**Cam one project – REACT NATIVE**

**Cam-one project**

**loginPage.js**

import { View,Text, Button, TextInput, TouchableOpacity } from "react-native";

import { StyleSheet } from "react-native";

export default function MyLogin(){

    return(

        <View style={style.container}>

            <Text style={style.text}>

                Login

            </Text>

            <TextInput style={style.textInput} placeholder="Username"/>

            <TextInput style={style.textInput} placeholder="Password"secureTextEntry/>

            <TouchableOpacity title='login'style={style.button}>

                <Text style={style.buttonText}> Login </Text>

            </TouchableOpacity>

        </View>

    );

}

const style = StyleSheet.create({

    container:

    {

        alignItems:'center',

        justifyContent:'center',

        flex:1,

        margin: 10,

    },

    button:

    {

        borderColor:'red',

        borderWidth:5,

        backgroundColor:'red',

        borderRadius: 30,

        alignItems:'center',

        justifyContent:'center',

    },

    buttonText: {

        fontSize:20,

        color:'white',

        textAlign:'center',

    },

    text: {

     color:'black',

     fontWeight: 'bold',

     fontSize: 30,

     textAlign:'center',

     marginVertical: 10,

    },

    textInput: {

        borderColor: 'gray',

        width: 200,

        padding: 5,

        borderWidth: 1,

        marginVertical: 10,

        borderRadius: 5,

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

export default function App() {

  return (

    <View style={styles.container}>

      <MyLogin/>

    </View>

    // <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',

    width: '100%',

  },

});

**TouchableOpacity**

xport default function MyLogin(){

    return(

        <View style={style.container}>

            <Text style={style.text}>

                Login

            </Text>

            <TextInput style={style.textInput} placeholder="Username"/>

            <TextInput style={style.textInput} placeholder="Password"secureTextEntry/>

            <TouchableOpacity title='login'style={style.button}>

                <Text style={style.buttonText}> Login </Text>

            </TouchableOpacity>

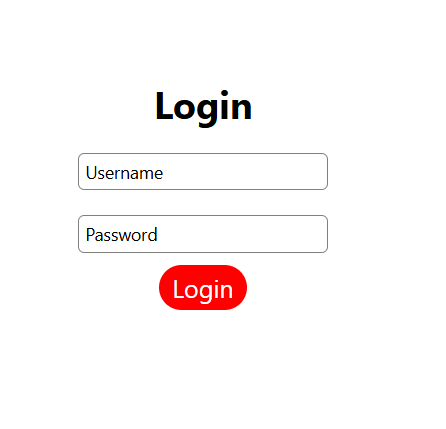
        </View>

    );

}

Here why we are using touchableOpacity instead of button tag is, we cannot change the color of the button, which <button/> component does not support the advanced stylings.

**OUTPUT**



**loginPage.js**

import { View,Text, Button, TextInput, TouchableOpacity } from "react-native";

import { StyleSheet } from "react-native";

export default function MyLogin(){

    return(

        <View style={style.container}>

            <Text style={style.text}>

                Login

            </Text>

            <View style = {style.label}>

            <Text>Username</Text>

            <TextInput style={style.textInput} placeholder="Username" placeholderTextColor={"gray"}/>

            </View>

            <View style = {style.label}>

            <Text>Password</Text>

            <TextInput style={style.textInput} placeholder="Password" placeholderTextColor={"gray"} secureTextEntry/>

            </View>

            <View style={style.labelforgotpass}>

            <Text>Forgot Password</Text>

            </View>

            <View style= {style.loginright}>

            <TouchableOpacity title='login'style={style.button}>

                <Text style={style.buttonText}> Login </Text>

            </TouchableOpacity>

            </View>

        </View>

    );

}

const style = StyleSheet.create({

    container:

    {

        alignItems:'center',

        justifyContent:'center',

        flex:1,

        margin: 10,

    },

    labelforgotpass:{

        alignItems:'flex-end',

        marginVertical: 5,

        width: 200,

    },

    label:

    {

        marginVertical: 5,

        alignItems:'flex-start',

    },

    loginright:{

        width:200,

        alignItems:"flex-end",

    },

    button:

    {

        borderColor:'gray',

        borderWidth:5,

        backgroundColor:'gray',

        borderRadius: 30,

        alignItems:'flex-end',

        justifyContent:'flex-end',

        flex:1,

        paddingHorizontal: 15,

        paddingVertical: 3,

    },

    buttonText: {

        color:'white',

        textAlign:'center',

    },

    text: {

     color:'black',

     fontWeight: 'bold',

     fontSize: 30,

     textAlign:'center',

     marginVertical: 10,

    },

    textInput: {

        borderColor: 'gray',

        width: 200,

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

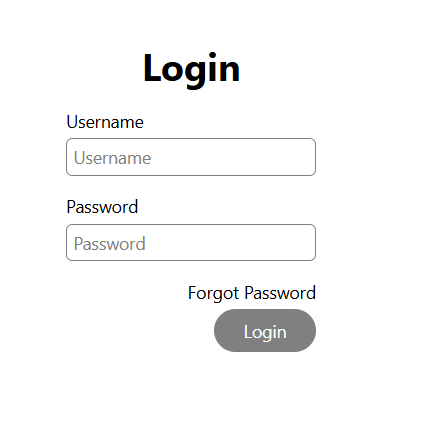
        borderRadius: 5,

    },

});

Here we should wrap the text content into <View> component if we want to move that to left or right. And we should also mention same width to move the content based on another one. For example, here I want to move the login button to the end which aligned with the <textInput>. So I mentioned that the width is 200 for both textInput and login button.

If we apply styles to the particular text without wrapping, it does not consider textInput alignment so there will be a issue in moving right or left. If we use <View> component then it will be consider the whole stuff as single unit.(username placeholder and username label)



**Going to change the color of the stuffs**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

export default function MyLogin() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    return (

        <View style={styles.container}>

            <Text style={styles.text}>Login</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    // onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button}>

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'maroon',

        borderWidth: 5,

        backgroundColor: 'maroon',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        width: '200',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginVertical: 10,

    },

    textInput: {

        borderColor: 'gray', // Default border color

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon', // Border color when focused or filled

    },

});

This line defines a functional component named MyLogin and exports it as the default export of the module.

const [focusedInput, setFocusedInput] = useState(null);

* This line uses the useState hook to create a state variable called focusedInput. focusedInput is initialized to null.
* setFocusedInput is a function used to update the focusedInput state.
* This state variable tracks which input field (username or password) is currently focused.

const [username, setUsername] = useState('');

* This line uses the useState hook to create a state variable called username.
* username is initialized to an empty string ('').
* setUsername is a function used to update the username state.
* This state variable stores the text entered in the username input field.

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

* This line uses the useState hook to create a state variable called password.
* password is initialized to an empty string ('').
* setPassword is a function used to update the password state.
* This state variable stores the text entered in the password input field.

<TextInput

style={[

styles.textInput,

(focusedInput === 'username' || username) && styles.textInputFocused

]}

 This line renders a TextInput component.

 The style prop applies an array of styles:

* styles.textInput for the default input field style.
* styles.textInputFocused conditionally if focusedInput is 'username' or if username is not empty.

onFocus={() => setFocusedInput('username')}

The onFocus prop sets focusedInput to 'username' when the input field gains focus.

onChangeText={text => setUsername(text)}

value={username}

 The onChangeText prop updates the username state with the entered text.

 The value prop sets the input field's value to the username state.

**CODE**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

export default function MyLogin() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>Login</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 5,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        // width: 200,

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginVertical: 10,

    },

    textInput: {

        borderColor: 'gray',

        width: 200,

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

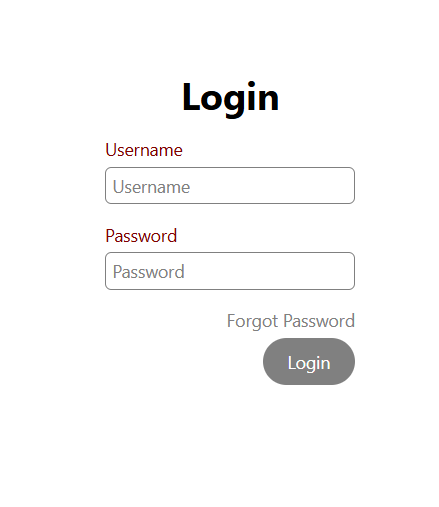
    textInputFocused: {

        borderColor: 'maroon',

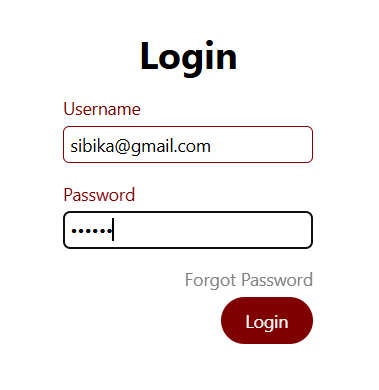
    },

});

**OUTPUT 1**



**OUTPUT 2**



const isFormFilled = username && password;

 **Purpose**: This line defines a variable isFormFilled that checks whether both username and password fields are non-empty.

 **Logic**: In JavaScript, the && operator returns the second operand (password in this case) if the first operand (username) is truthy. If username and password both have non-empty values, isFormFilled will be true. Otherwise, it will be false.

<TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

<Text style={styles.buttonText}>Login</Text>

</TouchableOpacity>

 **Purpose**: This is a React Native TouchableOpacity component used as a button. The button’s style changes conditionally based on the value of isFormFilled.

 **Style Prop Explanation**:

* styles.button: This is the default style applied to the button.
* isFormFilled && styles.buttonFilled: This is a conditional style. If isFormFilled is true, it will apply the styles.buttonFilled style in addition to styles.button. If isFormFilled is false, only styles.button will be applied.

LoginPage.js

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

export default function MyLogin() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {<Text style={styles.text}>LOGO</Text>

            /\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            {/\* <Text style={styles.text}>Login</Text> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text>If you are new to this? Register here</Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200, // Adjust the width as needed

        height: 200, // Adjust the height as needed

        marginBottom: 20, // Add some space below the image

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginVertical: 130,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    registerContainer: {

        width: '100%',

        justifyContent: "flex-end",

        flex: 1,

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

    icon: {

        marginHorizontal: 10,

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

export default function App() {

  return (

    <View style={styles.container}>

    <MyLogin/>

    </View>

    // <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',

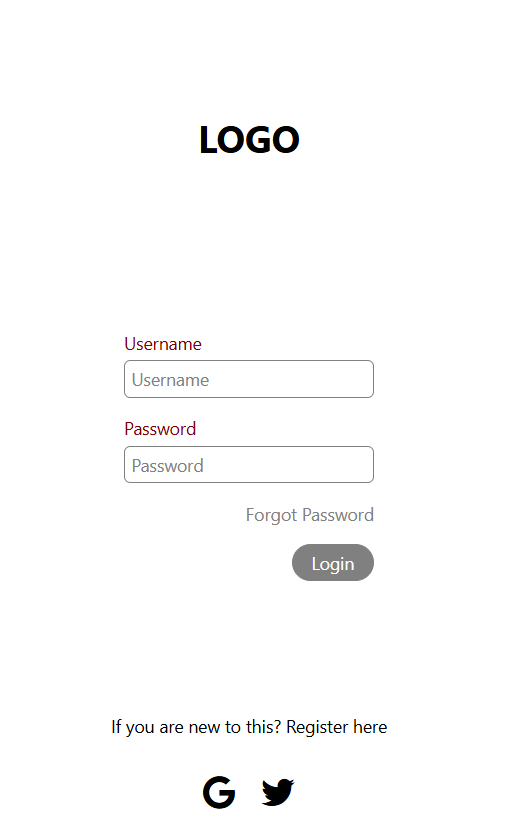
    justifyContent: 'center',

    width: '100%',

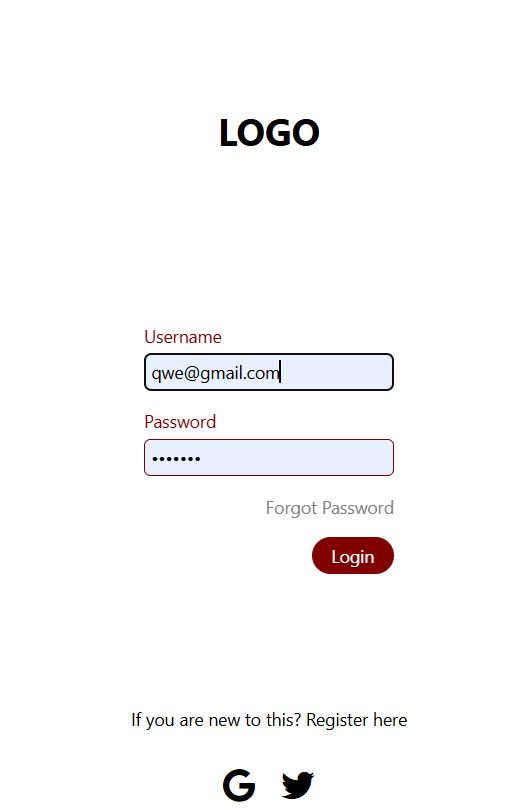
  },

});

**OUTPUT 1**



**OUTPUT 2**



**RegisterPage.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

export default function MyReg() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Name</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

export default function App() {

  return (

    <View style={styles.container}>

    <MyReg/>

    </View>

    // <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',

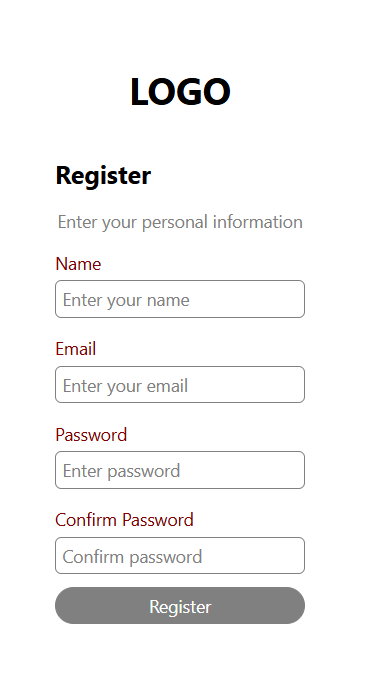
    justifyContent: 'center',

    width: '100%',

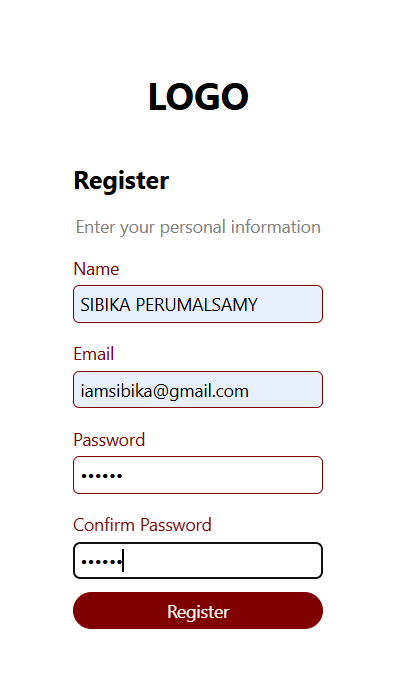
  },

});

