

# 교육사항, 프로젝트 정리

## Tech Stack.

파이썬, 자바스크립트, html, css, sql, restapi, tensorflow, mysql, postgresql, aws, gcp, git, docker, pandas, numpy, scikit-learn, opencv, ML/DL, YOLO

## Education/Training.

- 청년취업사관학교 서대  
문캠퍼스**
- 웹 기초: **HTML/CSS/JavaScript** 학습 및 미니 프로젝트 구현
  - 백엔드 & DB: **Python**, Flask 기반 CRUD 개발, SQLite/MySQL 활용
- Python & 생성형 AI을 활용  
한 풀스택 웹서비스 개발과정**
- 2025.06 - 2025.09
- <https://github.com/therapeuti/sesac618pm>
- 응용 기술: React 입문, 데이터 시각화 라이브러리, 다양한 API 연동 실습
  - 생성형 AI: **OpenAI api**를 활용한 **Langchain** 라이브러리, **RAG** 챗봇 개발
  - 클라우드/배포: **Docker** 및 **aws** 클라우드 환경에서 프로젝트 배포 경험
  - 프로젝트 경험: CRM 시스템, 생성형 AI 프로젝트, 팀 프로젝트 등 실전 중심 개발 경험 내용

- 청년취업사관학교 성북  
캠퍼스**
- 파이썬 활용한 데이터 분석 및 시각화
  - 머신러닝 및 딥러닝 모델링 및 활용
- 서비스 기획을 위한 데이터 기  
반 디지털헬스케어 전문가 양  
성과정**
- 2023.12 ~ 2025.03
- 웹 어플리케이션 구축(RestAPI, Flask, FastAPI)
  - 라즈베리 파이와 가속도 센서를 활용하여 데이터를 수집하고 AI 모델을 활용한 헬스 케어 서비스를 기획하고 개발하는 프로젝트
  - 사용기술 : pandas, numpy, scikit-learn, matplotlib, seaborn, , ML/DL, flask, fastapi

## Team Project.

- 🐾 반려동물의 목소리가 들려**
- 반려동물의 페르소나를 기반으로 한 LLM 대화형 반려동물 관리 서비스
- ▣ 구현한 핵심기능**
- **OpenAI api** 활용한 페르소나 생성 및 대화
  - **RAG** 기반 건강상담 챗봇
- ▣ 기술적 도전 및 문제해결 경험**
- 1 문서 벡터화 및 RAG 문서 검색 품질 문제**
- 청년취업사관학교 서대문캠퍼스 / 프로젝트 3팀
- 2025.08-2025.09**
- 팀 구성 및 역할
  - 3명 [팀장, 풀스택/배포]
- PDF, 웹, 한글 문서 등 다양한 형태로 존재. 컨텍스트 손실 우려가 컸음.  
⇒ 단순 토큰 단위 chunking 대신, **마크다운 변환 후 제목(#, ##) 기준으로 chunking** 하여 의미 단위 보존.
  - 크롤링된 의약품 문서(3만 건 이상)가 vectorDB의 대부분을 차지하고 있어 검색 결과를 왜곡하고 있었고, 문서 메타데이터 미활용, 반려동물 정보 미포함 등으로

- 기술 스택

Python, HTML/CSS/Javascript, AWS, Docker, github, action, OpenAI, Langchain, PostgreSQL, ChromaDB

<https://github.com/MyPetsVoice/my-pets-voice>

로 인해 관련도 낮은 결과가 반환됨.

- 의약품 문서와 일반 문서 컬렉션 분리
- 메타데이터 기반 필터링 및 하이브리드 서치(벡터+키워드 검색) 적용
- 쿼리에 반려동물 프로필 정보 추가

결과: 문맥이 유지된 상태로 vectorDB에 저장되어, 온전한 문서 형태로 검색이 가능해졌으며, 챗봇의 문서 매칭 정확도가 크게 향상되고, 답변 일관성 개선.

## 2 반려동물 특징을 반영한 페르소나 대화 구현 문제

- 단순히 대화 기능을 구현하는 것을 넘어, 실제 사용자의 반려동물이 할 법한 자연스러운 대화를 구현하는 것이 핵심 과제였습니다. 그러나 실제 반려동물을 키우는 사용자가 어떤 대화를 나눌지, 반려동물다운 답변이 무엇인지 기준을 잡기 어려워 효과적인 테스트 방법을 찾는 것이 난관이었습니다.
- 반려동물과의 일상을 다룬 웹툰을 테스트 데이터로 활용하는 방법을 고안했습니다. 웹툰 속 반려동물의 특징을 프롬프트로 구체화하고, 실제 에피소드 상황을 재현하여 웹툰 속 반응과 일치하는지를 평가 기준으로 삼았습니다. 프롬프트 구성, 모델 선택, 하이퍼파라미터를 체계적으로 조정하며 반복 테스트를 진행했습니다.
- 프롬프트에 반려동물의 상세한 특징과 특정 상황에서의 행동 패턴을 구체적으로 기술할수록 페르소나 재현도가 높아졌으며, GPT-4o 모델과 temperature 값 1.0 설정이 가장 자연스러운 대화를 생성한다는 것을 발견했습니다. 이를 통해 LLM의 창의성과 일관성 간의 균형을 찾는 방법을 체득했습니다.

