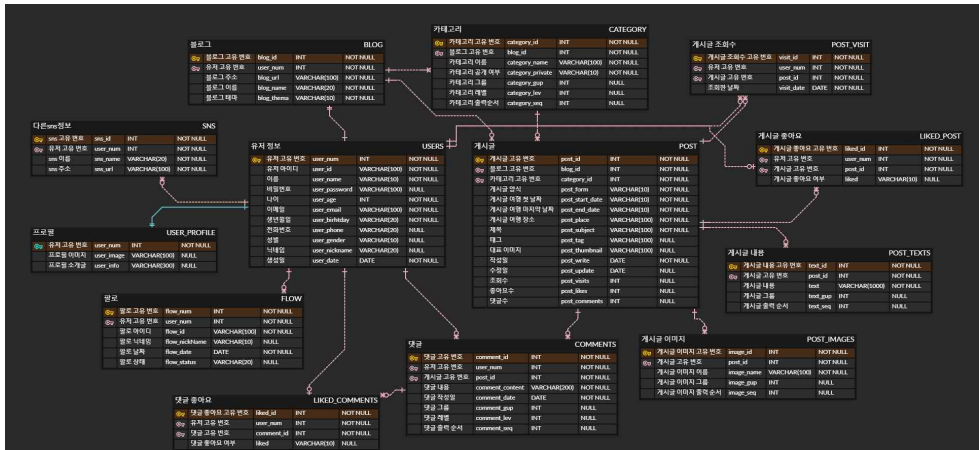


## 프로젝트 개요

- 주제 여행 블로그 - 개인 프로젝트(1명)
- 기간 2023.09.20 ~ 진행중
- GitHub [https://github.com/HYA6/Travel\\_Blog](https://github.com/HYA6/Travel_Blog)
- 개발 목적
  - Spring Boot 기반 웹 서비스 개발
  - 로그인 API, 회원가입, 외부 API 사용 등 해보지 않은 기능 구현
  - JPA 사용 및 H2 Database 활용
- 개발 환경
  - BACK-END Java, Spring Boot, JPA, Lombok, Json, Ajax, Maven
  - FRONT-END JavaScript, JQuery, Thymeleaf, CSS, Bootstrap
  - DB MySQL, H2 Database
  - SERVER Tomcat
  - API REST API, Naver Login API, Google Map API
  - ETC STS, GitHub



## ERD



## 프로젝트 기능 - 회원가입



## 프로젝트 기능 - 회원가입

Travel Blog

## JOIN US

회원가입

\* 표시는 필수로 입력해야 합니다.

\* 이름

\* 아이디

중복확인

\* 비밀번호

\* 비밀번호 확인

\* 생년월일

연도-월-일

\* 이메일

@ 직장 입력

-선택-

전화번호

00\*-000\*-0000

성별

☒ 선택안함 ☐ 남성 ☐ 여성

닉네임

회원 가입

## 프로젝트 기능 - 회원가입

사용 가능한 아이디입니다.

확인

### JOIN US

회원 가입

\* 표시는 필수로 입력해야 합니다.

\* 이름

이하영

\* 아이디

hya

중복 확인

\* 비밀번호

\*\*\*\*

\* 비밀번호 확인

\*\*\*\*

\* 이메일

### 회원가입 시 아이디 중복 확인 기능

#### ✓ 구현 과정

이미 입력된 다른 데이터들이 변경되지 않도록 **Fetch Ajax**를 사용하였습니다. 입력한 아이디를 DB에서 찾아 값이 없으면 올바른 응답 요청을 보내 회원가입을 이어갈 수 있도록 구현하였습니다.

## 프로젝트 기능 - 계층형 카테고리 생성

The image shows a 'CREATE CATEGORY' form with several sections and annotations. The form is titled 'CREATE CATEGORY' and has a subtitle '카테고리 생성하기'. Below the subtitle, it says '저장 버튼을 눌러야 내용이 저장됩니다.' (Press the save button for the content to be saved).

The form contains the following fields and controls:

- 1번째 카테고리** (1st Category): Includes '추가' (Add) and '삭제' (Delete) buttons. An annotation '서브 카테고리 생성' (Sub-category creation) points to the '추가' button.
- L1-1번째 카테고리** (L1-1 Category): Includes a '삭제' (Delete) button. An annotation '각 카테고리별 삭제' (Delete for each category) points to this button.
- L1-2번째 카테고리** (L1-2 Category): Includes a '삭제' (Delete) button.
- 2번째 카테고리** (2nd Category): Includes a lock icon, '추가' (Add), and '삭제' (Delete) buttons. An annotation '메인 카테고리 생성' (Main category creation) points to the '추가' button.
- 3번째 카테고리** (3rd Category): Includes '추가' (Add) and '삭제' (Delete) buttons.
- 카테고리 공개 여부** (Category public status): Includes radio buttons for '공개' (Public) and '비공개' (Private). The '비공개' option is selected.
- 저장** (Save) button: Located at the bottom right of the form. An annotation '카테고리 저장' (Category save) points to this button.
- 다음으로** (Next) button: Located at the bottom right of the form.

Annotations and arrows:

- '서브 카테고리 생성' (Sub-category creation) points to the '추가' button under '1번째 카테고리'.
- '각 카테고리별 삭제' (Delete for each category) points to the '삭제' button under 'L1-1번째 카테고리'.
- '메인 카테고리 생성' (Main category creation) points to the '추가' button under '2번째 카테고리'.
- '카테고리 저장' (Category save) points to the '저장' button.