**OUTPUT 1**



**OUTPUT 2**



**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

export default function MyForPass() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const isFormFilled = username;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>Don't worry! It occurs enter the email

                 address linked with your account</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Send Code</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

        width:200,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 300,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width:200,

        marginTop:10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

export default function App() {

  return (

    <View style={styles.container}>

    <MyForPass/>

    </View>

    // <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',

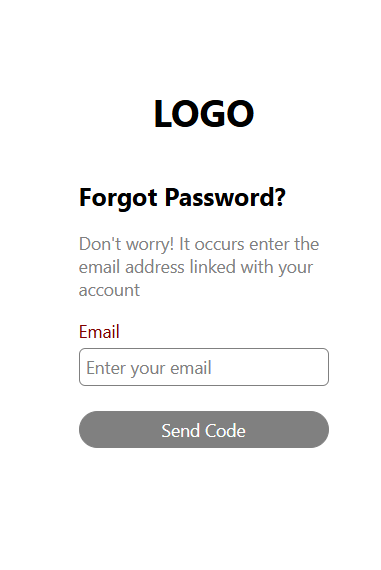
    justifyContent: 'center',

    width: '100%',

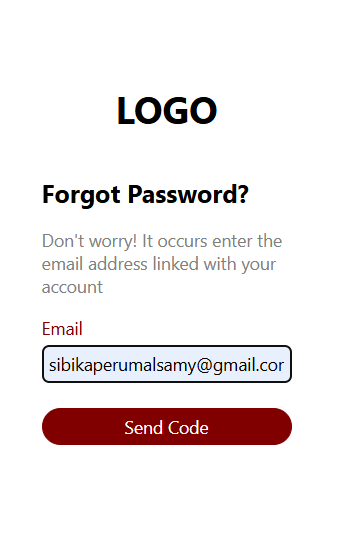
  },

});

**OUTPUT 1**



**OUTPUT-2**



**OTPVerification.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

export default function MyOtp() {

    // Initialize an array with 5 empty strings for OTP inputs

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        // Focus on the next input field if current one is filled

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        // Focus on the previous input field if text is deleted

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            // Assuming you have some verification logic here

            console.log('OTP Verified:', otp.join(''));

            navigation.navigate('CreateNewPass');

        } else {

            alert('Please enter the complete OTP.');

        }

    };

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>OTP Verification</Text>

            </View>

            <Text style={styles.description}>

                Enter the verification code we just sent to your email address

            </Text>

            <View style={styles.inputContainer}>

                {otp.map((value, index) => (

                    <TextInput

                        key={index}

                        style={[

                            styles.textInput,

                            (focusedInput === index || value) && styles.textInputFocused

                        ]}

                        placeholder=""

                        placeholderTextColor="gray"

                        onFocus={() => setFocusedInput(index)}

                        onChangeText={text => handleChange(text, index)}

                        value={value}

                        maxLength={1} // Only allow one character

                        keyboardType="number-pad" // Numeric keyboard for OTP

                    />

                ))}

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleVerify}

                >

                    <Text style={styles.buttonText}>Verify</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

        width: 200,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    description: {

        color: 'gray',

        marginBottom: 10,

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: 200,

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: 200,

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyOtp from './Pages/otpVerification';

export default function App() {

  return (

    <View style={styles.container}>

    <MyOtp/>

    </View>

    // <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',

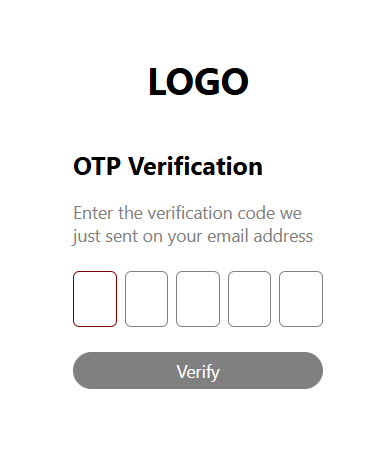
    justifyContent: 'center',

    width: '100%',

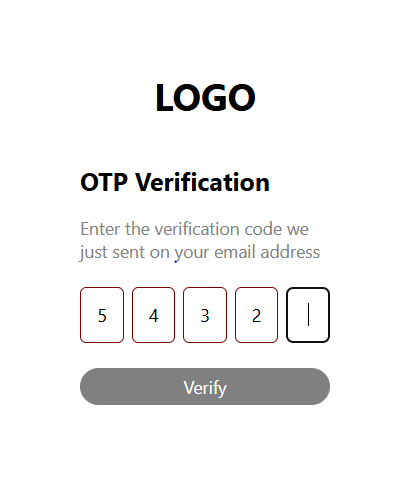
  },

});

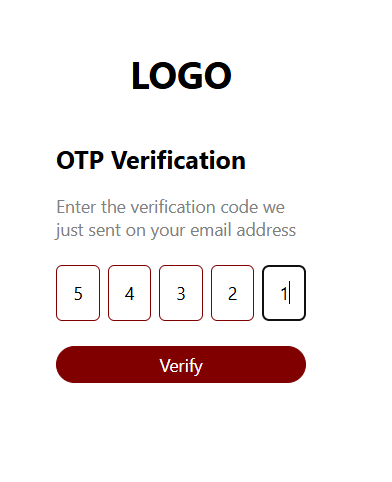
**OUTPUT 1**



**OUTPUT 2**



**OUTPUT 3**



**CreateNewPass.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

export default function MyCreNewPass() {

    const [focusedInput, setFocusedInput] = useState(null);

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

    const [conpassword, setConPassword] = useState('');

    const isFormFilled =   password  && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Create new password</Text>

                <Text style={styles.description}>Your new password must be unique from

                    those previously used

                </Text>

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>New password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Reset password</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop:10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyOtp from './Pages/otpVerification';

import MyCreNewPass from './Pages/createNewPassword';

export default function App() {

  return (

    <View style={styles.container}>

    <MyCreNewPass/>

    </View>

    // <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',

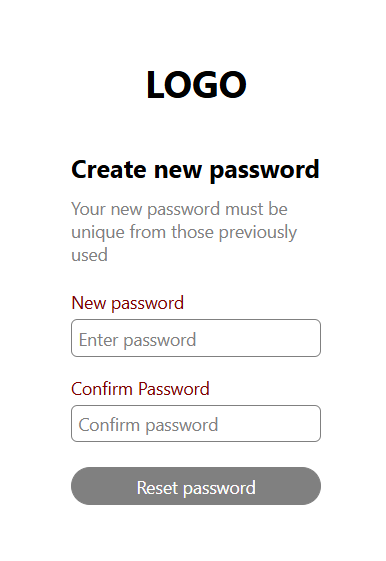
    justifyContent: 'center',

    width: '100%',

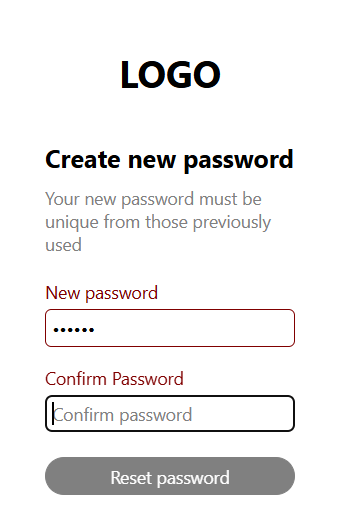
  },

});

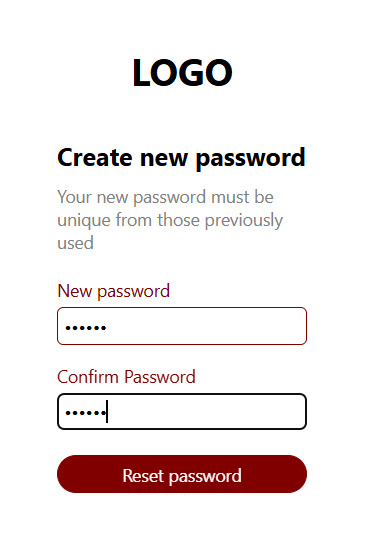
**OUTPUT -1**



**OUTPUT 2**



**OUTPUT 3**



**Dashboard.js**

import React from "react";

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

export default function MyDashboard() {

    const handleLogout = () => {

        // Implement your logout functionality here

        console.log("User logged out");

    };

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Dashboard for camera monitoring app</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button} onPress={handleLogout}>

                    <Text style={styles.buttonText}>Logout</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'maroon',

        borderWidth: 1,

        backgroundColor: 'maroon',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyOtp from './Pages/otpVerification';

import MyCreNewPass from './Pages/createNewPassword';

import MyDashboard from './Pages/dashboard';

export default function App() {

  return (

    <View style={styles.container}>

    <MyDashboard/>

    </View>

    // <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',

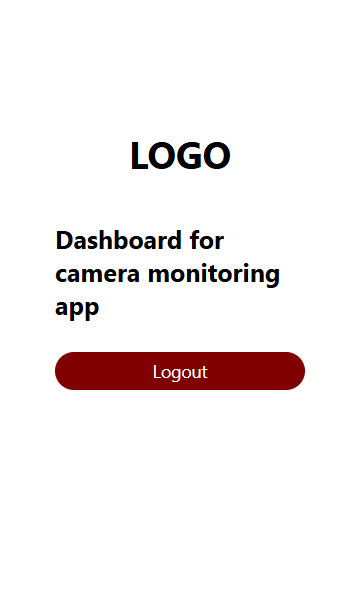
    justifyContent: 'center',

    width: '100%',

  },

});

**OUTPUT**



**registerPage.js after FIREBASE INTEGRATION**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

import { createUserWithEmailAndPassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyReg() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const [error, setError] = useState('');

    const handleRegister = () => {

        if (password !== conpassword) {

            setError('Passwords do not match');

            return;

        }

        createUserWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                console.log(user);

                setError(''); // Clear the error message upon successful registration

            })

            .catch((error) => {

                console.error(error);

                setError(error.message);

            });

    };

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Name</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleRegister}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**FirebaseAuth.js**

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { initializeAuth } from "firebase/auth";

// TODO: Add SDKs for Firebase products that you want to use

// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration

const firebaseConfig = {

  apiKey: "AIzaSyBI6cCSoO7ngINvfFc6KPrnsT2V2ApFB1o",

  authDomain: "cam-one-1.firebaseapp.com",

  projectId: "cam-one-1",

  storageBucket: "cam-one-1.appspot.com",

  messagingSenderId: "1062546701892",

  appId: "1:1062546701892:web:015b4b9bd2f7b7d8d98611"

};

let auth;

// Initialize Firebase

const app = initializeApp(firebaseConfig);

auth = initializeAuth(app)

export default auth;

**app.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import OtpVerification from './Pages/otpVerification';

import MyOtp from './Pages/otpVerification';

import MyCreNewPass from './Pages/CreNewPassword';

import MyDashboard from './Pages/Dashboard';

export default function App() {

  return (

    <View style={styles.container}>

    <MyReg/>

    </View>

    // <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',

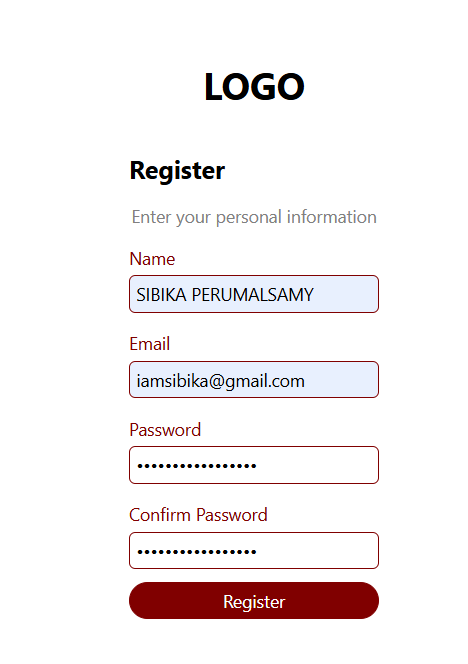
    justifyContent: 'center',

    width: '100%',

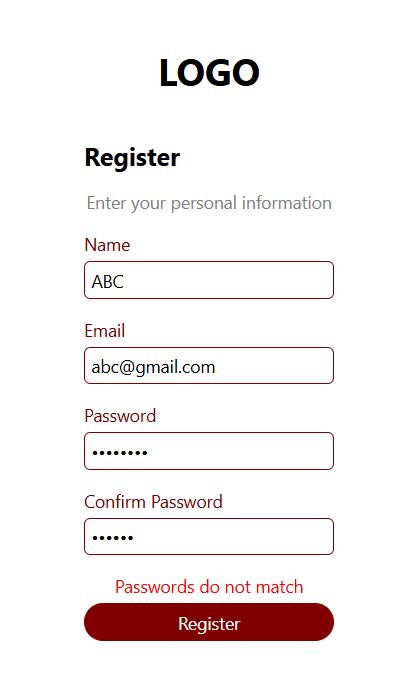
  },

});

