

# 갈피: 당신의 독서 여정을 위한 혁신적인 플랫폼

갈피(Galphi)는 독서 애호가들을 위한 혁신적인 **도서 추천 사이트**입니다. Java와 Oracle DB를 기반으로 한 이 플랫폼은 단순한 도서 목록 이상의 경험을 제공합니다. 세련된 디자인과 직관적인 인터페이스로 독서의 즐거움을 배가시키는 갈피는 **책갈피**처럼 여러분의 독서 경험을 기록하고, '**갈피를 잡다**'라는 표현처럼 **읽을 책의 정보를** 제공합니다. 로그인, 도서 검색, 상세 정보 열람, 실시간 댓글 및 평점 시스템 등 다양한 기능으로 새로운 독서 문화의 장을 열어갑니다. 갈피와 함께 더욱 풍성하고 의미 있는 독서 여정을 시작해보세요.



## 죽음의 역사

앤드루 도이그 | 브론스테인 | 출판일: 2023-02-22

평점: 8.9 ★★★★★☆

😊 완벽해요

THIS MORTAL COIL

# 갈피의 핵심 기능

## 1 풍부한 도서 정보

Oracle DB와 연동하여 방대한 양의 **도서 정보를 제공합니다**. 책의 제목, 저자, 출판사뿐만 아니라 줄거리, 리뷰, 관련 도서 추천까지 한눈에 볼 수 있습니다.

## 2 실시간 소통

사용자들이 자유롭게 의견을 나누고 평가할 수 있는 실시간 **댓글 시스템을 제공합니다**. 다른 독자들의 생각을 들어보고 자신의 의견을 공유하며 독서 경험을 풍성하게 만들 수 있습니다.

## 3 다양한 추천 시스템

사용자의 독서 경험이 바탕이 된 평점을 분석하여 베스트, 추천, 테마별 **다양한 도서 리스트를 제공합니다**. 새로운 책을 발견하고 독서의 지평을 넓힐 수 있는 기회를 제공합니다.

## 4 직관적인 인터페이스

**Bootstrap5를 활용한 세련된 디자인**과 사용자 친화적인 인터페이스로 누구나 쉽게 사용할 수 있습니다. 책을 찾고, 정보를 확인하고, 의견을 나누는 모든 과정이 매끄럽게 이루어집니다.

# 갈피의 개발 환경

## 백엔드 기술

갈피는 **Java 11**을 기반으로 하며, 강력하고 안정적인 **Spring Boot 프레임워크**를 사용하여 개발되었습니다. 이를 통해 높은 성능과 확장성을 확보했습니다. 데이터베이스로는 **Oracle DB**를 사용하여 대용량 데이터 처리와 복잡한 쿼리 실행을 효율적으로 수행합니다.

## 프론트엔드 기술

사용자 인터페이스는 **Bootstrap 5**를 활용하여 구축되었습니다. 이를 통해 반응형 디자인과 모던한 UI 컴포넌트를 쉽게 구현할 수 있었습니다. 또한, **JavaScript**를 활용하여 동적인 기능과 부드러운 사용자 경험을 제공합니다.

## 개발 도구 및 방법론

**Spring Tool Suite 4(STS4)**를 사용하여 효율적인 코딩과 디버깅을 수행했습니다. **애자일 방법론**을 채택하여 빠른 피드백과 지속적인 개선 **TDD(Test Driven Development)** 방식을 통해 안정적이고 유지보수가 용이한 코드를 작성했습니다.



# 갈피의 주요 기능 요구사항

## 1 정보 제공

갈피는 사용자에게 풍부한 책 정보를 제공합니다. 책의 기본 정보뿐만 아니라 상세한 줄거리, 저자 소개, 출판사 정보 등을 포함합니다. 또한, 다른 사용자들의 댓글과 평점을 통해 책에 대한 다양한 의견을 확인할 수 있습니다.

