

Learn

Think

# Portfolio

Create



**박창선, CHANGSEON PARK**

FrontEnd Developer



cscs0829@naver.com



010 9458 1351

# 목차

**Profile** 자기소개

---

**Abilities** 기술 스택

---

**Project** 경력기술

---

**Project** 경력기술

---

**end** 마무리

---

# Profile.

## Introduction

Name	박창선
Birth	1999.08.29
Mobile	010.9458.1351
E.mail	cscs0829@naver.com
Education	2025.02 조선대학교 졸업
major	전) 전자공학과 복) AI헬스케어

## Educations

course
2024.11 ~ 2025.05 DCX기반 빅데이터 분석서비스 개발자 과정(960H)

## Experience

course
2022.03 ~ 2023.01 코로나19 선별진료소 병상 배정 및 문의 응대
2022.01 ~ 2022.02. 헬스허브 인턴
2020.12 ~ 2021.02 무주덕유산리조트 리프트 안전요원

## Awards & Certifications

Subject
2024.12 조선대학교 IT Festival(종합학술대회) 동상
2024.12 조선대학교 캡스톤디자인 경진대회 동상
2024.06 조선대학교 AI 헬스케어 캡스톤디자인 경진대회 은상
2021.10 컴퓨터활용2급

# Abilities

기술 스택

## Front End



## Server



## Back End



## ETC



## DB



# Project.

## GoStop

**팀명 : Error Bros**  
**팀장 : 손민찬**  
**팀원 : 박창선 정경현**

### 주요 기능

- ✓ 원하는 휴게소 편의시설 검색 GO !
- ✓ 휴게소 식당 메뉴를 한눈에 GO !
- ✓ 휴게소 도착 전 메뉴 주문 GO !
- ✓ 게시판을 통해 다른 사용자와 소통 GO !

### 주요 기술

- ✓ 크롤링&머신러닝을 통해 약 3,000개 데이터 수집
- ✓ Kakao Map&Pay API 활용 및 시각화
- ✓ 관리자 계정을 통해 회원/휴게소/리뷰 관리
- ✓ 휴게소의 평가 지표 시각화

### 주요 화면/콘텐츠



✓ 검색한 휴게소 주소/편의시설/리뷰 조회 가능    ✓ 휴게소 별 선택한 메뉴 카카오페이 결제 가능

### 기술 스택

Front End

Back End

API

IDE

Data&Server

Etc



## 고속도로 휴게소 온라인 플랫폼 (Go Stop)

설명 사용자 휴게소를 검색하여 편의시설 정보를 제공받고 음식 메뉴를 사전에 주문받는 서비스

Date 2025.01 ~ 2025.03

참여자 3명

담당역할 프론트엔드

Skills 언어 : Java(JDK 1.8), JavaScript  
서버 : Apache Tomcat 9.0  
프레임 워크 : Spring  
DB : MySQL (8.0.41)  
IDE : VisualStudio Code, STS3 (Eclipse 4.21.0)  
API : 카카오 맵 API, 포트원 API

목표 비대면 주문 시스템을 통해 음식 주문 및 결제, 대기 시간 단축  
휴게소 편의시설 정보 제공, 리뷰 시스템 구축 등 다양한 서비스 제공

담당 프론트엔드 개발 및 반응형 웹 페이지 구축  
UI/UX 디자인 및 웹 페이지 레이아웃 설계

성과 Kakao Map API를 활용하여 휴게소 위치 시각화 및 포트원 API를 이용한 결제 기능을 구현  
게시판 기능을 통해 사용자 간의 정보 공유 및 소통을 활성화  
웹사이트를 한눈에 볼 수 있게 시각화하여 사용자에게 더 직관적으로 정보를 제공

