PORTFOLIO JSP

남 재 민

CONTENT

I. 개발 배경

01 개발 배경

02 개발 환경

Ⅱ. 프로젝트 구성

03 구성

04 주요 기능

05 데이터베이스

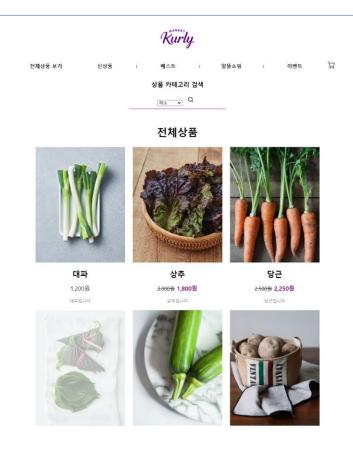
III. 라이브러리

06 라이브러리 및 코드

- (1) manager
- (2) customer

1. 개발 배경

0 1 개발 배경



온라인 웹쇼핑몰 마켓컬리의 사이트 클론 프로젝트 입니다. Java, JSP, CSS, Javascript, 그리고 DBMS는 Mysql을 사용하였으며, MVC2 모델로 Controller 및 각종 API를 구현하는 것에 초점을 맞추어 진행한 프로젝트 입니다. 관리자 / 사용자 기능을 각각 구현하여, 서로 다른 권한의 계정으로 로그인 했을 경우 기능 및 페이지를 구분하였으며 로그인, 상품 관리, 게시판 기능을 중점으로 구현한 웹사이트 프로젝트 입니다.

회원가입 | 로그인 | 고객센터

0 1 개발 배경

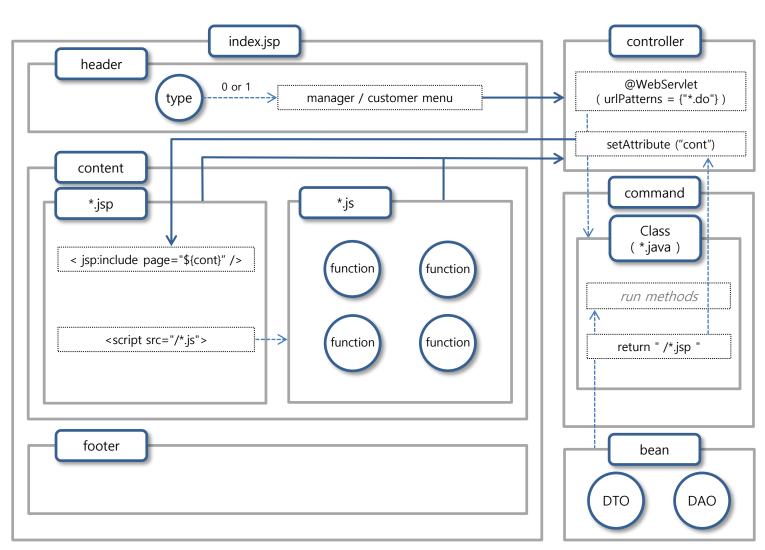
- 작업 기간: 2021년 1월 ~ 2월
- 담당한 영역: 모두
- 사용한 기술 stack
- Java, JSP, CSS, Javascript
- DBMS: Mysql
- MVC2, Ajax, Servlet
- Apache Tomcat
- 중점을 둔 부분 : MVC2 모델의 구축

0 2 개발 환경

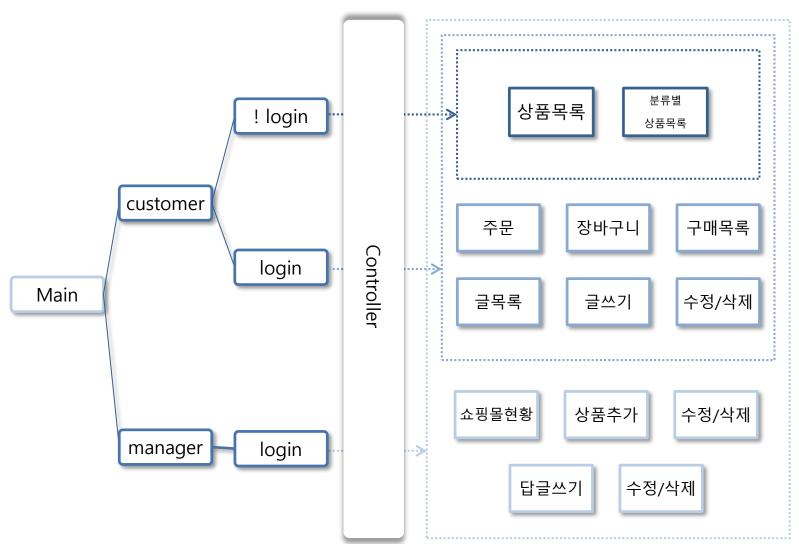


11. 프로젝트 구성

03 구성



04 주요 기능



04 데이터베이스

[관리자]