## 2 사용자 관리

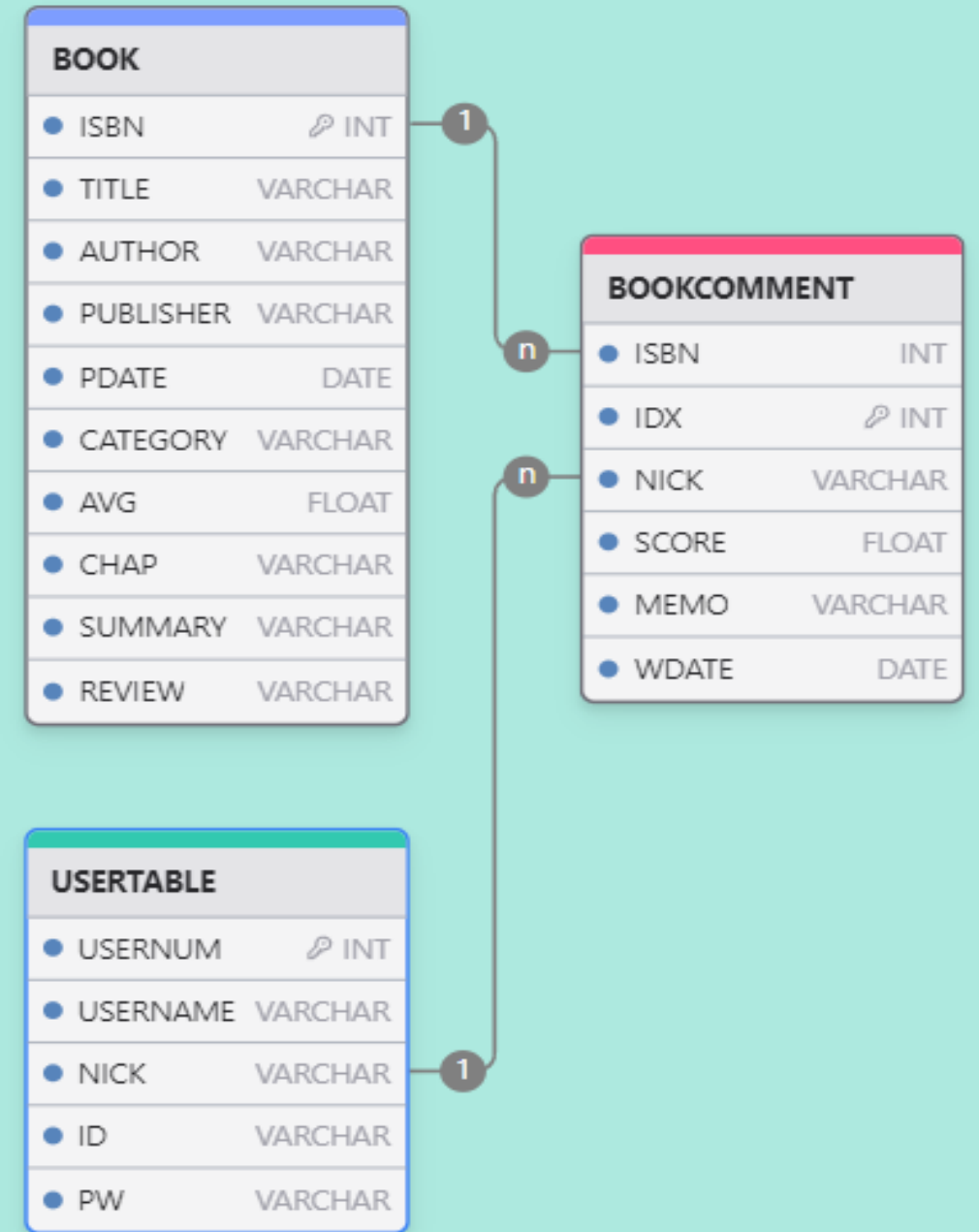
안전하고 편리한 로그인 및 회원가입 기능을 제공합니다. 사용자는 개인 아이디를 만들고 회원 고유의 닉네임을 설정할 수 있으며, 이를 통해 다른 사용자들이 의견을 남기는 댓글 교류의 장에 합류할 수 있습니다. 로그인한 사용자에게는 댓글 작성, 수정, 삭제 등의 추가 기능이 제공됩니다.

## 3 테마 / 레이아웃 개선

갈피가 여름 테마로 돌아왔습니다. 기존의 가을 감성이 묻어나는 테마색을 변경하여 현재 계절은 여름에 맞게 디자인하였고, 좀 더 풍부한 인덱스 페이지와 직관적이고 사용자 친화적인 페이지 레이아웃으로 재구성하고자 노력했습니다. 메인 색인 파스텔톤 블루와 함께 더 시원해진 갈피를 즐겨보세요.

