

PORTFOLIO

끈기와 열정이 넘치는 개발자, 이석호

CONTENTS



001 Developer SH

- 기본 인적사항
- 보유기술 숙련도



002 자격증 및 교육이수 내역

- 자격증
- 교육이수 내역



003 프로젝트

- 프로젝트 소개
- 담당 업무
- 상세 내용
- 프로젝트 개선 사항

1.1 기본 인적사항

Personal Information



이름	이석호	영문	Lee Seok Ho	한문	李錫昊
생년월일	1994.05.18			나이	26세
휴대폰	010-5157-3715	전화번호	-		
E - mail	lsh3715@gmail.com	SNS	Instargram / Facebook		
주소	인천광역시 미추홀구 석정로323번길 20-1, A동 502호				

학력사항 (최종학력: 백석대학교(4년) 졸업)

재학기간	학교명 및 전공	학점	구분
2013.03 ~ 2019.08	백석대학교 정보보호학과	3.2	졸업
2010.03 ~ 2013.02	대인고등학교	-	졸업

1.2 보유 기술 숙련도

업무 수행시 자주 사용되는 프로그램의 숙련도 표시

JAVA



Spring 4



DB



JavaScript



HTML/CSS



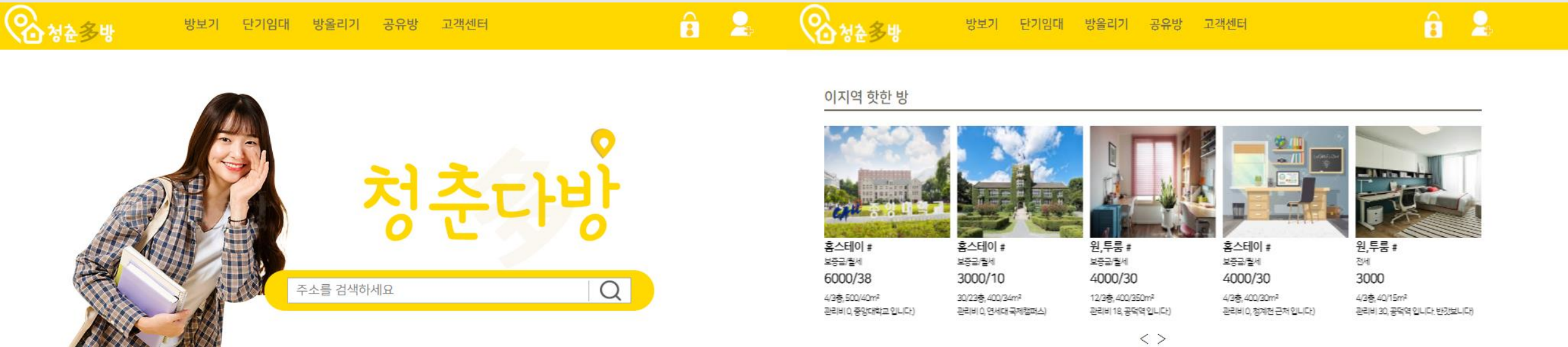
2.1 자격증 및 교육이수 내역

어학 자격증 및 컴퓨터 자격증 목록

취득일	자격증 명	기관	합격/점수
2019.05.22	정보처리기사	한국산업인력공단	최종합격

기간	기관/장소	내용
2019.07 ~ 2019.12	쌍용강북교육센터	JAVA를 이용한 웹 응용SW개발자 양성과정

3.1 부동산 중개 사이트 '청춘多방'

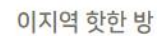


#주소 #서울 마포구 마포대로 지하 100, 공덕역

인실	보증금/보증금	전세	관리비
-	-	3000	30

총/건물층수 : 4층/3층
관리비 포함항목 : 수도, 인터넷, TV
반려동물 : X
난방종류 : 중앙난방

전용면적 : 40m² / 15평
베란다/발코니 : X
입주가능일 : 2019-11-21



홈스테이 #
보증금/월세
6000/38
4/3층, 500/40m²
관리비 0, 중앙대학교 입니다)



홈스테이 #
보증금/월세
3000/10
30/23층, 400/34m²
관리비 0, 연세대 국제캠퍼스



원,투룸 #
보증금/월세
4000/30
12/3층, 40Q/350m²
관리비 18, 공덕역입니다)



