

2023 (주)코덱스브리지 개발팀(백엔드 개발자) 신입 7

# PORTFOLIO

CONTACT



# INTRODUCE

신입 개발자 [redacted]입니다.

마르지 않는 백두산 샘물처럼  
항상 끊임없이 공부하는 개발자가 되겠습니다.

## Contact



### Education

2015.03~2020.02 [redacted] 졸업



### Project

2022.07~2023.01 미래융합교육원 팀프로젝트 참여



### Award

2023.01 미래융합교육원 팀프로젝트 최우수상

# Skills

## Language



## Front-End



## Back-End



## Database



## ETC



# 원가드 프로젝트

**프로젝트 기간** 2022. 11. 17 ~ 2023. 01. 10 (7주)

**플랫폼** 관리자 Web, Android, iOS

**참여 인원** 5명 (팀명: 삼삼오오)

**참여 기업** 에스지텍

**개발 환경**

**서버** Flask, MQTT, GCP, Django

**데이터 베이스** MariaDB

**객체 인식** Anaconda, YOLO V4, Pycharm

**앱** Flutter – Android Studio,  
iOS(Xcode), VSCode

## 프로젝트 소개

참여 기업 (주)에스지텍의 자회사인  
원가드의 핵심 기술인 창호 센서 기능과  
미래융합교육원 수업시간에 배운  
딥러닝과 MQTT 를 활용하여  
관리자용 Web 페이지 구현,  
사용자용 App 인터페이스 구현해보고자  
삼삼오오(팀명)는 이번 프로젝트에  
참가하게 되었습니다.

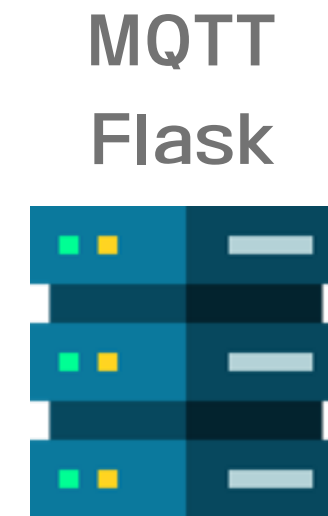
## 담당 역할

설계 – DB 설계 및 App 설계  
CCTV 객체 인식 – 기능 구현  
앱 제작 – 주로 앱을 담당하여 첫 앱을 만들기 시작함  
로그인 구현 – DB 연동,  
로그인 세션 유지 – Shared Preference,  
회원가입 – 양식에 맞춰야만 회원가입,  
비밀번호 – 암호화하여 db에 저장 및 비밀번호 찾기 구현  
히스토리 – 센서 내역과 동영상 내역 히스토리 구현  
기기관리 – 기기등록 및 삭제 기능  
iOS – 유일한 맥북 보유자로 ios 앱 구현 – android와  
동일하게 100% 구현 성공  
그 외 – 기획 발표 및 ppt 제작

