



Wrap UP 리포트_StockSense

프로젝트 개요

문제 정의 및 해결 과제

최근 유통업계의 급격한 변화 속에서 대형마트 및 소매점의 점포 감소, 무인점포의 경제적 어려움, 자영업자의 지속적인 폐업 증가 등이 주요 이슈로 부각되고 있다.

[2024.05.20 메트로신문 '소비 트렌드 변했다' 대형마트 점포 7년 새 37개 감소..."본업 집중 효율성 강화"]

[2024.02.18 한국경제 "월세가 160만원인데 매출이 60만원"...'무인점포'의 눈물]

[2024.02.18 서울경제 "1년도 못 버티고 폐업...코로나때보다 힘들어" 깊어지는 자영업자 한숨]

따라서, 재고 부족이나 과잉 재고로 인한 손실이 발생하여 비효율적인 재고관리가 이루어지고 매장 관리자가 데이터를 수동으로 분석하고 의사결정을 내려야 하는 낮은 효율성 구조 문제, 소비 트렌드 및 리콜 상품의 정보를 놓쳐 잠재 매출 감소 및 운영 비용 증가와 같은 문제가 발생한다.

해결 방안

- AI 기반 실시간 재고 관리 시스템 도입
 - 판매량 데이터를 머신러닝을 통해 판매량을 예측하여 재고 부족 시 관리자에게 즉각 알림 제공.
- 통합 대시보드 제공
 - 재고 상태, 판매 데이터를 통합하여 시각화된 대시보드로 제공.
 - 관리자들이 데이터를 직관적으로 파악하고 빠르게 의사결정을 내릴 수 있도록 지원.
- 맞춤형 매장 관리 솔루션 제공
 - 최신 트렌드 상품 추천 및 리콜 상품 방지 솔루션 제공.

⇒ 트렌드 & 리콜 상품을 파악해 자동으로 최적의 재고를 관리

프로젝트 팀 구성 및 역할

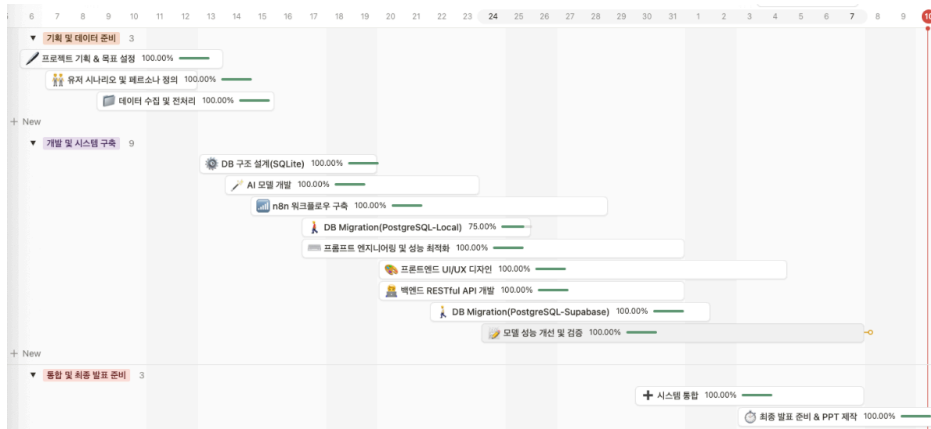
- 팀구성

김태한	- Webshop 가상 환경 세팅 - 주문 Agent(ReAct + Reflexion) 개발 - n8n 기반 Workflow 자동화
문채원	- n8n 기반 Workflow 자동화 - 웹 프론트엔드(UI/UX) 개발 - 트렌드 상품 OCR 및 프롬프트 엔지니어링
서동환	- 프론트엔드(UI/UX) 구현 - 챗봇 API 연결 및 프롬프트 튜닝 - 기간 별 매출 및 인기 상품 데이터 조회백엔드 엔드포인트 구현
윤남규	- DB(PostgreSQL & Supabase) setup, migration - n8n recall workflow 구현

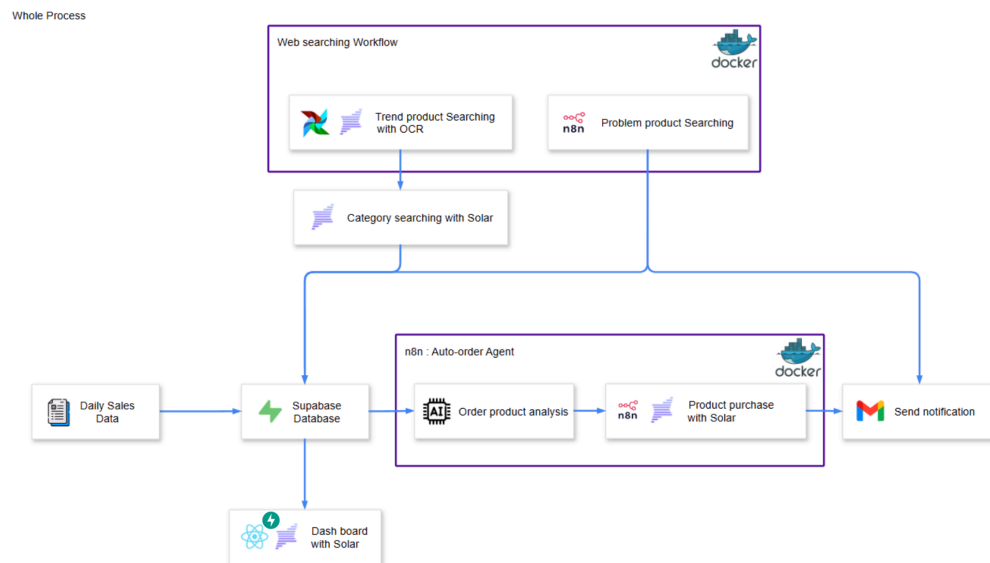
	- Airflow 기반의 Category Searching 구현
이재훈	- Dashboard 백엔드 및 프론트엔드 구현 - 판매량 예측 모델 개발
장지우	- Inventory 백엔드 및 프론트엔드 구현 - 자동 주문 & Dashboard 실시간 통합 아키텍처 설계

