Assignment 3 - Little Lemon newsletter sign-up

Overview

For this assignment, you will create a simple app in React Native that allows users to sign up for the Little Lemon newsletter to ensure that they always get the latest updates from the restaurant.

This task will entail creating two screens and setting up a visual layout with core components such as **Text**, **Image** and **View**, configuring some **TextInput** and **Pressable** components that the user can interact with, and setting up React Navigation so that users can move between screens.

You are encouraged to apply all the skills that you acquired during this course, and you can revisit the previous course materials as needed if you need to refresh your knowledge or find some inspiration.

Instructions

The Little Lemon app should have a **Welcome** screen and a **Subscribe** screen. Users will start on the **Welcome** screen and move to the **Subscribe** screen, where they can fill out and submit a form. The steps below will guide you through the specific requirements.

Step 1: Download the starter code

Download the ZIP file linked below and extract the files. These files will need to be imported into a development environment for editing.

3_little-lemon-starter-code.zip

Step 2: Build and Setup code

Your app can be built using React Native CLI, Expo CLI, or Expo Snack. Visit the links below for installation and setup instructions as needed:

- Setting up React Native CLI (click on the **React Native CLI Quickstart** tab)
- Setting up Expo CLI (click on the **Expo Go Quickstart** tab)

- A guide to Expo Snack, an in-browser React Native environment with no local setup required

Tip: Use whichever method you have been using to run code throughout this course.

Step 3: Setup Stack Navigation

You will need React Navigation to move between the two screens of the app. The **App.js** file has already been completed for you in the starter code.

The **navigators/RootNavigator.js** file needs additional setup. Make sure to add the stack screens to complete the navigation setup.

```
import * as React from "react";
import { createNativeStackNavigator } from "@react-navigation/native-stack";
import WelcomeScreen from "../screens/WelcomeScreen";
import SubscribeScreen from "../screens/SubscribeScreen";

const Stack = createNativeStackNavigator();

const RootNavigator = () => {
  return (
    <Stack.Navigator>
      {/* Set up stack navigation to move between welcome screen and subscribe screen
here */}
    </Stack.Navigator>
  );
};

export default RootNavigator;
```

Step 4: Welcome screen component

The Welcome screen should have a Little Lemon logo and some text to welcome users. It should also contain a **Pressable** button that can navigate to the Subscribe screen. The text on this button should read “Newsletter”.

