지도기반 SNS

팀　　명 :　maps
지도 교수 : 이병천 교수님
팀　　장 :　김정식
팀　　원 :　김현욱
　　서영우
　　이용석

2022. 11
중부대학교 정보보호학과

목 차

1. 서론

1.1 연구 배경 ····· 4
1.2 연구 필요성 ····· 4
1.3 연구 목적 및 주제 선정 ····· 4

2. 관련 연구

2.1 JAVA ····· 4
2.2 Spring Boot····· 5
2.3 Gradle····· 5
2.4 JPA ····· 5
2.5 JavaScript ····· 5
2.6 Thymeleaf ····· 6
2.7 AWS····· 6
2.8 Heroku····· 6

3. 본론

3.1 시스템 구성 ····· 6
3.2 웹 사이트 ····· 12

4. 결론

4.1 결론 ····· 14
4.2 기대 효과 ····· 14

5. 별첨

5.1 소스 코드 ····· 15
5.2 발표 자료 ····· 2

1. 서론

1.1 연구 배경

Java Spring으로 웹기반 SNS를 만들어 이용자가 단순히 사진 및 글을 작성하는 것이 아닌 지도 기반으로 사진을 올려 다른 사용자들이 한눈에 보기 쉽도록 작성하는 웹페이지를 개발하였습니다.

1.2 연구 필요성

SNS의 발전으로 다양한 핫 플레이스 및 맛집들이 공유가 되어지고 있는데 위치를 정확히 알아내지 못하는 경우도 존재하고 안다고 하더라도 인터넷에 검색을 하는 일이 많다. 이를 해결하기 위해 더욱 직관적으로 위치를 볼 수 있도록 해주는 웹 기반 SNS를 개발하였다.

1.3 연구 목표

다양한 사람들이 자신만의 장소 및 공유하고 싶은 장소를 자유롭게 올리고 공유 가능하도록 하는 것이 목표이며, 추후 지도를 이용한 다양한 기능을 추가해 더 많은 사용자들이 즐길 수 있는 웹 SNS를 개발하는 것이 목표이다.

2. 관련 연구

2.1 JAVA

자바(Java)는 1995년 썬 마이크로시스템즈에서 발표한 객체 지향 프로그래밍 언어입니다. 자바는 가능한 적은 종속성을 갖도록 설계되었으며 "Programmers wirte once, run anywhere(WORA)" 와 같이 한번 작성한 코드를 모든 플랫폼에서 작동 시킬 수 있는 범용적인 언어입니다. 전 세계의 많은 Back end 개발자가 선택하는 언어이며 전 세계적으로 보고된 개발자는 9백만 명입니다. 또한 Android 앱 개발을 위한 유일한 공식 언어입니다.
자바는 Amazon, Twitter, Netflix 등 많은 서비스에서 사용하고 있으며

게임 콘솔, 슈퍼컴퓨터 등 많은 곳에서 실행 가능합니다.
대한민국 전자정부표준 프레임워크는 Java 프레임워크인 Spring 입니다

2.2 Spring

JAVA의 웹 프레임워크로 JAVA 언어를 기반으로 사용한다. JAVA로 다양한 어플리케이션을 만들기 위한 프로그래밍 틀이라 할 수 있다.
옛날에 비교하면 지금은 JAVA의 활용도가 높아졌고 따라서 프로젝트 규모도 커졌다. JAVA를 이용한 기술은 JSP, MyBatis, JPA 등 여러가지가 있는데 즉, 이 기술들이 프로젝트에 많이 쓰인다고 할 수 있다. Spring은 이 기술들을 더 편하게 사용하기 위해 만들어진 것이다.
프로젝트를 진행하다 보면 아무리 분업을 해도 분명 중복되는 코드가 있기 마련이다. Spring은 이런 중복코드의 사용률을 줄여주고, 비즈니스 로직을 더 간단하게 해줄 수 있다.
결론적으로 Spring이란 JAVA 기술들을 더 쉽게 사용할 수 있게 해주는 오픈소스 프레임 워크이다.

2.3 Gradle

Gradle은 그루비를 이용한 빌드 자동화 시스템이다. Groovy와 유사한 도메인 언어를 채용하였으며, 현재 안드로이드 앱을 만드는데 필요한 안드로이드 스튜디오의 공식 빌드 시스템이기도 하다. Java, C/C++, 파이썬 등과 같은 여러 가지 언어를 지원한다.

2.4 JPA

자바 퍼시스턴스 API 또는 자바 지속성 API(Java Persistence API, JPA)는 자바 플랫폼 SE와 자바 플랫폼 EE를 사용하는 응용프로그램에서 관계형 데이터베이스의 관리를 표현하는 자바 API이다.
기존에 EJB에서 제공되던 엔터티 빈(Entity Bean)을 대체하는 기술이다. 자바 퍼시스턴스 API는 JSR 220에서 정의된 EJB 3.0 스펙의 일부로 정의가 되어 있지만 EJB 컨테이너에 의존하지 않으며 EJB, 웹 모듈 및 Java SE 클라이언트에서 모두 사용이 가능하다. 또한, 사용자가 원하는 퍼시스턴스 프로바이더 구현체를 선택해서 사용할 수 있다.

2.5 JavaScript

자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이다. 이 언어는 웹 브라우저 내에서 주로 사용되며, 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다. 또한 Node.js와 같은 런타임 환경과 같이 서버 프로그래밍에도 사용되고 있다. 자바스크립트는 본래 넷스케이프 커뮤니케이션즈 코퍼레이션의 브렌
던 아이크(Brendan Eich)가 처음에는 모카(Mocha)라는 이름으로, 나중에는 라이브스크립트(LiveScript)라는 이름으로 개발하였으며, 최종적으로 자바스크립트가 되었다. 자바스크립트가 썬 마이크로시스템즈의 자바
와 구문이 유사한 점도 있지만, 이는 사실 두 언어 모두 C 언어의 기본 구문에 바탕을 뒀기 때문이고, 자바와 자바스크립트는 직접적인 연관성은 약하다. 이름과 구문 외에는 자바보다 셀프나 스킴과 유사성이 많다. 자바스크립트는 ECMA스크립트(ECMAScript)의 표준 사양을 가장 잘 구현한 언어로 인정받고 있으며 ECMAScript 5(ES5)까지는 대부분의 브라우저에서 기본적으로 지원되었으나 ECMAScript 6 이후부터는 브라우저 호환성을 위해 트랜스파일러로 컴파일된다.

