

도전하며 달려가는 개발자



"안녕하세요 🙋"

항상 **목표**를 세우고 **행동**하는 이준경입니다."

하나의 목표가 아닌 다채로운 목표로 성장하기 위해 노력합니다.

자신 뿐만 아니라 팀과 목표를 공유하여 서로를 이끌어 갑니다.

현재 **Back-End 개발자**를 목표로 달려가고 있습니다!

☎ _ 010-3155-0419

🔄 _ <https://github.com/away0419>

✉ _ away0419@naver.com

🎓 Education

2022.01-2022.12 삼성 청년 SW 아카데미 7기 (삼성)

2021.03-2021.08 스프링 개발자 과정 (ITWILL)

2014.03-2018.02 컴퓨터 공학 전공

💼 Career

2023.01~ KTDS

🏆 Prize

2022 최종 프로젝트 우수상 (삼성 청년 SW 아카데미)

🎸 Etc

2018.03-2020.06 육군 통신 장교 (중위)

2021 정보 처리 기사

🔧 Skill

상 : 서비스 환경에 맞춰 최적화를 수행할 수 있는 높은 수준의 기술 능력

중 : 주요 이슈 트러블 슈팅을 할 수 있으며 프로젝트를 수행함에 문제가 없음

하 : 일부 프로젝트에서 사용한 경험이 있으며 기본 기능을 구현할 수 있음

Spring Boot

- 5개 이상의 프로젝트에서 사용한 경험이 있으며 가장 자신 있는 기술입니다.
- Spring Cloud, WebSocket, AWS 등 라이브러리 사용에 익숙합니다.

MySQL, Oracle

- 30개의 테이블로 구성된 ERD를 설계하고 작성한 경험이 있습니다.
- 성능 최적화를 고민하고 필요에 따라 반 정규화를 적용했습니다.

Java

- 자료 구조를 활용하여 기능을 최적화 할 수 있습니다.
- 모듈화로 중복 코드를 제거하고 재사용성을 높였습니다.

Etc

- Redis, Git, HTML, CSS, JSP, Vue, JQuery, Spring Cloud, Python, Fast API, AWS, Oracle Cloud, Docker, Nginx, RabbitMQ, Jenkins

★ Project (4개)

Beanzido 2022.10-2022.11 (6인)

개요	근처 사람들과 대화할 수 있는 지역 기반 실시간 채팅 서비스 입니다. 철저한 익명으로 부담 없이 이용 가능합니다.
개발환경	Spring Boot, FastAPI, RabbitMQ, JPA, AWS, Jenkins, Docker, Git, Nginx, MySQL, Redis, Grafana
담당역할	팀장, 백엔드(50%), 서버 구성(100%), 기획 및 설계 (30%), 산출물 작성(50%)

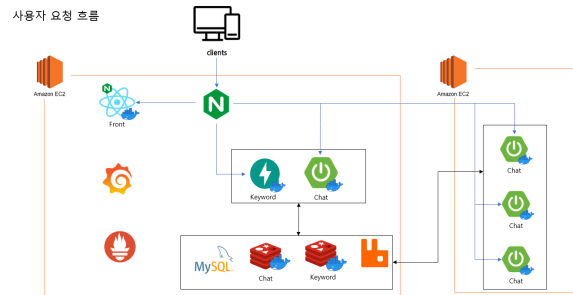
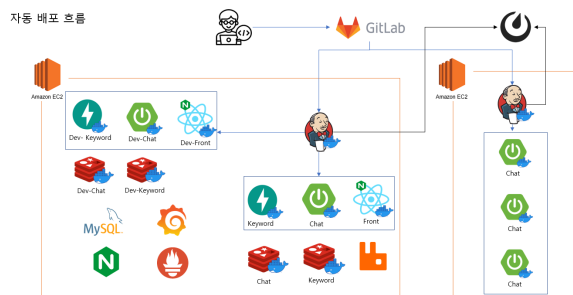
■ 담당 구현 기능 및 작업

