

F o r v a c a t i o n

2차 프로젝트

강민재 정병수 최영진

CONTENTS

01

개발과정

- 보유/사용 기술 목록
- 주요 기술 관계도
- 순서도
- 개발기간
- DB 테이블

02

기술설명

- 최영진
- 강민재
- 정병수

03

소감

- 정병수
- 강민재
- 최영진



GitHub



Visual Studio Code



draw.io



DBBeaver



HTML



JS



CSS



Apache Tomcat



Java
JDBC



JSON



Bootstrap



eclipse



Java™
© Logos/Icons

01



01

	단계	9/16	9/17	9/18	9/19	9/20
기획	프로젝트기획					
설계	데이터베이스설계					
코딩	화면구현					
	코딩					
구현	디버깅					

	단계	9/23	9/24	9/25	9/26	9/27
기획	프로젝트기획					
설계	데이터베이스설계					
코딩	화면구현					
	코딩					
구현	디버깅					

Members Table

속성	속성타입	제약조건	내용
Mid	Varchar(20)	Primary key	아이디
Pw	Varchar(20)	Not null	비밀번호
name	Varchar(20)	Not null	이름
Birth	Varchar(20)	Not null	생년월일
Gender	Varchar(20)	Not null	성별
Email	Varchar(50)	Not null	이메일
Tel	Varchar(20)	Not null	전화번호

Calendarlist Table

속성	속성타입	제약조건	내용
Schedulenum	Int(11)	Primary key Auto increment	스케줄 ID
Mid	Varchar(50)		회원ID
Checkdate	Date		날짜
Content	Varchar(100)		내용

Reservation Table

속성	속성타입	제약조건	내용
Resnum	Varchar(100)		예약번호
Regnum	Varchar(100)		사업자번호
Wnum	Varchar(100)		직원번호
Resdate	Varchar(100)		예약일자
Type	Varchar(100)		업종
Etc	Varchar(100)		요청사항
State	Varchar(100)		예약상태
Kdate	Datetime		요청일시
Resgroup	Int(11)		예약그룹

Address Table

속성	속성타입	제약조건	내용
Address1	Varchar(100)		시군주소
Address2	Varchar(100)		구읍면주소

Worker Table

속성	속성타입	제약조건	내용
Wnum	Int(11)	Primary key Auto increment	직원번호
Mid	Varchar(20)	Not null	회원ID
Type1	Varchar(100)		업종1
Period1	Int(11)		경력1
Type2	Varchar(100)		업종2
Period2	Int(11)		경력2
Type3	Varchar(100)		업종3
Period3	Int(11)		경력3
Address1	Varchar(100)		시군주소

Worker Table

속성	속성타입	제약조건	내용
Address2	Varchar(100)		구입면주소
Wcontent	Text		자기소개
Img	Varchar(100)		사진

Workerboard Table

속성	속성타입	제약조건	내용
Wrid	Int(11)	Primary key Auto increment	리뷰ID
Resnum	Varchar(100)		예약번호
Wnum	Int(11)		직원번호
Mid	Varchar(100)		회원ID
Rating	Decimal(2,1)		평점
Wrcontent	Text		리뷰내용
Wrdate	Datetime		작성일시

User Table

속성	속성타입	제약조건	내용
Regnum	Varchar(100)		사업자번호
Mid	Varchar(100)		회원번호
Storename	Varchar(100)		사업장이름
Postcode	Varchar(100)		우편번호
Roadaddress	Varchar(100)		도로명주소
Detailaddress	Varchar(100)		상세주소
Type	Varchar(100)		업종
Ssize	Varchar(100)		규모
Sinfo	Varchar(100)		사업장소개
Simgsrc	Varchar(100)		사업장사진

아이디 조건 및 중복체크

아이디

비밀번호

비밀번호 재확인

이름

생년월일

성별

성별 ▼

이메일

휴대전화

가입하기



```
$("#id").keyup(function() {
    var idValue = $("#id").val().replace(/ /gi, "");
    var pattern = /^[a-zA-Z0-9]/;

    if (idValue.length < 5 || pattern.test($("#id").val())) {
        $("#idcheck").html("<font color = red>아이디는 5자 이상 영문, 숫자로 입력해야 합니다.</font>");
        count4 = 1;
    } else {
        count4 = 0;
        $.ajax({
            url: "overlapcheck.jsp",
            data: {
                id: idValue
            },
            success: function(result) {
                count = result;
                if (count == 1) {
                    $("#idcheck").html("<font color = red>중복된 아이디 입니다.</font>");
                } else {
                    $("#idcheck").html("<font color = green>사용 가능한 아이디입니다.</font>");
                }
            }
        });
    }
});
```

```
String id = request.getParameter("id");

MemberDAO dao = new MemberDAO();
MemberDTO dto = new MemberDTO();
ArrayList<MemberDTO> list = dao.selectAll();

int data = 0;
for(int i = 0; i < list.size(); i++){
    MemberDTO dto2 = list.get(i);
    if(id.equals(dto2.getId())){
        data = 1;
    }
}

public ArrayList<MemberDTO> selectAll() throws Exception {
    //2. DB 로 접속한다.
    DBConnectionMgr mgr = DBConnectionMgr.getInstance();
    Connection con = mgr.getConnection();
    System.out.println("2. DB 연결 OK.");

    //3) sql문 가져와
    String sql = "select * from members";
    PreparedStatement ps = con.prepareStatement(sql);
    System.out.println("3. sql 가져와 OK.");

    //4) sql문 실행 요청
    ResultSet rs = ps.executeQuery();
    System.out.println("4. sql 실행 요청 OK.");
    MemberDTO dto2 = null;
    ArrayList<MemberDTO> list = new ArrayList<MemberDTO>();
    while (rs.next()) {
        dto2 = new MemberDTO();
        //ORM ( object : 자바 rdb : table 항목 mapping)
        dto2.setId(rs.getString(1));
        dto2.setPw(rs.getString(2));
        dto2.setName(rs.getString(3));
        dto2.setBirth(rs.getString(4));
        dto2.setGender(rs.getString(5));
        dto2.setEmail(rs.getString(6));
        dto2.setTel(rs.getString(7));
        list.add(dto2);
    }

    rs.close();
    ps.close();
}
```

비밀번호 조건 및 확인

아이디

비밀번호

비밀번호 재확인

이름

생년월일

성별

이메일

휴대전화

```
$("#pw").keyup(function() {  
    if ($("#pw").val().length < 8) {  
        $("#pwcheck").html("<font color = red>패스워드는 8자 이상 입력해야합니다.</font>");  
        count3 = 0;  
    } else {  
        $("#pwcheck").html("<font color = green>사용할 수 있는 패스워드입니다.</font>");  
        count3 = 1;  
    }  
});  
  
$("#pw2").keyup(function() { //pw와 pw재확인 값에 따른 출력 메시지 설정  
    if ($("#pw").val() == $("#pw2").val()) {  
        $("#pwcheck2").html(  
            "<font color = green>패스워드가 동일합니다.</font>");  
        count2 = 0;  
    } else {  
        $("#pwcheck2").html(  
            "<font color = red>패스워드가 동일하지 않습니다.</font>");  
        count2 = 1;  
    }  
})  
//console.log($("#pwrecheck").text()); //출력값 확인
```


이메일 생년월일 형식확인

아이디

비밀번호

비밀번호 재확인

이름

생년월일

xxxx-xx-xx 형식으로 입력해주세요.

성별
성별

이메일

xxxxx@xxx.xxx 형식으로 입력해주세요.

휴대전화

-없이 입력해주세요.