2.6 Thymeleaf

Thymeleaf는 웹 및 웹이 아닌 환경 모두에서 작동할 수 있는 Java XML/XHTML/HTML5 템플릿 엔진입니다. MVC 기반 웹 애플리케이션의 뷰 레이어에서 XHTML/HTML5를 제공하는 데 더 적합하지만 오프라인 환경에서도 모든 XML 파일을 처리할 수 있습니다.

2.7 AWS S3

아마존 S3는 아마존 웹 서비스에서 제공하는 온라인 스토리지 웹 서비스이다. 아마존 S3는 웹 서비스 인터페이스를 통해 스토리지를 제공한다. 아마존이 S3를 출시하고 최초로 공개적으로 웹서비스를 이용할 수 있게 된 것은 2006년 3월 미국 그리고 유럽은 2007년 11월이다.

2.8 Heroku

헤로쿠 주식회사는 웹 애플리케이션 배치 모델로 사용되는 여러 프로그래밍 언어를 지원하는 클라우드 PaaS이다.

3. 본론

3.1 시스템 구성

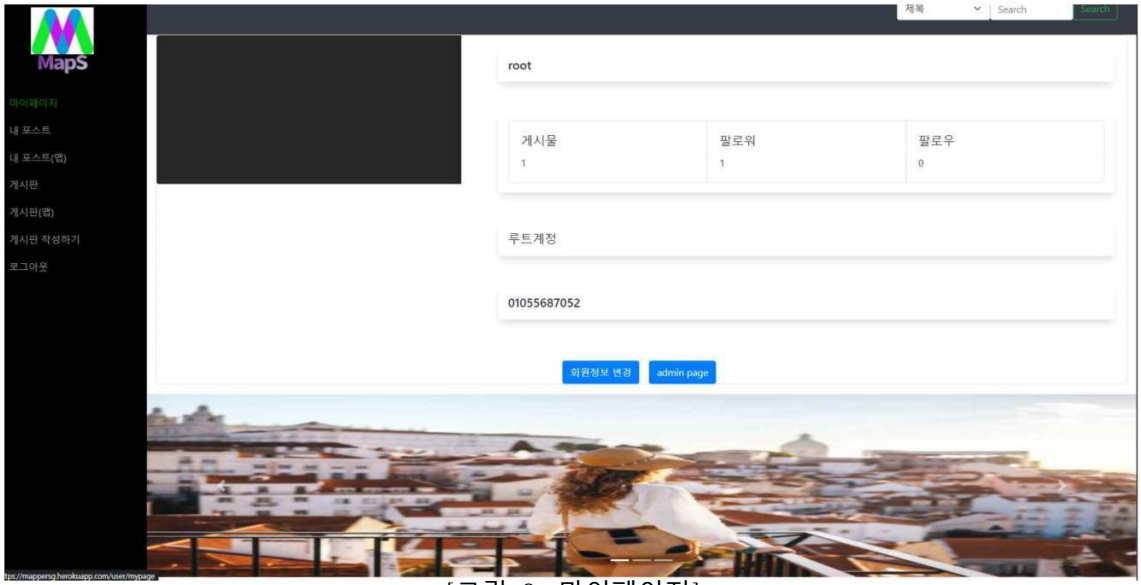
백엔드는 JAVA Spring Boot, Security를 사용하고, 프론트는 Spring과 자주 사용하는 Thymeleaf를 사용하여 개발하였다.
Kakaomap API를 사용하여 지도상에 다양한 기능을 구현하였으며, ajax를 사용하여 좋아요 및 팔로우 등의 기능을 비동기식으로 처리하였다.

3.2 웹사이트

Java

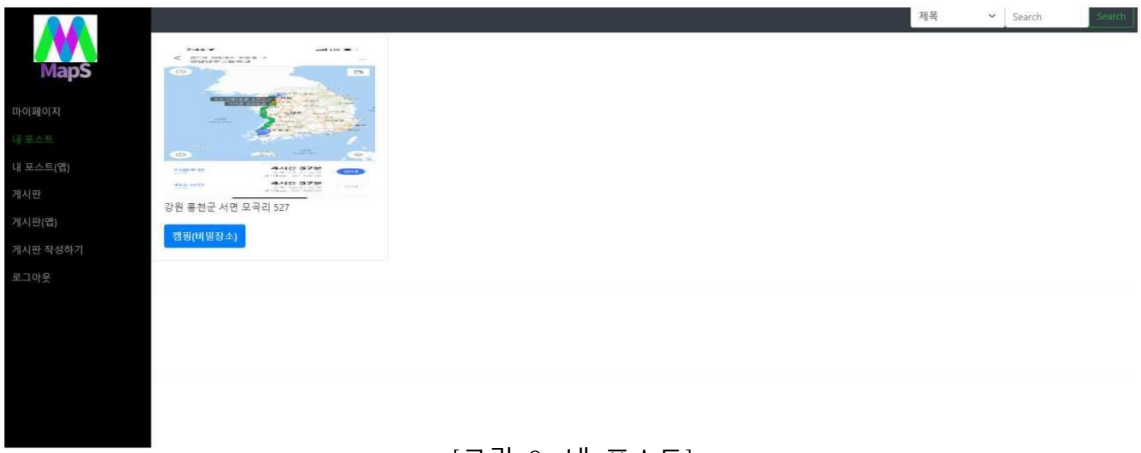
[그림 1. 웹페이지 로그인, 회원가입 화면]

먼저 웹페이지에 접속하면 메인페이지, 회원가입, 로그인 과정을 통해 실제 웹페이지에 접속이 가능하다.



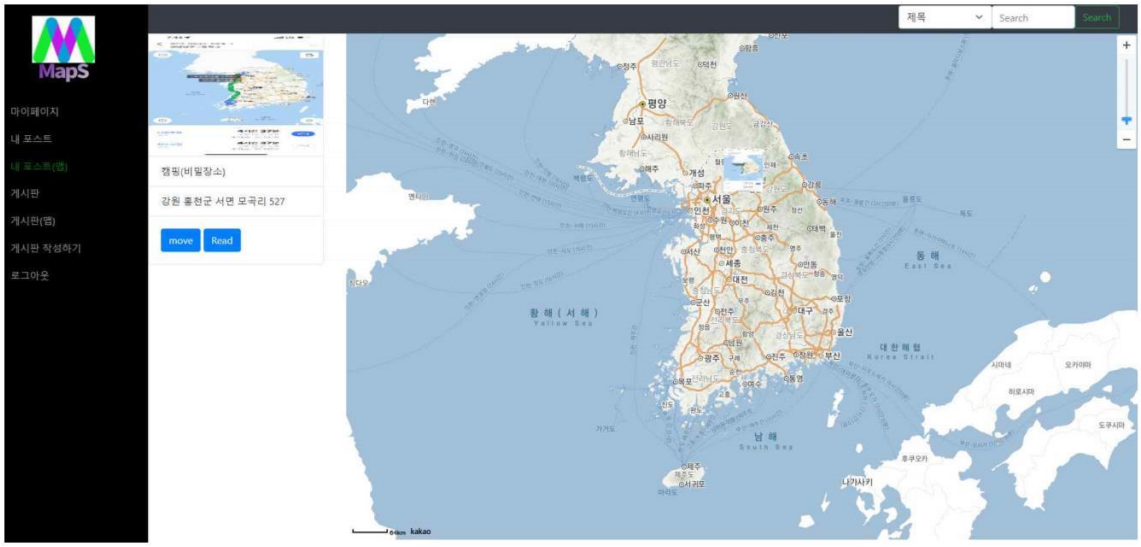
[그림 2. 마이페이지]