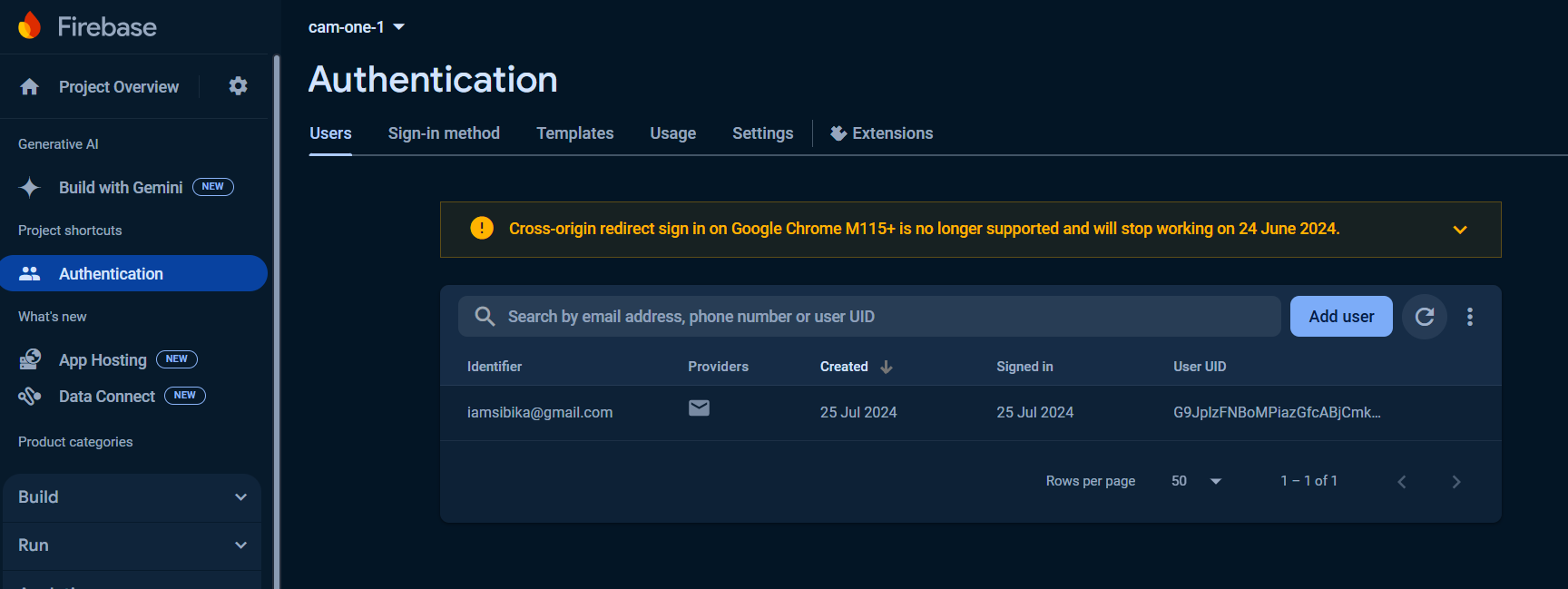
**OUTPUT**



**Error ( Passwords do not match)**



**In firebase authentication,**



**LoginPage.js**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                console.log(user);

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            {/\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text>If you are new to this? Register here</Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200,

        height: 200,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

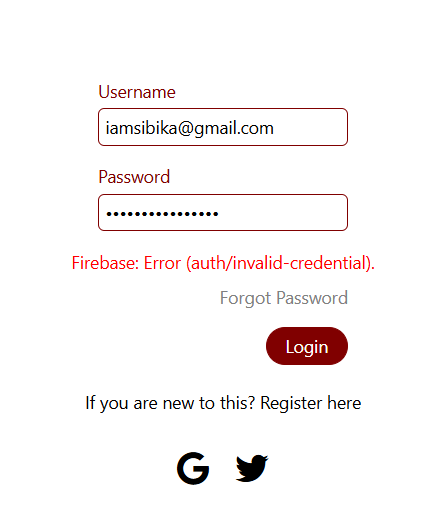
    icon: {

        marginHorizontal: 10,

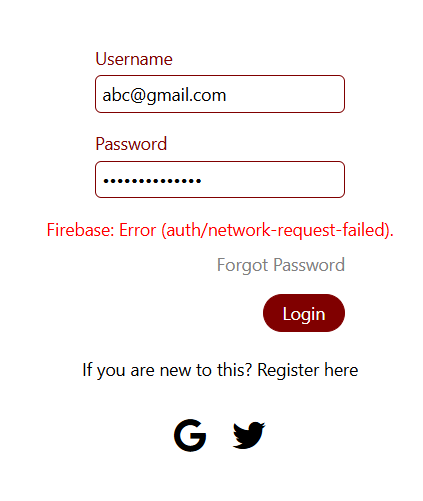
    },

});

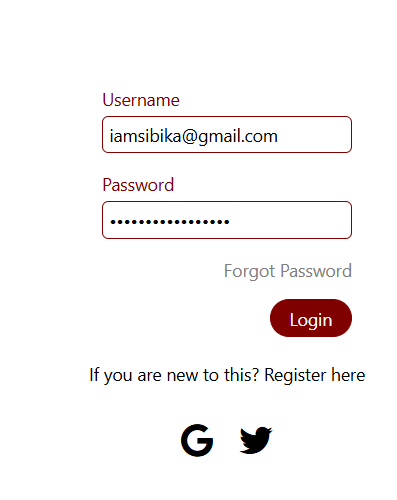
**OUTPUT FOR INVALID CREDENTIAL (ERROR)**



**Output -2 (one who didn’t have an account)**



**Output-3 (LoggedIn)**



Otherwise, it will be logged in!

**CREATE NEW PASSWORD**

**createNewPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import { updatePassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyCreNewPass() {

    const [focusedInput, setFocusedInput] = useState(null);

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

    const [conpassword, setConPassword] = useState('');

    const [errorMessage, setErrorMessage] = useState('');

    const handleResetPassword = () => {

        if (password !== conpassword) {

            setErrorMessage('Passwords do not match');

            return;

        }

        const user = auth.currentUser;

        updatePassword(user, password).then(() => {

            setErrorMessage(''); // Clear the error message on success

            Alert.alert('Success', 'Password updated successfully');

        }).catch((error) => {

            setErrorMessage(error.message);

        });

    };

    const isFormFilled = password && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Create new password</Text>

                <Text style={styles.description}>

                    Your new password must be unique from those previously used

                </Text>

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>New password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {errorMessage ? (

                <Text style={styles.errorText}>{errorMessage}</Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleResetPassword}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Reset password</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    errorText: {

        color: 'red',

        marginVertical: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

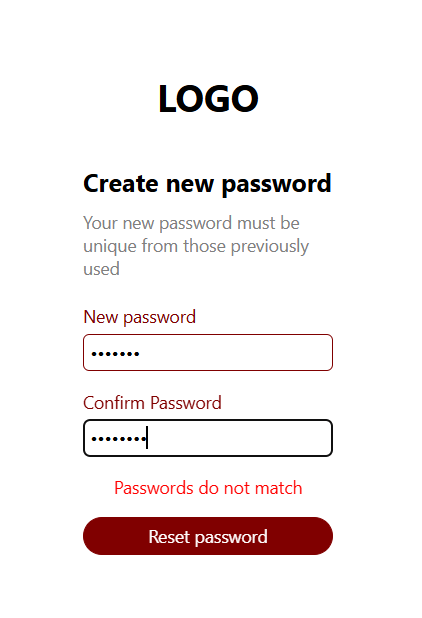
        color: 'white',

        textAlign: 'center',

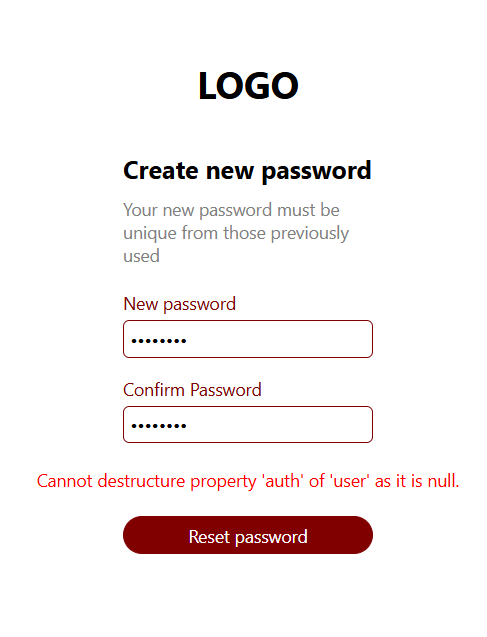
    },

});

**OUTPUT**



**OUTPUT 2**



**Forgot Password.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

export default function MyForPass() {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const isFormFilled = username;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>Don't worry! It occurs enter the email

                 address linked with your account</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText}>Send Code</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

        width:200,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 300,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width:200,

        marginTop:10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

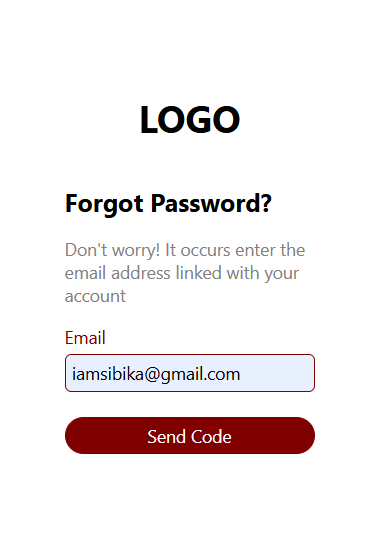
        color: 'white',

        textAlign: 'center',

    },

});

**OUTPUT**



**OtpVerification.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert } from "react-native";

export default function MyOtp({ navigation }) {

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            const otpCode = otp.join('');

            // Assuming you have an API endpoint for OTP verification

            fetch('https://your-backend-api.com/verify-otp', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ otp: otpCode }),

            })

                .then(response => response.json())

                .then(data => {

                    if (data.success) {

                        Alert.alert('Success', 'OTP Verified');

                        navigation.navigate('CreateNewPass');

                    } else {

                        Alert.alert('Error', 'Invalid OTP');

                    }

                })

                .catch(error => {

                    console.error('Error:', error);

                    Alert.alert('Error', 'Something went wrong. Please try again.');

                });

        } else {

            Alert.alert('Error', 'Please enter the complete OTP.');

        }

    };

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>OTP Verification</Text>

            </View>

            <Text style={styles.description}>

                Enter the verification code we just sent to your email address

            </Text>

            <View style={styles.inputContainer}>

                {otp.map((value, index) => (

                    <TextInput

                        key={index}

                        style={[

                            styles.textInput,

                            (focusedInput === index || value) && styles.textInputFocused

                        ]}

                        placeholder=""

                        placeholderTextColor="gray"

                        onFocus={() => setFocusedInput(index)}

                        onChangeText={text => handleChange(text, index)}

                        value={value}

                        maxLength={1} // Only allow one character

                        keyboardType="number-pad" // Numeric keyboard for OTP

                    />

                ))}

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleVerify}

                >

                    <Text style={styles.buttonText}>Verify</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

        width: 200,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    description: {

        color: 'gray',

        marginBottom: 10,

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: 200,

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: 200,

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

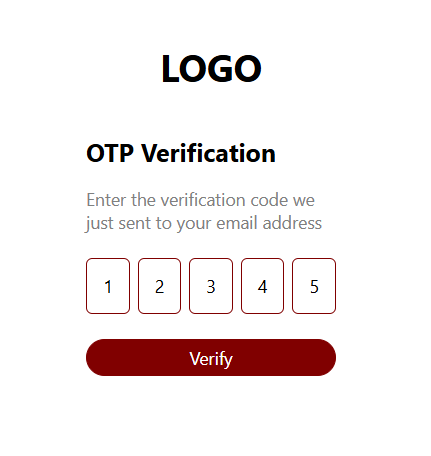
        color: 'white',

        textAlign: 'center',

    },

});

**OUTPUT**



**MyDashboard.js**

// MyDashboard.js

import React from 'react';

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

import { signOut } from 'firebase/auth';

import auth from '../Services/FirebaseAuth'; // Adjust path based on your folder structure

export default function MyDashboard({ navigation }) {

    const handleLogout = () => {

        signOut(auth)

            .then(() => {

                // Sign-out successful.

                console.log('User logged out');

                // Redirect or navigate to login screen after logging out

                navigation.navigate('Login'); // Replace 'Login' with your login screen route

            })

            .catch((error) => {

                // An error happened.

                console.error('Sign Out Error', error);

            });

    };

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Dashboard for Camera Monitoring App</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button} onPress={handleLogout}>

                    <Text style={styles.buttonText}>Logout</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1, // Ensure it takes up the full screen

        justifyContent: 'center',

        alignItems: 'center',

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'maroon',

        borderWidth: 1,

        backgroundColor: 'maroon',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonText: {

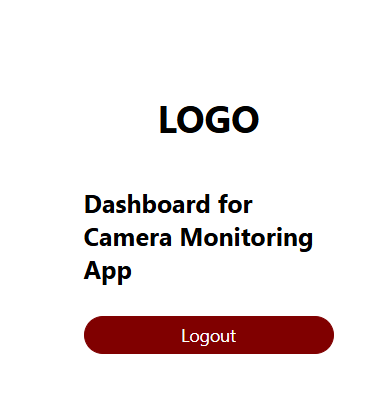
        color: 'white',

        textAlign: 'center',

    },

});

**OUTPUT**



**NAVIGATE REGISTER SCREEN FROM LOGIN SCREEN**

**LoginPage.js**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                console.log(user);

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')// to navigate from register here to registerpage (accessed in below lines)

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            {/\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text onPress={goToRegister}>If you are new to this? Register here</Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200,

        height: 200,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

    icon: {

        marginHorizontal: 10,

    },

});

