

- 장바구니에 담긴 물건을 주문하고 나서 session이 종료되는 것을 막기 위해서 thankCustomer.jsp 의 session.invalidate()를 주석처리 해 주었다.

```
49 /*****  
50 //session.invalidate();  
51 PreparedStatement pstmt = null;  
52 String sql = "DELETE FROM cart";  
53  
54 pstmt = conn.prepareStatement(sql);  
55 pstmt.executeUpdate();  
56  
57 if (pstmt != null)  
58     pstmt.close();  
59 if (conn != null)  
60     conn.close();  
61 |  
62 *****/
```

#### <thankCustomer.jsp>

- 장바구니에서 삭제하기 버튼을 눌렀을 때 session이 종료되는 것을 막기 위해서 deleteCart.jsp의 session.invalidate()를 주석처리 해 주었다.

```
12 /*****  
13 //session.invalidate();  
14 PreparedStatement pstmt = null;  
15 String sql = "DELETE FROM cart";  
16  
17 pstmt = conn.prepareStatement(sql);  
18 pstmt.executeUpdate();  
19  
20 if (pstmt != null)  
21     pstmt.close();  
22 if (conn != null)  
23     conn.close();  
24 *****/
```

#### <deleteCart.jsp>

- 장바구니 페이지와 주문 처리 페이지에 사용할 DB table 'cart'을 생성해 주었다.

```
1 USE WebMarketDB;  
2  
3 create table cart (  
4     cartId varchar(100) not null,  
5     userId varchar(10) not null,  
6     productId varchar(100) not null,  
7     productName varchar(100) not null,  
8     price INTEGER  
9 ) default CHARSET=utf8;
```

cartId는 로그인 시 생성된 sessionId, userId는 로그인 아이디, productId는 물품 ID, productName은 물품 이름, price는 물품의 가격을 의미한다.

#### <cart.sql>

- 장바구니 페이지에 데이터베이스를 연동해 주었다.

```

1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.util.ArrayList"%>
3 <%@ page import="dto.Product"%>
4 <%@ page import="dao.ProductRepository"%>
5 <%@ include file="dbconn.jsp" %>

```

데이터베이스에 연동해 주기 위해 addCart.jsp에 dbconn.jsp를 include 해주었다.

```

31 /*
32  *****/
33
34 PreparedStatement pstmt = null;
35 String cartId = session.getId();
36 String userId = (String)session.getAttribute("sessionId");
37 String sql = "insert into cart values(?,?,?,?)";
38
39 pstmt = conn.prepareStatement(sql);
40 pstmt.setString(1, cartId);
41 pstmt.setString(2, userId);
42 pstmt.setString(3, id);
43 pstmt.setString(4, product.getPname());
44 pstmt.setInt(5, product.getUnitPrice());
45 pstmt.executeUpdate();
46
47 if (pstmt != null)
48     pstmt.close();
49 if (conn != null)
50     conn.close();
51
52 /*****/
53

```

기존의 addCart.jsp 에서 새로운 session을 생성하여 장바구니를 만들어 물품을 추가하는 것을 DB table을 이용해 sql문을 이용하여 insert 하는 방식으로 수정하였다. PreparedStatement 객체를 생성하여 sql문을 실행해서 주문하고자 하는 물품을 DB table에 삽입해 주었다.

로그인 시 생성된 sessionId인 cartId를 session.getId()를 사용하여 얻어오고, 로그인 아이디인 userId를 session.getAttribute("sessionId")를 이용하여 얻고, 제품의 아이디인 id는 기존 코드에 존재하는 request.getParameter("id")를 이용해 얻고, 제품 이름을 product.getPname()을 이용해 가져오고, 제품 가격을 얻기 위해 product.getUnitPrice()를 이용하였다.

