

이민제의 PORTFOLIO



click

# 목차

## 이민제의 PORTFOLIO



### 1, CaxiDogi

- 반려동물을 위한 사료 구매 및 미용, 수영장을 예약 하는 사이트
- 2020.03.27 ~ 2020.05.11(5인 팀프로젝트)

### 2 , Oh My Baby!

- 베이비 시터 구인구직 및 예약사이트
- 2020.02.03 ~ 2020.02.21(4인 팀프로젝트)

# CaxiDogi



HOME ABOUT US CONTACT US



## PRODUCT



2019.10.10

2019.10.10

2019.10.10

2019.10.10

## SHOP



2019.10.10

2019.10.10

2019.10.10

2019.10.10

## POOL



2019.10.10

2019.10.10

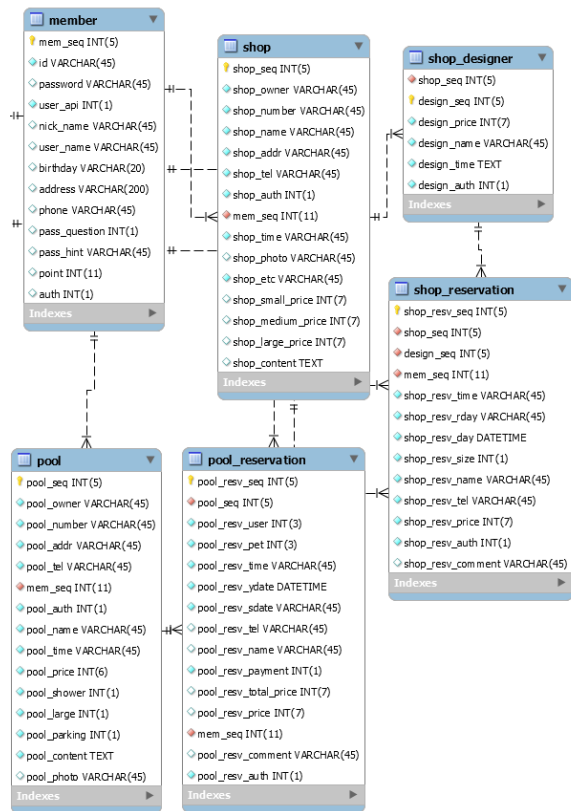
2019.10.10

2019.10.10

# 개발언어 및 환경



## DB MODEL &amp; CALSS



```

cd.com.a.dao
├── admin_noticeDao.java
├── basketDao.java
├── CompanyDao.java
├── MemberDao.java
├── MyPageDao.java
├── MyPageOrderDao.java
├── orderDao.java
├── PoolDao.java
└── ShopDao.java
  
```

```

cd.com.a.daolmpl
├── admin_noticeDaolmpl.java
├── basketDaolmpl.java
├── CompanyDaolmpl.java
├── MemberDaolmpl.java
├── MyPageDaolmpl.java
├── MyPageOrderDaolmpl.java
├── orderDaolmpl.java
├── PoolDaolmpl.java
└── ShopDaolmpl.java
  
```

```

cd.com.a.service
├── admin_noticeService.java
├── basketService.java
├── CompanyService.java
├── MemberService.java
├── MyPageOrderService.java
├── MyPageService.java
├── orderService.java
├── PoolService.java
└── ShopService.java
  
```

```

> cd.com.a.serviceml
├── admin_noticeServiceml.java
├── basketServiceml.java
├── CompanyServiceml.java
├── MemberServiceml.java
├── MyPageOrderServiceml.java
├── MyPageServiceml.java
├── orderServiceml.java
├── PoolServiceml.java
└── ShopServiceml.java
  
```

```

cd.com.a.controller
├── adminMainController.java
├── basketController.java
├── CompanyController.java
├── mainController.java
├── MemberController.java
├── MyPageOrderController.java
├── noticeController.java
├── orderController.java
├── PoolController.java
├── SampleController.java
├── SellerPoolController.java
└── ShopController.java
  
