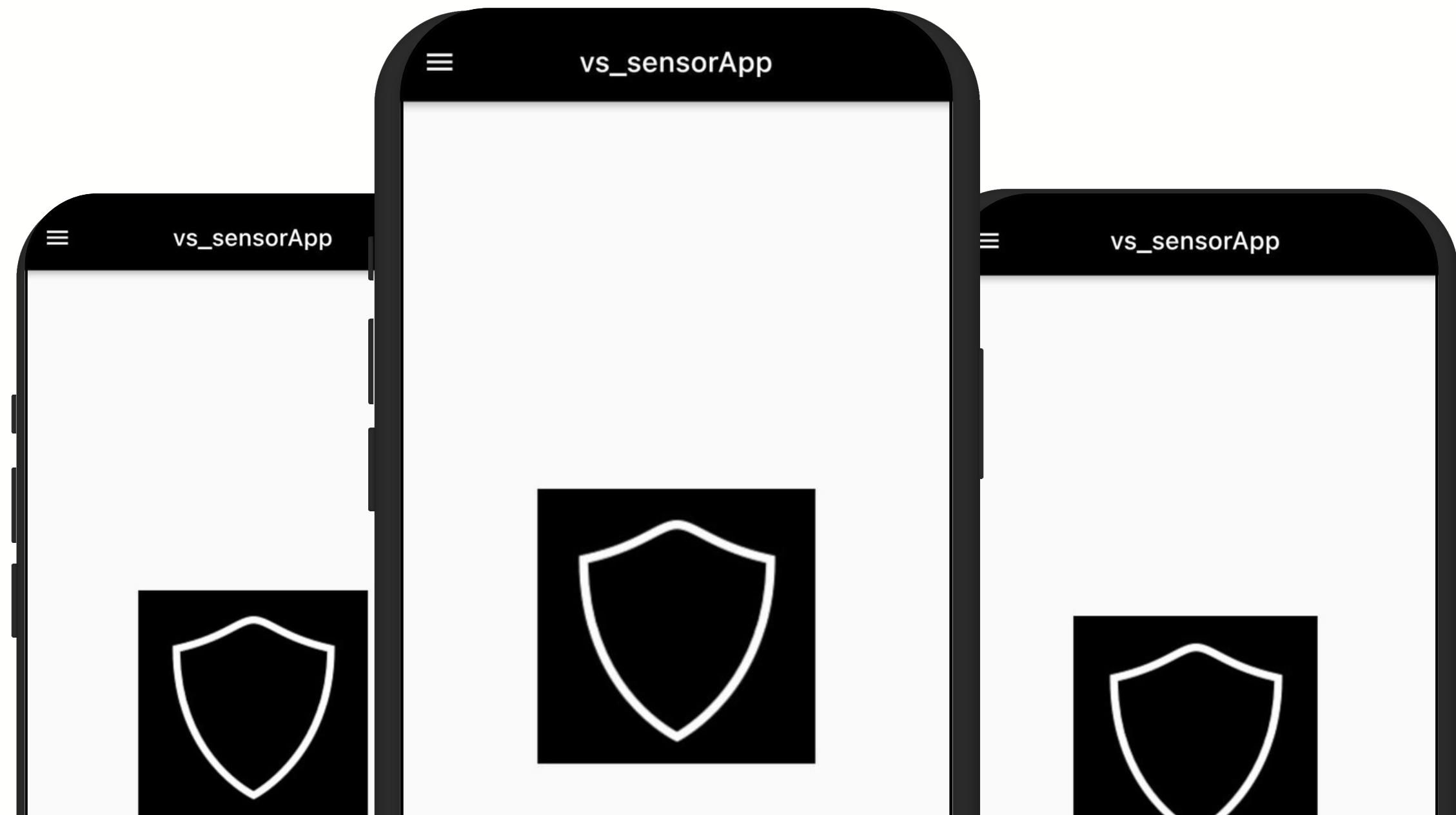


# Vehicle safety system

V.S.S



# Names:

Maha Mohammed Alosaimi - 43900820

Ahlaam Dakhel Al-mutairi - 43902395

Arwa Saud Alghamdi - 43901324

Rahaf Abdullah Bamukhier - 43901507

Hanan Abdullah Alotaibi - 43903405

Nouf Mohammed Algouba - 43903274

Supervised by:

Dr. FAHAD ALSHALAWI

**NEXT →**

# Introduction

**NEXT →**



# Topic

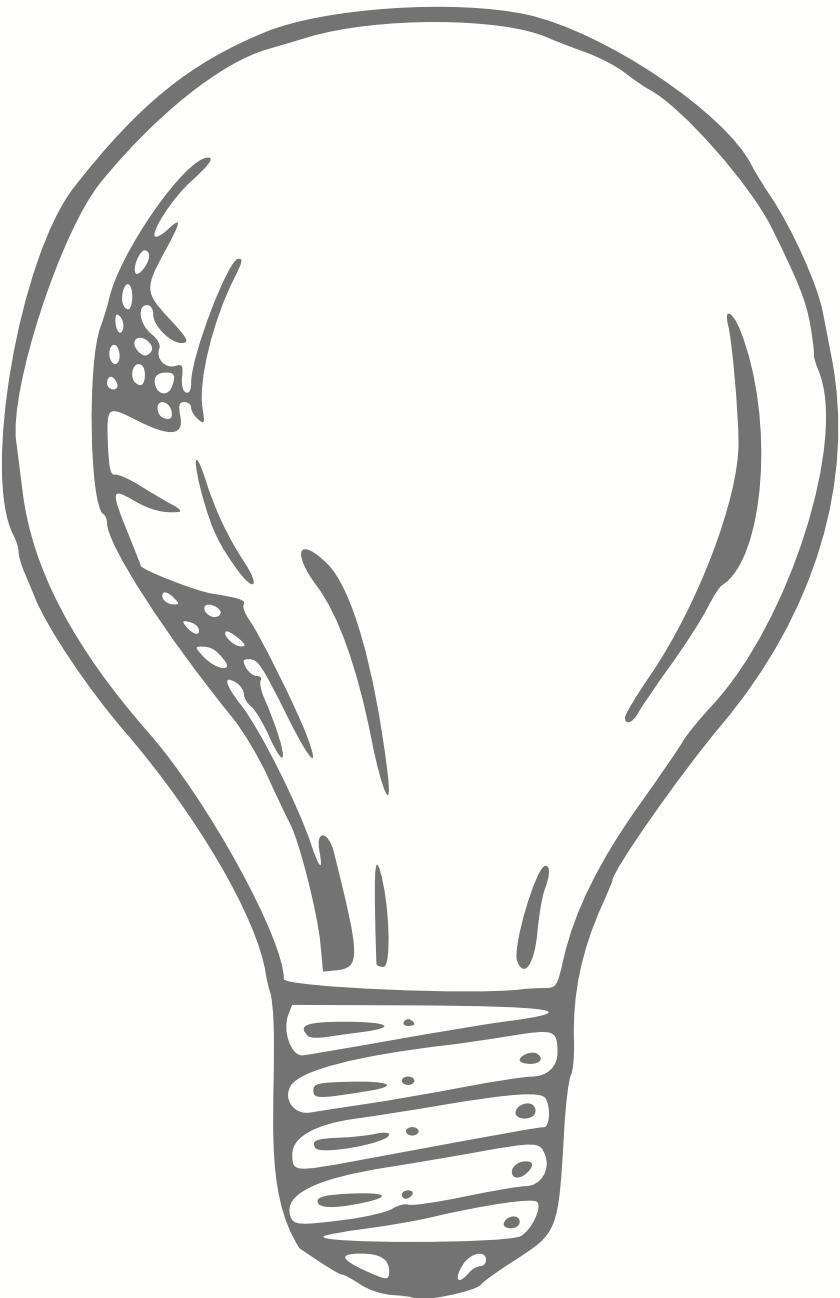
- Review System Idea
- Mobile Application Design
- Program and database
- Hardware (PIR)
- Test
- Conclusions and future work

**NEXT →**

# Review General Idea

Smart phone application connected with high sensitive motion sensor ,it sent alert to the app when it detects any movement inside the Vehicle when it's off

Develop our app  
by applying Internet of things (IoT)  
Designing an interface to let the user sign up and connect the  
PIR sensor

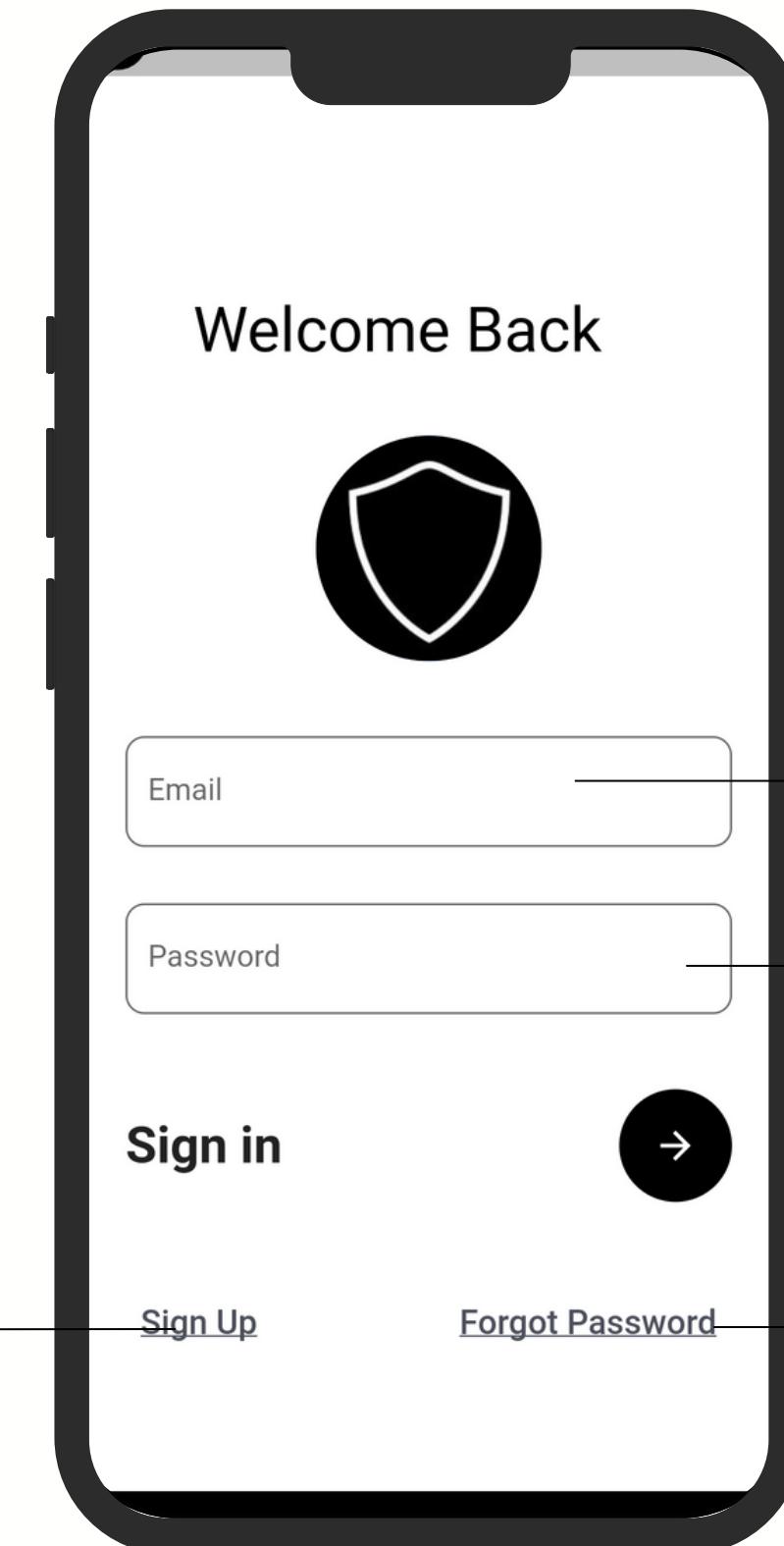


Easy to use  
inexpensive to buy  
good Design to see  
good Power Control and high secure

NEXT →

Notch

# Sign in page



● **create new account**

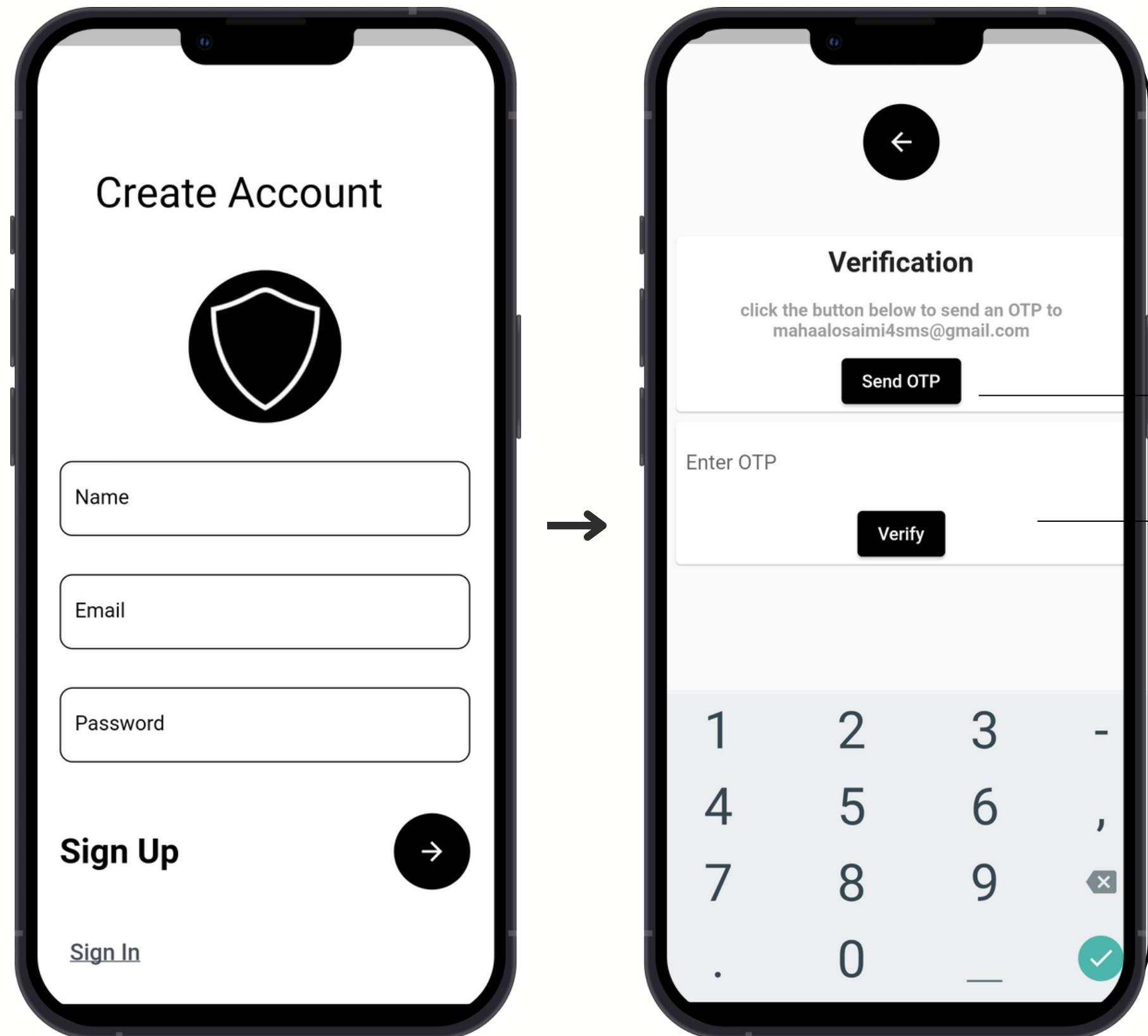
● **The user will enter his email address**