# 갈피의 데이터베이스 설계

테이블명	주요 필드	설명
BOOK	ISBN, TITLE, AUTHOR, AVG, REVIEW	도서 정보를 저장하는 테이블
BOOKCOMMENT	ISBN, IDX, NICK, SCORE, MEMO, WDATE	사용자의 도서 리뷰와 평점을 저장하는 테이블
USERTABLE	USERNUM, USERNAME, NICK, ID, PW	사용자 정보를 기억하는 테이블





# 갈피의 주요 페이지 소개



## 인덱스 페이지

갈피의 첫 인상을 결정짓는 **메인 페이지**입니다. **최신 도서 추천, 인기 도서 목록, 평점 베스트셀러** 등 다양한 정보를 한눈에 볼 수 있도록 구성되어 있습니다. 사용자의 관심사에 따른 테마별 추천도 이 페이지에서 확인할 수 있습니다.



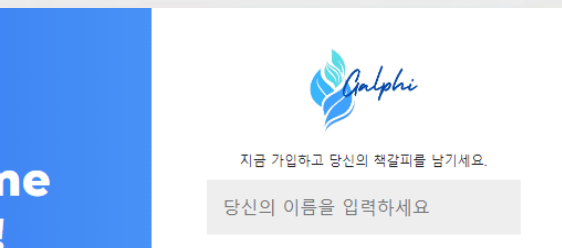
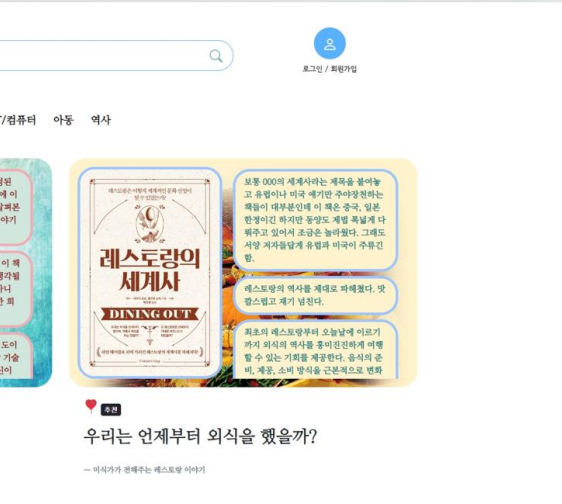
## 도서 리스트 페이지

사용자가 원하는 조건으로 도서를 검색하고 결과를 확인할 수 있는 **페이지**입니다. 다양한 검색 옵션과 페이지 기능을 제공하여 원하는 책을 쉽게 찾을 수 있습니다. 각 도서의 간단한 정보와 평점을 미리 볼 수 있어 선택에 도움을 줍니다.



