
Teu-Ching Teu-Ching

Back-end Developer

Lee Hyeyeong

Project

사용 언어 및 구축 환경

- **Back-end**

Java, Spring Framework, Mybatis

- **Server**

AWS, Ubuntu Linux

- **Front-end**

HTML, CSS, Javascript, JQuery

- **Management**

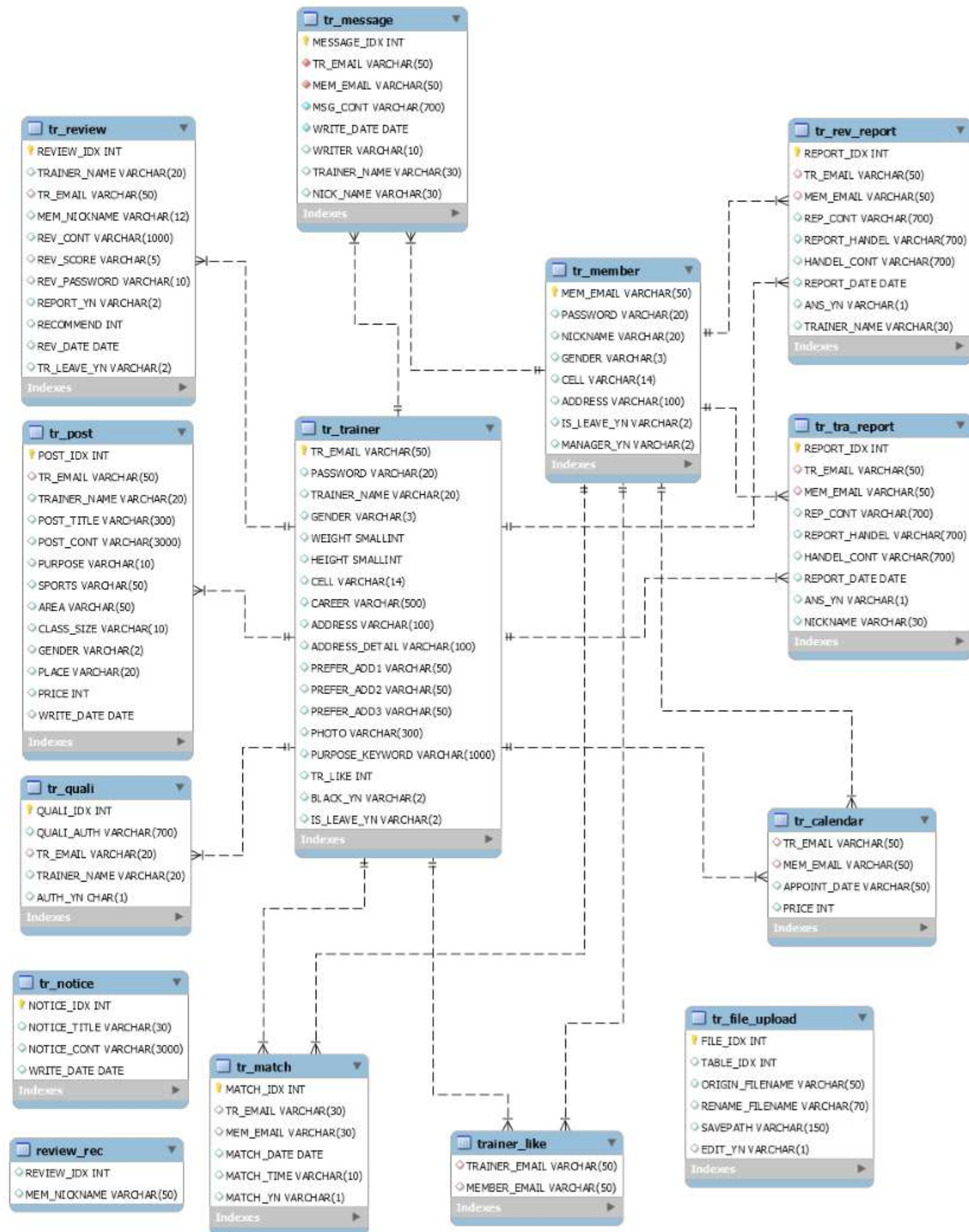
Github

- **Database**

MySQL

ERD

- TR_TRAINER, TR_MEMBER의 EMAIL 컬럼을 기준으로 관계 생성
- TR_TRAINER, TR_MEMBER의 기본키를 제외한 모든 기본키는 INT로 생성하여 AUTO_INCREMENT 적용



