



happy harvest

**e-mail** [ijklo2@naver.com](mailto:ijklo2@naver.com)

**Github** <https://github.com/Lee-Jae-Woong/shop>



프로젝트 명

Happy Harvest(수확행)

프로젝트 목표

쇼핑몰의 주요기능 파악 및 구현

개발환경

Tomcat / SpringFramework/ Maven / OricleDB/ Mybatis

개발언어

java / javascript / css / html /Jquery(Ajax)

IDE(개발도구)

Eclipse, Sqldeveloper, VScode

# ≡ 목차

---

→ 로그인/회원가입

→ 상품관리

→ 장바구니

→ 주문하기

→ 마이페이지

# Oracle DB테이블 정보

## 회원정보

THEZONE.USER_INFO	
P * USERNUM	NUMBER
* NAME	VARCHAR2 (20 BYTE)
* ID	VARCHAR2 (50 BYTE)
* PW	VARCHAR2 (100 BYTE)
* EMAIL	VARCHAR2 (50 BYTE)
* TEL	VARCHAR2 (20 BYTE)
* DATA	DATE
* MADDR	VARCHAR2 (20 BYTE)
* LADDR	VARCHAR2 (200 BYTE)
* DLADDR	VARCHAR2 (100 BYTE)
* LASTDATA	DATE
USER_INFO_PK (USERNUM)	

## 상품정보

THEZONE.GDS_INFO	
P * GDSNUM	NUMBER
* GDSNAME	VARCHAR2 (100 BYTE)
* CATECODE	NUMBER
* GDSPRICE	NUMBER
* GDSCOLOR	VARCHAR2 (100 BYTE)
* GDSSTOCK	NUMBER
* GDSDETAIL	VARCHAR2 (500 BYTE)
* GDSIMG	VARCHAR2 (500 BYTE)
* THUMBIMG	VARCHAR2 (100 BYTE)
* REGDATE	DATE
* GDSDESC	VARCHAR2 (500 BYTE)
GDS_INFO_PK (GDSNUM)	

THEZONE.CATE_INFO	
P * CATENUM	NUMBER
* CATENAME	VARCHAR2 (100 BYTE)
* CATECODE	NUMBER
CATE_INFO_PK (CATENUM)	

THEZONE.GDS_REVIEW	
P * RVNUM	NUMBER
* GDSNUM	NUMBER
* ID	VARCHAR2 (50 BYTE)
* RVTEXT	VARCHAR2 (2000 BYTE)
* RVIMG	VARCHAR2 (200 BYTE)
* RVDATE	DATE
* RVSCORE	NUMBER
GDS_REVIEW_PK (RVNUM)	

## 장바구니

THEZONE.USER_CART	
P * CARTNUM	NUMBER
* USERNUM	NUMBER
* GDSNUM	NUMBER
* CARTCOUNT	NUMBER
* CARTCOLOR	VARCHAR2 (50 BYTE)
USER_CART_PK (CARTNUM)	

## 결제/적립금

THEZONE.PAYMENT	
P * PAYNUM	NUMBER
* USERNUM	NUMBER
* ORDNAME	VARCHAR2 (500 BYTE)
* ORDNUM	VARCHAR2 (500 BYTE)
* ORDCOLOR	VARCHAR2 (500 BYTE)
* ORDCOUNT	VARCHAR2 (500 BYTE)
* ORDPRICE	VARCHAR2 (500 BYTE)
* PAYDATE	TIMESTAMP
* RECEIVER	VARCHAR2 (100 BYTE)
* EMAIL	VARCHAR2 (100 BYTE)
* PHONE	VARCHAR2 (100 BYTE)
* ADDR1	VARCHAR2 (100 BYTE)
* ADDR2	VARCHAR2 (200 BYTE)
* ADDR3	VARCHAR2 (100 BYTE)
* MESSAGE	VARCHAR2 (200 BYTE)
* DEPOSITOR	VARCHAR2 (100 BYTE)
* BANKNAME	VARCHAR2 (100 BYTE)
* PAYMETHOD	VARCHAR2 (100 BYTE)
* CAL	NUMBER
* POINT	NUMBER
* USEPOINT	NUMBER
* USECOUPONCAL	NUMBER
* CPNUM	NUMBER
PAYMENT_PK (PAYNUM)	

## 쿠폰정보

THEZONE.MGRCOUPON	
P * MGRCP	NUMBER
* CPNAME	VARCHAR2 (50 BYTE)
* ITEMSCOPE	VARCHAR2 (50 BYTE)
* CPRATE	NUMBER
* EXPDATE	VARCHAR2 (50 BYTE)
* CPCODE	VARCHAR2 (20 BYTE)
* CPCODE2	VARCHAR2 (20 BYTE)
* CPCODE3	VARCHAR2 (20 BYTE)
* CPCODE4	VARCHAR2 (20 BYTE)
MGRCOUPON_PK (MGRCP)	

THEZONE.COUPON	
P * CPNUM	NUMBER
* CP CODE	VARCHAR2 (20 BYTE)
* USERNUM	NUMBER
* USECP	VARCHAR2 (100 BYTE)
COUPON_PK (CPNUM)	

