

김종조 신입

남, 2000 (25세)

이메일 desk1614@naver.com | 휴대폰 010-5270-8614 | 전화번호 010-5270-8614

주소 (02130) 서울 중랑구 봉우재로



학력

명지전문대학

대학교(2,3년) 졸업

전공

기계공학과

경력

-

희망연봉

회사내규에 따름

포트폴리오

<https://kimjon...>

나의 스킬

- MyBatis
- Java
- JavaScript
- HTML5
- CSS3
- Spring Boot
- JSP
- OracleDB
- Thymeleaf
- Spring Data JPA
- React
- QueryDSL
- JSON Web Token
- Spring Security
- jQuery
- MySQL
- MariaDB

학력 대학교(2,3년) 졸업

졸업 **명지전문대학(2·3년제)** 기계공학과

경력기술서

1. 프로젝트명: [최우수상] 수리·인테리어 전 과정을 한 번에 제공하는 통합 플랫폼 서비스

- 수행 기간 : 2025.11 ~ 2025.12 (약 2개월)

- 개발 인원 : 4명

- 주요 역할 : 서비스 핵심 로직(결제/정산/관리자) Full-Stack 개발

- 회원 서비스 : 커뮤니티(게시판), 자재 구매/결제, 장바구니, 대화형 견적 요청 기능 구현
- 관리자 서비스 : 전문가·업체 승인 프로세스, 이용 내역 관리, 정산 및 통계 대시보드 구현
- 설계 및 문서화 : 팀 공통 설계 기반 요구사항 정의서, 컴포넌트 명세서, ERD 작성 및 관리

- 업무 성과 :

1) 이커머스 결제 및 정산 시스템의 Full-Stack 구현

- 구매 프로세스 및 결제 연동 : 자재 옵션 선택부터 장바구니 담기, Toss Payments API를 활용한 실시간 결제까지의 전 과정을 단독 구현

- 결제 데이터 검증 로직 : 보안을 위해 프론트엔드 결제 요청 전, 백엔드와 DB의 상품 데이터를 대조하여 최종 금액을 재검증함으로써 결제 데이터의 신뢰성 확보

- 정산 자동화 로직 설계 : 판매 금액 및 거래 데이터를 기반으로 전문가와 판매 업체의 정산 금액을 자동 집계하는 백엔드 로직을 설계

2) 관리자 승인 워크플로우 및 운영 통계 대시보드 구축

- 상태 패턴 기반 승인 로직 : 일반 회원의 전문가 전환 및 입점 신청 시 발생하는 '대기-승인-반려'의 상태 변화를 백엔드 상태 패턴 기반으로 설계하여 유지보수성 향상

- 데이터 시각화 및 UI 구현 : Chart.js 라이브러리를 활용하여 수익 추이 및 회원 수 등 주요 운영 지표를 시각화한 관리자 전용 통계 대시보드 UI 구현

3) 커뮤니티 활성화 및 대화형 견적 요청 시스템 개발

- 커뮤니티 및 이미지 업로드 : 수리 노하우 공유를 위한 게시판 CRUD를 구현하고, 다중 이미지 업로드와 비동기 통신을 적용하여 페이지 새로고침 없이 빠른 화면 응답성 확보

- 선택형 단계별 견적 시스템 : 복잡한 견적 과정을 직접 입력하는 대신 정해진 답변을 버튼으로 선택하는 대화형 UI로 구현하여 사용자의 접근성 향상

2. 프로젝트명: [최우수상] 정신건강 자가테스트와 병원 예약·진료를 제공하는 정신건강 플랫폼 서비스

- 수행 기간 : 2025.08 ~ 2025.10 (약 2개월)

- 개발 인원 : 4명

- 주요 역할 : 의료 데이터 보안 및 사용자 계정 통합 Full-Stack 개발

1) 회원 서비스 : OAuth 2.0 기반 소셜 로그인 및 이메일 계정 병합 로직 구현

2) 병원 관리자 서비스 : 진료 상태별 진단서 수정/조회 권한 제어 및 환자 이력 관리 개발

3) 사이트 관리자 서비스 : 병원 입점 심사 프로세스 및 고유 코드 자동 발급 시스템 구축

- 업무 성과 :

