input type="checkbox"/> SELECT	 Frank <input type="checkbox"/> SELECT	 Grace <input type="checkbox"/> SELECT
 Heidi <input type="checkbox"/> SELECT	 Ian <input type="checkbox"/> SELECT	 Joan <input type="checkbox"/> SELECT	 Karl <input type="checkbox"/> SELECT	 Laura <input type="checkbox"/> SELECT	 Mallet <input type="checkbox"/> SELECT	 Naomi <input type="checkbox"/> SELECT
 Oscar <input type="checkbox"/> SELECT	 Peggy <input type="checkbox"/> SELECT	 Quentin <input type="checkbox"/> SELECT	 Rachel <input type="checkbox"/> SELECT	 Samir <input type="checkbox"/> SELECT	 Tara <input type="checkbox"/> SELECT	 Ulysses <input type="checkbox"/> SELECT
 Vera <input type="checkbox"/> SELECT	 Walter <input type="checkbox"/> SELECT	 Xavier <input type="checkbox"/> SELECT	 Yvonne <input type="checkbox"/> SELECT	 Zack <input type="checkbox"/> SELECT		

# Squad


Selected players: 0

Select 10 random players

Select all players


Unselect all players

**GENERATE TEAMS**




Alice

☐ **SELECT**



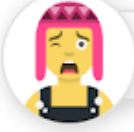
Bob

☐ **SELECT**



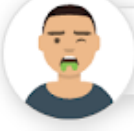
Carol

☐ **SELECT**



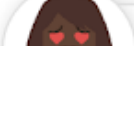
Dave

☐ **SELECT**



Eve

☐ **SELECT**



Frank

☐ **SELECT**

## Part 2: Functionality (45 min)

The app is missing some core pieces of functionality, namely the ability to select players and generate teams. You should implement this functionality by completing the stories below.

The bulk of the application code can be found in the **src/components/App.js** file, with its tests located in **App.test.js**.

### Selection Stories

Story 1:

As a user  
When I select or unselect a player  
Then I should see the selected player count update

Story 2:

As a user  
When I click the appropriate button  
Then I should select/unselect all players

Story 3:

As a user  
When I click the appropriate button  
Then I should select a random set of players

### Team Generation Stories

There is a `<TeamModal />` component that will open when Generate Teams is clicked. This component accepts 2 teams that consist of an array of player objects. You can use this component to validate your team generation.

Story 4:

As a user  
When I click Generate Teams  
Then the modal should open showing both teams

Story 5:

As a user  
When I click Generate Teams  
Then the modal should open showing both teams  
And both teams should be different every time it is generated