


# 항상 고민하는 개발자 이경돈 입니다.

Backend

Java

Spring

## Contact

 Kyeongdon.lee97@gmail.com

# Profile

# 이 경 돈

Backend Developer

Email [Kyeongdon.lee97@gmail.com](mailto:Kyeongdon.lee97@gmail.com)

GitHub <https://github.com/wgrgwg>

Blog <https://wgrgwg.tistory.com/>

LinkedIn <https://www.linkedin.com/in/kyeongdon-lee/>

## About me

안녕하세요! 신입 백엔드 개발자 이경돈입니다.

- 문제 상황을 분석하고 해결 방안을 고민하는 과정을 좋아합니다.
- 새로운 기술을 두려워하지 않고 도전하며 배우는 과정을 중요하게 생각합니다.
- 학습한 내용을 꾸준히 기록합니다.
- 부족한 부분을 찾고 채워 나가는 지속적인 성장을 추구합니다.

## Education

2023.03 ~ 2025.02

충남대학교 컴퓨터공학과 졸업

2021.03 ~ 2023.02

한밭대학교 정보통신공학과

## Certificate

2025.06

정보처리기사

2023.06

SQLD

2026.01

OPlc IH

# Skill

---

## Confident



**Java**



**Spring Boot**



**Spring Data JPA**



**MySQL**



**Git/GitHub**

## Experienced



**Redis**



**Docker**



**Spring Security**



**Linux(Ubuntu)**

## Knowledgeable



**GitHub Actions**



**Caddy**



**GCP  
Compute Engine**

# Project – Somniverse

## Somniverse – 꿈 기록 서비스



개인 프로젝트 (2025.06~2025.09)

### 개요

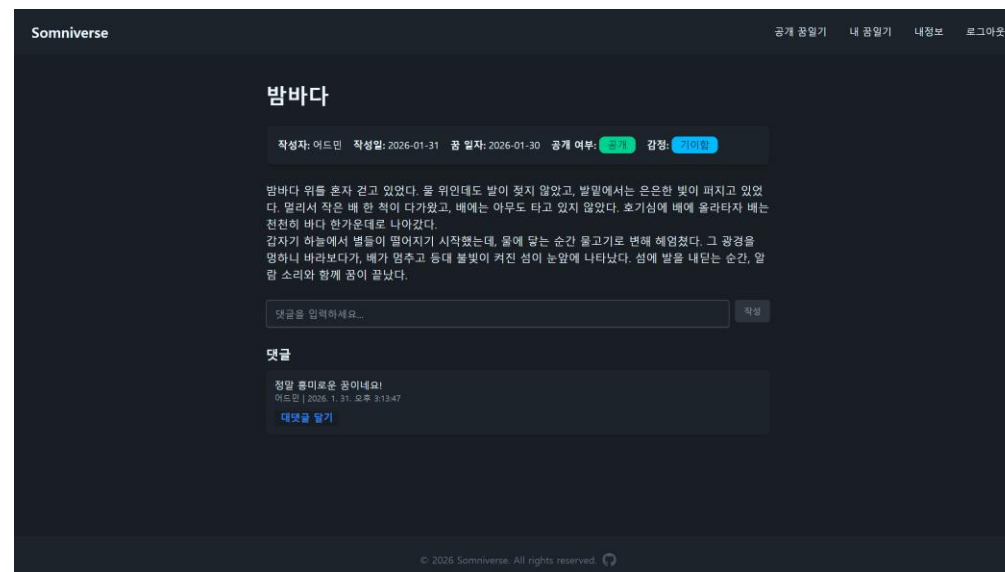
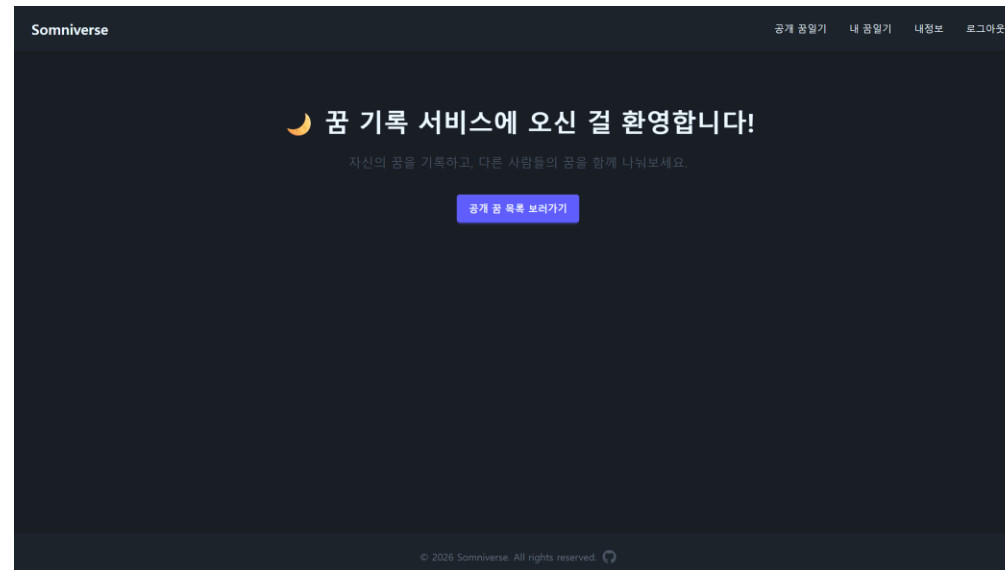
Somniverse는 사용자가 자신의 꿈을 기록하고 꿈에 담긴 감정을 분석하며 다른 사람들과 꿈을 공유하고 공감과 의견을 나눌 수 있는 꿈 기록 서비스입니다.