● **The password of the user**

● **Forgot password**

## Vehicle safety system

V.S.S

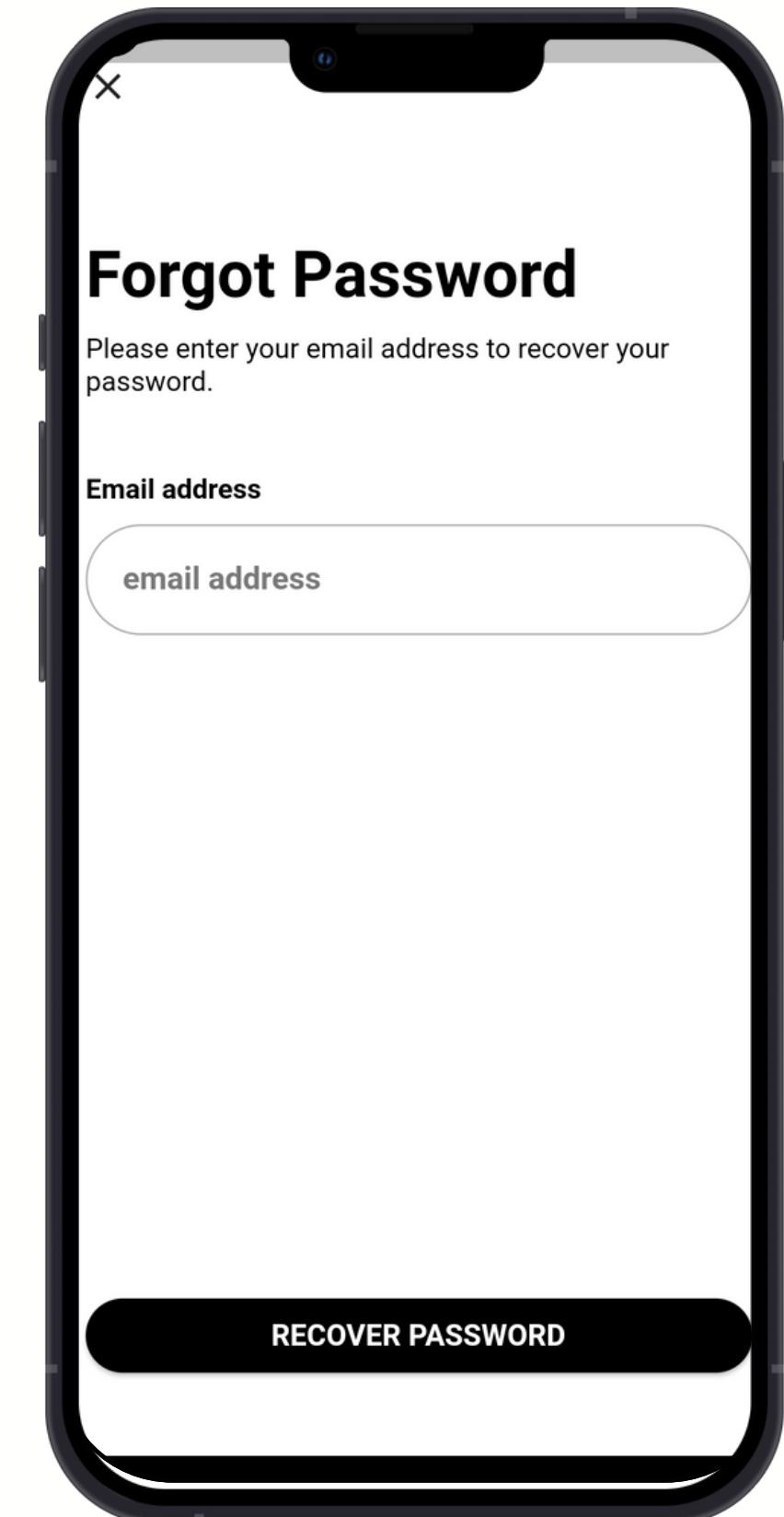


# Sign up page

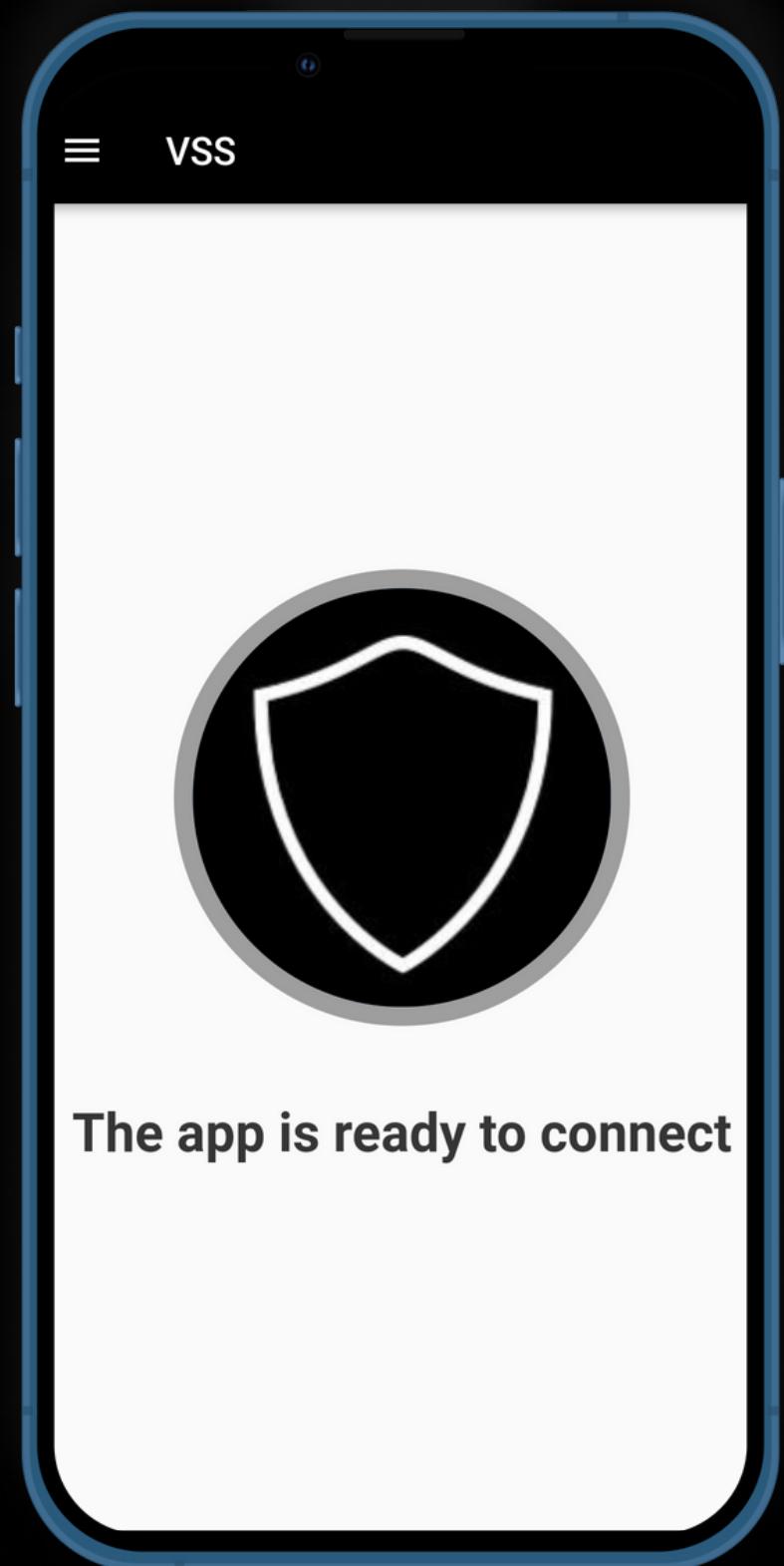
- Send the OTP to the user email

- Write the OTP code

# Forgot password page

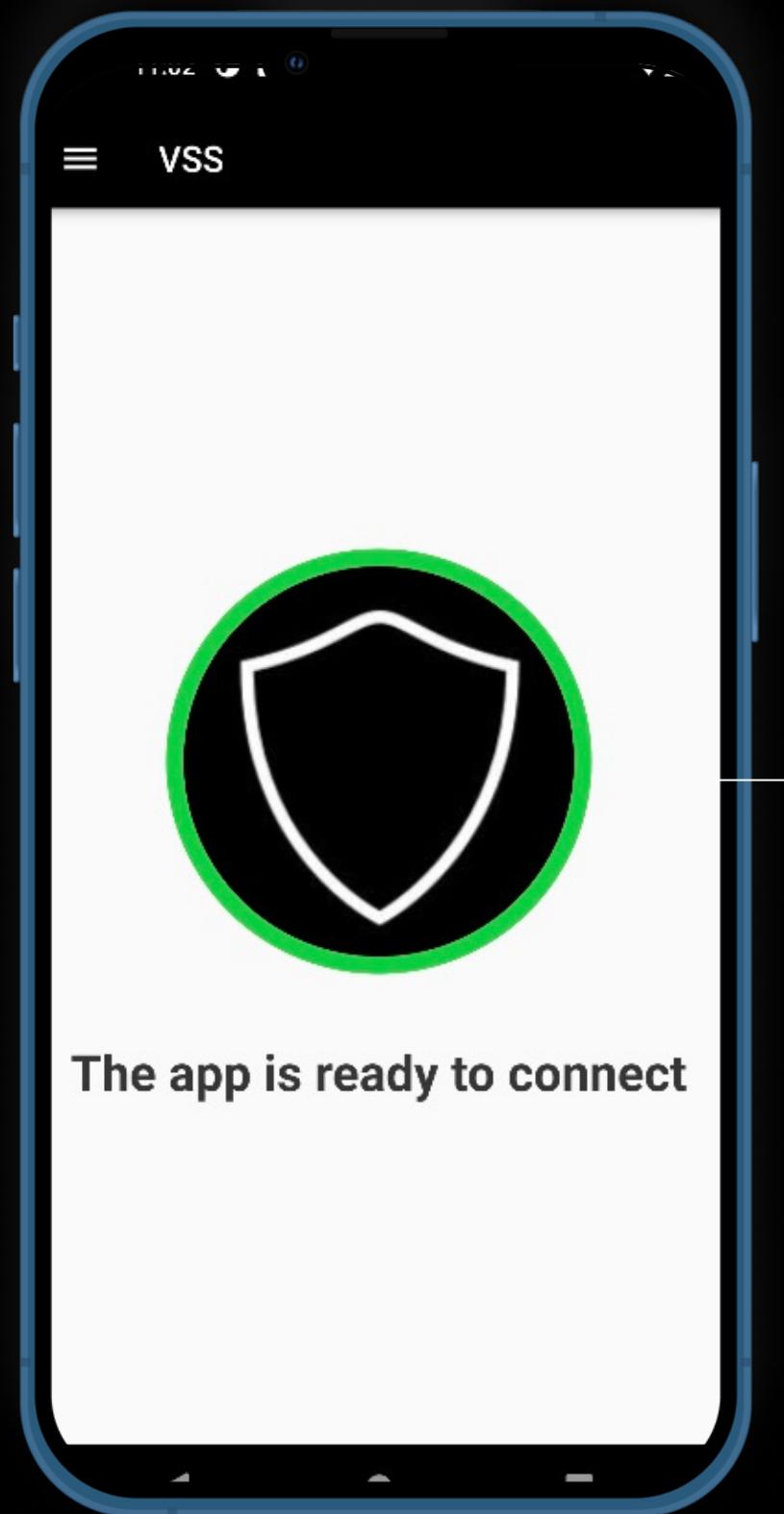


**forgot password**  
the system will  
recover the users  
password and  
send it to the  
user email



# Home page

**WHEN THE SENSOR DETECT MOTION  
THE LOGO IN THE MEDDLE WILL TURN GREEN**



# Home page

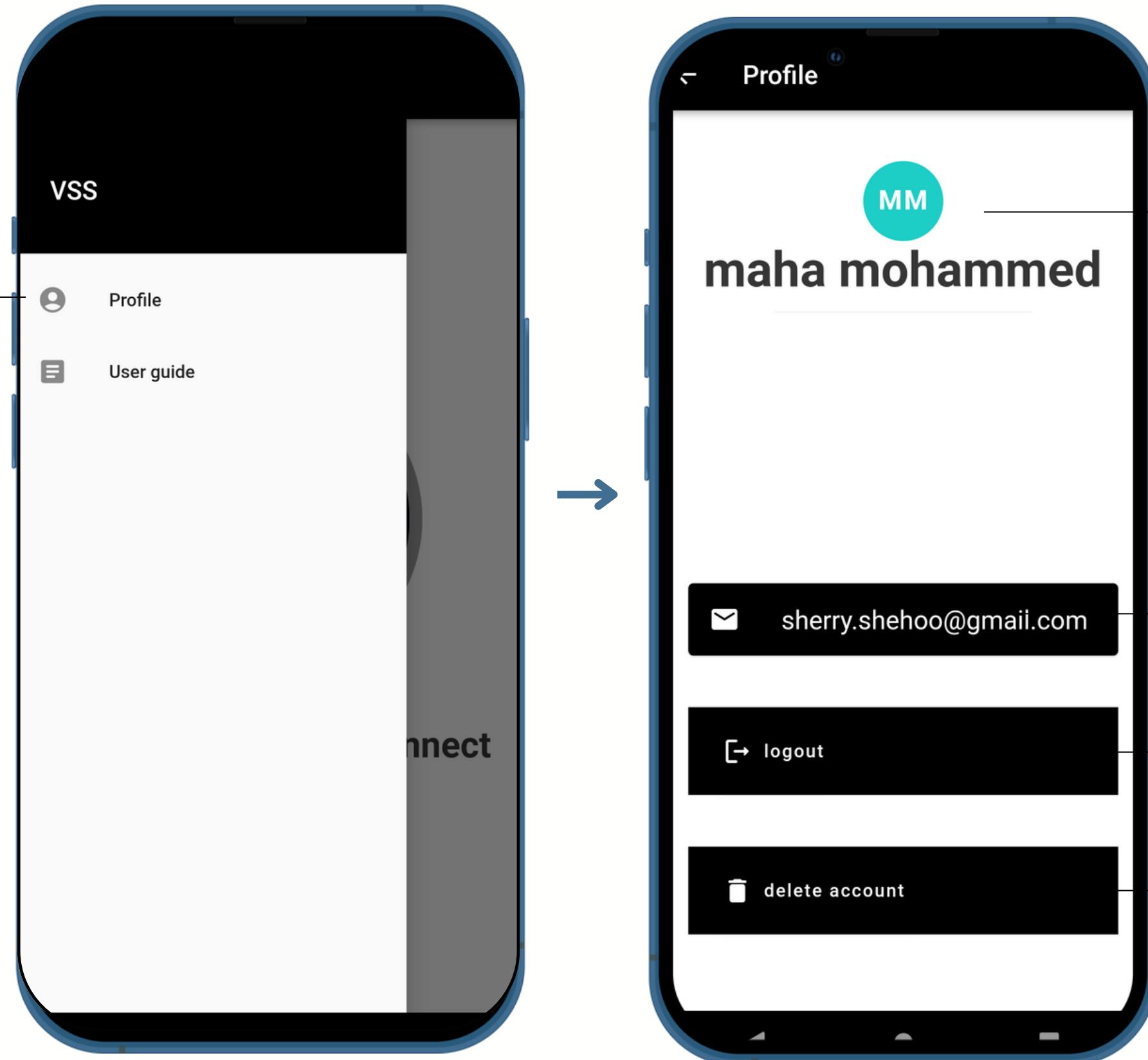
- THE LOGO TURNS GREEN
-

## Vehicle safety system

V.S.S

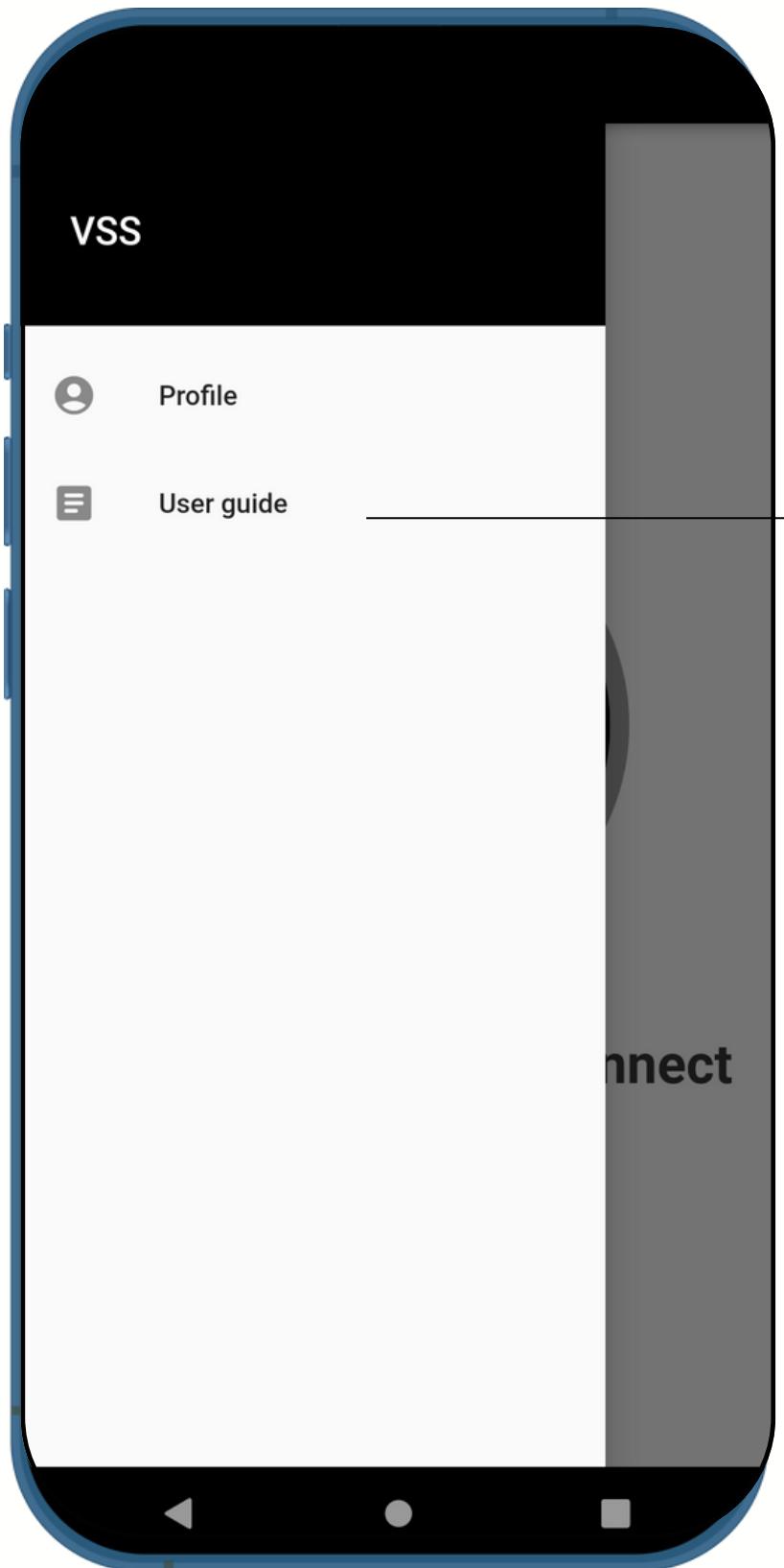
- to go to the profile page

# profile page



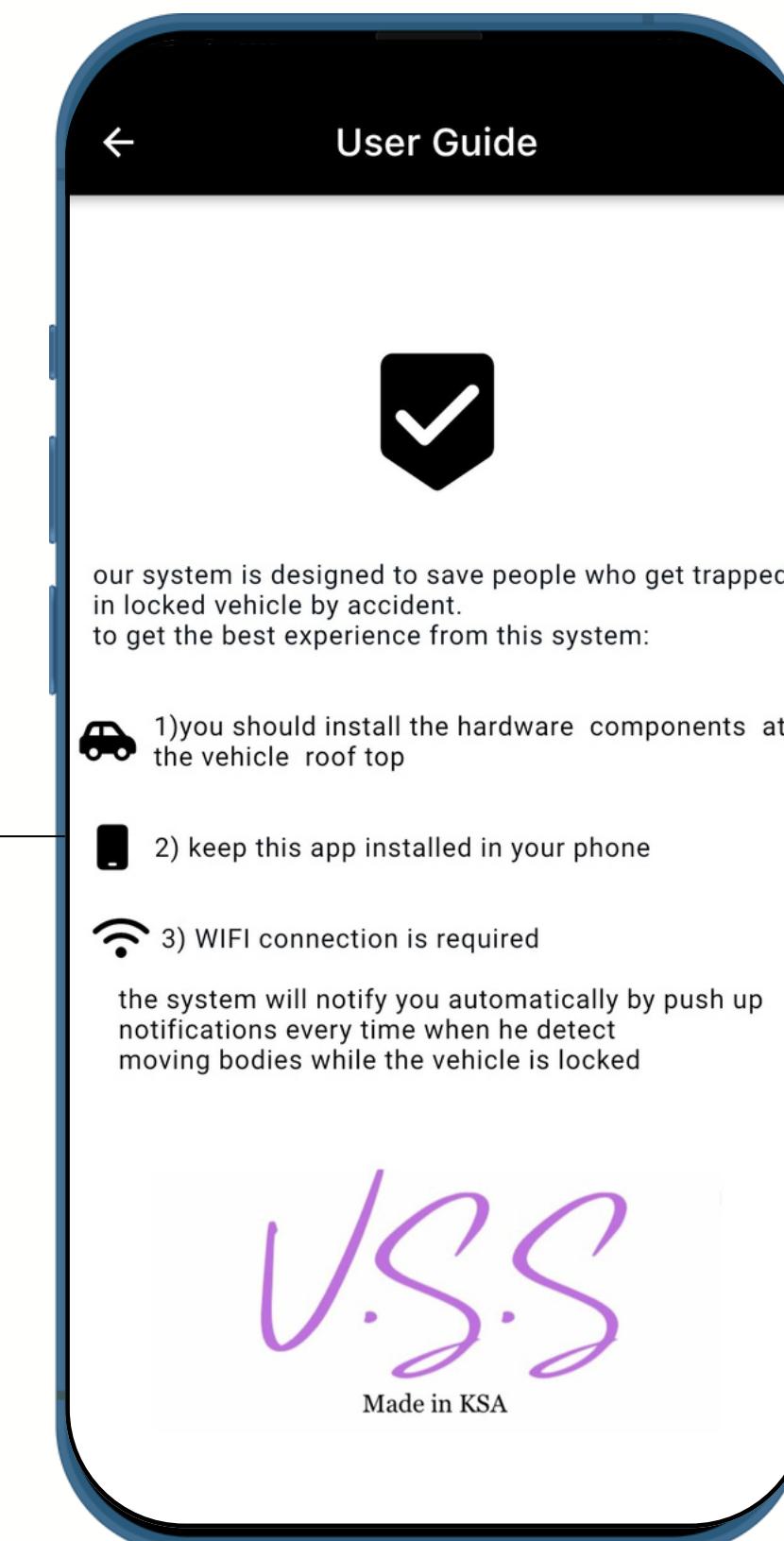
- User name
- User email address
- logout button
- delete account button

# User guide page



● **to go to the user guide page**

● **The user guide**



# program and database (functions and codes)

**NEXT →**

# Sign in code

```
),
Form(
    autovalidateMode:
        AutovalidateMode.onUserInteraction,
    child: TextFormField(
        validator: (value) =>
            value!.isEmpty || !value.contains("@")
                ? "enter a valid email"
                : null,
        style: TextStyle(color: Colors.black),
        onChanged: (value) {
            email = value;
        },
        decoration: InputDecoration(
            fillColor: Color.fromARGB(255, 255, 255, 255),
            filled: true,
            hintText: "Email",
            border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(10),
            ),
        ),
    ),
);
```

```
color: Colors.white,  
onPressed: () async {  
  try {  
    final user = await _auth  
        .signInWithEmailAndPassword(  
          email: email,  
          password: password);  
    if (user != null) {  
      Navigator.pushNamed(context, 'Email');  
    }  
  } on FirebaseAuthException catch (e) {  
    if (e.code == 'user-not-found') {  
      ScaffoldMessenger.of(context)  
          .showSnackBar(const SnackBar(  
