

ShoesProject



@reallygreatsite

목차

- 1 주제 / 주 타겟 / 선정이유
- 2 팀원 [일정 / 역할]
- 3 사용된 스킬 / 언어
- 4 주요 코드와 시연
- 5 앞으로 추후 방향
- 6 각자의 느낀 점



주제 / 주 타겟 / 선정이유

- 주제: 한정판 신발을 사고 파는 이커머스
- 주 타겟: 한정판 신발을 사고 싶은 고객과 팔고 싶은 고객
- 주제 선정 이유: 한정판 신발을 사고 싶은 고객들의 리즈를 충족시키기 위해 프로젝트를 기획하게 되었습니다.

팀원 [일정 / 역할]

이름	역할	담당	브랜치
김경주	백엔드 및 프론트엔드 작업	팀장	김경주
안태섭	백엔드 및 프론트엔드 작업	팀원 및 git 관리	안태섭
인상	백엔드 및 프론트엔드 작업	팀원	인상
강현규	백엔드 및 프론트엔드 작업	팀원	강현규

팀원 [일정 / 역할]

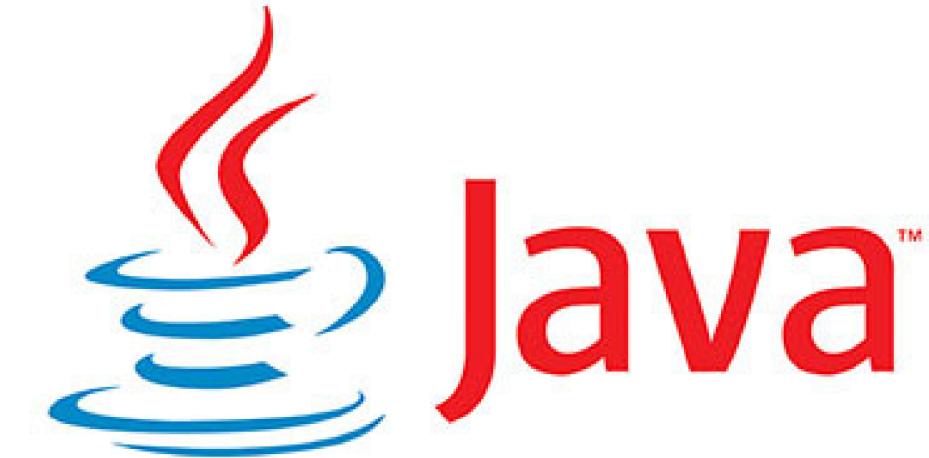
10/28(금) ~ 11월 2(수)	10/31(월) ~ 11월 2(수)	11월 3(목) ~ 11월 7(월)	11월 8(수) ~ 11월 10(목)	11월 11(금)	11월 14(월)
주제 선정/ 프로토타입 제 작	데이터베이스 설계 및 프로토타입 작 성	코드 작성 / 작성된 코드 보 완	제품 상세페이지 작성	코드 보완 / 발표 준비	발표 준비 / 10시 발표 및 시연

