민정원 포트폴리오

- 노션으로 연결
- Git으로 연결

목 차

<1차 프로젝트>

OKSUSU

MVC1, JSP를 이용하여 장바구니 및 결제 구현하기

목 차

1차 프로젝트 _ 장바구니 및 결제

1) 사용기술

slack

draw.io

eclipse

Java
JDBC

HTML
CSS
JS

Kakao
Maps API

Google Sheets

Notion

DBeaver

GitHub

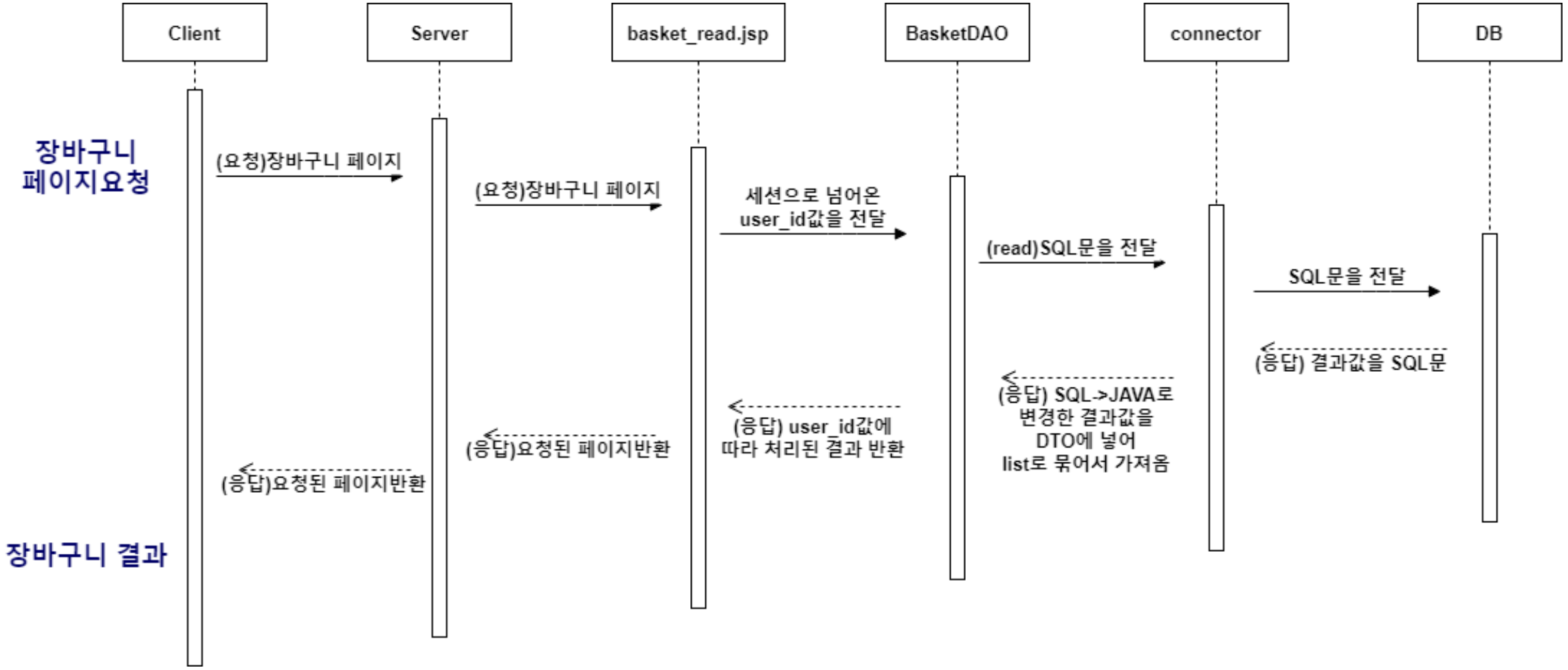
Apache
TOMCAT

Kakao Developers

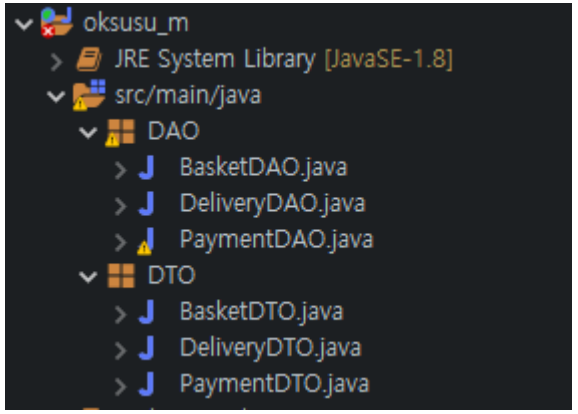
2) 프로젝트 개요

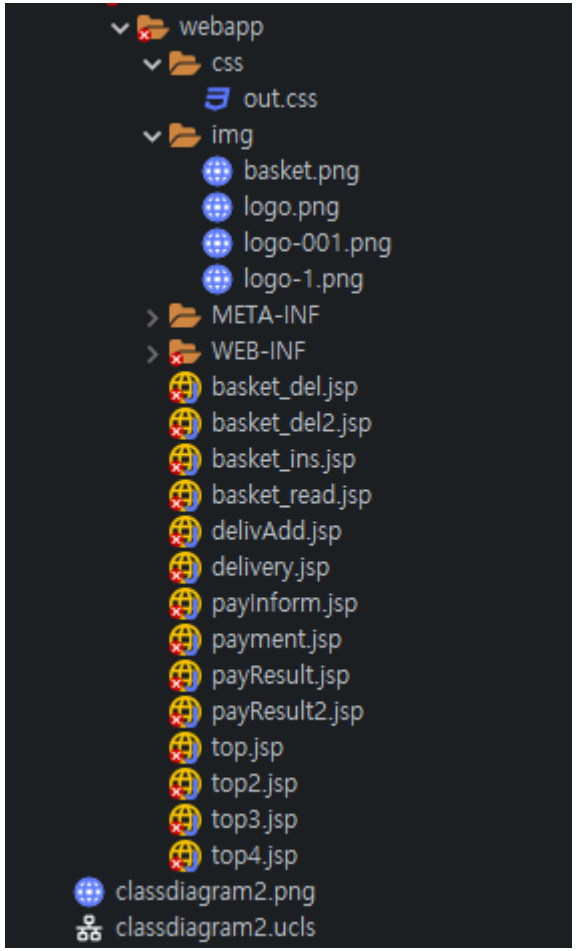
프로젝트명	중고 물건 거래 홈페이지 만들기
진행인원	5명
담당업무	장바구니 및 결제
구현	장바구니 기능과 API활용하여 결제
목표	장바구니를 DB에 저장하고 물품 담기, 삭제(전체/개별)기능 구현할 것 배송지 목록을 회원가입정보와 별도로 DB에 저장하여 관리할 것 장바구니에서 결제를 할 때, 결제금액에 대한 정보를 나타낼 것 물품 상세페이지와 장바구니페이지, 두 페이지에서 결제 할 것 결제한 정보는 DB에 저장하여 관리할 것
개발환경	Windows10
사용도구	Eclipse, MYSQL, Dbeaver, Apache Tomcat, Notion, Slack, Google Spreadsheet, draw.io, ObjectAid, Kakao Map
사용기술	JAVA, CSS, JAVASCRIPT, JSP, MVC1, OPEN API
결과	장바구니와 결제 기능을 목표와 같이 구현함

