

포트폴리오

* 끊임없는 배움과 성장, 진정한 개발자를 꿈꾸는 AI 엔지니어 *



이름 : 박종욱

Email : park363@daum.net

github : <https://github.com/jwpark363>

home : <https://jwpark363.github.io/myhome/>

AI 프로젝트 (최신 순)

순번	프로젝트	설명
1	K-Lingo	언리얼 엔진, LLM, RAG, FastAPI를 이용한 외국인 대상 한국어 교육 솔루션(진행중)
2	WanteDash	LLM, RAG, 구글 drive API 툴(개발), 디스코드API 툴을 이용한 학원 행정 자동화 솔루션
3	근태의전설	Yolo 오브젝트 인식 기술을 이용한 AI기반 지능형 비접촉식 근태관리 시스템
4	K-Food Demon Hunter	CNN 기반 이미지 분류모델과 재료기반 추천 시스템을 이용한 한식 정보 제공 시스템
5	CONE	저품질 기사 검증을 통한 현황 분석을 통한 해결책 제시를 위한 기사 제공 시스템

K-Lingo 프로젝트

언리얼 엔진, LLM, RAG, FastAPI를 이용한 외국인 대상 한국어 교육 솔루션 (현재 진행중)

<https://github.com/jwspark363/klingo-api>

WanteDash 프로젝트

LLM, RAG, 구글 driver API 툴, 디스코드 API 툴을 이용한 학원 행정 자동화 솔루션

- 개요

- RAG, LangGraph, LangChain 기술과 Chat Open API를 활용한 학원 지원 AI 에이전트 시스템
- 많은 학원들이 구글 시트와 디스코드를 활용하여 업무를 진행하고 있고 대부분 수동 업무가 많아 구글 API와 Discord API 기능 툴 개발
- 2025년 10월 13일부터 31일까지 3주간 3명이 진행

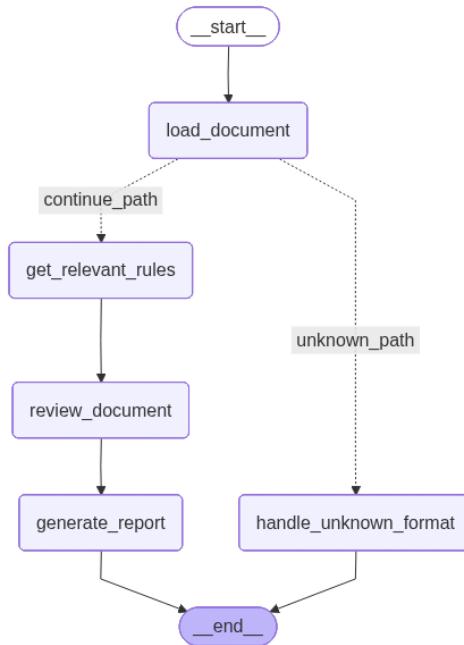
- 역할

- 구글 API 툴 개발, 툴을 이용한 구글 시트 & 드라이버 업무 처리용 에이전트 개발
- 개발된 에이전트를 이용한 백엔드와 에이전트용 프론트 개발
- PostgreSQL과 연동을 위한 SQLChatMessage 처리용 커스텀 클래스 개발
- PostgreSQL 도커 세업, 벡터 스토어 기능 추가