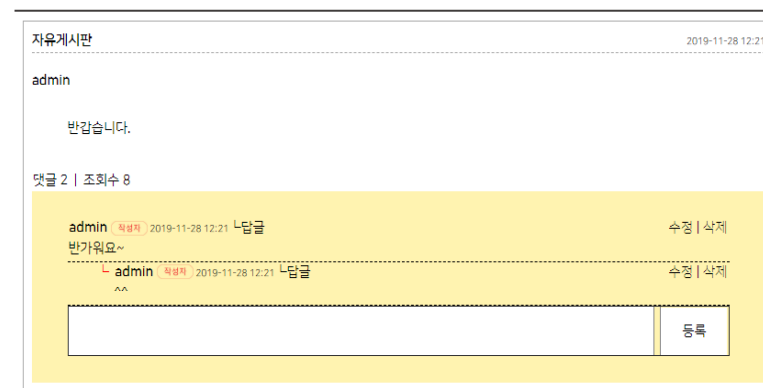
홈스테이 #
보증금/월세
4000/30
4/3층, 400/30m²
관리비 0, 청계천 근처입니다)



원,투룸 #
전세
3000
4/3층, 40/15m²
관리비 30, 공덕역입니다. 반값보니다



자유게시판



글쓰기 답글 수정 삭제 목록

3.1 부동산 중개 사이트 '청춘多방'

개발 PLATFORM		프로젝트 개요
언어	JAVA, HTML, CSS3, JAVASCRIPT, JSP/SERVLET,	Spring Framework를 활용하여 젊은 청춘들의 편한 쉼터를 찾아주는 부동산 중개 사이트 '청춘多방'
Datebase	Oracle 11g	
Framework	Spring 4.0	
Web Servers	Apache Tomcat	
IDE & Tools	Spring Tool Suite4, VSCODE	
형상 관리	git	

3.2 담당 업무

담당업무

업무 내용	비고
회원 관리(로그인, 회원가입, 세션) 구현 및 데이터 암호화	AES Crypto, AJAX 비동기 통신
회원 인증을 위한 mailsender 라이브러리 사용하여 이메일 발송, 회원 간의 거래 추진을 위하여 smssender API 사용하여 SMS발송	mailsender, smssender
방 올리기, 상세페이지 구현 및 방을 보여주는 지도 API 사용	Kakao API
회원 마이페이지(찜한 목록, 나의 방, 회원정보 수정 및 탈퇴) 구현	-

3.3 상세 내용(회원 관리)

회원가입

로그인

이메일회원가입

test1@naver.com

사용중인 이메일입니다. 확인해주세요 :p

대식호

아주 멋진 닉네임이군요!

* 비밀번호(영문,숫자,특수문자 포함 8자리 이상)

* 비밀번호 확인(영문,숫자,특수문자 포함 8자리 이상)

* 성명(한글)

생년월일(8자리로 입력)

연락처(- 구분없이 입력)

☒ 남성 ☐ 여성

회원가입

이메일

비밀번호

로그인

회원가입

이메일 찾기 패스워드 찾기

M_EMAIL	M_PWD
leesh3715@naver.com	c9HPiFkS5zEWZo8lsxZLRA==
whdydxoz@naver.com	asdf1234@
test1@naver.com	CosIT8LZi4LHONFG+Qsotg==
admin	PXvIbPeposVfSe8tozbYbQ==

```
@RequestMapping(value = "/login.do", method = RequestMethod.POST)
public ModelAndView modify(@RequestParam("input_email") String input_email,
    @RequestParam("input_pwd") String input_pwd, HttpServletRequest request, HttpServletResponse response)
    throws Exception {

    memDTO mdto = new memDTO();
    mdto.setM_email(input_email);
    mdto.setM_pwd(AESCrypto.encrypt(input_pwd));
    System.out.println(AESCrypto.encrypt("1234"));
    System.out.println(AESCrypto.encrypt("0000"));
    ModelAndView mav = new ModelAndView();
    HttpSession session = request.getSession();
    memDTO mem = mservice.login(mdto);

    if (mem == null) {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<script>");
        out.println("alert('아이디와 비밀번호를 확인하세요')");
        out.println("history.back()");
        out.println("</script>");
    }

    import java.security.Key;

    public class AESCrypto {

        private static final String ALGORITHM = "AES";
        private static final String myEncryptionKey = "LEESEOKH01234567";
        private static final String UNICODE_FORMAT = "UTF8";

        if (mem == null) {
            response.setContentType("text/html;charset=UTF-8");
            PrintWriter out = response.getWriter();
            out.println("<script>");
            out.println("alert('아이디와 비밀번호를 확인하세요')");
            out.println("history.back()");
            out.println("</script>");
        }
    }
}
```