사용된 스킬 / 언어 / 툴 1

HTML



JS



CSS



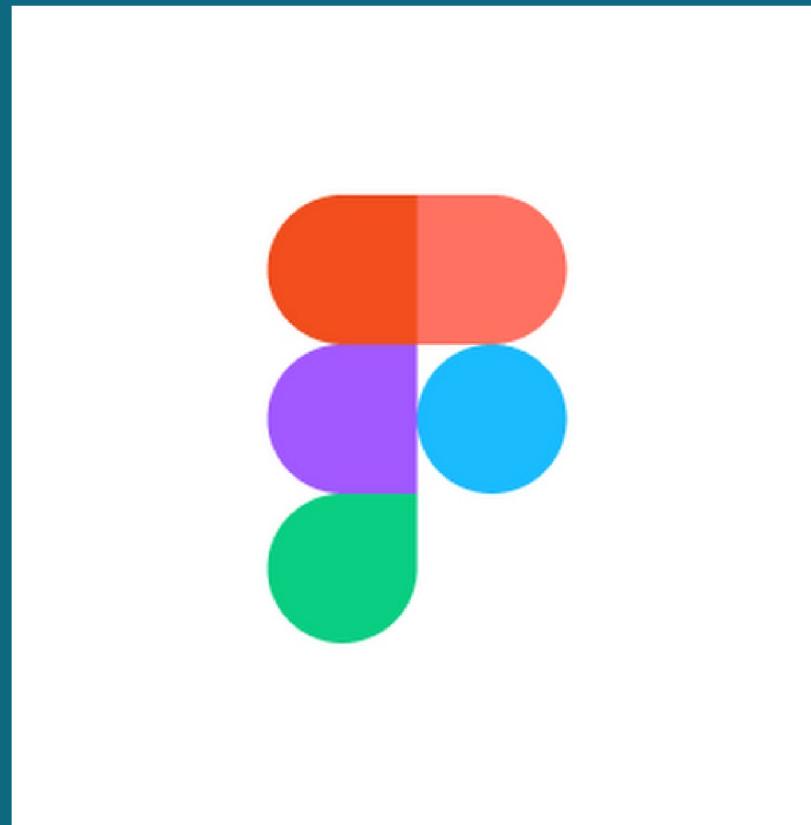
MySQL™



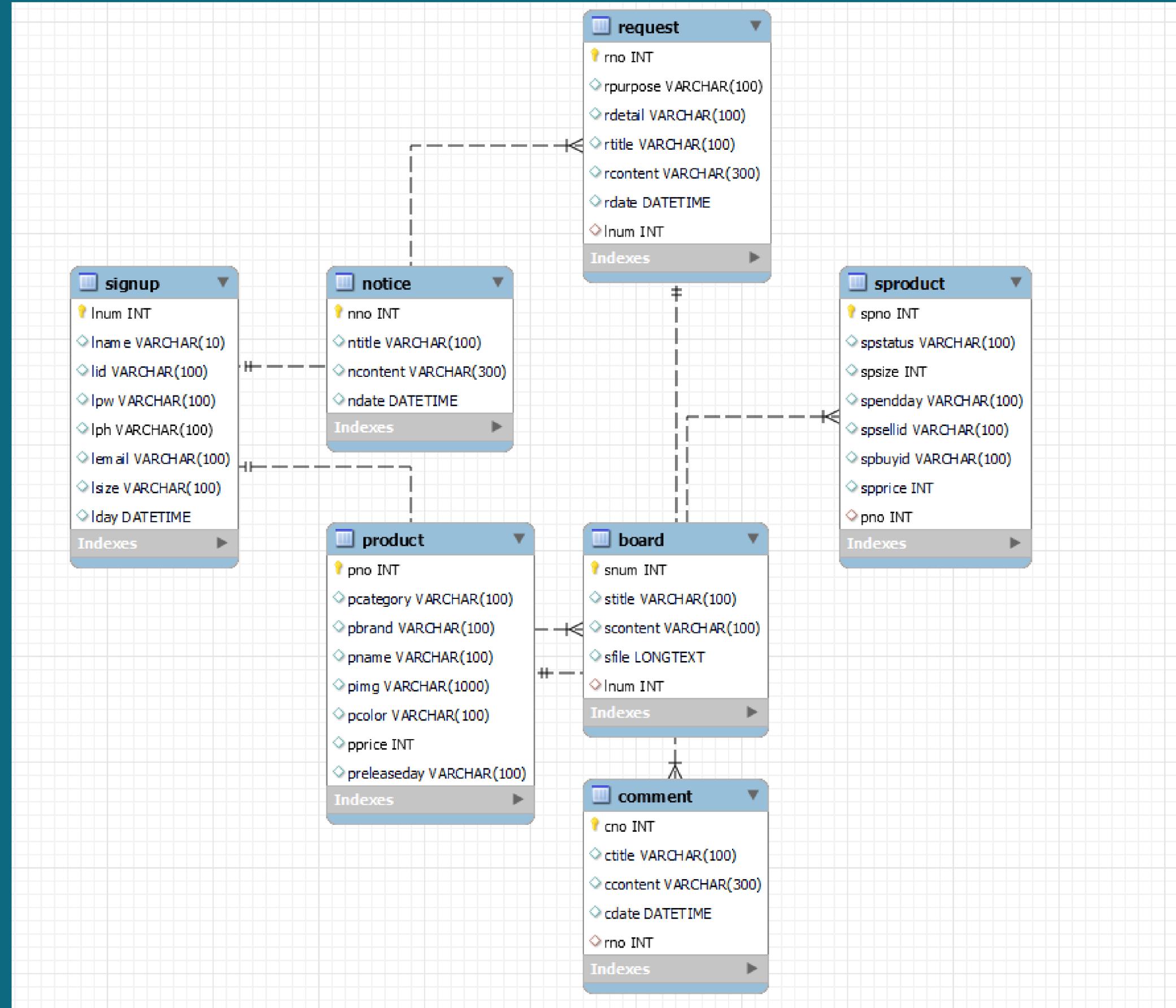
JSP



사용된 스킬 / 언어 / 툴 2



DB 설계도



request : 문의사항

notice : 공지사항

sproduct: 소분류 상품

product : 대분류 상품

board : 셀럽게시판

comment : 댓글

signup : 회원가입

주요 코드와 시연: 미디어 쿼리



```
/*-----가로 : 640px 이하 시작-----*/
@media (max-width: 640px) {}

/*-----가로 : 640px 이하 시작-----*/
/*-----가로 : 769px 이상 960px 이하 시작-----*/
@media (min-width: 641px) and (max-width: 768px) {}

/*-----가로 : 641px 이상 768px 이하 끝-----*/
/*-----가로 : 769px 이상 960px 이하 시작-----*/
@media (min-width: 769px) and (max-width: 967px) {}

/*-----가로 : 769px 이상 967px 이하 끝-----*/
/*-----가로 : 968px 이상 1097px 이하 시작-----*/
@media (min-width: 968px) and (max-width: 1098px) {}

/*-----가로 : 968px 이상 1097px 이하 끝-----*/
/*-----가로 : 1098px 이상 1281px 이하 시작-----*/
@media (min-width: 1098px) and (max-width: 1281px) {}

/*-----가로 : 1098px 이상 1281px 이하 끝-----*/
/*-----가로 : 1282px 이상 1536px 이하 시작-----*/