1) 중복 가입 방지 및 소셜 계정 통합 시스템 구축

- 소셜 로그인 API 연동 : OAuth 2.0 기반의 소셜 로그인 기능을 구현하여 사용자 접근성을 높이고 간편 가입 환경 조성

- 계정 병합 프로세스 설계 : 소셜 로그인 시 기존 이메일 가입 이력이 존재할 경우, 중복 계정 생성을 방지하고 기존 계정에 인증 수단을 안전하게 연결하는 로직 구현

2) 진료 상태 기반의 의료 데이터 접근 권한 관리

- 데이터 무결성 및 수정 제어 : '진료 중' 상태에서만 진단서 작성이 가능하도록 제한하고, 진료 완료 후에는 데이터 임의 변경이 불가능하도록 차단하여 의료 기록의 신뢰성 확보

- 조건부 조회 권한 설계 : 보안을 위해 진료 중일 때만 환자의 전체 이력을 공개하고, 진료 외 상황에서는 관리자 본인이 작성한 문서만 열람 가능하도록 데이터 접근 제어 적용

3) 병원 입점 심사 워크플로우 및 자동화 로직 개발

- 입점 승인 및 코드 발급 : 병원 입점 신청 현황 조회부터 심사, 승인까지의 운영 프로세스를 구축하고, 승인 시 병원별 고유 식별 코드를 자동 생성하는 시스템 구현

- Google SMTP 기반 이메일 알림 : Google SMTP 서버를 연동한 이메일 발송 로직을 구현하여, 승인 완료 시 병원 고유 코드를 신청자에게 실시간으로 전송

- 회원 활동 상태 관리 : 전체 회원 정보를 조회하고 활동 상태를 실시간으로 제어할 수 있는 관리자 전용 기능 구축

3. 프로젝트명: 여행지 추천과 여행 물품 구매·대여를 제공하는 여행 플랫폼 서비스

- 수행 기간 : 2024.05 ~ 2024.06 (약 1개월)

- 개발 인원 : 6명

- 주요 역할 : WebSocket 기반 실시간 소통 및 커머스 핵심 기능 Full-Stack 개발

1) 커뮤니케이션 서비스 : Spring WebSocket을 활용한 1:1 채팅 및 실시간 알림 시스템 구축

2) 인증 및 계정 서비스 : 소셜 로그인 연동 및 사용자 계정 관리(가입/찾기) 기능 개발

3) 이커머스 서비스 : 장바구니 및 관심상품 저장 등 사용자 편의 중심의 쇼핑 기능 구현

- 업무 성과 :

1) WebSocket 기반 실시간 1:1 채팅 및 알림 시스템 구축

- 자동 핸드셰이크 및 상시 연결 : 사용자가 서비스 로그인 시 별도의 액션 없이 Spring WebSocket 서버와 자동으로 핸드셰이크를 수행하도록 설계하여 실시간 소통 환경 구축

- 실시간 읽음 처리 및 카운트 : 메시지 송수신 시 읽음 상태 정보를 업데이트하는 로직을 구현하고, 카카오톡 방식의 '읽지 않은 메시지 수' 실시간 표시 기능 제공

- 관리자 전용 소통 채널 : 일반 회원과 관리자 간의 1:1 매칭 채팅 기능을 통해 실시간 고객 응대 및 즉각적인 피드백 지원

2) 사용자 편의 중심의 계정 관리 및 소셜 인증 시스템 구현

- 소셜 로그인 API 연동 : OAuth 2.0 기반 소셜 로그인 기능을 제공하여 번거로운 절차 없는 간편 가입 및 서비스 진입 속도 향상

- 통합 계정 관리 서비스 : 회원가입, 아이디 찾기, 비밀번호 재설정 등 필수적인 사용자 관리 기능을 구현하여 유연한 계정 복구 환경 지원

3) 쇼핑 편의 기능(장바구니 및 관심상품)개발

- 장바구니 데이터 관리 : 사용자가 선택한 여러 상품 정보를 보관하고, 수량 변경이나 선택 삭제 등이 실시간으로 반영되는 장바구니 기능 구현

- 관심상품 저장 시스템 : 다시 보고 싶은 상품을 즐겨찾기로 등록하여 마이페이지에서 언제든지 확인할 수 있도록 하는 개인화 저장 로직 구축

4. 프로젝트명: 씨앗 구매와 커뮤니티 기능을 제공하는 씨앗 플랫폼 서비스

- 수행 기간 : 2024.04 ~ 2024.05 (약 1개월)

