

PROFILE



TEL



010-6376-1105

학력



강릉원주대학교 유아교육학과

GIT



<https://github.com/hyojinnn>

GITBOOK



<https://hyojin-2.gitbook.io/hyojinportfolio/>

MAIL



yhjin92@gmail.com

CAREER



2021.07.15 - 2021.12.28

더조은컴퓨터아카데미 '자바기반 안드로이드 앱개발' 교육수료

TECH STACK

언어



DATABASE



개발툴



형상관리



프로젝트 소개

지금 나의 상황에 딱 맞는 멘토는 어디에?

쉽고 빠른 매칭과 다양한 수업 콘텐츠가 존재하는 숨은 잔디

늘어나는 수요에 비해 만족스럽지 않던 온라인 강의와

복잡한 카테고리 속에서 찾기 힘들던 나의 멘토 / 멘티들!

이젠 숨은 잔디에서 폭 넓게 비교 후 선택해보세요.

만족도 높은 학습을 위해 1 대 1 코칭 환경을 제공합니다!



프로젝트 기획의도

왜 IT 온라인 강의 플랫폼을 선택했는지?



온라인 교육산업의 증가

프리랜서가 직업으로서 선호도가 높아지고, 코로나 19 여파로 특정 전문기술을 보유한 유능한 프리랜서 전문가의 채용이 늘면서 온라인 교육자에 대한 수요도 증가했다.



수요에 비해 낮은 만족도

동영상의 품질 문제부터, 서버 상태로 번번이 일어나는 출결 체크 불량, 사용하는 기기 문제, 강의하는 강의자의 장비 문제 등등 세부적으로 매우 많은 단점들을 온라인 강의의 단점으로 뽑고 있다.



IT 관련 온라인 교육시장 현황

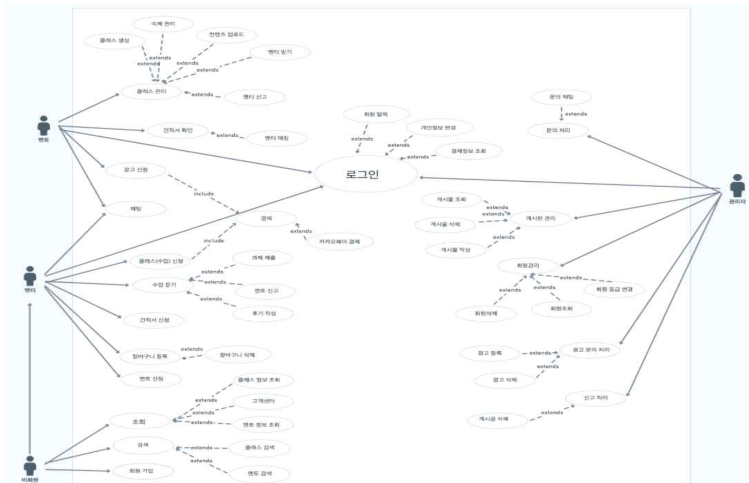
프로그래밍, 디지털 마케팅, IT 등 전문기술 교육 분야 교육자 74%는 교사 자격증 없이 온라인 플랫폼을 통해 수업을 올리고 있는 것으로 나타났다. 교육 자격증 유무와 관계없이 자신의 기술을 활용해 온라인 교육을 부수입원으로 활용하는 예는 더욱 늘어날 것으로 보인다.

FINAL PROJECT

프로젝트 명	숨은잔디
프로젝트 목적	IT 온라인 강의 플랫폼
사용언어	HTML, CSS, JS, JQuery, JSTL, BootStrap, semantic UI
사용 물	SPRING(5.2.9)
DB/Server	Oracle / Tomcat(8.5)
OS	Window10
개발기간	2021.11.12 ~ 2021.12.27 (6주)
개발인원	7명
담당업무	관리자페이지(회원조회, 관리자관리, 신고기능), 마이페이지(수강완료, 수강후기페이지)

FINAL PROJECT

구현기능

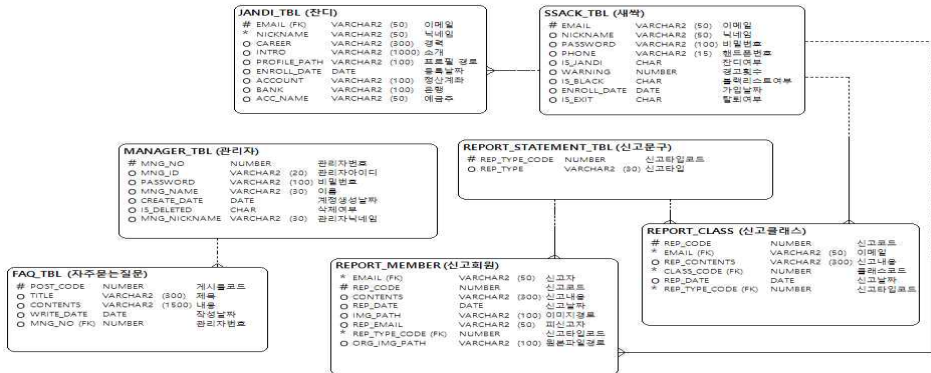


개발기간

전체기간	1주	2주	3주	4주	5주	6주
프로젝트 기획						
UI 설계 및 구현						
데이터베이스 설계						
기능구현						
테스트 및 디버그						