```

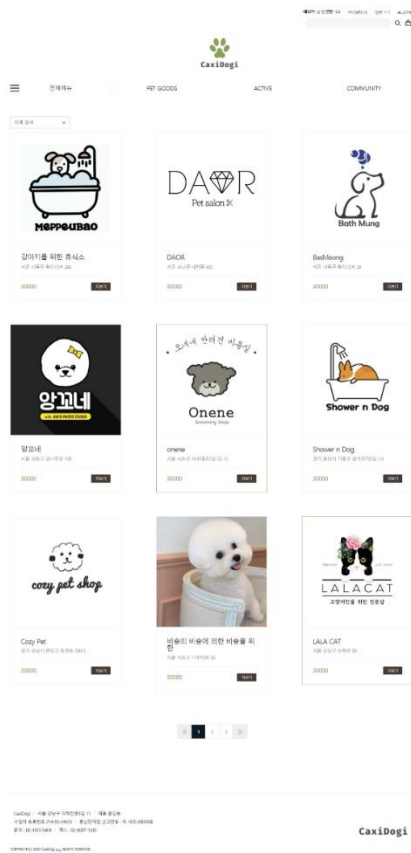
```

cd.com.a.model
├── admin_noticeDto.java
├── AdminPoolParam.java
├── adminShopParam.java
├── amountParam.java
├── basketDto.java
├── companyDto.java
├── groundDto.java
├── groundResvDto.java
├── kakaoAmount.java
├── KakaoCancelDto.java
├── KakaoPayApproveDto.java
├── kakaoSelectDto.java
├── memberDto.java
├── myBuyParam.java
├── order_PrdParam.java
├── order_PrdParamList.java
├── orderDetailParam.java
├── poolDto.java
└── PoolListParam.java
  
```

```

├── poolParam.java
├── poolResvDto.java
├── poolResvParam.java
├── productDto.java
├── ProductListParam.java
├── ProductParam.java
├── productSaleDto.java
├── productSuperDto.java
├── saleBasketParam.java
├── SelectedCardInfoDto.java
├── shopDesignerDto.java
├── shopDto.java
├── shopListParam.java
├── shopPagingParam.java
├── shopResvDto.java
├── shopSellerPagingParam.java
├── shopSellerResvParam.java
└── shopShowResvParam.java
  
```

# 강아지 미용 shop 페이지



## 글목록, 검색, 페이징 쿼리

```
<select id="getShopList" parameterType="cd.com.a.model.shopListParam" resultType="cd.com.a.model.shopDto">
select @rownum:=@rownum+1 as rownum, shop_seq, shop_owner, shop_number, shop_name, shop_addr, shop_tel, shop_auth,
mem_seq, shop_time, shop_photo, shop_etc, shop_small_price, shop_medium_price,
shop_large_price, shop_content
from shop
where (@rownum:=0)=0 and shop_auth=1
<if test="shopAddr == 1">
and shop_addr like concat('%',{shopAddrSeo},'%')
</if>
<if test="shopAddr == 2">
and shop_addr like concat('%',{shopAddrKyeong},'%')
</if>
<if test="shopAddr == 3">
and not shop_addr like concat('%',{shopAddrSeo},'%') and not shop_addr like concat('%',{shopAddrKyeong},'%')
</if>
order by shop_seq desc
limit #{start}, #{recordCountPerPage}
</select>

<select id="getShopCount" resultType="java.lang.Integer">
select count(*) from shop
where shop_auth=1
<if test="shopAddr == 1">
and shop_addr like concat('%',{shopAddrSeo},'%')
</if>
<if test="shopAddr == 2">
and shop_addr like concat('%',{shopAddrKyeong},'%')
</if>
<if test="shopAddr == 3">
and not shop_addr like concat('%',{shopAddrSeo},'%') and not shop_addr like concat('%',{shopAddrKyeong},'%')
</if>
</select>
```



## 기능소개

- 승인 받은 shop이 9개씩 화면에 나타납니다. 지역별(서울, 경기, 기타)로 선택하여 확인 가능합니다. 상세보기에 shop 관련 정보 확인 가능합니다.
- 검색용 param 을 따로 만들었습니다.
- Javascript change 이벤트로 결과 값이 변했을 때마다 원하는 화면을 보여주게 했습니다.

# Shop 디테일 & 예약 페이지



CaxiDogi

HOME > ACTIVE > DAOR

DAOR

DAOR  
Pet salon &

소개글 DAOR에 보신들 부탁하지 않게 해보겠습니다.

기본비용 가격

10,000 ~ 18,000

전화번호 02-4563-7815

주소 서울 강남구 대치동 605

May 2020						
Su	Mo	Tu	We	Th	Fr	Sa
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	31

예약시간 - 2500원

11:00 12:00 14:00 16:00

예약하기



기능소개

- 예약 시간은 append를 이용해서 보여줬습니다.
- 예약의 경우 날짜를 클릭 후, 디자이너를 선택하면 ajax로 JSON으로 예약된 시간을 가져와서 contains를 통해 겹치는 값이 있으면 클릭하지 못하게 하였습니다.