- 개발 인원 : 3명

- 주요 역할 : 이커머스 핵심 인터페이스 설계 및 고객 지원 서비스 Full-Stack 개발

1) 쇼핑 서비스 : 상품 카탈로그 및 장바구니 시스템 단독 설계 및 전체 로직 구현

2) 보안 및 인증 서비스 : Google SMTP 연동 계정 찾기 및 비밀번호 암호화 처리

3) 관리 및 상담 서비스 : 상품 재고 관리자 페이지 및 페이지 처리가 적용된 문의 게시판 구축

- 업무 성과 :

1) 이커머스 쇼핑 프로세스 단독 설계 및 풀스택 개발

- 쇼핑 로직 전담 구현 : 상품 목록 조회부터 상세 정보 제공, 장바구니 담기까지 결제 이전의 핵심 쇼핑 프로세스 전체를 독자적으로 설계 및 구축

- 장바구니 상태 관리 : 사용자가 선택한 상품 정보를 DB에 영속적으로 보관하고, 수량 변경 및 선택 삭제가 실시간으로 반영되는 장바구니 시스템 구현

- 상품 접근성 최적화 : 대량의 씨앗 상품을 효율적으로 탐색할 수 있도록 카테고리별 목록 조회 및 상세 페이지 연동 로직 구축

2) 보안 가이드라인을 준수한 계정 관리 및 인증 시스템 구축

- Spring Security 기반 보안 강화 : Spring Security에서 제공하는 BCryptPasswordEncoder를 적용하여 비밀번호를 단방향 해시로 암호화함으로써 데이터베이스 보안성 확보

- Google SMTP 기반 계정 복구 : 외부 메일 서버를 연동한 이메일 인증을 통해 아이디 찾기 및 비밀번호 재설정 기능을 구현하여 계정 접근성 향상

3) 플랫폼 운영 및 사용자 상담 시스템 개발

- 실시간 상품 관리 도구 : 신규 상품 등록, 가격 수정 및 실시간 품질/판매 상태 전환 기능을 통해 운영자가 상품 현황을 즉각 제어할 수 있는 관리 환경 구축
- 사용자 전용 상담 게시판 : 상품 및 서비스 이용 중 발생하는 궁금증을 자유롭게 남길 수 있는 사용자 전용 상담 커뮤니티 공간 제공

경험/활동/교육

2025.07 ~ 2025.12

KOSTA 한국소프트웨어기술진흥협회

교육이수내역

Java를 기반으로 웹 애플리케이션의 서버 로직을 구현하며 객체지향 개념과 백엔드 개발의 기본 구조를 학습했습니다. 비즈니스 로직을 중심으로 안정적인 서버 코드를 작성하는 경험을 쌓았습니다.

MariaDB를 사용해 SQL을 작성하고 CRUD 기능을 구현했으며, 데이터 구조를 고려한 테이블 설계와 쿼리 작성 경험을 통해 데이터베이스와 서버 간의 연동 흐름을 이해했습니다.

HTML, CSS, JavaScript와 jQuery를 활용해 웹 화면을 구성하고 사용자 인터랙션을 처리했으며, 이후 React를 학습하며 컴포넌트 기반 UI 구조와 상태 관리 개념을 이해하고 프론트엔드와 백엔드를 분리한 개발 환경을 경험했습니다.