### 역할

- 서버 및 RESTful API 개발, 핵심 기능 구현, 테스트 작성
- JWT, OAuth2 인증 시스템 구현
- ERD 기반 DB 설계, 연관 로직 구현
- 배포 환경 구축 및 CI/CD 파이프라인 구성

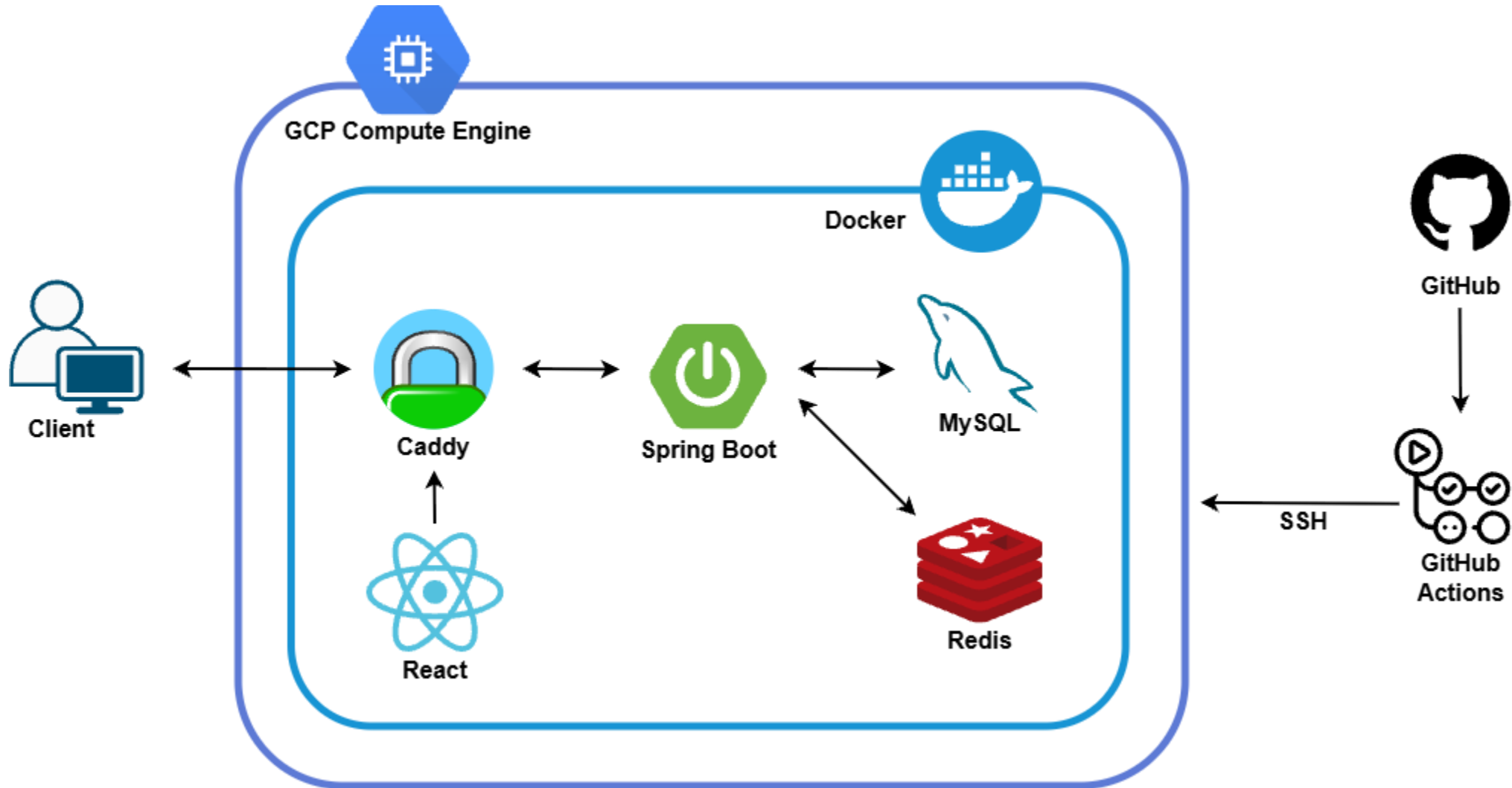
### 기술

- Java, Spring Boot, Spring Data JPA, Spring Security
- MySQL, Redis
- Docker, GCP Compute Engine, GitHub Actions, Caddy



# Project – Somniverse

## Architecture



# Project – Somniverse

## N+1 문제 해결 [Blog](#)

**문제** 게시글 목록 조회 시 게시글과 연관된 작성자 정보를 참조할 때마다 지연 로딩 쿼리가 추가로 발생했습니다. 이로 인해 단순 목록 조회임에도 총 N+3회의 쿼리가 실행되어 응답 속도가 저하되었습니다.

**해결** JPQL의 JOIN FETCH 구문을 사용하여 Dream 엔티티 조회 시 Member 엔티티를 즉시 함께 로딩하도록 쿼리를 튜닝하였습니다. 불필요한 추가 쿼리를 모두 제거하여 조회 쿼리를 감소시켰습니다. **(N+3회 → 3회)**

**문제** 댓글 목록 페이징 조회 시 대댓글 정보를 Fetch Join으로 가져올 경우 메모리 이슈 발생 위험을 확인했습니다. 컬렉션(대댓글) Fetch Join과 Paging을 함께 사용할 경우 Hibernate가 모든 데이터를 메모리에 로딩하여 처리하기 때문에 시스템 장애로 이어질 수 있었습니다.

**해결** 안정적인 Paging과 성능 확보를 위해 2단계 조회(List 조회 + Group By 집계) 방식으로 설계를 변경하였습니다. 부모 댓글 목록은 페이징 쿼리로 효율적으로 조회하고, 대댓글 개수는 IN절과 GROUP BY를 활용한 별도의 집계 쿼리로 한 번에 계산했습니다. 결과적으로 메모리 누수 문제를 방지하고 쿼리를 **2회로** 최소화 했습니다.

