

# Health\_Mate

조장 | 정시원

조원 | 김정학 | 김영철

# 목 차

01

팀원 소개 및 역할

02

프로젝트 목표

03

기획 내용

04

협업 활동

05

추가 기능

06

트러블 슈팅

07

소감

04

동작 시연

## 정시원

- Board - CRUD
- Post - 좋아요 구현
- Post - 파일 업로드
- Post - 프론트 작업

## 김정학

- Post - CRUD
- Post - page 처리
- Post - 정렬 구현
- Post - 검색 구현
- Post - ViewCount 로직
- User - CRUD
- User - 로그인 세션 로직
- User - 마이페이지

## 김영철

- Comment - CRUD
- Comment - page 처리
- Comment - 정렬 구현
- Board - 슬라이드 구현
- Board - 프론트 작업

# 프로젝트 목표

## 서비스 목표

- Health\_Mate의 목표는 사용자가 운동과 건강에 대한 정보와 경험을 공유하고 서로 격려할 수 있는 커뮤니티를 제공하는 것입니다.
- 운동 목표를 설정하고 성과를 기록하며, 사용자 간의 피드백과 동기부여를 통해 건강한 생활습관을 지원합니다.

## 서비스 대상

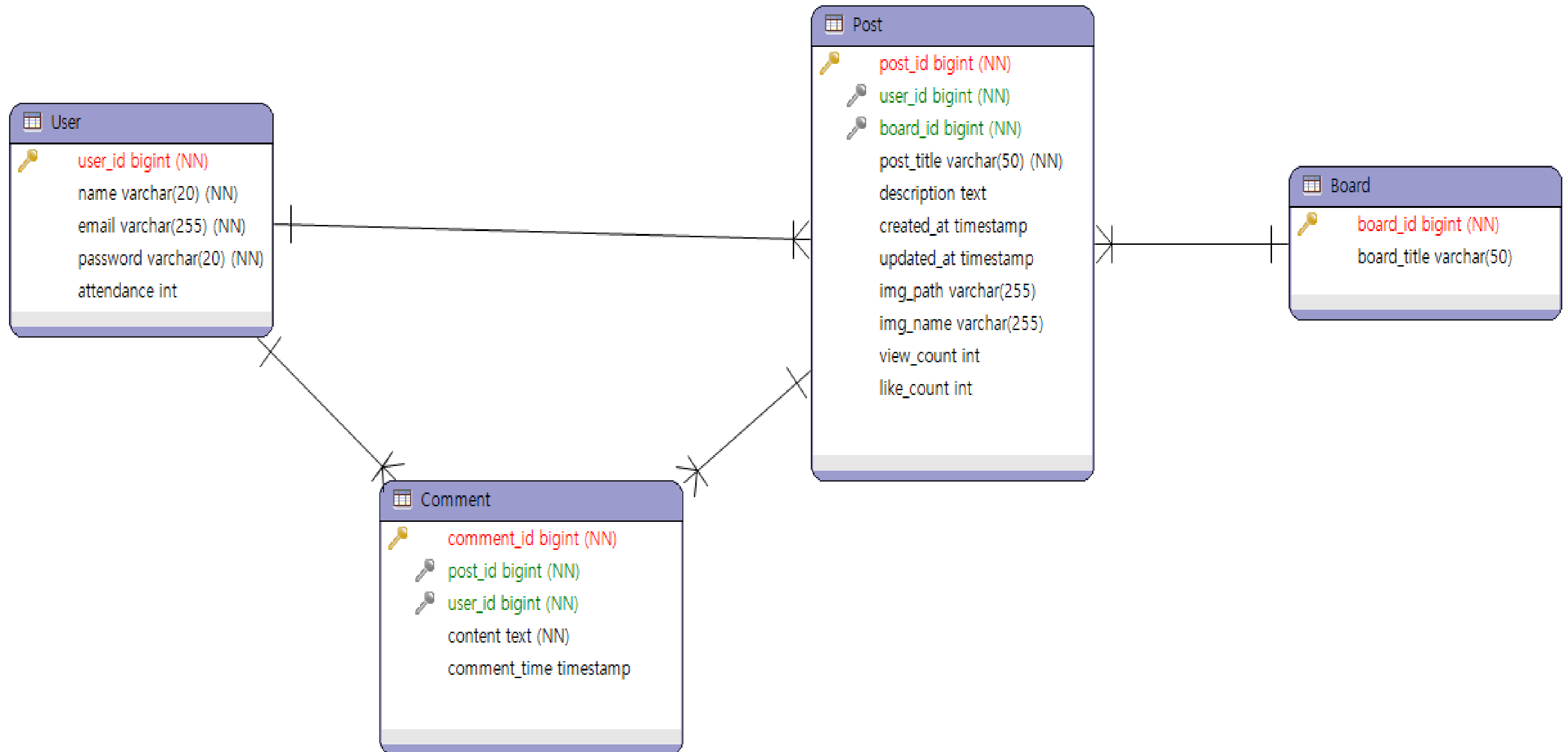
1. 정시원
  - 운동 초보자
  - 운동 시작에 대한 정보를 찾고 피드백을 통해 지속적인 동기부여를 받고자 함.
2. 김영철
  - 운동 애호가
  - 자신만의 운동 노하우와 꿀팁을 공유하고 다른 사용자들과 경험을 나누고자 함.
3. 김정학
  - 건강에 관심 있는 일반인
  - 운동과 건강에 관련된 최신 정보, 팁, 그리고 동기부여를 얻고자 함.
4. 커뮤니티에서의 피드백을 원하는 모두
  - 자신의 운동 계획과 목표에 대해 피드백을 받고 성취감을 느끼고자 함.