예약된 시간 가져오는 쿼리

```
<select id="getShopResv" parameterType="cd.com.a.model.shopResvDto" resultType="java.lang.String">
  select shop_resv_time
  from shop_reservation
  where shop_seq=#{shop_seq} and design_seq=#{design_seq} and shop_resv_rday=#{shop_resv_rday} and shop_resv_auth=0
</select>
```



예약된 시간을 클릭 못하게 하고 동적할당 하는 스크립트 & 예약 된 시간 보내는 컨트롤러

```
$(document).ready(function(){
    $("#sel").on("change", function(){
        //alert("seq" + $("#sel").val());
        var design_time = $("option:selected", this).attr('design_time');
        //alert("design_time:" + design_time);
        var design_seq = $("option:selected", this).attr('design_seq');
        //alert("design_seq:" + design_seq);
        var sTime = design_time.split("/");
        //alert("sTime:" + sTime);

        if($("#sel") > option:selected).val() != -1){
            $("#mtime").hide();
        }

        $("#mtime").remove();
        for(var i=0; i<sTime.length; i++){
            $("#mtime").append("<li class='timeBtn'><button type='button' class='btn' id='sbtn' value='"+sTime[i]+">"+sTime[i]+"/></li>");
        }
        var shop_seq = $("#shop_seq").val();
        var shop_resv_rday = $("#shop_resv_rday").val();
        var allData = ["design_seq":design_seq, "shop_seq":shop_seq, "shop_resv_rday":shop_resv_rday];

        $("#mtime").show();
        $.ajax({
            url: 'getResvTime.do',
            type: 'post',
            data: allData,
            success: function(data){
                str = JSON.stringify(data);
                var obj_length = Object.keys(str).length;
                var v = new Array();

                for (key in str){
                    v[key] = str[key];
                }
                for(var i=0; i<v.length; i++){
                    $("#button:contains '"+v[i]+"')").addClass('cant');
                    $("#button:contains '"+v[i]+"')").attr("disabled","disabled");
                }
            }
        });
        $(document).on('click', '.btn', function(){
            var pcode = $(this).val(); //리더가 element id values를 가져오는
            $(this).attr("name", "shop_resv_time");
            $("#shop_resv_time").val(pcode);
        });
    },
    error: function(){
        alert('err');
    }
});
});
```

## user shop 예약 내역 확인 페이지



 예약된 내역 가져오는 쿼리 & 날짜 계산해서 취소 하는 쿼리



## 기능소개

- 예약한 샵, 예약 내역 확인 가능합니다.
- join을 이용하여 해당 샵의 필요한 정보들을 다 불러왔습니다.
- xml에서 날짜를 확인한 후 값을 가져와서 사용 이를 전부터 취소 가능하게 했습니다.
- 변경, 취소 완료 됐을 경우 ajax를 사용하여 사용자에게 알려줬습니다.
- 페이지도 다 구현했습니다.

```
<select id="showShopResv" parameterType="cd.com.a.model.shopPagingParam" resultType="cd.com.a.model.shopShowResvParam">
  select @rownum:=@rownum+1 as rownum, r.*, s.shop_name, s.shop_addr, s.shop_tel, s.shop_time, s.shop_photo, d.design_name
  from shop_reservation r
  join shop s on r.shop_seq = s.shop_seq
  join shop_designer d on r.design_seq = d.design_seq
  where (@rownum:=0)=0 and shop_resv_auth=0 and r.mem_seq=#{mem_seq}
  order by shop_resv_rday asc
  limit #{start},#{recordCountPerPage}
</select>
```

```
<select id="shopCancelTimeCheck" parameterType="cd.com.a.model.shopResvDto" resultType="java.lang.Integer">
  select count(shop_resv_rday) from shop_reservation
  where shop_seq=#{shop_seq} and shop_resv_auth=0 and shop_resv_seq=#{shop_resv_seq}
  <![CDATA[and shop_resv_rday > curdate()+ INTERVAL 2 day]]>
</select>
```

```
cancelShopResv").click(function(){
var shop_seq = $(this).attr("shop_seq");
var shop_resv_auth = $(this).attr("shop_resv_auth");
var shop_resv_seq = $(this).attr("shop_resv_seq");
Swal.fire({
  title: '취소하시겠습니까?',
  text: "",
  icon: 'warning',
  showCancelButton: true,
  confirmButtonColor: '#3085d6',
  cancelButtonColor: '#d33',
  confirmButtonText: '네'
}).then((result) => {
  if (result.value) {
    $.ajax({
      url: "/cancelShopResv.do",
      type: 'post',
      data: {"shop_seq" : shop_seq,
            "shop_resv_auth" : shop_resv_auth,
            "shop_resv_seq" : shop_resv_seq},
      success: function(data){
        //alert("success");
        if(data == 'ok'){
          Swal.fire({
            title: '취소되었습니다!',
            text: '',
            icon: 'success'
          }).then(function(){
            window.location="showShopResv.do");
          })
        }else if(data == 'no'){
          alert("예약 날짜로부터 이전전은 취소가 불가능합니다");
        }
      }
    },
    error: function(e){
      alert("err");
    }
  )
});
});
```