Link <https://github.com/2024-SMHRD-DCX-BigData-16/Error-Bros.git>

# 프로젝트 중 말은 부분

## 1. 네이버 카페 크롤링

```
[15]:
from selenium import webdriver as wb
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from tqdm import tqdm
import pandas as pd
import time
import re

def preprocess_sentence_kr(w):
    w = w.strip()
    w = re.sub(r"[^0-9가-힣?.!,,:]+", " ", w)
    w = w.strip()
    return w

[16]:

driver = wb.Chrome()

[17]:

url = 'https://www.naver.com'
driver.get(url)

[18]:

keyword = '고속도로 휴게소'
search_input = driver.find_element(By.ID, 'query')
search_input.send_keys(keyword+'\n')

[19]:

# 2. "카페" 탭 클릭
driver.find_element(By.LINK_TEXT, '카페').click()
```

- 네이버 카페에서 휴게소 데이터를 수집
- 수집한 데이터를 불용어 처리

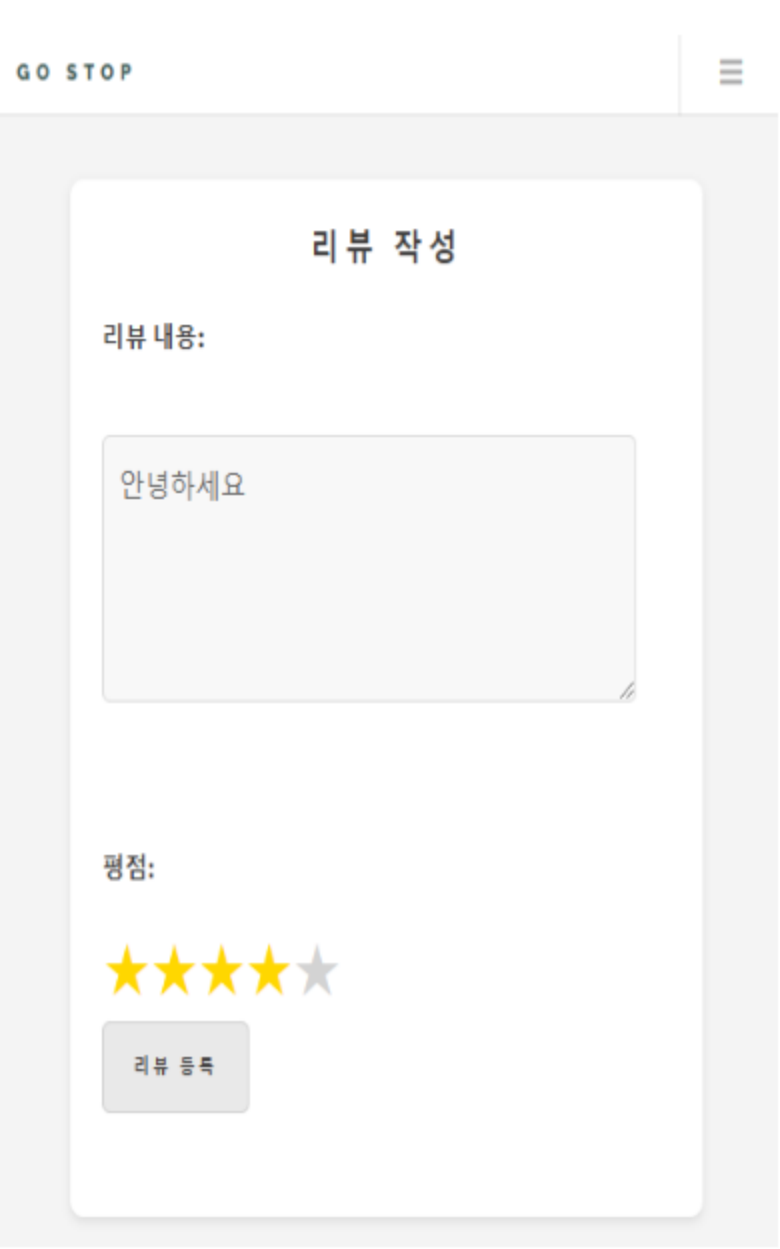
## 2. 음식 대기 순번 및 리뷰 목록 기능 구현



- 사용자에게 음식 대기 순번 정보를 실시간으로 제공
