

# Better Together

같이 챌린지하죠  
권세희(팀장) 김완식 김은화  
서재준 윤세홍 최승아

---

**1** 기획 의도

**2** 프로젝트 소개

**3** DataBase

**4** 주요 기능

**5** 개발 환경

---

# 1. 기획 의도

코로나로 인해 외부 활동이 많이 줄어든 요즘같은 시기에 크고 작은 생활습관을 챌린지로 만들어 도전하며 서로 의지를 북돋아 목표를 달성하게 해주기 위한 플랫폼을 제공하고자 하는 프로그램입니다.

## 2. 프로젝트 소개

### **Mission**

같은 목표를 가지고 있는 사람들이 모여 각자 원하는 챌린지 그룹 신청 후 지키고 싶은 나와 의 약속을 돈을 걸어두어 동기부여를 해줍니다.

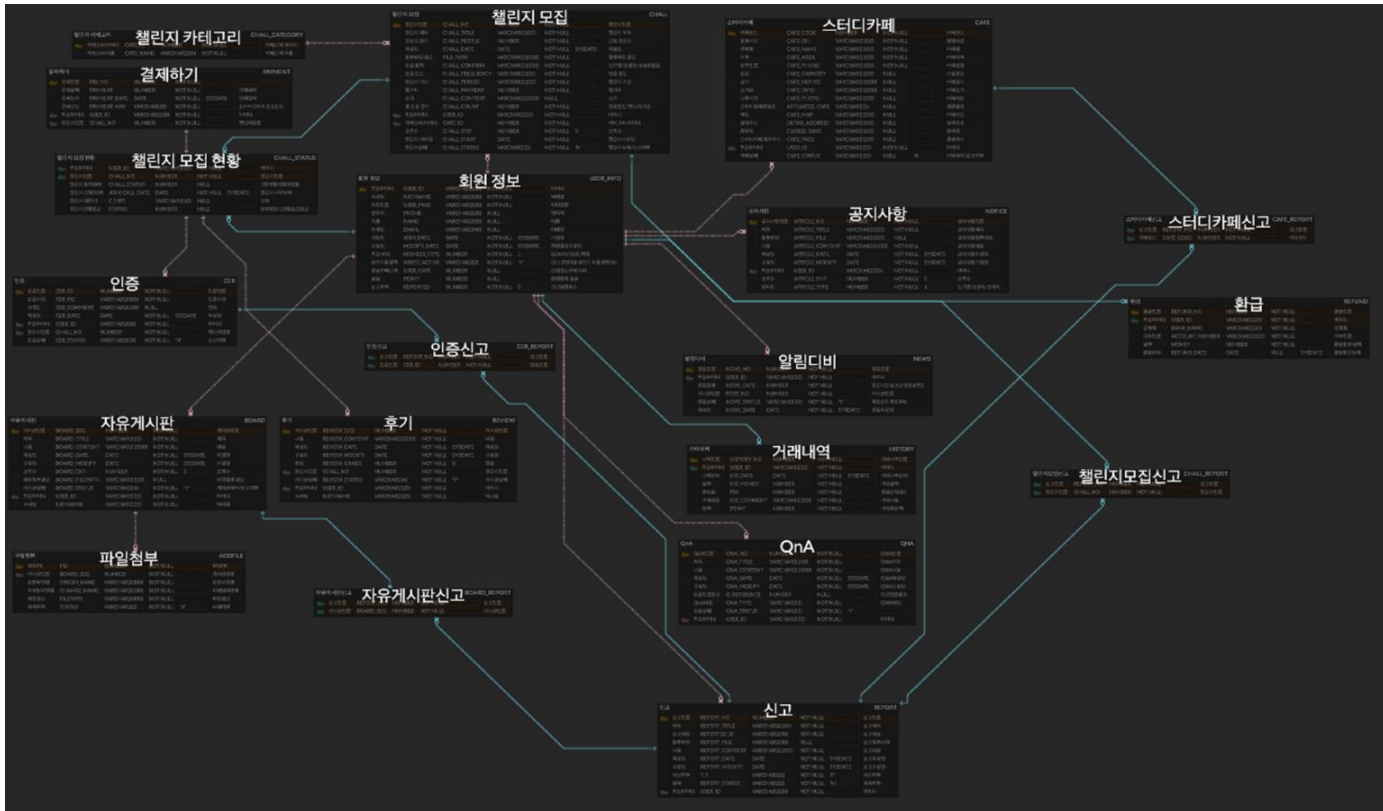
### **Community**

같은 목표를 가진 사람들과 함께 할 수 있어 공동체 의식과 서로의 도전을 응원하는 커뮤니티를 제공합니다.

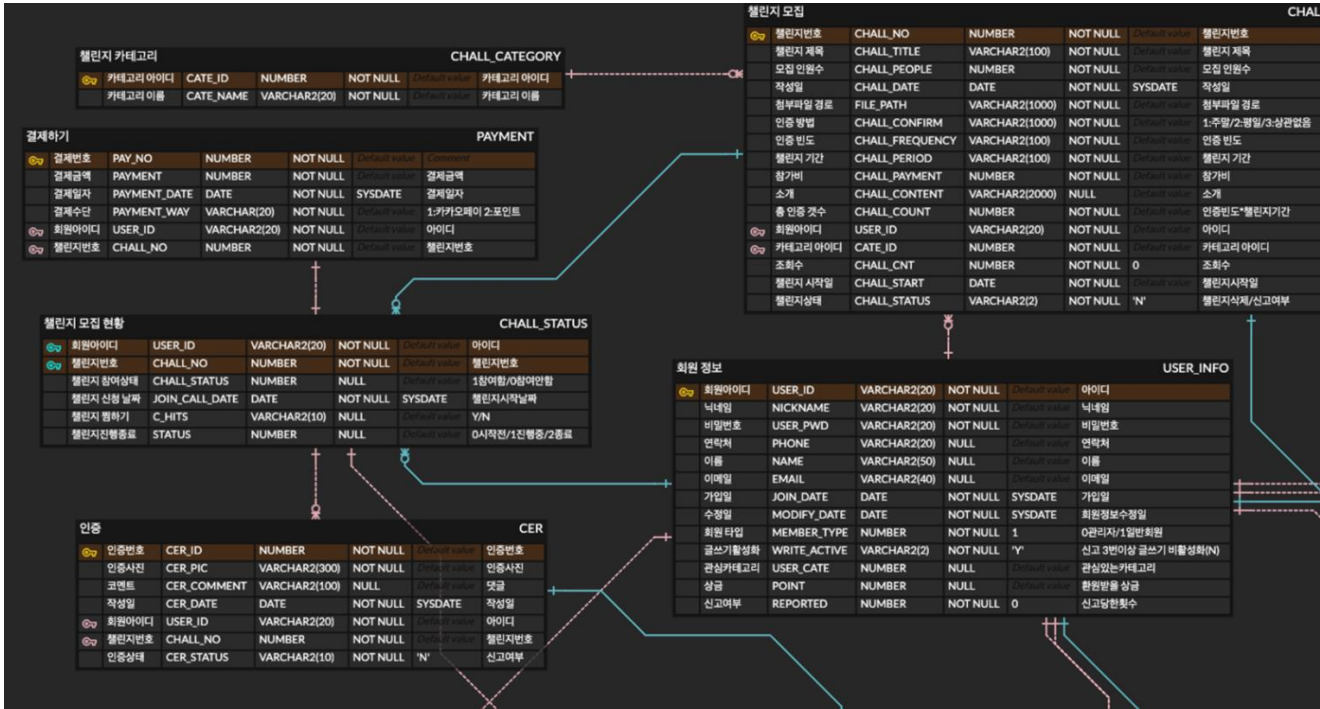
# 3. DataBase

---

# DataBase



# DataBase



챌린지 모집 현황 Table  
회원이 챌린지에 참가 또는 종료된 여부와 챌린지 점하기 여부를 저장하기 위한 디비

인증 Table : 회원이 참가한 챌린지에 한 인증을 저장하기 위한 엔티티이며 추후 달성률을 계산하고 챌린저 top-ten을 계산

# DataBase

신고					REPORT	
신고번호	REPORT_NO	NUMBER	NOT NULL	Default value	신고번호	
제목	REPORT_TITLE	VARCHAR2(100)	NOT NULL	Default value	신고제목	
신고대상	REPORTED_ID	VARCHAR2(30)	NOT NULL	Default value	신고대상	
첨부파일	REPORT_FILE	VARCHAR2(30)	NULL	Default value	신고첨부파일	
내용	REPORT_CONTENT	VARCHAR2(1000)	NOT NULL	Default value	신고내용	
작성일	REPORT_DATE	DATE	NOT NULL	SYSDATE	신고작성일	
수정일	REPORT_MODIFY	DATE	NOT NULL	SYSDATE	신고수정일	
사실여부	T_F	VARCHAR2(2)	NOT NULL	'F'	사실여부	
상태	REPORT_STATUS	VARCHAR2(2)	NOT NULL	'N'	삭제여부	
회원아이디	USER_ID	VARCHAR2(20)	NOT NULL	Default value	아이디	

인증신고		CER_REPORT				
신고번호	REPORT_NO	NUMBER	NOT NULL	Default value	신고번호	
인증번호	CER_ID	NUMBER	NOT NULL	Default value	인증번호	

자유게시판신고		BOARD_REPORT				
신고번호	REPORT_NO	NUMBER	NOT NULL	Default value	신고번호	
게시글번호	BOARD_SEQ	NUMBER	NOT NULL	Default value	게시글번호	

스터디카페신고		CAFE_REPORT				
신고번호	REPORT_NO	NUMBER	NOT NULL	Default value	신고번호	
카페코드	CAFE_CODE	NUMBER	NOT NULL	Default value	카페코드	

챌린지모집신고		CHALL_REPORT				
신고번호	REPORT_NO	NUMBER	NOT NULL	Default value	신고번호	
챌린지번호	CHALL_NO	NUMBER	NOT NULL	Default value	챌린지번호	