프로젝트 수행 절차 및 방법

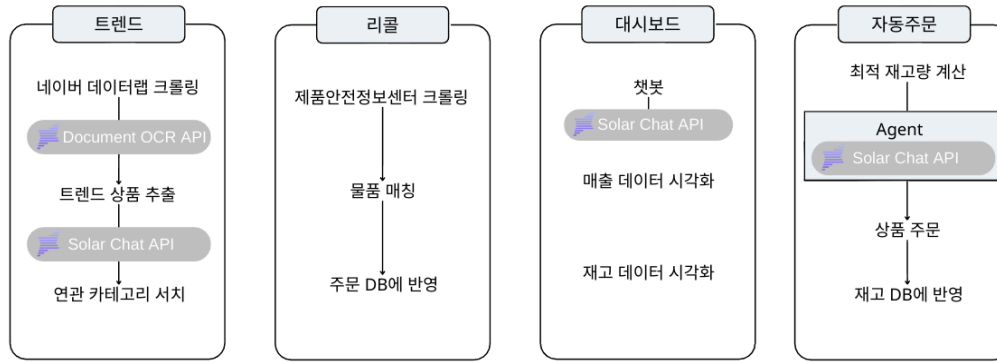
프로젝트 타임라인



프로젝트 워크플로우



주요 기능

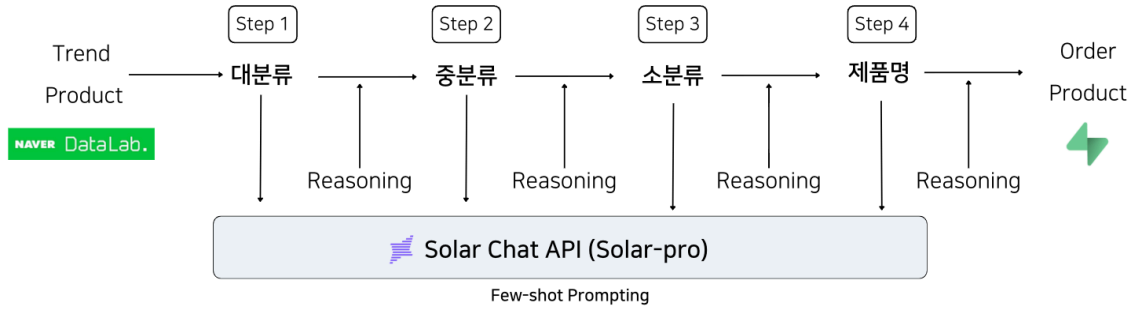


- DB Repository : Supabase & PostgreSQL

- PostgreSQL을 활용해 네트워크 원격 접속 및 다중 사용자가 동시에 데이터베이스에 접근하고 수정이 가능하도록 관리
- Huge traffic handling 을 위해서 auto-scaling & CDE caching 을 활용
- 실제 Amazon.com 에서 사용하는 9500개의 taxonomy 를 활용해 서비스 완성도를 높임

- Trend Searching Workflow

- Naver Datalab 에서 실시간 트렌드 상품을 불러오고, 마케팅으로 활용하는 Agent
- Airflow DAG 를 통해서 일정 주기마다 불러온 트렌드 상품들이 실제 판매 중인지 확인.
- 실제 판매 중인지 확인하는 기능은 Few-shot(solar), Hierarchy, Ensemble 를 활용해 구현했으며, 실제 판매 중인 연관 상품을 추천 및 자동 주문.
- 또한 관련 정보를 요약해 gmail 로 전송 및 Chatbot 에서 추가 정보 제공
- 이를 통해서 개인 사업자가 마케팅 정보를 바로 활용하기 어렵던 문제를 해결하고자 기여함.
- Category Search
 - 계층 구조 기반의 카테고리 탐색
 - Hierarchy (main > sub1 > sub2 > sub3) 기반으로 작동하는 parent & child 구조 → 검색 범위 제한
 - Parent category 탐색 결과 및 reasoning 과정을 child category 에서 Context로 추가 활용
ex. If input_text = Potato chip and main = food , then add more context(ex. Potato chip is food) when searching sub1 .
- Few Shot 기반의 카테고리 분류
 - Reasoning 과정에서 각 카테고리의 레벨별 예제를 구축하고 이를 Prompt로 활용
 - 이를 통해서 더 많은 Context를 사용함으로써 Category Searching Algorithm 를 개선함
 - Child category 는 valid_categoriy(child category) 해당하는 내용