# 구동방식



wireless 통신



서버



딥러닝을 통한  
객체 인식



데이터 전송



센서, 카메라 통해  
전달된 이벤트 기록



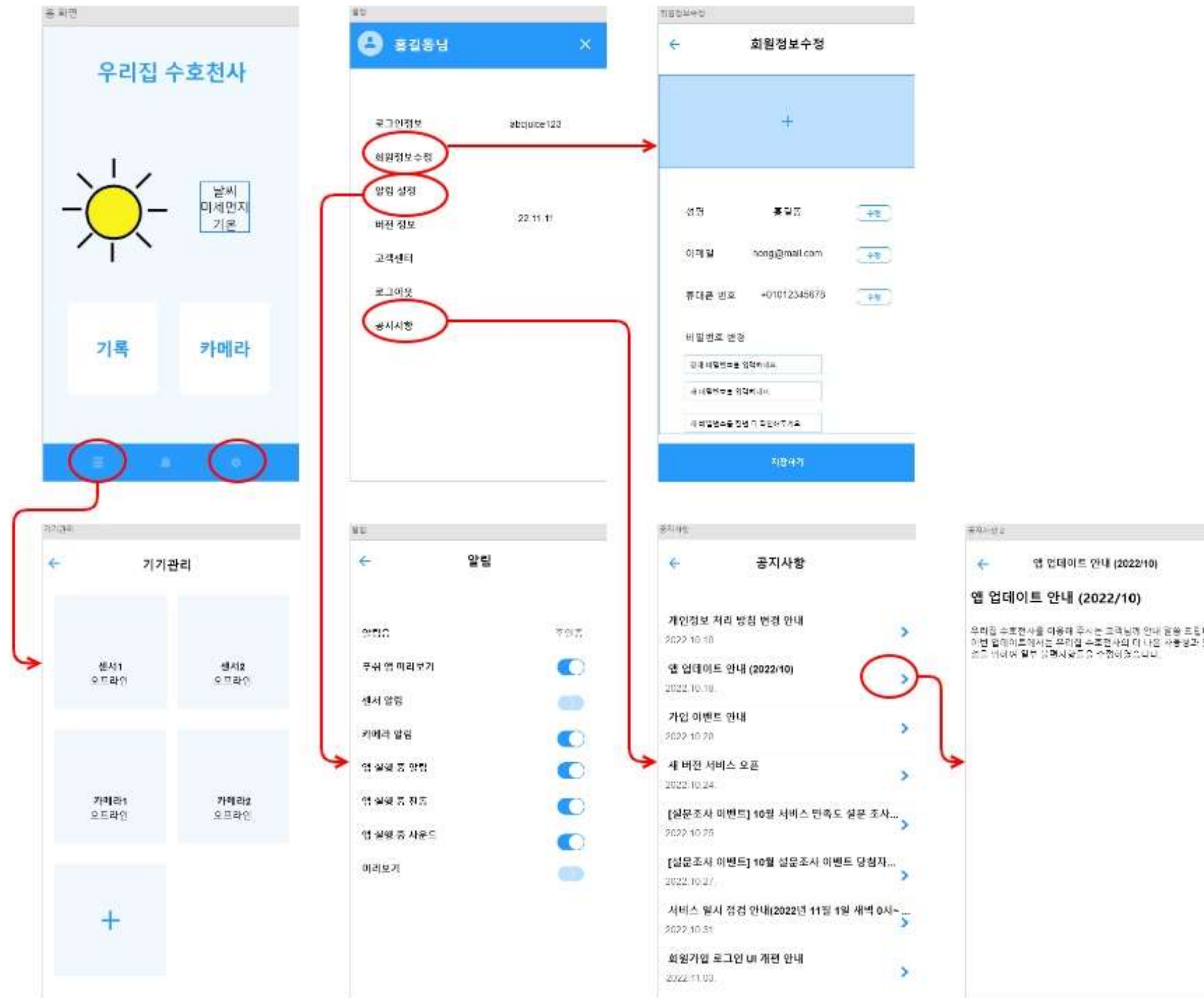
# 개발일정

구분		개발 내용	1주	2주	3주	4주	5주	6주	7주
설계		시스템 설계	●	●					
		DB 설계	●	●					
		앱 설계	●	●					
구현	CCTV 객체인식	Yolo v4 기능 구현			●				
		CCTV 영상 사람 인식 기능 추가			●				
		MQTT 연동				●			
	서버 및 환경 설정	MQTT 세팅/ DB 제작				●			
		Flask, Django 서버 구축					●		
		Django 관리자 페이지 제작						●	
	앱 제작	로그인/회원가입/메인페이지제작				●			
		히스토리 페이지 제작					●		
		MQTT 연동					●		
		푸시 알림 기능 추가					●		
		마이페이지 제작						●	
		기기 관리 페이지 제작							●
		IOS 앱 제작						●	●
기타		깃 세팅					●		
		통합 테스트						●	●



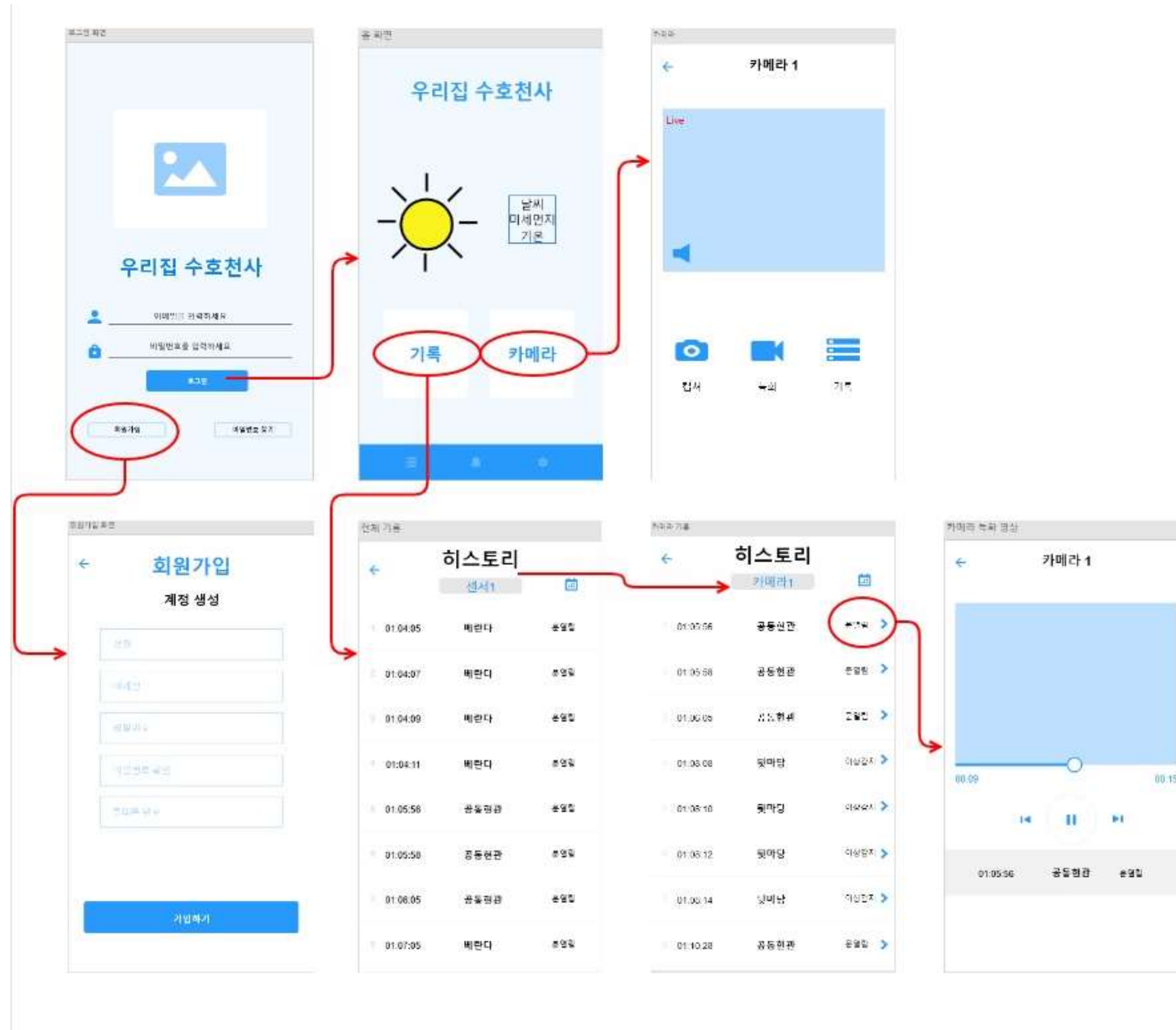
# 설계 – App 설계(1/2)