각 카테고리별 신고가 들어왔을 때 신고 작성품에 작성된 내용은 신고디비에 insert하고  
각 카테고리별 신고디비에 신고번호와 각 카테고리의 게시물 번호를 insert 하여 신고게시물 확인



## DataBase

거래내역					HISTORY
내역번호	HISTORY_NO	NUMBER	NOT NULL	Default value	거래내역번호
회원아이디	USER_ID	VARCHAR2(20)	NOT NULL	Default value	아이디
내역날짜	HIS_DATE	DATE	NOT NULL	SYSDATE	거래내역날짜
금액	HIS_MONEY	NUMBER	NOT NULL	Default value	거래금액
출입금	PM	NUMBER	NOT NULL	Default value	출금0/입금1
거래내용	HIS_COMMENT	VARCHAR2(100)	NOT NULL	Default value	거래내용
잔액	POINT	NUMBER	NOT NULL	Default value	거래후잔액

포인트로 관련 입출금 내역을 보여주기 위한 디비

포인트로 챌린지 결제, 포인트 충전, 챌린지 삭제로 인한 환불, 챌린지 종료시 환급, 포인트 현금으로 환급에 대한 모든 내용을 insert

포인트로 현금로 환급받을 때 돈을 받을 은행명과 계좌번호 금액 환급날자를 저장하는 디비

환급					REFUND
환급번호	REFUND_NO	NUMBER	NOT NULL	Default value	환불번호
회원아이디	USER_ID	VARCHAR2(20)	NOT NULL	Default value	아이디
은행명	BANK_NAME	VARCHAR2(20)	NOT NULL	Default value	은행명
계좌번호	ACCOUNT_NUMBER	VARCHAR2(100)	NOT NULL	Default value	계좌번호
금액	MONEY	NUMBER	NOT NULL	Default value	환급받을 금액
환불날짜	REFUND_DATE	DATE	NULL	SYSDATE	환불받은날짜

알림디비					NEWS
알림번호	NEWS_NO	NUMBER	NOT NULL	Default value	알림번호
회원아이디	USER_ID	VARCHAR2(20)	NOT NULL	Default value	아이디
알림종류	NEWS_CATE	NUMBER	NOT NULL	Default value	챌린지0/신고1/질문답변2/신고처리3
게시글번호	POST_NO	NUMBER	NOT NULL	Default value	게시글번호
알림상태	NEWS_STATUS	VARCHAR2(10)	NOT NULL	'Y'	확인전Y, 확인후N
작성일	NEWS_DATE	DATE	NOT NULL	SYSDATE	알림작성일

사용자에게 알려줄 내용을 저장하는 디비 알림 종류 컬럼으로 챌린지가 삭제되어 신청이 취소되었는지, 회원이 작성한 게시물이 신고당했는지, 질문에 관리자가 답변을 달았는지, 신고한 게시물이 처리가 되었는지를 알려주는 디비

# 4. 주요기능

---



# Login/Logout Code Review

- 비밀번호 찾기

권세희

# 비밀번호 찾기

```
<%  
if(request.getAttribute("msg")!=null){  
    if(request.getAttribute("msg").equals("success")){  
        String pwd=(String)request.getAttribute("pwd");  
        String email=(String)request.getAttribute("email");  
        String name=(String)request.getAttribute("name");  
    }  
%>  
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/emailjs-com@2/dist/email.min.js"></script>  
<script type="text/javascript">  
    (function() {  
        emailjs.init("user_ngBiYc4zImyZx2AEZwZwU");  
    })();  
</script>  
  
<script type="text/javascript">  
  
(function(){  
    emailjs.send("better_together","template_dajtxjn",{  
        name: '<%=name%>',  
        pwd: '<%=pwd%>',  
        email: '<%=email%>'  
    });  
})();  
alert("비밀번호를 회원님의 이메일로 발송하였습니다.");  
window.close();  
  
</script>  
<%} else{ %>  
<script>  
alert("비밀번호 찾기에 실패하였습니다.");  
</script>  
<% } %>  
}%>
```

이름, 아이디 이메일로 비밀번호를 찾아 jsp로 가져와, emailJs에 만들어 놓은 template에 맞게 파라미터를 setting해 전달하면 회원 가입시 입력한 email로 비밀번호를 전송시켜준다.

## ☆ BetterTogether 비밀번호 찾기

보낸사람 **VIP** BetterTogether<sehee1425@gmail.com>  
받는사람 <ii1425@naver.com>

안녕하세요 Better Together입니다.  
권세희 회원님의 비밀번호는 1234 입니다.  
[인생을 바꾸는 건 결심보다 행동!](#)  
전 국민 목표 달성 프로젝트 Better Together  
많은 관심 부탁드립니다.



# Challenge Join Code Review

- 챌린지 개설 (인증 빈도 선택 사항)
- 챌린지 꺾하기 - 꺾 / 꺾 취소
- 결제 기능 - API / 포인트
- 결제 완료 후 - 참여중으로 +1
- 챌린지 리스트 - 시작전/진행중/종료 챌린지 분류 (날짜계산)
- 챌린지 상세보기 - 시작전/진행중/종료 챌린지 텍스트 구별
- 내가 모집한 챌린지 - 수정 / 삭제 버튼 [시작전에만 보이게]

## 챌린지 개설 (인증 빈도 선택 사항)

```
<script type="text/javascript">
function Activity(name, list){
    this.name = name; // 주말/평일/상관없음 vlaue
    this.list = list; // 해당되는 select option list
}

var acts = new Array(); //배열
acts[0] = new Activity('주말', ['1', '2']);
acts[1] = new Activity('평일', ['1', '2', '3', '4', '5']);
acts[2] = new Activity('상관없음', ['1', '2', '3', '4', '5', '6', '7']);

function updateFreq(str){
    var frm = document.myForm; //입력 폼 name --> myForm
    var oriLen = frm.frequency.length; //select option "name" 가져오기 --> frequency
    var numActs;

    for (var i = 0; i < acts.length; i++){
        if (str == acts[i].name) { //str: 주말/평일/상관없음 1,2,3
            numActs = acts[i].list.length; // 해당되는 select option 길이 가져와서 담기
            for (var j = 0; j < numActs; j++)
                frm.frequency.options[j] = new Option(acts[i].list[j],
                acts[i].list[j]);
            for (var j = numActs; j < oriLen; j++)
                frm.frequency.options[numActs] = null;
        }
    }
}
</script>
```

챌린지 개설시 - 평일, 주말, 상관없음에 따라 인증 빈도 다르게 선택

1. 평일: 최대가 5

2. 주말: 최대가 2

3. 상관없을때 : 최대가 7

평일	주말	상관 없음
<p>인증 방법 <input type="radio"/> 주말 <input checked="" type="radio"/> 평일 <input type="radio"/> 상관없음</p> <p>인증 빈도 <input type="text" value="1"/></p> <p>챌린지 기간 <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> <input type="text" value="4"/> <input type="text" value="5"/></p> <p>챌린지 시작일</p>	<p>인증 방법 <input checked="" type="radio"/> 주말 <input type="radio"/> 평일 <input type="radio"/> 상관없음</p> <p>인증 빈도 <input type="text" value="1"/></p> <p>챌린지 기간 <input type="text" value="1"/> <input type="text" value="2"/></p> <p>챌린지 시작일</p>	<p>인증 방법 <input type="radio"/> 주말 <input type="radio"/> 평일 <input checked="" type="radio"/> 상관없음</p> <p>인증 빈도 <input type="text" value="1"/></p> <p>챌린지 기간 <input type="text" value="1"/> <input type="text" value="2"/> <input type="text" value="3"/> <input type="text" value="4"/> <input type="text" value="5"/> <input type="text" value="6"/> <input type="text" value="7"/></p> <p>챌린지 시작일</p>

## 챌린지 찜하기 - 찜 / 찜 취소

```
// 1. chall_status 테이블에 해당 번호, 유저아이디 컬럼이 있는지 확인 부터 하기
ChallengeStatus cs = new ChallService().selectChallStatus(challNo, userId);
```

```
// hits_status 가져오기 (찜 상태 알아내기위해!!)
String hits_status = new ChallService().selectHits(challNo, userId);
```

```
// chall_status 가져오기 (참여중인지 아닌지 여부)
int chall_status = new ChallService().selectJoinChallStatus(challNo, userId);
```

```
// 찜하기 경우의 수로 나누기 ( 1. 참여:0 찜: Y / 2. 참여:0 찜: N / 3. 참여:1 찜: N / 4. 참여:1 찜: Y
if (cs != null && hits_status.equals("Y") && chall_status == 0) {
    // 참여중은 아닌데 찜만 한 경우
    result = new ChallService().deleteChallHits(challNo, userId);
} else if (cs != null && chall_status == 1 && hits_status.equals("Y")) {
    // 참여중이면서 찜하기가 'Y' -- > 찜하기 취소하고 싶음 (삭제말고! 업데이트로) --> C_hits를 'N'으로 업데이트
    result = new ChallService().updateHitsStatus1(challNo, userId); // 'N'으로
} else if (cs != null && chall_status == 0 && hits_status.equals("N")) {
    result = new ChallService().updateHitsStatus(challNo, userId); // 'Y'로
} else { // cs != null + hits_status -> 'N'
    // 참여중인데 찜하기가 'N' -- > 'Y'로 바꿔주기
    // result = new ChallService().insertChallHits(challNo, userId);
    result = new ChallService().updateHitsStatus(challNo, userId); // 'Y'로
}
}
```

- 찜하기 경우의 수 [챌린지 상태 DB]

1) 챌린지 참여중은 아닌데 찜 취소한 경우-> 찜 DELETE

2) 챌린지 참여중인데 찜을 취소할때-> 찜 'N'으로 UPDATE (DELETE 해버리면 회원이 참여하고 있는 챌린지도 지워지기 때문에 UPDATE)