## 필수 기능

1. 회원가입 및 로그인 기능
  - 사용자 인증을 위한 회원가입과 로그인 기능.
2. 운동 게시판
  - 사용자가 운동과 관련된 글을 작성하고 공유할 수 있는 게시판 기능. 게시물 작성, 수정, 삭제 기능 제공.
3. 댓글 기능
  - 각 게시물에 댓글을 남길 수 있는 기능으로, 사용자 간 소통 지원
4. 마이페이지 기능
  - 사용자 정보 수정 및 작성한 게시글 및 댓글 확인 가능.

# 기획 내용

domain	<ul style="list-style-type: none"><li>- User</li><li>- Board</li><li>- Post</li><li>- Comment</li></ul>	Repository	<ul style="list-style-type: none"><li>- Jdbc Template<ul style="list-style-type: none"><li>• CRUD 기능에서 사용</li></ul></li><li>- JPA<ul style="list-style-type: none"><li>• @Entity 기능에서 사용</li><li>• 페이징 기능에서 사용</li></ul></li></ul>
DB	<ul style="list-style-type: none"><li>- MySQL</li></ul>	Build tool	<ul style="list-style-type: none"><li>- Gradle<ul style="list-style-type: none"><li>• 스크립트가 간결한 스크립트</li><li>• 유연한 의존성 관리</li><li>• 빠른 빌드 속도와 뛰어난 성능</li></ul></li></ul>



ID	업무구분(대)	업무구분(중)	요구사항	요구사항 내용	기능	중요도
P1	게시글(Post)	게시글 목록	게시글 목록을 조회	전체 게시글 목록조회	게시글은 한 페이지에 10개씩 최신순으로 보여 줌.	5
P2	게시글(Post)	게시글 목록	게시글 목록 조회 방식	전체 게시글 목록 조회 방식	인기순, 조회순, 최신순 등과 같이 선택하여 목록을 출력	2
P3	게시글(Post)	게시글 목록	게시글 목록을 조회	검색 버튼에서 검색된 게시글 목록 조회	게시글 제목으로 검색.	4
P4	게시글(Post)	게시글 조회	게시글을 조회	게시글 클릭 시 작성된 게시글의 상세 페이지로 이동	게시글 제목, 작성자, 조회수, 좋아요수, 업로드된 시간, 제목, 내용, 첨부된 파일 확인 가능.	5
P5	게시글(Post)	게시글 작성	게시글을 작성	게시글 작성 버튼 누르면 게시글 작성 페이지로 이동	로그인을 한 회원에게만 게시글 작성 권한 부여.	4
P6	게시글(Post)	게시글 작성	게시글을 작성	게시글 작성 폼으로 이동	게시글 제목, 내용, 첨부파일만 작성.	5
P7	게시글(Post)	게시글 좋아요	게시글에 좋아요 버튼	좋아요 버튼을 누르면 해당 게시글에 좋아요 수가 증가함.	로그인한 회원만 가능.	3
P8	게시글(Post)	게시글 수정	게시글을 수정	게시글 수정 버튼 누르면 수정 페이지로 이동	기존에 작성되었던 게시글 내용을 보여줌.	4
P9	게시글(Post)	게시글 수정	게시글을 수정	게시글 수정 폼으로 이동	게시글을 수정한 회원만 수정 가능	5
P10	게시글(Post)	게시글 수정	게시글을 수정	게시글 수정 완료	게시글이 업로드 된 시간은 수정된 시간으로 업데이트.	4
P11	게시글(Post)	게시글 삭제	게시글을 삭제	삭제 버튼 누르면 게시글 삭제	게시글을 작성한 회원만 삭제 가능.	5
P12	게시글(Post)	게시글 삭제	게시글을 삭제	게시글 삭제 확인	게시글 삭제 전에 알람 메시지 출력 후 동의하면 삭제 진행.	2
p13	게시글(Post)	게시글 방문	조회수 증가	게시글을 방문하면 조회수 증가	ID당 하나의 게시글 조회수 증가	2