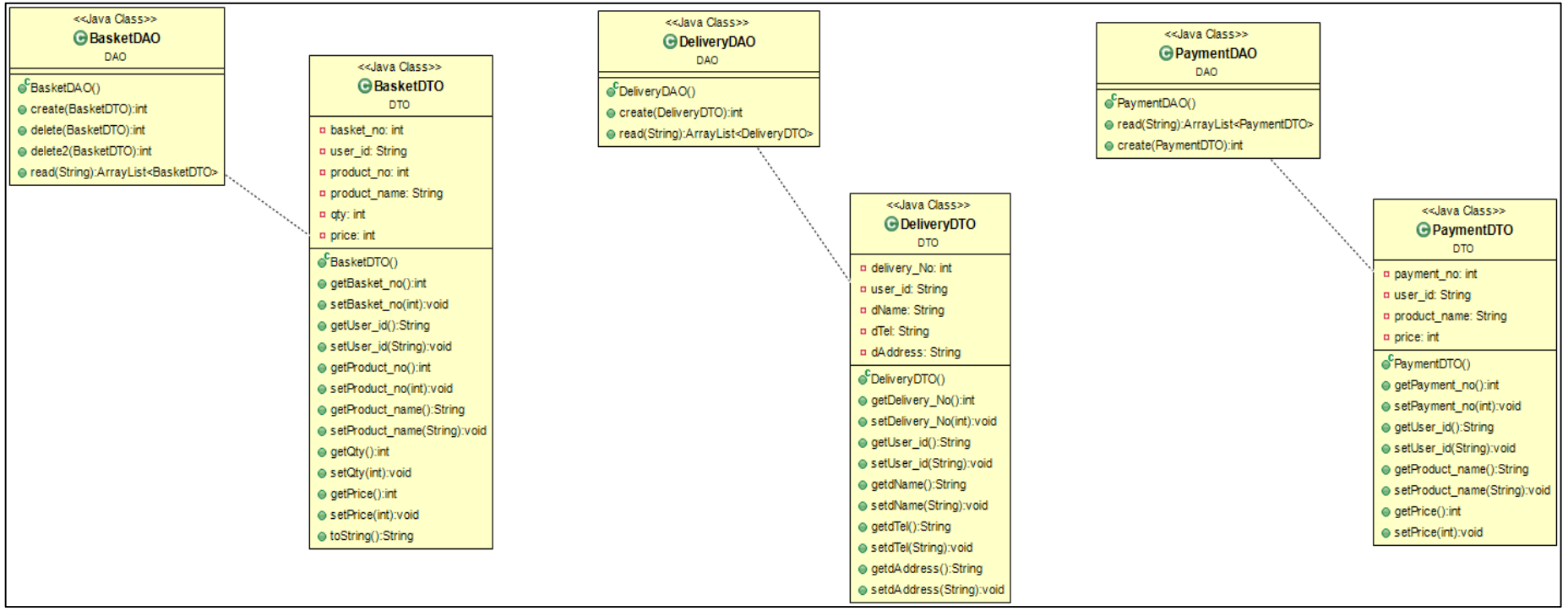
3) 시퀀스 다이어그램


Client
Server
basket_read.jsp
BasketDAO
connector
DB
장바구니
페이지요청
(요청)장바구니 페이지
(요청)장바구니 페이지
세션으로 넘어온
user_id값을 전달
(read)SQL문을 전달
SQL문을 전달
(응답) 결과값을 SQL문
(응답) SQL->JAVA로
변경한 결과값을
DTO에 넣어
list로 묶어서 가져옴
(응답) user_id값에
따라 처리된 결과 반환
(응답)요청된 페이지반환
(응답)요청된 페이지반환
장바구니 결과

4) Tree 및 Class Diagram







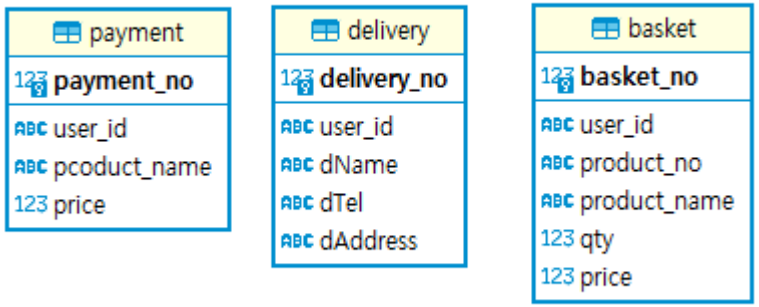
5) ERD 및 제약조건

4조_1차프로젝트

파일 수정 보기 삽입 서식 데이터 도구 부가기능 도움말 몇 초 전에 마지막으로 수정했습니다.

테이블명	basket				작성자	민정원	
테이블설명	구매할 물품을 저장하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	basket_no	int	n	pk	자동증가		
2	user_id	varchar(100)	nn			회원ID	세션으로 가져오기
3	product_no	int	nn			제품번호	건웅 페이지에서 값가져오기
4	product_name	varchar(100)	nn			제품이름	건웅 페이지에서 값가져오기
5	qty	int	nn			수량	건웅 페이지에서 값가져오기
6	price	int	nn			가격	건웅 페이지에서 값가져오기
테이블명	payment				작성자	민정원	
테이블설명	물품을 결제 하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	payment_no	int	nn	pk	자동증가		
2	user_id	varchar(100)	nn			주문회원ID	세션으로 가져오기
3	product_name	varchar(100)	nn			주문상품명	건웅 페이지에서 값가져오기
4	price	int	nn			가격	건웅 페이지에서 값가져오기
테이블명	delivery				작성자	민정원	
테이블설명	배송지 목록을 저장하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	deliveryNo	int	n	pk	자동증가		
2	user_id	varchar(100)	nn			회원ID	세션으로 가져오기
3	dName	varchar(100)	nn			배송받을 이름	셀렉트이용하기
4	dTel	varchar(100)	nn			배송받을 연락처	
5	dAddress	varchar(100)	nn			배송받을 주소	

하영 건웅 정원 정훈 석준



6) 협업툴 사용(Google Sheets)

Google Sheets

4조_1차프로젝트

파일 수정 보기 삽입 서식 데이터 도구 부가기능 도움말 몇 초 전에 마지막으로 수정했습니다.