담당 기능 및 설명

담당기능 상위권 트레이너, 키워드 검색, 검색어 검색, 공지사항, 게시글, 슬라이드

상위권 트레이너

메인 페이지 접속 시 TR_LIKE 컬럼의 값이 가장 큰 3명의 트레이너를 LIST로 받아옴

```
<select id="selectTopTrainer" resultMap="trainerResult">
  <![CDATA[select @rownum:=@rownum+1, t.* from (select * from TR_TRAINER t order by tr_like desc) t, (SELECT @rownum:=0) TMP where @rownum < 3]]>
</select>
```

```
public ModelAndView index() {
    ModelAndView mav = new ModelAndView();
    List<Trainer> tList = indexService.topTrainers();
    mav.addObject("tList", tList);
    mav.setViewName("landing/landing");
    return mav;
}
```

ModelAndView의 addObject를 통해 Attribute 추가

→ JSTL Core forEach를 사용하여 JSP 파일에서 반복문으로 List 출력

담당 기능 및 설명

키워드 검색

운동목적	<input type="checkbox"/> 체형교정 <input type="checkbox"/> 건강증진 <input type="checkbox"/> 다이어트 <input type="checkbox"/> 바디프로필 <input type="checkbox"/> 취미 <input type="checkbox"/> 대회준비 <input type="checkbox"/> 재활치료 <input type="checkbox"/> 벌크업 <input type="checkbox"/> 기초체력 <input type="checkbox"/> 시험준비
운동종목	<input type="checkbox"/> 요가 <input type="checkbox"/> 필라테스 <input type="checkbox"/> 스피닝 <input type="checkbox"/> 헬스 <input type="checkbox"/> 크로스핏 <input type="checkbox"/> 수영 <input type="checkbox"/> 기타
지역	<input type="checkbox"/> 등록지 기준 <input type="checkbox"/> 타지역
수업 인원	<input type="checkbox"/> 1:1 <input type="checkbox"/> N:1
성별	<input type="checkbox"/> 남 <input type="checkbox"/> 여
장소	<input type="checkbox"/> INDOOR EXERCISES <input type="checkbox"/> OUTDOOR EXERCISES

