

# SNS PROJECT: OHALL

경북산업직업전문학교 빅데이터반

4조 | 박지호 | 오세룡 | 조영탁 | 이범식 | 김혜민

# WELCOME TO THE OHALL



Spring, MySQL Project

# PROJECT OUTLINE

## :프로젝트 목차

1. 프로젝트 개요
2. 주제선정과 요구사항
3. 팀 구성원 소개와 역할
4. 개발일정과 개발과정
5. 프로젝트 수행과정
6. 프로젝트 수행결과
7. 오류수정
8. 자체평가의견





# PROJECT OUTLINE

## :프로젝트 개요

프로젝트명

- OHALL (SNS)

프로젝트 개발자

- 박지호, 오세룡, 조영탁, 이범식, 김혜민

프로젝트 개발일자

- 2022.10.31 ~ 2022.11.10

프로젝트 개발환경

- Eclipse
- JAVA 8
- Spring 5.2.5
- MySQL
- Visual Studio Code
- Python



# PROJECT

---

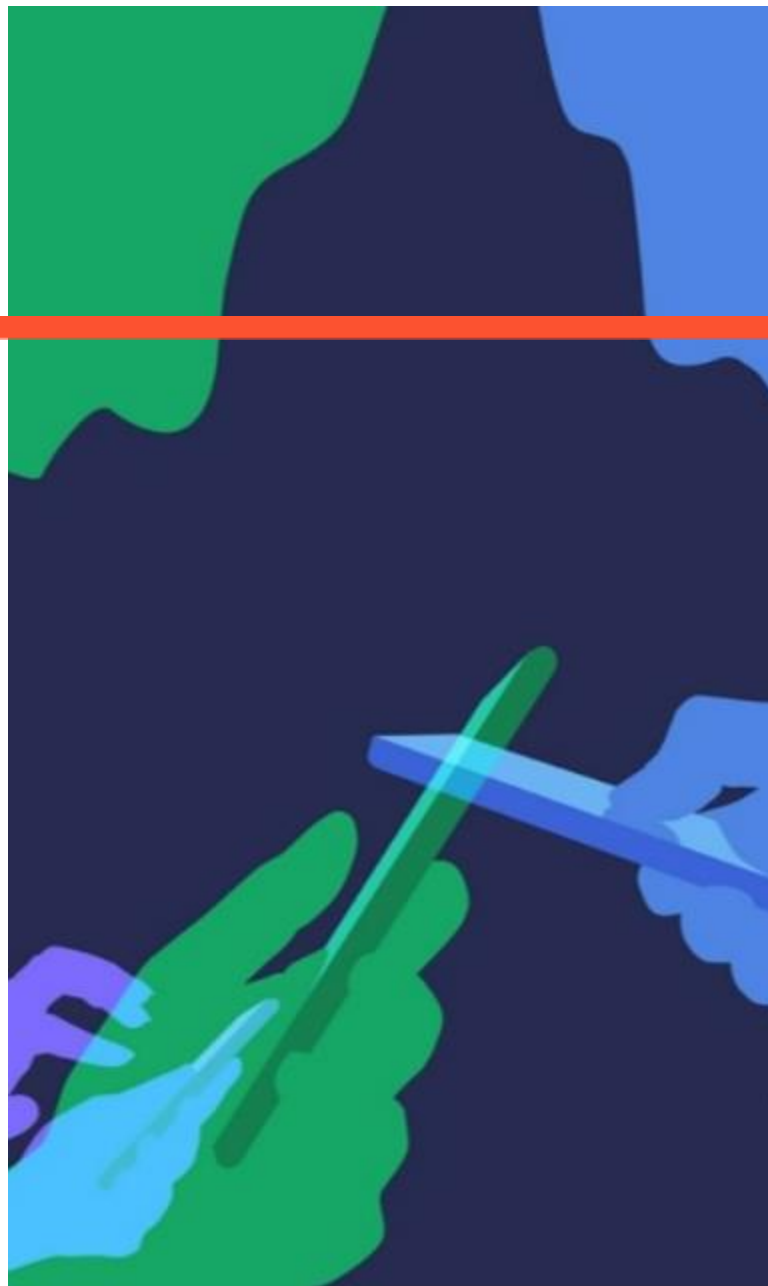
## VISION : 주제 선정

앞서 진행했던 DB프로젝트와 연계하여 사람들이 가장 많이 이용하는 애플리케이션인 SNS를 주제로 선정하였습니다.

그동안 실습한 내용을 바탕으로 SNS 페이지 및 게시판, 대시보드까지 구현한다면 저희에게 많은 공부가 될 것이라고 판단하였습니다.

또한 우리가 늘 사용하며 친근하게 접하는 SNS이기에 팀 프로젝트를 진행하고 공부하는 내내 팀원들의 흥미와 재미를 불러 일으킬 것이라고 예상하였습니다.

모두가 실생활에서 사용하기 때문에 장단점이나 개선점, 새로운 아이디어를 제시하는 데에도 부담이 적어, 앞으로 업데이트를 진행하기에 좋을 것이라고 생각하였습니다.