@media (min-width: 1282px) and (max-width: 1536px) {}

/*-----가로 : 1282px 이상 1536px 이하 끝-----*/
```

주요 코드와 시연 : CHART.JS



```
<!-- 그래프 css 설정은 여기서 하는겁니다. 시작 -->
<canvas id="myChart" style="width:100%;max-width:1000px"></canvas>
<!-- 그래프 css 설정은 여기서 하는겁니다. 끝 -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.9.4/Chart.js"></script>
<script type="text/javascript" src="../js/detail.js"></script>
```



```
// new Chart("dom으로 가져올 id", {객체인데 그래프에 설정 하고싶은 것들을 넣어주는겁니다.} )
new Chart("myChart", {
  type: "line",
  data: {
    labels: xValues, // 라벨을 xValues라는 변수를 쓸거다. //datasets은 아마 그래프 css설정 인거 같아요.
    datasets: [
      data: yValues, // data에는 yValues를 넣을겁니다.
      fill: false, // 그래프와 x축을 적분처럼 다 색을 칠할건지 ==> 근데 이건 안하는게 이쁠
      backgroundColor: "#3136ff", // 그래프 색은 빨간색이 좋을거 같아요.
      borderColor: "#3136ff", // 이게 꼭짓점 없애는 마다. 시작
      pointRadius: 3,
      lineTension : 0 // 이게 꼭짓점 없애는 마다. 끝
    ]
  }, // datasets를 이렇게 쓸거다.
  options: {...} // 옵션으로 이렇게 쓸거다.
});
```