**문제** 게시글 삭제(soft delete) 시 연관된 댓글을 삭제하는 과정에서 UPDATE 쿼리가 N번 발생하는 문제가 발생했습니다. 반복문을 순회하며 각 댓글마다 개별적인 UPDATE 쿼리를 실행하여 트랜잭션 시간이 길어지는 문제가 있었습니다.

**해결** 컬렉션을 순회하지 않고 JPQL 벌크 연산을 도입하여 직접 쿼리를 수행했습니다. @Modifying 쿼리를 사용하여 조건에 맞는 모든 댓글의 상태(isDeleted)를 단일 UPDATE 쿼리로 일괄 변경했습니다. 이를 통해 데이터 개수에 비례해서 늘어나던 쿼리를 감소시켜 처리 성능을 개선했습니다. **(N+3회 → 2회)**

✓ Test Results	559 ms	✓ Tests passed: 1 of 1 test - 559 ms
✓ DreamDeleteQueryCountTest	559 ms	총 실행된 쿼리 수 = 103
✓ Dream 삭제 시 발생하는 쿼리 수	559 ms	2025-10-06T22:22:10.799+09:00
		877800 nanoseconds spent ac



✓ Test Results	460 ms	✓ Tests passed: 1 of 1 test - 460 ms
✓ DreamDeleteQueryCountTest	460 ms	총 실행된 쿼리 수 = 2
✓ Dream 삭제 시 발생하는 쿼리 수	460 ms	2025-10-06T22:42:01.144+09:00
		507700 nanoseconds spent ac

< Dream 삭제 로직 쿼리 개수 test >

# Project – Somniverse

## POST 요청 재시도 시 중복 생성 방지 [Blog](#)

**문제** 네트워크 지연이나 타임아웃 발생 시 클라이언트가 POST 요청을 재시도하는 과정에서 데이터 정합성 문제가 확인되었습니다. POST 메서드는 기본적으로 멍등하지 않기 때문에 서버가 첫 번째 요청을 처리했음에도 응답이 유실되어 재요청이 들어온 경우 동일한 리소스가 중복 생성되는 현상이 발생했습니다.

**해결** 요청의 고유성을 식별하고 처리 결과를 저장하는 멍등키 패턴을 도입하여 데이터 무결성을 확보했습니다.

- 클라이언트가 요청 헤더에 고유한 멍등키를 포함하여 보내도록 하고 Redis를 활용해 키와 최초 응답 데이터를 매핑하여 저장했습니다.
  - 요청 진입 시 Redis를 조회하여 동일한 멍등키 기록이 존재할 경우, 비즈니스 로직을 재실행하지 않고 저장된 최초 응답을 즉시 반환하도록 처리했습니다.
- 이를 통해 네트워크 이슈로 인한 **중복 요청 상황에서도 데이터 중복 생성을 차단**하여 시스템 신뢰성을 높였습니다.

## API 과부하 방지 및 시스템 안정성 확보 (Rate Limit) [Blog](#)

**문제** 특정 IP나 사용자가 짧은 시간동안 과도한 API 요청을 보낼 경우 서버 리소스가 고갈되어 전체 서비스 장애로 이어질 위험이 있었습니다. 특히 로그인 시도나 생성형 AI 호출과 같이 리소스 소모가 큰 로직의 경우 무제한 요청에 대해서 서비스 안정성이 취약했습니다.

**해결** 토큰 버킷 알고리즘을 구현한 Bucket4j 라이브러리를 도입하여, 클라이언트(IP 또는 User 식별자)로 단위 시간당 요청 허용량을 설정했습니다.

결과적으로 **허용량을 초과한 요청에 대해서는 즉시 429 Too Many Requests를 반환하여 비즈니스 로직 실행을 차단**함으로써 악의적인 트래픽이나 급격한 부하로부터 서버 리소스를 보호하고 서비스 안정성을 높였습니다.