FINAL PROJECT

ERD



FINAL PROJECT

화면구현 관리자 - 새싹조회

등록된 새싹 조회

이메일	이름	닉네임	잔디 여부	경고 횟수	불역 리스트	가입 날짜
dbtjsal465@gmail.com	김철수	훈이	Y	0	N	2021-12-28
dbtjsal113@gmail.com	유선미	선미는 새싹	N	0	N	2021-12-27
test1@gmail.com	테스트	둥그라미	N	0	N	2021-12-27
jandi9@test.com	이천수	이천수	Y	0	N	2021-12-26
jandi8@test.com	박신양	박신양	Y	0	N	2021-12-26
jandi7@test.com	김종구	김종구	Y	0	N	2021-12-26
jandi6@test.com	박신혜	박신혜	Y	0	N	2021-12-26
jandi5@test.com	박해수	박해수	Y	0	N	2021-12-26
jandi4@test.com	이정재	이정재	Y	0	N	2021-12-26
jandi3@test.com	정우성	정우성	Y	0	N	2021-12-26

① 선택 🔍

② < 1 2 3 4 5 >

① 카테고리별로 검색할 수 있도록 쿼리문을 작성한다.

② 한 페이지에 10개의 게시글이 나올 수 있도록 페이지처리한다.

FINAL PROJECT

화면구현 관리자 - 새싹조회

①

```
<where>
  <if test='searchCondition == "1"'>
    A.NICKNAME LIKE '%' || #{ searchValue } || '%'
  </if>
  <if test='searchCondition == "2"'>
    AND A.EMAIL LIKE '%' || #{ searchValue } || '%'
  </if>
  <if test='searchCondition == "3"'>
    AND A.NAME LIKE '%' || #{ searchValue } || '%'
  </if>
</where>
```

②

```
// 새싹염버 목록 페이지 처리
Map<String, String> searchMap = new HashMap<>();
searchMap.put("searchCondition", searchCondition);
searchMap.put("searchValue", searchValue);
System.out.println("searchMap : " + searchMap);

// 새싹 염버 수 조회
int totalCount = managerService.selectSsackTotalCount(searchMap);
System.out.println("totalCount : " + totalCount);

// 한 페이지 당 10개씩 조회
int limit = 10;
int buttonAmount = 5;
```

①

카테고리별로 검색할 수 있도록
쿼리문을 작성한다.






②

한 페이지에 10개의 게시글이
나올 수 있도록 페이지처리한다.

FINAL PROJECT

화면구현 관리자 - 잔디조회

① 등록된 잔디 조회

이메일	이름	내역명	잔디 등록 일자
 kimsoack@naver.com	김석	21세기 장영실	2021-12-30
 parksoack@naver.com	박석	파닉스 이스티	2021-12-30
 hong@gmail.com	홍이	엑시현코딩	2021-12-30
 Leejung@gs.com	리정정	혁신 매거진	2021-12-30
 gabae@kacha.com	가가바	Uno 프로젝트	2021-12-25

선택

00000000

②



21세기 장영실

잔디 경력

원) 서경직업전문학교 정보기술개발 직업훈련교사
수원스마트융합대학원 운영
US전자 MC사업부
일본 아이지전 (株)東海電機 Security사업부

소개

교육하고 책 쓰는 게 좋아요.
양평군의 상촌포항 우육의 재능과 장영실 포항 교육 클러터를 운영중입니다.

① 새싹과는 다르게 잔디의 프로필 사진이 추가되어 한페이지에 5개의 게시글이 나올 수 있도록 한다.

② 게시글을 클릭하면, 원하는 잔디의 상세정보를 확인가능하다.

FINAL PROJECT

화면구현 관리자 - 잔디조회

①

```
// 잔디멤버 페이지 처리하기
Map<String, String> searchMap = new HashMap<>();
searchMap.put("searchCondition", searchCondition);
searchMap.put("searchValue", searchValue);
System.out.println("searchMap : " + searchMap);

// 잔디 멤버 수 조회하기
int totalCount = managerService.selectJandiTotalCount(searchMap);
System.out.println("totalCount : " + totalCount);

// 한 페이지에 5개씩 조회하기
int limit = 5;
int buttonAmount = 5;
```

②

```
<c:forEach var="jandiMember" items="${ sessionScope.jandilist }">
  <tr class="move_jandiBtn" onclick="location.href='${ pageContext.servletContext.contextPath }/findJandi/jandiProfile/${ jandiMember.email }'">
    <td><img src='${ pageContext.servletContext.contextPath }/resources/uploadFiles/profile/${ jandiMember.profile_path }' style="width:70px;"></td>
    <td>${ jandiMember.email }</td>
    <td>${ jandiMember.acc_name }</td>
    <td>${ jandiMember.nickName }</td>
    <td>${ jandiMember.enroll_date }</td>
  </tr>
</c:forEach>
```

① 새싹과는 다르게 잔디의 프로필 사진이 추가되어 한 페이지에 5개의 게시글이 나올 수 있도록 한다.

