



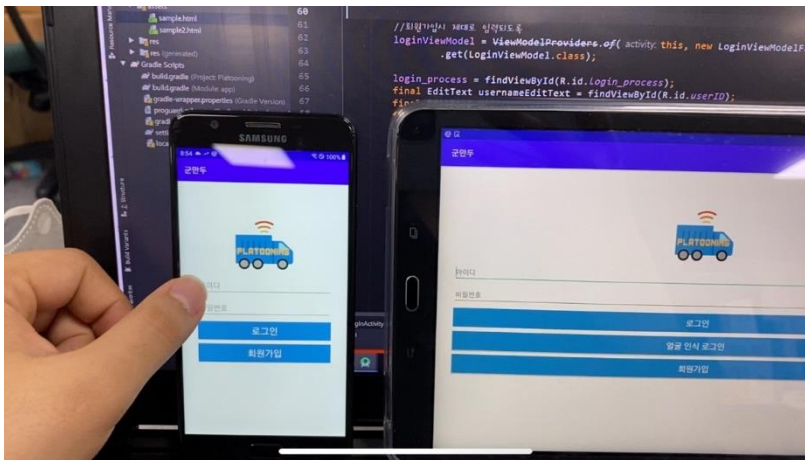
앱 서비스: 테블릿 로그인

차량 테블릿

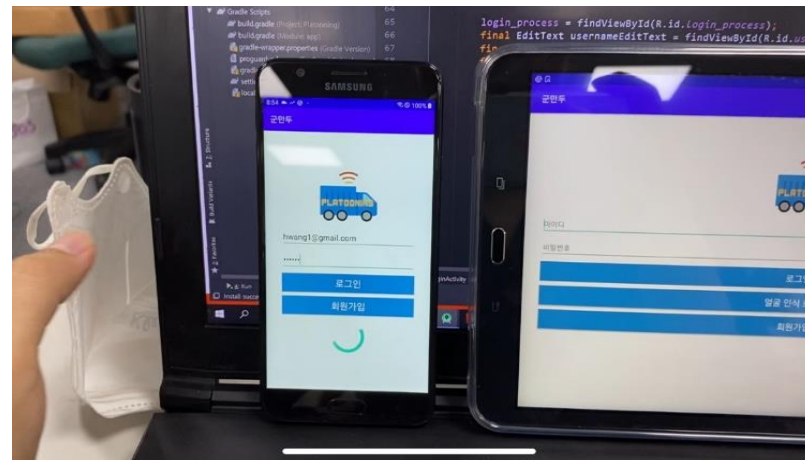


# 앱 서비스: 군집 주행 차량 제어

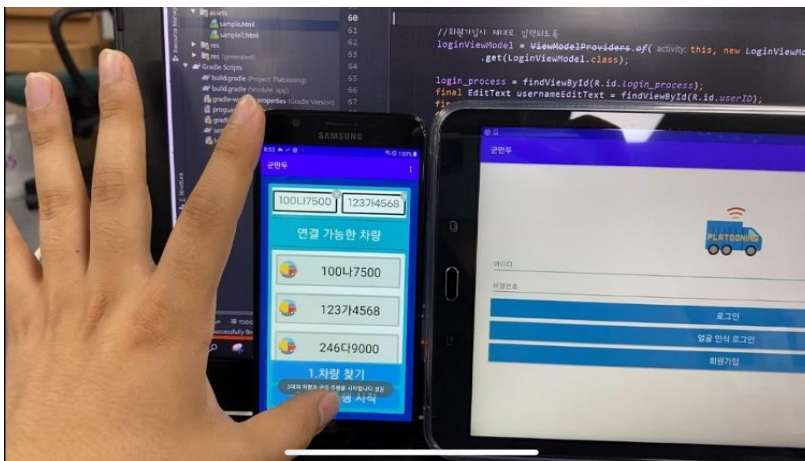
차량 내부 테블릿을 이용하기 위해서는 스마트폰 어플로 군집제어할 차량을 **우선적으로 선택해야** 한다  
차량을 이용하기 위해서는 테블릿에 **꼭 로그인을 해야 한다.**



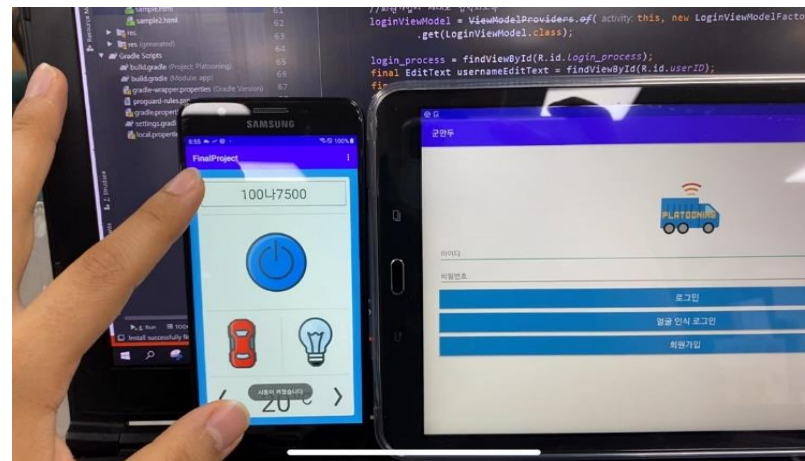
1. 스마트폰 로그인



2. 샘플 아이디: hwang1@gmail.com



3. 군집 제어 차량 선택



4. 차량 시동 On



## 안 서비스 · 구진 주한 차량 제

```
Run SQL Command Line
Commit complete.

SQL> select * from member;

      IDX USERID      PHONE      BIRTH      GENDER      NAME      PASSWORD
-----
71 hwang1@gmail.com 01012341234 940213 0 hwang 123456
cAESNHjMQ1eW83bz6D45VH: APA91bFkXy_LUzIC30KRbI5JdX38soz7BXcHx4WnVP5yFya05yroCuu7SveaVeew3WeViEWBgE8nXE3p8yxpoFckFOJDZ9LoyVqctecSNczV7oBJ5X5zfQVzYQuWqHTVfIuE_KorOr3
hwang

SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> select * from member;

      IDX USERID      PHONE      BIRTH      GENDER      NAME      PASSWORD
-----
77 test1@gmail.com 01012341234 940213 0 황제윤 123456
cAESNHjMQ1eW83bz6D45VH: APA91bFkXy_LUzIC30KRbI5JdX38soz7BXcHx4WnVP5yFya05yroCuu7SveaVeew3WeViEWBgE8nXE3p8yxpoFckFOJDZ9LoyVqctecSNczV7oBJ5X5zfQVzYQuWqHTVfIuE_KorOr3

SQL> select * from member;

      IDX USERID      PHONE      BIRTH      GENDER      NAME      PASSWORD
-----
71 hwang1@gmail.com 01012341234 940213 0 hwang 123456
cAESNHjMQ1eW83bz6D45VH: APA91bFkXy_LUzIC30KRbI5JdX38soz7BXcHx4WnVP5yFya05yroCuu7SveaVeew3WeViEWBgE8nXE3p8yxpoFckFOJDZ9LoyVqctecSNczV7oBJ5X5zfQVzYQuWqHTVfIuE_KorOr3
hwang

SQL> select userid, face from member;

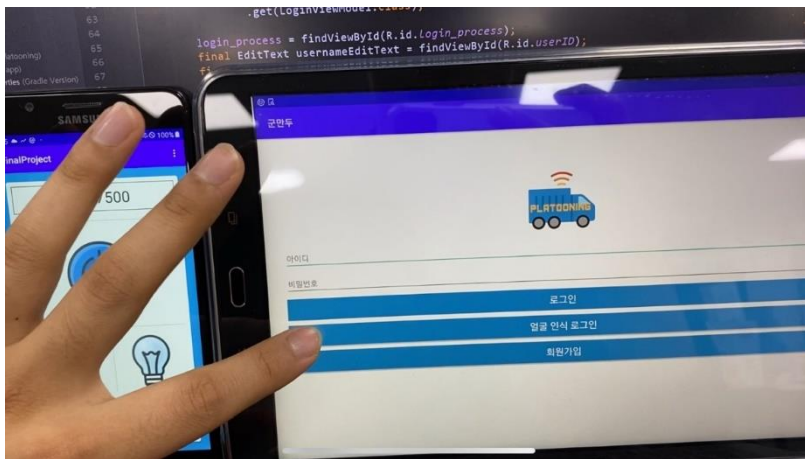
      USERID      FACE
-----
test1@gmail.com
hwang1@gmail.com hwang

SQL>
```