테이블명	basket				작성자	민정원	
테이블설명	구매할 물품을 저장하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	basket_no	int	n	pk	자동증가		
2	user_id	varchar(100)	nn			회원ID	세션으로 가져오기
3	product_no	int	nn			제품번호	건웅 페이지에서 값가져오기
4	product_name	varchar(100)	nn			제품이름	건웅 페이지에서 값가져오기
5	qty	int	nn			수량	건웅 페이지에서 값가져오기
6	price	int	nn			가격	건웅 페이지에서 값가져오기
테이블명	payment				작성자	민정원	
테이블설명	물품을 결제 하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	payment_no	int	nn	pk	자동증가		
2	user_id	varchar(100)	nn			주문회원ID	세션으로 가져오기
3	product_name	varchar(100)	nn			주문상품명	건웅 페이지에서 값가져오기
4	price	int	nn			가격	건웅 페이지에서 값가져오기
테이블명	delivery				작성자	민정원	
테이블설명	배송지 목록을 저장하는 TABLE						
Num	Column	Type	Null	Key	Default	Comment	
1	deliveryNo	int	n	pk	자동증가		
2	user_id	varchar(100)	nn			회원ID	세션으로 가져오기
3	dName	varchar(100)	nn			배송받을 이름	셀렉트이용하기
4	dTel	varchar(100)	nn			배송받을 연락처	
5	dAddress	varchar(100)	nn			배송받을 주소	

하영 건웅 정원 정훈 석준

6) 협업툴 사용(Notion)

Notion

작업자별 작업

JEONGWON MIN 13

- 페이지디자인마무리 — 8/10(화)
- 결제_주문상품/주문자정보/배송정보 — 8/9(월)
- 메인페이지 — 8/2(월)
- 장바구니_DBread — 8/2(월)
- 상품상세_장바구니담기 기능 — 정원, 건웅 맞춰보기 — 8/5(목)
- 상품상세_장바구니담기 후 read연결 — 8/4(수)
- 장바구니_상품삭제 후 read페이지연결 — 8/4(수)
- 장바구니_상품삭제 후 read페이지연결 — 8/4(수)
- 상품상세_장바구니담기 — 8/3(화)
- 결제_결제하기 — 외부의 카드결제 또는 카카오뱅크 결제와 연동하기 — 8/5(목)
- 장바구니_상품삭제 — 8/3(화)
- 장바구니_배송지변경 — 배송지관련 테이블 따로 빼기 — 8/4(수)
- 장바구니_주문하기 — 8/4(수)
- 기본내용(테이블정의서/워크플로우)

6) 협업툴 사용(slack)

slack

1차 로고 컨펌부탁드립니다~

logo-001.png

1

이하영 오전 11:19
진행시켱!

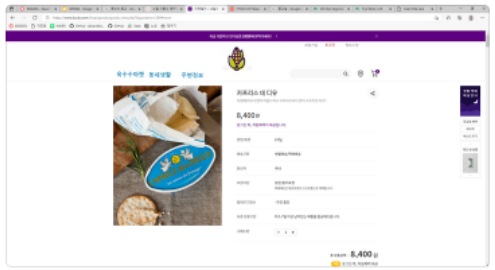
강석준 오전 11:19
저는 너무좋습니다!

JEONGWON MIN 오전 11:20
메인CSS 80%정도 진행되었습니다.
사이즈는 width: 1500px, height: 1500px 입니다~
height는 짧아져도 상관없지만 width는 맞춰서 작업부탁드립니다!

1

JEONGWON MIN 오후 6:05
메인페이지 초안입니다~

이미지 18.png

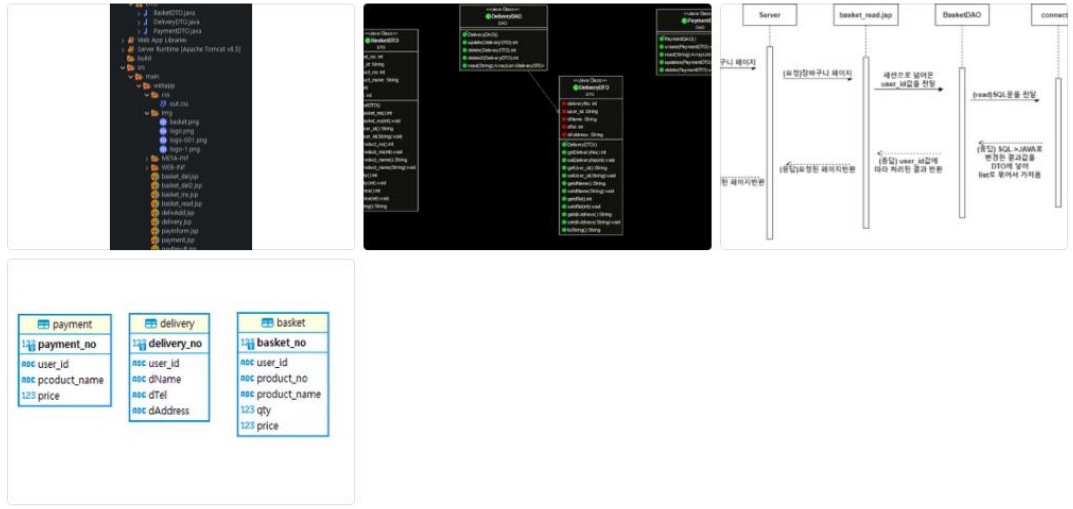


JEONGWON MIN 오후 6:13

1. 가입, 로그인, 정보수정 => 하영 : 스키마 1개 일반 회원or사업자 회원으로 구분하여 가입되도록 / 가능하다면 핸드폰 번호 인증
2. 옥수수마켓 (등록, 수정, 삭제, 검색)=> 건웅 : 전화로 문의 / 결제하기 / 장바구니 버튼 만들어주세요. 결제하기와 장바구니는 정원과 연결 / 시간이 되신다면 찜하기 기능
3. 결제 , 장바구니=> 정원 =>실제 거래가 가능하도록, 메인의 버튼은 장바구니만 결제는 옥수수마켓상품페이지와 장바구니와 연결하기
4. 동네생활 => 정훈 : 가능하다면, 좋아요 개수에 따라 게시판의 글 순서 변경
5. 주변정보(글쓴이 권한부여) => 석준 : 카테고리 2~3개 정도로 /가능하다면, 좋아요 개수에 따라 게시판의 글 순서 변경

(편집됨)

