

CONTACT.

✉ seeunchoi99@gmail.com

💻 github.com/dahyen0o

📖 sechoi.tistory.com

TECH SKILL.

Programming Language

Java, Python

Framework

Spring Boot

Database, ORM

MySQL(MariaDB), JPA

etc

Git, Github, AWS EC2

EDUCATION.

📖 서강대학교 컴퓨터공학과
학사 졸업

2019년 3월~2023년 2월

AWARD.

📖 서강대학교 캡스톤디자인2
우수 프로젝트 경진대회
최우수상

2022년 12월 | 현대모비스

최세은

문제 해결을 좋아하는 서버 개발자입니다.

INTRODUCE.

- 안정적인 서비스를 위해 항상 고민합니다. 테스트와 의존성 관리를 통해 변경에 유연한 코드를 작성하려고 노력합니다.
- 자동화를 통해 다양한 불편함을 해소해왔습니다.
 - 자동 TIL 봇 (github.com/dahyen0o/auto-TIL)
 - 출석/퇴실 체크 알림 봇 (github.com/dahyen0o/ssafy-attendance-notification-bot)

WORK EXPERIENCE.

우아한테크캠프 6기

우아한형제들 | 2023년 7월~8월 (2개월)

- 참고: github.com/dahyen0o/woowa-tech-camp
- 페어 프로그래밍 및 팀 프로젝트 진행
 - Java 어플리케이션 개발을 통한 TDD와 OOP 학습
 - 웹 어플리케이션 개발을 통한 Spring Boot, JPA, ATDD 학습
- <좋은 코드, 나쁜 코드>, <친절한 SQL 튜닝> 스터디 진행

백엔드 인턴

미리디 | 2022년 7월~8월 (2개월)

- Spring Boot, TypeScript, RDBMS, JSP를 사용해 비즈하우스 웹 페이지에 상품 추가
- Java 코드로 관리하던 상품 배송 정보를 JSON 형식으로 변환해 데이터베이스에서 관리하도록 변경하는 리팩토링에 참여

EXTRA EXPERIENCE.

삼성 청년 SW 아카데미

Java 전공반 | 2023년 1월~2023년 6월 (6개월)

- 알고리즘, Java, 웹 개발(Spring Boot, MySQL, JavaScript) 학습
- 매주 알고리즘 스터디를 진행하며 백준 골드2 달성
- 반을 위한 Mattermost 알림 봇 개발
(github.com/dahyen0o/ssafy-attendance-notification-bot)

멋쟁이사자처럼 서강대

백엔드 트랙 | 2022년 1월~2022년 12월 (1년)

- Django를 중심으로 웹 개발 학습
- DevStory 프로젝트에 백엔드 팀원으로 참여
(github.com/DEMODAY-devStory)

PROJECT.

ZI9ZA9

지구재구 - 의류 쇼핑몰

백엔드 팀원(백엔드 4명) | 2023년 8월(3주)

우아한테크캠프 팀 프로젝트

<https://github.com/wootecam-gugucon/shopping-mall>

사용 언어 및 기술

- Java 17, Spring Boot, Spring Security
- MariaDB, Spring Data JPA
- Github Actions, Docker, Flask, AWS EC2

기여

조회 쿼리 최적화

대용량 데이터(주문 500만개, 주문 품목 1250만개)을 사용하는 쿼리 최적화

- 특정 키워드로 상품을 검색한 후 주문 많은 순으로 정렬하는 슬로우 쿼리 등장
 - 원인: LIKE 구문과 1250만 개의 데이터가 있는 이너 테이블을 인덱스 없이 Nested Loop 방식으로 JOIN
 - 커버링 인덱스 적용으로 데이터 탐색 시간 단축
 - 결과: 48s → 24ms로 실행 시간 단축
- 상품 검색 시 '주문자의 성별 및 나이대' 조건 및 '별점순' 정렬 기능을 추가하며 슬로우 쿼리 등장
 - 원인: 600만 개의 별점 데이터, 500만 개의 주문 데이터를 탐색하며 커버링 인덱스를 적용해도 3s 이상 소요
 - 통계 테이블을 추가해 데이터 역정규화 → 조회할 데이터의 개수를 크게 낮춤
 - 결과: 해당 쿼리 외에도 모든 조회 쿼리 실행 시간을 15ms 이하로 단축