## 🚀 인사이트 및 성과

- AI 기술 심화 : OpenAI API, LangChain, RAG 등 최신 LLM 기술 실습 및 최적화 경험
- 실무 감각 강화 : CI/CD, 배포, 코드 컨벤션 및 문서화를 통한 프로덕션 수준 개발 경험
- 리더십 경험 : 기술적 의사결정 및 팀 내 협업·이슈 관리 주도



MAJU(마주)



구현한 핵심 기능

저소득·한부모 가정 학생  
도 질 높은 교육과 진로 멘  
토링을 받을 수 있는 교육  
매칭 플랫폼

- 자연어 기반 AI 매칭 알고리즘
- AI 레벨테스트 문제 생성 및 분석



기술적 도전 및 해결 과정

2025 우먼테크 해커톤 /

Wosha-Itsha

2025.07 ~ 2025.08

- 팀 구성 및 역할

- 5명(기획/디자인 1, 프론트 1, 백엔드 2,

AI 1)

- AI 백엔드 구축

## 1 AI 레벨테스트 문제 형식 오류

- chatGPT를 활용하여 사용자 정보를 토대로 레벨테스트 문제를 만들도록 하였으나, 지문 항목이 텅 비어있고, 지문에 들어가야 할 내용이 문제 항목에 연달아 나오는 등 지정한 문제 형식대로 만들지 않는 경우가 있었음.
- 그 시기에 GPT-5가 출시가 되었음. GPT-5 기능 중 CFG 형식으로 답변 형식을 강제할 수 있었음. CFG를 활용하여 레벨테스트 문제를 지정한 형식대로 출력하도록 하여 문제를 해결하였음.

## 2 임베딩 모델 비용 문제

- 기술 스택
  - `Flask`, `Langchain`,
  - `OpenAI`, `ClaudeCode`
- 사용자가 추가될 때마다 매칭 알고리즘을 실행시켜야 하는데, 매번 임베딩을 하면 시간이 오래 걸리고 임베딩 모델 비용이 증가한다는 문제가 있었음.
- 처음 임베딩 시 임베딩 정보를 캐시에 저장하도록 하여, 매번 임베딩을 하지 않고 캐시 정보를 사용하도록 함.

<https://github.com/MAJU-MENTORING>



## 인사이트 및 성과

[https://github.com/MAJU-MENTORING/backend\\_python](https://github.com/MAJU-MENTORING/backend_python)

- Claude Code를 처음으로 사용하여 바이브 코딩으로 프로젝트를 진행함으로써 기존 역량을 뛰어넘는 결과물을 만들어낼 수 있었음.

## AI 회전초밥 자동 계산 시스템

회전초밥 접시를 촬영하면  
자동 인식하여 결제금액을  
계산하는 시스템

모바일 앱 개발 협동조합 / AI팀

2024.09 - 2024.10

- 팀 구성 및 역할

4명 [PM, AI/백엔드/배포]

- 기술 스택

`Python`, `YOLO`, `Fastapi`, `GCP`

## 구현한 핵심기능

- 다양한 YOLO 모델 파인튜닝
- UX 중심의 프론트 설계 : 사용자의 결과 검토/수정이 빠르게 진행되도록 UI 구현
- FastAPI + WebSocket 기반 서버 구축으로 추론 지연 최소화
- GCP Cloud Build를 이용한 CI/CD 파이프라인 구성

## 기술적 도전 및 해결 과정

### 1 실시간 추론 지연 및 정확도 저하 문제

- flutter 앱 환경에서 YOLO 추론 시 지연이 길고 정확도가 낮은 문제 발생.
- 학습데이터 증가, 파라미터 튜닝, 모델별(v5, v8s, v8n 등) 비교 테스트, 앱 내장 방식 및 서버 연동 방식 비교하여 최적의 방식 탐색.
- FastAPI + WebSocket 기반 서버를 구성하여, 비동기 실시간 추론으로 스트리밍 시 딜레이 최소화

[https://github.com/therapeuticai/autocalculate\\_yolo](https://github.com/therapeuticai/autocalculate_yolo)

AI

💡 결과: 실시간 추론 속도 개선(1초 내), 인식 정확도 90% 이상 달성.