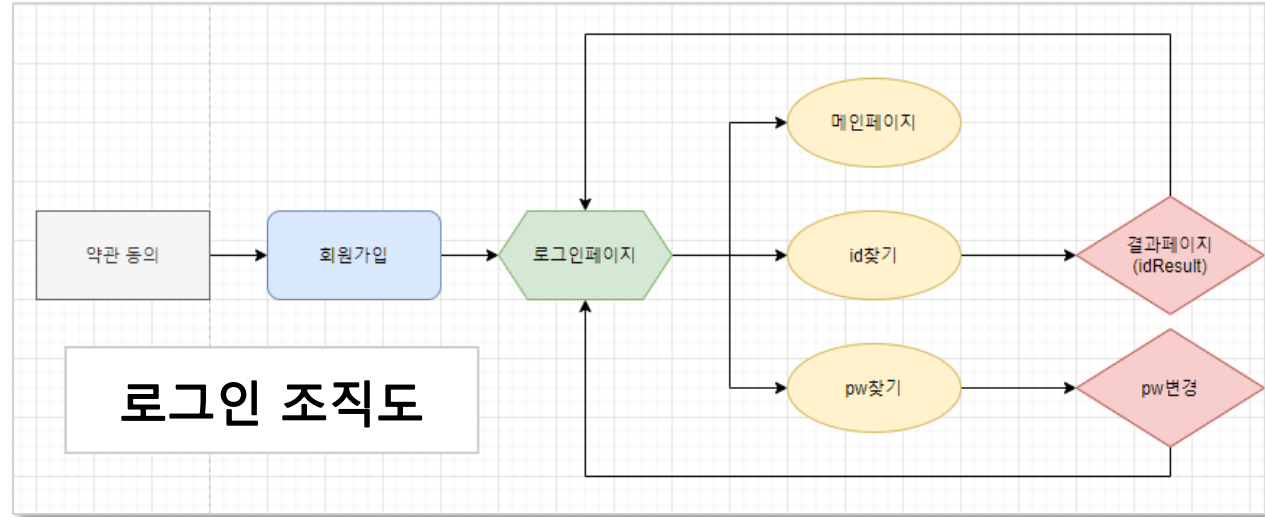
# 로그인/회원가입

1. 로그인

2. 회원가입

3. Id/pw찾기

4. 회원정보 수정



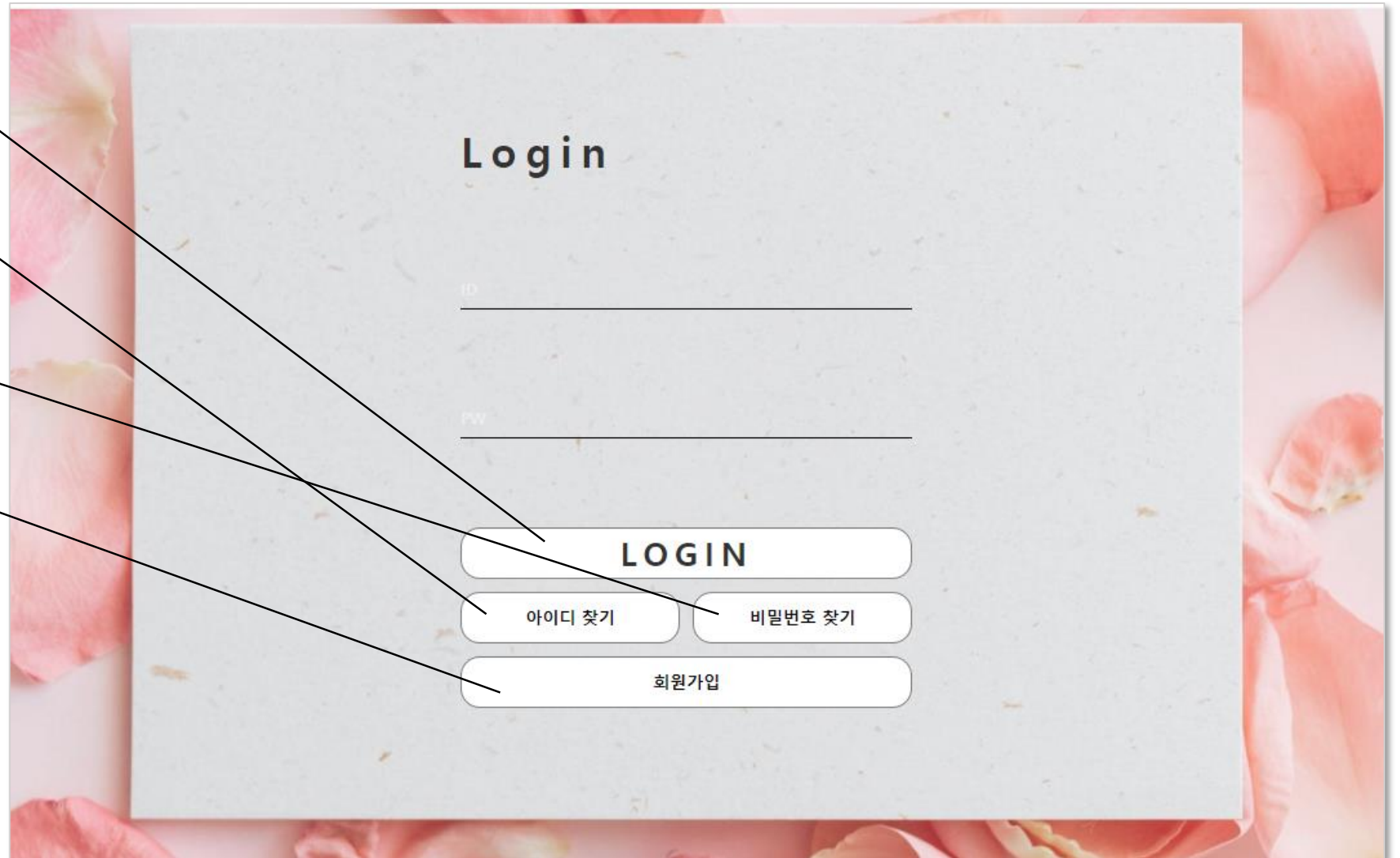
# 1. 로그인

암호화된 비밀번호를 비교하여  
로그인 성공 여부 판단

이름/이메일/전화번호를 가지고  
아이디 검색

아이디/이메일/전화번호를 가지고  
비밀번호 변경페이지로 이동

회원가입 페이지로 이동



The image shows a login form titled "Login" on a light gray background. The form includes two input fields: "ID" and "PW". Below the input fields are three buttons: a large "LOGIN" button, a smaller "아이디 찾기" (Find ID) button, and a smaller "비밀번호 찾기" (Find Password) button. At the bottom of the form is a button labeled "회원가입" (Sign Up). Four lines with arrows point from the text boxes on the left to specific elements on the form: the first line points to the "LOGIN" button, the second line points to the "아이디 찾기" button, the third line points to the "비밀번호 찾기" button, and the fourth line points to the "회원가입" button.

Login

ID

PW

LOGIN

아이디 찾기

비밀번호 찾기

회원가입

## 2. 회원가입

DB에 저장된 아이디와 비교해서  
중복결과를 알려줌

입력된 비밀번호 암호화

에러 발생시 유효성 검사로  
에러 메시지 반환

버튼 클릭 시 도로명 가져오기 창 오픈

클릭 시 home으로 이동

입력된 값을 Mapper로 Insert하여  
db에 저장함

### Sign Up

이름  ! 회원 이름은 2~15 글자여야 합니다

아이디  ! 아이디는 4~20 글자여야 합니다  
중복확인

비밀번호

비밀번호확인  ! 숫자 영문자 특수 문자를 포함한 8 ~ 12 자를 입력하세요.

이메일  ! 비밀번호를 확인해주세요  
! 이메일 형식으로 작성해주세요

전화번호  ! "-"없이 입력해주세요

### Address

우편번호  검색

도로명 주소

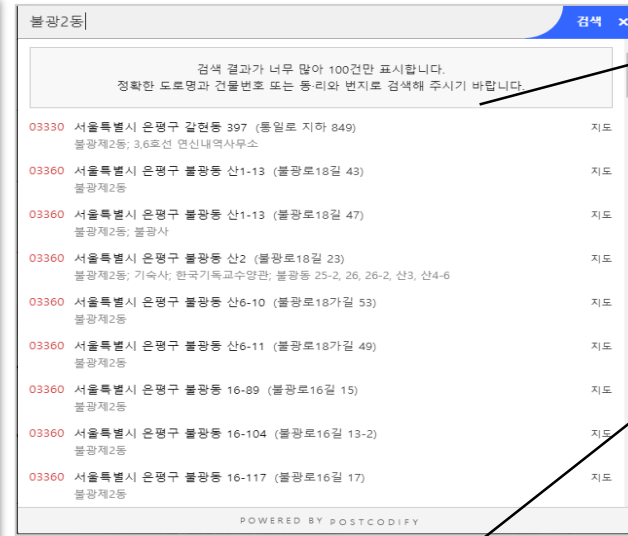
상세주소



## 2-1. 도로명 주소 가져오기

오픈 API 활용 → 우편주소 OR 도로명 주소를 불러오기

```
<div class="addrinfo">
  <h3 class="addrtitle">Address</h3>
  <div>
    <h4 class="addr">우편번호</h4>
    <div class="addrinput">
      <form:input path="mAddr" name="mAddr" class="postcodify_postcode5" />
      <button type="button" id="postcodify_search_button">검색</button>
      <script src="//ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
      <script src="//d1p7wdleee1q2z.cloudfront.net/post/search.min.js"></script>
      <script>
        $(function() {
          $("#postcodify_search_button").postcodifyPopUp();
        });
      </script>
    </div>
    <form:errors path="mAddr" class="addrerror"/>
  </div>
  <h4 class="addr">도로명 주소</h4>
  <div class="addrinput">
    <form:input path="lAddr" name="lAddr" class="postcodify_address" />
    <form:errors path="mAddr" class="addrerror"/>
  </div>
  <h4 class="addr">상세주소</h4>
  <div class="addrinput">
    <form:input path="dlAddr" name="dlAddr" class="postcodify_details" />
    <form:errors path="dlAddr" class="addrerror"/>
  </div>
</div>
```



도로명 주소 찾기

주소 선택 시 자동 입력  
(상세주소는 입력 안됨)