- WebSocket을 이용한 **실시간 통신** 구현
- AWS Rekognition으로 **이미지 AI 분석** 후 특정 카테고리 일치율 70% 이상이면 **필터링**
- **보이어-무어 알고리즘**을 이용한 패턴 매칭으로 **비속어 필터링**
- Python 형태소 분석 라이브러리를 활용하여 지역 별 한국어, 영어 **키워드 분석** 후 랭킹 제작
- 채팅 서버를 **로드 밸런싱**으로 구축 후 RabbitMQ로 메시지 공유
- Grafana, Prometheus로 서버 성능 **모니터링** 시각화
- Jmeter를 이용한 **과부하 테스트** 진행
- Git의 Branch 별 **자동 배포** 관리
- 개발 서버와 배포 서버 동시 운영
- 전체 일정 관리 및 문서 작성



지역별 키워드를 한눈에
국내 지역의 실시간 키워드를
순위별로 확인 *Local Keywords*
가장 많이 언급된 키워드가 우선순위가 확인됩니다.

**내 위치를 중심으로
실시간 주변 이슈 확인**
Realtime Communication
내 주변 이용자와 익명으로 소통
안정이나 주변 이슈 등 시시공공한 게시물을 한눈에 공유
자주한 시간 순서
부적절한 게시글이나 사용은 필터링되는 서비스



! 발생한 문제와 해결 방법

RabbitMQ로 서버 간 메시지 공유

내용	메시지 공유 서비스를 구현하는 과정에서 RabbitMQ 어노테이션에 Queue를 설정할 때 Const 오류가 발생했습니다.
원인	- 자동 배포를 진행할 때 동일한 프로젝트로 2개의 서버를 운영하는 로드 밸런싱이기 때문에 Queue 이름을 명시적으로 표시할 수 없었습니다. (서버마다 큐가 달라야 하기 때문입니다.) - env에서 Queue 이름을 받아오려 했으나 Const 오류 발생
해결 방법	RabbitMQ 공식 문서를 찾아본 결과 Queue를 임시 생성할 수 있는 설정이 있어 이를 활용했습니다.
결과	성공적으로 서로 다른 Queue를 하나의 Exchange에 연결하여 메시지 공유를 구현했습니다.

WebSocket 이미지와 글 동시 전송

내용	WebSocket 통신으로 사진과 글을 서버로 동시에 보내면 매개 타입 오류가 발생했습니다. 따로 보낼 경우 중간에 다른 사람이 글을 보내면 내가 동시에 보낸 글과 사진 사이에 다른 사람의 글이 남게 됩니다.
원인	WebSocket의 경우 사진은 바이너리 메시지로 요청하지만 글은 텍스트 메시지로 요청 해야하기 때문입니다.
해결 방법	Image를 Base64로 변환하여 JSON 형식으로 글과 합친 뒤 String으로 변환하여 요청을 보냈습니다.
결과	이미지와 글을 동시에 전송하여 중간에 다른 메시지가 오지 않도록 구현했습니다.