주요 코드와 시연 : 햄버거 드롭다운



```
<section class="brand">
    <div class="brand_content">이달의 인기 브랜드</div>
    <div class="month_popular_brand">
        <div>1.Nike &nbsp;&nbsp;&nbsp;</div>
        <div>2.밀레 &nbsp;&nbsp;&nbsp;</div>
        <div>3.THE NORTH FACE &nbsp;&nbsp;&nbsp;</div>
    <!-- 여기서부터 햄버거 드롭다운 바 시작 -->
    <a href="#" class="menu_bar">
        <span></span>
        <span></span>
        <span></span>
    </a>
    <div class="menu_open">
        <ul>
            <li class="dept1">
                <a href="#">1~10순위</a>
                <ul>
                    <li class="dept2">
                        <a href="#">1.Nike &nbsp;&nbsp;&nbsp;</a>
                    </li>
                    <li class="dept2">
                        <a href="#">2.밀레 &nbsp;&nbsp;&nbsp;</a>
                    </li>
                    <li class="dept2">
                        <a href="#">3.THE NORTH FACE &nbsp;&nbsp;&nbsp;</a>
                    </li>
                </ul>
            </li>
        </ul>
    </div>
    <!-- 여기서부터 햄버거 드롭다운 바 끝 -->
</section>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
$(document).ready(function() {
    var gnb = $('#gnb');// 마우스 over 시
    gnb.mouseenter(function() {
        $('.inner_menu').show();// menu bg
        var menuHeight = $('#header').outerHeight();
        var inmeHegiht = $('.inner_menu').outerHeight();
        $('.hd_bg').css({
            'top': menuHeight + 'px',
            height: inmeHegiht + 'px'
        });
    });// 마우스 leave 시
    gnbmouseleave(function() {
        $('.inner_menu').hide();
        $('.hd_bg').css('height', '0')
    });
    //dept2 hover시 dept1 active
    $('.dept1').mouseenter(function() {
        $(this).children().addClass('active');
        $(this).siblings().children().removeClass('active')
    });
    $('.dept1').mouseleave(function() {
        $(this).children().removeClass('active');
    });
    /* 로그인을 안했을때 햄버거 메뉴 */
    $('.menu_bar').click(function(){
        $(this).toggleClass('active');
        $('.menu_open').slideToggle();
    });
});

// 드롭다운 닫기
$(window).scroll(function () {
    var scrollValue = $(document).scrollTop();
    console.log(scrollValue);
    if(scrollValue>10 && findDropdown() == 'noMem'){
        noMemberDropdowncontent = document.querySelector('.menu_open')
        noMemberDropdowncontent.style.display = "none";
    }
});
```