### 2 빠른 결과 검증을 위한 UX 설계

- 결제와 직접 연결된 기능이므로 AI 추론 결과의 검증이 중요.
- 사용자가 결과를 한눈에 검토하고 수정할 수 있도록 UI 설계:
  - 각 접시 옆에 추론 라벨을 표시하고, 폰트 색상을 접시 색상과 일치시켜 한 눈에 결과를 판별할 수 있도록 함.
  - 한두 번의 터치로 수정 가능한 인터페이스 설계

💡 결과: 추론 오류가 발생하여 사용자가 결과를 수정하더라도 최장 10초이내 결제 금액 도출이 가능해짐.

## 인사이트 및 성과

- 프론트엔드 기술(HTML, CSS, JS)를 처음으로 학습하며 AI를 넘어 풀스택 개발로 확장의 계기가 됨
- 프로젝트 관리 역량 강화 : 기술 선택, 일정 조율, 품질 관리 전반 리딩
- 실제 고객과의 커뮤니케이션을 통해 AI 프로덕트 개발 프로세스 전반을 경험함.
  - 요구사항을 직접 듣고 서비스 기획 및 문제 해결을 위한 기술 선택

- 피드백을 반영하여 UI/UX 및 추론 결과 검증 프로세스 개선

## 배움의 모든 것

### 구현한 핵심기능

콘텐츠 기반 필터링 및 자연어 기반 취미 추천 알고리즘 개발

- 자연어 처리 기반 추천 알고리즘 구현

- 자연어로 된 사용자 프로필과 취미 활동 description을 형태소 분석 → TF-IDF 벡터화 → 코사인 유사도 계산

모바일 앱개발 협동조합 / AI팀

2024.07 - 2024.08

- Flask 서버 구축 : 실제 웹서비스에 적용 가능한 구조로 구현

- 알고리즘 설계 전 과정을 100장 분량의 기술 보고서로 문서화

- 팀 구성 및 역할

2명 [PM, ML/백엔드]

### 기술적 도전 및 해결 과정

- 기술 스택

Python, kiwipiepy,

Scikit-learn, Flask

- ML/DL 모델을 적용할 만한 실제 사용자 데이터가 존재하지 않음.

- 문제 해결을 위해 다양한 사용자 프로필의 더미데이터를 직접 생성.

- 성별, 연령대, 선호 취미, 활동 강도, 학습 목적 등을 포함한 사용자 프로필 설계

- 각 취미의 설명(description)에 카테고리, 활동 빈도, 관련 키워드 등을 반영

- 협업 필터링 대신, 데이터 부재 상황에서도 가능한 콘텐츠 기반 필터링 선택.

- 취미 설명(description)과 사용자 프로필을 형태소 분석 후 TF-IDF 벡터화 → 코사인 유사도 계산으로 취미를 추천.

- 추천 결과를 시각화하여 비교·검증하고, 키워드 중심의 개선 반복.

 결과: 사용자 성향과 취미 설명 간 유사도 기반의 맞춤형 추천 구현.

### 인사이트 및 성과

- 자연어 처리 기반 추천 시스템 설계 경험. 이후 또 다른 매칭 알고리즘 설계(MAJU 프로젝트)의 기반이 됨.

- 데이터 생성 및 전처리 역량 강화

- Flask 서버 개발 및 배포 구조 설계 경험

- 문서화 및 기술보고 역량 향상 (100p 기술 문서 작성)

- 협업 경험 : 자바, 프론트엔드 개발자와의 협업으로 API 연동 및 테스트 진행



## AI 대화친구 앱초

### 구현한 핵심 기능

아동 대상 LLM 기반 음성 대화 서비스

- 대화 콘텐츠 기획 및 시나리오 설계 (퀴즈놀이, 끝말잇기 등)

- 아동 맞춤형 대화 데이터셋 생성 및 전처리

모바일 앱개발 협동조합 / AI팀

2024.05 - 2024.07

- LLM 파인튜닝 및 하이퍼파라미터 튜닝 (temperature, top\_p 등)

### 기술적 도전 및 해결 과정

- 팀 구성 및 역할

## 5명 [PM, LLM 파인튜닝, 서비스 운영]

### 기술 스택

네이버클라우드, 하이퍼클로  
비X, GCP, gemini

### AI 대화친구 서비스 소개 영상

#### 1 LLM 할루시네이션 문제

- 모델이 엉뚱한 응답을 생성하거나 퀴즈 정답을 잘못 판별하는 문제가 발생.
- 프롬프트 엔지니어링**을 통해 페르소나를 명확히 정의하고, 각 놀이별로 **프롬프트 템플릿을 세분화**.
- Few-shot 프롬프팅 + COT (Chain of Thought)** 기법으로 정답 판별 정확도 향상.

