



6조 세미프로젝트

PETmu





CONTENTS.

01
팀 소개

05
테이블 정의

09
Q&A

02
프로젝트 소개

06
모델링

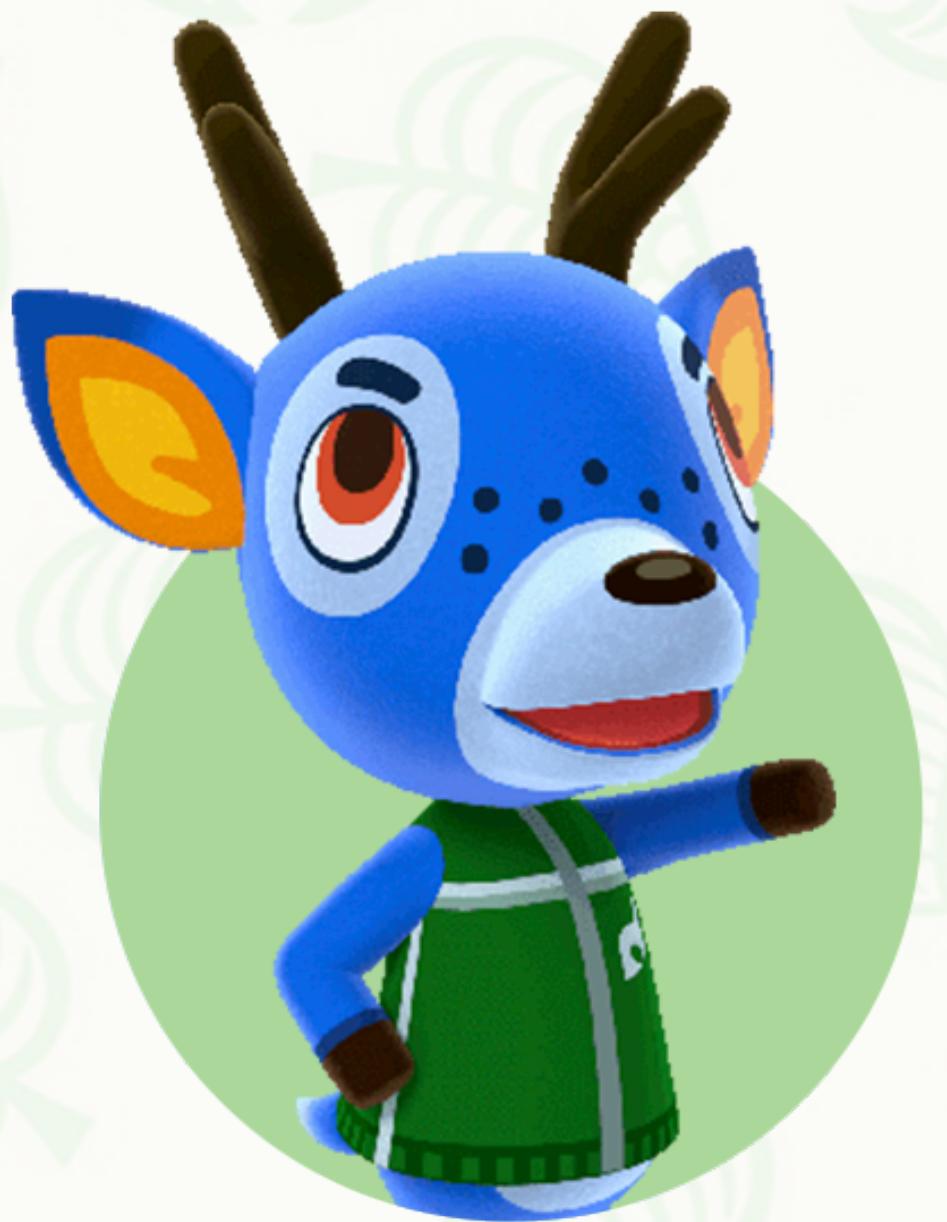
03
레퍼런스 분석

07
구현 가능 및 시연

04
개발환경 및 일정

08
개선사항

01 팀소개.



김용주

후기정보게시판



백지은

사진게시판
자유게시판



임기창

헤더/풋터
메인인덱스
API(주소, 지도)

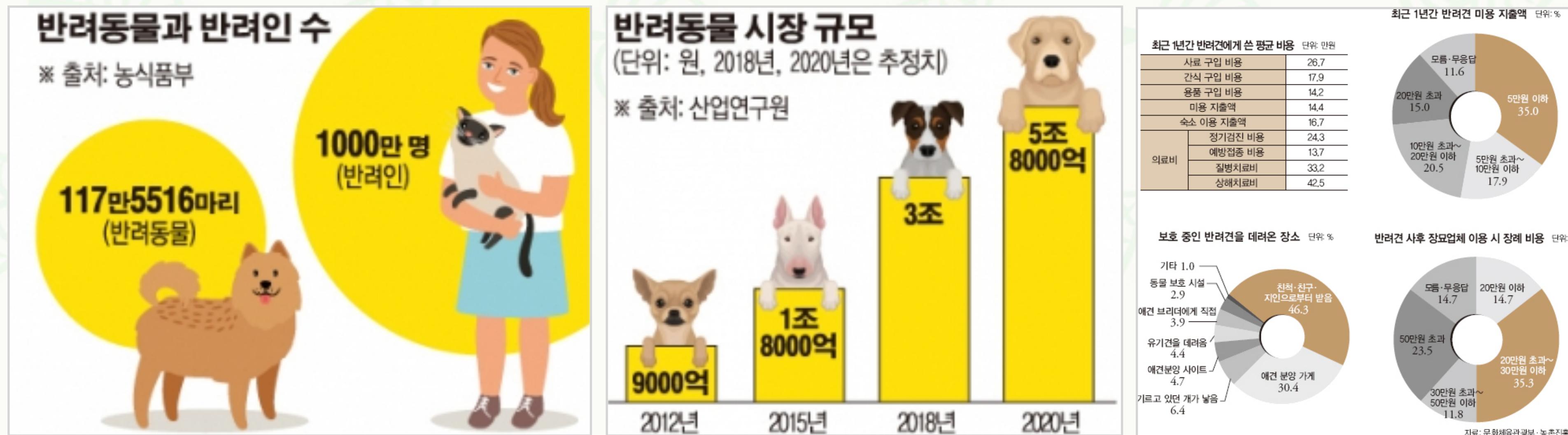


장민수

회원관리
회원탈퇴

02 프로젝트 소개.

트렌드 서치

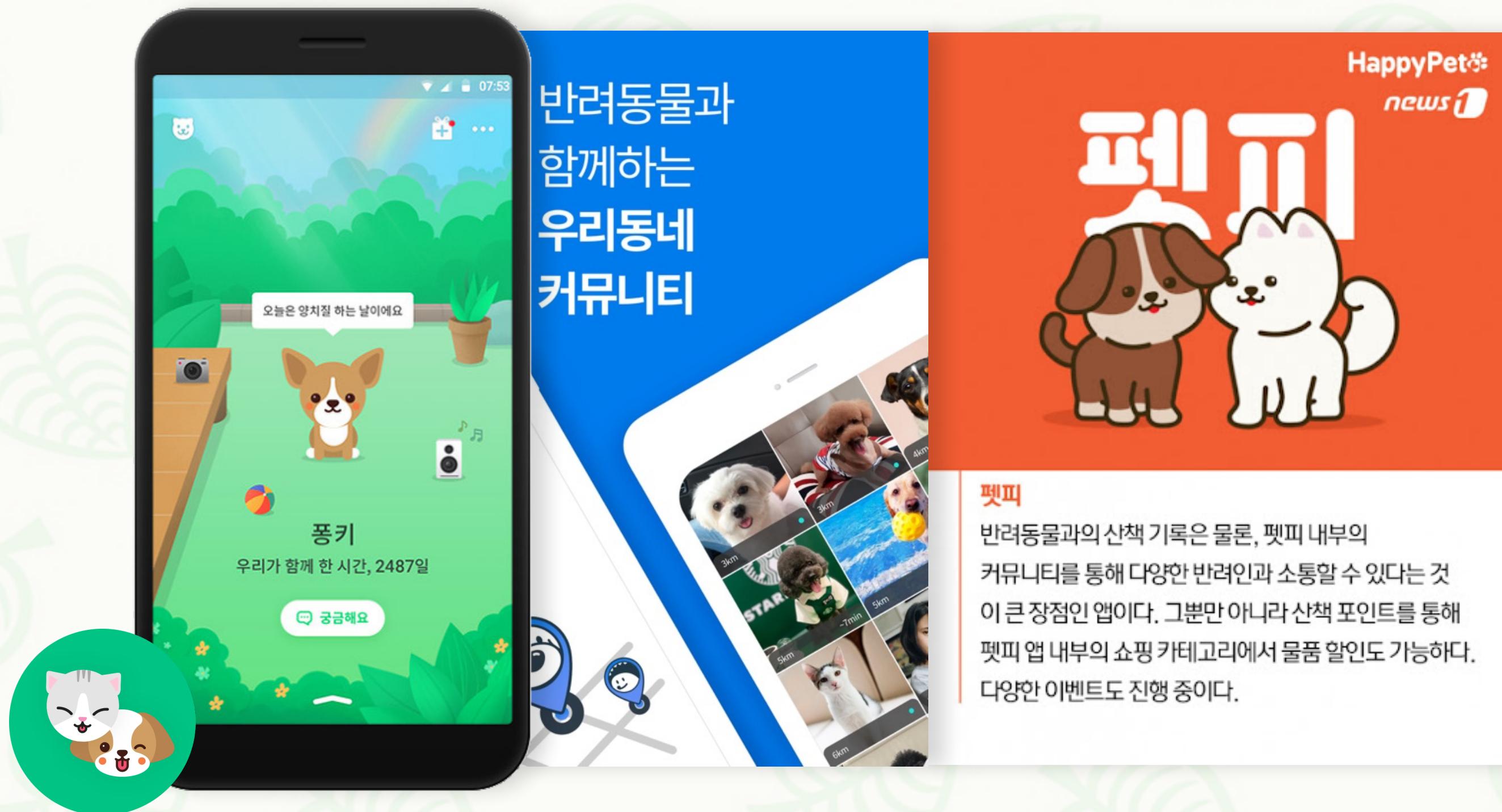


우리나라에서 반려동물을 양육하는 인구는 1000만 명, 반려동물은 약 118만 마리로 추정되며, 이에 따라 펫코노미(pet+economy) 시장 규모는 급격히 불어나고 있다.

그 중 반려인들의 최대 관심사는 반려동물의 건강이다. 비싸더라도 반려동물에게 질 좋은 사료나 간식을 구매하고, 펫보험과 애견전문 장례업체들도 생겨나는 추세를 보이고 있다.

02 프로젝트 소개.

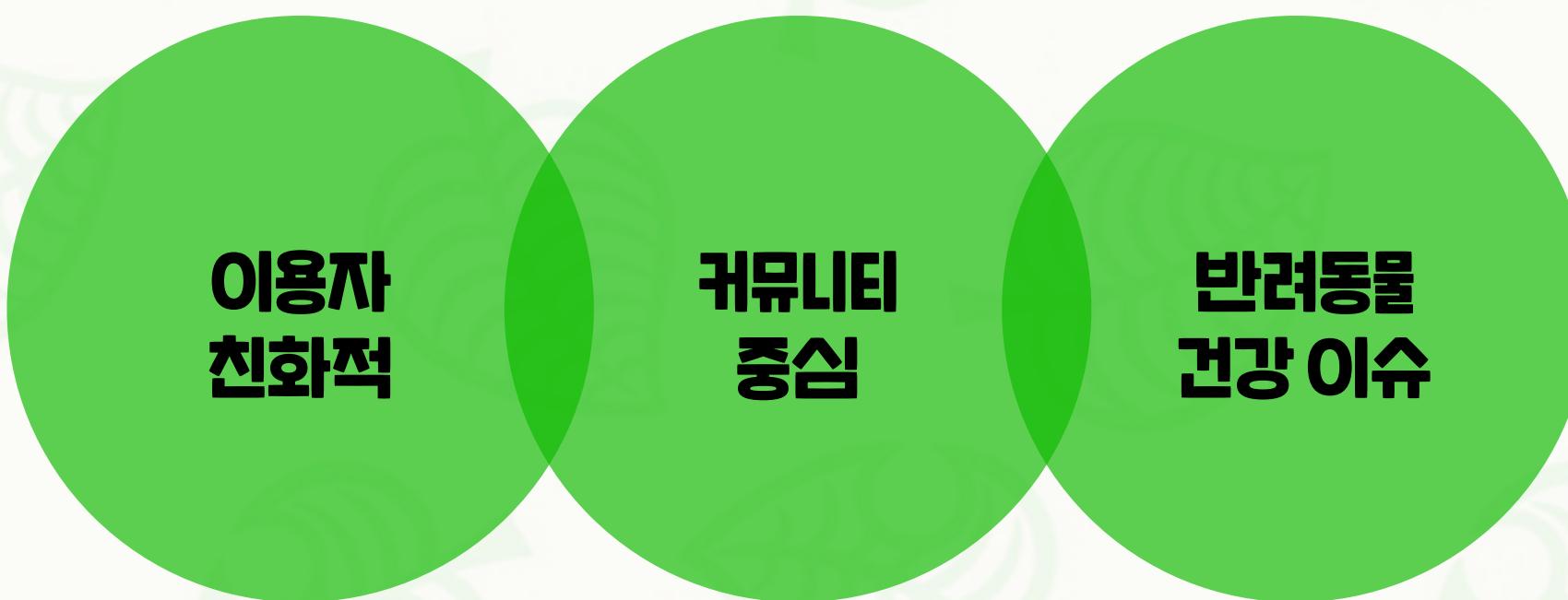
트렌드 서치



디지털세대인 반려인들은 정보를 얻기위해 인터넷에 의존하며, 펫 커뮤니티가 발빠르게 생겨나고 있다.

02 프로젝트 소개.

기획의도



이러한 상황을 바탕으로 이번 프로젝트 방향을 반려인들의 최대 관심사인 반려동물 건강 이슈를 중심으로 반려동물 보호자가 필요한 정보들을 얻을 수 있고, 공유할 수 있는 이용자 친화적 커뮤니티 사이트인 ‘펫뮤’를 제작하려 한다.

03 레퍼런스 분석.



