

# 프로젝트 수행 경력

구분	프로젝트명	기간	내용	비고
1	AI 이미지 생성	2024.03 ~ 2024.03	open api: 이미지 및 영상 생성	
2	UI, 로그인 제작	2024.01 ~ 2024.02	ini: 로그인/UI 제작	
3	웹 지도 제작	2024.02 ~ 2024.02	folium: 지도 및 상세 설명 만들기	
4	영상 제작	2024.01 ~ 2024.02	cv2: 이미지 활용한 영상제작	
5	RTSP 영상 호출	2024.02 ~ 2024.02	Thread: RTSP주소를 DB에 저장하여 재생	
6	소셜 지역 살리기	2024.02 ~ 2024.03	소켓,FTP 통신: 다중 클라이언트/서버 연결, 각 클라이언트별 개별 정보 받기	
7	개화 예측 제작	2024.03 ~ 2024.04	머신러닝: 봄철 꽃 개화 예측	
8	웹 구직사이트 만들기	2024.04 ~ 2024.04	웹크롤링: 구직 사이트 만들기	
9	차량 및 문자 인식 제작	2024.05 ~ 2024.06	YOLOv5: 차량 및 차량 번호판, 문자 인식	

# 개발 포트폴리오

2024.03.07

- ▶ 프로젝트명 : AI 이미지 생성
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, open api
- ▶ 담당 업무 : 이미지 생성 및 변환
- ▶ 목 적 : open api 이해

## ◆ 화면구성 및 기능 요약설명

```
# JSON 형식으로 변환
response = json.loads(r.content)
return response

# 프롬프트에 상세한 이미지
prompt = "Korean women, Age: 20s, medium length hair, dog type face, looks like a celebrity, cute, symmetry, frontal, A gentle smile"
negative_prompt = "facial asymmetry, ugly, ugly face, dog, puppy, Hanbok, chinese, asymmetric eyes"

# 이미지 생성하기 REST API 호출
response = t2i(prompt, negative_prompt)

# 응답의 첫 번째 이미지 생성 결과 출력하기
image_url = response.get("images")[0].get("image")
print(f"이미지 URL: {image_url}")

# 이미지 열기
result = Image.open(urllib.request.urlopen(image_url))
result.show()
# result.show()
# 이미지 저장하기
try:
    result.save(r"C:\study\20240307\image_create.png", 'PNG')
    print("이미지가 성공적으로 저장되었습니다.")
except Exception as e:
    print(f"이미지 저장 오류: {e}")
```



```
# JSON 형식으로 변환
response = json.loads(r.content)
return response

# Base64 인코딩
def imageToString(img):
    img_byte_arr = io.BytesIO()
    img.save(img_byte_arr, format='PNG')
    my_encoded_img = base64.encodebytes(img_byte_arr.getvalue()).decode('ascii')
    return my_encoded_img

# 이미지 파일 불러오기
img = Image.open('image_create1.png')

# 이미지를 Base64 인코딩하기
img_base64 = imageToString(img)

# 이미지를 Base64 인코딩한 문자열 저장
img_arr = []
img_arr.append(img_base64)

# 이미지 변환하기 REST API 호출
response = upscale(img_arr, 4)

# 응답의 첫 번째 이미지 생성 결과 출력하기
result = Image.open(urllib.request.urlopen(response.get("images")[0]))
try:
    result.save(r"C:\study\20240307\image_scaling.png", 'PNG')
    print("이미지가 성공적으로 저장되었습니다.")
except Exception as e:
    print(f"이미지 저장 오류: {e}")
```

◆ open api 코드를 입력합니다.

◆ 원하는 단어와, 원치 않는 단어를 입력하여 사진을 생성합니다.

◆ 해당 단어에 따라 사진이 생성되었습니다.

◆ 다음 코드를 실행 하여 해당 사진을 보정합니다.



