**Exercise: Create a login page**

**Overview**

You’ve already done quite a bit with the **TextInput** component, which includes storing the user’s input in the local state and styling the text box.

This time around, you’ll use **TextInput** props to create a login page for the Little Lemon app.

**Scenario**

In this exercise, your assignment is to build a simple login page for the Little Lemon app. You will create an email address and password text input box and allow the user to enter the requested information.

Also, make sure a suitable keyboard type is displayed when the user focuses on a text input box. For the password box, the characters should be hidden/secure during user input. You will do this by using the **keyboardType** and **secureTextEntry** props of the **TextInput** component.

Below is the starter code for the **LoginScreen.js** and **App.js** files in your project:

***LoginScreen.js***

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

import { ScrollView, Text, StyleSheet} from 'react-native';

export default function LoginScreen() {

  return (

    <ScrollView style={styles.container}>

      <Text style={styles.headerText}>Welcome to Little Lemon</Text>

      <Text style={styles.regularText}>Login to continue </Text>

    </ScrollView>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

  headerText: {

    padding: 40,

    fontSize: 30,

    color: '#EDEFEE',

    textAlign: 'center',

  },

  regularText: {

    fontSize: 24,

    padding: 20,

    marginVertical: 8,

    color: '#EDEFEE',

    textAlign: 'center',

  },

});

***App.js***

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

import \* as React from 'react';

import { View, Text, StyleSheet } from 'react-native';

import LittleLemonHeader from './components/LittleLemonHeader';

import LittleLemonFooter from './components/LittleLemonFooter';

import LoginScreen from './LoginScreen';

export default function App() {

  return (

    <>

      <View style={styles.container}>

        <LittleLemonHeader />

        <LoginScreen />

      </View>

      <View style={styles.footerContainer}>

        <LittleLemonFooter />

      </View>

    </>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#333333',

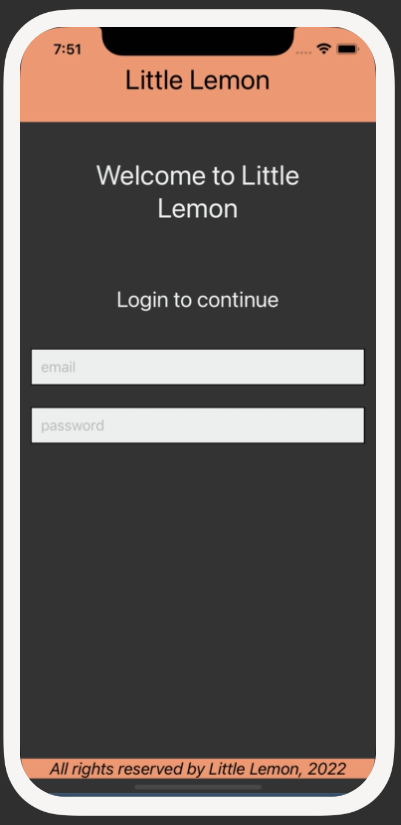
  },

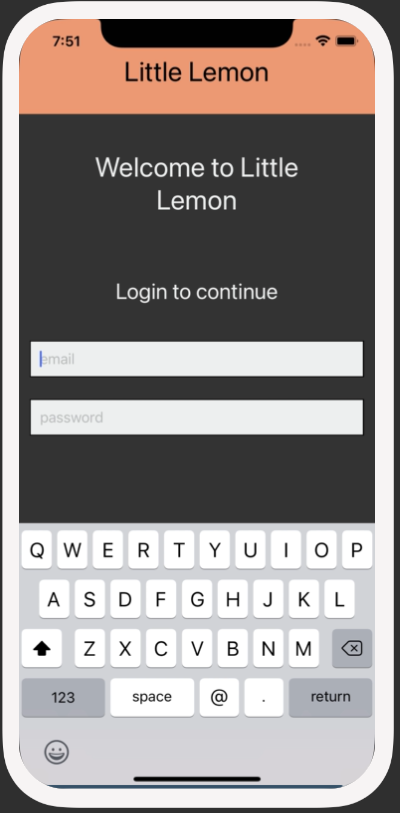
  footerContainer: { backgroundColor: '#333333' },

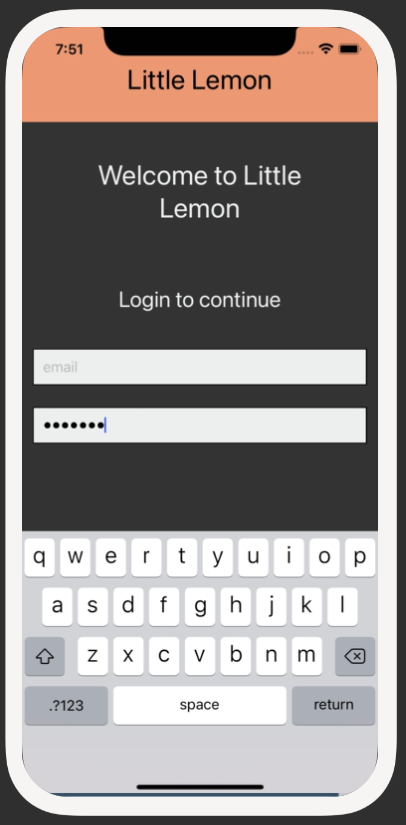
});

You will be adding the text input boxes to **LoginScreen.js**.

Refer to the emulator screenshots below to understand how your app should appear after you complete this exercise:







Colors that you will use:

#EDEFEE, black and white

**Instructions**

**Step 1: Configure TextInput components within Login Screen**

This exercise's first step is configuring the **TextInput** components within the **Login** Screen. It will accept the user’s input. You will need to configure two text input components; one for the email address and the other for the password.

Make sure to use the correct keyboard types and secure the password.

**Step 2: Store user input within TextInput as local state**

Next, you will store the user’s inputs as local state within the component.

**Hint**: You will need to store two separate local states for the email address and password.

**Step 3: Style the component**

In this step, you will style the login screen that you have created to match the screenshots. Make sure to provide meaningful names for all your styles.

**Hint**: Use the previous exercise and solution as a reference and follow a similar styling pattern.

**Conclusion**

By completing this exercise, you will demonstrate your understanding and ability to configure and utilize the **TextInput** component to accept user’s input within a text box.

Mark as completed

Like

Dislike

Report an issue