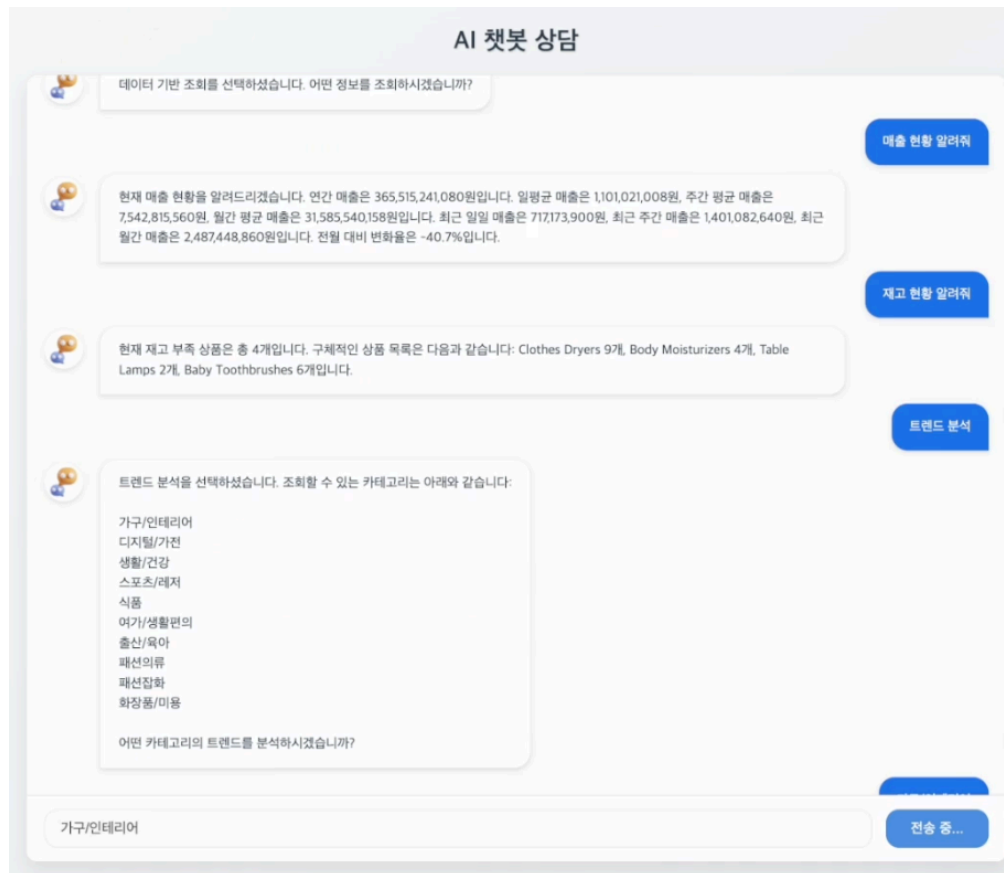


• 리콜 상품 파악 Workflow

- [safetykorea\(제품안전정보센터\)](#)의 공시 정보를 통해서 리콜 상품 파악
- 만약 리콜 상품이 판매중이라면, 사업자에게 알림 제공 및 자동으로 재고에서 제거함으로써 사후 문제를 예방함.

• 대시보드

1. 챗봇 페이지



- 챗봇을 이용해 트렌드 분석 재고 분석 답변 받을 수 있다.

2. 대시보드 페이지



- 일, 주, 월, 연간 매출등이 보이고 많은 매출이 내는 제품과 트렌드 상품등을 보여준다.

3. 인벤토리 페이지

재고 관리

시작 일: 2022년 01월01일 종료 일: 2022년 04월01일

상품명 검색

다분류: 상품유: 소분류:

☐ 재고 내림차순 ☐ 품 평균 판매량 오름차순 ☐ 품 평균 판매량 오름차순 ☐ 재고고려

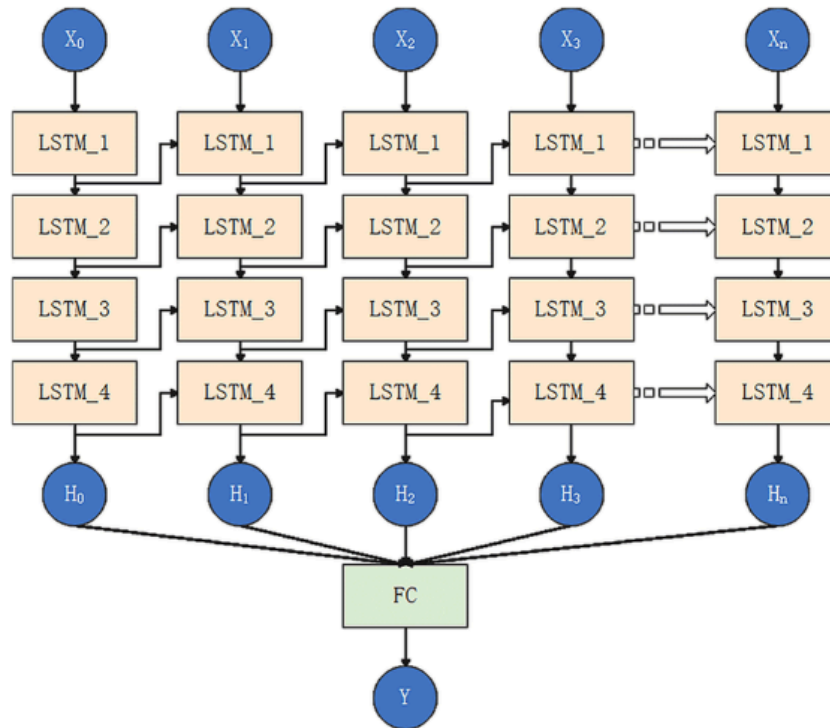
제품명	일 평균 판매량	일 평균 판매량	재고량	최소 재고 기준	상태	주문
Body Moisturizers	1.9	0.1	4	10	▲ 재고 부족	×
Baby Toothbrushes	13.8	0.5	6	10	▲ 재고 부족	×
Table Lamps	55.3	1.8	8	20	▲ 재고 부족	×
Baby Stroller Inner Tubes & Tires	261.4	8.7	122	14	■	-
Medical Scalpel Blades	1.6	0.1	128	10	■	-
Children's Cartoon Humor Books	114.5	3.8	133	12	■	-
Law Enforcement Biographies	56.8	1.9	137	10	■	-
Russian Literature	22.3	0.7	140	10	■	-
Endocrine System Diseases	294.9	9.8	145	12	■	-
Fruit Gardening	7.4	0.2	158	10	■	-
Sega Game Gear Games	0.6	0.0	160	10	■	-
Children's Reptile & Amphibian Books	346.4	11.5	179	38	■	-
Commercial Access Cards	22.9	0.8	183	10	■	-
Women's Pants	74.5	2.5	193	10	■	-
Cargo Liners	1540.5	51.4	198	192	■	-
Breadcrumbs & Seasoned Coatings	201.2	6.7	204	12	■	-
Scuba Diving	30.8	1.0	205	10	■	-
Ranchera	1039.9	34.7	210	92	■	-
Steam Cleaner & Steam Mop Accessories	13.5	0.5	218	10	■	-
Cat Grooming Clippers & Blades	17.1	0.6	222	10	■	-
Desk Supplies Holders & Dispensers	2.0	0.1	228	10	■	-
Dating	325.3	10.8	237	66	■	-
Hair Conditioner	78.8	2.6	239	12	■	-

- 현재 상품의 재고 현황을 보여준다.