메인화면, 회원정보 수정 및 기기관리, 알림 설정, 공지사항 화면



# 설계 – App 설계(2/2)

## 회원가입 라이브 영상 및 히스토리 화면





# 사용자 페이지 주요 기능(1/4)

로그인, 회원가입 및 비밀번호 찾기 화면



아이디

비밀번호

로그인

회원가입

비밀번호 찾기

앱 실행 첫 화면



회원가입

아이디

이메일

비밀번호

비밀번호 확인

이름

휴대폰 번호

주소

회원가입

회원가입 화면



비밀번호 찾기

아이디

이메일

비밀번호 찾기

비밀번호 찾기



비밀번호 변경

아이디

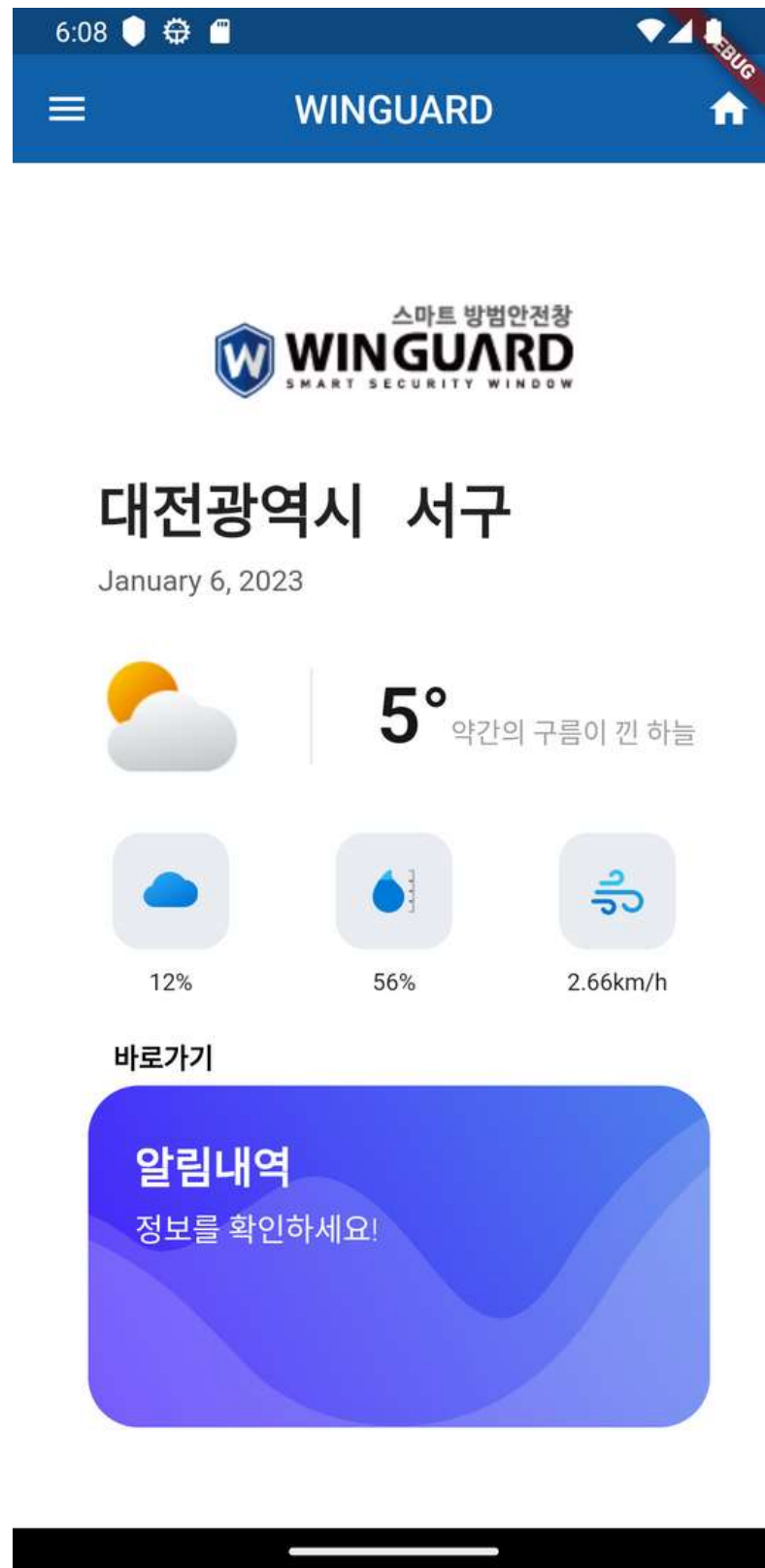
비밀번호

비밀번호 확인

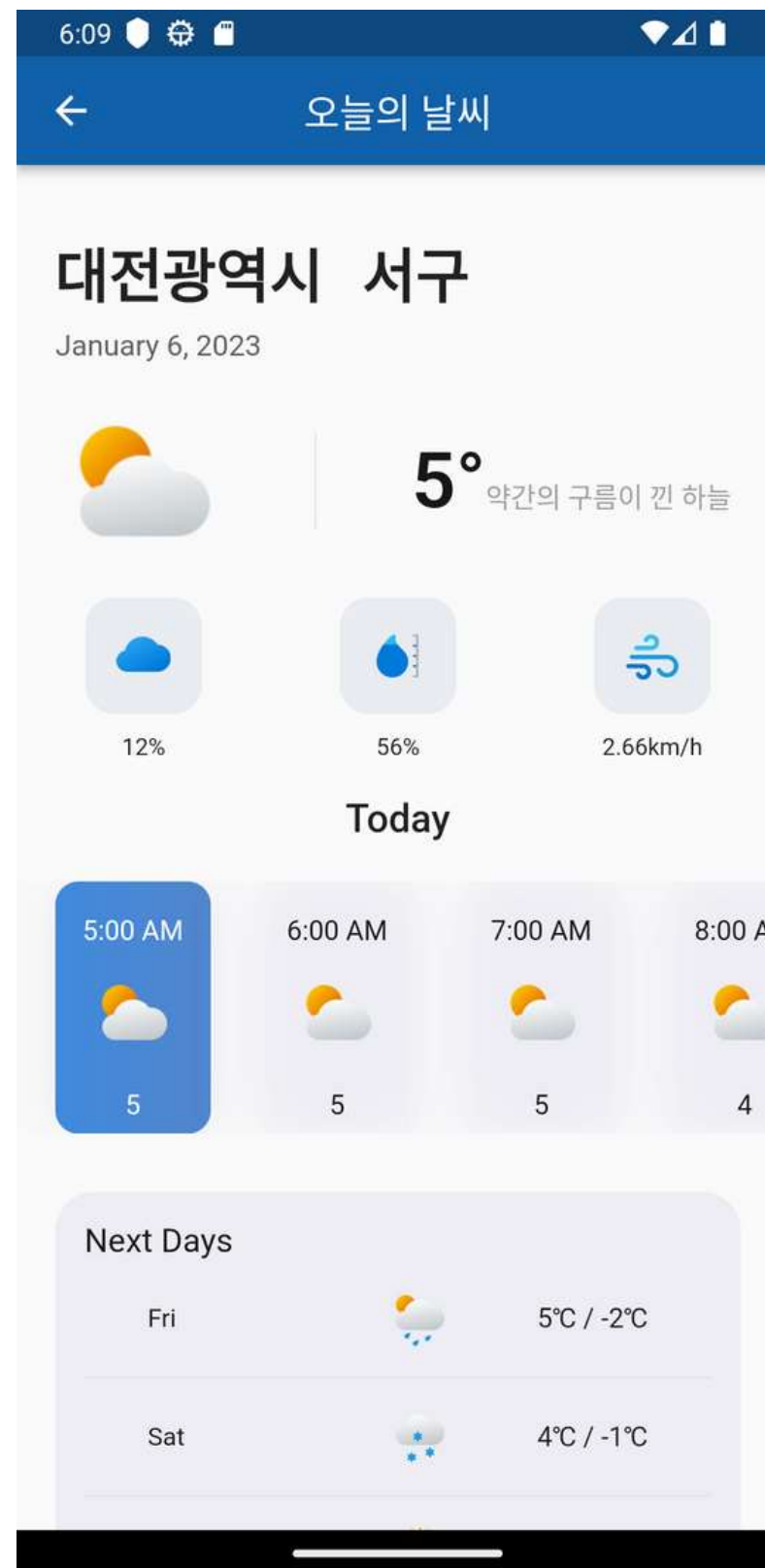
비밀번호 변경

비밀번호 재설정

# 사용자 페이지 주요 기능(2/4)



메인 화면



날씨 세부 정보

## 메인 화면 센서 히스토리 화면

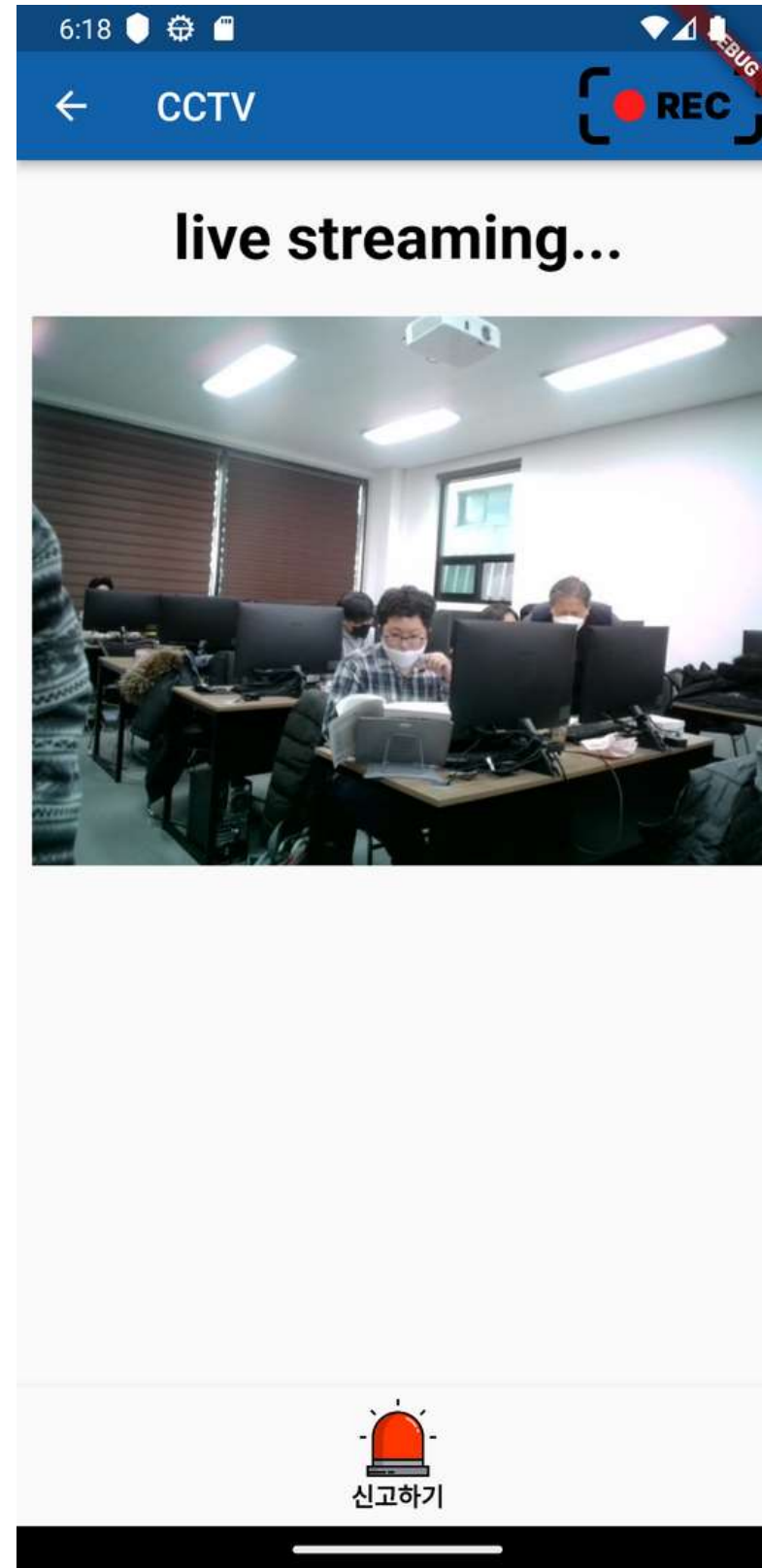


알림내역