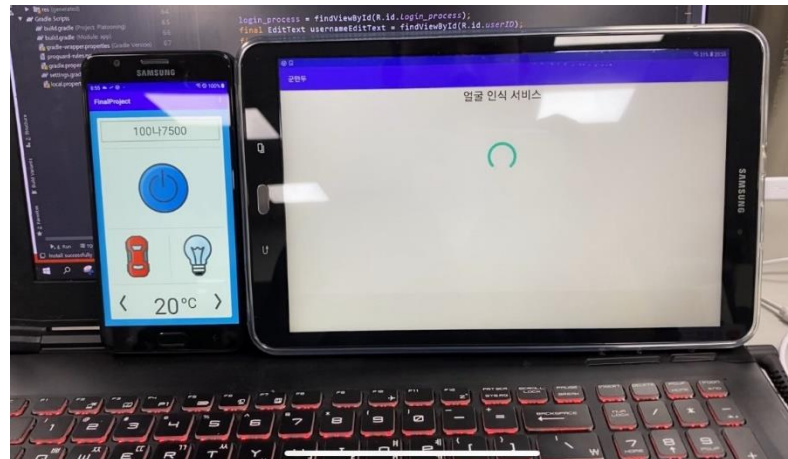
회원 Member DB를 확인해보면  
hwang1@gmail.com 에 얼굴인  
식 서비스가 등록되어 있고 '황  
제윤'의 얼굴 정보(hwang)가  
입력되어 있다.



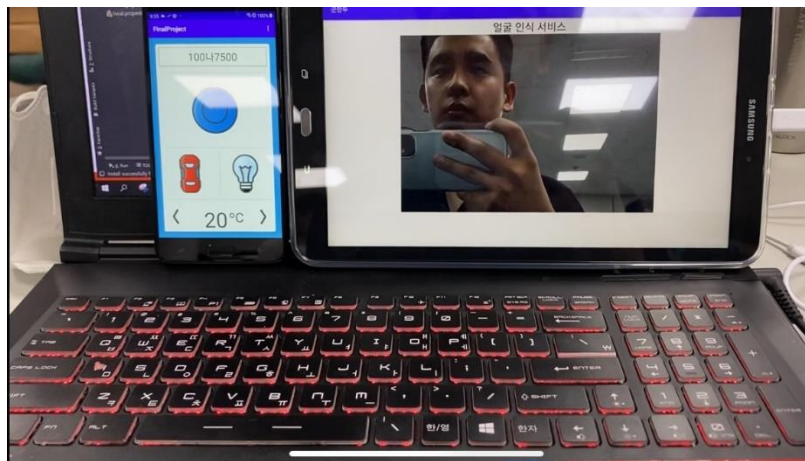
# 앱 서비스: 군집 주행 차량 제어



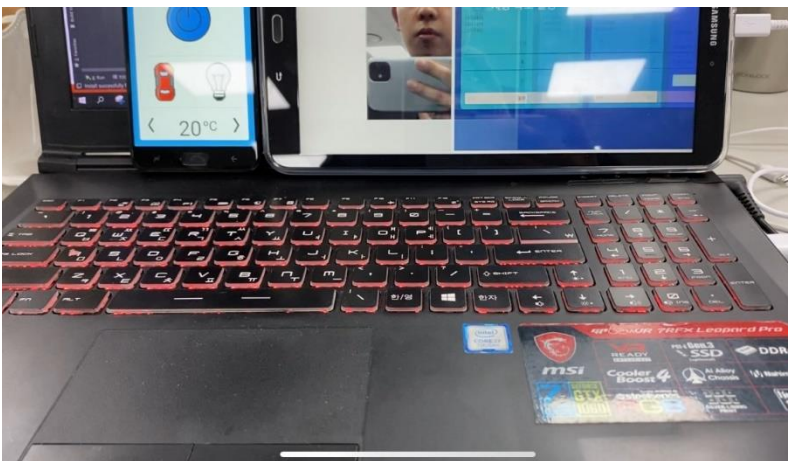
1. 차량 테블릿 로그인 - 얼굴인식 클릭



2. 얼굴 인식 서비스 로딩



3. 카메라가 뜨고 얼굴을 인식하는 중이다.