마이펫



- 지역별, 검진 종류에 따른 동물병원 찾기
- 사이트와 연계된 병원으로 비용할인 존재
- 반려동물 관련 장소 정보 획득



- 사이트와 연계된 장소만 찾을 수 있음
- 동물병원과 장소 정보만 존재



위트



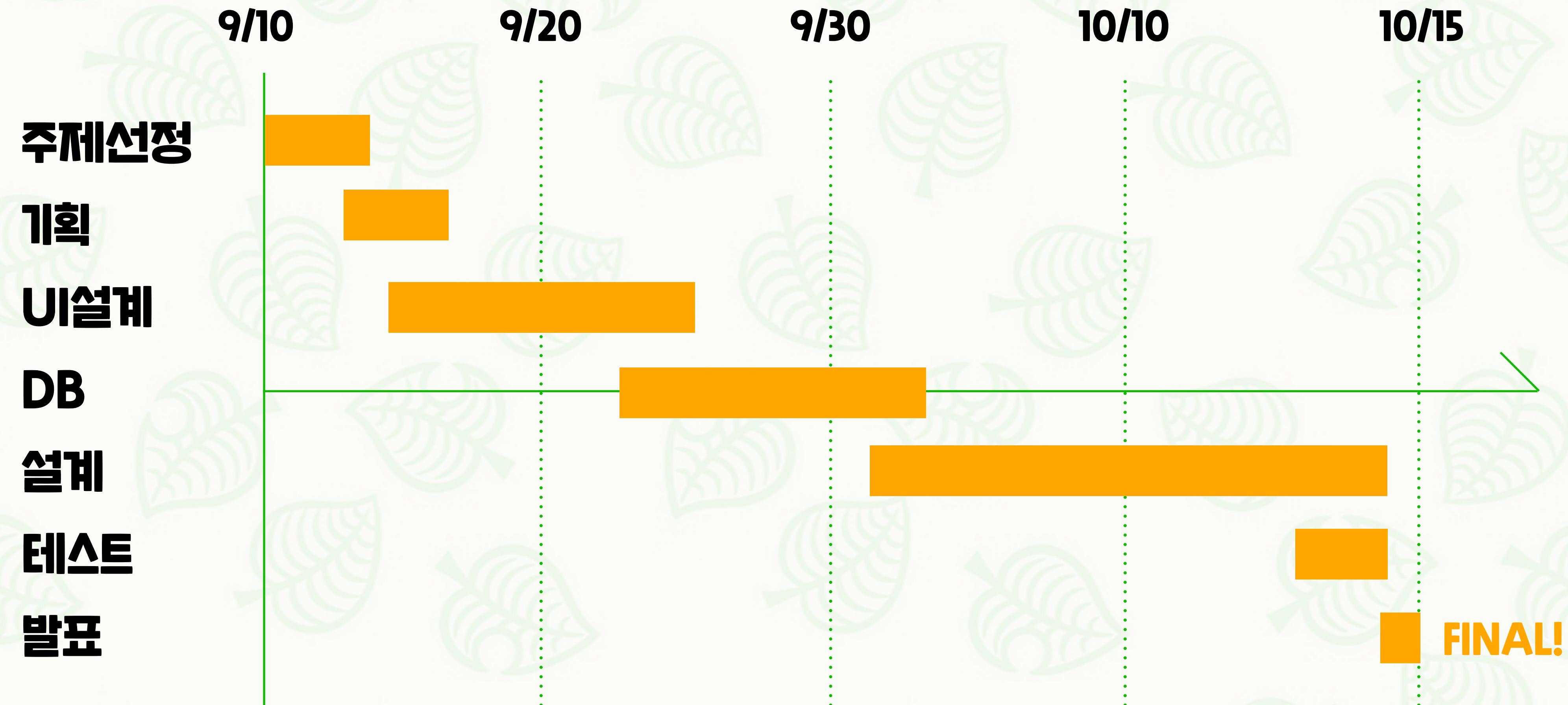
- 커뮤니티 중심 스토어 형성
- 스토어를 통해 쉽게 용품 구입 가능
- 다양한 이벤트 진행



- 반려동물 관련 정보를 수집하기 어려움

04 개발환경 및 일정.

개발일정



05 테이블정의.

회원(MEMBER)

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
MEMBER	USERID	아이디	VARCHAR2(30)	NOT NULL	PK
	PASSWORD	비밀번호	VARCHAR2(100)	NOT NULL	
	NICKNAME	닉네임	VARCHAR2(60)	NOT NULL	
	USERNAME	이름	VARCHAR2(30)	NOT NULL	UNIQUE
	BIRTH	생년월일	DATE	NOT NULL	
	EMAIL	이메일	VARCHAR2(50)	NOT NULL	UNIQUE
	ADDRESS	주소	VARCHAR2(1000)	NOT NULL	
	PET_TYPE	펫종류	VARCHAR2(900)	NOT NULL	
	PET_NAME	펫이름	VARCHAR2(60)		
	LEVEL_TYPE	등급	NUMBER	NOT NULL	FK
	ENROLLDATE	가입일	DATE		

05 테이블정의.

게시판

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
BOARD	BNO	게시글 번호	NUMBER		PK
	CATE_NO	카테고리 번호	NUMBER		FK
	BWRITER_ID	작성자	VARCHAR2(30)		FK
	BTITLE	게시글 제목	VARCHAR2(600)	NOT NULL	
	BCONTENT	내용	VARCHAR2(3000)	NOT NULL	
	BCOUNT	조회수	NUMBER		기본값 : 0
	BFILE	첨부파일	VARCHAR2(600)		
	LIKECOUNT	좋아요	NUMBER		기본값 : 0
	BDATE	작성일	DATE		기본값 : SYSDATE
	STATUS	삭제여부	VARCHAR(1)		기본값 : 'Y'
	CCOUNT	댓글 수	NUMBER		기본값 : 0

05 테이블정의.

댓글

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
COMMENTS	CNO	댓글 번호	NUMBER		PK
	BNO	게시글 번호	NUMBER		FK
	CWRITER_ID	작성자	VARCHAR2(30)		FK
	CCONTENT	댓글내용	VARCHAR2(1000)	NOT NULL	
	CDATE	작성일	DATE		기본값 : SYSDATE
	STATUS	삭제여부 상태	VARCHAR2(1)		기본값 : 'Y'

05 테이블정의.

첨부파일

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
ATTACHMENT	FNO	첨부파일 번호	NUMBER		PK
	BNO	게시글 번호	NUMBER		FK
	ORIGINALNAME	원본이름	VARCHAR2(300)		
	CHANGENAME	변경이름	VARCHAR2(300)		
	FLEVEL	게시판 종류	NUMBER		
	UPLOADDATE	등록일	DATE		기본값 : SYSDATE
	STATUS	삭제여부 상태	VARCHAR2(1)		기본값 : 'Y'

05 테이블정의.

카테고리, 좋아요, 접근 등급

카테고리

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
CATEGORY	CATE_NO	카테고리 번호	NUMBER		PK
	CATE_NAME	카테고리 명	VARCHAR2(50)		

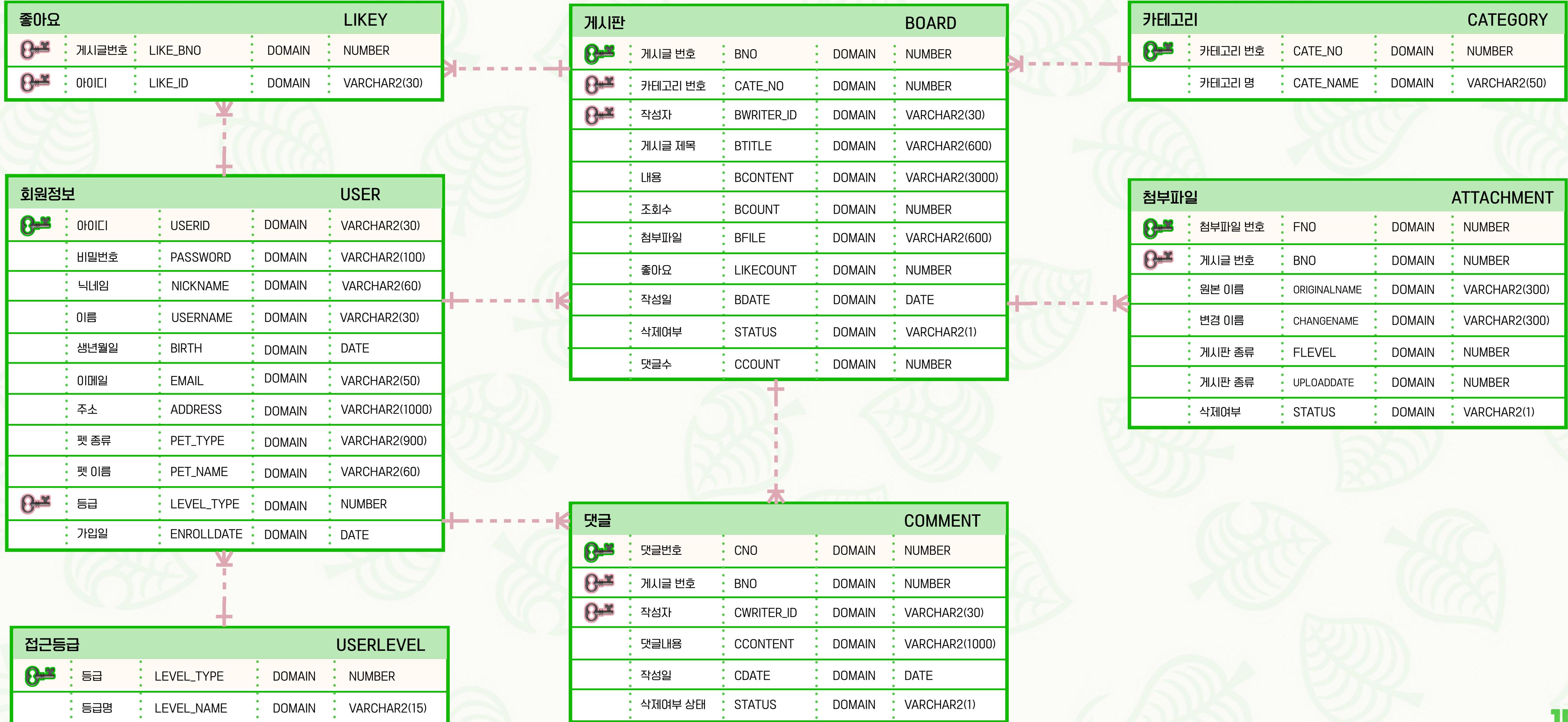
좋아요

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
LIKEY	LIKE_BNO	게시글 번호	NUMBER		FK
	LIKE_ID	아이디	VARCHAR2(30)		FK

접근 등급

테이블명	컬럼명	컬럼	타입	NULL 여부	비고
USERLEVEL	LEVEL_TYPE	등급	NUMBER		PK, 1 / 2
	LEVEL_NAME	등급명	VARCHAR2(15)		일반회원 / 관리자

06 모델링.



07 구현 기능 및 시연.

회원가입

아이디 중복, 유효성 체크, 비밀번호 유효성 체크, 이메일 입력 동물 반려 여부 (키움, 기타 - 반려동물 정보 입력 / 안키움)

로그인

아이디 존재 여부 확인, 아이디/비밀번호 찾기

회원탈퇴

가입정보 확인

마이페이지

개인정보 확인, 수정, 내가 쓴 게시글/ 댓글 확인

주변 동물병원

API 이용 위치 기반으로 주변 병원 찾기

게시판

동물병원 정보/후기공유, 동반입장 가능한 곳 정보/후기 공유, 호텔링/펫택시/애견학교 등(기타 반려동물 관련 장소의 정보/후기 공유)
우리 댕댕 자랑(사진게시판), 잡담게시판

07 구현 기능 및 시연.

Index

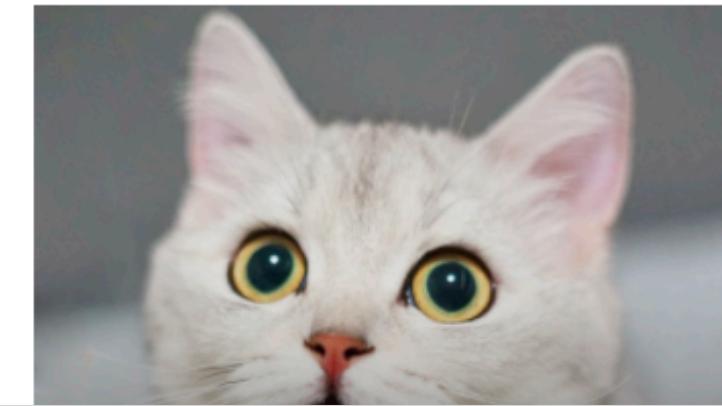
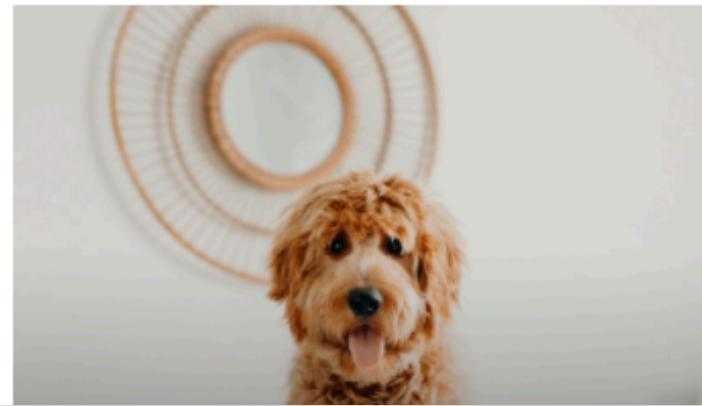
홍 | 마이페이지 | 로그아웃 |

PETmu

≡ 전체 카테고리 우리동네 동물병원 후기·정보 자유게시판 나의 댕냥 자랑



FOCUS ON



^

07 구현 기능 및 시연.

모달화면

FOCUS ON

사지 말고 입양하세요.

나와 동물을 사랑하는 방법

긴 연휴가 찾아올 때마다 사람들은 설레는 마음으로 여행을 계획하는 등 미루던 일을 꿈꿉니다. 하지만 누군가에게는 평생 잊지 못할 잔인한 순간이 될 수도 있다는 사실을 아시나요? 설날과 추석, 여름휴가가 지나고나면 항상 수많은 반려동들이 버려졌다는 소식이 들리오고, 유기동물이 많다고 소문이 난 동네에는 일부러 찾아온 관광객들이 고의로 버리고 가는 일도 허다합니다. 그렇게 버려진 아이들은 보호소에서 새로운 주인을 기다리겠지만, 그리 희망적인 일은 아닐겁니다. 하나님의 생명을 책임지는 일은 결코 가볍지 않고 우리는 더 많은 것을 알아야 하니까요!

언제나 뱃튜와 함께 와쳐주세요. 사지 말고, 입양하세요!

미소가 매력적인 땅콩이 | 2021년 1월 21일 입양

사랑하는 가족 맞이하기 전 입양 가이드 A-Z

나와 동물을 사랑하는 방법

백신접종의 종류

PETMU

COPYRIGHT © 2021 PETMU, ALL RIGHTS RESERVED. SEMIPROJECT | KH Academy reserved by Yongjoo, Minsu, Gichang, Jieun

FOCUS ON

사랑하는 가족을 맞이하기 전 준비해주세요.

