

기말 Term Project

주제: 1학년 때 제작한 개인 홈페이지 또는
새로 작성한 홈페이지에 스프링 부트 프로젝트로 제작

프로젝트 명: 캠퍼스 분실물 찾기 서비스
(CAMPUS LOST & FOUND)

Github -> <https://github.com/chakihwan/FrameworkTermProject>

한국폴리텍대학

서울정수캠퍼스

인공지능소프트웨어과

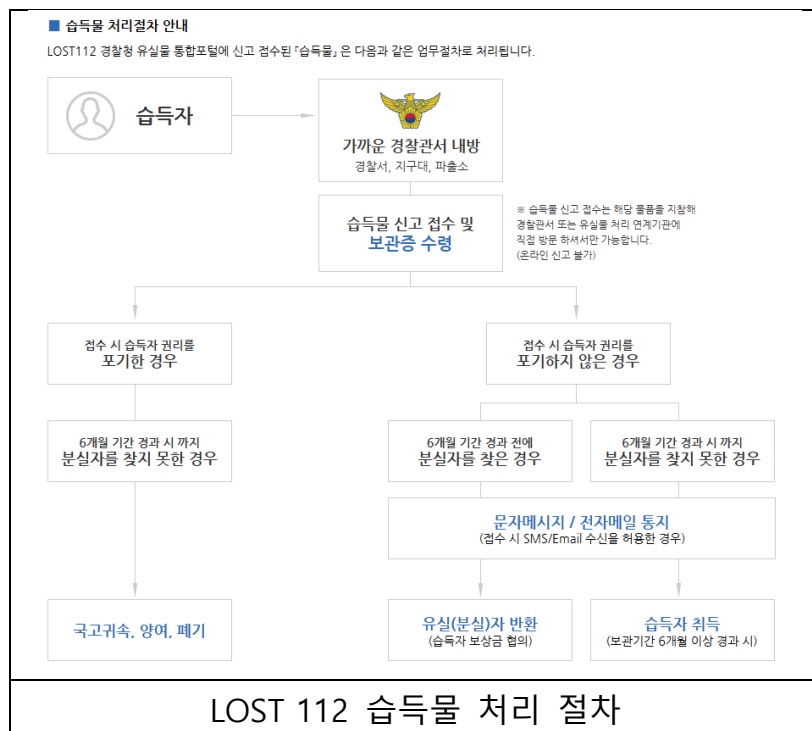
2401110268

차기환

1. 프로젝트 개요

1.1. 주제 선정 배경

최근 교내에서 타인의 '에어팟 4'를 습득한 경험이 있습니다. 주인을 찾아주기 위해 경찰청 유실물 종합관리시스템(LOST112)을 이용하려 했으나, 복잡한 가입 절차와 본인 인증 과정이 필요했습니다. 특히, 습득물은 온라인으로 게시할 수 없고 반드시 경찰서나 파출소에 직접 방문하여 제출해야만 했으며, 분실자가 게시한 글을 찾더라도 개인정보 보호로 인해 직접 연락할 방법이 없었습니다.



결국 시간을 내어 파출소에 방문해야만 했던 이 경험을 통해, "같은 캠퍼스 구성원끼리라면 복잡한 절차 없이, 더 빠르고 직관적으로 소통하며 물건을 찾아줄 수 없을까?" 라는 문제의식을 가지게 되었습니다.

1.2. 프로젝트 목적

본 프로젝트는 Spring Boot와 React를 활용하여 대학 캠퍼스 전용 분실물 분실/습득 웹 서비스를 구축하는 것을 목표로 합니다. 기존 시스템의 불편함을 해소하기 위해 다음과 같은 핵심 가치를 제공합니다.

1. 접근성: 경찰서 방문 없이 교내에서 습득물 정보를 공유하고 안전하게 보

관/전달.

2. 신속성: 카카오톡 오픈채팅 및 비밀 댓글을 통한 당사자 간 직접 소통 채널 제공.
3. 편의성: 직관적인 UI/UX(드래그 앤 드롭 업로드, 카드형 리스트)와 회원 인증 기반의 신뢰성 확보.

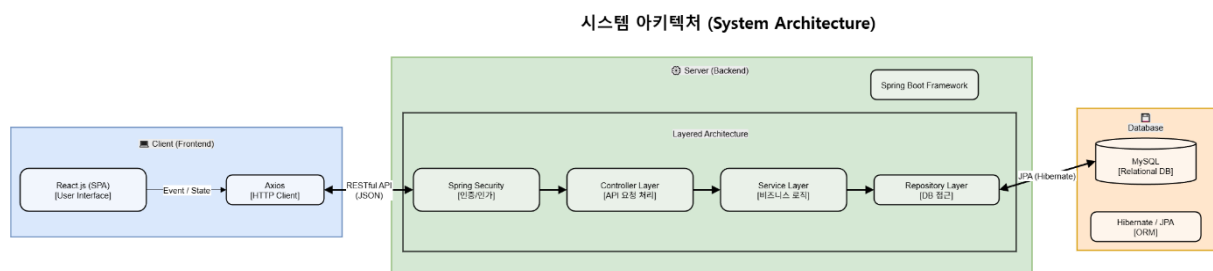
2. 시스템 설계 및 개발 환경

2.1. 개발 환경 (Development Environment)

구분	상세 내용	비고
OS	Windows 10	
Language	Java 17, JavaScript (ES6+)	
Backend	Spring Boot 4.0.0	Spring Security, Spring Data JPA
Frontend	React.js (v18)	Axios, React Router, CSS3
Database	MySQL	Hibernate ORM
IDE	IntelliJ IDEA, VS Code	

2.2. 시스템 아키텍처 (System Architecture)

본 시스템은 RESTful API 기반의 클라이언트-서버 구조로 설계되었습니다.



