

프레임워크 DB 연동

이름	정민규
학번	2401110265
소속	인공지능소프트웨어과
과목	프레임워크실습
작성일	2025-12-08

1. 주제 및 개요

본 프로젝트는 1학년 때 제작했던 "버디(반려견) 홈페이지"를 기반으로 하여, **Spring Boot + JPA + Spring Security**를 사용해 **회원가입, 로그인/로그아웃, 댓글 기능이 연동되는 웹 애플리케이션**으로 확장 개발한 것이다.

기존에 제작했던 정적 HTML 페이지는 View 레이어로 재사용하고, Spring Boot MVC 구조 안에서 Controller·Service·Repository 계층을 새로 구축하여 동적 웹 애플리케이션으로 확장하였다.본 프로젝트에서 구현된 핵심 기능은 다음과 같다.

- 회원가입 (JPA 저장)
- 로그인 / 로그아웃 (Spring Security 기반)
- 역할(Role) 기반 권한 처리 (ADMIN / USER)
- 페이지별 댓글 작성·조회·삭제 기능
- 관리자(admin) 페이지에서 회원 목록 및 특정 회원의 댓글 조회
- 마이페이지(로그인 사용자 정보 조회)
- Info / Photo / Video 페이지 모두에서 댓글 가능

2. 시스템 설계도 (구조도)

아래는 본 프로젝트의 전체 구조를 나타낸 시스템 설계도이다.



설계 방식은 **Spring MVC 아키텍처 패턴**을 기반으로 하고 있으며,
Controller → Service → Repository → Database 흐름으로 요청이 처리된다.

특히 댓글 기능은 `pageld(INFO, PHOTO_IN, PHOTO_OUT, VIDEO_IN, VIDEO_OUT)`를 기준으로 페이지별 댓글을 구분한다.

3. DB 설계 (ER-Diagram)

본 프로젝트에서 사용하는 주요 엔티티는 **Member**와 **Comment** 두 가지이다.

✓ (1) Member 엔티티

필드명	타입	설명
id	Long(PK)	회원 고유 ID
name	String	사용자 이름
email	String	로그인 ID(중복 불가)
password	String	비밀번호(BCrypt 암호화)
role	Enum(Role.USER / ADMIN)	권한
createdAt	LocalDateTime	생성 일시

✓ (2) Comment 엔티티

필드명	타입	설명
id	Long(PK)	댓글 고유 ID
content	String	댓글 내용
pageId	String	어느 페이지에서 달린 댓글인지
createdAt	LocalDateTime	작성 시각
author	Member(FK)	작성 회원

Member (1)	-----	(N) Comment
id PK		id PK
email		content
password		pageId
name		createdAt
role		member_id (FK)

회원 1명은 여러 개의 댓글을 작성할 수 있으므로 **1:N 관계**이다.

DB 구현은 다음과 같이 하였다:

```
CREATE DATABASE frameworkgimal CHARACTER SET utf8mb4;

CREATE USER 'framework'@'localhost' IDENTIFIED BY 'framework1234';

GRANT ALL PRIVILEGES ON frameworkgimal.* TO 'framework'@'localhost';
```

4. 시스템 흐름도

✓ (1) 회원가입 흐름

사용자 → /signup → AuthController
→ MemberService.register()
→ MemberRepository.save()
→ DB 저장 → 로그인 페이지로 이동

✓ (2) 로그인 흐름

사용자 → /login
Spring Security → CustomUserDetailsService
→ MemberRepository.findByEmail()
→ 인증 성공 → 세션 생성 → Home으로 이동

✓ (3) 댓글 작성 흐름

사용자(로그인 상태)
↓
페이지에서 댓글 작성 (content + pagelId)
↓
CommentController.addComment()
↓

CommentService.saveComment()

↓

CommentRepository.save()