## 프로젝트 요구사항

데이터베이스 프로젝트 때 썼던 테이블을 활용하거나  
그에 상응하는 규모의 테이블을 활용할 것 ✓

공공 데이터를 파이썬을 활용하여 가져오고,  
데이터베이스에 추가해서 차트를 d3.js로 구현할 것 ✓

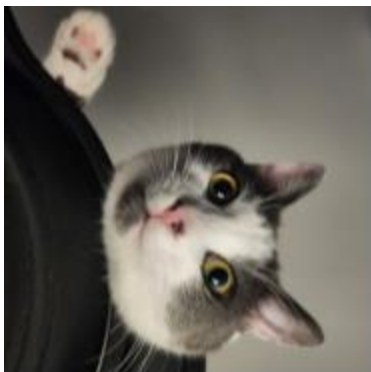
JSP 혹은 스프링 중 선택하여 프로젝트 진행할 것 ✓

모든 구성원들이 빠짐없이 기여할 수 있도록 진행할 것 ✓

이하 다음 상황들을 반드시 고려할 것

- 프로젝트 개요 ✓
- 팀 구성원 역할 소개 ✓
- 프로젝트 수행절차 ✓
- 프로젝트 수행결과 ✓
- 자체평가의견 ✓

# TEAM MEMBERS : 조원 소개



박지호

웹페이지 코드 작성  
DB 구축 및 연결  
공공데이터 활용 및 D3 제작  
PPT 작성  
발표



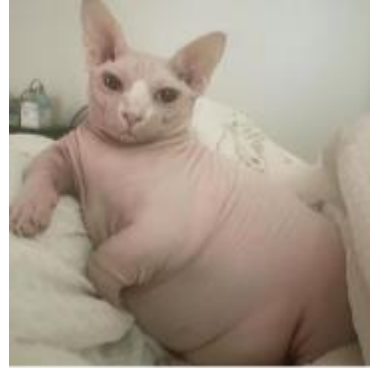
오세룡

Spring 로직 전반적인 작성  
MVC패턴 구축 및 연결  
DB 테이블 생성 및 연결



조영탁

웹페이지 코드 작성  
MVC패턴 구축 및 연결  
DB 테이블 생성 및 연결  
공공데이터 활용 및 D3 제작



이범식

공공데이터 수집  
DB 구축 및 연결  
SNS 로고 디자인



김혜민

웹페이지 코드 작성  
SNS 로고 디자인  
PPT 작성

# PROJECT

---

## HISTORY : 개발과정

### STEP 1

주제선정  
개발방향논의  
요구사항파악

### STEP 2

Spring 개발환경 구축  
JSP, CSS 코드작성

### STEP 3

My SQL DB 구축 및 연결

### STEP 4

공공데이터 수집  
D3 chart 구현

### STEP 5

PPT 작성



---

# PROJECT EXECUTION PROCESS

: 프로젝트 수행과정

# DEVELOPMENT ENVIRONMENT:폴더구조

- ▼ sns [OHALLproject main]
  - > Deployment Descriptor: sns
  - > Spring Elements
  - > JAX-WS Web Services
  - > Java Resources
  - > Deployed Resources
  - ▼ src
    - ▼ main
      - ▼ java
        - ▼ com
          - ▼ project
            - ▼ sns
              - > controller
              - > d3
              - > dao
              - > dto
              - > service

- ▼ resources
  - > sqlmap
  - log4j.xml
  - ▼ webapp
    - ▼ resources
      - > img
      - > js
      - > style
    - ▼ WEB-INF
      - ▼ spring
        - > appServlet
        - root-context.xml
      - ▼ views
        - > board
        - > component
        - > d3
        - > user
    - web.xml

# DEVELOPMENT ENVIRONMENT:ROOT-CONTEXT

```
<bean id="dataSource"
      class="org.apache.commons.dbcp2.BasicDataSource"
      destroy-method="close">
  <property name="driverClassName"
    value="com.mysql.cj.jdbc.Driver" />
  <property name="url"
    value="jdbc:mysql://localhost:3306/projectdb?serverTimezone=UTC&useSSL=false&
    useUnicode=true&characterEncoding=UTF-8" />
  <property name="username" value="root" />
  <property name="password" value="1234" />
</bean>
```

MySQL DB 연결



```
<bean id="sqlSessionFactory"
      class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="mapperLocations"
    value="classpath:/sqlmap/**/*.xml" />
</bean>
<bean id="sqlSessionTemplate"
      class="org.mybatis.spring.SqlSessionTemplate">
  <constructor-arg index="0" ref="sqlSessionFactory" />
</bean>
```

# MVC PATTERN: 모델, 뷰, 컨트롤러 패턴

## MODEL

- 📁 > dao
  - 📄 FeedDAO.java
  - 📄 UserDAO.java
- 📁 > dto
  - 📄 CommentDTO.java
  - 📄 FeedDTO.java
  - 📄 LoginDTO.java
  - 📄 ProfileDTO.java
  - 📄 UploadFileDTO.java
  - 📄 UserDTO.java
- 📁 > service
  - 📄 FeedService.java
  - 📄 FeedServiceImpl.java
  - 📄 UserService.java
  - 📄 UserServiceImpl.java

## VIEW

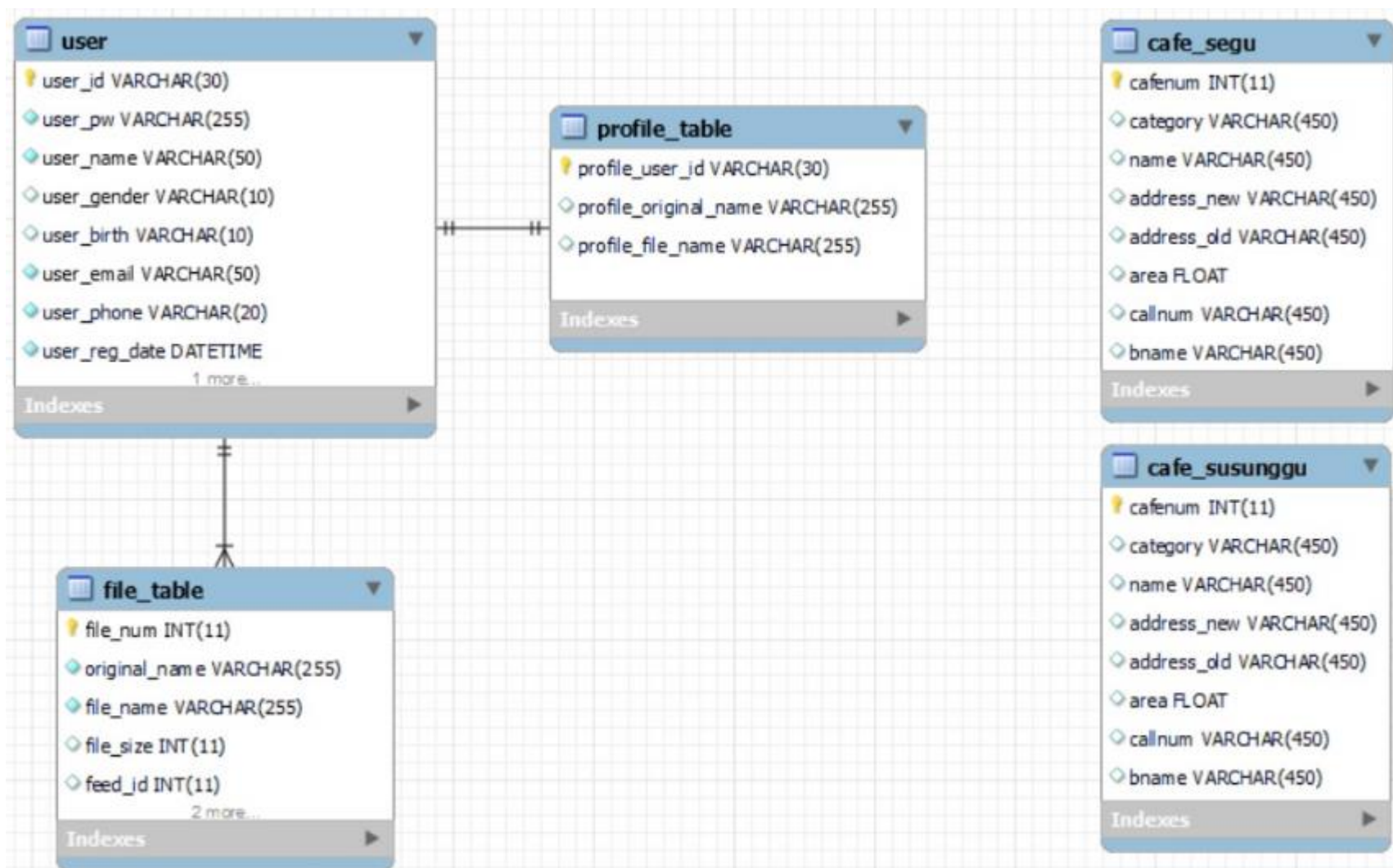
- 📁 > views
  - 📁 > board
    - 📄 main.jsp
    - 📄 modifyFeed.jsp
    - 📄 newFeed.jsp
  - 📁 > component
    - 📄 header.jsp
  - 📁 > d3
    - 📄 d3.jsp
  - 📁 > user
    - 📄 detailUser.jsp
    - 📄 join.jsp
    - 📄 login.jsp
    - 📄 modifyUser.jsp

