



(단위: °C)

날씨알리미

충남 지역혁신프로젝트 IoT

한밭대학교 전자공학과 김영환
한밭대학교 전자공학과 김보선

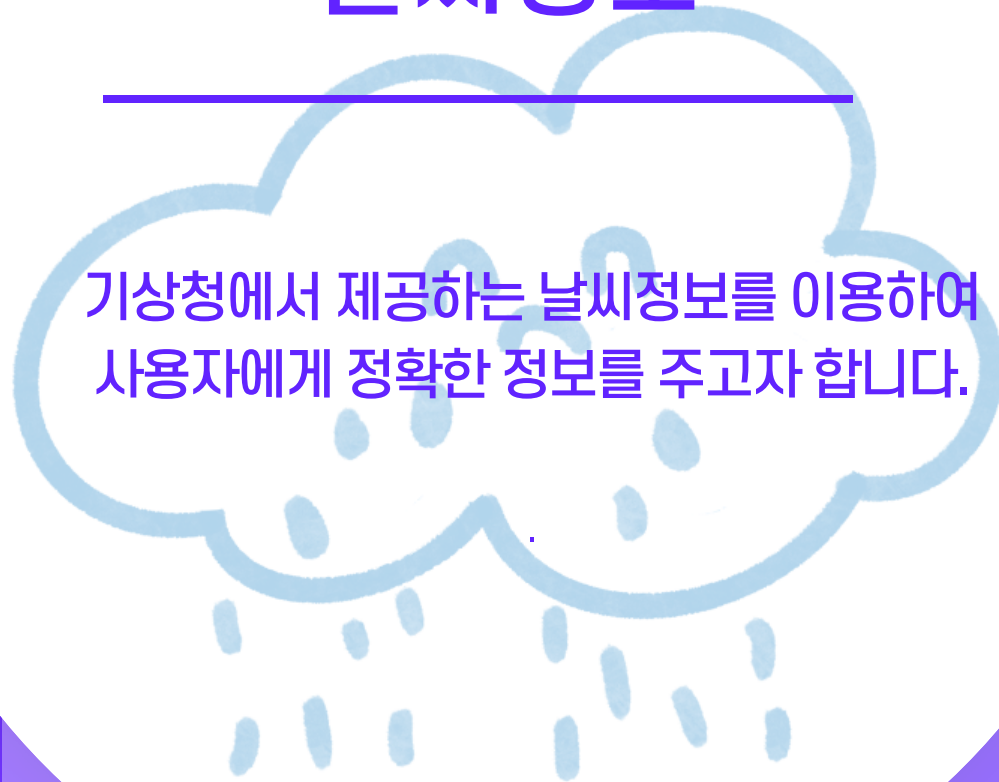
목차

.....	1.날씨정보
.....	2.주요기능
.....	3.실제화면
.....	4.사용기술