- Client (Frontend): React 기반의 SPA(Single Page Application)로 구현되어 사

용자 인터페이스를 담당하며, Axios를 통해 서버와 비동기 통신을 수행합니다.

- Server (Backend): Spring Boot 프레임워크를 기반으로 하며, Controller-Service-Repository 계층 구조를 따릅니다. Spring Security를 통해 인증/인가를 처리합니다.
- Database: MySQL 데이터베이스를 사용하여 회원 정보, 게시글, 댓글 데이터를 관계형으로 저장하며, JPA(Hibernate)를 통해 객체 지향적으로 관리합니다.

3. 데이터베이스 설계 (E-R Diagram)

시스템의 핵심 엔티티인 회원(Member), 분실물(LostItem), 댓글(Comment) 간의 관계를 설계하였습니다.

3.1. 엔티티 관계 설명 (Entity Relationship)

1. Member (회원)

- 설명: 서비스 이용자의 계정 정보를 관리하는 엔티티입니다.
- Primary Key (PK): id (Long, Auto Increment)
- 주요 속성:
 - username (아이디, Unique, Not Null): 로그인 시 사용하는 고유 아이디
 - password (비밀번호, Not Null): 암호화되어 저장되는 비밀번호
 - studentId (학번, Unique, Not Null): 교내 구성원 인증을 위한 고유 학번
 - name (이름): 사용자의 실명
 - phoneNumber (전화번호): 연락처 정보
 - role (권한): 사용자 권한 구분 (ENUM: USER, ADMIN)

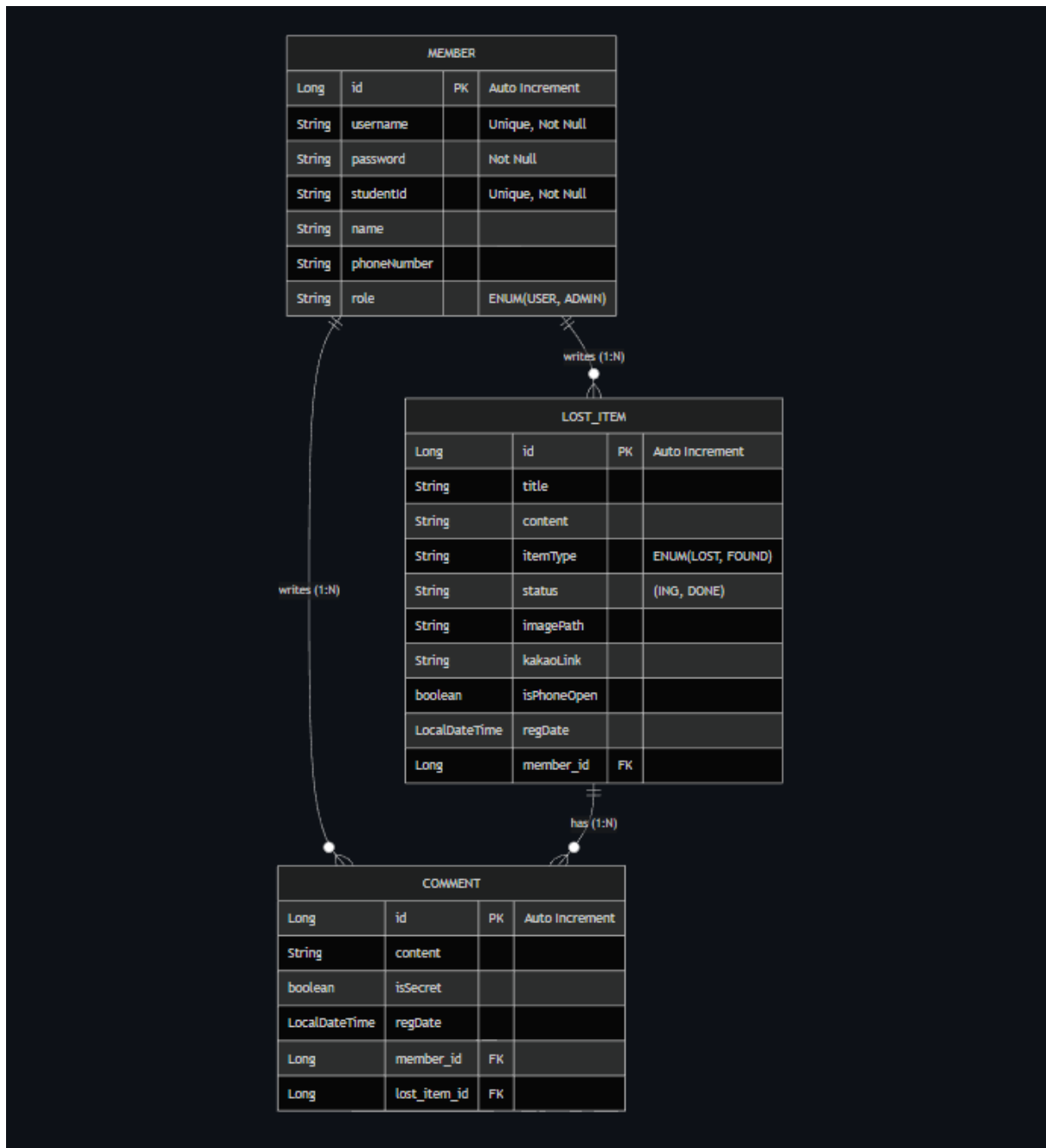
2. LostItem (분실물 게시글)