마이페이지에서는 개인정보와 게시물,팔로워,팔로우 수를 확인 할 수 있고, 회원정보 변경 또한 가능하다. 루트계정일 경우 루트페이지에 접속도 가능하다.



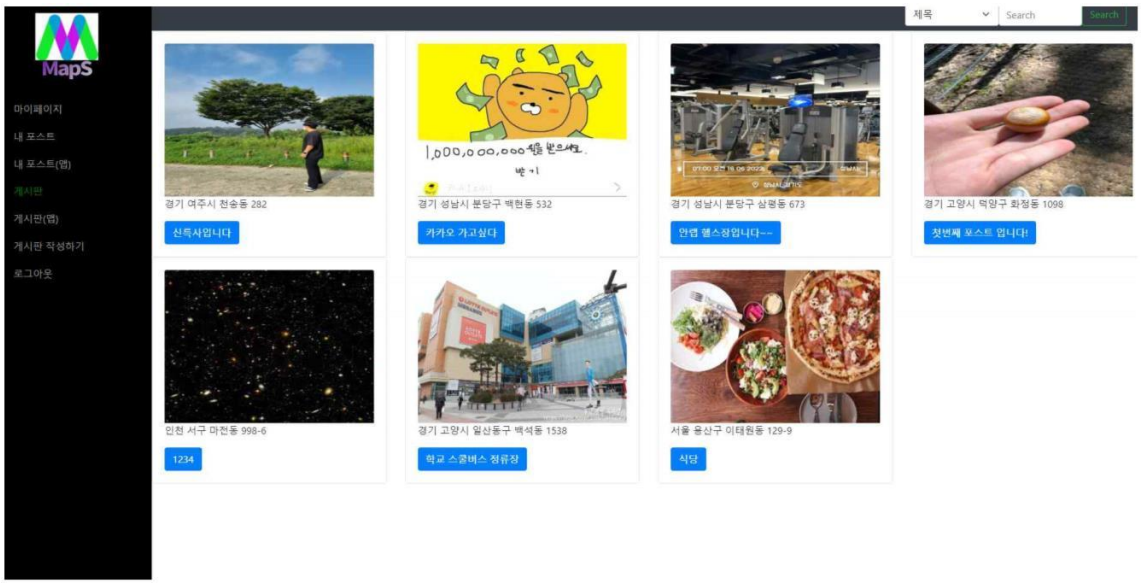
[그림 3. 내 포스트]

내 포스트 페이지에서는 내가 올린 글을 모두 확인할 수 있다.



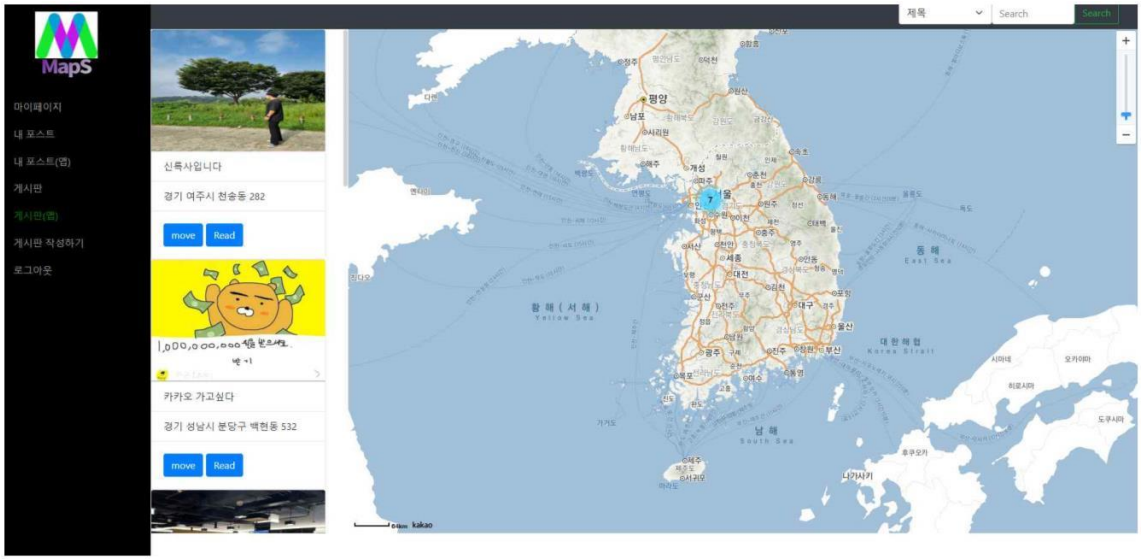
[그림 4. 내 포스트(지도)]

내 포스트(지도) 페이지에서는 내가 올린글과 함께 지도에 사진이 등록된 화면까지 볼 수 있다.



[그림 5. 전체 포스트]

전체 포스트에는 모든 사용자들이 전체공개로 올린 게시물을 볼 수 있다.

