

황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text" value="uesr05"/>	사용할 수 있는 아이디입니다.
비밀번호	<input type="password" value="*****"/>	
비밀번호 확인	<input type="password" value="*****"/>	
닉네임	<input type="text" value="수아"/>	사용할 수 있는 닉네임입니다.
이름	<input type="text" value="이수아"/>	
생년월일	<input type="text" value="880725"/>	
휴대전화번호	<input type="text" value="010-2834-1236"/>	
이메일주소	<input type="text" value="sunea24e@naver.com"/>	인증번호 전송 사용할 수 있는 이메일입니다.
인증번호	<input type="text" value="jj9AqJbM6Y"/>	인증하기 인증번호가 메일로 전송 되었습니다.
주소	<input type="text" value="서울 강남구 논현로139길 7"/> <input type="text" value="1층"/>	주소검색
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여	
신분	<input checked="" type="radio"/> 학생 <input type="radio"/> 선생님	

회원가입

취소

```
$('#mId').focusout(function() {  
  
    let findStr = $('#mId').val(); // input_id에 입력되는 값  
  
    if ($('#mId').val().length < 6) { //mId 6자 보다 작으면  
        $("#checkId").html('사용할 수 없는 아이디입니다.');
```

1. 비동기 방식 ajax 로 아이디 데이터 가지고 DB 데이터 확인작업 (닉네임, 이메일 동일)

황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text" value="uesr05"/>	사용할 수 있는 아이디입니다.
비밀번호	<input type="password" value="*****"/>	
비밀번호 확인	<input type="password" value="*****"/>	
닉네임	<input type="text" value="수아"/>	사용할 수 있는 닉네임입니다.
이름	<input type="text" value="이수아"/>	
생년월일	<input type="text" value="880725"/>	
휴대전화번호	<input type="text" value="010-2834-1236"/>	
이메일주소	<input type="text" value="sunea24e@naver.com"/>	<input type="button" value="인증번호 전송"/> 사용할 수 있는 이메일입니다.
인증번호	<input type="text" value="jj9AqJbM6Y"/>	<input type="button" value="인증하기"/> 인증번호가 메일로 전송 되었습니다.
주소	<input type="text" value="서울 강남구 논현로139길 7"/> <input type="text" value="1층"/>	<input type="button" value="주소검색"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여	
신분	<input checked="" type="radio"/> 학생 <input type="radio"/> 선생님	

회원가입

취소

```
//이메일 형식 확인-ok
$("#mEmail").on("input",function() {
    //
    var EmailRegEx = /^[A-Za-z0-9_]+[A-Za-z0-9]*@[1]{1}[A-Za-z0-9]+[A-Za-z0-9]*[.]{1}[A-Za-z]{1,3}$/; // 이메일 정규식
    var EmailVal = $("#mEmail").val();
    if (!EmailRegEx.test(EmailVal)) { // mEmail 이메일 정규식 맞는지 체크
        console.log("이메일 형식을 바르게 입력해주세요. console");
        emailChk = true;
        $("#checkmEmail_info").html("이메일 형식이 맞지 않습니다.");
        $("#checkmEmail_info").attr('color', 'red');
    } else {
        emailChk = false;
        $("#checkmEmail_info").html("");
    }
});

//동료체크 - 이메일 -ok
$("#mEmail").focusout(function() {
    let findStr = $("#mEmail").val(); // input_id에 입력되는 값

    $.ajax({
        type : 'post',
        async : false, //false가 기본값임 - 비동기
        url : 'findStr', //MemberFindStrAjaxController.java 주소 이동
        dataType : 'text', //리턴값 형식
        data : {
            str : findStr,
            type : 'mEmail'
        },
        success : function(data, textStatus) {

            if (data == 'not-usable') {
                $("#checkmEmail_info").html('사용할 수 없는 이메일입니다. ');
                $("#checkmEmail_info").attr('color', 'red');
                emailChk = true;
            } else {
                $("#checkmEmail_info").html('사용할 수 있는 이메일입니다. ');
                $("#checkmEmail_info").attr('color', 'green');
                emailChk = false;
            }
        },
        error : function(data, textStatus) {
            console.log('error');
        }
    }) //ajax
});
```

2. 비동기 방식 ajax 이메일 중복 구현, 이메일 체크하는 유효성 부분을 구현

황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text" value="숫자,영문대소문자 6~15자리"/>
비밀번호	<input type="text" value="8자이상 영문대소문자, 숫자, 특수문자가 각각 1개이상"/>
비밀번호 확인	<input type="text" value="비밀번호 확인"/>
닉네임	<input type="text" value="한글, 영문, 특수문자를 포함한 2 ~ 12글자"/>
이름	<input type="text"/>
생년월일	<input type="text" value="예) 210210"/>
휴대전화번호	<input type="text" value="예) 010-2222-3333"/>
이메일주소	<input type="text" value="예) id@domain.com"/>
인증번호	<input type="text"/>
주소	<input type="text" value="상세주소 입력해 주세요"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
신분	<input checked="" type="radio"/> 학생 <input type="radio"/> 선생님

인증번호 전송

인증하기

주소검색

회원가입

취소

```
/* 인증번호 이메일 전송 */
$("#bnt_Cert1").click(function(){
    var email = $("#mEmail").val(); // 입력한 이메일
    $("#certEmail").val(email);

    $.ajax({
        type : 'post',
        async : false, //false가 기본값임 - 비동기
        url : 'certEmail',
        dataType : 'text',
        data : {
            mEmail: email
        },
        success : function(data, textStatus) {
            console.log('data: '+data);
            $("#certChk").val(data);
            $("#cert_info").html("인증번호가 메일로 전송 되었습니다.");
            $("#cert_info").attr('color', 'green');
        },
        error : function(data, textStatus) { //자바 익셉션 시 ajax 실패시
            console.log('ajax error');
            $("#certChk").val("");
            $("#cert_info").html("인증번호가 메일로 전송 실패되었습니다. ");
            $("#cert_info").attr('color', 'red');
        }
    });
});
```

3. 버튼 클릭시 인증번호 이메일로 전송

dlfdj@naver.com

인증번호 전송 사용할 수 있는 이메일입니다.

Lr6098B270

인증하기 인증번호가 메일로 전송 되었습니다.