4. 얼굴인식 후 hwang1@gmail.com 계정으로 로그인 성공

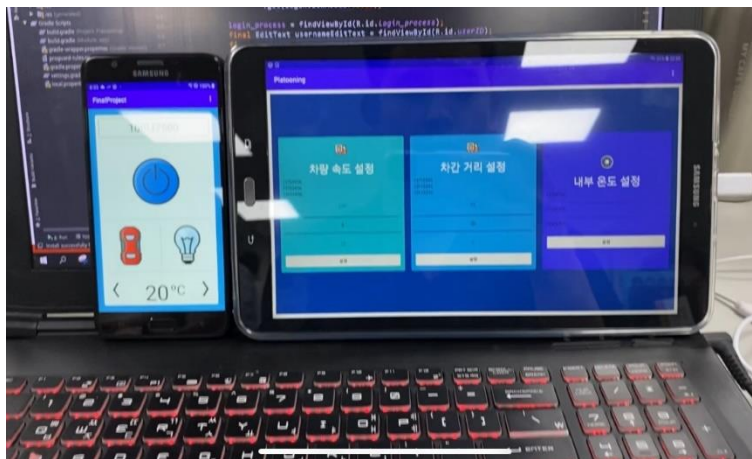


앱 서비스: 군집 주행 차량 제  
어 차량 테블릿

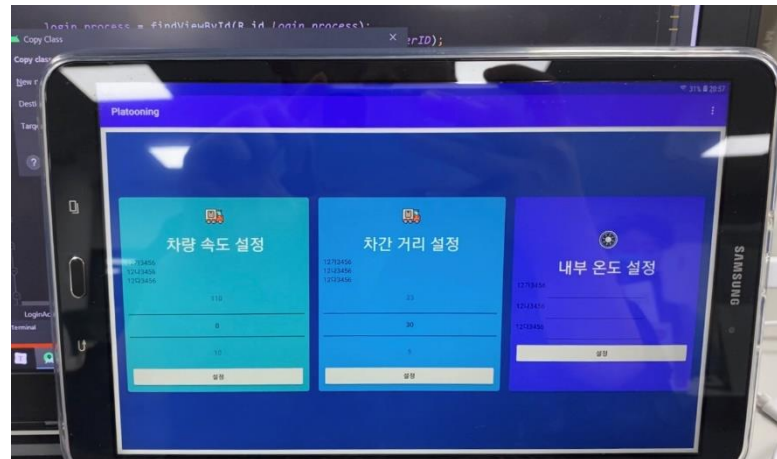




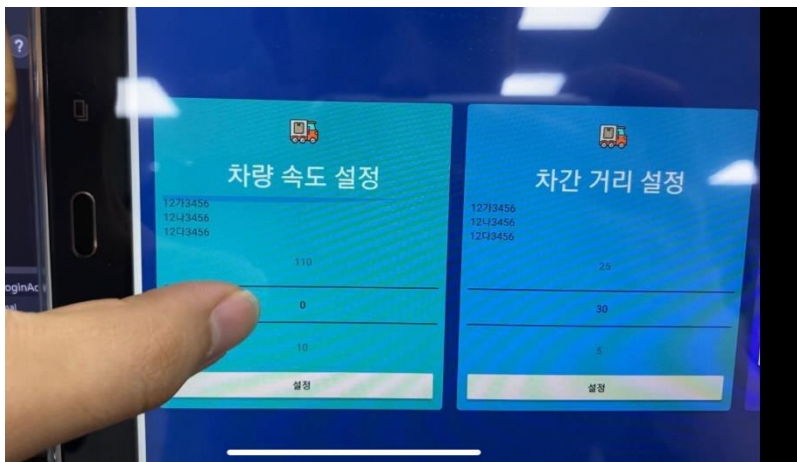
# 앱 서비스: 군집 주행 차량 제어



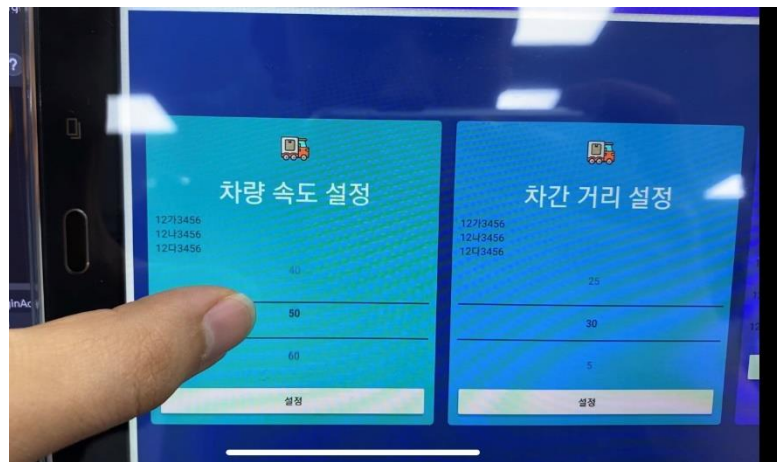
1. 차량 테블릿 메인 화면



2. 차량제어: 속도, 차간 거리, 온도



3. 차량 제어: 속도 설정



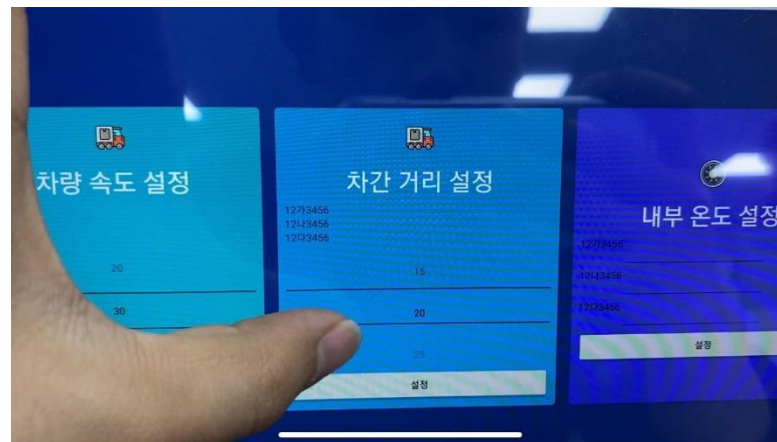
4. 속도 설정값을 변경할 수 있다



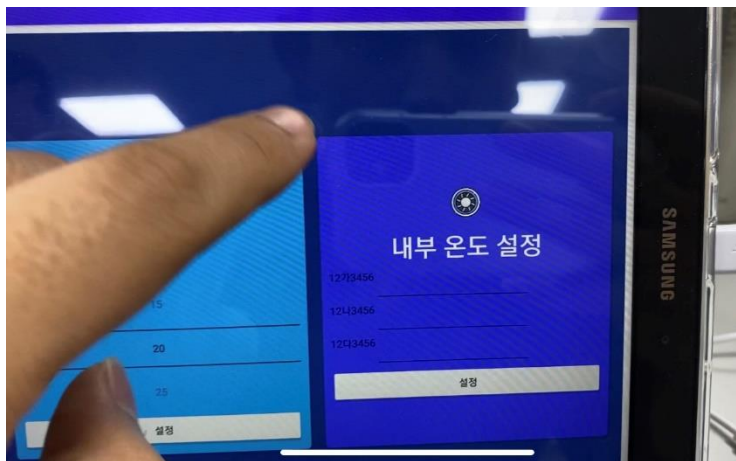
