**[Renton Technical College CSI-248](https://github.com/EmeryCSI/csi248-guidedactivity5" \l "renton-technical-college-csi-248)**

[**Guided Activity 5**](https://github.com/EmeryCSI/csi248-guidedactivity5#guided-activity-5)

This repository is a part of CSI-248 at Renton Technical College.

[**Guided Activity 5 - If the images are not loading use the word doc in this repository GA5\_instructions.docx**](https://github.com/EmeryCSI/csi248-guidedactivity5#guided-activity-5---if-the-images-are-not-loading-use-the-word-doc-in-this-repository-ga5_instructionsdocx)

[**Setup Expo - If you have already done this in class you may skip this step**](https://github.com/EmeryCSI/csi248-guidedactivity5#setup-expo---if-you-have-already-done-this-in-class-you-may-skip-this-step)

1. Sign up for an account at <https://expo.dev/>
2. Install Expo Go on your mobile device in the Apple App Store or Google Play Store
3. Sign in to the Expo Go App using your account you made at expo.dev

[**Clone the repo**](https://github.com/EmeryCSI/csi248-guidedactivity5#clone-the-repo)

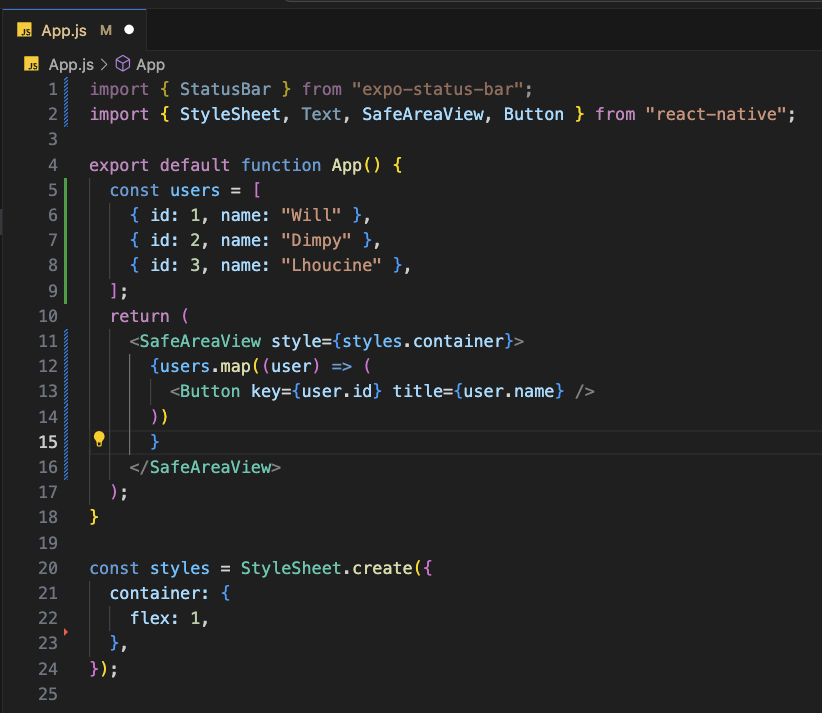
1. Clone the repository to your local machine. (Do not use OneDrive for assignments in this course!)
2. Make note of the folder where you cloned the repository.
3. After you have cloned this repository navigate to your local repository using the cd command.
4. Open the repository in Visual Studio Code by typing code .
5. Create a folder to store the requested Screenshots for this assignment.

[**Creating your first react-native project**](https://github.com/EmeryCSI/csi248-guidedactivity5#creating-your-first-react-native-project)