황선애 - 회원가입(학생& 선생님)

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("doPost - Checkemail 이메일 인증 ");

    // ajax로 들어옴
    PrintWriter out = response.getWriter();
    String mEmail = request.getParameter("mEmail");

    // 인증 번호 생성기
    StringBuffer temp = new StringBuffer();
    Random rnd = new Random();
    for (int i = 0; i < 10; i++) {
        int rIndex = rnd.nextInt(3);
        switch (rIndex) {
            case 0:
                // a-z
                temp.append((char) ((int) (rnd.nextInt(26)) + 97));
                break;
            case 1:
                // A-Z
                temp.append((char) ((int) (rnd.nextInt(26)) + 65));
                break;
            case 2:
                // 0-9
                temp.append((rnd.nextInt(10)));
                break;
        }
    }
    String random = temp.toString();
    System.out.println("random " + random);

    // mail server 설정
    String host = "smtp.gmail.com";
    String user = "testminkyotest@gmail.com"; // 자신의 구글 계정
    String password = "wjaldsry1212"; // 자신의 구글 패스워드

    // 메일 받을 주소
    //String to_email = m.getEmail();
    String to_email = mEmail;
    //String to_email = "sunea24@naver.com";
```

```
// SMTP 서버 정보를 설정한다.
Properties props = new Properties();
props.put("mail.smtp.host", host);
props.put("mail.smtp.port", 465);
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.ssl.enable", "true");
props.put("mail.smtp.starttls.enable", "true");
props.put("mail.debug", "true");

// 이메일 보내기
Session session = Session.getDefaultInstance(props, new javax.mail.Authenticator() {
    protected PasswordAuthentication getPasswordAuthentication() {
        return new PasswordAuthentication(user, password);
    }
});
try {
    MimeMessage msg = new MimeMessage(session);
    msg.setFrom(new InternetAddress(user, "VISITOR"));

    InternetAddress to = new InternetAddress(to_email);
    msg.setRecipient(Message.RecipientType.TO, to);
    // 메일 제목
    msg.setSubject("홈페이지 방문해주셔서 감사합니다. 인증번호 발송 메일입니다.");
    // 메일 내용
    msg.setText("인증번호 : " + random + "입니다." + "해당 인증번호 확인란에 기입하여 주세요");

    Transport.send(msg);
    System.out.println("인증번호 이메일 전송");

} catch (Exception e) {
    e.printStackTrace(); // TODO: handle exception
}

HttpSession saveKey = request.getSession();
saveKey.setAttribute("AuthenticationKey", random);

out.print(random);
out.flush();
out.close();
}
```

인증번호 생성 후 이메일 보내기
- 회원가입 및 비밀번호 재설정부분 동일하게 구현

황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text"/>
비밀번호	<input type="password"/>
비밀번호 확인	<input type="password"/>
닉네임	<input type="text"/>
이름	<input type="text"/>
생년월일	<input type="text"/>
휴대전화번호	<input type="text"/>
이메일주소	<input type="text"/>
인증번호	<input type="text"/>
주소	<input type="text"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
신분	<input checked="" type="radio"/> 학생 <input type="radio"/> 선생님

인증번호 전송

인증하기

주소검색

회원가입

취소

```
$("#bnt_Cert2").click(function(){  
    if( $("#certChk").val() != "" && $("#certChk").val() == $("#cert").val()){  
        alert("인증 되었습니다");  
  
        //$("#cert_info2").html("인증 되었습니다.");  
        certChk = false;  
    }else{  
        alert("인증번호를 다시 확인해주세요.");  
        //$("#cert_info2").html("인증번호를 확인해주세요.");  
        certChk = true;  
    }  
});
```

localhost:8081 내용:
인증번호를 다시 확인해주세요.

확인

선

localhost:8081 내용:
인증 되었습니다

확인

황선애 - 회원가입(학생& 선생님)



인증번호 전송 사용할 수 있

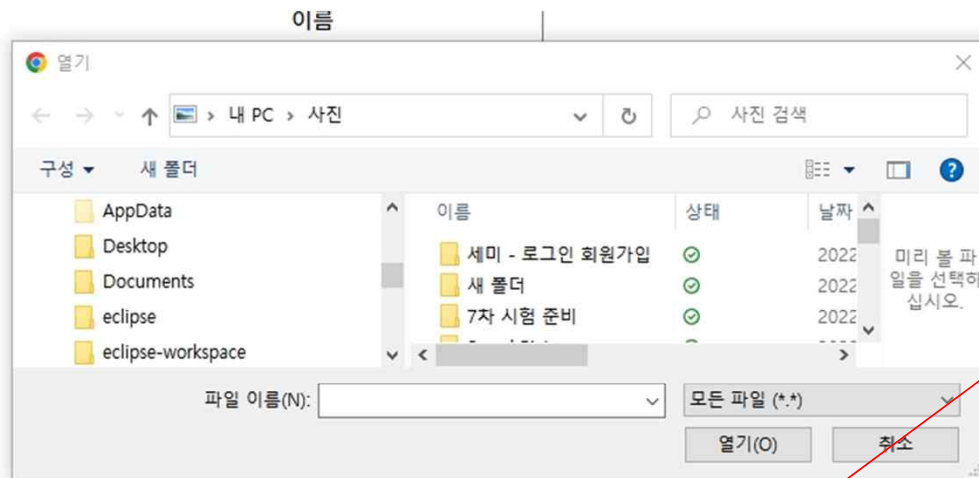
인증하기 인증번호가

주소검색

```
5 function open_Postcode() { //다음 카카오 주소찾기
7     new daum.Postcode({
3         oncomplete : function(data) {
9             console.log(data);
9             // 우편번호와 주소 정보를 해당 필드에 넣는다.
1            document.getElementById("mAddress1").value = data.roadAddress;
2
3            }
4        }).open();
5    }
```

4.주소 API

황선애 - 회원가입(학생& 선생님)



성별 ☒ 남 ☐ 여

신분 ☐ 학생 ☒ 선생님

증명서류 선택된 파일 없음

회원가입

취소

```
//신분 선생님 클릭시 증명서 첨부 활성화 - ok
$("#fileupload").hide(); // 증명서 첨부 숨기기

$("#radioT").click(function() { // 신분 선생님 radioT 클릭시 화면 보이기
    $("#fileupload").show();
});

