**Solution: Create a login page**

**Overview**

In this reading, you will review the solution code for the exercise.

The code block below outlines the outcome of the exercise in its entirety.

***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

32

33

34

35

36

37

38

39

40

import React, { useState } from 'react';

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

export default function LoginScreen() {

  const [email, onChangeEmail] = useState('');

  const [password, onChangePassword] = useState('');

  return (

    <ScrollView style={styles.container}>

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

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

      <TextInput

        style={styles.inputBox}

        value={email}

        onChangeText={onChangeEmail}

        placeholder={'email'}

        keyboardType={'email-address'}

      />

      <TextInput

        style={styles.inputBox}

        value={password}

        onChangeText={onChangePassword}

        placeholder={'password'}

        keyboardType={'default'}

        secureTextEntry={true}

      />

    </ScrollView>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

  headerText: {

    padding: 40,

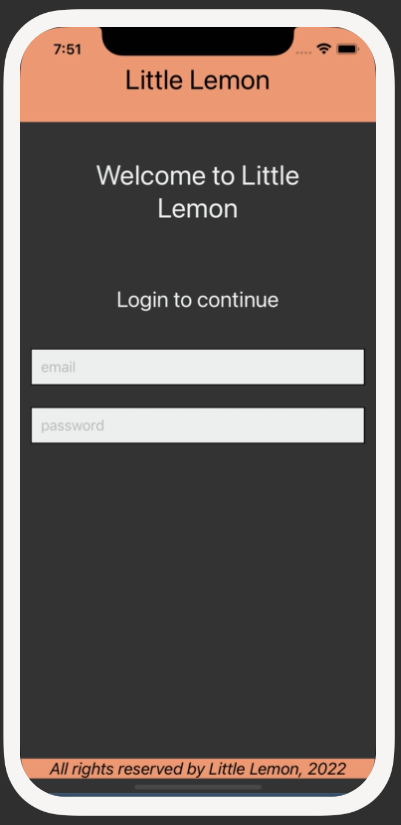
    fontSize: 30,

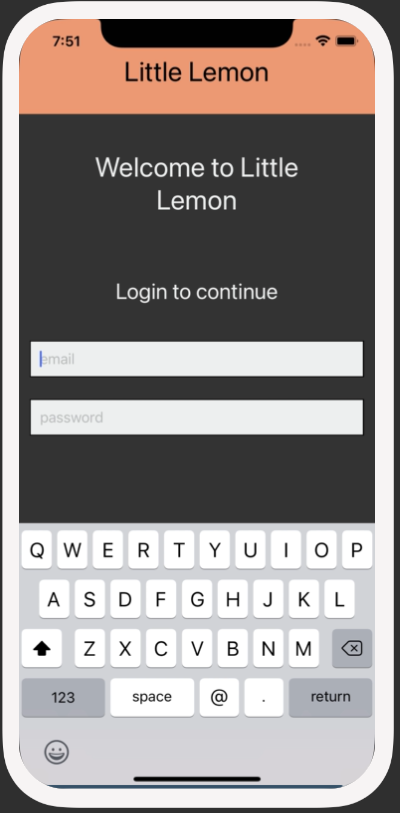
    color: '#EDEFEE',

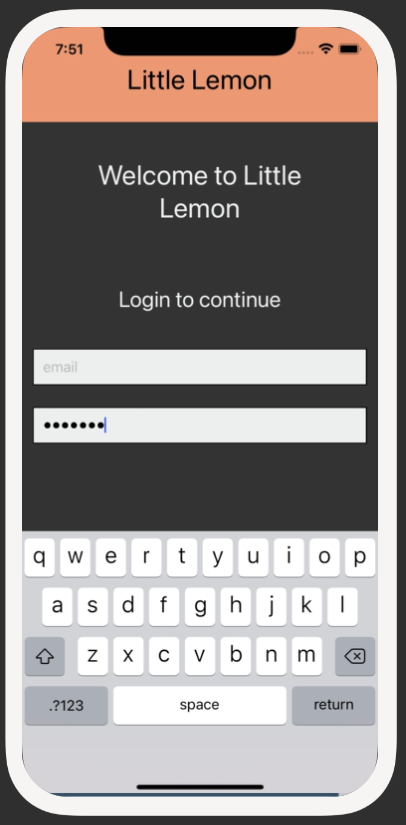
    textAlign: 'center',

  },

The code above would produce the following output on the emulator:







**Procedure**

Next, review the breakdown of the individual steps below.

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

Within the **Welcome** Screen, configure the **TextInput** component to accept the user’s input.

Import **TextInput** from React Native.

1

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

Configure **TextInput** components for email and password*:*

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

  <TextInput

        value={email}

        onChangeText={onChangeEmail}

        placeholder={'email'}

        keyboardType={'email-address'}

      />

      <TextInput

        value={password}

        onChangeText={onChangePassword}

        placeholder={'password'}

        keyboardType={'default'}

        secureTextEntry={true}

      />

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

In this step, make sure to import the **useState** hook from React and create two local state variables for storing the email and password that are entered in the text input boxes.

1

import React, { useState } from 'react';

1

2

  const [email, onChangeEmail] = useState('');

  const [password, onChangePassword] = useState('');

**Step 3: Style the component**

Once you have coded a working component, it is time to style it. In this step, you will style the text input box components to match the screenshots.

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

32

33

34

35

36

37

38

39

40

import React, { useState } from 'react';

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

export default function LoginScreen() {

  const [email, onChangeEmail] = useState('');

  const [password, onChangePassword] = useState('');

  return (

    <ScrollView style={styles.container}>

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

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

      <TextInput

        style={styles.inputBox}

        value={email}

        onChangeText={onChangeEmail}

        placeholder={'email'}

        keyboardType={'email-address'}

      />

      <TextInput

        style={styles.inputBox}

        value={password}

        onChangeText={onChangePassword}

        placeholder={'password'}

        keyboardType={'default'}

        secureTextEntry={true}

      />

    </ScrollView>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

  headerText: {

    padding: 40,

    fontSize: 30,

    color: '#EDEFEE',

    textAlign: 'center',

  },