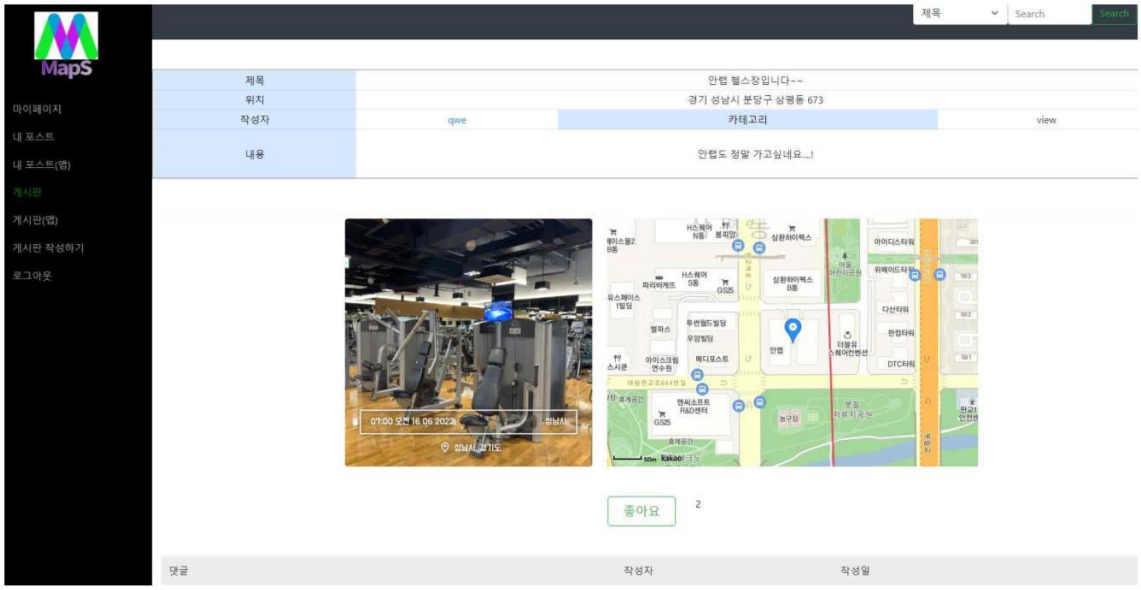


[그림 6. 전체 포스트(지도)]

전체 포스트(지도)에는 모든 사용자들이 전체공개로 올린 게시물을 지도형식과 게시물 형식으로 확인이 가능하다.

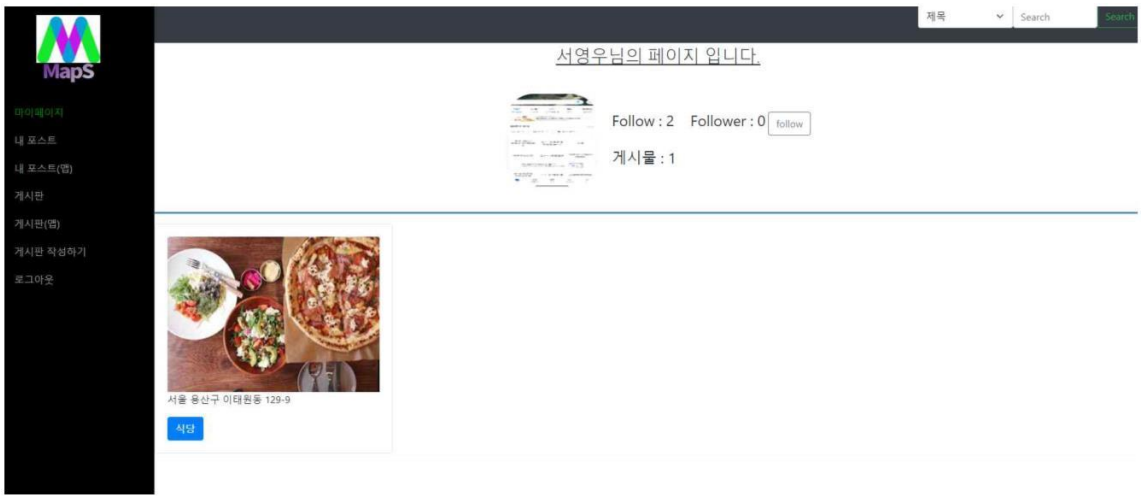
[그림 7. 게시판 작성]

게시판 작성 페이지에서는 사진, 위치, 카테고리, 공개여부 등을 선택 및 등록하여 글을 쓸 수 있다.



[그림 8. 게시물 뷰 페이지]

게시물 뷰 페이지에서는 게시물의 정보를 읽을 수 있고, 좋아요, 댓글 등의 기능이 가능하다.



[그림 9. 개인 페이지]

개인 페이지에서는 해당 사용자가 올린 전체공개 글을 확인 할 수 있고, 팔로워, 팔로우, 등록 게시물 수를 확인 할 수 있다. 이때 팔로우를 하게 되면 친구공개로 등록했던 게시물도 확인이 가능하다.

4. 결론

4.1 결론

지도기반 SNS를 통하여 사용자들이 여행 및 출장 시 주변의 다양한 정보를 한눈에 보기 쉽게 획득할 수 있다.

4.2 기대효과

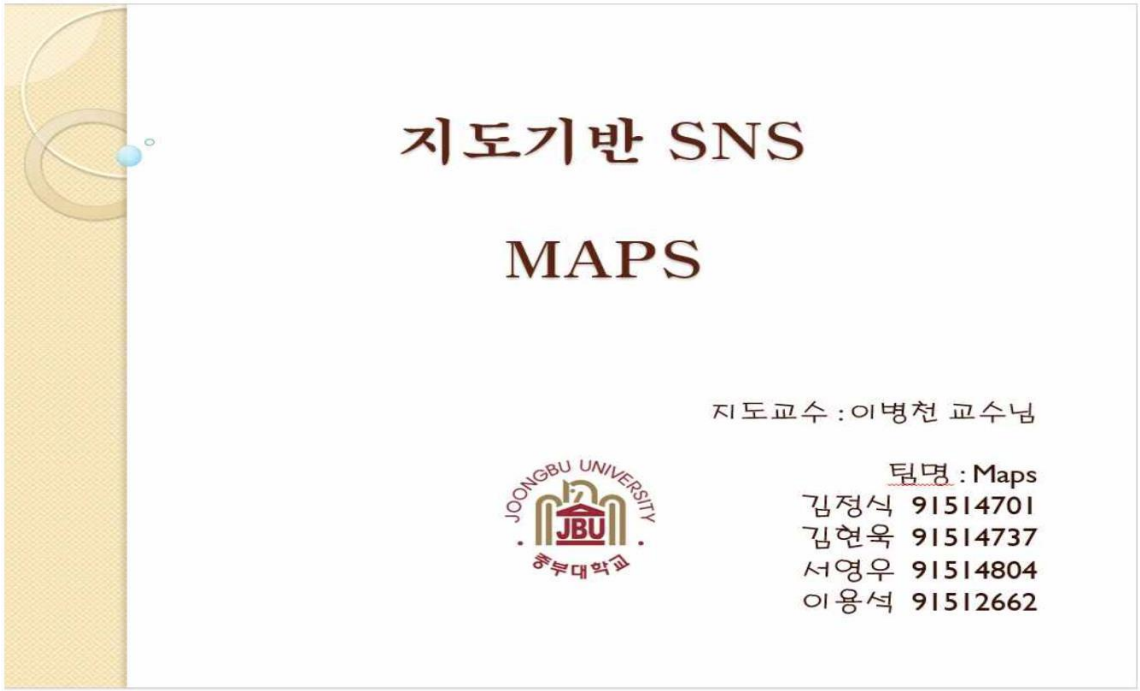
프로젝트 발표 후 계속해서 방송 및 Category분류 등 다양한 기능을 추가하여 사용자들이 더 즐길 수 있는 컨텐츠를 제공하여 많은 사용자들이 유입할 수 있게 하고, 지도기반 SNS를 모바일 앱까지 출시하여 많은 사용자들이 지도상에서 정보를 쉽게 얻고 즐길 수 있도록 이용할 수 있을 것이다.