## 앱 서비스: 군집 주행 차량 제어



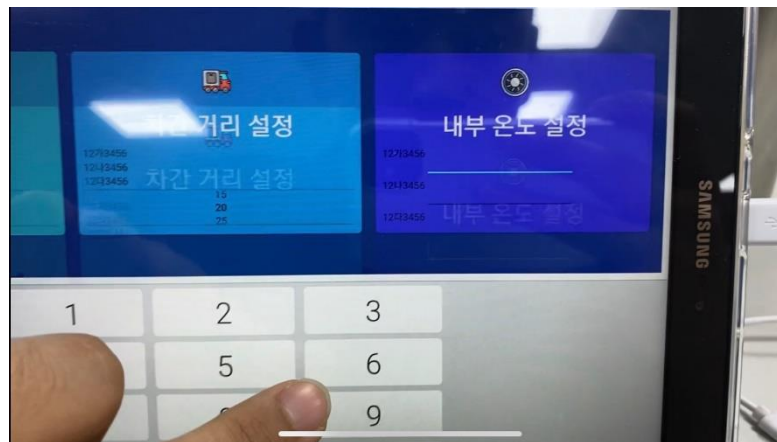
1. 차량 제어: 차간 거리 설정



2. 차간 거리 설정값을 변경할 수 있다



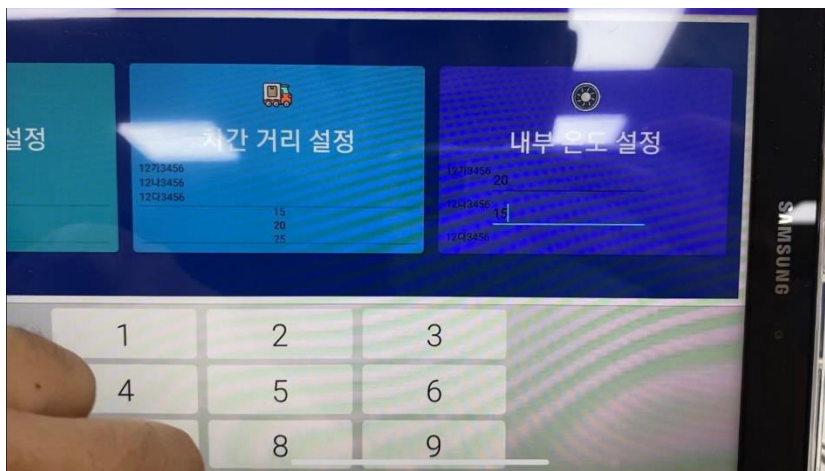
3. 차량 제어: 차량 내부 온도 설정



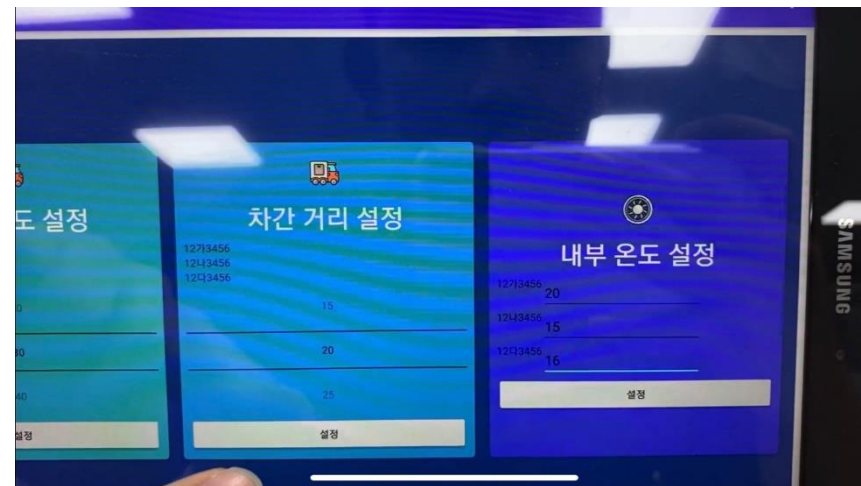
4. 속도 설정값을 변경할 수 있다



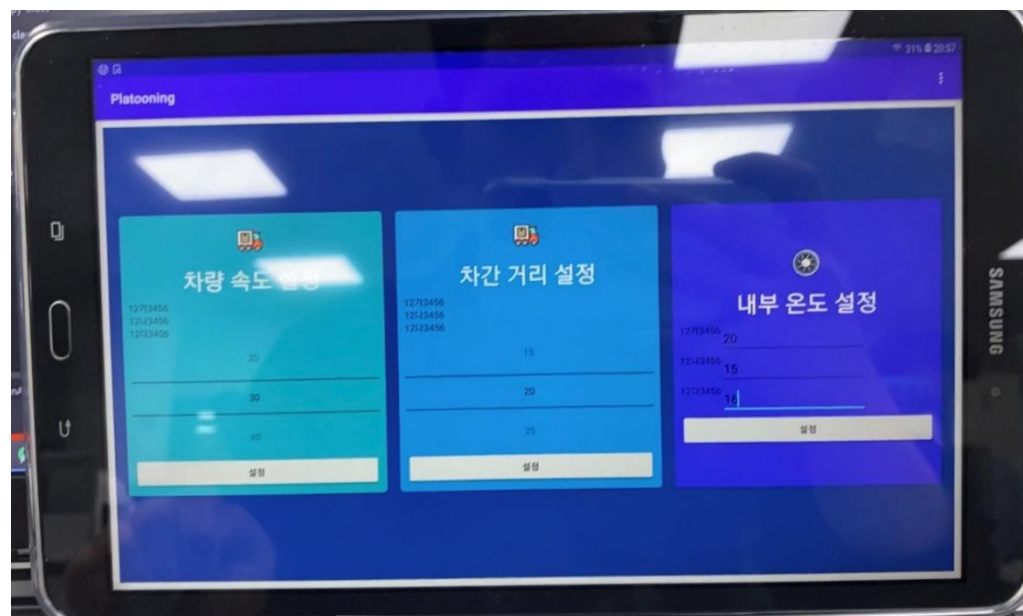
## 앱 서비스: 군집 주행 차량 제어



1. 3개 차량 온도 설정 중



2. 3개 차량 온도 설정 완료

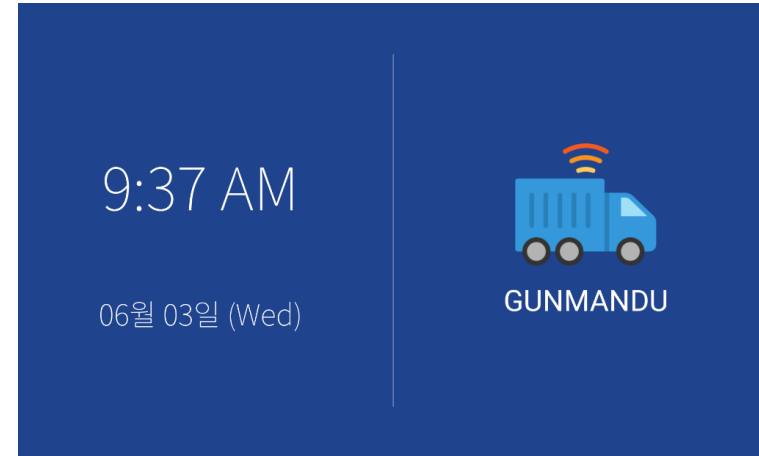
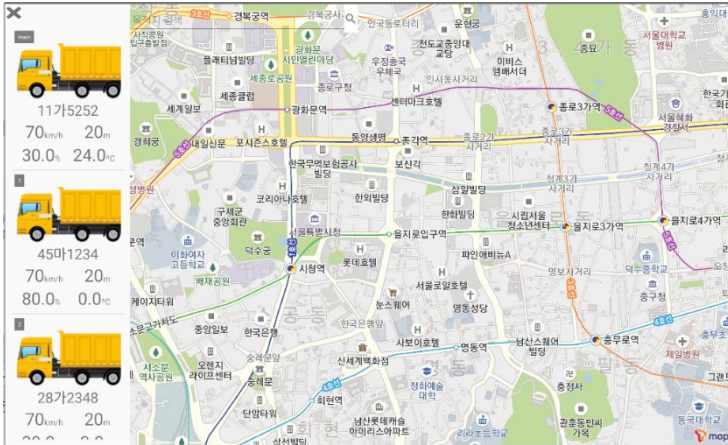