            content: Text(  
              "No user found for that email."),  
          ));  
    }  
  }  
}
```

sherry@gmail.com

....

## Sign in



[Sign Up](#)

[Forgot Password](#)

No user found for that email.

```
    } else if (e.code == 'wrong-password') {
      ScaffoldMessenger.of(context)
        .showSnackBar(const SnackBar(
          content: Text(
            "Wrong password provided for that user."),
        ));
    }
  }
}
```

sherry.shehoo@gmail.com

..

## Sign in



[Sign Up](#)

[Forgot Password](#)

Wrong password provided for that user.

# Sign up code

```
Form(  
  autovalidateMode:  
    AutovalidateMode.onUserInteraction,  
  child: TextFormField(  
    validator: (value) =>  
      value!.isEmpty || !value.contains("@")  
        ? "enter a valid email"  
        : null,  
    style: TextStyle(  
      color: Color.fromARGB(255, 0, 0, 0)),  
    onChanged: (value) {  
      email = value;  
    },
```



```
UserCredential newUser = await _auth  
  .createUserWithEmailAndPassword(  
    email: email,  
    password: password);  
User? user = newUser.user;  
await user?.updateDisplayName(name);  
await user?.reload();  
User? latestUser =  
  FirebaseAuth.instance.currentUser;  
print('${latestUser?.displayName}');  
Navigator.pushNamed(context, 'Email');
```

```
} on FirebaseAuthException catch (e) {
    if (e.code == 'weak-password') {
        ScaffoldMessenger.of(context)
            .showSnackBar(const SnackBar(
                content: Text(
                    "The password provided is too weak."),
            )));
    }
}
```

mahaalosaimi@gmail.com

..|

**Sign Up**



Sign In

The password provided is too weak.

```
sherry.shehoo@gmail.com
```

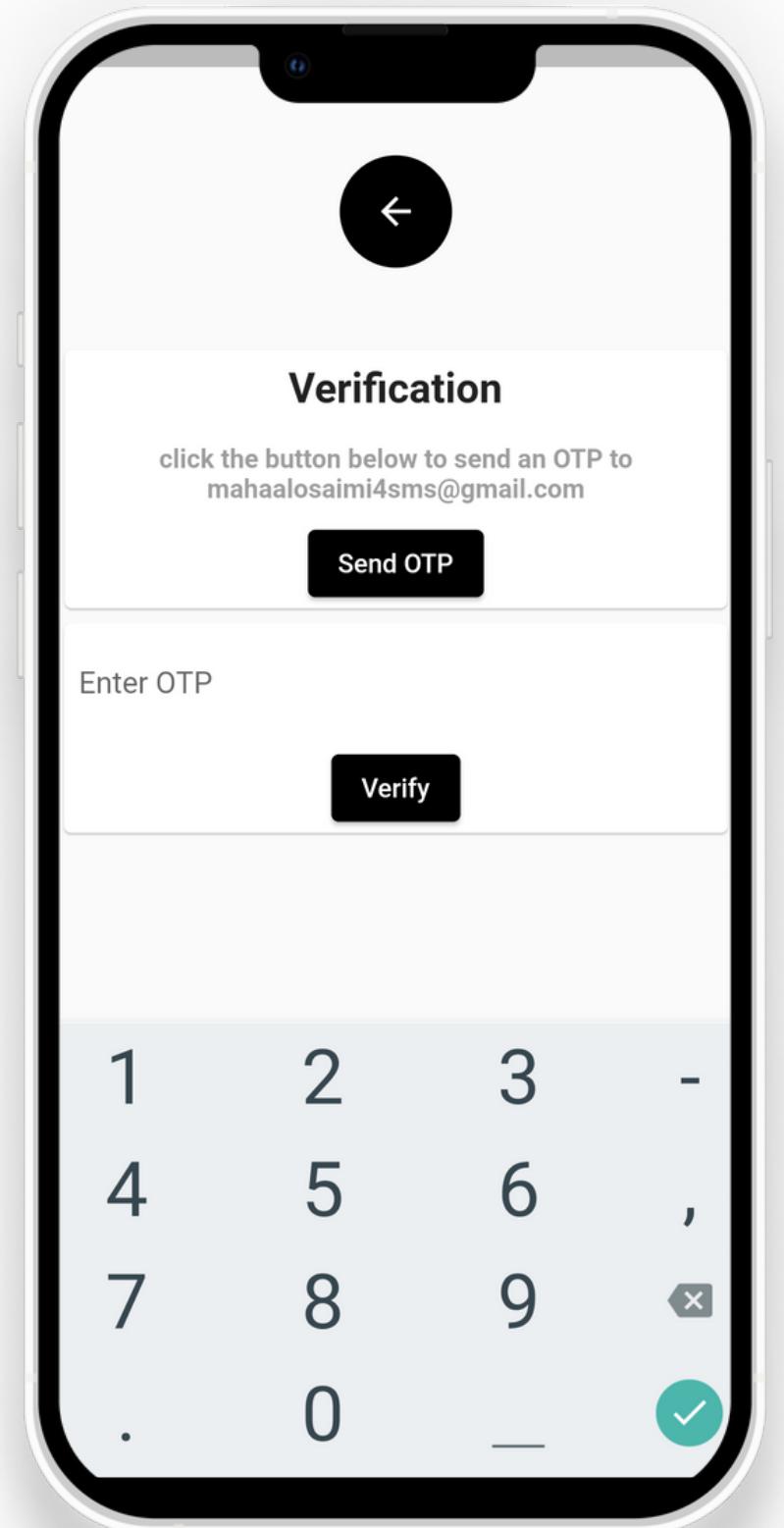
```
.....
```

## Sign Up



The account already exists for that email.

```
));
} else if (e.code ==
  'email-already-in-use') {
  ScaffoldMessenger.of(context)
    .showSnackBar(const SnackBar(
      content: Text(
        "The account already exists for that email."),
    )));
}
} catch (e) {
  print(e);
}
```