# 사용자 페이지 주요 기능(3/4)

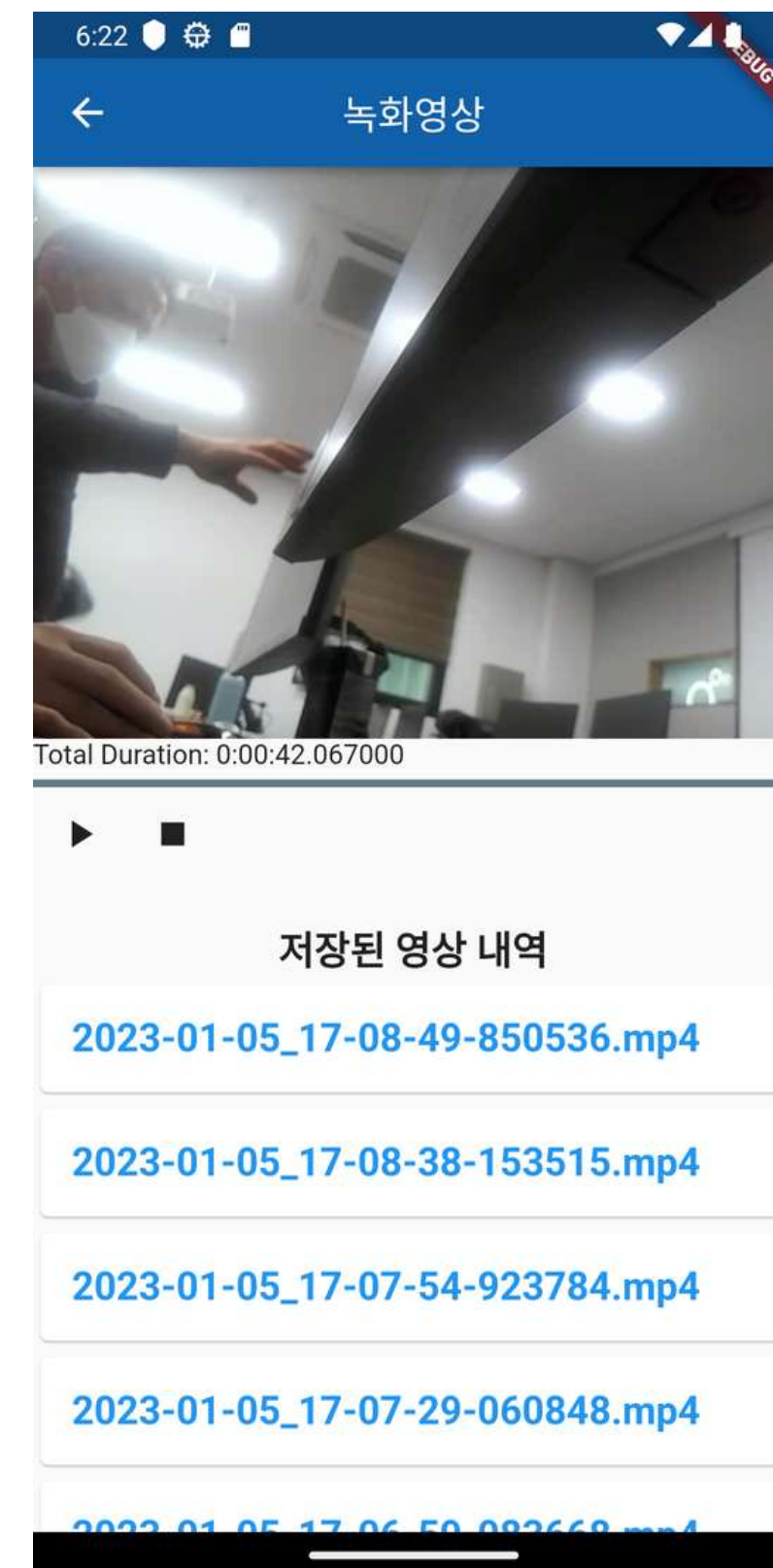


메인 화면



실시간 영상 확인

## 라이브 영상 및 영상 히스토리 화면



녹화 영상 확인

# 사용자 페이지 주요 기능(4/4)

## 기기등록 화면



대전광역시 서구

January 6, 2023



6° 맑음



4%



50%



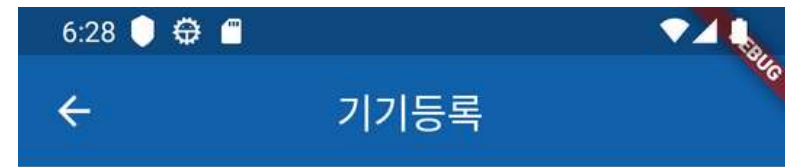
2.57km/h

바로가기

기기등록

자신의 기기를 등록하세요!