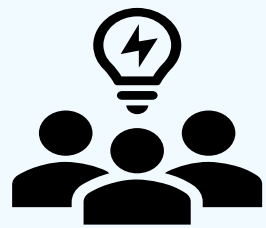
ID	업무구분(대)	업무구분(중)	요구사항	요구사항 내용	기능	중요도
B1	게시판(Board)	게시판 조회	게시판 목록을 조회	전체 게시판 목록 조회		5
B2	게시판(Board)	게시판 작성	게시판을 작성	게시판 클릭시 해당 게시글 페이지로 이동		5
B3	게시판(Board)	게시판 작성	게시판을 작성	게시판 작성 폼으로 이동	게시판 이름만 작성.	5
B4	게시판(Board)	게시판 수정	게시판을 수정	게시판 수정 폼으로 이동		5
B5	게시판(Board)	게시판 삭제	게시판을 삭제	게시판 삭제 버튼 누르면 게시판 삭제		5
B6	게시판(Board)	게시판 삭제	게시판을 삭제	게시판 삭제 확인	게시판 삭제 전에 알람 메시지 출력 후 동의하면 삭제 진행.	3
B7	게시판(Board)	게시판 권한	게시판 권한		게시판 작성, 수정, 삭제는 관리자만 가능	2



ID	업무구분(대)	업무구분(중)	요구사항	요구사항 내용	기능	중요도
C1	댓글(Comment)	댓글 조회	댓글을 조회	게시물 전체 댓글 조회	댓글은 한 페이지에 10개씩 최신순으로 보여줌.	5
C2	댓글(Comment)	댓글 작성	댓글을 작성	댓글 작성 버튼 클릭 시 해당 게시글에 댓글 작성	댓글 내용만 작성.	5
C3	댓글(Comment)	댓글 수정	댓글을 수정	댓글 수정 버튼 클릭 시 수정 해당 페이지에서 수정	기존에 작성되었던 댓글 내용을 보여줌.	4
C4	댓글(Comment)	댓글 수정	댓글을 수정	댓글 수정 버튼 클릭 시 수정 해당 페이지에서 수정	댓글을 작성한 회원만 수정 가능	5
C5	댓글(Comment)	댓글 수정	댓글을 수정	댓글 수정 완료	댓글이 업로드 된 시간은 수정된 시간으로 업데이트.	3
C6	댓글(Comment)	댓글 삭제	댓글을 삭제	삭제 버튼 누르면 댓글 삭제	댓글을 작성한 회원만 삭제 가능.	5
C7	댓글(Comment)	댓글 삭제	댓글을 삭제	댓글 삭제 확인	댓글 삭제 전에 알람 메시지 출력 후 동의하면 삭제 진행.	2

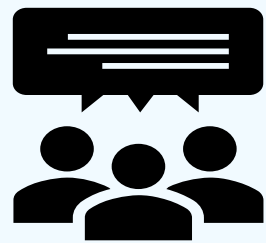
ID	업무구분(대)	업무구분(중)	요구사항	요구사항 내용	기능	중요도
U1	사용자(User)	회원가입	회원 가입	회원 가입 폼	name, password, email 정보 입력	5
U2	사용자(User)	로그인	로그인	Id, password를 통한 로그인	name, password 입력 후 로그인	4
U3	사용자(User)	로그아웃	로그아웃	로그아웃 버튼 클릭 시 로그아웃	회원 권한 기능 사용 불가	3
U4	사용자(User)	회원탈퇴	회원 삭제	마이 페이지에서 회원탈퇴 버튼 클릭	회원 정보 DB에서 삭제	5
U5	사용자(User)	정보 수정	회원 정보 수정	정보 수정 폼	회원 정보 DB에 업데이트	5
U6	사용자(User)	활동 조회	게시물 조회	사용자가 등록한 게시물을 보여줌	사용자 Id를 통해 게시물 리스트를 가져옴	2
U7	사용자(User)	활동 조회	게시물 조회	게시물 제목을 클릭하여 이동	해당 게시물의 상세 페이지로 이동	1
U8	사용자(User)	활동 조회	댓글 조회	사용자가 등록한 댓글을 보여줌	사용자 Id를 통해 댓글 리스트를 가져옴	2
U9	사용자(User)	활동 조회	댓글 조회	댓글 내용을 클릭하여 이동	댓글 Id를 통해 게시물 상세 페이지로 이동	1