입양 가이드 A-Z

평생을 행복하게 함께할 나의 가족, 반려동물! 유기동물을 입양할 때 복잡한 절차가 기다릴 것 같다고요? 사실은 그리 어렵지 않답니다. 뱃튜와 함께 자세히 알아보려 가실까요?

유기동물 입양 방법

① A-J군 구청에서 보호하고 있는 유기동물 중 보호시설을 공고한 지 10일이 지난 후 주인이 나타나지 않는 경우 일반인에게 분양할 수 있습니다.
② 입양 보호시설에 미리 전화로 문의하시고, 담당자의 안내에 따라 방문 일시 등을 예약합니다.
③ 입양 시 신분증 복사본 2회과 계집, 개 풀, 목걸이 등 필요한 물품을 준비하고 보호시설을 방문해 입양 계약서에 작성해야 합니다.
④ 입양 보호시설에는 신청자 본인의 직접 방문해야 합니다.
⑤ 미성년자에게는 반려동물을 분양하지 않습니다. 분양을 원하는 미성년자는 부모님의 허락을 얻어 반드시 부모님과 함께 방문해야 합니다. 유기동물 보호센터에 따라 입양 시 일부 경비가 청구될 수 있으니 참고해주시길 바랍니다.

멍멍! 너살 좋은 호두 | 2020년 12월 25일 입양

사랑하는 가족 맞이하기 전 입양 가이드 A-Z

나와 동물을 사랑하는 방법

백신접종의 종류

PETMU

COPYRIGHT © 2021 PETMU, ALL RIGHTS RESERVED. SEMIPROJECT | KH Academy reserved by Yongjoo, Minsu, Gichang, Jieun

FOCUS ON

반려동물과 오래오래 행복하기

백신접종의 종류

갈복송이와 함께 살아가며 멀리가 멀 수 없는 것이 바로 예방 접종이죠. 보통 어릴 땐 강아지는 5번, 고양이는 3번 접종을 해요. 우리 아기들이 과연 어떤 주사를 맞게 되는 걸까요?

보편적으로 많이 맞는 예방 접종의 종류를 알려드립니다.

강아지

- 1. 종합 7종 백신(DA2PPV)
- 2. 코로나 침경 예방접종
- 3. 엔셀프로 예방접종
- 4. 공연별 예방접종

* 층입백신은 5차까지 예회 접종, 코로나와 캔消息称는 2번씩 접종하게 됩니다!

고양이

- 1. 고양이 전염성 비기관지염(FVR, FELINE VIRAL RHINOTRACHEITIS)
- 2. 박색질구 강소증(FPL, FELINE PANLEUKOPENIA)
- 3. 쥐리시 바이러스(FCV, FELINE CALICIVIRUS)

사랑하는 가족 맞이하기 전 입양 가이드 A-Z

나와 동물을 사랑하는 방법

백신접종의 종류

PETMU

COPYRIGHT © 2021 PETMU, ALL RIGHTS RESERVED. SEMIPROJECT | KH Academy reserved by Yongjoo, Minsu, Gichang, Jieun

07 구현 기능 및 시연.

모달화면

```
<meta charset="UTF-8">
<title>PETMU</title>
<link rel="stylesheet"
      href="<%=request.getContextPath()%>/resources/css/petcss.css">
<link rel="stylesheet"
      href="<%=request.getContextPath()%>/resources/css/header.css" />

<link rel="shortcut icon" type="image/x-icon"
      href="<%=request.getContextPath()%>/resources/images/petmu.ico" />
</head>

<body>
    <%@ include file="/views/common/header.jsp"%>

    <div id="best">
        <div class="bestboard">
            
        </div>
        <div class="bestboard">
            
        </div>
    </div>

    <h1 align="center">FOCUS ON</h1>

    <div class="container">
        <!-- 첫 번째 Modal을 여는 클래스 -->
        <div class="btn">
            
        </div>

        <!-- 첫 번째 Modal -->
        <div class="modal">
            <!-- 첫 번째 Modal의 내용 -->
            <div class="modal-content">
                <div class="close">&times;</div>
                
            </div>
        </div>
    </div>
```

```
<div class="close">&times;</div>

</div>

<!-- 두 번째 Modal을 여는 클래스 -->
<div class="btn">
    
</div>

<!-- 두 번째 Modal -->
<div class="modal">
    <!-- 두 번째 Modal의 내용 -->
    <div class="modal-content">
        <div class="close">&times;</div>
        
    </div>
</div>

<!-- 세 번째 Modal을 여는 클래스 -->
<div class="btn">
    
</div>

<!-- 세 번째 Modal -->
<div class="modal">
    <!-- 세 번째 Modal의 내용 -->
    <div class="modal-content">
        <div class="close">&times;</div>
        
    </div>
</div>

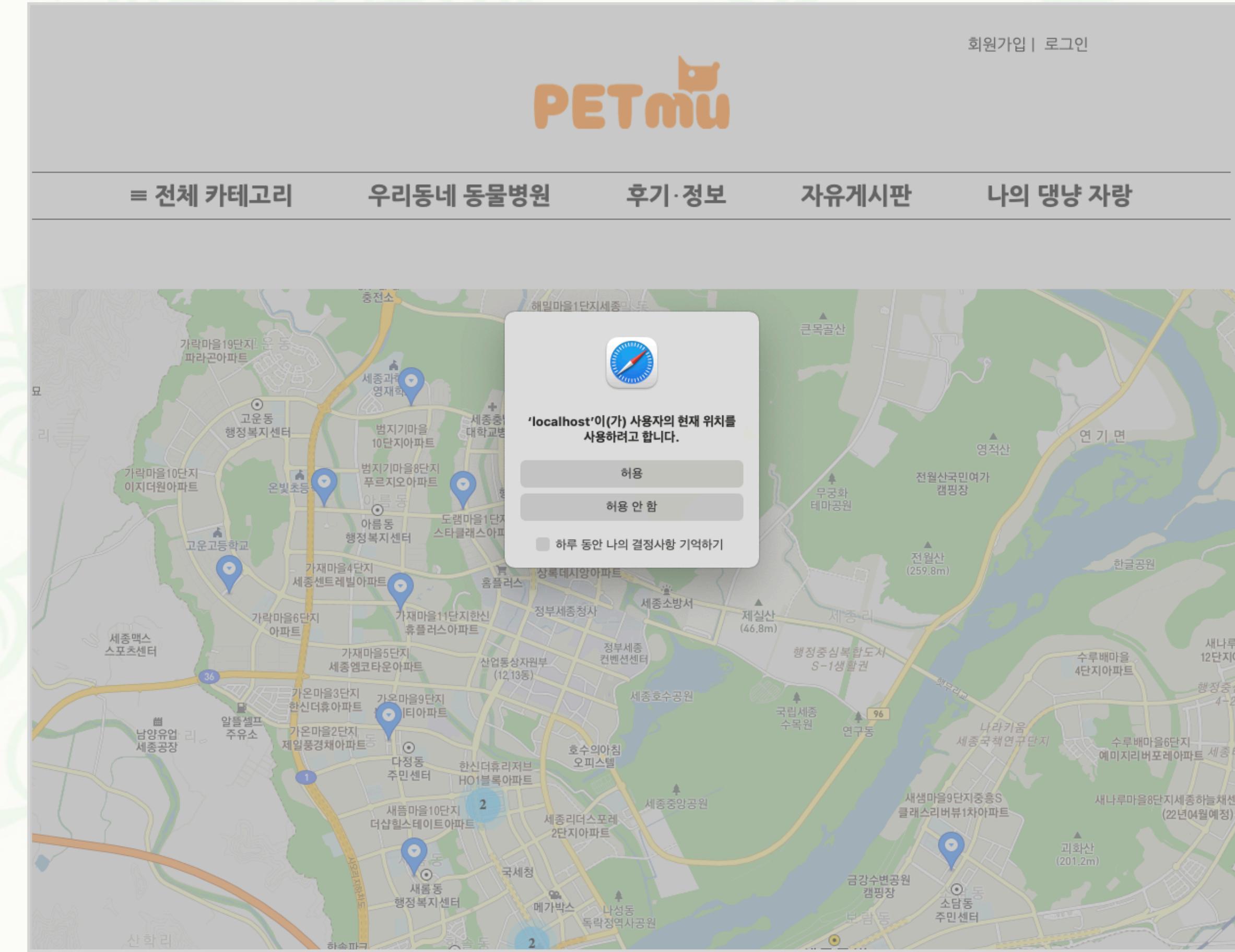
</body>
<script>
    // Modal을 가져옵니다.
    var modals = document.getElementsByClassName("modal");
    // Modal을 띄우는 클래스 이름을 가져옵니다.
    var btns = document.getElementsByClassName("btn");
    // Modal을 닫는 close 클래스를 가져옵니다.
    var spans = document.getElementsByClassName("close");
    var funcs = [];

    // Modal을 띄우고 닫는 클릭 이벤트를 정의한 함수
    function Modal(num) {
        var modal = modals[num];
        var span = spans[num];
        var btn = btns[num];
        span.onclick = function() {
            modal.style.display = "none";
        }
        btn.onclick = function() {
            modal.style.display = "block";
        }
    }
    // Modal을 띄우는 버튼을 클릭할 때 실행되는 함수
    function openModal() {
        var num = document.getElementById("num").value;
        Modal(num);
    }
</script>
```

07 구현 기능 및 시연.

API

```
// 지도를 생성한다  
var map = new kakao.maps.Map(mapContainer, mapOption);  
  
// 마커 클러스터러를 생성합니다  
var clusterer = new kakao.maps.MarkerClusterer({  
    map: map, // 마커들을 클러스터로 관리하고 표시할 지도 객체  
    averageCenter: true, // 클러스터에 포함된 마커들의 평균 위치를 클러스터 마커로 사용할지 여부  
    minLevel: 6 // 클러스터 할 최소 지도 레벨  
});  
  
// HTML5의 geolocation으로 사용할 수 있는지 확인합니다  
if (navigator.geolocation) {  
  
    // GeoLocation을 이용해서 접속 위치를 얻어옵니다  
    navigator.geolocation.getCurrentPosition(function (position) {  
  
        var lat = position.coords.latitude, // 위도  
            lon = position.coords.longitude; // 경도  
  
        var locPosition = new kakao.maps.LatLng(lat, lon), // 마커가 표시될 위치  
            message = '<div style="padding:5px;">현재위치</div>'; // 인포윈도우에 표시될 내용  
  
        // 마커와 인포윈도우를 표시합니다  
        displayMarker(locPosition, message);  
    });  
}
```



<지도 위치 서비스>

07 구현 기능 및 시연.

API

```
// 마커들을 저장할 변수 생성
var markers = [];
for (var i = 0; i < 데이터.length; i++) {
    // 지도에 마커를 생성하고 표시한다.
    var marker = new kakao.maps.Marker({
        position: new kakao.maps.LatLng(데이터[i][0], 데이터[i][1]), // 마커의 좌표
        map: map // 마커를 표시할 지도 객체
    });

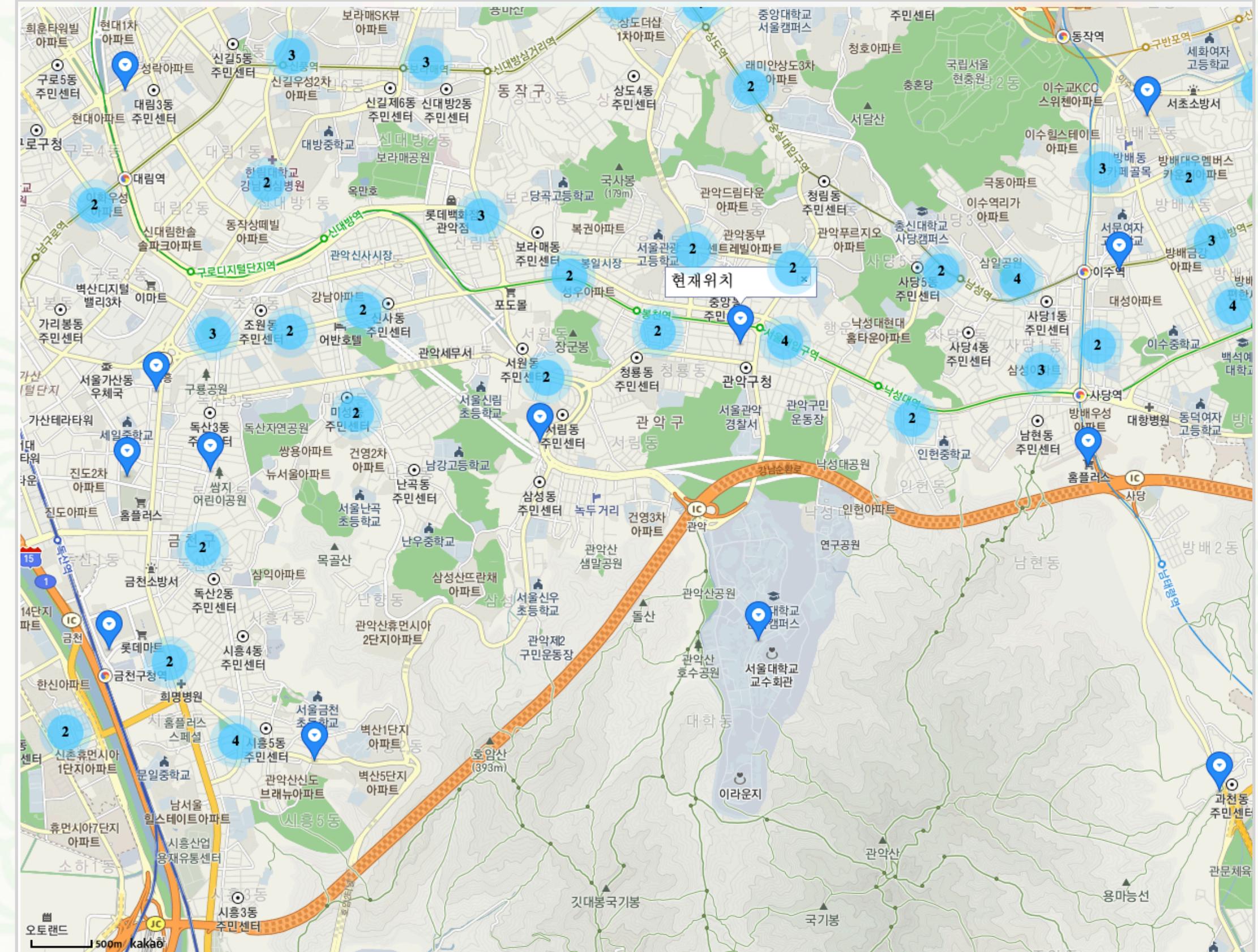
    iwPosition = new kakao.maps.LatLng(데이터[i][0], 데이터[i][1]),
    iwRemoveable = true; // removeable 속성을 true로 설정하면 인포윈도우를 닫을 수 있는 x버튼

    // 인포윈도우를 생성하고 지도에 표시합니다
    var infowindow = new kakao.maps.InfoWindow({
        position: iwPosition,
        content: 데이터[i][2],
        removable: iwRemoveable
    });

    // 생성된 마커를 마커 저장하는 변수에 넣음
    markers.push(marker);

    kakao.maps.event.addListener(marker, 'mouseover', makeOverListener(map, marker, infowindow));
    kakao.maps.event.addListener(marker, 'mouseout', makeOutListener(infowindow));
}

// 클러스터러에 마커들을 추가합니다
clusterer.addMarkers(markers);
```



<위치를 잡은 화면>

07 구현 기능 및 시연.