- 구현기술

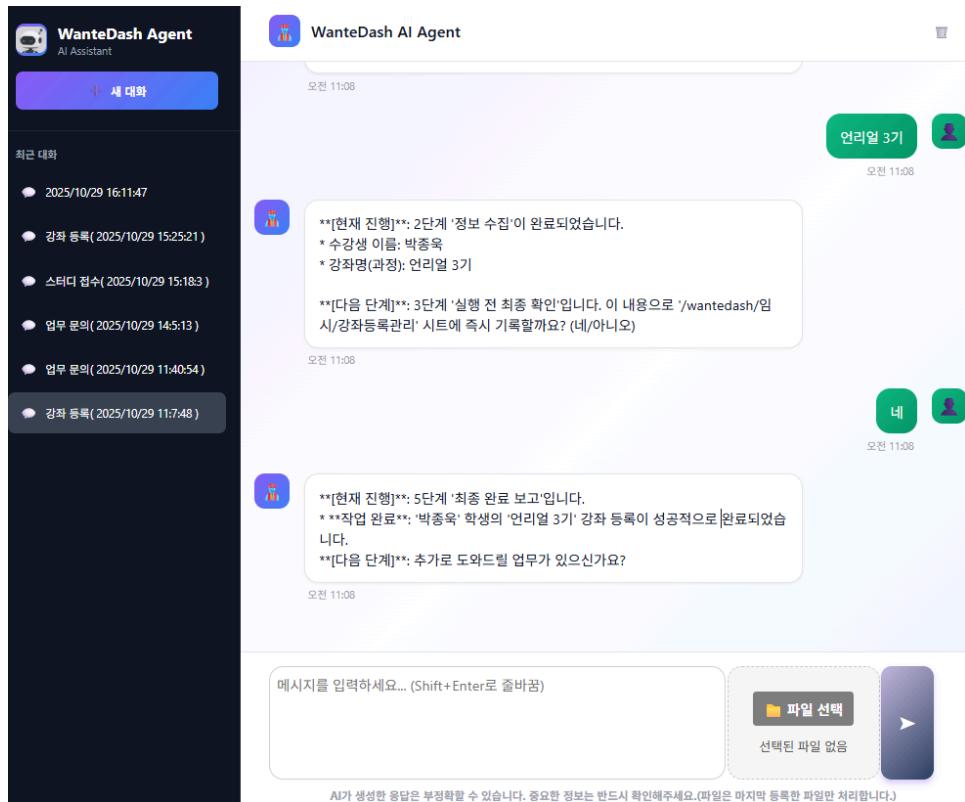


- Google Drive & Sheet API 이용한 Agent 용 툴 개발
- Python LangChain 이용 LLM AI 에이전트 개발
- LangChain의 Message History를 session과 conversation을 기준으로 PostgreSQL에 저장 할 수 있도록 Custom Message 클래스를 개발
- Python LangGraph 이용 LLM AI 문서 검토 에이전트 개발
 - 지정된 양식과 규칙에 맞게 docx 문서를 작성하였는지 검토하는 기능



문서 검토용 에이전트 그래프

- RAG Vector DB 사용을 위한 PostgreSQL 도커 세팅
- HTML + Javascript 이용 클라이언트 구현



에이전트 프론트

- 성과

- LangChain, LangGraph 을 이용한 에이전트 기능 및 서비스 구현
- 에이전트 구현중 DB(PostgreSQL) 연동을 위한 커스텀 클래스 구현
- 구글 Drive API 이용한 툴 개발 및 적용

- 리뷰

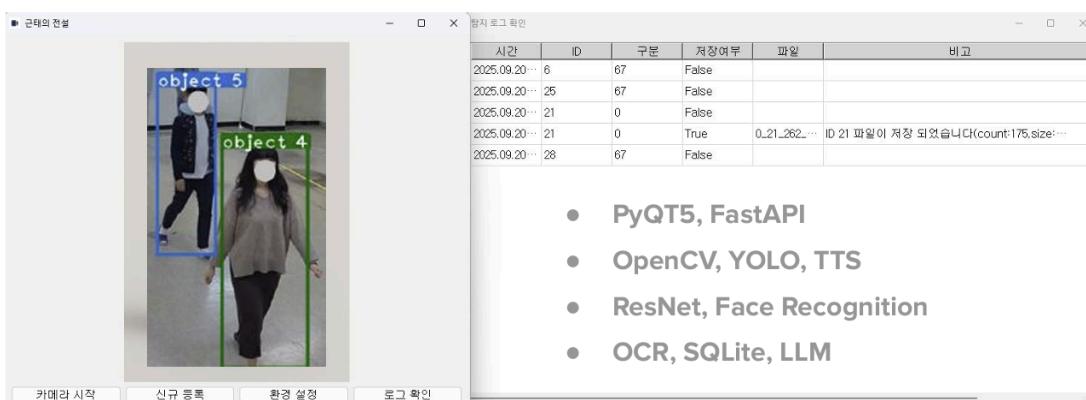
- 구글 Driver API 툴 구현 시 Credential 처리 까지 포함된 툴 개발 진행으로 초기 어려움이 있었음
 - Credential 부분을 툴과 분리 구현으로 극복
- 여러 양식에 적용 가능한 형태 설계를 진행하여 실제 적용 가능한 형태로 구현됨
 - 설정 부분을 구글 시트를 활용한 형태로 개발하여 높은 확장성을 갖출 수 있었음
- 각 Session별 Conversation을 구별하여 저장 할 수 있는 구조로 메시지 클래스 Custom
 - Custom Class를 개발하고 PostgreSQL과 연동함으로써 각 대화의 히스토리 저장과 대화를 연속하여 진행 할 수 있는 형태로 구현됨으로써 완성도 높은 시스템을 구현 하였음