*Address*

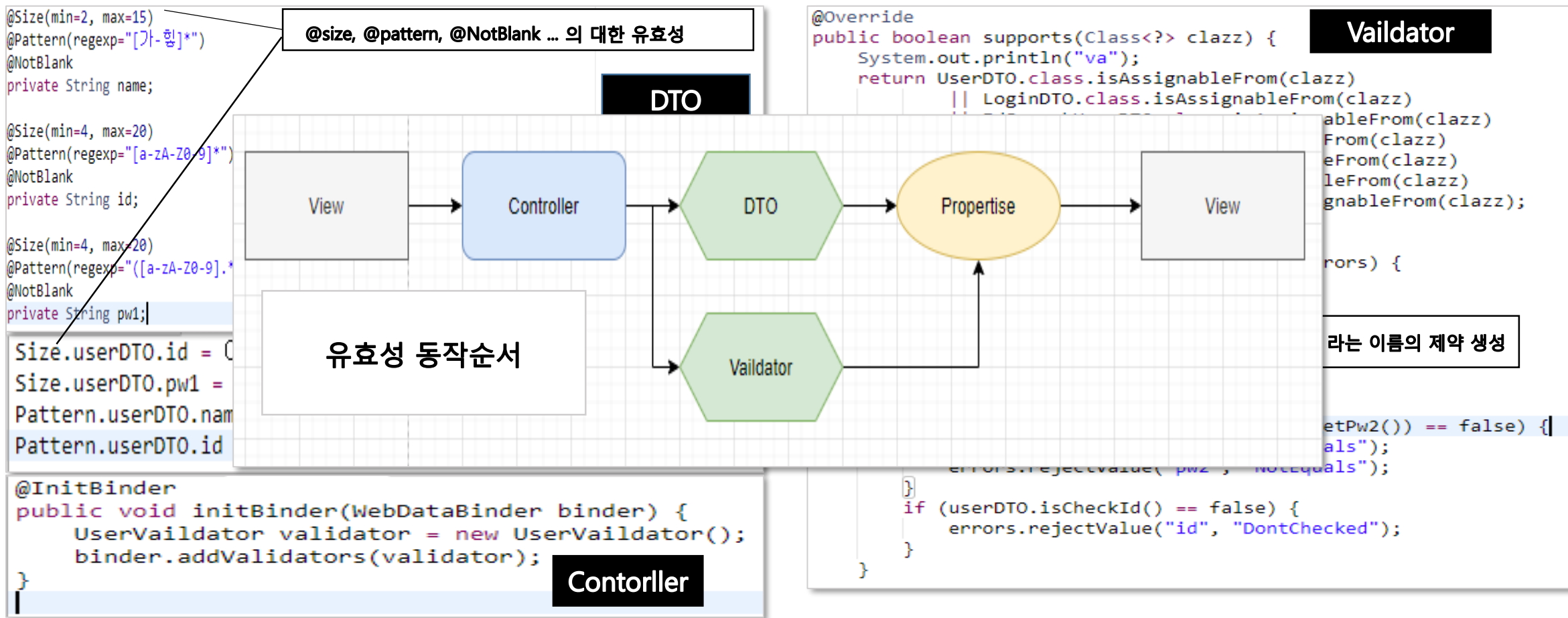
우편번호	03360	검색
도로명 주소	서울특별시 은평구 불광로18길 43	
상세주소	301호	



## 2-2. 유효성 검사

- 1) Validation(@Size,@Pattern,@NotBlank ... ) 사용 → DTO/VO의 매개변수에 제약 >> Error 발생 시 화면에 출력
- 2) DTO/VO 바인딩 → 새로운 유효성을 생성하여 개발자가 필요한 상황에서 사용 가능.

유효성 검사에 대한 이름 생성 후 Controller에서 Error 발생 시 유효성을 호출하여 view 로 표시.



## 2-3. Password 암호화

- 1) BCryptPasswordEncoder의 'encode() 메소드' 사용 → 입력된 값을 임의의 문자열로 변환 → DB에 저장
- 2) 암호화 된 값을 DB에서 호출할 경우, DB에 저장 된 문자열을 반환
- 3) 암호를 인증 할 경우 'matches() 메소드' 를 사용  
→ '새롭게 입력된 값' 과 DB에 있는 '암호화 된 문자열'과 비교

>> 같다면 true, 같지 않으면 false값 반환

```
@RequestMapping(value = "/login/login_proc", method = {RequestMethod.POST,
    RequestMethod.GET})
public String loginproc(
    @Valid @ModelAttribute("loginUserDTO") LoginDTO loginUserDTO,
    BindingResult result, HttpSession session) {

    if (result.hasErrors()) {
        System.out.println("result.hasErrors");

        String pw1 = userService.selectLoginPw(loginUserDTO);
        String pw2 = loginUserDTO.getPw1();
        if (passwordEncoder.matches(pw2, pw1) == true) {
            loginUserDTO.setLogLogin(true);
            System.out.println("result.hasErrors");
            userService.loginSelect(loginUserDTO);
            session.setAttribute("loginUser", loginUserBean);
            return "redirect:/";
        }
        loginUserDTO.setLogLogin(false);
        return "user/login";
    }
    return "user/login";
}
```

**Contorller**

**Matches()메소드 사용 예시**

```
@PostMapping(value = "/join/join_proc")
public String join_proc(@Valid @ModelAttribute("userDTO") UserDTO userDTO,
    BindingResult result) {
    if (result.hasErrors()) {
        return "user/join";
    }
    String pw1 = passwordEncoder.encode(userDTO.getPw1());
    userDTO.setPw1(pw1);
    userService.insertUserInfo(userDTO);
    return "redirect:joinResult";
}
```

**Contorller**

**encode()메소드 사용 예시**

데이터베이스의 저장된 암호 : \$2a\$10\$brgd/N/Pvy9MMFuXNiVvMOhN5m.uBB1TZzwIEea.pYHltKpQakmS  
로그인시 새롭게 입력된 값 : wkdgns1243!#  
matches()메소드의 값 : true

**로그인 성공 시 나오는 log값**

데이터베이스의 저장된 암호 : \$2a\$10\$brgd/N/Pvy9MMFuXNiVvMOhN5m.uBB1TZzwIEea.pYHltKpQakmS  
로그인시 새롭게 입력된 값 : a1234567!  
matches()메소드의 값 : false

**로그인 성공 시 나오는 log값**

SQL | 인출된 모든 행: 2(0.129초)

USERNUM	NAME	ID	PW
1	1 name	id	a1234567!
2	2 조장훈	spahslzk	\$2a\$10\$brgd/N/Pvy9MMFuXNiVvMOhN5m.uBB1TZzwIEea.pYHltKpQakmS

## 4. 회원정보 수정

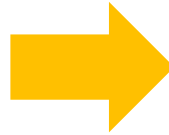
- 마이페이지에서 회원정보수정 클릭 시 인증페이지로 이동 → 비밀번호 인증 후 '수정 페이지'로 이동
- 비밀번호 인증 시 로그인과 같은 방식으로 암호화된 비밀번호 비교

### 회원정보 수정

아이디      spahslzk

비밀번호      \_\_\_\_\_

회원정보 확인하기



### 회원정보 수정

이름	조장훈
아이디	spahslzk
비밀번호	_____
비밀번호 확인	_____
이메일	_____
전화번호	_____

### 주소

우편번호	_____	검색
도로명 주소	_____	

로그인 버튼과 같이 비밀번호 인증 후  
수정페이지로 이동

회원탈퇴

클릭 시 최종 확인 후 DB에서 삭제

변경하기

클릭 시 DB에서 수정

# 🔑 상품관리



Spring

Peony

리뷰 4개

마감임박

₩2000 x 1ea



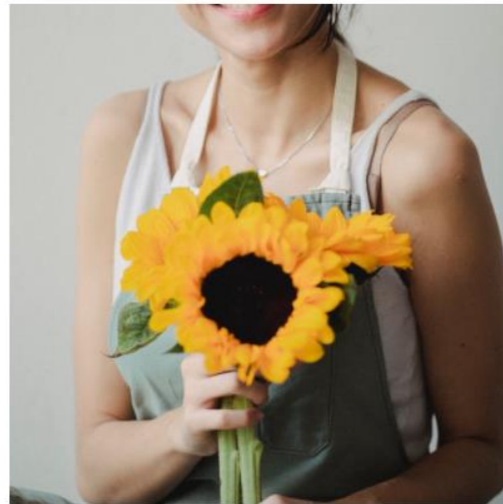
Summer

Rose

리뷰 2개

Sold Out

₩2500 x 1ea



Fall

Sunflower

리뷰 0개

₩5000 x 1ea

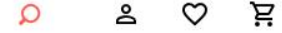



# 1. 상품 리스트



김동률 님 로그인  
반갑습니다.