API

```
// 마커들을 저장할 변수 생성
var markers = [];
for (var i = 0; i < 데이터.length; i++) {
    // 지도에 마커를 생성하고 표시한다.
    var marker = new kakao.maps.Marker({
        position: new kakao.maps.LatLng(데이터[i][0], 데이터[i][1]), // 마커의 좌표
        map: map // 마커를 표시할 지도 객체
    });

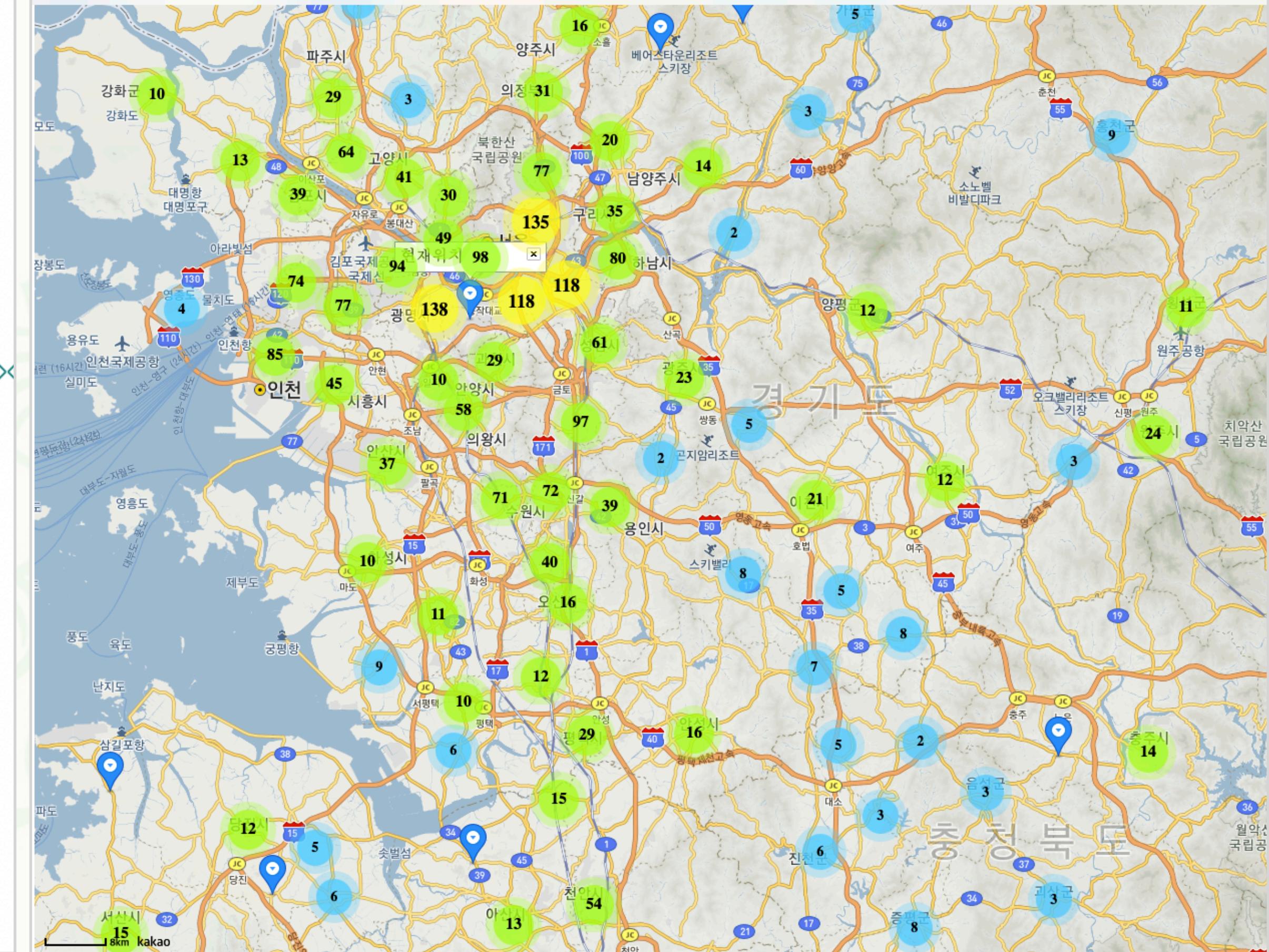
    iwPosition = new kakao.maps.LatLng(데이터[i][0], 데이터[i][1]),
    iwRemoveable = true; // removeable 속성을 true로 설정하면 인포윈도우를 닫을 때마다 마커도 삭제된다.

    // 인포윈도우를 생성하고 지도에 표시합니다
    var infowindow = new kakao.maps.InfoWindow({
        position: iwPosition,
        content: 데이터[i][2],
        removable: iwRemoveable
    });

    // 생성된 마커를 마커 저장하는 변수에 넣음
    markers.push(marker);

    kakao.maps.event.addListener(marker, 'mouseover', makeOverListener(map, marker, infowindow));
    kakao.maps.event.addListener(marker, 'mouseout', makeOutListener(infowindow));
}

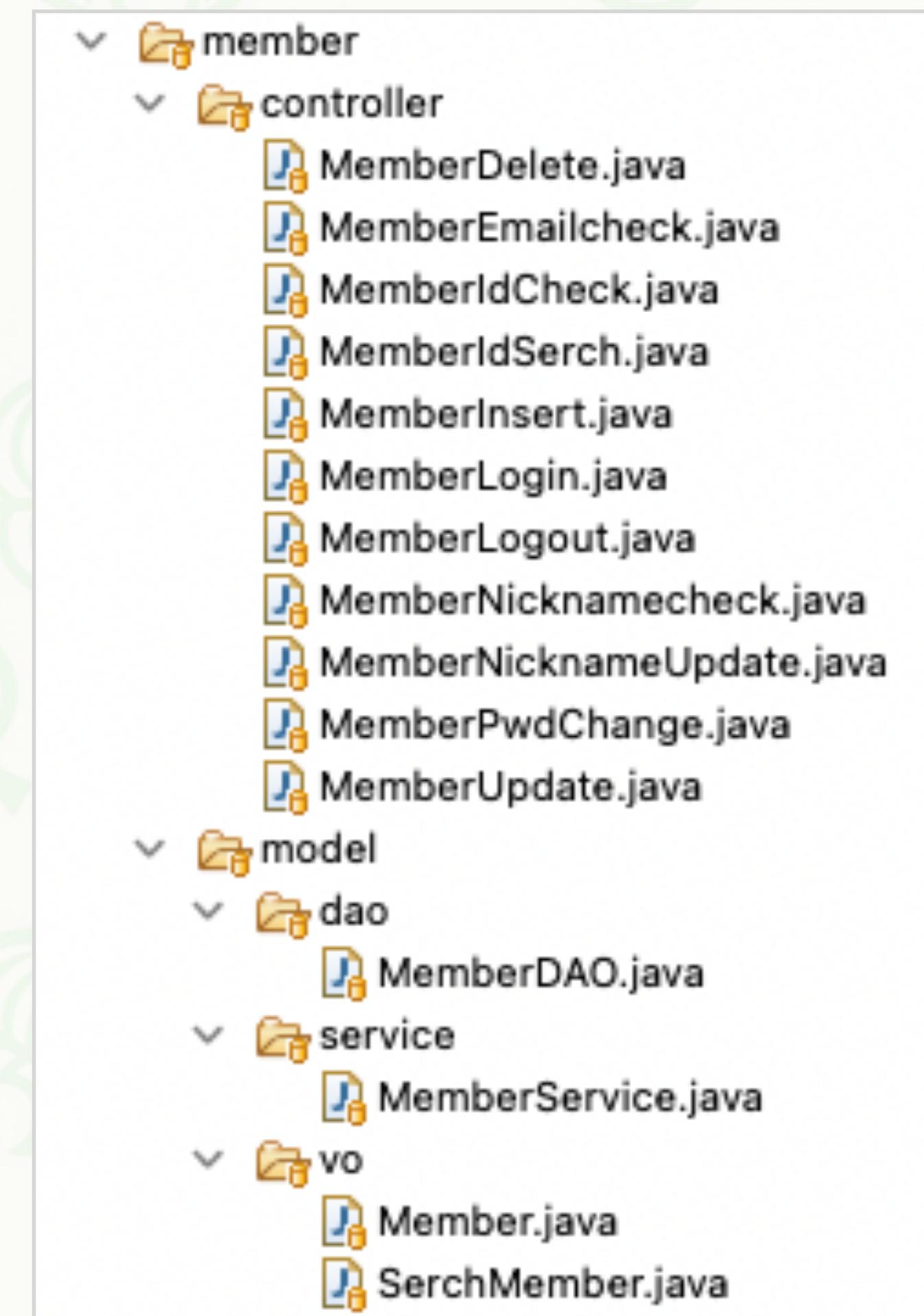
// 클러스터러에 마커들을 추가합니다
clusterer.addMarkers(markers);
```



<축소 시 화면>

07 구현 기능 및 시연.

Member Class



07 구현 기능 및 시연.

회원가입

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberInsert
 */
@WebServlet("/insert.do")
public class MemberInsert extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberInsert() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
String userId = request.getParameter("userId");
String userPwd = request.getParameter("userPwd");
String nickname = request.getParameter("nickname");
String userName = request.getParameter("userName");

String bd = request.getParameter("birth");

java.sql.Date birth= java.sql.Date.valueOf(bd);

String email = request.getParameter("email");
String address = request.getParameter("zipCode") + "|"
    + request.getParameter("address1") + "|"
    + request.getParameter("address2");
String petType= String.join(" ", request.getParameterValues("petType"));
String petName= request.getParameter("petName");
    }
```

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberIdcheck
 */
@WebServlet("/idcheck.do")
public class MemberIdCheck extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberIdCheck() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
String userId = request.getParameter("userId");
MemberService service = new MemberService();
int result = service.idcheck(userId);
response.getWriter().print(result);
}

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
// TODO Auto-generated method stub
doGet(request, response);
    }
```

<회원가입>

<아이디 중복 확인>

localhost:8088 내용:
이미 사용 중인 아이디입니다.

PETMU

회원가입

아이디
 ✓

비밀번호
 * 숫자, 영문자, 특수문자 포함 8~16자리 이내

비밀번호 확인

닉네임
 ✓

이름

확인

07 구현 기능 및 시연.

로그인

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberLogin
 */
@WebServlet("/login.do")
public class MemberLogin extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberLogin() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        String userId = request.getParameter("userId");
        String userPwd = request.getParameter("userPwd");

        // 여기까지 문제 없음!
        System.out.println("서블릿 전달 확인 : " + userId + "/" + userPwd);

        Member loginMember = new Member(userId, userPwd);
        MemberService service = new MemberService();

        loginMember = service.selectMember(loginMember);
        System.out.println("받아온 회원 정보 확인 : " + loginMember);
        if( loginMember != null ) {
            // 로그인 성공!
            HttpSession session = request.getSession();
        }
    }
}
```

<로그인>

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberLogout
 */
@WebServlet("/logout.do")
public class MemberLogout extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberLogout() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        HttpSession session = request.getSession(false);

        if(session != null) {
            session.invalidate();
        }
        response.sendRedirect("index.jsp");
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
        // TODO Auto-generated method stub
        doGet(request, response);
    }
}
```

<로그아웃>



로그인

ID

PASSWORD

◀ 로그인 | ▶ 회원가입

07 구현 기능 및 시연.

아이디/비밀번호 찾기

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberIdSerch
 */
@WebServlet("/idserch.do")
public class MemberIdSerch extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberIdSerch() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        String userName = request.getParameter("userName");
        String email = request.getParameter("email");

        System.out.println("서블릿 전달 확인 : " + userName + "/" + email);
        SerchMember serchMember = new SerchMember(userName, email);
        MemberService service = new MemberService();
        String result = service.memberSerch(serchMember);
        System.out.println("서블릿에서 받은 result = " + result);

        response.getWriter().print(result);
    }
}
```

ID / PW 찾기

아이디 비밀번호

회원 가입 시 작성한 이름과 E-Mail을 입력해주세요.

이름

E-Mail

아이디 찾기



COPYRIGHT © 2021 PETMU, ALL RIGHTS RESERVED. SEMIPROJECT | KH Academy reserved by Yongjoo, Minsu, Gichang, Jieun

07 구현 기능 및 시연.

마이페이지

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="com.kh.petmu.board.model.vo.*, java.util.*"%>

<%
ArrayList<Board> list = (ArrayList<Board>)request.getAttribute("list");
PageInfo pi = (PageInfo)request.getAttribute("pageInfo");

int startPage = pi.getStartPage();
int endPage = pi.getEndPage();
int maxPage = pi.getMaxPage();
int currentPage = pi.getCurrentPage();
int listCount = pi.getListCount();
%>

<!DOCTYPE html>
<html>
<head>
    <title>PETMU : 마이페이지</title>
    <link rel="petmu icon" href="<%=request.getContextPath()%>/resources/images/petmu.ico">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <script src="https://kit.fontawesome.com/c10cbac54f.js" crossorigin="anonymous"></script>
    <style>
        #td{
            font-size: 20px;
            padding-right: 300px;
        }
        #tr{
            height: 50px;
        }
        #btn{
            color: coral;
            font-size: 20px;
            font-weight: 600;
        }
    </style>

```

마이페이지

홍	
아이디	user01
이름	홍길동
생년월일	2021-10-15
E-Mail	user01@example.com
주소	46760 부산 강서구 신호동 294 로운
반려동물 정보	(없음) 없음

개인정보수정 회원탈퇴

내가 작성한 게시글

No.	제목	카테고리	작성일	조회수	추천수

07 구현 기능 및 시연.

