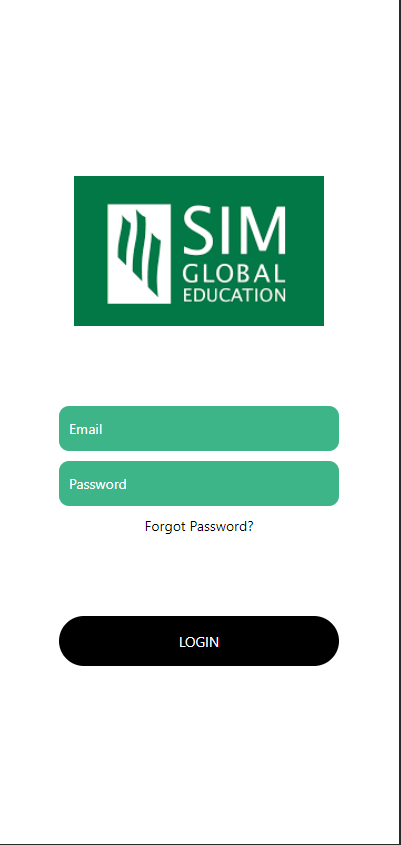


**Workshop Practice 1**

**CS - Mobile Development**

In this practical lab, we will be practicing on building interface page. This practice we will try to attempt building a simple login page that looks like the following:



The first few steps will be provided and the rest is meant for research/discover to achieve layout.

1. Create a brand new react native project.

|  |
| --- |
| Quick start command:  **expo init --npm**  **expo start or npm start** |

1. Open the App.js to start working on your application. You should have a similar code structure as shown below:

|  |
| --- |
| import { StatusBar } from 'expo-status-bar';  import React from 'react';  import { StyleSheet, Text, View } from 'react-native';    export default function App() {  return (  <View style={styles.container}>  <Text>Open up App.js to start working on your app!</Text>  <StatusBar style="auto" />  </View>  );  }    const styles = StyleSheet.create({  container: {  flex: 1,  backgroundColor: '#fff',  alignItems: 'center',  justifyContent: 'center',  },  }); |

1. On the second line of code, we import React to use JSX. Then we import the StyleSheet, Text, and View components from React Native.
2. Next, find a nice logo image online. We will add an Image to the View Layout. Start by removing the text component, and replace it with the logo image. Don't forget to import the Image component at the top.

|  |
| --- |
| <Image source={require('.assets//image.png')} /> |

Should look similar to this:

|  |
| --- |
| import React from 'react';  import { StyleSheet, Text, View, Image } from 'react-native';    export default class App extends React.Component {  render(){  return (  <View style={styles.container}>  <Image source = {require("./assets/logo.png")}/>  </View>  );  }  } |

1. Add the following styles to the image. Here we add a marginBottom: 40 style to make some space between the image and the text inputs.

|  |
| --- |
| const styles = StyleSheet.create({  container: {  flex: 1,  backgroundColor: '#fff',  alignItems: 'center',  justifyContent: 'center',  },    image :{  marginBottom: 40    }    }); |

1. Next, add text input fields for the email and password. First, add state properties to hold the email and password. State in React Native is used on components that change over time. For example, the information in the **TextInput** keeps changing as users input their information. The initial state of the email and password will be empty.
2. Start by importing **useState**, as shown below. The **useState** function will allow our functional components to be stateful.

|  |
| --- |
| import React, { useState } from 'react'; |

1. Then initialize the state by adding the following code in the App function.

|  |
| --- |
| export default function App() {  const [email, setEmail] = useState('');  const [password, setPassword] = useState('');  return (  // rest of the code  );  } |

1. We use a View to wrap each text input for easy styling. Add this code below the image.

|  |
| --- |
| <View style={styles.inputView}>  <TextInput  style={styles.TextInput}  placeholder="Email"  placeholderTextColor="white"  onChangeText={(email) => setEmail(email)}  />  </View>  <View style={styles.inputView}>  <TextInput  style={styles.TextInput}  placeholder="Password"  placeholderTextColor="white"  secureTextEntry={true}  onChangeText={(password) => setPassword(password)}  />  </View> |

1. Add the following styles to the inputView and textInput props.

|  |
| --- |
| inputView: {  backgroundColor: "#3EB489",  borderRadius: 10,  width: "70%",  height: 45,  marginBottom: 10,  paddingLeft: 10,  paddingRight: 10,  alignItems: "left",  },  TextInput: {  height: 50,  }, |

**The next few steps are meant for discovery and experiment.**

1. Add the **Forgot Password?** button below the text input fields. We'll use a **TouchableOpacity** button, which changes opacity when pressed.
2. Style the Forget Password? Button.
3. Finally, add the Login button. Add the following code below the forgot password button.
4. Style the Login Button.
5. Customize the login page however you like.