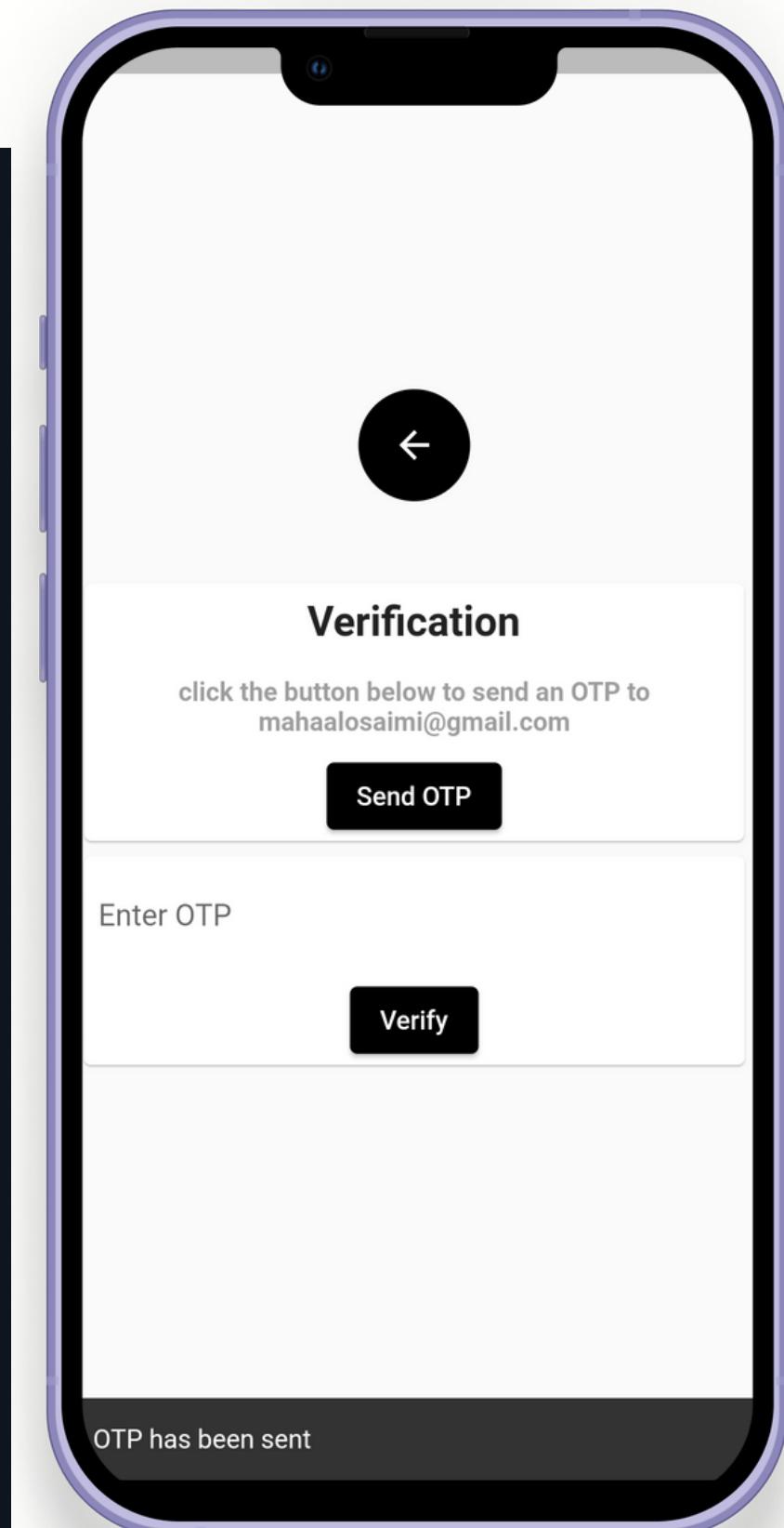


PAGE OF

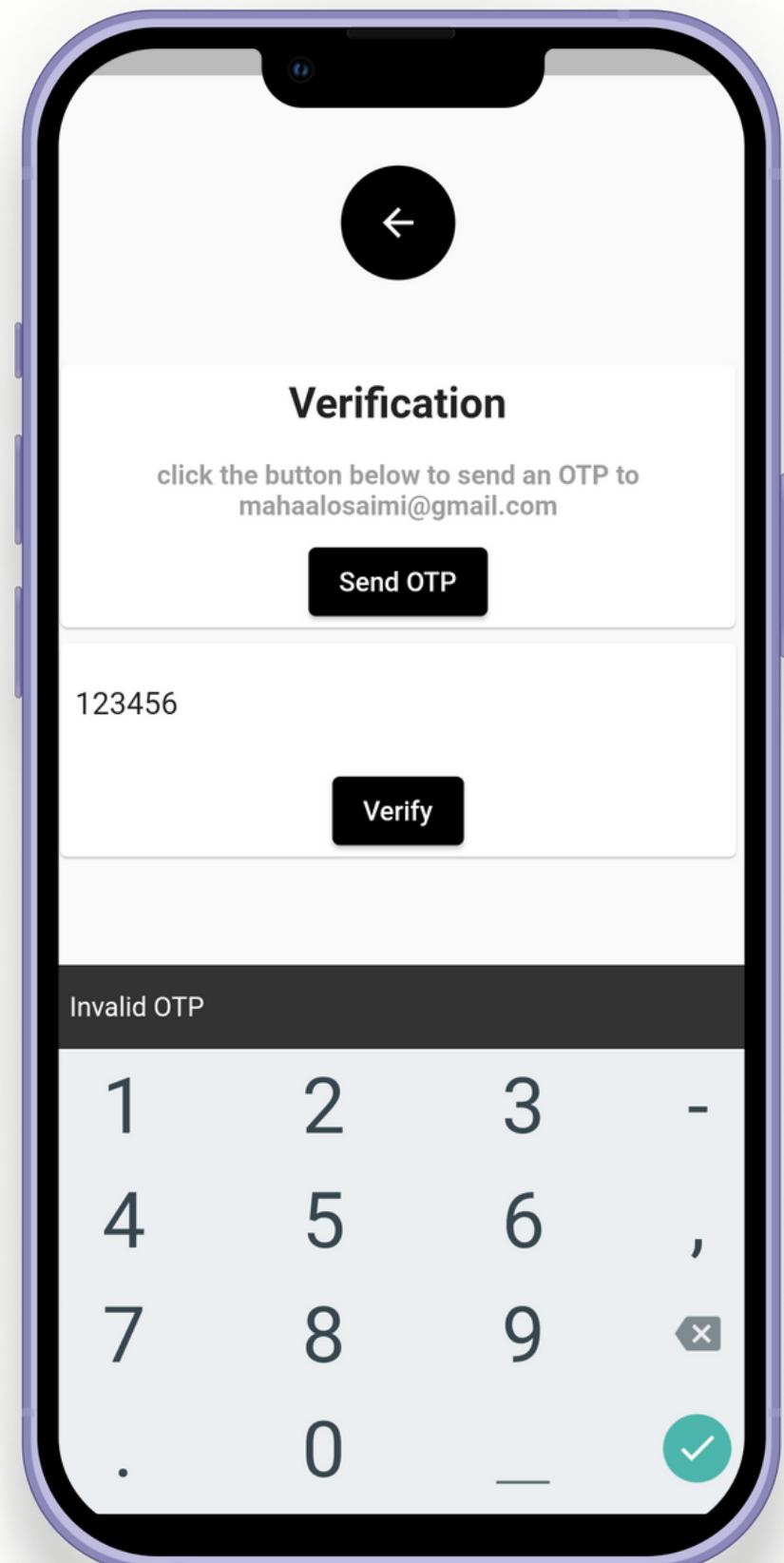
# OTP CODE

# OTP CODE

```
ElevatedButton(  
    style: ElevatedButton.styleFrom(  
        backgroundColor: Color.fromARGB(255, 0, 0, 0),  
    ),  
    onPressed: () async {  
        myauth.setConfig(  
            appEmail: "me@rohitchouhan.com",  
            appName: "Email OTP",  
            userEmail: email.text,  
            otpLength: 6,  
            otpType: OTPType.digitsOnly);  
        if (await myauth.sendOTP() == true) {  
            ScaffoldMessenger.of(context)  
                .showSnackBar(const SnackBar(  
                    content: Text("OTP has been sent"),  
                ));  
        } else {  
            ScaffoldMessenger.of(context)  
                .showSnackBar(const SnackBar(  
                    content: Text("Oops, OTP send failed"),  
                ));  
        }  
    },  
    child: const Text("Send OTP")),  
,  
,
```



