

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



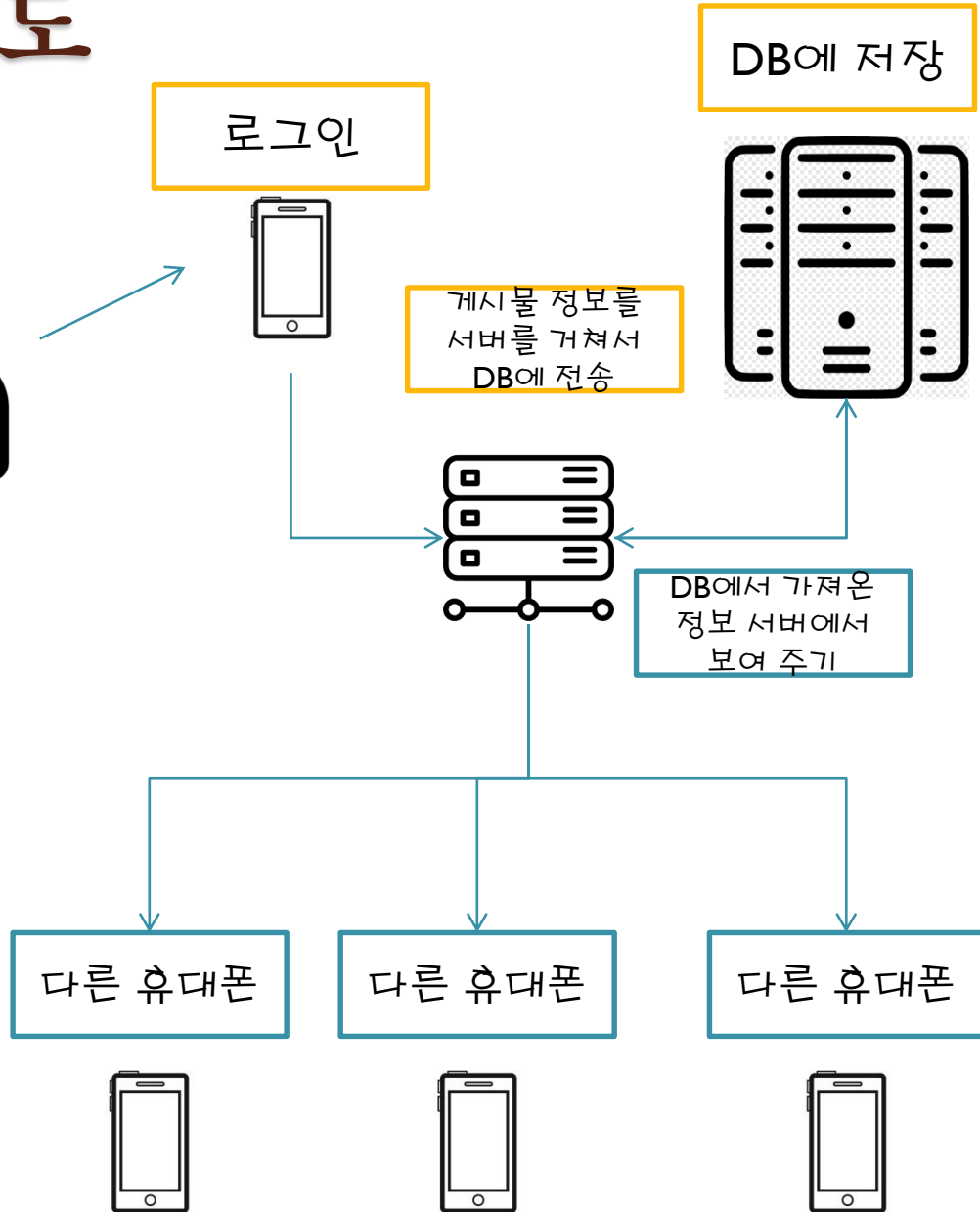
조원

| 이름 | 역할 |
|----------|-------------|
| 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.