차량 제어 완료



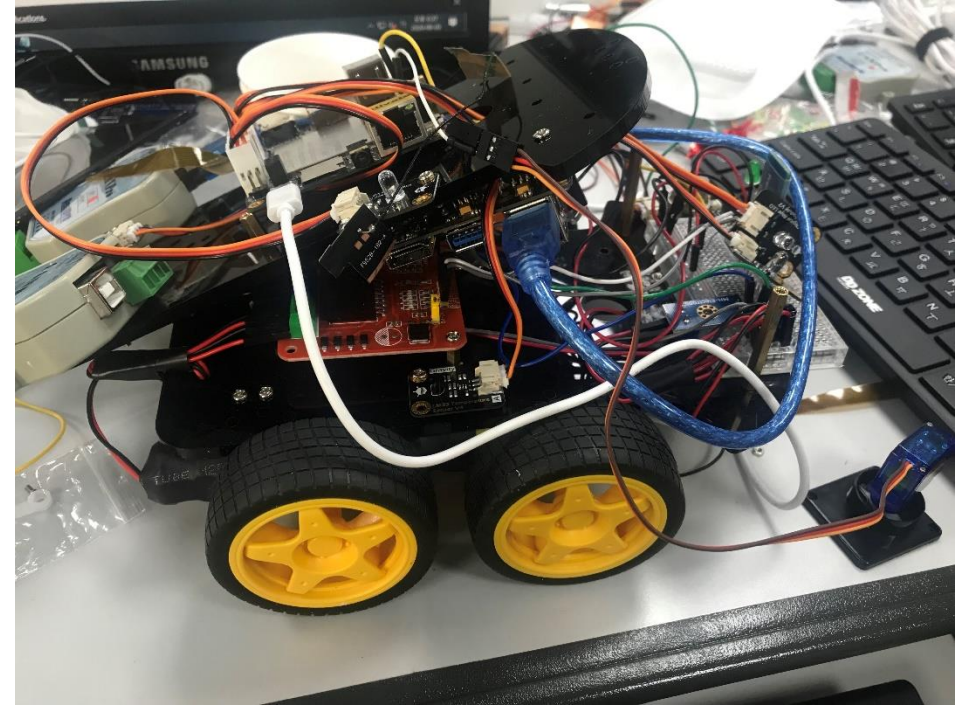
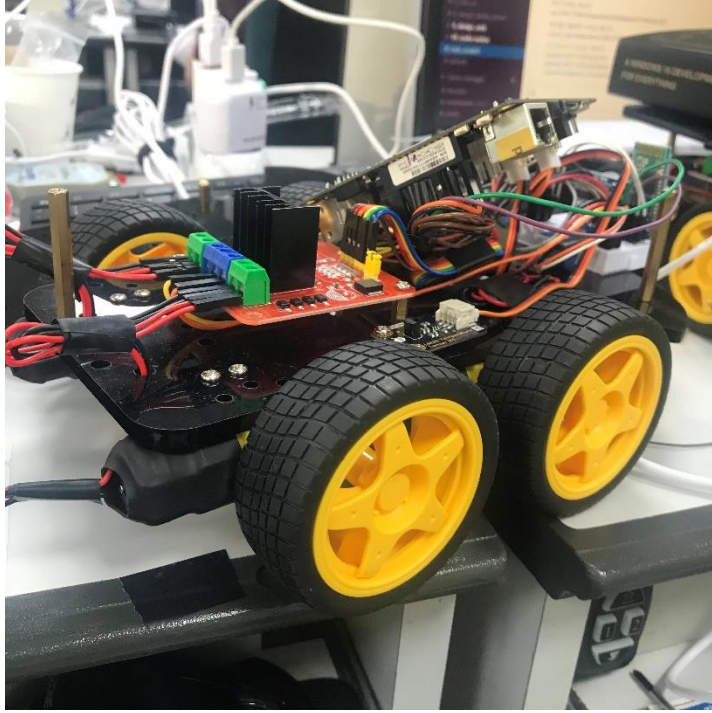


# 앱 서비스: 군집 주행 차량 제어





차량 서비스: 군집 주행 차량  
차량 (RC카)






# 웹 서비스: 차량관제와 빅데이터 분석






# 관리자 로그인

AdminLogin





Sign In

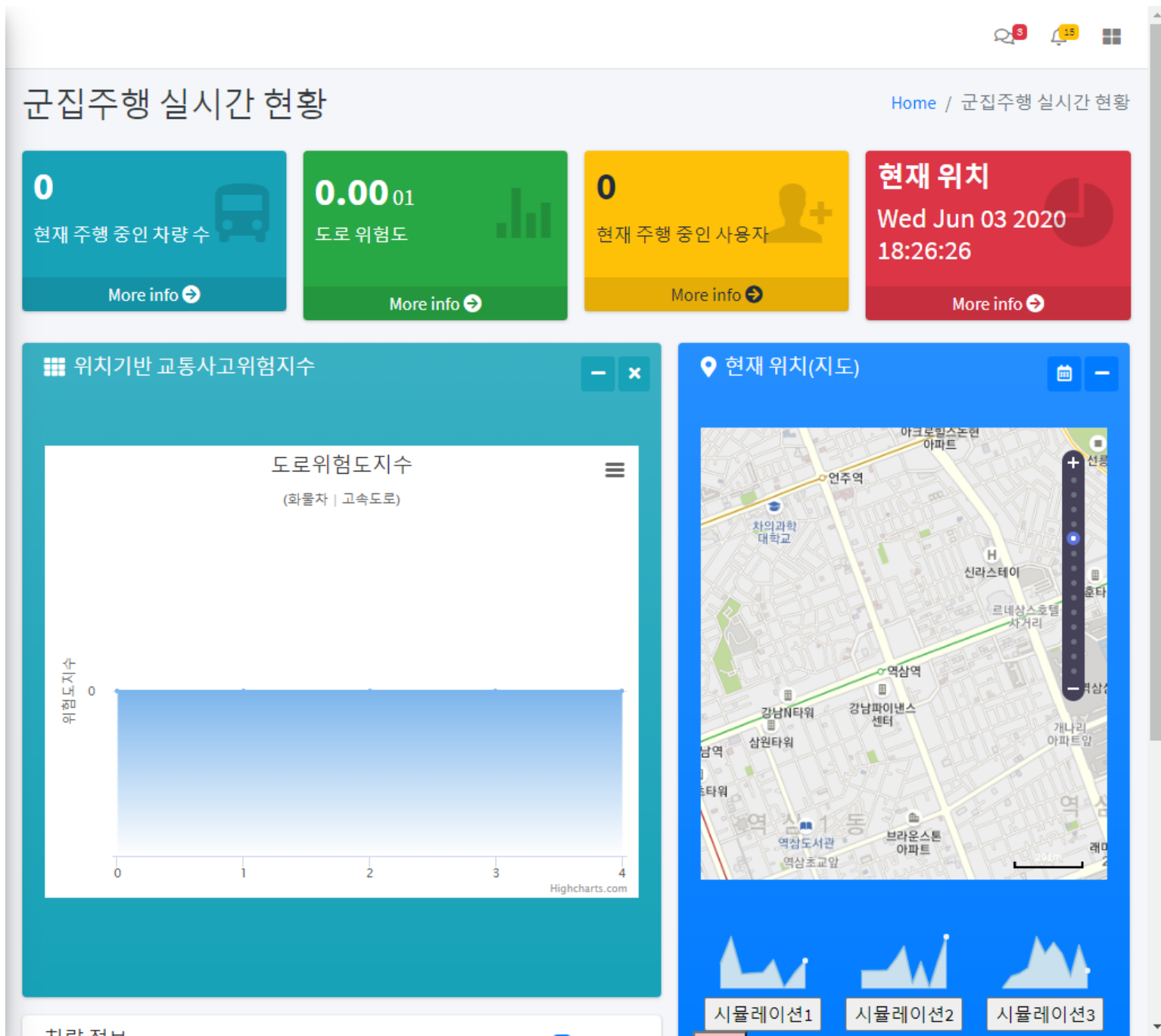
관리자 로그인을 지원한다.