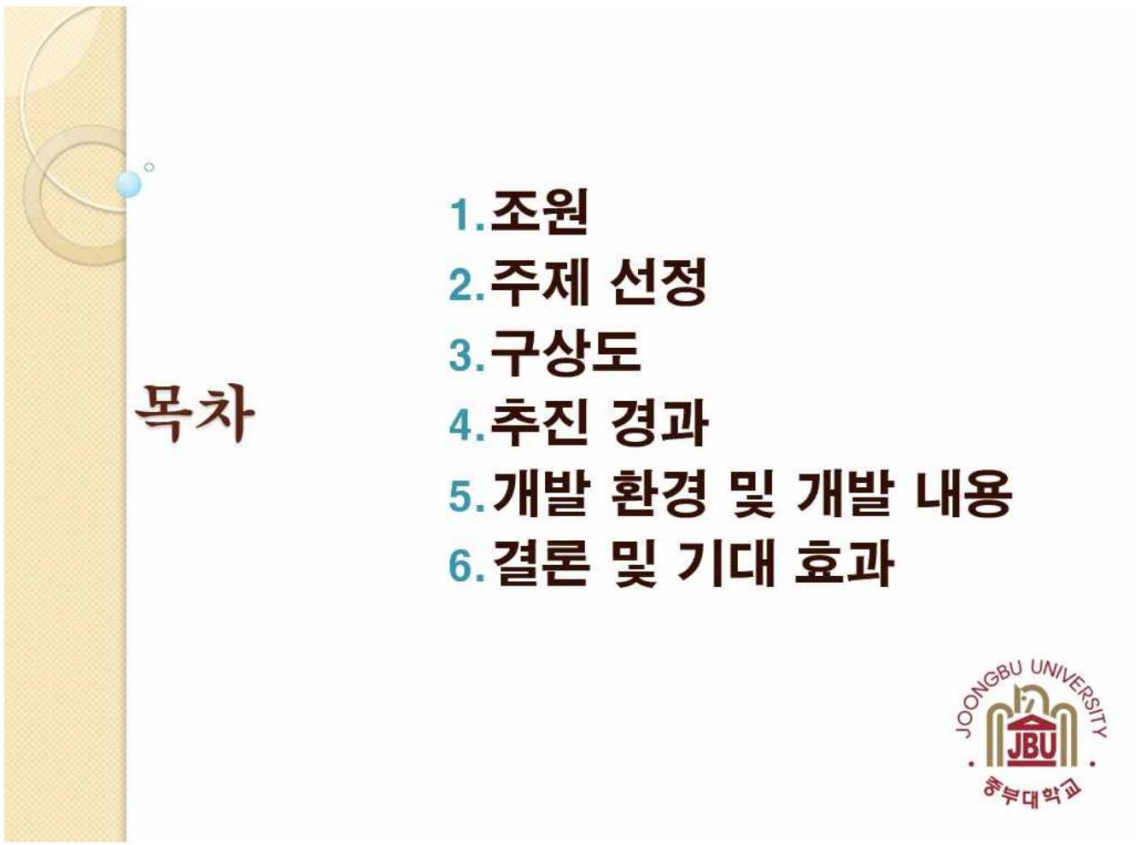
5. 별첨

5.1 소스코드

https://github.com/wjdtlr0822/jol

5.2 발표자료


지도기반 SNS
MAPS
지도교수 : 이병천 교수님
팀명 : Maps
김정식 91514701
김현욱 91514737
서영우 91514804
이용석 91512662


목차
1. 조원
2. 주제 선정
3. 구상도
4. 추진 경과
5. 개발 환경 및 개발 내용
6. 결론 및 기대 효과

조원

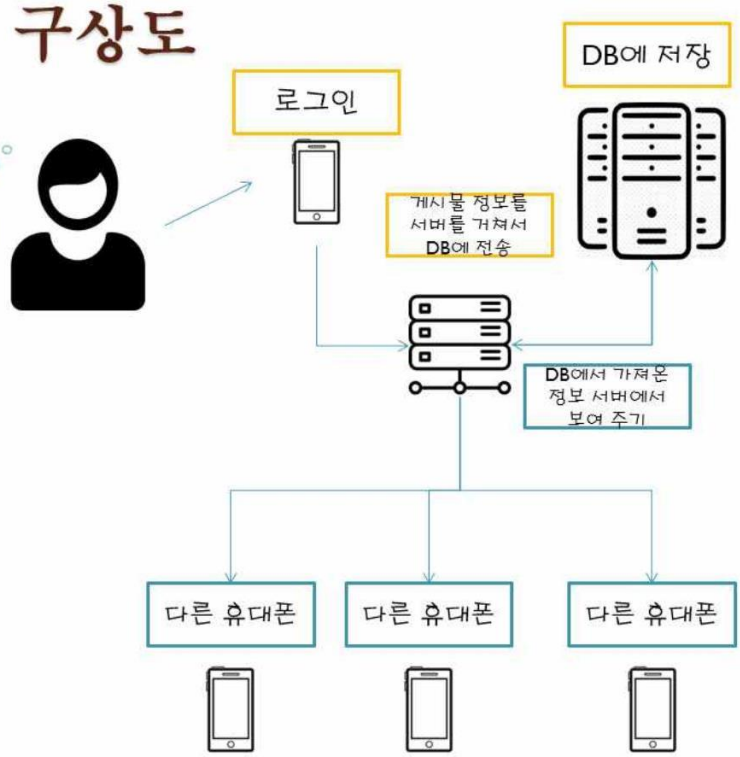
이 름	역 할
김정식 (팀장)	프로그램 개발
김현욱	자료수집 PPT 작성
이용석	디자인 및 보고서작성
서영우	디자인 및 보고서작성