$("#radioS").click(function() { // 신분 학생 radioS 클릭시 화면 숨기기
    $("#fileupload").hide();
});
```

주소검색

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
    System.out.println("doPost - join");
    // 파일 저장 경로
    String fileSavePath = "upload";
    // 파일 크기 10M 제한
    int uploadSizeLimit = 10 * 1024 * 1024;
    String encType = "UTF-8";
    ServletContext context = getServletContext();
    String uploadPath = context.getRealPath(fileSavePath);

    // 업로드 할 폴더 존재여부확인 - 없다면 생성
    File path = new File(uploadPath);
    if(!path.exists()) {
        path.mkdirs();
    }
    System.out.println("uploadPath:" + uploadPath);

    MultipartRequest multi = new MultipartRequest(request, // request 객체
        uploadPath, // 서버 상 업로드 될 디렉토리
        uploadSizeLimit, // 업로드 파일 크기 제한
        encType, // 인코딩 방법
        new DefaultFileRenamePolicy() // 동일 이름 존재 시 새로운 이름 부여 방식
    ); // 파일 저장 완료
```

황선애 - 회원가입(학생& 선생님)

아이디	<input type="text" value="uesr"/>	사용할 수 없는 아이디입니다.
비밀번호	<input type="password" value="....."/>	패스워드 형식이 맞지 않습니다.
비밀번호 확인	<input type="password" value="....."/>	비밀번호가 다릅니다. 다시 확인해 주세요.
닉네임	<input type="text" value="민"/>	닉네임은 2자리 이상입니다.

```
// 아이디 형식 확인 -ok
$("#mId").on("input", function() {
    var idRegEx = /^[0-9a-zA-Z]{6,15}$/; //숫자 대소문자 6~15자리 정규식
    var idVal = $("#mId").val();
    if (!idRegEx.test(idVal)) { //idVal값이 정규식에 맞는지 체크
        console.log("아이디 형식을 바르게 입력해주세요.");
        idChk = true;
        $("#checkId").html("아이디 형식이 맞지 않습니다.");
        $("#checkId").attr('color', 'red');
    } else {
        idChk = false;
        $("#checkId").html("");
    }
});
```

```
//패스워드 형식 확인 -ok
$("#pw").on("input",function() {
    // 8자리이상 대소문자, 숫자, 특수문자가 각각 1개 이상 (패스워드 체크시 활용)
    var passwordRegEx = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[!@#%&*~]).{8,}$;
    var passwordVal = $("#pw").val();
    if (!passwordRegEx.test(passwordVal)) { //passwordVal값이 정규식에 맞는지 체크
        console.log("패스워드 형식을 바르게 입력해주세요.");
        pwdChk = true;
        $("#pwd_info").html("패스워드 형식이 맞지 않습니다.");
        $("#pwd_info").attr('color', 'red');
    } else {
        pwdChk = false;
        $("#pwd_info").html("");
    }
});
```

5. 회원가입 유효성 검사

- 아이디, 비밀번호, 닉네임, 생년월일, 핸드폰 등 동일한 형식으로 아이디, 비밀번호 형식외 생략

6. 비밀번호 재 확인

```
function check_pw() { //비밀번호 확인 - ok
    var p = document.getElementById('pw').value;
    var p_cf = document.getElementById('pw_cf').value;

    if (p != p_cf) {
        document.getElementById('pw_check_msg').innerHTML = "비밀번호가 다릅니다. 다시 확인해 주세요.";
        $("#pw_check_msg").attr('color', 'red');
    } else {
        document.getElementById('pw_check_msg').innerHTML = "";
    }
    if (p_cf == "") {
        document.getElementById('pw_check_msg').innerHTML = "";
    }
}
```


황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text" value="uesr05"/>	사용할 수 있는 아이디입니다.
비밀번호	<input type="password" value="*****"/>	
비밀번호 확인	<input type="password" value="*****"/>	
닉네임	<input type="text" value="수아"/>	사용할 수 있는 닉네임입니다.
이름	<input type="text" value="이수아"/>	
생년월일	<input type="text" value="880725"/>	
휴대전화번호	<input type="text" value="010-2834-1236"/>	
이메일주소	<input type="text" value="sunea24e@naver.com"/>	<input type="button" value="인증번호 전송"/> 사용할 수 있는 이메일입니다.
인증번호	<input type="text" value="jj9AqJbM6Y"/>	<input type="button" value="인증하기"/> 인증번호가 메일로 전송 되었습니다.
주소	<input type="text" value="서울 강남구 논현로139길 7"/> <input type="text" value="1층"/>	<input type="button" value="주소검색"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여	
신분	<input checked="" type="radio"/> 학생 <input type="radio"/> 선생님	

```
//필수 입력정보 입력되었는지 확인하는 함수
function checkValue() {
    //아이디 6자 이상
    if ($("#mId").val().length < 6) {
        alert("아이디는 6자 이상입니다.");
        return;
    }
    //아이디
    if (idChk) {
        alert("아이디를 확인하세요.");
        return;
    }
    if ($("#pw").val().length == 0) {
        alert("비밀번호를 입력하세요.");
        return;
    }
    if ($("#pw").val().length < 8) {
        alert("비밀번호는 8자 이상입니다.");
        return;
    }
    if (pwdChk) {
        alert("비밀번호를 확인하세요.");
        return;
    }
    //비밀번호== 비밀번호확인 일치 하지 확인
    if (document.getElementById('pw').value != document
        .getElementById('pw_cf').value) {
        alert("비밀번호가 일치하지 않습니다. 확인해 주세요");
        document.getElementById('pw_cf').focus();
        return;
    }
}
```

7. 회원가입 input 박스 입력사항 체크

황선애 - 회원가입(학생& 선생님)

회원가입

아이디	<input type="text" value="uesr0"/>	사용할 수 없는 아이디입니다.
비밀번호	<input type="password" value="....."/>	패스워드 형식이 맞지 않습니다.
비밀번호 확인	<input type="password" value="....."/>	비밀번호가 다릅니다. 다시 확인해 주세요.
닉네임	<input type="text" value="민초"/>	사용할 수 없는 닉네임입니다.
이름	<input type="text" value="이수아"/>	
생년월일	<input type="text" value="88072"/>	생년월일 형식이 맞지 않습니다.
휴대전화번호	<input type="text" value="010-2834-123"/>	phone 형식을 바르게 입력해주세요.
이메일주소	<input type="text" value="sunea24@naver.com"/>	<input type="button" value="인증번호 전송"/> 사용할 수 없는 이메일입니다.
인증번호	<input type="text" value="jj9AqJbM6Y"/>	<input type="button" value="인증하기"/> 인증번호가 메일로 전송되었습니다.
주소	<input type="text" value="서울 강남구 논현로139길 7"/> <input type="text" value="1층"/>	<input type="button" value="주소검색"/>
성별	<input type="radio"/> 남 <input checked="" type="radio"/> 여	
신분	<input type="radio"/> 학생 <input checked="" type="radio"/> 선생님	
증명서류	<input type="button" value="파일 선택"/> 선택된 파일 없음	