## CONTROLLER

- 📁 > controller
  - 📄 FeedController.java
  - 📄 UserController.java



# DB DESIGN:DB설계



# PROCEDURE:수행절차

SPRING



DATABASE



VISUALIZATION

- JSP 작성
- HTML 작성
- JS 작성
- CSS 작성

- My SQL 연결
- 공공데이터 연결

- 웹페이지 구현
- D3 Chart 구현

# REGISTRATION:회원가입



OHALL



## 회원가입

### Personal Details

아이디

ohall

비밀번호

....

이름

관리자

아이디 중복 확인

아이디 중복 확인

비밀번호 확인 비밀번호가 일치하지 않음.

...

성별

남성 ☐ 여성 ☐ 비공개 ☒

생일

2022-11-11



Email

ohall\_admin@gmail.com

핸드폰 번호

010-1234-5678

회원가입 ▶

## REGISTRATION:회원가입

```
@Controller
```

```
public class UserController {
```

```
    @Autowired
```

```
    UserService userService;
```

```
    @GetMapping(value="/join")
```

```
    public String createUser() {
```

```
        return "/user/join";
```

```
    }
```

### Controller

주소 창에 주소를 입력하면  
알맞은 주소로 보냅니다.



# REGISTRATION:회원가입

```
<c:import url="../component/header.jsp" />
<div class="mycontainer">
  <header>회원가입</header>
  <form action="join" method="POST">
    <div class="form first">
      <div class="details personal">
        <span class="title">Personal Details</span>
        <div class="fields">
          <div class="input-field">
            <label>아이디</label>
            <input type="text" id="user_id" name="user_id" placeholder="아이디" required>
          </div>
          <div class="input-field">
            <label>비밀번호</label>
            <input id="user_pw" type="password" name="user_pw" placeholder="비밀번호" autocomplete="on" required>
          </div>
          <div class="input-field">
            <label>이름</label>
            <input type="text" name="user_name" placeholder="이름" required>
          </div>
          <div class="input-field">
            <label>아이디 중복 확인</label>
            <input type="button" id="idChk" style="border: 0; outline: none; background-color: #4070f4; color: #fff;"
          </div>

          <div class="input-field">
            <label>비밀번호 확인<span id="checkNotice"></span></label>
            <input type="password" id="pw_check" placeholder="비밀번호 확인" autocomplete="on" required />
          </div>
        </div>
      </div>
    </div>
  </form>
</div>
```

## JSP

데이터를 입력하여  
컨트롤러로 보냅니다.

# REGISTRATION:회원가입



## Controller

JSP에서 데이터를 넘겨받아  
알맞은 서비스를 호출합니다.

```
@Transactional
@PostMapping(value="/join")
public String createUserPost(UserDTO dto) {
    this.userService.createUser(dto);
    this.userService.initProfile(dto);
    return "redirect:login";
}
```

# REGISTRATION:회원가입

## Service

DTO의 모델을 활용하여  
필요한 코드를 실행합니다.

@Service

```
public class UserServiceImpl implements UserService {  
    @Autowired  
    UserDao userDao;  
  
    @Override  
    public String createUser(UserDTO dto) {  
        PasswordEncoder p = new BCryptPasswordEncoder();  
        dto.setUser_pw(p.encode(dto.getUser_pw()));  
  
        int affectRowCount = this.userDao.userInsert(dto);  
        if(affectRowCount == 1) {  
            return dto.getUser_id();  
        }  
        return null;  
    }  
}
```

## REGISTRATION:회원가입

```
public class UserDTO {  
    private String user_id;  
    private String user_pw;  
    private String user_name;  
    private String user_gender;  
    private String user_birth;  
    private String user_email;  
    private String user_phone;  
    private String user_reg_date;  
}
```

### DTO

데이터를 저장해 두는  
DTO 모델입니다.