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyOtp from './Pages/otpVerification';

import MyCreNewPass from './Pages/CreNewPassword';

import MyDashboard from './Pages/Dashboard';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import { NavigationContainer } from '@react-navigation/native';

const Stack = createNativeStackNavigator();

export default function App() {

  return (

    // <View>

    //   <navigation/>

    // </View>

   <NavigationContainer>

    <Stack.Navigator initialRouteName='Login'>

      <Stack.Screen name='Register' component={MyReg}/>

      <Stack.Screen name='Login' component={MyLogin}/>

      <Stack.Screen name='ForgotPassword' component={MyForPass}/>

      <Stack.Screen name='OTPverification' component={MyOtp}/>

      <Stack.Screen name='CreateNewPassword' component={MyCreNewPass}/>

      <Stack.Screen name='Dashboard' component={MyDashboard}/>

    </Stack.Navigator>

   </NavigationContainer>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

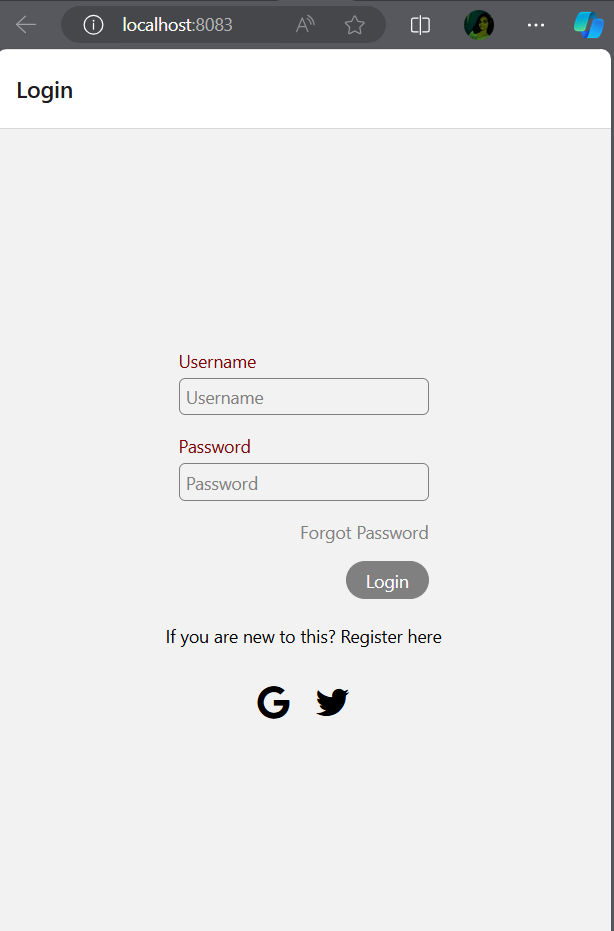
    width: '100%',

  },

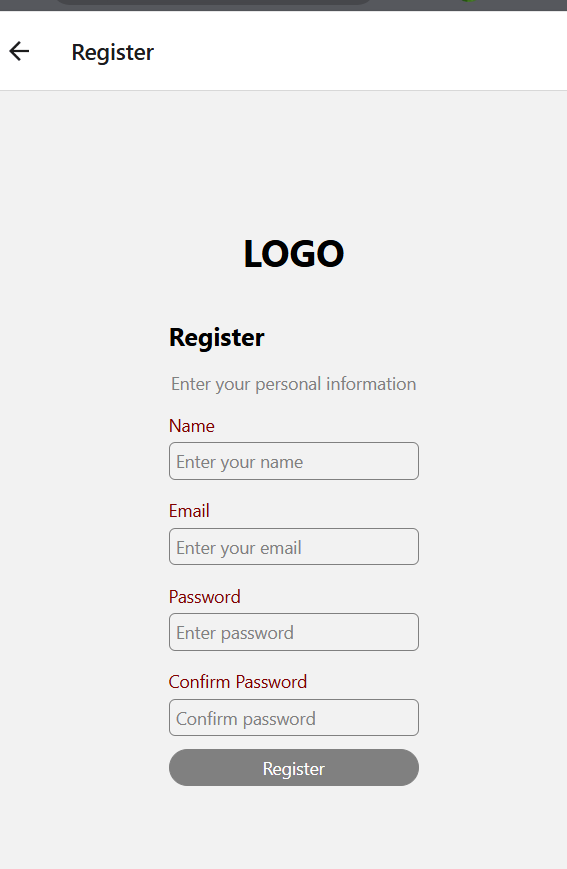
});

Initially, it shows the login page and after that if we click in register here mean, it will go to the register page.

**OUTPUT 1**



**OUTPUT 2**



**LoginPage.js**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                console.log(user);

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')

    }

    const goToForPass = () => {

        navigation.navigate('ForgotPassword')

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            {/\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText} onPress={goToForPass}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text onPress={goToRegister}>If you are new to this? Register here</Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200,

        height: 200,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

    icon: {

        marginHorizontal: 10,

    },

});

If we wanna redirect to forgot password

**BUTTON NAVIGATION**

**Register to dashboard**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

import { createUserWithEmailAndPassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyReg({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const [error, setError] = useState('');

    const handleRegister = () => {

        if (password !== conpassword) {

            setError('Passwords do not match');

            return;

        }

        createUserWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

               navigation.navigate('Dashboard')

                setError(''); // Clear the error message upon successful registration

            })

            .catch((error) => {

                console.error(error);

                setError(error.message);

            });

    };

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Name</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleRegister}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

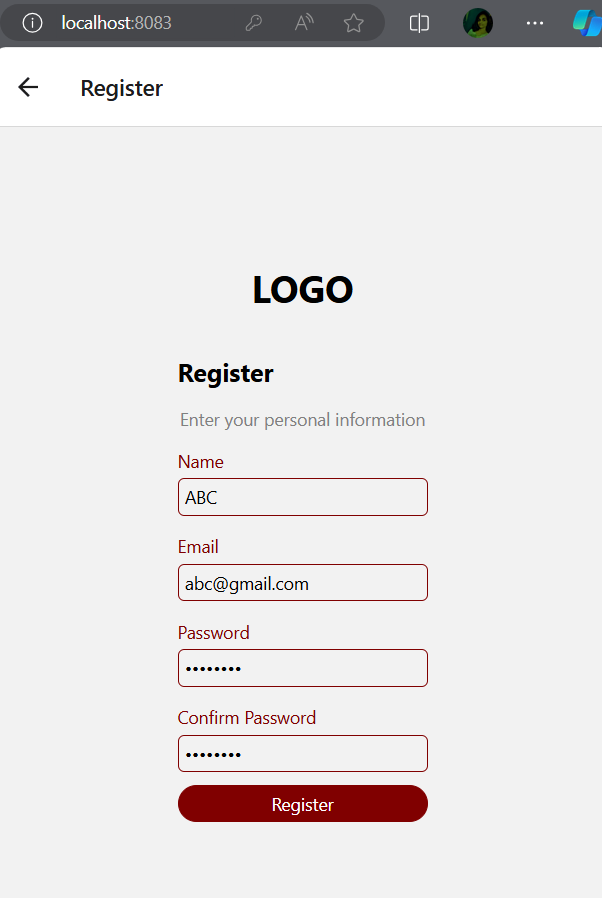
        color: 'white',

        textAlign: 'center',

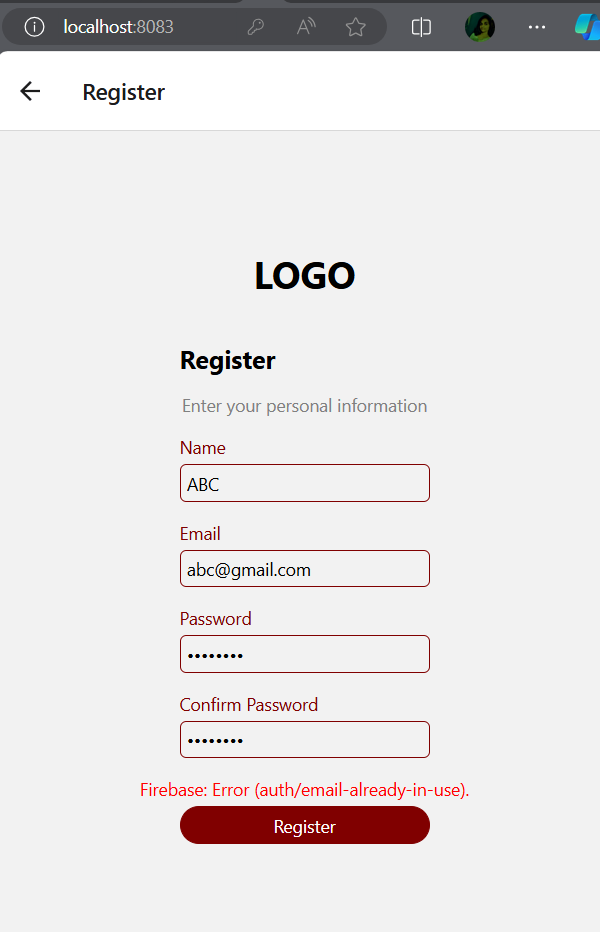
    },

});

**OUTPUT 1**



**Once I clicked to register, then the page will be redirect to dashboard page.**



If you trying to register using same mail.

**LOGIN TO DASHBOARD**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                navigation.navigate('Dashboard')

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')

    }

    const goToForPass = () => {

        navigation.navigate('ForgotPassword')

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            {/\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText} onPress={goToForPass}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text onPress={goToRegister}>If you are new to this? Register here</Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200,

        height: 200,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

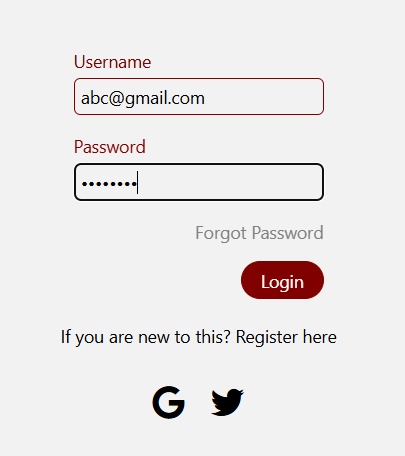
    icon: {

        marginHorizontal: 10,

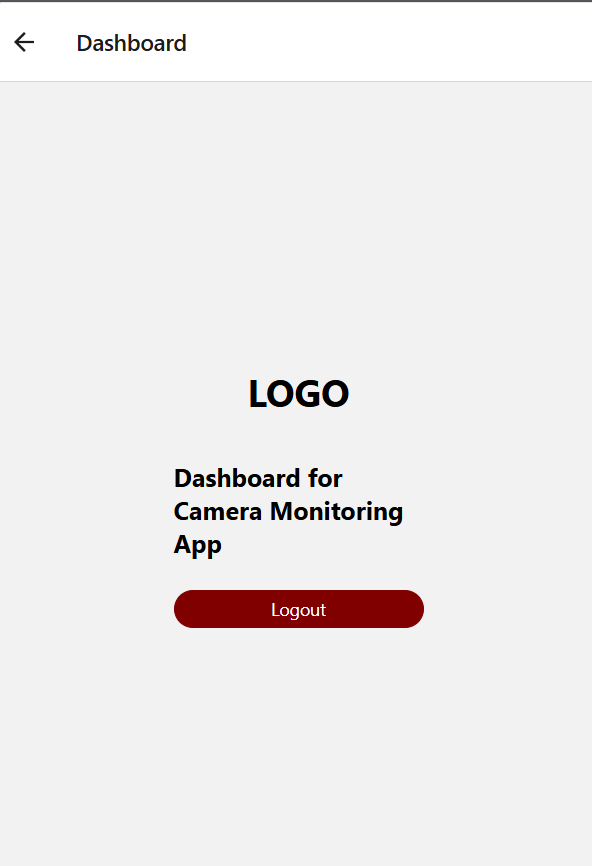
    },

});

**OUTPUT 1**



**OUTPUT 2**



**FORGOT PASSWORD TO OTP VERIFICATION**

**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert } from "react-native";

export default function MyOtp({ navigation }) {

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            const otpCode = otp.join('');

            // Assuming you have an API endpoint for OTP verification

            fetch('https://your-backend-api.com/verify-otp', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ otp: otpCode }),

            })

                .then(response => response.json())

                .then(data => {

                    if (data.success) {

                        Alert.alert('Success', 'OTP Verified');

                        navigation.navigate('CreateNewPass');

                    } else {

                        Alert.alert('Error', 'Invalid OTP');

                    }

                })

                .catch(error => {

                    console.error('Error:', error);

                    Alert.alert('Error', 'Something went wrong. Please try again.');

                });

        } else {

            Alert.alert('Error', 'Please enter the complete OTP.');

        }

    };

    return (

        <View style={styles.container}>

            <View style={styles.contentContainer}>

                <Text style={styles.text}>LOGO</Text>

                <View style={styles.headerContainer}>

                    <Text style={styles.registerHeading}>OTP Verification</Text>

                </View>

                <Text style={styles.description}>

                    Enter the verification code we just sent to your email address

                </Text>

                <View style={styles.inputContainer}>

                    {otp.map((value, index) => (

                        <TextInput

                            key={index}

                            style={[

                                styles.textInput,

                                (focusedInput === index || value) && styles.textInputFocused

                            ]}

                            placeholder=""

                            placeholderTextColor="gray"

                            onFocus={() => setFocusedInput(index)}

                            onChangeText={text => handleChange(text, index)}

                            value={value}

                            maxLength={1} // Only allow one character

                            keyboardType="number-pad" // Numeric keyboard for OTP

                        />

                    ))}

                </View>

                <View style={styles.loginRight}>

                    <TouchableOpacity

                        style={[styles.button, isFormFilled && styles.buttonFilled]}

                        onPress={handleVerify}

                    >

                        <Text style={styles.buttonText}>Verify</Text>

                    </TouchableOpacity>

                </View>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

    },

    contentContainer: {

        width: 200,

        alignItems: 'center',

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: '100%',

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        marginBottom: 10,

        textAlign: 'left',

        width: '100%',

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: '100%',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    loginRight: {

        width: '100%',

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

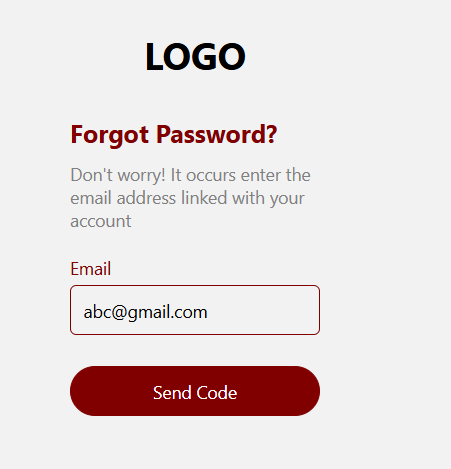
        color: 'white',

        textAlign: 'center',

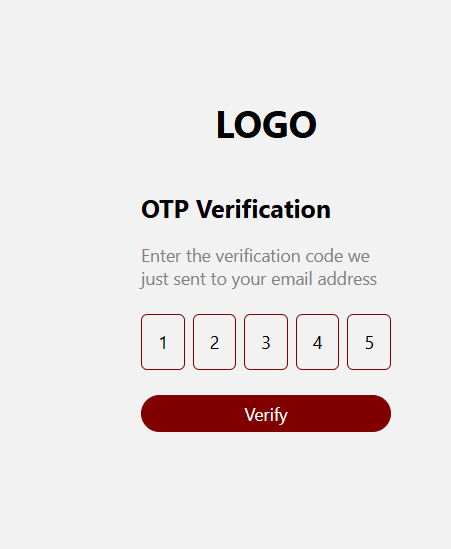
    },

});

**OUTPUT 1**



**OUTPUT 2**



**OTP VERIFICATION TO CREATE NEW PASSWORD**

**otpVerification.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert } from "react-native";

export default function MyOtp({ navigation }) {

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            const otpCode = otp.join('');

            // Assuming you have an API endpoint for OTP verification

            fetch('https://your-backend-api.com/verify-otp', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ otp: otpCode }),

            })

                .then(response => response.json())

                .then(data => {

                    if (data.success) {

                        Alert.alert('Success', 'OTP Verified');

                        navigation.navigate('CreateNewPass');

                    } else {

                        Alert.alert('Error', 'Invalid OTP');

                    }

                })

                .catch(error => {

                    console.error('Error:', error);

                    Alert.alert('Error', 'Something went wrong. Please try again.');

                });

        } else {

            Alert.alert('Error', 'Please enter the complete OTP.');