Search

키워드에 클릭 이벤트를 추가하여 키워드 클릭 시 JavaScript에 값이 저장되도록 함

지역은 등록지 기준, 타지역으로 나뉨

① 등록지 기준

1. 회원 : 회원 정보의 주소에서 '시'만 추출
2. 비회원 : 등록지 기준 설정 불가능 알림을 띄움

② 타지역

'시' 단위로 선택할 수 있음

③ 선택안함

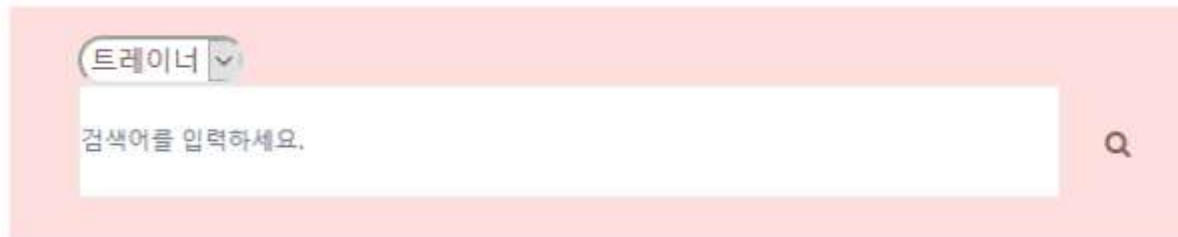
1. 회원 : 회원 정보의 주소에서 '시'만 추출
2. 비회원 : 서울로 자동 설정한다는 알림창 띄움
수락할 경우, 서울로 설정
수락하지 않을 경우, 타지역 설정

```
function keywordSearch() {  
    $.ajax({  
        url: "http://52.78.116.59:8080/teuching/matching/keyword.do",  
        data: {  
            purpose : purpose,  
            sports : sports,  
            area : area,  
            classSize : classSize,  
            gender : gender,  
            place : place  
        },  
        success: function(v) {  
            for(i = 0; i < v.length; i++) {  
                pList.push(v[i]);  
            }  
        }  
    })  
    .done(function() {  
        createTable();  
    })  
}
```

Ajax를 이용하여 비동기 통신 진행
가져온 값을 Controller로 보내 Post List를 가져옴

담당 기능 및 설명

검색어 검색



select로 트레이너와 게시글 선택

트레이너 : TRAINER_NAME 컬럼 값들 중 해당 검색어를 포함하고 있는 행만 추출

게시글 : POST_TITLE 컬럼 값들 중 해당 검색어를 포함하고 있는 행만 추출

```
document.getElementById("inputBtn").addEventListener('click', function(e) {
    pList = new Array();
    var input = document.getElementById("keywordInput");
    var option = document.getElementById("searchOption");

    if(input.value == '') {
        alert("검색어를 입력해주세요.");
    } else {
        $.ajax({
            url : "http://52.78.116.59:8080/teuching/matching/searchword.do",
            data : {
                "input" : input.value,
                "option" : option.value
            },
            success : function(v) {
                for(i = 0; i < v.length; i++) {
                    pList.push(v[i]);
                }
            }
        })
        .done(function() {
            createTable();
        })
    }
})
```

Ajax를 이용하여 비동기 통신 진행

가져온 값을 Controller로 보내 Post List를 가져옴

담당 기능 및 설명

공지사항 1. 페이징

Notice

제목	내용	작성일자	작성자
2020년 6월 12일 서버점검안내	안녕하세요. TEUCHING 개발팀입니다.	2020-06-11	TEUCHING
2020년 6월 10일 서버점검안내	안녕하세요. TEUCHING 개발팀입니다.	2020-06-09	TEUCHING
2020년 6월 9일 추가점검안내	안녕하세요. TEUCHING 개발팀입니다.	2020-06-09	TEUCHING
2020년 6월 9일 서버점검안내	안녕하세요. TEUCHING 개발팀입니다.	2020-06-08	TEUCHING
2020년 6월 8일 서버점검안내	안녕하세요. TEUCHING 개발팀입니다.	2020-06-07	TEUCHING

한 번에 5개의 게시글만 확인할 수 있음

다음 페이지, 이전 페이지 클릭 시 start 값 가공하여 Controller에 전달 → DB에서 Limit에 사용

```
function back(start, el) {  
  if(start <= 1) {  
    alert("첫 번째 페이지입니다.");  
  } else {  
    el.href = "http://52.78.116.59:8080/teuching/notice/notice.do?start=" + (start-5) + "&end=" + (start-1);  
  }  
}  
  
function next(end, el, size) {  
  if((end + 1) > size) {  
    alert("마지막 페이지입니다.");  
  } else {  
    el.href = "http://52.78.116.59:8080/teuching/notice/notice.do?start=" + (end+1) + "&end=" + (end+5);  
  }  
}
```

```
<select id="selectNotice" resultMap="noticeMap">  
  select * from TR_NOTICE order by notice_idx desc limit ${start}, 5  
</select>
```

담당 기능 및 설명

공지사항 2. 등록, 변경, 삭제

관리자 계정으로 로그인 했을 때에만 해당 기능 사용 가능 (세션에 등록)