② 게시글을 클릭하면, 원하는 잔디의 상세정보를 확인가능하다.

FINAL PROJECT

화면구현 관리자 - 관리자계정 관리

① 관리자 계정 정보

관리자이름	아이디	닉네임	계정 비활성화
<input type="checkbox"/> 양진진	manager4@test.com	manager4	Y
<input checked="" type="checkbox"/> 이지환	manager8@test.com	manager8	N
<input type="checkbox"/> 한소리	manager1@test.com	manager1	N
<input type="checkbox"/> 송그래	manager2@test.com	manager2	N
<input type="checkbox"/> 유시은			Y
<input type="checkbox"/> 김소소			N
<input type="checkbox"/> 이투리			N

해당 관리자계정을 비활성화하시겠습니까?

확인

취소

계정 비활성화

① 비활성화 하고 싶은 관리자를 체크한 후 버튼을 누르면 모달창 활성화한다.

② 비활성화 완료 후 완료모달창 활성화한다.

③ 비활성화가 완료되면 상태값이 'Y'로 변경한다.

②

계정 비활성화가 완료되었습니다.

확인

③

관리자이름	아이디	닉네임	계정 비활성화
<input type="checkbox"/> 양진진	manager4@test.com	manager4	Y
<input type="checkbox"/> 이지환	manager8@test.com	manager8	Y
<input type="checkbox"/> 한소리	manager1@test.com	manager1	N

FINAL PROJECT

화면구현 관리자 - 관리자계정 관리

①

```
var chkArray = new Array();
$("input:checkbox[name='check']:checked").each(function(){
    chkArray.push(this.value);
});
e.preventDefault();

if(chkArray == '') {
    $("#nullModal").fadeIn();
    $(".btn").click(function(){
        $("#nullModal").fadeOut();
    });
} else {
    $("#deleteModal").fadeIn();
    $(".deletebtn1").click(function(){
        $.ajax({
            type: "POST",
            url: "${ pageContext.servletContext.contextPath }/manager/inactiveManager",
            data: {checkbox : chkArray},
            success: function(data){
                console.log("완료");
            },
            error: function(xhr,status,error){
                console.log("에러");
            }
        });
    });
}
```

①

체크된 값을 확인하여 ajax로 넘겨준다.

②

체크된 값을 조회하여 상태값을 'Y'로 바꾸어주는 쿼리문 작성한다.

②

```
// manager, checkbox 값 확인하기
System.out.println("manager : " + manager);
System.out.println("checkbox : " + checkbox);

// 체크된 값 확인하여 비활성화 하기
int result = managerService.inactiveManager(checkbox);
```

```
<update id="inactivemanager" parameterType="java.util.ArrayList">
    UPDATE MANAGER_TBL
    SET IS_DELETED = 'Y'
    WHERE MNG_NO IN (
        <foreach collection="list" item="item" index="index" separator=",">
            #{ item }
        </foreach>
    )
</update>
```

FINAL PROJECT

화면구현 관리자 - 관리자계정 생성

관리자 계정 생성

이름		
양시형	올바른 이름 형식입니다.	①
닉네임		
manager10	사용 가능한 닉네임입니다.	②
아이디		
manager7@test.com	사용 가능한 이메일입니다.	③
비밀번호		

비밀번호 확인		
****	비밀번호가 일치합니다.	④

관리자 계정 생성

이름		
asdf	올바르지 않은 이름 형식입니다.	
닉네임		
manager4	이미 있는 닉네임입니다.	
아이디		
manager4@test.com	이미 있는 이메일 주소입니다.	
비밀번호		

비밀번호 확인		
***	비밀번호가 일치하지 않습니다.	

- ① 정규식을 이용하여 올바른 이름인지 체크한다.
- ② ajax로 값을 전달하여 DB에 동일한 닉네임이 있는지 확인한다.
- ③ 닉네임과 동일하게 ajax로 값을 전달하여 확인한다.
- ④ 비밀번호가 동일한지 value값을 확인하여 체크한 후 암호화 처리하여 전달한다.

FINAL PROJECT

화면구현 관리자 - 관리자계정 생성

①

```
/* 관리자 계정 생성 - 이름 체크 */
$(function () {
    $("#mgrName").keyup(function () {
        let filter = /^[가-힣]+$/;

        if (!filter.test($(this).val())) {
            $(".nameCheck")
                .html("올바르지 않은 이름 형식입니다.")
                .css("background", "lightpink");
        } else {
            $(".nameCheck")
                .html("올바른 이름 형식입니다.")
                .css("background", "#E3FFEC");
        }
    });
});
```