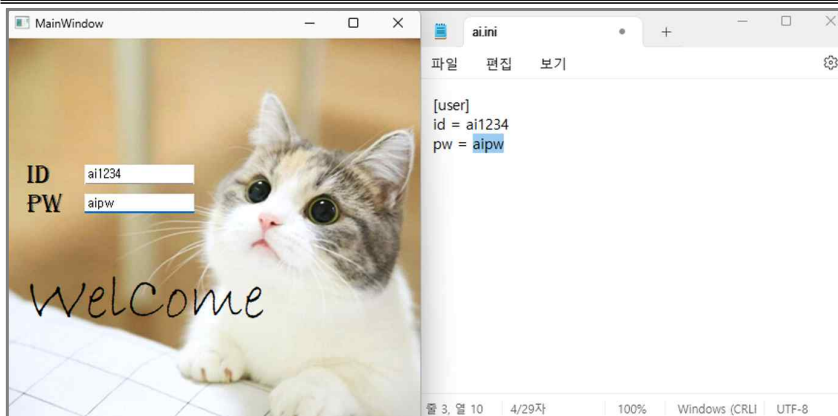
◆ 보정이 완료된 사진입니다.

# 개발 포트폴리오

2024.01.29 - 2024.02.05

- ▶ 프로젝트명 : UI, 로그인 제작
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, Qt Designer, ini
- ▶ 담당 업무 : 로그인 화면 제작 및 코딩
- ▶ 목 적 : Python 기초 코드 및 Qt Designer 원리 이해, ini파일 연결

## ◆ 화면구성 및 기능 요약설명



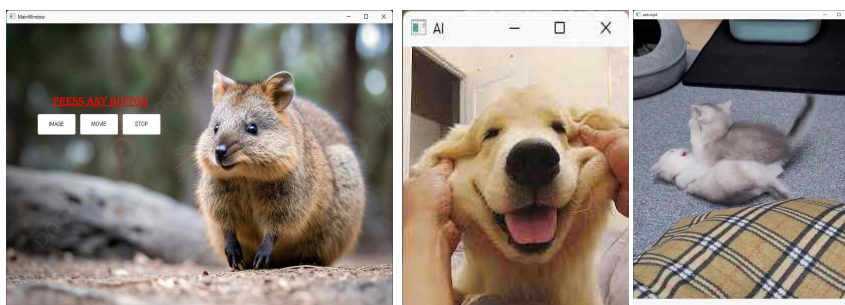
- ◆ ini파일을 호출하여 저장된 ID, Pw를 이용한 로그인 제작



- ◆ ID 또는 PW 미입력시 입력문구 변경 됩니다.



- ◆ ID 또는 PW 가 일정횟수 오류시, 오류 문구 송출, 일정시간동안 사용불가



- ◆ 로그인 성공시 UI 전환 됩니다.

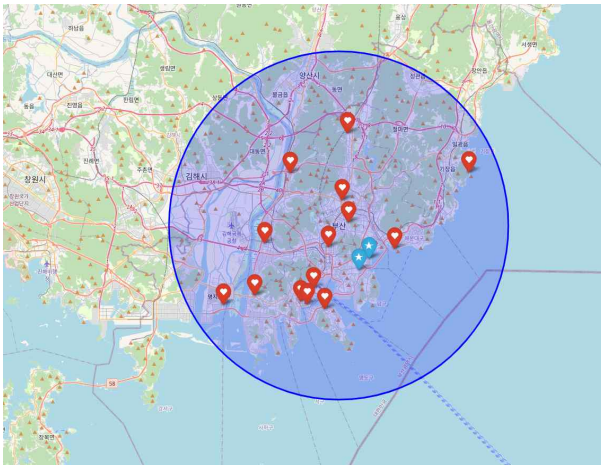
- ◆ 메뉴 클릭시 사진 및 영상 으로 화면전환 생성

# 개발 포트폴리오

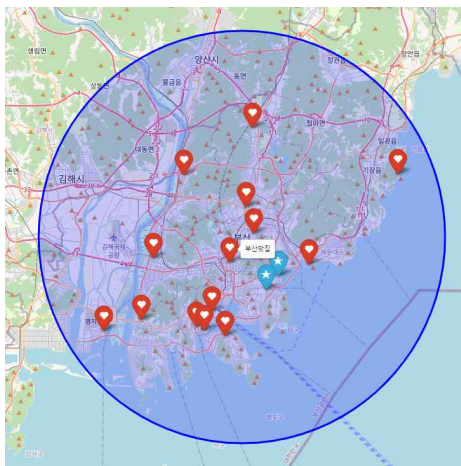
2024.02.16 - 2024.02.20

- ▶ 프로젝트명 : 웹 지도 제작
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, folium
- ▶ 담당 업무 : 지도 및 마커, 설명사진 제작
- ▶ 목 적 : folium을 사용한 웹 지도 생성