# OTP CODE

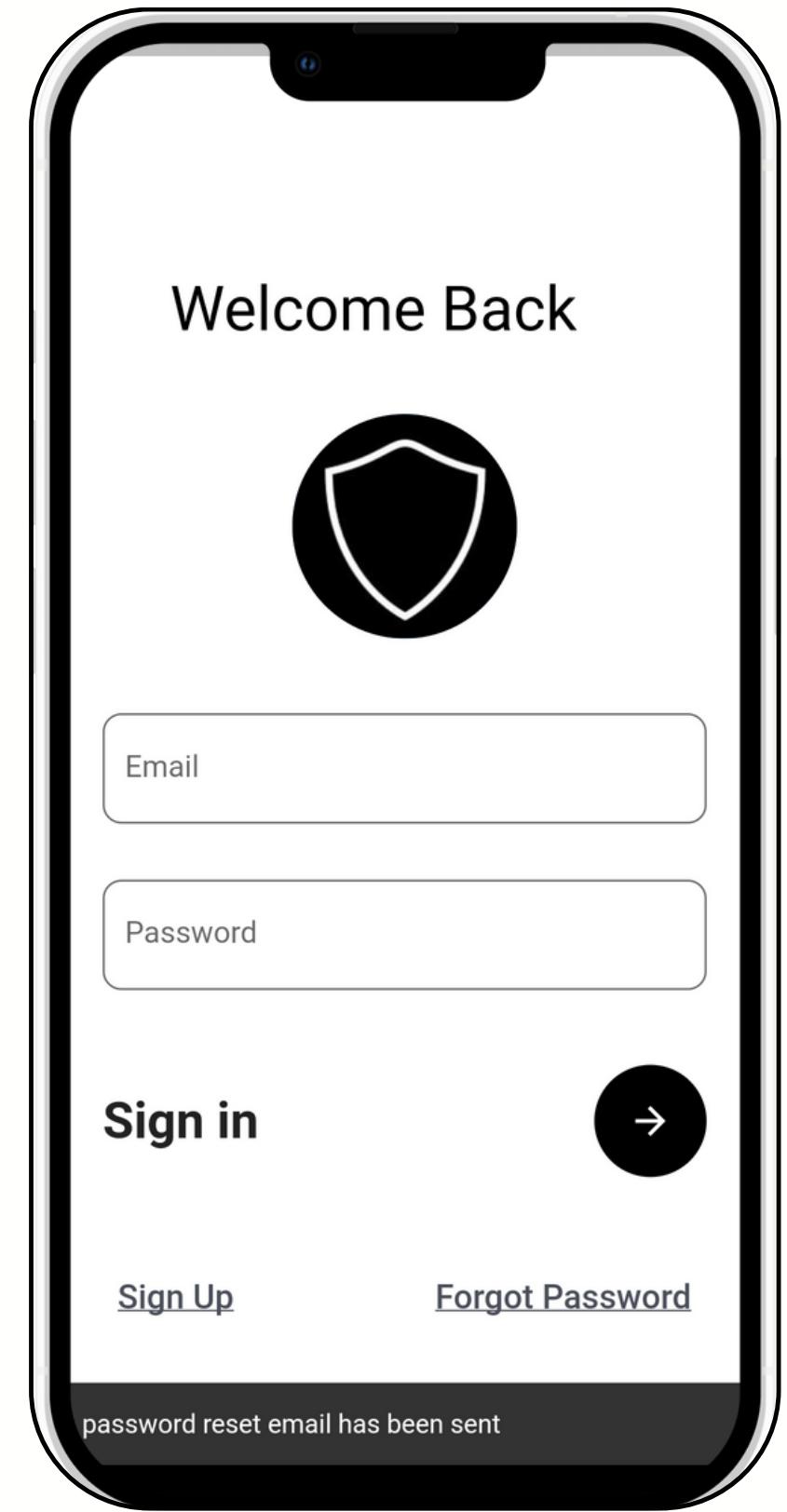
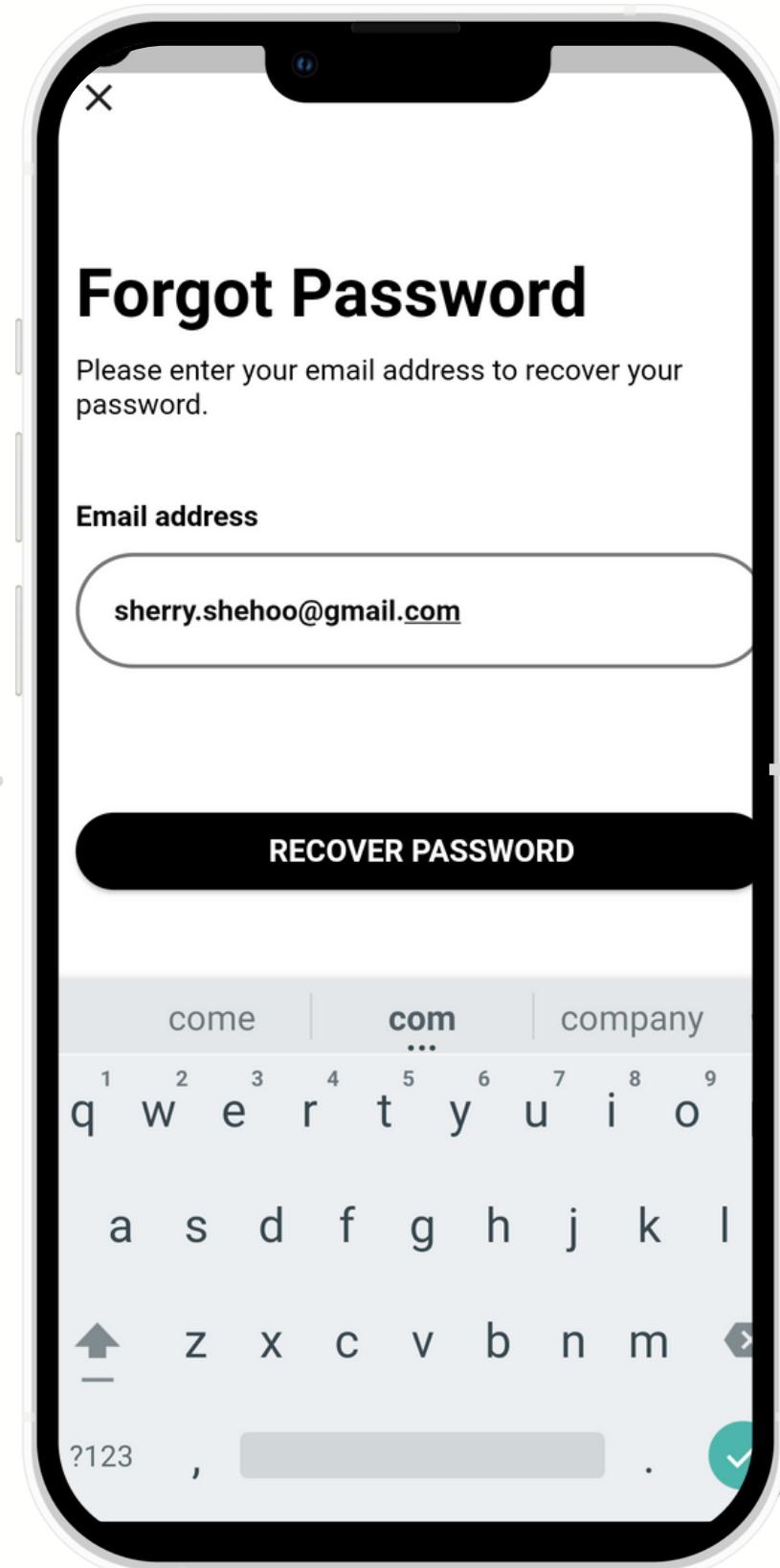
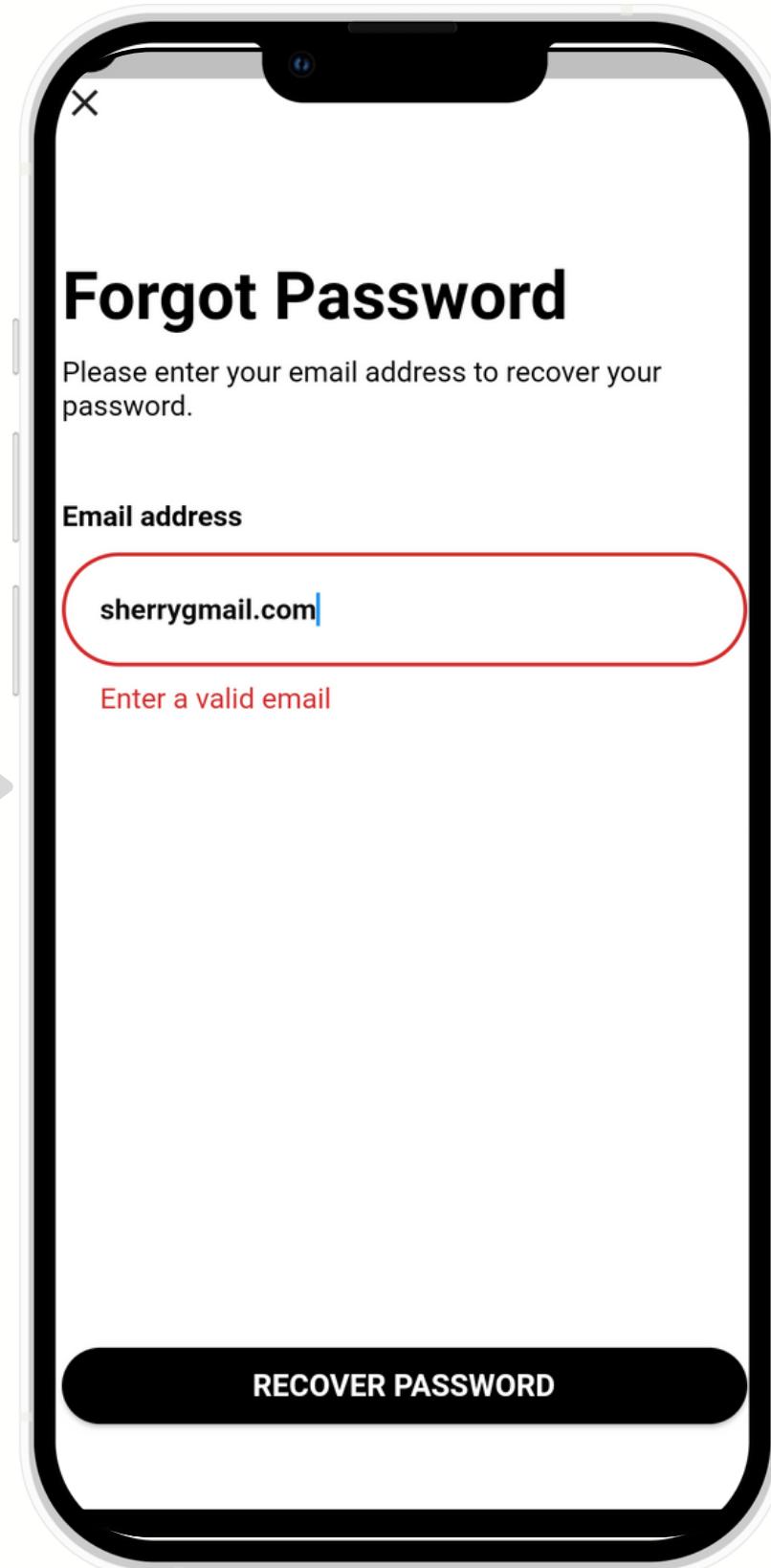
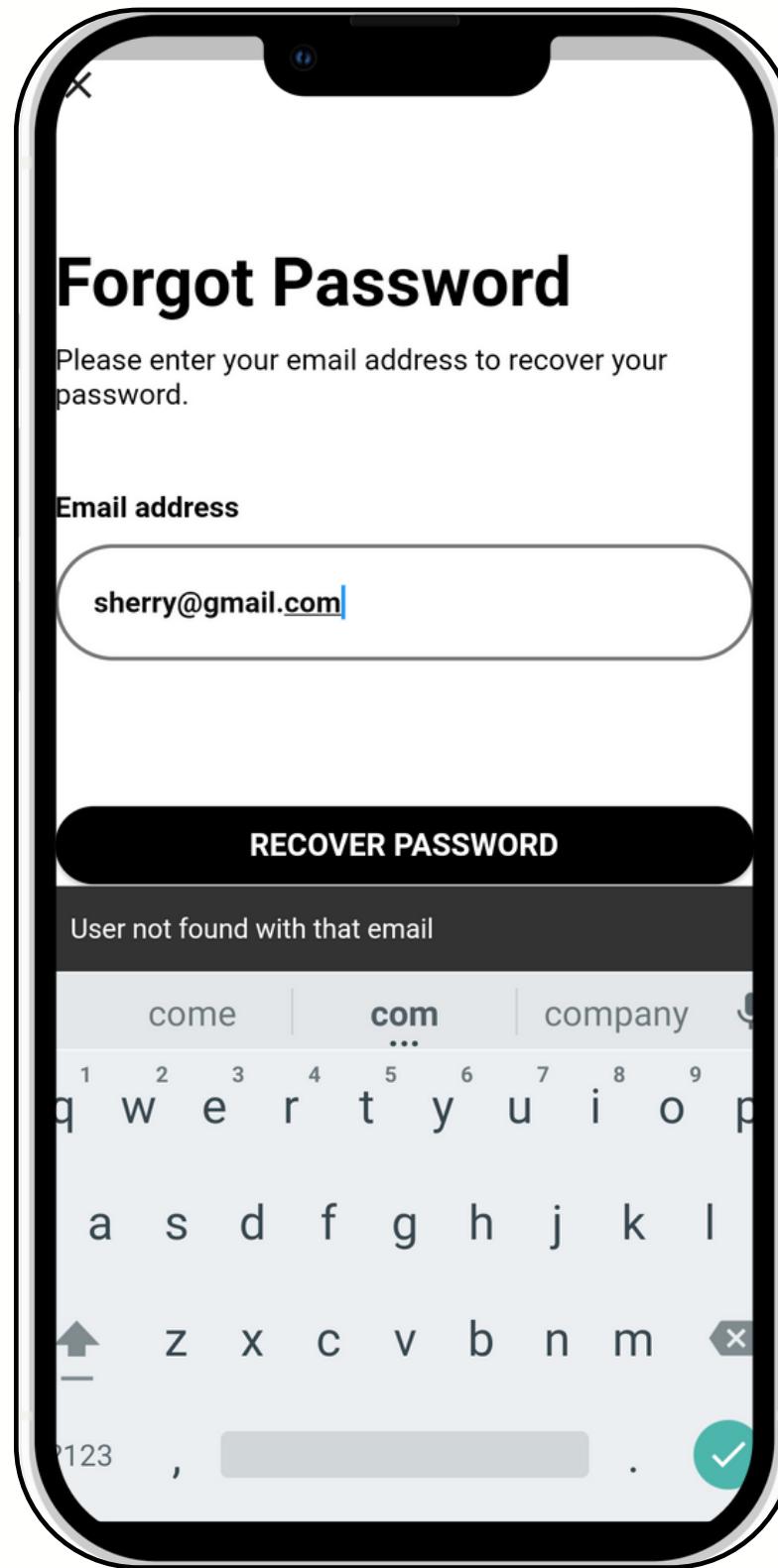


```
),
ElevatedButton(
  style: ElevatedButton.styleFrom(
    backgroundColor: Color.fromARGB(255, 0, 0, 0),
  ),
  onPressed: () async {
    if (await myauth.verifyOTP(otp: otp.text) == true) {
      ScaffoldMessenger.of(context)
        .showSnackBar(const SnackBar(
          content: Text("OTP is verified"),
        ));
      Navigator.pushNamed(context, 'home_page');
    } else {
      ScaffoldMessenger.of(context)
        .showSnackBar(const SnackBar(
          content: Text("Invalid OTP"),
        ));
    }
  },
  child: const Text("Verify")),
],
```

# forgot password code

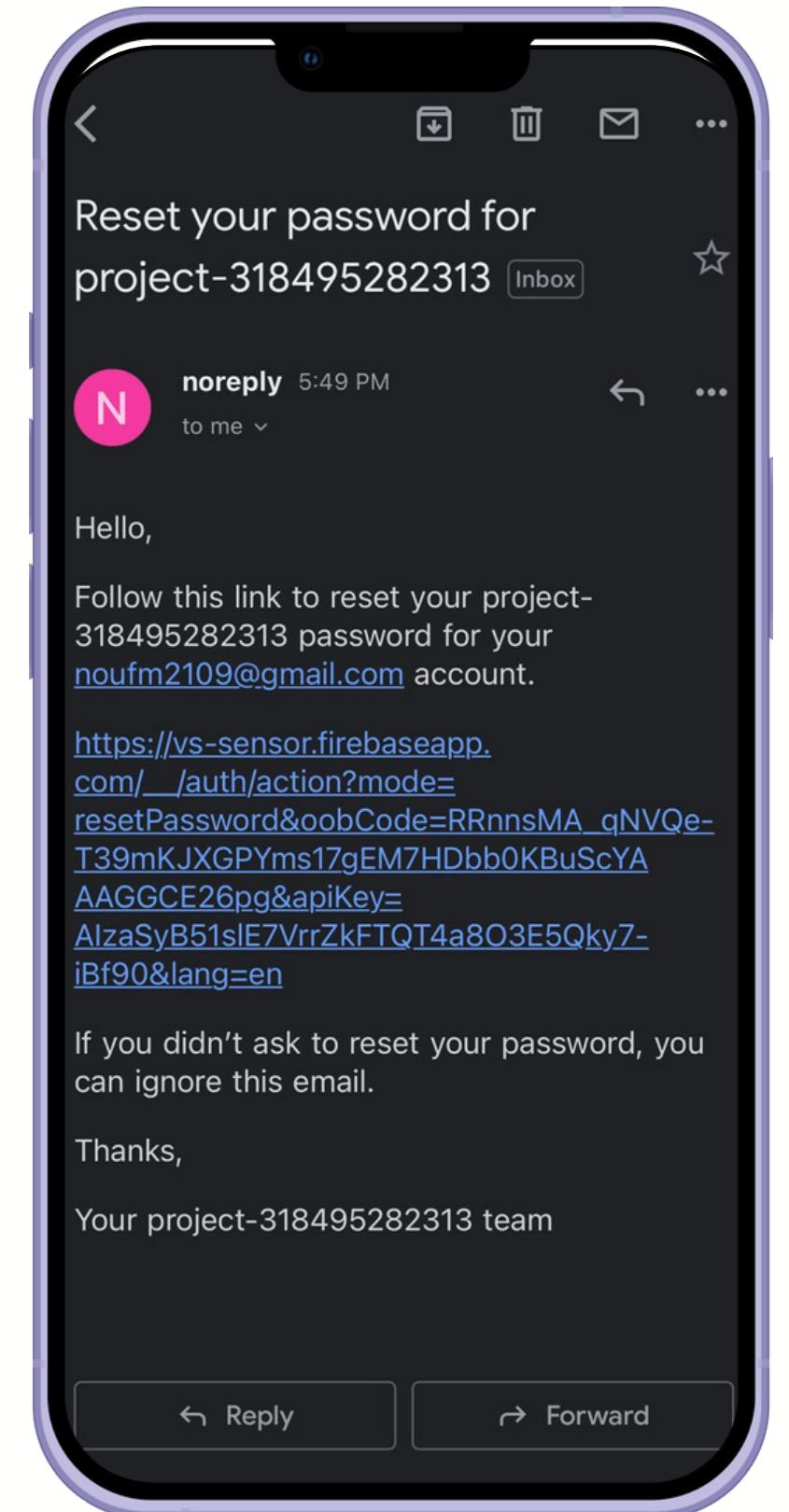
```
email: _emailController.text.trim());  
if (_status == AuthStatus.successful) {  
  ScaffoldMessenger.of(context)  
    .showSnackBar(const SnackBar(  
      content:  
        Text("password reset email has been sent"),  
    ));  
  Navigator.pop(context);  
} else {  
  ScaffoldMessenger.of(context)  
    .showSnackBar(const SnackBar(  
      content: Text("User not found with that email"),  
    ));  
}  
}
```

```
child: Form(  
  autovalidateMode: AutovalidateMode.onUserInteraction,  
  child: TextFormField(  
    obscureText: false,  
    controller: _emailController,  
    validator: (value) =>  
      value!.isEmpty || value.contains("@")  
        ? "Enter a valid email"  
        : null,  
  
    autofocus: false,  
    style: const TextStyle(  
      fontSize: 15,  
      fontWeight: FontWeight.bold,  
      color: Color.fromARGB(255, 16, 16, 16),  
    ),  
    decoration: const InputDecoration(  
      contentPadding:  
        EdgeInsets.symmetric(vertical: 20, horizontal: 20),  
      border: OutlineInputBorder(  
        borderSide: BorderSide(  
          width: 2,  
          color: Colors.purple  
        )  
      )  
    )  
  )  
);
```