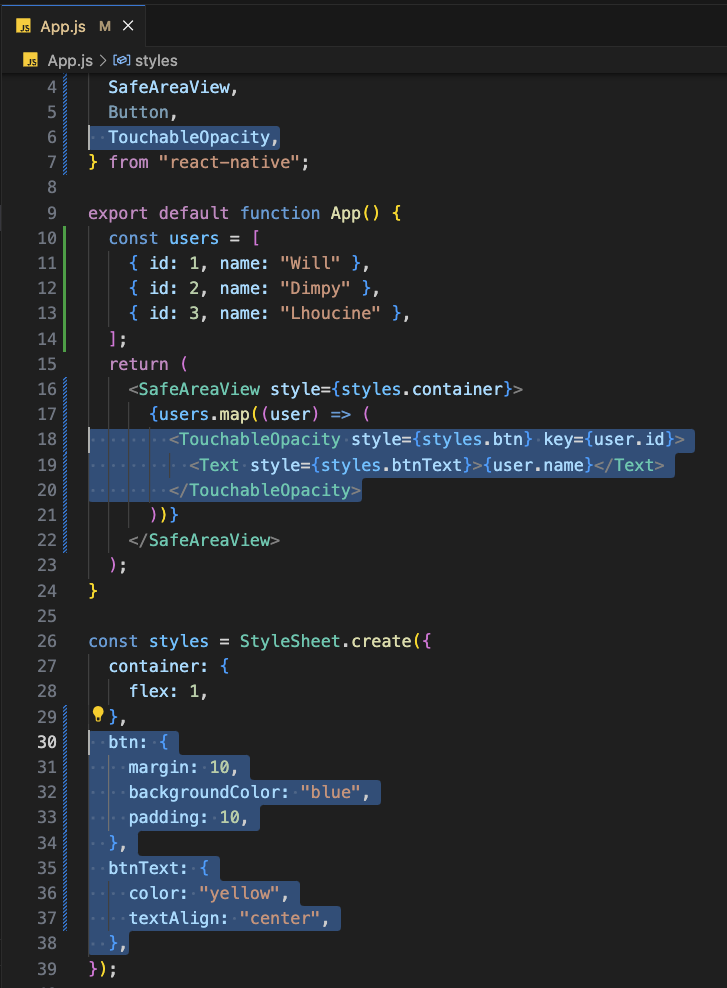
1. Open the terminal in Visual Studio Code and create a new expo app by running npx create-expo-app custom-button
2. Run cd custom-button
3. You are now in the root project of the folder. Notice that there are some files that look familiar such as package.json and App.js
4. Run npx expo install react-dom react-native-web @expo/webpack-config to enable react-native for web
5. Run npx expo start to launch the project. You will see a QR Code in the terminal.
6. Type w to launch the app in a web browser. You should now see your app running in your default browser.
7. Scan this QR Code with the Camera on IOS or using the Expo Go app on Android and you App will launch on your device.

[**Understanding Pressable**](https://github.com/EmeryCSI/csi248-guidedactivity5#understanding-pressable)

1. Replace App.js with the following:

[](https://private-user-images.githubusercontent.com/102991550/280841639-ef7bb326-ee5e-48c1-ba25-59e016f017c2.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQxNjM5LWVmN2JiMzI2LWVlNWUtNDhjMS1iYTI1LTU5ZTAxNmYwMTdjMi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1jNGNhYTYwMTg4ODhhZjVhZTUzYWI0YTcwNWNiOTEwYTE1NWQ1Yzk0YmMxOTBlZDczNWRkOGFjMzUwNzY3NDZmJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.eXv5gkXJKOCLsNoYo4gg6I9l5s81mXDZ27W6A5Yc-jI)

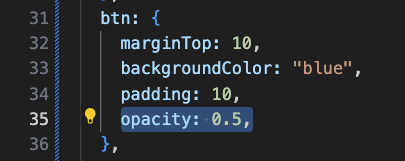
1. The button component works for quick projects but it is very limited in what can be done with it. For example the background color can be change but not the text color.
2. Button has a nice effect when it is clicked on however that effect cannot be customized.
3. Make the following changes to App.js

[](https://private-user-images.githubusercontent.com/102991550/280842091-38627306-4b68-45c2-af23-f3326f7e00f6.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQyMDkxLTM4NjI3MzA2LTRiNjgtNDVjMi1hZjIzLWYzMzI2ZjdlMDBmNi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT05OGNkZjk2MWYxMjBmNmEzMDE2NWFlZTU1NjViODY0MDMwNGRmZDliM2Y3M2M4MWMyZDkxNzM3M2NlZDU1Nzg5JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.SSyjbmHK2yXMRiwUMJoK9wk5KFZCZ7-Kk1Nf86eI9-k)

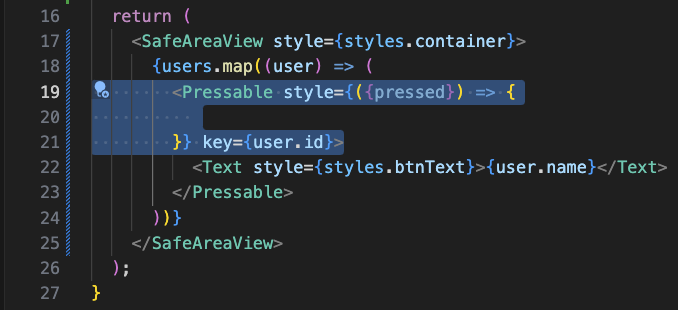
1. TouchableOpacity offers the effect when a button is clicked as well as the ability to change both background color and the text color. Notice that the text inside of the button is its own component. There is no title prop used.
2. Change the TouchableOpacity component to a Pressable.