가입하기



```
var exptext = /^[A-Za-z0-9_\.\\-]+@[A-Za-z0-9\\-]+\.[A-Za-z0-9\\-]+/;  
var exptext2 = /^[0-9]+-[0-9]+-[0-9]+/;
```

```
});  
$("#email").keyup(function() {  
    if (exptext.test($("#email").val()) == false) {  
        $("#emailcheck").html(  
            "<font color = red>이메일 형식을 확인해주세요</font>");  
        count5 = 1;  
    } else {  
        $("#emailcheck").html(  
            "<font color = green>바른 형식입니다.</font>");  
        count5 = 0;  
    }  
});  
$("#birth").keyup(function() {  
    if (exptext2.test($("#birth").val()) == false) {  
        $("#birthcheck").html(  
            "<font color = red>형식을 확인해주세요</font>");  
        count5 = 1;  
    } else {  
        $("#birthcheck").html(  
            "<font color = green>바른 형식입니다.</font>");  
        count5 = 0;  
    }  
});
```

가입

아이디

비밀번호

비밀번호 재확인

이름

생년월일

성별

이메일

휴대전화

```
$("#join").click(function() {  
    if (count == 1) {  
        alert("중복된 아이디는 회원가입하실 수 없습니다.");  
        return false;  
    } else if (count2 == 1 || count3 == 0) {  
        alert("비밀번호 재확인이 필요합니다.");  
        return false;  
    } else if ($("#id").val() == "" || $("#pw").val() == "" || $("#pw2").val() == "" || ($("#name").val() == "" ||  
        $("#birth").val() == "" || ($("#gender").val() == "gender" || ($("#email").val() == "" || ($("#tel").val() == "")) {  
        alert("모든 정보들을 입력해야 합니다.");  
        return false;  
    } else if (count4 == 1){  
        alert("아이디 재확인이 필요합니다.");  
        return false;  
    } else if (count5 == 1){  
        alert("이메일 형식을 확인해주세요.");  
        return false;  
    }  
})
```

```
String id = request.getParameter("id");  
String pw = request.getParameter("pw");  
String name = request.getParameter("name");  
String birth = request.getParameter("birth");  
String gender = request.getParameter("gender");  
String email = request.getParameter("email");  
String tel = request.getParameter("tel");  
  
//2. DAO 사용하여 DB 처리  
MemberDAO dao = new MemberDAO();  
MemberDTO dto = new MemberDTO();  
dto.setMid(id);  
dto.setPw(pw);  
dto.setName(name);  
dto.setBirth(birth);  
dto.setGender(gender);  
dto.setEmail(email);  
dto.setTel(tel);  
dao.insert(dto);  
  
public void insert(MemberDTO dto) throws Exception {  
    //2. DB 로 전송한다.  
    DBConnectionMgr mgr = DBConnectionMgr.getInstance();  
    Connection con = mgr.getConnection();  
    System.out.println("2. DB 연결 OK.");  
  
    //3) sql문 객체화  
    String sql = "insert into members values (?, ?, ?, ?, ?, ?)";  
    PreparedStatement ps = con.prepareStatement(sql);  
    ps.setString(1, dto.getMid());  
    ps.setString(2, dto.getPw());  
    ps.setString(3, dto.getName());  
    ps.setString(4, dto.getBirth());  
    ps.setString(5, dto.getGender());  
    ps.setString(6, dto.getEmail());  
    ps.setString(7, dto.getTel());  
    System.out.println("3. sql 객체화 OK.");  
  
    //4) sql문 실행 요청  
    ps.executeUpdate();  
    System.out.println("4. sql 실행 요청 OK.");  
  
    ps.close();  
    con.close();  
}
```

수정 및 탈퇴

아이디

비밀번호 확인

비밀번호 수정

비밀번호 수정을 원하지 않을 시 원래 비밀번호를 기입해주세요.

이름

성별

이메일

휴대전화

수정하기

탈퇴하기

```
String id = (String)session.getAttribute("id");
```

```
MemberDTO dto = new MemberDTO();
MemberDAO dao = new MemberDAO();
dto.setMid(id);
dao.select(dto);
```

```
value = "<%=id%>";
```

```
public MemberDTO select(MemberDTO dto) throws Exception {
    //2. DB 로 접속한다.
    DBConnectionMgr mgr = DBConnectionMgr.getInstance();
    Connection con = mgr.getConnection();
    System.out.println("2. DB 연결 OK.");

    //3) sql문 객체화
    String sql = "select * from members where mid = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, dto.getMid());

    System.out.println("3. sql 객체화 OK.");

    //4) sql문 실행 요청
    ResultSet rs = ps.executeQuery();
    System.out.println("4. sql 실행 요청 OK.");

    if (rs.next()) {
        //ORM ( object : 자바 pojo : table명 목록 mapping)
        dto.setMid(rs.getString(1));
        dto.setPw(rs.getString(2));
        dto.setName(rs.getString(3));
        dto.setBirth(rs.getString(4));
        dto.setGender(rs.getString(5));
        dto.setEmail(rs.getString(6));
        dto.setTel(rs.getString(7));
    }
    rs.close();
    ps.close();
    con.close();
}
```

```
var pw = "<%=dto.getPw()%>";
var id = "<%=session.getAttribute('id')%>";
var count = 0;
var count2 = 0;
$("#pw").keyup(function() {
    if($("#pw").val() == pw){
        $("#pwcheck").html("<font color = green>비밀번호 확인 체크.</font>");
        count = 0;
    }else{
        $("#pwcheck").html("<font color = red>비밀번호를 확인해주세요.</font>");
        count = 1;
    }
});
$("#pw2").keyup(function() {
    if($("#pw2").val() == "" || $("#pw2").val().length < 8){
        $("#pwcheck2").html("<font color = red>비밀번호는 8자 이상 입력해주세요.</font>");
        count2 = 1;
    }else{
        $("#pwcheck2").html("<font color = green>비밀번호 수정이 가능합니다.</font>");
        count2 = 0;
    }
});
```

수정 및 탈퇴

아이디

비밀번호 확인

비밀번호 수정

비밀번호 수정을 원하지 않을 시 원래 비밀번호를 기입해주세요.

이름

성별

이메일

휴대전화

```

$("#update").click(function() {
    if($("#id").val() != id) {
        alert("아이디는 수정할 수 없습니다.");
        return false;
    }
    if(count == 1 || $("#pw").val() == "")
        alert("정보를 확인해주세요.");
        return false;
    }
    if($("#pw2").val() == ""){
        alert("비밀번호 수정칸의 주의사항을 확인하세요.");
        return false;
    }
    String name = request.getParameter("name");
    String gender = request.getParameter("gender");
    String email = request.getParameter("email");
    String tel = request.getParameter("tel");
    String mid = request.getParameter("id");
    String pw = request.getParameter("pw2");

    MemberDAO dao = new MemberDAO();
    MemberDTO dto = new MemberDTO();
    dao.update(pw, name, gender, email, tel, mid);
});

public void update(String pw, String name, String gender, String email, String tel, String mid) throws Exception {
    // 2. DB 연결
    DBConnectionMgr mgr = DBConnectionMgr.getInstance();
    Connection con = mgr.getConnection();
    System.out.println("2. DB 연결 성공");

    // 3. SQL 문 작성
    String sql = "update members set pw = ?, name = ?, gender = ?, email = ?, tel = ? where mid = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, pw);
    ps.setString(2, name);
    ps.setString(3, gender);
    ps.setString(4, email);
    ps.setString(5, tel);
    ps.setString(6, mid);
    ps.executeUpdate();
    System.out.println("3. SQL문 실행 성공");

    // SQL문 전송 요청
    System.out.println("4. SQL문 전송 요청 성공");
    ps.close();
    con.close();
}

```

```

$("#delete").click(function() {
    var id = $("#id").val();
    $.ajax({
        url : "delete.jsp",
        data : {
            id : id
        },
        success : function(res) {
            // 3. SQL 문 결정
            String sql = "delete from members where mid = ?";
            PreparedStatement ps = con.prepareStatement(sql);
            ps.setString(1, mid);
            ps.executeUpdate();
            System.out.println("3. SQL문 결정 성공");

            // SQL문 전송 요청
            System.out.println("4. SQL문 전송 요청 성공");
            ps.close();
            con.close();
        }
    });
});

public void delete(String mid) throws Exception {
    DBConnectionMgr mgr = DBConnectionMgr.getInstance();
    Connection con = mgr.getConnection();
    System.out.println("2. DB 연결 성공");

    // 3. SQL 문 결정
    String sql = "delete from members where mid = ?";
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, mid);
    ps.executeUpdate();
    System.out.println("3. SQL문 결정 성공");

    // SQL문 전송 요청
    System.out.println("4. SQL문 전송 요청 성공");
    ps.close();
    con.close();
}

```

02

로그인 및 로그아웃



ID

Password

아이디 또는 비밀번호를 확인해주세요.