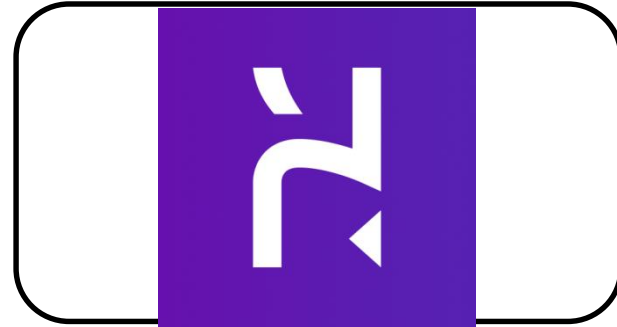
구상도



추진 경과

| | 3월 | 4월 | 5월 | 6월 | 7월 | 8월 | 9월 | 10월 |
|----------------|----|----|----|----|----|----|----|-----|
| 자료 조사 및 연구 | ■ | ■ | ■ | □ | □ | □ | □ | □ |
| 프론트엔드 개발 | □ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| 백엔드 개발 | □ | ■ | ■ | ■ | ■ | ■ | ■ | □ |
| 지도 시스템 | □ | □ | ■ | ■ | ■ | ■ | ■ | □ |
| 페이지 보완 및 점검 | □ | □ | □ | □ | □ | ■ | ■ | ■ |

개발 환경



개발 내용(1/11)



```
<body id="page-top">
  <!-- Navigation -->
  <div class="logo">
    <i class="fa fa-plane" aria-hidden="true"><span>MAPS</span></i>
  </div>
  <!-- Header Starts -->
  <section id="Banner" class="content-section">
    <div class="container content-wrap text-center">
      <h1>MAPS</h1>
      <h3>
        <em>환영합니다</em>
      </h3>
      <a class="btn btn-primary btn-xl smooth-scroll" href="/login">LOG IN</a>
      <a class="btn btn-primary btn-xl smooth-scroll" href="/user/signup">회원가입</a>
    </div>
    <div class="overlay"></div>
  </section>
```


개발 내용(2/11)

maps register

test1

서영우

파일 선택 1.jpg

01011111111

SUBMIT

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

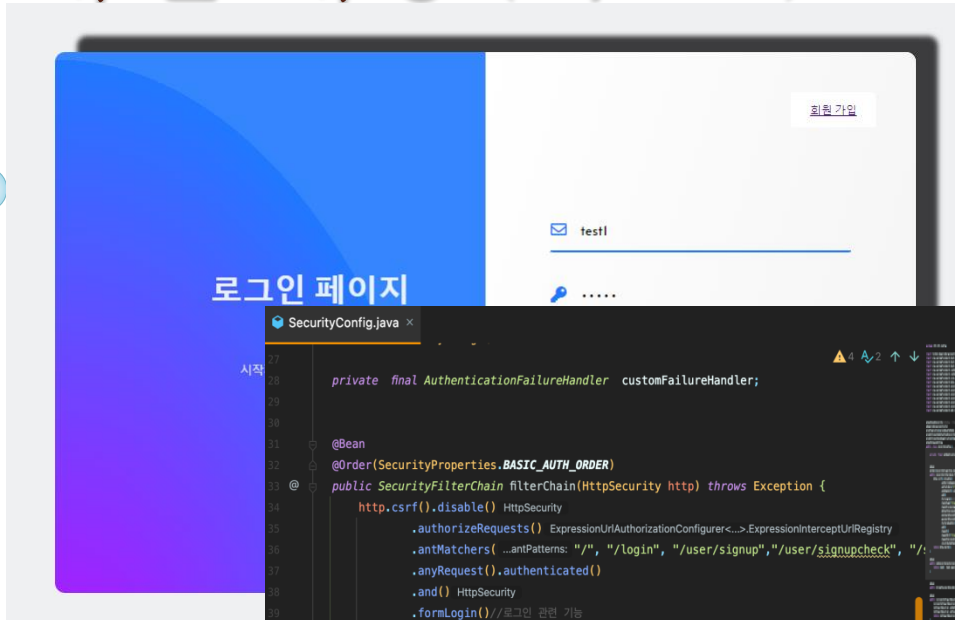
@PostMapping("/user/signup")
public String signupcheck(Register register, @RequestParam("profile") MultipartFile file, Authentication authentication) throws IOException {