3) 참여중인데 찜한 경우 -> 찜 'Y'로 UPDATE

찜 취소



찜하기 +1

찜하기



찜하기 +2

# API

# 결제 기능 - API / 포인트

```
$(function(){
var IMP = window.IMP; // 생략가능
IMP.init('imp63477997'); // 'iamport' 대신 부여받은 "가맹점 식별코드"를 사용
```

```
IMP.request_pay({
  pg : 'kakaopay',
  pay_method : 'card',
  merchant_uid : 'merchant_' + new Date().getTime(),
  name : 'Better Together 콜린지 결제',
  amount : <%=ch.getChallPay()%>,
  buyer_email : '<%=email%>',
  buyer_name : '<%=name%>',
  buyer_tel : '<%=phone%>',
  buyer_addr : '',
  buyer_postcode : '123-456',
  //m_redirect_url : 'http://www.naver.com'
}, function(rsp) {
  console.log(rsp);
  if (rsp.success) {
    var msg = '결제가 완료되었습니다.';
    msg += '\n고유ID : ' + rsp.imp_uid;
    msg += '\n상점 거래ID : ' + rsp.merchant_uid;
    msg += '\n결제 금액 : ' + rsp.paid_amount;
    msg += '\n카드 승인번호 : ' + rsp.apply_num;
    location.href = '<%=request.getContextPath()%>/pay/insert?challNo=<%= ch.getChallNo()>';
  } else {
    var msg = '결제가 실패하였습니다.';
    msg += '에러내용 : ' + rsp.error_msg;
    //실패시 다시 페이지로 이동
    location.href = '<%= request.getContextPath()%>/commu/main";
  }
}
```

```
<script>
const challenge_btn = document.getElementById('challenge_btn');
challenge_btn.addEventListener('click',function(){
var radioVal = $('#input[name="payment"] :checked').val();
console.log(radioVal);
var url = '<%=request.getContextPath()%>/order/paySuccess?challNo=<%=ch.getChallNo()%>$payment=2';
// 팝업 가운데에 띄우기
var popupWidth = 550;
var popupHeight = 300;
var popupX = (window.screen.width / 2) - (popupWidth / 2);
// 만큼 팝업창 width 크기의 1/2 만큼 보정값으로 해주어야함
var popupY = (window.screen.height / 2) - (popupHeight / 2);
// 만큼 팝업창 height 크기의 1/2 만큼 보정값으로 해주어야함
if(radioVal == 1){
$('#challNoForm').attr("action", "<%=request.getContextPath()%>/order/paySuccess");
$('#challNoForm').submit();
} else if(radioVal == 2) {
//window.open(url, "포인트 결제", "width=350, height=300, scrollbars=no");
window.open(url, "카드 결제", "status=no, height=" + popupHeight + ", width=" + popupWidth + ", "<%=request.getContextPath()%>");
$('#challNoForm').submit();
}
});
```

## 포인트

## 결제 수단

카카오페이  포인트 결제

## -결제 API (카카오페이)

- 1) 결제 정보를 나타냄 (회원 정보 및 결제 금액)
- 2) 결제 성공 시 뜨는 alert 내용과 성공시 이동하는 페이지
- 3) 결제 실패 시 이동하는 페이지

## -포인트 결제 (팝업창)

- 1) 회원 보유 포인트로 결제 후 회원 디비 포인트 컬럼 업데이트 해줌

## API

## 포인트



emi\_project/order/paySuccess?challNo=146&payer

### 포인트 결제

보유 금액	9810000 포인트
결제할 금액	10000 원
남은 금액	9800000 원
비밀번호 확인	<input type="text"/>



## 결제 완료 후 - 참여중으로 +1

```
// 1: 카카오페이인지, 2: 회원보유포인트인지 판단
int payment = Integer.parseInt(request.getParameter("payment"));

int result = 0;
// 1. chall_status 테이블에 해당 번호, 유저아이디 컬럼이 있는지 확인 부터 하기
ChallengeStatus cs = new ChallService().selectChallStatus(challNo, userId);

// 결제 후 금액 (포인트 결제시)
// payment == 2 일 경우 남은 point로 user_info 업데이트 해주기
int point = 0;
Challenge ch = null;
if(payment == 2) {
    point = Integer.parseInt(request.getParameter("point"));
    Member updateMember = new MemberService().updatePoint(point, loginUser);

    //로그인 세션 값 변경
    request.getSession().setAttribute("loginUser", updateMember);

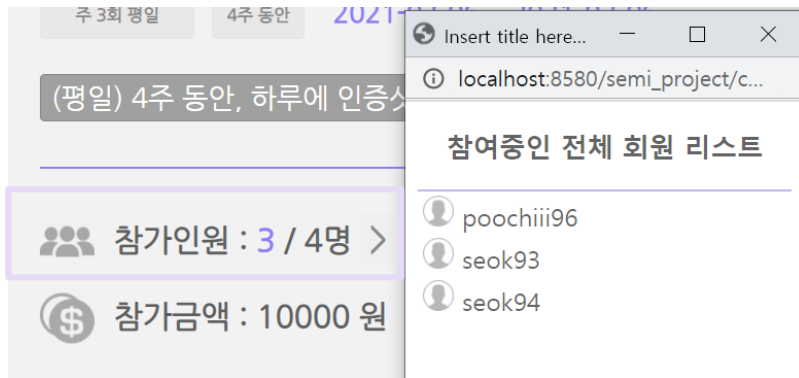
    // success시에 같이 넘겨주기 위해
    ch = new ChallService().selectChall(challNo);
}

if(cs == null) {
    result = new ChallService().insertChallStatus(challNo,userId);
    // 챌린지 모집현황 인원 insert (참여중)
    result = new ChallService().updateChallStatus(challNo, userId);
} else {
    // 챌린지 모집현황 인원 insert (참여중)
    result = new ChallService().updateChallStatus(challNo, userId);
}
```

1)결제 완료 후 챌린지 현황 디비에 참여중으로 업데이트 해주고 참가인원 옆에 < 버튼 클릭시 팝업창이 뜨면 아이디 추가 되어있음

2)중복 참여 못함 (참가하기 버튼 클릭 시 alert창뜸)

3) 참가인원 만원일 경우 참가 못함



## 챌린지 리스트 - 시작전/진행중/종료 챌린지 분류 (날짜계산)

### 시작전

```
        , CHALL_START
        , CHALL_STATUS
    FROM CHALL
    JOIN CHALL_CATEGORY USING(CATE_ID)
    JOIN USER_INFO USING (USER_ID)
    WHERE
        SYSDATE << CHALL_START
        AND CHALL_STATUS = 'N'
    ORDER BY CHALL_NO DESC)T)
RNUM BETWEEN ? AND ?
```

### 진행중

```
        , CHALL_START
        , CHALL_STATUS
    FROM CHALL
    JOIN CHALL_CATEGORY USING(CATE_ID)
    JOIN USER_INFO USING (USER_ID)
    WHERE
        SYSDATE BETWEEN CHALL_START AND CHALL_START + (TO_NUMBER(CHALL_PERIOD)*7)
        AND CHALL_STATUS = 'N'
    ORDER BY CHALL_NO DESC)T)
BETWEEN ? AND ?
```

### 종료

```
        , CHALL_START
        , CHALL_STATUS
    FROM CHALL
    JOIN CHALL_CATEGORY USING(CATE_ID)
    JOIN USER_INFO USING (USER_ID)
    WHERE
        CHALL_START + (TO_NUMBER(CHALL_PERIOD)*7) <= SYSDATE
        AND CHALL_STATUS = 'N'
    ORDER BY CHALL_NO DESC)T)
BETWEEN ? AND ?
```

1) 챌린지 시작날짜를 기준으로 퀴리문에 날짜 계산식 추가해서 시작전/진행중/종료 챌린지 분류함

2) 시작전에만 신고버튼이 보이고 시작일까지 카운트다운으로 나타냄

3) 진행중일때는 챌린지 참가 못함 (진행중일때 인증보기 버튼 보임)

4) 종료일때는 참가하기 버튼 비활성화 함

시작전

진행중

종료

# 챌린지 상세보기 - 시작전/진행중/종료 챌린지 텍스트 구별

```
// 오늘 날짜 구하기
Date date= new Date();
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
String today = sdf.format(date);
```

```
//진행중 : to < today < to + Integer.parseInt(ch.getChallPeriod()*7)
//종료 : today >= to + Integer.parseInt(ch.getChallPeriod()*7
```

```
//시작날짜 + 기간
DateFormat df = new SimpleDateFormat("yyyy-MM-dd");
Calendar cal = Calendar.getInstance();
cal.setTime(from); // 챌린지 시작일로 세팅
cal.add(Calendar.DATE, Integer.parseInt(ch.getChallPeriod()*7); // 챌린지 시작일 + 기간 세팅
String period = df.format(cal.getTime()); // 챌린지 시작일 + 기간
System.out.println("period(챌린지 끝나는 날): " + period);
```

```
// 시작전 : sysdate < chall_start
int compare1 = today.compareTo(startDate);
// compare1 < 0

// 진행중 : startDate < today(sysdate) < period(챌린지 끝나는 날)
int compare = startDate.compareTo(today);
int compare2 = today.compareTo(period);
//compare < 0 && compare2 < 0

// 종료 : period(챌린지 끝나는 날) <= today(sysdate)
int compare3= period.compareTo(today);
// compare3 < 0
```

```
<script>
// 챌린지 시작일 세팅
var countDownDate = new Date("<%=month%> <%=day%> <%=year%> 00:00:00").getTime();
```

```
// 1초마다 카운트
var x = setInterval(function() {
// 현재 날짜/시간 가져오기
var now = new Date().getTime();
// 챌린지 시작일 - 현재 날짜
var distance = countDownDate - now;

// 일,시,분,초 계산
var days = Math.floor(distance / (1000 * 60 * 60 * 24));
var hours = Math.floor((distance % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
var minutes = Math.floor((distance % (1000 * 60 * 60)) / (1000 * 60));
var seconds = Math.floor((distance % (1000 * 60)) / 1000);

// 화면에 출력
document.getElementById("countDown").innerHTML = days + "일 " + hours + "시간 "
+ minutes + "분 " + seconds + "초 ";

// 카운트 진행중 & 종료 문구 바꾸기
```