회원가입

취소

```

if (emailChk) {
    alert("이메일을 확인하세요.");
    return;
}
if (phoneChk) {
    alert("핸드폰번호를 확인하세요.");
    return;
}
//생년월일 미입력시 확인
if ($("#mBirth").val().length < 6) {
    alert("생년월일을 제대로 입력하세요.");
    return;
}
if (birthChk) {
    alert("생년월일을 확인하세요.");
    return;
}
//주소 미입력시 확인
if (document.getElementById('mAddress1').value == "") {
    alert("주소를 입력해주세요.");
    document.getElementById('postcode_button').focus();
    return;
}
//성별 라디오 미 선택시 확인
if ($("#input[name=genderFm]:checked").length == 0) {
    alert("성별을 바르게 입력해주세요.");
    return;
}
//신분구분 미 선택시 확인
if ($("#input[name=roleSt]:checked").length == 0) {
    alert("신분을 바르게 입력해주세요.");
    return;
}
if(certChk){
    alert("이메일인증을 해주세요.");
    return;
}
if($("#mEmail").val() != $("#certEmail").val()){
    alert("인증했던 이메일이 아닙니다. 다시 인증 해주세요.");
    return;
}
chkVal = true;
    
```

7. 회원가입 input 박스 입력사항 체크

황선애 - 회원가입(학생& 선생님)

```
//회원가입 버튼 클릭시
$('#btn_register').on("click", function() {
    checkValue(); //회원가입 형식 맞게 들어갔는데 체크하는 함수
    if (chkVal) {
        var frm = document.frm_register;
        frm.action = "join";
        frm.method = 'post';
        frm.submit();
    }
});
```

8. 회원가입시 버튼 클릭시 post 방식

- 데이터 가지고 Service > Dao 이동 > DB에 회원 insert 하고 Dao > Service > 학생 No 확인 후 > 학생 No insert > 연필 기본insert 완료

```
// 기존 code에서 multi로 바꿔 줘여함.
MemberVo memberVo = new MemberVo();

System.out.println("client파일명"+multi.getParameter("joinupload")); // 업로드한 클라이언트 pc에 있던 파일명
String fileName = multi.getFilesystemName("joinupload"); // 서버에 업로드한 파일명
System.out.println("서버 저장 파일명" + fileName);
String mCertificate = fileSavePath + "/" + fileName;
memberVo.setmCertificate(mCertificate);

memberVo.setmId(multi.getParameter("mId"));
memberVo.setmPw(multi.getParameter("mPw"));
memberVo.setmNickname(multi.getParameter("mNickName"));
memberVo.setmName(multi.getParameter("mName"));
memberVo.setmBirth(multi.getParameter("mBirth"));
memberVo.setmPhone(multi.getParameter("mPhone"));
memberVo.setmEmail(multi.getParameter("mEmail"));
memberVo.setmAddress(multi.getParameter("mAddress1") + " " + multi.getParameter("mAddress2"));
memberVo.setGenderFm(multi.getParameter("genderFm"));
memberVo.setRoleSt(multi.getParameter("roleSt"));

System.out.println("memberVo: " + memberVo);
PrintWriter out = response.getWriter();
try {
    int i = new MemberService().insertMember(memberVo);

    if (i < 1) {
        // 회원가입페이지 이동
        out.print("<script>alert('가입 실패 다시 시도해주세요.');
```

황선애 - 회원가입(학생& 선생님)

```
// 회원가입 ok
public int insertMember(MemberVo vo) {
    int result = 0;
    Connection conn = getConnection();
    setAutocommit(conn, false);
    System.out.println("회원가입 시작");
    result = dao.insertMember(conn, vo);
    System.out.println("member 테이블 insert 결과 : " + result);
    if (result < 1) {
        rollback(conn);
        close(conn);
        return 0;
    }
    System.out.println("member 테이블 insert 결과 : " + result);
    if (vo.getRoleSt().equals("S")) { // 받은 값이 S이면
        System.out.println("학생입니다.");
        String sNo = new StudentService().readStudentCheck();// readStudent
        if (sNo == null) { // 수정 요청으로 수정
            sNo = "S1";
        } else {
            int no = Integer.parseInt(sNo) + 1;
            sNo = "S".concat(no + "");
        }
        StudentVo sVo = new StudentVo(); //
        sVo.setmId(vo.getmId());
        sVo.setsNo(sNo);
        result = stuDao.insertStudent(conn, sVo);
        if (result < 1) {
            rollback(conn);
            close(conn);
            return 0;
        }
    }
}
```

```

    if (vo.getRoleSt().equals("T")) {
        System.out.println("선생님 입니다.");
        result = teacherDao.insertTeacherInit(conn, vo.getmId());
        if (result < 1) {
            rollback(conn);
            close(conn);
            return 0;
        }
    }
    if (result == 1) {
        // 연필 테이블에 기본 insert
        result = penDao.insertPencilInit(conn, vo.getmId());

        if (result < 1) {
            rollback(conn);
            close(conn);
            return 0;
        }
    }
    commit(conn);
    close(conn);
    return result;
}
```

황선애 - 회원가입(학생& 선생님)

```
//회원가입
public int insertMember(Connection conn, MemberVo vo) {
    int result = 0;

    String sql =
        "INSERT INTO THDUO.MEMBER"
        +"(M_ID, M_PW, M_NAME, M_NICKNAME, M_BIRTH, M_ADDRESS, M_PHONE, M_EMAIL, GENDER_FM, ROLE_ST, M_DATE)"
        +"VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, SYSTIMESTAMP)";

    try {
        pstmt = conn.prepareStatement(sql);

        pstmt.setString(1, vo.getMId());
        pstmt.setString(2, vo.getMpw());
        pstmt.setString(3, vo.getMName());
        pstmt.setString(4, vo.getMNickname());
        pstmt.setString(5, vo.getMBirth());
        pstmt.setString(6, vo.getMAddress());
        pstmt.setString(7, vo.getMPhone());
        pstmt.setString(8, vo.getMEmail());
        pstmt.setString(9, vo.getGenderFm());
        pstmt.setString(10, vo.getRoleSt());

        result = pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(pstmt);
    }

    return result;
}
```

```
// 회원가입시 학생S_no insert
public int insertStudent(Connection conn, StudentVo vo) {
    int result = 0;
    String sql = "INSERT INTO THDUO.MEMBER_STUDENT" + "(M_ID, S_NO)" + "VALUES(?, ?)";
    try {
        pstmt = conn.prepareStatement(sql);

        pstmt.setString(1, vo.getMId());
        pstmt.setString(2, vo.getSNo());

        result = pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(pstmt);
    }

    return result;
}
```