테이블명	manager				
No	Col_ID	Туре			
1	id	VARCHAR			
2	pw	VARCHAR			
3	name	VARCHAR			

[상품정보]

테이블명	item				
No	Col_ID	Туре			
1	item_number	INT			
2	item_cagtegory	VARCHAR			
3	item_name	VARCHAR			
4	item_price	INT			
5	item_stock	INT			
6	item_image	VARCHAR			
7	item_info	VARCHAR			
8	discount_rate	VARCHAR			
9	reg_date	VARCHAR			
10	sold	int			

04 데이터베이스

[구매목록]

테이블명	buy				
No	Col_ID	Туре			
1	customer_id	VARCHAR			
2	customer_name	VARCHAR			
3	cart_number	INT			
4	item_name	VARCHAR			
5	buy_price	INT			
6	buy_count	INT			
7	item_image	VARCHAR			
8	buy_date	VARCHAR			
9	howpay	VARCHAR			
10	address	VARCHAR			

[게시판]

테이블명	board				
No	Col_ID	Туре			
1	num	INT			
2	writer	VARCHAR			
3	title	VARCHAR			
4	pw	VARCHAR			
5	reg_date	DATE			
6	ref	INT			
7	re_step	INT			
8	re_level	INT			
9	readcount	INT			
10	content	VARCHAR			

04 데이터베이스

[회원]

테이블명	customer				
No	Col_ID	Туре			
1	id	VARCHAR			
2	pw	VARCHAR			
3	name	VARCHAR			
4	reg_date	VARCHAR			
5	tel	VARCHAR			
6	address	VARCHAR			
7	email	VARCHAR			

[장바구니]

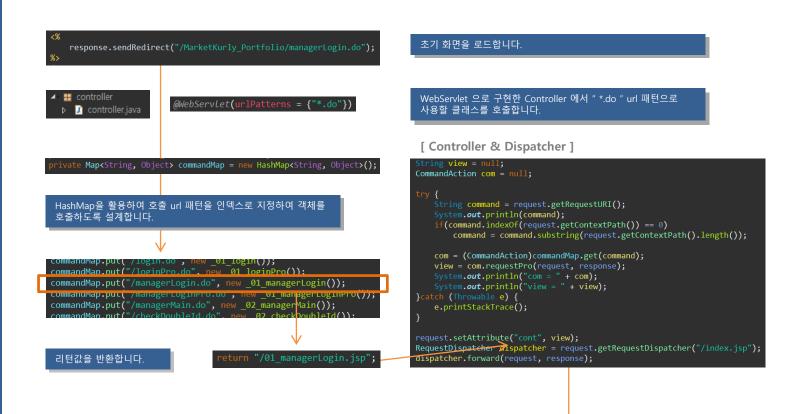
테이블명	cart				
No	Col_ID	Туре			
1	cart_number	INT			
2	buyer	VARCHAR			
3	item_name	VARCHAR			
4	buy_price	INT			
5	buy_count	INT			
6	item_image	VARCHAR			

111. 라이브러리

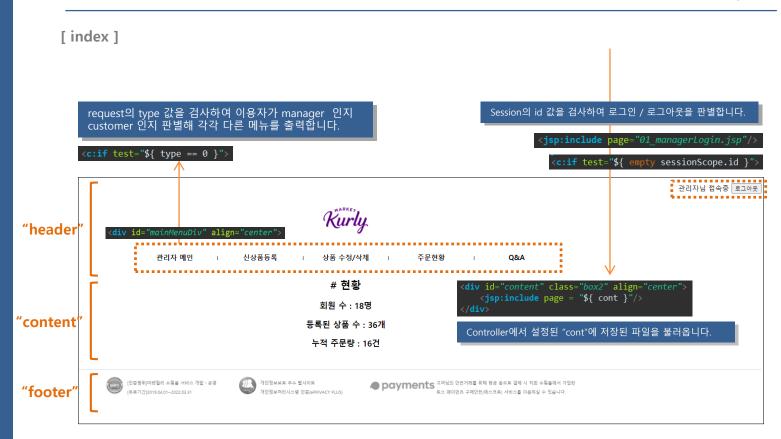
(1) manager

[manager]