        }

    };

        const verify = () => {

            navigation.navigate('CreateNewPassword')

        }

    return (

        <View style={styles.container}>

            <View style={styles.contentContainer}>

                <Text style={styles.text}>LOGO</Text>

                <View style={styles.headerContainer}>

                    <Text style={styles.registerHeading}>OTP Verification</Text>

                </View>

                <Text style={styles.description}>

                    Enter the verification code we just sent to your email address

                </Text>

                <View style={styles.inputContainer}>

                    {otp.map((value, index) => (

                        <TextInput

                            key={index}

                            style={[

                                styles.textInput,

                                (focusedInput === index || value) && styles.textInputFocused

                            ]}

                            placeholder=""

                            placeholderTextColor="gray"

                            onFocus={() => setFocusedInput(index)}

                            onChangeText={text => handleChange(text, index)}

                            value={value}

                            maxLength={1} // Only allow one character

                            keyboardType="number-pad" // Numeric keyboard for OTP

                        />

                    ))}

                </View>

                <View style={styles.loginRight}>

                    <TouchableOpacity

                        style={[styles.button, isFormFilled && styles.buttonFilled]}

                        onPress={verify}

                    >

                        <Text style={styles.buttonText} >Verify</Text>

                    </TouchableOpacity>

                </View>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

    },

    contentContainer: {

        width: 200,

        alignItems: 'center',

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: '100%',

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        marginBottom: 10,

        textAlign: 'left',

        width: '100%',

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: '100%',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    loginRight: {

        width: '100%',

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

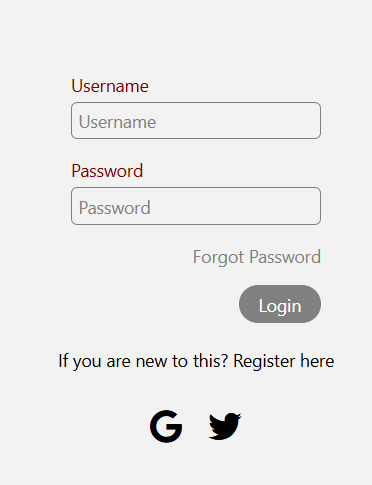
        color: 'white',

        textAlign: 'center',

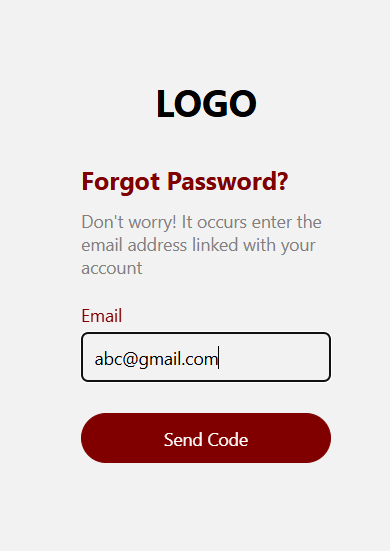
    },

});

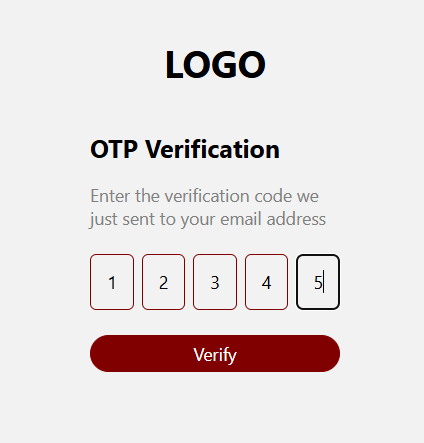
**OUTPUT 1**



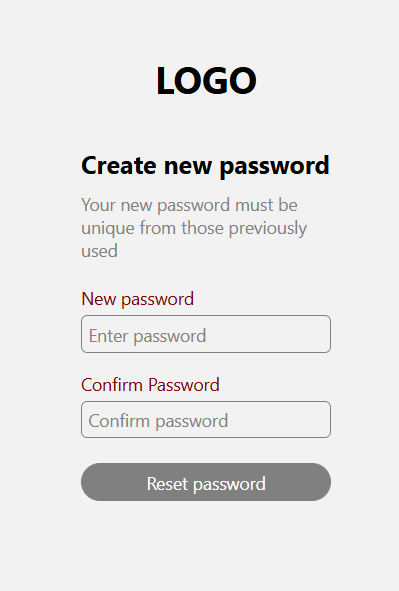
**OUTPUT 2**



**OUTPUT 3**



**OUTPUT 4**



**CREATE NEW PASSWORD TO LOGIN**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import { updatePassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyCreNewPass({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

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

    const [conpassword, setConPassword] = useState('');

    const [errorMessage, setErrorMessage] = useState('');

    const handleResetPassword = () => {

        if (password !== conpassword) {

            setErrorMessage('Passwords do not match');

            return;

        }

        const user = auth.currentUser;

        updatePassword(user, password).then(() => {

            setErrorMessage(''); // Clear the error message on success

            Alert.alert('Success', 'Password updated successfully');

        }).catch((error) => {

            setErrorMessage(error.message);

        });

    };

    const resetPass = () => {

        navigation.navigate('Dashboard')

    }

    const isFormFilled = password && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Create new password</Text>

                <Text style={styles.description}>

                    Your new password must be unique from those previously used

                </Text>

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>New password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {errorMessage ? (

                <Text style={styles.errorText}>{errorMessage}</Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={resetPass}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Reset password</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "maroon",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'maroon',

    },

    errorText: {

        color: 'red',

        marginVertical: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'maroon',

        backgroundColor: 'maroon',

    },

    buttonText: {

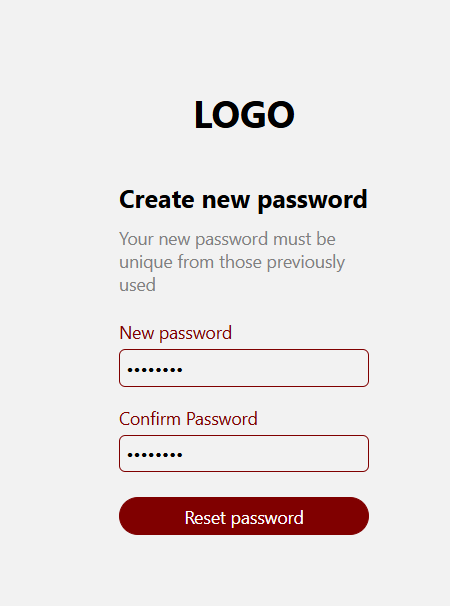
        color: 'white',

        textAlign: 'center',

    },

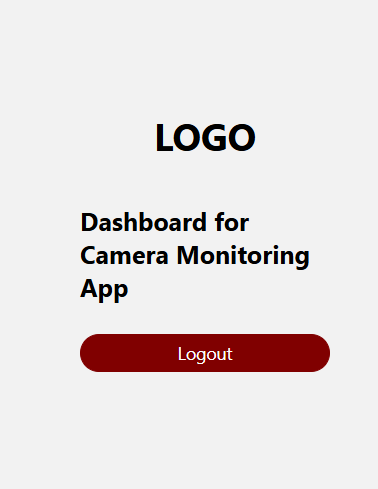
});

**OUTPUT 1**



**OUTPUT 2**

**REDIRECTS TO Dashboard PAGE**



**FINAL CODE AND OUTPUT**

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyOtp from './Pages/otpVerification';

import MyCreNewPass from './Pages/CreNewPassword';

import MyDashboard from './Pages/Dashboard';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import { NavigationContainer } from '@react-navigation/native';

const Stack = createNativeStackNavigator();

export default function App() {

  return (

    // <View>

    //   <navigation/>

    // </View>

   <NavigationContainer>

    <Stack.Navigator initialRouteName='Login'>

      <Stack.Screen name='Register' component={MyReg}/>

      <Stack.Screen name='Login' component={MyLogin}/>

      <Stack.Screen name='ForgotPassword' component={MyForPass}/>

      <Stack.Screen name='OTPverification' component={MyOtp}/>

      <Stack.Screen name='CreateNewPassword' component={MyCreNewPass}/>

      <Stack.Screen name='Dashboard' component={MyDashboard}/>

    </Stack.Navigator>

   </NavigationContainer>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

    width: '100%',

  },

});

**LoginPage.js**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                navigation.navigate('Dashboard')

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')

    }

    const goToForPass = () => {

        navigation.navigate('ForgotPassword')

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            {/\* <Image

                source={require('../assets/cam-one-main-logo.png')}

                style={styles.image}

            /> \*/}

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText} onPress={goToForPass}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text style={{color:'gray'}} onPress={goToRegister}>If you are new to this? <Text style= {{color: 'black', textDecorationLine:'underline'}}>Register here</Text></Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 200,

        height: 200,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

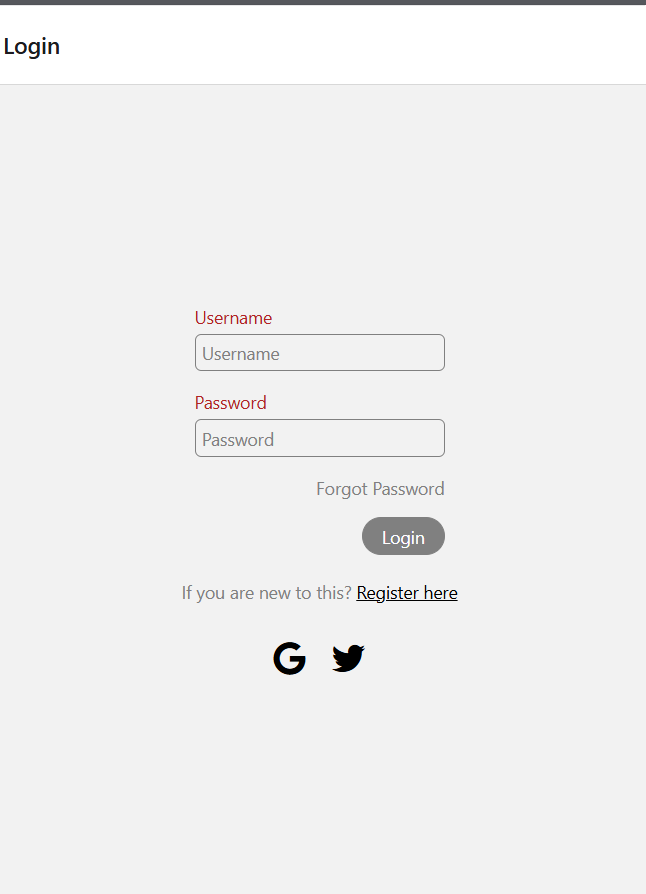
    },

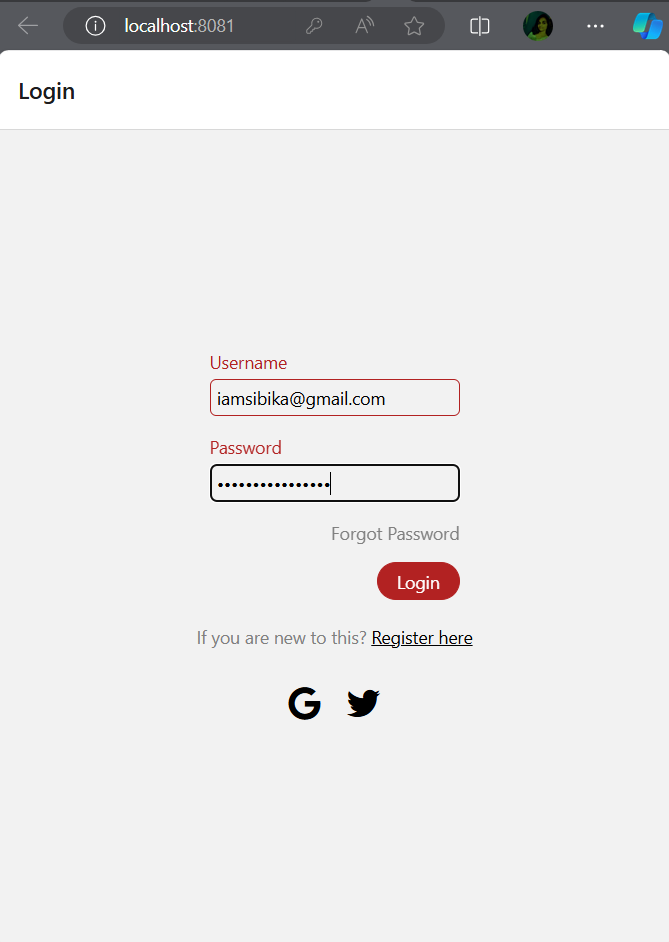
    icon: {

        marginHorizontal: 10,

    },

});





**RegisterPage.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

import { createUserWithEmailAndPassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyReg({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const [error, setError] = useState('');

    const handleRegister = () => {

        if (password !== conpassword) {

            setError('Passwords do not match');

            return;

        }

        createUserWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

               navigation.navigate('Dashboard')

                setError(''); // Clear the error message upon successful registration

            })

            .catch((error) => {

                console.error(error);

                setError(error.message);

            });

    };

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleRegister}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

        color: 'firebrick',

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        color: 'firebrick',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

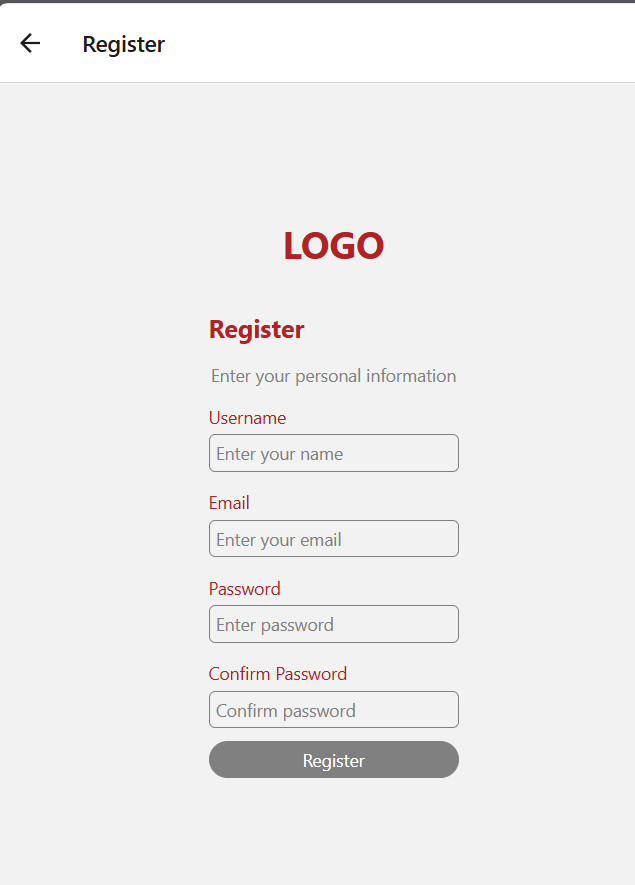
    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});