## 예약 취소 하는 js



## seller shop 예약 내역 확인 페이지



## 셀러페이지

## 업체 등록 정보

Shop  
Pool

## 예약정보관리

Shop 예약  
Pool 예약

## 오늘 예약 현황

Pool

## 미용샵 예약 리스트

샵 선택

전체

업체 번호	샵 이름	예약 이용일	시간	예약자 이름	건종	디자이너	상태
21	DAOR	2020-05-13	1200	배타미	중형건	임시완	예약
21	DAOR	2020-05-13	1400	차현	소형건	임시완	예약
15	비송의 비송에 의한 비송을 위한		2020-05-13	1800	배타미	소형건	현승희 예약취소

&lt; 1 &gt;



해당 셀러의 샵 예약된 내역 가져오는 쿼리

```

<select id="getSellerShopResvList" parameterType="cd.com.a.model.shopSellerPagingParam" resultType="cd.com.a.model.shopSellerResvParam" >
select * from(
select @rownum:=@rownum+1 as rownum, r.*, d.design_name, s.shop_name, s.shop_addr, s.shop_tel, s.shop_time, s.shop_photo, s.mem_seq as seller_seq
from shop_reservation r
join shop s on r.shop_seq = s.shop_seq
join shop_designer d on r.design_seq = d.design_seq
where (@rownum=>0) and s.mem_seq=#{memSeq}
<if test="shopSeq != null and shopSeq != -1">
and r.shop_seq=#{shopSeq}
</if>
<if test="shopResvAuth != null and shopResvAuth != -1">
and r.shop_resv_auth=#{shopResvAuth}
</if>
)fsub
order by fsub.shop_resv_day desc
limit #{start}, #{recordCountPerPage}
</select>

<select id="getSellerResvCount" parameterType="cd.com.a.model.shopSellerPagingParam" resultType="java.lang.Integer">
select count(*) from(
select @rownum:=@rownum+1 as rownum, r.*, d.design_name, s.shop_name, s.shop_addr, s.shop_tel, s.shop_time, s.shop_photo, s.mem_seq as seller_seq
from shop_reservation r
join shop s on r.shop_seq = s.shop_seq
join shop_designer d on r.design_seq = d.design_seq
where (@rownum=>0) and s.mem_seq=#{memSeq}
<if test="shopSeq != null and shopSeq != -1">
and r.shop_seq=#{shopSeq}
</if>
<if test="shopResvAuth != null and shopResvAuth != -1">
and r.shop_resv_auth=#{shopResvAuth}
</if>
)fsub
order by fsub.shop_resv_day desc
</select>

```



## 기능소개

- Join을 이용해서 각 샵의 정보, 디자이너 정보를 가져와서 셀러가 소유한 샵에 대한 예약내역을 보여줍니다.
- 예약 상태(예약, 예약취소)에 따라서도 검색이 가능합니다.
- 마찬가지로 Javascript change 이벤트로 결과 값이 변했을 때마다 원하는 화면을 보여주게 했습니다.
- 예약 내역 디테일도 확인 가능합니다.
- 페이지도 다 구현했습니다.

# Oh My Baby!



Oh My Baby

Service 고객센터



방문해주셔서 감사합니다

우리 Oh My baby는 2020년 업계 최초로 비드캠프 인증(BA)을 획득한 영유아 서비스 전문 브랜드입니다.  
국내 영유아 보육/직아 서비스 시스템을 선도해서 대중화하였으며 높은 인지도와 고객들의  
영유아 관련 다양한 서비스를 제공하고 있습니다

회사 소개 보기

- 베이비 시터 구인구직 및 예약사이트  
2020.02.03 ~ 2020.02.21(4인 팀프로젝트)