주제 선정

- ✓ **지도를 통해서 특정 위치를 친구들과 함께 공유할 수 있는 새로운 플랫폼을 구축하기 위해 선정하였다.**
  (EX. 맛 집 ,풍경)

- ✓ **친구들과 급작스럽게 모임을 할 때 빠르고 정확하게 장소를 공유할 수 있는 서비스를 구축하고자 하였다.**

구상도



추진 경과

	3월	4월	5월	6월	7월	8월	9월	10월
자료 조사 및 연구								
프로트엔드 개발								
백엔드 개발								
지도 시스템								
페이지 보완 및 점검								

개발 환경

spring boot

Thymeleaf

kakaomap
Maps API

JPA
Java Persistence API

개발 내용(1/11)

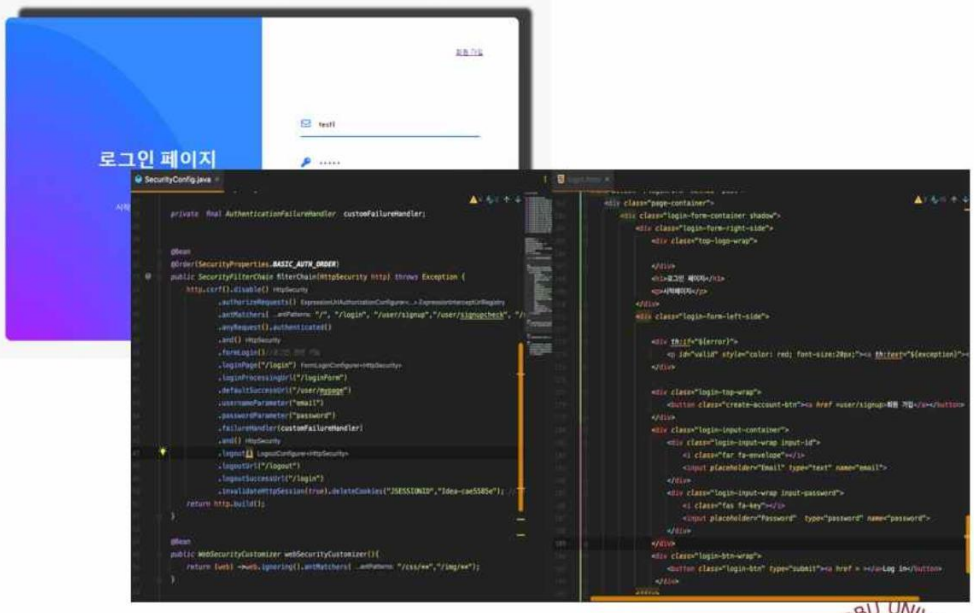
개발 내용(2/11)


```
@GetMapping("/user/signup")
public String signup() { return "user/signup"; }

@PostMapping("/user/signup")
public String signupcheck(Register register, @RequestParam("profile") MultipartFile file, Authentication authentication) throws IOException {

    if (!file.isEmpty()) {
        String filename = UUID.randomUUID() + "-" + file.getOriginalFilename();
        String imgurl = amazonS3Service.upload(file, filename);
        register.setProfileImage(imgurl);
        if (userService.save(register)) {
            return "redirect:login";
        } else {
            return "redirect:signuperror";
        }
    } else {
        return "redirect:signuperror";
    }

}
```

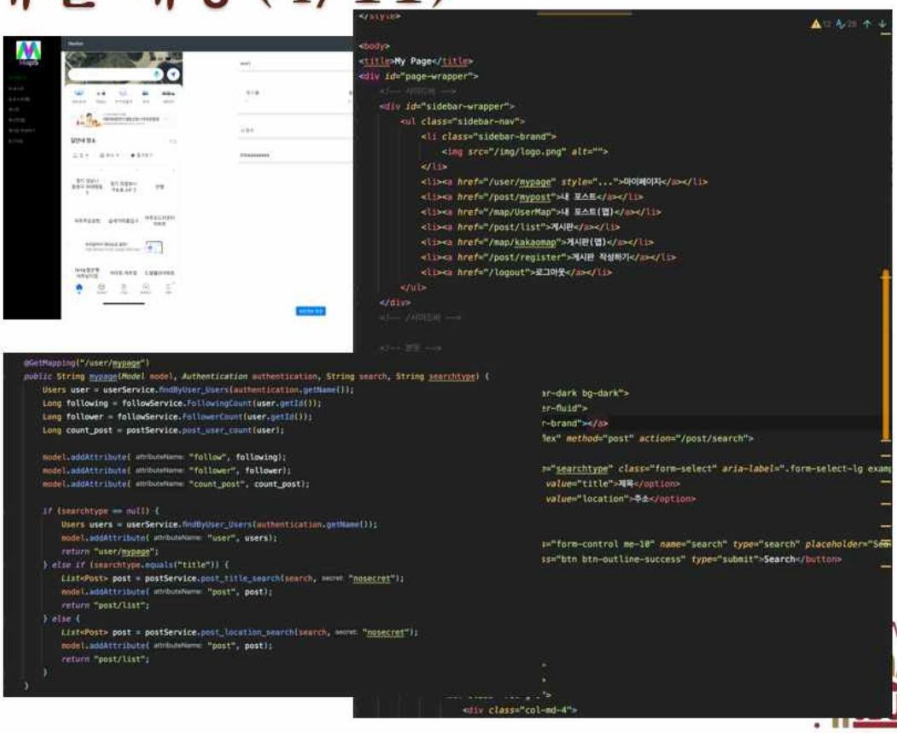
```
public String upload(MultipartFile multipartFile,String s3FileName) throws IOException{
    ObjectMetadata objMeta = new ObjectMetadata();
    objMeta.setContentLength(multipartFile.getInputStream().available());
    amazonS3.putObject(new PutObjectRequest(bucket, s3FileName, multipartFile.getInputStream(), objMeta));
    return amazonS3.getUrl(bucket, s3FileName).toString();
}
```

```
@Transactional
public boolean save(Register register) {
    if (userRepository.findUserByEmail(register.getEmail()).isPresent()) {
        return false;
    }
    register.setPassword(encoder.encode(register.getPassword()));
    userRepository.save(register.toEntity());
    return true;
}
```