**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Dimensions } from "react-native";

const screenWidth = Dimensions.get('window').width;

export default function MyForPass({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const isFormFilled = username;

    const sendCode = () => {

        navigation.navigate('OTPverification')

    }

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>Don't worry! It occurs enter the email

                 address linked with your account</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText} onPress={sendCode}>Send Code</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

        paddingHorizontal: 20,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

    },

    headerContainer: {

        alignItems: 'flex-start',

        width: 200,

        marginBottom: 10,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color: 'firebrick',

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        textAlign: 'left',

        marginBottom: 20,

        width: 200,

    },

    inputContainer: {

        width: 200,

        marginBottom: 20,

    },

    labelText: {

        color: 'firebrick',

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 10,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    loginRight: {

        width: 200,

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 10,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**OTP verification.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert } from "react-native";

export default function MyOtp({ navigation }) {

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            const otpCode = otp.join('');

            // Assuming you have an API endpoint for OTP verification

            fetch('https://your-backend-api.com/verify-otp', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ otp: otpCode }),

            })

                .then(response => response.json())

                .then(data => {

                    if (data.success) {

                        Alert.alert('Success', 'OTP Verified');

                        navigation.navigate('CreateNewPass');

                    } else {

                        Alert.alert('Error', 'Invalid OTP');

                    }

                })

                .catch(error => {

                    console.error('Error:', error);

                    Alert.alert('Error', 'Something went wrong. Please try again.');

                });

        } else {

            Alert.alert('Error', 'Please enter the complete OTP.');

        }

    };

        const verify = () => {

            navigation.navigate('CreateNewPassword')

        }

    return (

        <View style={styles.container}>

            <View style={styles.contentContainer}>

                <Text style={styles.text}>LOGO</Text>

                <View style={styles.headerContainer}>

                    <Text style={styles.registerHeading}>OTP Verification</Text>

                </View>

                <Text style={styles.description}>

                    Enter the verification code we just sent to your email address

                </Text>

                <View style={styles.inputContainer}>

                    {otp.map((value, index) => (

                        <TextInput

                            key={index}

                            style={[

                                styles.textInput,

                                (focusedInput === index || value) && styles.textInputFocused

                            ]}

                            placeholder=""

                            placeholderTextColor="gray"

                            onFocus={() => setFocusedInput(index)}

                            onChangeText={text => handleChange(text, index)}

                            value={value}

                            maxLength={1} // Only allow one character

                            keyboardType="number-pad" // Numeric keyboard for OTP

                        />

                    ))}

                </View>

                <View style={styles.loginRight}>

                    <TouchableOpacity

                        style={[styles.button, isFormFilled && styles.buttonFilled]}

                        onPress={verify}

                    >

                        <Text style={styles.buttonText} >Verify</Text>

                    </TouchableOpacity>

                </View>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

    },

    contentContainer: {

        width: 200,

        alignItems: 'center',

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: '100%',

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color:'firebrick',

        marginBottom: 10,

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        marginBottom: 10,

        textAlign: 'left',

        width: '100%',

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: '100%',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    loginRight: {

        width: '100%',

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**Create New Password.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet } from "react-native";

import { updatePassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyCreNewPass({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

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

    const [conpassword, setConPassword] = useState('');

    const [errorMessage, setErrorMessage] = useState('');

    const handleResetPassword = () => {

        if (password !== conpassword) {

            setErrorMessage('Passwords do not match');

            return;

        }

        const user = auth.currentUser;

        updatePassword(user, password).then(() => {

            setErrorMessage(''); // Clear the error message on success

            Alert.alert('Success', 'Password updated successfully');

        }).catch((error) => {

            setErrorMessage(error.message);

        });

    };

    const resetPass = () => {

        navigation.navigate('Dashboard')

    }

    const isFormFilled = password && conpassword;

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Create new password</Text>

                <Text style={styles.description}>

                    Your new password must be unique from those previously used

                </Text>

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>New password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {errorMessage ? (

                <Text style={styles.errorText}>{errorMessage}</Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={resetPass}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Reset password</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color:'firebrick',

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    errorText: {

        color: 'red',

        marginVertical: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',fire

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**Dashboard.js**

// MyDashboard.js

import React from 'react';

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

import { signOut } from 'firebase/auth';

import auth from '../Services/FirebaseAuth'; // Adjust path based on your folder structure

export default function MyDashboard({ navigation }) {

    const handleLogout = () => {

        signOut(auth)

            .then(() => {

                // Sign-out successful.

                console.log('User logged out');

                // Redirect or navigate to login screen after logging out

                navigation.navigate('Login'); // Replace 'Login' with your login screen route

            })

            .catch((error) => {

                // An error happened.

                console.error('Sign Out Error', error);

            });

    };

    return (

        <View style={styles.container}>

            <Text style={styles.text}>LOGO</Text>

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Dashboard for Camera Monitoring App</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button} onPress={handleLogout}>

                    <Text style={styles.buttonText}>Logout</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1, // Ensure it takes up the full screen

        justifyContent: 'center',

        alignItems: 'center',

        margin: 10,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'firebrick',

        borderWidth: 1,

        backgroundColor: 'firebrick',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**ADDING LOGO OF CAM ONE PROJECT**

By using the <Image/> tag, we can add an image in the application.

First of all, we should upload the image from our computer to assets directory in the application.

Copy the image and open assets -> reveal as a file explorer ->assets -> paste the image. And it will be shown in the assets directory. After that should copy the path of the image in assets directory and paste it in the <Image source = {require(path)}/>. Before that should import the Image from react-native.

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

<Image

                source={require('../assets/cam-one-main.png')}

                style={styles.image}

            />

**LOGIN PAGE CODE**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                navigation.navigate('Dashboard')

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')

    }

    const goToForPass = () => {

        navigation.navigate('ForgotPassword')

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            <Image

                source={require('../assets/cam-one-main.png')}

                style={styles.image}

            />

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText} onPress={goToForPass}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text style={{color:'gray'}} onPress={goToRegister}>If you are new to this? <Text style= {{color: 'black', textDecorationLine:'underline'}}>Register here</Text></Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 170,

        height: 170,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

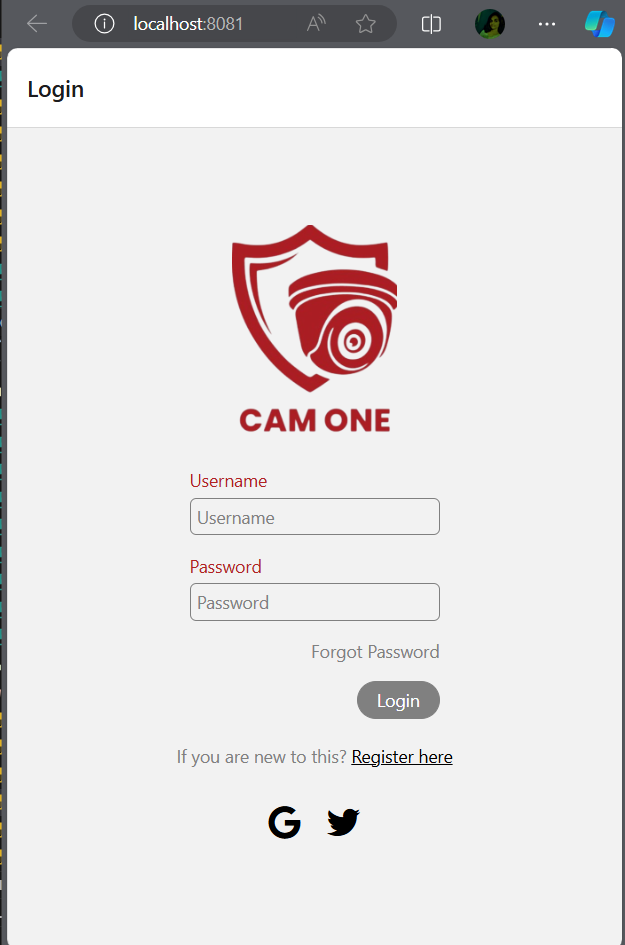
    icon: {

        marginHorizontal: 10,

    },

});

**SCREEN**



**RegisterPage.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

import { createUserWithEmailAndPassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyReg({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const [error, setError] = useState('');

    const handleRegister = () => {

        if (password !== conpassword) {

            setError('Passwords do not match');

            return;

        }

        createUserWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

               navigation.navigate('Dashboard')

                setError(''); // Clear the error message upon successful registration

            })

            .catch((error) => {

                console.error(error);

                setError(error.message);

            });

    };

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Image source = {require('../assets/cam-one-main.png')}

            style = {styles.image}

            />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleRegister}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        height:100,

        width:100,

       resizeMode:'contain',

       marginBottom: 20,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

        color: 'firebrick',

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        color: 'firebrick',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

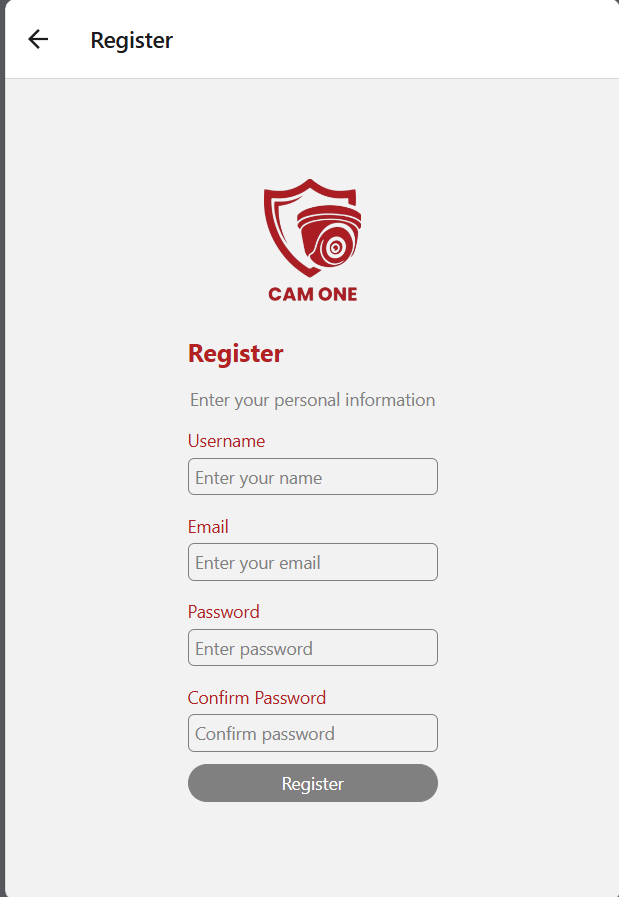
        color: 'white',

        textAlign: 'center',

    },

});

**SCREEN**



**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Dimensions, Image } from "react-native";

const screenWidth = Dimensions.get('window').width;

export default function MyForPass({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const isFormFilled = username;

    const sendCode = () => {

        navigation.navigate('OTPverification')

    }

    return (

        <View style={styles.container}>

            <Image

            source={require('../assets/ForgotPass-removebg-preview.png')}

            style = {styles.image}

            />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>Don't worry! It occurs enter the email

                 address linked with your account</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity style={[styles.button, isFormFilled && styles.buttonFilled]}>

                    <Text style={styles.buttonText} onPress={sendCode}>Send Code</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

        paddingHorizontal: 20,

    },

    image: {

        height: 200,

        width: 200,

        resizeMode:'contain',

        marginBottom: 20,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

    },

    headerContainer: {

        alignItems: 'flex-start',

        width: 200,

        marginBottom: 10,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color: 'firebrick',

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        textAlign: 'left',

        marginBottom: 20,

        width: 200,

    },

    inputContainer: {

        width: 200,

        marginBottom: 20,

    },

    labelText: {

        color: 'firebrick',

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 10,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    loginRight: {

        width: 200,

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 10,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

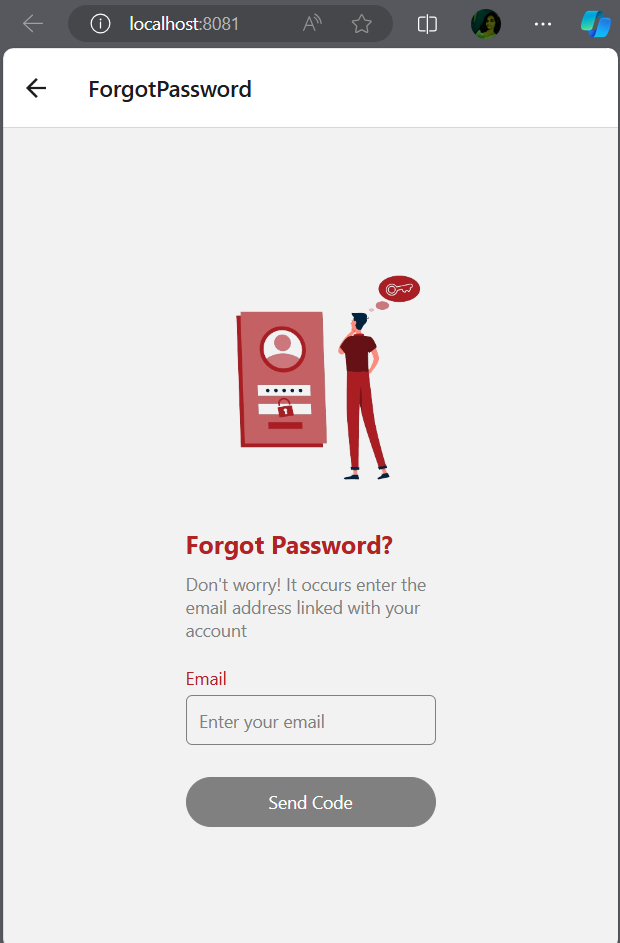
        color: 'white',

        textAlign: 'center',

    },

});

**SCREEN**



**OtpVerification.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Alert, Image } from "react-native";

export default function MyOtp({ navigation }) {

    const [otp, setOtp] = useState(['', '', '', '', '']);

    const [focusedInput, setFocusedInput] = useState(null);

    const isFormFilled = otp.every(value => value.length > 0);

    const handleChange = (text, index) => {

        const newOtp = [...otp];

        newOtp[index] = text;

        setOtp(newOtp);

        if (text.length === 1 && index < 4) {

            setFocusedInput(index + 1);

        }

        if (text.length === 0 && index > 0) {

            setFocusedInput(index - 1);