# 개발언어 및 환경



# DB MODEL & CALSS



BABY_MEMBER						
Table logical name						
Num	PK	AI	FK Null	Logical Name	Name	Type
1	✓				ID	VARCHAR2(300)
2					PASSWORD	VARCHAR2(300)
3					NAME	VARCHAR2(200)
4			✓		AGE	NUMBER(30)
5			✓		GENDER	VARCHAR2(50)
6			✓		ADDRESS	VARCHAR2(800)
7			✓		MONEY	VARCHAR2(2000)
8			✓		PHONE_NUMBE	VARCHAR2(300)
9			✓		BANK_ACCOUNT	VARCHAR2(2000)
10			✓		BANK_ACCOUNT	VARCHAR2(2000)
11					AUTH	NUMBER(30)
12			✓		POINT	VARCHAR2(2000)
13			✓		CUSTOMER_RE	DATE
14					REGISTER_DAT	DATE
15			✓		UN_REGISTER	DATE
16			✓		LICENSE	VARCHAR2(200)
17			✓		WANT_PAY	VARCHAR2(500)
18			✓		CAREER	VARCHAR2(400)
19			✓		HIRE_DATE	DATE
20			✓		EXPIRE_DATE	DATE
21			✓		PROFILE_PHOTO	VARCHAR2(2000)
22			✓		INTRODUCE	VARCHAR2(3000)
23			✓		START_WORK_	DATE
24			✓		END_WORK_D	DATE
25			✓		WANT_DATE	VARCHAR2(500)
26			✓		WANT_LOCAL	VARCHAR2(400)
27			✓		WANT_TIME	VARCHAR2(300)

SITTER_BOARD						
Table logical name						
Num	PK	AI	FK Null	Logical Name	Name	Type
1	✓				SEQ	NUMBER(30)
2			✓		ID	VARCHAR2(300)
3			✓		TITLE	VARCHAR2(200)
4			✓		CONTENT	VARCHAR2(2000)
5			✓		FILE_NAME	VARCHAR2(1000)
6			✓		ORIGINAL_FILE	VARCHAR2(2000)
7			✓		READCOUNT	NUMBER(30)
8			✓		REGISTER_DAT	DATE

SITTER_BOARD_COMMENT						
Table logical name						
PK	AI	FK Null	Logical Name	Name	Type	
	✓			SEQ	NUMBER(30)	
		✓		BOARD_COMMI	VARCHAR2(2000)	
		✓		REGISTER_DAT	DATE	

- admin
- commission
- db
- employee
- file
- mail
- main
- mypage
- sitterboard
  - AdminSitterBoardSearchPagingServlet.java
  - SitterBoardCommentDto.java
  - SitterBoardCommentServlet.java
  - SitterBoardDao.java
  - SitterBoardDeleteServlet.java
  - SitterBoardDetailServlet.java
  - SitterBoardDto.java
  - SitterBoardFileDownloaderServlet.java
  - SitterBoardSearchPagingServlet.java
  - SitterBoardUpdateServlet.java
  - SitterBoardWriteServlet.java
  - SitterNoServlet.java
  - SitterOkServlet.java
  - SitterWriteUpdateServlet.java
- sql

## 베이비 시터 구인 구직 게시판



글목록, 페이징, 검색 쿼리

번호	구분	제목	작성자	등록일	조회수
1		시터 지원합니다	crend819	2020-05-23 14:26:47.0	4
2		베이비 시터 지원합니다.	crend819	2020-02-19 20:02:25.0	6
3		상제 된 곡입니다	crend819	2020-02-19 16:27:42.0	2
4	localhost:8090 내용: 관리자 권한입니다.			0-02-18 16:29:06.0	0
5				0-02-18 16:28:59.0	0
6				0-02-18 16:28:51.0	2
7				0-02-17 12:39:15.0	17
8				0-02-17 11:53:47.0	23
9		시터 지원합니다	222	2020-02-16 04:16:43.0	6
10		시터 지원합니다	222	2020-02-16 03:58:02.0	33