↓

DB 저장 후 해당 페이지로 redirect

작성자 또는 관리자만 댓글 삭제 가능

Pageld에 따라 댓글을 분기해서 불러옴

관리자에서 회원 목록 → 특정 회원 댓글로 이동함

✓ (4) 관리자(Admin) 기능 흐름

ADMIN 로그인

↓

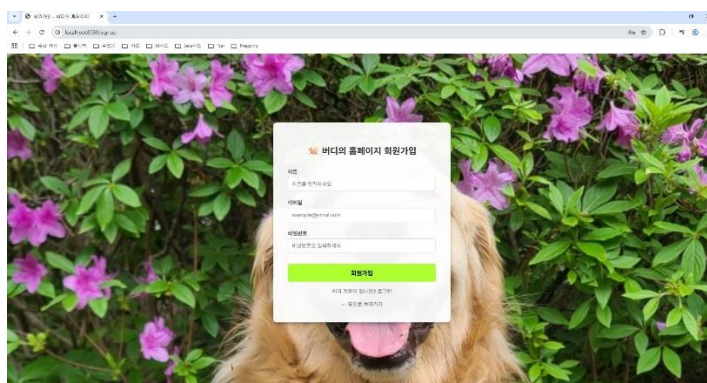
/admin/members → 전체 회원 목록 조회

/admin/members/{id} → 특정 회원 댓글 목록 확인

5. 주요 기능 설명

✓ (1) 회원가입 기능

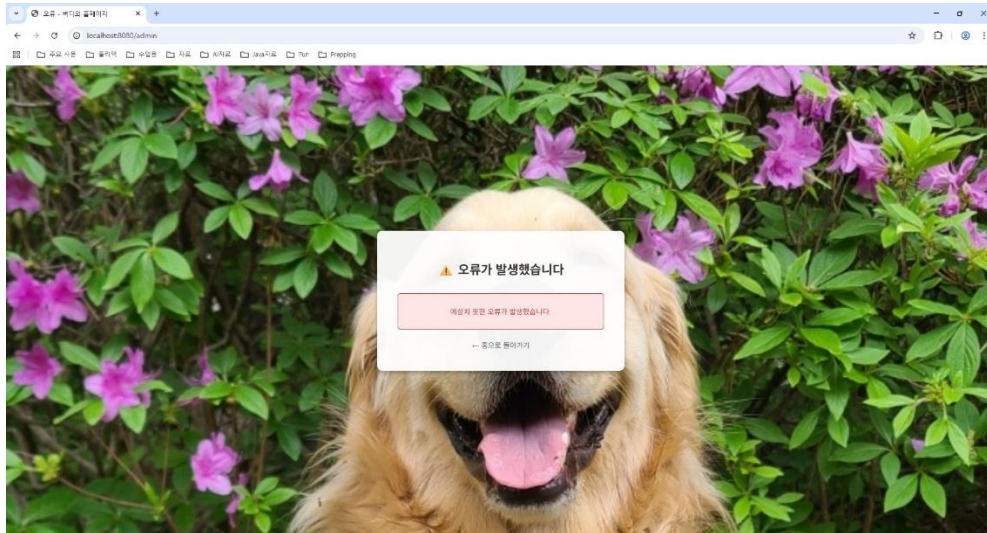
회원가입 페이지에서 이름, 이메일, 비밀번호를 입력하면 비밀번호는 BCrypt로 암호화되어 DB에 저장된다.



✓ (2) 로그인 / 로그아웃 기능

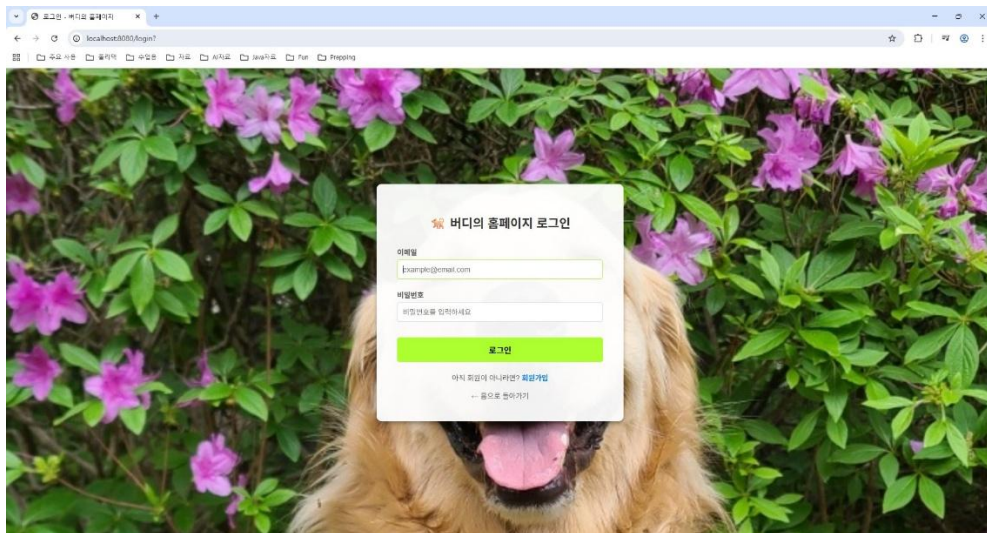
Spring Security 기반의 세션 인증 방식.