🔥 프로젝트 후기

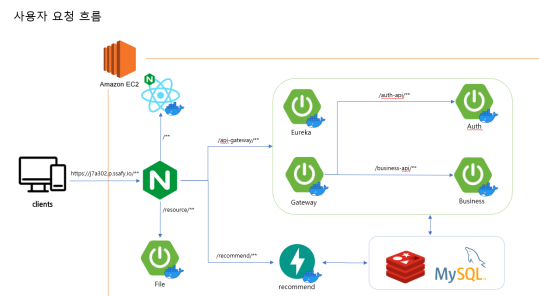
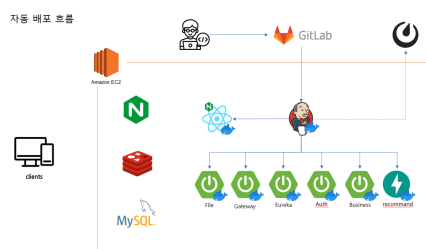
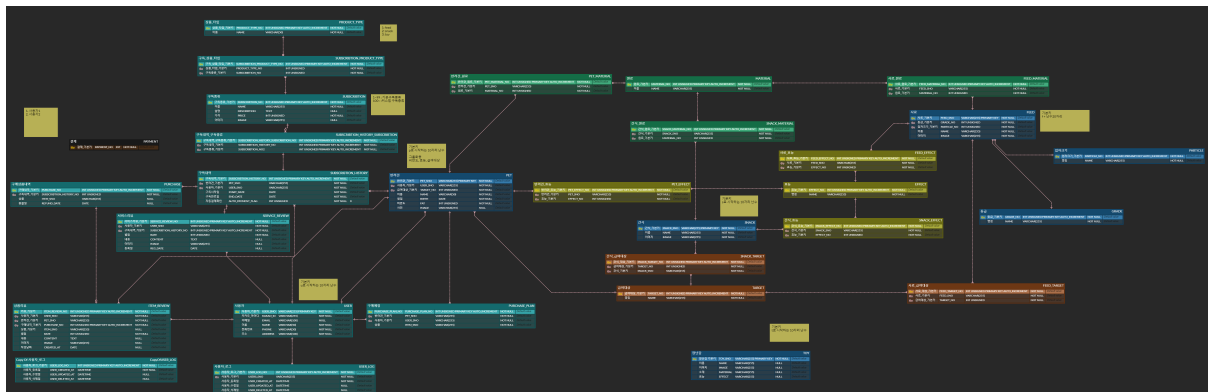
- 필수 기능인 실시간 채팅만 먼저 구현한 뒤 빠르게 배포하여 사용자의 피드백을 받았습니다. 이후 기능을 추가하고 개선해 나가는 과정에서 배포 서버와 개발 서버를 분리 하는 것이 편하다는 것을 느낄 수 있었습니다. 또한 개발자가 생각하는 서비스와 실제 사용자가 원하는 서비스가 다르다는 것을 배울 수 있었습니다.
- 서버 모니터링과 Jmeter로 과부하 테스트를 진행하며 실제 서비스 시 사용자의 수, 메시지량 등 트래픽 관리를 어떻게 할지 고민해 볼 수 있었습니다. 하나의 방법인 스케일 아웃을 적용하기 위해 Nginx를 이용하여 로드 밸런싱을 구축하였고 그 결과 최대 session 연결 수를 1초 150개에서 1초 300개로 증가시켰습니다. 또한 서버 간 메시지를 공유하고자 RabbitMQ를 활용했습니다. 이를 통해 프로젝트 기획 시 트래픽을 고려해야 한다는 것을 알 수 있었습니다.
- Keyword 분석을 위해 Python 라이브러리를 사용하였고 지역 별로 분류하고 다시 합치는 과정에서 Python의 자료구조를 학습할 수 있었습니다.
- AWS Rekognition을 사용하기 위해 블로그 위주로 탐색하였으나 결국 공식 문서를 보는 것이 중요하다는 것을 다시금 깨닫게 되었습니다.
- 비속어 필터링에서 보이어-무어 알고리즘을 적용하며 개념을 학습할 수 있었고, 실제 알고리즘을 문제 풀이가 아닌 기능에 활용하며 지금까지 배운 알고리즘도 이후 기능 구현에 사용해 볼 생각입니다.

🔗 <https://github.com/away0419/Beanzido>

개요	반려견을 위한 맞춤 상품 추천 과 정기 구독 서비스 입니다.
개발환경	Spring Boot, FastAPI, Spring Cloud, JPA, Selenium, Jenkins, Docker, AWS, Git, Nginx, MySQL, Redis
담당역할	부팀장, 백엔드(40%), 서버 구성(100%), 기획 및 설계(30%), 산출물 작성(30%)

담당 구현 기능 및 작업

- kakao API를 이용한 **소셜 로그인**
- **JWT**를 활용하여 결재, 개인정보 관련 서비스 사용 시 인증
- Selenium으로 네이버 스토어에서 상품, 리뷰 데이터 **크롤링**
- 수집 데이터 테이블 형식에 맞게 정제 후 데이터 입력
- 정기 구독, 커스텀 구독
- 상품, 구독 리뷰 작성, 조회
- Python 라이브러리를 이용하여 **협업 필터링**
- 콜드 스타트 해결을 위한 **콘텐츠 필터링**
- **API Gateway** 서버를 구성하여 공통 필터 관리 및 URL 확인 후 서버 연결
- Jenkins를 활용한 **자동 배포**
- Eureka 서버, Image 서버, Business 서버, Auth서버, API Gateway 서버 **구성**
- 자주 조회하는 칼럼 **파티셔닝** 등 DB 설계
- Back-End 일정 관리 및 문서 작업