# 차량 관제

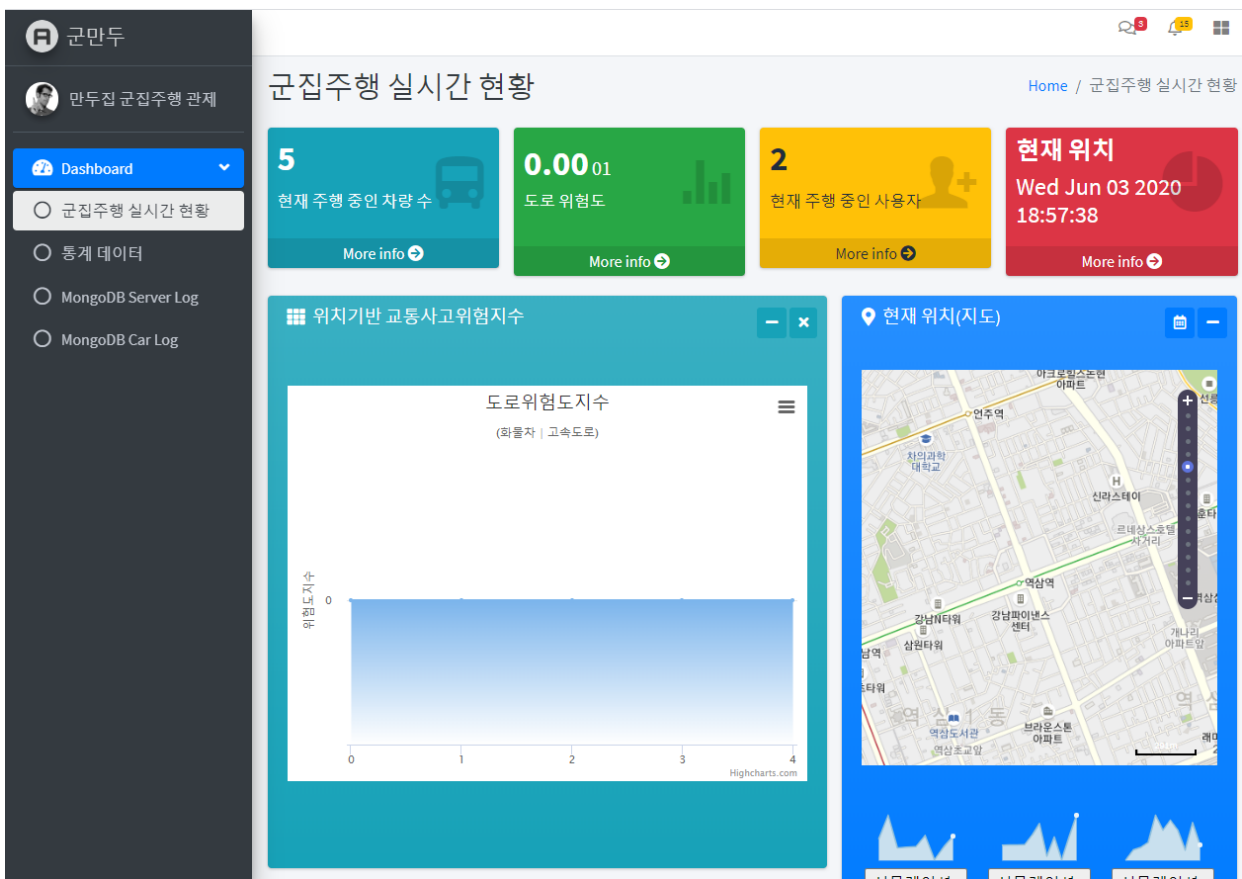
## 1. 첫 화면

5초 간격으로 실시간으로 DB의  
군집주행 중인 차량과  
현재 주행 중인 사용자 정보를  
업데이트한다.

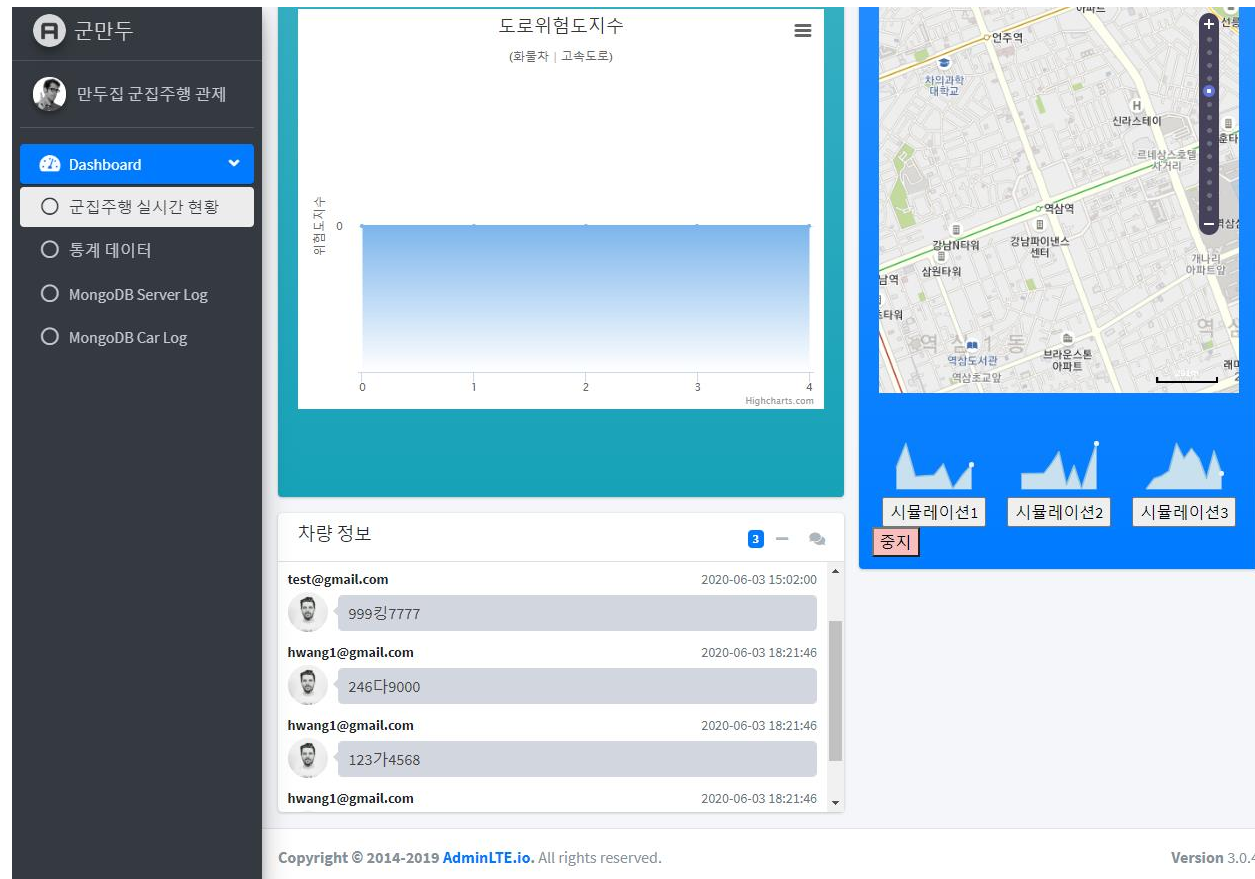




# 차량 관제



2. DB정보를 가져온 후  
차량 수와 현재 주행 중인 사용자 업데이트



3. 차량 정보(탑승 중인 사용자 id, 차량 번호)  
출력



# 차량 관제



실내에서 RC카의 GPS센서 정보를 읽어올 수 없으므로 Kml파일을 이용하여 군집주행 상황을 시뮬레이션 해준다.