① 게시글 작성 페이지로 이동

새로운 게시글 등록 시 noticeIdx로 0 전달, 기존 게시글 수정 시 NOTICE_IDX 컬럼 값 전달

```
public ModelAndView writePage(int noticeIdx) {  
    ModelAndView mav = new ModelAndView();  
    if(noticeIdx > 0) {  
        Notice notice = noticeService.noticeDetail(noticeIdx);  
        mav.addObject("notice", notice);  
    }  
    mav.setViewName("notice/writeNotice");  
    return mav;  
}
```

② 게시글 삭제

TR_NOTICE의 NOTICE_IDX의 값을 Controller로 전달, 해당 NOTICE_IDX의 값을 가지고 있는 NOTICE를 삭제함

```
public ModelAndView deleteNotice(String noticeIdx) {  
    ModelAndView mav = new ModelAndView();  
    int res = noticeService.deleteNotice(noticeIdx);  
    mav.setViewName("redirect:/notice/notice.do?start=0&end=4");  
    return mav;  
}
```


담당 기능 및 설명

공지사항 2. 등록, 변경, 삭제

③ 게시글 저장 or 변경

새로운 게시글 등록 시 noticeIdx로 0 전달, 기존 게시글 수정 시 NOTICE_IDX 컬럼 값 전달

```
public ModelAndView write(Notice notice) {
    ModelAndView mav = new ModelAndView();
    int res = 0;
    if(notice.getNoticeIdx() > 0 ) {
        res = updateNotice(notice);
    } else {
        res = insertNotice(notice);
    }

    mav.setViewName("redirect:/notice/notice.do?start=0&end=4");
    return mav;
}
```

기존 게시글 수정 시 NOTICE_TITLE과 NOTICE_CONT의 값을 변경함

```
<update id="updateNotice">
    update TR_NOTICE set notice_title = #{noticeTitle}, notice_cont = #{noticeCont} where notice_idx = #{noticeIdx}
</update>
```

새로운 게시글 등록 시 SQL문

```
<insert id="insertNotice">
    insert into TR_NOTICE values(0, #{noticeTitle}, #{noticeCont}, sysdate())
</insert>
```

담당 기능 및 설명

슬라이드(About Us)

슬라이드 옆 버튼에 클릭 이벤트 추가
클릭 시 아래 함수 실행

```
document.querySelectorAll('.slide_btn').forEach(function(el) {  
  el.addEventListener('click', function() {  
    move_text('true');  
  
    index = el.id.substring(el.id.length - 1);  
    move_slide(--index);  
    window.setTimeout(function() {  
      move_text('false');  
    }, 1500);  
  });  
});
```

```
function move_text(rgx) {  
  var title = document.querySelectorAll('.about_title');  
  var content = document.querySelectorAll('.about_content');  
  
  if(rgx == 'true') {  
    title.forEach(function(el) {  
      el.style.opacity = '0';  
      el.style.transform = 'translateY(100vh)';  
    })  
  
    content.forEach(function(el) {  
      el.style.opacity = '0';  
      el.style.transform = 'translateY(100vh)';  
    })  
  } else {  
    title.forEach(function(el) {  
      el.style.opacity = '1';  
      el.style.transform = 'translateY(44vh)';  
      el.style.transitionDuration = '1s';  
    })  
  
    content.forEach(function(el) {  
      el.style.opacity = '1';  
      el.style.transform = 'translateY(46vh)';  
      el.style.transitionDuration = '1s';  
    })  
  }  
}
```

```
function move_slide(idx) {  
  document.querySelectorAll('.slide_common').forEach(function(el) {  
    el.style.transform = 'translateY(-' + (idx * 100) + 'vh)';  
    el.style.transitionDuration = '1.5s';  
  });  
}
```

↑ 슬라이드 사진 이동 함수
idx의 값만큼 세로로 이동

← 슬라이드 안 글자 이동 함수
모든 글자를 세로로 이동