주요 코드와 시연: 회원 아이디를 식별해 구매, 판매 내역 출력

```
String lid = (String)request.getSession().getAttribute("lid");
int type = Integer.parseInt(request.getParameter("type"));
ArrayList<orderinfodetail> result = null;
if(type == 1) { // 만약 type == 1 일 경우 구매내역 출력 DAO 실행
    result = memberdao.getInstance().brload(lid);
} else if (type == 2) { // 만약 type == 2 일 경우 판매내역 DAO 실행
    result = memberdao.getInstance().srload(lid);
}
JSONArray array = new JSONArray();
for(int i = 0 ; i<result.size(); i++) {
    JSONObject object = new JSONObject();
    object.put("spno" , result.get(i).getSpno());
    object.put("spstatus" , result.get(i).getSpstatus());
    object.put("spsize" , result.get(i).getSpsize());
    object.put("spendday" , result.get(i).getSpendday());
    object.put("spsellid" , result.get(i).getSpsellid());
    object.put("spbuyid" , result.get(i).getSpbuyid());
    object.put("spprice" , result.get(i).getSpprice());
    object.put("pno", result.get(i).getPno());
    object.put("pcategory" , result.get(i).getPcategory());
    object.put("pbrand" , result.get(i).getPbrand());
    object.put("pname" , result.get(i).getPname());
    object.put("pimg" , result.get(i).getPimg());
    object.put("pcolor" , result.get(i).getPcolor());
    object.put("pprice" , result.get(i).getPprice());
    object.put("preleaseday" , result.get(i).getPreleaseday());
    array.add(object);
}

response.setCharacterEncoding( "UTF-8" );
response.getWriter().print(array);
}
```

```
// 구매내역 출력 [안테선 ]
public ArrayList<orderinfodetail> brload(String spbuyid){
    ArrayList<orderinfodetail> list = new ArrayList<>();
    String sql = "SELECT spno, spstatus, spsize , spendday, spsellid, spbuyid , spprice , product.*
    FROM sproduct, product where spbuyid = ?";
    try {
        ps = con.prepareStatement(sql);
        ps.setString(1, spbuyid);
        rs = ps.executeQuery();
        while(rs.next()) {
            orderinfodetail dto = new orderinfodetail(
                rs.getInt(1), rs.getString(2),
                rs.getInt(3), rs.getString(4),
                rs.getString(5), rs.getString(6),
                rs.getInt(7) , rs.getInt(8),
                rs.getString(9) , rs.getString(10),
                rs.getString(11) , rs.getString(12),
                rs.getString(13) , rs.getInt(14) , rs.getString(15)
            );
            list.add(dto);
            System.out.println(dto);
        }
        return list;
    }catch (Exception e) {System.out.println(e);}
    return list;
}

// 판매내역 출력 [안테선 ]
public ArrayList<orderinfodetail> srload(String spsellid){
    ArrayList<orderinfodetail> list = new ArrayList<>();
    String sql = "SELECT spno, spstatus, spsize , spendday, spsellid, spbuyid , spprice , product.*
    FROM sproduct, product where spsellid = ?";
    try {
        ps = con.prepareStatement(sql);
        ps.setString(1, spsellid);
        rs = ps.executeQuery();
        while(rs.next()) {
            orderinfodetail dto = new orderinfodetail(
                rs.getInt(1), rs.getString(2),
                rs.getInt(3), rs.getString(4),
                rs.getString(5), rs.getString(6),
                rs.getInt(7) , rs.getInt(8),
                rs.getString(9) , rs.getString(10),
                rs.getString(11) , rs.getString(12),
                rs.getString(13) , rs.getInt(14) , rs.getString(15)
            );
            list.add(dto);
            System.out.println(dto);
        }
        return list;
    }catch (Exception e) {System.out.println(e);}
    return list;
}
```

```
getbuylist()
function getbuylist(){
    $.ajax({
        url: "/shoesproject/product/psession",
        type: "post",
        data: { "type": 1 }, // type == 1
        async: false,
        success: function(re) {
            if (re) {
                let bns = JSON.parse(re)
                let html = "";
                let bns;
                for (let i = 0; i < bns.length; i++) {
                    bns = bns[i]
                    html +=
                        "<div class='str'>" + bns.spno + "</div> <br>"
                        + "<div class='str'>" + bns.spbuyid + "</div> <br>"
                        + `<div class='str'></div>` 
                        + "<div class='str'>" + bns.spsize + "</div> <br>" 
                        + "<div class='str'>" + bns.pname + "</div> <br>" 
                        + "<div class='str'>" + bns.preleaseday + "</div> <br>" 
                        + "<div class='str'>" + bns.spstatus + "</div> <br>" 
                }
                document.querySelector('.buy').innerHTML = html
            }
        }
    })
}

getsellist()
function getsellist(){
    $.ajax({
        url: "/shoesproject/product/psession",
        type: "post",
        data: { "type": 2 }, // type == 2
        async: false,
        success: function(re) {
            if (re) {
                let bns = JSON.parse(re)
                let html = '';
                let pimg = null;
                let bns = null;
                for (let i = 0; i < bns.length; i++) {
                    bns = bns[i]
                    pimg = `` 
                    html +=
                        "<div class='str'>" + bns.spno + "</div> <br>" 
                        + "<div class='str'>" + bns.spsellid + "</div> <br>" 
                        + pimg 
                        + "<div class='str'>" + bns.spsize + "</div> <br>" 
                        + "<div class='str'>" + bns.pname + "</div> <br>" 
                        + "<div class='str'>" + bns.preleaseday + "</div> <br>" 
                        + "<div class='str'>" + bns.spstatus + "</div> <br>" 
                }
                document.querySelector('.sell').innerHTML = html
            }
        }
    })
}
```