Servlet과 JSP를 통해 웹의 요청-응답 구조를 이해한 뒤, Spring Framework와 Spring Boot를 활용해 MVC 패턴 기반의 웹 애플리케이션을 개발했습니다. 이 과정에서 계층 분리를 통해 유지보수성과 확장성을 고려한 설계를 경험했습니다.

MyBatis를 사용해 SQL 중심의 데이터 접근 방식을 학습했고, 이후 JPA와 QueryDSL을 활용해 객체 중심의 데이터 처리와 동적 쿼리 작성 경험을 쌓으며 상황에 맞는 기술 선택의 중요성을 이해했습니다.

Git과 GitHub를 활용해 소스 코드 버전 관리를 진행하며 협업 과정에서의 개발 흐름을 익혔고, AWS EC2를 사용해 애플리케이션을 배포하며 실제 서비스 환경에서의 운영 경험을 쌓았습니다.

2024.01 ~ 2024.06

KH정보교육원

교육이수내역

Java 기반 웹 개발 교육 과정을 통해 백엔드 개발의 기초부터 웹 애플리케이션 구현까지의 전반적인 흐름을 학습했습니다. Java 문법과 객체지향 개념을 바탕으로 서버 로직을 구현하며, 웹 서비스가 어떻게 동작하는지 이해하는 데 집중했습니다.

OracleDB를 사용해 SQL을 작성하고 CRUD 기능을 구현했으며, 테이블 설계와 데이터 구조를 고려한 개발 경험을 쌓았습니다. 이를 통해 데이터베이스와 서버 로직이 유기적으로 연결되는 과정을 이해하게 되었습니다.

HTML, CSS, JavaScript를 활용해 기본적인 웹 화면을 구성하고, 사용자 입력에 따라 동적으로 동작하는 화면을 구현했습니다. 프론트엔드와 백엔드가 어떻게 연동되는지 경험하며 전체 서비스 흐름을 이해할 수 있었습니다.

Servlet과 JSP를 통해 요청과 응답 처리 방식을 학습한 뒤, Thymeleaf와 Spring Framework, Spring Boot를 활용해 MVC 패턴 기반의 웹 애플리케이션을 개발했습니다. 컨트롤러, 서비스, 데이터 접근 계층을 분리하여 유지보수성을 고려한 구조로 개발하는 경험을 했습니다.

또한 MyBatis를 사용해 SQL과 Java 객체를 매핑하며 데이터 접근 로직을 관리했고, Git과 GitHub를 활용한 버전 관리와 협업 과정을 통해 팀 프로젝트에서의 개발 흐름을 익혔습니다.

자격/어학/수상

| | |
|---------|--|
| 2025.12 | 웹 애플리케이션 개발 프로젝트 최우수상 2회 수상 (세미·파이널, Full-Stack 개발자 양성과정) 한국소프트웨어기술진흥협회 |
| 2025.04 | SQL개발자(SQLD자격) 최종합격 한국데이터베이스진흥센터 |
| 2022.11 | 정보처리산업기사 최종합격 한국산업인력공단 |
| 2021.10 | 컴퓨터활용능력2급 최종합격 대한상공회의소 |

포트폴리오 및 기타문서

포트폴리오 <https://kimjongjo.github.io/about-me/>

자기소개서

정해진 설계를 넘어, 논리적 해답을 찾아가는 개발자로

기계공학을 전공하며 공학적 사고와 설계 순서의 기본기를 익혔지만, 대부분의 전공 수업은 정해진 결과를 기준으로 설계를 구현하는 과정이 중심이었습니다. 반면 자동화시스템제어실습 수업에서 진행한 아두이노 코딩 과제는 동작 오류의 원인을 직접 찾고 코드를 수정하며 해결해 나가는 과정이 요구되었습니다. 카운트다운 기능이 정상적으로 동작하지 않을 때마다 코드와 회로를 하나씩 점검하며 수정했고, 작은 변경에도 즉각적으로 결과가 달라지는 경험을 통해 문제 해결 과정 자체에 흥미를 느끼게 되었습니다. 이 경험을 계기로 설계 결과를 따라가는 역할보다, 논리적으로 문제를 분석하고 개선해 나가는 개발 업무가 제 성향과 더 잘 맞는다는 판단을 하게 되었습니다.