확인



## 기능소개

- 시터로 가입한 사람만 글 작성 가능합니다.
- 이중 쿼리로 검색과 페이징을 구현했습니다.
- auth로 관리하여 로그인과 본인과 관리자 auth가 아닌 사람은 게시글을 읽을 수 없게 했습니다.

```
// 베이비 시터 글목록
public List<SitterBoardDto> getSitterBoardPagingList(String choice, String searchWord, int page) {
    String sql = " SELECT SEQ, ID, TITLE, CONTENT, FILE_NAME, "
        + " ORIGINAL_FILE_NAME, READCOUNT, REGISTER_DATE, DEL "
        + " FROM ";
    sql += "(SELECT ROW_NUMBER()OVER(ORDER BY REGISTER_DATE DESC) AS RNUM, "
        + " SEQ, ID, TITLE, CONTENT, FILE_NAME, "
        + " ORIGINAL_FILE_NAME, READCOUNT, REGISTER_DATE, DEL "
        + " FROM SITTER_BOARD ";
    String sqlWord = "";
    if(choice.equals("title")) {
        sqlWord = " WHERE TITLE LIKE '%" + searchWord.trim() + "%' ";
    } else if(choice.equals("writer")) {
        sqlWord = " WHERE ID LIKE '%" + searchWord.trim() + "%' ";
    } else if(choice.equals("content")) {
        sqlWord = " WHERE CONTENT LIKE '%" + searchWord.trim() + "%' ";
    }
    sql += sqlWord;
    sql += " ORDER BY REGISTER_DATE DESC ";
    sql += " WHERE RNUM >= ? AND RNUM <= ? ";
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    List<SitterBoardDto> list = new ArrayList<SitterBoardDto>();
    int start, end;
    start = 1 + 10 * page;
    end = 10 + 10 * page;
    try {
        conn = DBConnection.getConnection();
        System.out.println("1/6 getSitterBoardPagingList success");
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, start);
        pstmt.setInt(2, end);
        System.out.println("2/6 getSitterBoardPagingList success");
        rs = pstmt.executeQuery();
        System.out.println("3/6 getSitterBoardPagingList success");
        while(rs.next()) {
            int i = 1;
            SitterBoardDto dto = new SitterBoardDto(rs.getInt(i++), rs.getString(i++), rs.getString(i++), rs.getString(i++), rs.getString(i++),
                rs.getString(i++), rs.getInt(i++), rs.getString(i++), rs.getInt(i++));
            list.add(dto);
        }
        System.out.println("4/6 getSitterBoardPagingList success");
    } catch (SQLException e) {
        System.out.println("getSitterBoardPagingList fail");
        e.printStackTrace();
    } finally {
        DBClose.close(pstmt, conn, rs);
    }
    return list;
}
```

# 베이비 시터 구인 구직 글쓰기 페이지



Sitter

제목

제목을 입력해 주세요

작성자

csrf819@naver.com

내용

내용을 입력해 주세요

파일첨부

파일 선택 선택된 파일 없음

등록 취소



## 기능소개

- Multipart를 이용해서 파일업로드를 구현했습니다.
- 파일 이름이 겹칠 수 있기때문에 당사자가 올린 파일이름과, 현재시간으로 파일이름을 바꿔서 따로 저장했습니다.

```
public void processFunc(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    req.setCharacterEncoding("utf-8");
    String type = req.getParameter("type");
    System.out.println("type: " + type);
    String id = "";
    String title = "";
    String content = "";
    String fname = ""; // 사진명 생성

    String filename = ""; // 글자 생성

    if (type.equals("write")) {
        // 글자 생성
        // 글자 길이에 제한이 없음
        String savePath = getServletContext().getRealPath("upload");
        System.out.println("savePath: " + savePath);
        String yourTempDir = savePath;
        int yourMaxRequestSize = 100 * 1024 * 1024; // 우리가 올릴때 최대 사이즈 1 Mbyte으로
        int yourMaxMemorySize = 100 * 1024; // 1 Kbyte

        boolean isMultipart = ServletFileUpload.isMultipartContent(req);

        if (isMultipart == true) {
            // fileitem 생성
            DiskFileItemFactory factory = new DiskFileItemFactory(); // fileitem 생성
            factory.setSizeThreshold(yourMaxMemorySize); // 메모리에서 올릴
            factory.setRepository(new File(yourTempDir)); // 파일들 생성해두고 (지정된 경로에 올릴것임)

            ServletFileUpload upload = new ServletFileUpload(factory);
            upload.setSizeMax(yourMaxRequestSize); // 전송 크기, 1000000 (1Mbyte) 넘지않도록 제한

            // list MultipartItem -> id, title, content, file
            List<FileItem> items = null;
            try {
                items = upload.parseRequest(req);
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            // 전송된 데이터가 null이 아닐때 while
            Iterator<FileItem> it = items.iterator();

            // while
            while (it.hasNext()) {
                FileItem item = it.next();

                if (item.isFormField()) { // id, title, content
                    if (item.getFieldName().equals("id")) {
                        id = item.getString("id");
                    } else if (item.getFieldName().equals("title")) {
                        title = item.getString("title");
                    } else if (item.getFieldName().equals("content")) {
                        content = item.getString("content");
                    }
                } else { // fileupload로 올린것만 올림
                    if (item.getFieldName().equals("fileupload")) {
                        // fileupload 이름으로 items에 올림
                        fname = (new Date().getTime()) + "";

                        filename = processUploadFile(item, savePath, fname);
                    }
                }
            }
        } else {
            System.out.println("multipart가 아님");
        }

        SitterBoardDao dao = SitterBoardDao.getInstance();
        boolean isS = dao.writeSitterBoard(new SitterBoardDto(id, title, content, fname, filename));
        dao.insertComment(id);
        System.out.println("isS: " + isS);
        resp.sendRedirect("process.jsp?type=write&isS=" + isS);

        // forward("sitterwrite.jsp", req, resp);

        if (type.equals("moveSitterBoardList")) {
            resp.sendRedirect("sitterBoard.jsp");
        }
    }
}
```