황선애 - 로그인(일반회원 & 관리자)

관리자 및 회원 따라 로그인 페이지 다름

로그인

☒ 관리자 ☐ 일반회원

admin

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

로그인

☐ 관리자 ☒ 일반회원

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

[illegible]

```
<script type="text/javascript">
$(function() {
    $('#btn_Yes').on("click", function() {
        var frm = document.frm_login;

        if ($('#radio[name="mode"]:checked').val() == "member") {
            frm.action = "login";
            frm.method = 'post';
            frm.submit();
        }else{
            frm.action = "adminLogin";
            frm.method = 'post';
            frm.submit();
        }
    });
});
</script>
```

- 로그인 버튼 클릭시 radio 구분으로 회원, 관리자 로그인 post로 각자 다른 서블릿 이동

황선애 - 로그인(일반회원 & 관리자)

로그인

☐ 관리자 ☒ 일반회원

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("doPost - login");
    // 로그인시 id, pw 받기
    String mId = request.getParameter("id");
    String pw = request.getParameter("pwd");
    System.out.println(mId);
    System.out.println(pw);
    MemberVo vo = new MemberService().readFindId(mId);

    if (vo == null) {
        //로그인시 id 확인 작업
        request.setAttribute("login_msg", "아이디가 없습니다.");
        request.getRequestDispatcher("WEB-INF/view/member/login.jsp").forward(request, response);
        return;
    }

    // 로그인 id 와 pw 확인 시 없으면
    MemberVo vo2 = new MemberService().login(mId, pw);
    if (vo2 == null) {
        request.setAttribute("login_msg", "비밀번호가 틀렸습니다.");
        request.getRequestDispatcher("WEB-INF/view/member/login.jsp").forward(request, response);
        return;
    }
}
```

1. doPost 메소드 (일반회원) 이동
2. ID,PW 받아와 DB로 이동해 ID 있는지 확인작업 후 불일치 시 오류 메세지 출력 후 리턴
3. ID값 일치하면 ID,PW 가지고 DB 이동 데이터값 확인 후 불일치시 오류 메세지 출력

황선애 - 로그인(일반회원 & 관리자)

로그인

☐ 관리자 ☒ 일반회원

비밀번호가 틀렸습니다.

로그인

아이디 찾기 | 비밀번호 찾기 | 회원가입

```
<div class="card-body">
  <input type="text" name="id" class="form-control" placeholder="아이디" required="required"> <br>
  <input type="password" name="pwd" class="form-control" placeholder="비밀번호" required="required"><br>
  <p id="check" class="check"><%request.getAttribute("login_msg"); %>${login_msg}</p><br>
  <button type="button" id="btn_Yes" class="form-control">로그인</button>
</div>
```

1. DB 데이터 확인 후 불일치시 오류 메시지 표시

```
<% request.getSession().removeAttribute("msgLogin"); %>
<script>
$(function(){
  var msgLoginVal = '${msgLogin}';
  if(msgLoginVal != null && msgLoginVal != ""){
    alert('${msgLogin}');
    location.replace("login");
  }
});
</script>
```

```
<div class="links">
  <a href="findId">아이디 찾기 &nbsp;&nbsp;&nbsp;</a>
  <a href="findPw">&nbsp;&nbsp;&nbsp;비밀번호 찾기 &nbsp;&nbsp;&nbsp;</a>
  <a href="join">| &nbsp;&nbsp;&nbsp;회원가입</a>
</div>
```

2. a 태그 각 페이지 이동

황선애 - 로그인(일반회원 & 관리자)

로그인

☐ 관리자 ☒ 일반회원

로그인

아이디 찾기 | 비밀번호 찾기 | 회원가입

```
// 로그인시 id 확인작업
public MemberVo readFindId(Connection conn, String mId) {
    MemberVo retVo = null;
    String sql = "select * from member where M_ID = ?";
    try {
        pstmt = conn.prepareStatement(sql); // 연결
        pstmt.setString(1, mId); //
        rs = pstmt.executeQuery(); // 실행
        if (rs.next()) {
            retVo = new MemberVo();
            retVo.setmId(rs.getString("m_Id"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(rs);
        close(pstmt);
    }
    return retVo;
}

} else {
    // 로그인 id와 pw 맞으면 메인이동
    request.getSession().setAttribute("ssMV", vo2);
    String mNickname = vo2.getmNickname();
    System.out.println("받은 알람 서비스 호출 mNickname:" + mNickname);
    ArrayList<AlarmVo> result = new AlarmService().receiveIdList(mNickname);
    System.out.println("리스트 결과:" + result);
    request.getSession().setAttribute("nicknameList", result);
    response.sendRedirect(request.getContextPath() + "/"); // 절대경로를 의미하며 -context root가 없음.
    return;
}
}
```

4.DB데이터 ID 데이터 확인

6. DB 데이터 ID,PW 확인 후 일치하면 로그인 후 메인이동

황선애 - 로그인(일반회원 & 관리자)

```
// 회원 로그인
public MemberVo login(Connection conn, String mId, String pw) {
    MemberVo retVo = null;
    String sql = "SELECT m.M_ID, M_PW, M_NAME, M_NICKNAME, M_BIRTH, M_ADDRESS, M_PHONE, M_EMAIL, GENDER_FM, ROLE_ST, M_DATE,"
        + " M_CERTIFICATE, M_ALARM_YN, t_no,s_no, T_RECRUIT_YN "
        + "FROM MEMBER m left outer join t_profile t on m.m_id = t.m_id full outer join member_student ms on m.m_id = ms.m_id "
        + "where m.M_ID=? and m_pw=?";

    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, mId);
        pstmt.setString(2, pw);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            retVo = new MemberVo();

            retVo.setmId(rs.getString("m_Id"));
            retVo.setmPw(rs.getString("m_pw"));
            retVo.setmName(rs.getString("M_NAME"));
            retVo.setmNickname(rs.getString("m_Nickname"));
            retVo.setmBirth(rs.getString("m_Birth"));
            retVo.setmAddress(rs.getString("m_Address"));
            retVo.setmPhone(rs.getString("m_phone"));
            retVo.setmEmail(rs.getString("m_email"));
            retVo.setGenderFm(rs.getString("gender_Fm"));
            retVo.setRoleSt(rs.getString("role_St"));
            retVo.setmDate(rs.getTimestamp("m_Date"));
            retVo.setmCertificate(rs.getString("M_CERTIFICATE"));
            retVo.setmAlarmYn(rs.getString("M_ALARM_YN"));
            retVo.settNo(rs.getString("t_no"));
            retVo.settRecruitYn(rs.getString("t_Recruit_Yn"));
            retVo.setsNo(rs.getString("s_no"));