1) sysdate & 챌린지 시작일 포맷

2) 챌린지 시작일 + 기간 = 끝나는날 세팅

3) compareTo 함수를 이용하여 판별문으로 텍스트 구별함

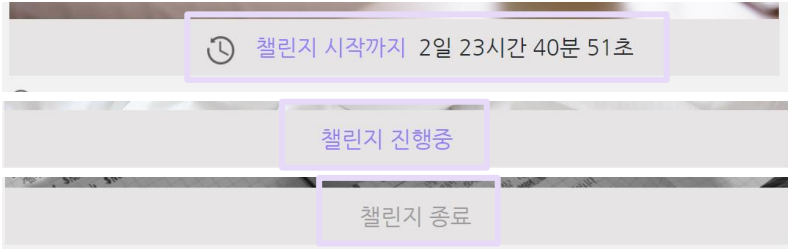
- 시작전: 카운트다운
- 진행중: 챌린지 진행중
- 종료: 챌린지 종료

카운트다운

시작전

진행중

종료



## 내가 모집한 챌린지 - 수정 / 삭제 버튼 [시작전에만 보이게]

```
//오늘 날짜 구하기
Date date= new Date();
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
String today = sdf.format(date);
```

```
<%
    Date from = ch.getChallStart();
    SimpleDateFormat transFormat = new SimpleDateFormat("yyyy-MM-dd");
    String startDate = transFormat.format(from);
%>
<% int compare = today.compareTo(startDate);%>
<!-- 시작전 챌린지만 보이게 -->
<%if(!(compare < 0)) {%>
    <button id="updateBtn" disabled>수정</button>
    <button id="deleteBtn" disabled>삭제</button>
```

1) 내가 모집한 챌린지 수정 / 삭제버튼은 시작전에만 수정 / 삭제 가능함

- 시작날짜를 받아와서 포맷해주고 sysdate와 비교하여 판별문 넣음

진행중 / 종료 - 비활성화

주 1회

3주 동안

수정

삭제

시작전 - 활성화

주 4회

2주 동안

수정

삭제



# Study Cafe Code Review

- 스테디카페 등록
- 제휴 선택 시 결제
- 스테디 카페 수정 / 삭제
- Kako Maps, Postcodify 등사용

김완식

## 카페등록시 일반과 제휴선택

```
String cafe_photo3 = multiRequest.getFilesystemName("cafe_photo3");  
  
Cafe n = new Cafe(cafe_oh,cafe_name,cafe_area,cafe_phone, cafe_capacity ,cafe_notice,cafe_info,  
  
System.out.println("cafe_map:" +cafe_photo2);
```

```
int result = new CafeService().insertCafe(n) ;
```

```
if(AFFILIATED_CAFE.equals("Y")) {  
    request.setAttribute("cafe_name", cafe_name);  
    request.setAttribute("cafe_phone", cafe_phone);  
  
    request.getRequestDispatcher("/views/studycafe/cafeAPIPay.jsp").forward(request, response);
```

```
}else {
```

```
if(result > 0) {
```

```
    response.sendRedirect(request.getContextPath() + "/cafe/list");  
} else {  
    request.setAttribute("msg", "카페 등록에 실패하였습니다.");  
    request.getRequestDispatcher("/views/common/errorPage.jsp").forward(request, response);  
}
```

제휴정보

일반 ▼ 제휴약관

일반  
제휴

취소 등록

1. 제휴로 선택해도 결제전에 우선 디비에 저장

2. 제휴의 벨류값이 Y이면 결제창으로 이동(네임값과 번호값을 포함하여)

3. 일반카페이면 바로 그냥 카페 리스트출력화면으로 이동

## 카페결제 폼으로 이동

```
<%@page language="java" contentType="text/html; charset=UTF-8"%>
pageEncoding="UTF-8"
import="java.util.ArrayList , studycafe.model.vo.*, member.model.vo.Member"%>
```

```
String cafe_name = (String.valueOf(request.getAttribute("cafe_name")));
String cafe_phone = (String.valueOf(request.getAttribute("cafe_phone")));
```

```
//System.out.println("cafe_name1 : " + cafe_name);
//System.out.println("cafe_phone1 : " + cafe_phone);
```

```
Member m = (Member)session.getAttribute("loginUser");
```

```
}, function(rsp) {
    console.log(rsp);
    if (rsp.success) {
        var msg = '결제가 완료되었습니다.';
        msg += '\n고유ID : ' + rsp.imp_uid;
        msg += '\n상점 거래ID : ' + rsp.merchant_uid;
        msg += '\n결제 금액 : ' + rsp.paid_amount;
        msg += '\n카드 승인번호 : ' + rsp.apply_num;
        location.href="<%=request.getContextPath()%>/cafe/list";
```

```
} else {
    var msg = '결제에 실패하였습니다.';
    msg += '에러내용 : ' + rsp.error_msg;
    //실패시 이동할 페이지
```

```
location.href="<%=request.getContextPath()%>
/CafeDeleteServlet2?cafe_name=<%=cafe_name%>&cafe_phone=<%= cafe_phone %>";
```

```
alert(msg);
});
```

1.결제폼에 받아온 네임값과 번호값을 스트링에 저장

2.정상적으로 완료되면 카페리스트출력화면으로 이동

3.결제취소시 받아온 네임값과 번호값을 딜리트 서블릿으로 넘겨줌

## 결제 취소(실패)시 카페딜리트 서블릿

```
//결제 취소시 딜리트카페서블릿
```

```
String cafe_name = request.getParameter("cafe_name");  
String cafe_phone = request.getParameter("cafe_phone");
```

```
System.out.println(cafe_phone);
```

```
//결제 취소시 삭제
```

```
int result = new CafeService().deleteCafe2(cafe_name,cafe_phone);
```

```
if(result > 0) {
```

```
    response.sendRedirect(request.getContextPath() + "/cafe/list");
```

```
} else {
```

```
    request.getRequestDispatcher("/views/common/errorPage.jsp").forward(request, response);
```

```
    request.setAttribute("msg", "계시글 삭제에 실패했습니다.");
```

```
}
```

```
}
```

```
/**
```

```
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
```

```
 */
```

1. 받아온 카페네임과 번호값을 이용해서 방금 등록된 카페정보를 삭제

2. 정상적으로 삭제가 되면 카페리스트 화면으로 이동

3. 중복카페이름이 있을시 삭제를 방지하기 위해 번호값도 같이 사용



# 카페 수정시 이미지 null 값 해결

## 1. 카페 수정폼

```
<form action="<%=request.getContextPath()%>/cafe/update"
method="post" enctype="multipart/form-data"> |
<input type="hidden" name="Cafe_code" value="<%=c.getCafe_code()%>">
<%
if (c.getCafe_photo() != null) {
%>
<input type="hidden" name="Cafe_photo"
value="<%=c.getCafe_photo()%>">
<%
}
%>
```

1. 수정전 카페의 이미지 값이 있으면 히든값으로 보냄

## 2. 카페 수정 서블렛

```
String Cafe_photo = multiRequest.getParameter("Cafe_photo"); //히든값으로 이미지 값
String Cafe_photo2 = multiRequest.getParameter("Cafe_photo2");
String Cafe_photo3 = multiRequest.getParameter("Cafe_photo3");
System.out.println("Cafe_photo" + " " +Cafe_photo); // 새로운 수정이미지 파일이 없으면
```

2. 업데이트서블렛에 히든값으로 받아온 카페 이미지정보를 담는다

```
cafe.setCafe_phone(multiRequest.getParameter("cafe_phone"));
cafe.setCafe_capacity(multiRequest.getParameter("cafe_capacity"));
cafe.setCafe_notice(multiRequest.getParameter("cafe_notice"));
cafe.setCafe_info(multiRequest.getParameter("cafe_info"));
```

```
if(multiRequest.getFilesystemName("cafe_photo") == null) {
    cafe.setCafe_photo(Cafe_photo);
} else {
    cafe.setCafe_photo(multiRequest.getFilesystemName("cafe_photo"));
}
```

3. 만약 새로 추가된 카페이미지가 없으면 히든값으로 받은 이미지정보값을 다시 담는다.

```
cafe.setAFFILIATED_CAFE(multiRequest.getParameter("AFFILIATED_CAFE"));
cafe.setCafe_map(multiRequest.getParameter("cafe_map"));
cafe.setDetail_address(multiRequest.getParameter("detail_address"));
cafe.setClosed_day(multiRequest.getParameter("closed_day"));
cafe.setCafe_page(multiRequest.getParameter("cafe_page"));
cafe.setCafe_code(Cafe_code);
```

# Postcodify와 kakao Maps 연동

## 1. 검색 api

카페홈페이지

도로명주소

상세주소

대표이미지  선택된 파일 없음

추가이미지  선택된 파일 없음

## 2. 카카오맵 api

```
var map = new kakao.maps.Map(mapContainer, mapOption);
```

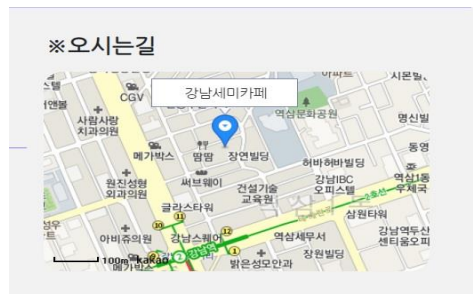
```
var geocoder = new kakao.maps.services.Geocoder();
```

```
geocoder.  
.addressSearch(  
  '<%= c.getCafe_map() %>',  
  function(result, status) {  
  
    if (status === kakao.maps.services.Status.OK) {  
  
      var coords = new kakao.maps.LatLng(  
        result[0].y, result[0].x);  
  
      var marker = new kakao.maps.Marker(  
        {  
          map : map,  
          position : coords  
        });  
  
      var infowindow = new kakao.maps.InfoWindow(  
        {  
          content :  
            '<div style="width:150px;text-align:center;padding:6px 0;"><%= c.getCafe_name() %></div>',  
        });  
      infowindow.open(map, marker);  
  
      map.setCenter(coords);  
    }  
  });
```

1. 카페 위치를 도로명 주소로 받아 cafe\_map에 저장

2. 카카오 맵 api에 도로명주소 입력하는곳에 저장된 cafe\_map값을 저장.