    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:signup?error";
        }
    } else {
        return "redirect:signup?error";
    }
}
}
```

```
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;
}
```

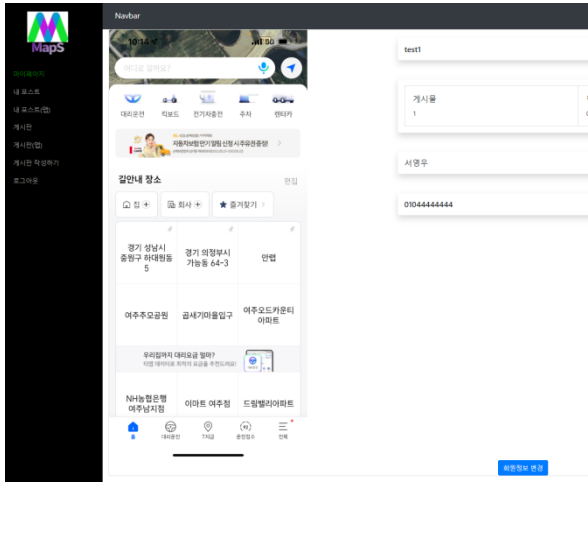
개발 내용 (3/11)



```
SecurityConfig.java
시작
27 private final AuthenticationFailureHandler customFailureHandler;
28
29
30
31 @Bean
32 @Order(SecurityProperties.BASIC_AUTH_ORDER)
33 @Configuration
34 public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
35     http.csrf().disable() HttpSecurity
36         .authorizeRequests() ExpressionUrlAuthorizationConfigurer<...>.ExpressionInterceptUrlRegistry
37         .antMatchers( ...antMatchers: "/", "/login", "/user/signup", "/user/signupcheck", "/:
38         .anyRequest().authenticated()
39         .and() HttpSecurity
40         .formLogin() // 로그인 관련 기능
41         .loginPage("/login") FormLoginConfigurer<HttpSecurity>
42         .loginProcessingUrl("/loginForm")
43         .defaultSuccessUrl("/user/mypage")
44         .usernameParameter("email")
45         .passwordParameter("password")
46         .failureHandler(customFailureHandler)
47         .and() HttpSecurity
48         .logout() LogoutConfigurer<HttpSecurity>
49         .logoutUrl("/logout")
50         .logoutSuccessUrl("/login")
51         .invalidateHttpSession(true).deleteCookies("JSESSIONID","Idea-cae5585e"); //
52     }
53     return http.build();
54 }
55
56 @Bean
57 public WebSecurityCustomizer webSecurityCustomizer(){
58     return (web) ->web.ignoring().antMatchers( ...antMatchers: "/css/**","/img/**");
59 }
60 }

login.html
162 <div class="page-container">
163
164 <div class="login-form-container shadow">
165 <div class="login-form-right-side">
166 <div class="top-logo-wrap">
167
168 </div>
169 <h1>로그인 페이지</h1>
170 <p>시작페이지</p>
171 </div>
172 <div class="login-form-left-side">
173
174 <div th:if="${error}">
175 <p id="valid" style="color: red; font-size:20px;"><a th:text="${exception}"></a>
176 </div>
177
178 <div class="login-top-wrap">
179 <button class="create-account-btn"><a href =user/signup>회원 가입</a></button>
180 </div>
181 <div class="login-input-container">
182 <div class="login-input-wrap input-id">
183 <i class="far fa-envelope"></i>
184 <input placeholder="Email" type="text" name="email">
185 </div>
186 <div class="login-input-wrap input-password">
187 <i class="fas fa-key"></i>
188 <input placeholder="Password" type="password" name="password">
189 </div>
190 </div>
191 <div class="login-btn-wrap">
192 <button class="login-btn" type="submit"><a href = ></a>Log in</button>
193 </div>
194 </div>
195 </div>
```

개발 내용(4/11)