로그인

아이디를 잊으셨나요?
비밀번호를 잊으셨나요?
메인화면으로

```
String id = request.getParameter("id"); // id 와 pw 에 입력된 값을 받아옴
String pw = request.getParameter("pw");

MemberDAO dao = new MemberDAO();
MemberDTO dto = new MemberDTO();

ArrayList<MemberDTO> list = dao.selectAll();

int data = 0;

for(int i = 0; i < list.size(); i++){
    MemberDTO dto2 = list.get(i);
    if(id.equals(dto2.getId()) && pw.equals(dto2.getPw())){ // 입력된 아이디와 비밀번호가 일치하는지 확인
        data = 1;
        session.setAttribute("id", id); // session set
    }
}

out.print(data);
```

```
<%
    session.invalidate(); //session 초기화
    response.sendRedirect("main.jsp"); // 다시 메인화면으로
%>
```

로그인 및 로그아웃 시 버튼

사업자등록

인력등록

로그아웃

회원정보수정

```
<%  
if(session.getAttribute("id")==null){  
    // 로그인 되지 않았을 때  
}%>  
  
<button class="btn btn-primary my-2 my-sm-0" type="button" onclick = "login()">  
    로그인  
</button>  
  
<% }else{ %>  
  
<button class="btn btn-primary my-2 my-sm-0" type="button" onclick = "logout()">  
    로그아웃  
</button>  
  
<%} %>
```

```
<%  
if(session.getAttribute("id")==null){  
}%>  
  
<button class="btn btn-primary my-2 my-sm-0" type="button" onclick = "join()">  
    회원가입  
</button>  
<%}else{ %>  
<button class="btn btn-primary my-2 my-sm-0" type="button" onclick = "change()">  
    회원정보수정  
<%} %>  
</button>
```

검색

시작날짜 :

~

종료날짜 :

업종 :

지역 : 전체

```
String start = request.getParameter("start");
String end = request.getParameter("end");
String type = request.getParameter("type");
String address1 = request.getParameter("address1");
String address2 = request.getParameter("address2");
JoinDAO joinDAO = new JoinDAO();
JoinDTO joinDTO = new JoinDTO();

joinDTO.setStartdate(start);
joinDTO.setEnddate(end);
joinDTO.setType1(type);
joinDTO.setAddress1(address1);
joinDTO.setAddress2(address2);
MemberDTO memberDTO = new MemberDTO();
MemberDAO memberDAO = new MemberDAO();

ArrayList<JoinDTO> list = joinDAO.search(joinDTO);
ArrayList<MemberDTO> namelist = new ArrayList();
for(int i=0; i<list.size(); i++){
    joinDTO = list.get(i);
    memberDTO.setMid(joinDTO.getMid());
    memberDTO = memberDAO.select(memberDTO);
    namelist.add(memberDTO);
}
request.setAttribute("namelist", namelist);
request.setAttribute("list", list); //wnum, mid, image
request.setAttribute("start", start);
request.setAttribute("end", end);
request.setAttribute("type", type);
request.setAttribute("address1", address1);
request.setAttribute("address2", address2);

RequestDispatcher dispatcher = request.getRequestDispatcher("searchresult.jsp");
dispatcher.forward(request, response);
```

```
String sql;
PreparedStatement ps;
if(dto.getAddress2().equals("전체")) {
    sql = "select wnum, mid, image from worker where address1=? and (type1=? or type2=? or type3=?) and mid not in (select mid from calendarlist where checkdate between ? and ?)";
    ps = con.prepareStatement(sql);
    ps.setString(1, dto.getAddress1());
    ps.setString(2, dto.getType1());
    ps.setString(3, dto.getType1());
    ps.setString(4, dto.getType1());
    ps.setString(5, dto.getStartdate());
    ps.setString(6, dto.getEnddate());
}else{
    sql = "select wnum, mid, image from worker where address1=? and address2=? and (type1=? or type2=? or type3=?) and mid not in (select mid from calendarlist where checkdate between ? and ?)";
    ps = con.prepareStatement(sql);

    ps.setString(1, dto.getAddress1());
    ps.setString(2, dto.getAddress2());
    ps.setString(3, dto.getType1());
    ps.setString(4, dto.getType1());
    ps.setString(5, dto.getType1());
    ps.setString(6, dto.getStartdate());
    ps.setString(7, dto.getEnddate());
}

//4) sql문 실행 요청
ResultSet rs = ps.executeQuery();

while (rs.next()) {
    dto = new JoinDTO();
    System.out.println(rs.getInt(1));
    //ORM ( object : 자바 rdb : table명속 mapping)
    dto.setWnum(rs.getInt(1));
    dto.setMid(rs.getString(2));
    dto.setImage(rs.getString(3));
}
```



검색 결과

시작날짜 : 2019-10-04 -종료날짜 : 2019-10-05 업종 : 카페 * 지역 : 서울 전체 검색

전문인력



예약하기 =

☒  taglibs-standard-impl-1.2.5.jar☒  taglibs-standard-spec-1.2.5.jar

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
```

```
<form action="search.jsp" class="form-inline">
  <p>
    시작날짜 : <input type="text" class="datepicker2" id="startdate"
      name="start" value="${start}" autocomplete="off">
  </p>
  <p>~</p>
  <p>
    종료날짜 : <input type="text" class="datepicker2" id="enddate"
      name="end" value="${end}" autocomplete="off">
  </p>
  <p>
    업종 : <select name="type" id="datepicker">
      <option class="form-control mr-sm-2" value="cafe">
        <%( "${type}.equals("cafe") )> {<%> selected <%>}>카페</option>
      <option class="form-control mr-sm-2" value="factory">
        <%( "${type}.equals("factory") )> {<%> selected <%>}>공장</option>
      <option class="form-control mr-sm-2"
        value="design"> <%( "${type}.equals("design") )> {<%> selected <%>}>서점</option>
      </select>
  </p>
  &nbsp;
  <p>
    지역 : <input type="text" id="address1" name="address1"
      value="${address1}"> <input type="text" id="address2"
      name="address2" value="${address2}">
  </p>
  <input class="btn btn-primary my-2 my-sm-0" type="submit"
    id="search" value="검색">
  </p>
</form>
```

```
<c:forEach items="${namelist}" var="dto2">
<p>${dto2.name}</p>
</c:forEach>
<c:forEach items="${list}" var="dto">
<div class="col-md-4">
<h2>
<div class="col-md-12">

</div>
</h2>
<p>
<form action="reservation/reservation.jsp" method="post">
<input type="hidden" name="wnum" value="${dto.wnum}"> <input
type="hidden" name="start" value="${start}"> <input
type="hidden" name="end" value="${end}"> <input
type="hidden" name="type" value="${type}"> <input
type="hidden" name="address1" value="${address1}"> <input
type="hidden" name="address2" value="${address2}"> <input
class="btn" type="submit" value="예약하기 »">
</form>
</p>
</div>
</c:forEach>
```


DatePicker

2019-10-04 | ~종료날짜 : 2

October 2019						
Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

```
<link href="css/bootstrap.min.css" rel="stylesheet">
<!-- DatePicker -->
<link rel="stylesheet"
      href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
<!-- DatePicker -->
```

```
<!-- DatePicker -->
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<!-- DatePicker -->
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<!-- DatePicker -->
<script>
    $(function() {
        $(".datepicker2").datepicker({
            dateFormat : 'yy-mm-dd' //datepicker 의 포맷 형식 지정
        });
    });
</script>
```

```
<input type="text" class="datepicker2"
```

02

기술설명 - 강민재

1. 예약내역확인 및 Fullcalendar API
2. 이미지 미리보기/업로드 및 다음 주소입력 API,
검색을 활용한 예약요청 DB처리
3. 예약 수락/거절 처리 및 Fullcalendar API

1. 예약내역확인 및 Fullcalendar API

1-A 예약 요청내역 출력(인력, Browser)

worker님 환영합니다.

For Vacation

[사업자등록](#)[인력정보](#)[로그아웃](#)[회원정보수정](#)

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨

[휴무일 캘린더로](#)

1-A 예약 요청내역 출력(인력, CODE&DB1)

```
16    <%
17        ReservationDTO dto = new ReservationDTO();
18        ReservationListDAO dao = new ReservationListDAO();
19        dto.setWnum(wnum);
20        //wnum를 가지고 dao.select(dto) 실행
21        ArrayList<ReservationDTO> list = dao.selectAll(dto);
22        // select 사용하여 세션id해당 data 모두 list화 하여 넘겨주기
23    %>
24    //state 상태에 따른 버튼 변경
25    for (var i = 0; i < <%=list.size()%>; i++) {
26        var check = <u>$("#statecheck"+i).text();</u>
27        if (check=="0") {
28            $("#append"+i).append("<button class='statechange' id='stateaccept'>예약확정"</button>");
29            $("#append"+i).append("<button class='statechange' id='staterefuse'>예약취소"</button>");
30        }else if(check=="1"){
31            $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약확정됨"</button>");
32        }else{
33            $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약취소됨"</button>");
34        }
35    }
36    }//for
```

- JSP를 활용하여 DB내 예약요청 내역 배열화
- 예약요청 state column 값에 따른 상태 버튼 차별화하여 append

1-A 예약 요청내역 출력(인력, CODE&DB2)