.....	5.사용코드
.....	6.발전사항

날씨알리미 주요기능

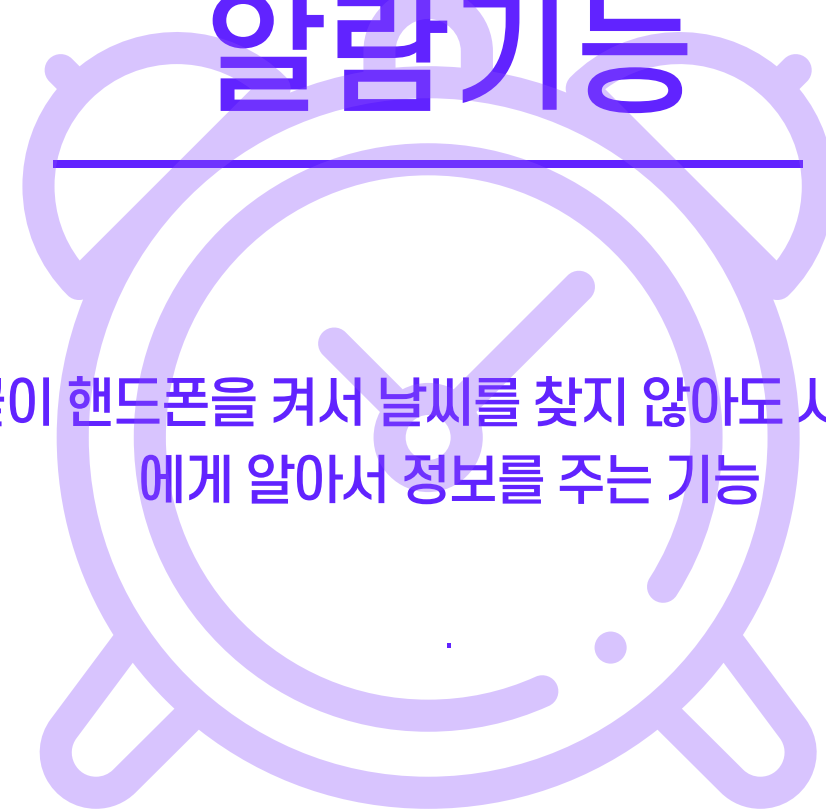
날씨정보

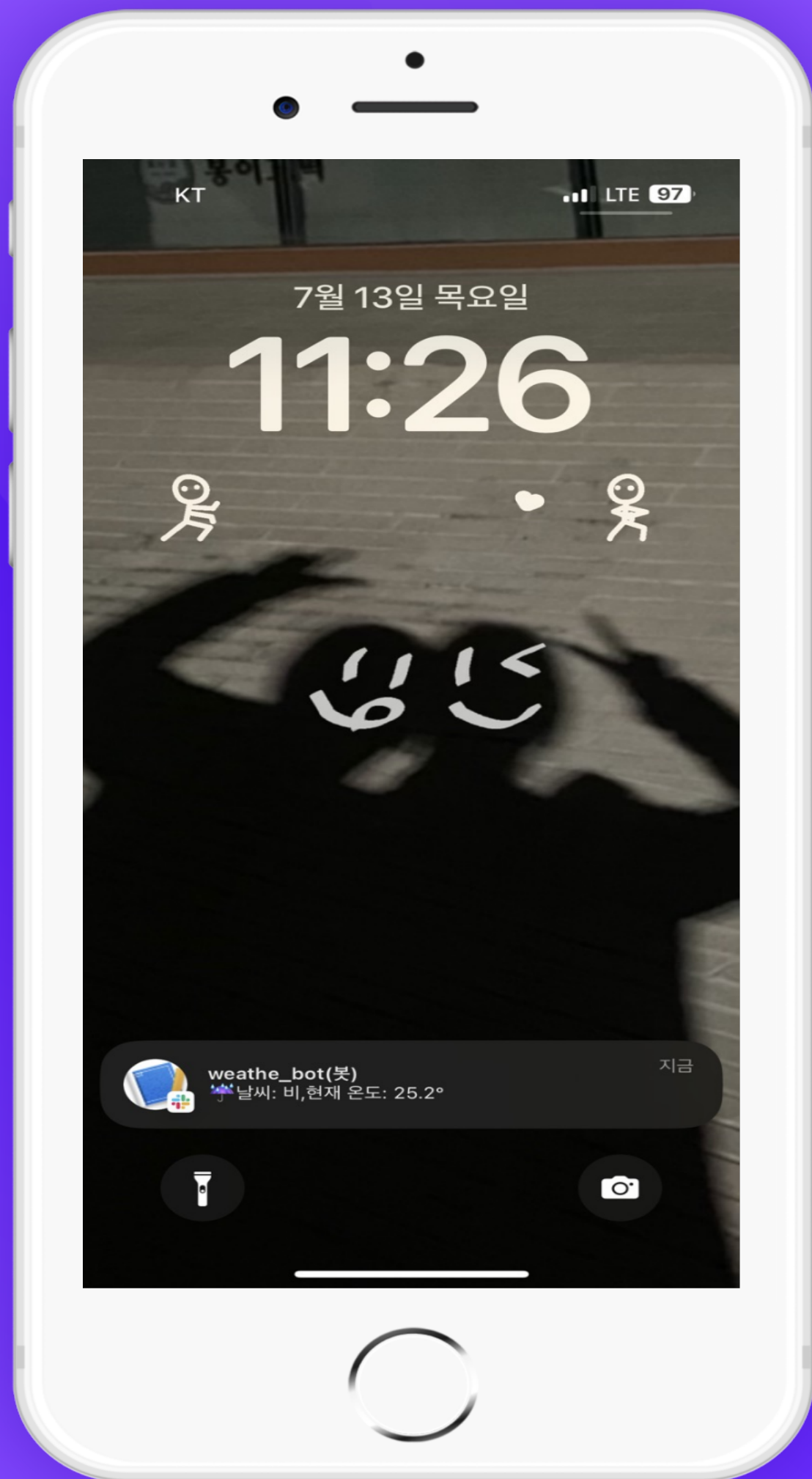
기상청에서 제공하는 날씨정보를 이용하여
사용자에게 정확한 정보를 주고자 합니다.