**<addCart.jsp>**

```

1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.util.ArrayList"%>
3 <%@ page import="dto.Product"%>
4 <%@ page import="dao.ProductRepository"%>
5 <%@ include file="dbconn.jsp" %>

```

데이터베이스에 연동해 주기 위해 cart.jsp에 dbconn.jsp를 include 해주었다.

```

40 <th>비교</th>
41 </tr>
42 <%
43 /*****
44 Statement stmt = conn.createStatement();
45 String sql = "SELECT * , COUNT(productId) as cnt , SUM(price) as total FROM cart group by productId";
46 ResultSet rs = stmt.executeQuery(sql);
47 int sum = 0;
48
49 %>
50 <%
51 while(rs.next()){
52     sum += rs.getInt("total");
53 %>
54
55 <tr>
56 <td><%=rs.getString("productId")%> - <%=rs.getString("productName") %></td>
57 <td><%=rs.getInt("price") %></td>
58 <td><%=rs.getInt("cnt") %>
59 <td><%=rs.getInt("total")%></td>
60 <td><a href="/removeCart.jsp?id=<%=rs.getString("productId") %>" class = "badge badge-danger"> 삭제</a></td>
61 </tr>
62
63
64 <%
65 }
66 if (stmt != null)
67     stmt.close();
68 if (conn != null)
69     conn.close();
70 /*****/
71 %>
72 <tr>
73 <th></th>
74 <th></th>
75 <th>총액</th>
76 <th><%=sum%></th>
77 <th></th>

```

생성 전 addCart.jsp에서 생성한 session을 얻어와 상품 리스트를 하나씩 출력하던 코드를 수정한 addCart.jsp에서 DB Table cart에 입력한 제품 정보를 읽어오는 코드로 수정하였다.

Statement 객체를 생성하여 모든 Column의 값과 제품의 ID의 중복되는 숫자를 세는 cnt Column과 각 제품의 총 가격인 total Column 값을 가져오는 sql 문을 실행한다. 그 후 반복문에서 sql 실행문의 결과 값인 rs를 사용하여 모든 제품의 id, 이름, 가격, 수량, 소계를 화면에 표시해준다.

**<cart.jsp>**

```

1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.util.ArrayList"%>
3 <%@ page import="dto.Product"%>
4 <%@ page import="dao.ProductRepository"%>
5 <%@ include file="dbconn.jsp" %>

```

데이터베이스에 연동해 주기 위해 removeCart.jsp에 dbconn.jsp를 include 해주었다.

```

17
20 /*****
21     PreparedStatement pstmt = null;
22     String sql = "DELETE FROM cart where productId= ?";
23     pstmt = conn.prepareStatement(sql);
24     pstmt.setString(1, id);
25     pstmt.executeUpdate();
26
27     if (pstmt != null)
28         pstmt.close();
29     if (conn != null)
30         conn.close();
31 /*****
32     response.sendRedirect("cart.jsp");
33 %>

```

장바구니에 존재하는 비고란의 삭제 버튼을 누르면 해당 제품이 장바구니에서 삭제되는 코드이다. 수정 전 removeCart.jsp에서는 addCart.jsp에서 생성된 session을 불러와서 상품 리스트 하나씩 가져온 다음 해당되는 제품을 리스트에서 제거하는 방식으로 구현되어 있다. 수정 후 removeCart.jsp에서는 DB table cart에서 해당 제품의 id를 가진 데이터를 삭제해주는 방식으로 구현되어 있다. PreparedStatement 객체를 생성한 후 해당 제품의 id를 가진 데이터를 삭제해주는 sql 문을 실행하였다.

**<removeCart.jsp>**

```

1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="dto.Product"%>
3 <%@ page import="dao.ProductRepository"%>
4 <%@ include file="dbconn.jsp" %>

```

데이터베이스에 연동해 주기 위해 deleteCart.jsp에 dbconn.jsp를 include 해주었다.

```

12 /*****
13     //session.invalidate();
14     PreparedStatement pstmt = null;
15     String sql = "DELETE FROM cart";
16
17     pstmt = conn.prepareStatement(sql);
18     pstmt.executeUpdate();
19
20     if (pstmt != null)
21         pstmt.close();
22     if (conn != null)
23         conn.close();
24 /*****

```

장바구니 전체를 지우는 코드이다. 수정 전 deleteCart.jsp에서는 addCart.jsp에서 생성된 세션을 invalidate하여 장바구니를 삭제하였다. 수정 후 deleteCart.jsp에서는 DB table cart의 데이터 값을 삭제하는 방식으로 구현되어 있다. PreparedStatement 객체를 생성한 후 cart 테이블의 모든 데이터 값을 삭제해주는 sql문을 실행하였다.