```
51◉      <%  
52          //for문으로 selectAll table생성  
53          if(list.size()>0){  
54              for(int i=0; i<list.size(); i++){  
55                  %>  
56◉          <tr class="reservationvalue">  
57              <td><%=list.get(i).getResnum()%></td>  
58              <td><%=list.get(i).getRegnum()%></td>  
59              <td><%=list.get(i).getResdate()%></td>  
60              <td><%=list.get(i).getType()%></td>  
61              <td><%=list.get(i).getKdate()%></td>  
62              <td><%=list.get(i).getEtc()%></td>  
63              <td id = "statecheck<%=i%>"><%=list.get(i).getState() %></td>  
64              <td id="append<%=i%>"></td>  
65          </tr>  
66◉      <%  
67          }//for  
68          //list에 값이 없을 경우 table에 예약내역 없음 출력  
69          }else{ %>  
70◉      <tr>  
71          <td colspan="8">예약내역이 존재하지 않습니다.</td>  
72      </tr>  
73◉      <%  
74          }//if  
75          %>  
76      </table>
```

- JSP를 활용하여 DB내 예약요청 내역 table내 해당 column값 출력.
- list가 null일 경우, 예약내역 없음 출력.

1-A 예약 요청내역 출력(인력, CODE&DB3)

```
86     String sql = "select * from reservation where wNum=?";
87     PreparedStatement ps = con.prepareStatement(sql);
88     ps.setInt(1, dto.getWnum());
89     System.out.println("3. sql문 객체화 ok..");
90
91     // 4) SQL문 실행 요청
92     ResultSet rs = ps.executeQuery();
93     System.out.println("4. sql문 실행 요청 ok..");
94     ReservationDTO dto2 = null;
95     ArrayList<ReservationDTO> list = new ArrayList<ReservationDTO>();
96     while(rs.next()) {
97         dto2 = new ReservationDTO();
98         dto2.setResnum(rs.getString(1));
99         dto2.setRegnum(rs.getString(2));
100        dto2.setWnum(rs.getInt(3));
101        dto2.setResdate(rs.getString(4));
102        dto2.setType(rs.getString(5));
103        dto2.setEtc(rs.getString(6));
104        dto2.setState(rs.getString(7));
105        dto2.setKdate(rs.getTimestamp(8));
106        dto2.setResgroup(rs.getInt(9));
107        list.add(dto2);
108    }
109    return list;
```

- 예약내역 추출을 위한 DAO.
- 인력번호를 통한 예약요청 내역 조회 및 list화.

02

1-B 휴무일 입력 기능(인력, Browser1)

worker님 환영합니다.

For Vacation

사업자등록

인력정보

로그아웃

회원정보수정

< >

today

September 2019

휴무입력

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4 휴무	5	6	7
8	9	10	11	12	13	14
15	16 cafe	17 cafe	18 휴무	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

휴무일:

02

1-B 휴무일 입력 기능(인력, Browser2)

worker님 환영합니다.

For Vacation

[사업자등록](#)[인력정보](#)[로그아웃](#)[회원정보수정](#)

<

>

today

September 2019

휴무입력

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4 휴무	5	6	7
8	9	10	11	12	13	14
15	16 cafe	17 cafe	18 휴무	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

휴무일: 2019-09-13

02

1-B 휴무일 입력 기능(인력, Browser3)

worker님 환영합니다.

For Vacation

[사업자등록](#)[인력정보](#)[로그아웃](#)[회원정보수정](#)

<

>

today

September 2019

휴무입력

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6	7
			휴무			
8	9	10	11	12	13	14
					휴무	
15	16	17	18	19	20	21
	cafe	cafe	휴무			
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

휴무일:

1-B 휴무일 입력 기능(인력, CODE&DB1-1)

```
91      //캘린더 db입력 코드 시작
92      //해당 id의 일정 list json 전환하여 넣기
93      $.ajax({
94          url: "http://localhost:8888/project2/1_4_CalendarList.jsp",
95          dataType: "json",
96          success: function(result) {
97              for(var i=0; i<$(result).length; i++){
98                  var content = result[i].content;
99                  var checkdate = result[i].checkdate; //date 형식?
100                  var date = new Date(checkdate + 'T00:00:00'); // will be in local time 변환
101                  if (!isNaN(date.valueOf())) { // valid?
102                      calendar.addEvent({
103                          title: content,
104                          start: date, //선택한 날짜 가져와서 해당날짜에 추가됨
105                          allDay: true //false시 반나절기입됨.
106                      })
107                  }
108              }
109          }
110      }); //캘린더 db ajax
111      //캘린더 db입력 코드 끝
```

- Fullcalendar API와 JQuery Ajax 내 for문을 사용하여 인력 캘린더 내용 출력.
- JSP를 이용하여 캘린더 DB를 JSON화하여 캘린더에 적용.

1-B 휴무일 입력 기능(인력, CODE&DB1-2)

```
8  <%
9      String id = (String)session.getAttribute("id");
10     CalendarListDTO dto = new CalendarListDTO();
11     dto.setId(id);
12     CalendarListDAO dao = new CalendarListDAO();
13     // select 사용하여 세션id해당 data 모두 list화 하여 넘겨주기
14     //id를 가지고 dao.select(dto) 실행
15     ArrayList<CalendarListDTO> list = dao.select(dto);
16     JSONArray jarray = null;
17     //json array 생성
18     try{
19         jarray = new JSONArray();
20         for(int i =0; i<list.size(); i++){
21             JSONObject jobject = new JSONObject();
22             jobject.put("checkdate", list.get(i).getCheckdate());
23             jobject.put("content", list.get(i).getContent());
24             jarray.add(jobject);
25             //검색된 일정을 for문을 이용하여 jsonarray에 삽입.
26         }
27     }catch (Exception e) {
28         e.printStackTrace();
29     }
30     %>
31     <%= jarray.toJSONString() %>
```

- JSON ARRAY 형태를 이용하여 DB 추출 및 삽입.

1-B 휴무일 입력 기능(인력, CODE&DB1-3)

```
52     public ArrayList<CalendarListDTO> select(CalendarListDTO dto) throws Exception {
53
54         DBConnectionMgr mgr = DBConnectionMgr.getInstance();
55         Connection con = mgr.getConnection();
56         System.out.println("DB연결 완료..");
57
58         String sql = "select * from calendarlist where mid=?";
59         PreparedStatement ps = con.prepareStatement(sql);
60         ps.setString(1, dto.getId());
61         System.out.println("3. sql문 객체화 ok..");
62
63         // 4) SQL문 실행 요청
64         ResultSet rs = ps.executeQuery();
65         System.out.println("4. sql문 실행 요청 ok.. select");
66         CalendarListDTO dto2 = null;
67         ArrayList<CalendarListDTO> list = new ArrayList<CalendarListDTO>();
68         while (rs.next()) {
69             dto2 = new CalendarListDTO();
70             dto2.setId(rs.getString(2));
71             dto2.setCheckdate(rs.getString(3));
72             dto2.setContent(rs.getString(4));
73             list.add(dto2);
74         }
75         System.out.println("5. list 삽입 완료");
76         return list;
77     }
```

- 인력의 mid를 활용하여 캘린더 일정을 배열 형태로 추출.

1-B 휴무일 입력 기능(인력, CODE&DB2)

```
14 <input type="hidden" id="mid" value="<%=session.getAttribute("id")%>">
15 <script>
16     var sessionid = $("#mid").val();
17     //세션id ajax사용 위해 hidden에서 가져오기
18     //캘린더 생성
19     document.addEventListener('DOMContentLoaded', function() {
20         var calendarEl = document.getElementById('calendar');
21         var calendar = new FullCalendar.Calendar(calendarEl, {
22             plugins: ['interaction', 'dayGrid'],
23             selectable: true,
24             header: {
25                 left: 'prev,next today',
26                 center: 'title', //연월
27                 right: 'addEventButton' //일정 입력 버튼
28             },
```

- Fullcalendar 기능 설정 부분.

1-B 휴무일 입력 기능(인력, CODE&DB3-1)

```
30      addEventButton: {
31          text: '휴무입력',
32          click: function() {
33              var dateStr = document.getElementById("start-date").value; //날짜 선택 후 값 가져오기
34              var date = new Date(dateStr + 'T00:00:00'); // will be in local time
35              var content = "휴무";
36              //해당 날짜 db존재 여부 확인 및 존재시 진행 거부 ajax 시작
37              $.ajax({
38                  url: "1_2_RestCheck.jsp",
39                  data:{
40                      id : sessionid,
41                      checkdate : dateStr
42                  },
43                  success: function(result) {
44                      if (result=="false") { //선택일 일정 존재시 휴무입력 불가능
45                          alert("해당 날짜에 일정이 존재합니다. 다른날을 선택하세요.")
46                          return false;
47                      } else { //선택일 일정 없을 시 휴무 입력 시작
48                          if (!isNaN(date.valueOf())) { // valid?
49                              calendar.addEvent({
50                                  title: content,
51                                  start: date, //선택한 날짜 가져와서 해당날짜에 추가됨
52                                  allDay: true //false시 반나절기입됨.
53                              }),
54                              //alert('Great. Now, update your database...');
55                              //ajax db입력 라인
```

- 캘린더일정 DB를 확인하여, 해당 mid 및 날짜에 일정이 있을 경우, 휴무 추가 불가능 alert 출력.

1-B 휴무일 입력 기능(인력, CODE&DB3-2)