[웹사이트 구조 및 매니저 로그인 화면 로드]



[manager]



모든 페이지는 index를 바탕으로 사용자의 요청에 따라 호출된 클래스의 서로 다른 return 값을 cont 변수에 저장하여 content div에 출력합니다. 또한 header의 우측 상단에는 manager / customer 또는 login / !login 상태에 따라 다른 메뉴를 출력합니다.

[manager]

[신상품 등록 (1)]

<pre>request.setAttribute(return "/04_addNewIte()</pre>	("type", new Inte		품등록	
관리자 메인	! ✓ ○ 신상품등록	Ku.		Q&A
	상품 카테고리	채소 ✔		
	상품명			
	상품가격		원	
	상품재고			
	이미지	파일 선택 선택된 파일	없음	
	상품정보			
	할인율		96	
		상품 추	가하기	
[인당범위]마켓컬리 쇼핑을 서비: [유료기간]2019 04 01~2022 03 31		보호 우수 웹사이트 처리시스템 인종(ePRIVACY PLUS)	■ payments 고객님의 안전거래를 위해 현금 표스 페이먼츠 구매안전(에스크	등으로 걸제 시 저희 쇼핑몰에서 가입한 로, 서비스를 이용하실 수 있습니다.
<form <mark="" method="</td><td>" post"="">id="upFo</form>	rm1" action=	- ="/MarketKurly_Portfolio/d	nddNewItemPro.do"	
			Form – input 태그를 활용하여 등록힐 정보를 처리할 객체를 호출합니다.	· 상품 정보를 저장하여

[manager]

[신상품 등록 (2)]

```
tring realFolder = "";
String fileName="";
MultipartRequest imageup = null;
String saveFolder = "/img";
String encType = "UTF-8";
int max size = 2 * 1024 * 1024;
ServletContext context = request.getSession().getServletContext();
realFolder = context.getRealPath(saveFolder);
   imageup = new MultipartRequest(request, realFolder, max size, encType, new DefaultFileRenamePolicy());
   Enumeration<?> files = imageup.getFileNames();
   while(files.hasMoreElements()) {
                                                                                                  MultipartRequest를 활용하여 상품
       fileName = imageup.getFilesystemName(name);
                                                                                                  정보 및 이미지 파일을 담은 후
                                                                                                  itemDTO 클래스에 저장하여
}catch(Exception e) {
                                                                                                  managerDAO 클래스를 통해 데이
   e.printStackTrace();
                                                                                                  터베이스에 저장합니다.
String item name = imageup.getParameter("name");
                                                                    itemDTO dto = new itemDTO();
String item category = imageup.getParameter("category");
                                                                   dto.setItem category(item category);
String item price = imageup.getParameter("price");
                                                                   dto.setItem name(item name);
String item stock = imageup.getParameter("stock");
                                                                   dto.setItem price(Integer.parseInt(item price));
String item info = imageup.getParameter("info");
                                                                   dto.setItem stock(Integer.parseInt(item stock));
 String discount rate = imageup.getParameter("discountRate");
                                                                   dto.setItem info(item info);
                                                                   dto.setDiscount rate(Integer.parseInt(discount rate));
                                                                    f(fileName != null) {
                                                                       dto.setItem image(fileName);
                                                                       dto.setItem image("error.jpg");
 nanagerDAO dao = managerDAO.getInstance();
dao.insertNewItem(dto);
```

[manager]

[상품 목록 및 관리]



[manager]

「상품 수정 및 삭제]

- 상품 수정

```
int item number = Integer.parseInt(request.getParameter("item number"));
System.out.println("updateItme - item number : " + item number);
request.setAttribute("beanForUpdate", bean);
선택한 상품의 item_number로 상품 정보를 불러옵니다.
    번호
                           가격
                           <input type="text" name="price" value="${ update.getItem price() }">
   카테고리
            채소 🗸
    상품명
            당근
                                                       상품 정보를 input 태그 안에 출력하여 수정할 수 있도록 합니다.
            2500
    가격
    재고
   이미지
            파일 선택 선택된 파일 없음
                                     <form method="post" action="/MarketKurly Portfolio/updateItemPro.do"</pre>
            당근입니다
   상품 정보
                                                       managerDAO dao = managerDAO.getInstance();
```