- 리뷰 목록을 통해 사용자들의 리뷰를 확인 가능

Title	Go stop
Rule	프론트엔드

## 3. 리뷰 기능 구현 및 애니메이션 효과

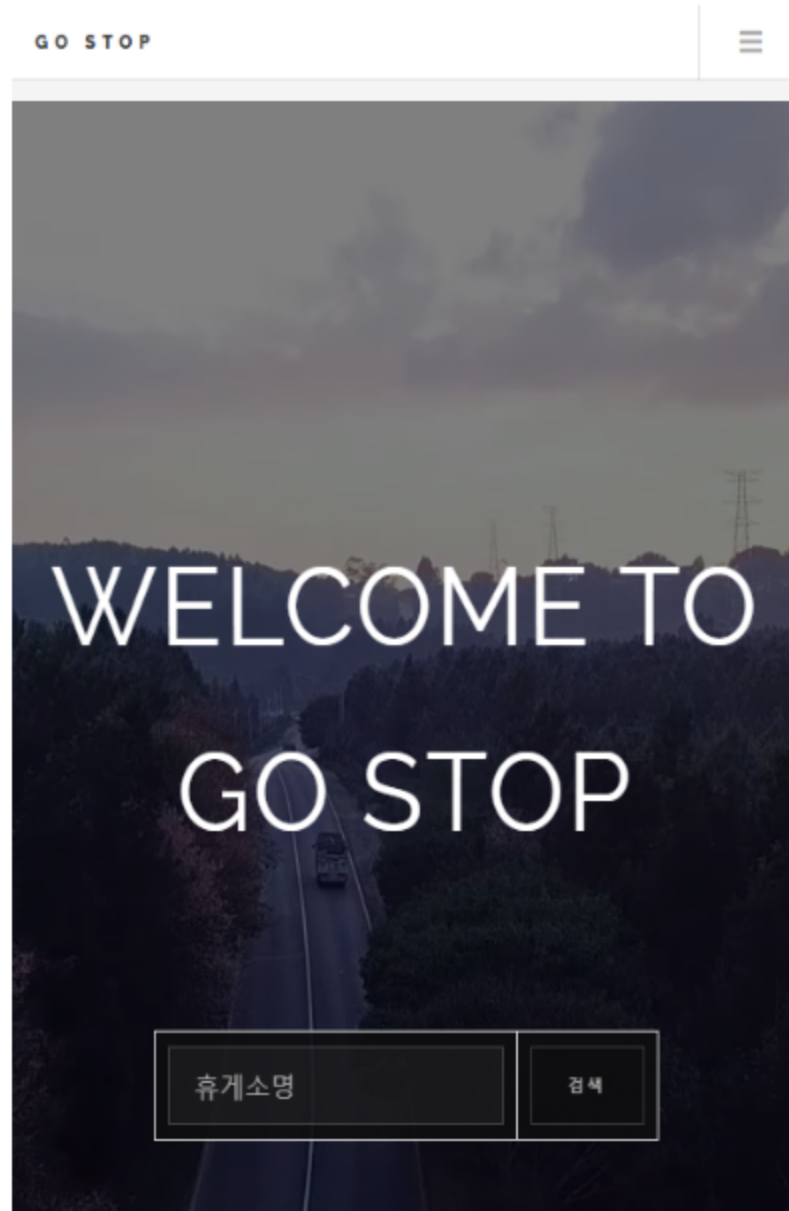


- 사용자가 리뷰를 작성하고 평점을 남길 수 있는 기능을 구현
- CSS를 사용하여 별점에 애니메이션 효과를 적용



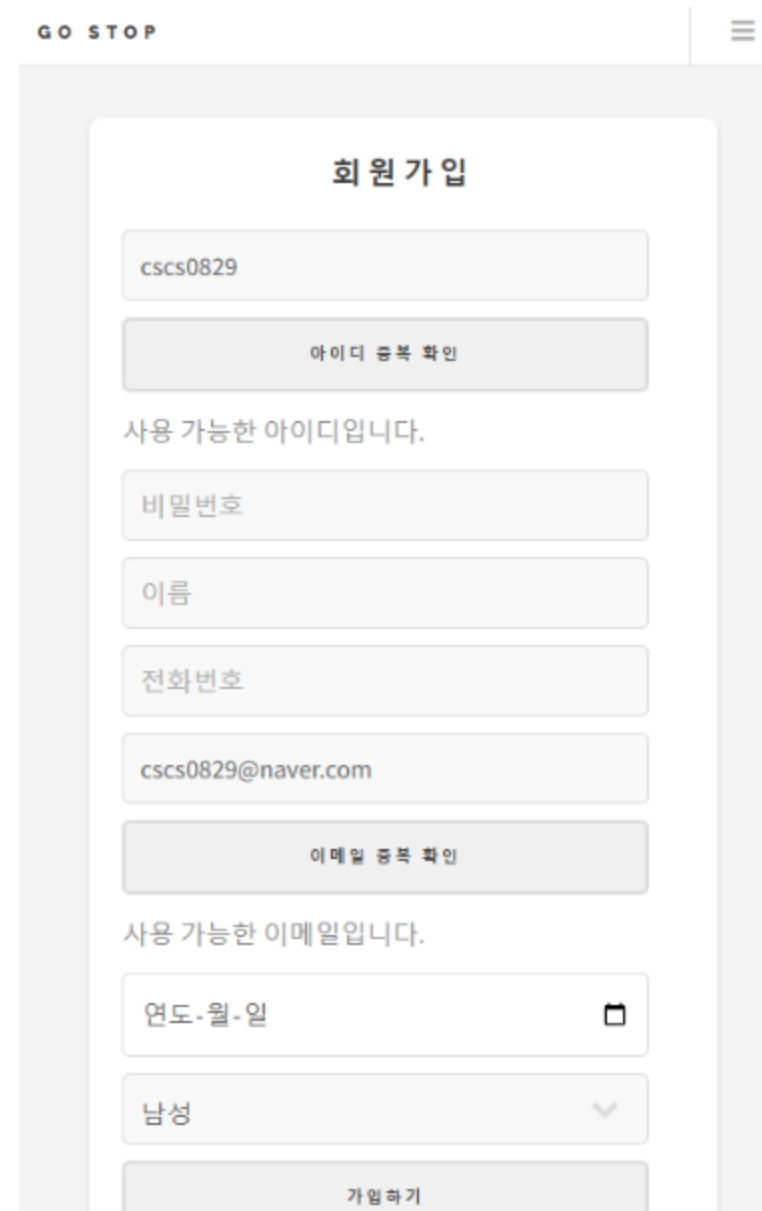
# 프로젝트 중 맡은 부분

## 4. 반응형 페이지구현



- 다양한 기기에 최적화된 반응형 웹 페이지를 구현

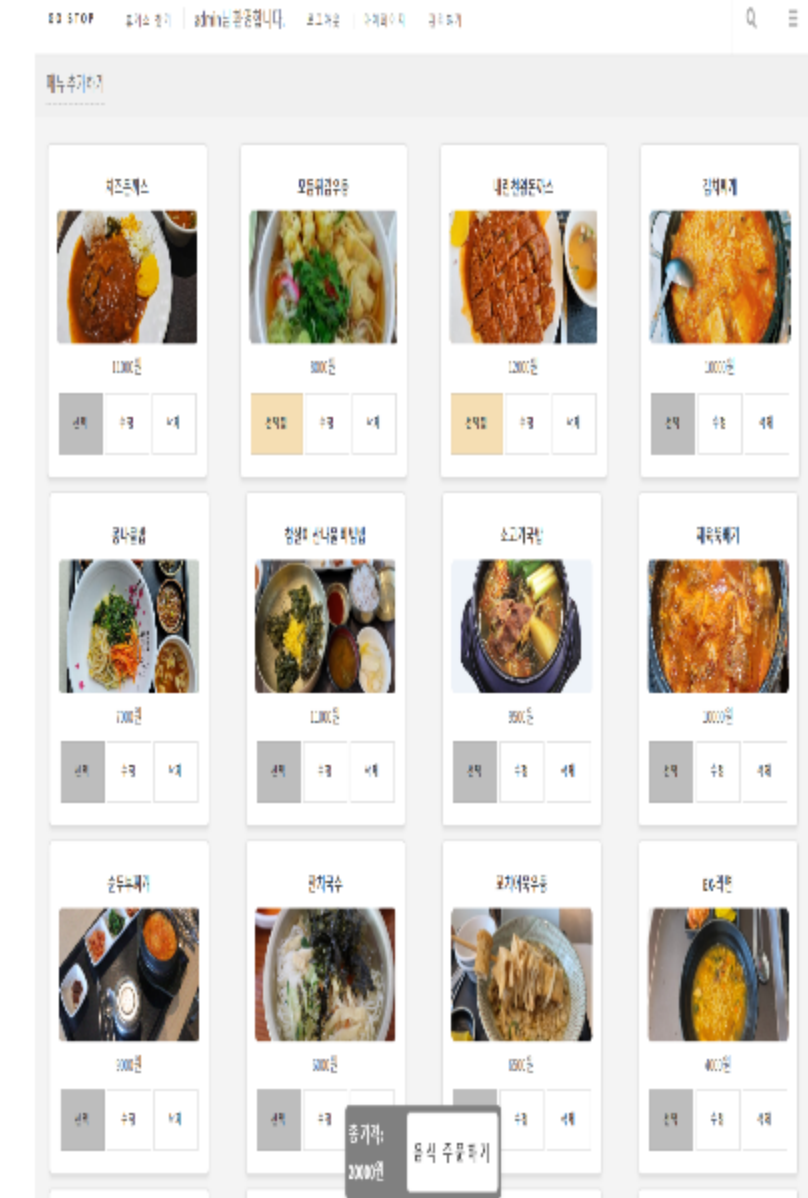
## 5. 중복 아이디 및 이메일 검증



- AJAX를 활용하여 실시간으로 중복 아이디 및 이메일을 검증