결제 성능 향상

- 토스페이먼츠 결제 모듈을 통한 결제 과정에서 재고 업데이트 관련 성능 향상 (자세히)
 - 원인: 재고에 대한 갱신 손실(Lost Update) 문제를 인식 후 비관적 락 추가 → 결제 승인 시 외부 API 통신과 같은 트랜잭션에서 실행되며 같은 상품을 다른 사용자가 주문 시 긴 대기 발생
 - 결제 요청 시 재고가 미리 차감되는 로직으로 변경
 - 완료되지 않은 주문에 대한 재고는 Spring Scheduler를 통해 30분마다 복구
 - 결과: 결제 승인 TPS를 309% 개선(32→99), 데이터 정합성을 지키면서 성능 향상에 성공 → 사용성 개선

테스트를 이용해 안정성 기여

- ATDD 적용
 - 모든 사용자 시나리오에 대해 인수 테스트 추가
 - 결과: Jacoco 기준 테스트 커버리지 95% 달성 → 별다른 QA 없이도 서비스의 안정성을 크게 향상
- CI를 통해 실행 실패한 테스트가 존재하는 경우 merge 제한

Floney

플로니 - 함께 기록하는 공유가계부 앱

백엔드 팀원(iOS 1명, 안드로이드 1명, 백엔드 3명, 기획·디자인 2명) | 2023년 3월~진행중

<https://github.com/Floney-2023/Floney-Server> (🍏 iOS 다운로드)

사용 언어 및 기술

- Java 17, Spring Boot, Spring Security
- MySQL 8.0, Spring Data JPA, QueryDSL, Redis
- Github Actions, AWS EC2, AWS RDS, Amazon S3, AWS Code Deploy, AWS ELB
- Prometheus, Grafana

기여

데이터 정합성 해결

- MySQL에서 유니크 인덱스와 전용 문법을 사용해 Repeatable Read 격리 수준에서의 **동시성 문제 해결** ([자세히](#))
 - 원인: 특정 조건을 만족하는 데이터는 유일해야 함 → 데이터가 있으면 UPDATE, 없으면 INSERT 필요
 - 하나의 쿼리로 조건에 따른 INSERT 혹은 UPDATE를 실행
 - 결과: 별도의 Lock 없이 빠른 성능으로 도메인 제약 사항을 만족시켜 사용성 향상
 - 이전 코드로 인해 생성된 잘못된 데이터들은 다음달 아침 전까지 수정하여 사용자 불편 또한 빠르게 해소

레거시 리팩토링

- 변화하는 도메인과 추가되는 코드에 대응하기 어려웠던 **카테고리** 관련 코드를 스키마 설계부터 리팩토링 리드 ([자세히](#))
 - 한 달 동안 9000줄이 넘는 코드 수정
 - Flyway 스크립트로 변경된 스키마에 따라 데이터 마이그레이션 진행
 - 결과: 가독성 향상, 일부 로직에서 쿼리를 한번 더 실행하던 불편함 해결, 데이터 정합성을 Lock 없이 간단하게 해결

인프라 구축

- Github Actions, AWS S3, AWS Code Deploy로 환경에 따른 CI/CD 구축
- AWS ELB, Route 53를 사용해 HTTP 요청에 SSL 인증서 적용
- Flyway와 git submodule을 도입해 환경/버전에 따라 구분되는 값을 용이하게 관리
- 데이터베이스 보안 강화
 - Load Balancer에서 데이터베이스(AWS RDS) 직접 접근이 불가하도록 설정
 - 비밀번호 등 민감정보는 Bcrypt 알고리즘을 통해 해싱된 값을 저장