②

```
/* 관리자 계정 생성 - 닉네임 체크 */
$("#mgrNickName").on("keyup", function () {
    const nickName = $("#mgrNickName").val();
    $.ajax({
        url: "${pageContext.servletContext.contextPath}/manager/nickNameCheck",
        data: {nickName: nickName},
        type: "POST",
        success: function(data){
            if(data == "true"){
                console.log(data);
                $(".nickNameCheck").html('이미 있는 닉네임입니다.')
                $("#mgrNickName").focus().css("background", "lightpink");
            } else {
                $(".nickNameCheck").html('사용 가능한 닉네임입니다.')
                $("#mgrNickName").focus().css("background", "#E3FFEC");
            }
        },
        error: function(xhr, status, error){
            console.log(error);
        }
    });
});
```

③

```
/* 관리자 계정 생성 - 이메일 체크 */
$("#mgrId").on("keyup", function(){
    const email = $("#mgrId").val();
    $.ajax({
        url: "${pageContext.servletContext.contextPath}/manager/emailCheck",
        data: {email: email},
        type: "POST",
        success: function(data){
            if(data == "true"){
                console.log(data);
                $(".emailCheck").html('이미 있는 이메일 주소입니다.')
                $("#mgrId").focus().css("background", "lightpink");
            } else {
                $(".emailCheck").html('사용 가능한 이메일입니다.')
                $("#mgrId").focus().css("background", "#E3FFEC");
            }
        },
        error: function(xhr, status, error){
            console.log(error);
        }
    });
});
```

④

```
// 관리자 password 암호화 처리하기
manager.setPassword(passwordEncoder.encode(manager.getPassword()));

// 관리자 계정 생성 확인
if(!managerService.msregistMember(manager)) {
    throw new MemberRegisterException("관리자 계정 생성에 실패했습니다.");
} else {
    // 회원 페이지로 리다이렉트하기
    return "redirect:/manager/manproduce";
}
```

① 정규식을 이용하여 올바른 이름인지 체크한다.

② ajax로 값을 전달하여 DB에 동일한 닉네임이 있는지 확인한다.

③ 닉네임과 동일하게 ajax로 값을 전달하여 확인한다.

④ 패스워드가 동일한지 value값을 확인하여 체크한 후 암호화 처리하여 전달한다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리

신고된 회원 조회

이메일	닉네임	내용	신고 사용	신고 날짜	신고 처리 여부
leehy@naver.com	리하이	ㅇㅇ	확인	2021-12-30	Y
lee@gmail.com	리선생	ㅎㅎ	확인	2021-12-30	N
nam@gmail.com	남남	너무 기분파해요.	확인	2021-12-27	Y
ssack3@test.com	소리질러	강의가 너무 비싸요	확인	2021-12-30	N
ssack3@test.com	소리질러	성도양호	확인	2021-12-34	Y
ssack3@test.com	소리질러	불량이 없으신것 같네요.	확인	2021-12-23	Y
nam@gmail.com	남남	기분 나쁘게 말했어요.	확인	2021-12-30	Y
nam@gmail.com	남남	너무 기분파해요.	확인	2021-12-16	Y
nam@gmail.com	남남	신고신고	확인	2021-12-16	Y
adghy123@naver.com	양정진	불러올라	확인	2021-12-16	Y

선택

0000

신고된 회원 상세내용

신고카테고리

욕설

이메일

ssack3@test.com

닉네임

소리질러

이미지 첨부

spring

신고사유

강의가 너무 비싸요

신고

반려

① 신고된 회원 관련 게시글을 클릭하면 모달창을 통해 신고상세내용을 확인할 수 있다.

② 신고/반려 처리를 통해 상태값이 변경된다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리

①

```
// * 회원 상세 신고된 회원 상세 내용 띄어주기 *  
$(function(){  
    $("#repbtn").click(function(e){  
        console.log($(e.target).parent().children("input[type=hidden]").val());  
        const repCode = $(e.target).parent().children("input[type=hidden]").val();  
        $("#repCodeinput").val(repCode);  
        $.ajax({  
            url: "${ pageContext.servletContext.contextPath }/manager/repDetail",  
            data: {repCode : repCode},  
            type: 'GET',  
            async: false,  
            success: function(data){  
                console.log('출력값');  
                console.log(data.repCategory);  
                $("#repCategory").val(data.repCategory);  
                $("#repEmail").val(data.repEmail);  
                $("#repNickName").val(data.repNickName);  
                $("#repContents").val(data.repContents);  
                $("#isBlack").val(data.isBlack);  
                $("#imgPath").prop('src', "${ pageContext.servletContext.contextPath }/resources/uploadFiles/reportFiles/"  
                    + data.imgPath);  
            },  
            error: function(xhr, status, error){  
                console.log(error);  
                console.log(xhr);  
            }  
        });  
        $("#repMembermodal").show();  
    });  
});
```

① ajax를 이용하여 모달창에 신고 상세내용을 띄어준다.