- 향후 자주 사용하는 기능에 대한 템플릿 기능과 어드민용 챗 명령어를 추가한다면 훌륭한 시스템으로 활용 가능하다고 판단됨

근태의전설 프로젝트

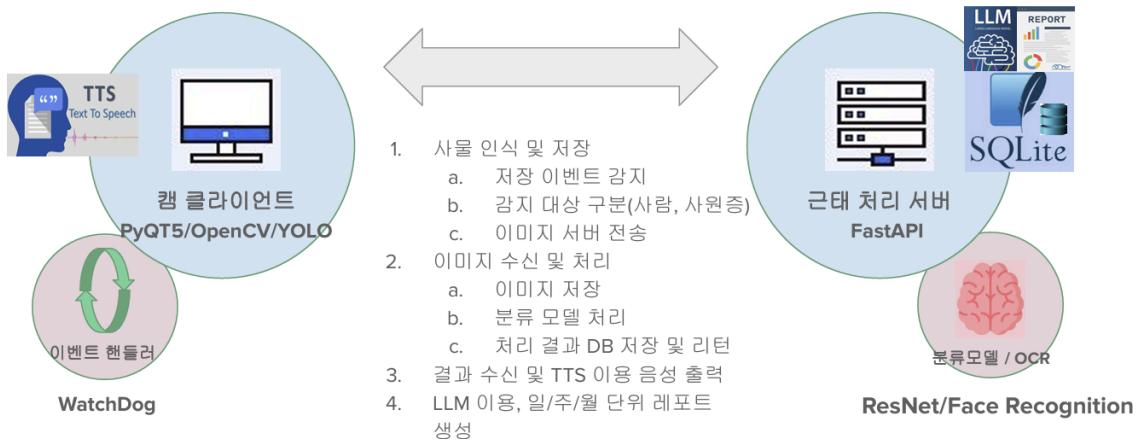
AI기반 지능형 비접촉식 근태관리 시스템

- 개요
 - Yolo 객체 인식 기술을 활용한 근태시스템 개발
 - 객체 인식 기술을 활용한 프로젝트 검토중 근태시스템이 가장 적절하다고 판단되어 진행
 - 2015년 9월 8일부터 24일까지 3주간 2명이 진행
- 역할
 - Yolo와 OpenCV를 이용한 객체 인식 및 추적 기술을 이용한 클라이언트 프로그램 개발
 - WatchDog 이벤트 핸들러를 통해 적정 조건(사이즈와 연속 인식)에 부합하는 인식된 객체(사람, 핸드폰)를 이미지 캡처 하여 저장 후 근태처리용 백엔드 서버로 전송
 - 전송 처리 결과를 TTS로 음성 표출
 - 얼굴 인식 및 분류 모델중 하나인 Face Recognition 모델 테스트 및 적용 코드 작성
- 구현기술
 - PyQT5, OpenCV, Yolo : CS 버전 객체탐지 및 추적용 클라이언트 개발



- WatchDog : 지정된 조건의 객체 인식 처리용 비동기 이미지 핸들러

- gTTS : 구글 TTS 라이브러리, 텍스트를 음성 파일로 변환
- Face Recognition 모델 : dlib 기반 Face 검출, 인식, 비교, 특징 추출 라이브러리