개발에 대한 관심을 단순한 흥미로 끝내지 않기 위해 독학으로 기초 학습을 이어가며 정보처리산업기사와 SQLD 자격증을 취득했습니다. 그러나 개념을 이해하는 것과 이를 실제 개발에 적용하는 것 사이에는 분명한 차이가 있다는 한계도 느끼게 되었습니다. 이에 팀 단위로 기능을 구현하고 협업 경험을 쌓기 위해 국비 지원 개발자 과정에 지원했습니다.

국비 교육 과정에서는 비전공자로서 부족한 부분을 보완하고자 문제를 체계적으로 정리하는 학습 방식을 유지했습니다. 오류가 발생하면 에러 메시지와 로그를 기반으로 원인을 분석하고, 해결 과정을 기록해 동일한 문제가 반복되지 않도록 했습니다. 이러한 습관은 프로젝트를 안정적으로 진행하는 데에도 도움이 되었습니다.

또한 팀 프로젝트에서는 요구사항 해석에 차이가 발생했을 때, 기능을 실제로 사용한다고 가정하고 사용자 입장에서 혼란스럽거나 불편할 수 있는 부분을 먼저 고민했습니다. 기능 흐름을 따라가며 입력 방식과 처리 순서에서 애매한 지점을 정리해 팀원들과 공유했고, 이를 기준으로 요구사항을 다시 맞추며 개발 방향을 조율했습니다. 그 결과 개발 기준을 명확히 할 수 있었고, 불필요한 수정 없이 일정에 맞춰 프로젝트를 완성할 수 있었습니다.

이러한 경험을 통해 저는 단순히 기능을 구현하는 데 그치지 않고, 문제의 원인을 먼저 고민하며 팀과 함께 더 나은 방향을 만들어가는 개발자로 성장하고자 합니다.

자연된 타임라인 앞에서 '내 기능'보다 '우리 프로젝트'를 선택하다

저의 장점은 적극적으로 개선점을 찾아내고, 이를 팀과 공유하며 협력하는 자세입니다. 프로젝트 설계 단계에서 회원 테이블의 로그인 방식을 일반 로그인과 소셜 로그인을 모두 포함한 단일 컬럼으로 설계한 것을 보고, ERD Cloud를 활용해 로그인 방식을 별도 테이블로 분리하고 회원과 로그인 방식이 명확히 관계를 맺도록 개선한 모델링을 직접 제안했습니다. 또한 팀원의 코드에서 반복되는 로직을 발견했을 때, 별도 메서드로 분리하여 재사용성을 높인 코드를 작성해 제안한 경험도 있습니다. 이러한 경험을 통해, 단순히 문제를 지적하는 데 그치지 않고 실질적인 개선안을 찾아 팀 성과로 연결할 수 있는 능력이 제 강점임을 확인했습니다.

하지만 때로는 더 나은 구조를 만들고 싶다는 의욕이 너무 앞서서, 팀 전체의 일정이나 동료의 상황을 충분히 살피지 못하는 실수를 하기도 했습니다. 프로젝트 중반에 도메인별로 필터링 처리가 미흡한 부분을 발견하고, 보안상 꼭 고쳐야 한다는 생각에 시큐리티 담당 팀원에게 로직 수정을 요청했습니다.

당시에는 시스템의 완성도를 높이는 게 최선이라고 생각했지만, 제 요청 때문에 그 팀원의 계획했던 개인 일정이 하루이틀씩 뒤로 밀리는 걸 옆에서 보게 되었습니다. 제 제안이 팀원에게 업무 부담을 주고 전체 타임라인을 혼드는 걸 보면서 아차 싶었고, 큰 책임감을 느꼈습니다.