• 자동 주문 Agent

- Stacked LSTM 기반 판매량 예측

- 시계열 데이터 분석
 - 모델 : Stacked LSTM



Stacked LSTM Architecture

- 데이터셋 : Amazon Categories + 온라인 채널 제품 판매량 예측 AI 온라인 해커톤 학습 데이터셋
- 성능 : PSFA 기준 약 0.55

모델	PSFA Score	특징
LSTM	0.48	구조는 단순하여 우리의 데이터셋에 성능 한계를 확인
GRU	0.50	계산 비용이 낮아 inference 속도가 빠르지만 성능이 다소 낮음
Stacked LSTM	0.55	다층 구조로 단기와 장기 의존성을 잘 학습한다고 판단 → 최종 선택

- ReOrder Point
 - 최소 재고 주문 수량 $ROP = (\text{평균 일일 수요} \times \text{리드 타임}) + \text{안전재고}$
 - 평균 일일 수요: LSTM으로 예측한 판매량 사용
 - 총, 재고 부족 상품 $ROP + \text{트렌드 상품}$ 을 주문 에이전트에 전달
- Order_Agent
 - Agent 구조
 - Environment : Webshop
 - Model : solar-pro(Chat API)
 - Reasoning/Planning : ReAct + one-shot prompting

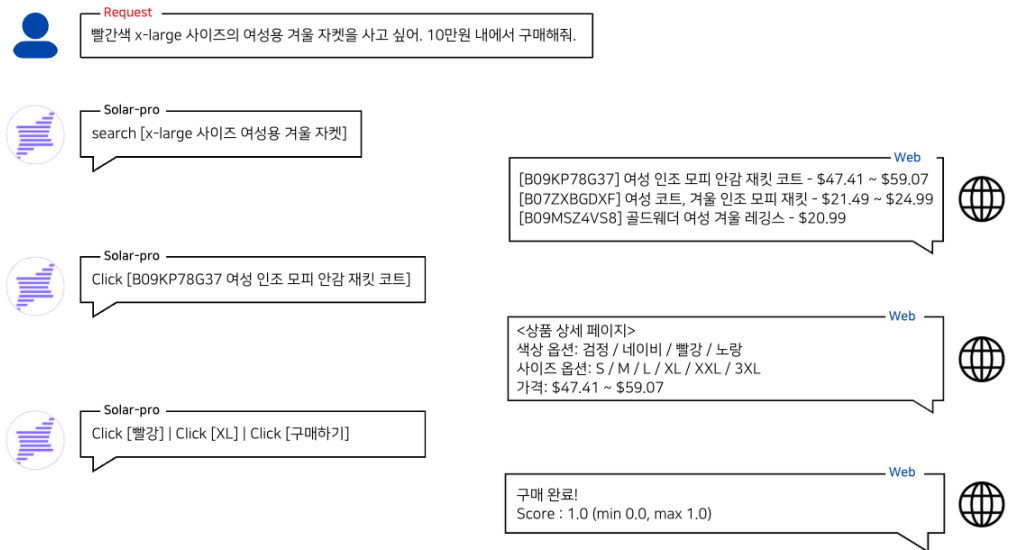
- Memory : Reflexion

■ Webshop

- 웹 기반 시뮬레이션 이커머스 환경 및 에이전트의 이커머스 문제 해결력 평가를 위한 벤치마크
- [amazon.com](https://www.amazon.com)에서 스크랩된 1,000개의 상품 활용
- 12,087개의 instructions
- Webshop을 활용하여 Observation을 얻는 방법
 - Action → url 형태로 수정 → request GET을 통해서 해당 html을 읽어옴 → BeautifulSoup 라이브러리를 활용해서 html 파싱을 진행 → 필요한 부분(Observation)만 text 추출

■ ReAct

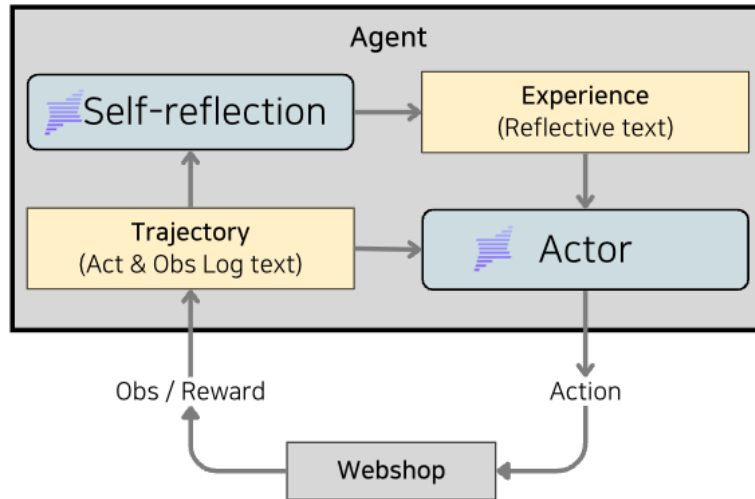
- Reasoning + Acting 결합하여 사고(think)와 행동(search, click)을 적절히 수행
- 진행 과정
 1. 입력 : one-shot 예제 + Instruction
 - a. 출력 : LLM모델이 가장 적합한 Action을 도출
 2. 입력 : 1번 입력 + 1번 출력 + 1번 Action으로 나온 Observation
 - a. 출력 : LLM모델이 가장 적합한 Action을 도출



◦ Reflexion

- AI의 자기 피드백(Self-Reflection) 기법
- 문제와 이전에 생성한 답, 정답 여부(실패)를 주고 왜 실패했는지, 어떻게 하면 다음 시도에 성공할 수 있는 지를 묻고 그 결과를 **memory**에 저장하는 방식
- 위 ReAct로 Webshop Task를 풀다가 **실패한 경우** few-shot과 함께 자기 피드백을 생성하게 함