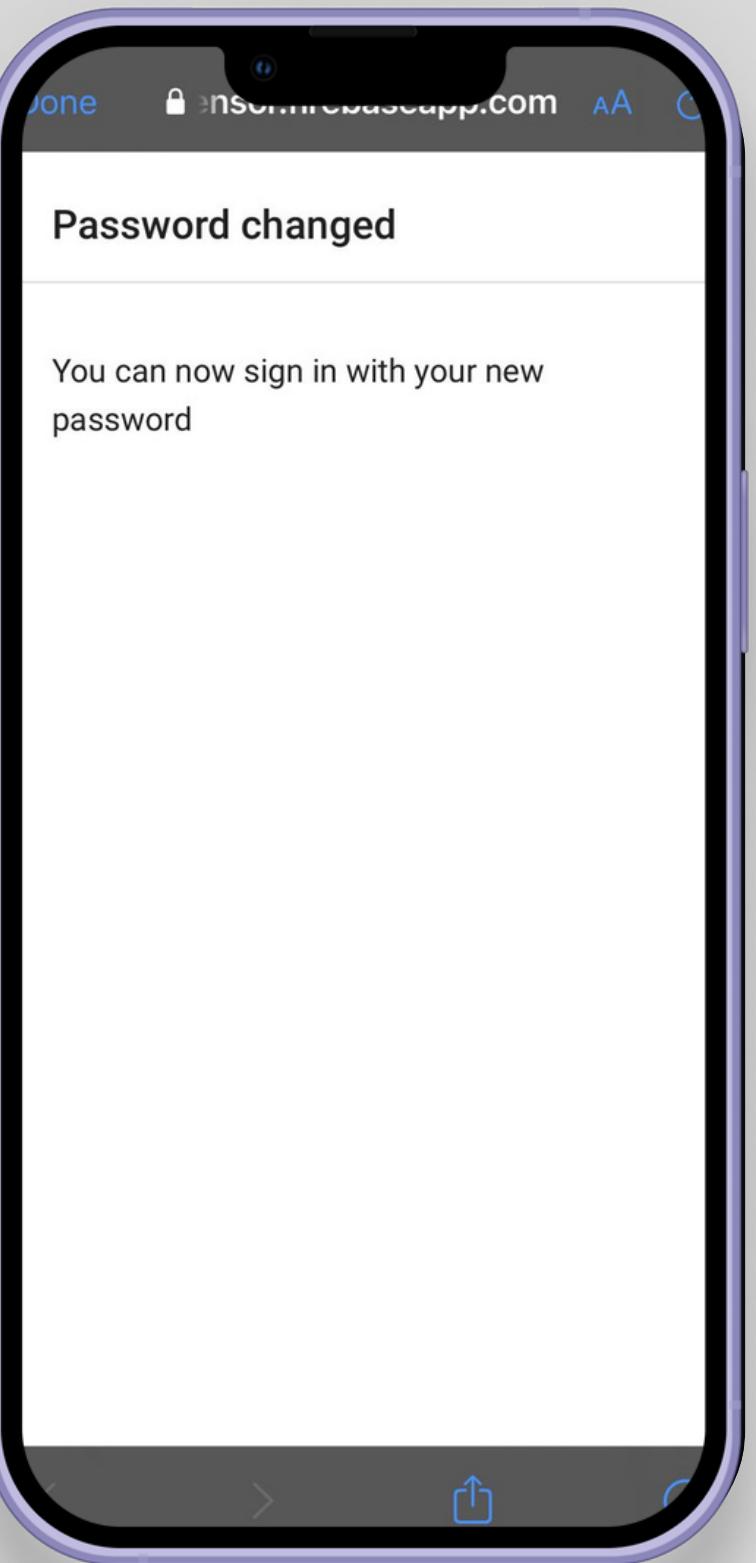


**FORGOT PASSWORD**

# Massage to reset password

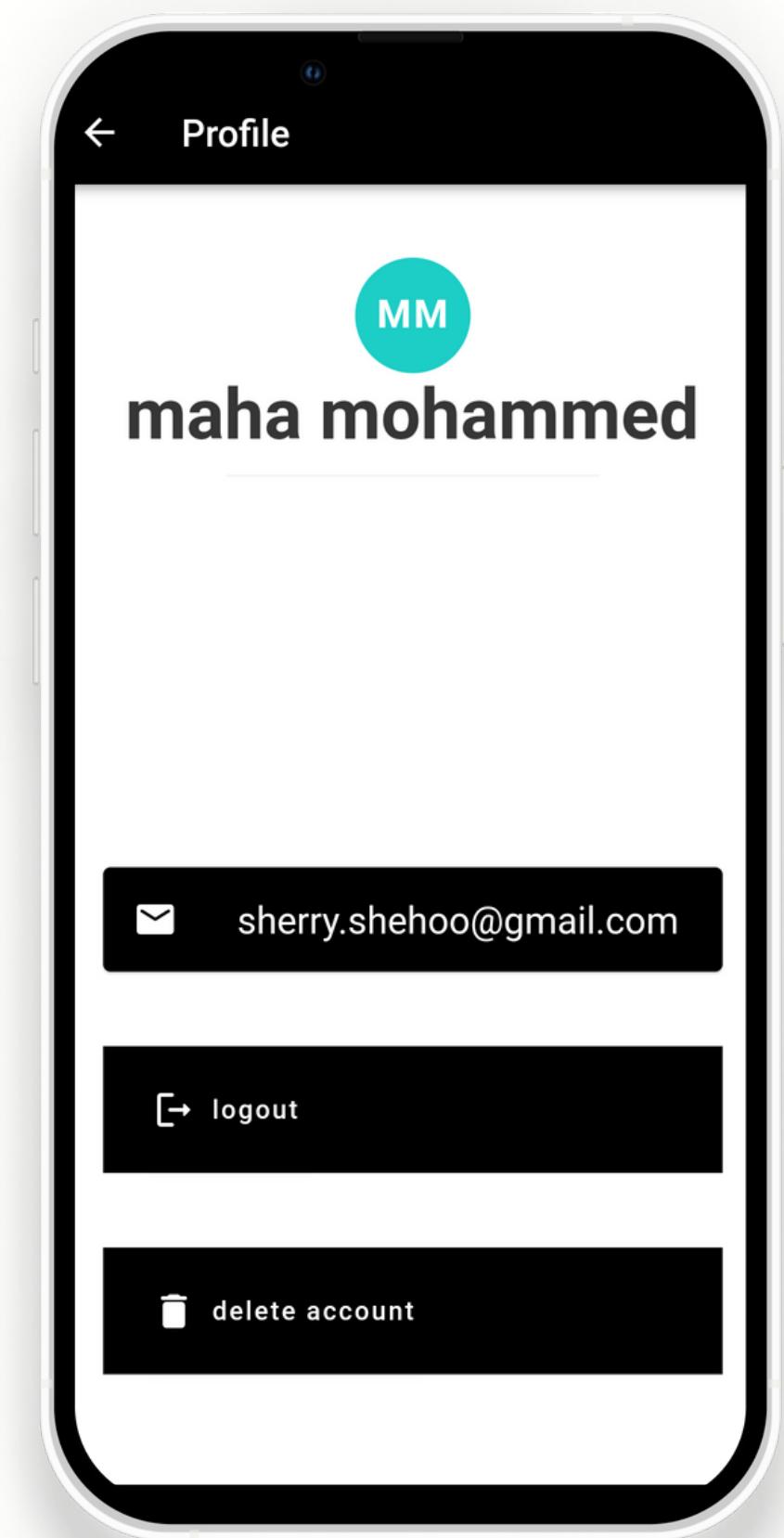


# Forgot password code

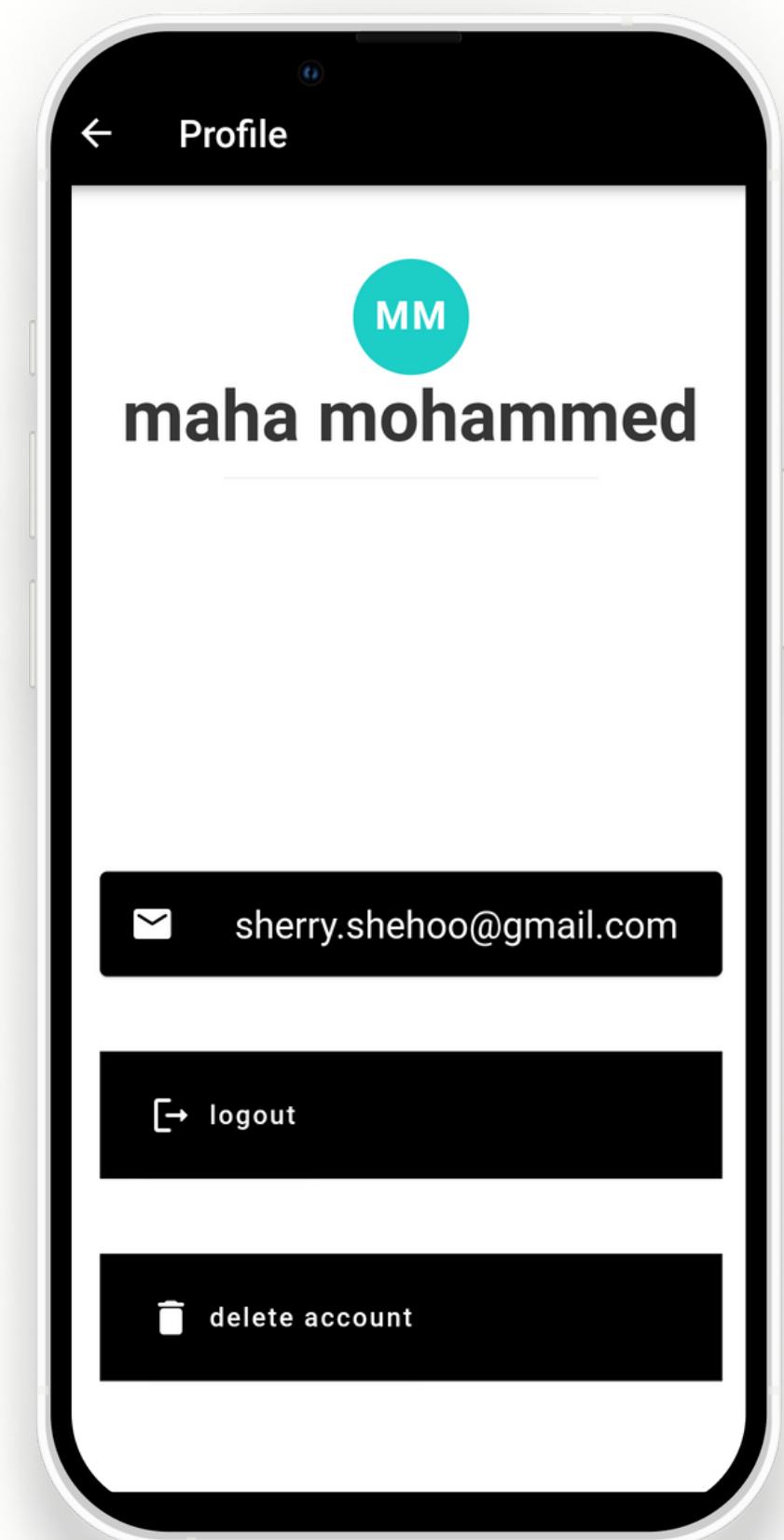


# Profile

```
ProfilePicture(  
    name: '${latestUser?.displayName}',  
    radius: 31,  
    fontsize: 21,  
    random: true,  
    count: 2,  
),  
Text(  
    '${latestUser?.displayName}',  
    style: TextStyle(  
        fontSize: 38.0,  
        color: Color.fromARGB(255, 55, 54, 54),  
        fontWeight: FontWeight.bold,  
    ),
```