All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매






Spring  
Peony  
리뷰 4개


마감임박  
₩2000 x 1ea






Summer  
Rose  
리뷰 2개


Sold Out  
₩2500 x 1ea





Fall  
Sunflower  
리뷰 0개

₩5000 x 1ea



[정적 Data]  
상품 코드 (Spring, Summer ...)  
상품명  
상품 가격

[동적 Data]  
Stock 관리 (마감임박, Sold Out...)  
리뷰 개수  
관심상품 여부 확인



# 1. 상품 리스트

```
@RequestMapping(value = "/shoplist", method = {RequestMethod.POST, RequestMethod.GET})  
public String shopList(Model model){
```

```
List<GdsInfoDTO> gdsInfoList = gdsService.getAllGdsList();
```

① 상품 목록 전체 가져오기

```
if(loginUserBean.isCheck() == true) {  
    List<Integer> userLikeList = userLikeService.getUserLikeGdsList(loginUserBean.getUserNum());
```

```
    for(GdsInfoDTO dto : gdsInfoList) {  
        long num1 = dto.getGdsNum();  
        for(Integer integer : userLikeList) {  
            long num2 = (long) integer;  
            if(num1 == num2) {  
                dto.setLike(true);  
            }  
        }  
    }  
}
```

② 로그인 여부 확인 후, 위시리스트 목록 가져오기

③ 상품목록 DTO에, 회원의 위시 여부 체크

```
public class GdsInfoDTO {  
  
    private long gdsNum;  
    private String cateName;  
    private String gdsName;  
    private int gdsPrice;  
    private int gdsStock;  
    private String thumbImg;  
    private int review_count;  
    private boolean like;  
  
    ...  
}
```

# 1. 상품 리스트



Spring

Peony

리뷰 4개

마감임박

₩2000 x 1ea



[shoplist.jsp]

```
<c:choose>
  <c:when test="${dto.gdsStock < 1 }">
    <span class="stock_state">Sold Out</span>
  </c:when>
  <c:when test="${dto.gdsStock < 30 }">
    <span class="stock_state">마감임박</span>
  </c:when>
</c:choose>
```

[GdsMapper.java]

```
public interface GdsMapper {
    ...
    @Select("select a.gdsNum, a.gdsName , b.cateName, a.gdsPrice , a.gdsStock, a.thumbImg " +
        "count(c.gdsNum) as review_count from gds_info a " +
        "left join gds_review c on a.gdsNum=c.gdsNum " +
        "left join cate_info b on a.catecode=b.catecode " +
        "group by a.gdsNum, a.gdsName, b.cateName, a.gdsPrice, a.gdsStock, a.thumbImg " +
        "order by a.gdsNum")
    List<GdsInfoDTO> getAllGdsList();
    ...
}
```

[DB]

	GDSNUM	GDSNAME	CATENAME	GDSPRICE	GDSSTOCK	THUMBIMG	REVIEW_COUNT
1	1	Peony	Spring	2000	22	1_thumb.jpg	4
2	2	Rose	Summer	2500	0	2_thumb.jpg	2
3	3	Sunflower	Fall	5000	939	3_thumb.jpg	2



# 1. 상품 리스트

## ① 로그인 여부 체크



Spring

마감임박

Peony

₩2000 x 1ea

리뷰 47



Click발생

Value값으로 구별

```
$(document).ready(function() {  
    $(".favorite > img").on("click", function() {  
        let like = $(this).parent().attr('value');  
        let gdsNum = $(this).attr('value');  
        let check= "<c:out value='${sessionScope.loginUser.check}'/>";  
        if(check === "false"){  
            alert("로그인 후 사용가능합니다");  
            location.href='${root}login';  
            return false;  
        }  
    })  
})
```

```
let userNum = "<c:out value='${sessionScope.loginUser.userNum}'/>";  
let result={"gdsNum" : gdsNum, "userNum": userNum};
```

```
if(like == 'like'){
```

## ② 상품번호 / 회원 번호 저장

```
fetch('${root}shop/like',{  
    method: 'POST',  
    headers:{  
        'Content-Type':'application/json;charset=utf-8'  
    },  
    body:JSON.stringify(result)  
})
```

## ③ JSON형태로 데이터 전송

```
$(this).prop("src", "images/favorite-black.svg");  
like.setAttribute('value','dislike');
```

```
}else{
```

```
fetch('${root}shop/dislike',{  
    method: 'POST',  
    headers:{  
        'Content-Type':'application/json;charset=utf-8'  
    },  
    body:JSON.stringify(result)  
})
```

<img> src 변경

```
$(this).prop("src", "images/favorite_border.svg");  
like.setAttribute('value','like');
```

## 2. 상품 페이지



상품번호: 3

Sunflower ₩5000

맥시밀리언

색상 --선택하세요-- ▼ 수량 --선택하세요-- ▼

구매하기

② 장바구니 추가하기



① 옵션 선택 후 구매하기

## 2. 상품 페이지 - 리뷰 작성하기

### ① 평점 선택

업로드

평 점 ☐1 ☐2 ☐3 ☐4 ☒5

Send



예쁘고 싱싱한 꽃이에요!! 잘 받았습니다 :) 선물용으로도 딱이에요 |

### ② 이미지 파일 선택

- 파일 사이즈 제한(3Mb)
- 파일 개수 제한 (3개 이하)

```
function readAndPreview(files){

    let reader = new FileReader();
    let maxSize = 3 * 1024 * 1024;
    let fileSize = 0;
    reader.addEventListener("load",function(){

        if(files.size > maxSize){
            alert("3MB이하의 파일만 업로드 할수 있습니다");
            return false;
        }
        let image = new Image();
        image.width = "200";
        image.height = "200";
        image.title = files.name;
        image.src = this.result;
        filenames.push(image.title);

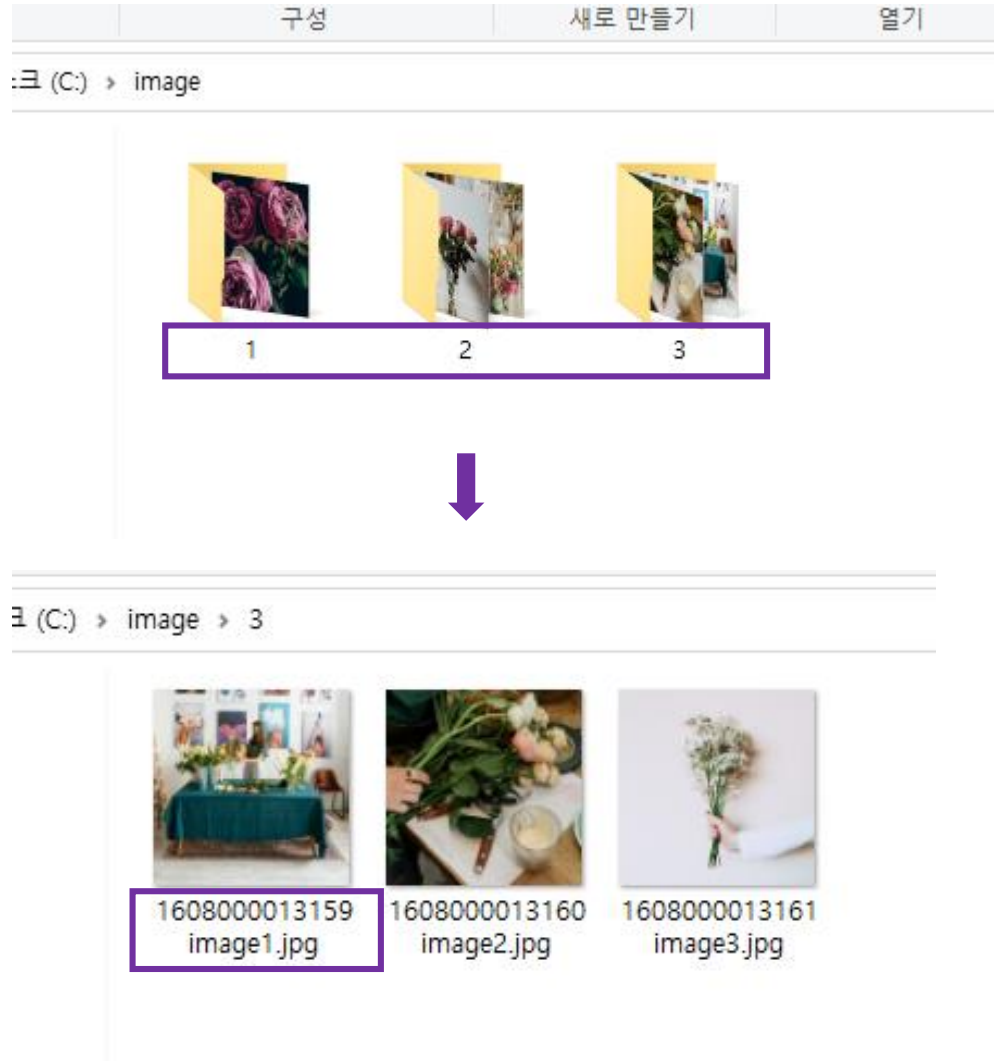
        document.getElementById("thumbings").appendChild(image);
    },false);

    if(files){
        reader.readAsDataURL(files);
    }

    if(files){
        if(files.length > 3){
            alert("3개 이하만 업로드 가능합니다 ");
            return false;
        }
        [].forEach.call(files,readAndPreview);
    }

}
```