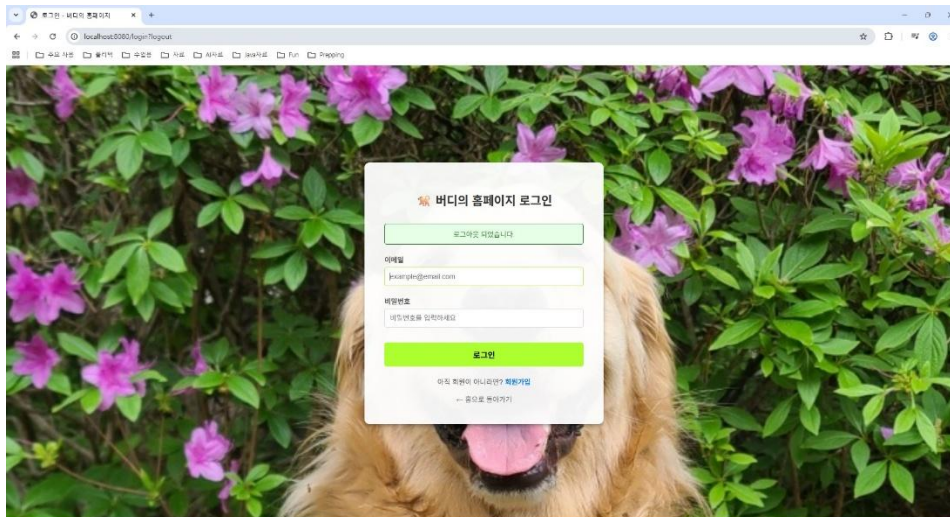
- /admin/** → ADMIN만 허용



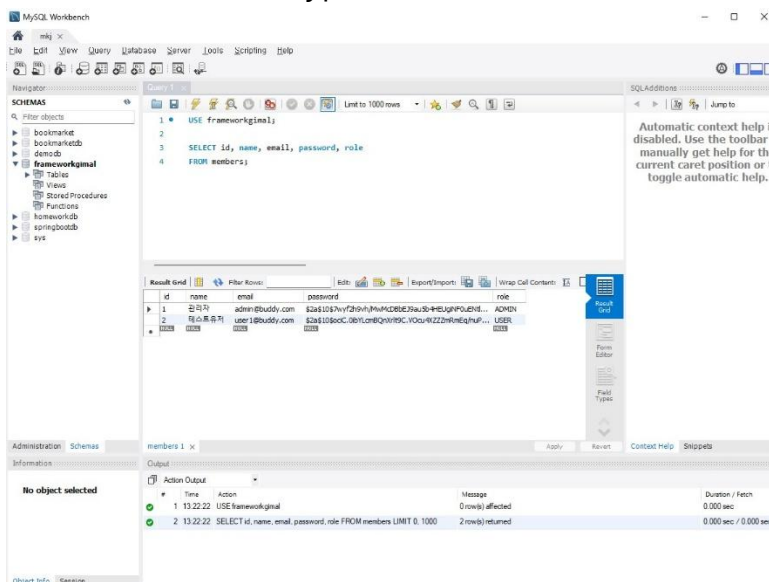
(일반회원으로 <http://localhost:8080/admin/members/> 접속시의 오류)

- 로그인, 로그아웃 URL





- 비밀번호 암호화(BCrypt)



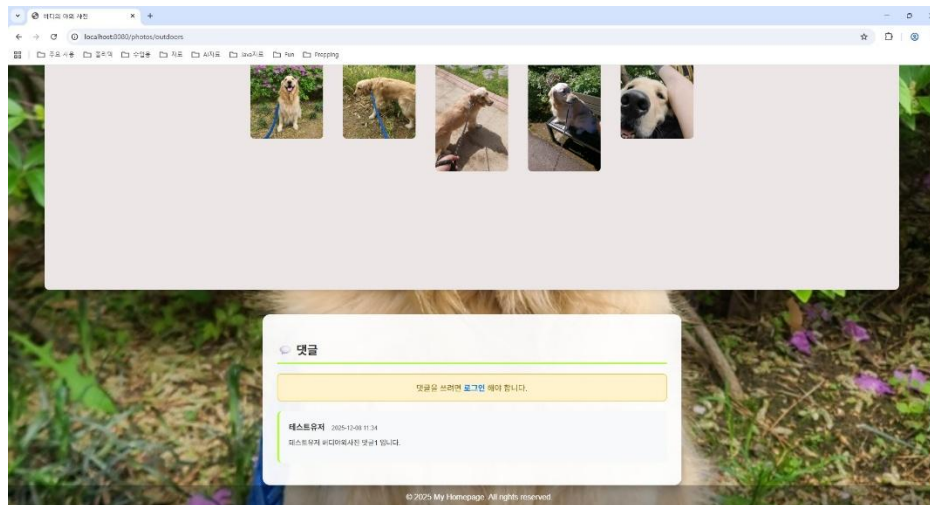
- UserDetailsService가 Member 엔티티를 인증에 사용
- 작성자/관리자 댓글 삭제 규칙(SpEL + Thymeleaf 조건)

우측 상단에 로그인 상태가 항상 표시되며, 로그인 상태에 따라 버튼이 자동 변경된다.