- 성과

- 객체 인식 및 추적 기술 통한 이미지 처리 기능 구현
- CNN 기반 이미지 분류 기능이외 얼굴 인식 모델 테스트

CNN 기반 분류 모델	객체 탐지 기반 분류 모델
ResNet	Face Recognition
많은 양의 이미지 필요	품질이 좋은 이미지 필요
많은 학습 시간 필요	학습 시간 짧음
높은 분류 성능	성능 검증 필요
학습 데이터 기준 결과 리턴	유사도와 거리 기반 결과 리턴
신규 입사자 배치 처리	실시간 신규 입사자 처리 가능

- 다양한 기능(객체 인식, TTS, 이벤트 감지 및 처리, LLM)을 활용한 시스템 구현
- PyQt5이용한 클라이언트 기능 구현
- 리뷰
 - 객체 인식 후 연속된 프레임의 동일 객체 처리 방법 문제 발생
 - 동일 객체에 대한 추적 기술을 활용하여 해당 객체에 부여된 아이디와 연속 인식 프레임 수를 저장하고 특정 사이즈(예 사람 300*300)에 도달 할 경우 해당 객체를 추출하도록 로직 개발 및 적용하여 처리함
 - PyQt5를 이용한 클라이언트 개발 경험 부족으로 화면간 상태관리와 의존성 문제 발생

- 시스템 전반에서 공용으로 사용될 객체와 환경 정보들 그리고 화면간 공유 정보들을 싱글톤 모델을 활용하여 처리 가능하도록 설계 및 개발 진행
- GUI 부분과 객체 인식 및 처리 로직 부분을 분리 설계 및 개발 진행

K-Food Demon Hunter(캐데헌) 프로젝트

Korean Food Image Classification Project

- 개요
 - K-Food Demon Hunters 프로젝트는 K-Food 이미지 분류를 통해 사용자가 K-Food를 쉽게 식별하고 정보를 얻을 수 있도록 돋는 것을 목표하며 이를 통해 K-Food에 대한 인식을 높이고 접근성을 개선하는데 기여 할 수 있을 것을 기대함
 - K-Food 이미지 분류 모델 개발, 레시피를 활용한 추천 모델 개발과 이를 위한 서비스 구축 진행
 - 2025년 8월 13일 부터 9월 1일까지 3주간 3명이 진행
- 역할
 - 레시피 데이터 수집 및 추천 시스템
 - 한식진흥원 레시피 데이터 12*17 건 크롤링 통한 데이터 수집
 - 농촌진흥원 한식 요리 정보(공공데이터포털) 수집
 - 수집한 데이터 전처리
 - 수집한 한식진흥원과 농촌진흥원 데이터와 이미지 학습 데이터 간 코드 맵핑 작업
 - 음식 레시피를 웹에 사용하기 위한 JSON 파일 생성

```

56906_제육볶음.json > ...
{
    "code": "56906",
    "menu": "제육볶음",
    "qty": "4인분",
    "ingredient": [
        {
            "title": "메인",
            "ingredients": [
                "돼지고기(삼겹살) 500g",
                "양파 1/4개(250g)",
                "실팠 9대(90g)",
                "식용유 2큰술"
            ]
        },
        {
            "title": "매운 양념장",
            "ingredients": [
                "고추장 4큰술",
                "고춧가루 2큰술",
                "간장 2큰술",
                "설탕 2큰술",
                "다진 파 2작은술",
                "다진 마늘 1작은술",
                "생강즙 1큰술",
                "청주 2큰술",
                "후춧가루 약간"
            ]
        }
    ],
    "image": "https://www.hansik.or.kr//resources/img/recipe/food112.jpg",
    "recipe": [
        {
            "title": "준비하기",
            "desc": [
                "돼지고기 삼겹살을 0.3cm 두께로 얇게 썬 후 한 입 크기로 썬다.",
                "양파는 길이로 채 썬다.",
                "실팠는 6대는 4cm 길이로 썰고, 나머지 3대는 고명용으로 송송 썬다.",
                "고추장에 나머지 재료를 섞어서 매운 양념장을 만든다.",
                "손질한 돼지고기에 양념장을 냉고 주물러 간이 고루 배도록 30분 동안 재워둔다."
            ]
        }
    ]
}