## 도서 상세 페이지

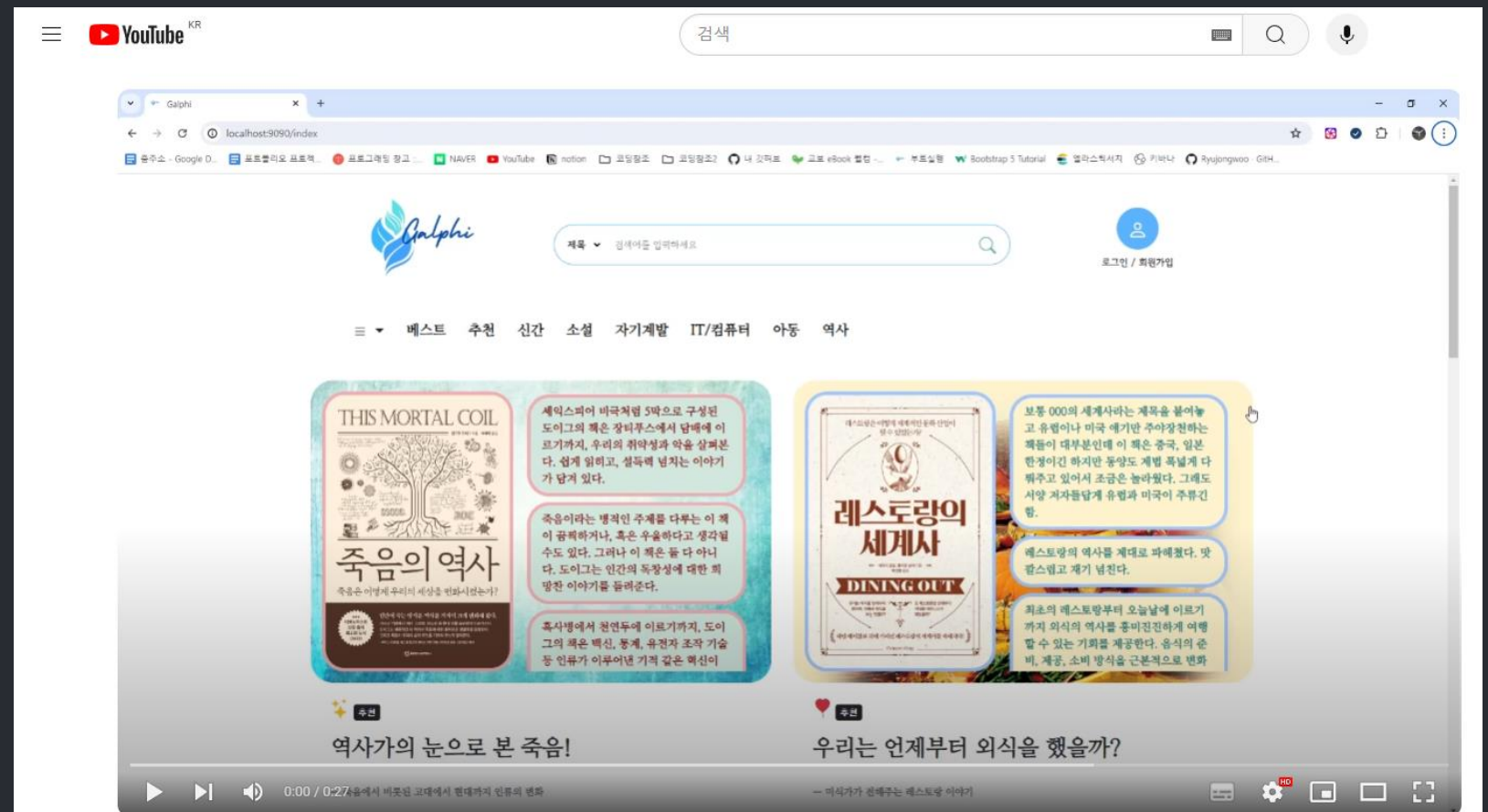
선택한 도서의 모든 정보를 확인할 수 있는 **페이지**입니다. 책 표지, 저자 정보, 출판사, ISBN 등의 기본 정보부터 상세한 줄거리, 독자 리뷰, 관련 도서 추천까지 제공합니다. 이 페이지에서 **직접 리뷰를 작성하고 평점**을 남길 수 있습니다.



# 갈피의 시연 영상

## 유튜브 링크

<https://youtu.be/BM5jBhdn9n4>



리메이크갈피 시연영상(준비중)