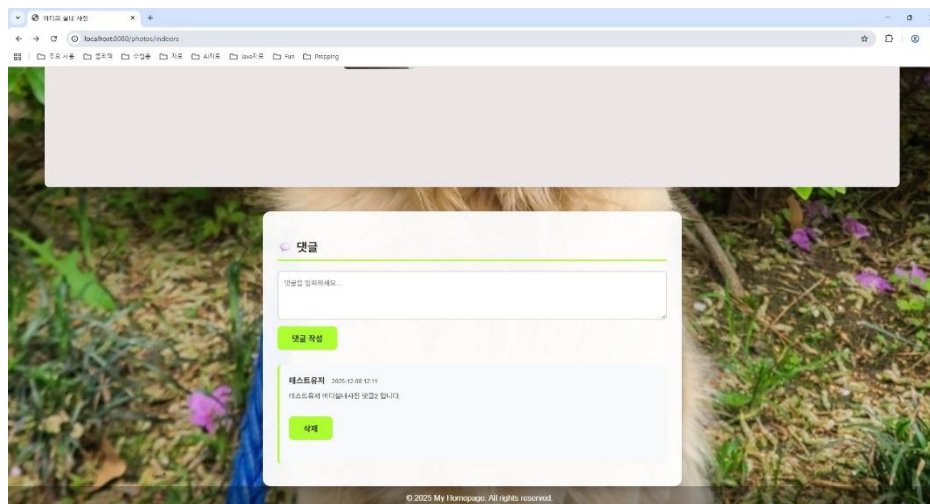
✓ (3) 댓글 기능 (페이지별 구분)

모든 주요 페이지(Info, Photo, Video)에 댓글 작성이 가능하며 pageId를 기반으로 각각 분리 저장된다.