- 상품 삭제

할인율

등록일

판매량

2021-02-18 16:38:01

다시작성

수정하기

```
int item number = Integer.parseInt(request.getParameter("item number"));
System.out.println("delItem - item number : " + item number);
anagerDAO.getInstance().deleteItem(item number);
```

삭제 또한 마찬가지로 삭제할 상품의 item_numbe로 managerDAO의 deleteItem 메소드를 호출하여 데이터베 이스에서 상품 정보를 삭제합니다.

수정할 상품 정보를 updateItemPro 클래스에 전달하면

managerDAO 객체에 저장된 updateItem 메소드를 실행

dao.updateItem(item number, dto);

하여 상품 정보를 수정합니다

[manager]

[주문 현황]



managerDAO의 getAllOrderList 메소드를 실행하여 데이터베 이스에서 불러온 orderList를 request에 저장하여 출력합니다.

c:f	ForEach	var="	order"	items=	"\${ ord	derL:	ist }"> 주문한	변황	Q&A
번호	아이디	주문자명	상품명	가격	구매갯수	이미지	구매일자	결제수단	주소
1	test	김혜민	연어	14000원	1개		2021-02-28 15:28:37	계좌이체	서울시 양천구
2	test	김혜민	당근	2250원	1개	Whi	2021-02-28 15:28:37	계좌이체	서울시 양천구
3	test	김혜민	치즈돈가스	13000원	17#		2021-02-28 15:28:37	계좌이체	서울시 양천구
4	test	김혜민	연어	14000원	17#		2021-02-28 15:28:37	계좌이체	서울시 양천구
5	test	김혜민	당근	2250원	1개	Mili	2021-02-28 15:44:28	계좌이체	서울시 양천구
6	test1	김효성	당근	2250원	1개	Whi	2021-02-28 19:03:39	신용카드	서울시 양천구 목동
7	test1	김효성	대파	1200원	17#	W	2021-02-28 19:03:39	신용카드	서울시 양천구 목동
8	test1	김효성	파프리카	2700원	17#	66	2021-02-28 19:03:39	신용카드	서울시 양천구 목동
9	test1	김효성	닭다리	10000원	17#		2021-02-28 19:03:39	신용카드	서울시 양천구 목동
10	test1	김효성	닭가슴살	5000원	2개	20	2021-02-28 20:46:45	신용카드	서울시 양천구 목동
11	test1	김효성	닭가슴살	5000원	37#	27	2021-02-28 20:48:44	계좌이체	서울시 양천구 목동
12	test1	기능선	당가스산	5000원	278	2,	2021-02-28 20-49-38	계자이체	서울시 양체구 모토

[manager]

[Q&A]

```
pageSize = 10;
현재 페이지 번호, 현재 페이지에서 불러올 글 목록, 첫번째,
                                                           ring pageNum = request.getParameter("pageNum");
마지막 글 번호 등 출력할 정보를 request로 전달합니다.
                                                           f(pageNum == null) {
                                                             pageNum = "1";
                                                          nt number = 0;
                                                          .nt curPage = Integer.parseInt(pageNum);
                                                         count = boardDAO.getInstance().getAllCount();
                                                          nt startRow = (curPage - 1) * pageSize;
                                                          nt endRow = curPage * pageSize;
                                                          \rrayList<boardDTO> boardList = boardDAO.getInstance().getAllBoardList(startRow, endRow);
                                                         number = count - (curPage - 1) * pageSize;
 관리자 메인
                    신상품등록
                                     상품 수정/삭제
                                                          nt pageCount = count / pageSize + (count % pageSize == 0 ? 0 : 1);
                                                           nt startPage = 1;
                                      고객센터
                                                          f(curPage % pageSize != 0) {
                                                             startPage = (int)(curPage / pageSize) * pageSize + 1;
                                                             startPage = ((int)(curPage / pageSize) - 1) * pageSize + 1;
번호
                      제목
                                                  작성자
289 하이하이
                                                  test1
     [답변] 하이하이
                                                  관리자
                                                           nt endPage = startPage + pageSize - 1;
                                                           F(endPage > pageCount)
287
     [답변] 999
                                                  관리자
                                                             endPage = pageCount;
286 999
                                      <a href="/MarketKurly_Portfolio/adminShowBoardContent.do?num=${ board.getNum()</pre>
285
    999
284
    999
                                                   test
                                                                   2021-02-28
283
    999
                                                                   2021-02-28
                                                                                   0
                                                   test
282
    999
                                                                   2021-02-28
                                                                                   2
281
    999
                                                                   2021-02-28
                                                                                   2
                                                   test
    [답변] 999
                                                  관리자
                                                                   2021-02-28
                                                                                     게시판 페이지의 조건에 따라 [이전] 또는 [다음] 버튼이 표시
                            [1] [2] [3] [4] [5] [6] [7] [8] [9] [10 [다음]
                                                                                     되도록 설정하였습니다
                                       <c:if test="${ endPage < pageCount }">
                                           <a href="/MarketKurly_Portfolio/adminShowBoardList.do?pageNum=${ startPage + 10 }">[다음]</a>
```