```
</style>
<body>
<title>My Page</title>
<div id="page-wrapper">
  <!-- 사이드바 -->
  <div id="sidebar-wrapper">
    <ul class="sidebar-nav">
      <li class="sidebar-brand">
        
      </li>
      <li><a href="/user/mypage" style="...">마이페이지</a></li>
      <li><a href="/post/mypost">내 포스트</a></li>
      <li><a href="/map/UserMap">내 포스트 (맵)</a></li>
      <li><a href="/post/list">게시판</a></li>
      <li><a href="/map/kakaomap">게시판 (맵)</a></li>
      <li><a href="/post/register">게시판 작성하기</a></li>
      <li><a href="/logout">로그아웃</a></li>
    </ul>
  </div>
  <!-- /사이드바 -->
  <!-- 본문 -->
  <div class="col-md-4">
```

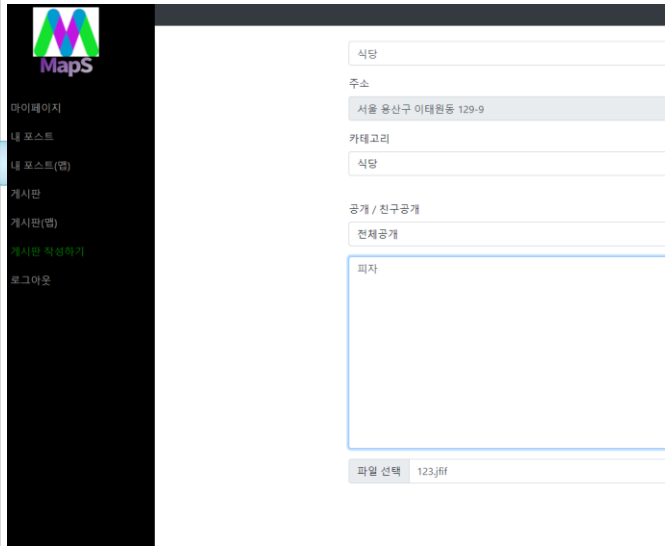
```
@GetMapping("/user/mypage")
public String mypage(Model model, Authentication authentication, String search, String searchtype) {
    Users user = userService.findByUser_Users(authentication.getName());
    Long following = followService.FollowingCount(user.getId());
    Long follower = followService.FollowerCount(user.getId());
    Long count_post = postService.post_user_count(user);

    model.addAttribute( attributeName: "following", following);
    model.addAttribute( attributeName: "follower", follower);
    model.addAttribute( attributeName: "count_post", count_post);

    if (searchtype == null) {
        Users users = userService.findByUser_Users(authentication.getName());
        model.addAttribute( attributeName: "user", users);
        return "user/mypage";
    } else if (searchtype.equals("title")) {
        List<Post> post = postService.post_title_search(search, secret: "nosecret");
        model.addAttribute( attributeName: "post", post);
        return "post/list";
    } else {
        List<Post> post = postService.post_location_search(search, secret: "nosecret");
        model.addAttribute( attributeName: "post", post);
        return "post/list";
    }
}
```



개발 내용 (5/11)



```
<div class="container-fluid">
  <div id="posts_list">
    <div class="container col-md-8">
      <form action="/post/register" method="post" enctype="multipart/form-data">
        <div class="form-group">
          <input type="text" name="title" class="form-control" placeholder="제목을 입력해주세요" id="title">
        </div>

        <div class="form-group">
          <label>주소</label>
          <input type="text" id="location_address" name="location" class="form-control" readonly onclick="goPopup();" placeholder="주소 입력" />
          <input id="location_x" name="location_x" hidden>
          <input id="location_y" name="location_y" hidden>
        </div>

        <div class="form-group">
          <label>카테고리</label>
          <select class="form-select" name="category">
            <option value="cafe">cafe</option>
            <option value="view">관광명소</option>
            <option value="restaurant">식당</option>
            <option value="museum">박물관</option>
            <option value="etc">기타</option>
          </select>
        </div>