- 설명: 분실물 또는 습득물에 대한 상세 정보를 저장하는 게시글 엔티티입니다.
- Primary Key (PK): id (Long, Auto Increment)
- Foreign Key (FK): member_id (N:1 관계, 작성자)
- 주요 속성:
 - title (제목): 게시글 제목
 - content (내용): 분실/습득물에 대한 상세 설명 (최대 1000자)
 - itemType (유형): 게시글 성격 구분 (LOST: 분실, FOUND: 습득)
 - status (상태): 물건의 현재 상태 (ENUM: ING: 찾는 중, DONE: 해결됨)
 - imagePath (이미지 경로): 업로드된 사진 파일의 저장 경로 (Nullable)
 - kakaoLink (카카오톡 링크): 오픈채팅 등 외부 연락망 링크 (Nullable)
 - isPhoneOpen (전화번호 공개 여부): 작성자의 전화번호 공개 설정 (Boolean)
 - regDate (등록일): 게시글 생성 일시 (TimeStamp)

3. Comment (댓글)

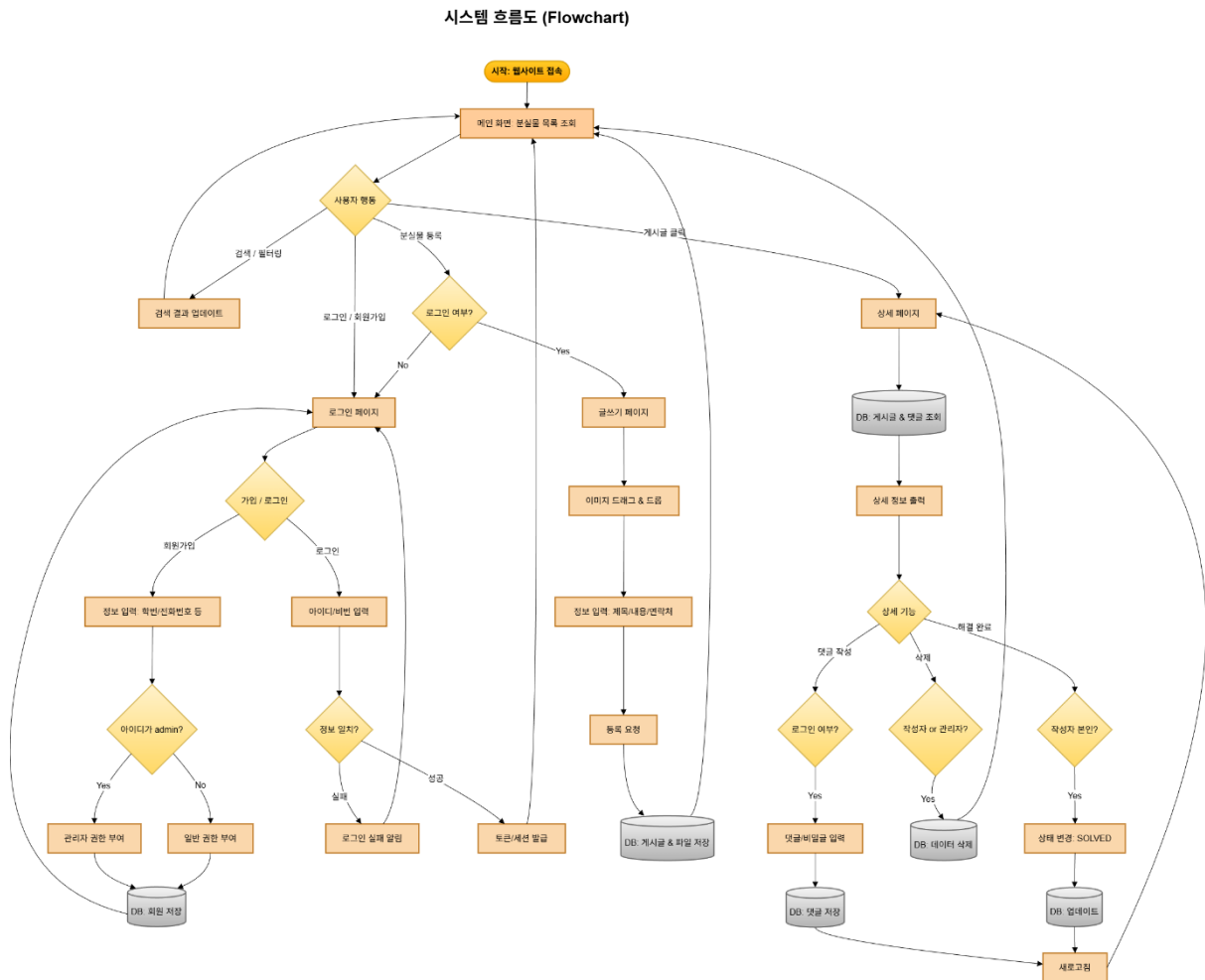
- 설명: 게시글에 대한 문의 및 소통 내용을 저장하는 엔티티입니다.
- Primary Key (PK): id (Long, Auto Increment)
- Foreign Key (FK):
 - lost_item_id (N:1 관계, 해당 게시글)
 - member_id (N:1 관계, 댓글 작성자)
- 주요 속성:

- content (내용): 댓글 본문
- isSecret (비밀글 여부): 작성자와 관리자만 볼 수 있는 비밀 댓글 설정 (Boolean)
- regDate (등록일): 댓글 생성 일시 (TimeStamp)



4. 시스템 주요 기능 및 흐름

4.1. 시스템 흐름도 (Flowchart)



- 접속 및 조회:** 사용자는 메인 화면에서 최신 분실물 목록을 카드 형태로 확인하고, 필터(분실/습득) 및 검색 기능을 이용할 수 있습니다.
- 회원가입/로그인:** 글 작성 및 댓글 이용을 위해 학번을 포함한 회원가입 후 로그인을 진행합니다. (admin 계정 생성 시 관리자 권한 부여)
- 분실물 등록:** 분실(LOST) 또는 습득(FOUND) 태그를 선택하고, 이미지를 드래그 앤 드롭으로 업로드한 뒤 연락처(카톡/전화번호) 공개 여부를 설정하여 등록합니다.
- 소통 및 완료:** 상세 페이지에서 댓글(비밀글 지원)로 소통하며, 물건을 찾으면 작성자가 상태를 해결됨(SOLVED)으로 변경하거나 게시글을 삭제하여 완료 처리합니다.

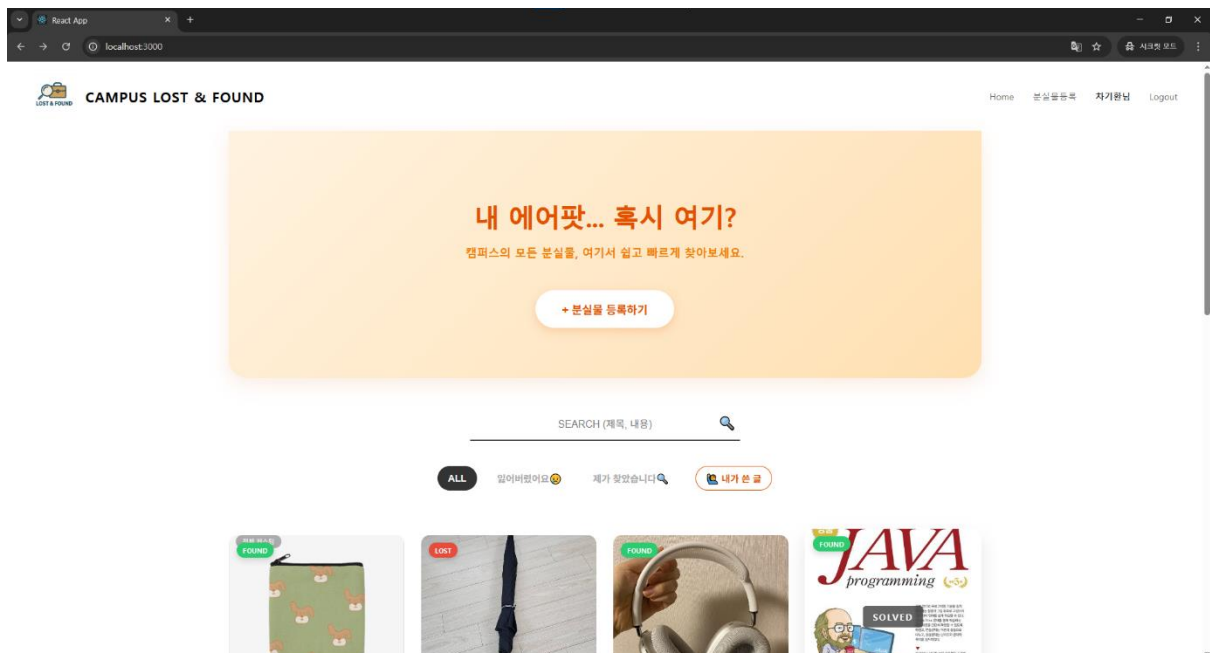
4.2. 주요 기능 상세

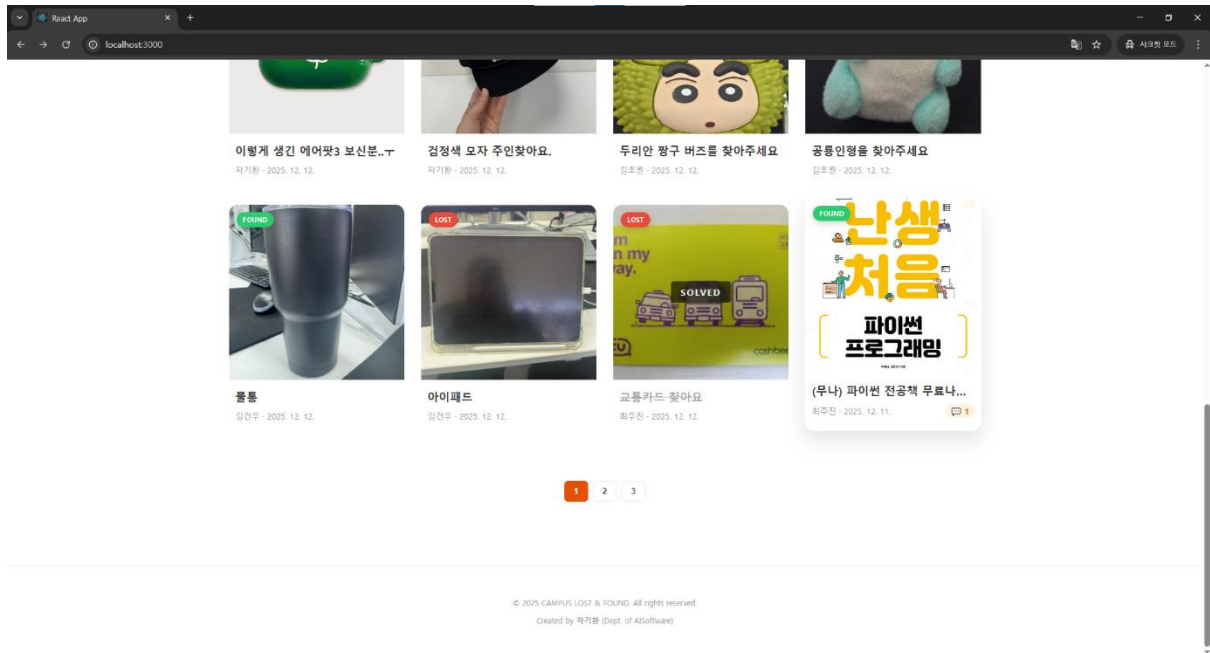
- **드래그 앤 드롭 업로드:** react-drag-drop-files 등의 라이브러리 없이 Native API를 활용하여 직관적인 파일 업로드 구현.
- **비밀 댓글 & 권한 관리:** 개인정보 보호를 위해 작성자와 관리자만 볼 수 있는 비밀 댓글 기능을 구현하였으며, 본인 글/댓글에 대한 삭제 권한 및 관리자(Admin)의 강제 삭제 권한을 구현.
- **UI/UX 최적화:** 메인 리스트에서 댓글 수를 배지로 미리 보여주어(💬 2) 사용자 참여 유도 및 편의성 증대.

5. 대표 실행 화면

5.1. 메인 화면 (목록 조회 및 필터링)

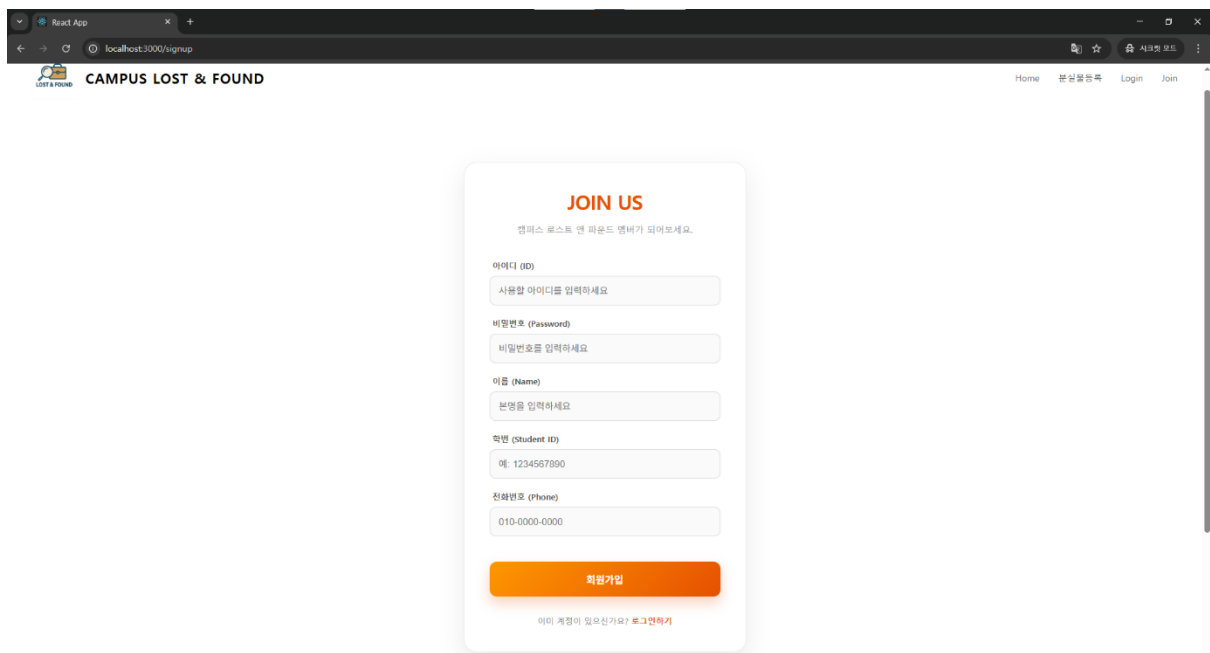
- 분실/습득 상태 태그와 해결 여부(SOLVED), 댓글 수를 직관적인 카드 UI로 제공합니다.
- 상단 필터 버튼을 통해 원하는 유형만 모아볼 수 있습니다.