메인 화면



기기번호

기기이름

기기등록

기기 등록

idx	user_id	serial_number	device_name
3	jdk	ds20221221-0001	도어센서
2	test9999	ds20221221-0002	도어센서
4	test9999	cam20021221-0001	카메라 센서

기기를 등록하면 사용자의  
아이디와 함께 DB에 저장



# 관리자 페이지 주요 기능(1/2)

## 원가드 관리자 페이지

### 원가드 관리

MYAPP		
Devices	+ 추가	✎ 변경
Historys	+ 추가	✎ 변경
User ips	+ 추가	✎ 변경
Users	+ 추가	✎ 변경
Videos	+ 추가	✎ 변경
NOTICE		
공지사항	+ 추가	✎ 변경
인증 및 권한		
그룹	+ 추가	✎ 변경
사용자(들)	+ 추가	✎ 변경

DB에 기록된 정보를 관리할 수 있는  
웹 페이지 구축

사용자에게 보여질 공지사항 등록 가능

관리자 권한 설정  
관리자들에게 게시판별 접근 권한을 설정하여  
체계적인 관리와 고객정보에 대한 보안유지 가능

# 관리자 페이지 주요 기능(2/2)

변경할 user 선택

USER 추가 +

Q  검색

액션:  실행 7 중 아무것도 선택되지 않았습니다.

<input type="checkbox"/>	USER ID	PASSWORD	NAME	PHONE NUMBER	ADDRESS	EMAIL
<input type="checkbox"/>	test9999	888df25ae35772424a560c7152a1de794440e0ea5cfee62828333a456a506e05	test	01012345678	qwer123	qwer321
<input type="checkbox"/>	test999	83cf8b609de60036a8277bd0e96135751bbc07eb234256d4b65b893360651bf2	test2	01023456789	qwer123	qwer321
<input type="checkbox"/>	qqq	a95bc16631ae2b6fadb455ee018da0adc2703e56d89e3eed074ce56d2f7b1b6a				
<input type="checkbox"/>	jdk	2ca883ba5c161911b5dcd5591f5240e67ac1fbed2db2df9da0374838668702a2	jeong	0101231243	daejeon	jeong@naver.com
<input type="checkbox"/>	asd	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	kim	010123456789	daejeon	abc@gmail.com
<input type="checkbox"/>	a	ca978112ca1bbdcfac231b39a23dc4da786eff8147c4e72b9807785afee48bb	a	a	a	a
<input type="checkbox"/>		e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855				

액션:  실행 86 중 아무것도 선택되지 않았습니다.

<input type="checkbox"/>	USER ID	SENSOR	STATUS	DATE TIME
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:29 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:28 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:28 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:28 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:28 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:13 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:11 오후
<input type="checkbox"/>	test9999	door sensor	door opened	2022년 12월 21일 11:11 오후



# 데이터 베이스

The screenshot shows the HeidiSQL interface. On the left, a tree view shows the database structure under 'winguard\_project', including schemas like 'information\_schema', 'mysql', 'performance\_schema', and tables like 'auth\_group', 'auth\_group\_permissi...', 'auth\_permission', 'auth\_user', 'auth\_user\_groups', 'auth\_user\_user\_permi...', 'Device', 'django\_admin\_log', 'django\_content\_type', 'django\_migrations', 'django\_session', 'History', 'User', and 'Video'. The 'History' table is selected. The main pane shows the 'History' table with 87 rows. The columns are 'idx', 'user\_id', 'sensor', 'status', and 'datetime'. The data shows multiple 'door opened' events for 'test9999' and 'test999'.

idx	user_id	sensor	status	datetime
1	test9999	door sensor	door opened	2022-12-20 15:21:54
2	test9999	door sensor	door opened	2022-12-20 15:22:07
3	test9999	door sensor	door opened	2022-12-20 15:22:09
4	test9999	door sensor	door opened	2022-12-20 15:22:10
5	test9999	door sensor	door opened	2022-12-20 15:22:11
6	test9999	door sensor	door opened	2022-12-20 15:34:05
7	test9999	door sensor	door opened	2022-12-20 15:35:09
20	test9999	door sensor	door opened	2022-12-21 10:01:29
21	test9999	door sensor	door opened	2022-12-21 10:01:31
22	test9999	door sensor	door opened	2022-12-21 10:01:51
23	test9999	door sensor	door opened	2022-12-21 10:02:34
24	test9999	door sensor	door opened	2022-12-21 10:02:35
25	test9999	door sensor	door opened	2022-12-21 10:02:35
26	test9999	door sensor	door opened	2022-12-21 10:02:35
27	test9999	door sensor	door opened	2022-12-21 10:02:35
28	test9999	door sensor	door opened	2022-12-21 10:02:36
29	test9999	door sensor	door opened	2022-12-21 10:02:36
30	test9999	door sensor	door opened	2022-12-21 10:02:36
31	test9999	door sensor	door opened	2022-12-21 10:02:36
32	test9999	door sensor	door opened	2022-12-21 10:17:18
33	test9999	door sensor	door opened	2022-12-21 10:19:40
34	test9999	door sensor	door opened	2022-12-21 10:19:42
35	test9999	door sensor	door opened	2022-12-21 10:19:42
36	test9999	door sensor	door opened	2022-12-21 10:19:42
37	test9999	door sensor	door opened	2022-12-21 10:20:58
38	test9999	door sensor	door opened	2022-12-21 10:21:13
39	test9999	door sensor	door opened	2022-12-21 10:21:25
40	test9999	door sensor	door opened	2022-12-21 10:24:56