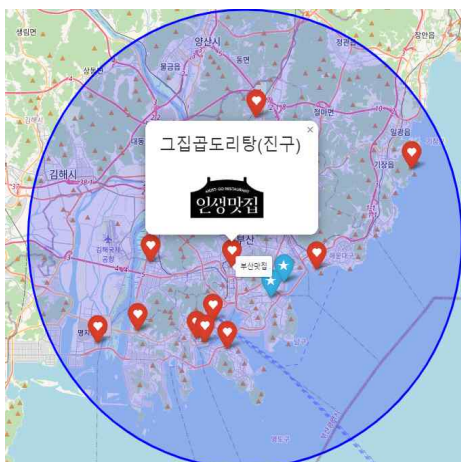
## ◆ 화면구성 및 기능 요약설명



- ◆ [지도상 부산시 맛집 리스트 구현]
- ◆ 실행시 해당 사진과 같이 지도가 실행되고 마커가 찍혀있습니다.



- ◆ 해당 마커에 마우스를 올리면 간략한 설명이 뜹니다.



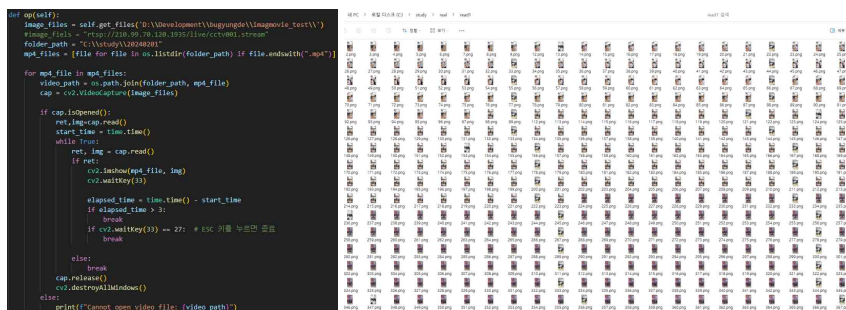
- ◆ 마커클릭시 상세설명과 해당 지역 관련 이미지를 표시합니다.

# 개발 포트폴리오

2024.01.31 - 2024.02.01

- ▶ 프로젝트명 : 영상 제작
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, CV2
- ▶ 인원 : 팀프로젝트 (5명)
- ▶ 담당 업무 : 조장, 영상 제작
- ▶ 목적 : Python을 통한 프레임의 이해

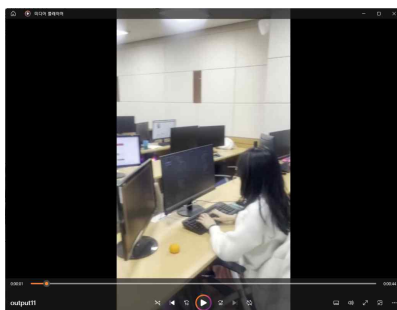
## ◆ 화면구성 및 기능 요약설명



- ◆ 앱을 통한 영상제작이 아니라, 이미지를 가지고 영상을 구현하는 프로그램을 제작했습니다.
- ◆ CV2를 이용하여 해당 경로의 사진 1405장을 사용하여 영상으로 제작했습니다.



- ◆ 실행시 영상제작 작업 시작합니다.
- ◆ 작업 취소요구시 ESC키를 누르면 작업을 취소 할 수 있습니다.



- ◆ 제작 완료시 해당경로에 영상이 저장됩니다.

