Hi,

Your task is to continue styling your Week1 app. In your first assignment, you were asked to study the "Layout with Flexbox" (https://reactnative.dev/docs/flexbox). Flexbox is a layout implementation that React Native uses to provide an efficient way for users to create UIs and control positioning.

If you are familiar with Flexbox and the first task for the week was easy, with minor changes, you are able to finetune the looks in addition to the layout. Otherwise I suggest you should use this time to also practice some more Flexbox. Note that Flexbox in RN has small differences to how it works on the web.

If you need some more help with Flexbox (some are CSS material):

Flexbox Froggy - A game for learning CSS flexbox
https://flexboxfroggy.com/
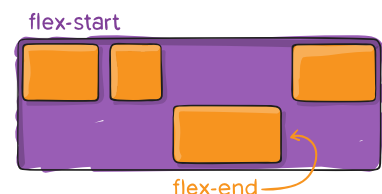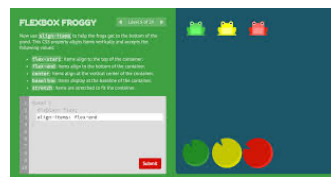
CSS-TRICKS a Complete guide to flexbox
https://css-tricks.com/snippets/css/a-guide-to-flexbox'

Build with Flexbox – Interactive Reference to experiment with layouts (for React Native)
http://flexbox.buildwithreact.com/
https://yogalayout.com/playground/
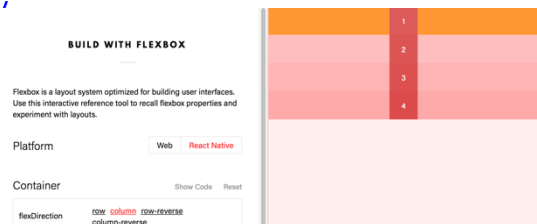
Now style your app a bit further. The idea of this lab is to get more used to the styling of RN apps. Your page could look something like these layouts, but please style it to your liking. Check the next page for more info:
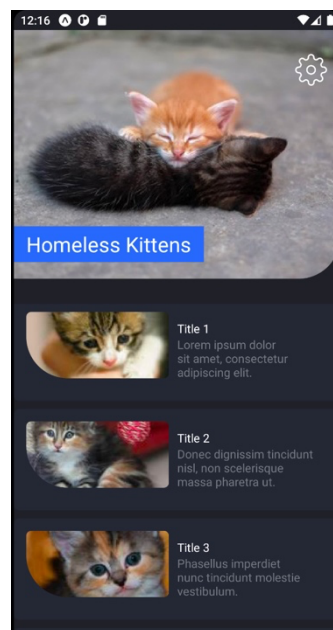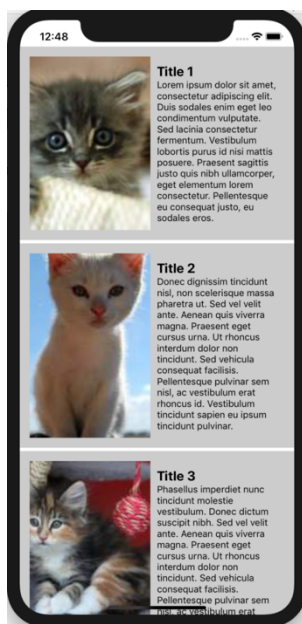
Styling in React:

First, read about styling in React: https://reactnative.dev/docs/style

With React Native, you don't use a special language or syntax for defining styles. You just style your application using JavaScript. All of the core components accept a prop named style. The style names and values usually match how CSS works on the web, except names are written using camel casing, e.g. backgroundColor rather than background-color. React Native doesn't have all of the capabilities that we have on the web.

React Native Styling Cheat Sheet
https://github.com/vhpoet/react-native-styling-cheat-sheet

From the cheat sheet you can quickly check which properties work and how they are used in RN. Looking at the cheat sheet you can quickly notice for example that React doesn't like shorthand properties.. :/

You could try to style the following:

Change and adjust the colors of the statusbar. https://reactnative.dev/docs/statusbar

Add a navigation/setting or any other icon. You can use:
https://www.npmjs.com/package/react-native-feather

https://feathericons.com/

Position Views on top of each other with absolute/relative layout

https://reactnative.dev/docs/flexbox#absolute--relative-layout

Header Image should be an <imageBackground>
https://reactnative.dev/docs/imagebackground

Use Flexbox for the spacing of the Views. Remember that you can make nested flex views.

Change the font size & color

Apply adequate margin/padding to elements

Try the border radius for rounding images or edges of some views