## 2. 상품 페이지 - 리뷰 작성하기



<ReviewController.java>

```
String path = "C:\\image\\"+gdsNum+"\\";

if(! new File(path).exists()) {
    new File(path).mkdir();
}
```

① 저장 경로를 설정하고 상품 번호에 따라 리뷰 사진을 관리

```
if(!fileList.get(0).isEmpty()){
    for(Iterator<MultipartFile> i = fileList.iterator(); i.hasNext();) {
        MultipartFile mf = i.next();

        String originFileName = mf.getOriginalFilename(); // 원본 파일 명
        long fileSize = mf.getSize();
        String saveFileName = System.currentTimeMillis() + originFileName;
        String realFileName = path + saveFileName;

        try {
            mf.transferTo(new File(realFileName));
        } catch (IllegalStateException e) {
            e.printStackTrace();
        } catch (IOException e) {
        }
    }
}
```

② currentTimeMillis()를 사용해 파일 이름 중복 방지

## 2. 상품 페이지 - 리뷰 가져오기

상품정보

리뷰작성

리뷰쓰기

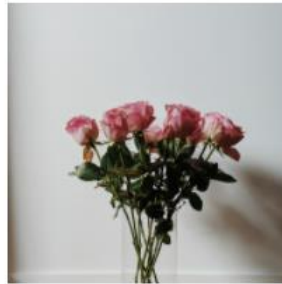
총 평점: 3.5(2개의 리뷰)

① 전체 리뷰 평점 생성

ID testId | 작성일: 2020-12-07 14:56:08 | 평 점: 2

테스트리뷰4

ID mong123 | 작성일: 2020-12-09 15:06:54 | 평 점: 5



배송 잘 받았습니다^^ 방을 꾸렸더니 환해졌어요

② 리뷰 정보(ID/작성일/평점/이미지/리뷰내용) 불러오기

## 2. 상품 페이지 - 리뷰 가져오기

ID mong123 | 작성일: 2020-12-09 15:06:54 | 평 점: 5



<shop.jsp>

```
<c:set var="rvImgArr" value="${fn:split(rdto.rvImg,',')}" />
<td>
<c:forEach var="rvImg" items="${rvImgArr }">
  <c:if test="${rvImg ne '' }">
    <div style="display: inline-block; margin: 10px 10px; width: 200px; height: 200px; overflow: hidden;">
      
    </div>
  </c:if>
</c:forEach>
</td>
```

배송 잘 받았습니다^^ 방을 꾸몄더니 환해졌어음

### ② 리뷰 정보(ID/작성일/평점/이미지/리뷰내용) 불러오기

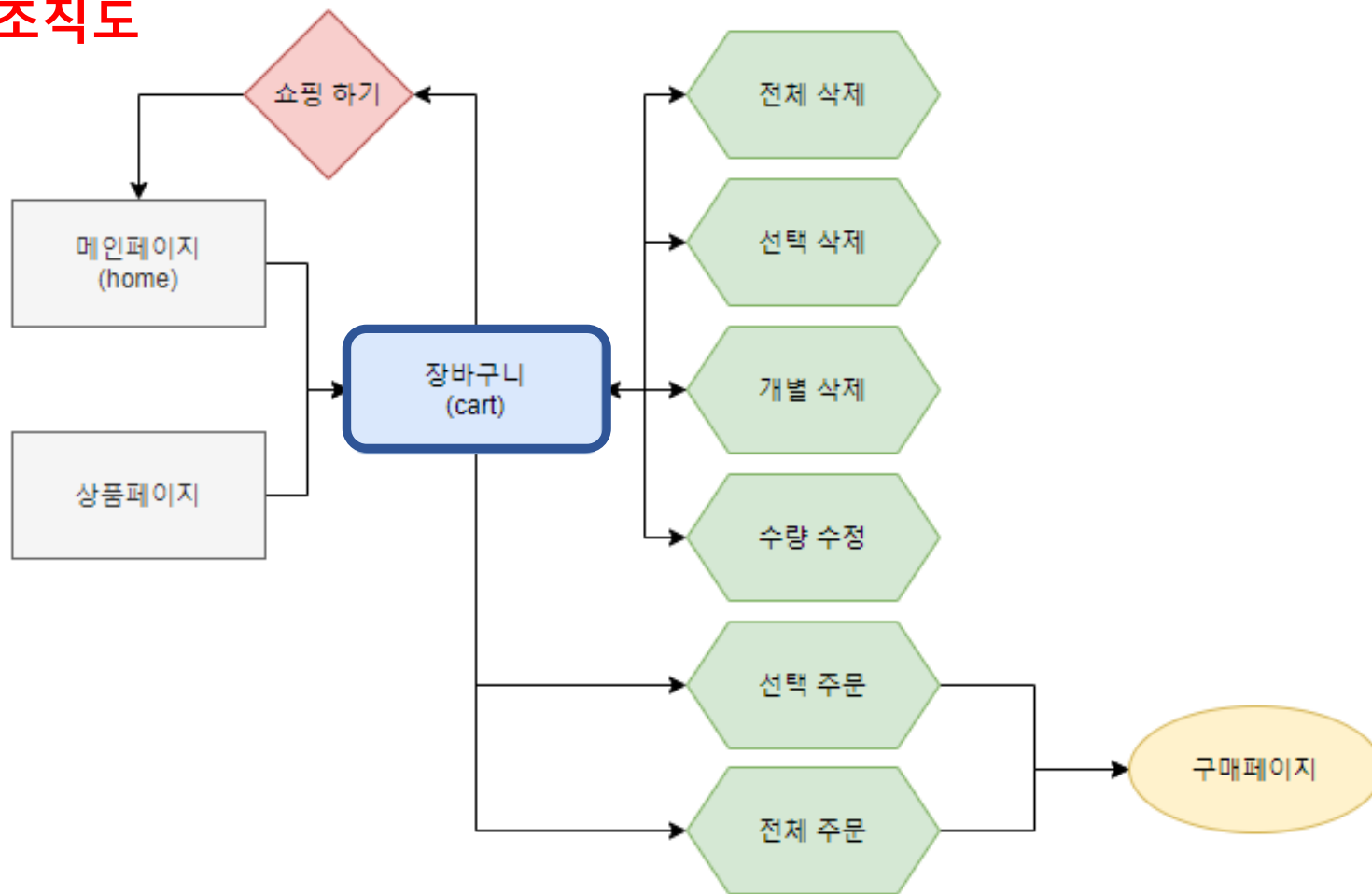
-

RVNUM	GDSNUM	ID	RVTEXT	RVIMG	RVDATE	RVSCORE
1	41	3mong123	예쁘고 싱싱...	1608000013159image1.jpg,1608000013160image2.jpg,1608000013161image3.jpg	20/12/15	5
2	44	3heesun0509	잘 받았습니...	(null)	20/12/15	4



# 장바구니

## 조직도





# 1. 주문 기능 (전체 주문)






회원님 로그인  
반갑습니다.