# REGISTRATION:회원가입

```
@Repository
public class UserDAO {
    @Autowired
    SqlSessionTemplate sqlSessionTemplate;

    public int userInsert(UserDTO dto) {
        return this.sqlSessionTemplate.insert("user.user_insert", dto);
    }
}
```

## DAO

원하는 SQL 쿼리문을  
실행합니다.

```
<mapper namespace="user">

<insert id="user_insert" parameterType="com.project.sns.dto.UserDTO" useGeneratedKeys="true" keyProperty="user_num">
    <![CDATA[
        insert into user(user_id, user_pw, user_name, user_gender, user_birth, user_email, user_phone)
        values
        (#{user_id}, #{user_pw}, #{user_name}, #{user_gender}, #{user_birth}, #{user_email}, #{user_phone});
    ]]>
</insert>
```

# LOGIN:로그인



OHALL



아이디

비밀번호

로그인

[회원가입](#) [비밀번호 잊음](#)

# LOGIN:로그인

## 반응형 UI 구현

웹페이지 창 사이즈 변경 시  
UI의 배열과 사이즈가 자동으로  
조절되도록 하였습니다.

### Registration

#### Personal Details

ID

Enter ID

PASSWORD

Enter Password

이름

Enter your Name

성별

남성 ☐ 여성 ☐ 비공개 ☐

생일

연도-월-일

Email

### Registration

#### Personal Details

ID

Enter ID

PASSWORD

Enter Password

이름

Enter your Name

성별

남성 ☐ 여성 ☐ 비공개 ☐

생일

연도-월-일

Email

Enter your Email

핸드폰 번호

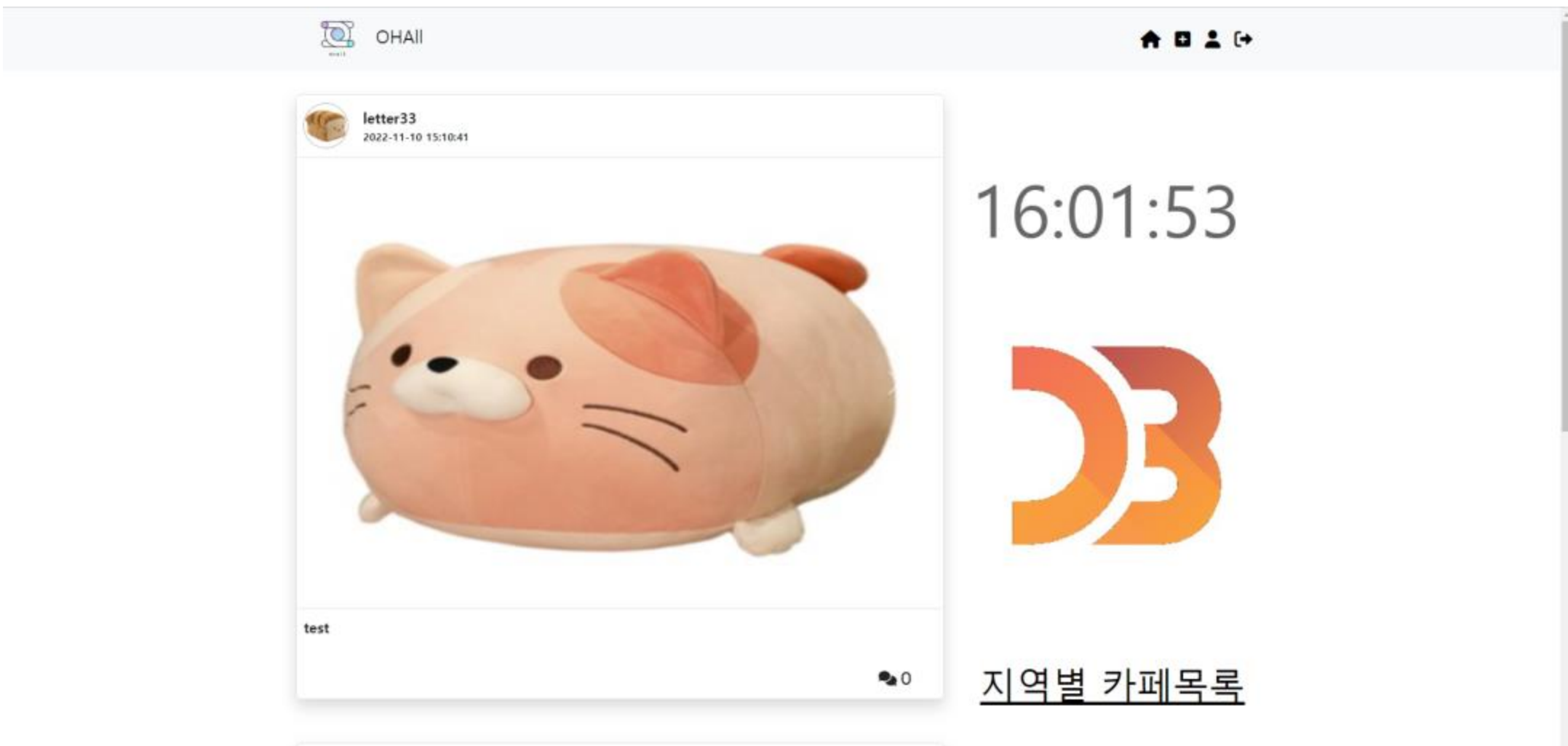
Enter your Mobile number

선호분야

Enter your preference

Submit >

# MAIN:메인화면





# MAIN:메인화면

```
<%-- 게시물 목록 출력--%>
<c:forEach var="feed" items="${feedList}" varStatus="status">
    <div id="carouselExampleIndicators${status.count}"
        class="carousel slide border shadow rounded mt-5"
        data-bs-ride="true">
        <div class="rounded-top border-bottom profile-header">

            <%-- 프로필 사진 출력 --%>
            <c:forEach var="profile" items="${profileList}">
            <c:choose>
            <%--프로필 이름에 맞는 프로필 사진 출력--%>
            <c:when test="${feed.feed_user_id eq profile.profile_user_id}">
                <c:choose>
```

# MAIN:메인화면

```
<!-- 프로필 사진이 없는 경우 기본 이미지 출력 --%>
  <c:when test="${profile.profile_file_name eq 'default-image.png' }">
    
  </c:when>
  <!-- 프로필 사진이 있는 경우 사진 출력 --%>
  <c:otherwise>
    
  </c:otherwise>
</c:choose>
</c:when>
</c:choose>
```

