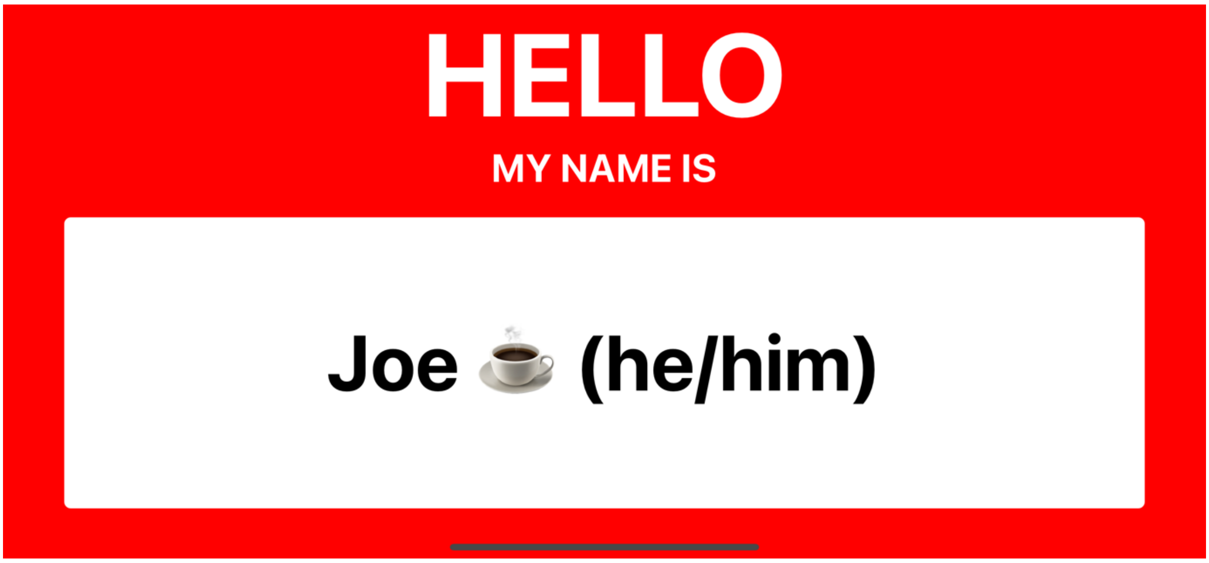
Name badge



HELLO

MY NAME IS

Joe ☕ (he/him)

In this first assignment, we are going to take a look at building a simple application in React Native. This is the first step in a long path to understanding React Native, so don't worry. I'm going to guide you through the steps in detail this time. We'll expand upon some of the more advanced elements shown in this assignment in the next few weeks.

By completing this project, you should form a good understanding of what it's like to program in this environment.

We're going to be making name badges! Follow my instructions and then feel free to customise. I've given you some examples of how you could expand upon this app at the end.

How do I submit this?

This graded assessment is one of five graded assessments which will make up your midterm assignment submitted in week 10. I strongly advise you to complete it now as it is crucial to expanding your understanding of the subsequent weeks. Once you have completed it, store it somewhere safe and do not share it with others. Instead, you can upload it with the other assignments during week 10.

Steps to complete

Firstly, we need to build a blank project. Remember, if you have forgotten how to do this, just skip back to the introduction video. I explain it in detail there. Once you are ready, let's continue.

(1) Landscape! With this project, we are closely replicating the "hello I'm…" badge format and this requires a landscape application! To make this work, we have to do a few things.

Firstly, it starts with installing a library from Expo. To do this, run

```
expo install expo-screen-orientation
```

in your terminal/command prompt. Note that you can't be running the application while doing this, so if you are, press control-c or command-c to stop the application. You also need to be in the project folder directory before running this command.

Once done, we have to import the library using

```
import * as ScreenOrientation from 'expo-screen-orientation';
```

And add the following line first thing within the App() function

```
ScreenOrientation.lockAsync(ScreenOrientation.OrientationLock.LANDSCAPE_LEFT);
```

You should notice the screen is now displaying landscape, perfect!

(2) Badge styling

We need to replicate the styling of the badge, so we'll use some stylesheet styles and view components.

- Firstly, change the background color in the *container* stylesheet style to red. It should be white by default.
- Then, also make sure the *container* style includes: **`alignItems: 'center',`** and **`justifyContent: 'center'`**. These styles will make sure that anything inside of our View container will be positioned in the center, both horizontally and vertically.

(3) Text Let's add some text. We will need to add a few text components and style them appropriately.

- Add two text elements inside of your View container. In the first, add "Hello" and, in the second, add "my name is". At first, your text will be boring black default text, but we'll add some styling next.
- Firstly, add a new style to the stylesheet called *welcomeText;* it should look like this:

```
welcomeText: {
    fontSize: 90,
    textTransform: 'uppercase',
    fontWeight: 'bold',
    color: 'white',
    textAlign: 'center'
},
```

- Make sure you've added commas at the end of the closing bracket, otherwise your code might throw an error.
- Add another style called *subtitleText* which should look like this:

```
subtitleText: {
    fontSize: 30,
    textTransform: 'uppercase',
    fontWeight: 'bold',
    color: 'white',
    marginBottom: 20,
    textAlign: 'center’
},
```

- Now we need to apply the styles to the two tags, apply the *welcomeText* to the “Hello” Text component and the *subtitleText* to the “my name is” Text component. You can do this by adding style={styles.welcomeText} to your tag, like below
  `<Text style={styles.welcomeText}>Hello</Text>`

- You should now have two text elements that are styled similarly to the image above.

(4) The box In the last step, we add the white box containing our name. This is essentially a View component, with a Text component inside.

- Add a View component after your last Text tag.
- Style it with the following styles: `width: "100%", height: "55%", backgroundColor: "white", borderRadius: 5, justifyContent:"center"`. This will require you to make a style like above and apply it to the View component.
- Inside the View, add a Text and style it with the following styles: `fontSize: 60, textAlign: "center", fontWeight: "bold"`.
- This should add the text and style it exactly like the image above.

(5) Customise it! Now you should hopefully have an app that looks similar to my screenshot. Now is the time to customise it. Add in your own name, perhaps some emojis that best represent you, pronouns or anything you want! Also, feel free to translate this into any language that you like! Make it feel uniquely yours and then post a screenshot of it to the group to say hello.

(6) A few additions We do a few things in this demo which I’d like to expand upon a little.

Firstly, you have probably noticed that if you have a device with a notch, your badge might go behind un-usable spaces. We can fix this by using a SafeAreaView component. This component applies safe margins to the view, based on device requirements. To use it, wrap everything inside of the View container in a `<SafeAreaView>` with the following style: `flex:1, width:"100%", height:"100%"`. This will fix the problem.

Also, you might be wondering what does borderRadius do? borderRadius rounds the edges of View components by a radius measured in pixels. This gives us nice round corners; you’ll see me use this a lot.

Also, what does textTransform do? It allows us to alter the text, so perhaps transforming it to uppercase in this case, or you can auto-capitalise and lowercase all text too.

Mark scheme

Description	Marks
Landscape mode	
No attempt made	0
Landscape mode works correctly	1
Layout	
No attempt made	0
The layout is incorrect or broken	1
The layout is correct and works as expected	2
Styling	
No attempt made	0
Some styling, however, some errors still remain	1
Most styling is correct and working	2
Styling is correct and identical to the example	3
Some improvements have been made to the styling	4
Significant improvements to the styling	5
Additions	
No improvements have been made to the layout/content	0
Some improvements were made	1
Significant improvements were made	2