# 프로젝트 개선 사항 및 성과

## 개선 영역

레이아웃

테마

기능 구현

개발 도구

## 이전 상태

기본적인 구조

갈색 계통 색상

기본적인 CRUD  
기능

이클립스, 스프링

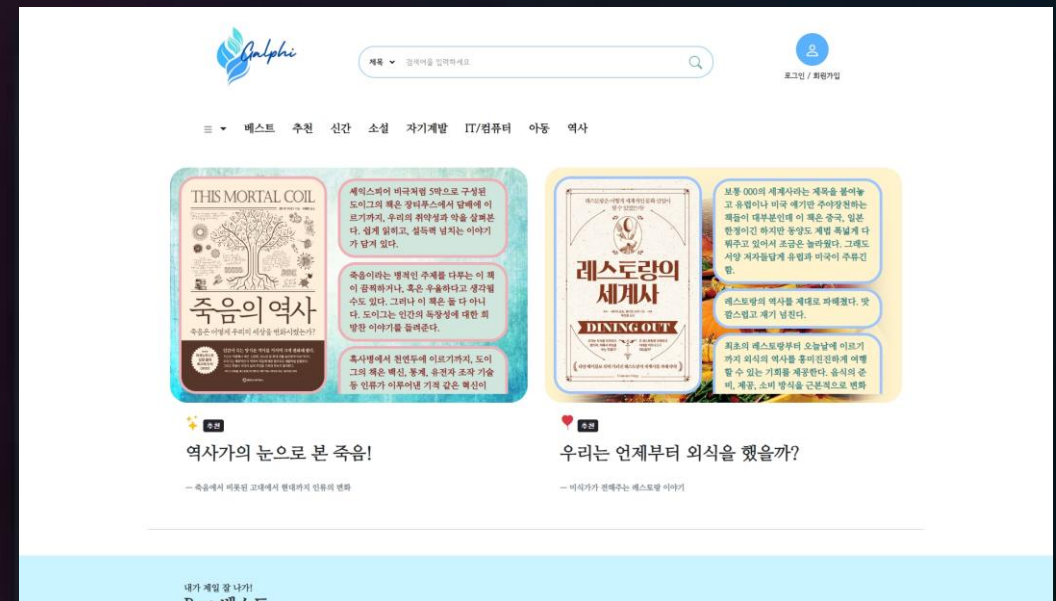
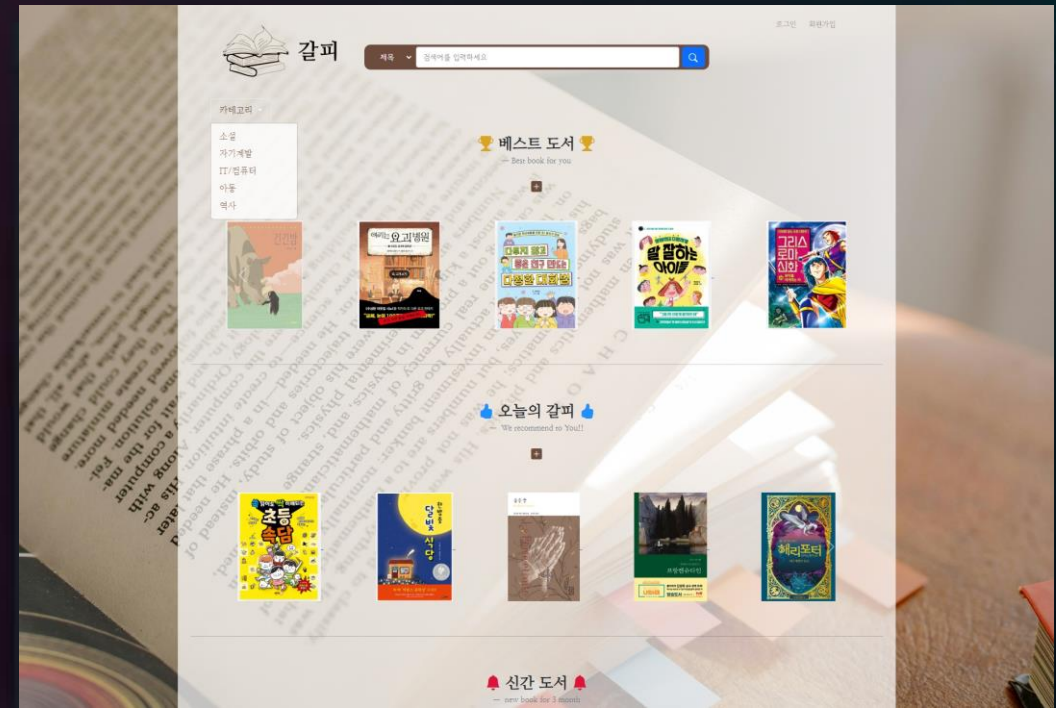
## 개선 후 상태

더 풍부하고 사용자  
친화적인 디자인

여름 테마와 세련된  
색상 팔레트 적용

검색, 페이징 등  
고급 기능 추가

스프링부트로 전환  
개발 효율성 향상





# 스프링부트 개발 中 학습 경험

스프링부트를 활용한 프로젝트 개발 과정에서 다양한 기술적 도전과 학습 기회를 경험했습니다. 이 과정은 단순히 코드를 작성하는 것을 넘어, 깊이 있는 문제 해결 능력과 기술적 이해도를 향상시키는 귀중한 시간이었습니다. 카멜 표기법과 스네이크 표기법의 차이, Entity 객체 생성 시 발생한 문제, JPA의 와일드카드 사용법 등 다양한 기술적 난관을 극복하며 실질적인 개발 경험을 쌓았습니다. 이러한 경험들은 개발자로서의 성장과 더불어 프로젝트의 전반적인 이해도를 높이는 데 크게 기여했습니다.



# 카멜 표기법과 스네이크 표기법의 차이

## 카멜 표기법 (CamelCase)

Java에서 주로 사용되는 표기법으로, 각 단어의 첫 글자를 대문자로 표기합니다.

예를 들어, 'pDate'와 같은 형태입니다. 이 표기법은 가독성이 좋고 Java 개발자들 사이에서 널리 사용됩니다.