# Project – WordWise

## WordWise – 영어 단어 학습 서비스



2025.01~2025.02

### 개요

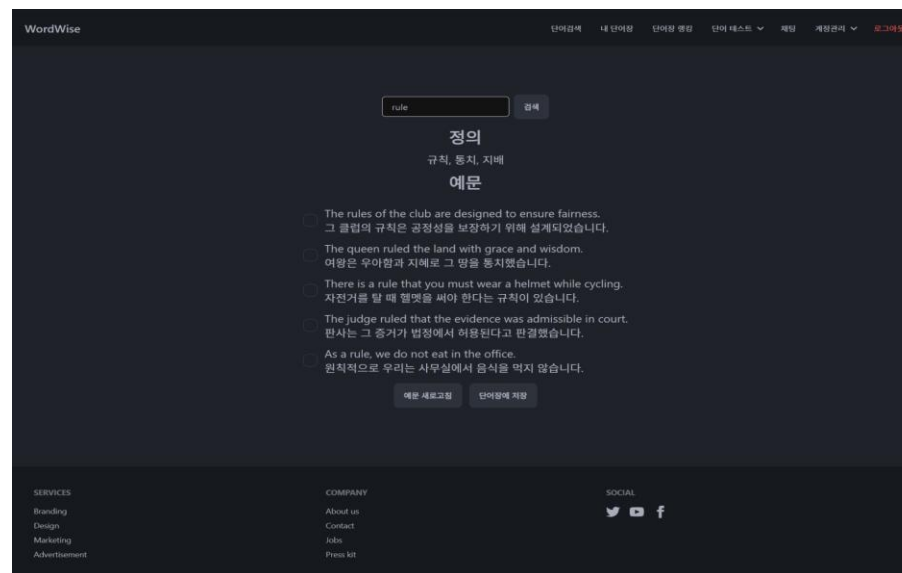
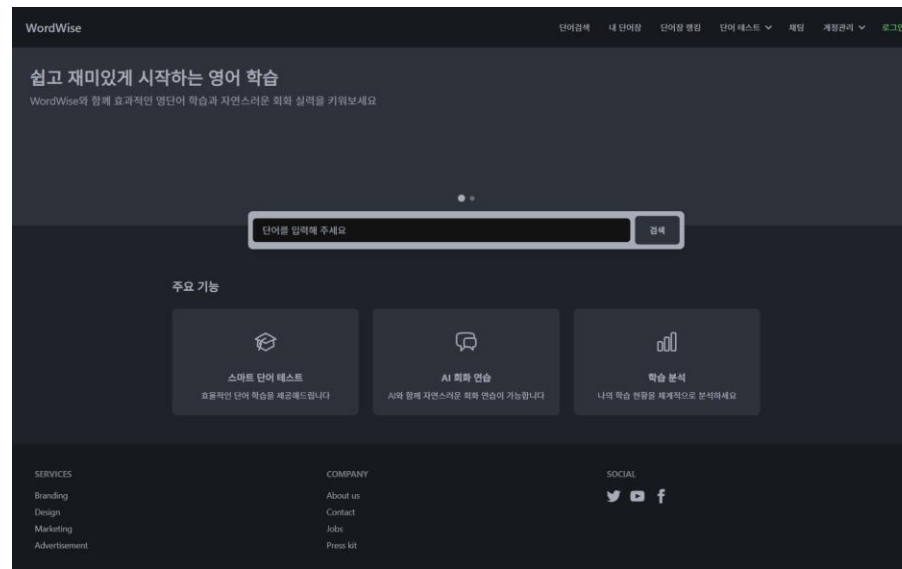
WordWise는 사용자가 단어의 뜻만 외우는 것이 아닌 예문을 생성해서 공부함으로써 단어의 여러 사용법을 익힐 수 있는 영어 단어 학습 서비스 입니다.

### 역할

- 서버 및 RESTful API 개발
- 생성형 AI API 활용 영어 단어 예문 생성 기능 구현
- ERD 기반 DB 설계, 연관 로직 구현

### 기술

- Java, Spring Boot, Spring Data JPA
- MariaDB
- Git 브랜치 전략에 따른 협업 개발





# 감사합니다

Contact



Kyeongdon.lee97@gmail.com