**<deleteCart.jsp>**



- 주문 처리 페이지를 데이터베이스에 연동하였다.

```
1 ge contentType="text/html; charset=utf-8"%>
2 ge import="java.util.ArrayList"%>
3 ge import="java.net.URLDecoder"%>
4 ge import="dto.Product"%>
5 ge import="dao.ProductRepository"%>
6 clude file="dbconn.jsp" %>
```

데이터베이스에 연동해 주기 위해 orderConfirmation.jsp에 dbconn.jsp를 include 해주었다.

```
76 /*****
77 Statement stmt = conn.createStatement();
78 String sql = "SELECT * , COUNT(productId) as cnt , SUM(price) as total FROM cart group by productId";
79 ResultSet rs = stmt.executeQuery(sql);
80 int sum = 0;
81 %>
82 <%
83
84 while(rs.next()){
85     sum += rs.getInt("total");
86 %>
87 <tr>
88     <td class="text-center"><em><%=rs.getString("productName")%> </em></td>
89     <td class="text-center"><%=rs.getInt("cnt")%></td>
90     <td class="text-center"><%=rs.getInt("price")%>원</td>
91     <td class="text-center"><%=rs.getInt("total")%>원</td>
92 </tr>
93
94 <%
95     }
96     if (stmt != null)
97         stmt.close();
98     if (conn != null)
99         conn.close();
100 |
101 /*****
102 %>
```

주문 내역을 확인하는 코드이다. 수정 전 orderConfirmation.jsp에서는 장바구니 생성 시 만들어진 session을 불러와 상품 리스트를 하나씩 출력해주었다. 수정 후 orderConfirmation.jsp에서는 DB table cart에 입력되어 있는 데이터 값을 반복문을 사용하여 읽어온다. Statement 객체를 생성하여 모든 Column의 값과 제품의 ID의 중복되는 숫자를 세는 cnt Column과 각 제품의 총 가격인 total Column 값을 가져오는 sql 문을 실행한다. 그 후 반복문에서 sql 실행문의 결과 값인 rs를 사용하여 모든 제품의 이름, 가격, 수량, 소계 그리고 총액을 화면에 표시해준다. 또한 배송 주소와 성명, 우편 번호, 주소 등을 표시해준다.

**<orderConfirmation.jsp>**

```
1 <%@ page contentType="text/html; charset=utf-8"%>
2 <%@ page import="java.net.URLDecoder"%>
3 <%@ include file="dbconn.jsp" %>
```

데이터베이스에 연동해 주기 위해 thankCustomer.jsp에 dbconn.jsp를 include 해주었다.

```
4 /*****
50 //session.invalidate();
51 PreparedStatement pstmt = null;
52 String sql = "DELETE FROM cart";
53
54 pstmt = conn.prepareStatement(sql);
55 pstmt.executeUpdate();
56
57 if (pstmt != null)
58     pstmt.close();
59 if (conn != null)
60     conn.close();
61 |
62 /*****
```

주문 완료 후 장바구니에 주문한 제품들을 삭제해 주는 코드를 thankCustomer.jsp에 추가해 주었다.