주요 코드와 시연



```
let cloadstate = false /* 답변한 내용 품 열고닫기 상태 저장 변수 */
// 답변내용 보기 품 생성
function commentview(rno) {
    $.ajax({
        url: "/shoesproject/admin/cload",
        data: { "rno": rno },
        success: function(re) {
            if (re) {
                let list = JSON.parse(re)
                for (let i = 0; i < list.length; i++) {
                    let c = list[i]
                    if (cloadstate == false) { // 답변하기 품이 열려있으면 답변하기 상세페이지 오픈
                        html = `<tr>
                            <td colspan="7" style="height: 500px; border: 1px solid black;">
                                <h4> 답변 한 제목 </h4> <br> <h5> ${c.ctitle} </h5> <br>
                                <h4> 답변 한 내용 </h4> <br> <h5> ${c.ccontent} </h5> <br>
                            </td>
                        </tr>`
                        cloadstate = true; // 답변하기 열리면 true 전환
                    } else { // 답변하기 품이 열려있을때 다시 버튼을 눌러주면
                        html = ""; // 공백처리
                        cloadstate = false; // 닫히면 false 전환
                    }
                    document.querySelector(`.commentform${rno}`).innerHTML = html
                }
            }
        }
    })
}
```



```
rloadDetailId()
function rloadDetailId() { // 내가 문의한 게시물만 출력
    $.ajax({
        url: "/shoesproject/rload",
        data: { "type": "memberd" },
        success: function(re) {
            if (re) {
                let list = JSON.parse(re)
                console.log(list)
                document.querySelector('.request').innerHTML =
                    "<tr><th>번호</th><th>제목</th><th>내용</th><th>날짜</th></tr>"
                for (let i = 0; i < list.length; i++) {
                    r = list[i]
                    rnum = list[i].rno
                    html = `<tr>
                        <td>${r.rno}</td>
                        <td>${r.rttitle}</td>
                        <td>${r.rcontent}</td>
                        <td class="board_date">${r.rdate}</td>
                        <td class="board_thing"><button type="button" class="board_btn"
                            onclick="commentview(${r.rno})">답변보기</td>
                        <td class="board_thing"><button type="button" class="board_btn"
                            onclick="rdelete(${r.rno})">삭제</button></td>
                    </tr>
                    <tr class="commentform${r.rno}"></tr>
                `
                document.querySelector('.request').innerHTML += html
            }
        }
    })
}
```

주요 코드와 시연

```
// 일반회원 공지출력
notice()
function notice() {
    $.ajax({
        url: "/shoesproject/nload",
        data: {"type" : "member" },
        success: function (re) {
            if(re){
                let list = JSON.parse(re)
                console.log(list)
                let html = "<tr><th>번호</th><th>제목</th><th>내용</th><th>날짜</th></tr>";
                for(let i = 0 ; i<list.length ; i++) {
                    notice = list[i]
                    html += "<tr>
                        + "<td>" + notice.nno + "</td>"
                        + "<td>" + notice.ntitle + "</td>"
                        + "<td>" + notice.ncontent + "</td>"
                        + "<td>" + notice.ndate + "</td>
                    + "</tr>"
                }
                document.querySelector('#notice_table').innerHTML = html
            }
        }
    })
}
```

일반회원과 어드민 TYPE 을 나누어 작업했습니다.

일반회원이 문의 -> 관리자 페이지에서 답변

일반회원의 NUMBER 혹은 ID를 식별해 본인이 문의한 답변만 확인 가능합니다.

DAO 쿼리문을 이용해 본인의 구매 , 판매내역을 확인 할 수 있으며 상태도 확인 가능합니다.