# MAIN:메인화면

```
<%-- 게시물 내용 중 아이디랑 게시물 작성 시간 출력 --%>
<div class="d-flex flex-column ms-3" style="font-weight: 600;">
  <div>${feed.feed_user_id }</div>
  <div class="time">${feed.feed_created_at }</div>
</div>
<%-- 로그인한 아이디가 게시물 작성자 아이디와 같으면 수정 삭제를 보여줌 --%>
<c:if test="${user_id eq feed.feed_user_id }">
  <div class="ms-auto me-3 d-flex">
    <form action="modify_feed">
      <input type="hidden" name="feed_id" value="${feed.feed_id }" />
      <button type="submit" class="fa-solid fa-pen fa-sm"></button>
    </form>
    <form action="delete_feed" method="POST">
      <input type="hidden" name="feed_id" value="${feed.feed_id }" />
      <button type="submit" class="fa-solid fa-eraser fa-sm"></button>
    </form>
  </div>
</c:if>
</div>
```

# MAIN:메인화면

```
<%-- 게시물 사진 출력 --%>
<c:forEach var="uploadFile" items="${uploadFileList }">
    <c:if test="${feed.feed_id eq uploadFile.feed_id }">
        <div class="carousel-item">
            
        </div>
    </c:if>
</c:forEach>
</div>
<%-- 게시물 넘기기 할때 해당 게시물 사진만 넘어가게 해주는 것. --%>
<button class="carousel-control-prev carousel-btn" type="button"
    data-bs-target="#carouselExampleIndicators${status.count }"
    data-bs-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Previous</span>
</button>
<button class="carousel-control-next carousel-btn" type="button"
    data-bs-target="#carouselExampleIndicators${status.count }"
    data-bs-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="visually-hidden">Next</span>
</button>
```



# MAIN:메인화면

```
<%-- 게시물 내용 출력 --%>
<div class="border-top rounded-bottom content position-relative">
  <p class="m-2 d-inline-block" style="font-weight: 600;">${feed.feed_content }</p>
  <div class="position-absolute comment-icons">

    <%-- 댓글 수 출력 --%>
    <i class="fa-solid fa-comments" id="toggle"
      onclick='$("#comments${status.count}").toggle
      (500)'>&nbsp;${feed.feed_comment_count }</i>
    <!-- <i class="fa-regular fa-heart">&nbsp;0</i> -->
  </div>
</div>
```



# MAIN:메인화면

```
<%-- 댓글 리스트 출력 --%>
<div class="border-top" id="comments${status.count}" style="display: none;">
  <c:forEach var="comment" items="${commentList}">
    <c:if test="${feed.feed_id eq comment.comment_feed_id}">
      <div class="row">
        <p class="m-2 col-10">${comment.comment_user_id} : ${comment.comment_content}
        <span class="time d-flex justify-content-end">${comment.comment_created_at}
        </span></p>
        <%-- 댓글 작성자 본인만 삭제 가능 --%>
        <c:if test="${user_id eq comment.comment_user_id}">
          <form action="/delete_comment" method="post" id="comment_form"
            class="col-1 row text-center">
            <input type="hidden" name="comment_feed_id" value="${feed.feed_id}" />
            <input type="hidden" name="comment_id" value="${comment.comment_id}" />
            <button class="delete-btn fa-solid fa-eraser fa-sm"
              type="submit"></button>
          </form>
        </c:if>
      </div>
    </c:if>
  </c:forEach>
</c:choose>
```

## MAIN:메인화면

```
<%-- 로그인 해야 댓글 입력 가능하게 해주는 기능. --%>
<c:when test="${empty user_id}">
  <div class="comment-wrapper">
    <input class="form-control comment-input d-inline" type="text"
      placeholder="로그인 후 입력가능합니다." name="comment_content" readonly>
    <button type="submit"
      class="d-inline fa-solid fa-paper-plane fa-lg send-icon"
      disabled></button>
  </div>
</c:when>
<c:otherwise>
<c:choose>
```

## MAIN:메인화면

```
<%-- 게시물 작성자가 탈퇴한 경우 댓글들 작성 불가능.--%>
<c:when test="${feed.feed_user_id eq '탈퇴한 회원' }">
    <div class="comment-wrapper">
        <input class="form-control comment-input d-inline" type="text"
            placeholder="탈퇴한 회원의 게시물입니다." name="comment_content" readonly>
        <button type="submit"
            class="d-inline fa-solid fa-paper-plane fa-lg send-icon"
            disabled></button>
    </div>
</c:when>
```

## MAIN:메인화면

```
<%-- 댓글 작성 가능 --%>
<c:otherwise>
  <form action="/new_comment" method="post">
    <input type="hidden" name="comment_feed_id"
      value="${feed.feed_id}" /> <input type="hidden"
      name="comment_user_id" value="${user_id}" />
    <div class="comment-wrapper">
      <input class="form-control comment-input d-inline" type="text"
        placeholder="댓글 입력하기" name="comment_content">
      <button type="submit"
        class="d-inline fa-solid fa-paper-plane fa-lg send-icon"></button>
    </div>
  </form>
```