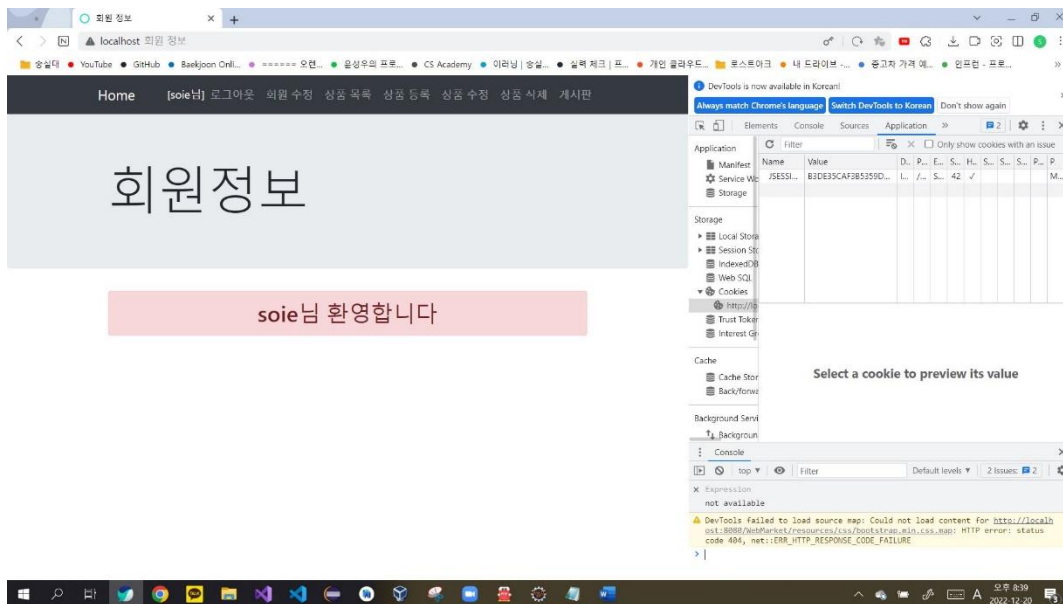
**<thankCustomer.jsp>**

## ● 추가로 수정한 것

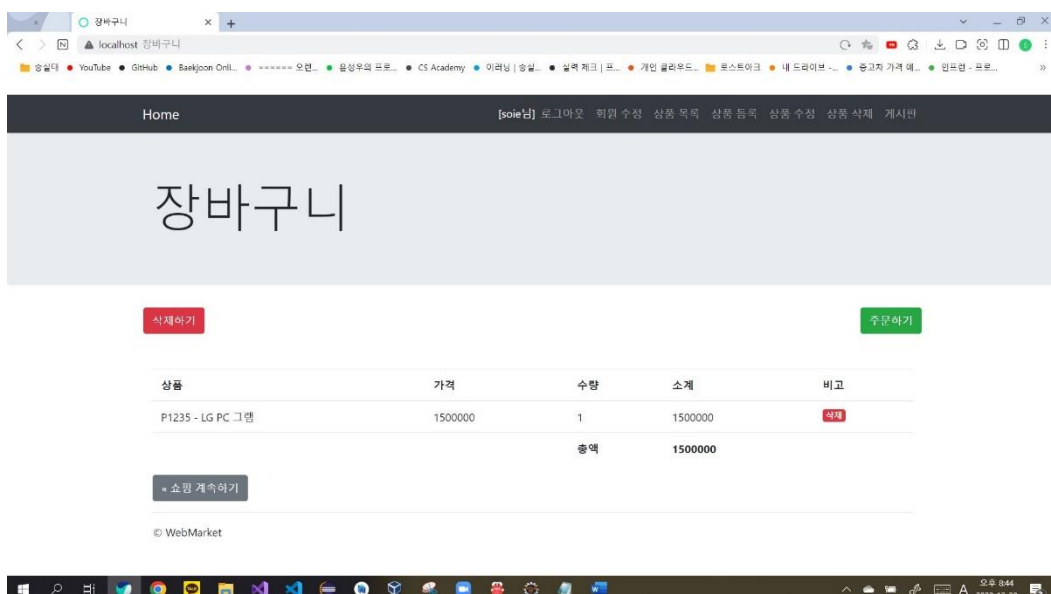
```
<c:choose>
  <c:when test="${empty sessionId}">
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">로그인 </a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/addMember.jsp"/>">회원 가입</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">상품 목록</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">상품 등록</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">상품 수정</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">상품 삭제</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LoginMember.jsp"/>">게시판</a></li>
  </c:when>
  <c:otherwise>
    <li style="padding-top: 7px; color: white"><%=sessionId%>님</li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/LogoutMember.jsp"/>">로그아웃 </a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/member/updateMember.jsp"/>">회원 수정</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/products.jsp"/>">상품 목록</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/addProduct.jsp"/>">상품 등록</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/editProduct.jsp?edit=update"/>">상품 수정</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/editProduct.jsp?edit=delete"/>">상품 삭제</a></li>
    <li class="nav-item"><a class="nav-link" href="<c:url value="/BoardListAction.do?pageNum=1"/>">게시판</a></li>
  </c:otherwise>
</c:choose>
```

로그인 해야 다른 메뉴에 접근할 수 있도록 menu.jsp를 수정하였다.