Reflexion



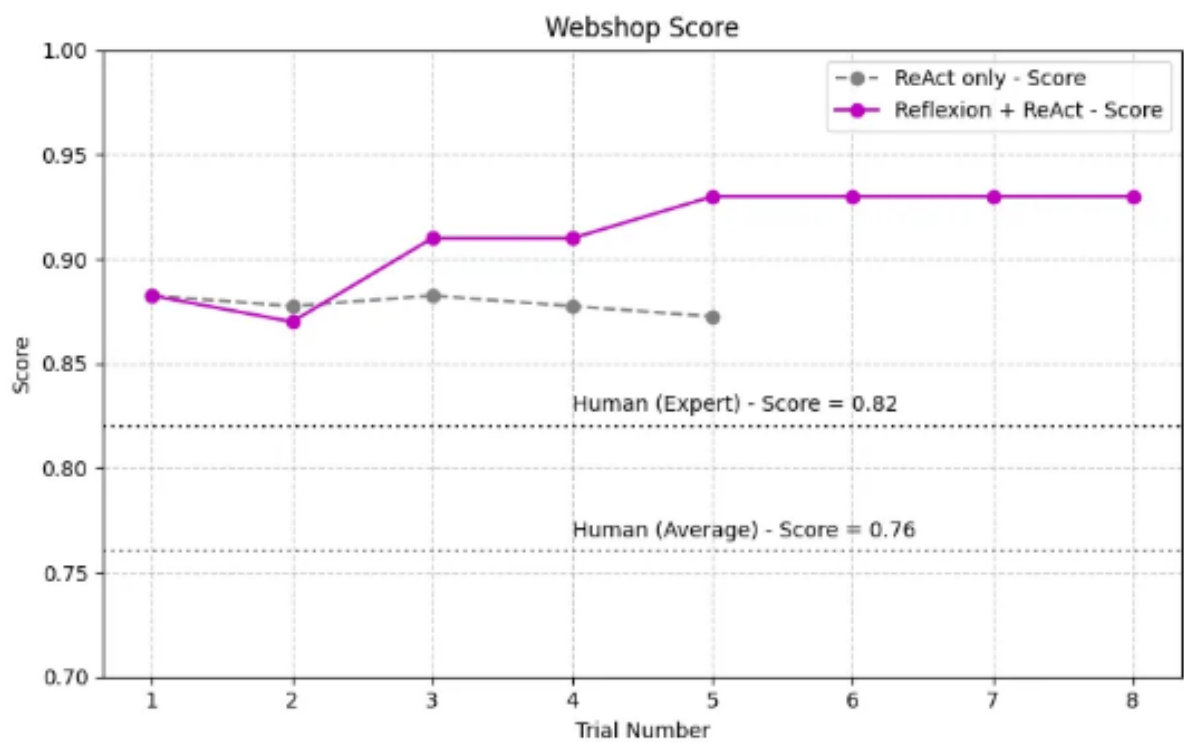
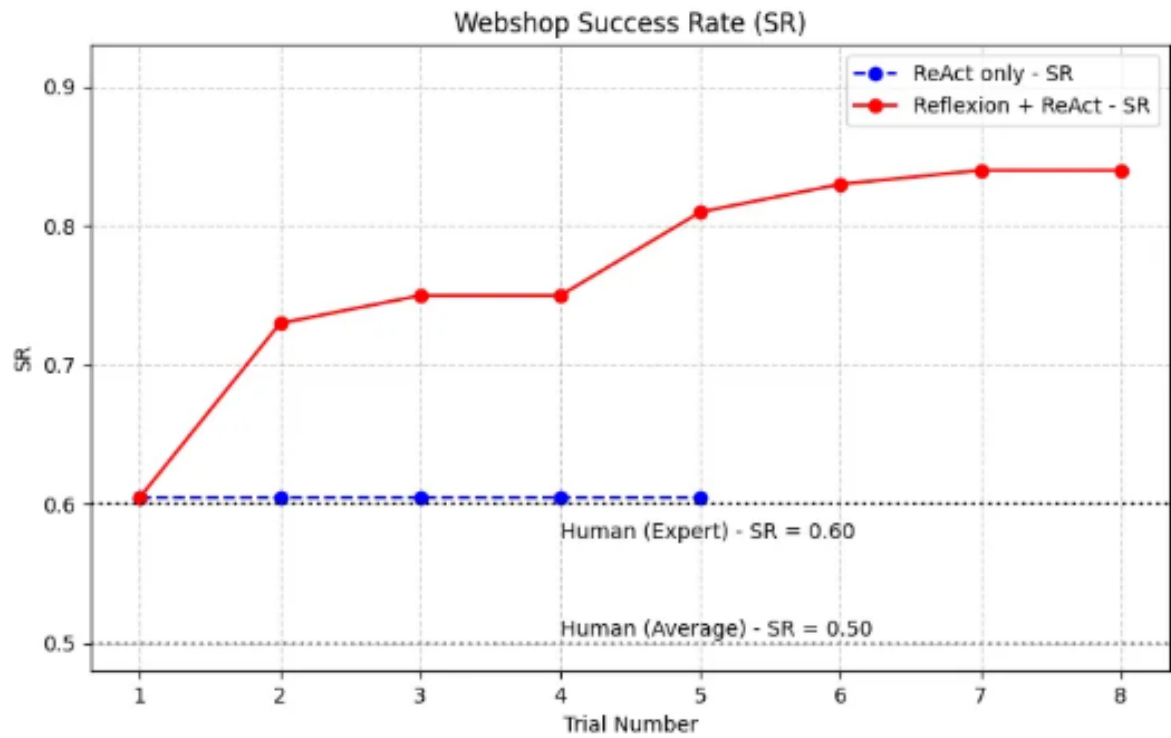
■ 진행 과정