All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매



## 장바구니

■	사진	상품명	수량	적립금	가격	배송비	선택
<input type="checkbox"/>		상품번호 : 41 [옵션] color : red				[본배송] 조건	<input type="checkbox"/>
<input type="checkbox"/>		상품번호 : 42 [옵션] color : yellow				[본배송] 조건	<input type="checkbox"/>
<input type="checkbox"/>		상품번호 : 43 [옵션] color : pink				[본배송] 조건	<input type="checkbox"/>

선택상품 삭제

전체상품 주문

선택상품 주문

장바구니 비우기

쇼핑 계속하기

```
@GetMapping(value = "/cart/order")
public String order(Model model, GdsCartDTO gdsCartDTO) {

    int userNum = loginUserBean.getUserNum();

    List<UserCartDTO> CartList = cartService.getGdsList(userNum);
    List<PaymentDTO> myPage = paymentService.myPage(userNum);
    List<CouponInfoDTO> cpList = couponService.cpList(userNum);
    PointDTO totalPoint = paymentService.totalPoint(userNum);
    model.addAttribute("CartList", CartList);
    model.addAttribute("mp", myPage);
    model.addAttribute("cpList", cpList);
    model.addAttribute("totalPoint", totalPoint);

    return "order";    }
```

### Service

회원정보

장바구니

적립금

쿠폰리스트

주문페이지

# 1. 주문 기능 (선택 주문)



회원님 로그인  
반갑습니다.

All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매



장바구니에 담긴 상품 내에서  
선택 구매 가능

## 장바구니

사진	상품명	수량	적립금	가격	배송비	선택
<input checked="" type="checkbox"/>	상품번호 : 41 [옵션] color : red	- 20 + 수정	1000.0	50000	[기본배송] 조건	x
<input type="checkbox"/>	상품번호 : 42 [옵션] color : yellow	- 30 + 수정	3000.0	150000	[기본배송] 조건	x
<input checked="" type="checkbox"/>	상품번호 : 43 [옵션] color : pink	- 10 + 수정	1000.0	50000	[기본배송] 조건	x

선택 상품 주문

체크된 상품의 파라미터 값인 상품번호(cartNum)를 받아와  
Service를 호출한 후 DTO에 전달

선택상품 삭제

전체상품 주문

선택상품 주문

```
@GetMapping(value = "/cart/orderSelect")
public String orderSelect(@RequestParam(value = "num") ArrayList<Integer>
    cartNums, Model model) {

    int userNum = loginUserBean.getUserNum();
    List<PaymentDTO> myPage = paymentService.myPage(userNum);
    List<UserCartDTO> CartList = cartService.orderSelect(cartNums);
    PointDTO totalPoint = paymentService.totalPoint(userNum);
    List<CouponInfoDTO> cpList = couponService.cpList(userNum );
    model.addAttribute("mp", myPage);
    model.addAttribute("CartList", CartList);
    model.addAttribute("totalPoint", totalPoint);
    model.addAttribute("cpList", cpList);
    return "order";
}
```

## 2. 수량 수정 기능





회원님 로그인  
반갑습니다.

All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매



### 장바구니

수정 버튼 클릭 시  
수량 변경

■	사진	상품명	수량	적립금	가격	배송비	선택
<input type="checkbox"/>		상품번호 : 41 [옵션] color : red	<div>- 20 + 수정</div>	1000.0	50000	[기본배송] 조건	<input type="button" value="x"/>
<input type="checkbox"/>		상품번호 : 43 [옵션] color : pink	<div>- 20 + 수정</div>	800.0	40000	[기본배송] 조건	<input type="button" value="x"/>
총 구매금액 : 90000 원 <b>90000</b> 원 (적립금 : 1800.0원)							
회원님은 1원이상 구매 시 결제 금액의 2%를 적립해 드립니다.							

선택상품 삭제

전체상품 주문

선택상품 주문

장바구니 비우기

쇼핑 계속하기

### Cotroller

```
@GetMapping(value = "/cart/cartUpdate")
public String cartUpdate(@RequestParam int cartCount,
    @RequestParam int cartNum, GdsCartDTO gdsCartDTO) {
```

### DAO

```
public void cartUpdate(GdsCartDTO gdsCartDTO) {
    cartMapper.cartUpdate(gdsCartDTO); }
```

### Service

```
public void cartUpdate(GdsCartDTO gdsCartDTO) {
    cartDAO.cartUpdate(gdsCartDTO); }
```

### Mapper

```
@Update("update user_cart set cartcount =
    #{cartCount} where cartnum = #{cartNum}")
void cartUpdate(GdsCartDTO gdsCartDTO);
```

### 구현 과정 중 문제 발생

상품이 다수인 경우, 수량 수정 기능 작동 X

### 문제 해결

수정할 테이블에 인덱스를 설정 (명확성 부여)

```
$(function() {
    $('.btnUp').click(function() {
        let n = $('.btnUp').index(this);
        let num = $(".txtSpin:eq(" + n + ")").val();
        if (num < 100) {
            num = $(".txtSpin:eq(" + n + ")").val(num * 1 + 1);
        } else {
            num = 100;
        }
    });
});
```

### 3. 삭제 기능 (선택 삭제, 단일 삭제)



회원님 로그인  
반갑습니다.

All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매

🔍 👤 ❤️ 🛒

장바구니에 담긴 여러 상품 중  
원하지 않는 상품을 선택하여 삭제 가능

#### 장바구니

단일상품 삭제

사진	상품명	수량	적립금	가격	배송비	선택
<input checked="" type="checkbox"/>	상품번호 : 41 [옵션] color : red	- 20 + 수정	1000.0	50000	[기본배송] 조건	<input checked="" type="checkbox"/>
<input type="checkbox"/>	상품번호 : 42 [옵션] color : yellow	- 30 + 수정	3000.0	150000	[기본배송] 조건	<input type="checkbox"/>
<input checked="" type="checkbox"/>			800.0	40000	[기본배송] 조건	<input checked="" type="checkbox"/>
총 구매금액 : 240000 원 240000원 (적립금 : 4800.0원)						
회원님은 1원이상 구매 시 결제 금액의 2%를 적립해 드립니다.						

선택상품 삭제

반복문으로 DAO 호출

```
public void deleteChoice(ArrayList<Integer> cartNums) {  
    int index =0;  
    for(int Count : cartNums) {  
        int cartNum = cartNums.get(index);  
        cartDAO.deleteChoice(cartNum);  
        index++;  
    }  
}
```

선택상품 삭제

선택상품 삭제

```
@GetMapping(value = "/cart/deleteChoice")  
public String deleteChoice  
(@RequestParam(value = "num") ArrayList<Integer> cartNums) {
```

```
public void deleteChoice(ArrayList<Integer> cartNums){  
    int index =0;  
    for(int Count : cartNums) {
```

```
public void deleteChoice(int cartNum) {  
    cartMapper.deleteChoice(cartNum);  
}
```

```
@Delete("delete user_cart where cartnum= #{cartNum}")  
void deleteChoice(int cartNum);
```

Cotroller

Service

DAO

Mapper

단일상품 삭제

```
@GetMapping(value = "/cart/delete")  
public String delete  
(@RequestParam int cartNum, GdsCartDTO gdsCartDTO) {
```

```
public void delete(GdsCartDTO gdsCartDTO) {  
    cartDAO.delete(gdsCartDTO);  
}
```

```
public void delete(GdsCartDTO gdsCartDTO) {  
    cartMapper.delete(gdsCartDTO);  
}
```

```
@Delete("delete user_cart where cartnum= #{cartNum}")  
void delete(GdsCartDTO gdsCartDTO);
```

### 3. 삭제 기능 (장바구니 비우기)




회원님 로그인  
반갑습니다.