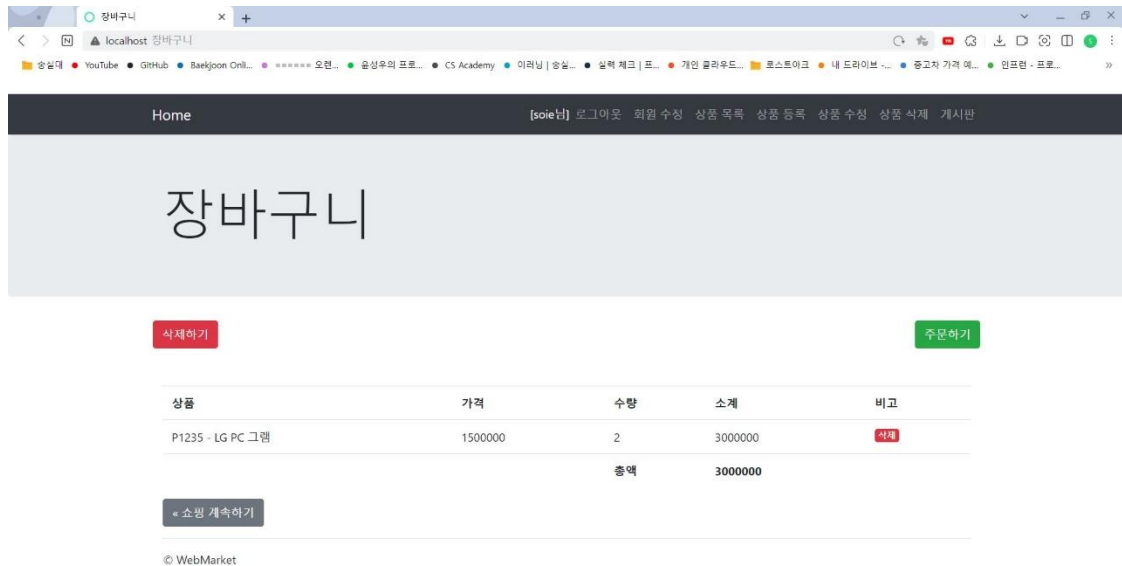
## ● 수정한 WebMarket 실행 결과



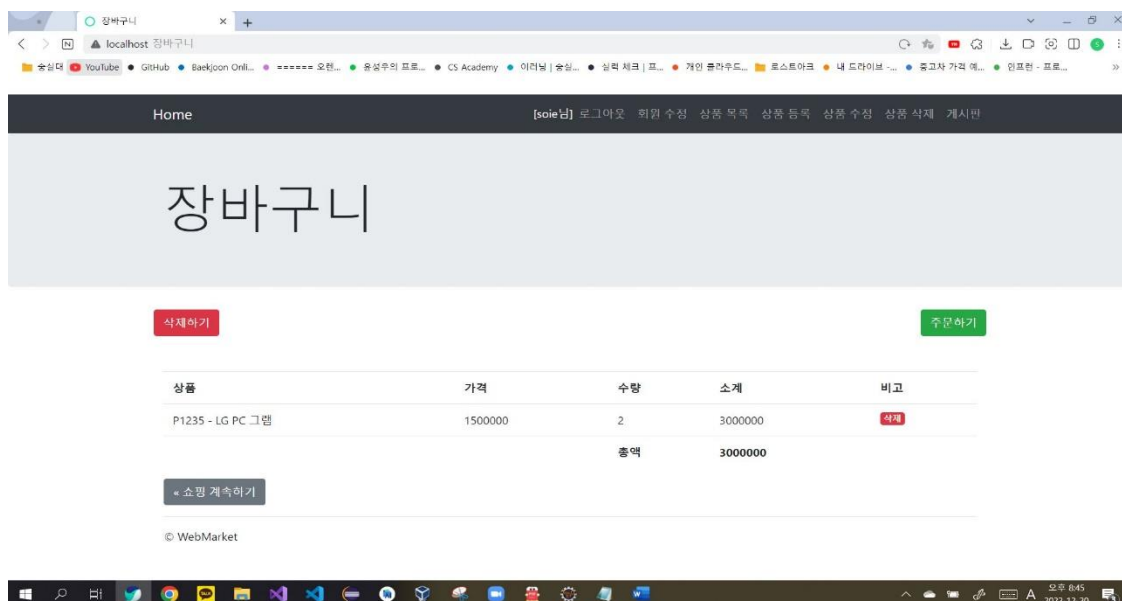