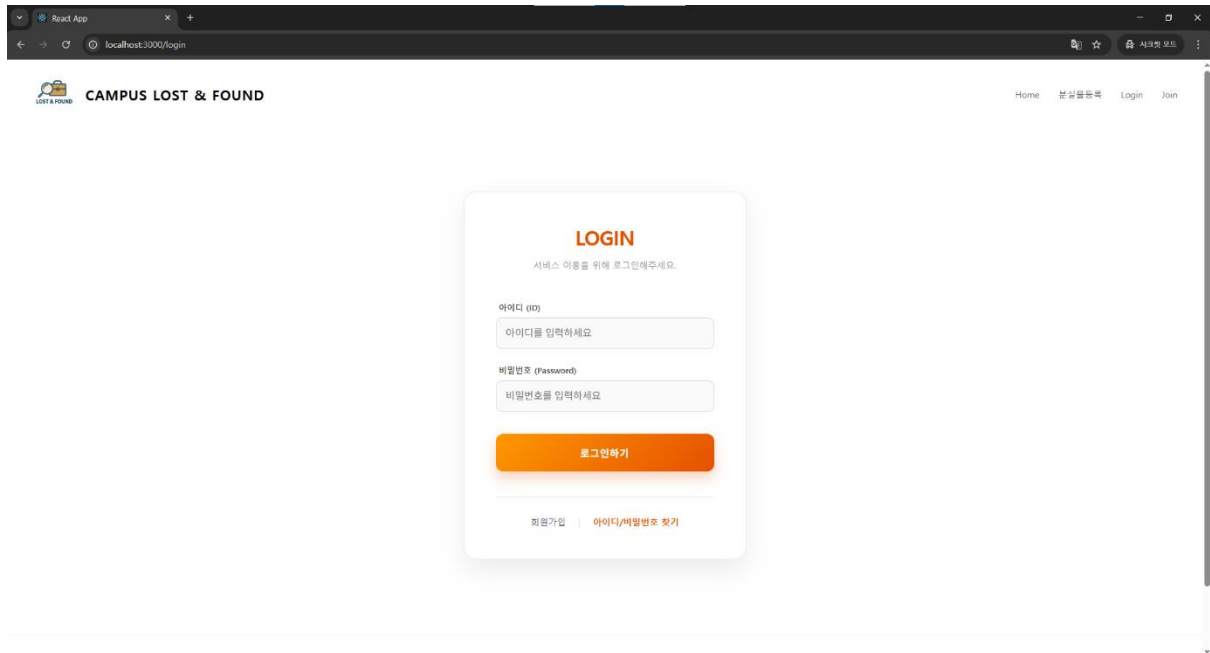




5.2. 회원가입 및 로그인

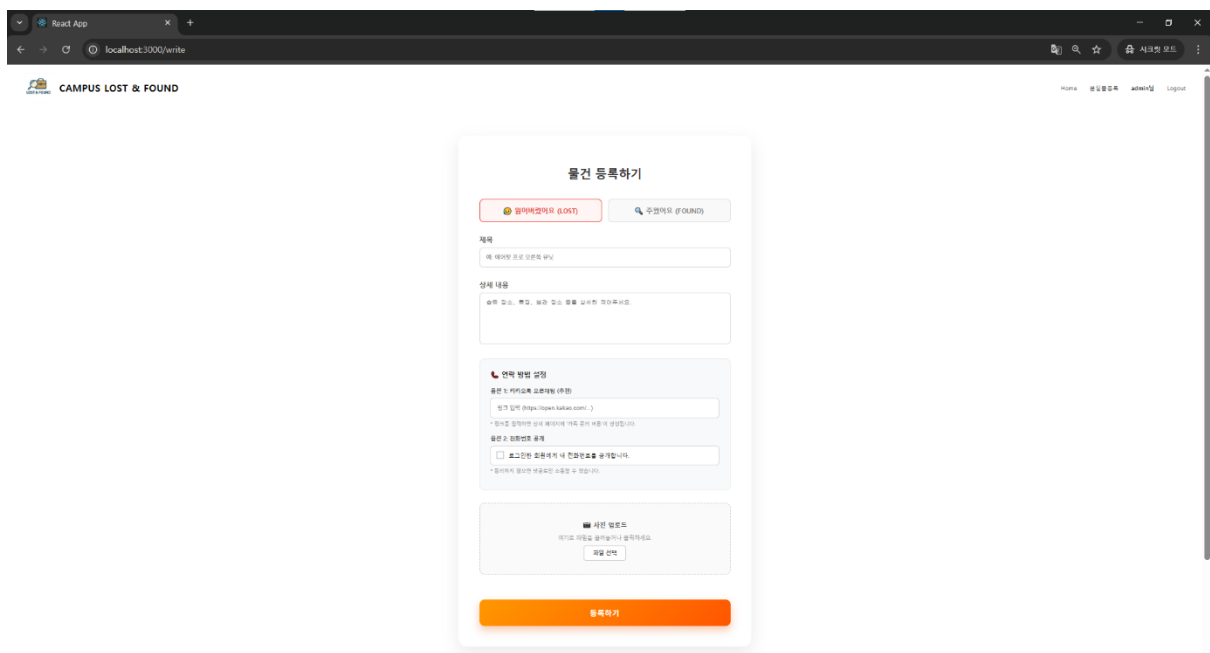
- 학번, 전화번호 등 교내 구성원 인증을 위한 필수 정보를 입력 받습니다.





5.3. 분실물 등록 (드래그 앤 드롭)

- 마우스 드래그 앤 드롭으로 사진을 쉽게 업로드하고, 카카오톡 오픈채팅 링크를 등록할 수 있습니다.



물건 등록하기

☐ 잃어버렸어요 (LOST)
 ☐ 찾았어요 (FOUND)

제목
예: 대학교 도서관 책 분실

상세 내용
분실 장소, 품목, 분실 장소, 분실 날짜, 분실 시간 등 상세히 작성해주세요.

연락 방법 설정

물건 조회 시 연락처 노출 여부 (선택)

* 연락처를 입력하면 상세 페이지에 자동 공유가 이루어집니다.

물건 조회 권한 설정