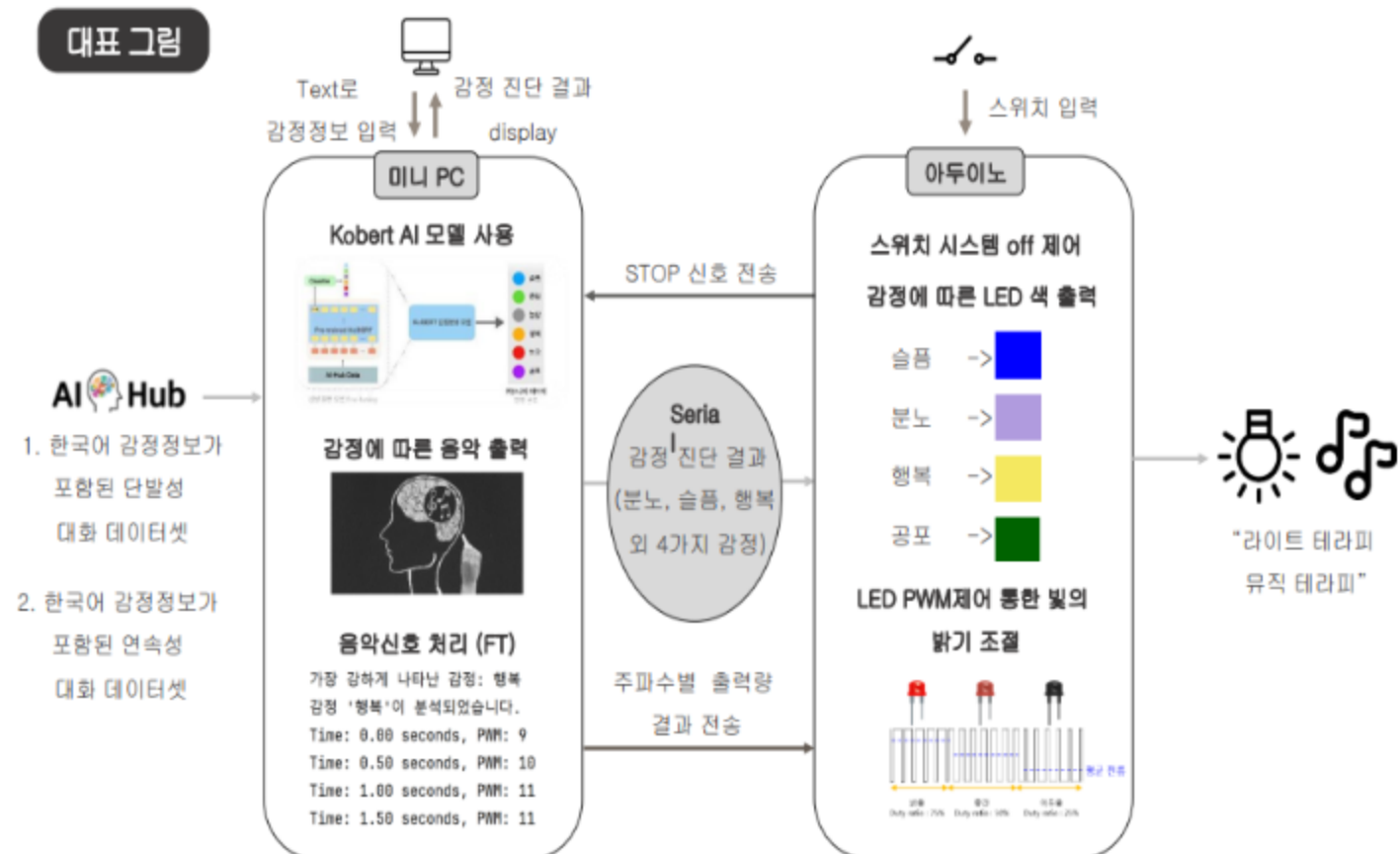
Title	Go stop
Rule	프론트엔드

## 6. UI/UX 디자인 및 레이아웃 설계



- 사용자 중심의 디자인 원칙을 적용하여 직관적이고 사용하기 쉬운 웹 페이지 레이아웃을 설계

# Project.



## KoBert 모델을 활용한 라이트테라피 조명 개발

설명	사용자가 입력한 텍스트를 분석하여 감정을 진단하고, 그에 맞는 음악과 조명을 제공하는 시스템
Date	2024.03 ~ 2024.06
참여자	4명
담당역할	프론트엔드 및 모델과 회로 시리얼 통신
라이브러리	언어 : Python KcBERT TensorFlow / PyTorch Streamlit Pygame serial /numpy/scipy
목표	한국어 감정 정보가 포함된 텍스트 데이터셋을 사용하여 KoBERT 모델을 학습시켜 감정을 진단 감정 진단 결과에 따라 LED 색상을 출력하고, 음악 신호 처리를 통해 빛의 밝기를 조절하는 시스템을 구현
담당	KoBERT 모델을 사용하여 감정을 7개로 분류하고 Streamlit을 사용하여 인터페이스 개발 미니 PC와 아두이노 간의 실시간 데이터 교환을 위해 시리얼 통신을 구현
성과	KoBERT 모델을 사용하여 텍스트 데이터셋에서 7가지 감정을 85%의 정확도로 분류 감정 분석 결과에 따라 LED를 출력하고, 음악 신호의 진폭값에 따라 빛의 밝기를 조절하는 시스템 구현
Link	<a href="https://github.com/cscs0829/capston_sentiment">https://github.com/cscs0829/capston_sentiment</a>