```
79 public boolean selectBefore(CalendarListDTO dto) throws Exception {
80
81     DBConnectionMgr mgr = DBConnectionMgr.getInstance();
82     Connection con = mgr.getConnection();
83     //System.out.println("DB연결 완료..");
84
85     String sql = "select * from calendarlist where mid=? and checkdate=?";
86     PreparedStatement ps = con.prepareStatement(sql);
87     ps.setString(1, dto.getId());
88     ps.setString(2, dto.getCheckdate());
89     //System.out.println("3. sql문 객체화 ok..");
90
91     // 4) SQL문 실행 요청
92     ResultSet rs = ps.executeQuery();
93     //System.out.println("가 확인");
94     if (rs.next()) {
95         //System.out.println("일정 채워져있음");
96         return false; //해당 날짜 값 있으면 진행정지
97     } else {
98         //System.out.println("일정 비어있음");
99         return true; //해당 날짜 값 없으면 진행 null
100     }
101 }
```

- mid 및 날짜를 활용한 DAO, SQL문 결과가 있을경우 false로 결과를 설정하여 일정확인 ajax에서 활용.

1-B 휴무일 입력 기능(인력, CODE&DB4-1)

```
56         $.ajax({
57             url: "1_3_RestInsert.jsp",
58             data:{
59                 id : sessionid, //세션 아이디 가져오기
60                 checkdate : dateStr, //선택한 날짜
61                 content : content //
62             },
63             success: function(result) {
64                 //alert("db저장 완료");
65             } //success:
66         }); //$.ajax
67         document.getElementById("start-date").value = null;
68         //휴무 입력 후, 휴무일 칸 clear
69     }else {
70         alert('휴무일을 선택해주세요.');
71     } //else 날짜 선택 안했을시
72     } //선택일 일정 없을 시 휴무 입력 끝
73 } //restcheck success close
74 }); //ajax
75 //해당 날짜 db존재 여부 확인 및 존재시 진행 거부 ajax 끝
76 } //click: function()
77 } //addEventButton:
78 }, //customButtons:
79 //날짜 클릭시 선택날짜 확인
```

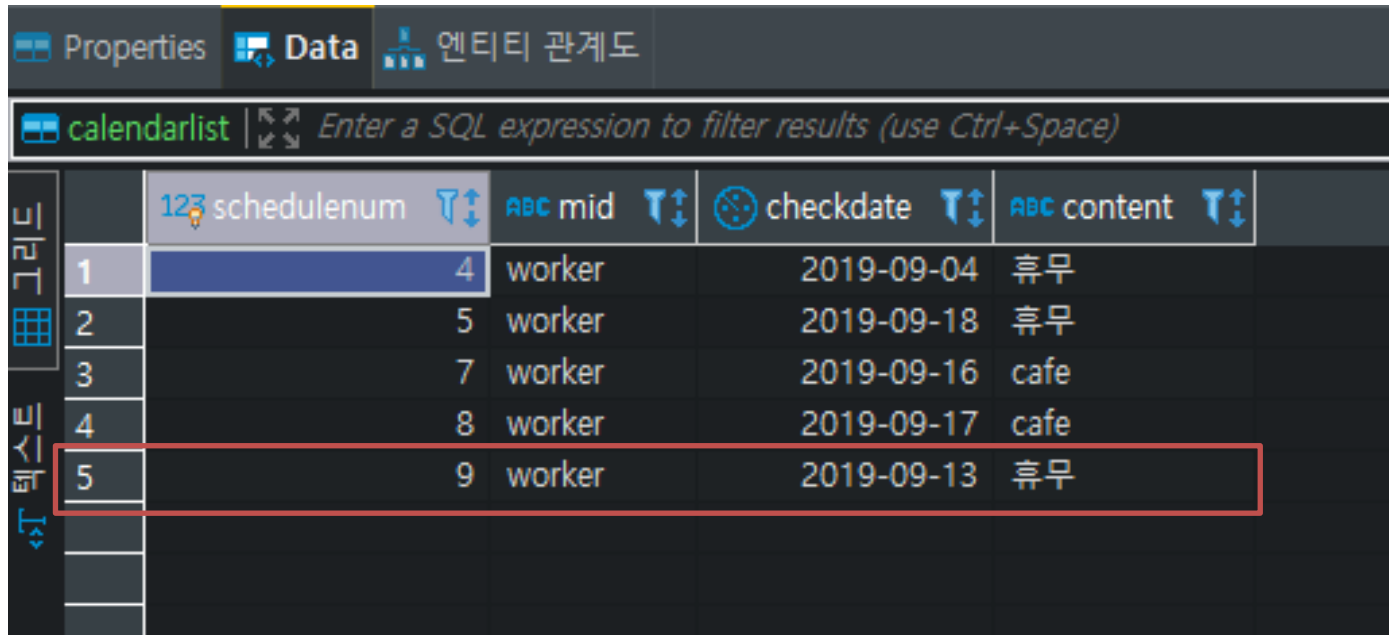
- 해당 날짜 일정 없을 경우 휴무일 입력 진행, 휴무일을 선택하지 않은 경우 alert 출력.

1-B 휴무일 입력 기능(인력, CODE&DB4-2)

```
10     public void insert(CalendarListDTO dto) throws Exception {  
11  
12         DBConnectionMgr mgr = DBConnectionMgr.getInstance();  
13         Connection con = mgr.getConnection();  
14  
15         String sql = "insert into calendarlist values(null,?,?,?)";  
16         PreparedStatement ps = con.prepareStatement(sql);  
17         ps.setString(1, dto.getId());  
18         ps.setString(2, dto.getCheckdate());  
19         ps.setString(3, dto.getContent());  
20  
21         // 4) SQL문 실행 요청  
22         ps.execute();  
23         System.out.println("cdao 4. sql문 실행 요청 ok..");  
24         System.out.println("예약일정 추가완료");  
25  
26         ps.close();  
27         con.close();  
28     } //휴무입력  
29
```

- 휴무일 입력 DAO.

1-B 휴무일 입력 기능(인력, CODE&DB1)



The screenshot shows a software interface with a 'Data' tab selected. Below the tab is a search bar with the text 'calendarlist' and a placeholder 'Enter a SQL expression to filter results (use Ctrl+Space)'. Below the search bar is a table with the following columns: '123 schedulenum', 'ABC mid', 'checkdate', and 'ABC content'. The table contains five rows of data. The fifth row is highlighted with a red box.

	123 schedulenum	ABC mid	checkdate	ABC content
1	4	worker	2019-09-04	휴무
2	5	worker	2019-09-18	휴무
3	7	worker	2019-09-16	cafe
4	8	worker	2019-09-17	cafe
5	9	worker	2019-09-13	휴무

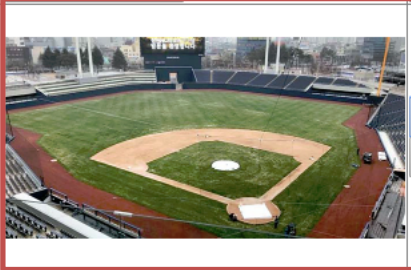
- 5번 row 휴무일이 반영된 DB.

2. 이미지 미리보기/업로드 및 다음 주소입력 API, 검색을 활용한 예약요청 DB처리

02

2-A 이미지 미리보기/업로드 및 주소입력 API(사업자, Browser1)

[사업자정보](#)
[인력등록](#)
[로그아웃](#)
[회원정보수정](#)

제목(모든 입력사항은 필수 입니다.)			
사업자번호:	<input type="text" value="123"/>		
중복확인 내용(사업자번호 입력창 넘어갈시 자동)			
사업장이름:	<input type="text" value="123"/>		
주소	<input type="button" value="우편번호 찾기"/>		
<input type="text" value="06112"/>	<input type="text" value="서울 강남구 논현로123"/>	<input type="text" value="123"/>	
업종:	<input type="text" value="카페"/>		
사업장규모:	<input type="text" value="12"/>	상시 근로인원, 월 매출등	
사업장 소개			
<input type="text" value="123"/>	사업장 소개 및 영업시간 등 상세		
사업장 소개 및 영업시간 등 상세			
<input type="button" value="사업자정보 수정 완료"/>		<input type="button" value="사업자등록 취소"/>	<input type="button" value="예약확인"/>
사업장 사진:	<div style="display: flex; align-items: center;">  <div style="margin-left: 10px; border: 1px solid red; padding: 5px;"> <input type="button" value="파일 선택"/> <input type="text" value="002.jpg"/> <input type="button" value="업로드"/> 파일 선택 후 누르시오. </div> </div>		

2-A 이미지 미리보기/업로드 및 주소입력 API(사업자, CODE&DB1)