            System.out.println(retVo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(rs);
        close(pstmt);
    }
    return retVo;
}
```

5. DB 데이터 ID,PW 확인 후 로그인

로그인

☐ 관리자 ☒ 일반회원

로그인

[아이디 찾기](#) | [비밀번호 찾기](#) | [회원가입](#)

황선애 - 아이디찾기

아이디 찾기

아이디찾기

비밀번호 찾기 | 로그인 | 회원가입

```
<div>
  <p class="card-title" style="color:#FA9D00;">아이디 찾기</p>
</div>
<div class="card-body">
  <input type="text" name="mName" id="mName" class="form-control" placeholder="이름" required autofocus><br>
  <input type="email" name="mEmail" id="mEmail" class="form-control" placeholder="이메일" required><br>
  <p class="check" id="check">${check}</p><br>
  <button id="btn-Yes" class="form-control type="button">아이디찾기</button>
</div>
```

```
<script type="text/javascript">
$("#btn-Yes").click(function(){
  $.ajax({
    url:"findId",
    type:"post",
    data:{mName: $("#mName").val(),
      mEmail: $("#mEmail").val(), },
    success:function(result){
      if(result == ""){
        //
        $("#id_content").html("<p>찾는 아이디가 없습니다.</p>");
      } else{
        $("#id_content").html("<p>회원님의 아이디는 " +result+ "입니다.</p>");
      }
      $(".modal").show();
    },
    error:function(){
    }
  });
});
```

1. 아이디찾기 ajax post방식

황선애 - 아이디찾기

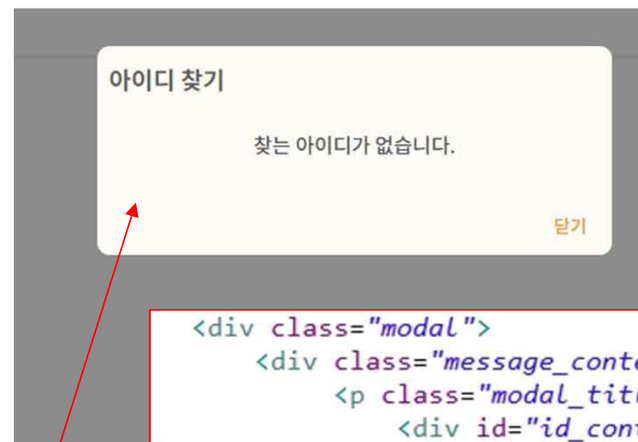
```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    // ajax로 들어옴  
    System.out.println("doPost - findId ajax로 들어옴");  
    PrintWriter out = response.getWriter();  
    // 아이디 찾기  
    String mEmail = request.getParameter("mEmail");  
    String mName = request.getParameter("mName");  
  
    Enumeration params = request.getParameterNames();  
    while(params.hasMoreElements()) {  
        String name = (String) params.nextElement();  
        System.out.print(name + " : " + request.getParameter(name) + " ");  
    }  
    System.out.println(mEmail);  
    System.out.println(mName);  
  
    MemberVo vo = new MemberService().readFindId(mEmail, mName);  
    System.out.println(vo );  
  
    if(vo == null) {  
        out.print("");  
    } else {  
        out.print(vo.getId());  
    }  
    out.flush();  
    out.close();  
}  
}
```

```
// id찾기-ok  
public MemberVo readFindId(String mEmail, String mName) {  
    MemberVo retVo = null;  
    Connection conn = getConnection();  
    retVo = dao.readFindId(conn, mEmail, mName);  
    close(conn);  
    return retVo;  
}
```

```
// id찾기  
public MemberVo readFindId(Connection conn, String mEmail, String mName) {  
    MemberVo retVo = null;  
    String sql = "select * from member where M_EMAIL = ? and M_NAME =?";  
    try {  
        pstmt = conn.prepareStatement(sql);  
        pstmt.setString(1, mEmail);  
        pstmt.setString(2, mName);  
        rs = pstmt.executeQuery();  
        if (rs.next()) {  
            retVo = new MemberVo();  
            retVo.setmId(rs.getString("m_Id"));  
        }  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        close(rs);  
        close(pstmt);  
    }  
    return retVo;  
}
```

1. 아이디찾기 ajax post방식
2. 이름, 이메일 데이터 가지고 MemberService 이동
3. Dao에서 ID 가지고 DB 확인 후 리턴

황선애 - 아이디찾기



4. DB 데이터 결과값 모달창으로 출력
- 아이디찾기, 비밀번호 재설정 부분
동일한 모달창 구현

```
<script type="text/javascript">
$("#btn-Yes").click(function(){
    $.ajax({
        url:"findId",
        type:"post",
        data:{mName: $("#mName").val(),
            mEmail: $("#mEmail").val(), },
        success:function(result){
            if(result == ""){
                //
                $("#id_content").html("<p>찾는 아이디가 없습니다.</p>");
            } else{
                $("#id_content").html("<p>회원님의 아이디는 " +result+ "입니다.</p>");
            }
            $(".modal").show();
        },
        error:function(){
        }
    });
});
```

```
<div class="modal">
    <div class="message_content">
        <p class="modal_title">아이디 찾기</p>
        <div id="id_content">

        </div>
        <div class="message_send">
            <button type="button" class="btn_close">닫기</button>
        </div>
    </div>
```

황선애 - 아이디찾기 & 비밀번호 찾기

아이디 찾기

이름을 입력해주세요.

아이디찾기

비밀번호 찾기 | 로그인 | 회원가입

비밀번호 찾기

아이디를 입력해주세요.

비밀번호 재설정

아이디 찾기 | 로그인 | 회원가입

1. 아이디찾기 input 박스 미입력시 오류 메시지 구현(비밀번호찾기 동일)

```
$("#mName").on("blur",function(){
    console.log("mName blur");
    if($("#mName").val() == ""){
        $('#check').text('이름을 입력해주세요.');
```

```
<div>
  <p class="card-title" style="color:#FA9D00;">아이디 찾기</p>
</div>
<div class="card-body">
  <input type="text" name="mName" id="mName" class="form-control" placeholder="이름" required autofocus><br>
  <input type="email" name="mEmail" id="mEmail" class="form-control" placeholder="이메일" required><br>
  <p class="check" id="check">${check}</p><br>
  <button id="btn-Yes" class="form-control type="button">아이디찾기</button>
</div>
```

황선애 - 비밀번호 찾기 & 비밀번호 재설정

비밀번호 재설정

임시 비밀번호가 메일로 전송 되었습니다.

닫기

비밀번호 찾기

a12345

aaa@gmail.com

비밀번호 재설정