## 스네이크 표기법 (Snake\_case)

주로 데이터베이스에서 사용되는 표기법으로, 단어 사이를 언더스코어 (\_)로 구분합니다.

예를 들어, 'P\_DATE'와 같은 형태입니다. 이 표기법은 SQL에서 자주 사용되며, 데이터베이스 필드명에 적합합니다.

## 스프링부트에서의 차이점

스프링부트는 기본적으로 스네이크 표기법을 사용하여 데이터베이스 필드와 매핑합니다. 따라서 'pDate'와 같은 카멜 표기법 변수는 'P\_DATE'로 인식되어 오류가 발생할 수 있습니다. 이를 해결하기 위해 변수명을 'pdate'로 수정하여 문제를 해결했습니다.

# Entity 객체 생성 시 ID 필드 문제 해결

## 1 문제 발견

Entity 객체 생성 시 데이터베이스의 ID 필드를 불러오지 못하는 현상이 발생했습니다. 이는 JPA의 @GeneratedValue 어노테이션과 관련된 문제였습니다.

## 2 원인 분석

@GeneratedValue 어노테이션의 strategy 설정이 데이터베이스의 ID 생성 방식과 일치하지 않아 발생한 문제였습니다. 이를 통해 데이터베이스 view의 구조와 sequence에 대한 이해의 필요성을 깨달았습니다.

## 3 해결 과정

@GeneratedValue 어노테이션의 strategy에 대해 심층적으로 공부하고, 데이터베이스의 ID 생성 방식에 맞게 설정을 변경했습니다. 이 과정에서 JPA와 데이터베이스 간의 상호작용에 대한 이해도가 크게 향상되었습니다.

## 4 결과 및 학습

설정 변경 후 Entity 객체가 정상적으로 ID 필드를 인식하여 문제가 해결되었습니다. 이 경험을 통해 ORM 기술과 데이터베이스 설계의 중요성을 더욱 깊이 이해하게 되었습니다.



# JPA 와일드카드 사용법 습득

## 1 Like 메소드 사용 시 발생한 오류

JPA의 Like 메소드를 사용하여 검색 기능을 구현하던 중, 검색어를 인식하지 못하는 오류가 발생했습니다. 이는 SQL 문법과 JPA 문법의 차이에서 비롯된 문제였습니다.

## 2 SQL vs JPA 와일드카드 차이점

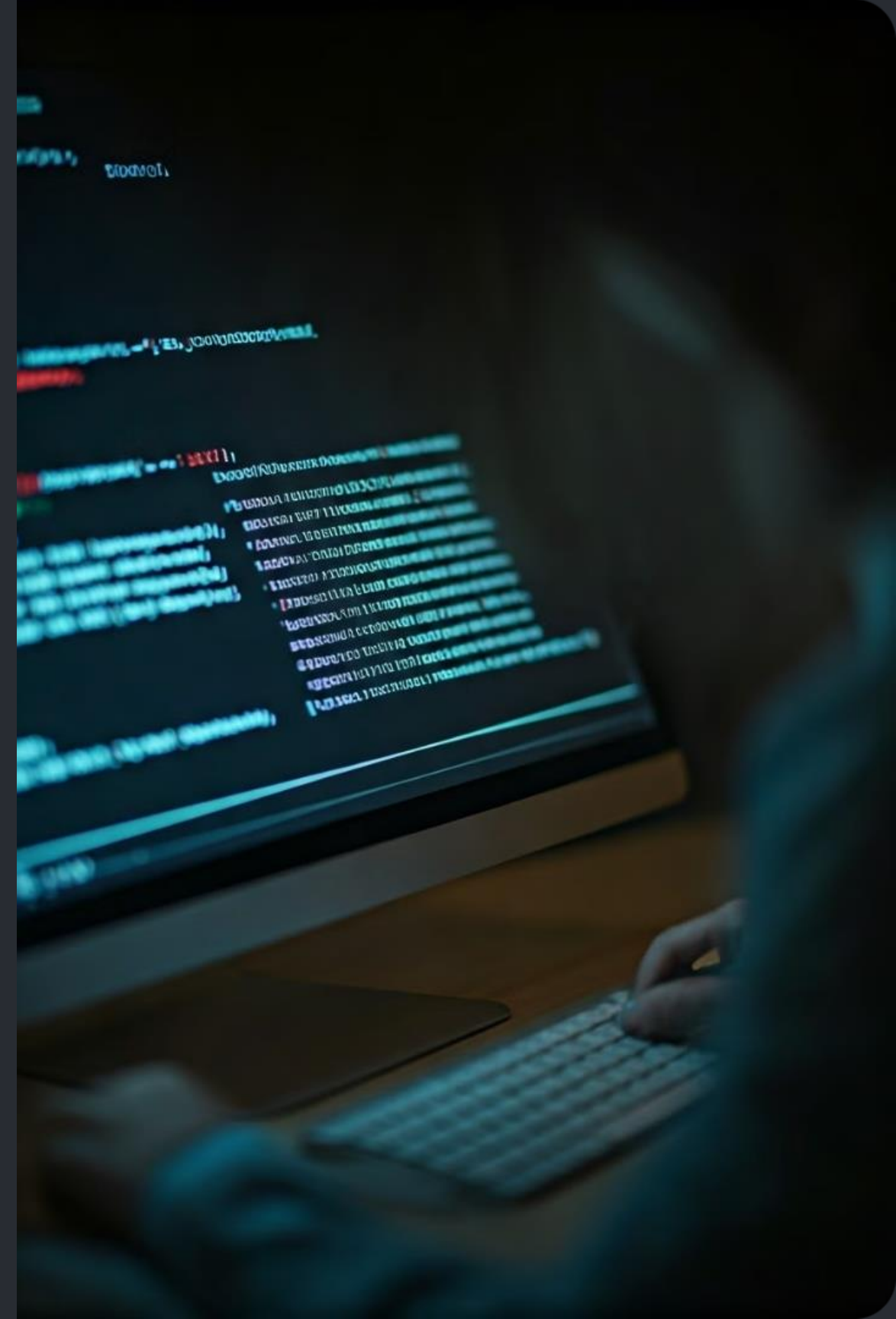
일반 SQL에서는 '%검색어%' 형식으로 포함 검색을 구현하지만, JPA의 Like 메소드는 '검색어' 형식으로 작성해야 합니다. 이러한 차이점을 인지하고 코드를 수정하여 문제를 해결했습니다.