## 프로젝트 기능 - 계층형 카테고리 생성

1번째 카테고리

1-1번째 카테고리

1-2번째 카테고리

2번째 메인 카테고리 이름

2번째 카테고리

카테고리 공개 여부 ☐ 공개 ☒ 비공개

모든 카테고리를 입력했으면 이 버튼을 클릭하세요.

### 카테고리 선택, 추가 및 삭제 기능

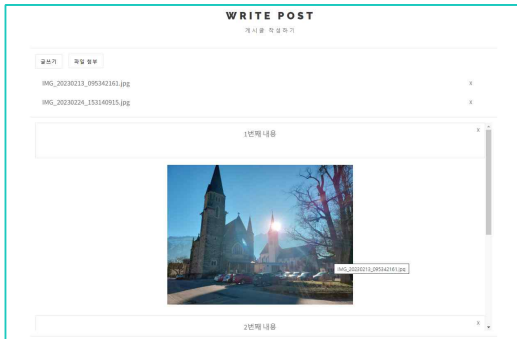
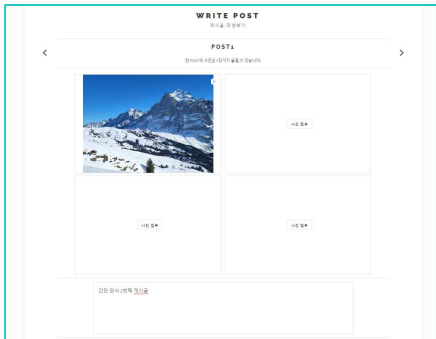
#### ✓ 구현 과정

JavaScript로 태그를 추가, 삭제하는 코드를 작성하여 **동적 페이지**를 구현하였습니다.

Ajax 비동기 통신을 이용해 각 카테고리를 선택(읽어오기), 저장, 삭제할 수 있도록 구현하였습니다.

REST API 기반으로 CRUD 기능을 구현해 JPA를 활용하여 DB와 연결했습니다.

## 프로젝트 기능 - 게시글 작성



### 간단 양식&기본 양식 게시글 작성

#### ✓ 구현 과정

간단 양식은 이미지와 텍스트를 구분하였고, 이미지는 각 파트 별로 사진 첨부가 가능하도록 구현하였습니다.

JavaScript를 이용하여 이미지를 표시하였으며 둘 다 **동적 페이지**로 구현하였습니다.

각 이미지와 텍스트는 버튼 하나에 저장할 수 있도록 **Ajax 비동기 통신**과 **REST API**를 사용하여 DB에 저장하도록 구현하였습니다.



## 프로젝트 기능 - 게시글 작성


WRITE POST

게시글 작성하기

대표 이미지

파일 선택

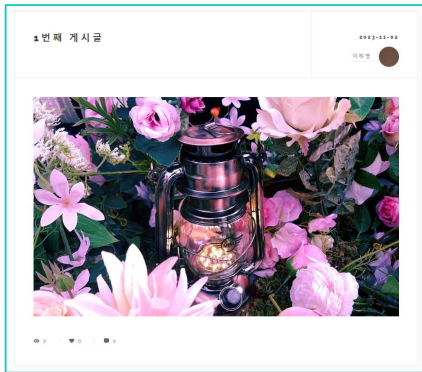
1572730025871-9.jpg



해시태그

Enter로 추가해보세요.

#태스트



### 대표 이미지 & 해시태그

#### ✓ 구현 과정

대표 이미지를 따로 업로드하도록 구현하였습니다.

대표 이미지는 게시글 DB에 추가하였고, 게시글 이미지는 따로 게시글 이미지 DB를 만들어 분리하였습니다.

해시태그는 텍스트 전부 DB에 저장 후 얻어올 때 ', '로 구분하여 얻어오도록 구현하였습니다.

## 프로젝트 기능 - 댓글 및 답글



### 댓글, 답글 작성 및 삭제

#### ✓ 구현 과정

댓글 DB에 외래키로 지정된 작성자 고유 번호를 이용하여 작성자 이름을 불러오게 구현하였습니다.

삭제 시 삭제한 여부가 DB로 저장되며 삭제된 댓글로 변경됩니다.

## 개인 프로젝트를 진행하면서 느낀 점



- ❖ 팀 프로젝트 때 해보지 못했던 로그인과 회원가입 기능을 구현해볼 수 있는 기회였습니다.
- ❖ 외부 API를 사용해볼 수 있었고 추후 채팅 및 메신저 기능 또한 추가할 예정입니다.
- ❖ 로컬 이미지 저장을 온전히 혼자 해볼 수 있었으며, 실무에서는 서버에 이미지를 저장하여 사용한다는 점을 알게 되었습니다.
- ❖ 템플릿, 데이터베이스, 프레임워크 등 하나하나 비교하면서 신중하게 선택하고 공부할 수 있었습니다.
- ❖ Ajax와 REST API를 주로 사용하여 로직을 구성하다보니 그에 관련하여 더 깊이 공부할 수 있었습니다.



- ❖ Ajax 비동기 처리 순서를 정리하는 데 어려움이 있었습니다.
- ❖ 프로젝트를 문서화시키는 작업도 중요하다는 것을 깨달았습니다.
- ❖ 프로젝트 시기와 취업 시기가 겹쳐 온전히 집중하지 못해 기본만 끝내놓은 것이 아쉽습니다.