# WRITING:글 작성



OHALL



 이미지 선택하기



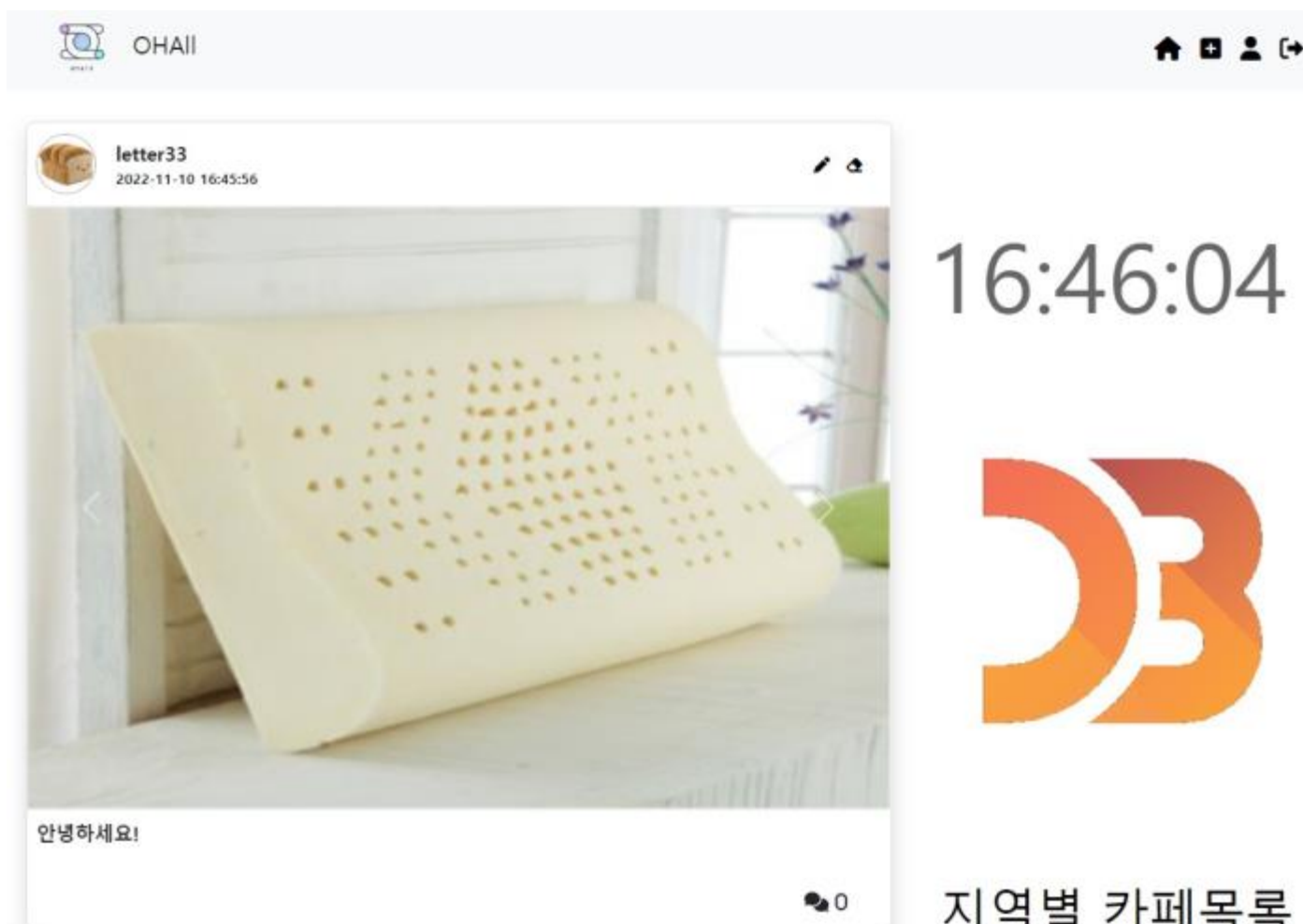
안녕하세요!

글 작성



# EDIT POST:게시글 수정

수정 전 게시글



16:46:04



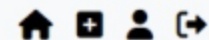
지역별 카페 목록

# EDIT POST:게시글 수정

게시글 수정 페이지



OHALL

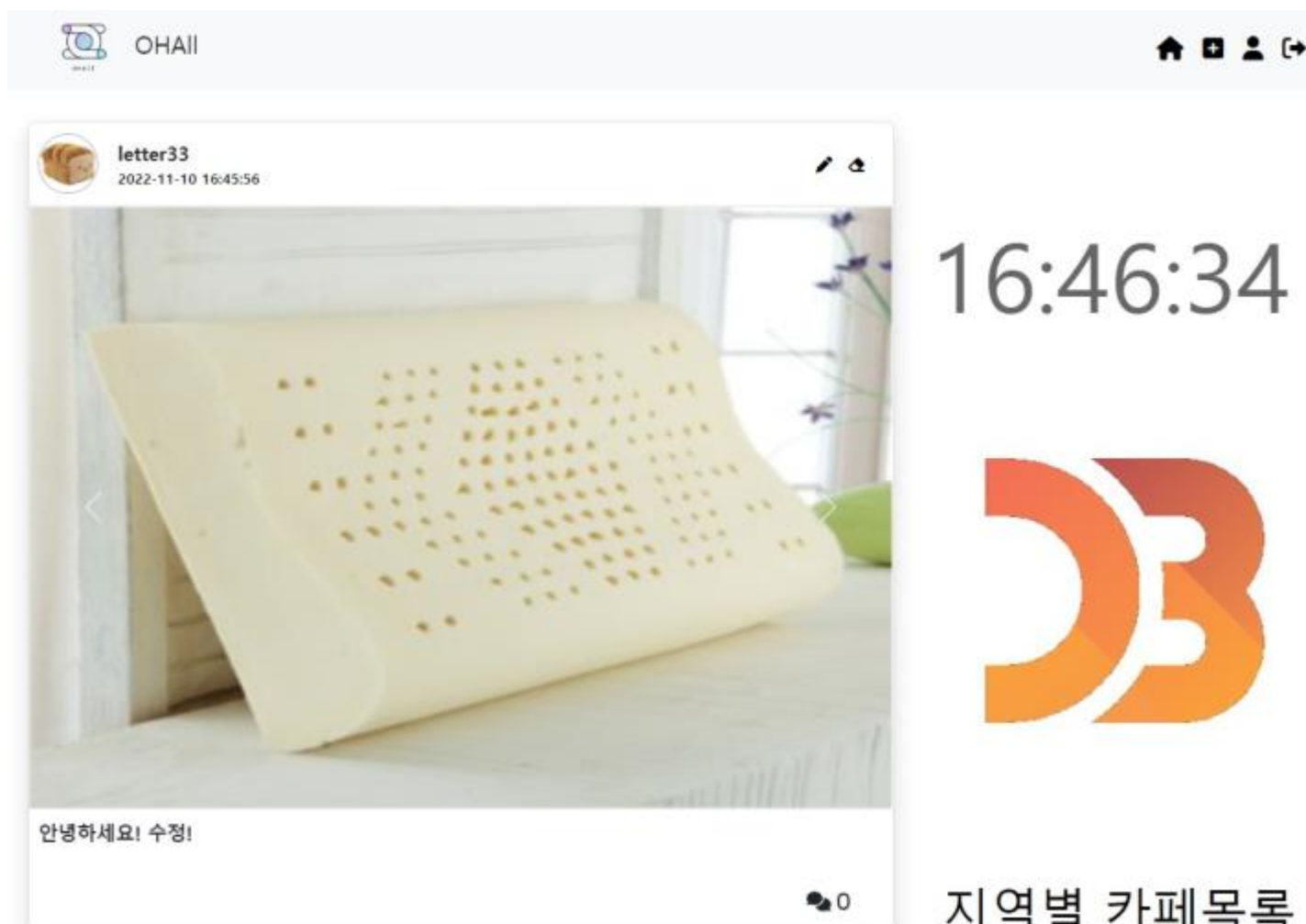


안녕하세요! 수정!