```

- 음식 레시피의 표준 음식명에 대한 표준 재료명에 대한 동일 단위 기준 재료양을 값으로 하는 테이블 데이터 자료 작성(추천 시스템 벡터 데이터, 사이즈 3093*1311)

```

data_pivot = data.pivot(index='음식_코드',columns='식품_코드',values='식품_중량')
target = data_pivot.fillna(0.0)
target.iloc[:,1295:].head()
✓ 0.0s

```

	식품 코드	F03108	F03109	F03110	F03111	F03112	F03113	F03114	F03115	F03116	F03117	F03118	F03119	F03123	F03338	F03485	F03489
음식 코드	D011001	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
D011002	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
D011003	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	210.0	0.0	0.0	0.0	
D011004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
D011005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

- 코사인 유사도 이용한 추천 시스템 학습 및 모델 생성 저장(서비스 활용)
 - React + Typescript 이용 웹 프론트 개발
 - 음식 이미지 이용 한식 분류용 모델과 추천 시스템 그리고 구성한 한식 JSON 정보를 활용한 프론트 개발
 - Vite + React + Typescript, react_webcam, styled_components 사용
 - gh-page 배포(아래 링크 참조)
 - <https://jwspark363.github.io/kfood/>
- 구현 기술
 - Python pandas, Requests, BeautifulSoup : 데이터 크롤, 전처리 및 웹 표출용 JSON 생성

- Scikit-learn, Joblib : 전처리 데이터 벡터화, 코사인 유사도 학습 및 모델 PKL 저장
- Vite + React + Typescript, react_webcam, styled_component : 웹 프론트 개발
- gh page : 프론트 배포
 - <https://jwpark363.github.io/kfood/>