```
57
58      //미리보기 평션
59      function readURL(input) {
60          if (input.files && input.files[0]) {
61              var reader = new FileReader(); //파일을 읽기 위한 FileReader객체 생성
62              reader.onload = function(e) {
63                  //파일 읽어들이기를 성공했을때 호출되는 이벤트 핸들러
64                  $('#preview').attr('src', e.target.result);
65                  //이미지 Tag의 SRC속성에 읽어들이 File내용을 지정
66                  //(아래 코드에서 읽어들이인 dataURL 형식)
67              }
68              reader.readAsDataURL(input.files[0]);
69              //File내용을 읽어 dataURL 형식의 문자열로 저장
70          }
71      } //readURL() --
72      //file 양식으로 이미지를 선택(값이 변경) 되었을때 처리하는 코드
73      $("#fileselect").change(function() {
74          //alert(this.value); //선택한 이미지 경로 표시
75          readURL(this);
76      });
77      //미리보기 평션
```

- 미리보기 구현을 위한 CODE
- preview 위치에 선택한 이미지의 경로를 삽입한다.

2-A 이미지 미리보기/업로드 및 주소입력 API(사업자, CODE&DB2)

```
33 //이미지 업로드 폼서
34 $("#imgsubmit").submit(function() {
35     var formdata = new FormData();
36     file = $("#fileselect")[0].files[0];
37     formdata.append("file1", file);
38     $.ajax({
39         url: "upload_result.jsp",
40         data: formdata,
41         processData: false,
42         contentType: false,
43         type: 'POST',
44         success: function(result) {
45             result2 = result.trim();
46             //파일전송완료 후 result 업로드 이미지 파일 경로
47             var src="img/store/"+result2;
48             $("#sImgsrc").val(src);
49             //업로드 후 해당 경로 hidden input에 입력
50         }
51     })
52     $("#joinmemsubmit").attr("disabled",false);
53     //이미지 업로드 후 joinmemsubmit button 활성화
54     return false;
55 }); //submit function
56 //이미지 업로드 폼서
57
```

- COS.jar을 이용한 이미지 업로드 ajax 코드.
- 이미지 업로드 완료 후 사업자 등록버튼 활성화 구현.

2-A 이미지 미리보기/업로드 및 주소입력 API(사업자, CODE&DB3)

```
19 <script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
20 <script src="js/postcode.js"></script>
116 </td></td>
117 </td>
118 <button type="button" onclick="execDaumPostcode()" value="우편번호 찾기">우편번호 찾기</button>
119 <!-- 주소입력 api 실행 버튼 -->
120 </td>
121 </tr>
122 <tr>
123 <td>
124 <input type="text" id="postcode" name="postcode" placeholder="우편번호">
125 <!-- 우편번호 입력창 (주소입력 api에서 가져오기) -->
126 </td>
127 <td>
128 <input type="text" id="roadAddress" name="roadAddress" placeholder="도로명주소">
129 <!-- 주소 입력창 (주소입력 api에서 가져오기) -->
130 </td>
131 <td>
132 <span id="guide " style="color: #999; display: none"></span>
133 <input type="text" id="detailAddress" name="detailAddress" placeholder="상세주소">
134 <!-- 상세주소 사용자 입력 -->
135 </td>
136 </tr>
```

- 다음 주소입력 API구현 부분.

2-B 사업자 예약 요청 처리/출력(사업자, Browser1)

user1님 환영합니다.

For Vacation

사업자정보

인력등록

로그아웃

회원정보수정

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨

- 1-A와 동일.

02

2-B 사업자 예약 요청 처리/출력(사업자, Browser2, 검색기능 참조)

근무가능지역

시/군 서울

구/읍/면 null

소개

1

[더보기](#)

기타정보

테스트 예약

예약하기

- 테스트용 예약 완료.

02

2-B 사업자 예약 요청 처리/출력(사업자, Browser3)

For Vacation

사업자정보

인력등록

로그아웃

회원정보수정

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-06user1	123	2019-09-06	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약대기중
worker2019-09-07user1	123	2019-09-07	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약대기중
worker2019-09-08user1	123	2019-09-08	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약대기중
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약대기중

- 검색시 입력한 값에 대한 예약 진행 결과.

2-B 사업자 예약 요청 처리/출력(사업자, CODE&DB1)

```
6<String regnum= (String)request.getParameter("regnum");
7 //regnum(사업자등록시 입력)을 받아와 해당 사업자의 예약요청 내역 리스트 출력
8 %>
9 <input type="hidden" id="regnum" value="<%=regnum%>">
10<script>
11     $(function() {
12         var regnum =$("#regnum").val();
13         //예약요청 내역 출력을 위한 ajax 전송용 regnum
14         //변수 hidden value에 넣어두어 script내 생성
15     <%
16         ReservationDTO dto = new ReservationDTO();
17         ReservationListDAO dao = new ReservationListDAO();
18         dto.setRegnum(regnum);
19         //id를 가지고 dao.select(dto) 실행
20         //select 사용하여 regnum 해당 data 모두 list화 하여 넘겨주기
21         ArrayList<ReservationDTO> list = dao.selectAllU(dto);
22     %>
23     for (var i = 0; i < <%=list.size()%>; i++) {
24         var check = $("#statecheck"+i).text();
25         if (check=="0") {
26             $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약대기중"+</button>");
27         }else if(check=="1"){
28             $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약확정됨"+</button>");
29         }else{
30             $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약취소됨"+</button>");
31         }//if
32     }//for
33     });
34 </script>
```

- 1-A와 동일하며, 검색을 위해 사용하는 값을 변경.

02

2-B 사업자 예약 요청 처리/출력(사업자, CODE&DB2)

	ABC resnum	ABC regnum	ABC wnum	ABC resdate	ABC type	ABC etc	ABC state	kdate	123 resgroup
1	worker2019-09-16user1	123	1	2019-09-16	cafe	ㄱㄱ	1	2019-09-27-18.13.45 +0900	1
2	worker2019-09-17user1	123	1	2019-09-17	cafe	ㄱㄱ	1	2019-09-27-18.13.45 +0900	1
3	worker2019-09-09user1	123	1	2019-09-09	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
4	worker2019-09-10user1	123	1	2019-09-10	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
5	worker2019-09-11user1	123	1	2019-09-11	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
6	worker2019-09-06user1	123	1	2019-09-06	cafe	테스트 예약	0	2019-09-28-17.02.59 +0900	3
7	worker2019-09-07user1	123	1	2019-09-07	cafe	테스트 예약	0	2019-09-28-17.02.59 +0900	3
8	worker2019-09-08user1	123	1	2019-09-08	cafe	테스트 예약	0	2019-09-28-17.02.59 +0900	3
9	worker2019-09-09user1	123	1	2019-09-09	cafe	테스트 예약	0	2019-09-28-17.02.59 +0900	3

- 6~9 row, 예약 요청이 DB에 반영된 모습.

3. 예약 수락/거절 처리 및 Fullcalendar API

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, Browser1)

사업자등록

인력정보

로그아웃

회원정보수정

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-06user1	123	2019-09-06	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약확정 예약취소
worker2019-09-07user1	123	2019-09-07	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약확정 예약취소
worker2019-09-08user1	123	2019-09-08	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약확정 예약취소
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-28 17:02:59.0	테스트 예약	0	예약확정 예약취소

휴무일 캘린더로

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, Browser2)

사업자등록

인력정보

로그아웃

회원정보수정

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-06user1	123	2019-09-06	cafe	2019-09-28 17:02:59.0	테스트 예약	1	예약확정됨
worker2019-09-07user1	123	2019-09-07	cafe	2019-09-28 17:02:59.0	테스트 예약	1	예약확정됨
worker2019-09-08user1	123	2019-09-08	cafe	2019-09-28 17:02:59.0	테스트 예약	-1	예약취소됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-28 17:02:59.0	테스트 예약	-1	예약취소됨

휴무일 캘린더로

02

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, Browser3)

For Vacation

[사업자등록](#) [인력정보](#) [로그아웃](#) [회원정보수정](#)

< >		today		September 2019		휴무입력	
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
1	2	3	4	5	6	7	
			휴무		cafe	cafe	
8	9	10	11	12	13	14	
					휴무		
15	16	17	18	19	20	21	
	cafe	cafe	휴무				
22	23	24	25	26	27	28	
29	30	1	2	3	4	5	
6	7	8	9	10	11	12	

휴무일:

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(사용자, Browser4)

user1님 환영합니다.

For Vacation

사업자정보

인력등록

로그아웃

회원정보수정

예약번호	사업자번호	예약날짜	업종	예약요청시간	기타	현 상태	출력
worker2019-09-16user1	123	2019-09-16	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-17user1	123	2019-09-17	cafe	2019-09-27 18:13:45.0	ㄱㄱ	1	예약확정됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-10user1	123	2019-09-10	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-11user1	123	2019-09-11	cafe	2019-09-27 18:15:10.0	ㄱㄱㄱㄱ	-1	예약취소됨
worker2019-09-06user1	123	2019-09-06	cafe	2019-09-28 17:02:59.0	테스트 예약	1	예약확정됨
worker2019-09-07user1	123	2019-09-07	cafe	2019-09-28 17:02:59.0	테스트 예약	1	예약확정됨
worker2019-09-08user1	123	2019-09-08	cafe	2019-09-28 17:02:59.0	테스트 예약	-1	예약취소됨
worker2019-09-09user1	123	2019-09-09	cafe	2019-09-28 17:02:59.0	테스트 예약	-1	예약취소됨

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, CODE&DB1)