All.전체상품 Spring.봄 Summer.여름 Fall.가을 Winter.겨울 Etc SET.판매



## 장바구니

■	사진	상품명	수량	적립금	가격	배송비	선택
<input type="checkbox"/>		상품번호 : 41 [옵션] color : red	- 15 + 수정	750.0	37500	[기본배송] 조건	<input type="checkbox"/>
<input type="checkbox"/>		상품번호 : 42 [옵션] color : yellow	- 50 + 수정	5000.0	250000	[기본배송] 조건	<input type="checkbox"/>
<input type="checkbox"/>		상품번호 : 43 [옵션] color : pink	- 30 + 수정	1200.0	60000	[기본배송] 조건	<input type="checkbox"/>

총 구매금액 : 347500 원 347500 원

선택상품 삭제

장바구니 비우기

상품페이지

전체상품 주문

선택상품 주문

장바구니 비우기

쇼핑 계속하기

```
@GetMapping(value = "/cart/deleteAll")  
public String deleteAll(GdsCartDTO gdsCartDTO) {  
}
```

Cotroller

```
public void deleteAll(long userNum) {  
    cartDAO.deleteAll(userNum);  
}
```

Service

```
public void deleteAll(long userNum) {  
    cartMapper.deleteAll(userNum);  
}
```

DAO

```
@Delete("delete user_cart where usernum = #{userNum}")  
void deleteAll(long userNum);
```

Mapper



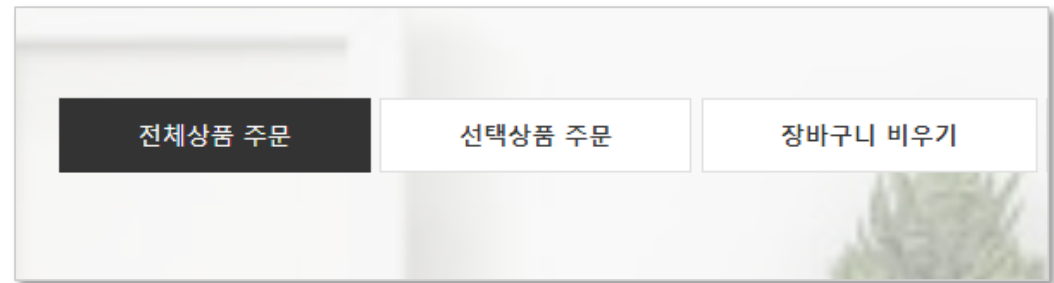
# ➦ 주문하기

## 방법 1. 상품 상세 페이지에서 바로 구매



Product detail page for Sunflower. The page shows a heart icon, product number 3, the name 'Sunflower', and the price '₩5000'. Below this, there are dropdown menus for '색상' (Color) and '수량' (Quantity), both with the placeholder text '--선택하세요--'. At the bottom, there is a blue button labeled '구매하기' (Purchase) and a circular icon with a shopping bag.

## 방법 2. 장바구니에서 선택 구매



Shopping cart interface showing three buttons: '전체상품 주문' (Order all products), '선택상품 주문' (Order selected products), and '장바구니 비우기' (Clear cart).

## 주문 과정 (결제 포함)

1. 상품/장바구니 주문리스트 불러오기
2. 쿠폰등록/쿠폰/적립금/최종결제금액 계산하기
3. 주문 후 장바구니 내역 삭제하기 / 재고 계산

# 1-1. 상품/장바구니 주문 리스트 불러오기

## 1. 상품 바로 구매

Peony ₩2000  
작약

색상 --선택하세요-- 수량 --선택하세요--

**구매하기**

## 2. 장바구니 구매

**전체상품 주문**    선택상품 주문

```
// 상품 주문하기
@PostMapping(value = "/order")
public String order(@ModelAttribute("ordDto") PaymentDTO paymentDTO, GdsOrd

    선택한 데이터

    return "order";
}

// 장바구니 전체주문
@GetMapping(value = "/cart/order")
public String order(Model model, GdsCartDTO gdsCartDTO) {

    선택한 데이터

    return "order";
}

// 장바구니 선택 주문
@GetMapping(value = "/cart/orderSelect")
public String orderSelect(@RequestParam(value = "num") ArrayList<Integer> c

    선택한 데이터

    return "order";
}
```



## 1-2. 결제 페이지 ( /order)에서 불러오기

### 1. 각각의 Controller에서 정보 넘기기

```
//주문
model.addAttribute("OrdDto", ordDto);

//장바구니 전체주문
List<UserCartDTO> CartList = cartService.getGdsList(userNum);
model.addAttribute("CartList", CartList);

// 장바구니 선택주문
List<UserCartDTO> CartList = cartService.orderSelect(cartNums);
model.addAttribute("CartList", CartList);
```



### 2. 방법1. 방법2. 로 구분하여 리스트 가져오기



```
<c:if test="${not empty orderData }">
    ${orderList 테이블}
</c:if>

<c:if test="${CartList }">
    <c:forEach items="${CartList }" >
        ${cartList 테이블}
    </c:forEach>
</c:if>
```



### 3. 방법1. / 방법2. 존재하는 데이터의 리스트 테이블만 가져오기

주문리스트

사진	상품명	수량	적립금	가격
	Peony:1 [옵션] color : pink	10개	40.0원	2000원
	Peony:1 [옵션] color : orange	20개	40.0원	2000원

총 구매금액 : 60,000원 (적립금 : 1,200 원)

회원님 구매 시 결제 금액의 2%를 적립해 드립니다.

## 2. 쿠폰 / 적립금 <쿠폰>

상품금액		배송비		쿠폰할인		적립금사용		결제 예정금액
20,000원	⊕	무료	⊖	10000원	⊖	1200원	⊖	8800원
쿠폰		VVIP 0.5% 할인 쿠폰	▼	적용				
적립금		1200		사용		보유적립금 : 1200원		

적립금은 100원 단위로 사용하실 수 있습니다. 구매금액의 2%가 적립됩니다.

### 사용 전 <쿠폰 등록>

VVIP

-

-

-

쿠폰번호등록

wwip

16자리 입니다. 예) ABCD-AB12-CD34-12E3

### 등록 후 <쿠폰 리스트>

보유쿠폰 ▼

보유쿠폰

블랙프라이데이 0.3% 할인 쿠폰

회원가입축하 0.15% 할인 쿠폰

VVIP 0.5% 할인 쿠폰

### 등록 후 <쿠폰 사용>

보유쿠폰 ▼

보유쿠폰

회원가입축하 0.15% 할인 쿠폰

VVIP 0.5% 할인 쿠폰

## 2-1. 쿠폰 페이지 사용 전 <쿠폰 등록>

maxlength 속성 사용

```
<input type="text" size=6 maxlength=4 name="cpCode"
      style='text-align:center;text-transform : uppercase;'
- <input type="text" size=6 maxlength=4 name="cpCode2"
      style='text-align:center;text-transform : uppercase;'
- <input type="text" size=6 maxlength=4 name="cpCode3"
      style='text-align:center;text-transform : uppercase;'
- <input type="text" size=6 maxlength=4 name="cpCode4"
      style='text-align:center;text-transform : uppercase;'
```

CSS 설정 : uppercase

< couponDTO >

```
void setCpCode(String cpCode){
    this.cpCode = cpCode.toUpperCase();
}
```

사용자 관점  
: 대문자

실제 입력 값  
: 소문자



The image shows a coupon registration form and a coupon graphic. The coupon graphic is red with the text "COUPON FOR YOU" and a warning: "쿠폰 코드가 일치하지 않거나 이전에 발급된 쿠폰 코드는 등록할 수 없습니다." (Coupon code does not match or coupon code already issued previously cannot be registered). The form has a "VVIP" dropdown, four input fields for the coupon code, and a "쿠폰번호등록" button. Below the input fields, it says "16자리 입니다. 예) ABCD-AB12-CD34-12E3". A diagram shows arrows from the "사용자 관점 : 대문자" (User perspective: uppercase) and "실제 입력 값 : 소문자" (Actual input value: lowercase) boxes pointing to the "vvip" and "VVIP" labels in the form.