        }

    };

    const handleVerify = () => {

        if (isFormFilled) {

            const otpCode = otp.join('');

            // Assuming you have an API endpoint for OTP verification

            fetch('https://your-backend-api.com/verify-otp', {

                method: 'POST',

                headers: {

                    'Content-Type': 'application/json',

                },

                body: JSON.stringify({ otp: otpCode }),

            })

                .then(response => response.json())

                .then(data => {

                    if (data.success) {

                        Alert.alert('Success', 'OTP Verified');

                        navigation.navigate('CreateNewPass');

                    } else {

                        Alert.alert('Error', 'Invalid OTP');

                    }

                })

                .catch(error => {

                    console.error('Error:', error);

                    Alert.alert('Error', 'Something went wrong. Please try again.');

                });

        } else {

            Alert.alert('Error', 'Please enter the complete OTP.');

        }

    };

        const verify = () => {

            navigation.navigate('CreateNewPassword')

        }

    return (

        <View style={styles.container}>

            <View style={styles.contentContainer}>

                <Image

                source={require('../assets/OTPVerify-removebg-preview.png')}

                style ={styles.image}

                />

                <View style={styles.headerContainer}>

                    <Text style={styles.registerHeading}>OTP Verification</Text>

                </View>

                <Text style={styles.description}>

                    Enter the verification code we just sent to your email address

                </Text>

                <View style={styles.inputContainer}>

                    {otp.map((value, index) => (

                        <TextInput

                            key={index}

                            style={[

                                styles.textInput,

                                (focusedInput === index || value) && styles.textInputFocused

                            ]}

                            placeholder=""

                            placeholderTextColor="gray"

                            onFocus={() => setFocusedInput(index)}

                            onChangeText={text => handleChange(text, index)}

                            value={value}

                            maxLength={1} // Only allow one character

                            keyboardType="number-pad" // Numeric keyboard for OTP

                        />

                    ))}

                </View>

                <View style={styles.loginRight}>

                    <TouchableOpacity

                        style={[styles.button, isFormFilled && styles.buttonFilled]}

                        onPress={verify}

                    >

                        <Text style={styles.buttonText} >Verify</Text>

                    </TouchableOpacity>

                </View>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

    },

    image: {

        height:200,

        width: 200,

        resizeMode:'contain',

        marginBottom: 20,

    },

    contentContainer: {

        width: 200,

        alignItems: 'center',

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: '100%',

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color:'firebrick',

        marginBottom: 10,

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        marginBottom: 10,

        textAlign: 'left',

        width: '100%',

    },

    inputContainer: {

        flexDirection: 'row',

        justifyContent: 'space-between',

        width: '100%',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: 35,

        height: 45,

        textAlign: 'center',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    loginRight: {

        width: '100%',

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

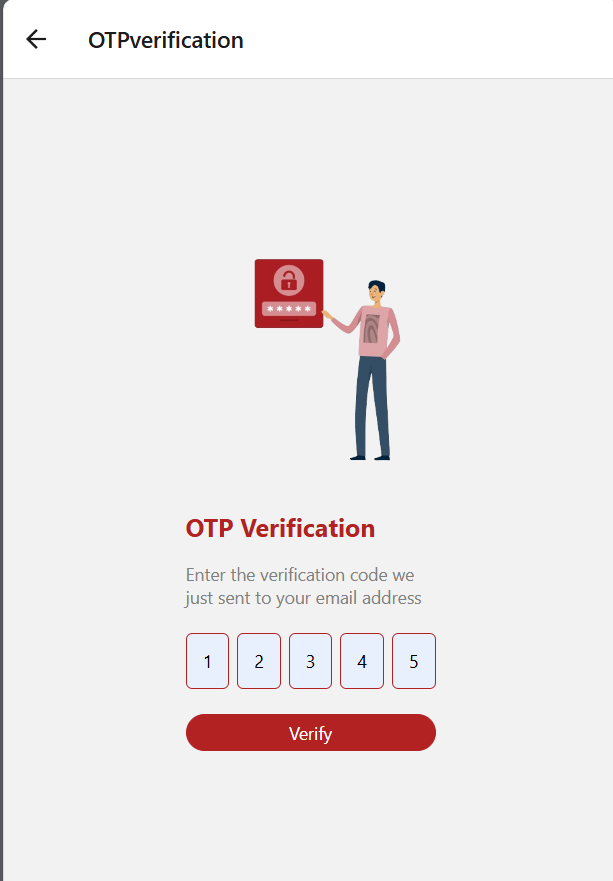
        color: 'white',

        textAlign: 'center',

    },

});

**SCREEN**



**CreateNewPass.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image} from "react-native";

import { updatePassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyCreNewPass({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

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

    const [conpassword, setConPassword] = useState('');

    const [errorMessage, setErrorMessage] = useState('');

    const handleResetPassword = () => {

        if (password !== conpassword) {

            setErrorMessage('Passwords do not match');

            return;

        }

        const user = auth.currentUser;

        updatePassword(user, password).then(() => {

            setErrorMessage(''); // Clear the error message on success

            Alert.alert('Success', 'Password updated successfully');

        }).catch((error) => {

            setErrorMessage(error.message);

        });

    };

    const resetPass = () => {

        navigation.navigate('Dashboard')

    }

    const isFormFilled = password && conpassword;

    return (

        <View style={styles.container}>

           <Image

           source={require('../assets/CreateNewPass-removebg-preview.png')}

           style={styles.image}

           />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Create new password</Text>

                <Text style={styles.description}>

                    Your new password must be unique from those previously used

                </Text>

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>New password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {errorMessage ? (

                <Text style={styles.errorText}>{errorMessage}</Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={resetPass}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Reset password</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        height:200,

        width:200,

        resizeMode:'contain',

        marginBottom: 20,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color:'firebrick',

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    errorText: {

        color: 'red',

        marginVertical: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

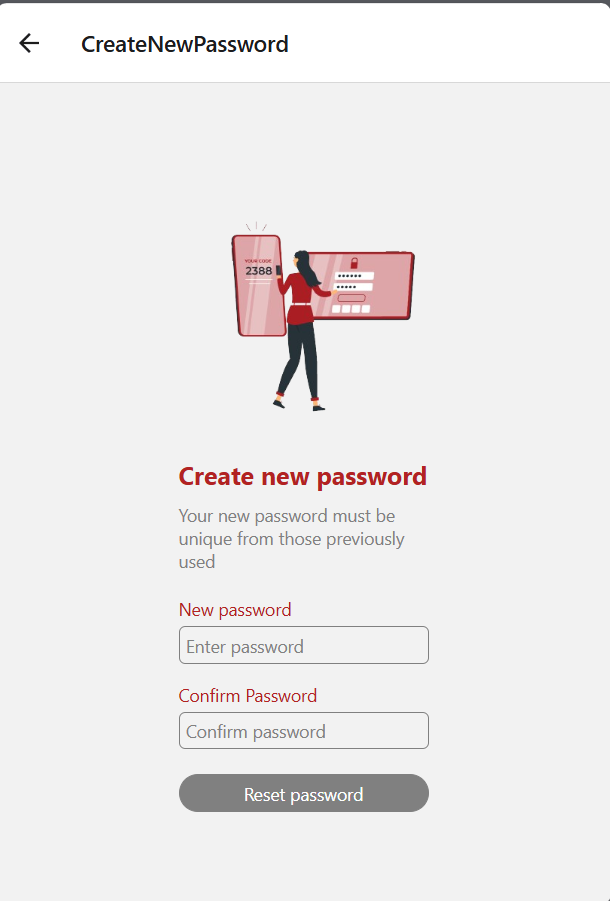
        color: 'white',

        textAlign: 'center',

    },

});

**SCREEN**



**Dashboard.js**

// MyDashboard.js

import React from 'react';

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

import { signOut } from 'firebase/auth';

import auth from '../Services/FirebaseAuth'; // Adjust path based on your folder structure

export default function MyDashboard({ navigation }) {

    const handleLogout = () => {

        signOut(auth)

            .then(() => {

                // Sign-out successful.

                console.log('User logged out');

                // Redirect or navigate to login screen after logging out

                navigation.navigate('Login'); // Replace 'Login' with your login screen route

            })

            .catch((error) => {

                // An error happened.

                console.error('Sign Out Error', error);

            });

    };

    return (

        <View style={styles.container}>

            <View style={styles.headerContainer}>

                {/\* <Text style={styles.registerHeading}>Dashboard for Camera Monitoring App</Text> \*/}

            </View>

            <Image

            source={require('../assets/Dashboard-removebg-preview.png')}

            style={styles.image}

            />

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button} onPress={handleLogout}>

                    <Text style={styles.buttonText}>Logout</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1, // Ensure it takes up the full screen

        justifyContent: 'center',

        alignItems: 'center',

        margin: 10,

    },

    image: {

        height: '90%',

        width: '90%',

        resizeMode: 'contain',

        marginBottom: 20,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'firebrick',

        borderWidth: 1,

        backgroundColor: 'firebrick',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonText: {

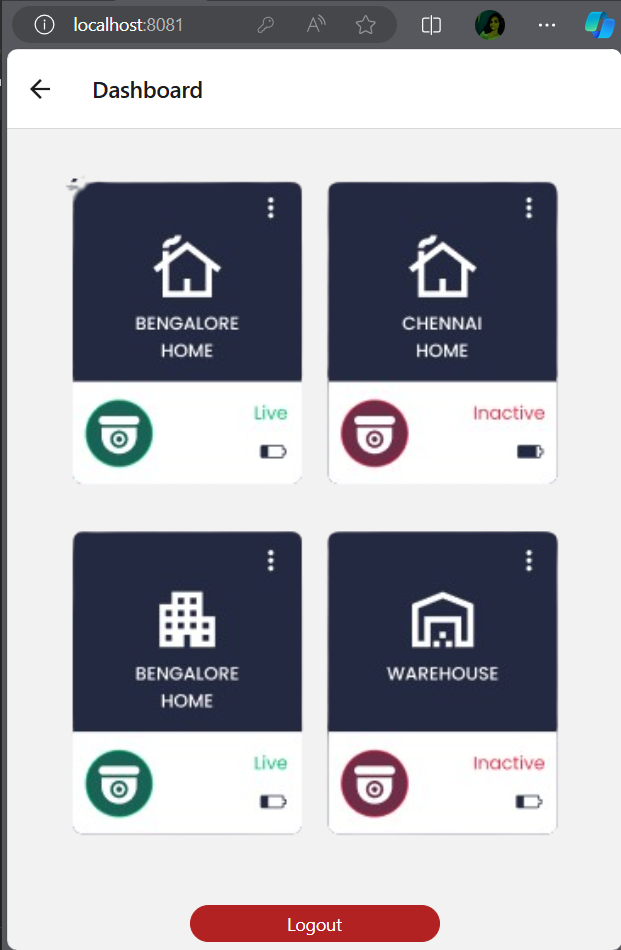
        color: 'white',

        textAlign: 'center',

    },

});

**SCREEN**



**Here we are going to validating the forgot password like we are going to integrating the forgot password page to the firebase.**

**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Dimensions, Image } from "react-native";

import { getAuth, sendPasswordResetEmail } from "firebase/auth"; // Import sendPasswordResetEmail

const screenWidth = Dimensions.get('window').width;

export default function MyForPass({ navigation }) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const [message, setMessage] = useState('');

    const [messageType, setMessageType] = useState(''); // 'success' or 'error'

    const isFormFilled = username;

    const sendCode = async () => {

        if (!username) {

            setMessage('Please enter a valid email address');

            setMessageType('error');

            return;

        }

        console.log('Sending password reset email to:', username);

        try {

            const auth = getAuth(); // Get the auth instance

            await sendPasswordResetEmail(auth, username); // Call sendPasswordResetEmail

            console.log('Password reset email sent to:', username);

            setMessage('Password reset email sent!');

            setMessageType('success');

            navigation.navigate('OTPverification');

        } catch (error) {

            console.error('Error sending password reset email:', error.code, error.message);

            setMessage(`Error: ${error.message}`);

            setMessageType('error');

        }

    };

    return (

        <View style={styles.container}>

            <Image

                source={require('../assets/ForgotPass-removebg-preview.png')}

                style={styles.image}

            />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>

                Don't worry! It happens. Enter the email address linked with your account.

            </Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            {message ? (

                <Text style={[styles.message, messageType === 'success' ? styles.successMessage : styles.errorMessage]}>

                    {message}

                </Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={sendCode}

                >

                    <Text style={styles.buttonText}>Send Code</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

        paddingHorizontal: 20,

    },

    image: {

        height: 200,

        width: 200,

        resizeMode: 'contain',

        marginBottom: 20,

    },

    headerContainer: {

        alignItems: 'flex-start',

        width: 200,

        marginBottom: 10,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color: 'firebrick',

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        textAlign: 'left',

        marginBottom: 20,

        width: 200,

    },

    inputContainer: {

        width: 200,

        marginBottom: 20,

    },

    labelText: {

        color: 'firebrick',

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 10,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    message: {

        marginVertical: 10,

        padding: 10,

        borderRadius: 5,

        textAlign: 'center',

        width: '100%',

    },

    successMessage: {

        color: 'green',

        backgroundColor: '#d4edda',

        borderColor: '#c3e6cb',

    },

    errorMessage: {

        color: 'red',

        backgroundColor: '#f8d7da',

        borderColor: '#f5c6cb',

    },

    loginRight: {

        width: 200,

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 10,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**FirebaseAuth.js**

// Import the functions you need from the SDKs you need

import { initializeApp } from "firebase/app";

import { getAuth, initializeAuth } from "firebase/auth";

// Your web app's Firebase configuration

const firebaseConfig = {

  apiKey: "AIzaSyBI6cCSoO7ngINvfFc6KPrnsT2V2ApFB1o",

  authDomain: "cam-one-1.firebaseapp.com",

  projectId: "cam-one-1",

  storageBucket: "cam-one-1.appspot.com",

  messagingSenderId: "1062546701892",

  appId: "1:1062546701892:web:015b4b9bd2f7b7d8d98611"

};

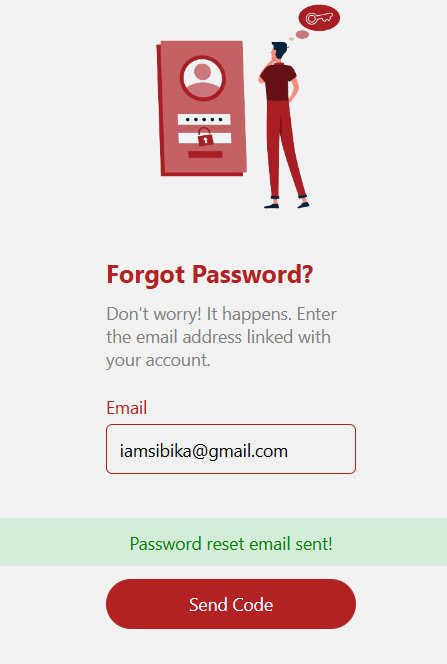
// Initialize Firebase

const app = initializeApp(firebaseConfig);