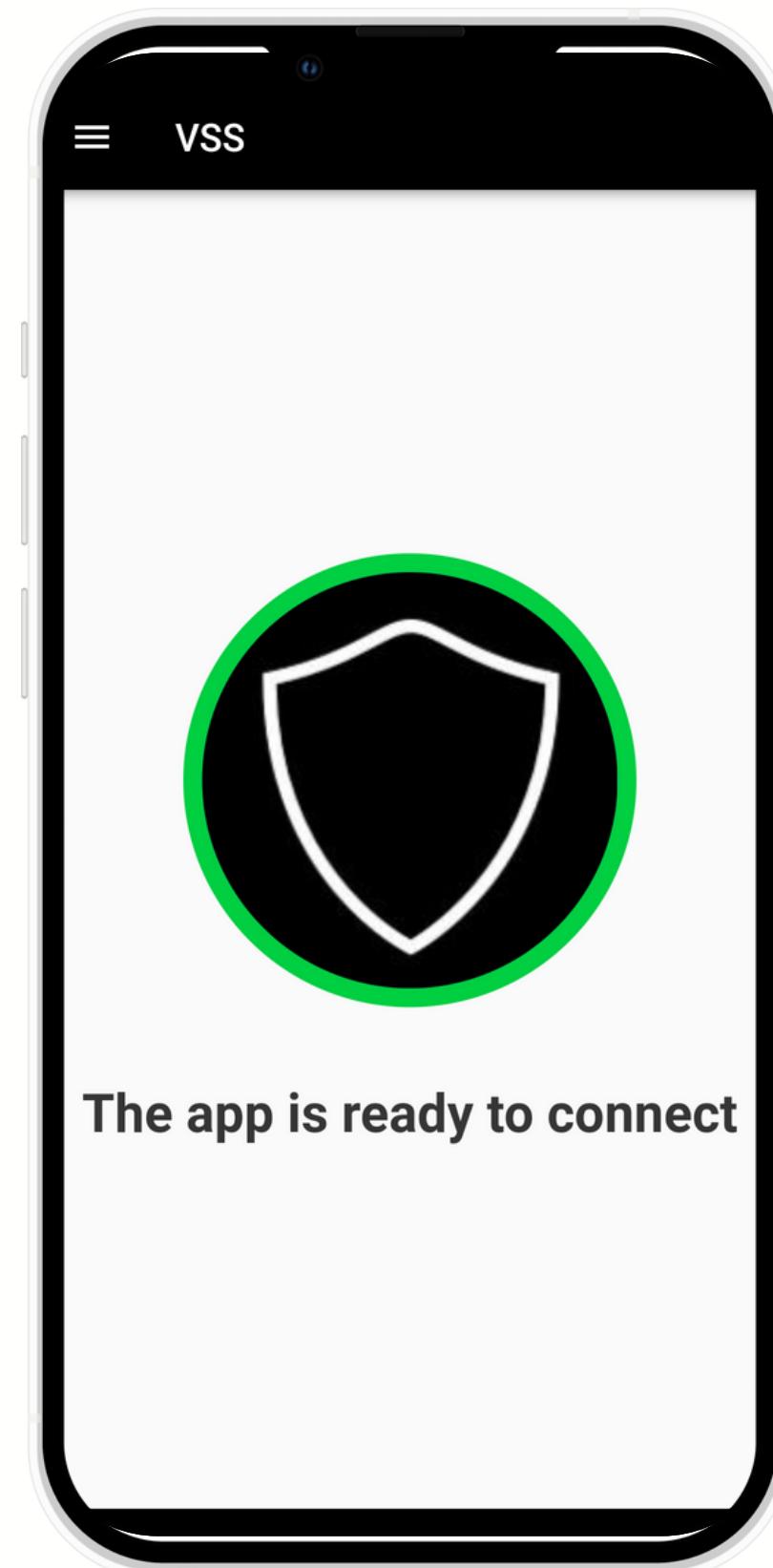


# Profile



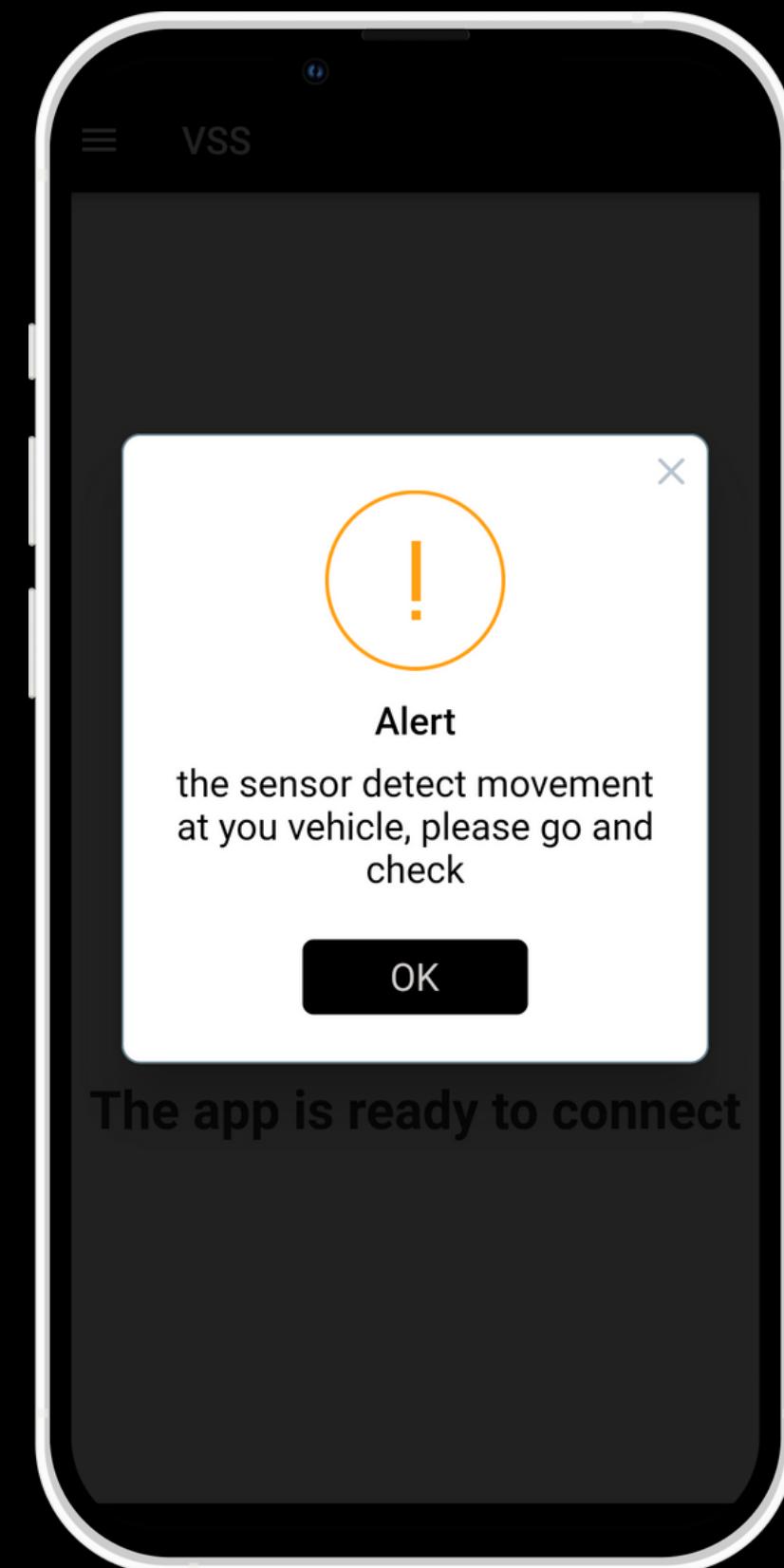
```
children: [
    FloatingActionButton.extended(
        heroTag: "logout",
        onPressed: () async {
            await FirebaseAuth.instance.signOut();
            Navigator.pushReplacementNamed(context, "login");
        },
        icon: Icon(
            Icons.logout_rounded,
            size: 24.0,
        ), // Icon
        label: Text('logout'),
        backgroundColor: Colors.black,
    ), // FloatingActionButton.extended
],
), // Row
), // Container
Container(
    margin: EdgeInsets.fromLTRB(30.0, 20.0, 30.0, 20.0),
    padding: EdgeInsets.all(10.0),
    color: Colors.black,
    child: Row(
        children: [
            FloatingActionButton.extended(
                heroTag: "Delete",
                onPressed: () async {
                    await user.delete();
                    Navigator.pushNamed(context, 'register');
                },
                icon: Icon(
                    Icons.delete_forever_rounded,
                    size: 24.0,
                ), // Icon
                label: Text('Delete account'),
                backgroundColor: Colors.black,
            ), // FloatingActionButton.extended
],
),
```

# Home page



# Home page

# Notification



# Notification

```
FirebaseDatabase database = FirebaseDatabase.instance;
DatabaseReference ref = FirebaseDatabase.instance.ref("State");
late String data;
//late String statedata;
String dataOut = '';

@Override
void initState() {
    super.initState();
    read();
}
```

```
Future read() async {
    //DatabaseReference starCountRef = FirebaseDatabase.instance.ref('State'); old comments
    ref.onValue.listen(DatabaseEvent event) {
        data = event.snapshot.value.toString();
        if (data == 'detected') {
            Alert(
                context: context,
                title: "Alert",
                type: AlertType.warning,
                desc:
                    "the sensor detect movement at you vehicle, please go and check",
                buttons: [
                    DialogButton(
                        child: Text(
                            "OK",
                            style: TextStyle(
                                color: Color.fromARGB(255, 211, 206, 206), fontSize: 20), // TextStyle
                        ), // Text
                        onPressed: () => Navigator.pop(context),
                        color: Colors.black,
                        width: 120,
                    ) // DialogButton
                ],
            ).show(); // Alert
        }
        print(data);
        //return data;
        data = data;
        setState(() {
            this.dataOut = data;
        });
    });
}
```

**Vehicle safety system**

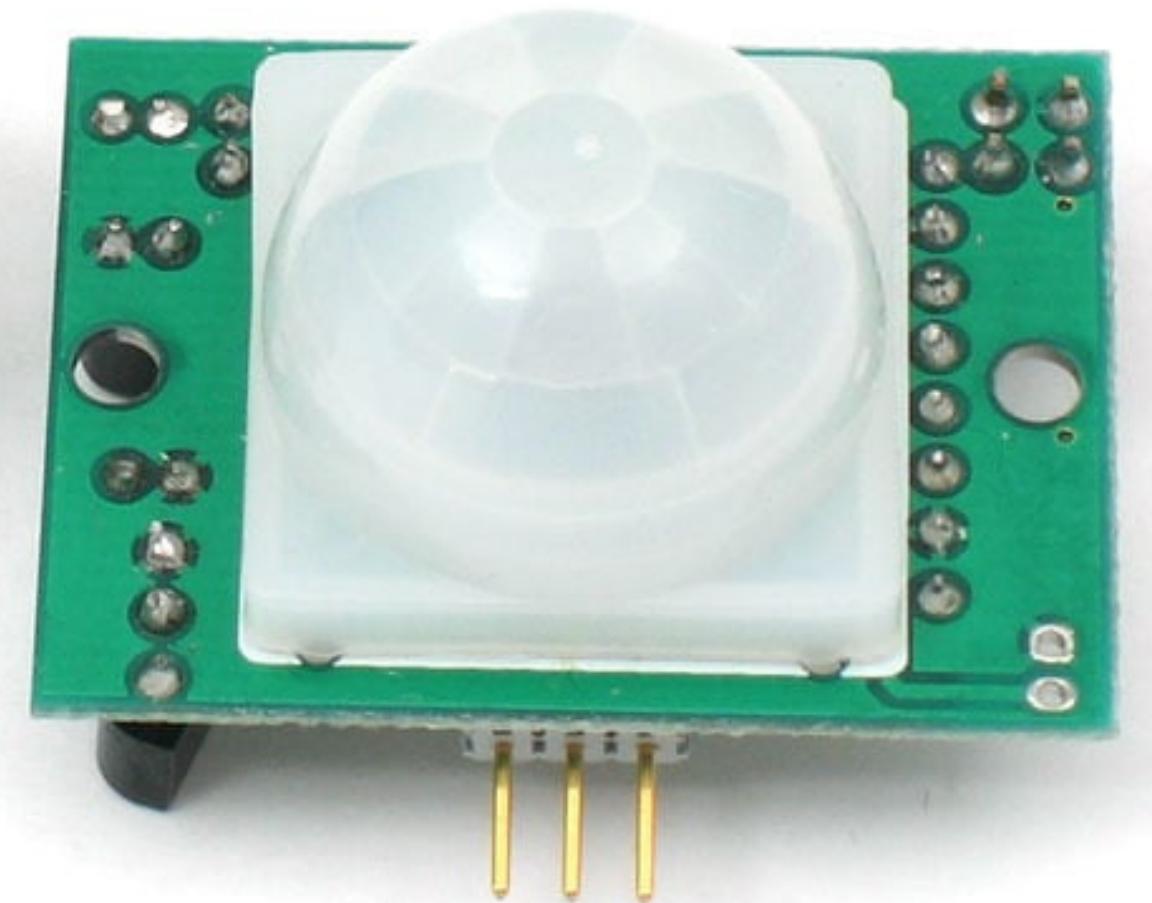
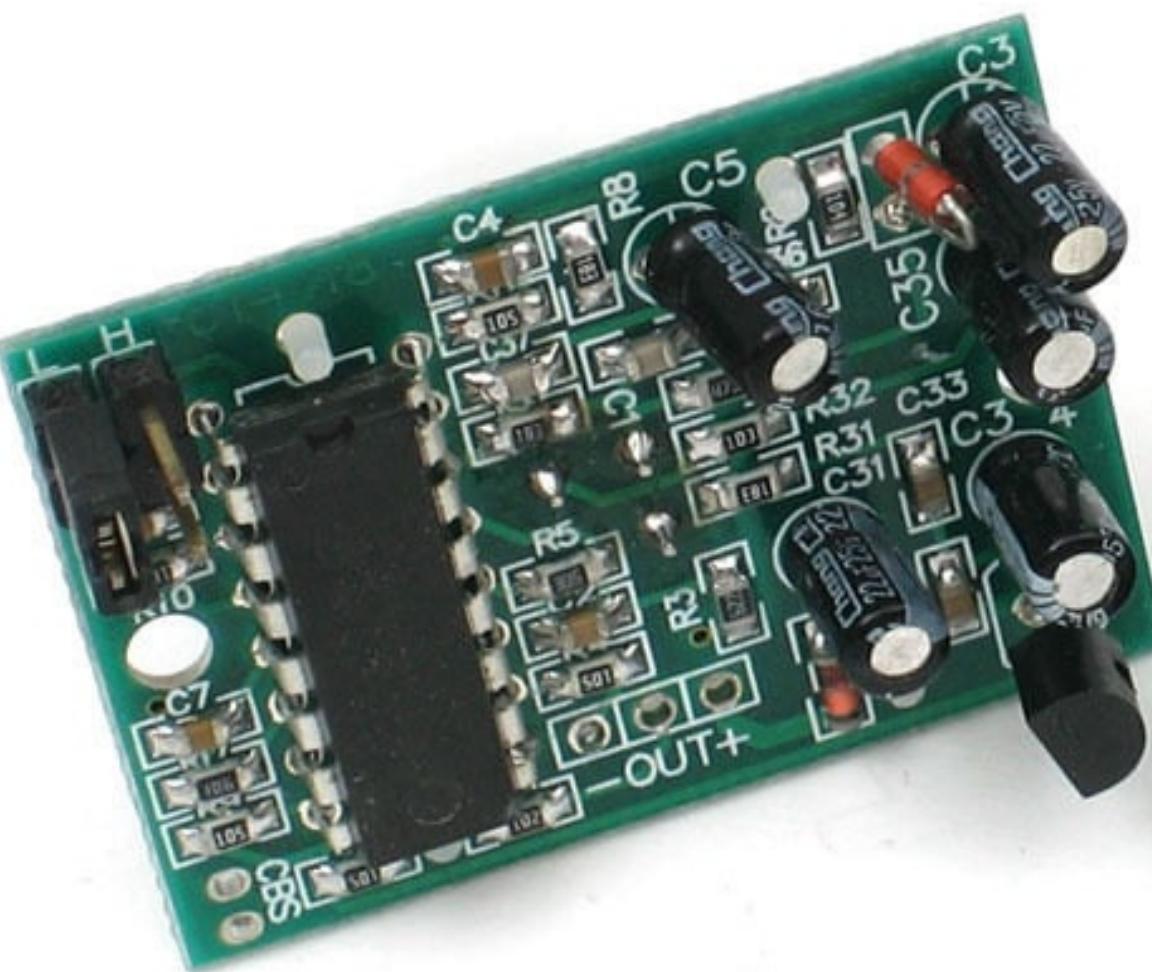
V.S.S