<쿠폰 코드 등록>

mgrCoupon 테이블								
mgrCp	cpName	itemScope	cpRate	expDate	cpCode	cpCode2	cpCode3	cpCode4
1111	VVIP	전체상품 적용가능	0.5	2021.12.31	VVIP	VVIP	VVIP	VVIP
1211	블랙프라이데이	전체상품 적용가능	0.3	2021.12.31	AB12	AA11	BB22	12AB
1311	회원가입축하	전체상품 적용가능	0.15	2021.12.31	CD23	CC22	DD33	23CD

1. 일치하는 코드가 있는지 확인



-  -  -  쿠폰번호등록

쿠폰번호는 영문자+숫자 조합이며, 총 16자리 입니다. 예) ABCD-AB12-CD34-12E3

2. 이전에 발급 된적이 있는지 확인  
( useNum과 cpCode로 check)

coupon 테이블			
cpNum	cpCode	userNum	useCp
1	VVIP	1	사용가능
2	AB12	1	사용완료
3	CD23	1	사용완료
4	AB12	2	사용가능
5	CD23	2	사용가능

전체 쿠폰 목록

쿠폰번호	쿠폰이름	적용범위	할인액(률)	사용기간	사용여부
1311	VVIP	전체상품 적용가능	50%	2021.12.31	사용가능
1211	블랙프라이데이	전체상품 적용가능	30%	2020.12.31	사용완료

# <보유 쿠폰 불러오기>

'쿠폰내역' 페이지 ( /couponPage )

쿠폰번호등록

쿠폰번호는 영문자+숫자 조합이며, 총 16자리 입니다. 예) ABCD-AB12-CD34-12E3

## 전체 쿠폰 목록

쿠폰번호	쿠폰이름	적용범위	할인액(률)	사용기간	사용여부
1311	VVIP	전체상품 적용가능	50%	2021.12.31	사용가능
1211	블랙프라이데이	전체상품 적용가능	30%	2020.12.31	사용완료

## '마이페이지' ( /myPage )

### 사용가능 쿠폰

전체보기

쿠폰번호	쿠폰이름	적용범위	할인액(률)	사용기간	사용여부
1311	VVIP	전체상품 적용가능	0.5	2021.12.31	사용가능

## '주문서작성' 페이지 ( /order )

### 주문상품 할인적용

상품금액	배송비	쿠폰할인
40,000원	무료	0원
쿠폰	보유쿠폰	적용
적립금	보유쿠폰	사용
	VVIP 0.5% 할인 쿠폰	

CouponMapper

```
public interface CouponMapper {
    @Insert("insert into coupon(cpNum, cpCode, ... ")
    void regCoupon(CouponDTO couponDTO);

    @Select("SELECT A.cpNum, ... WHERE A.cpCode= B.cpCode
    and A.userNum = #{userNum} and A.useCP='사용가능'")
    List<CouponInfoDTO> cpList(int userNum);

    @Select("SELECT A.cpNum, ... WHERE A.cpCode= B.cpCode
    and A.userNum = #{userNum}")
    List<CouponInfoDTO> allCpList(int userNum);

    @Select("SELECT cpNum from coupon where ... ")
    String existCp(CouponDTO couponDTO);

    @Update("UPDATE coupon SET useCp= '사용 완료' ... ")
    void updateCp(int cpNum);
}
```

cpList

allCpList

## 2-2. 쿠폰 사용

### Controller

```
@PostMapping(value = "/payment")
public String payment(@ModelAttribute("paymentDTO")
    paymentService.insertPayment(paymentDTO);
    int cpNum = paymentDTO.getCpNum();
    couponService.updateCp(cpNum);
    return "redirect:/paymentResult";
}
```

### CouponMapper

```
@Update("UPDATE coupon SET useCp= '사용완료' WHERE cpNum=#{cpNum} ")
void updateCp(int cpNum);
```

useCp 가 '사용가능' → '사용완료' 로 변경

USERNUM	USECP	MGRCP	CPNAME	ITEMSCOPE	CPRATE
1	사용가능	1211	블랙프라이데이	전체상품 적용가능	0.3
1	사용가능	1311	회원가입축하	전체상품 적용가능	0.15
1	사용가능	1111	VVIP	전체상품 적용가능	0.5

USERNUM	USECP	MGRCP	CPNAME	ITEMSCOPE	CPRATE
1	사용완료	1211	블랙프라이데이	전체상품 적용가능	0.3
1	사용가능	1311	회원가입축하	전체상품 적용가능	0.15
1	사용가능	1111	VVIP	전체상품 적용가능	0.5

보유쿠폰 ▼

보유쿠폰

블랙프라이데이 0.3% 할인 쿠폰

회원가입축하 0.15% 할인 쿠폰

VVIP 0.5% 할인 쿠폰



보유쿠폰 ▼

보유쿠폰

회원가입축하 0.15% 할인 쿠폰

VVIP 0.5% 할인 쿠폰

'사용가능' 데이터 : 3개 → 2개

## 2. 쿠폰 / 적립금 <적립금>

상품금액		배송비		쿠폰할인		적립금사용		결제 예정금액
20,000원	⊕	무료	⊖	10000원	⊖	1200원	⊖	8800원
쿠폰	<input type="text" value="VVIP 0.5% 할인 쿠폰"/>		<input type="button" value="적용"/>					
적립금	<input type="text" value="1200"/>		<input type="button" value="사용"/>		보유적립금 : 1200원			
적립금은 100원 단위로 사용하실 수 있습니다. 구매금액의 2%가 적립됩니다.								

[ 보유 적립금 불러오기 ] [ 사용 전 유효성 검사 ] [ 적립금 적용 ] [ 사용 적립금 / 구매 적립금 저장 ]



## <적립금 사용 전 유효성 검사>

1200    **사용**    보유적립금 : 1200원

### 3가지의 유효성 검사

1. 숫자만 입력하기
2. 적립금 100원 단위 사용하기
3. 보유적립금 적게 사용하기

onkeypress="inNumber();

```
function minPoint(){
    var A = document.getElementById('couponnum').value
    var B = document.getElementById('totalPoint').value

    if(A%100 != 0){
        alert("적립금은 100원 단위로 이용하실 수 있습니다.");
    }
    else if(A>B){
        alert("적립금이 부족합니다.");
    }
    else{
        getInputValue(); calCulation(smart); getInputValue();
    }
}
```

## <적립금 사용 / 적립금 적립>

기존 보유 적립금

USERNUM	TOTALPOINT
1	1200

적립금	<input type="text" value="1200"/>	사용	보유적립금 : 1200원
-----	-----------------------------------	----	---------------

구매 완료

적립되는 적립금과  
사용되는 적립금을 따로 저장

적립된 적립금

USERNUM	POINT	USEPOINT
1	176	1200
1	1200	0

← 사용된 적립금

```
@PostMapping(value = "/payment")
public String payment(@ModelAttribute("paymentDTO") PaymentDTO paymentDTO,
    paymentService.insertPayment(paymentDTO);

    int cpNum = paymentDTO.getCpNum();
    couponService.updateCp(cpNum);

    return "redirect:/paymentResult";
}
```

적립금 / 쿠폰 적용 => 최종 결제 (금액)

## <적립금 사용 / 적립금 적립>

```
@Select("SELECT userNum, SUM(point-usePoint) as totalPoint FROM payment where userNum=${userNum} GROUP BY userNum")  
PointDTO totalPoint(int payNum);
```

```
PointDTO totalPoint = paymentService.totalPoint(userNum);  
model.addAttribute("totalPoint", totalPoint);
```

GROUP BY

USERNUM	POINT	USEPOINT
1	( 176	- 1200)
1	( 1200	- 0)

SUM



USERNUM	TOTALPOINT
1	176

적립된 적립금과 사용 적립금을 합쳐서 가져옴

### 3. 주문 후 장바구니 내역 삭제하기 / 재고 계산

주문자동의

☐ 상기 결제정보를 확인하였으며, 구매진행에 동의합니다.

최종 결제금액	<b>40,000</b> 원 (적립예정: 800원) (적립 예정금액과 최종 적립금액은 상이할 수 있습니다. 주문 완료 시 지급되는 적립금을 확인해주시기 바랍니다.)
---------	--

주문하기

주문취소

```
String[] cartCounts = cartCount.split(",");
String[] gdsNums = gdsNum.split(",");

if(!cartNum.equals("fail")) {
    String[] arrCartNums = cartNum.split(",");

    for(String cn : arrCartNums) {
        cartService.deleteOne(Integer.parseInt(cn));
    }
}
```

장바구니 구매 후  
내역 삭제

```
int index = 0;
for(String gn : gdsNums) {
    int gdsStock = gdsService.getGdsStock(gn);
    int selStock = Integer.parseInt(cartCounts[index]);
    int result = gdsStock-selStock;

    gdsService.updateStock(gn,result);
    index++;
}
```

주문 수량 체크 후  
상품 stock → update