```
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    System.out.println("doPost - findPw 비밀번호 정보 조회 ");
    // ajax로 들어옴
    PrintWriter out = response.getWriter();
    // 비밀번호찾기
    String mId = request.getParameter("mId");
    String mEmail = request.getParameter("mEmail");
    // 데이터 확인용
    Enumeration params = request.getParameterNames();
    while (params.hasMoreElements()) {
        String name = (String) params.nextElement();
        System.out.print(name + " : " + request.getParameter(name) + "    ");
    }
    // id email 확인
    int result = new MemberService().readFindPw(mId, mEmail);
    if (result == 0) {
        out.print("empty");
        out.flush();
        out.close();
        return;
    }
}
```

비밀번호 재설정 post 이동

- 아이디 이메일 정보 받아서 DB 데이터 확인 후 일치 하면 인증번호 생성 후 새 비밀번호 업데이트 후 이메일 전송 후 모달창에 메시지 출력

황선애 - 관리자(전체 회원조회)

관리자 페이지

회원정보 조회

선생님 승인여부

선생님 신고관리

게시판관리

중전금액 조회

자주 묻는 질문관리

로그아웃

회원정보 조회

아이디	성명	닉네임	생년월일	주소	핸드폰	이메일	성별	신분	가입일자	알람여부
a12345	홍길동	홍기	111111	서울시관악구신림동50	010-1111-1111	aaa@gmail.com	F	S	2022-05-13 16:29:11.965	Y
b12345	이민호	민호	222222	경기도 부천시 삼곡본동 340	010-2211-1111	bbb@gmail.com	F	S	2022-05-13 16:29:11.981	Y
c12345	이영희	영희	900423	서울시 강남구 논현동 224	010-3311-1111	ccc@gmail.com	F	T	2022-05-13 16:29:11.981	Y
d12345	김보미	보미	000506	서울시 용산구 한남동 24	010-4411-1111	ddd@gmail.com	M	T	2022-05-13 16:29:11.996	Y
asfdgk	김지아	지아	990701	경기 성남시 분당구 분당내곡로 117 null	010-2345-1234	sunea24@naver.com	F	S	2022-05-13 21:05:10.822	Y
g12345	정동현	석	910102	서울시 구로구 신도림동 10	010-2345-6788	rrr@gmail.com	M	T	2022-05-13 16:29:12.012	Y
e12345	송민호	MINO	940104	서울시 영등포구 신길1동	010-4411-1111	ggg@gmail.com	M	T	2022-05-13 16:29:12.028	Y
h12345	전소연	과외과외	990918	서울시 관악구 보라매동 80	010-7878-1131	hhh@gmail.com	F	T	2022-05-13 16:29:12.043	Y
i12345	김이안	평생가는국어	990826	서울시 도봉구 도봉동 86-7	010-7902-1081	iii@gmail.com	M	T	2022-05-13 16:29:12.043	Y
j12345	신사랑	사회1타강사	930102	서울시 중구 송파로3가	010-2449-8899	jjj@gmail.com	F	T	2022-05-13 16:29:12.059	Y
k12345	정인하	방가방가	880105	서울시 종로구 팔판동 33-2	010-5345-7428	kkk@gmail.com	F	T	2022-05-13 16:29:12.059	Y
m12345	장민우	초코송이	950505	서울시 은평구 용암3동	010-2222-3333	mmm@gmail.com	M	T	2022-05-13 16:29:12.074	Y
n12345	김인하	호병	830407	서울특별시 용산구 한남동	010-1111-1234	nnn@naver.com	F	T	2022-05-13 16:29:12.09	Y
o12345	한동원	자상한선생님	950512	서울특별시 구로구 구로동	010-1111-5678	ooo@naver.com	M	T	2022-05-13 16:29:12.106	Y
p12345	이소안	제니	960324	서울특별시 양천구 목동	010-2222-1234	ppp@naver.com	F	T	2022-05-13 16:29:12.121	Y
q12345	박수안	과학교청고수	921208	서울특별시 은평구 역촌동	010-2222-5678	qqq@gmail.com	F	T	2022-05-13 16:29:12.137	Y
r12345	김도현	고등문과전문	880928	서울특별시 강서구 가양1동	010-3333-1234	rrr@gmail.com	M	T	2022-05-13 16:29:12.153	Y
s12345	장근희	제이크	871010	서울특별시 성동구 행당2동	010-3333-5678	sss@gmail.com	M	T	2022-05-13 16:29:12.153	Y
t12345	박영구	서울대영어국어	980508	서울특별시 동작구 흑석동	010-4444-1234	ttt@gmail.com	M	T	2022-05-13 16:29:12.168	Y
u12345	오재현	사과	710717	서울특별시 동대문구 장안동	010-4444-5678	uuu@gmail.com	M	T	2022-05-13 16:29:12.184	Y
uesr01	이소라	소라	970701	서울 관악구 신림동2길 5 203호	010-234-1254	soea@naver.com	F	S	2022-05-15 21:22:36.751	Y
user02	이미영	미영1	990304	서울 강남구 강남대로 478 502호	010-4422-1124	mio2@naver.com	M	S	2022-05-15 21:42:20.903	Y
asdgfh	김주아	주아	980803	경기 성남시 수정구 대왕판교로 1187 null	010-2345-2353	sunea241@naver.com	F	S	2022-05-13 21:22:33.359	Y
user03	이덕진	덕진	980702	경기 성남시 분당구 고기로 25 203호	010-2344-1234	ejrws@naver.com	M	S	2022-05-15 21:44:53.564	Y

```

<div id="memberList">
  <p class="member-title">회원정보 조회</p>
  <div class="member-tableAll">
    <table class="member-tableAll">
      <thead>
        <tr id=title>
          <th>아이디</th>
          <th>성명</th>
          <th>닉네임</th>
          <th>생년월일</th>
          <th>주소</th>
          <th>핸드폰</th>
          <th>이메일</th>
          <th>성별</th>
          <th>신분</th>
          <th>가입일자</th>
          <th>알람여부</th>
        </tr>
      <thead>
        <c:forEach items="${retVolist}" var="vo">
          <tr>
            <td>${vo.getId()}</td>
            <td>${vo.getName()}</td>
            <td>${vo.getNickname()}</td>
            <td>${vo.getBirth()}</td>
            <td>${vo.getAddress()}</td>
            <td>${vo.getmPhone()}</td>
            <td>${vo.getEmail()}</td>
            <td>${vo.getGenderFm()}</td>
            <td>${vo.getRoleSt()}</td>
            <td>${vo.getDate()}</td>
            <td>${vo.getAlarmYn()}</td>
          </tr>
        </c:forEach>
      </thead>
    </table>
  </div>
</div>

```

황선애 - 관리자(전체 회원조회)