! 발생한 문제와 해결 방법

상품 리뷰 조회 시간 14초 → 1초 단축


내용	상품의 리뷰를 조회 할 경우 14초 이상 걸려 응답이 느려졌습니다.
원인	- 데이터베이스 설계 시 상품(사료,간식,상품)을 각각의 테이블로 나누고 이를 관계 설정해주는 테이블을 설정하지 않음. 이로 인해 리뷰 테이블의 상품에 인덱스가 자동 설정 되지 않아 조회 시 시간 증가했습니다. - JPA N+1 문제로 리뷰에 연결된 다른 테이블의 정보를 다시 조회하며 시간 증가했습니다.
해결 방법	- 리뷰 테이블의 상품 필드에 인덱스를 설정했습니다. - Fetch Join을 통해 JPA N+1 문제를 해결했습니다. - ENTITY에 배치 사이즈를 설정했습니다.
결과	14초 이상 걸리던 조회 시간을 1초 이하로 줄일 수 있었습니다.

데이터 정제 및 DB 저장

내용	- 상품과 리뷰 데이터를 크롤링하고 이를 CSV로 변환 할 때 구분자로 인한 오류가 발생했습니다. - CSV 파일을 MySQL workbench에서 Import 할 경우 상당한 시간이 소요되었습니다.
원인	- Java에서 CSV로 변환할 경우 구분자가 ,로 고정되어 리뷰에 포함된 ,를 구분하지 못했습니다. - workbench로 import 하는 경우 데이터를 각각 삽입 후 커밋하기 때문이었습니다.
해결 방법	- Java의 경우 Excel 파일로 저장한 뒤 Excel에서 CSV로 변환하여 만들었습니다. Python으로 테스트 한 결과 구분자를 구별 할 수 있는 라이브러리가 있어 문제가 되지 않았습니다. - 직접 SQL문을 작성하여 CSV를 불러와 데이터를 한번에 넣어 해결했습니다.
결과	- 수집한 데이터를 CSV 변환할 수 있었습니다 - 50만개 이상의 데이터 저장 시간을 1시간 이상에서 10초 이하로 줄일 수 있었습니다.

🔥 프로젝트 후기

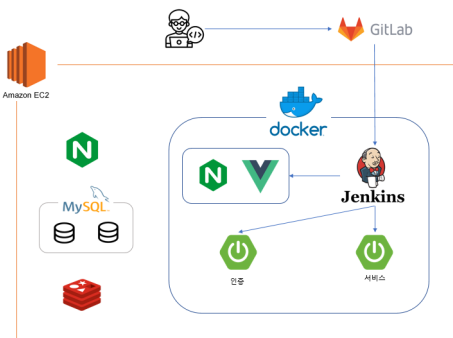
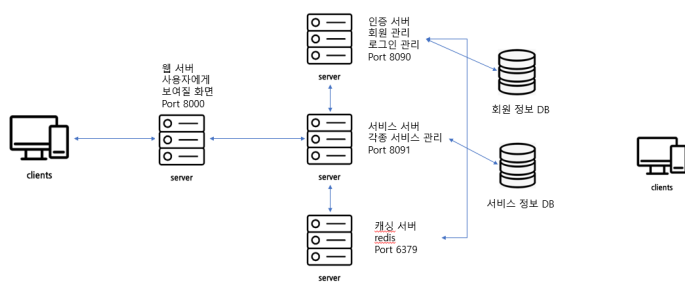
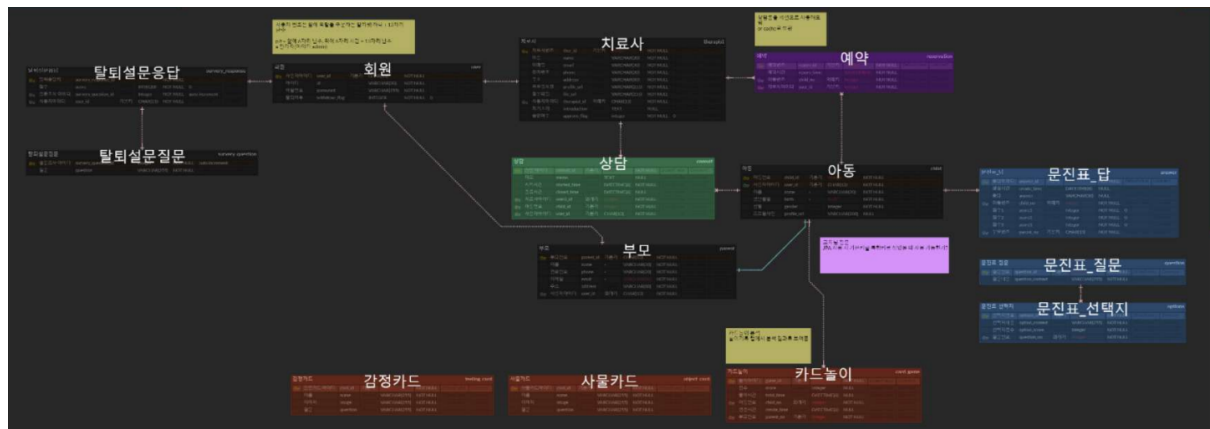
- 데이터를 수집, 정제하는 과정에서 Python의 강점을 느낄 수 있었습니다.
- Spring Cloud를 이용하여 API Gateway 서버를 구축하고 공통 필터를 구현하면서 MSA 구성 시 API Gateway가 필요한 이유를 학습할 수 있었습니다.
- 추천 기능을 구현하기 위해 콘텐츠 기반, 협업 기반 알고리즘의 개념을 학습할 수 있었고 이는 다른 프로젝트에서도 유용하게 사용할 수 있다 생각합니다.
- JWT를 구성하며 Access Token이 필요한 이유를 알게 되었고 이를 활용한 인증 방식 흐름을 이해할 수 있었습니다.