개인정보 수정

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberUpdate
 */
@WebServlet("/update.do")
public class MemberUpdate extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberUpdate() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    // 개인정보 수정 창에서 입력한 닉네임 받아오기

    String userPwd = request.getParameter("userPwd");
    String address = request.getParameter("zipCode") + "|"
        + request.getParameter("address1") + "|"
        + request.getParameter("address2");
    String petType = String.join(", ", request.getParameterValues("petType"));
    String petName = request.getParameter("petName");

    // 해당 회원을 구분짓는 ID 받아오기
    HttpSession session = request.getSession(false);
    Member m = (Member)session.getAttribute("member");
    // 변경한 개인정보 등록하기
    m.setUserPwd(userPwd);
    m.setAddress(address);
```

<정보 수정>



개인정보수정

아이디	user01
비밀번호	<small>* 숫자, 영문자, 특수문자 포함 8~16자리 이내</small>
비밀번호 확인	
닉네임	홍
이름	홍길동
생년월일	2021-10-15
이메일	user01@example.com
주소	

07 구현 기능 및 시연.

개인정보 수정

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberNicknameUpdate
 */
@WebServlet("/nicknameUpdate.do")
public class MemberNicknameUpdate extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberNicknameUpdate() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        // 닉네임 변경창에서 입력한 닉네임 받아오기
        String nickname = request.getParameter("nickname");

        // 해당 회원을 구분짓는 ID 받아오기
        HttpSession session = request.getSession(false);
        Member m = (Member)session.getAttribute("member");
        // 변경한 닉네임 등록하기
        m.setNickname(nickname);
        System.out.println("변경한 회원 정보 확인 : " + m);
        MemberService service = new MemberService();
        int result = service.nicknameUpdate(m);
    }
}
```

<닉네임 변경>

```
package com.kh.petmu.member.controller;

import java.io.IOException;

/**
 * Servlet implementation class MemberPwdChange
 */
@WebServlet("/changepw.do")
public class MemberPwdChange extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public MemberPwdChange() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        String userId = request.getParameter("userId");
        String userName = request.getParameter("userName");
        String email = request.getParameter("email");
        String userPwd = request.getParameter("userPwd");

        System.out.println("서블릿 전달 확인 : " + userName + "/" + email + "/" + userId +
            Member m = new Member(userId, userName, email, userPwd);
            MemberService service = new MemberService();
            int result = service.changePw(m);
            if(result > 0) {
                // 비밀번호 변경 성공!
                response.sendRedirect("views/member/login_page2.jsp");
            } else {
        }
    }
}
```

<비밀번호 변경>

PETmu

개인정보수정

아이디	user01
비밀번호	<input type="text"/> * 숫자, 영문자, 특수문자 포함 8~16자리 이내
비밀번호 확인	<input type="text"/>
닉네임	홍
이름	홍길동
생년월일	2021-10-15
이메일	user01@example.com
주소	<input type="text"/>

07 구현 기능 및 시연.

회원탈퇴

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
String userId = request.getParameter("userId");
String userPwd = request.getParameter("userPwd");

// 여기까지 문제 없음!
System.out.println("탈퇴 서블릿 전달 확인 : " + userId + "/" + userPwd);

Member loginMember = new Member(userId, userPwd);

MemberService service = new MemberService();
loginMember = service.selectMember(loginMember);

System.out.println("받아온 탈퇴 회원 정보 확인 : " + loginMember);

if( loginMember != null ) {
    // 로그인 성공!

    System.out.println("삭제할 회원 아이디 : " + userId);

    int result = service.deleteMember(userId);

    if( result > 0 ) {

        HttpSession session = request.getSession(false);
        session.invalidate();
        response.sendRedirect("index.jsp");
    } else {

        RequestDispatcher view
        = request.getRequestDispatcher("views/common/errorPage.jsp");
        request.setAttribute("error-msg", "회원 정보 삭제 실패");
        view.forward(request, response); }
```

회원탈퇴

- ✓ 사용하고 계신 아이디의 탈퇴가 진행되면, 재사용 및 복구가 불가능합니다.

탈퇴한 아이디는 본인과 타인 모두 재사용 및 복구가 불가하오니 신중하게 선택하시기 바랍니다.

- ✓ 탈퇴 후에도 게시판형 서비스에 등록한 게시물은 그대로 남아 있습니다.

자유게시판, 사진게시판 등에 올린 게시글 및 댓글은 탈퇴 시 자동 삭제되지 않고 그대로 남아 있습니다.
삭제를 원하는 게시글이 있다면 반드시 [탈퇴 전 삭제하시기 바랍니다.](#)

탈퇴 후에는 회원정보가 삭제되어 본인 여부를 확인할 수 있는 방법이 없어, 게시글을 임의로 삭제해드릴 수 없습니다.

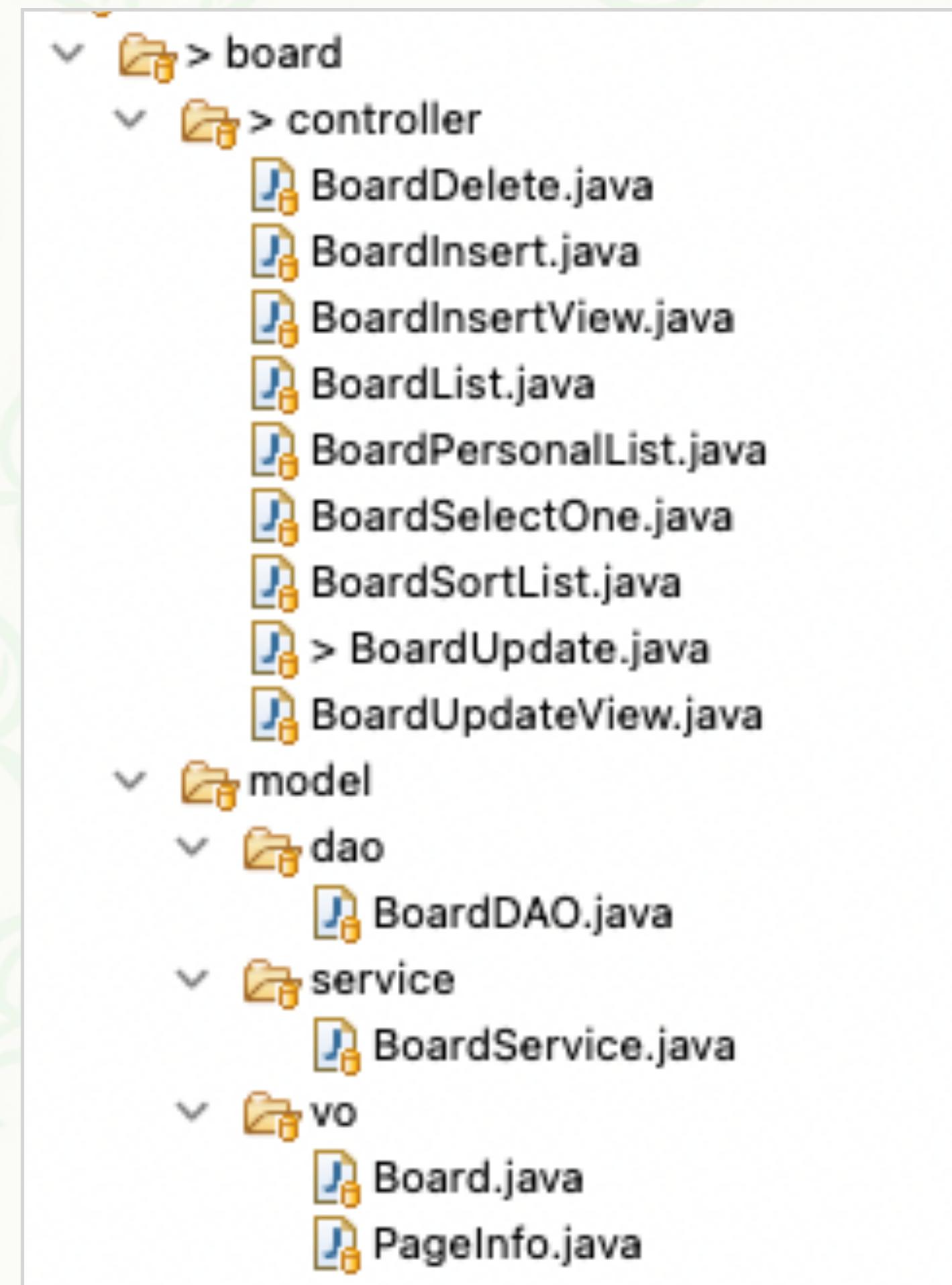
- ✓ 탈퇴 후에는 해당 아이디로 다시 가입할 수 있으며 다른 데이터는 복구할 수 없습니다.
게시판형 서비스에 남아 있는 게시글은 탈퇴 후 삭제할 수 없습니다.

개인정보 보호를 위해 아이디와 비밀번호를 확인해주세요.

[탈퇴하기](#)

07 구현 기능 및 시연.

Board Class



07 구현 기능 및 시연.

제시글 메인화면

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="com.kh.petmu.board.model.vo.* , com.kh.petmu.comment.model.vo.* , java.util.*%>

<%--vo임포트하고 board, comment list 가져오기 --%>

<%
Board b = (Board)request.getAttribute("board");
ArrayList<Comment> colist = (ArrayList<Comment>)request.getAttribute("colist");
int cmtCount = (int)request.getAttribute("cmtCount");
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>PETMU : 병원 / 약국 정보</title>
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.15.4/css/all.css" integrity="sha384-DyZ88mC6Up2uq54h/KRgHuoeGwBcD4Ng95."/>
<link rel="stylesheet" href="<%= request.getContextPath()%>/resources/css/sideMenu.css" />
<link rel="stylesheet" href="<%= request.getContextPath()%>/resources/css/header.css" />
<link rel="stylesheet" href="<%= request.getContextPath()%>/resources/css/boardDetail.css" />

<link rel="shortcut icon" type="image/x-icon" href="<%= request.getContextPath()%>/resources/images/petmu.ico" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css" />
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Jua&display=swap" rel="stylesheet">

<script src="<%= request.getContextPath()%>/resources/js/jquery-3.6.0.min.js" type="text/javascript"></script>

</head>
<body>
<%@ include file="/views/common/header.jsp" %>
<div class="main">
<%@ include file="/views/common/sideMenu.jsp" %>

<div class="boardArea">
<div id="boardTitle" style="font-family: 'Jua', sans-serif; font-size : 30px; padding-bottom : 10px;">
<i class="far fa-edit"></i>
<label style="color : orange;"> 병원 / 약국 정보 공유</label>
</div>
<div id="content">
```

동물병원 / 약국

병원 / 약국 정보 공유

이용 후기

동반입장 장소

장소 공유

이용 후기

기타 장소

장소 공유

이용 후기

▣ 병원 / 약국 정보 공유

No.	제목	글쓴이	작성일	조회수	추천수
1	안녕하세요	홍	2021-10-15	0	👍 +0

제목 검색어를 입력해주세요 검색

<메인화면>

07 구현 기능 및 시연.

제시글 메인화면

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<%
int cateNo = (int)request.getAttribute("cate");
%>

<div class="sideMenu">
<div id="sideArea">
<ul>
<li class="listTitle">동물병원 / 약국
<ul>
<%if(cateNo == 3) {%
<li style="border-bottom: solid 3px orange; ">
<a href="<%= request.getContextPath()%>/selectList.bo?cate=3" style="font-weight: bold;">병원 / 약국 정보 공유
</li>
<%} else {%
<li class="nonFocus"><a href="<%= request.getContextPath()%>/selectList.bo?cate=3">병원 / 약국 정보 공유</a>
<br/>
%}>
<%}%
<%if(cateNo == 4) {%
<li style="border-bottom: solid 3px orange; ">
<a href="<%= request.getContextPath()%>/selectList.bo?cate=4" style="font-weight: bold;">이용 후기
</li>
<%}%
<%} else {%
<li class="nonFocus"><a href="<%= request.getContextPath()%>/selectList.bo?cate=4">이용 후기</a>
<br/>
%}>
<%}%
</ul>
<li class="listTitle">동반입장 장소
<ul>
<%if(cateNo == 5) {%
<li style="border-bottom: solid 3px orange; ">
<a href="<%= request.getContextPath()%>/selectList.bo?cate=5" style="font-weight: bold;">장소 공유
</li>
<%} else {%
<li class="nonFocus"><a href="<%= request.getContextPath()%>/selectList.bo?cate=5">장소 공유</a>
<br />
%}>
<%}%
</ul>
</li>
</ul>
</div>
</div>
```

동물병원 / 약국

병원 / 약국 정보 공유

이용 후기

동반입장 장소

장소 공유

이용 후기

기타 장소

장소 공유

이용 후기

병원 / 약국 정보 공유

No.	제목	글쓴이	작성일	조회수	추천수
1	안녕하세요	홍	2021-10-15	0	+ 0

제목 : 검색어를 입력해주세요

검색

<사이트바>

07 구현 기능 및 시연.

게시글 검색

```
// 게시글 리스트 가져오기
if(target == null && keyword == null) {
    // 검색하지 않았을 때
    list = service.selectList(cateNo, currentPage, limit);
} else if(target.equals("title") && keyword != null) {
    // 제목으로 검색
    list = service.selectTitleList(cateNo, currentPage, limit, keyword);
} else if(target.equals("writerId") && keyword != null) {
    // 아이디로 검색
    list = service.selectWriterIdList(cateNo, currentPage, limit, keyword);
} else if(target.equals("writerNick") && keyword != null) {
    // 닉네임으로 검색
    list = service.selectWriterNickList(cateNo, currentPage, limit, keyword);
}

String page2 = null;

if(list != null) {
    PageInfo pi = new PageInfo(startPage, endPage, maxPage, currentPage, limit, listCount);

    request.setAttribute("cate", cateNo);
    request.setAttribute("list", list);
    request.setAttribute("pageInfo", pi);

    // 카테고리 번호에 맞는 게시글 페이지 등록
    switch(cateNo) {
        case 3 : page2 = "views/board/info_review/hospital/infoList.jsp";
        break;
        case 4 : page2 = "views/board/info_review/hospital/reviewList.jsp";
        break;
        case 5 : page2 = "views/board/info_review/place/infoList.jsp";
    }
}
```

▣ 병원 / 약국 정보 공유

No.	제목	글쓴이	작성일	조회수	추천수
	<input type="text" value="검색어를 입력해주세요"/> <input type="button" value="검색"/>				

07 구현 기능 및 시연.

게시글 등록

```
package com.kh.petmu.board.controller;

import java.io.IOException;

/**
 * Servlet implementation class BoardInsert
 */
@WebServlet("/insert.bo")
public class BoardInsert extends HttpServlet {
    private static final long serialVersionUID = 1L;