[](https://private-user-images.githubusercontent.com/102991550/280842568-15855f52-ce5c-4766-98f1-ce4442ed5101.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQyNTY4LTE1ODU1ZjUyLWNlNWMtNDc2Ni05OGYxLWNlNDQ0MmVkNTEwMS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT05MDU5YWI3YmQyNjQzOGRkMWQ0NWMwMjkxODlkOGQ2YmU4ZjUyMGYwMzljODM5NjJlMTcxZTljNjdlMDlhN2YyJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.8u2cvWMHaUdrfHTqfBNMVHFGUodmrbvjUniiN6kbTrY)

1. Now noticed when you click on the buttons there is no effect to show that is has been clicked on. It is possible to create this effect with Pressable, however we much create it ourselves.
2. Notice what happens if you add an opacity to the btn style:

[](https://private-user-images.githubusercontent.com/102991550/280843399-3cd35233-5cec-4ed0-b5ff-d04654710ae2.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQzMzk5LTNjZDM1MjMzLTVjZWMtNGVkMC1iNWZmLWQwNDY1NDcxMGFlMi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1mZDc0Zjc5NWNhZjc3MDQ2OWRhZWUzMzk4ZGUwMWFhN2VkYzRiMzQ4Yzk3NjY1Y2NlMGJjY2EwOTYzZGU1YTIwJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.TEWb4RNRyI7K1COozX8x2AeCnRMG3ANmf11BfrSKKAY)

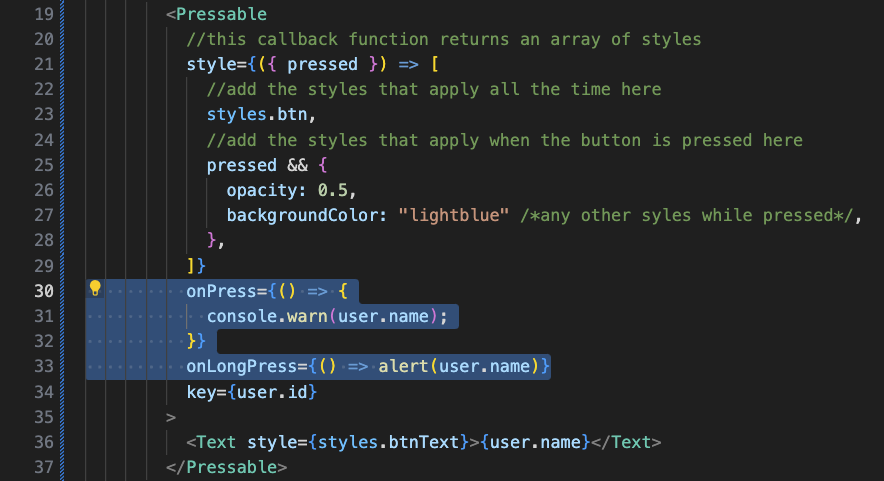
1. The buttons will appear faded. We can make this happen on press by supplying a callback function to the style prop.
2. Delete the opacity: 0.5 from the btn style.
3. The style prop of the pressable is now supplying a callback function, that callback function is taking in a parameter called pressed.
4. pressed will be true when the button is pressed or false otherwise.

[](https://private-user-images.githubusercontent.com/102991550/280844070-f3c48249-f1ad-48c0-975d-edd36110ff82.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQ0MDcwLWYzYzQ4MjQ5LWYxYWQtNDhjMC05NzVkLWVkZDM2MTEwZmY4Mi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT02MTNkMGM0NjE1ZGQzNWFmYjZhY2M1ZTIwYmI1OGQxMzc3Y2JmN2M4ZmM2OWE3NzBhNzg3N2FkZWNhNzVlMzllJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.HFXZ2_y2IAQKvwMpGalUIVd2zcCgxZBWcpC72zjcwOw)

1. Using this pressed parameter we can supply an array of styles to the style prop:

[](https://private-user-images.githubusercontent.com/102991550/280845968-401b41f2-ac4e-4a47-af9d-7cefe80c3e69.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQ1OTY4LTQwMWI0MWYyLWFjNGUtNGE0Ny1hZjlkLTdjZWZlODBjM2U2OS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT1iYmU5YjA1YTViZThmMWRiNmViZjY3YjIxNzdmYjhjM2U1MmRlMDQwMjJlZDc5N2I1MDRkNWNjNDljZmIxY2FhJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.I96YXLgs3mXgzVMW616pyLXt3Kac48IAwo3Lo26myL8)

1. Notice now when you click the button you will see the opacity and background change
2. Pressable has multiple press events that you can use. onHoverIn onHoverOut onLongPress onPress onPressIn onPressOut
3. Lets add a press and a longPress event:

[](https://private-user-images.githubusercontent.com/102991550/280846720-efc5a2f4-c865-484a-8cb1-8784025e021b.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODQ2NzIwLWVmYzVhMmY0LWM4NjUtNDg0YS04Y2IxLTg3ODQwMjVlMDIxYi5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT02NmU4OGJiZGI5MmNiNGQzYTI1NGUyYzNkNTViZWY1MDZlZmUzNzExMTg3OGUyYjE4ZjNiNTg1YzY5YWVkNWRlJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.vliHbNG3RXBmIeOc-UYKwathRHmNiivb8mNFHyc04TU)

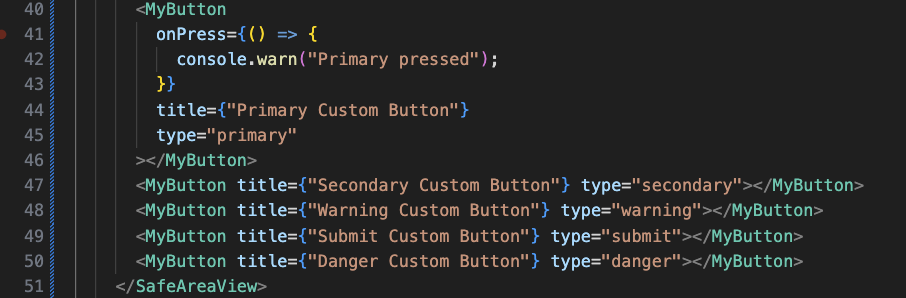
1. Click your pressable and take a screenshot of the output
2. Click and hold on your pressable and take a screenshot of the output.

[**Create a custom button component**](https://github.com/EmeryCSI/csi248-guidedactivity5#create-a-custom-button-component)

1. As you can see, this is quite a lot of code to repeat everywhere that you want a button.
2. It is common practice to create a custom button for your app that you can re-use.
3. Typically you will pre-define a few different colors within the component and then select which one you want rendered using a prop.
4. Common button types are primary, secondary, submit, warning and danger.
5. Create a CustomButtom component that takes the following props onPress, //callBack for press onLongPress, //callback for longpress title, //this will be the text of the button type = "primary", // this will change the appearance of the button and the text, default is primary disabled = false, //buton is not disabled by default pressedOpacity = 0.5 //default opacity when pressed
6. You can chose whatever colors you like for the different types of buttons. Here is an example

[](https://private-user-images.githubusercontent.com/102991550/280852664-474e9551-f77e-4f9e-b0cf-7ad82c9f0775.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODUyNjY0LTQ3NGU5NTUxLWY3N2UtNGY5ZS1iMGNmLTdhZDgyYzlmMDc3NS5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT00NGI2YTZmNTFkYWIyNDMwODBkMmIzZTgxYjQwOWQ2MzNhMTAyMzk5YzJmNzNjYWU1MmVmNzViYjhlODIyNjUzJlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.iA3WeGCg2MLp2CyfrnVWeJoPpMv88TxGwwQXqUdH9Eg)

1. Once complete render the custom buttons in App.js and take a screenshot.
2. Notice how much easier it is to use these buttons than the Pressable element we made earlier.

[](https://private-user-images.githubusercontent.com/102991550/280852936-e6fdd0ce-deee-4ad9-a601-49f6372aabb4.png?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTEiLCJleHAiOjE2OTkzMDcxMDQsIm5iZiI6MTY5OTMwNjgwNCwicGF0aCI6Ii8xMDI5OTE1NTAvMjgwODUyOTM2LWU2ZmRkMGNlLWRlZWUtNGFkOS1hNjAxLTQ5ZjYzNzJhYWJiNC5wbmc_WC1BbXotQWxnb3JpdGhtPUFXUzQtSE1BQy1TSEEyNTYmWC1BbXotQ3JlZGVudGlhbD1BS0lBSVdOSllBWDRDU1ZFSDUzQSUyRjIwMjMxMTA2JTJGdXMtZWFzdC0xJTJGczMlMkZhd3M0X3JlcXVlc3QmWC1BbXotRGF0ZT0yMDIzMTEwNlQyMTQwMDRaJlgtQW16LUV4cGlyZXM9MzAwJlgtQW16LVNpZ25hdHVyZT0zMGI3ZjBiMjc2ZjFkMWJiMGUxNGVjNjUwNzk2NjdiMTI0MzAyNmZlMDE1ZDc1M2U0OThmOTJmMWE4NTI1NjY1JlgtQW16LVNpZ25lZEhlYWRlcnM9aG9zdCZhY3Rvcl9pZD0wJmtleV9pZD0wJnJlcG9faWQ9MCJ9.lIDEs2IZfAu0FLgoTPsftCd2ejDJvkGfseP-XP3bGsE)

9 Since you made the component yourself you have full control over its appearance, however you also have the ease of use of the built-in Button component.

1. Take a screenshot of your final functioning app add it to the screenshots folder.
2. git add .
3. git commit -m "Assignment Complete"