        <br>

        <div class="form-group">
          <label>공개 / 친구공개</label>
          <select class="form-select" name="secret">
            <option value="nosecret">전체공개</option>
            <option value="secret">친구공개</option>
          </select>
        </div>

        <div class="form-group">
          <input type="text" name="profile" class="form-control" id="profile" placeholder="내용을 입력해주세요" style="width: 100%; height: 40px;" />
          <input type="file" name="profile" class="form-control" multiple="multiple" id="file">
        </div>

        <div class="form-group">
          <input type="button" value="등록" class="btn btn-primary me-md-2" type="button" onclick="registerCheck()" />
          <input type="button" value="취소" class="btn btn-primary me-md-2" type="button" />
        </div>
      </form>
    </div>
  </div>
</div>
```

```
@PostMapping("/post/register")
public String register_post(PostRegister postRegister, Authentication authentication, @RequestParam("profile") MultipartFile multipartFile) throws IOException {
    if (!multipartFile.isEmpty()) {
        String s3FileName = UUID.randomUUID().toString() + "-" + multipartFile.getOriginalFilename();
        String imgUrl = amazonS3Service.upload(multipartFile, s3FileName);

        Users users = userService.findByUser(authentication.getName());
        postRegister.setImgurl(imgurl);
        postRegister.setUsers(users);

        if (postService.save(postRegister)) {
            return "redirect:list";
        } else {
            return "redirect:register?error";
        }
    } else {
        return "redirect:register?error";
    }
}
```



개발 내용(6/11)



마이페이지

내 포스트

내 포스트(맵)

게시판

게시판(맵)

게시판 작성하기

로그아웃



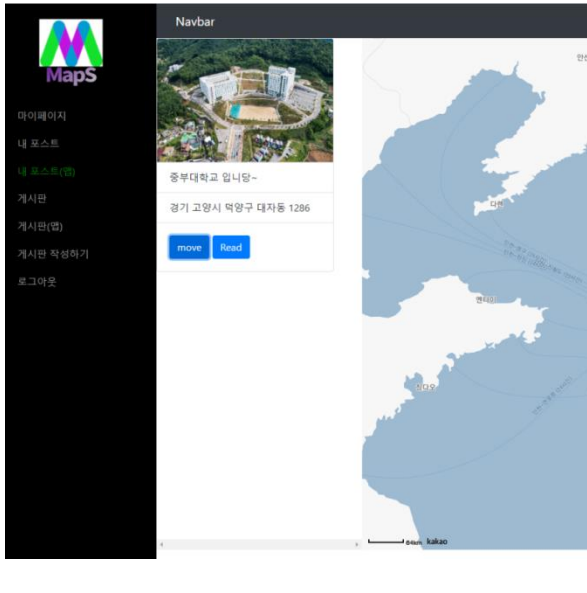
서울 용산구

식당

```
<div class="row">
  <div class="col-sm-3" th:each="post:${post}">
    <div class="card">
      <div class="card-body">
        
        <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>
</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();