☒ 소그린팅 회원에게만 권한을 부여합니다.
 ☐ 소그린팅 회원에게만 권한을 부여합니다.

* 연락처를 입력하면 상세 페이지에 자동 공유가 이루어집니다.

사진 업로드

5.4. 상세 화면 (비밀 댓글 및 관리)

- 작성자와 관리자만 확인할 수 있는 비밀 댓글 기능이 적용되어 있습니다.
- 작성자 본인 및 관리자는 게시글과 댓글을 삭제할 수 있는 버튼이 활성화됩니다.

물건 등록하기

☐ 잃어버렸어요 (LOST)
 ☐ 찾았어요 (FOUND)

제목
예: 대학교 도서관 책 분실

상세 내용
분실 장소, 품목, 분실 장소, 분실 날짜, 분실 시간 등 상세히 작성해주세요.

연락 방법 설정

물건 조회 시 연락처 노출 여부 (선택)

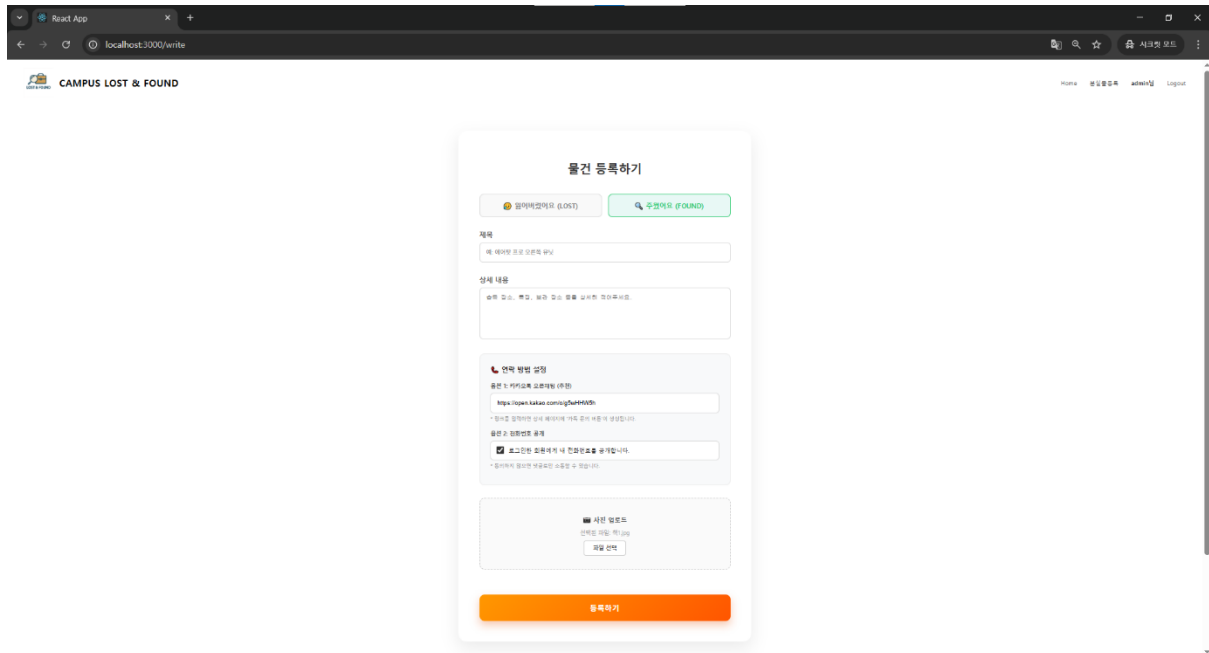
* 연락처를 입력하면 상세 페이지에 자동 공유가 이루어집니다.

