Exercise (Instructions): Picking an Image

Objectives and Outcomes

In this exercise you will use the ImagePicker API from Expo SDK to access the camera to obtain a picture for use in our app. At the end of this exercise you will be able to:

- Configure your app to use the ImagePicker
- Obtain a picture from the Camera using the ImagePicker

Obtaining a Picture from the Camera

- Open LoginComponent.js and update its content as follows:

```
import React, { Component } from 'react';
import { View, StyleSheet, Text, ScrollView, Image } from 'react-native';
import { Input, CheckBox, Button, Icon } from 'react-native-elements';
import { SecureStore, Permissions, ImagePicker } from 'expo';
import { createBottomTabNavigator } from 'react-navigation';
import { baseUrl } from '../shared/baseUrl';

class LoginTab extends Component {

    constructor(props) {
        super(props);

        this.state = {
            username: '',
            password: '',
            remember: false
        }
    }

    componentDidMount() {
        SecureStore.getItemAsync('userinfo')
            .then((userdata) => {
                let userinfo = JSON.parse(userdata);
                if (userinfo) {
                    this.setState({username: userinfo.username});
                    this.setState({password: userinfo.password});
                    this.setState({remember: true})
                }
            })
    }

    static navigationOptions = {
        title: 'Login',
        tabBarIcon: ({ tintColor }) => (
            <Icon
              name='sign-in'
              type='font-awesome'
              size={24}
              iconStyle={{ color: tintColor }}
            />
          )
    };

    handleLogin() {
        console.log(JSON.stringify(this.state));
        if (this.state.remember)
            SecureStore.setItemAsync('userinfo', JSON.stringify({username: this.state.username, password: this.state.password}))
                .catch((error) => console.log('Could not save user info', error));
        else
            SecureStore.deleteItemAsync('userinfo')
                .catch((error) => console.log('Could not delete user info', error));

    }

    render() {
        return (
            <View style={styles.container}>
                <Input
                    placeholder="Username"
                    leftIcon={{ type: 'font-awesome', name: 'user-o' }}
                    onChangeText={(username) => this.setState({username})}
                    value={this.state.username}
                    containerStyle={styles.formInput}
                    />
                <Input
                    placeholder="Password"
                    leftIcon={{ type: 'font-awesome', name: 'key' }}
                    onChangeText={(password) => this.setState({password})}
                    value={this.state.password}
                    containerStyle={styles.formInput}
                    />
                <CheckBox title="Remember Me"
                    center
                    checked={this.state.remember}
                    onPress={() => this.setState({remember: !this.state.remember})}
                    containerStyle={styles.formCheckbox}
                    />
                <View style={styles.formButton}>
                    <Button
                        onPress={() => this.handleLogin()}
                        title="Login"
                        icon={
                            <Icon
                                name='sign-in'
                                type='font-awesome'
                                size={24}
                                color= 'white'
                            />
                        }
                        buttonStyle={{
                            backgroundColor: "#512DA8"
                        }}
                        />
                </View>
                <View style={styles.formButton}>
                    <Button
                        onPress={() => this.props.navigation.navigate('Register')}
                        title="Register"
                        clear
                        icon={
                            <Icon
                                name='user-plus'
                                type='font-awesome'
                                size={24}
                                color= 'blue'
                            />
                        }
                        titleStyle={{
                            color: "blue"
                        }}
                        />
                </View>
            </View>
        );
    }

}

class RegisterTab extends Component {

    constructor(props) {
        super(props);

        this.state = {
            username: '',
            password: '',
            firstname: '',
            lastname: '',
            email: '',
            remember: false,
            imageUrl: baseUrl + 'images/logo.png'
        }
    }

    getImageFromCamera = async () => {
        const cameraPermission = await Permissions.askAsync(Permissions.CAMERA);
        const cameraRollPermission = await Permissions.askAsync(Permissions
          .CAMERA_ROLL);

        if (cameraPermission.status === 'granted' && cameraRollPermission.status
          === 'granted') {
            let capturedImage = await ImagePicker.launchCameraAsync({
                allowsEditing: true,
                aspect: [4, 3],
            });
            if (!capturedImage.cancelled) {
                console.log(capturedImage);