사용자 정보, 기기정보,  
알림내역 등을 DB에 저장

사용자의 비밀번호는  
암호화해서 저장

user_id	password	name	phone_number	address	email
asd	a665a43920422f9d417e4867efdc4fb8a04a1f31f1fa07e...	kim	010123456789	daejeon	abc@gmail.com
jdk	2ca883ba5c161911b5dcd5591f5240e67ac1fbed2db2df9...	jeong	0101231243	daejeon	jeong@naver.com
test999	83cf8b609de60036a8277bd0e96135751bbc07eb234256...	test2	01023456789	qwer123	qwer321
test9999	888df25ae35772424a560c7152a1de794440e0ea5cfee6...	test	01012345678	qwer123	qwer321

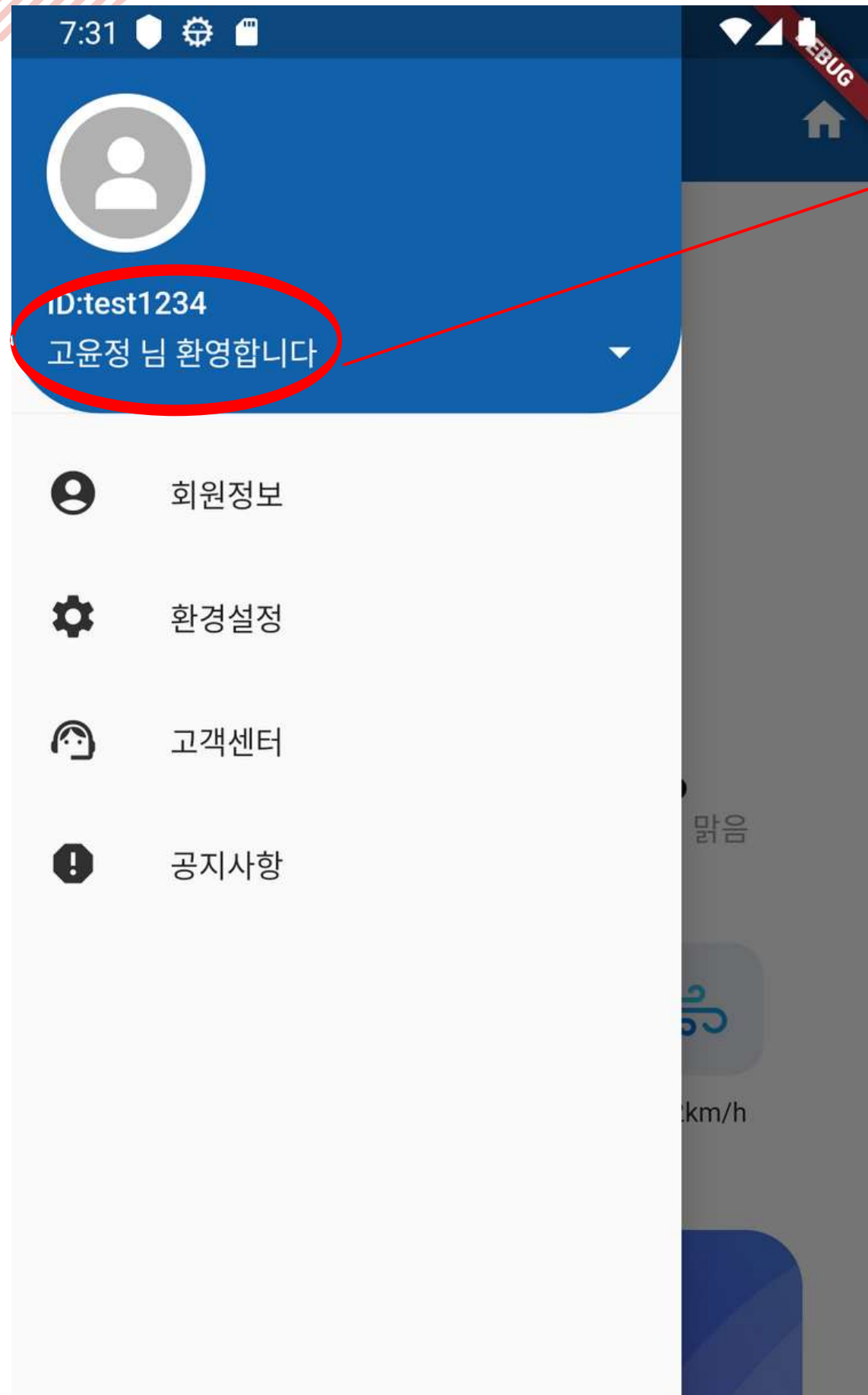
# 푸시 알림



센서가 감지되면 MQTT로 신호 전송

MQTT로 받은 신호를  
사용자의 휴대폰으로 푸시로 알림

# 로그인 세션 유지 – Shared Preference



Flutter의 Shared Preference를 이용해서  
로그인 시 사용자의 아이디를 캐시에 저장

DB에서 해당 아이디에 대한 사용자의 이름을  
불러옴

알림 내역 페이지로 이동 시 DB에서 해당  
아이디로 기록된 내역만 조회

로그인 상태를 캐시에 저장하여 로그아웃 하지 않고  
앱을 종료하면 다시 실행할때 로그인 상태가 유지



# 플러터 코드 화면 예시- 로그인 구현

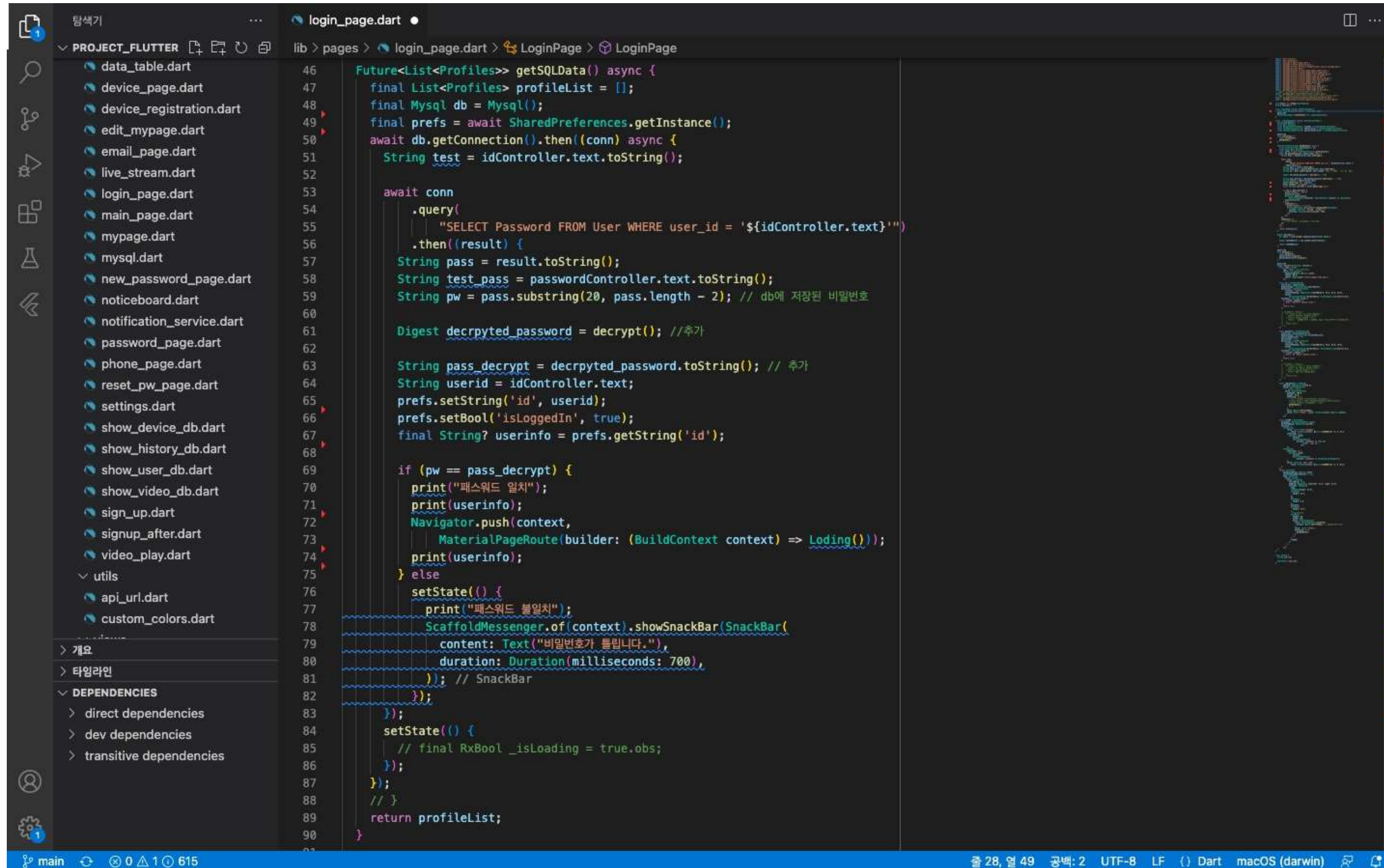
github

android :

[https://github.com/thunderbl  
oom/project\\_flutter.git](https://github.com/thunderbl<br/>oom/project_flutter.git)

ios :

[https://github.com/thunderbl  
oom/project\\_flutter\\_ios.git](https://github.com/thunderbl<br/>oom/project_flutter_ios.git)

A screenshot of an IDE (likely VS Code) showing a Flutter project. The left sidebar displays a file explorer with a list of Dart files under 'PROJECT\_FLUTTER'. The main editor area shows the 'login\_page.dart' file, which contains Dart code for a login function. The code includes database queries, password decryption, and state management. The bottom status bar shows 'main', '0', '1', '615', '줄 28, 열 49', '공백: 2', 'UTF-8', 'LF', 'Dart', and 'macOS (darwin)'.

```
Future<List<Profiles>> getSQLData() async {  