## 파일 업로드 servlet

```
// upload 함수
// insert, delete, select, update
// update가 가장 어려움 해보면 -> 제목, 파일 등 바꾸지 못하게
public String processUploadFile(FileItem fileitem, String dir, String fname) throws IOException {
    String filename = fileitem.getName(); // 경로 (서버(폴더)명, 클라이언트명) + 파일명으로 불러옴
    long sizeInBytes = fileitem.getSize();

    // 파일명 생성
    if (sizeInBytes > 0) { // d:\tmp\adc.txt d:\tmp\adc.txt
        int idx = filename.lastIndexOf("\\"); // 서버에서 올린 파일이름에 마지막에 있는 지점을 찾아옴
        if (idx == -1) {
            // 오른쪽에서부터 (\\) 즉 d:\tmp\adc.txt 이렇게 되어있으면
            idx = filename.lastIndexOf("/");
        }
        // 파일명하고 확장자만 코드로 뽑음
        filename = filename.substring(idx + 1); // 시작하는 부분의 줄바꿈을 잘 하고 써야 함
        // 예: \adc.txt 에서 abc.txt만 갖고올것
        // 우리가 올릴때는 다 내려와서, 파일명

        // File uploadFile = new File(dir, fname + ".tmp");

        File uploadFile = new File(dir, fname);
        try {
            fileitem.write(uploadFile);
        } catch (IOException e) {
            //
        }
        return filename; // 파일명
    }
}
```

# 베이비 시터 구인 구직 게시판 디테일 페이지



승인 거절

베이비 시터 지원합니다.

파일 : 이민제이력서.docx

등록일 2020-02-19

작성자인cred819

조회수4

안녕하세요 잘 부탁드립니다.

관리자  
답격되었습니다.

Comments

관리자 권한입니다.

등록



## 기능소개

- 관리자는 시터 지원자의 글, 이력서를 본 후 승인 및 거절하고 댓글을 달수있게 구현하였습니다.
- 승인 된 시터는 돌봄 신청 페이지에 보이게 되고 거절된 시터는 로그인시 바로 거절되었다고 알림창이 뜹니다.



## 파일 다운로드servlet, 댓글 다는 쿼리

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    System.out.println("file doGet");
    String filename = req.getParameter("filename");
    String sseq = req.getParameter("seq");
    int seq = Integer.parseInt(sseq);
    String type = req.getParameter("type");
    String ofilename = req.getParameter("ofilename");

    System.out.println("filename:" + filename);

    if(type.equals("download")) {
        BufferedOutputStream out = new BufferedOutputStream(resp.getOutputStream());
        String filepath = "";

        // tomcat(Server)
        filepath = mConfig.getServletContext().getRealPath("/upload");

        // 클라이언트
        filepath = "d:\\tmp";
        //filepath = filepath + "\\" + filename + ".temp";
        filepath = filepath + "\\" + filename;
        System.out.println("파일명: " + filepath);

        File f = new File(filepath); // 파일이 생성되는 부분을 여백
        // 이 파일이 있는지 먼저 체크함.

        if(f.exists() && f.canRead()) {
            //System.out.println("파일 읽음.");
            resp.setHeader("Content-Disposition", "attachment; filename=" + ofilename + ".");
            resp.setHeader("Content-Transfer-Encoding", "binary");
            resp.setHeader("Content-Length", "" + f.length()); // 파일의 크기
            resp.setHeader("Pragma", "no-cache"); // 캐시를 사용하지 않음
            resp.setHeader("Expires", "-1"); // 기한

            //파일 읽기, 차례대로
            BufferedInputStream fileInput = new BufferedInputStream(new FileInputStream(f));
            byte buffer[] = new byte[BUFFER_SIZE]; // 버퍼 크기

            int read = 0;
            while( (read = fileInput.read(buffer)) != -1) {
                out.write(buffer, 0, read);
            }
            fileInput.close();
            out.flush();