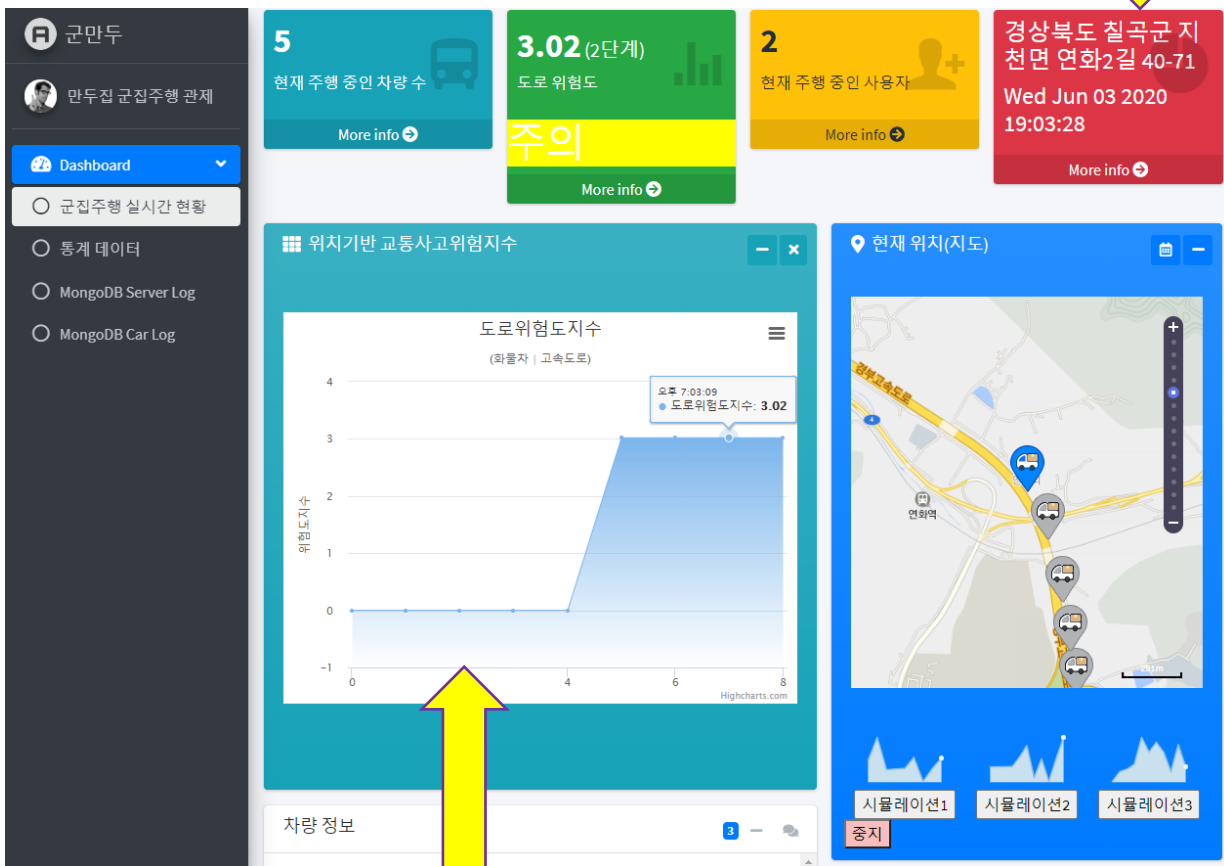
세부링크 도로위험도 지수 API를 이용하여 날씨, 차종, 소통상황, 도로구간에 따른 도로위험도를 보여준다.





# 차량 관제

Tmap Reverse Geocoding을  
이용하여 실시간 위치 표시



Highcharts를 이용한  
실시간 도로위험도지수 그래프

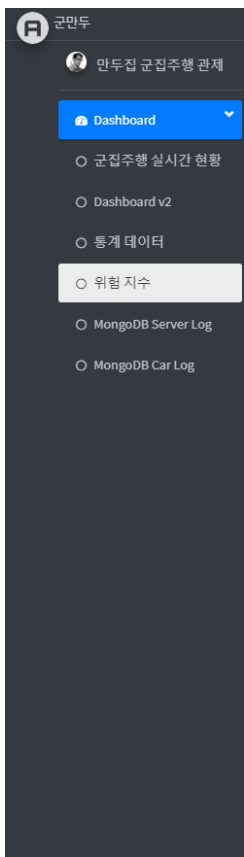
도로 위험도 지수와  
평가 매트릭스 4단계  
(안전, 주의, 위험, 심각)



도로 위험도 지수 3단계 이상이 되면  
FCM으로 태블릿에 주의 메시지 전송



# 데이터 분석



## 1. 2007-2019년(13년) 교통사고 데이터 분석

```
In [1]: import pandas as pd
data = pd.read_csv("./accident_hwamul_csv.csv")
print(data)

print("원본 특성:\n", list(data.columns), "\n")
data_dummies = pd.get_dummies(data)
print("get_dummies 후의 특성:\n", list(data_dummies.columns))

display(data_dummies.head())

features = data_dummies.loc[:, ['가해운전자 연령', '요일_금요일', '요일_목요일', '요일_수요일', '요일_월요일', '요일_일요일', '요일_화요일', '기상상태_눈', '기상상태_맑음', '기상상태_비', '기상상태_안개', '기상상태_호우', '가해운전자 성별_남', '가해운전자 성별_여', '시간대_아간', '시간대_주간', '월_10월', '월_11월', '월_12월', '월_1월', '월_2월', '월_3월', '월_4월', '월_5월', '월_6월', '월_7월', '월_8월', '월_9월']]

X = features.values
y = data_dummies['도로위험도'].values

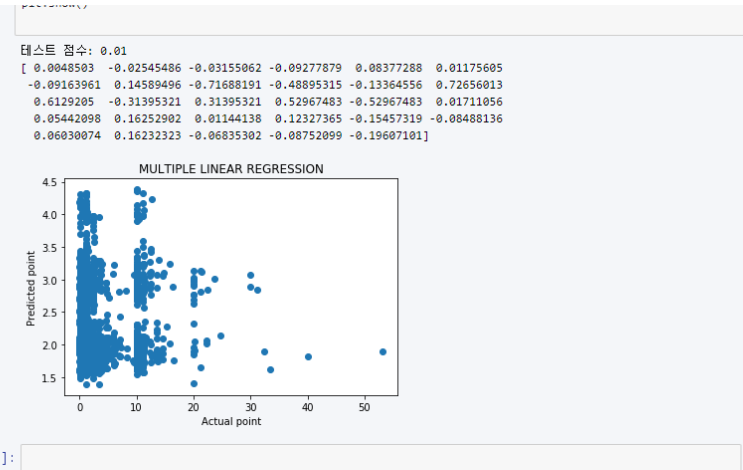
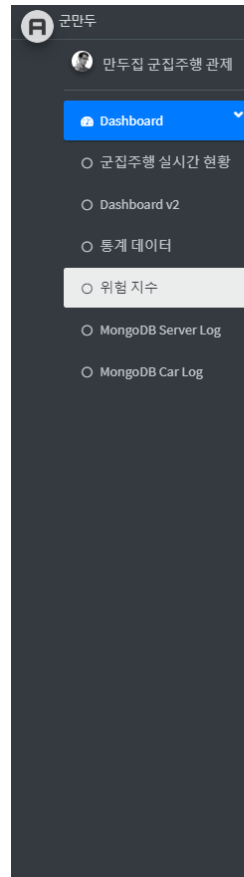
print("X.shape: {} y.shape: {}".format(X.shape, y.shape))
```

	요일	기상상태	가해운전자 성별	가해운전자 연령	시간대	월	도로위험도
0	금요일	맑음	남	45	아간	1월	0.033
1	수요일	맑음	남	58	주간	2월	0.068
2	월요일	맑음	남	48	주간	2월	1.168
3	목요일	맑음	남	37	주간	3월	0.068
4	화요일	맑음	남	49	주간	4월	0.136
...	...	...	...	...	...	...	...
11353	화요일	맑음	남	59	주간	12월	0.068
11354	화요일	맑음	남	60	주간	12월	0.068
11355	토요일	맑음	남	64	주간	12월	0.305
11356	토요일	맑음	남	66	주간	12월	0.068
11357	화요일	맑음	남	71	주간	12월	0.169