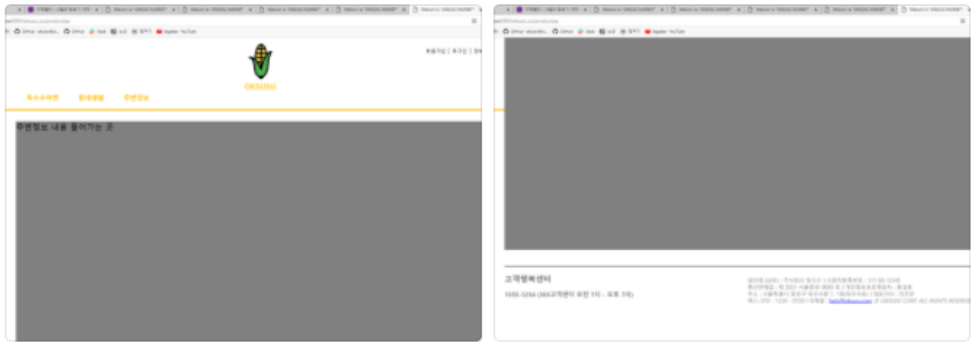
JEONGWON MIN 오후 9:48
4개 파일



---

css 초안

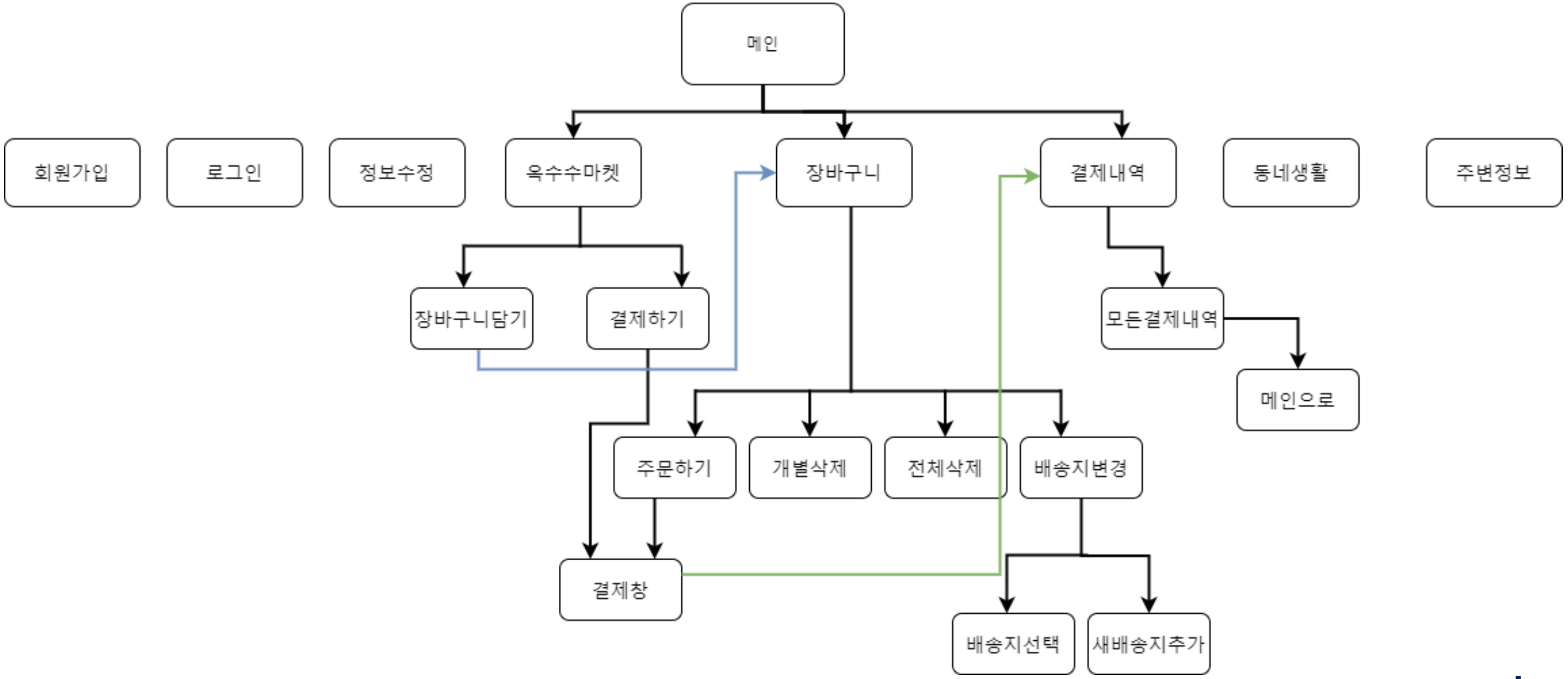
2개 파일



JEONGWON MIN 오전 12:29
<해야할 일>
장바구니페이지 내 및 배송지 read 배송지 변경 수정
장바구니에서 결제할 내용 넘겨주기
결제내역 table에 저장 후 read해오기 결제내역 버튼과 연결
메인페이지 물건리스트 뿌리기
물건 상세페이지에서 결제하기버튼 연결하기?(건웅씨한테 확인)

3개 파일

7) 흐름도


메인
회원가입
로그인
정보수정
옥수수마켓
장바구니
결제내역
동네생활
주변정보
장바구니담기
결제하기
모든결제내역
메인으로
주문하기
개별삭제
전체삭제
배송지변경
결제창
배송지선택
새배송지추가

8) 코드 분석

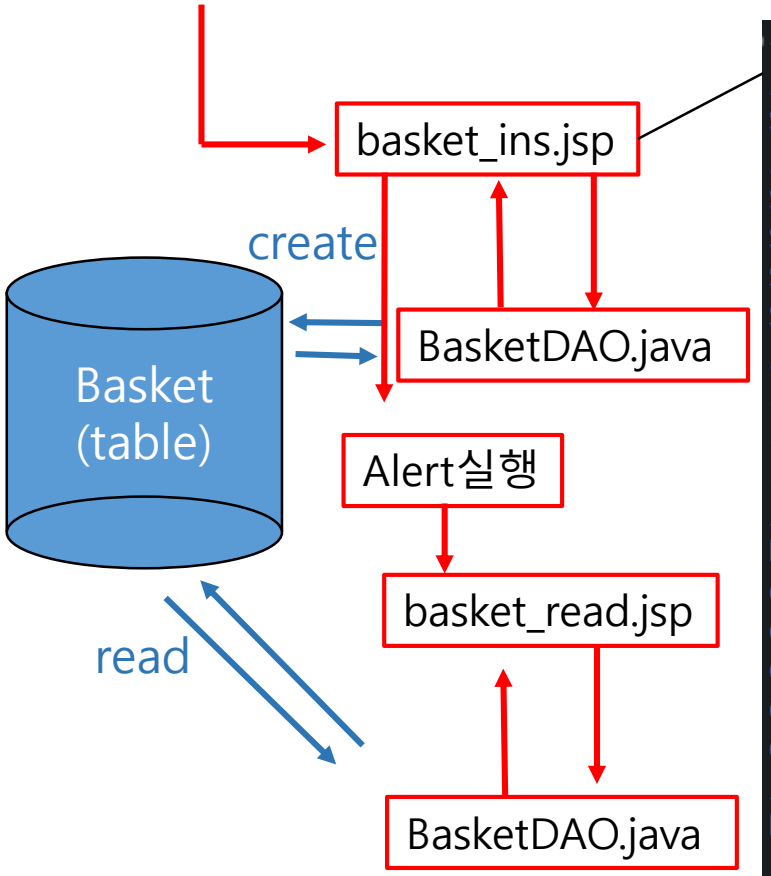
주요기능 : Table CRUD, 회원 ID 세션활용, OPEN API사용, 금액 콤마찍기

- (8-1) 중고물품 상세페이지_장바구니담기
- (8-2) 중고물품 상세페이지_ 바로 결제
- (8-3) 장바구니 목록 읽어 오기
- (8-4) 장바구니 전체 삭제
- (8-5) 장바구니 개별 삭제
- (8-6) 장바구니 물품에 따른 결제금액 변경 및 세 자리마다 콤마 찍어 출력하기
- (8-7) 배송지 목록 읽어오기
- (8-8) 배송지 추가하기 (1),(2)
- (8-9) 우편번호 검색(OPEN API)
- (8-10) 결제(OPEN API)
- (8-11) 결제페이지_값 넘기기