주요 코드와 시연: 관리자페이지 카테고리별 상품 등록현황



```
public class SpregistDao extends dao {  
  
    private static SpregistDao dao = new SpregistDao();  
    public static SpregistDao getInstance() { return dao; }  
  
    public ArrayList<SpregistDto> supload(int pno) {  
        ArrayList<SpregistDto> list = new ArrayList<>();  
        String sql = "select * from sproduct where pno = " + pno;  
  
        try {  
            ps = con.prepareStatement(sql);  
            rs = ps.executeQuery();  
            while(rs.next()) {  
                SpregistDto dto = new SpregistDto(  
                    rs.getInt(1), rs.getString(2),  
                    rs.getInt(3), rs.getString(4),  
                    rs.getString(5), rs.getString(6),  
                    rs.getInt(7), rs.getInt(8)  
                );  
                list.add(dto);  
            }  
            return list;  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
        return list;  
    }  
}
```



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,  
IOException {  
    int pno = Integer.parseInt(request.getParameter("pno"));  
  
    ArrayList<SpregistDto> list = SpregistDao.getInstance().supload(pno);  
  
    JSONArray array = new JSONArray();  
    for(int i = 0 ; i<list.size() ; i++) {  
        JSONObject object = new JSONObject();  
        object.put("spno", list.get(i).getSpno());  
        object.put("spstatus", list.get(i).getSpstatus());  
        object.put("spsize", list.get(i).getSpsize());  
        object.put("spendday", list.get(i).getSpendday());  
        object.put("spsellid", list.get(i).getSpsellid());  
        object.put("spbuid", list.get(i).getSpbuyid());  
        object.put("spprice", list.get(i).getSpprice());  
        object.put("pno", list.get(i).getPno());  
        array.add(object);  
    }  
    response.setCharacterEncoding("UTF-8");  
    response.getWriter().print(array);  
}
```

주요 코드와 시연: 메인페이지에 상품리스트 등록 및 출력



```
public boolean pregist(String pcategory, String pbrand, String  
pname, String pimg, String pcolor, int pprice, String preleaseday) {  
  
    String sql = "insert into product(pcategory, pbrand, pname,  
pimg, pcolor, pprice, preleaseday)"  
        + "values(?, ?, ?, ?, ?, ?, ?)";  
  
    try {  
        ps = con.prepareStatement(sql);  
        ps.setString(1, pcategory);  
        ps.setString(2, pbrand);  
        ps.setString(3, pname);  
        ps.setString(4, pimg);  
        ps.setString(5, pcolor);  
        ps.setInt(6, pprice);  
        ps.setString(7, preleaseday);  
        ps.executeUpdate();  
  
        return true;  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
    return false;  
}
```



```
public ArrayList<PregistDto> ploadMain() {  
    ArrayList<PregistDto> list = new ArrayList<>();  
    String sql = "select * from product order by pno desc limit 8";  
    try {  
        ps = con.prepareStatement(sql);  
        rs = ps.executeQuery();  
        while(rs.next()) {  
  
            PregistDto dto = new PregistDto(  
                rs.getInt(1), rs.getString(2),  
                rs.getString(3), rs.getString(4),  
                rs.getString(5), rs.getString(6),  
                rs.getInt(7), rs.getString(8)  
            );  
            list.add(dto);  
        }  
        return list;  
    } catch (Exception e) {  
        System.out.println(e);  
    }  
    return list;  
}
```

주요 코드와 시연: 메인페이지 상품별 페이지 로딩



```
public class DetailDao extends dao {  
    private static DetailDao dao = new DetailDao();  
    public static DetailDao getInstance() { return dao; }  
  
    //소분류 카테고리  
    public PregistDto detail(int pno) {  
        String sql = "select * from product where pno = "+pno;  
        try {  
            ps = con.prepareStatement(sql);  
            rs = ps.executeQuery();  
  
            if(rs.next()) {  
                PregistDto dto = new PregistDto()  
  
                    rs.getInt(1), rs.getString(2),  
                    rs.getString(3), rs.getString(4),  
                    rs.getString(5), rs.getString(6),  
                    rs.getInt(7), rs.getString(8)  
                );  
                return dto;  
            }  
        } catch (Exception e) {  
            System.out.println(e);  
        }  
        return null;  
    }  
}
```