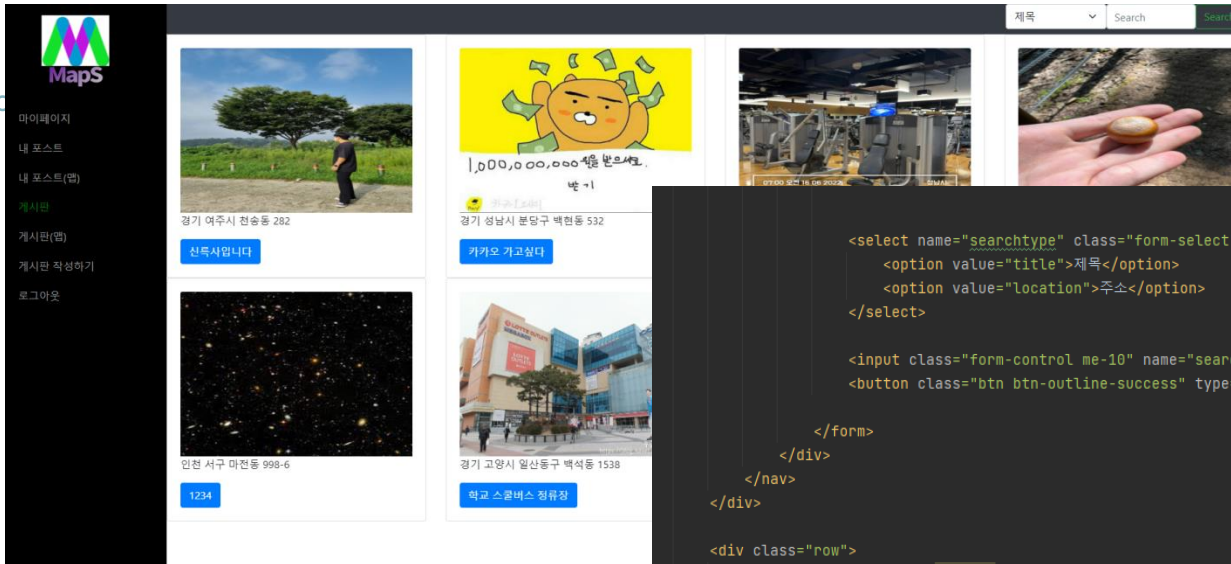
        Iterator<Post> postIt = post.iterator();
        while (postIt.hasNext()) {
            Post post1 = postIt.next();
            JsonObject object = new JsonObject();
            String category = post1.getCategory();
            String location_x = post1.getLocation_x();
            String location_y = 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);
            object.addProperty(property: "imgurl", imgurl);
        }
    }
}
```

```
<div class="left" style="...">
<div th:each="post:${post}">
    <div class="card" style="...">
        
        <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([[${post['location_x']}]], ...)">move
                <a th:href="@{/post/postview/{id}(id=${post.id})}" class="btn btn-primary">Read
            </div>
        </div>
    </div>
</div>
</div>

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

개발 내용(8/11)



```

<select name="searchtype" class="form-select" aria-label=".form-select-lg example"
  <option value="title">제목</option>
  <option value="location">주소</option>
</select>

<input class="form-control me-10" name="search" type="search" placeholder="Search"
<button class="btn btn-outline-success" type="submit">Search</button>

</form>
</div>
</nav>
</div>

<div class="row">
  <div class="col-sm-3" th:each="post:${post}">
    <div class="card">
      <div class="card-body">
        <img alt="" src="" class="card-img-top" data-bbox="515 568 615 615" />
        <p class="card-text" th:text="${post.location}">location</p>
        <p class="card-text" th:text="${post.title}" th:href="@{/post/postview/{id}(id=${post.id})}" class="
  
```

```

@GetMapping("post/list")
public String listPost(Model model, String search, String searchtype) {
    if (searchtype == null) {
        model.addAttribute("post", postService.findAllBysecret("nosecret"));
        return "post/list";
    } else if (searchtype.equals("title")) {
        List<Post> post = postService.post_title_search(search, secret: "nosecret");
        model.addAttribute("post", post);
        return "post/list";
    } else {
        List<Post> post = postService.post_location_search(search, secret: "nosecret");
        model.addAttribute("post", post);
        return "post/list";
    }
}
}

```

개발 내용(9/11)