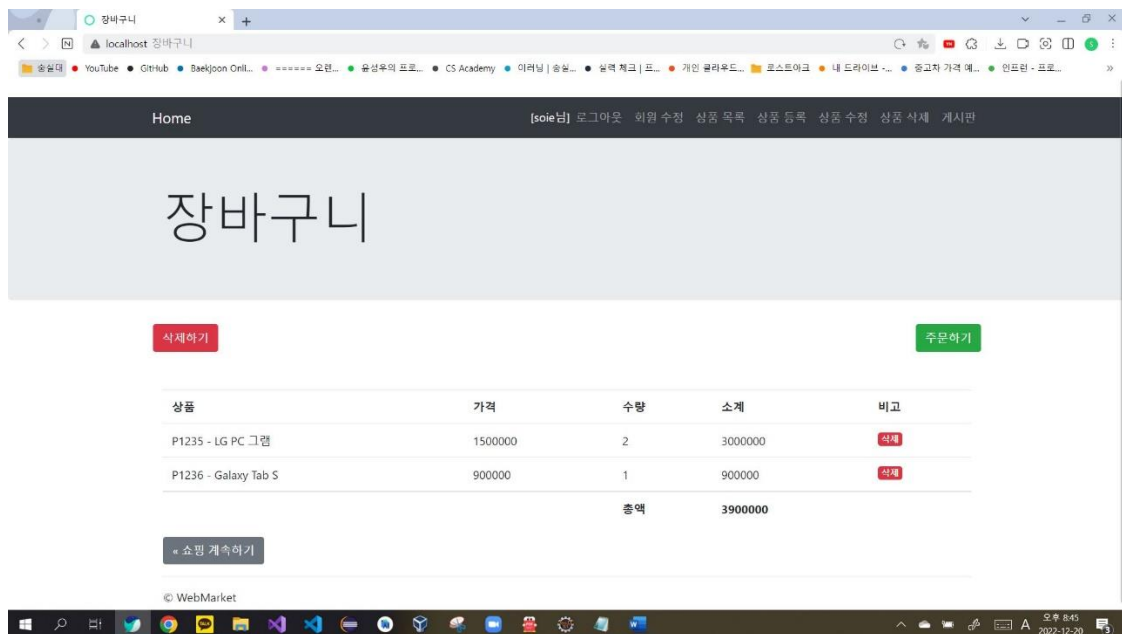
로그인 시 생성된 SessionId: B3DE35CAF3B5359D7FE4D5E8983898680