```
7<%int wnum= Integer.parseInt(request.getParameter("wnum"));
8 //wnum(인력 등록시 pk)를 받아와 해당 인력의 예약 내역 리스트 출력
9 %>
10 <input type="hidden" id="wnum" value="<%=wnum%>">
11<script>
12     $(function() {
13         var wnum =$("#wnum").val();
14         //예약 확정/취소를 위한 ajax 전송용 wnum
15         //변수 hidden value에 넣어두어 script내 생성
16     <%
17         ReservationDTO dto = new ReservationDTO();
18         ReservationListDAO dao = new ReservationListDAO();
19         dto.setWnum(wnum);
20         //wnum를 가지고 dao.select(dto) 실행
21         ArrayList<ReservationDTO> list = dao.selectAll(dto);
22         // select 사용하여 세션id해당 data 모두 list화 하여 넘겨주기
23     %>
24         //state 상태에 따른 버튼 변경
25         for (var i = 0; i < <%=list.size()%>; i++) {
26             var check = $("#statecheck"+i).text();
27             if (check=="0") {
28                 $("#append"+i).append("<button class='statechange' id='stateaccept'>예약확정"+"</button>");
29                 $("#append"+i).append("<button class='statechange' id='staterefuse'>예약취소"+"</button>");
30             }else if(check=="1"){
31                 $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약확정됨"+"</button>");
32             }else{
33                 $("#append"+i).append("<button class='statechange' id='stateaccept' disabled='true'>예약취소됨"+"</button>");
34             }//if
35         }//for
```

- 1-A의 반복, 예약요청 이후 인력의 예약요청 내역 페이지 접속.

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, CODE&DB1)

```
37 //confirm accept
38 $("#reservationvalue").on("click", "#stateaccept", function() {
39 //예약상태 변경 버튼
40 var acceptcheck = confirm("예약을 수락하시겠습니까?");
41 if (acceptcheck) {
42 //예약수락
43 //버튼 누를시 테이블 갯 받기
44 console.log("갯 받기 버튼 눌렀다");
45 var tdArr = new Array(); // 배열 선언
46 var checkBtn = $(this);
47 // checkBtn.parent() : checkBtn의 부모는 <td>이다.
48 // checkBtn.parent().parent() : <td>의 부모이므로 <tr>이다.
49 var tr = checkBtn.parent().parent();
50 var td = tr.children();
51 //console.log("클릭한 Row의 모든 데이터 : "+tr.text());
52 var reserveNum = td.eq(0).text();
53 var regNum = td.eq(1).text();
54 var reserveDate = td.eq(2).text();
55 var type1 = td.eq(3).text();
56 var Kdate = td.eq(4).text();
57 var address1 = td.eq(5).text();
58 var state = 1;
59 //ajax로 버튼값에 따라 전송장소 변경
60 $.ajax({
61 url: "ReservationAccept.jsp",
62 data:{
63 wnum : wnum,
64 resnum : reserveNum,
65 regNum : regNum,
66 reserveDate : reserveDate,
67 type1 : type1,
68 Kdate : Kdate,
69 state : state
70 },
71 success: function(result){
72 location.reload();
73 }//success
74 });//ajax
```

- 예약 확정 버튼 클릭시 작동하는 ajax, 해당 버튼의 row data를 ajax로 전송.

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, CODE&DB1)

```
84 //confirm refuse
85 $(".reservationvalue").on("click", "#staterefuse", function() {
86 //$( "#reservation-confirm" ).click(function(){
87 //예약상태 변경 버튼
88 var refusecheck = confirm("예약을 거절하시겠습니까?");
89 if (refusecheck) {
90 //예약거절
91 console.log("refuse 완료");
92 //버튼 누를시 테이블 갱 받기
93 console.log("값 받기 버튼 눌렀다");
94 var tdArr = new Array(); // 배열 선언
95 var checkBtn = $(this);
96 // checkBtn.parent() : checkBtn의 부모는 <td>이다.
97 // checkBtn.parent().parent() : <td>의 부모이므로 <tr>이다.
98 var tr = checkBtn.parent().parent();
99 var td = tr.children();
100 //console.log("클릭한 Row의 모든 데이터 : "+tr.text());
101 var reserveNum = td.eq(0).text();
102 var regNum = td.eq(1).text();
103 var reserveDate = td.eq(2).text();
104 var type1 = td.eq(3).text();
105 var Kdate = td.eq(4).text();
106 var address1 = td.eq(5).text();
107 var state = -1; //예약거절시 state -1로 변경
108 //ajax로 버튼값에 따라 전송장소 변경
109 $.ajax({
110 url: "ReservationRefuse.jsp",
111 data:{
112 wnum : wnum,
113 resnum : reserveNum,
114 regNum : regNum,
115 reserveDate : reserveDate,
116 type1 : type1,
117 Kdate : Kdate,
118 state : state
119 },
120 success: function(result){
121 location.reload();
122 }//success
123 });//ajax
```

- 예약 취소 버튼 클릭시 작동하는 ajax, 해당 버튼의 row data를 ajax로 전송.

02

3-A 요청 예약 수락/거부 처리 및 캘린더 연동(인력, CODE&DB1)

	ABC resnum	ABC regnum	ABC wnum	ABC resdate	ABC type	ABC etc	ABC state	kdate	123 resgroup
1	worker2019-09-16user1	123	1	2019-09-16	cafe	ㄱㄱ	1	2019-09-27-18.13.45 +0900	1
2	worker2019-09-17user1	123	1	2019-09-17	cafe	ㄱㄱ	1	2019-09-27-18.13.45 +0900	1
3	worker2019-09-09user1	123	1	2019-09-09	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
4	worker2019-09-10user1	123	1	2019-09-10	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
5	worker2019-09-11user1	123	1	2019-09-11	cafe	ㄱㄱㄱㄱ	-1	2019-09-27-18.15.10 +0900	2
6	worker2019-09-06user1	123	1	2019-09-06	cafe	테스트 예약	1	2019-09-28-17.02.59 +0900	3
7	worker2019-09-07user1	123	1	2019-09-07	cafe	테스트 예약	1	2019-09-28-17.02.59 +0900	3
8	worker2019-09-08user1	123	1	2019-09-08	cafe	테스트 예약	-1	2019-09-28-17.02.59 +0900	3
9	worker2019-09-09user1	123	1	2019-09-09	cafe	테스트 예약	-1	2019-09-28-17.02.59 +0900	3

- 6~9 row, state column에 예약 확정/취소 버튼 액션으로 state 변경이 DB에 반영된 모습.

02

기술설명 - 정병수

1. Autocomplete API

1-2. jsonArray

2. Include 활용

3. Toggle버튼 활용

1. Autocomplete API

1. DB처리