알람기능

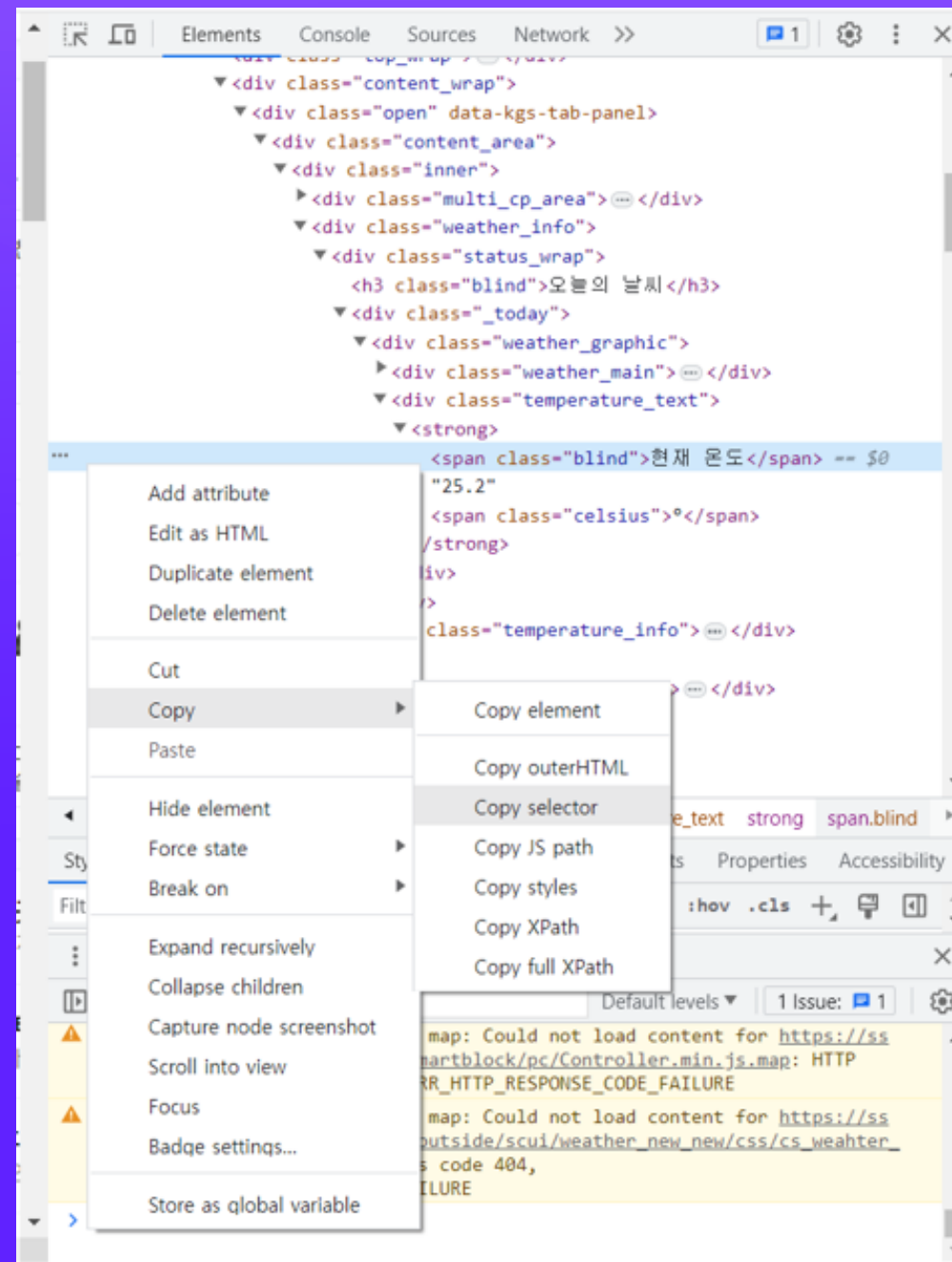
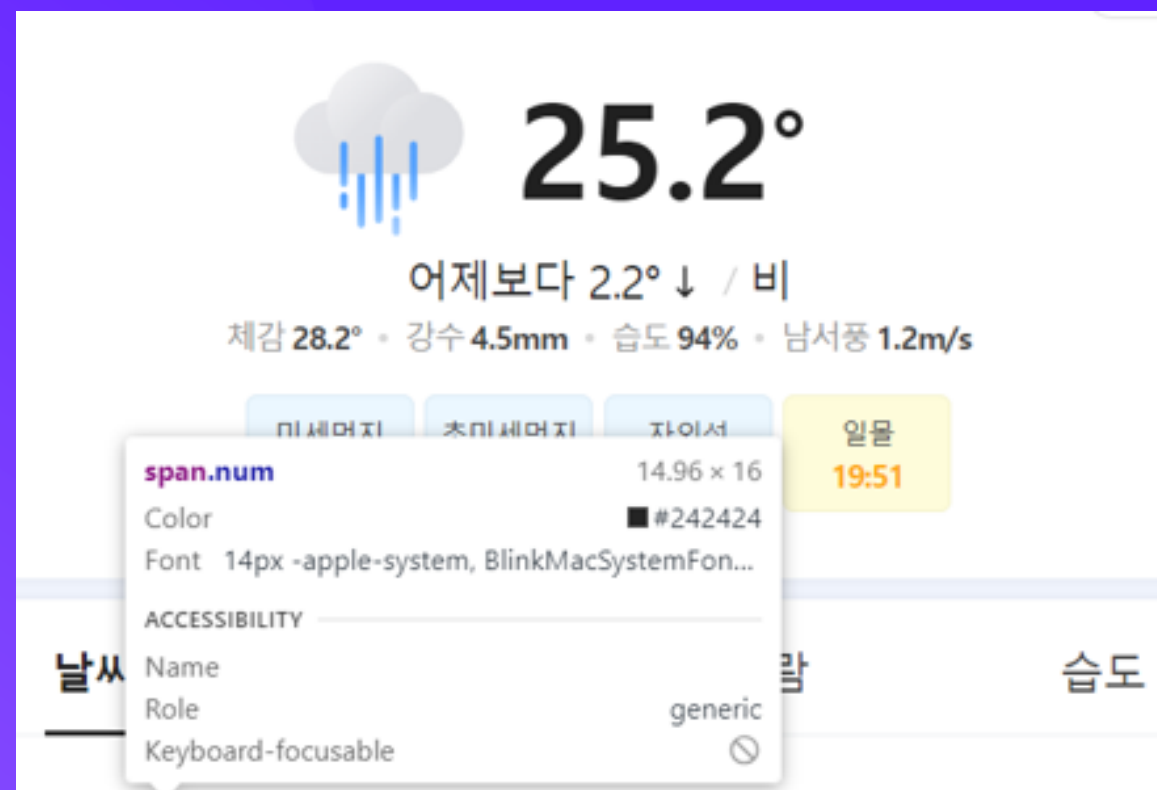
굳이 핸드폰을 켜서 날씨를 찾지 않아도 사용자
에게 알아서 정보를 주는 기능





실제 화면

이 날씨알리미를 사용하면 설정한 아침에 한 번 또 비오기 전에 한 더 알람이
가게끔하여 사용자가 날씨정보를 쉽게 얻을 수 있고 비를 대비할 수 있다.



사용기술

HTML 크롤링

정보찾기 | 원하는 정보가 있는 페이지로 이동

소스코드 | F12 를 눌러 해당 페이지의 소스코드를 볼 수 있음

원하는 selector찾기 | 원하는 값의 아이콘을 눌러 해당 소스코드의 위치로 이동하여 selector를 복사

코드작성 | 원하는 값이 출력되게끔 코드를 작성한다.