            //resp.setHeader("Content-Disposition", "attachment; filename=" + originalFileName + ".");
        }
    }

    // 댓글 달기
    public boolean comment(int seq, SitterBoardCommentDto cDto) {
        String sql = " INSERT INTO SITTER_BOARD_COMMENT "
            + " (SEQ, BOARD_COMMENT, REGISTER_DATE) "
            + " VALUES(?, ?, SYSDATE) ";

        Connection conn = null;
        PreparedStatement psmt = null;

        int count = 0;
        System.out.println("===== \n");

        try {
            conn = DBConnection.getConnection();
            System.out.println("1/6 comment success");
            psmt = conn.prepareStatement(sql);
            psmt.setInt(1, cDto.getSeq());
            psmt.setString(2, cDto.getComment());
            System.out.println("2/6 comment success");

            count = psmt.executeUpdate();
            System.out.println("3/6 comment success");

        } catch (SQLException e) {
            System.out.println("comment fail");
            e.printStackTrace();
        } finally {
            DBClose.close(psmt, conn, null);
        }

        return count>0?true:false;
    }
}
```

# Node.js로 만들어 보는 로그인



node mysql

[home](#) [login](#) [register](#)

## login project

혼자 해보는 node.js 로그인, 게시판 프로젝트

수료 막바지에 node.js를 이틀정도 배웠습니다. 너무 짧게 배워서 제대로 알고 있는 것 같지 않아서 수료 뒤에 혼자 공부를 시작했습니다.  
그리고 또, javascript 부분이 부족하다고 느껴 시작한 아직 진행 중인 작은 프로젝트 입니다.



## 로그인 페이지



node mysql

home bbb님 환영합니다. logout 계시판

node mysql

home login register

Login Form

ID

Password

Login



## 기능소개

- 비밀번호가 틀렸는지 확인과 로그인을 하면  
세션에 저장하고 있습니다.



## 로그인 코드

```
exports.loginAf = async (req, res) => {
  console.log('loginAf 들어옴');

  try {
    const {id, pwd} = req.body;
    if(!id || !pwd){
      return res.status(400).render('../views/auth/login', {
        message: 'Please provide an id and password'
      })
    }
    db.query('SELECT * FROM member_ WHERE id = ?', [id], async(error, result) => {
      console.log(result);
      //console.log('result: ' + result[0].id);

      if(!result || !(await bcrypt.compare(pwd, result[0].pwd))){
        res.status(401).render('../views/auth/login', {
          message: 'id or password is incorrect '
        })
      }

      // 세션
      else if(result[0].id != 0){

        console.log('uid : ' + result[0].id);

        req.session.uid = result[0].id;
        req.session.isLogined = true;
        req.session.save(function() {
          res.redirect('/');
          console.log('session id: ' + req.session.uid);
        })
      }
    })
  } catch (error) {
    console.log(error);
  }
}
```

# 회원가입 페이지



node mysql

[home](#) [login](#) [register](#)

Register Form

ID

bbb

NAME

bbb

Email Address

bbb@bbb.bbb

Password

\*\*\*

PasswordConfirm

\*\*\*

Register User

That id is already in use

Password do not match



## 기능소개

- 회원가입 진행 시 있는 이미 있는 아이디와 입력한 비밀번호가 틀리면 알려주고 있습니다.



## 회원가입 코드

```
exports.register = (req, res) => {
  console.log(req.body); // form에서 보낸걸 읽는다는거 같음

  // const name = req.body.name;
  // const email = req.body.email;
  // const password = req.body.password;
  // const passwordConfirm = req.body.passwordConfirm;
  // 위에 코드랑 같은 말
  const { id, name, email, pwd, passwordConfirm } = req.body;

  db.query('SELECT id FROM member_ WHERE id = ?', [id], async (error, result) =>{
    console.log('=====');

    if(error){
      console.log(error);
    }

    if( result.length > 0){
      console.log('id check: ');

      return res.render('../views/auth/register',{
        message: 'That id is already in use'
      });
    } else if(pwd !== passwordConfirm){
      return res.render('../views/auth/register',{
        message: 'Password do not match'
      });
    }

    let hashedPassword = await bcrypt.hash(pwd, 8);
    console.log(hashedPassword);

    db.query('INSERT INTO member_ SET ?', {id: id, name: name, email: email, pwd:hashedPassword }, (error, result)->{
      if(error){
        console.log(error);
      }else {
        console.log(result);

        return res.render('../views/auth/register',{
          message: 'User registered'
        });
      }
    })
  });
}
```

감사합니다! |



slslflfl819@naver.com