1. Agent가 Webshop 내 "A" Instruction에서 문제 해결 **실패**(첫 번째 시도는 기본 ReAct Agent와 동일)
2. 자기 피드백 진행(**Reflexion**) & 메모리에 저장
 - a. 입력: Few-shot 예제 + "A" Instruction의 문제 해결 **로그**(Instruction, Action, Observation)
 - b. 출력: "A" Instruction의 문제 해결 과정이 실패한 이유 + 다음 시도에서 변경할 것
3. "A"Instruction **재시도**
 - a. 입력: one-shot 예제 + **메모리(2의 자기 피드백 내용)** + "A" Instruction
 - b. 출력: 가장 적합한 Action을 도출
4. 3번에서 **또 실패 시 2번**(자기 피드백) 다시 진행 → 2개의 자기 피드백을 주고 다시 시도(**8 Trial = 실패 시 최대 8번의 추가 시도**)

○ Agent 성능 평가

- Webshop 벤치마크 내 **200개**의 instructions 시도 후 성능 비교
- ReAct only의 경우 Temperature를 시도마다 0.2씩 증가 시켜서 진행했지만 큰 성능 변화가 없어서 Trial 5에서 종료함.
- Reflexion + ReAct는 6번의 시도 이후 거의 유사한 reflexion이 도출되었고 성능의 향상이 없어서 Trial 8에서 종료함.

○ Webshop 벤치마크



- 200개의 task에서의 성공 확률 및 스코어 점수.
Human (Average) : 데이터 생성에 참여한 13명, 500개의 test task 진행
Human (expert) : 데이터 생성에 참여한 13명 중 가장 성과가 우수한 7명

사람과 비교했을 때 ReAct only는 전문가 수준과 비슷하였다.
Reflexion 프레임워크를 도입하여 매우 높은 성능 향상을 도출해냈다.

- DB에서 주문 목록 불러오기 및 구매 Agent 활용 과정 workflow

- Order_product 테이블에서 **sub3**값을 읽어옴
- Agent에 들어가는 입력 **Webshop Instruction** 부분을 “i am looking for {sub3}.”로 변경.
- Agent 실행-1 Trial (Reflexion을 굳이 하지 않은 이유: 실패의 대부분이 webshop에 sub3으로 검색했을 때 검색 결과가 없어서 생김. 위 경우 **reflexion**을 통해서 해결이 불가능함. 시도 횟수가 늘어나면 걸리는 시간이 배가 됨.)
- “click[Buy Now]”가 모델에서 출력되면 정상적으로 구매를 한 것으로 가정함.

- 프로젝트 혁신성 및 사회적 영향력

- 혁신성

- 기존의 단순 수동 재고 관리 방식을 AI 기반의 자동화 시스템으로 전환하여 운영 효율성을 극대화
 - 최신 소비 트렌드 분석 및 리콜 상품 감지를 통한 신속한 대응으로 매출 증대 및 소비자 신뢰 확보
 - 머신러닝 기반 판매량 예측 및 소비 트렌드 상품 파악을 활용한 지능형 자동 주문 시스템으로 비용 절감 및 재고 최적화
 - 통합 대시보드 및 챗봇 기능을 통해 관리자 편의성을 향상시키고, 실시간 데이터 분석 기반 의사결정을 지원

- 사회적 영향력

- 중소 규모의 자영업자 및 무인 매장 운영자의 운영 부담 감소 및 경쟁력 강화
 - 소비자들에게 안전하고 트렌드에 맞는 상품 제공을 통한 만족도 증가
 - 리콜 상품의 신속한 감지 및 제거로 인한 안전한 유통망 구축
 - AI 및 데이터 기반 자동화 솔루션을 활용하여 지속 가능한 스마트 리테일 환경 조성

- 향후 개발 계획

- Category Searching Algorithm 고도화

- 트렌드 상품이 매출에 많은 도움이 된다고 판단
 - 유사 트렌드 상품이 실제 판매중인 상품인지 신뢰도를 제공하는 평가용 데이터셋을 구축

- 판매 수량 예측 모델 성능 향상

- 현재 주문량은 판매예측량 + 안전재고량 기반으로 계산
 - 최대한 맞춤형 재고 관리를 위해서는 더 정확한 판매예측량이 필요
 - 따라서 Imputation 및 데이터 전처리를 통해 모델의 성능을 향상

- 자동 주문 Agent 정밀도 향상

- 임베딩 기반 유사도 분석을 활용해 사용자 요청과 유사한 예제를 찾아 Few-Shot Learning 적용

- 다양한 웹사이트에서의 적용

- 이번 프로젝트에서 사용한 Webshop의 논문에 따르면 Webshop에서의 성능과 아마존, 이베이와 같은 실제 e-commerce 환경에서도 좋은 성능을 보임
 - 국내 e-commerce 환경인 네이버 쇼핑, 쿠팡등에 HTML Parsing을 통해 환경을 통일하여 사업성을 확장

참고 논문

- [1] (Shunyu Yao (2022). "WebShop: Towards Scalable Real-World Web Interaction with Grounded Language Agents". NeurIPS 2022)
- [2] (Shunyu Yao (2023). "ReAct: Synergizing Reasoning and Acting in Language Models". ICLR 2023)
- [3] Noah Shinn (2023). "Reflexion: Language Agents with Verbal Reinforcement Learning". NeurIPS 2023)
- [4] Yukun Dong, Yu Zhang, Fubin Liu, Xiaotong Cheng (2021). "Reservoir Production Prediction Model Based on a Stacked LSTM Network and Transfer Learning"