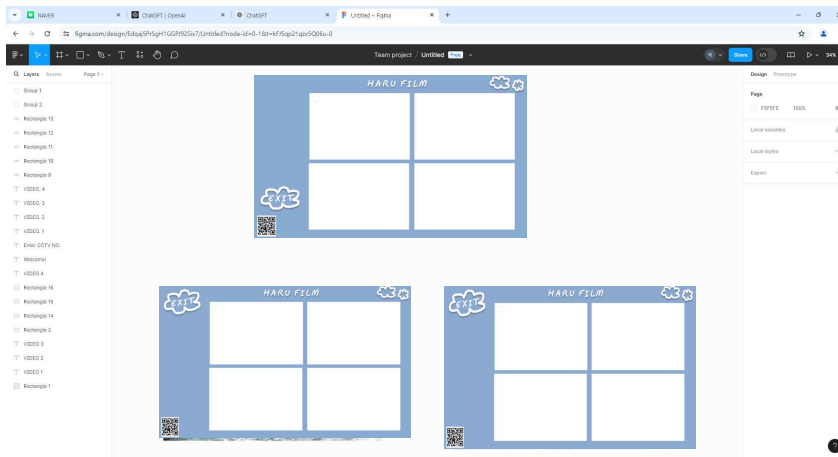


# 개발 포트폴리오

2024.01.31 - 2024.02.01

- ▶ 프로젝트명 : RTSP 영상 호출
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, Figma, SQLiteDB, CV2, Thread
- ▶ 담당 업무 : 화면 제작 및 RTSP 송출 코딩, DB저장
- ▶ 목 적 : RTSP호출을 통한 Thread 코드 이해

## ◆ 화면구성 및 기능 요약설명



- ◆ 피그마를 이용하여 해당 UI의 화면을 제작하였습니다.



- ◆ 실행시 해당 UI가 나옵니다
- ◆ Enter CCTV NO.에 개방된 RTSP 주소를 입력후 저장하면 DB에 저장됩니다.



- ◆ DB에 저장되어있는 CCTVno를 입력후 실행버튼을 누르면 해당RTSP영상이 송출됩니다.

# 개발 포트폴리오

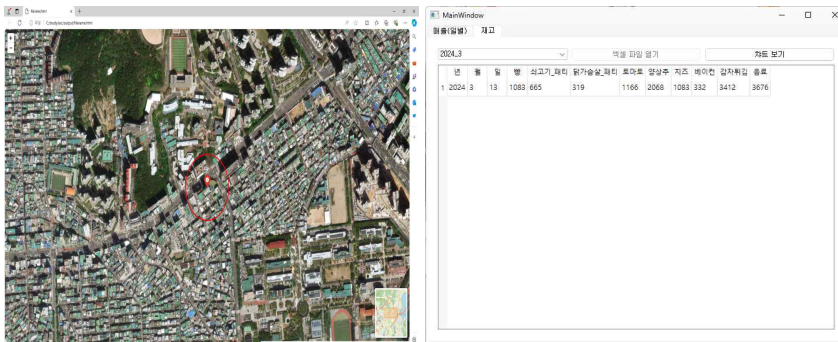
2024.02.27 - 2024.03.13

- ▶ 프로젝트명 : (종합 프로젝트) 소멸 지역 살리기
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, Figma, SQLiteDB, Thread, folium, socket, FTP통신
- ▶ 인원 : 팀프로젝트 (5명)
- ▶ 담당 업무 : 재고 관리 담당 및 코딩
- ▶ 목적 : FTP와 socket통신을 이용해 앞서 배운 내용을 복습

## ◆ 화면구성 및 기능 요약설명



- ◆ 피그마를 이용하여 해당 UI의 배경을 제작하였습니다.
- ◆ 서버 실행시 해당 UI가 실행됩니다.
- ◆ Client와 Server는 서로의 서버를 소켓 통신을 통해 연결합니다. 이를 통해 양방향 통신이 가능하게 됩니다.



- ◆ 해당 지점 사진 클릭시 해당 지점의 위치를 나타내는 지도를 출력합니다.
- ◆ 해당 지점 클릭시 해당 지점의 저장되어있는 매출 및 재고가 출력됩니다.