②

```
// JSON 객체 생성하여 데이터 담기  
JSONObject jsonObject = new JSONObject();  
jsonObject.put("repEmail", repMember.getRepEmail());  
jsonObject.put("repCategory", repMember.getReportStatementDTO().getRepType());  
jsonObject.put("repContents", repMember.getContents());  
jsonObject.put("imgPath", repMember.getImgPath());  
jsonObject.put("repNickName", repMember.getNickName());  
jsonObject.put("isBlack", repMember.getIsBlack());  
  
System.out.println("확인출 : \n" + jsonObject);  
  
// JSONObject를 string으로 변환하여 전달하기  
return jsonObject.toJSONString();
```

② json 객체를 이용하여 string으로 변환하여 다시 전달해준다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리

신고 회원 상세내용

신고카테고리

육살

이메일

ram@gmail.com

닉네임

람람

이미지 첨부

TEAM
Introduce



신고사유

신고

반려

신고처리가 완료되었습니다.

확인취소

신고 날짜	신고 처리 여부
2021-12-24	W
2021-12-16	Y
2021-12-16	Y
2021-12-16	N
2021-12-16	Y
2021-12-16	W
2021-12-16	Y
2021-12-16	Y
2021-12-16	Y
2021-12-16	W

① 신고처리 시 상태값이 'Y'로 변경된다.

② 신고처리 시 상태값이 'N'로 변경된다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리

```
/**
 * 신고회원생성내용 (신고처리)
 * @author 최정
 */
@PostMapping("confirm")
public String reportConfirmMember(@ModelAttribute ReportMemberDTO repMember) {

    // repMember에 담긴 신고된 회원의 상세정보 확인하기
    System.out.println("repMember : " + repMember);

    // 신고된 회원 신고처리 하기
    int result = managerService.modifyConfirmMember(repMember);

    if(result > 0 ) {
        System.out.println("신고처리성공");
        // 신고처리 된 회원 MemberDTO에서 조회하기
        MemberDTO member = managerService.selectRepMember(repMember);
        System.out.println("member : " + member);
    }

    /**
     * 신고된 회원이 블랙리스트 회원이 아닐 경우 경고 +1이 추가되며,
     * 경고가 2을경우 블랙리스트 회원으로 처리되어 IS_BLACK에 'Y'로 변경된다.
     */
    if(member.getIsBlack() != 'Y') {
        if(member.getWarning() < 3) {

            int result2 = managerService.updateMemberWarning(member);

        }
        if(member.getWarning() == 2) {

            int result3 = managerService.updateMemberBlack(member);

        }
    }
}
```

①

```
<!-- 신고된 회원 신고처리 -->
<update id="modifyConfirmMember" parameterType="ReportMemberDTO">
    UPDATE REPORT_MEMBER_TBL
    SET REP_YN = 'Y'
    WHERE REP_CODE = #{ repCode }
</update>
```

```
<!-- 신고된 회원 관리자처리 -->
<update id="modifySendbackMember" parameterType="ReportMemberDTO">
    UPDATE REPORT_MEMBER_TBL
    SET REP_YN = 'N'
    WHERE REP_CODE = #{ repCode }
</update>
```

①

신고 또는 처리 여부에 따라
상태값이 'W'에서
'Y', 'N'으로 변경된다.

②

신고처리 시 회원의 누적 경고가
+ 1 처리되며,
누적경고횟수가 2인 회원은
블랙리스트 회원으로 변경된다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리(블랙리스트)

신고 회원 상세내용

신고카테고리 육살

이메일 ram@gmail.com

닉네임 람람

이미지 첨부

TEAM Introduce

신고사유

신고 반려

localhost:8585 내용:
이미 블랙리스트 회원입니다.

확인

누적 경고로 인해 블랙리스트 조치되었습니다. 로그인 가능까지 29일 남았습니다.

확인

이메일 찾기 비밀번호

① 회원이 이미 블랙리스트회원일 경우 모달창이 나타난다.

② 블랙리스트 회원이 로그인을 시도할 시 로그인 실패와 동시에 로그인까지 남은 기한이 나타난다.

FINAL PROJECT

화면구현 관리자 - 신고된 회원 관리(블랙리스트)

①

```
/* 신고버튼 클릭 후 '확인' 버튼 클릭 시 블랙리스트 회원인지 조회하기 */
$('#con-btn').click(function(event){

    let isBlack = $('#isBlack').val();
    console.log(isBlack);

    if(isBlack == 'Y') {
        alert('이미 블랙리스트 회원입니다. ');
    } else {
        $('#resultModal').show();
    }
    event.preventDefault();
});
```

②

```
<!-- 경고 3번 된 회원 블랙리스트 처리 -->
<update id="updateMemberBlack" parameterType="MemberDTO">
    UPDATE SSACK_TBL S
        SET S.IS_BLACK = 'Y'
        , S.BLACK_END_DATE = SYSDATE + 30
    WHERE EMAIL = #{ email }
</update>
```

① 쿼리문에서 BLACKCODE를 조회하여 블랙리스트 회원인지 아닌지 확인한다.

② 누적경고가 3번이 될 시 30일동안 로그인 제한이 걸린다.