# 프로젝트 중 말은 부분

Title	Go stop
Rule	프론트엔드

## 1. 인터페이스 구현

### 감정분석(Sentiment Analysis)

글에 나타난 공포, 놀람, 분노, 슬픔, 중립, 행복, 혐오의 정도를 비율로 알려드립니다.

이곳에 글 입력(100자 이하 권장):

아 오늘 뭐하지

시작

{

"공포" : 0.07057388126850128

"놀람" : 0.0002933132345788181

"분노" : 0.0034623260144144297

"슬픔" : 0.01157877966761589

"중립" : 0.9107363820075989

"행복" : 0.0025046253576874733

"혐오" : 0.0008506650337949395

}

가장 강하게 나타난 감정: 중립

- Streamlit을 통한 인터페이스 구현

## 2. 시리얼 통신

```
# 시리얼 통신 설정
SERIAL_PORT = 'COM3' # Arduino의 포트를 여기에 입력하세요 (예: '/dev/tty
BAUD_RATE = 9600

1개의 사용 위치
def send_to_arduino(emotion):
    with serial.Serial(SERIAL_PORT, BAUD_RATE, timeout=1) as ser:
        ser.write(emotion.encode('utf-8'))

# 사용자 입력 받기
def main():
    print('감정분석 프로그램입니다. 글에 나타난 공포, 놀람, 분노, 슬픔, 중립,
    user_input = input("이 곳에 글 입력(100자 이하 권장): ")

    result = inference(user_input)
    print('감정 분석 결과:', result[0])
    print(f'가장 강하게 나타난 감정: {result[1]}')

    # 중립 감정인 경우 재입력 요청
    if result[1] == '중립':
        print("감정이 중립으로 나타났습니다. 다른 텍스트를 입력해 주세요.")
    else:
        # 결과를 Arduino로 전송
        send_to_arduino(result[1])
        print(f"Arduino로 '{result[1]}' 감정을 전송했습니다.")

if __name__ == "__main__":
    main()
```

- 감정 결과에 따라 빛의 세기 조절을 위한 아두이노와 통신 구현

## 3. KoBert 모델을 이용한 감정 분석

```
{'train_acc': 0.75,
 'train_f1': 0.75,
 'train_loss': 0.5262309908866882,
 'train_precision': 0.8571428571428571,
 'train_recall': 0.6666666666666666}

{'train_acc': 0.8125,
 'train_f1': 0.7857142857142856,
 'train_loss': 0.3342723548412323,
 'train_precision': 0.7333333333333333,
 'train_recall': 0.8461538461538461}

{'train_acc': 0.875,
 'train_f1': 0.8823529411764706,
 'train_loss': 0.2689765691757202,
 'train_precision': 0.8823529411764706,
 'train_recall': 0.8823529411764706}

{'train_acc': 0.875,
 'train_f1': 0.8461538461538461,
 'train_loss': 0.3174639940261841,
 'train_precision': 0.7857142857142857,
 'train_recall': 0.9166666666666666}

{'train_acc': 0.75,
 'train_f1': 0.5555555555555556,
 'train_loss': 0.38240528106689453,
 'train_precision': 0.8333333333333334,
 'train_recall': 0.4166666666666667}
```

- AI Hub 데이터와 KoBERT를 활용한 감정 분석 모델 구현

# Thank You!

여기까지 읽어주셔서 감사합니다.

---

*Connect.*

GitHub [github.com/cscs0829](https://github.com/cscs0829)

PHONE 010.9458.1351

E-Mail [cscs0829@naver.com](mailto:cscs0829@naver.com)

---