[manager]

[글 내용 확인]

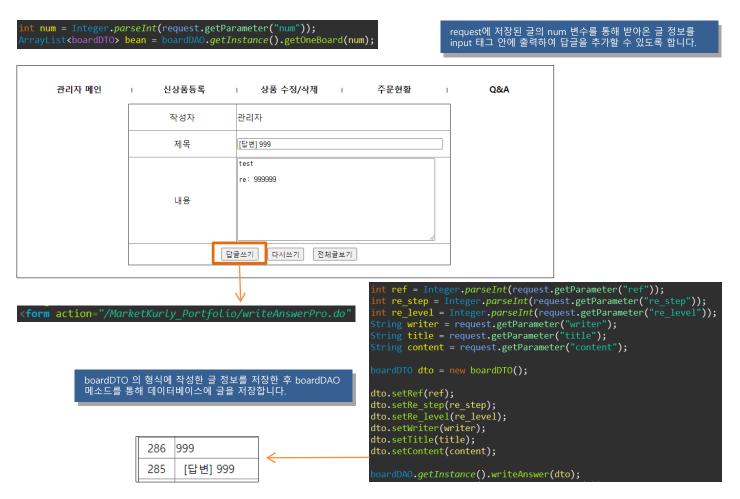


request에 저장된 글의 num 변수를 통해 데이터베이스에서 글 정보를 불러옵니다.



[manager]

[답글 쓰기]



(2) customer

[customer]

[웹사이트 구조 및 고객 메인 화면 로드]



[customer]



모든 페이지는 index를 바탕으로 사용자의 요청에 따라 호출된 클래스의 서로 다른 return 값을 cont 변수에 저장하여 content div에 출력합니다. 또한 header의 우측 상단에는 manager / customer 또는 login / !login 상태에 따라 다른 메뉴를 출력합니다.

[customer]

[회원 가입 (1)]



[customer]

[회원 가입 (2) - 중복 체크]

```
$("#checkDoubleEmail").click(function(){
var stId = false;
var stEmail = false;
                                                                                id 또는 email 에 저장된 값을 가져와 customerDAO의 메소드
                                                                                를 활용하여 결과값을 check 값에 저장합니다
$("#checkDoubleId").click(function(){
    if($("#id").val()){
        $.ajax({
                                                                               String id = (String)request.getParameter("id");
           type: "POST",
                                                                                int check = customerDAO.getInstance().checkDoubleId(id);
           url : "/MarketKurly Portfolio/checkDoubleId.do",-
           contentType: "application/x-www-form-urlencoded; charset=UTF-8",
                                                                               request.setAttribute("id", id);
                                                                               request.setAttribute("check", new Integer(check));
           data : {
                'id' : $("#id").val()
           success : function(data){
                                                                                결과 페이지에서 매개변수로 저장된 check 변수에 저장된 값
                                                                                을 검사하여 중복 유무를 검사 후 결과를 반환합니다.
               var loc = data.indexOf(str);
               var len = str.length;
               var check = data.substr(loc+len,1);
                                                                                                           <p id="ck">${check}
               if(check == "1"){
                   alert("중복된 아이디입니다.");
                   $("#id").val("");
                                                                                        사용할 수 있는 아이디 입니다.
                   alert("사용할 수 있는 아이디 입니다.");
                   id=$("#id").val();
                   stId = true;
                                                                                        중복된 아이디입니다.
        alert("사용할 아이디를 입력하세요.");
                                                                                        사용할 아이디를 입력하세요.
        $("#id").focus();
```