물건 조회 권한 설정

☒ 소그린팅 회원에게만 권한을 부여합니다.
 ☐ 소그린팅 회원에게만 권한을 부여합니다.

* 연락처를 입력하면 상세 페이지에 자동 공유가 이루어집니다.

사진 업로드



6. 결론 및 고찰

이번 프로젝트를 통해 Spring Boot와 React를 연동한 풀스택 웹 개발의 전 과정을 경험할 수 있었습니다.

특히 JPA를 활용하여 데이터베이스를 객체 지향적으로 다루는 방법과, Spring Security를 통한 인증/인가 로직 구현에 대해 깊이 있게 학습하였습니다.

초기 기획 단계에서 느꼈던 기존 유실물 시스템의 불편함을 '사용자 편의성 중심'의 UI와 '자율적인 소통 기능'으로 해결하고자 노력하였으며, 드래그 앤 드롭이나 비밀 댓글과 같은 디테일한 기능을 구현하며 완성도를 높였습니다.

향후에는 교내 인증 API 연동이나 모바일 앱 확장을 통해 실제 캠퍼스에서 활용 가능한 수준으로 발전시키고 싶습니다.

[향후 개선 계획 / 현재 문제점]

본 프로젝트는 기능 구현에 초점을 맞추었으나, 실제 상용 서비스를 위해서는 다음과 같은 기술적 고도화가 필요하다고 판단됩니다.

1. **환경 분리:** 하드코딩된 파일 경로와 API 주소를 환경 변수(Environment Variables)로 추출하여, 개발/운영 환경의 이식성을 높여야 합니다.

- 문제점:

1. 파일 업로드 경로: D:/FrameworkTermProject/... 처럼 드라이브명이 포함된 절대 경로를 사용 중입니다. 맥(Mac)이나 리눅스 서버에서는 D: 드라이브가 없어 에러가 납니다.
2. 프론트엔드 API 주소: http://192.168.xx.xx:8081 처럼 IP가 고정되어 있습니다. 와이파이가 바뀌거나 서버 IP가 변경되면 프론트엔드 코드를 다시 빌드해야 합니다.

- 개선 방향:

"현재 개발 편의를 위해 로컬 절대 경로와 고정 IP를 사용하였으나, 향후에는 application.properties의 프로파일(Profile) 기능을 활용하여 OS별 경로를 분리하고, 프론트엔드에서는 **환경 변수(.env)**를 사용하여 API 주소를 동적으로 관리하도록 개선할 예정입니다."

2. **예외 처리 통합:** @ControllerAdvice를 활용한 전역 예외 처리기를 도입하여, 중복 코드를 제거하고 일관된 에러 응답을 클라이언트에게 전달해야 합니다.

- 문제점:

1. 모든 API 메소드마다 try { ... } catch (Exception e) { ... } 코드가 반복됩니다.

- 개선 방향:

"현재는 컨트롤러 레벨에서 개별적으로 예외를 처리하고 있으나, 추후 **@ControllerAdvice와 @ExceptionHandler**를 도입하여 전역적으로 예외를 핸들링하고, 클라이언트에게 통일된 에러 응답 포맷(Error Response)을 제공하는 구조로 리팩토링이 필요합니다."

3. **데이터 검증 강화:** 프론트엔드뿐만 아니라 백엔드 DTO 계층에서도 @Valid를 통한 엄격한 유효성 검사를 적용하여 데이터 무결성을 확보해야 합니다.

- 문제점:

1. 학번에 숫자가 아닌 문자가 들어오거나, 비밀번호가 너무 짧은 경우에도 서버로 요청이 넘어갈 수 있습니다.

- 개선 방향:

"데이터 무결성을 위해 DTO 객체에 **@Valid 어노테이션과 Bean Validation(예: @NotNull, @Size, @Pattern)**을 적용하여, 잘못된 입력값이 서버 로직으로 진입하기 전에 검증하는 과정을 추가할 계획입니다."