                this.setState({imageUrl: capturedImage.uri });
            }
        }

    }

    static navigationOptions = {
        title: 'Register',
        tabBarIcon: ({ tintColor, focused }) => (
            <Icon
              name='user-plus'
              type='font-awesome'
              size={24}
              iconStyle={{ color: tintColor }}
            />
          )
    };

    handleRegister() {
        console.log(JSON.stringify(this.state));
        if (this.state.remember)
            SecureStore.setItemAsync('userinfo', JSON.stringify({username: this
              .state.username, password: this.state.password}))
                .catch((error) => console.log('Could not save user info', error
                  ));
    }

    render() {
        return(
            <ScrollView>
            <View style={styles.container}>
                <View style={styles.imageContainer}>
                    <Image
                        source={{uri: this.state.imageUrl}}
                        loadingIndicatorSource={require('./images/logo.png')}
                        style={styles.image}
                        />
                    <Button
                        title="Camera"
                        onPress={this.getImageFromCamera}
                        />
                </View>
                <Input
                    placeholder="Username"
                    leftIcon={{ type: 'font-awesome', name: 'user-o' }}
                    onChangeText={(username) => this.setState({username})}
                    value={this.state.username}
                    containerStyle={styles.formInput}
                    />
                <Input
                    placeholder="Password"
                    leftIcon={{ type: 'font-awesome', name: 'key' }}
                    onChangeText={(password) => this.setState({password})}
                    value={this.state.password}
                    containerStyle={styles.formInput}
                    />
                <Input
                    placeholder="First Name"
                    leftIcon={{ type: 'font-awesome', name: 'user-o' }}
                    onChangeText={(lastname) => this.setState({firstname})}
                    value={this.state.firstname}
                    containerStyle={styles.formInput}
                    />
                <Input
                    placeholder="Last Name"
                    leftIcon={{ type: 'font-awesome', name: 'user-o' }}
                    onChangeText={(lastname) => this.setState({lastname})}
                    value={this.state.lastname}
                    containerStyle={styles.formInput}
                    />
                <Input
                    placeholder="Email"
                    leftIcon={{ type: 'font-awesome', name: 'envelope-o' }}
                    onChangeText={(email) => this.setState({email})}
                    value={this.state.email}
                    containerStyle={styles.formInput}
                    />
                <CheckBox title="Remember Me"
                    center
                    checked={this.state.remember}
                    onPress={() => this.setState({remember: !this.state
                      .remember})}
                    containerStyle={styles.formCheckbox}
                    />
                <View style={styles.formButton}>
                    <Button
                        onPress={() => this.handleRegister()}
                        title="Register"
                        icon={
                            <Icon
                                name='user-plus'
                                type='font-awesome'
                                size={24}
                                color= 'white'
                            />
                        }
                        buttonStyle={{
                            backgroundColor: "#512DA8"
                        }}
                        />
                </View>
            </View>
            </ScrollView>
        );
    }
}

const styles = StyleSheet.create({
    container: {
        justifyContent: 'center',
        margin: 20,
    },
    imageContainer: {
        flex: 1,
        flexDirection: 'row',
        margin: 20
    },
    image: {
      margin: 10,
      width: 80,
      height: 60
    },
    formInput: {
        margin: 20
    },
    formCheckbox: {
        margin: 20,
        backgroundColor: null
    },
    formButton: {
        margin: 60
    }
});

const Login = createBottomTabNavigator({
    Login: LoginTab,
    Register: RegisterTab
}, {
    tabBarOptions: {
        activeBackgroundColor: '#9575CD',
        inactiveBackgroundColor: '#D1C4E9',
        activeTintColor: '#ffffff',
        inactiveTintColor: 'gray'
    }
});

export default Login;
```

- Then, open MainComponent.js and update it as follows:

```
. . .

  const LoginNavigator = createStackNavigator({
      Login: Login
    }, {
    navigationOptions: ({ navigation }) => ({
      headerStyle: {
          backgroundColor: "#512DA8"
      },
      headerTitleStyle: {
          color: "#fff"
      },
      title: 'Login',
      headerTintColor: "#fff",
      headerLeft: <Icon name="menu" size={24}
        iconStyle={{ color: 'white' }}
        onPress={ () => navigation.toggleDrawer() } />
    })
  });

. . .
```

- Save all the changes and do a Git commit with the message "Picking an Image".

Conclusions

In this exercise we use the ImagePicker API to obtain an image from the camera.

Mark as completed