```
1 <%@page import="org.json.simple.JSONObject"%>
2 <%@page import="org.json.simple.JSONArray"%>
3 <%@page import="java.util.ArrayList"%>
4 <%@page import="bean.AddressDTO"%>
5 <%@page import="bean.AddressDAO"%>
6 <%@page language="java" contentType="text/html; charset=UTF-8"
7     pageEncoding="UTF-8"%>
8     <!-- 주소데이터 db에서 불러와서 jsonArray로 처리 -->
9 <%
10     AddressDAO addressDAO = new AddressDAO();
11     AddressDTO addressDTO = new AddressDTO();
12     ArrayList<AddressDTO> list = addressDAO.selectAll(addressDTO);
13     JSONArray addressList = new JSONArray();
14
15     for (int i = 0; i < list.size(); i++) {
16         addressList.add(list.get(i).getAddress1());
17     }
18 %>
19 <%=addressList.toJSONString()%>
```

DB에서 select실행한 결과를 JSONArray 타입의 list에 담는 부분
Auto Complete API를 활용하는 화면에서 ajax result값 처리하기 위해
toJSONString만 화면에 출력

1. 배열 및 API 처리

```
/* 주소 auto complete 기능 사용하기 위한 js 배열 정의 */
var list1 = [];
var list2 = [];

/* jsonArray를 db에 저장되어 있는 데이터 가지고 온 뒤 js배열로 담기 */
$("#address1").focus(function() {

    $.ajax({

        url : "../selectregworker.jsp",
        dataType : "json",
        success : function(result) {
            for (var i = 0; i < result.length; i++) {
                list1[i] = result[i];
            }
        }

    });

});
```

JSONArray로 출력한 결과가 JS배열과 console에서는 똑같이 출력되나 형식이 달라 autocomplete기능을 수행하지 못함.

Result의 각 속성을 미리 선언해둔 JS배열 타입의 변수에 삽입.

02

1. 결과

파일 선택 후 누르시오.

경력업종1	cafe ▼
경력기간1(개월)	<input type="text"/>
경력업종2	▼
경력기간2	<input type="text"/>
경력업종3	▼
경력기간3	<input type="text"/>
근무지역(시/군)	시
근무지역(구/읍/면)	강릉시
본인소개	동해시
	삼척시
	속초시
	원주시
	춘천시
	태백시
	고양시
	과천시
	광명시
	광주시
	구리시
	군포시
	김포시
	남양주시

그림과 같이 기능 수행

2. Include 활용

2. Include 활용 예

```
<body>
  <!-- review화면과 worker화면을 한화면에 담으려고 각 각 include한 부분
  컨트롤러를 안써서 이번 프로젝트에서 뭔가 해보려다가 가장 큰 실수 한 부분
  이거 때문에 코드가 복잡하고 퍼포먼스가 굉장히 안좋아짐
  원래 include는 첫줄 menu만 계속 가져오려고 생각해냈던 것.... -->
  <%@include file="../menu.jsp"%>
  <br>
  <br>
  <br>

  <!-- worker화면과 review화면에 있는 값도 같이 insert해줘야하는 부분이 있어서 form택 안에 include 포함 -->
  <form action="insertreservation.jsp">
    <table align="center">
      <tr>
        <td width="100"></td>
        <td><%@include file="../worker/worker.jsp"%></td>
        <td width="100"></td>
        <td><%@include file="../review/review.jsp"%></td>
        <td width="100"></td>
      </tr>
    </table>
  </form>
```

<%@include file=""%>로 코드 활용
미리 작성한 view를 한 화면에 담기 위함.

2. 결과

1 For Vacation

사업자등록 인력등록 로그아웃 회원정보수정

2

전문인력

경력사항
업종 cafe
기간 36개월

근무가능지역
시/군 서울
구/읍/면 서대문구

3

3.2 ★★★★★
●1●2●3●4●5 ★★★★★

댓글입력

등록

worker ★★★★★

worker ★★★★★

worker ★★★★★

1. menu.jsp
2. worker.jsp
3. review.jsp

3. Toggle버튼 활용

3. Toggle버튼 활용 예

```
#dis2 {  
    display: none;  
}
```

```
/* 토글 버튼으로 리뷰 더보기 기능 정의 */  
function() {  
    var flag = 1;  
    $("#btn").on("click", function() {  
        if(flag==1){  
            $(this).html($(this).text().replace($("#btn").text(), '닫기'));  
            flag=0;  
        }else if(flag==0){  
            $(this).html($(this).text().replace($("#btn").text(), '더보기'));  
            flag=1;  
        }  
        $("#dis2").slideToggle("middle");  
    });  
}
```

기본보기를 닫은 상태로 시작

SlideToggle 버튼으로 리뷰 더보기 기능을 구현

Flag를 정의하여 토글 열림 닫힘을 구분

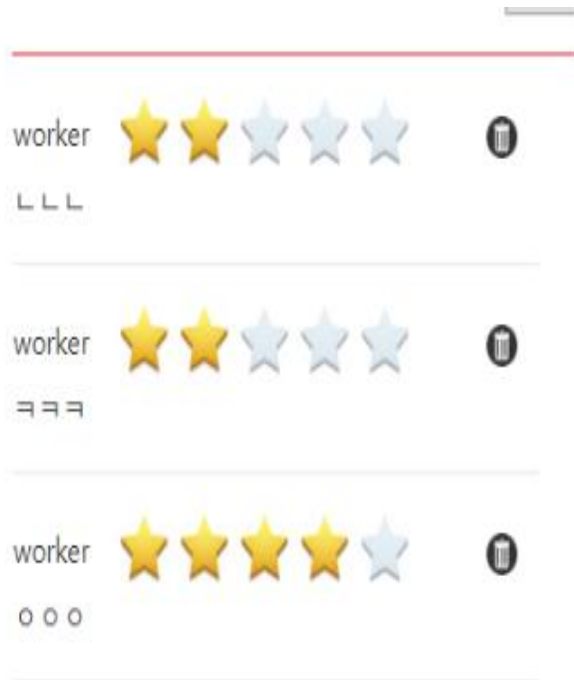
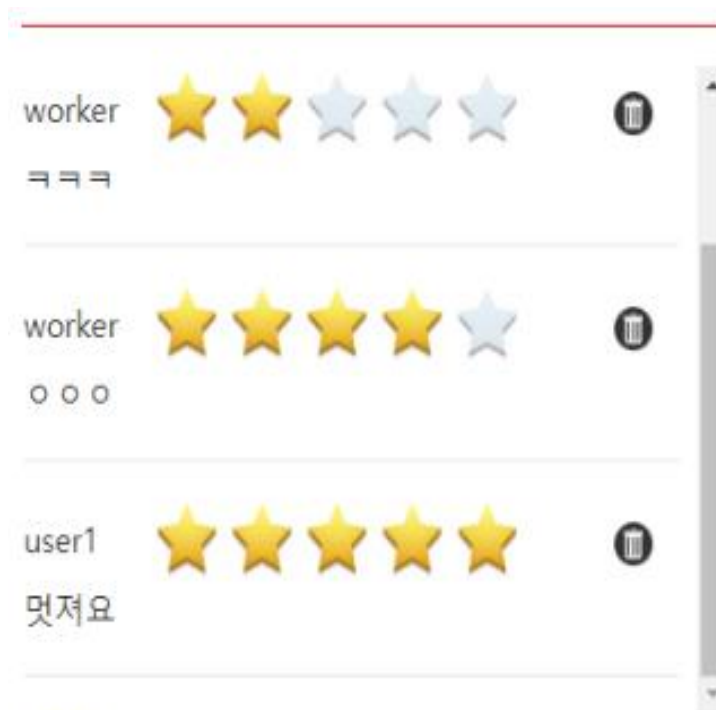
더보기 클릭 시 버튼은 닫기 버튼으로 바뀌도록 정의

3. 출력 구현

```
<!-- 댓글 갯수가 3개보다 많을 경우 나머지 댓글 출력 -->
<%
  if (list2.size() > 3) {
    for (int i = 3; i < list2.size(); i++) {
      workerBoardDTO2 = list2.get(i);
    }
  }
  <tr>
    <td width="64" style="word-break: break-all;"><%=workerBoardDTO2.getMid()%></td>
    <td style="word-break: break-all;">
      <%
        if (workerBoardDTO2.getRating() == 1) {
          %>  <%
        } else if (workerBoardDTO2.getRating() == 2) {
          %>  <%
        } else if (workerBoardDTO2.getRating() == 3) {
          %>  <%
        } else if (workerBoardDTO2.getRating() == 4) {
          %>  <%
        } else if (workerBoardDTO2.getRating() == 5) {
          %>  <%
        }
      %>
    </td>
  </tr>
</pre>
```

이후 4번째 부터 토글버튼으로 출력하기 위한 select

3. 결과

[더보기](#)[닫기](#)

왼쪽 더보기 누르기 전 3개의 리뷰만 출력되는 화면 예

오른쪽 더보기를 누른 후 닫기 버튼으로 바뀌고,
나머지 리뷰가 출력되는 화면 예

**THANK
YOU**