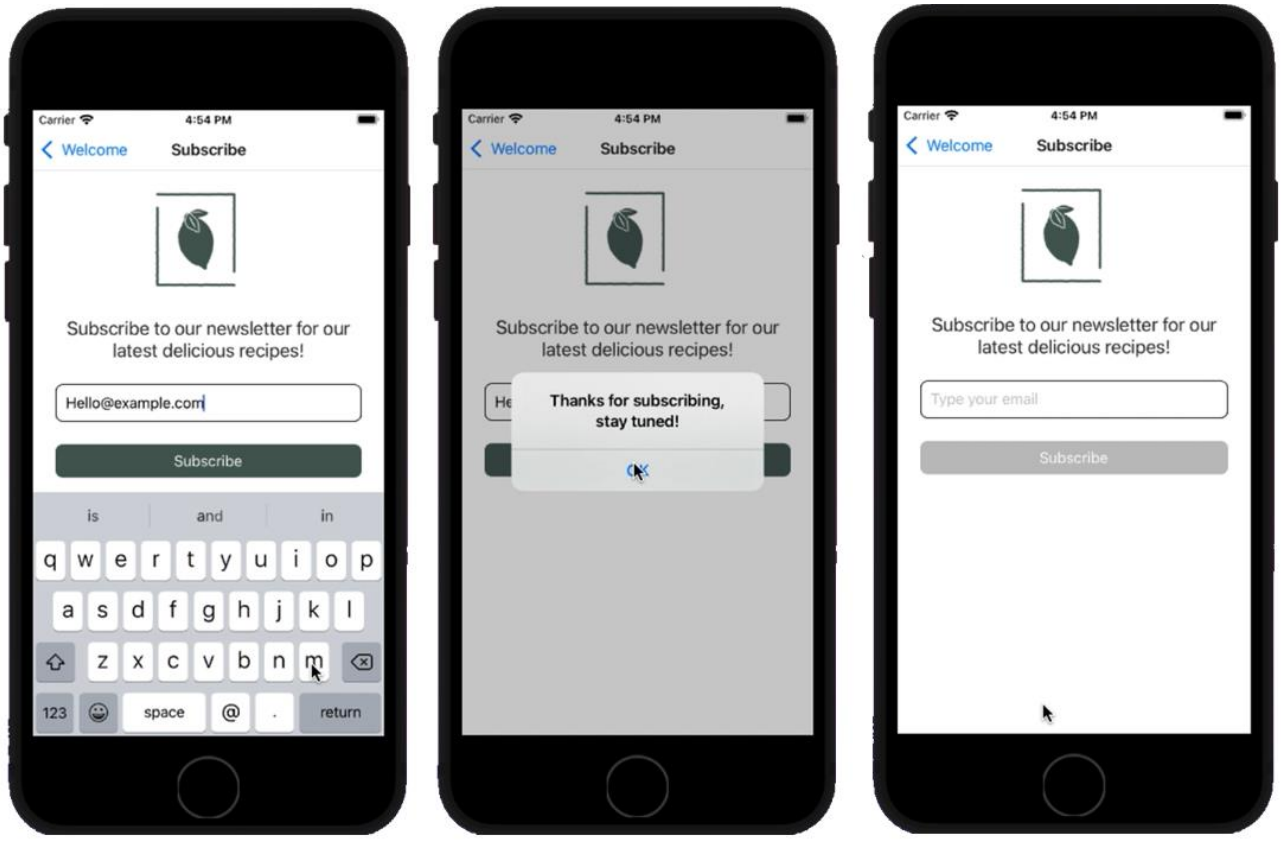
Add the code to complete the Welcome Screen component. You will need components such as **View**, **Text**, **Image** and **Pressable** to achieve the desired output. Make sure to use the **navigation** prop to move between the screens.

```
import * as React from 'react';
import { View } from 'react-native';

const WelcomeScreen = ({ navigation }) => {
  // Add welcome screen code here.
  return <View></View>;
};
```

```
export default WelcomeScreen;
```

Step 5: Subscribe Screen component



The Subscribe screen must contain a text input box that allows the user to enter their email address and a button that reads “Subscribe”. Clicking on this button will then show an alert message to the user to confirm the subscription. The entire workflow is illustrated in the screenshots above. Use this as an inspiration to get your own Little Lemon app setup.

Add the code to complete the Subscribe screen component. In addition to the components that you used in the Welcome screen, you will also use the **TextInput** component to configure the text input box to enter an email address and the **Alert** component to display an alert message.

```
import * as React from 'react';

const SubscribeScreen = () => {
  // Add subscribe screen code here
  return <View></View>;
};
```

```
export default SubscribeScreen;
```

Step 6: Validate email

You can use the **validateEmail** util method, which is part of the starter code, to determine if the user has entered a valid email. If the email is valid, then the Subscribe button should be enabled. If the email is invalid, then the button should be disabled. You will simply have to work on the UI to enable and disable the button. The **validateEmail** util method is already available! You can import it with the following line of code:

```
import { ValidateEmail } from '../utils';
```

Step 7: Style the components

Once you get the basic functionality completed, revisit the components and ensure proper styling has been applied. Use the StyleSheet API to abstract styles and keep code clean. Make sure to add meaningful names to the styles as well.

Grading Criteria

At the peer review step, you will submit your completed assignment files. These will be downloaded by at least two of your peers, who will view your code and interact with your app in an emulator. They will review your work based on the following criteria:

- Is the Native Stack Navigation setup appropriately in the *navigators/Root Navigator.js* file?
- Is there a Welcome screen component that contains **View**, **Text**, **Image** and **Pressable** components?
- Is there a “Newsletter” button on the Welcome screen that moves to the Subscribe screen when clicked?
- Is there a **TextInput** component on the Subscribe screen which displays a text input box that prompts the user to enter an email address?
- Is there a button on the Subscribe screen that is only enabled after text has been entered in the text input box above?

- Does clicking the Subscribe button trigger an alert containing a confirmation message?
- Is the Little Lemon logo image rendered on both the Welcome Screen and the Subscribe Screen using the **Image** component?
- In the code, are StyleSheets used to store styles outside of the component's render?