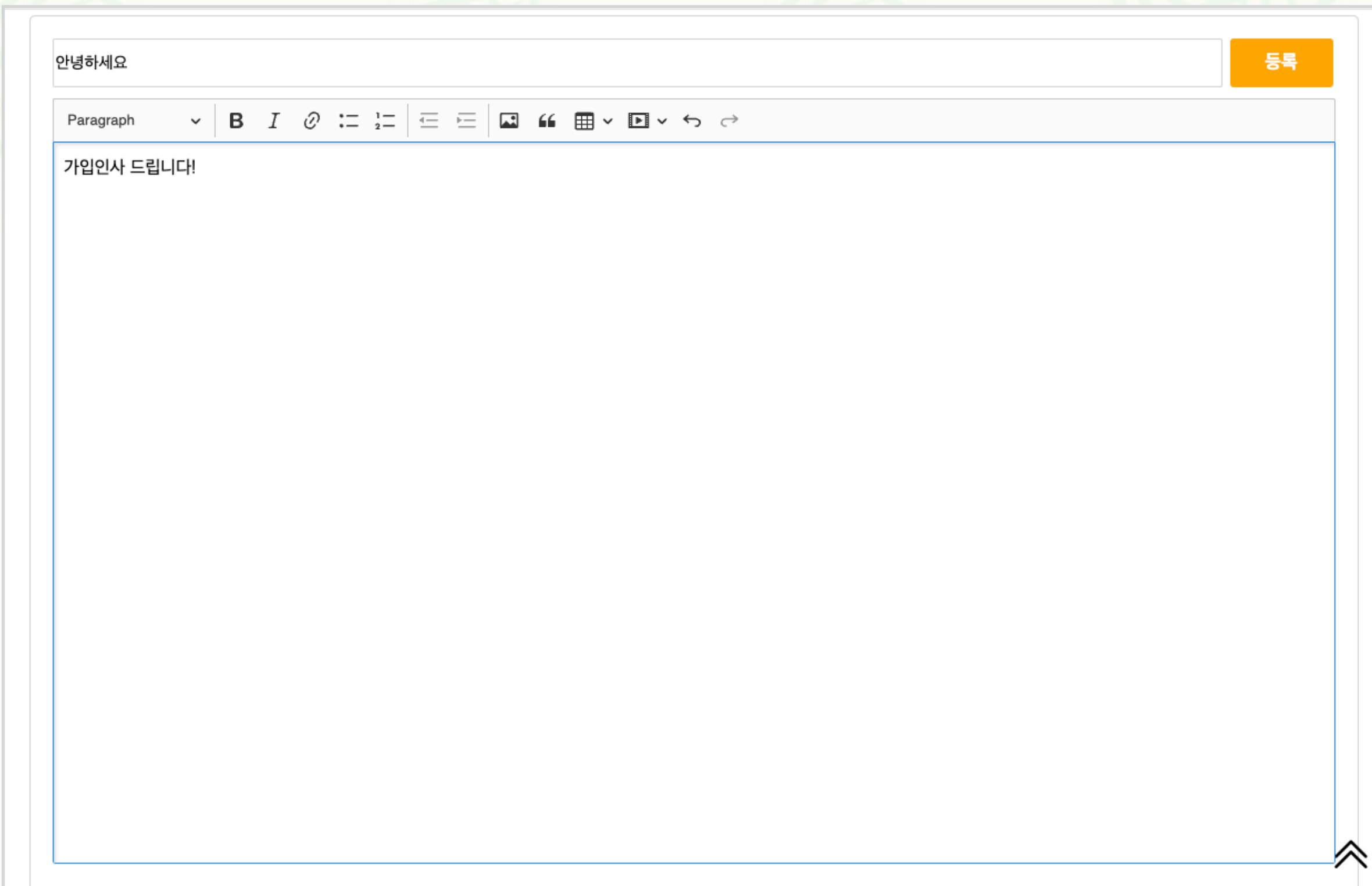
    /**
     * @see HttpServlet#HttpServlet()
     */
    public BoardInsert() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        int cateNo = Integer.parseInt(request.getParameter("cate"));

        String bwriterId = request.getParameter("bwriterId");
        String bwriterNick = request.getParameter("bwriterNick");
        String btitle = request.getParameter("btitle");
        String bcontent = request.getParameter("bcontent");

        // 게시글 저장 vo
        Board b = new Board();

        b.setBwriterId(bwriterId);
        b.setBwriterNick(bwriterNick);
        b.setBtitle(btitle);
        b.setBcontent(bcontent);
        b.setcateNo(cateNo);
```



07 구현 가능 및 시연.

제작

```
package com.kh.petmu.board.controller;

import java.io.IOException;

/**
 * Servlet implementation class BoardUpdate
 */
@WebServlet("/update.bo")
public class BoardUpdate extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public BoardUpdate() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
        int cateNo = Integer.parseInt(request.getParameter("cate"));

        int bno = Integer.parseInt(request.getParameter("bno"));
        String btitle = request.getParameter("btitle");
        String bcontent = request.getParameter("bcontent");

        // 게시글 저장 vo
        Board b = new Board();

        BoardService service = new BoardService();

        b = service.updateView(bno);

        b.setBtitle(btitle);
        b.setBcontent(bcontent);

        int result = service.updateBoard(b);

        String page = "";
```

07 구현 기능 및 시연.

게시글 추천

```
/**  
 * Servlet implementation class likeUpdate  
 */  
@WebServlet("/likeUpdate.do")  
public class likeUpdate extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public likeUpdate() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        int bno = Integer.parseInt(request.getParameter("bno"));  
        String userId = request.getParameter("id");  
  
        //System.out.println("서블릿 체크 : " + bno + userId);  
        likeService service = new likeService();  
  
        // 해당 게시글에 좋아요를 눌렀는지 db로 확인  
        int result1 = service.selectLike(bno, userId);  
  
        int like = 0;  
        String text = null;  
  
        if(result1 == 0) { // 좋아요를 누른적이 없으면  
            // bno의 like 값 +1  
            int result2 = service.updateLike(bno);  
  
            // 좋아요 개수 가져오기  
            like = service.getLikeCount(bno);  
        }  
    }  
}
```

The screenshot shows a web application interface. On the left, a code editor displays Java code for a servlet named 'likeUpdate'. The code handles HTTP GET requests to update a post's like count and retrieve the current count. On the right, there are two main sections: a 'Post Detail' view and a 'Confirmation Dialog'.

Post Detail View:

- Header:** 신세 가네고디, 주디, localhost:8088 내용: 이 게시글을 추천 하시겠습니까?
- Content:** 동반입장 장소 정보 ...
- User Info:** 안녕하세요 (user01)
- Comments:** 0 Like (수정, 삭제)
- Text Area:** 댓글 0 (등록)
- Buttons:** 목록

Confirmation Dialog:

- Header:** 신세 가네고디, 주디, localhost:8088 내용: 이 게시글을 추천 하시겠습니까?
- Buttons:** 취소, 확인

07 구현 기능 및 시연.

게시글 삭제

```
* @see HttpServlet#HttpServlet()
*/
public BoardDelete() {
    super();
    // TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
    int cateNo = Integer.parseInt(request.getParameter("cate"));
    int bno = Integer.parseInt(request.getParameter("bno"));

    int result = new BoardService().deleteBoard(bno);

    String page = "";
    if(result > 0) {
        request.setAttribute("cate", cateNo);
        // 각 카테고리별 리스트로 이동
        switch(cateNo) {
            case 3 : page = "selectList.bo?cate=3";
            break;
            case 4 : page = "selectList.bo?cate=4";
            break;
            case 5 : page = "selectList.bo?cate=5";
            break;
            case 6 : page = "selectList.bo?cate=6";
            break;
            case 7 : page = "selectList.bo?cate=7";
            break;
            case 8 : page = "selectList.bo?cate=8";
            break;
        }
        response.sendRedirect(page);
    } else {
        request.setAttribute("error-msg", "게시글 삭제에 실패하였습니다.");
        request.getRequestDispatcher("views/common/errorPage.jsp").forward(request, response);
    }
}
```

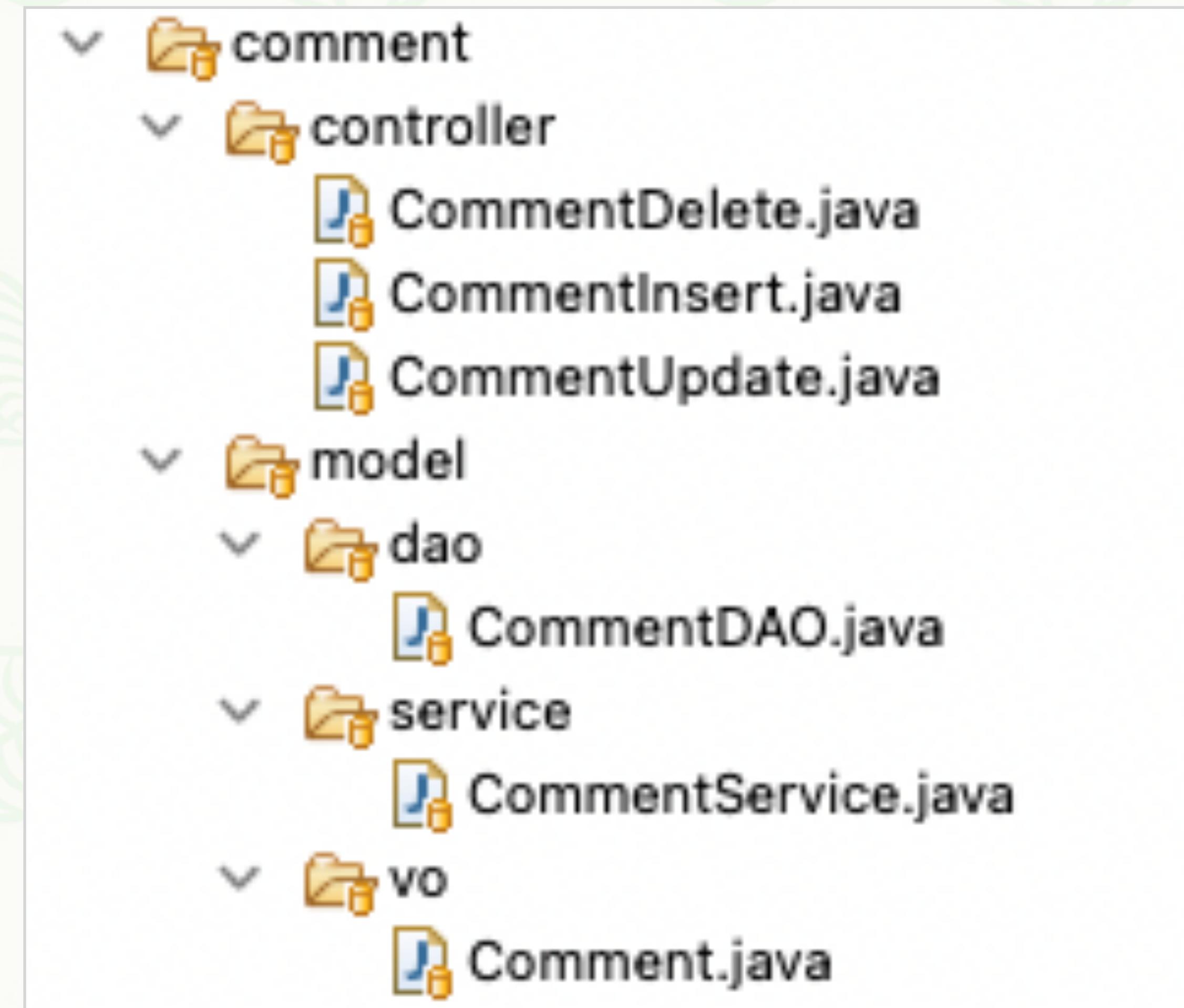
The screenshot shows a web application interface. At the top, there's a navigation bar with tabs: '전체 카테고리', '우리' (selected), '게시판', and '나의 냉장고 자당'. Below the navigation, a modal dialog is open with the title '동반입장 장소 정보'. It contains the message 'localhost:8088 내용: 삭제시 복구 할 수 없습니다. 삭제하시겠습니까?' with two buttons: '취소' and '확인'. The main content area shows a post with the following details:

- 작성자:** 흥 (user01)
- 작성일:** 2021-10-15
- 조회수:** 1
- 추천수:** 0
- 댓글 수:** 0

The post content is: '가입인사 드립니다!'. Below the post, there are 'Like' and 'Delete' buttons. A comment input field is present with the placeholder '댓글을 남겨 보세요.' and a 'Reply' button. At the bottom right of the post area, there's a '목록' (List) button.

07 구현 기능 및 시연.

Comment Class



07 구현 기능 및 시연.

댓글 등록

```
/**  
 * Servlet implementation class CommentInsert  
 */  
@WebServlet("/cmtInsert.co")  
public class CommentInsert extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public CommentInsert() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        // ajax로 보낸 데이터 가져오기  
        int bno = Integer.parseInt(request.getParameter("bno"));  
        String cwriterId = request.getParameter("cwriterId");  
        String cwriterNick = request.getParameter("cwriterNick");  
        String ccontent = request.getParameter("ccontent");  
  
        // 데이터 담을 vo  
        Comment co = new Comment();  
  
        co.setBno(bno);  
        co.setCwriterId(cwriterId);  
        co.setCwriterNick(cwriterNick);  
        co.setCcontent(ccontent);  
  
        // System.out.println(co); 확인 완료  
  
        // 서비스로 보내기  
        CommentService service = new CommentService();
```

동반입장 장소 정보 공유

안녕하세요

홍 (user01)

2021-10-15

조회수 2 추천수 1 댓글 1

가입인사 드립니다!

1 Like

수정 삭제

댓글 1

댓글을 남겨 보세요.

등록

홍

안녕하세요!

목록

2021-10-15

07 구현 기능 및 시연.

댓글 삭제

```
/**  
 * Servlet implementation class CommentDelete  
 */  
@WebServlet("/cmtDelete.co")  
public class CommentDelete extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public CommentDelete() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        int cno = Integer.parseInt(request.getParameter("cno"));  
        int bno = Integer.parseInt(request.getParameter("bno"));  
  
        //System.out.println("서블릿 : " + cno); //확인 완료  
        //System.out.println("서블릿 : " + bno); //확인 완료  
  
        int result = new CommentService().deleteComment(cno, bno);  
  
        if(result > 0) {  
            response.getWriter().print(result);  
        } else {  
            request.setAttribute("error-msg", "댓글 삭제 실패");  
            request.getRequestDispatcher("views/common/errorPage").forward(request, response);  
        }  
    }  
  