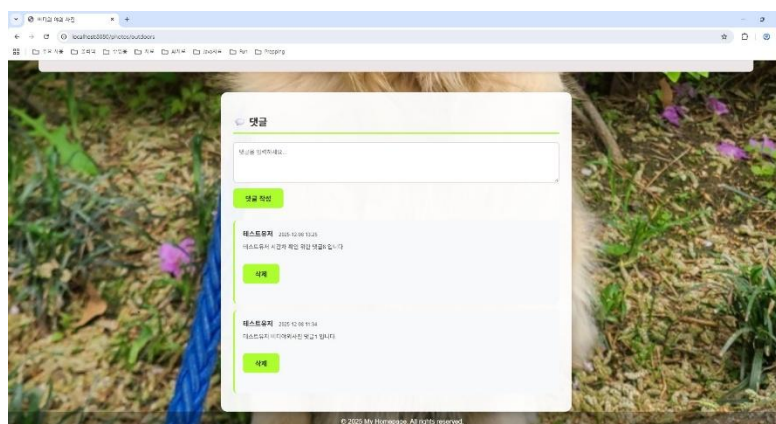
- 로그인 사용자만 댓글 작성 가능



- 작성자 본인 또는 관리자만 삭제 가능

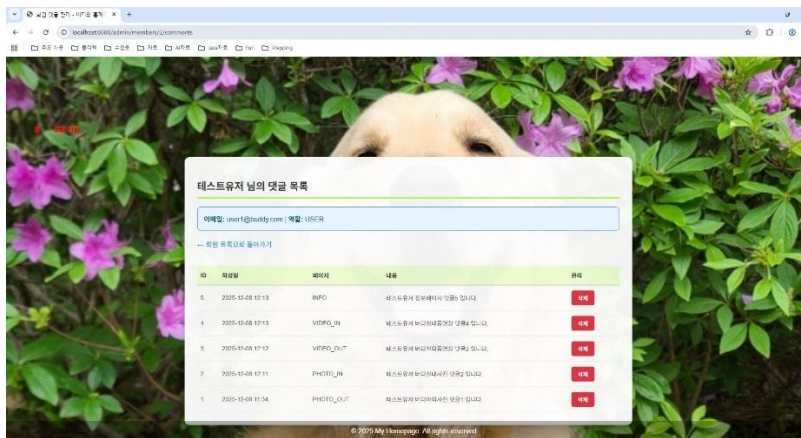


- 최신순 정렬

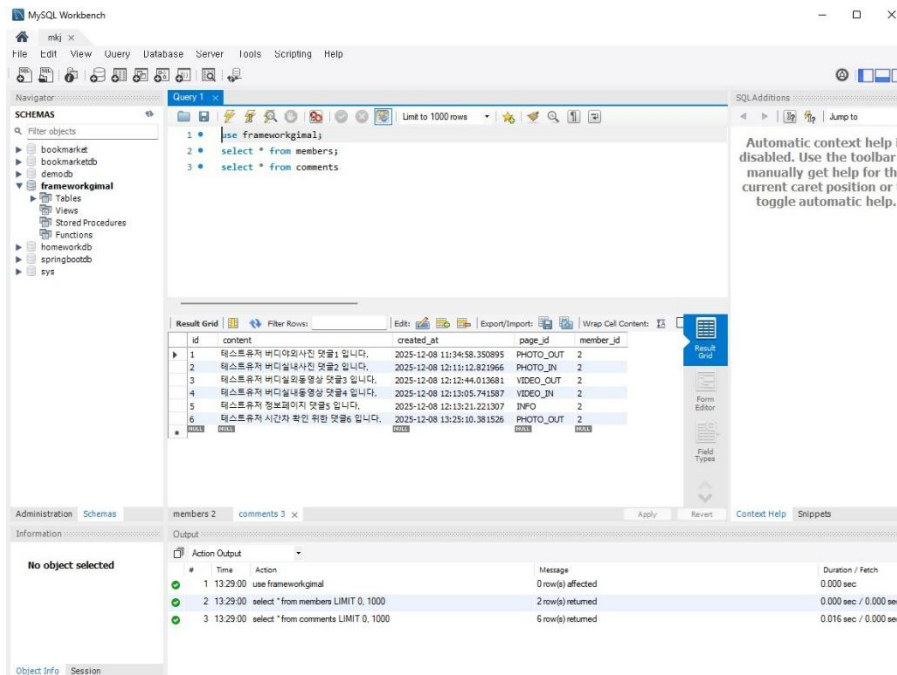


- page_id는 INFO, PHOTO_IN, PHOTO_OUT, VIDEO_IN, VIDEO_OUT 다섯 가

지 문자열로 구분되며, 각 페이지별 댓글을 분리 저장하기 위한 ENUM 역할을 함



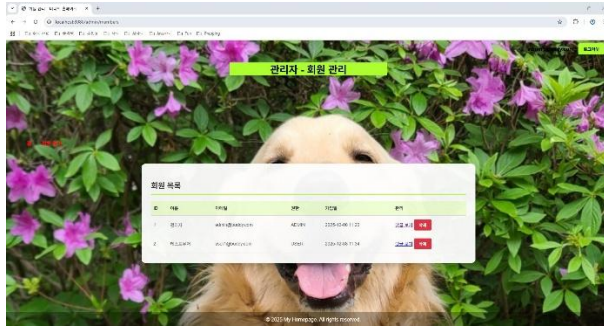
- MySQL 저장 성공 후 동작함



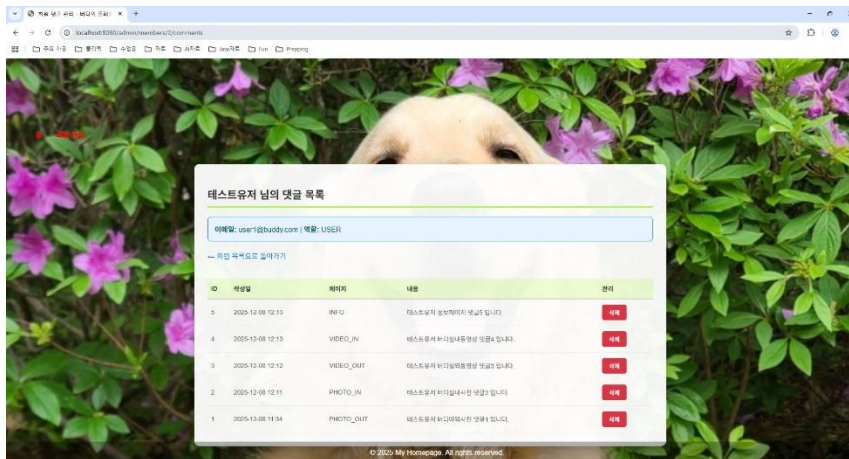
✓ (4) 관리자 전용 기능

관리자(Admin 권한)를 가진 계정만 접근 가능:

- 전체 회원 목록 조회



- 특정 회원이 작성한 모든 댓글 조회



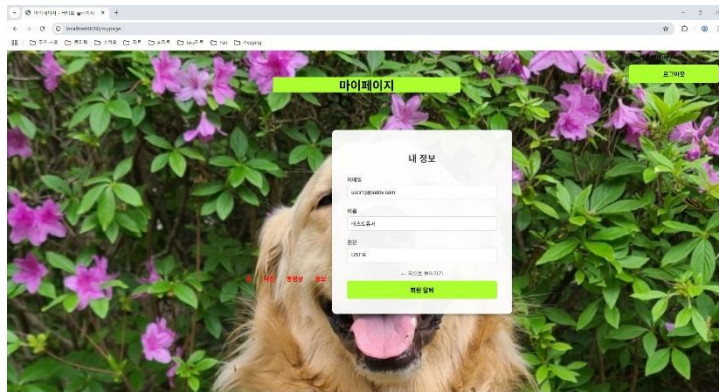
보안상 /admin/** 경로는 Spring Security에서 ADMIN 권한에게만 허용된다.