Menu	Count	Sales
빅맥세트	78	54800
상하리 스프라이츠 최저 가격	312	322000
상하리 스프라이츠 최저 가격 세트	98	43000
1955 버거	220	247200
1955 버거 세트	98	47000
1955 버거 세트	210	248370
빅맥세트	54	40000
빅맥세트	188	182300
빅맥세트	35	54000
빅맥세트	172	178000
빅맥세트	54	27700
빅맥세트	184	182400
음료 단종	1810	300000
음료 단종	1710	328000

재고현황	
빵	1987
식고기 패티	993
달걀송실 패티	669
새우 패티	325
토마토	1172
양상추	1987
치즈	993
배어컨	325
감자튀김	3292
음료	3480

- ◆ 클라이언트 실행시 UI가 실행됩니다.
- ◆ 해당 지점의 물품별 판매 수가 입력되면 일별 매출과 재고 현황이 자동으로 집계되며 DB에 저장됩니다.

2024.03.29. – 2024.04.08

- ## ◆ 화면구성 및 기능 요약설명

◆ 기상청의 날씨 데이터  
(기온, 습도, 강수량, 일사량)의  
데이터를 모읍니다.

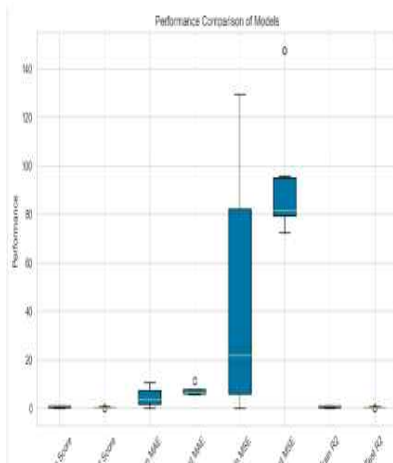
◆ 벚꽃, 매화의 개화일을 예측 할  
모델을 선정 합니다.

◆ 데이터를 학습(train)과 검증(val)으로 나누는 것은 모델의 성능을 평가하고 과적합(overfitting)을 방지하기 위해 train, val 데이터를 나눈 후 훈련을 진행 합니다.

◆ 결과가 나오면 예측을 하기 적합한 모델을 선정 하고 그 모델을 하이퍼 파라메타 튜닝을 통해 개화시기를 예측 합니다.

◆ 훈련한 결과와 얼마나 적합한지를 비교하기 쉽게 matplotlib을 이용하여 한눈에 결과를 쉽게 차트로 나타냅니다.

◆ 한쪽 막대가 더 길다면, 이는 예측 값이 실제 값과 차이가 있음을 나타냅니다. 예측 값이 실제 값보다 크거나 작음을 나타내며, 모델의 성능이 좋지 않음을 시사할 수 있습니다.





# 개발 포트폴리오

2024.04.12 - 2024.04.26

- ▶ 프로젝트명 : 맞춤형 채용 사이트 제작
- ▶ 운영 체제 : Window11 pro
- ▶ 개발 환경 : Python, Django, SQLite, jupyter Notebook
- ▶ 인원 : 팀프로젝트 (5명)
- ▶ 담당 업무 : 웹크롤링을 통한 데이터 수집, footer디자인
- ▶ 목적 : 웹 크롤링 및 Django를 이용한 홈페이지

## ◆ 화면구성 및 기능 요약설명



- ◆ Django를 이용해 채용 사이트 UI 제작

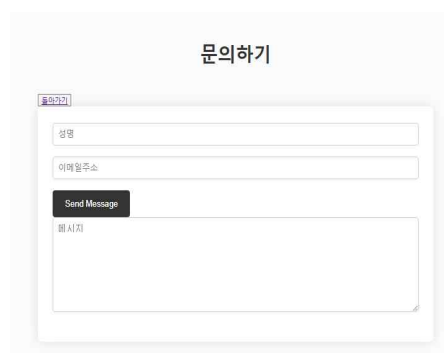


- ◆ 회원가입한 사람들을 대상으로 정보 제공



- ◆ 회원가입 창을 누르면 회원가입 가능

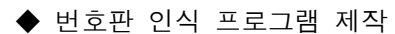
- ◆ 회원가입 정보가 없는 사람들은 로그인 이용 안내



- ◆ 고객센터 버튼을 누르면 고객센터 문의를 남길 수 있다.

2024.05.14 – 2024.06.17

- ## ◆ 화면구성 및 기능 요약설명



- ## 센터 프로그램