SLACK을 이용한 스마트폰으로 문자 보내기



사물인터넷 IoT 과정

CODE

```
import requests
from bs4 import BeautifulSoup
import json

# Slack 웹훅 URL 설정
slack_webhook_url = "https://hooks.slack.com/services/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"

# 날씨 정보 URL 설정
weather_url = "https://search.naver.com/search.naver?where=nexearch&sm=top_hyt&fbm=0&ie=utf8&query=%EB%82%A0%EC%94%A8"

def send_slack_webhook(str_text):
    """
    입력된 텍스트를 Slack 웹훅으로 전송하는 함수
    """
    Args:
        str_text (str): 전송할 텍스트

    Returns:
        str: 전송 결과 ("OK" 또는 "error")
    """
    headers = {
        "Content-type": "application/json"
    }
    data = {
        "text": str_text
    }
    res = requests.post(slack_webhook_url, headers=headers, data=json.dumps(data))
    if res.status_code == 200:
        return "OK"
    else:
        return "error"
```

CODE

```
def get_current_temperature():
    # 날씨 정보 가져오기
    response = requests.get(weather_url)
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')

    # 현재 온도 요소 선택
    tmp_elements = soup.select('#main_pack > section.sc_new.cs_weather_new._cs_weather > div._tab_flicking > div.content_w

    # 온도 추출
    print(tmp_elements[0].text)
    t = tmp_elements[0].text
    t = t.split('도')
    return t[1]

def get_current_weather():
    # 날씨 정보 가져오기
    response = requests.get(weather_url)
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')

    # 현재 날씨 요소 선택
    wth_elements = soup.select('#main_pack > section.sc_new.cs_weather_new._cs_weather > div._tab_flicking > div.content_w

    # 날씨 추출
    print(wth_elements[0].text)
    w = wth_elements[0].text
    return w
```

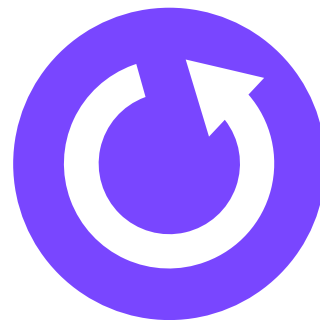

CODE

```
def get_icon(current_weather):
    # 날씨 아이콘 매핑
    if current_weather == '맑음':
        icon = '☀'
    elif current_weather == '구름많음':
        icon = '☁'
    elif current_weather == '구름조금':
        icon = '☁'
    elif current_weather == '흐림':
        icon = '☁'
    elif current_weather == '흐리고 비':
        icon = '🌧'
    elif current_weather == '비':
        icon = '🌧'
    else:
        icon = '🌧'
    return icon

# 현재 온도 가져오기
current_temperature = get_current_temperature()
# 현재 날씨 가져오기
current_weather = get_current_weather()
# 현재 날씨 아이콘 가져오기
current_icon = get_icon(current_weather)

# 메시지 생성
message = "{}날씨: {}, 현재 온도: {}".format(current_icon, current_weather, current_temperature)

# Slack 웹훅으로 메시지 전송
print(send_slack_webhook(message))
```



자동 알람 시스템

파이썬으로 실행파일과 가상 환경을 만들어 작업 스케줄러로 따로 실행없이 자동적으로 알람이 오는 시스템



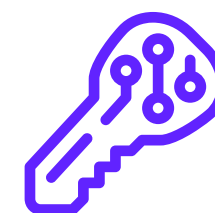
비 예보 시스템

현재 시간을 이용하여 비오기 한시간 전에 알람이 오게 하여 비를 대비 할 수 있도록 하는 예보 시스템



특보 시스템

태풍, 폭염, 폭우 같은 특보 시스템에 대한 정보를 얻어 사용자에게 추가적인 정보를 제공하는 시스템



사용자 맞춤

설정 시스템

일정 미세먼지 농도 이상일시 알람, 비 예보가 있어도 설정한 강수확률에 따른 사용자에게 맞춤 시스템

DEVELOPE

감사합니다.

김보선 김영한



🔍 사물인터넷IoT

IoT # 잘하고싶다 # 돈 많이벌고싶다