3. 마커이름에 표시될 부분에 카페이름 저장





# Community Code Review

- 메인 페이지 - 인기 챌린지 슬라이드
- 메인 페이지 - 챌린저 **TOP10**
- 자유게시판
- 후기

윤세홍

## 인기 챌린지 - query

```
-- 랭킹뷰 만들기
CREATE OR REPLACE VIEW RANKLIST
AS SELECT ROWNUM RNUM, B.*
FROM
  (SELECT RANK()OVER(ORDER BY CNT DESC)RANKING,
    CHALL_NO,
    D.CNT,
    CHALL_TITLE,
    FILE_PATH,
    CHALL_FREQUENCY,
    CHALL_PERIOD,
    CHALL_START,
    CHALL_STATUS
  FROM
    (SELECT A.CNT,
      CHALL_NO,
      CHALL_TITLE,
      FILE_PATH,
      CHALL_FREQUENCY,
      CHALL_PERIOD,
      CHALL_START,
      C.CHALL_STATUS
    FROM
      (SELECT CHALL_NO, COUNT(*) CNT
      FROM CHALL_STATUS
      WHERE C_HITS = 'Y'
      GROUP BY CHALL_NO)A
    JOIN CHALL C USING(CHALL_NO)
    WHERE CHALL_STATUS = 'N')D)B;
```

1. 챌린지 상태 테이블 -> 째 상태 'Y'
2. 챌린지 테이블 -> 챌린지 정보
3. JOIN 후 째 갯수 카운트
4. 카운트 랭크순으로 정렬 VIEW 생성
5. ROWNUM 1 ~ 18 까지 슬라이드에 올리는 작업

## 챌린저 TOP10

```
-- TOPTEN LIST
CREATE OR REPLACE VIEW TOPTENLIST
AS SELECT RANK()OVER(ORDER BY A.CNT DESC)RANKING, A.CNT, USER_ID
FROM
    (SELECT COUNT(*) CNT, USER_ID
    FROM CER
    WHERE CER_STATUS = 'N'
    GROUP BY USER_ID
    ORDER BY COUNT(*) DESC)A;
```

1. 로그인 유저의 인증 갯수 카운트
2. 카운트 랭크순으로 정렬 VIEW 생성
3. ROWNUM 1 ~ 10까지 출력 후

세로 무한 슬라이드

```
@keyframes slide {
  0% {margin-top: 0;}
  5%{margin-top:0;}
  10%{margin-top:-30px;}
  15%{margin-top:-30px;}
  20%{margin-top:-60px;}
  25%{margin-top:-60px;}
  30%{margin-top:-90px;}
  35%{margin-top:-90px;}
  40%{margin-top:-120px;}
  45%{margin-top:-120px;}
  50%{margin-top:-150px;}
  55%{margin-top:-150px;}
  60%{margin-top:-180px;}
  65%{margin-top:-180px;}
  70%{margin-top:-210px;}
  75%{margin-top:-210px;}
  80%{margin-top:-240px;}
  85%{margin-top:-240px;}
  90%{margin-top:-270px;}
  95%{margin-top:-270px;}
  100%{margin-top:0;}
}
```

## 내가 쓴 글 리스트 따로 보기

```
<entry key="selectMyBoardList">
  SELECT
    RNUM,
    BOARD_SEQ,
    BOARD_TITLE,
    BOARD_CONTENT,
    BOARD_DATE,
    BOARD_MODIFY,
    BOARD_CNT,
    BOARD_FILEPATH,
    BOARD_STATUS,
    USER_ID,
    NICKNAME
  FROM (SELECT ROWNUM RNUM, A.*
        FROM
          (SELECT
            BOARD_SEQ,
            BOARD_TITLE,
            BOARD_CONTENT,
            BOARD_DATE,
            BOARD_MODIFY,
            BOARD_CNT,
            BOARD_FILEPATH,
            BOARD_STATUS,
            USER_ID,
            NICKNAME
          FROM BOARD
          WHERE BOARD_STATUS = 'Y'
          ORDER BY BOARD_SEQ DESC)A
        WHERE USER ID = ?)
  WHERE RNUM BETWEEN ? AND ?
</entry>
```

내가 쓴 글 보기

Button

내가 쓴 글만 모아 볼 수 있는 별도의 페이지에서  
수정 / 삭제 가능하도록 편리하게 제공하고자 함  
페이징 처리를 위한 ROWNUM과  
USER\_ID 조건 추가 하여 SELECT

## 후기 메인 - 별점순, 최신순 정렬 출력

별점순▼

최신순▼

Button

```
<!-- 후기 목록 -->
<div class="reviewWrap">
  <% if(sortByGradeList != null) { %>
  <% for(Review r : sortByGradeList) { %>
  <div class="reviewdiv">
    <table class="reviewTb">
      <tr class="firstTr">
        <td class="nickname"><%= r.getNickname() %></td>
        <td class="challName"><%= r.getChallTitle() %></td>
        <td class="starArea"><%if(r.getrGrade() == 5) { %> *****
          <% } else if(r.getrGrade() == 4) { %>*****
          <% } else if(r.getrGrade() == 3) { %>***
          <% } else if(r.getrGrade() == 2) { %>**
          <% } else { %>*<% } %>
        </td>
        <td class="createDate"><%= r.getCreateDate() %></td>
      </tr>
      <tr class="secondTr">
        <td class="contentArea"><textarea readonly cols="116"><%= r.getrContent() %></textarea></td>
      </tr>
    </table>
  </div>
  <% } %>
  <% } %>
</div>
```

## 후기 등록(1) - 내가 참여했던 챌린지 선택 & 별점 매기기

```
<div id="challSelectArea">
  <select id="challSelect">
    <% if(cList != null) { %>
      <% for(Challenge c : cList) {%>
        <option value=<%= c.getChallNo() %>><%= c.getChallTitle() %></option>
      <% } %>
    <% } %>
  </select>
</div>
```

챌린지의 시퀀스 값 넘기고 챌린지의 제목 출력

```
<div id="starArea">
  <p id="star">
    <a href="javascript:void(0)" value=1>★</a>
    <a href="javascript:void(0)" value=2>★</a>
    <a href="javascript:void(0)" value=3>★</a>
    <a href="javascript:void(0)" value=4>★</a>
    <a href="javascript:void(0)" value=5>★</a>
  </p>
</div>
```

일주일에 6번 5km 러닝 뛰기 ▾





## 후기 등록(2) - 내용 작성 글자수 제한 이벤트

```
<!-- 글자수 동적 카운팅 -->
<script>
$('#reviewWriteArea').keyup(function (e){
  var content = $(this).val();
  $('#counter').html("(" + content.length + " / 300)");

  if(content.length > 300){
    alert('최대 300자까지 입력 가능합니다. ');
    $(this).val(content.substring(0, 300));
    $('#counter').html("(300 / 300)");
  }
});
</script>
```

300자 글자 수 제한

텍스트 입력시 글자 수 동적 카운팅

300자 넘을 시 alert창 띄운 후

초과된 내용 잘리도록 구현

(57 / 300)

최대 300자까지 입력 가능합니다.

확인

## 후기 등록(3) - Ajax 통신

```
success : function(data){

    // 갱신 된 rList를 후기 div에 적용
    reviewWrap = $(".reviewWrap");
    reviewWrap.html(""); // 기존 리뷰 정보 초기화

    // 새로 받아온 갱신된 댓글 리스트들을 for문을 통해 다시 table에 추가
    for(var key in data){

        var reviewdiv = $("<div class='reviewdiv'></div>");
        var reviewTb = $("<table class='reviewTb'></table>");
        var firstTr = $("<tr class='firstTr'></tr>");
        var secondTr = $("<tr class='secondTr'></tr>");
        var writerTd = $("<td class='nickname'></td>").text(data[key].nickName);
        var challTitleTd = $("<td class='challName'></td>").text(data[key].challTitle);
        var gradeTd;
        if(data[key].rGrade == 5){
            gradeTd = $("<td class='starArea'></td>").text("*****");
        }else if(data[key].rGrade == 4){
            gradeTd = $("<td class='starArea'></td>").text("****");
        }else if(data[key].rGrade == 3){
            gradeTd = $("<td class='starArea'></td>").text("***");
        }else if(data[key].rGrade == 2){
            gradeTd = $("<td class='starArea'></td>").text("**");
        }else {
            gradeTd = $("<td class='starArea'></td>").text("*");
        }
        var createDateTd = $("<td class='createDate'></td>").text(data[key].createDate);
        var contentTd = $("<td class='contentArea'><textarea readonly cols='116'></textarea></td>").text(data[key].rContent);

        firstTr.append(writerTd, challTitleTd, gradeTd, createDateTd);
        secondTr.append(contentTd);
        reviewTb.append(firstTr, secondTr);
        reviewdiv.append(reviewTb);
        reviewWrap.append(reviewdiv);
    }
}
```

## 후기 수정 - Ajax

```
<script>
$(function(){
    $("#submit").click(function(){
        var rId = <%= r.getId() %>;
        var content = $("#contentArea").val();

        $.ajax({
            url : "<%= request.getContextPath() %>/review/update",
            type : "post",
            dataType : "json",
            data : {rId : rId, content : content, star : star},
            success : function(data){
                window.opener.document.location.reload();
                window.close();
            },
            error : function(e){
                console.log(e);
            }
        });
    });
});
</script>
```