[11358 rows x 7 columns]  
원본 특성:  
['요일', '기상상태', '가해운전자 성별', '가해운전자 연령', '시간대', '월', '도로위험도']

get\_dummies 후의 특성:  
['가해운전자 연령', '도로위험도', '요일\_금요일', '요일\_목요일', '요일\_수요일', '요일\_월요일', '요일\_일요일', '요일\_화요일', '기상상태\_눈', '기상상태\_맑음', '기상상태\_비', '기상상태\_안개', '기상상태\_호우', '가해운전자 성별\_남', '가해운전자 성별\_여', '시간대\_아간', '시간대\_주간', '월\_10월', '월\_11월', '월\_12월', '월\_1월', '월\_2월', '월\_3월', '월\_4월', '월\_5월', '월\_6월', '월\_7월', '월\_8월', '월\_9월']

가해운전자 연령	도로위험도	요일_금요일	요일_목요일	요일_수요일	요일_월요일	요일_일요일	요일_화요일	기상상태_눈	기상상태_맑음	기상상태_비	기상상태_안개	기상상태_호우	시간대_아간	시간대_주간	월_10월	월_11월	월_12월	월_1월	월_2월	월_3월	월_4월	월_5월	월_6월	월_7월	월_8월	월_9월
45	0.033	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
58	0.068	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
48	1.168	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
37	0.068	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
49	0.136	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	



## 2. 2019년(1년) 교통사고 데이터 분석[전체, 성별]

```
In [1]: import pandas as pd
data = pd.read_csv("./accident_hwamul_gender_male.csv")
print(data)

print("원본 특성:\n", list(data.columns), "\n")
data_dummies = pd.get_dummies(data)
print("get_dummies 후의 특성:\n", list(data_dummies.columns))

display(data_dummies.head())

features = data_dummies.loc[:, ['기상상태_눈', '기상상태_맑음/호우', '기상상태_비', '기상상태_안개', '시간대_아간', '시간대_주간']]

X = features.values
y = data_dummies['도로위험도'].values

print("X.shape: {} y.shape: {}".format(X.shape, y.shape))
```

세부링크 도로위험도지수 API 도출방법 분석  
(머신러닝)





# 데이터 분석

## 통계 데이터

[Home](#) / 통계 데이터

지난 10년 간 연도별 사고 건수

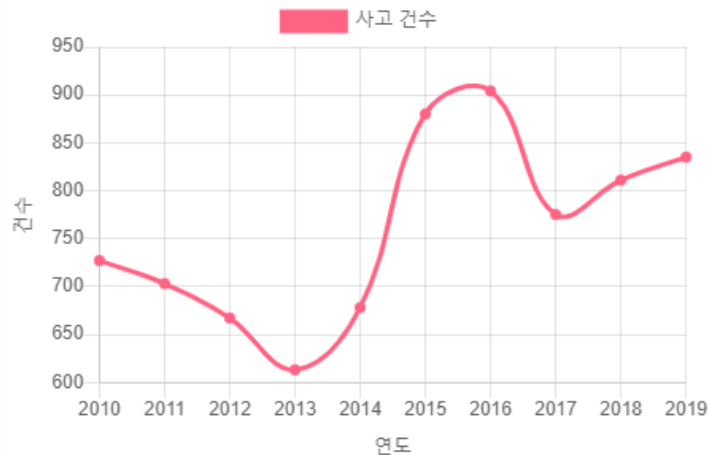
[데이터 확인](#)

835

2019년 화물차 사고 건수

↑ 2.9 %

작년 대비



월별 사고 합계

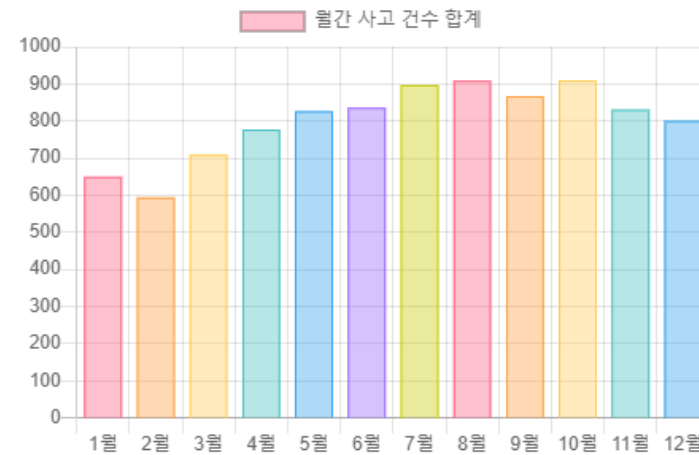
[데이터 확인](#)

801 건

2019년 12월 발생 사고 건수

↓ 3.7%

지난 달 대비



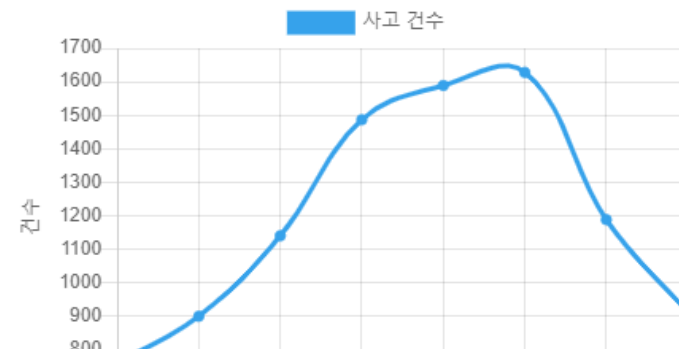
요일별 교통사고 통계 (전체)



요일	건수
월요일	1414
화요일	1546
수요일	1469
목요일	1481

시간대별 사고 통계

[데이터 확인](#)







# 데이터 분석

## 통계 데이터

[Home](#) / 통계 데이터

### MongoDB Realtime LOG

[T]:2020/06/03/18:26:23 [LOGIN]:admin

[T]:2020/06/03/18:21:46 [SetPltning]:hwang1@gmail.com=>246ㄷ9000

[T]:2020/06/03/18:21:46 [SetPltning]:hwang1@gmail.com=>123가4568

[T]:2020/06/03/18:21:46 [SetPltning]:hwang1@gmail.com=>100ㄴ7500

[T]:2020/06/03/18:21:34 [CarDetachAll]:hwang1@gmail.com

[T]:2020/06/03/18:20:23 [SetPltning]:hwang1@gmail.com=>246ㄷ9000

[T]:2020/06/03/18:20:23 [SetPltning]:hwang1@gmail.com=>123가4568

[T]:2020/06/03/18:20:23 [SetPltning]:hwang1@gmail.com=>100ㄴ7500

[T]:2020/06/03/18:12:04 [CarDetachAll]:hwang1@gmail.com

[T]:2020/06/03/18:09:24 [LoginByFace]:hwang1@gmail.com=hwang->100ㄴ7500

[T]:2020/06/03/18:09:14 [SetPltning]:hwang1@gmail.com=>246ㄷ9000

[T]:2020/06/03/18:09:14 [SetPltning]:hwang1@gmail.com=>123가4568

[T]:2020/06/03/18:09:14 [SetPltning]:hwang1@gmail.com=>100ㄴ7500

[T]:2020/06/03/18:09:01 [CarDetachAll]:hwang1@gmail.com



# 데이터 분석

## 통계 데이터

[Home](#) / 통계 데이터

### MongoDB Car LOG

2020/06/03/11:37:03->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:36:53->[id]:null[T]29[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:36:09->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:50->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:43->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:38->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:21->[id]:null[T]29[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:10->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:35:03->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:34:51->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:34:11->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:33:52->[id]:null[T]null[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:32:55->[id]:null[T]29[D]null[V]null[F]null[G]null[C]null

2020/06/03/11:32:48->[id]:null[T]28[D]null[V]null[F]null[G]null[C]null