## 3 JPA의 추가 메소드 학습

이 과정에서 JPA의 StartingWith, EndingWith, Containing 메소드에 대해 추가로 학습했습니다. 이러한 메소드들은 더 세밀한 검색 조건을 구현할 때 유용하게 사용될 수 있습니다.

## 4 검색 기능 최적화

학습한 내용을 바탕으로 검색 기능을 최적화했습니다. 이를 통해 사용자 경험을 향상시키고, 데이터베이스 쿼리의 효율성을 개선할 수 있었습니다.





# 회원가입 기능 개선 방안

## 한글 ID 방지

현재 시스템에서는 한글 ID 사용이 가능한 상태입니다. 이를 개선하기 위해 정규표현식을 활용하여 영문과 숫자만 허용하는 ID 유효성 검사를 구현할 계획입니다. 이를 통해 시스템의 일관성을 유지하고 잠재적인 문제를 예방할 수 있습니다.

## 비밀번호 유효성 강화

현재의 비밀번호 정책을 더욱 강화할 필요가 있습니다. 최소 8자 이상, 대소문자, 숫자, 특수문자를 포함하도록 요구사항을 설정하고, 이를 프론트엔드와 백엔드 모두에서 검증하는 로직을 추가할 예정입니다. 이를 통해 사용자 계정의 보안성을 크게 향상시킬 수 있습니다.

## 이메일 인증 시스템

회원가입 시 이메일 인증 절차를 추가하여 실제 사용자 확인 과정을 거치도록 할 계획입니다. 이는 스프링의 JavaMailSender를 활용하여 구현할 수 있으며, 이를 통해 스팸 계정 생성을 방지하고 시스템의 신뢰성을 높일 수 있습니다.