### 실시간 피드백

코딩 진행 상황을 공유하고, 작업 내용에 대한 즉각적인 피드백을 주고받으며 효율적인 협업을 이루었습니다.  
이를 통해 코드 품질을 높이고, 프로젝트 진행 속도를 향상시킬 수 있었습니다.



### 커뮤니케이션 도구

프로젝트 진행 중 Discord와 엘리스 플랫폼을 통해 팀원들과 소통했습니다.  
Discord에서는 주로 채팅을 통한 실시간 대화를 나누고, 일정 관리 및 공지 사항을 공유했습니다.



### 화면공유 및 오류 해결

엘리스 플랫폼의 화면 공유 기능을 활용하여 각자 작업 중 발생한 오류나 어려운 지점을 공유했습니다.  
팀원들이 함께 화면을 보며 문제를 분석하고 해결 방안을 모색함으로써, 신속하게 오류를 수정할 수 있었습니다.

# 진행 사항

11-4(시작)

11-7

11-11

11-13

11월 4일

- 프로젝트 주제 선정
- ERD 설계 작업 진행

11월 5일

- 기본적인 CRUD 작성
- 와이어프레임을 작성
- 프로젝트 구조, UI 구성 결정
- 프로젝트 개발 스펙 설정

11월 6일

- 기능명세서 작성
- Comment, Post, Board의 기본적인 구조 설정 작업 완료

11월 7일 ~ 8일

- 로그인, 회원가입 기능 구현
- DB셋팅
- 추가 기능 구현

11월 11일

- 백엔드 작업 마무리
- 프론트엔드 작업 시작

11월 12일

- 초기 프론트엔드 디자인 및 스타일링 작업 진행

11월 13일

- 프론트엔드 디자인 완성

11월 14일

- 발표 준비 및 프로젝트 기간 동안의 내용 정리 작업

# 추가 기능



## 파일 추가

- MultipartFile 사용하여 구현
- 파일은 외부 경로에 저장
- 커스텀 어노테이션을 통해 유효성 검사 진행



## 검색

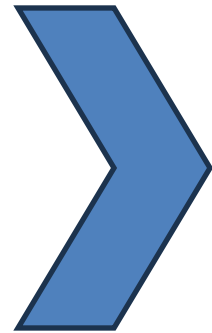
- Title 을 이용해 구현
- 검색한 내용을 정렬 조건에 맞게 출력
- Paging과 함께 작동가능



## 정렬 및 페이징

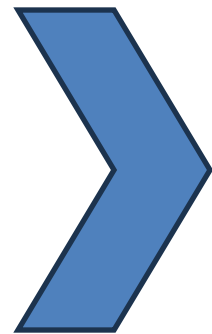
- JPA를 활용해 구현
- 최신순, 과거순 등 정렬 옵션 제공

문제



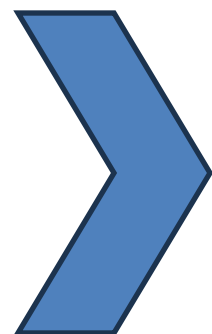
이미지 파일을 외부 경로에 저장했지만, 브라우저에서 이미지가 바로 로드되지 않음.

원인



파일을 외부 경로에 저장하면 프로젝트 내 static 폴더처럼 브라우저에서 직접 접근할 수 없어 외부 경로의 파일을 웹에서 접근할 수 있도록 URL 매핑 설정이 필요함.