[customer]

```
[회원 가입 (3) – 회원 가입]
 $("#join").click(function(){
    if(!stId){
        alert("아이디 중복확인이 필요합니다.");
                                                                              저장된 변수를 통해 중복 확인 유무를 먼저 검사합니다.
    }else if(!stEmail){
        alert("이메일 중복확인이 필요합니다.");
                                                                                              function checkIt() {
        var stEmpty = checkIt();
                                                                                                  if(!$("#id").val()) {
                                                                                                     alert("아이디를 입력하세요");
        if(stEmpty){
                                                                                                     $("#id").focus();
                                                                       모든 필드를 정상적으로
                                                                                                     return false;
                                                                       입력하였는지 검사합니다.
                   id : $("#id").val(),
                   pw : $("#pw").val(),
                   name : $("#name").val(),
                                                                                                  if(!$("#pw").val()) {
                   tel: $("#tel").val(),
                                                                                                     alert("비밀번호를 입력하세요");
                   address : $("#address").val(),
                                                                                                     $("#passwd").focus();
                   email : $("#email").val()
                                                                                                     return false;
           $.ajax({
                                                                                                  if(!$("#name").val()) {
               type: "POST",
                                                                                                     alert("사용자 이름을 입력하세요");
               url : "/MarketKurly Portfolio/joinPro.do",
                                                                                                     $("#name").focus();
               data : query,
                                                                                                     return false;
               success : function(data){
                                                                                                  if(!$("#tel").val()) {
                   alert("회원가입이 완료되었습니다.");
                                                                                                     alert("전화번호를 입력하세요");
                   window.location.href = "/MarketKurly Portfolio/index.do";
                                                                                                     $("#tel").focus();
                                                                                                     return false;
               error : function(data){
                   alert("오류가 발생했습니다.")
                                                                                                  if(!$("#address").val()) {
                                                                                                     alert("주소를 입력하세요");
                                                                                                     $("#address").focus();
                                                                                                     return false;
                                                                                                  if(!$("#email").val()) {
                                                                                                     alert("이메일을 입력하세요");
                                                                                                     $("#email").focus();
String id = request.getParameter("id");
                                                                                                     return false;
String pw = request.getParameter("pw");
                                                전달된 데이터를 customerDAO의 메소드를 활용
 String name = request.getParameter("name");
                                                하여 데이터베이스에 회원 정보를 추가합니다.
 String tel = request.getParameter("tel");
                                                                                                  return true;
 String address = request.getParameter("address");
 String email = request.getParameter("email");
 customerDAO.getInstance().insertMember(id, pw, name, tel, address, email);
```

[customer]

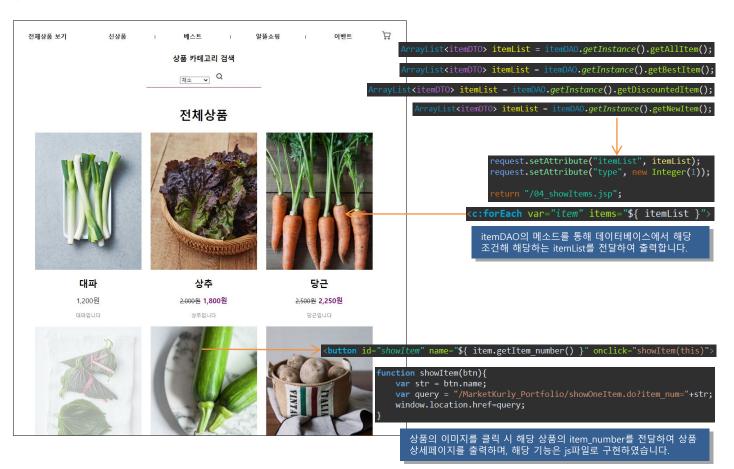
[로그인 & 로그아웃]