- 마이페이지
- 내 포스트
- 내 포스트(맵)
- 게시판
- 게시판(맵)
- 게시판 작성하기
- 로그아웃

| | |
|-----|-------|
| 제목 | |
| 위치 | |
| 작성자 | test1 |
| 내용 | |



```
<header class="py-5">
<table class="table table-sm text-center" border="2" width="100%" th:if="${post.users.id}=${user_id}">
<tr>
<td colspan="1" class="table-primary">제목</td>
<td colspan="3" th:text="${post.title}">제목</td>
</tr>
<tr>
<td colspan="1" class="table-primary">위치</td>
<td colspan="3" th:text="${post.location}">위치</td>
</tr>
<tr>
<td class="table-primary">작성자</td>
<td colspan="3" th:href="@{/user/userpage/{postuser}(postuser=${postuser})}" th:text="${postuser}">작성자</td>
<td class="table-primary">카테고리</td>
<td th:text="${post.category}"></td>
</tr>
<tr>
<td class="table-primary" height="80px" colspan="1" style="vertical-align: top;">내용</td>
<td colspan="3" th:text="${post.text}" style="vertical-align: top;"></td>
</tr>
</table>
```

```
<table class="table table-sm text-center" border="2" width="100%" th:if="${post.users.id}!=${user_id}">
<tr>
<td colspan="1" class="table-primary">제목</td>
<td colspan="3" th:text="${post.title}">제목</td>
</tr>
<tr>
<td colspan="1" class="table-primary">위치</td>
<td colspan="3" th:text="${post.location}">위치</td>
</tr>
<tr>
<td class="table-primary">작성자</td>
<td colspan="3" th:href="@{/user/userpage/{postuser}(postuser=${postuser})}" th:text="${postuser}">작성자</td>
<td class="table-primary">카테고리</td>
<td th:text="${post.category}"></td>
</tr>
```

```
@GetMapping("/post/postview/{id}")
public String postview(Model model, @PathVariable("id") Long id, Authentication authentication, String searchtype, String search) {
    Post postDto = postService.findpost(id);
    Users users = userService.findByUser_Users(postDto.getUsers().getId());

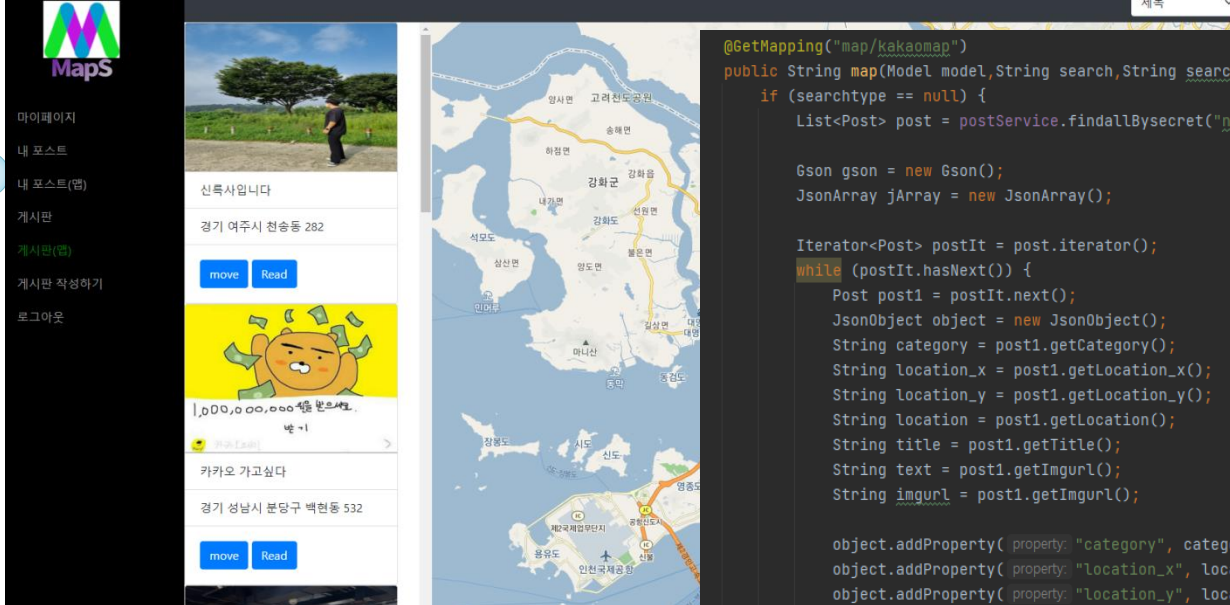
    List<Comments> comments = commentService.findbypostcomment(id);

    Long user_id = userService.findByUser_Long(authentication.getName());
    model.addAttribute("user_id", user_id);
    model.addAttribute("postuser", users.getEmail());
    if(likesService.likecheck(authentication.getName(), id)){
        Likes likes = likesService.findLike(authentication.getName(), id);
        model.addAttribute("like", likes.getId());
        model.addAttribute("likes", attributeValue: true);
    }
    else {
        model.addAttribute("like", attributeValue: 0);
        model.addAttribute("likes", attributeValue: false);
    }
    int count = likesService.count(id);
    model.addAttribute("count", count);

    if (comments != null && !comments.isEmpty()) {
        model.addAttribute("comments", comments);
    }
    model.addAttribute("id", id);
    model.addAttribute("post", postDto);
}
```