해결



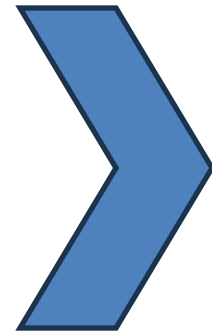
```
@Configuration
public class WebConfig implements WebMvcConfigurer {

    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("/files/**") // view에서 접근할 경로
                .addResourceLocations("file:///C:/Prj1/Files/"); // 실제 파일 저장 경로
    }
}
```

1. WebMvcConfigurer를 구현한 config 클래스를 생성하여 외부 경로를 URL에 매핑함.
2. 특정 URL 경로를 설정하여 이 경로로 들어오는 요청이 실제 외부 경로에 있는 파일로 연결되도록 지정.

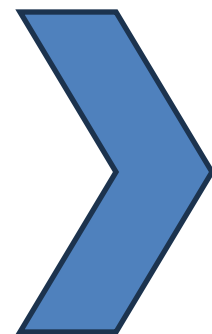
예) /files/로 시작하는 URL 경로로 들어오는 요청은 실제로 /C:/Prj1/Files 내의 파일로 매핑 됨.

문제



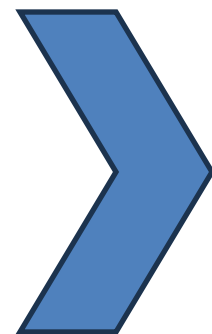
ID 하나로 같은 post를 여러 번 방문 해도 Post의 ViewCount가 중복적으로 증가하는 문제가 생김

원인



Post를 방문한 ID를 저장해주는 저장소의 부재

해결



```
@Slf4j
public class PostController {

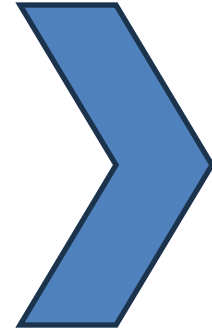
    private final Map<Long, Set<Long>> userViewRecords = new ConcurrentHashMap<>(); 1 usage

    private final PostService postService; 16 usages
```

PostController에 postId를 key값으로 하는 map을 생성하여 방문한 postId를 관리하여 해결

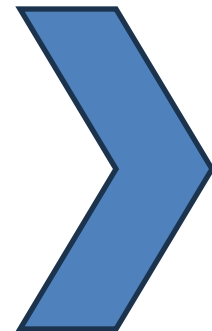


문제



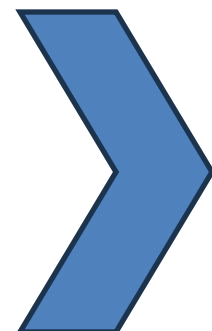
@PathVariable , @RequestParam 를 사용할때 Spring에서  
파라미터 오류가 발생

원인



Gradle 설정 문제로 추측

해결



```
@PostMapping("/{postId}/comments")  
public String saveComment(@PathVariable("postId") Long postId,  
                           @RequestParam("content") String content,  
                           @RequestParam("boardId") Long boardId,
```

@PathVariable , @RequestParam의 value 기본값을 명시 해주거나  
gradle이 아닌 maven으로 build 함으로써 오류를 해결

정시원

이번 팀 프로젝트를 통해 배운 내용을 실제로 적용해보면서 배웠던 내용을 조금 더 이해할 수 있었습니다. 그리고 프로젝트를 하면서 오류도 많이 나고 막혔을 때도 많았는데 그럴 때마다 팀원들과 서로 도움을 주고 받으면서 문제를 해결해 나갈 수 있어 좋았습니다.

김정학

팀 프로젝트를 진행하면서 코딩실력 뿐만 아니라 협업과 팀워크의 중요성을 깨닫게 되었습니다. 또한 이론으로 배운 기술들을 직접 사용해보면서 백엔드 기술에 대해 조금 더 친숙해지는 경험이 되었습니다.

김영철

프로젝트를 진행하면서 이론 공부 당시에는 충분히 이해하지 못했던 부분들을 실제로 적용해보면서 많은 깨달음을 얻었습니다. 실제 구현 과정에서 예상치 못한 오류들이 발생하면서 팀원들과 도움을 주고 받으면서 문제를 해결해 나갔고 저의 부족함을 채우는 계기가 된 것 같아서 좋았습니다. 이번 프로젝트를 통해 얻은 경험이 앞으로의 학습과 개발에 큰 밑거름이 될 것 같습니다.

[localhost:8080/board/list?](http://localhost:8080/board/list?)

# Q&A

Source

[- cloud\\_track / class\\_05 / web\\_project1 / Team05 / main · GitLab](#)

[-\(2\) 5팀](#)