8-1 중고물품 상세 페이지 _ 장바구니 담기

marketView.jsp

```
<button onclick="location.href='basket_ins.jsp?Product_no=<%=dto2.getProduct_no()%>
&Product_name=<%=dto2.getProduct_name()%>&Qty=<%=dto2.getQty()%>&Price=<%=dto2.getPrice()%>'">장바구니</button>
```




```
<%
///세션으로 id 받기
String user_id = (String)session.getAttribute("memberId");
//marketView.jsp에서 값 받아오기
String Product_no = request.getParameter("Product_no");
String Product_name = request.getParameter("Product_name");
String Qty = request.getParameter("Qty");
String Price = request.getParameter("Price");
//int로 변환
int Product_no2 = Integer.parseInt(Product_no);
int Qty2 = Integer.parseInt(Qty);
int Price2 = Integer.parseInt(Price);
//basket_no를 dto에 넣기
BasketDTO dto = new BasketDTO();
dto.setUser_id(user_id);
dto.setProduct_no(Product_no2);
dto.setProduct_name(Product_name);
dto.setQty(Qty2);
dto.setPrice(Price2);
//dao에 delete2를 실행
BasketDAO dao = new BasketDAO();
int result = dao.create(dto);
out.println("<script> alert('장바구니에 추가되었습니다.'); location.href='basket_read.jsp'; </script>");
%>
```

8-2 중고물품 상세 페이지 _ 바로결제

marketView.jsp

```
<script>
function inform() {
    var url = "payInform.jsp?product_name=<%=dto2.getProduct_name()%>&qty=<%=dto2.getQty()%>&price=<%=dto2.getPrice()%>";
    var name = "popup inform";
    var option = "width = 490, height = 550, top = 400, location = no"
    window.open(url, name, option);
}
</script>
<button onclick="inform()" >바로결제</button>
```

payInform.jsp

```
<%  //세션으로 id 받기
    String user_id = (String)session.getAttribute("memberId");
   // marketView.jsp에서 값받아오기
    String product_name = request.getParameter("product_name");
    String qty = request.getParameter("qty");
    String price = request.getParameter("price");
   //int로 변환
    int qty2 = Integer.parseInt(qty);
    int price2 = Integer.parseInt(price);
  //가격에 세자리 콤마찍기
    DecimalFormat dc = new DecimalFormat("###,###,###");
    int total = qty2 * price2;
    String total2 = dc.format(total);
    System.out.print("총액은 "+ total2);
%>
```

```
<h2>주문 내역</h2>
<hr>
<table>
  <tr>
    <td>결제하실 물품은 :</td>
    <td id="td2"><%= product_name %></td>
  </tr>
  <tr>
   <td>결제하실 수량은 :</td>
   <td id="td2"><%= qty %></td>
  </tr>
  <tr>
   <td>결제하실 금액은 :</td>
   <td id="td2"><%= total2 %></td>
   </tr>
</table>
<br><br>
   <hr>
   <div>결제를 원하시면, 아래의 주문하기 버튼을 클릭해주세요!</div>
   <button id="b4" onclick= "location.href = 'payment.jsp?product_name=<%=product_name%>&total=<%=total%>'">주문하기</button>
```

payment.jsp

marketView.jsp에서 버튼 클릭 시, payInform.jsp로 연결되며 이때 여러 개의 값이 전달됨.

주문하기 버튼 클릭시, payment.jsp연결되며 payInform.jsp에 받아온 값을 payment.jsp로 전달해줌.

8-3 장바구니 페이지 _ 장바구니 목록 읽어오기

basket_read.jsp

```jsp
String user_id = (String)session.getAttribute("MemberId");
```

```jsp
//Basket table에서 데이터를 가져오기
BasketDAO dao = new BasketDAO();
ArrayList<BasketDTO> list = dao.read(user_id);
```

```jsp
for (int i = 0; i < list.size(); i++) {

    BasketDTO dto = list.get(i);
    int orderQty = list.get(i).getQty();
    int orderPrice = list.get(i).getPrice();
```

```jsp
<tr>
    <td><%=dto.getProduct_name()%></td>
    <td><%=dto.getQty()%></td>
    <td><%=price2%></td>
```

list에 담긴 여러 개의 dto를 for문을 이용해 하나씩 꺼내 화면에 나타내 준다.

BasketDAO.java

```java
public ArrayList<BasketDTO> read(String user_id) {
    System.out.println("R전달된 user_id는 " + user_id);
    ResultSet rs = null;
    ArrayList<BasketDTO> list = new ArrayList<>();
    try {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("1. connector연결 성공!!!");
        String url = "jdbc:mysql://localhost:3306/oksusu";
        String username = "root";
        String password = "1234";
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("2. oksusu db연결 성공!!!");
        String sql = "select * from basket where user_id= ?";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, user_id);
        System.out.println("3. sql문 생성 성공!!!");
        rs = ps.executeQuery();
        System.out.println("4. sql문 전송 전송");
        while (rs.next()) {
            System.out.println("검색 결과가 있음!");
            BasketDTO dto = new BasketDTO();
            int basket_no = rs.getInt(1);
            String user_id2 = rs.getString(2);
            int product_no = rs.getInt(3);
            String product_name = rs.getString(4);
            int qty = rs.getInt(5);
            int price = rs.getInt(6);
            dto.setBasket_no(basket_no);
            dto.setUser_id(user_id2);
            dto.setProduct_no(product_no);
            dto.setProduct_name(product_name);
            dto.setQty(qty);
            dto.setPrice(price);
            list.add(dto);
        }

    } catch (ClassNotFoundException e) {// 1단계,클래스가 없으면 어떡할래?
        System.out.println("1번에러 >> 드라이버없음!");
        e.printStackTrace();// 에러정보 자세하게 알려주는 코드
    } catch (SQLException e) {// 2-4단계, SQL문과 관련되서 문제있으면 어떡할래?
        System.out.println("2-4번에러 >> DB관련된 처리하다 에러발생!");
        e.printStackTrace();// 에러정보 자세하게 알려주는 코드
    }
    System.out.println(list.size());
    return list;
}
```

8-4 장바구니 페이지 _ 장바구니 전체삭제

basket_read.jsp

```jsp
<!-- 장바구니 삭제 기능 : (연결)basket_del.jsp -->
<form action="basket_del.jsp">
<button id="b1">전체삭제</button>
</form>
```

basket_del.jsp

```jsp
<%
//user_id를 세션으로 받아오기
String user_id = (String)session.getAttribute("user_id");

//user_id를 dto에 넣어서 dao에 delete를 실행
BasketDTO dto = new BasketDTO();
dto.setUser_id(user_id);
BasketDAO dao = new BasketDAO();
int result = dao.delete(dto);

//결과값은 삭제한 row수와 동일 : 0이 아니면 안내창, 0일 경우 안내창
if (result != 0) {
    out.println("<script> alert('전부 삭제했습니다.'); location.href='basket_read.jsp';</script>");
  }
else{
    out.println("<script> alert('삭제할 내역이 없습니다.'); location.href='basket_read.jsp';</script>");
    }
%>
```

DAO에서 delet한 뒤 resul에 retur된 수 = 테이블의 삭제된 row 수

즉, result != 0은 장바구니에 담겼던 물품이 삭제된 것, 안내창: "전부삭제했습니다. "

Result=0 이라면 삭제된 물품이 없다, 안내창: "삭제할 내역이 없습니다. "

8-5 장바구니 페이지 _ 장바구니 개별삭제

basket_read.jsp

```jsp
<td><a href="basket_del2.jsp?basket_no=<%=dto.getBasket_no()%>">
<button id="b2">X</button>
</a></td>
```

basket_del2.jsp

```jsp
<%
//basket_read에서 넘겨준 basket_no를 가져오기
String basket_no =  request.getParameter("basket_no");

//String -> int로 변환
int basket_no2 = Integer.parseInt(basket_no);

//basket_no를 dto에 넣어서 dao에 delete2를 실행
BasketDTO dto = new BasketDTO();
dto.setBasket_no(basket_no2);
BasketDAO dao = new BasketDAO();
int result = dao.delete2(dto);

//안내창
out.println("<script> alert('삭제완료'); location.href='basket_read.jsp';</script>");

%>
```

버튼 클릭 시, basket_del2.jsp로 연결되며 이때 Basket_no값이 전달됨.

basket_del2.jsp에서 Basket_no값을 전달받아 BasketDAO를 실행하며, 이때 삭제완료라는 안내창을 실행.