✓ (5) 마이페이지 기능

로그인된 사용자 본인의 정보만 조회할 수 있으며

- 이름
- 이메일
- 권한

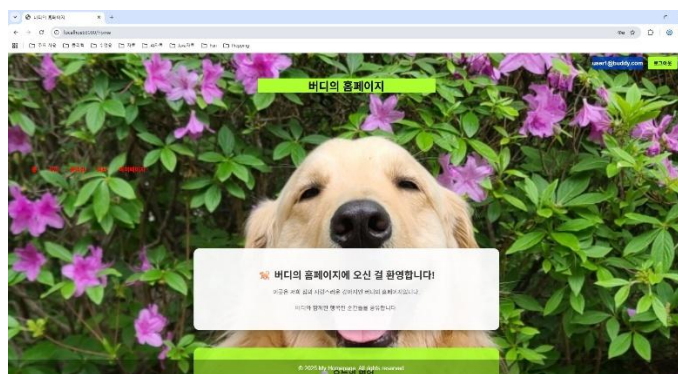
이 노출된다.



회원탈퇴는 선택사항이지만, 구현되어 있으면 추가 점수 요소가 된다.

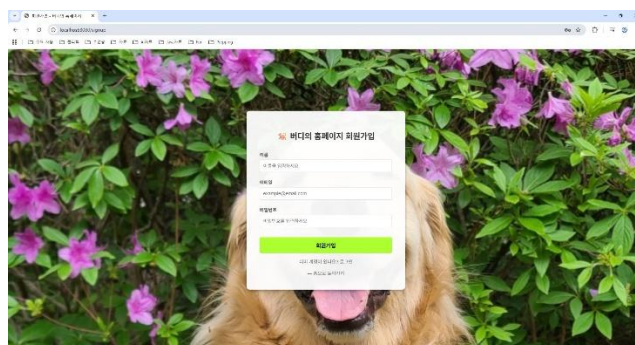
6. 대표 실행 화면

1. 홈페이지 메인 화면



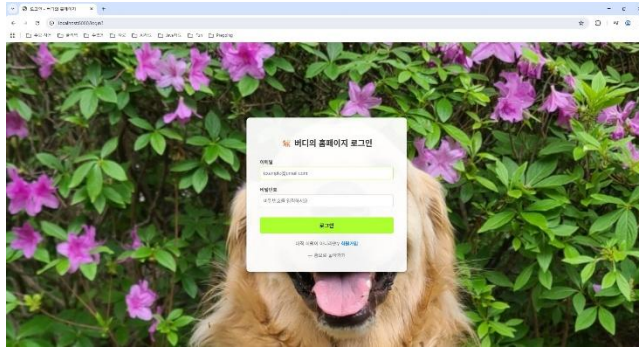
- 배경 이미지 + 네비게이션 + 로그인 영역 표시

2. 회원가입 화면

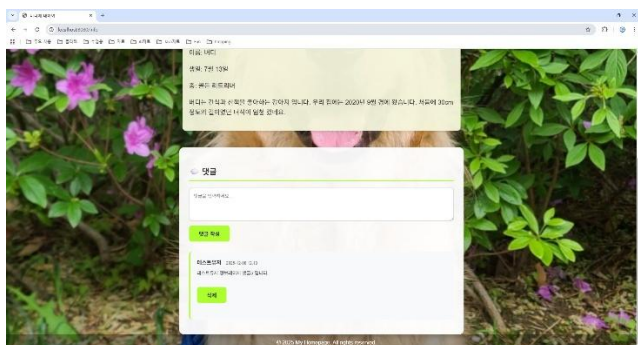
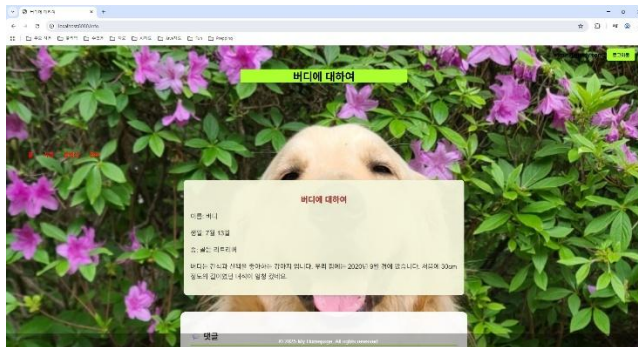