FINAL PROJECT

화면구현 관리자 - 신고된 클래스 관리

신고된 클래스조회

①

신고한 회원	신고된 클래스	관리자내명	내용	신고 사유	신고 처리 여부
ran@gmail.com	AWS 중/상급자를 위한 강의	츄가이버	ss	욕설	Y
trsvy2004@gmail.com	스토리지론	진한	아이유가 지브니요	욕설	Y
honey@hollbang.com	프로그래밍 시작하기 : 파이썬 입문	코태의 달인	희상하하	욕설	Y
asdfghj1290@naver.co	버전 관리 시스템 Git	Android Rookie	왜그리지?	욕설	Y
jandi7@test.com	[조금] 40분만에 풀이하는 SaaS	환진디디	여학생들거봐	욕설	Y
Leekung@yju.com	만들고 비교하여 학습하는 리팩트	프론트의 존재	성의가 없어.	욕설	N
asdfghj1290@naver.co	C++20 풀이보기	Lu89 Developer	목차가 1개밖에 없어	욕설	Y
asdfghj1290@naver.co	[조금] 40분만에 풀이하는 SaaS	환진디디	dodddd	욕설	Y
trsvy2004@gmail.com	Flutter 중급 - 클린 아키텍처	Lu89 Developer	이상해요 강의가	욕설	Y
hvan@gmail.com	웹 앱오면서 좋은 안드로이드 앱 개발하기	Lu89 Developer	강의에 성의가 없어서 지적	욕설	Y

선택



신고 클래스 상세내용

신고카테고리

신고된 클래스

닉네임

신고사유

②

① 신고된 클래스를 가진 회원에게 신고처리가 들어간다.

② 신고클래스 상세내용에서 신고 및 반려처리 한다.

③ 신고진행중 : 'W'
신고처리 : 'Y'
반려처리 : 'N'

FINAL PROJECT

화면구현 관리자 - 신고된 클래스 관리

①

```
<!-- 신고된 클래스 신고 처리 -->
<update id="modifyConfirmClass" parameterType="ReportClassDTO">
    UPDATE REPORT_CLASS_TBL
    SET REP_YN = 'Y'
    WHERE REP_CODE = #{ repCode }
</update>

<!-- 신고된 클래스 반려처리 -->
<update id="modifySendbackClass" parameterType="ReportClassDTO">
    UPDATE REPORT_CLASS_TBL
    SET REP_YN = 'N'
    WHERE REP_CODE = #{ repCode }
</update>
```

① 신고된 클래스 상태값을 신고 및 반려처리에 따라 'Y', 'N'으로 변경처리 한다.

② 신고클래스를 가진 회원에게 누적 경고 + 1 처리 한다.

```
<!-- 신고처리 된 클래스의 잔다 경고처리 -->
<update id="updateClassWarning">
    UPDATE SSACK_TBL S
    SET S.WARNING = S.WARNING + 1
    WHERE EMAIL = #{ EMAIL }
</update>
```


화면구현 마이페이지 - 누적경고 확인

① 새싹 누적된 경고

총 3회

날짜	사유
2021-12-16	유료
2021-12-16	유료
2021-12-16	유료

클래스 누적된 경고

총 3회

날짜	클래스	사유
2021-12-22	[초급] 40분만에 풀어보는 Sass	유료
2021-12-21	[초급] 40분만에 풀어보는 Sass	유료
2021-12-21	[초급] 40분만에 풀어보는 Sass	유료

- ① 마이페이지에서 회원과 클래스에 누적된 경고횟수를 확인한다.

화면구현 마이페이지 - 누적경고 확인

①

```
/* 회원의 경고 횟수 조회 */
int memberTotalCount = mypageService.selectMemberTotalCount(map);
System.out.println("memberTotalCount : " + memberTotalCount);

/* 회원의 경고 내역 가져오기 */
List<ReportMemberDTO> reportMember = new ArrayList<>();
reportMember = mypageService.selectReportMember(map);
for (ReportMemberDTO rm : reportMember) {
    System.out.println(rm);
}

/* 잔디회원의 클래스 경고 횟수 조회 */
int classTotalCount = mypageService.selectClassTotalCount(map);
System.out.println("classTotalCount : " + classTotalCount);

/* 잔디회원의 클래스 경고 내역 가져오기 */
List<ReportClassDTO> reportClass = new ArrayList<>();
reportClass = mypageService.selectReportClass(map);
for (ReportClassDTO rc : reportClass) {
    System.out.println(rc);
}
```

①

회원의 경고횟수와 클래스에 적용된 경고 횟수를 SELECT문을 이용하여 조회해서 가져온다.

FINAL PROJECT

화면구현 관리자 - 블랙리스트 회원 관리

① 블랙리스트 회원 조회

이메일	이름	닉네임	블랙리스트 해지 날짜	블랙리스트 해지
ram@gmail.com	임예람	람람	2022-01-26	② 해지

선택 ▼

< 1 >

이름 닉네임

블랙리스트를 해지하시겠습니까?

확인 취소

< 1 >

① 블랙리스트 처리 된 회원을 조회한다.

② 관리자 권한으로 블랙리스트 회원을 임의로 해지시킬 수 있다.