8-6 장바구니 페이지 _ 장바구니 물품에 따른 결제금액 변경 및 세 자리마다 콤마 찍어 출력하기

```
//결제금액에 사용하기 위한 변수
int sum = 0;
```

```
//숫자 세 자리마다 콤마 찍기
DecimalFormat dc = new DecimalFormat("###,###,###");
```

```
//list에 들어있는 결과데이터를 읽어오기 위한 for문
for (int i = 0; i < list.size(); i++) {
    BasketDTO dto = list.get(i);
    //읽어온 Qty, Price를 연산하기 위해 int로 변환
    int orderQty = list.get(i).getQty();
    int orderPrice = list.get(i).getPrice();

    //Qty과 Price를 곱한 소계를 누적
    sum = sum + (orderQty * orderPrice);
```

```
<%  String sum2 = dc.format(sum); %>
<ol id="ol2">
    <li><%=sum2%> 원</li>
    <li>3,000 원</li>
    <li> <!-- 물건소계 + 배송비 : 콤마찍어주기 -->
        <% int total = sum + 3000;
        String total2 = dc.format(total);%>
        <%=total2%>원
```

```
//Price에 콤마찍어주기
String price2 = dc.format(dto.getPrice());
```

```
<td><%=price2%></td>
```

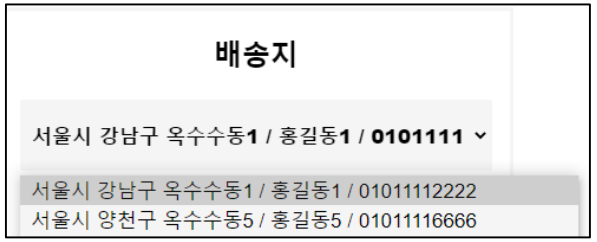
물품명	수량	가격
bag	1	1,000

상품금액 :	1,000 원
배송비 :	3,000 원
결제예정금액 :	4,000원

물품명	수량	가격
bag	2	500

상품금액 :	1,000 원
배송비 :	3,000 원
결제예정금액 :	4,000원

8-7 장바구니 페이지 _ 배송지 목록 읽어오기



basket_read.jsp

```jsp
String user_id = (String)session.getAttribute("MemberId");
```

```jsp
//Delivery table에서 데이터를 가져오기
DeliveryDAO delDao = new DeliveryDAO();
ArrayList<DeliveryDTO> delList = delDao.read(user_id);
```

```jsp
<!-- Delivery table에서 데이터를 가져오기 -->
<select>
    <%
    for (DeliveryDTO dto : delList) {
    %>
    <option value="<%=dto.getDelivery_No()%>">
        <%=dto.getdAddress()%> /
        <%=dto.getdName()%> /
        <%=dto.getdTel()%>
```

list에 담긴 여러 개의 dto를 for문을 이용해 하나씩 꺼내 화면에 셀렉트로 나타내 준다.

DliveryDAO.java

```java
public ArrayList<DeliveryDTO> read(String user_id) {
    System.out.println("배송지Read 전달된 user_id는 " + user_id);
    ResultSet rs = null;
    ArrayList<DeliveryDTO> list = new ArrayList<>();
    try {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("1. connector연결 성공!!!");
        String url = "jdbc:mysql://localhost:3306/oksusu";
        String username = "root";
        String password = "1234";
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("2. oksusu db연결 성공!!!");
        String sql = "select delivery_no, user_id, dName, dTel, dAddress from delivery where user_id=?;";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, user_id);
        System.out.println("3. sql문 생성 성공!!!");
        rs = ps.executeQuery();
        System.out.println("4. sql문 전송 전송");
        while (rs.next()) {
            System.out.println("검색 결과가 있음!");
            DeliveryDTO dto = new DeliveryDTO();
            int delivery_no = rs.getInt(1);
            String user_id2 = rs.getString(2);
            String dName = rs.getString(3);
            String dTel = rs.getString(4);
            String dAddres = rs.getString(5);
            //int dDefalt2 = rs.getInt(6);
            dto.setDelivery_No(delivery_no);
            dto.setUser_id(user_id2);
            dto.setdName(dName);
            dto.setdTel(dTel);
            dto.setdAddress(dAddres);
            //dto.setdDefalt(dDefalt2);
            list.add(dto);
```

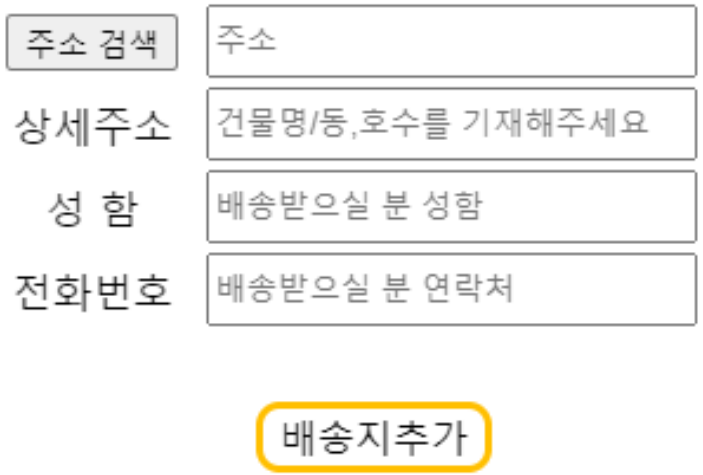
8-8 장바구니 페이지 _ 배송지 추가하기(1)

basket_read.jsp

```
//배송지추가 버튼 연결 : 입력창에 입력된 데이터를 delivAdd.jsp로 넘기기
function dAdd() {
    dNameValue = document.getElementById("dName").value
    dTelValue = document.getElementById("dTel").value
    dAdressValue = document.getElementById("sample5_address").value + " "
            + document.getElementById("sample6_address").value;
    alert(dNameValue + ", " + dTelValue + ", " + dAdressValue);
    location.href = "delivAdd.jsp?dName=" + dNameValue + "&dTel="
            + dTelValue + "&dAddress=" + dAdressValue
```

delivAdd.jsp

버튼 클릭 시, dAdd()가 실행
-> delivAdd.jsp로 연결되며, 값(dName, dTel, dAdress)이 전달




```
<!-- 입력받은 데이터를 Delivery table에 insert -->
<table id="tb2">
    <tr id="tb2r">
        <td id="tb2d"><input type="button"
            onclick="sample5_execDaumPostcode()" value="주소 검색"></td>
        <td id="tb2d2"><input type="text" id="sample5_address"
            placeholder="주소" class="in"></td>
    </tr>
    <tr id="tb2r">
        <td id="tb2d">상세주소</td>
        <td id="tb2d2"><input class="in" type="text"
            id="sample6_address" placeholder="건물명/동,호수를 기재해주세요"></td>
    </tr>
    <tr id="tb2r">
        <td id="tb2d">성 함</td>
        <td id="tb2d2"><input class="in" type="text" id="dName"
            placeholder="배송받으실 분 성함"></td>
    </tr>
    <tr id="tb2r">
        <td id="tb2d">전화번호</td>
        <td id="tb2d2"><input class="in" type="text" id="dTel"
            placeholder="배송받으실 분 연락처"></td>
    </tr>
</table>
<button id="b3" text-align="center" onclick="dAdd()">배송지추가</button>
```

8-8 장바구니 페이지 _ 배송지 추가하기(2)

basket_read.jsp에서 전달된 값을 delivAdd.jsp로 받아 DeliveryDAO.java를 실행.