이 상황을 해결하기 위해 저는 팀의 우선순위를 다시 따져보았습니다. 당시 제가 맡았던 '1:1문의'나 '정보 상세' 기능보다 팀원이 담당한 '마이페이지 공구 대여' 로직이 프로젝트의 핵심이라고 판단했습니다. 그래서 제 기능 구현을 뒤로 미루거나 과감히 덜어내고, 그 시간만큼 팀원의 업무를 도와 함께 마이페이지 로직을 개발하며 팀의 전체적인 마감 속도를 맞추 수 있었습니다. 이 경험을 통해 아무리 좋은 제안이라도 팀의 상황을 배려하지 못하면 독이 될 수 있다는 것과, 때로는 팀을 위해 내 것을 내려놓는 유연함이 필요하다는 것을 배웠습니다.

이후로는 제안하기 전, '이 작업이 동료의 흐름에 어떤 영향을 줄까?'를 먼저 고민하게 되었습니다. 지금은 팀의 컨디션과 기한을 먼저 살피고, 필요한 부분부터 단계적으로 개선해 나가는 유연함을 갖추려 노력하고 있습니다. 입사 후에도 기술적 욕심과 팀의 목표 사이에서 균형을 잘 잡으며, 동료들과 발맞춰 안정적으로 성과를 내는 개발자가 되겠습니다.

핵심역량

[Spring Boot: 구조가 탄탄하고 보안이 확실한 서버를 만듭니다]

Java와 Spring Boot로 서버를 개발할 때, 나중에 코드를 수정하거나 기능을 추가하기 편하도록 계층(Controller - Service - Repository)을 명확히 나누는 것을 원칙으로 합니다. 특히 보안은 서비스의 기본이라고 생각해서 Spring Security와 JWT, OAuth2를 활용한 인증 시스템을 꼼꼼히 구축해왔습니다. 단순한 로그인을 넘어 이메일 인증 같은 실무적인 로직까지 직접 처리해 보며 안전한 서버를 만드는 법을 익혔습니다.

[데이터베이스 : 꼼꼼한 모델링부터 효율적인 쿼리 작성까지]

Oracle, MySQL, MariaDB 등 다양한 환경에서 비즈니스 로직에 딱 맞는 데이터 구조를 설계하는 데 익숙합니다. 직접 ERD를 그리며 데이터가 꼬이지 않도록 무결성을 챙기고, MyBatis와 JPA(QueryDSL)를 상황에 맞춰 활용하며 복잡한 동적 쿼리도 효율적으로 작성합니다. 단순히 데이터가 나오는 것에 만족하지 않고, 인덱스나 쿼리 튜닝을 고민하며 시스템의 전체적인 성능을 높이는 데 집중합니다.

[연동 및 통신 : 프론트엔드와 매끄럽게 소통하는 데이터 흐름]

프론트엔드와 백엔드 사이의 안정적인 소통이 서비스의 질을 결정한다고 생각합니다. React 환경에서 Axios로 데이터를 주고받을 때, 서버에 잘못된 값이 들어오지 않도록 유효성 검사를 꼼꼼하게 설계하여 데이터의 신뢰도를 높이는 데 신경을 썼습니다. 또한 Jotai를 활용해 사용자 데이터를 관리하고 React Router로 페이지 이동을 처리하는 등 프론트엔드의 동작 원리를 직접 경험해 보았습니다. 덕분에 화면에서 어떤 데이터가 어떤 시점에 필요한지 정확히 파악하여, 효율적인 API 구조를 설계하고 유연하게 개발할 자신이 있습니다.

[운영 및 배포 : 서비스가 구동되는 전체 흐름을 경험했습니다]

로컬 환경에서 코딩을 마치는 것에 그치지 않고, 작성한 기능이 실제 서버 환경에서 어떻게 작동하는지 확인하기 위해 직접 배포를 진행해 보았습니다. Gradle 도구를 활용해 프로젝트를 빌드하고, Apache Tomcat 서버에 올려서 제가 짠 코드가 실제 서비스로 이어지는 기본적인 과정을 경험했습니다. 아직 인프라 지식이 깊지는 않지만, 내 컴퓨터 밖의 실행 환경에서도 코드가 잘 돌아가는지 끝까지 확인해 본 경험이 있기에, 새로운 운영 환경을 마주하더라도 빠르게 적응하고 부딪히며 배울 준비가 되어 있습니다.