팝업 창을 띄우고 후기 수정 내용 Ajax로

통신 후 부모창 새로고침 및 팝업 닫기



# Certification Code Review

- 인증하기
- 달성률 계산
- Payback

## 인증하기

```
Calendar calendar = Calendar.getInstance();  
int day1 = calendar.get(Calendar.DAY_OF_WEEK);
```

오늘 날짜에 맞는 요일->1~7(일~토)로 받아  
오기

```
<!-- 요일에 맞게 버튼 활성화 -->  
<%if(confirm.equals("상관없음")||confirm.equals("평일")&&(day1<=6)&&(day1>1)||confirm.equals("주말")&&(day1==1||day1==7)){ %>  
<button type="submit" class="comment">인증하기</button>  
<%} else if(confirm.equals("평일")&&(day1==1||day1==7)){%>  
<button type="button" class="comment" onClick="alert('해당 챌린지의 인증은 평일만 가능합니다.')">인증하기</button>  
<%} else if(confirm.equals("주말")&&(day1<=6)&&(day1>1)){%>  
<button type="button" class="comment" onClick="alert('해당 챌린지의 인증은 주말만 가능합니다.')">인증하기</button>  
<%} %>
```

평일만 인증하는 챌린지인데 day1이 1또는 7이면 주말이므로 경고창을 띄우줌

주말만 인증하는 챌린지인데 day1이 2와 6 사이면 평일이므로 경고창을 띄워줌

localhost:8580 내용:

해당 챌린지의 인증은 평일만 가능합니다.

확인

## 달성률 계산하기

```
SELECT COUNT(DISTINCT TO_CHAR(CER_DATE, 'YYYY-MM-DD')), CHALL_COUNT, COUNT(DISTINCT TO_CHAR(CER_DATE, 'YYYY-MM-DD'))/CHALL_COUNT AS 달성률
FROM CER
JOIN CHALL USING(CHALL_NO)
WHERE CER.USER_ID = ?
AND CHALL_NO = ?
AND CER_STATUS = 'N'
GROUP BY CHALL_COUNT
```

회원이 한 챌린지에 하루에 몇개의 인증을 해도 달성률 계산에는 하루에 한개로 계산하여 인증의 갯수를 카운트해 챌린지의 총인증 갯수로 나누어 달성률을 계산한다.

```
SELECT
AVG(달성률)
FROM (SELECT CNT/(SELECT CHALL_COUNT FROM CHALL WHERE CHALL_NO = ?) 달성률
FROM (SELECT C.USER_ID, COUNT(DISTINCT TO_CHAR(CER_DATE, 'YYYY-MM-DD')) CNT
FROM CER C
WHERE CHALL_NO = ?
AND CER STATUS = 'N'
GROUP BY C.USER_ID))
```

챌린지에 참여한 `user_id`값으로 `group by`하여 각 회원의 인증 갯수를 구한 뒤, 그 챌린지의 총 인증갯수를 가져와 나누어 회원 별 달성률의 평균을 구하여 챌린지별 달성률을 계산한다.

## 달성률 계산하기-payback

```
SELECT
COUNT(*)
FROM (SELECT CNT/(SELECT CHALL_COUNT FROM CHALL WHERE CHALL_NO = ?) 달성률
      FROM (SELECT C.USER_ID, COUNT(DISTINCT TO_CHAR(CER_DATE, 'YYYY-MM-DD')) CNT
            FROM CER C
            WHERE CHALL_NO = ?
            AND CER_STATUS = 'N'
            GROUP BY C.USER_ID))
WHERE 달성률=1
```

회원별 달성률을 서브 쿼리로 구하고 달성률이 1인 사람을 카운트하여 100% 달성률인 사람의 수를 구한다. 85%미만 달성률인 사람의 수도 이와 같이 계산

```
//돌려받을 금액
int money=(int)(c.getChallPay());
```

```
//달성률이 100%이면 추가금액 지급
if(cer.getAchieve()==1){
    money = c.getChallPay()+(c.getChallPay()*num2)/num1;
}
```

```
//챌린지 상태가 2인지 아닌지 2면 상금 다시 받을수 없음
int challstatus = Integer.parseInt(String.valueOf(request.getAttribute("challstatus")));
```

달성률이 85%인 사람은 챌린지금액을 모두 환급받는다.  
85%미만 달성한 사람은 챌린지 금액이 환급되지 않는다.  
100%달성한 사람에게 최소 달성률을 통과하지 못하여 챌린지 금액을 환급받지 못한 돈을 나누어 추가 지급한다.

## 달성률 계산하기-payback(중복수령방지)

```
UPDATE
  CHALL_STATUS
SET
  CHALL_STATUS = 2
WHERE
  USER_ID = ?
AND
  CHALL_NO = ?
```

종료된 챌린지의 상금 수령을 중복으로 하는 것을 막기 위하여 챌린지 모집현황 디비의 챌린지 현황을 2(종료)로 변경







# Report Code Review

- 신고작성
- 신고처리
- 게시글 비활성화
- 글쓰기 비활성화

## 신고하기

```
//신고 디비에 insert하는 과정
//카테고리별 신고 디비에 신고번호와 카테고리번호를 넣어줌
switch(category){
case "인증" :
    result = new ReportService().insertConfirmReport(r, no);
    break;
case "챌린지" :
    result = new ReportService().insertChallReport(r, no);
    break;
case "자유게시판" :
    result = new ReportService().insertBoardReport(r, no);
    break;

case "카페" :
    result = new ReportService().insertCafeReport(r, no);
    break;
}
```

자유게시판, 카페, 인증, 챌린지에서 신고버튼 클릭 시 **category**값으로 인증, 챌린지, 자유게시판, 카페라는 값을 넘겨줘서 각 카테고리에 맞게 신고디비와 각 카테고리신고 디비에 신고내역을 넣어준다

## 신고하기

```
INSERT ALL
INTO REPORT
(
    REPORT_NO
    , REPORT_TITLE
    , REPORTED_ID
    , REPORT_CONTENT
    , REPORT_DATE
    , REPORT_MODIFY
    , T_F
    , USER_ID
)
VALUES
(
    SEQ_REPID.NEXTVAL
    , ?
    , ?
    , ?
    , SYSDATE
    , SYSDATE
    , DEFAULT
    , ?
)
INTO CER_REPORT
VALUES
(
    SEQ_REPID.CURRVAL
    , ?
)
INTO NEWS
VALUES
(
    SEQ_NEID.NEXTVAL
    , ?
    , 1
    , SEQ_REPID.NEXTVAL
    , DEFAULT
    , SYSDATE
)
SELECT *
FROM DUAL
```

각 카테고리에 신고내역을 넣어줄때 신고번호가 시퀀스를 사용하여 값을 넣어주고 카테고리별 신고 디비에 신고번호를 CURRVAL를 사용하여 넣어주기 위해 insertAll을 사용해서 insert해준다.

이때, 각 카테고리 별 챌린지가 신고되었다는 사실을 게시물 작성자에게 알려주기 위해 NEWS 데이터베이스에도 함께 값을 넣어준다.

챌린지, 자유게시판, 카페, 인증 모두 동일한 방법으로 신고디비에 insert

## 신고처리-글쓰기 비활성화

```
UPDATE
  USER_INFO
SET
  REPORTED = REPORTED + 1,
  WRITE_ACTIVE = CASE WHEN REPORTED >=2 THEN 'N' ELSE WRITE_ACTIVE END,
  REPORTED_DATE = CASE WHEN REPORTED >=2 THEN SYSDATE ELSE REPORTED_DATE END
WHERE
  USER_ID = (SELECT REPORTED_ID
             FROM REPORT
             WHERE REPORT_NO = ?)
```

신고 처리가 될 때 user\_info 디비의 신고 횟수를 증가시켜 준다. 이때 신고횟수가 3번이 되면 글쓰기 활성화 컬럼값을 N으로 변경시켜준다. 그리고 신고당한 날짜를 넣어주고 그 날짜로부터 30일간 인증을 제외한 모든 게시글 글쓰기가 제한된다.

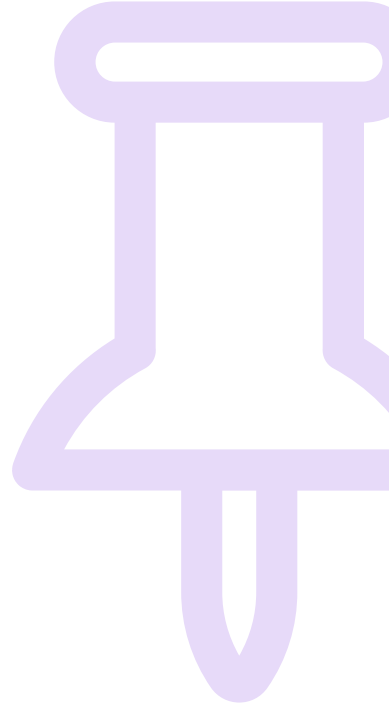
🕒 글쓰기 활성화까지 28일 18시간 45분 34초

USER\_INFO의 글쓰기 활성화 컬럼 값이 N일 경우, 신고당한 날을 기준으로 30일간 글쓰기비활성화  
내가 당한 신고 내역 페이지에서 카운트다운으로 회원에게 알림 (카운트 다운 계산은 챌린지 시작일까지 계산하는 것과 동일)

## 신고처리- 신고 게시물 비활성화

```
SELECT
  C.CAFE_CODE
, CH.CHALL_NO
, CE.CER_ID
, B.BOARD_SEQ
FROM
  REPORT R
  LEFT JOIN CAFE_REPORT C ON (R.REPORT_NO=C.REPORT_NO)
  LEFT JOIN CHALL_REPORT CH ON (R.REPORT_NO=CH.REPORT_NO)
  LEFT JOIN CER_REPORT CE ON (R.REPORT_NO=CE.REPORT_NO)
  LEFT JOIN BOARD_REPORT B ON (R.REPORT_NO=B.REPORT_NO)
WHERE
  R.REPORT_NO = ?
```