개발 내용(10/11)



```
@GetMapping("map/kakaoapi")
public String map(Model model, String search, String searchtype) {
    if (searchtype == null) {
        List<Post> post = postService.findAllBysecret("nosecret");

        Gson gson = new Gson();
        JSONArray jsonArray = 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.getText();
            String imgUrl = post1.getImgurl();

            object.addProperty("category", category);
            object.addProperty("location_x", location_x);
            object.addProperty("location_y", location_y);
            object.addProperty("location", location);
            object.addProperty("title", title);
            object.addProperty("text", text);
            object.addProperty("imgurl", imgUrl);
        }
    }
}
```

```
<div id="page-content-wrapper">
    <div>
        <nav class="navbar navbar-dark bg-dark">
            <div class="container-fluid">
                <a class="navbar-brand"></a>
                <form class="d-flex">
                    <input type="text" value="Search" />
                    <button type="submit" value="Search" />
                </form>
            </div>
        </nav>
        <div class="left">
            <div class="post">
                <div class="card">
                    <img alt="Post Image" />
                    <ul>
                        <li>An item</li>
                        <li>An item</li>
                    </ul>
                    <div class="card-body">
                        <button type="button" value="move" />
                        <a href="#" />
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
on(jsonArray);
);
@Name("json", json);
@Name("post", post);
```



개발 내용(11/11)

```
<div class="container-fluid">
  <table class="table table-striped" id="commentlist">
    <tr>
      <td>댓글</td>
      <td>작성지</td>
      <td>작성일</td>
    </tr>
    <tr th:each="comments:${comments}">
      <td th:text="${comments.comments}"></td>
      <td th:text="${comments.user?.email}"></td>
      <td th:text="${comments.createdDate}"></td>
    </tr>
  </table>
</div>
```

```
<!-- 댓글 입력-->
```

```
<div class="card">
```

```
  <div class="card-header bi bi-chat-right-dots"> Write a Comment</div>
```

```
  <form class="card-text" method="POST">
```

```
    <div class="card-body">
```

```
      <textarea id="comment" class="form-control" rows="4" placeholder="댓글을 입력하세요"></textarea>
```

```
    </div>
```

```
    <div class="card-footer">
```

```
      <a class="btn btn-primary btn-lg px-4 me-sm-3 text-center" href="#features" onclick="return dataSend()">submit</a>
```

```
    </div>
```

```
  </form>
```

```
</div>
```

```
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);
```

```
  });
```

```
}
```

```
  {
```

```
    strategy = GenerationType.SEQUENCE)
```

```
  = false)
```

```
  }comments;
```

```
    created_date")
```

```
    createdDate = LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy.MM.dd HH:mm"));
```

```
  }
```

```
    modified_date")
```

```
    modifiedDate = LocalDateTime.now().format(DateTimeFormatter.ofPattern("yyyy.MM.dd HH:mm"));
```

```
  = FetchType.LAZY)
```

```
  = "post_id")
```



결론 및 기대 효과

▪ 결론

1. 지도를 기반으로 한 SNS 구축완료
2. 게시물 작성 시 지도 활용의 다양성
3. 게시물 작성 시 사용자간의 빠른 위치 파악 가능

▪ 기대 효과

1. 지도를 기반으로 SNS를 활성화 하여 사용자간의 만남이 더욱 편해질 수 있음
2. 기존의 SNS 방식을 활용하면서 추가적인 기능을 통하여 사용자 유치가 가능할 것으로 보임



감사합니다