    /**  
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)  
     */
```

동반입장 장소 정보 ~~

localhost:8088 내용:

삭제시 복구 할 수 없습니다. 삭제하시겠습니까?

작성자: 흥 (user01)

작성일: 2021-10-15

조회수 2 추천수 1 댓글 1

취소 확인

가입인사 드립니다!

1 Like

수정 삭제

댓글 1

댓글을 남겨 보세요.

등록

작성자: 흥

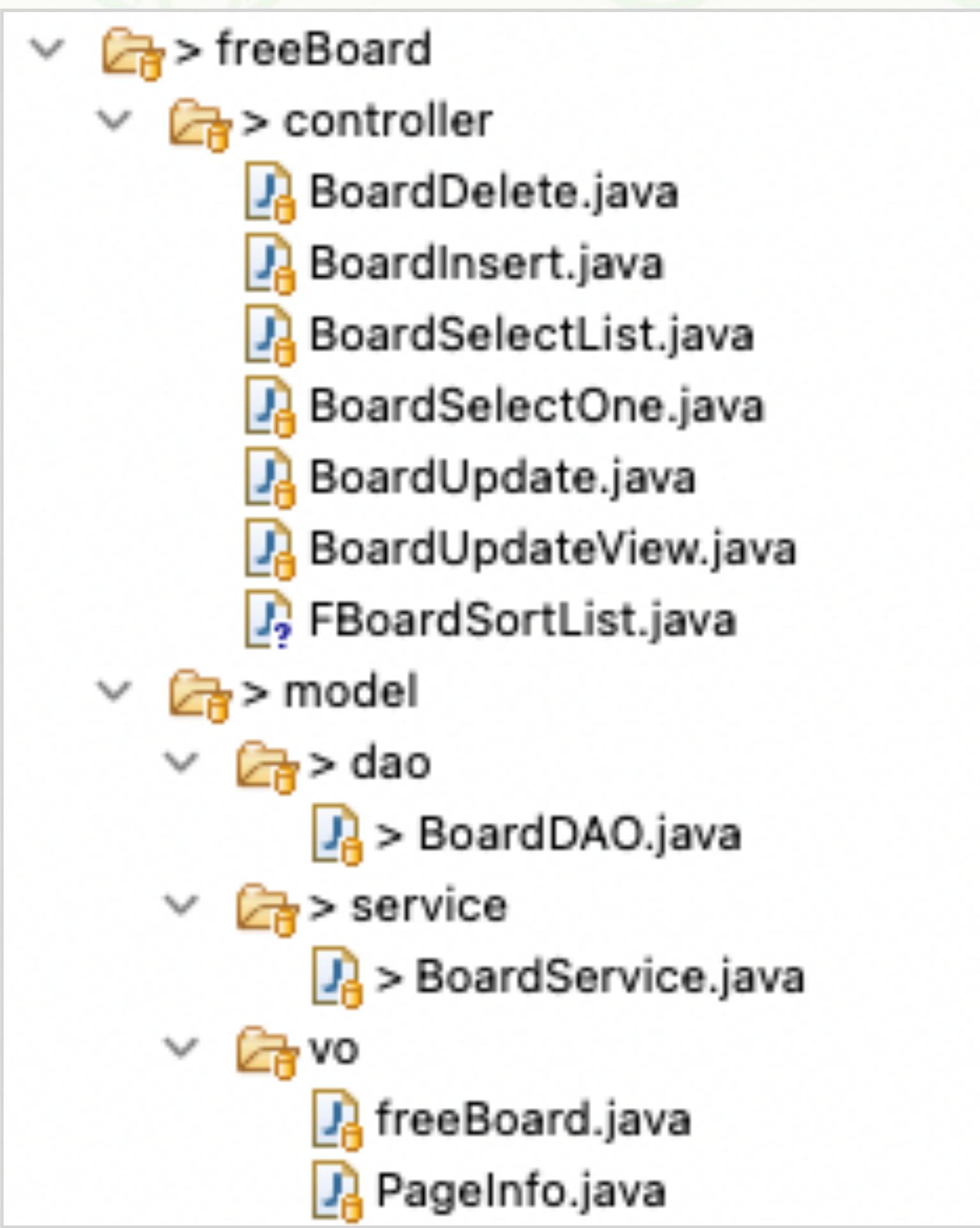
작성일: 2021-10-15

안녕하세요!

목록

07 구현 기능 및 시연.

FreeBoard Class



07 구현 기능 및 시연.

자유게시판

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" import="com.kh.petmu.freeBoard.model.vo.*, java.util.*" %>
<%
ArrayList<freeBoard> list = (ArrayList<freeBoard>)request.getAttribute("list");
 PageInfo pi = (PageInfo)request.getAttribute("pi");
 int listCount = pi.getListCount();
 int currentPage = pi.getCurrentPage();
 int maxPage = pi.getMaxPage();
 int startPage = pi.getStartPage();
 int endPage = pi.getEndPage();
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>PETMU : 자유게시판</title>
<script src="<%= request.getContextPath()%>/resources/js/jquery-3.6.0.min.js"></script>
<!-- CSS 적용 -->
<link rel="stylesheet" href="<%= request.getContextPath()%>/resources/css/header.css" />
<link rel="stylesheet" href="<%= request.getContextPath()%>/resources/css/FboardList.css" />
<!-- 타이틀로고 -->
<link rel="shortcut icon" type="image/x-icon" href="<%= request.getContextPath()%>/resources/images/petmu.ico" />
<!-- fontawesome 아이콘cdn -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css" />
<!-- 구글폰트 cdn -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Jua&display=swap" rel="stylesheet">
</head>
<body>
<%@ include file="../common/header.jsp" %>
```

The screenshot shows the PETMU website's free board section. At the top, there is a navigation bar with links for 'Home', 'My Page', 'Logout', '전체 카테고리' (All Categories), '우리동네 동물병원' (Our Neighborhood Animal Hospital), '후기·정보' (Reviews·Information), '자유게시판' (Free Board), and '나의 댕냥 자랑' (Show off my puppy). The main content area has a title '자유게시판' (Free Board) and a search bar with dropdown menus for '제목' (Title) and '작성일' (Date). Below the search bar, there is a message '검색어를 입력해주세요' (Please enter a search term). At the bottom, there is a copyright notice: 'COPYRIGHT © 2021 PETMU, ALL RIGHTS RESERVED. SEMIPROJECT | KH Academy reserved by Yongjoo, Minsu, Gichang, Jieun'.

07 구현 가능 및 시연.

자유게시글 작성

```
/**  
 * Servlet implementation class BoardInsert  
 */  
@WebServlet("/insert.fb")  
public class BoardInsert extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public BoardInsert() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        // 파일 업로드용 서블릿  
        // MultipartRequest  
  
        // 1. 전송받은 최대크기 설정하기  
        int maxSize = 1024 * 1024 * 10; // 10MB  
  
        // 2. multipart/form-data 형식으로 인코딩 되어 왔는지 확인  
        if(! ServletFileUpload.isMultipartContent(request)) {  
  
            //에러 페이지  
            request.setAttribute("error-msg", "multipart로 전송되지 않았습니다.");  
  
            request.getRequestDispatcher("views/common/errorPage.jsp")  
                .forward(request, response);  
        }  
  
        // 3. 받아온 파일을 저장할 경로 설정하기  
        String root = request.getServletContext().getRealPath("/");  
        String savePath = root + "resources/boardUploadFiles";
```

07 구현 기능 및 시연.

파일첨부

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // 파일 업로드용 서블릿
    // MultipartRequest

    // 1. 전송받은 최대크기 설정하기
    int maxSize = 1024 * 1024 * 10; // 10MB

    // 2. multipart/form-data 형식으로 인코딩 되어 왔는지 확인
    if(! ServletFileUpload.isMultipartContent(request)) {

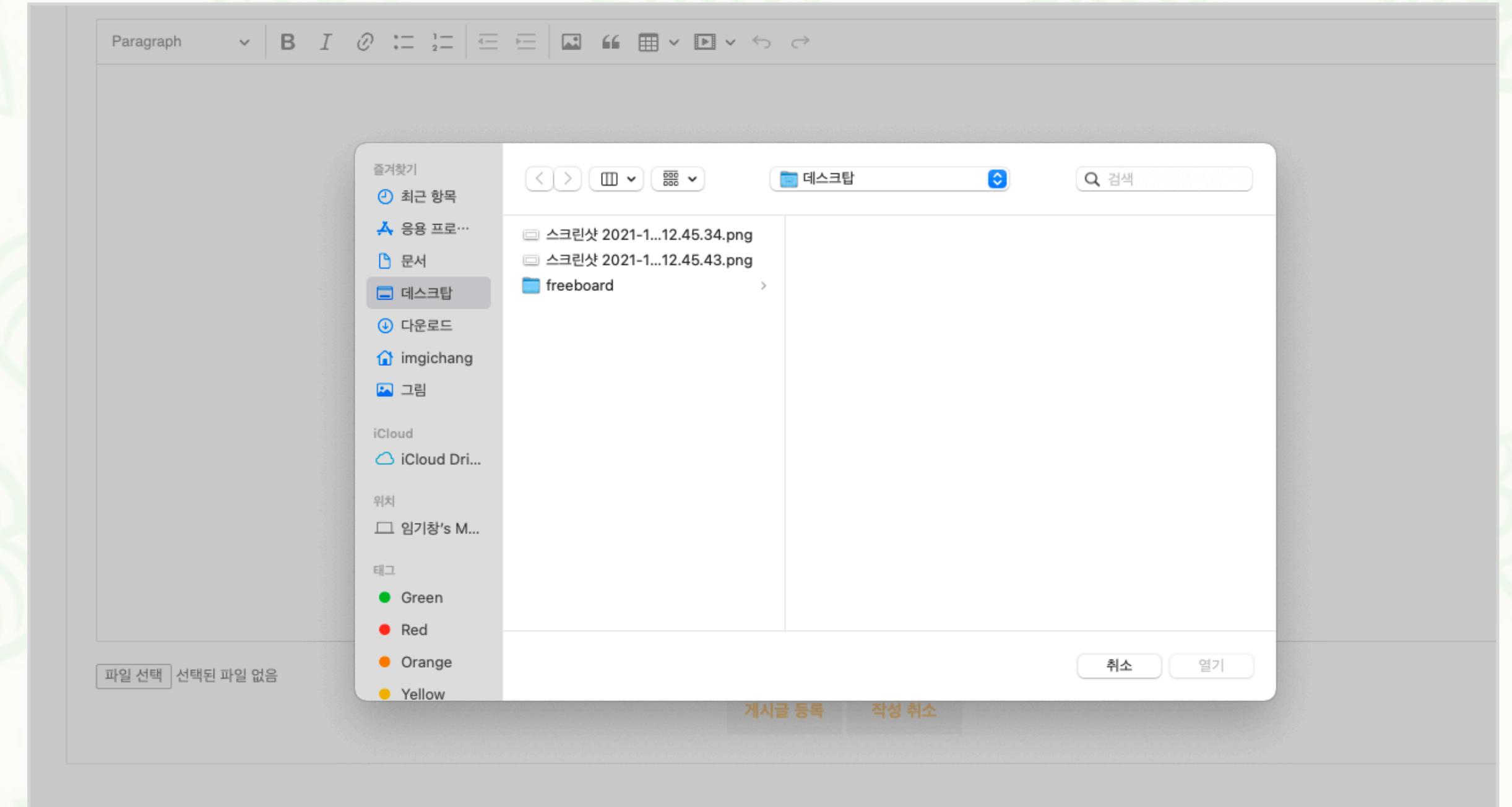
        //에러 페이지
        request.setAttribute("error-msg", "multipart로 전송되지 않았습니다.");
        request.getRequestDispatcher("views/common/errorPage.jsp")
            .forward(request, response);
    }

    // 3. 받아온 파일을 저장할 경로 설정하기
    String root = request.getServletContext().getRealPath("/");
    String savePath = root + "resources/boardUploadFiles";

    // 4. 위에서 설정한 정보를 바탕으로 MultipartRequest 생성하기
    MultipartRequest mre = new MultipartRequest(request, savePath,
        maxSize, "UTF-8",
        new DefaultFileRenamePolicy());

    // 5-1. 파일이 아닌 기본 전송값 처리
    String bwriterId = mre.getParameter("bwriterId");
    String bwriterNick = mre.getParameter("bwriterNick");
    String btitle = mre.getParameter("btitle");
    String bcontent = mre.getParameter("bcontent");

    // 5-2. 파일 저장 및 정보 처리
}
```



07 구현 가능 및 시연.

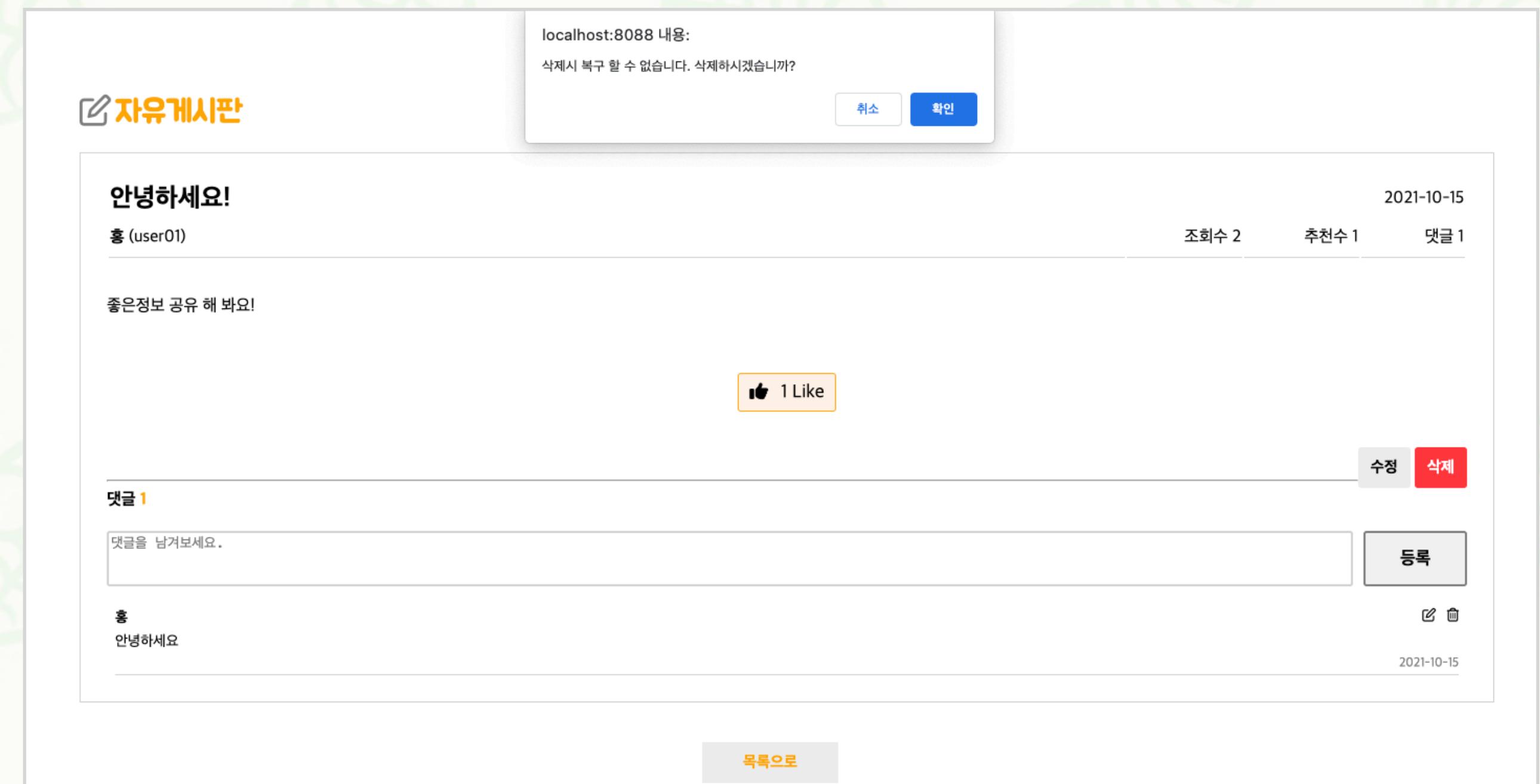
자유마시글 수정

```
/**  
 * Servlet implementation class BoardUpdate  
 */  
  
@WebServlet("/update.fb")  
public class BoardUpdate extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public BoardUpdate() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        BoardService service = new BoardService();  
  
        // 1. JSP에서 multipart로 전송했는지 확인  
        if( ! ServletFileUpload.isMultipartContent(request) ) {  
            request.setAttribute("error-msg", "multipart 전송이 아닙니다!");  
            request.getRequestDispatcher("views/common/errorPage.jsp").forward(request, response);  
        }  
  
        // 2. 전송받을 최대 크기  
        // 10MB  
        int maxSize = 1024 * 1024 * 10;  
  
        // 3. 저장할 폴더 위치  
        String root = request.getServletContext().getRealPath("/");  
        String savePath = root + "resources/boardUploadFiles";  
        // ** 해당 위치에 폴더가 미리 만들어져 있어야 한다!  
  