 <https://github.com/away0419/daldalhagae>

개요	상담사와 자폐 아동을 연결해주고 상담을 받을 수 있는 화상 상담 서비스 입니다.
개발환경	Spring Boot, JPA, Jenkins, Docker, AWS, Git, Nginx, MySQL
담당역할	백엔드(40%), 서버 구성(100%), 기획 및 설계(20%), 산출물 작성(20%)

■ 담당 구현 기능 및 작업

- 회원 분류(부모, 상담사) 별 회원 가입
- 아동 등록, 수정
- 아동 증상 확인 후 해당 전문 상담사 추천
- 상담사 아동 1 : 1 상담 방
- 문진표 작성
- 자주 조회 되는 정보 Redis 캐싱 적용
- 상담 예약 및 관리
- 민감한 정보 테이블 분리 및 DB 설계
- 자동 배포 및 서버 구성



! 발생한 문제와 해결 방법

OpenVidu를 이용한 화상 채팅

내용	OpenVidu를 사용하기 위해 Certbot 인증서를 발급 받아 연동하는 과정에서 인증서를 인식하지 못하는 에러 발생했습니다.
원인	OpenVidu의 기본 docker-compose.yml에서 설정된 volumes이 정상적으로 적용되지 않아 인증서를 가져오지 못하고 결과적으로 인증서를 인식하지 못하는 것이 문제였습니다.
해결 방법	공유되는 폴더에 인증서를 직접 명령어로 복사했습니다.
결과	인증서를 정상적으로 인식하고 화상 채팅 연결 성공했습니다.

WebSocket 이미지와 글 동시 전송

내용	문진표 결과를 DB에 저장하는 과정에서 에러가 발생했습니다.
원인	DB의 필드가 varchar(255)로 설정된 상태에서 255byte 보다 큰 문장을 저장하려 했기 때문이었습니다.
해결 방법	해당 결과를 저장하는 필드의 타입을 TEXT로 변경하여 크기 제한을 늘렸습니다.
결과	문진표의 결과가 정상적으로 DB에 저장할 수 있었습니다.