DAO에서 creadte한 뒤 resul에 retur된 수 =테이블의 삽입된 row수
즉, result != 0=>배송지추가완료, 안내창: "배송지가 추가되었습니다. "
Else=>배송지추가실패 안내창 : "배송지가 추가되지 않았습니다. "

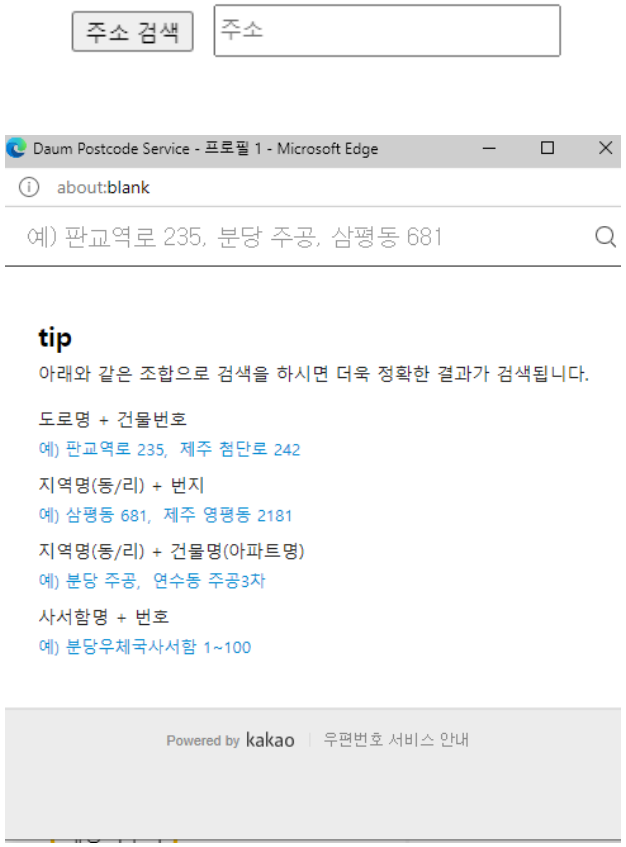
delivAdd.jsp

```jsp
<%
String user_id = (String)session.getAttribute("user_id");
String dName = request.getParameter("dName");
String dTel = request.getParameter("dTel");
String dAddress = request.getParameter("dAddress");

DeliveryDTO dto = new DeliveryDTO();
dto.setUser_id(user_id);
dto.setdName(dName);
dto.setdTel(dTel);
dto.setdAddress(dAddress);

DeliveryDAO dao = new DeliveryDAO();
int result = dao.create(dto);
if (result != 0) {
    out.println("<script> alert('배송지가 추가되었습니다.'); location.href='basket_read.jsp';</script>");
} else {
    out.println("<script> alert('배송지가 추가되지 않았습니다.'); location.href='basket_read.jsp';</script>");
}
%>
```

DeliveryDAO.java

```java
public int create (DeliveryDTO dto) {
    System.out.println("배송지추가>>전달된 user_id는 " + dto.getUser_id());
    System.out.println("배송지추가>>전달된 dName는 " + dto.getdName());
    System.out.println("배송지추가>>전달된 dTel는 " + dto.getdTel());
    System.out.println("배송지추가>>전달된 dAeeress는 " + dto.getdAddress());
    int result=0;
    try {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("1. connector연결 성공!!!");
        String url = "jdbc:mysql://localhost:3306/oksusu?useUnicode=true&characterEncoding=utf8";
        String username = "root";
        String password = "1234";
        Connection con = DriverManager.getConnection(url, username, password);
        System.out.println("2. oksusu db연결 성공!!!");
        String sql = "insert into delivery(user_id, dName, dTel, dAddress) values (?, ?, ?, ?);";
        PreparedStatement ps = con.prepareStatement(sql);
        ps.setString(1, dto.getUser_id());
        ps.setString(2, dto.getdName());
        ps.setString(3, dto.getdTel());
        ps.setString(4, dto.getdAddress());
        System.out.println("3. sql문 생성 성공!!!");
        result = ps.executeUpdate();
        System.out.println("4. sql문 전송 전송");
        System.out.println(result);
    } catch (ClassNotFoundException e) {// 1단계,클래스가 없으면 어떡할래?
        System.out.println("1번에러 >> 드라이버없음!");
    } catch (SQLException e) {// 2-4단계, SQL문과 관련되서 문제있으면 어떡할래?
        System.out.println("2-4번에러 >> DB관련된 처리하다 에러발생!");
        e.printStackTrace();
    }
    return result;
```

8-9 장바구니 페이지 _ 우편번호 검색 (OPEN API사용)