FINAL PROJECT

화면구현 관리자 - 블랙리스트 회원 관리

①

```
SELECT
  F.RNUM
, F.EMAIL
, F.NICKNAME
, F.IS_BLACK
, F.BLACK_END_DATE
, F.NAME
FROM (SELECT ROWNUM RNUM
      , D.EMAIL
      , D.NICKNAME
      , D.IS_BLACK
      , D.BLACK_END_DATE
      , D.NAME
      FROM (SELECT
              S.EMAIL
            , S.NICKNAME
            , S.IS_BLACK
            , S.BLACK_END_DATE
            , S.NAME
            FROM SSACK_TBL S
            <where>
            S.IS_BLACK = 'Y'
```

②

```
<!-- 블랙리스트 해지 -->
<update id="updateBlackCancelMember">
  UPDATE SSACK_TBL S
    SET S.IS_BLACK = 'N'
      , S.WARNING = 0
    WHERE S.EMAIL = #{ email }
</update>
```

① DB에서 IS_BLACK가 'Y'인
블랙리스트 회원을 조회한다.

② 블랙리스트 해지 시 누적 경고
횟수가 0으로 초기화되며,
IS_BLACK가 'N'으로 변경된다.

FINAL PROJECT

화면구현

마이페이지 - 수강완료클래스 관리

reym님

수강완료 클래스

수강완료 클래스

클래스관리

수강중인 클래스

수강완료 클래스

받은 클래스

구독내역

클래스 후기

설정

내정보

정보수정

비밀번호 변경

회원탈퇴

선택

수강에 완료된 클래스가 없거나, 조회된 결과가 없습니다.

[클래스 신청하기](#)

선택

Q

카테고리	클래스제목	강사/상태	수강완료	수강후기
웹 개발	웹 개발을 만들어 배우는 Next	완료	2021-12-30	수강후기작성
웹 개발	버튼스프링 3 Bootstrap 5 - 기초부터 웹 프로젝트 만들기	완료	2021-12-30	수강후기작성
데이터/사이언스	카운트 다운으로 데이터 분석 입문하기	완료	2021-12-30	수강후기작성
프론트 엔드	[프론트] 40분만에 끝내는 SaaS	완료	2021-12-30	수강후기작성
프론트 엔드	리액트 리플로우를 활용하여	완료	2021-12-30	수강후기작성

1/1

- ① 수강완료된 클래스가 있을때와 없을 때 화면이 다르게 나타나게 구현한다.

FINAL PROJECT

화면구현 마이페이지 - 수강완료클래스 관리

```
<c:if test="${ empty finishList[0].status }">
  <div class="unlist-text">
    수강이 완료된 클래스가 없거나, 조회된 결과가 없습니다.
  <br><br><br>
  <a href="${ pageContext.servletContext.contextPath }/findclass/findAllClassMain" class="finish-go"><u>클래스 신청하러가기</u></a>
</div>
</c:if>
```

```
<c:if test="${ !empty finishList[0].status }">
  <table class="ui single line table finishtable">
    <thead>
      <tr>
        <th>카테고리</th>
        <th>클래스제목</th>
        <th>강사닉네임</th>
        <th>수강완료</th>
        <th>수강후기</th>
      </tr>
    </thead>
```

- ① <c:if>를 이용하여 수강완료된 클래스가 있는지 DB에서 조회하여 리스트의 값에 따라 화면이 다르게 나타날 수 있도록 구현한다.

FINAL PROJECT

화면구현 마이페이지 - 수강완료클래스 관리

수강완료 클래스

선택

10주완성 C++ 프로그래밍

카테고리	강사	수강기간
알고리즘 - 자료구조	최희의 달인	2021-12-10-2021-12-26

①

② 클래스는 만족하셨나요? ★ ★ ★ ★ ★

작성완료 취소

① 클래스 카테고리, 강사닉네임, 수강기간 등 클래스와 관련된 정보를 모달창에 띄워준다.

② 후기 작성 시 클래스 별점을 매길수 있도록 구현한다.

FINAL PROJECT

화면구현 마이페이지 - 수강완료클래스 관리

①

```
/* 클래스 정보가 적힌 리뷰 작성 modal 창 띄워주기 */
$("#reviewbtn").on('click', function(e){
    let $reviewbtn = e.target;
    let $tr = $($reviewbtn).parent().parent();
    console.log($tr);
    console.log($($tr).children().eq(0).html());
    console.log($($tr).children().eq(4).children().eq(0).val());
    console.log("classCode : " + $($tr).children().eq(6).children().eq(0).val());
    $('<div>#reviewmodal</div>').val($($tr).children().eq(0).html());
    $('<div>#reviewmodal</div>').val($($tr).children().eq(1).html());
    $('<div>#reviewmodal</div>').val($($tr).children().eq(2).html());
    $('<div>#reviewmodal</div>').val($($tr).parent().children().eq(3).val());
    $('<div>#reviewmodal</div>').val($($tr).parent().children().eq(4).val());
    $('<div>#reviewmodal</div>').val($($tr).parent().children().eq(1).val() + "-" + $($tr).children().eq(3).html());
    $('<div>#reviewmodal</div>').fadeIn();
    e.preventDefault();
});
```

②

```
/* 별점 표시 - 기본 3개 */
$( document ).ready(function() {
    $( "#pstar a" ).click(function() {
        $(this).parent().children("a").removeClass("on");
        $(this).addClass("on").prevAll("a").addClass("on");
        var starRate = $(this).attr('id');
        console.log(starRate);
        $("#star").val(starRate);
        return false;
    });
});
```

① 스크립트를 이용하여 각 요소의 위치를 찾아서 클래스를 클릭 할때마다 각 클래스의 정보가 화면에 나타나도록 한다.