개인 회고

개인회고_김태한

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - 자동 주문 시스템을 구축하면서 AI Agent 개념을 이해하고 프로젝트에 적용해 보았습니다. ReAct, Reflexion 등의 개념을 논문을 통해 익히고, 이를 실제 코드에 적용하면서 이론이 코드의 어떤 부분에 활용되는지와 그 동작 방식을 이해하게 되었습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?
 - 이번 프로젝트에서 가장 큰 문제점은 업스테이지의 Solar Chat API를 통한 구현으로 Fine Tuning이 불가능한 점이었습니다. 이를 해결하기 위해 다양한 프롬프팅 기법을 활용한 LLM의 추론 능력 향상에 초점을 맞춰 Reflexion이라는 언어 형태의 자기 피드백 기법을 도입하였습니다. 문제 해결에 실패할 경우 실패 원인과 개선 계획을 자연어 형태로 모델로 다시 입력을 함으로써 모델을 학습시키지 않고도 WebShop의 문제 해결률을 60%에서 84%까지 향상시켰습니다.
- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 자동 주문 기능을 구현하기 위해 WebShop 환경을 세팅하고 DB에 저장된 물건 리스트를 받아와 자동 주문 Agent를 활용하여 가상 구매를 하는 부분을 엔드포인트로 구현하였습니다. 또한, 주문 결과를 n8n workflow 툴을 통해 DB에 업데이트하고 유저에게 메일을 발송하는 워크플로우를 만들어 자동화하였습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - 예제 데이터셋을 구축하고 임베딩을 통해 실제 요청과 가장 유사도가 높은 예제를 Few-Shot Learning으로 활용하여 Agent의 성능을 좀 더 향상시키고 싶습니다. 또한, WebShop이라는 정제된 환경이 아닌 쿠팡이나 네이버 쇼핑과 같은 실제 한국 이커머스 환경에 적용하여 고도화를 진행할 예정입니다.

개인회고_문채원

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - n8n을 활용한 로우코드 자동화를 처음 접하며, 다양한 노드를 조합하여 워크플로우를 구성하는 경험을 하였습니다. 또한, FastAPI와 n8n Webhook 트리거를 연동하여 백엔드 API와의 데이터 흐름을 최적화하는 시도를 진행하였으며, 이를 통해 비동기 API 처리 방식과 효율적인 데이터 연동 구조를 구축하는 경험을 쌓았습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?
 - 월간 트렌드 상품 예측 챗봇에서 제품 카테고리나 소비자 행동, 조회 기간의 주기적 특성 간의 연관성을 충분히 반영하지 못하는 한계가 있었습니다. 이를 해결하기 위해 Zero-shot, Few-shot, Chain-of-Thought 등 다양한 프롬프팅 기법을 실험한 결과, 개별 기법만으로는 분석의 깊이나 논리적 일관성이 부족하다는 점을 확인하였고 Thought

Process를 포함한 Few-shot Prompting을 구현하여 챗봇이 보다 신뢰도 높은 응답을 생성할 수 있도록 최적의 프롬프트를 구성하였습니다.

- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 크롤링된 이미지를 기반으로 Solar OCR API를 활용해 일정한 패턴을 인식하고, 이를 Supabase에 저장하는 방안을 구현하였습니다. 또한, OCR 분석 결과를 활용하여 n8n을 통해 유저에게 트렌드 상품 정보 메일을 발송하는 워크플로우를 추가하여 데이터 전달을 자동화하였습니다. 이와 함께, DB에 저장된 트렌드 상품 데이터를 활용한 AI 기반 트렌드 해석 챗봇 엔드포인트를 구현하였으며, 웹 프론트엔드에서는 사용자가 주요 기능을 직관적으로 탐색할 수 있도록 사이드바 UI를 개발 및 개선하였습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - LangChain이나 LangGraph 같은 AI 워크플로우 프레임워크를 활용하여 AI 기반 멀티스텝 자동화를 설계하고, 더 유연한 환경에서 최적의 워크플로우를 구현하고 싶습니다. 또한, 현재 내부 DB 데이터에 의존하는 방식인 챗봇 기능을 확장하여 SERP API와 소비자 동향 리포트, 전자상거래 트렌드 데이터 등 다양한 외부 정보를 반영한 RAG 기법을 적용함으로써, 보다 신뢰도 높은 답변을 제공하고 실시간 트렌드를 효과적으로 반영할 수 있도록 고도화할 계획입니다.

개인회고_서동환

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - n8n이라는 자동화 툴을 사용해 코드 없이 전반적인 워크플로우를 구성하였습니다. 초기에는 정보나 오픈소스 자료가 상대적으로 부족해 어려움을 겪었으나, 조원들과 함께 강의를 듣고 소스코드를 참고하며 해결하기 위해 노력하였습니다. 또한, 강의를 통해 학습했던 FastAPI를 활용하여 라우팅 및 비동기 처리 과정을 깊이 이해할 수 있었으며, Docker를 통해 실행 환경을 효율적으로 관리할 수 있었습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?
 - Upstage에서 제공된 Solar Chat API를 활용해 챗봇 기능을 구현하였습니다. 초기에는 질문에 일관된 답변을 제공하지 못해 프롬프트 구성에 어려움을 겪었지만, instruction을 적절히 추가하고 프롬프트를 수정하여 답변의 퀄리티를 개선할 수 있었습니다. 또한, 데이터베이스에서 값을 직접 불러오는 방식이 시간이 오래 걸려, 미리 함수를 정의해 값을 계산하고 이를 전달하는 방식을 적용하여 활용성을 높였습니다.
- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 사용자가 이전 판매 기록과 트렌드 정보를 직관적으로 확인할 수 있도록 대시보드, 챗봇, 인벤토리 페이지의 프론트엔드(UI/UX)를 설계 및 구현하였습니다. 또한, 데이터베이스의 정보를 기반으로 FastAPI를 활용하여 기간별 매출, 최다 매출 상품 등의 데이터를 제공하는 엔드포인트를 설계하고, 이를 처리하는 함수를 구현하여 데이터를 효과적으로 전달할 수 있도록 구성하였습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - 다음 프로젝트에서는 웹 크롤링이나 Chat API를 단순히 연결하는 방식을 넘어, CV나 NLP 요소를 추가하여 모델링 과정까지 포함시켜 프로젝트의 완성도를 더욱 높여보고 싶습니다.

개인회고_윤남규

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - 이전부터 개인적으로 공부해왔던, 다양한 기술 스택을 실제로 적용해볼 수 있었습니다. 노코드 기반으로 빠른 기능 구현이 가능한 n8n, PostgreSQL & pgvector 을 활용하는 협업용 데이터베이스인 Supabase, batch serving 으로 유용한 도구인 Airflow, 완성도 높은 코드를 위한 Ruff까지 사용해봄으로써 사용할 수 있는 기술 스택의 범위가 넓어졌다는 느낌을 받았습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?