글 수정

# EDIT POST:게시글 수정

수정 후 게시글

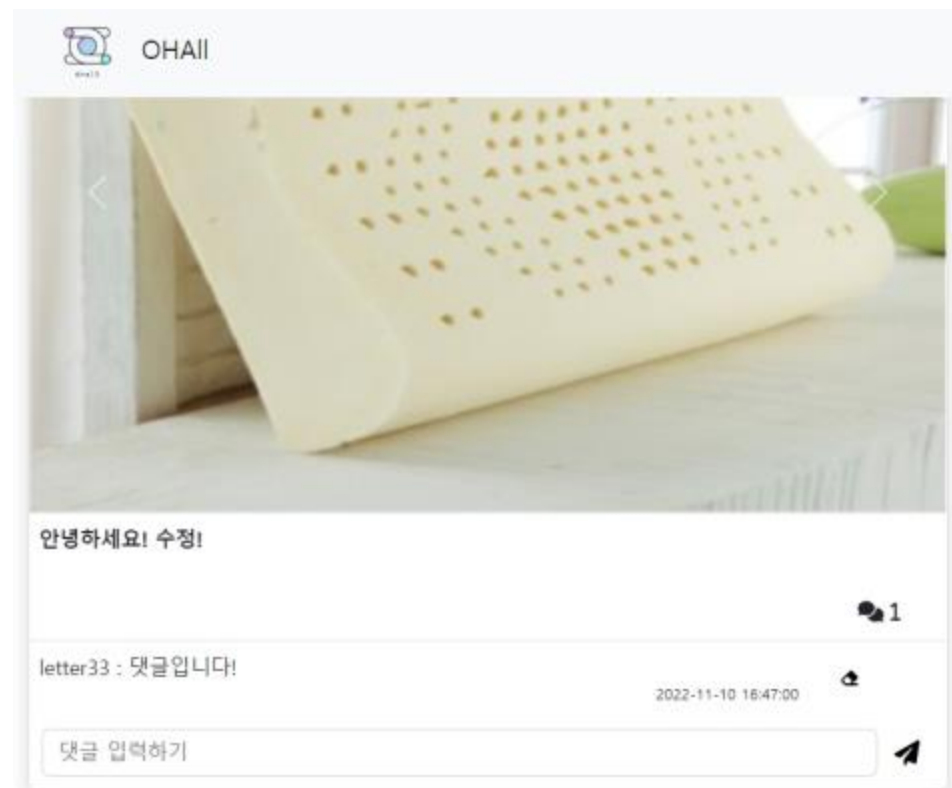


16:46:34








지역별 카페목록

# COMMENT:댓글 작성




# USER INFORMATION:내 정보조회 및 회원탈퇴

 OHALL



프로필 사진



Ohall

### 내 정보

Personal Details


아이디	이름	성별
<input type="text" value="ohall"/>	<input type="text" value="관리자"/>	남성 <input type="checkbox"/> 여성 <input type="checkbox"/> 비공개 <input checked="" type="checkbox"/>
생일	Email	핸드폰 번호
<input type="text" value="2022-11-11"/>	<input type="text" value="ohall_admin@gmail.com"/>	<input type="text" value="01012345678"/>
가입일		
<input type="text" value="2022-11-10 16:00:02"/>		





[정보 수정 >](#)

[회원탈퇴](#)




# CHANGING INFORMATION:내 정보수정

 OHALL



프로필 사진



## 내 정보 수정

### Personal Details

아이디	비밀번호	이름
<input type="text" value="ohall"/>	<input type="password" value="비밀번호"/>	<input type="text" value="관리자"/>
성별	생일	Email
남성 <input type="checkbox"/> 여성 <input type="checkbox"/> 비공개 <input checked="" type="checkbox"/>	<input type="text" value="2022-11-11"/>	<input type="text" value="ohall_admin@gmail.com"/>
핸드폰 번호		
<input type="text" value="01012345678"/>		

내 정보 수정 완료 ▶

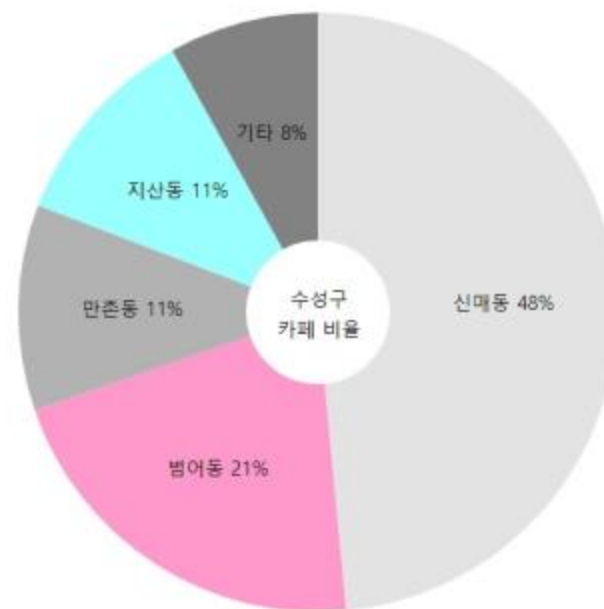
# DATABASE: 공공데이터DB

cafenum	category	name	address_new	address_old	area	callnum	bname
884	휴게음식점	벨케익(Bell cake)	대구광역시 수성구 범어로20길 68, 1층 (범어동)	대구광역시 수성구 범어동 213-6 1층	0	0507-1393-4...	커피숍
885	휴게음식점	제이샌드(J.SAND)	대구광역시 수성구 수성로 393, 상가동 1층 107...	대구광역시 수성구 수성동4가 1090-8 수성하이...	0	053-744-7613	커피숍
886	휴게음식점	험블앤멤블(humb...	대구광역시 수성구 수성로14길 34, 1층 (상동)	대구광역시 수성구 상동 384-6 1층	0	0507-1361-1...	커피숍
887	휴게음식점	수북하이수성점	대구광역시 수성구 돌안로71길 8, 1층 (수성동4...	대구광역시 수성구 수성동4가 1024-43 1층	0	053-759-5554	커피숍
888	휴게음식점	로지트(Rogit)황...	대구광역시 수성구 회망로36길 130-32, 제1동 ...	대구광역시 수성구 황금동 795-7 1층 제1동	0	데이터 미집계	커피숍
889	휴게음식점	그림이글에게	대구광역시 수성구 돌안로81길 27, 1층 (수성동...	대구광역시 수성구 수성동4가 1133-32 1층	0	0507-1319-1...	커피숍