임의의
대칭키를 사용하여, 암호화

```
// 이메일 텍스트창에서 커서 이동시 중복확인 및 정규화 체크
$(function() {
    $('#m_email').blur(function(){
        var m_email = $('#m_email').val();
        $.ajax({
            async : false,
            url : "${pageContext.request.contextPath}/checkemail.do?mem_email="+ m_email,
            type : "post",
            success : function(data){
                console.log("1 = 중복o / 0 = 중복x : "+ data);
                if(data == 1){
                    $('#emailcheck').text("사용중인 이메일입니다. 확인해주세요 :p");
                    $('#emailcheck').css("color", "red");
                } else {
                    if(emailJ.test(m_email)){
                        // 이메일 정규 체크
                        $('#emailcheck').css("color", "green");
                        $('#emailcheck').text("사용 가능한 아이디입니다.");
                    } else if(m_email == ""){
                        $('#emailcheck').text('이메일을 입력해주세요 :)');
                        $('#emailcheck').css('color', 'red');
                    } else {
                        $('#emailcheck').text("이메일 형식이 올바르지 않습니다.");
                        $('#emailcheck').css('color', 'red');
                    }
                }
            }, error : function() {
                console.log("실 - 패");
            }
        });
    });
});
```

```
// 이메일 중복 체크 [AJAX]
@RequestMapping(value = "/checkemail.do", method = RequestMethod.POST)
@ResponseBody
public int checkemail(@RequestParam("mem_email") String m_email) {
    return mservice.checkemail(m_email);
}
```

비동기 통신을 사용하여, 컨트롤러를 통해
DB의 값과 비교, 값 여부를 리턴

3.3 상세 내용(회원 관리)

비밀번호 찾기

☐ 이메일로 인증하기

회원정보에 등록된 이름과
전화번호가 같아야 합니다.

이름

전화번호 (- 빼고 입력)

비밀번호 찾기

답장 전체답장 전달 삭제 스팸신고

☆ 청년다방에서 알려드립니다. ✉

보낸사람 ☆ <leesh3715@naver.com>

받는사람 <leesh3715@naver.com>

안녕하세요.

회원님의 비밀번호는 0000 입니다.

```
if(this.mservice.findPwd(mdto)!=null) {
    mdto2=this.mservice.findPwd(mdto);
}else {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<script>");
    out.println("alert('이름과 전화번호를 확인하세요')");
    out.println("history.back()");
    out.println("</script>");
}

String decrypt_pwd = AESCrypto.decrypt(mdto2.getM_pwd());

if(mdto2.getM_name()!=null) {
    String setfrom = "leesh3715@naver.com";
    String tomail = mdto2.getM_email(); // 받는 사람 이메일
    String title = "청년다방에서 알려드립니다."; // 제목
    String content = "안녕하세요.\r\n" +
        "\r\n" +
        "회원님의 비밀번호는 "+decrypt_pwd+" 입니다.\r\n" +
        "\r\n" +
        ""; // 내용
    try {
        MimeMessage message = mailSender.createMimeMessage();
        MimeMessageHelper messageHelper = new MimeMessageHelper(message,
            true, "UTF-8");
        messageHelper.setFrom(setfrom); // 보내는사람 생략하면 정상작동을 안함
        messageHelper.setTo(tomail); // 받는사람 이메일
        messageHelper.setSubject(title); // 메일제목은 생략이 가능하다
        messageHelper.setText(content); // 메일 내용
        mailSender.send(message);
    }
```