- 이름 / 이메일 / 비밀번호 입력폼

3. 로그인 화면

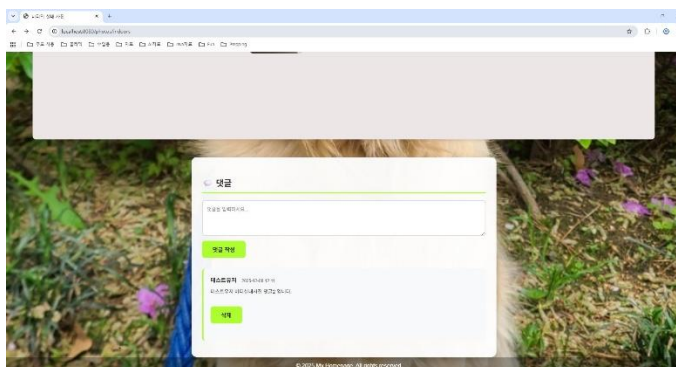


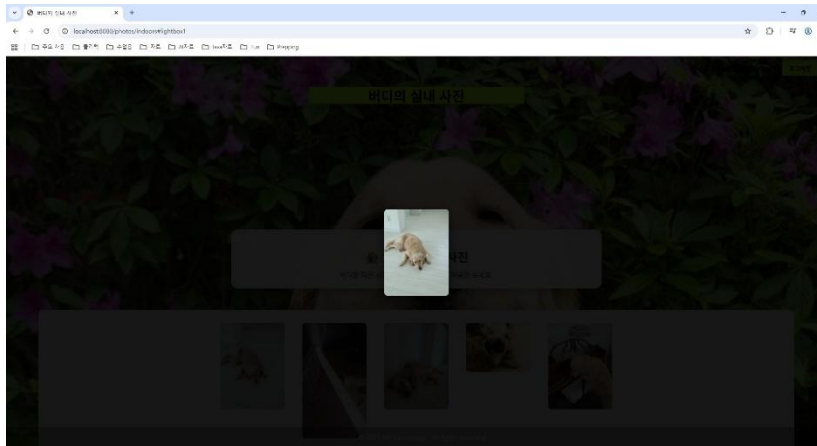
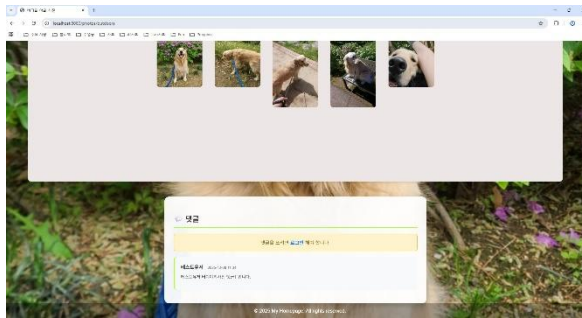
4. 정보 페이지 InfoPage



- 반려견 소개 + 댓글 목록

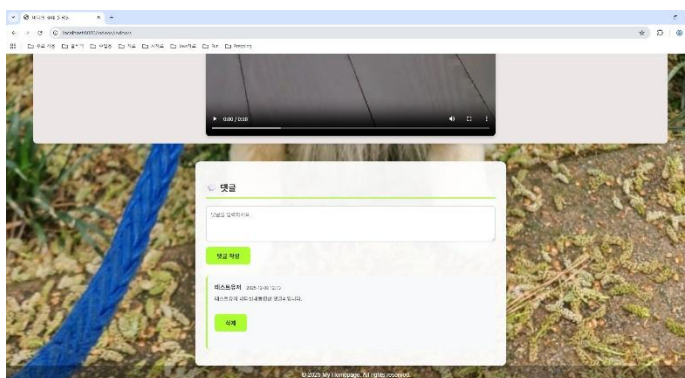
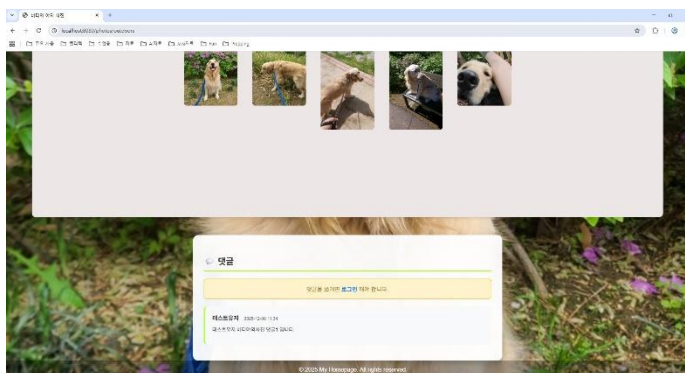
5. 사진 페이지 (실내/실외)