결과: LLM 응답 일관성 향상, 잘못된 정답 비율 약 60% 감소.

#### 2 파인튜닝 및 하이퍼파라미터 튜닝

- 단순 프롬프트로 해결되지 않는 “대화 스타일” 문제 해결을 위해 아동 콘텐츠(유튜브, 교육 서적 등)를 참고해 **유아 대화체 데이터셋 직접 제작**.
- 파인튜닝을 통해 **대화 톤의 일관성 확보**, temperature, top\_p 조정으로 다양성과 정확도 균형 달성.

결과: 자연스러운 아동 대화 톤 구현 및 사용자 몰입도 향상.

#### 3 정답 판별 로직 추가

- LLM의 오판을 완전히 방지하기 위해 **Python 규칙 기반 정답 판별 로직을 별도로 구현**.
- 사용자 입력이 LLM에 전달되기 전 검증하도록 파이프라인을 수정하여, 모델 출력의 신뢰도를 높임.

결과: 퀴즈 모드 정답 판별 정확도 대폭 향상, 할루시네이션 근본적으로 감소.

## 🚀 인사이트 및 성과

- 직접 기획하고, 세상에 출시한 **첫 LLM 서비스**
- 프롬프트 엔지니어링과 할루시네이션 제어 역량 강화**
- 서비스 기획 및 대화 콘텐츠 설계 경험**
- 프로젝트 리딩 및 팀 운영 경험**

## .drawImage("image.png") 이미지와 IMU 센서를 활용한 양치질 패턴 식별

CNN, GRU 모델을 활용하여 사용자의 양치질 패턴 분류하는 IoT 서비스

## ⚙️ 구현한 핵심 기능

- 양치질 이미지 전처리(hand-detection, face-detection 모델, opencv 사용)
- 양치질 이미지 분류를 위한 CNN 모델링
- flask 웹 어플리케이션 구축

## 🧠 기술적 도전 및 해결 과정

### 청년취업사관학교 성북캠퍼스

2024.02 ~ 2024.03

#### 팀 구성 및 역할

5명, [팀장, CNN 모델링, flask 백엔드 구축]

#### 1 양치질 이미지 분류 문제

- 단순히 양치질 이미지만으로는 배경이나, 칫솔 색상 등 노이즈가 많아 양치질 부위를 분류하는데 어려움이 있었음.
- 양치질 부위는 얼굴과 손의 위치 관계와 관련이 깊으므로 분류 시 해당 컨텍스트를 제공하고자 했음. 구글 Mediapipe의 hand-detection 모델과 face-detection 모델을 먼저 적용하여 얼굴과 손의 위치를 파악하도록 한 후 학습시킴.

- 기술 스택
  - Val accuracy : 0.8~0.89 / Val loss : 0.45 ~0.33 정도로 정확도는 증가하고 손 실값은 감소했음.
  - Python, Flask, OpenCV,
  - Mediapipe, Tensorflow,
  - RaspberryPie, IMU

[https://docs.google.com/presentation/d/1\\_hIHc5kXbBYs\\_Gzr9ur0Qqa3ciPHZmjU7OO8WWgW3ns/edit?usp=sharing](https://docs.google.com/presentation/d/1_hIHc5kXbBYs_Gzr9ur0Qqa3ciPHZmjU7OO8WWgW3ns/edit?usp=sharing)

---

## Mini Project.

### 테이블 오더 시스템(바이 이브코딩)



#### 주요 기능

- 주문 메뉴 등록 시 실시간 주문 현황 반영
- QR코드 기반으로 식당 주문 현황 관리 시스템
- 고객이 QR코드로 음식 주문

#### 기술 스택

Python, Django,  
Docker, Github action

<https://github.com/therapeuti/tableorder>

---

### 모동숲 섬 꾸미기(바이 브코딩)



#### 주요 기능

- 닌텐도 게임 “모여봐요 동물의 숲” 섬 지도 스크린샷을 배경으로 섬 꾸미기 계획을 세울 수 있는 웹 어플리케이션
- 섬 지도 스크린샷을 격자무늬에 맞춰 업로드
  - 섬 지도 모양을 배경으로 하는 캔버스 위에 도로, 절벽, 강물 등을 칠할 수 있음.
  - 다양한 모양, 사이즈, 색상의 페인트 도구
  - 건물, 나무, 꽃 등의 오브젝트를 배치할 수 있음.
  - PC/아이패드용 반응형 디자인

#### 기술 스택

React, Konva

<https://github.com/2in1-team39/team39-aws-hackathon>