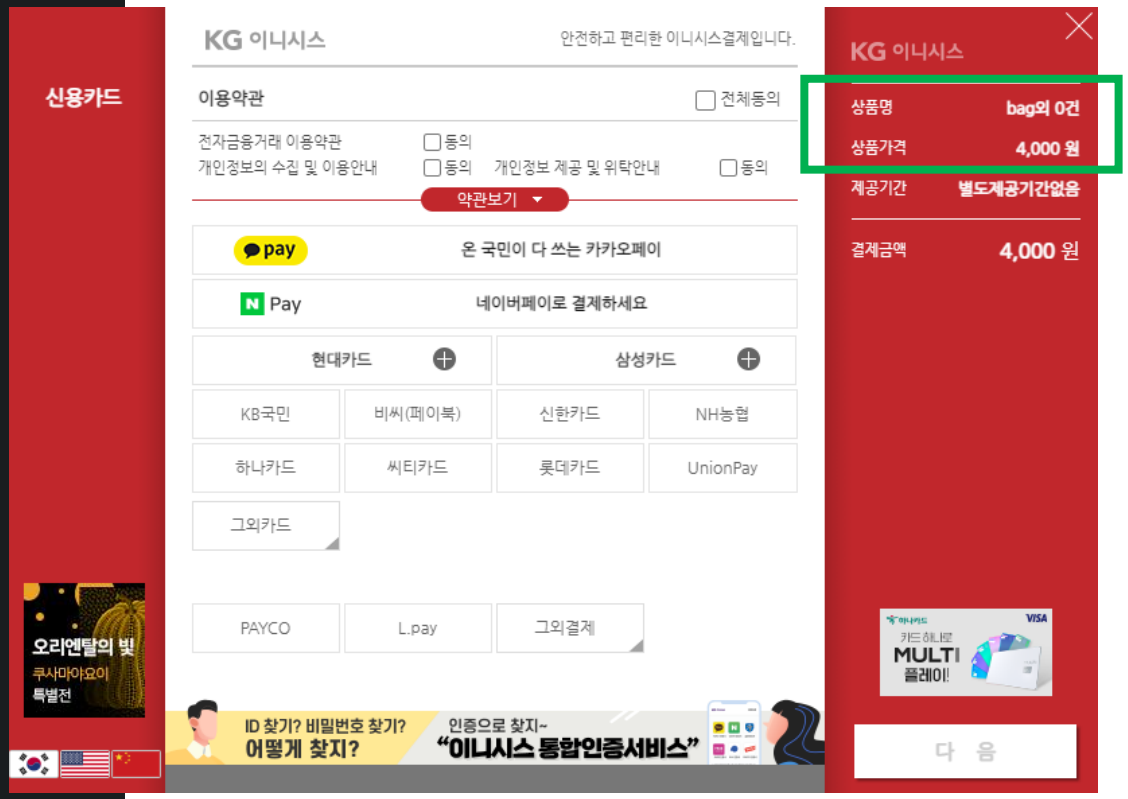

```
<script src="//t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
<script src="//dapi.kakao.com/v2/maps/sdk.js?appkey=0518358d739a04cf7e091d4c708fe499&libraries=services"></script>
<script>
    //다음지도api사용
    //주소-좌표 변환 객체를 생성
    var geocoder = new daum.maps.services.Geocoder();

    function sample5_execDaumPostcode() {
        new daum.Postcode({
            oncomplete : function(data) {
                var addr = data.address; // 최종 주소 변수
                // 주소 정보를 해당 필드에 넣는다.
                document.getElementById("sample5_address").value = addr;
                // 주소로 상세 정보를 검색
                geocoder.addressSearch(data.address, function(results, status) {
                    // 정상적으로 검색이 완료됐으면
                    if (status === daum.maps.services.Status.OK) {
                        var result = results[0]; //첫번째 결과의 값을 활용
                    }
                });
            }
        }).open();
    }
```

```
<table id="tb2">
    <tr id="tb2r">
        <td id="tb2d"><input type="button"
            onclick="sample5_execDaumPostcode()" value="주소 검색"></td>
        <td id="tb2d2"><input type="text" id="sample5_address"
            placeholder="주소" class="in"></td>
    </tr>
```

8-10 결제 페이지 _ 결제 (OPEN API사용)

```
<script type="text/javascript"
    src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
 <!-- iamport.payment.js -->
<script type="text/javascript"
    src="https://cdn.iamport.kr/js/iamport.payment-1.1.5.js"></script>

<script type="text/javascript">
        var IMP = window.IMP; // 생략가능
        IMP.init('iamport'); // 'iamport' 대신 부여받은 "가맹점 식별코드"를 사용
        //IMP.request_pay({json}), function(x){})
        IMP.request_pay({
        pg : 'inicis', // version 1.1.0부터 지원.
        pay_method : 'card',
        merchant_uid : 'merchant_' + new Date().getTime(),
        name : '<%= product_name %>',
        amount : <%= total %>,
        buyer_email : 'iamport@siot.do',
        buyer_name : '<%= user_id %>',
        buyer_tel : '010-1234-5678',
        buyer_addr : '서울특별시 강남구 삼성동',
        buyer_postcode : '123-456',
        m_redirect_url : 'www.yourdomain.com/payments/complete'
        }, function(rsp) {
            if ( rsp.success ) {
                var msg = '결제가 완료되었습니다.';
                msg += '고유ID : ' + rsp.imp_uid;
                msg += '상점 거래ID : ' + rsp.merchant_uid;
                msg += '결제 금액 : ' + rsp.paid_amount;
                msg += '카드 승인번호 : ' + rsp.apply_num;
                console.log(msg)
                alert(msg)
                location.href = "payResult2.jsp?product_name=<%= product_name %>&total=<%= total%>"
            } else {
                var msg = '결제에 실패하였습니다.';
                msg += '에러내용 : ' + rsp.error_msg;
                alert(msg)
                location.href = "basket_read.jsp"
            }
        });
```

(30페이지 참고)



8-11 결제 페이지 _ 값 넘기기

basket_read.jsp

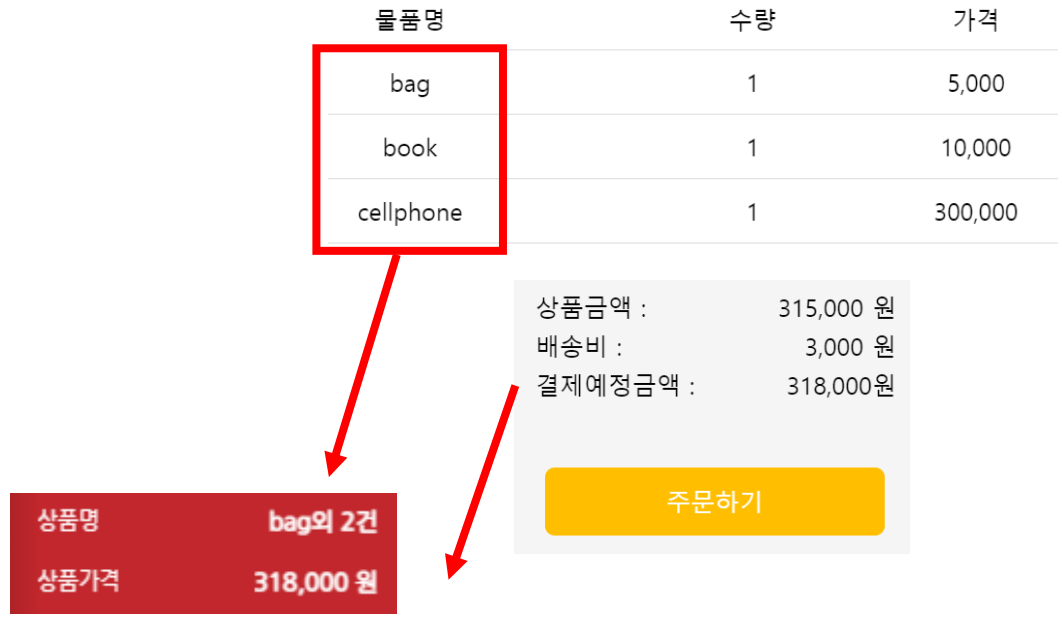
```
//payment.jsp로 결제정보를 넘길 때 사용하기 위한 변수
String result = null;
```

```
//list에 들어있는 결과데이터를 읽어오기 위한 for문
for (int i = 0; i < list.size(); i++) {
```

```
//결제정보에 넣기 위해 첫번째 물품명을 넣어줌
result = list.get(0).getProduct_name();
```

```
<% //result2에 물품명 뒤로 들어갈 문구를 넣어줌
} String result2 = result + "외 " + (list.size() - 1) + "건";
%>
```

```
<!-- 주문하기 : 결제창띄우기, payment.jsp로 결제정보 넘기기 -->
<button id="b4" onclick="location.href = 'payment.jsp?product_name=<%=result2%>&total=<%=total%>'">주문하기</button>
```



payment.jsp

```
String product_name = request.getParameter("product_name");
String total = request.getParameter("total");
```

```
name : '<%= product_name %>',
amount : <%= total %>,
```

필요한 것
(1) 첫번째 list의 물품명
(2) list의 개수
For문 앞에서 result 변수선언하여 for