# DATABASE:D3차트



OHALL



## 지역별 카페 목록

서구

연번 업종명 업소명

- 1 휴게음식점 동원다방
- 2 휴게음식점 이디야커피대구광장코아점

도로명 주소

- 대구광역시 서구 문화로 273-1, 지하1층 (평리동)  
대구광역시 서구 달구벌대로 1703 (내당동)

지번 주소

- 대구광역시 서구 평리동 870-4 지하1  
대구광역시 서구 내당동 463-34

면적

- 67.66  
46.0

전화번호

- 053-527-6063  
053-525-5577

업태명

- 커피숍  
커피숍

---

# PROJECT PERFORMANCE RESULTS

: 프로젝트 수행결과

# PROJECT DEMONSTRATION

프로그램 시연 영상을 재생하겠습니다.



# ERROR CORRECTION

: 오류수정



- CSS 코드가 제대로 적용이 안 되는 오류.
  - : 클래스 이름 앞에 .을 빠뜨림. 다시 수정하여 구동하니 오류없이 잘 됨.
- 서버 가동이 제대로 안 되는 오류.
  - : 루트 커넥트에서 DB를 잘 연결하거나 언어를 코드에 맞게 잘 인코딩하여 해결함.

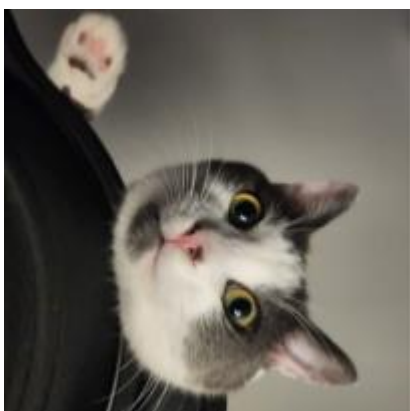


- 최신 게시글 삭제 후, 작성 시 업로드 했던 이미지가 제대로 나오지 않는 오류.
  - : 게시글 생성 시 첨부 된 이미지가 해당 테이블에 있는 가장 큰 아이디 값을 참조하는 방식에서 가장 최근 게시글 아이디를 저장하는 테이블을 따로 만들어 그 값을 참조하도록 변경함.



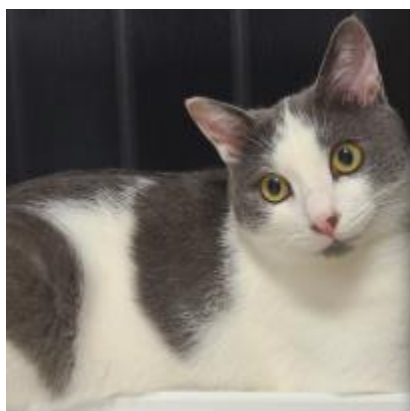
- 언어 인코딩이 달라서 파이썬에서 DB로 넘길 때 호환 오류가 발생.
  - csv 파일을 메모장에서 UTF-8언어로 변환하여 입력하는 것으로 해결함.

# SELF EVALUATION : 자체평가



박지호

Spring 개발환경 세팅과 웹 구현을 위해 다양한 코드페이지들을 공부하고 이해하는 것이 힘들었습니다. 좋은 팀원들의 많은 도움을 받아 완성했습니다. 모두에게 감사합니다!



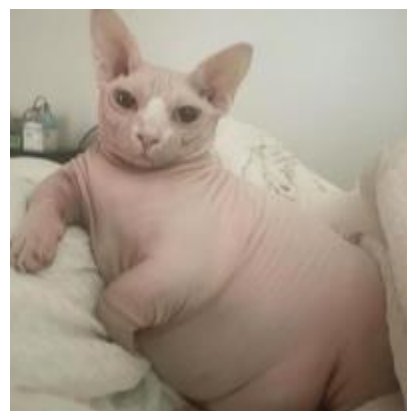
오세룡

만들면서 잘 안되는 부분은 검색하면서 공부하였다. 힘들긴 했지만 만들고 돌아가는 걸 보니 뿌듯했고 나름 재미있었다.



조영탁

주제선정부터 PPT 제작까지 모두가 열정적으로 참여하여 프로젝트가 즐겁고 보람찼습니다. MVC패턴 및 D3에 대해서 더 공부하고 코드를 구현하는데 성공해서 뿌듯했습니다.



이범식

묵묵하게 작업을 커버해 준 세룡씨, 예쁘게 포장해 준 헤민씨, 필요한 자료를 찾아서 힘을 주는 지호씨, 해야 될 일을 차근차근 해결한 영탁씨, 팀원들에게 당근만 준 채찍없는 간수인 저. 다들 고생 많았습니다.



김헤민

전에 진행했던 DB 프로젝트 보다 난이도가 많이 높아졌다. 개인적으로 부족한 부분이 많아 기여가 적어서 아쉽다. 기본기의 중요성을 다시 한 번 느낄 수 있는 프로젝트였다.

**THANK YOU!**

A black and white photograph of a night landscape. In the foreground, there's a calm body of water reflecting the sky. To the right, a rocky shoreline is visible. In the background, there are jagged mountains. The sky is dark and filled with stars, with a bright, glowing aurora borealis (Northern Lights) visible in the center. Overlaid on the left side of the image is the text 'THANK YOU!' in a bold, white, sans-serif font. A thick red horizontal line extends from the end of the text across the middle of the image.