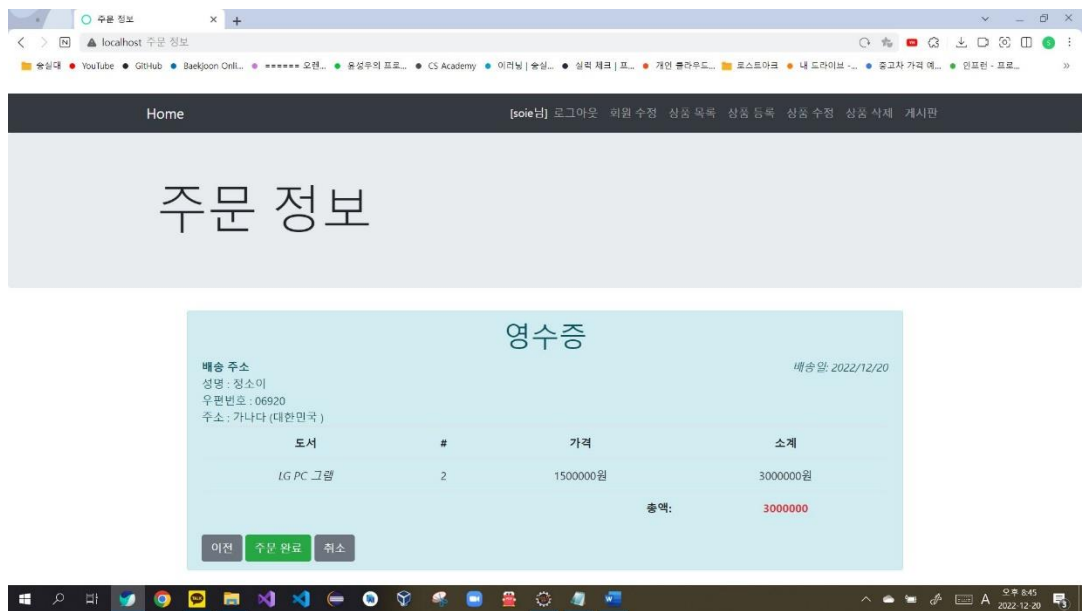
장바구니에 제품을 담았음



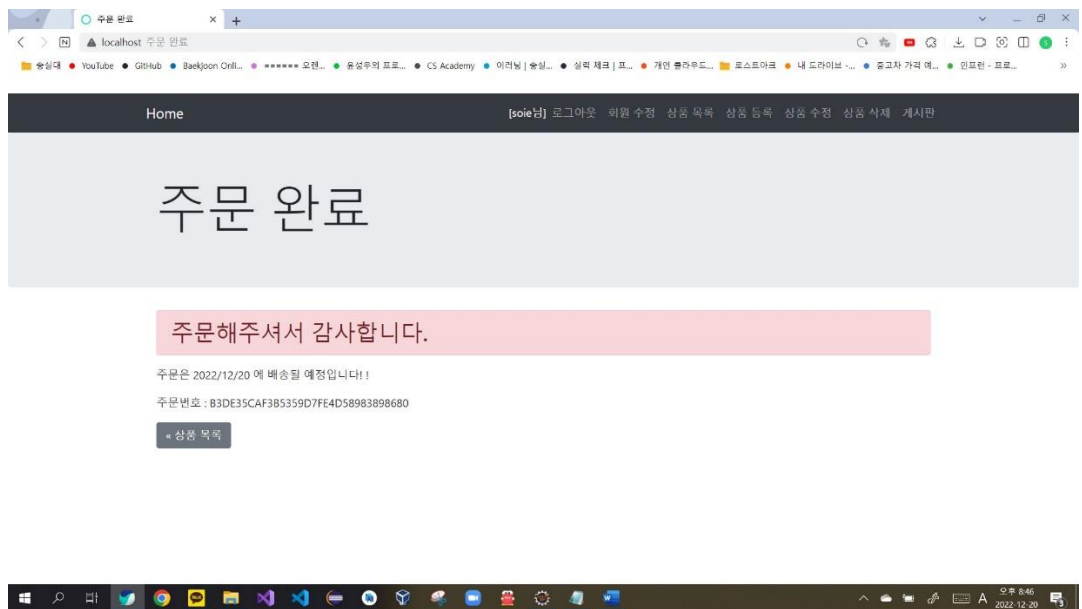
똑 같은 제품을 등록했을 때 수량이 늘어난 모습



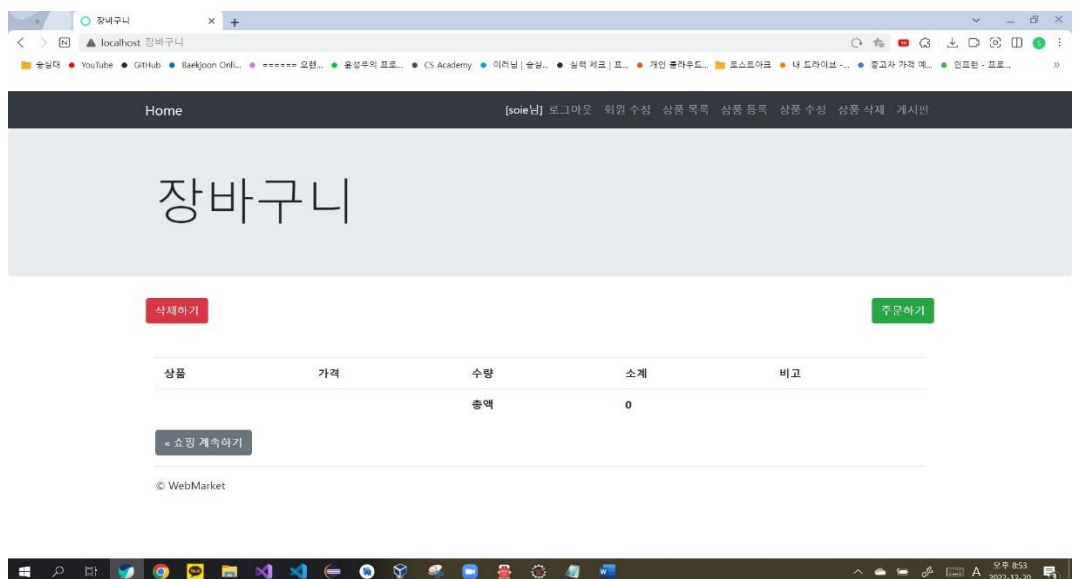
삭제 버튼을 눌렀을 때 제품이 잘 삭제되고 있는 것을 볼 수 있다.



주문 정보 페이지에서 장바구니의 정보가 잘 표현되는 것을 볼 수 있다.



주문 번호 cartId가 로그인 시 생성된 sessionId와 같은 것을 볼 수 있다.



주문 완료 후 다시 장바구니로 돌아가면 주문한 제품은 삭제된 것을 볼 수 있다.