각 카테고리의 게시물을 비활성화 시켜주기 위해 신고번호에 맞는 신고 게시물을 가져온다



## 신고처리- 신고 게시물 비활성화

```
//신고게시물 비활성화시키기
if(r.getBoard_seq()!=0) { //자유게시판 신고
    result = new BoardService().disabledPost(r.getBoard_seq());
} else if(r.getCafe_code()!=0) { //카페 신고
    result = new CafeService().disabledPost(r.getCafe_code());
} else if(r.getCer_id()!=0) { // 인증 신고
    result = new CerService().disabledPost(r.getCer_id());
} else if(r.getChall_no()!=0) { // 챌린지모집 신고

    result = new ChallService().disabledPost(r.getChall_no());
    if(result>0) {
        //챌린지에 참여하고 있는 유저 환급
        // 게시물 삭제 성공한 경우
        // 참여중인 인원들 모두 환불 시키기
        int result1 = new MemberDao().refundMoney(conn,r.getChall_no());

        //거래 내역 디비에 환급된거 넣어주기
        int result2 = new HistoryDao().refundChall(conn,r.getChall_no());
        //삭제당한 인원들이 로그인시 확인가능하도록 news디비에 정보 넣기
        int result3 = new NewsDao().insertChall(conn,r.getChall_no());

        // chall_status 디비에 chall_status 1을 0으로 변경
        int result4 = new ChallDao().deleteChallStatus(conn, r.getChall_no());

        if(result1>0&&result2>0&&result3>0&&result4>0) {
            result = result1;
        } else {
            result = 0;
        }
    }
}
}
```

가져온 게시물에 카테고리별 번호를 확인하여 카테고리에 맞는 게시물 번호를 가져가 해당 게시물을 비활성화 시킨다.

챌린지 게시물 비활성화는 참여신청한 유저에게 환급, 챌린지가 삭제되었다는 알림, 환급된 포인트의 거래내역 디비까지 값을 insert해줘야한다.(알림에서 함께 설명)

## 챌린지 신고 추가 작업-챌린지금액 환급

```
UPDATE USER_INFO
SET
    POINT = POINT + (SELECT CHALL_PAYMENT
                     FROM CHALL
                     WHERE CHALL_NO = ?)
, MODIFY_DATE = SYSDATE
WHERE USER_ID IN (SELECT USER_ID
                  FROM CHALL_STATUS
                  WHERE CHALL_NO = ?
                  AND CHALL_STATUS !=0)
```

챌린지가 신고 또는 개설자에 의해 삭제되면 챌린지참가 금액을 환급해줘야하므로 참여한 모든 유저의 point값을 변경

```
INSERT INTO HISTORY
(
    HISTORY_NO
, USER_ID
, HIS_DATE
, HIS_MONEY
, PM
, HIS_COMMENT
, POINT
)
```

해당 챌린지에 참여 신청한 모든 유저의 환급을 HISTORY디비에 챌린지 이름과 챌린지 삭제로 인한 환불이라는 멘트를 concat으로 문자열을 합치고 거래내역 디비에 insert해주기 위해 insert into table select구문을 사용해 챌린지에 참여한 모든인원에 대해 insert를 한번에 진행한다.

```
SELECT SEQ_HID.NEXTVAL
, CS.USER_ID
, SYSDATE
, CHALL_PAYMENT
, 1
, CONCAT(C.CHALL_TITLE, ' 챌린지 삭제로 인한 환불')
, (SELECT POINT FROM USER_INFO WHERE USER_ID = CS.USER_ID)
FROM CHALL_STATUS CS
JOIN CHALL C USING(CHALL_NO)
WHERE CHALL_NO = ?
AND CS.CHALL_STATUS = 1
```

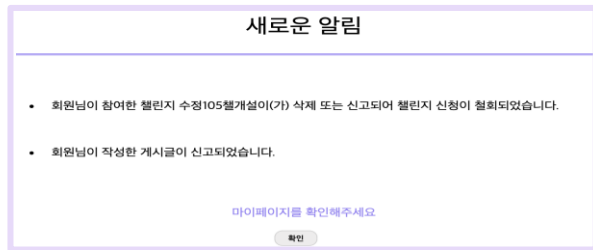
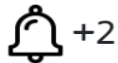


# News Code Review

- 챌린지 삭제, 신고로 인해 챌린지가 삭제
- 나의 게시물이 신고처리
- 내가 한 신고가 처리
- 내가 한 질문에 답변



## 알림 확인



새로운 알림이 있을 경우, 알림의 종류에 따라 유저에게 챌린지 신청이 취소가 되었는지, 게시물이 신고가 되었는지, 질문에 대한 답변이 달렸는지 alert창으로 띄워 알려준다.

```
<%for(int i=0; i<news.size();i++){ %>
<!-- 챌린지0 신고1 질문답변2 -->
<%if(news.get(i).getNews_cate()==1){ %>
<li>회원이 작성한 게시글이 신고되었습니다.</li>

<%}else if(news.get(i).getNews_cate()==0){ %>
<br>
<li>회원이 참여한 챌린지 <%=news.get(i).getTitle() %>이(가) 삭제 또는 신고되어 챌린지 신청이 철회되었습니다.</li>
<%}else if(news.get(i).getNews_cate()==3){ %>
<li>회원이 신고한 <%=news.get(i).getTitle() %>이(가) 신고처리 되었습니다.</li>
<%}else if(news.get(i).getNews_cate()==2){ %>
<li>회원의 질문<%=news.get(i).getTitle() %>에 관리자가 답변을 달아주었습니다.</li>
<%} %>
```

```
<%
if(request.getAttribute("result")!=null){
    if(request.getAttribute("result").equals("success")){
```

팝업창 닫히면 새로 알려줄 알림을 0으로 만들어 주기 위해 부모 창을 reload하며 알림팝업을 닫아준다.

```
<script>
opener.parent.location.reload();
window.close();
</script>
<%}
}%>
```



# Point History Code Review

- 포인트의 입금 출금 내역
- 포인트 환급
- 포인트 충전

## 거래내역 확인

출입금	거래 내역	금액	거래 후 잔액	날짜
출금	현금으로 환급	1000	25000	2021-02-06 15:25:06.0

```
SELECT RNUM, HIS_DATE, HIS_MONEY, PM, HIS_COMMENT, POINT
FROM (SELECT ROWNUM RNUM, HIS_DATE, HIS_MONEY, PM, HIS_COMMENT, POINT
      FROM HISTORY
      WHERE USER_ID = ?
      ORDER BY HISTORY_NO DESC)
WHERE RNUM BETWEEN ? AND ?
```

```
while (rset.next()) {
    java.util.Date date = rset.getTimestamp(2);
    list.add(new History(date, rset.getInt(3), rset.getInt(4), rset.getString(5), rset.getInt(6)));
}
```

포인트의 출금 입금내용을 날짜역순으로 보여줄 때 같은날 거래된 내역을 시간순으로 보기 위해 거래내역 날짜 컬럼을 Timestamp로 받아와 java.util.Date로 캐스팅하여 list에 추가해준다.



# My Page Code Review

- 참여했던 챌린지 그룹
- 참여중인 챌린지 그룹
- 꺾어놓은 챌린지 그룹 / 꺾어놓기 취소

김은화

## 참여했던 챌린지 그룹(1)

```
<%if(month>0){ %>
<section id="content-1">
  <p><%= month %>개월전 참여했던 챌린지 그룹</p>
  <div class="line"></div>
</section>
<%}else{ %>
  <section id="content-1">
    <p>참여했던 챌린지 그룹</p>
    <div class="line"></div>
  </section>
<%} %>
```

-참여했던 챌린지 그룹 3개월, 6개월, 12개월 전  
버튼 클릭 시 해당 개월 수의 챌린지 목록 확인

```
<form action="<%= request.getContextPath() %>/joined/list" method="post">
  <span id="btnType_area">
    <button id="threeMonsBtn" type="submit" name="month" value="3">3개월 전</button>
    <button id="sixMonsBtn" type="submit" name="month" value="6">6개월 전</button>
    <button id="oneyearBtn" type="submit" name="month" value="12">12개월 전</button>
  </span>
</form>
```

## 참여했던 챌린지 그룹(2)

```
<% for(Challenge chall : cList) { %>
<tr>
<td id="chall_no">
  <input type="hidden" value="<%= chall.getChallNo() %>">
</td>
<td>
  
</td>
<td>카테고리 : <%= chall.getCateName() %></td>
<td>챌린지 제목 : <%= chall.getChallTitle() %></td>
<td>
  챌린지 기간 : <%=chall.getChallStart() %>~<%= chall.getEndDate() %>
</td>
<% } %>
</tr>
</table>
</div>

</div>
<% } else { %>
<div id="joined1">참여했던 챌린지그룹이 없습니다.</div>
<% } %>
```

- 카테고리, 챌린지 제목, 챌린지 기간 확인 가능

- 목록 클릭 시 챌린지 인증 페이지로 이동

```
<script>
  // 챌린지 상세보기 기능 (jQuery를 통해 작업)
  $(function(){
    $("#joinedList td").mouseenter(function(){
      $(this).parent().css({"background" : "#e0dbf890", "cursor":"pointer"});
    }).mouseout(function(){
      $(this).parent().css("background", "none");
    }).click(function(){
      var challNo = $(this).parent().children().children().eq(0).val();
      location.href='<%= request.getContextPath() %>/confirm/result?cno='+challNo;
    });
  });
</script>
```