        // 4. 파일 처리를 위한 multipart 객체 선언  
        MultipartRequest mre = new MultipartRequest(request, savePath, maxSize, "UTF-8")
```

07 구현 기능 및 시연.

자유게시글 삭제

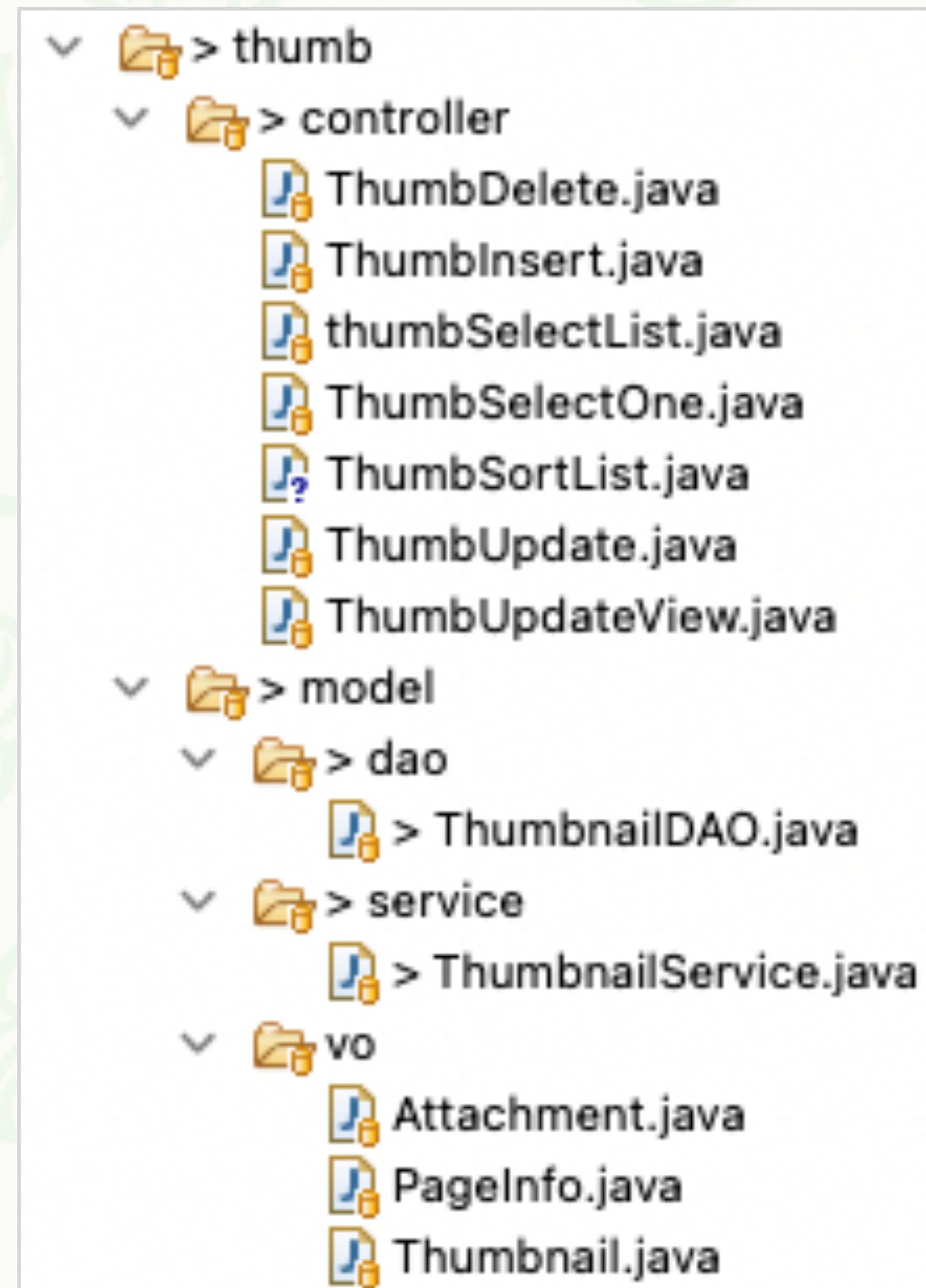
```
/**  
 * Servlet implementation class BoardDelete  
 */  
@WebServlet("/delete.fb")  
public class BoardDelete extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public BoardDelete() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  
        String savePath = request.getServletContext().getRealPath("resources/boardUploadFiles");  
  
        int bno = Integer.parseInt(request.getParameter("bno"));  
        BoardService service = new BoardService();  
        freeBoard fb = service.updateView(bno);  
  
        // 삭제할 게시글의 첨부파일까지 삭제  
        if( fb.getFile() != null ) {  
            new File(savePath + "/" + fb.getFile()).delete();  
        }  
  
        int result = service.deleteBoard(bno);  
        if (result > 0) {  
            response.sendRedirect("list");  
        }  
    }  
}
```



The screenshot shows a web application interface. On the left, there is a code editor window displaying Java code for a servlet named 'BoardDelete'. The code handles a GET request and checks if a file exists at a specific path before deleting it. On the right, there is a preview of a post titled '안녕하세요!' (Hello!) by 'user01' from '2021-10-15'. The post content is '좋은정보 공유 해 봐요!'. Below the post, there is a 'Like' button with '1 Like'. At the top right of the post, there is a small modal dialog with the text 'localhost:8088 내용: 삭제시 복구 할 수 없습니다. 삭제하시겠습니까?' (Content: Deleting will not be recovered. Do you want to delete?). The dialog has two buttons: '취소' (Cancel) and '확인' (Confirm). At the bottom right of the post, there are '수정' (Edit) and '삭제' (Delete) buttons. A '등록' (Register) button is also visible on the far right. A '목록으로' (List) button is located at the bottom center of the preview area.

07 구현 기능 및 시연.

Thumb Class



07 구현 기능 및 시연.

게시글 메인화면

```
<%@ include file="../common/header.jsp" %>

<section>
    <br>
    <div id="Bhead">
        <i id="cateIcon" class="far fa-images"></i>
        <label class="category">나의 댕냥 자랑</label>
        <p id="photoInfo">
            반려동물의 모습을 사진으로 자랑하세요!
        </p>
    </div>

    <div align="right" style="width: 1340px; padding-bottom: 5px;">
        <select name="sortList" id="sortlist">
            <option value="">:::정렬:::</option>
            <option ${param.sort == "new"} ? "selected" : "" value="new">최신순</option>
            <option ${param.sort == "cmt"} ? "selected" : "" value="cmt">댓글순</option>
            <option ${param.sort == "like"} ? "selected" : "" value="like">추천순</option>
        </select>
    </div>

    <hr />
    <div id="thumbnailArea">
        <% for(Thumbnail thumb : list) { %>
            <div class="thumb-list" align="center">
                <div>
                    <input type="hidden" name="bno" value="<%= thumb.getBno() %>" />
                    
                </div>
            </div>
        <% } %>
    </div>
</ul>
```

홍 | 마이페이지 | 로그아웃 |

PETmu

≡ 전체 카테고리 우리동네 동물병원 후기·정보 자유게시판 나의 댕냥 자랑

나의 댕냥 자랑
반려동물의 모습을 사진으로 자랑하세요!

...정렬... ▾

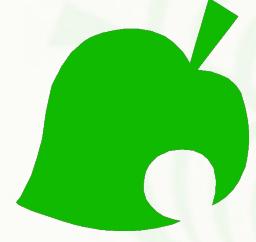
작성하기

« < > »

제목 검색어를 입력해주세요 검색

PETmu

COPYRIGHT © 2021 PETmu. ALL RIGHTS RESERVED. SEMIPROJECT I KH Academy reserved by Yongin Minsu Gichang Jeun



07 구현 가능 및 시연.

제시글 작성

```
/**  
 * Servlet implementation class ThumbInsert  
 */  
@WebServlet("/insert.tn")  
public class ThumbInsert extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public ThumbInsert() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        // 1. multipart 전송 여부 확인  
        if ( ! ServletFileUpload.isMultipartContent(request)) {  
            request.setAttribute("error-msg", "multipart 전송이 아닙니다.");  
            request.getRequestDispatcher("views/common/errorPage.jsp").forward(request, response);  
        }  
  
        // 2. 파일 사이즈 선언  
        int maxSize = 1024 * 1024 * 30;  
  
        // 3. 저장 경로 선언  
        String root = request.getServletContext().getRealPath("/");  
        String savePath = root + "resources/thumbFiles";  
  
        // 4. MultipartRequest 객체 선언  
        MultipartRequest mre = new MultipartRequest(request, savePath, maxSize, "UTF-8"  
                , new MyRenamePolicy());  
  
        // 5. 파일 저장 수행할 객체 선언  
        ArrayList<String> originNames = new ArrayList<>();
```

우리 아이 좀 보세요 ㅠ

Paragraph ▼ B I ② := 1= 2= ≡ ≡ | ▣ “ ” ■ ▼ ▶ ◀ ↪ ↪

너무 이쁘죠?!

대표사진

게시글 등록

작성 취소

07 구현 가능 및 시연.

제시글 수정

```
/**  
 * Servlet implementation class ThumbUpdate  
 */  
@WebServlet("/update.tn")  
public class ThumbUpdate extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public ThumbUpdate() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        if( ! ServletFileUpload.isMultipartContent(request)) {  
            request.setAttribute("error-msg", "multipart 전송이 아닙니다.");  
            request.getRequestDispatcher("views/common/errorPage.jsp").forward(request, response);  
        }  
  
        int maxSize = 1024 * 1024 * 10;  
        String savePath = request.getServletContext().getRealPath("/resources/thumbFiles");  
  
        MultipartRequest mre = new MultipartRequest(request, savePath, maxSize, "UTF-8", new MyRenamePolicy());  
  
        int bno = Integer.parseInt(mre.getParameter("bno"));  
        ThumbnailService service = new ThumbnailService();  
  
        // 원본 게시글 조회  
        HashMap<String, Object> hmap = service.getUpdateView(bno);  
        ArrayList<String> originNames = new ArrayList<>();  
        ArrayList<String> changeNames = new ArrayList<>();  
  
        Enumeration<String> files = mre.getFileNames(); // 전송한 파일들의 태그명
```

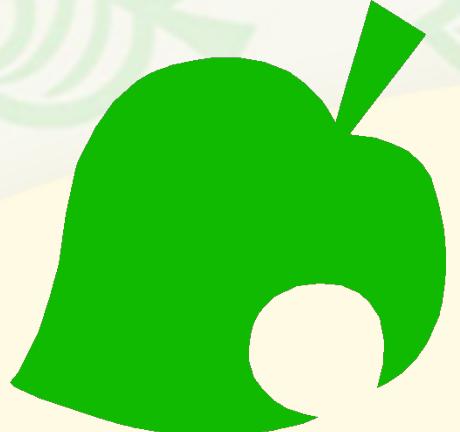
07 구현 기능 및 시연.

게시글 삭제

```
package com.kh.petmu.thumb.controller;  
  
import java.io.IOException;  
  
/**  
 * Servlet implementation class ThumbDelete  
 */  
@WebServlet("/delete.tn")  
public class ThumbDelete extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    /**  
     * @see HttpServlet#HttpServlet()  
     */  
    public ThumbDelete() {  
        super();  
        // TODO Auto-generated constructor stub  
    }  
  
    /**  
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
     */  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
    int bno = Integer.parseInt(request.getParameter("bno"));  
    ThumbnailService service = new ThumbnailService();  
  
    String savePath = request.getServletContext().getRealPath("/resources/thumbFiles");  
  
    int result = service.deleteThumbnail(bno, savePath);  
  
    if( result > 0 ) {  
        response.sendRedirect("selectList.tn");  
    } else {  
        request.setAttribute("error-msg", "사진 게시글 삭제 실패!");  
    }  
}
```

The screenshot shows a web application interface. On the left, there is a code editor window displaying Java code for a servlet named 'ThumbDelete'. The code handles a GET request and checks if a parameter 'bno' is provided. It then calls a 'deleteThumbnail' method from a 'ThumbnailService' object, passing the 'bno' and a save path. If the result is greater than 0, it redirects to 'selectList.tn'. Otherwise, it sets an attribute 'error-msg' with the value '사진 게시글 삭제 실패!'.

On the right, there is a browser window showing a confirmation dialog. The dialog has a title 'localhost:8088 내용:' and a message '삭제시 복구 할 수 없습니다. 삭제하시겠습니까?'. Below the message are two buttons: '취소' (Cancel) and '확인' (Confirm). To the right of the dialog, there is a sidebar with the heading '나의 댕냥 자랑' and the subtext '반려동물의 모습을 사진으로 자랑하세요!'. At the bottom of the sidebar, there is a search bar with the placeholder '검색어를 입력해주세요' and a '검색' button. The overall theme of the application appears to be pet-related, specifically for sharing pictures of pets.



THANK YOU!