- 노코드 기반의 **n8n**을 활용해 빠르게 원하는 기능을 구현할 수 있었지만, 코드 기반의 커스텀 로직 구현이 제한적이었던 점이 한계로 느껴졌습니다. 따라서 프로젝트 후반부에 코드 기반의 **Apache Airflow**로 기능 전환을 팀원들과 논의 후 결정했고, **DAG** 기반의 자유로운 로직을 도입함으로써 성능을 크게 개선할 수 있었습니다.
- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 프로젝트에 필요한 데이터베이스의 전처리 및 스키마 설계를 주도했습니다. 간편하게 사용할 수 있는 **SQLite**부터 시작해서, 확장성 및 유연성이 뛰어난 **PostgreSQL**, 그리고 최종적으로 동시 작업이 가능한 **Supabase**까지 데이터베이스를 옮기는 과정을 주도했습니다. 또한 OCR 및 크롤링을 활용해 불러온 트렌드 상품의 실제 유사 상품을 추천해주는 알고리즘을 개발함으로써 “트렌드 탐색 - 재고 관리 - 가상 주문” 까지 전 과정을 자동화하는 흐름을 완성했습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - 다양한 오픈 소스의 장단점을 비교하는 능력이 부족한 것 같아서 아쉽습니다. 예를 들어서 이번 프로젝트는 **batch serving**이 목적이므로, **Airflow**가 유용하게 사용될 수 있었지만 이를 프로젝트 막바지에 떠올려 급하게 기능을 전환했던 점이 아쉽습니다. 점점 사용할 수 있는 기술 스택은 많아지면서, 필요한 상황에 어울리는 적당한 도구를 사용할 수 있도록 빠르게 판단하는 능력이 필요한 것 같습니다.

개인회고_이재훈

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - n8n이라는 새로운 툴을 배우면서 사용해보고 Agent에 대한 개념과 개발 방법등을 알 수 있게되었고 이러한 기능을 사용자에서 보여주도록 FastAPI와 React를 이용해 백엔드와 프론트엔드를 개발하면서 웹에 대한 지식도 알게 되었습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?
 - 기존의 Dash 라이브러리를 이용해 대시보드 작업을 진행할 계획이었지만 Dash의 한계를 느끼고 FastAPI + React로 변경하면서 백엔드의 많은 엔드포인트와 초반 UI를 설계하는데 어려움을 겪었습니다. 팀원들의 의견을 많이 물어보기도 하고 다양한 웹사이트에서 insight를 얻어가면서 진행했고 결과적으로 정상적인 웹 사이트 개발에 성공했습니다.
- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 백엔드 및 프론트엔드, 판매량 예측 모델 부분입니다. 사용자에게 우리의 서비스를 보여주기 위해 FastAPI, React 개발 작업을 진행했고 자동 주문 Agent에 사용되는 주문량 계산을 위한 판매량 예측 모델을 개발했습니다. 시계열 데이터 분석 모델을 이용해 약 PFSA 스코어 기준 0.55의 성능을 기록하는 모델을 만들었습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - 이번 프로젝트에 CV적 요소가 없어 아쉬웠습니다. CV 요소도 포함하는 주제를 하고 싶고 팀원들을 통해서 여러 오픈 소스나 기술들을 많이 알게 되었습니다. 다양한 툴을 경험하면서 다음 프로젝트를 진행하면 좋을 것 같습니다.

개인회고_장지우

- 이번 프로젝트에서 어떤 기술적 도전과 성장이 있었나요?
 - Agent를 구현하기 위해 팀원 전부 처음 접한 n8n 툴을 활용하는 과정에서, 다같이 강의를 듣고 응용해보며 결과물을 낼 수 있었습니다. 또한 Fastapi를 활용하며 비동기 처리 방식, 실시간 웹소켓 방식과 엔드포인트 구축에 대해 많이 배우고 사용해보았으며, React를 활용해 프론트엔드를 구현할 수 있었습니다. 이렇게 다양한 프레임워크와 Github, Jira, Notion이라는 다양한 협업툴을 사용해보며 많은 지식과 경험을 쌓을 수 있었습니다.
- 프로젝트 진행 중 만난 문제를 어떻게 해결했나요?
 - n8n으로 구현된 자동 주문 agent를 활용해 주문 현황을 실시간으로 불러와 Fastapi로 엔드포인트를 구축하는 과정에서 실시간 정보 처리가 안되는 문제가 발생해 어려움을 겪었습니다. 이를 해결하기 위해 n8n에서 webhook으로 주문 현황 정보를 전달하고, 해당 정보를 헤더의 크기가 작아 효율적이고 실시간 네트워킹이 가능한 웹소켓을 이용해 구현할

수 있었습니다. 또한 인벤토리 관련 데이터를 Supabase에서 불러오는 과정에서, 서버 과부하 문제가 있었지만 이를 청크사이즈로 불러오는 방식으로 해결할 수 있었습니다.

- 프로젝트에서 본인이 기여한 핵심적인 부분은 무엇인가요?
 - 사용자의 제품들, 재고 현황, 재고 부족 정보, 주문해야할 수량, 주문 현황 등을 직관적으로 볼 수 있도록 인벤토리의 프론트엔드(UI/UX)를 설계 및 구현하였으며 Fastapi를 활용해 엔드포인트를 구축하고 웹소켓과 n8n의 Webhook을 활용해 실시간 정보를 불러오는 기능을 구현하였습니다. 또한 인벤토리 관련 데이터를 Supabase에서 효율적으로 불러올 수 있도록 구현하였습니다.
- 다음 프로젝트에서 개선하고 싶은 점은 무엇인가요?
 - 부스트캠프에서 배웠던 CV적 요소를 활용할 수 있게끔 프로젝트를 진행하고 싶고, 데이터베이스를 처음부터 확실하게 설계, 구축해두고 개발을 진행할 것입니다. 또한 팀원들을 통해 알게된 다양한 툴을 사용해보고 싶습니다.