## 참여중인 챌린지 그룹

```
<% if(loginUser!=null && cList!=null) { %>
  <div id="joining">
    <div id="joiningChall">
      <table id="joiningList">
<% for(Challenge chall : cList) { %>
  <tr>
    <td>
      <input type="hidden" name="challNo" value="<%= chall.getChallNo() %>">
    </td>
    <td>
      
    </td>
    <td>카테고리 : <%= chall.getCateName() %></td>
    <td>챌린지 제목 : <%= chall.getChallTitle() %></td>
    <td>
      챌린지 기간 : <%=chall.getChallStart() %>~<%= chall.getEndDate() %>
    </td>
  </tr>
<% } %>
</table>
</div>
</div>
<% } else { %>
  <div id="joined1">참여했던 챌린지그룹이 없습니다.</div>
<% } %>
```

- 참여중인 챌린지 그룹 카테고리, 챌린지 제목  
챌린지 기간 확인 가능
- 챌린지 그룹 목록 클릭 시 챌린지 그룹  
상세 페이지로 이동

## 찜하기한 챌린지 그룹(1)

```
<%@ include file="../../common/common_ui.jsp" %>
<section id="btSection" class="Like_content">
  <section id="content-1">
    <p>찜하기한 챌린지 그룹</p>
    <div class="line"></div>
  </section>
  <% if(loginUser != null && cList!=null) { %>
    <div id="Like">
      <div id="LikeChall">
        <table id="LikeList">
          <% for(Challenge chall : cList) { %>
            <tr>
              <td>
                <input type="hidden" name="challNo" value="<%= chall.getChallNo() %%">
              </td>
              <td>
                
              </td>
              <td>카테고리 : <%= chall.getCateName() %></td>
              <td>챌린지 제목 : <%= chall.getChallTitle() %></td>
              <td><button id="LikeDeleteBtn" class="LikeUpdateBtn">찜하기 취소</button></td>
            </tr>
          <% } %>
        </table>
      </div>
    </div>
  <% } else { %>
    <div id="Like1">찜하기한 챌린지 그룹이 없습니다.</div>
  <% } %>
```

찜하기한 챌린지 그룹 카테고리, 챌린지 제목  
확인 가능

목록 클릭 시 챌린지 상세페이지로 이동



## 찜하기한 챌린지 그룹(2)

```
$(function(){
  //챌린지 찜하기 취소 버튼 클릭 시 찜하기한 챌린지에서 삭제하기
  //id는 고유한 값이기 때문에 버튼을 id로 하면 순서대로 적용은 가능하지만 중간에 내가 다른 찜하기 취소를 클릭;
  //챌린지 리스트로 넘어가게 된다
  //class로 버튼 속성을 주면 순서대로 누르지 않아도 적용 가능
  $("#likeUpdateBtn").click(function(e){
    //likeList에 이벤트를 걸어놔서 버튼에도 이벤트가 걸림 stopProagation으로 막아줌
    e.stopPropagation();
    //td에 있는 버튼
    var num = $(this).parent().parent().parent().children().eq(0).text();
    //찜하기 취소 버튼 클릭 시 넘어가게 하기
    location.href="<%=" request.getContextPath() %>/like/hits?challNo=" +num;
    alert("찜하기 취소가 되었습니다");
  });
});
```

```
// 챌린지 상세보기 기능 (jQuery를 통해 작업)
$(function(){
  $("#likeList td").click(function(){
    var challNo = $(this).parent().children().children().eq(0).val();
    location.href="<%=" request.getContextPath() %>/chall/join?challNo="+challNo;
  });
});
```

-찜하기한 챌린지 그룹 클릭 시

챌린지 상세 페이지로 이동

- 찜하기 취소 클릭 시 ‘찜하기 취소가

되었습니다’ 라는 alert 창을 띄우고

로그인 유저가 찜하기 한 챌린지 그룹의

찜하기 취소가 됨

# Contact Code Review



- 공지사항
- Q&A
- FAQ
- 관리자 페이지
- 카카오톡 1대1 채팅

서재준

## Q&A 질문에 대한 답변

1

```
String title = request.getParameter("title");
String content = request.getParameter("content");
int qnaNo = Integer.parseInt(request.getParameter("qnaNo"));
int reference = Integer.parseInt(request.getParameter("reference"));
```

```
QnA a = new QnA();
a.setQnaTitle(title);
a.setQnaContent(content);
a.setQnaNo(qnaNo);
```

```
int result = new QnAService().updateAnswer(a);
```

```
int qnaNo = Integer.parseInt(request.getParameter("qnaNo"));
```

```
QnA q = new QnAService().selectQuestion(qnaNo);
```

2

```
QnA a = new QnAService().selectAns(qnaNo);
if(a!=null) {
    request.setAttribute("a", a);
}
```

1. 질문에 대한 답변을 할 때 레퍼런스 값에 질문 번호를 넣어 데이터 작성

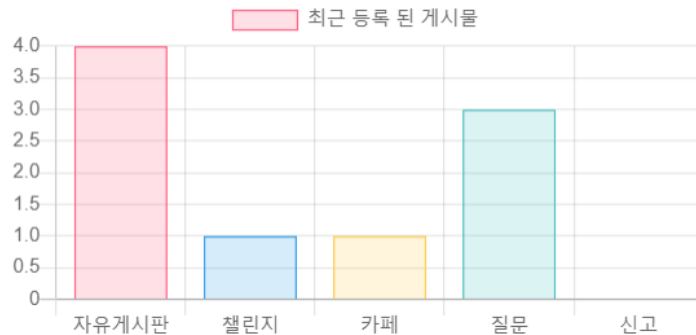
2. 질문자세히 보기를 할 때 답변데이터(a)가 있으면 답변을 질문(q)에 담아서 같이 보여줌

## 관리자페이지(1)

```
<section id="chaSection">
  <!-- 차트div -->
  <div id="chart">
    <div class="container">
      <canvas id="myChart" class="myChart"></canvas>
    </div>
  </div>

<script>
var ctx = document.getElementById('myChart');
var myChart = new Chart(ctx, { type: 'bar', data: { labels: ['자유게시판', '챌린지', '카페', '질문', '신고'],
  datasets: [{ label: '최근 등록 된 게시물', data: [<%= board %>, <%= chall %>, <%= cafe %>, <%= qna %>, <%= report %>],
  backgroundColor: [ 'rgba(255, 99, 132, 0.2)',
    'rgba(54, 162, 235, 0.2)',
    'rgba(255, 206, 86, 0.2)',
    'rgba(75, 192, 192, 0.2)',
    'rgba(153, 102, 255, 0.2)'],
  borderColor: [ 'rgba(255, 99, 132, 1)',
    'rgba(54, 162, 235, 1)',
    'rgba(255, 206, 86, 1)',
    'rgba(75, 192, 192, 1)',
    'rgba(153, 102, 255, 1)'],
  borderWidth: 1 } ] },
  options: { scales: { yAxes: [{ ticks: { beginAtZero: true } } ] } } });
```

1. 값을 담아주는 div 설정
2. 라벨에 맞는 값을 담아주는 스크립트 작성
3. 부트스트랩을 이용한 막대그래프 생성

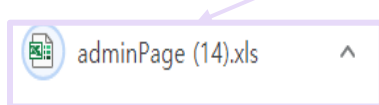


## 관리자페이지(2)

```
<%@ page language="java" contentType="application/vnd.ms-excel; charset=UTF-8"  
pageEncoding="UTF-8"%>
```

```
<%  
response.setHeader("Content-Type", "application/vnd.ms-exls");  
response.setHeader("Content-Disposition", "inline; filename=adminPage.xls");
```

```
<div id="btnArea">  
  <button id="download" type="button" class="btn1"  
    onclick="location.href='<%= request.getContextPath() %>/administratorPage'">다운로드</button>  
</div>
```



1. JSP페이지에 엑셀다운로드 할 수 있는 소스를 줌

2. 바로 저장이 될 수 있는 타입을 설정

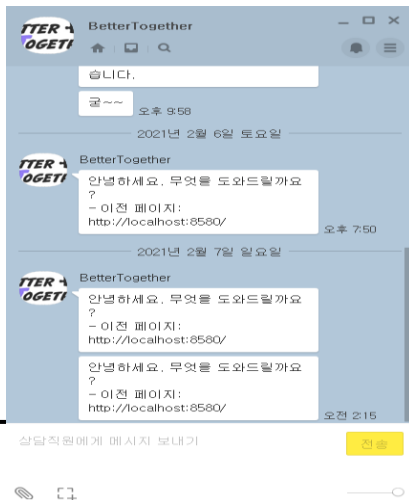
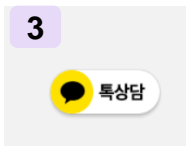
3. 저장 될 이름 입력

4. 다운로드 버튼 클릭 시 다운로드 됨

## 카카오톡 1대1 채팅

```
<div id="kakao-talk-channel-chat-button"></div>

<script src="//developers.kakao.com/sdk/js/kakao.min.js"></script>
<script type='text/javascript'>
  //<![CDATA[
    // 사용할 앱의 JavaScript 키를 설정 1 주세요.
    Kakao.init('dfa18ae3b3f4417156f0f25f0de131eb');
    // 카카오톡 채널 1:1채팅 버튼을 생성합니다.
    Kakao.Channel.createChatButton({
      container: '#kakao-talk-channel-chat-button',
      channelPublicId: '_xotneK' // 카카오톡 채널 홈 URL에 명시된 ID
    });
  //]]>
</script>
```



카카오톡에서 제공하는 오픈API를 활용한 1대1 서비스 기능

1. 카카오톡에서 제공하는 알맞은 자바스크립트 키값을 넣어줌
2. 사용할 채널의 ID를 입력
3. 화면 왼쪽 하단에 특상담 버튼 클릭 시 관리자와 1대1 채팅으로 연결

# 5. 개발환경

---

# Development Tools

SQL Developer



Eclipse



Visual Studio



Google Drive



bootstrap



github



I'mport

I'mport;

EmailJs



Kakao chat



Kakao map



Summernote

summernote

Language

Java  
Html  
CSS  
jQuery  
javaScript



# Thanks

Do you have any questions?

---