🔥 프로젝트 후기

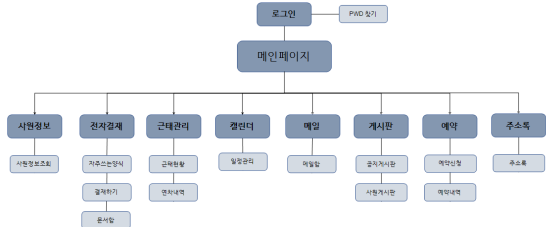
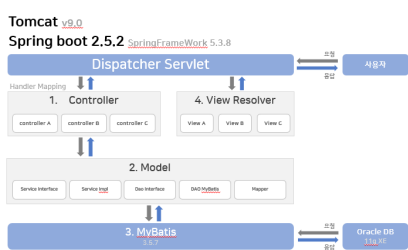
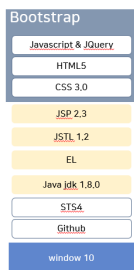
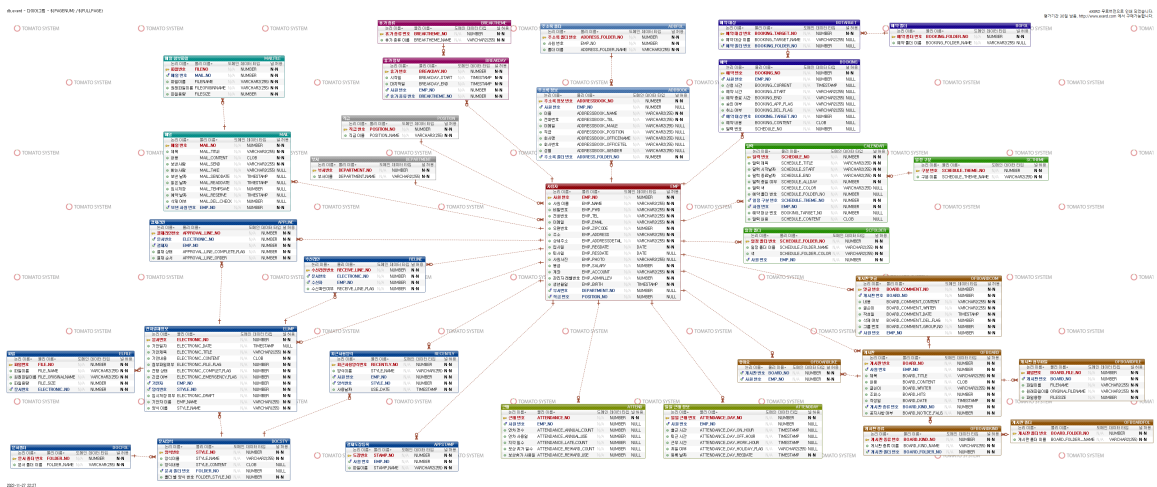
- 삼성 SW 아카데미에서 실시한 첫 팀 프로젝트로 협업을 통하여 개발 외에도 기획 및 설계, API 작성, JIRA 사용법 등을 직접 해보며 습득할 수 있었고 중요성을 깨닫게 되었습니다.
- 서버를 구성하는 과정에서 MSA 구성, Docker의 개념과 사용법, 자동 배포를 위한 Jenkins, 웹 서버와 리버스 프록시 역할을 하는 Nginx를 학습하고 기초를 다질 수 있었으며 개발 역량을 크게 높일 수 있었습니다.
- JPA의 경우 스스로 학습하는 것이 쉽지 않았으나 컨설턴트님의 도움과 팀원과의 소통을 통해 프로젝트에 적용할 수 있었습니다. Query문을 모두 작성할 필요가 없어 개발 시간을 단축할 수 있었습니다.

 <https://github.com/away0419/bomnal>

개요	회사에 필요한 기능을 제공하여 업무 효율을 높여주는 그룹웨어 서비스 입니다.
개발환경	Spring Boot, MyBatis, Oracle, JSP
담당역할	팀장, 풀스택(40%), 기획 및 설계(30%), 산출물 작성(40%), 배포(100%)

특이사항

- **SMTP** 이메일 인증
- FullCalendar 이용한 개인 **일정**
- 일정 그룹 체크 박스로 해당 그룹에 포함된 모든 일정 일괄 보임/숨김
- Excel 업로드로 데이터 입력, 조회 가능
- 출/퇴근 클릭 시 정보 저장 및 그래프로 **시각화**
- 스케줄러를 이용한 **자동 퇴근**
- **AJAX** 활용
- 관리자용 페이지



 [_https://github.com/away0419/Group](https://github.com/away0419/Group)

 <https://www.notion.so/IU-5b66c5df1e954c2dac5963d0ea4357ad>