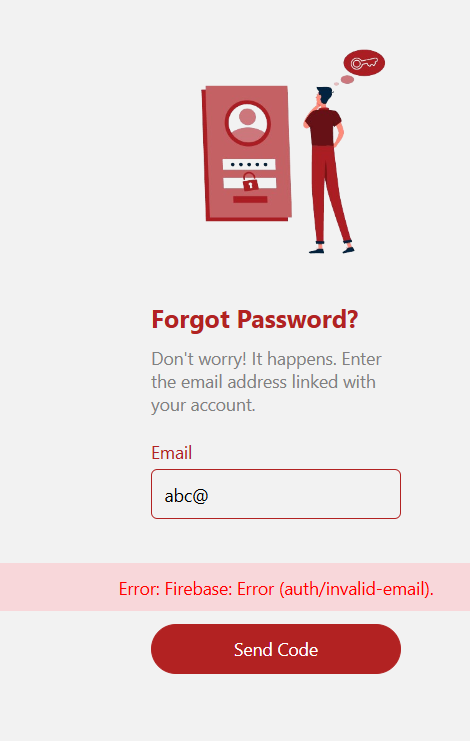
const auth = getAuth(app);

export default auth;

**SCREEN**



**It actually sends an mail with password reset link to the above mentioned mail.**



**------------------\*\*\*\*\*\*\*\*\*\*\*\*-------------------------\*\*\*\*\*\*\*\*\*\*\*-------**

**CHANGING FRONTEND BASED ON BACKEND SUPPORTS**

**App.js**

import { StatusBar } from 'expo-status-bar';

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

import MyLogin from './Pages/LoginPage';

import MyReg from './Pages/RegisterPage';

import MyForPass from './Pages/ForgotPassword';

import MyDashboard from './Pages/Dashboard';

import { createNativeStackNavigator } from '@react-navigation/native-stack';

import { NavigationContainer } from '@react-navigation/native';

const Stack = createNativeStackNavigator();

export default function App() {

  return (

    // <View>

    //   <navigation/>

    // </View>

   <NavigationContainer>

    <Stack.Navigator initialRouteName='Login'>

      <Stack.Screen name='Register' component={MyReg}/>

      <Stack.Screen name='Login' component={MyLogin}/>

      <Stack.Screen name='ForgotPassword' component={MyForPass}/>

      <Stack.Screen name='Dashboard' component={MyDashboard}/>

    </Stack.Navigator>

   </NavigationContainer>

  );

}

const styles = StyleSheet.create({

  container: {

    flex: 1,

    backgroundColor: '#fff',

    alignItems: 'center',

    justifyContent: 'center',

    width: '100%',

  },

});

**LoginPage.js**

import { signInWithEmailAndPassword } from "firebase/auth";

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome';

import auth from '../Services/FirebaseAuth';

export default function MyLogin({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [error, setError] = useState('');

    const handleLogin = () => {

        signInWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

                navigation.navigate('Dashboard')

            })

            .catch((error) => {

                setError(error.message);

            });

    };

    const goToRegister = () => {

        navigation.navigate('Register')

    }

    const goToForPass = () => {

        navigation.navigate('ForgotPassword')

    }

    const isFormFilled = username && password;

    return (

        <View style={styles.container}>

            {/\*<Text style={styles.text}>LOGO</Text>\*/}

            <Image

                source={require('../assets/cam-one-main.png')}

                style={styles.image}

            />

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Username"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.forgotPasswordContainer}>

                <Text style={styles.forgotPasswordText} onPress={goToForPass}>Forgot Password</Text>

            </View>

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleLogin}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Login</Text>

                </TouchableOpacity>

            </View>

            <View style={styles.registerContainer}>

                <Text style={{color:'gray'}} onPress={goToRegister}>If you are new to this? <Text style= {{color: 'black', textDecorationLine:'underline'}}>Register here</Text></Text>

            </View>

            <View style={styles.iconContainer}>

                <Icon name="google" size={30} style={styles.icon} />

                <Icon name="twitter" size={30} style={styles.icon} />

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        width: 170,

        height: 170,

        marginBottom: 20,

        resizeMode: 'contain',

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    registerContainer: {

        width: '100%',

        alignItems: 'center',

        marginVertical: 20,

    },

    iconContainer: {

        flexDirection: 'row',

        justifyContent: 'center',

        marginVertical: 10,

    },

    icon: {

        marginHorizontal: 10,

    },

});

**RegisterPage.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Image } from "react-native";

import Icon from 'react-native-vector-icons/FontAwesome'; // Importing FontAwesome icons

import { createUserWithEmailAndPassword } from "firebase/auth";

import auth from '../Services/FirebaseAuth';

export default function MyReg({navigation}) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

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

    const [conpassword, setConPassword] = useState('');

    const [name, setName] = useState('');

    const [error, setError] = useState('');

    const handleRegister = () => {

        if (password !== conpassword) {

            setError('Passwords do not match');

            return;

        }

        createUserWithEmailAndPassword(auth, username, password)

            .then((userCredential) => {

                const user = userCredential.user;

               navigation.navigate('Dashboard')

                setError(''); // Clear the error message upon successful registration

            })

            .catch((error) => {

                console.error(error);

                setError(error.message);

            });

    };

    const isFormFilled = username && password && name && conpassword;

    return (

        <View style={styles.container}>

            <Image source = {require('../assets/cam-one-main.png')}

            style = {styles.image}

            />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Register</Text>

            </View>

            <Text style={styles.description}>Enter your personal information</Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Username</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'name' || name) && styles.textInputFocused

                    ]}

                    placeholder="Enter your name"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('name')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setName(text)}

                    value={name}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'password' || password) && styles.textInputFocused

                    ]}

                    placeholder="Enter password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('password')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setPassword(text)}

                    value={password}

                />

            </View>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Confirm Password</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'conpassword' || conpassword) && styles.textInputFocused

                    ]}

                    placeholder="Confirm password"

                    placeholderTextColor="gray"

                    secureTextEntry

                    onFocus={() => setFocusedInput('conpassword')}

                    onBlur={() => setFocusedInput(null)}

                    onChangeText={text => setConPassword(text)}

                    value={conpassword}

                />

            </View>

            {error ? <Text style={styles.errorText}>{error}</Text> : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={handleRegister}

                    disabled={!isFormFilled}

                >

                    <Text style={styles.buttonText}>Register</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        alignItems: 'center',

        justifyContent: 'center',

        flex: 1,

        margin: 10,

    },

    image: {

        height:100,

        width:100,

       resizeMode:'contain',

       marginBottom: 20,

    },

    text: {

        color: 'black',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30, // Adjusted margin

        marginTop: 10,

        color: 'firebrick',

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        color: 'firebrick',

        fontSize: 20,

        marginBottom: 10, // Space below the heading

    },

    description: {

        color: 'gray',

        marginBottom: 10, // Space below the description

    },

    inputContainer: {

        width: 200,

        marginVertical: 5,

        alignItems: 'flex-start',

    },

    labelText: {

        color: "firebrick",

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 5,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    errorText: {

        color: 'red',

        marginVertical: 5,

    },

    forgotPasswordContainer: {

        alignItems: 'flex-end',

        marginVertical: 5,

        width: 200,

    },

    forgotPasswordText: {

        color: 'gray',

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**ForgotPassword.js**

import React, { useState } from "react";

import { View, Text, TextInput, TouchableOpacity, StyleSheet, Dimensions, Image } from "react-native";

import { getAuth, sendPasswordResetEmail } from "firebase/auth"; // Import sendPasswordResetEmail

const screenWidth = Dimensions.get('window').width;

export default function MyForPass({ navigation }) {

    const [focusedInput, setFocusedInput] = useState(null);

    const [username, setUsername] = useState('');

    const [message, setMessage] = useState('');

    const [messageType, setMessageType] = useState(''); // 'success' or 'error'

    const isFormFilled = username;

// const sendCode = async () => {

//     if (!username) {

//         setMessage('Please enter a valid email address');

//         setMessageType('error');

//         return;

//     }

//     try {

//         await sendPasswordResetEmail(auth, username);

//         setMessage('A password reset email has been sent!');

//         setMessageType('success');

//     } catch (error) {

//         console.error('Error sending password reset email:', error.code, error.message);

//         if (error.code === 'auth/user-not-found') {

//             setMessage('This email address is not registered');

//             setMessageType('error');

//         } else {

//             setMessage('An error occurred. Please try again.');

//             setMessageType('error');

//         }

//     }

// };

    const sendCode = async () => {

        if (!username) {

            setMessage('Please enter a valid email address');

            setMessageType('error');

            return;

        }

        console.log('Sending password reset email to:', username);

        try {

            const auth = getAuth(); // Get the auth instance

            await sendPasswordResetEmail(auth, username); // Call sendPasswordResetEmail

            console.log('Password reset email sent to:', username);

            setMessage('Password reset email sent!');

            setMessageType('success');

        } catch (error) {

            console.error('Error sending password reset email:', error.code, error.message);

            setMessage(`Error: ${error.message}`);

            setMessageType('error');

        }

    };

    const goToLogin = () => {

        navigation.navigate('Login')

    }

    return (

        <View style={styles.container}>

            <Image

                source={require('../assets/ForgotPass-removebg-preview.png')}

                style={styles.image}

            />

            <View style={styles.headerContainer}>

                <Text style={styles.registerHeading}>Forgot Password?</Text>

            </View>

            <Text style={styles.description}>

                Don't worry! It happens. Enter the email address linked with your account.

            </Text>

            <View style={styles.inputContainer}>

                <Text style={styles.labelText}>Email</Text>

                <TextInput

                    style={[

                        styles.textInput,

                        (focusedInput === 'username' || username) && styles.textInputFocused

                    ]}

                    placeholder="Enter your email"

                    placeholderTextColor="gray"

                    onFocus={() => setFocusedInput('username')}

                    onChangeText={text => setUsername(text)}

                    value={username}

                />

            </View>

            {message ? (

                <Text style={[styles.message, messageType === 'success' ? styles.successMessage : styles.errorMessage]}>

                    {message}

                </Text>

            ) : null}

            <View style={styles.loginRight}>

                <TouchableOpacity

                    style={[styles.button, isFormFilled && styles.buttonFilled]}

                    onPress={sendCode}

                >

                    <Text style={styles.buttonText}>Send Code</Text>

                </TouchableOpacity>

            </View>

            <Text style= {{color:'gray',marginTop: 20}}>Have you created your new password? <Text style = {{color:'black', marginTop:20, textDecorationLine:'underline'}}onPress={goToLogin}>Login here!</Text></Text>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1,

        justifyContent: 'center',

        alignItems: 'center',

        paddingHorizontal: 20,

    },

    image: {

        height: 200,

        width: 200,

        resizeMode: 'contain',

        marginBottom: 20,

    },

    headerContainer: {

        alignItems: 'flex-start',

        width: 200,

        marginBottom: 10,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        color: 'firebrick',

        textAlign: 'left',

    },

    description: {

        color: 'gray',

        textAlign: 'left',

        marginBottom: 20,

        width: 200,

    },

    inputContainer: {

        width: 200,

        marginBottom: 20,

    },

    labelText: {

        color: 'firebrick',

    },

    textInput: {

        borderColor: 'gray',

        width: '100%',

        padding: 10,

        borderWidth: 1,

        marginVertical: 5,

        borderRadius: 5,

    },

    textInputFocused: {

        borderColor: 'firebrick',

    },

    message: {

        marginVertical: 10,

        padding: 10,

        borderRadius: 5,

        textAlign: 'center',

        width: '100%',

    },

    successMessage: {

        color: 'green',

    },

    errorMessage: {

        color: 'red',

    },

    loginRight: {

        width: 200,

        alignItems: 'center',

    },

    button: {

        borderColor: 'gray',

        borderWidth: 1,

        backgroundColor: 'gray',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 10,

        width: '100%',

    },

    buttonFilled: {

        borderColor: 'firebrick',

        backgroundColor: 'firebrick',

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**Dashboard.js**

// MyDashboard.js

import React from 'react';

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

import { signOut } from 'firebase/auth';

import auth from '../Services/FirebaseAuth'; // Adjust path based on your folder structure

export default function MyDashboard({ navigation }) {

    const handleLogout = () => {

        signOut(auth)

            .then(() => {

                // Sign-out successful.

                console.log('User logged out');

                // Redirect or navigate to login screen after logging out

                navigation.navigate('Login'); // Replace 'Login' with your login screen route

            })

            .catch((error) => {

                // An error happened.

                console.error('Sign Out Error', error);

            });

    };

    return (

        <View style={styles.container}>

            <View style={styles.headerContainer}>

                {/\* <Text style={styles.registerHeading}>Dashboard for Camera Monitoring App</Text> \*/}

            </View>

            <Image

            source={require('../assets/Dashboard-removebg-preview.png')}

            style={styles.image}

            />

            <View style={styles.loginRight}>

                <TouchableOpacity style={styles.button} onPress={handleLogout}>

                    <Text style={styles.buttonText}>Logout</Text>

                </TouchableOpacity>

            </View>

        </View>

    );

}

const styles = StyleSheet.create({

    container: {

        flex: 1, // Ensure it takes up the full screen

        justifyContent: 'center',

        alignItems: 'center',

        margin: 10,

    },

    image: {

        height: '90%',

        width: '90%',

        resizeMode: 'contain',

        marginBottom: 20,

    },

    text: {

        color: 'firebrick',

        fontWeight: 'bold',

        fontSize: 30,

        textAlign: 'center',

        marginBottom: 30,

        marginTop: 10,

    },

    headerContainer: {

        marginVertical: 5,

        alignItems: 'flex-start',

        width: 200,

    },

    registerHeading: {

        fontWeight: 'bold',

        fontSize: 20,

        marginBottom: 10,

    },

    loginRight: {

        width: 200,

        alignItems: 'flex-end',

    },

    button: {

        borderColor: 'firebrick',

        borderWidth: 1,

        backgroundColor: 'firebrick',

        borderRadius: 30,

        alignItems: 'center',

        justifyContent: 'center',

        paddingHorizontal: 15,

        paddingVertical: 5,

        width: '100%',

        marginTop: 10,

    },

    buttonText: {

        color: 'white',

        textAlign: 'center',

    },

});

**OUTPUT SCREENS**