```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    Object pvalue = request.getSession().getAttribute("pno");  
    int pno = 0;  
    if(pvalue == null) { // int는 null값을 체크 못하니까 세션을 객체로  
        // 변환해서 null을 체크한다.  
        response.sendRedirect("mainhome.jsp");  
    } else {  
        pno = (Integer)pvalue;  
    }  
    PregistDto dto = DetailDao.getInstance().detail(pno);  
    JSONObject object = new JSONObject();  
    object.put("pno", dto.getPno());  
    object.put("pcategory", dto.getPcategory());  
    object.put("pbrand", dto.getPbrand());  
    object.put("pname", dto.getPname());  
    object.put("pimg", dto.getImg());  
    object.put("pcolor", dto.getPcolor());  
    object.put("pprice", dto.getPrice());  
    object.put("preleaseday", dto.getPreleaseday());  
    response.setCharacterEncoding("UTF-8");  
    response.getWriter().print(object);  
}
```

주요 코드와 시연: 셀럽 게시판

기존의 있던 셀럽 게시판에
클릭시 글 내용을 볼수 있습니다

삭제와 수정은 본인의 글일때만 출력이 되고 가능합니다

```
function selhi() {
    $.ajax({
        url: "/shoesproject/board/write",
        type: 'get',
        success: function(re) {
            let sjson = JSON.parse(re)

            let chtml = ''
            for (let i = 0; i < sjson.length; i++) {
                let simg = '/shoesproject/upload/' + sjson[i].sfile
                chtml +=
                    '<div class="seloveComponent">
                     + '<a href="/shoesproject/board/selovelist.jsp"></a>' +
                     + '<div class="userid">' + sjson[i].lid + '님</div>' +
                     + '<div class="usercontent">' + sjson[i].stitle + '</div>' +
                     + '</div>';
            }
            document.querySelector(".seloveimg").innerHTML += chtml;
        }
    })
}
```

앞으로 추후 방향

JSP 와 JDBC 로 작성한 코드를 리팩토링하면서 SPRING 과 MYBATIS 으로 구현하고자 합니다.

프론트는 리액트와 리덕스를 활용해 구현해보고 싶습니다.

또한 프로젝트를 AWS 환경에서 구현해보고 싶습니다.

프로젝트에서 온라인 결제 기능을 구현을 못했는데 끝나고 구현해보고 싶습니다.

각자의 느낀 점

안태섭: 입력 및 출력 코드를 입력하는 것에 대해 반복작업이 단순하면서도 복잡한데 많은 어려움을 극복해서 팀원들과 소통을 통해 문제해결 능력을 기를 수 있었고 첫 번째, 두 번째 팀플에서와 마찬가지로 팀원들과 소통이 정말 중요하다고 생각했습니다. 그리고 팀플과 별개로 학습하는데에 있어 정말 많은 도움이 되었습니다.

김경주: 프론트와 백엔드의 역할을 번갈아가면서 다 했는데 팀원들의 의사소통이 제일 중요한거 같습니다

인상: 기획부터 프론트엔드에서 백엔드 작업에 이르기까지 협업하는데 생각보다 생각할게 많은 것 같고 소통이 협업에서 정말 중요하다고 느꼈습니다. 전반적인 CRUD 작업을 이해하는데 많은 도움이 되었고, 앞으로 새로운 환경에서 협업할 때는 더 잘 할 수 있다는 자신이 생긴 것 같습니다. 협업하는데 있어서 팀원 한 명 한명이 정말 소중하다고 생각했습니다.

강현규 : 코드를 가져와서 써야 하는 것들도 생각보다 쉽지는 않은 거 같습니다.
그리고 POSITION 속성을 쓸 때 나중에 반응형으로 바꾸면 손이 많이 갈 수 있다는 것도 깨달았습니다.

shoesProject팀의 발표를 마치겠습니다.

감사합니다!

Casual Shoes



@reallygreatsite