**PIR SENSOR**

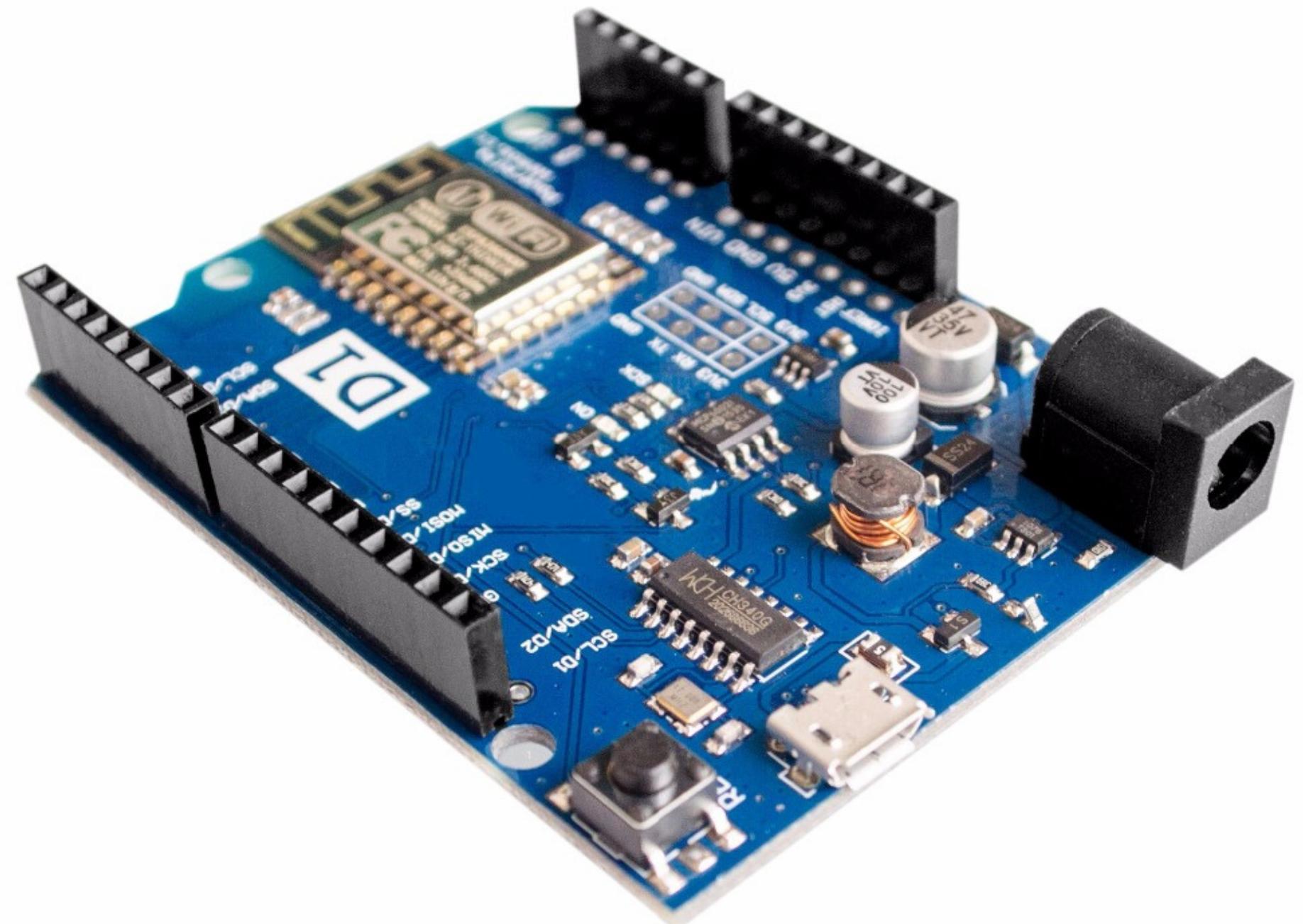
# HARDWARE

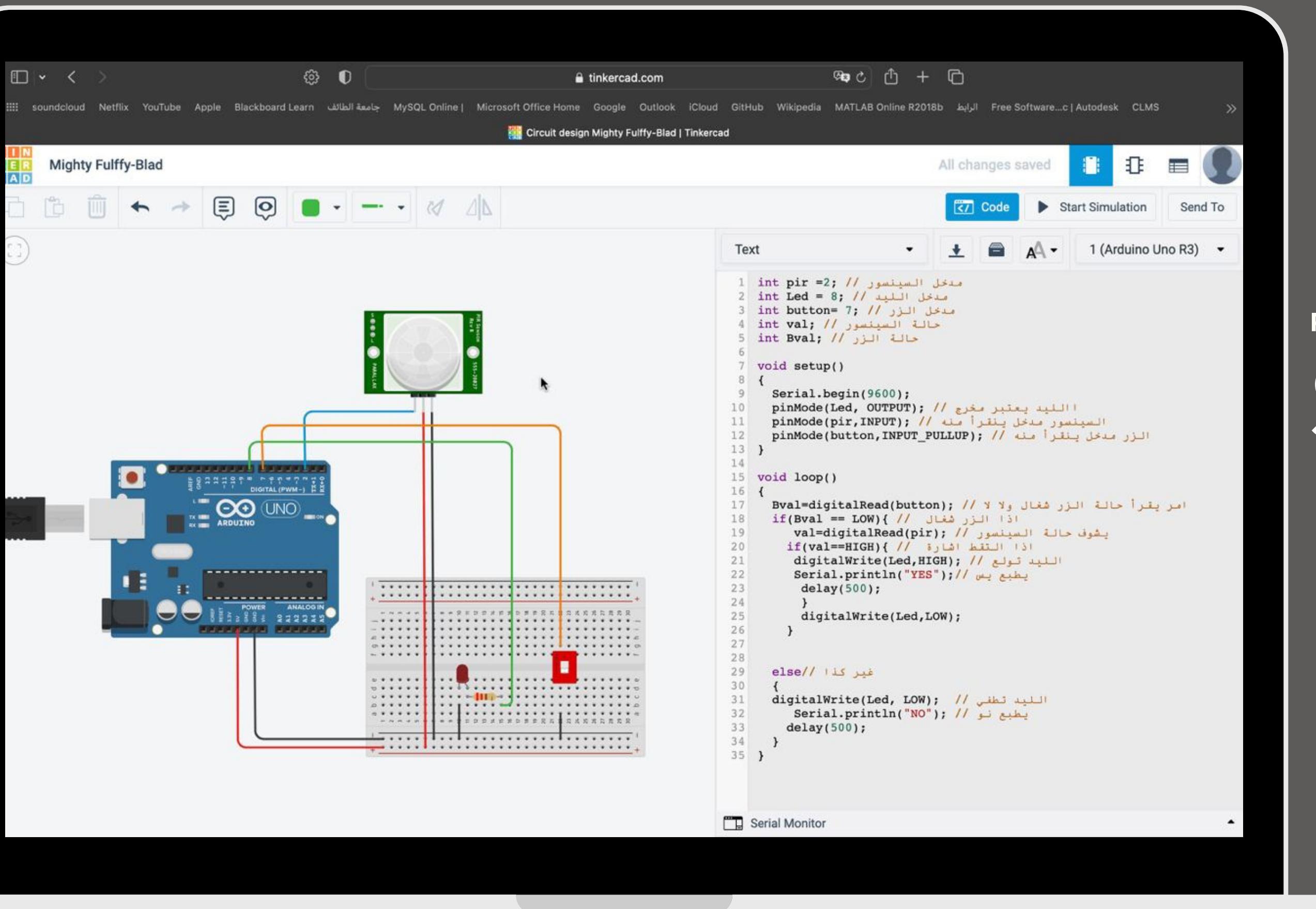
NEXT →

# PIR SENSOR



# WIFI ARDUINO

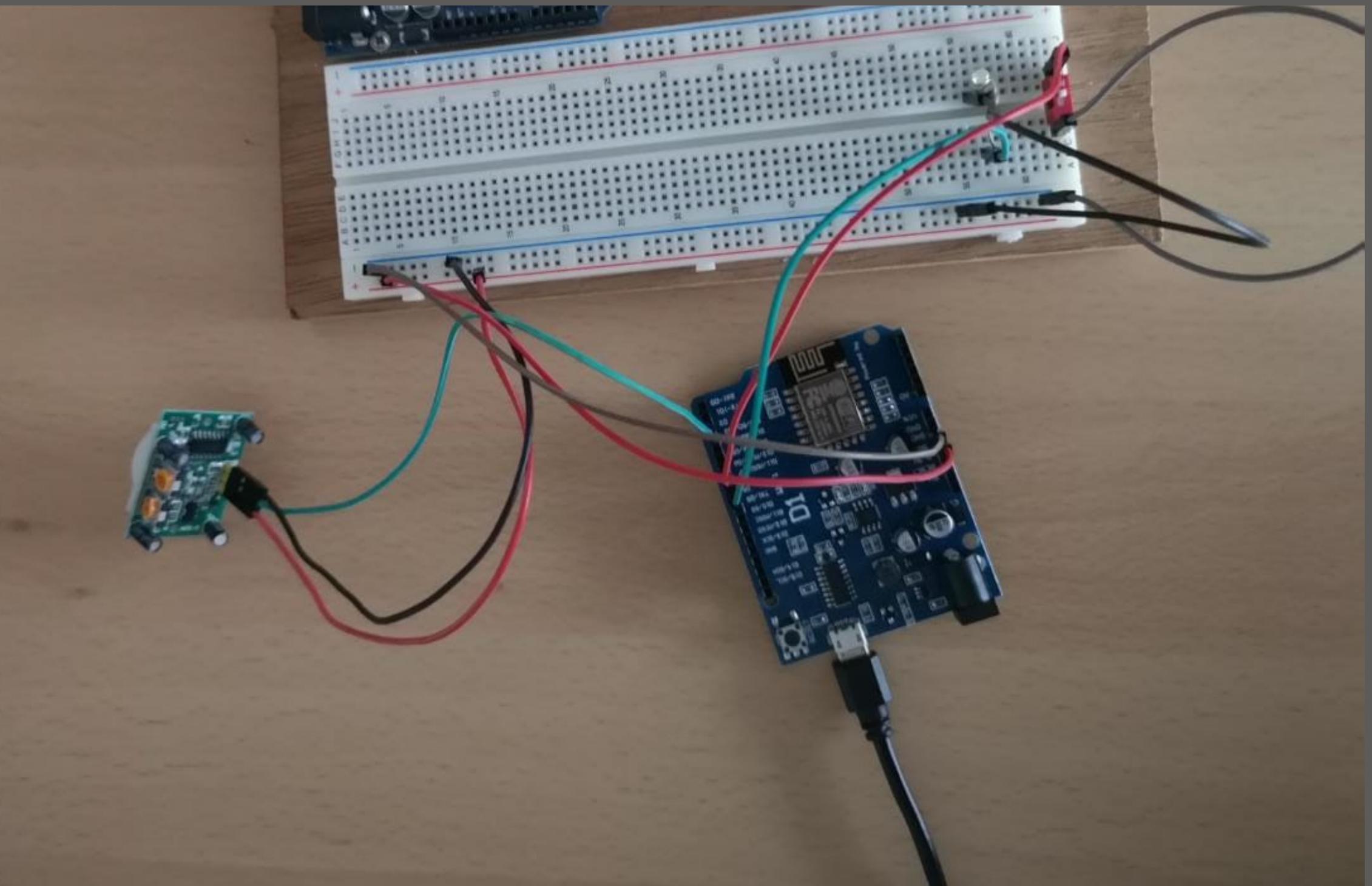




# PIR SENSOR Simulations

PIR SENSOR

# Results



PIR SENSOR

# Code

```
void loop()
{
    Bval=digitalRead(button);
    if(Bval == LOW){
        val=digitalRead(pir);
        if(val==HIGH){
            digitalWrite(Led,HIGH);
            Serial.println("motion");
            Firebase.setString(firebaseData,"State",motion);
            delay(500);
        }
        digitalWrite(Led,LOW);
    }
    else
    {
        digitalWrite(Led, LOW);
        Serial.println("no motion");
        Firebase.setString(firebaseData, "State",nomotion);
        delay(500);
    }
}
```

# Realtime Database

Data

Rules

Backups

Usage

🔗 <https://vs-sensor-default-rtdb.firebaseio.com>



<https://vs-sensor-default-rtdb.firebaseio.com/>

└ State: "nomotion"

PIR SENSOR

Database

# Realtime Database

Data    Rules    Backups    Usage

https://vs-sensor-default-rtdb.firebaseio.com/ 

https://vs-sensor-default-rtdb.firebaseio.com/

State: "detected"

PIR SENSOR

Database

**Vehicle safety system**

V.S.S

# APPLICATION TEST

NEXT →

# Login page test

**precondition**

Test case	Test steps	Test data	Expected results	Actual results	Pass/fail	
The account should be in the system.	<ul style="list-style-type: none"> <li>Valid email and password</li> </ul>	<b>1-Enter email Address</b>	<ul style="list-style-type: none"> <li>email: sherry.shehoo@gmail.com Password: test123</li> </ul>	<ul style="list-style-type: none"> <li>Login to system</li> </ul>	<ul style="list-style-type: none"> <li>Login successful</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
			<ul style="list-style-type: none"> <li>email: Sherry.sheho Password: test123</li> </ul>	<ul style="list-style-type: none"> <li>Unable to login until enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Error messages ask the user to enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Valid email and Invalid password</li> </ul>	<b>2-Then the password</b>	<ul style="list-style-type: none"> <li>email: sherry.shehoo@gmail.com Password: test12</li> </ul>	<ul style="list-style-type: none"> <li>Unable to login until enter a correct password</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that he enters incorrect password</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
			<ul style="list-style-type: none"> <li>email: Sherry.sheho Password: test12</li> </ul>	<ul style="list-style-type: none"> <li>Unable to login until enter a correct password and a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Error messages ask the user to enter a valid email address and a correct password</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>

# Register page test

## precondition

	Test case	Test steps	Test data	Expected results	Actual results	Pass/fail
<b>The account should not be in the system</b>	<ul style="list-style-type: none"> <li>Enter unregistered email in the system with strong password</li> </ul>	<b>1-enter your name</b> <b>2-Enter an email</b> <b>3-Enter a password</b>	<ul style="list-style-type: none"> <li>Name: Maha alosaimi</li> <li>email: mahaalosaimi4sms@gmail.com</li> <li>Password:Test123</li> </ul>	<ul style="list-style-type: none"> <li>register to the system</li> </ul>	<ul style="list-style-type: none"> <li>register successful</li> </ul>	<ul style="list-style-type: none"> <li>• Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Enter unregistered email in the system with weak password</li> </ul>		<ul style="list-style-type: none"> <li>Name: Maha alosaimi</li> <li>email: mahaalosaimi4sms@gmail.com</li> <li>Password:te</li> </ul>	<ul style="list-style-type: none"> <li>Unable to register</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that password is to weak</li> </ul>	<ul style="list-style-type: none"> <li>• Pass</li> </ul>
<b>The account should be in the system</b>	<ul style="list-style-type: none"> <li>Enter email previously registered in the system with strong password</li> </ul>	<b>1-enter your name</b> <b>2-Enter an email</b> <b>3-Enter a password</b>	<ul style="list-style-type: none"> <li>Name: Maha alosaimi</li> <li>email: Sherry.shehoo@gmail.com</li> <li>Password: Test123</li> </ul>	<ul style="list-style-type: none"> <li>Unable to register</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that the email already exists</li> </ul>	<ul style="list-style-type: none"> <li>• Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Enter email previously registered in the system with weak password</li> </ul>		<ul style="list-style-type: none"> <li>Name: Maha alosaimi</li> <li>email: Sherry.shehoo@gmail.com</li> <li>Password:te</li> </ul>	<ul style="list-style-type: none"> <li>Unable to register</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that password is to weak</li> </ul>	<ul style="list-style-type: none"> <li>• Pass</li> </ul>

# OTP verification test

## precondition

precondition	Test case	Test steps	Test data	Expected results	Actual results	Pass/fail
The account should not be in the system	<ul style="list-style-type: none"> <li>Enter matching OTP</li> </ul>	<ol style="list-style-type: none"> <li>click “send OTP” button</li> <li>check your email</li> <li>enter the OTP</li> </ol>	<ul style="list-style-type: none"> <li>OTP received via email: 945729</li> <li>OTP entered in the OTP field: 945729</li> </ul>	<ul style="list-style-type: none"> <li>Verify access to the system</li> </ul>	<ul style="list-style-type: none"> <li>verification successful</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Enter mismatching</li> </ul>	<ol style="list-style-type: none"> <li>click “send OTP” button</li> <li>check your email</li> <li>enter the OTP</li> </ol>	<ul style="list-style-type: none"> <li>OTP received via email: 945729</li> <li>OTP entered in the OTP field: 945700</li> </ul>	<ul style="list-style-type: none"> <li>Unable to verify access to the system</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that he enters invalid OTP</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>

# Forget password page test

## precondition

	Test case	Test steps	Test data	Expected results	Actual results	Pass/fail
<b>The account should be in the system</b>	<ul style="list-style-type: none"> <li>Enter email previously registered in the system</li> </ul>	<b>1-enter email address 2-press “RECOVER PASSWORD” button</b>	<ul style="list-style-type: none"> <li>Email: Sherry.shehoo@gmail.com</li> </ul>	<ul style="list-style-type: none"> <li>Send recover password email to the user email</li> </ul>	<ul style="list-style-type: none"> <li>Email sent successful</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Enter invalid email address</li> </ul>		<ul style="list-style-type: none"> <li>Email: Sherry.shehoogmail.com</li> </ul>	<ul style="list-style-type: none"> <li>Unable to send recover email until enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Error messages ask the user to enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
<b>The account should not be in the system</b>	<ul style="list-style-type: none"> <li>Enter unregistered email in the system</li> </ul>	<b>1-enter email address 2-press “RECOVER PASSWORD” button</b>	<ul style="list-style-type: none"> <li>Email: Hlmm@gmail.com</li> </ul>	<ul style="list-style-type: none"> <li>Unable to send recover password email</li> </ul>	<ul style="list-style-type: none"> <li>Error messages notify the user that no user found with that email</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>
	<ul style="list-style-type: none"> <li>Enter invalid email address</li> </ul>		<ul style="list-style-type: none"> <li>Email: Sherry.shehoogmail.com</li> </ul>	<ul style="list-style-type: none"> <li>Unable to send recover email until enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Error messages ask the user to enter a valid email address</li> </ul>	<ul style="list-style-type: none"> <li>Pass</li> </ul>

# Conclusion and future work

**NEXT →**

# CONCLUSIONS:

In our project we developed a system for parents who accidentally leave a baby or young child in a locked vehicles also the bus drivers they can use our system to make sure that no one is left in the bus. Our system will reduce the risk of children death, we tried to make our application easy to use and accessible to everyone.



# FUTURE WORKS

The future work for our upgrade project by put a camera inside the vehicle with the sensor ,also make system to control the vehicle to open window and turn on the air conditioning , if they take a long time to respond to the app notification it will directly send to police the vehicle information and the number also the location of that vehicle to save the baby or the person who locked inside the vehicle..



**THANK YOU FOR  
LISTENING**

Hope you like  
Our graduation project

**END**