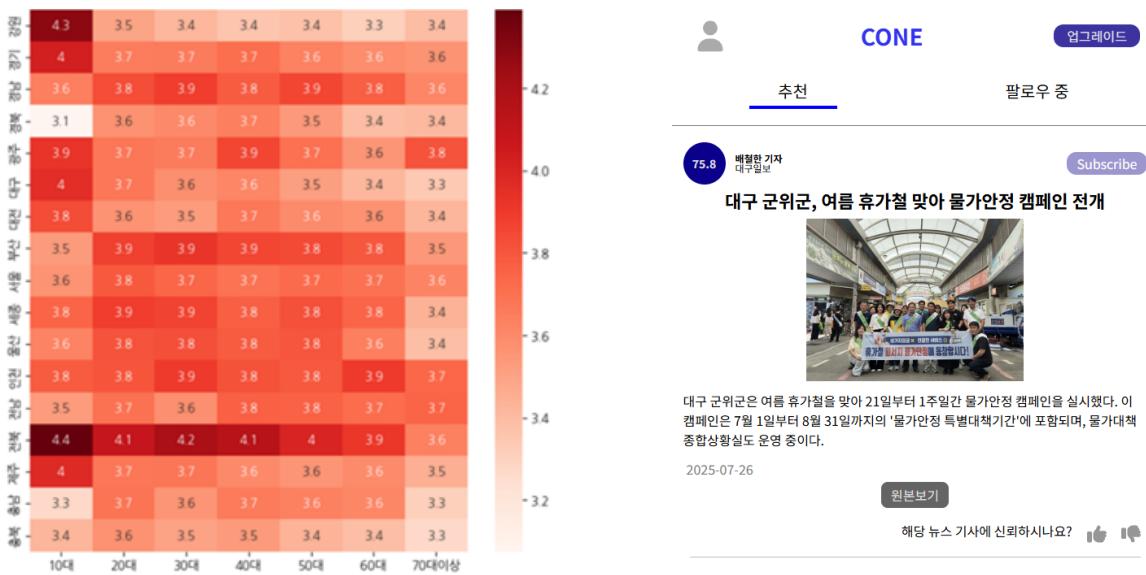


- 성과
 - CNN 분류 모델을 통해 K-Food 이미지 분류 성공률 90% 이상의 성능 달성
 - 레시피 재료 정보를 통한 높은 성능의 추천 시스템 구현
 - 이를 활용한 웹 서비스 개발
- 리뷰
 - 대량의 이미지 데이터로 초기 분류 성능향상에 너무 많은 에너지 소모
 - 회의를 통해 K-Food로 한정하고 대상 모델을 CNN기반 대표 모델인 EfficientNet, ResNet, ConvNext, VGG 모델 결정, 학습 가능 PC 3대 확보하여 1주일간 각 모델의 최종 분류기인 FC 부분을 조정하며 학습 진행 후 최종 모델 선정
 - 분류기, 추천시스템, 레시피 정보 데이터간 코드 불일치 발생
 - 코드 불일치로 분류 후 추천시스템과 웹앱을 통한 정보 표출에 어려움이 예상되어 3개의 코드를 하나의 매핑테이블을 작성하여 최종 시스템에 적용
 - 추천시스템 구현용 데이터 확보의 어려움

- 식당에 대한 평가 데이터는 많았으나 음식에 대한 평가 데이터 확보에 어려움이 있었음
- 1주일간 여러 데이터를 확인하고 검색 및 회의를 통해 최종 레시피 정보를 활용하는 것으로 결정
- 몇 건의 레시피 정보를 확보하여 확인하였으나 재료의 양에 대한 단위 문제 발생
- 최종 농촌진흥원 한식데이터에서 음식별 재료별 동일 기준 재료량 정보 확보

저품질 기사 검증 CONE 프로젝트

- 개요
 - 디지털 언론을 통한 저품질 기사에 대한 현황 분석과 해결을 위한 솔루션 제안
 - 뉴스빅데이터 해커톤을 계기로 시작된 프로젝트로 저품질 기사의 현황을 분석하고 이에 대한 실질적인 해결책을 제시하는 것을 목표로함
 - 2025년 7월 14일부터 8월 1일까지 3주간 3명이 진행함
- 역할
 - 뉴스 데이터 수집 및 분석
 - 빅카인즈 사이트 분석 하여 뉴스 크롤링 프로그램 개발 및 데이터 수집
 - 한국 언론 진흥 재단 언론 수용자 설문 조사 원본 데이터를 바탕으로 지역별 세대별 언론의 신뢰도 및 심각성에 대한 분석 진행
 - 파이썬 판다스와 Dash 라이브러리를 이용하여 시각화 및 앱 구현하고 PythonAnywhere Cloud에 배포 진행
 - React + Typescript 이용 프로토타입 프론트 구현, gh-page 배포
 - <https://jwspark363.github.io/clickbait/news>
 - 프로젝트 발표



• 구현 기술

- Python Pandas, Matplotlib : 데이터 전처리 및 분석 활용
- Vite + React + Typescript : 프론트 개발
- PythonAnywhere Cloud : 무료 클라우드 서버를 제공하고 있어 파이썬 기반 백엔드 개발 프로젝트 초기 단계 배포 및 테스트 용이
- GitHub Page : 정적 웹사이트 무료 호스팅 서비스로 GitHub 리포지토리를 이용하여 처리 가능하여 간단한 개인 포트폴리오, 블로그등에 많이 활용됨

• 성과

- 데이터 분석을 통한 문제점 인식 : 데이터 분석 결과 국내 언론 매체 시장에서 디지털 언론 매체의 비중 증가로 인해 저품질 기사가 양산되고 있으며 이로 인해 전체 언론 시장의 신뢰도가 하락하고 있음을 명확히 파악 할 수 있었음
- 저품질 기사 검증 솔루션 CONE을 기획하고 해당 솔루션의 백엔드 및 프론트 프로토타입을 성공적으로 구현하였음
- 프로젝트의 최초 목적에 맞게 뉴스빅데이터 해커톤에 제출하였음

• 리뷰

- 프로젝트 초기 분석 대상 데이터의 양이 너무 많아 방향 설정에 어려움이 있었음
 - 팀원들과 회의를 통해 목적에 적합한 데이터 위주 전처리를 통해 데이터 양을 줄일 수 있었고 이로 인해 분석을 마무리 할 수 있었음
- 프로토타입으로 개발된 백엔드와 프론트 배포를 위한 서버 선정에 어려움이 있었음

- 다양한 무료 클라우드 서버를 검토하고 테스트 하여 기술 스택과 목적에 맞는 서버 선정을 하였음, 백엔드 PythonAnywhere, 프론트 gh-page