# 상세 페이지 및 리스트 버튼 개선 계획

## 댓글 입력 창 레이아웃 개선

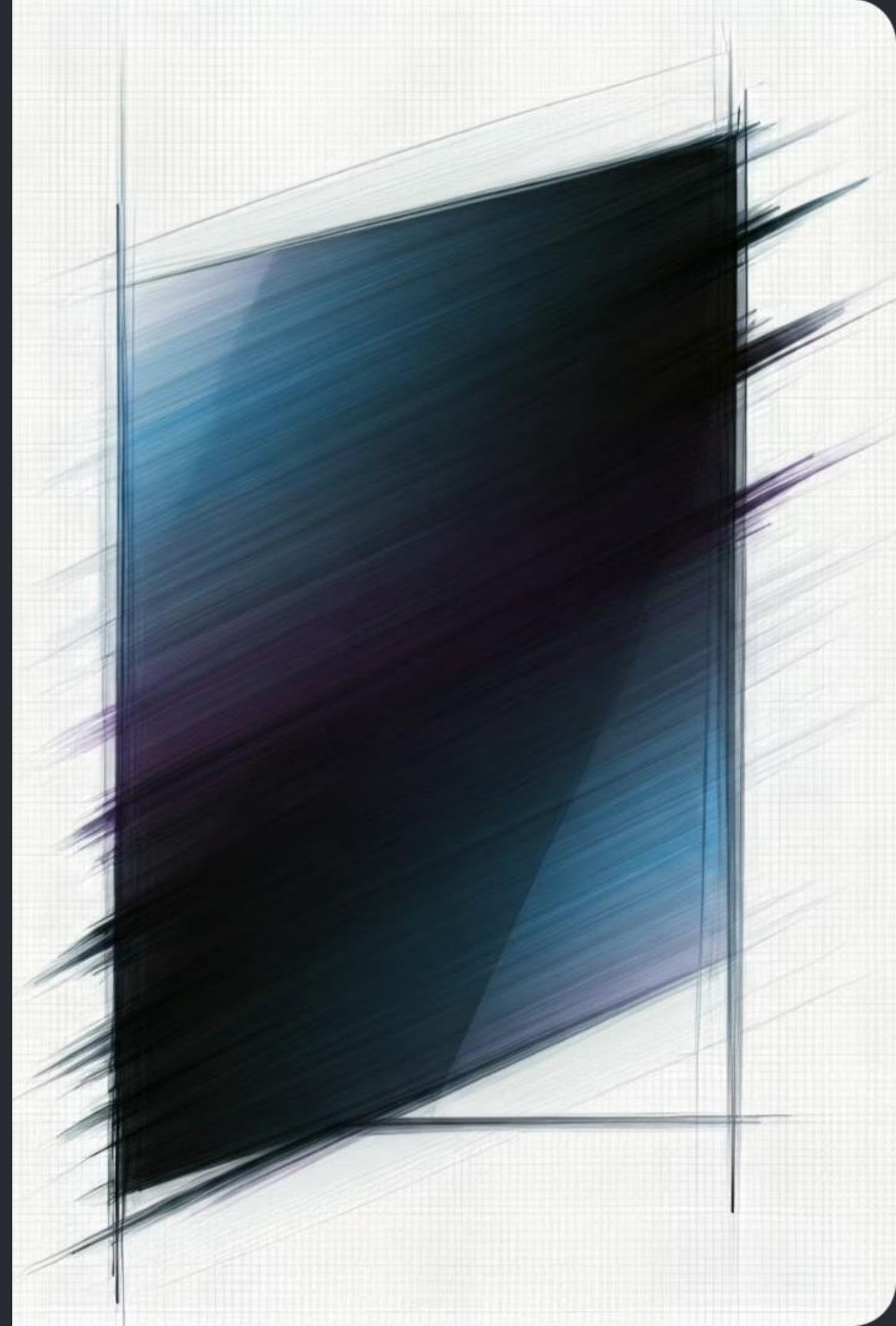
- 1 현재의 정적인 댓글 입력 창을 동적 버튼화하여 사용자 경험을 개선할 계획입니다. JavaScript를 활용하여 댓글 작성 버튼 클릭 시 입력 폼이 나타나도록 구현하고, 입력 완료 후 자동으로 접히는 기능을 추가할 예정입니다.

## 리스트 버튼 직관성 향상

- 2현재 페이지를 더 명확하게 표시하기 위해 CSS를 활용하여 활성 페이지 버튼의 스타일을 차별화할 계획입니다. 또한, 페이지 네비게이션 기능을 개선하여 사용자가 현재 위치를 쉽게 파악할 수 있도록 할 예정입니다.

## 반응형 디자인 적용

- 3모바일 환경에서의 사용성을 개선하기 위해 Bootstrap과 같은 프레임워크를 활용하여 반응형 디자인을 적용할 계획입니다. 이를 통해 다양한 디바이스에서 일관된 사용자 경험을 제공할 수 있습니다.



# 감사합니다.

발표자 & 개발자 김현담

깃허브: <https://github.com/hn922003>

블로그: <https://programstorage.tistory.com>

## 갈피(Galphi) - 도서 추천 사이트