- 상품 수정

```
String id = request.getParameter("id");
              로그인
                                        String pw = request.getParameter("pw");
                                        int check = customerDAO.getInstance().userCheck(id, pw);
  아이디를 입력하세요
                                       System.out.println("loginPro check = " + check);
                                        System.out.println("id = " + id);
  비밀번호를 입력하세요
                                        request.setAttribute("check", new Integer(check));
                                       request.setAttribute("id", id);
                                       request.setAttribute("type", new Integer(1));
                로그인
                                                                            전달된 데이터를 customerDAO의 메소드를 활용
                                                                            하여 데이터베이스에 있는 회원 정보와 비교 후
               회원가입
                                                                            check 값을 전달합니다.
<form method="post" action="/MarketKurly Portfolio/loginPro.do">
                                                                      전달받은 check 값을 검사하여 입력한 id, pw 값이 데이터
                                                                      베이스에 저장된 정보와 일치할 경우 index.do를 실행하며,
                                                                      그렇지 않을 경우 이전 페이지로 이동합니다.
onclick="location.href='/MarketKurly Portfolio/join.do'"
                                                       <c:if test="${ check == 1 }">
                                                          <c:set var="id" value="${ id }" scope="session"/>
- 로그아웃
                                                          <script>
                                                              location.href = '/MarketKurly Portfolio/index.do';
<c:remove var="id" scope="session"/>
                                                          </script>
<script>
   alert("로그아웃 되었습니다.");
   location.href = "/MarketKurly Portfolio/index.do";
                                                       <c:if test="${ check != 1 }">
 /script>
                                                          <script>
                                                              alert("아이디와 비밀번호를 확인하세요.");
로그아웃 되었습니다.
                                                              history.go(-1);
                                                          </script>
```

[customer]

[상품 목록 (1)]



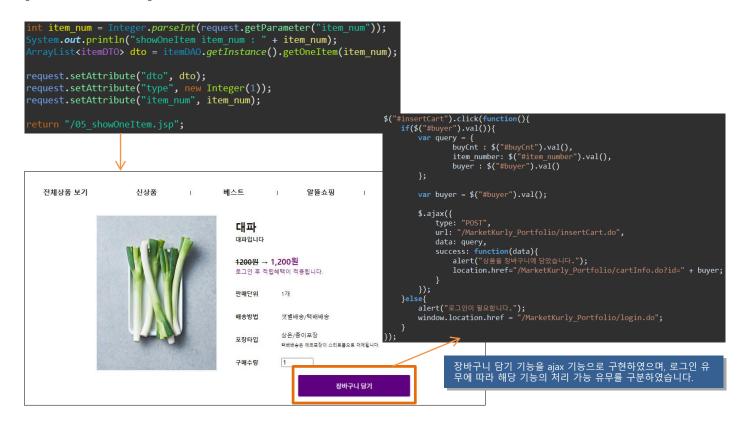
[customer]

[상품 목록 (2)]



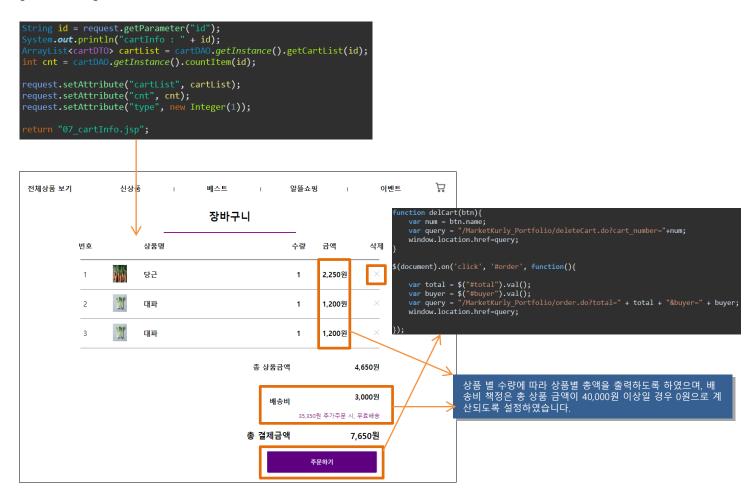
[customer]

[상품 상세페이지]



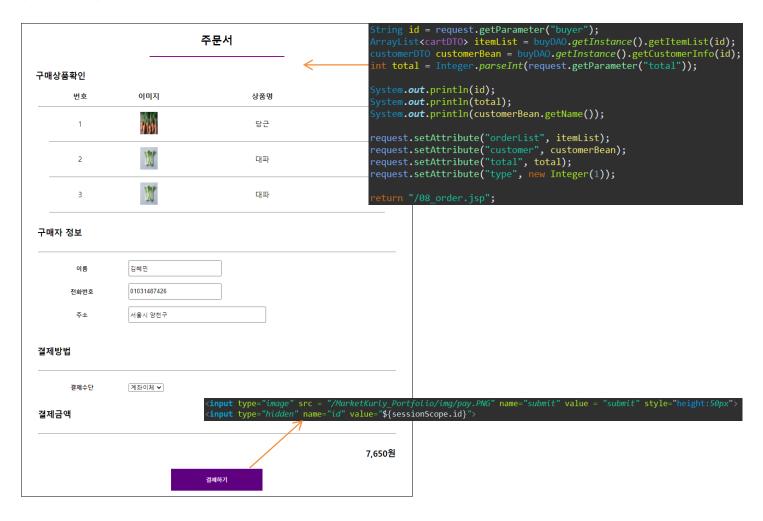
[customer]

