

# 김성민



## 프로필

코드의 책임과 테스트 가능성에 관심이 많은 Back-End 개발자입니다.

단순 기능이 동작하는 서비스를 만드는 것이 아니라, 유지보수의 비용이 적고, 신뢰성 높은 서비스를 만들기 위해 노력합니다.

이와 같은 코드를 작성하기 위해 OOP와 Hexagonal Architecture에 많은 관심을 가지고 서비스에 적용하는 노력을 하고 있습니다.

## 프로젝트

### 팀 성과 평가 서비스, Full-Stack — 2024.04~2024.05

주요 기술 - Java, Spring Boot, JPA, QueryDSL, MySQL, Keycloak, JavaScript, React, Docker  
팀의 성과를 평가하는 시스템을 개발하는 프로젝트에서 React와 Java, Spring을 이용하여 기획부터 Full-Stack 개발 전 과정을 수행했습니다.

### 알고리즘 학습 사이트 GPGT, Team Leader / Back-End — 2023.08~서비스중

주요 기술 - Java, Spring Boot, JPA, QueryDSL, MySQL, Github Action, AWS, Docker  
사용자의 알고리즘 학습 내역을 자동으로 수집해서, 포인트와 경고를 부여/관리하며  
유저의 학습 통계, 학습 동기부여, 커뮤니티 기능을 제공하여 구성원들의 동기부여와  
관리자에게 관리의 용이성을 제공하는 서비스를 개발하고 1년 가량 운영중입니다.

### CherryPicker, Back-End Leader — 2023.03~2023.06

주요 기술 - Java, Spring Boot, JPA, MySQL, Github Action, AWS, nGrinder  
지역내에서 개인화 음식점 추천 기능을 제공하는 '광진구 음식점 추천 어플'  
Cherry Picker의 Back-End 팀장을 맡아 Spring Boot와 JPA를 이용한 서버를  
졸업 프로젝트로 개발했습니다.

## 학력

세종대학교 - 컴퓨터공학 학사, 졸업, 2022.03 ~ 2024.02

인천대학교 - 컴퓨터공학 학사, 편입으로 인한 중퇴, 2018.03 ~ 2022.02

소래고등학교 - 이공계열, 졸업, 2015.03 ~ 2018.02

## 교육

### 42 Seoul, Cadet - 2023.03~2024.07

42서울은 과학기술정보통신부의 주최로 진행된 교육기관으로, 1년 반 가량의 기간 동안  
C, C++, Unix, Shell 등의 기술을 코드 리뷰를 통한 동료학습을 통해 학습했습니다.

### 삼성SDS 동계 대학생 알고리즘 특강 - 2023.02~2023.02

삼성SDS에서 졸업예정 대학생을 대상으로 오프라인으로 진행된 알고리즘 특강에  
참여하여 2주간 C++를 이용한 교육을 수료했습니다.

## 자격사항

- 삼성 SDS SW검정, Professional 등급
- 세종대학교 TOSC SW코딩역량평가, 1급 (C언어, 만점)
- TOEIC Speaking, IM3 등급

## 수상 내역

- 인천대학교 매트랩 Cody 챌린지, 우수상 수상

# 알고리즘 학습 사이트 GPGT - 2023.08~서비스 종



## 주요 기술

**Back-End.** Java, Spring Boot, JPA, Spring Security, QueryDSL, Python, Flask

**Front-End.** JavaScript, React

**Database.** MySQL

**DevOps.** AWS(EC2, RDS, S3), Docker, Docker Compose, Github Action

**OS.** Amazon Linux

## 서비스 소개

사용자의 알고리즘 학습 내역을 자동으로 수집해서 사용 유저에게 학습 통계, 학습 랭킹, 주간 팀 경쟁, 다른 유저들과 공유하는 학습 로드맵, 포인트/경고를 통한 학습 동기부여, 게시판을 통한 커뮤니티 기능을 제공합니다.

관리자에게 이벤트, 포인트/경고 로그, 민원, 유저 설정등이 가능한 Back Office를 제공합니다.

이와 같이 구성원들의 동기부여와 관리자에게 관리의 용이성을 제공하는 서비스를 10개월 가량 운영중입니다.

## 개발 과정

프로젝트 진행의 전 과정에서 Git과 Jira를 통해 협업했습니다.

저는 팀장으로 Java, Spring Boot와 Mysql, JPA, Querydsl을 이용한 백엔드 개발을 맡아서 진행했습니다.