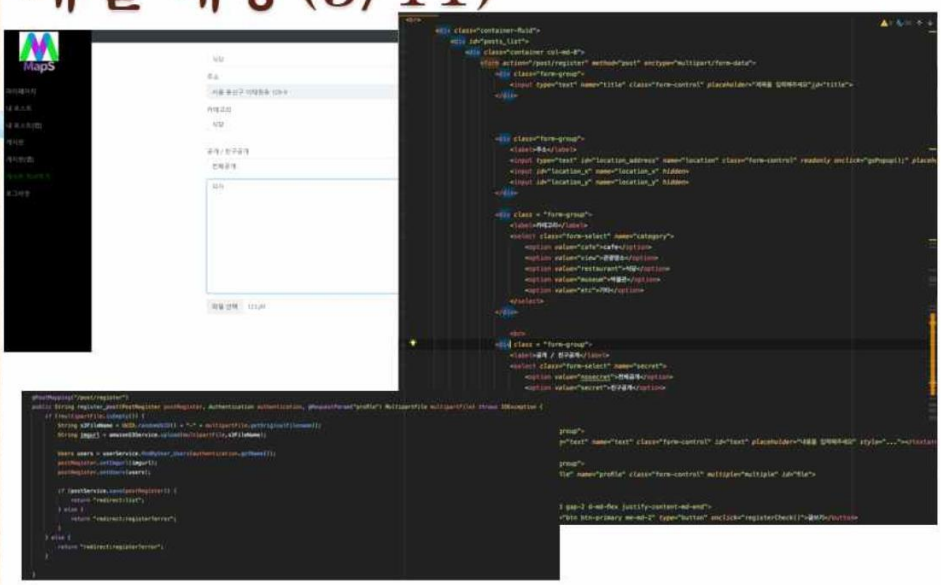
개발 내용(3/11)



개발 내용(4/11)



개발 내용(5/11)

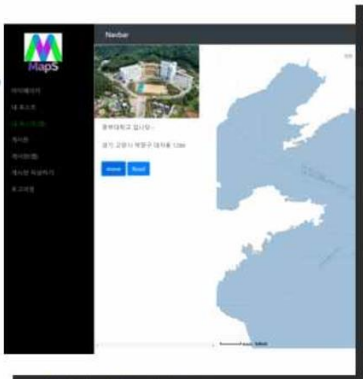


개발 내용(6/11)


```
<div class="row">
    <div class="col-sm-3" th:each="post:${post}">
        <div class="card">
            <div class="card-body">
                <img th:src="${post.imgurl}" class="card-img-top" alt="..." width="200"  height="250">
                <p class="card-text" th:text="${post.location}">location</p>
                <a th:text="${post.title}" th:href="@{/post/postview/{id}(id=${post.id})}" class="btn btn-primary">Go somewhere</a>
            </div>
        </div>
    </div>
```

```
@GetMapping("post/mypost")
public String mypost(Model model, Authentication authentication, String search, String searchtype) {
    Users users = userService.findByUser_Users(authentication.getName());
    if (searchtype == null) {
        List<Post> findall = postService.findPost_User(users);
        model.addAttribute( attributeName: "post", findall);
        return "post/mypost";
    } else if (searchtype.equals("title")) {
        List<Post> post = postService.post_userAndtitle_search(users, search);
        model.addAttribute( attributeName: "post", post);
        return "post/mypost";
    } else {
        List<Post> post = postService.post_userAndlocation_search(users, search);
        model.addAttribute( attributeName: "post", post);
        return "post/mypost";
    }
}
```


개발 내용(7/11)



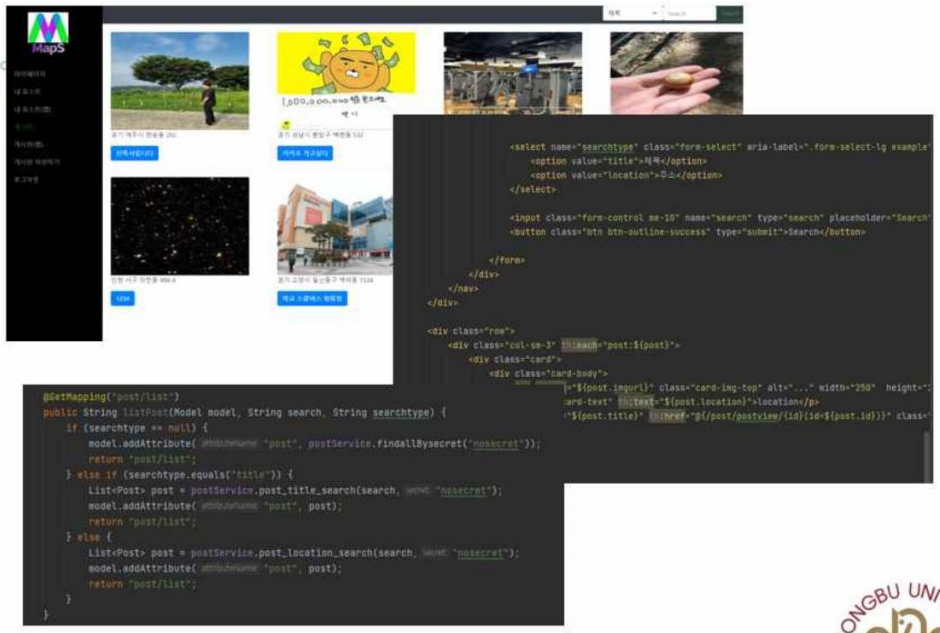

```
@GetMapping("map/userMap")
public String usermap(Model model, Authentication authentication, String search, String searchtype) {
    if (searchtype == null) {
        Users users = userService.findByUser_Users(authentication.getName());
        List<Post> post = postService.findPost_User(users);

        Gson gson = new Gson();
        JsonArray jArray = new JsonArray();

        Iterator<Post> postIt = post.iterator();
        while (postIt.hasNext()) {
            Post post1 = postIt.next();
            JsonObject object = new JsonObject();
            String category = post1.getCategory();
            String location_x = post1.getLocation_x();
            String location_y = post1.getLocation_y();
            String location = post1.getLocation();
            String title = post1.getTitle();
            String text = post1.getImgurl();
            String imgurl = post1.getImgurl();

            object.addProperty( property: "category", category);
            object.addProperty( property: "location_x", location_x);
            object.addProperty( property: "location_y", location_y);
            object.addProperty( property: "location", location);
            object.addProperty( property: "title", title);
            object.addProperty( property: "text", text);
                                                    ("imgurl", imgurl);

                                                    ray);

                                                    ("json", json);
                                                    ("post", post);
```

```
<div class="left" style="...">
<div th:each="post:${post}">
    <div class="card" style="...">
        <img th:src="${post.imgurl}" height="200" width="100" class="card-img-top" alt="...">
        <ul class="list-group list-group-flush">
            <li class="list-group-item" th:text="${post.title}">An item</li>
            <li class="list-group-item" th:text="${post.location}">An item</li>
        </ul>
        <div class="card-body">
            <div class="d-grid gap-2 d-md-block">
                <button class="btn btn-primary" th:onclick="movemap([[${post['location_x']}]],
                <a th:href="@{/post/postview/{id}(id=${post.id})}" class="btn btn-primary">Read
            </div>
        </div>
    </div>
</div>
</div>

<div class="right">
    <div id="map" style="..."></div>
</div>
</div>
```