```
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    System.out.println("doGet - memberList 페이지 이동");
    AdminVo vo = (AdminVo)request.getSession().getAttribute("ssMV");

    if(vo==null) {
        response.sendRedirect(request.getContextPath());
    } else {
        ArrayList<MemberVo> retVolist = new MemberService().readAllMember();
        System.out.println(retVolist);

        request.setAttribute("retVolist", retVolist);
        request.getRequestDispatcher("WEB-INF/view/member/memberList.jsp").forward(request, response);
    }
}
```

```
// 회원 전체조회 (관리자 회원전체 조회)
public ArrayList<MemberVo> readAllMember() {
    ArrayList<MemberVo> retVolist = null;
    Connection conn = getConnection();
    retVolist = dao.readAllMember(conn);
    close(conn);
    return retVolist;
}
```

```
// 회원 전체 읽기
public ArrayList<MemberVo> readAllMember(Connection conn) {
    ArrayList<MemberVo> retVolist = null;
    String sql = "select * from member";
    try {
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        retVolist = new ArrayList<MemberVo>();
        while (rs.next()) {
            MemberVo vo = new MemberVo();
            vo.setmId(rs.getString("m_Id"));
            vo.setmPw(rs.getString("m_pw"));
            vo.setName(rs.getString("M_NAME"));
            vo.setmNickname(rs.getString("m_Nickname"));
            vo.setmBirth(rs.getString("m_Birth"));
            vo.setmAddress(rs.getString("m_Address"));
            vo.setmPhone(rs.getString("m_phone"));
            vo.setmEmail(rs.getNString("m_Email"));
            vo.setGenderFm(rs.getNString("gender_Fm"));
            vo.setRoleSt(rs.getString("role_St"));
            vo.setmDate(rs.getTimestamp("m_Date"));
            vo.setmAlarmYn(rs.getString("M_ALARM_YN"));

            retVolist.add(vo);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        close(pstmt);
    }
    return retVolist;
}
```

황선애 - 관리자(전체 회원조회)

관리자 페이지

회원정보 조회

회원정보 조회

선생님 승인여부

선생님 신고관리

게시판관리

충전금액 조회

자주 묻는 질문관리

로그아웃

아이디	성명	닉네임	생년월일	주소	핸드폰	이메일	성별	신분	가입일자	알림여부
a12345	홍길동	홍기	111111	서울시관악구신림동50	010-1111-1111	aaa@gmail.com	F	S	2022-05-13 16:29:11.965	Y
b12345	이민호	민호	222222	경기도 부천시 삼곡본동 340	010-2211-1111	bbb@gmail.com				
c12345	이영희	영희	900423	서울시 강남구 논현동 224	010-3311-1111	ccc@gmail.com				
d12345	김보미	보미	000506	서울시 용산구 한남동 24	010-4411-1111	ddd@gmail.com				
asfdgk	김지아	지아	990701	경기 성남시 분당구 분당내곡로 117 null	010-2345-1234	sunea24@naver.com				
g12345	정동한	석학	910102	서울시 구로구 신도림동 10	010-2345-6788	rrr@gmail.com				
e12345	송민호	MINO	940104	서울시 영등포구 신길1동	010-4411-1111	ggg@gmail.com				
h12345	전소연	과외과외	990918	서울시 관악구 보라매동 80	010-7878-1131	hhh@gmail.com				
i12345	김이만	평생가는국어	990826	서울시 도봉구 도봉동 86-7	010-7902-1081	iii@gmail.com				
j12345	신사랑	사회1타점사	930102	서울시 중구 충무로3가	010-2449-8899	jjj@gmail.com				
k12345	장인하	방가방가	880105	서울시 종로구 팔판동 33-2	010-5345-7428	kkk@gmail.com				
m12345	장민우	초코송이	950505	서울시 은평구 용암3동	010-2222-3333	mmm@gmail.com				
n12345	김인하	호병	830407	서울특별시 용산구 한남동	010-1111-1234	nnn@naver.com				
o12345	한동원	자상한선생님	950512	서울특별시 구로구 구로동	010-1111-5678	ooo@naver.com				
p12345	이소안	제니	960324	서울특별시 양천구 목동	010-2222-1234	ppp@naver.com				
q12345	박수환	과학코칭고수	921208	서울특별시 은평구 역촌동	010-2222-5678	qqq@gmail.com	F	T	2022-05-13 16:29:12.137	Y
r12345	김도현	고등문과신문	880928	서울특별시 강서구 가림1동	010-3333-1234	rrr@gmail.com	M	T	2022-05-13 16:29:12.153	Y
s12345	정근희	제이크	871010							
t12345	박영구	서울대영어국어	980508							
u12345	오재현	사과	710717							
uesr01	이소라	소라	970701							
user02	이미영	미영1	990304							
asdgfhh	김주아	주아	980803							
user03	이덕진	덕진	980702							

```

<header>
  <div class="Logo">
    <a href="#">
      
    </a>
  </div>
  <nav class="menu">
    <ul>
      <c:choose>
        <c:when test="${not empty ssMV}">
          <li><a href="${pageContext.request.contextPath}/logout">로그아웃</a><span>&nbsp;&nbsp;&nbsp;</span></li>
        </c:when>
        <c:choose>
          </c:choose>
        </ul>
      </nav>
    </header>
  
```

```

<nav class="adminList">
  <p>관리자 페이지</p>
  <ul>
    <li><a href="${pageContext.request.contextPath}/memberList">회원정보 조회</a></li>
    <li><a href="${pageContext.request.contextPath}/teacherApprovalList">선생님 승인여부</a></li>
    <li><a href="${pageContext.request.contextPath}/teacherReportList">선생님 신고관리</a></li>
    <li><a href="${pageContext.request.contextPath}/BoardReportList">게시판관리</a></li>
    <li><a href="${pageContext.request.contextPath}/pencilChart?type=A">충전금액 조회</a></li>
    <li><a href="#">자주 묻는 질문관리</a></li>
  </ul>
</nav>
  
```