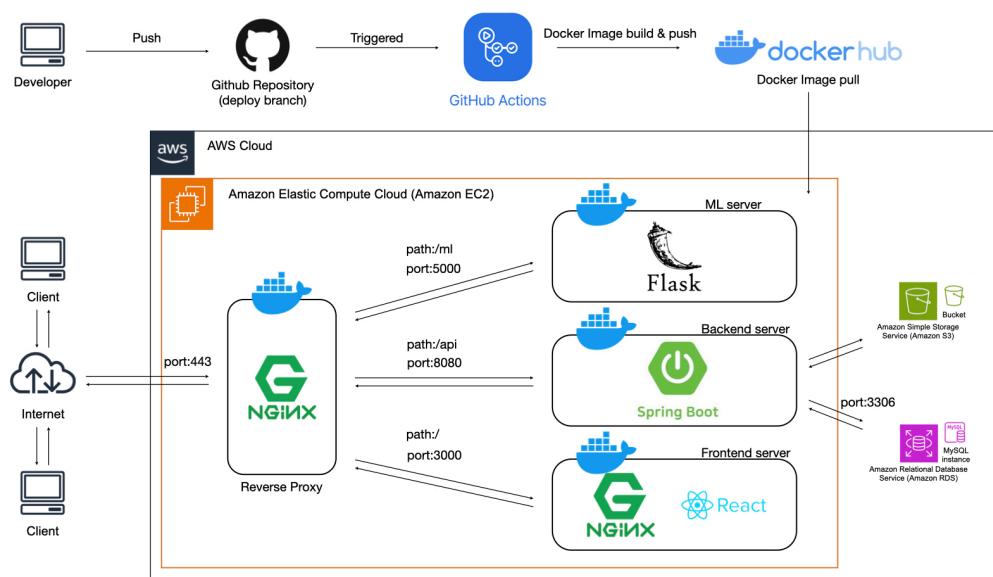
협업 과정에서 팀원들과 적극적으로 소통하며 의견을 모아 유저의 학습 의욕을 끌어올릴 수 있는 '포인트와 상점 시스템', '랜덤 문제 스트릭', '팀 단위 경쟁 시스템'과 같은 기능을 구상하여 서비스를 개발했습니다.

그 뒤, Docker와 Docker Compose, Github Action을 이용하여 AWS 클라우드에 CI/CD 파이프라인을 구축하여 현재까지 10개월 가량의 기간 일간 15~20명가량의 Active user를 가진 서비스를 제공하고 있습니다.

알고리즘 학습 서비스를 운영하며, 코드의 유지보수 비용을 낮추기 위해서 기존의 레이어드 아키텍처에 JPA Repository와 도메인 로직의 사이에 추상화를 적용하여 DB와 서비스 계층의 의존성을 풀어내는 Port-Adapter 패턴을 적용하는 리팩토링을 진행한 경험이 있습니다.

이 리팩토링을 통해 DB와 Mockito를 사용하지 않고 Spring과 JUnit만을 이용한 Mocking 단위 테스트 코드를 작성할 수 있게 코드를 개선한 경험이 있습니다.

## 서비스 구조도



# 알고리즘 학습 사이트 GPGT - 서비스 이미지



## Links

Site. <https://randps.kr>

Github Team. <https://github.com/GPGT-Algorithm-Study>

Github Back-End. <https://github.com/GPGT-Algorithm-Study/GPGT-Server>

Github Front-End. <https://github.com/GPGT-Algorithm-Study/GPGT-Client>

## CHERRY PICKER - 2023.03~2023.06



### 주요 기술

Back-End. Java, Spring Boot, JPA, Spring Security, Redis, nGrinder, Groovy

Front-End. Swift

Database. MySQL

DevOps. AWS(EC2, RDS, S3), Docker, Github Action

OS. Amazon Linux

### 서비스 소개

유저의 취향을 기반으로 음식점을 게임 형태로 추천해 주는 시스템을 제안 및 개발했습니다.

Cherry Picker는 초기 유저의 취향을 간단한 튜토리얼 게임을 통해 설정한 후 사용할 수 있습니다.

음식점을 추천받고 싶다면, Short-form video와 같이 등장하는 가게에 대해 오른쪽/왼쪽으로 화면을 스와이프해서 좋아요/싫어요를 하는 게임을 진행할 수 있습니다. 게임은 유저가 선호할만한 식당을 게임의 형태로 유저에게 추천해 줍니다.

결과적으로 추천받은 음식점을 저장할 수 있으며, 여러 가게에 대한 즐겨찾기 및 목적에 맞는 음식점을 확인할 수 있습니다.

기존에 진행한 게임이 누적되며 따라 유저의 취향을 점점 학습/분석하여 더 나은 추천 결과를 도출하고, 유저는 자신의 음식점 선호 패턴을 확인할 수 있습니다.

### 개발 과정

개인화 음식점 추천 기능을 제공하는 '광진구 음식점 추천 어플'의 Back-End 팀장을 맡아 Spring Boot와 JPA를 이용하여 서버를 개발했습니다. 사용자에게 적합한 음식점을 찾아내기 위해 30가지에 달하는 가게 분류 Tag와 Euclidean Distance를 이용한 음식점 추천 알고리즘을 설계 및 개발했습니다. 이를 통해 여러 가게의 기준 리뷰를 바탕으로 가게를 분류하고, 사용자의 선호를 파악하여 유사한 음식점을 추천해 주는 기능을 도입할 수 있었습니다.

추가로 서비스의 성능을 정량화하고 개선하기 위해 부하 테스트 오픈 소스 'nGrinder'를 이용해 유저가 앱을 사용하는 간단한 테스트 시나리오를 작성 후, 클라우드 환경의 서버에 가상 트래픽을 발생시켜 서버의 가용성을 테스트한 뒤, 중복된 조회로 인해 병목 현상을 유발하는 지점을 찾아 Redis를 이용한 캐싱을 적용해서 부하 테스트 결과와 클라우드상에서 기준보다 서버가 감당할 수 있는 가상 유저를 10%가량 늘리고 유저 경험을 개선한 경험이 있습니다.