# Chapter 2

# REVIEW OF THE RELATED LITERATURE AND STUDIES

import {

Text,

View,

ScrollView,

SafeAreaView,

Image,

} from 'react-native';

import React, { useState } from 'react';

import { router, Link } from 'expo-router';

import InputBox from '../../components/inputBox';

import CustomBtn from '../../components/button';

import { supabase } from '../../lib/supabase';

const SignIn = () => {

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

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

const [emailError, setEmailError] = useState({ condition: false, message: '' });

const [passwordError, setPasswordError] = useState({ condition: false, message: '' });

const [isLoading, setIsLoading] = useState(false);

const validateEmail = (email) => {

if (email.trim() === '' || (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email))){

setEmailError({ condition: true, message: 'Field is empty' });

} else {

setEmailError({ condition: false, message: '' });

}

};

const validatePassword = (password) => {

if (password.trim() === '') {

setPasswordError({ condition: true, message: 'Field is empty' });

} else {

setPasswordError({ condition: false, message: '' });

}

};

const signInWithEmail = async () => {

validateEmail(email);

validatePassword(password);

if (emailError.condition || passwordError.condition) {

return; // Stop if there are validation errors

}

setIsLoading(true);

try {

const { error } = await supabase.auth.signInWithPassword({

email,

password,

});

if (error) {

console.log('Sign In failed:', error.message);

setEmailError({ condition: true, message: "Email or Password is Incorrect"});

} else {

console.log("Sign in Successful");

router.push('../(tabs)/home');

}

} catch (e) {

console.log("Sign in failed due to:", e.message);

setEmailError({ condition: true, message: e.message });

} finally {

setIsLoading(false);

}

};

return (

<View>

<SafeAreaView className="bg-white h-full">

<ScrollView className="bg-white">

<View className="flex-1 items-center bg-white">

<View className="w-80 flex-row justify-center mt-32 mb-[40px]">

<View className="w-2/5 h-mx">

<Image className="w-24 h-24" source={require('../../assets/icon/redcross.png')} />

</View>

<View className="flex-1 h-mx ml-1">

<Text className="text-4xl text-customgray font-interbold">Philippine Red Cross</Text>

<Text className="text-base text-customgray font-interbold">Muntinlupa Branch</Text>

</View>

</View>

<View className="w-auto h-auto">

<InputBox

keyboardType="email-address"

value={email}

onChangeText={(emailVal) => {

setEmail(emailVal);

validateEmail(emailVal);

}}

title="Enter Email here"

borderWidth={emailError.condition ? 2 : 1}

borderColor={emailError.condition ? 'red' : '#D9D9D9'}

message={emailError}

/>

<InputBox

value={password}

onChangeText={(passwordVal) => {

setPassword(passwordVal);

validatePassword(passwordVal);

}}

title="Enter Password here"

borderWidth={passwordError.condition ? 2 : 1}

borderColor={passwordError.condition ? 'red' : '#D9D9D9'}

secureTextEntry

autoCapitalize="none"

message={passwordError}

customize={{ display: "none" }}

/>

<CustomBtn

title="Sign In"

onPress={signInWithEmail}

disable={isLoading}

isLoading={isLoading}

/>

<View className="m-[25]">

<Text className="color-customgray text-sm mx-auto">Don't have an Account?

<Link href={'./Email'}>

<Text className="font-interbold color-primaryRed"> Sign up</Text>

</Link>

</Text>

</View>

</View>

</View>

</ScrollView>

</SafeAreaView>

</View>

);

};

export default SignIn;