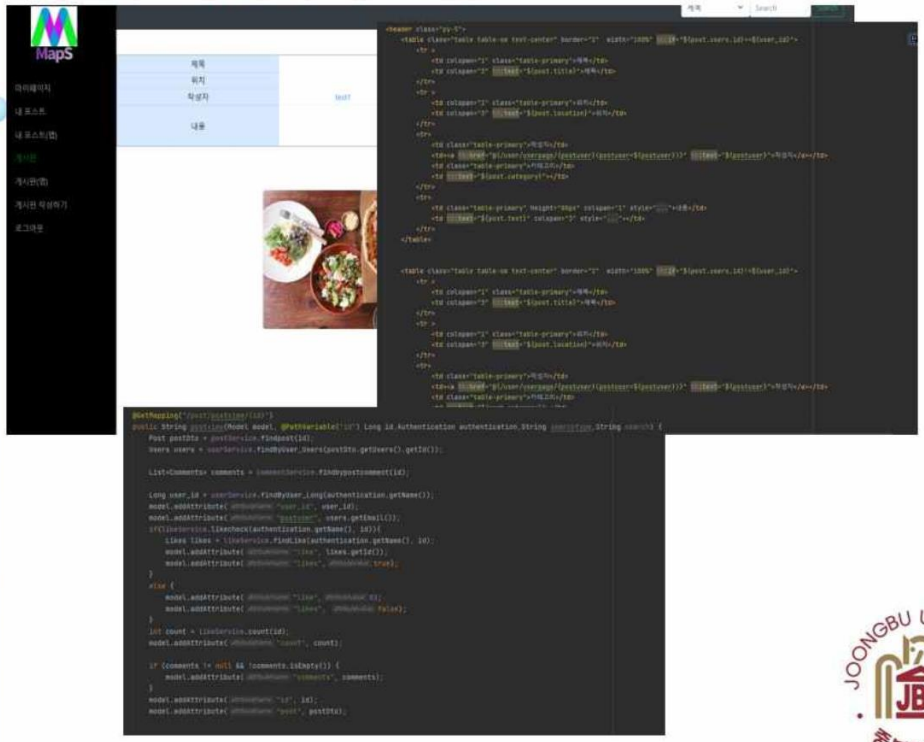
목차

1. 조원
2. 주제 선정
3. 구상도
4. 추진 경과
5. 개발 환경 및 개발 내용
6. 결론 및 기대 효과

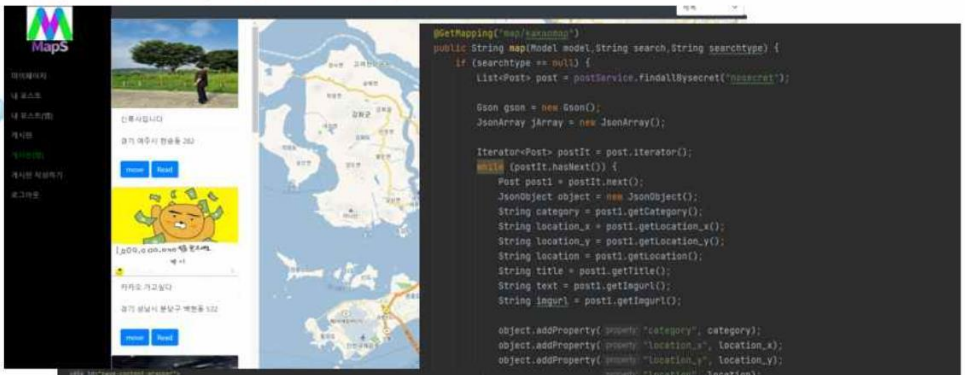
개발 내용(8/11)

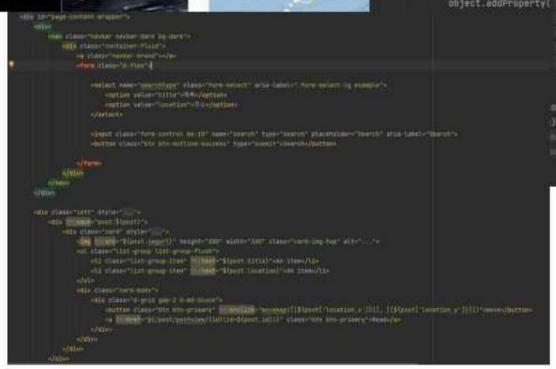


개발 내용(9/11)

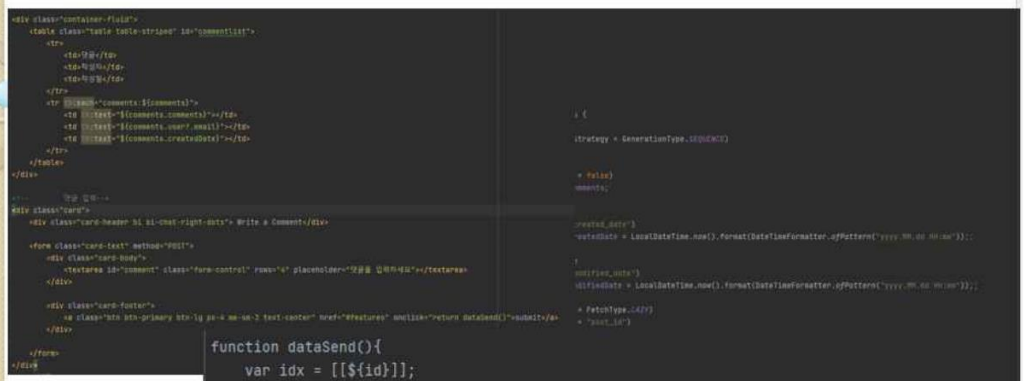


개발 내용(10/11)





개발 내용(11/11)



```
function dataSend(){
    var idx = [[${id}]];
    var commentT={
        id : idx,
        comment: $("#comment").val()
    };

    $.ajax({
        url: "/dataSend",
        type : "POST",
        data : commentT,
        })
        .done(function (fragment){
            $("#commentlist").replaceWith(fragment);
        });
}
```