[장바구니]



[customer]

[주문하기]



[customer]

[주문처리]

```
String customer id = request.getParameter("id");
                                                                                    Request에 저장된 id값을 가져와 해당 정보를 바탕으로 DB에
customerDTO customerBean = buyDAO.getInstance().getCustomerInfo(customer id);
                                                                                    서 고객정보를 customerDTO에 저장하고 장바구니 정보를 가
ArrayList<cartDTO> cartList = cartDAO.qetInstance().getCartList(customer id);
                                                                                    져와 저장합니다.
 nc paynow = inceger.purseinc(requesc.gecrarameter( nowray ));
String pay = "";
ovstem.out.println(customer id);
vstem.out.println(payhow);
 f(payhow == 1) {
   pay = "계좌이체";
                                                                                    추가로 입력한 결제 방식을 String 형식에 저장한 뒤 DB에 데
}else if(payhow == 2) {
                                                                                    이터를 저장합니다.
   pay = "신용카드";
 or(int i=0; i<cartList.size(); i++) {
   cartDTO bean = cartList.get(i);
   String customer_name = customerBean.getName();
   int cart number = bean.getCart number();
   String item name = bean.getItem name();
   int buy price = bean.getBuy price();
   int buy count = bean.getBuy count();
   String item_image = bean.getItem_image();
   String howpay = pay;
   String address = customerBean.getAddress();
                                                                                    받아온 고객 및 장바구니 정보를 바탕으로 buyDTO를 생성하
   System.out.println("item_name : " + item_name);
                                                                                    고, 이를 DB에 INSERT 합니다.
   System.out.println("buy count : " + buy count);
   buyDTO dto = new buyDTO();
   dto.setCustomer id(customer id);
   dto.setCustomer name(customer name);
   dto.setCart number(cart number);
   dto.setItem name(item name);
   dto.setBuy price(buy price);
   dto.setBuy_count(buy_count);
   dto.setItem_image(item_image);
   dto.setHowpay(howpay);
   dto.setAddress(address);
   buyDAO.getInstance().insertOrderList(dto);
   buyDAO.getInstance().updateSold(item name, buy count);
 uyDAO.getInstance().deleteCartList(customer id);
 eturn "/09 insertOrderList.jsp";
```

[customer]

[주문내역]

```
String id = request.getParameter("id");
System.out.println("checkMyOrderList id = " + id);
ArrayList<buyDTO> buyList = buyDAO.getInstance().getBuyList(id);
int cnt = buyList.size();
System.out.println(cnt);

request.setAttribute("buyList", buyList);
request.setAttribute("cnt", cnt);
request.setAttribute("type", new Integer(1));

RequestM 저장된 id값으로 DB에서 저장된 id에 해당하는
BuyList를 호출하여 출력합니다.
```

주문내역 확인						
번호		상품명	수량	가격	주문일	결제수단
1		연어	1	14000	2021-02-28 15:28:37	계좌이체
2	M	당근	1	2250	2021-02-28 15:28:37	계좌이체
3		치즈돈가스	1	13000	2021-02-28 15:28:37	계좌이체
4		연어	1	14000	2021-02-28 15:28:37	계좌이체
5	rin	당근	1	2250	2021-02-28 15:44:28	계좌이체

감사합니다.

현재 프로젝트의 코드는 아래 github 링크에서 확인하실 수 있습니다.

https://github.com/woalskdl/MarketKurly_Clone

추가로 현재 진행 중인 Spring 프로젝트는 아래 github 링크에서 확인하실 수 있습니다.

https://github.com/woalskdl/CallMeBack

- 관련 기술 스택 : Spring, SpringBoot, Mybatis, Ajax, Lombok, Bootstrap 등

남 재 민

woalskdl@gmail.com