JAVA mail sender 라이브러리를 사용하여
입력받은 이름, 전화번호 값과 DB의 값이 같으면
회원 비밀번호를 회원 이메일을 통해 전송

```
Coolsms coolsms = new Coolsms(api_key, api_secret);

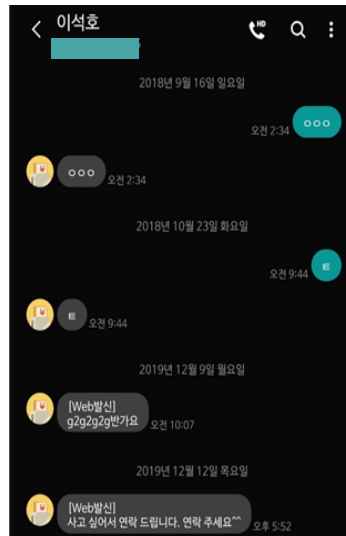
HashMap<String, String> set = new HashMap<String, String>();
set.put("to", "[전화번호]"); // 수신번호
set.put("from", "[전화번호]"); // 발신번호
set.put("text", send_text); // 문자내용 set.put("type", "800"); // 문자 타입

org.json.simple.JSONObject result = coolsms.send(set); // 보내기&전송결과받기

if ((boolean)result.get("status") == true) { // 메시지 보내기 성공 및 전송결과 출력
    System.out.println("성공"); System.out.println(result.get("group_id")); //그룹아이디
    System.out.println(result.get("result_code")); // 결과코드
    System.out.println(result.get("result_message")); // 결과 메시지
    System.out.println(result.get("success_count")); // 메시지아이디
    System.out.println(result.get("error_count")); // 여러개 보낼시 오류난 메시지 수 } else {
// 메시지 보내기 실패 System.out.println("실패");
System.out.println(result.get("code")); // REST API 에러코드
System.out.println(result.get("message")); // 에러메시지 }

response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<script>");
out.println("alert('SMS 전송 완료')");
out.println("history.back()");
out.println("</script>");
```

SMS API를 사용하여 입력한 메시지를 방 정보를
올린 회원의 전화번호에 전송



3.3 상세 내용(지도 API)



KAKAO 지도 API를 사용하여, map을 불러오고,
방 정보들을 DB에서 가져온 후, json 타입으로 자바스크립트
배열에 저장하여 마커와 인포 윈도우로 출력

```
var positions = [];
for(var i = 0; i < json.length; i++) {
    var Photosrc = json[i].r_photo.split('/');
    console.log(Photosrc[0]);
    positions.push({
        title : '<div class="r_wrap">' +
            '<div class="r_info">' +
            '<div class="r_title">' + json[i].r_no +
            '<div class="r_close" onclick="closeOverlay()" title="닫기"></div>' +
            '</div>' +
            '<div class="r_body">' +
            '<div class="r_img">' +
            '' +
            '</div>' +
            '<div class="r_desc">' +
            '<div class="r_ellipsis">' + json[i].r_address +
            '<div class="r_jibun ellipsis">방타입: '+json[i].r_btype+'&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& 가격: '+json[i].r_rentfee +'</div>' +
            '<div><a href="room_cont.do?r_no='+json[i].r_no+'" class="r_link">자세히 보기</a></div>' +
            '</div>' +
            '</div>' +
            '</div>' +
            '</div>'.
        latlng : new kakao.maps.LatLng(json[i].r_location1, json[i].r_location2)
    });
}
```

3.4 프로젝트 개선 사항

개선 사항	비 고
로그인 기존 세션 처리 : 각 페이지마다 자바 HttpSession 값을 가지고 넘어감 -> 개선 사항 : Interceptor를 이용하여 DispatcherServlet 이 Controller를 호출하기 전에 로그인 세션 처리를 통하여 불필요한 코드 수정	진행 예정
회원 관리 : 회원의 비밀번호를 DB에서 불러와 이메일로 전송 -> 개선 사항 : 회원의 비밀번호를 이메일로 전송하여, 비밀번호를 초기화 할 수 있는 링크를 제공	진행 예정
지도 API 활용 : DB의 값들을 불러와 마커와 인포 인도우를 통하여 MAP에 출력 -> 개선 사항 : 기존 기능보다 좀 더 상세하고, 자세하게 구현하여 확대-축소 바, 마커 클러스터러, 로드뷰 등 다양한 기능 추가	진행 예정
실행 환경 : 기존 local 환경에서 작업 -> 개선 사항 : AWS EC2 활용하여 클라우드 환경에서의 작업, 서버 및 배포	진행중 (20%)

부족한 부분은 수시로 추가예정 입니다.

감사합니다

지원자 이석호