② 기본적으로 별 3개가 나타나며, 마우스 클릭 이벤트를 이용하여 1개부터 5개까지 별점을 부여할 수 있도록 한다.

FINAL PROJECT

화면구현 마이페이지 - 수강후기 관리

수강후기

①



①

수강후기가 작성되면 후기게시판으로 리다이렉트 되어, 작성된 수강후기를 확인할 수 있다.

수강완료 클래스

②

카테고리	클래스제목	강사님네임	수강완료	수강후기
프론트 엔드	라플아 라플라를 알아봐!	람잔디	2021-12-26	후기작성완료
웹 개발	웹 게임을 만들어 배우는 React	람잔디	2021-12-26	수강후기작성
프론트 엔드	만들고 비교하며 학습하는 리액트	프론트의 품격	2021-12-26	수강후기작성

②

수강후기가 작성되면 수강완료 게시판에서는 '후기작성완료' 버튼이 나타난다.

FINAL PROJECT

화면구현 마이페이지 - 수강후기 관리

①

```
<c:if test="${ finishClass.reviewDTO.rvCode eq 0 || finishClass.reviewDTO.isDelete eq 'Y' || finishClass.reviewDTO.isDelete eq null }">
<td><button class="ui button reviewBtn" id="reviewBtn">수강후기작성</button></td>
</c:if>
<c:if test="${ finishClass.reviewDTO.rvCode ne 0 && finishClass.reviewDTO.isDelete eq 'N' }">
<td style="color: #527340; font-weight: 700;">후기작성완료</td>
</c:if>
```

①

<c:if>를 이용하여 DB에서 후기
리스트를 조회하여
후기를 작성했을때와 작성하지
않았을 때 버튼이 다르게 나타날 수
있도록 구현한다.

FINAL PROJECT

화면구현 마이페이지 - 수강후기 관리



- ① 삭제 버튼을 클릭 시 삭제알럿이 나타나며, 삭제가 완료된다.

FINAL PROJECT

화면구현 마이페이지 - 수강후기 관리

①

```
/**
 * 수강 후기 삭제
 * @author 표진
 */
@PostMapping("deleteReview")
public String deleteReview(@RequestParam int rvCode, HttpSession session, RedirectAttributes rttr) {

    // 가져온 파라미터 값 확인하기
    System.out.println("rvCode : " + rvCode);

    // 삭제 성공여부 확인하기
    if (mypageService.deleteReview(rvCode) > 0) {
        rttr.addFlashAttribute("deleteMessage", "삭제가 성공했습니다.");
    } else {
        rttr.addFlashAttribute("deleteMessage", "삭제에 실패했습니다.");
    }

    return "redirect:/mypage/review";
}
```

①

DB에서 리뷰코드를 조회하여
수강후기게시판에서 삭제할 수
있도록 구현한다.

```
<!-- 수강후기 삭제 -->
<update id="deleteReview" parameterType="_int">
    DELETE FROM CLASS_REVIEW_TBL
    WHERE REVIEW CODE = #{ rvCode }
</update>
```

파이널 프로젝트 후기

- 마지막 프로젝트를 진행하며, 마지막이라는 생각에 프로젝트 기간동안 모든 시간에 열정을 쏟아 했던 것 같다. 하지만, 생각처럼 기능구현이 이루어지지 않아 좌절했던 시간도 있었지만 강사님과 또 함께하는 팀원들의 도움으로 끝까지 포기하지 않고 해낼 수 있었다.

어려웠던 점

- 수강완료페이지에서 수강완료클래스 목록이 나타나고 후기를 작성했을 때와 안했을때 버튼을 다르게 구현하는 것이 있었는데, 쿼리문에 어떻게 조건을 주어도 원하는대로 나타나지 않아 어려움이 있었는데, 그래도 끝까지 포기하지 않고 다양하게 쿼리문을 작성해본 끝에 기능 구현에 성공할 수 있었다.
- 신고된 회원을 클릭했을 때 모달창으로 상세내용이 나타나게 하는 기능이 있었다. 세미때 요소를 찾는 것을 해보았기에 간단히 할 수 있을거라고 생각했는데, 생각보다 요소를 찾는 부분에서 많은 시간을 소비했다. 몇시간을 생각하고 작성한 끝에 각 요소를 찾는 것은 성공했지만 프로젝트가 끝나고 계속적으로 요소를 찾아보는 공부를 해야겠다는 생각이 들었다.

감사합니다