- Lightbox 기능 + 댓글 기능

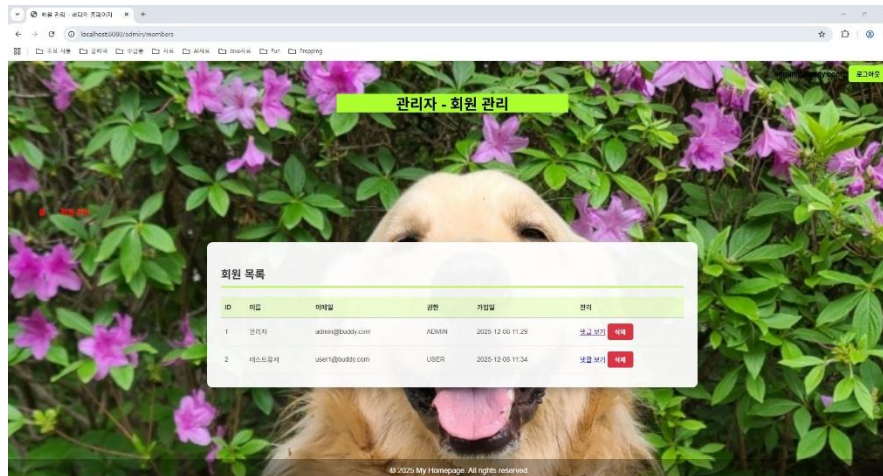
6. 동영상 페이지 (실내/실외)



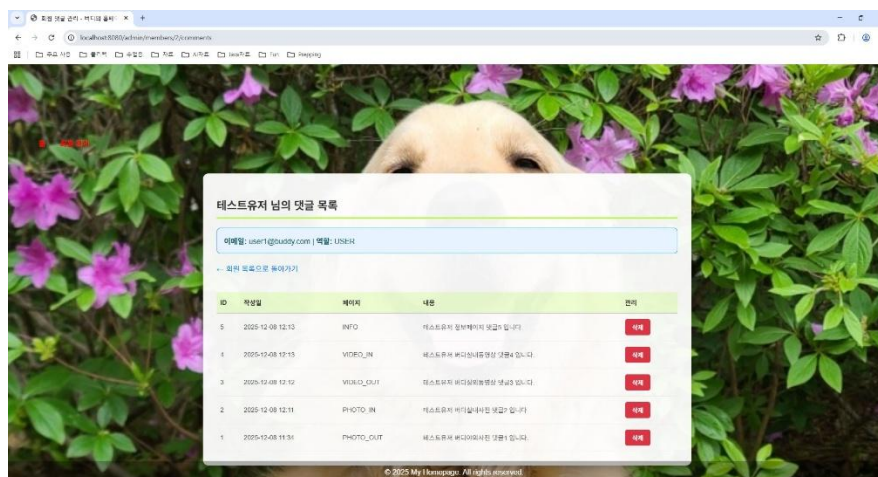
- 비디오 플레이어 + 댓글 기능

7. 관리자 페이지

- 회원 목록



- 회원별 댓글 목록



7. 개발 환경

- **Language:** Java 17
- **Framework:** Spring Boot 4.0.0
- **Database:** MySQL 8.0.43
- **JPA + Hibernate** 자동 스키마 생성
- **Build Tool:** Gradle
- **View Engine:** Thymeleaf

- **Security:** Spring Security + BCrypt
 - **IDE:** IntelliJ IDEA
 - **Github 주소:** <https://github.com/MKJPolytech/FrameworkTermProject>
-

8. 결론

본 프로젝트는 기존 정적 HTML 기반의 1학년 개인 홈페이지를 스프링 프레임워크 기반 MVC 웹 애플리케이션으로 발전시킨 것이다.

특히

- JPA 기반의 회원·댓글 데이터 관리
- Spring Security 기반 인증/권한 처리
- 페이지별 동적 댓글 기능
- 관리자 기능

등을 통해 **데이터 중심의 웹 서비스 구현 능력을 갖추는 데 초점을 맞췄다.**

이를 통해 Spring Boot 기반 웹 애플리케이션의 전반적인 개발 과정인

설계 → DB 구성 → MVC 개발 → 인증/권한 → 화면 연동

전체를 직접 구현하며 학습할 수 있었다.

또한, MySQL 8.0 기반으로 실제 스키마를 구성하고, 회원/댓글 데이터를 영구 저장함으로써 실서비스 구조에 가까운 구현을 경험할 수 있었다.