  final List<Profiles> profileList = [];  
  final Mysql db = Mysql();  
  final prefs = await SharedPreferences.getInstance();  
  await db.getConnection().then((conn) async {  
    String test = idController.text.toString();  
  
    await conn  
      .query(  
        "SELECT Password FROM User WHERE user_id = '${idController.text}'"  
      ).then((result) {  
        String pass = result.toString();  
        String test_pass = passwordController.text.toString();  
        String pw = pass.substring(20, pass.length - 2); // db에 저장된 비밀번호  
  
        Digest decrpyted_password = decrypt(); //추가  
  
        String pass_decrypt = decrpyted_password.toString(); // 추가  
        String userid = idController.text;  
        prefs.setString('id', userid);  
        prefs.setBool('isLoggedIn', true);  
        final String? userinfo = prefs.getString('id');  
  
        if (pw == pass_decrypt) {  
          print("패스워드 일치");  
          print(userinfo);  
          Navigator.push(context,  
            MaterialPageRoute(builder: (BuildContext context) => Loding()));  
          print(userinfo);  
        } else  
          setState(() {  
            print("패스워드 불일치");  
            ScaffoldMessenger.of(context).showSnackBar(SnackBar(  
              content: Text("비밀번호가 틀립니다."),  
              duration: Duration(milliseconds: 700),  
            )); // SnackBar  
          });  
        });  
        setState(() {  
          // final RxBool _isLoading = true.obs;  
        });  
      });  
      // }  
      return profileList;  
    }  
  }  
}
```

2023 (주)코덱스브리지 개발팀(백엔드 개발자) 신입 지원자

**THANK YOU**