

웹서버프로그래밍

[최종결과보고서]

프로그램명	GoalF Market
학번	2271387
이름	서준영

목 차

1. 프로젝트 개요	1
2. 프로젝트 진행 내용	1
2.1 프로젝트 진행 개요	2
2.2 프로젝트 전체 구성도	3
2.3 소개하고 싶은 페이지	9
3. 최종 결과	15
3.1 프로젝트 실행 결과	15
3.2 프로젝트 진행 특이사항	39
4. 프로젝트 진행 소감	41
5. 참고문헌	43

1. 프로젝트 개요

본 과제를 통해 진행하고자 하는 프로젝트의 이름은 'GoalF Market' 프로그램을 개발하고자 한다. GoalF Market은 골프와 관련된 용품을 판매하는 쇼핑몰 웹 사이트로, 목표를 의미하는 Goal과 Golf 단어를 합성해서 만들었다.

판매 할 물품에는 Club, Ball, Clothes가 있다. 교재에는 쇼핑몰 상단에 Home 버튼밖에 없지만, BootStrap의 Navbar Library를 활용해 각 판매 물품 페이지로 이동할 수 있게 한다. 로그인을 통해 사용자를 판매자와 구매자로 구분하고, 판매자는 물품 등록 페이지도 이용할 수 있게 한다. 세션을 활용해 장바구니 페이지도 구현하고, DB를 활용해 판매 물품의 등록, 조회, 편집 등을 가능하게 제작한다. DB는 MariaDB를 이용한다.

구매자의 현명한 소비를 위해 다른 웹 쇼핑몰의 가격과 비교할 수 있는 기능을 추가한다. 네이버 쇼핑에서 물품을 검색할 때, GET방식으로 URL query string에 값을 주는 방식인 것을 활용해 이 페이지로 이동시키는 방식을 활용한다.

상품의 배치나 가격, 필요한 항목 등은 '골마켓(<https://www.golmarket.co.kr/>)'을 참고해 제작한다.

외에도 개발 과정에서 추가하고 싶은 아이디어가 생각나거나 필요한 기능을 발견하면 추가한다.

평소에 취미로 부모님과 함께 즐기던 골프가 생각나 선정하게 됐다. 판매할 물품에 대한 쇼핑몰을 제작하려면 잘 아는 분야여야 하기 때문이다.

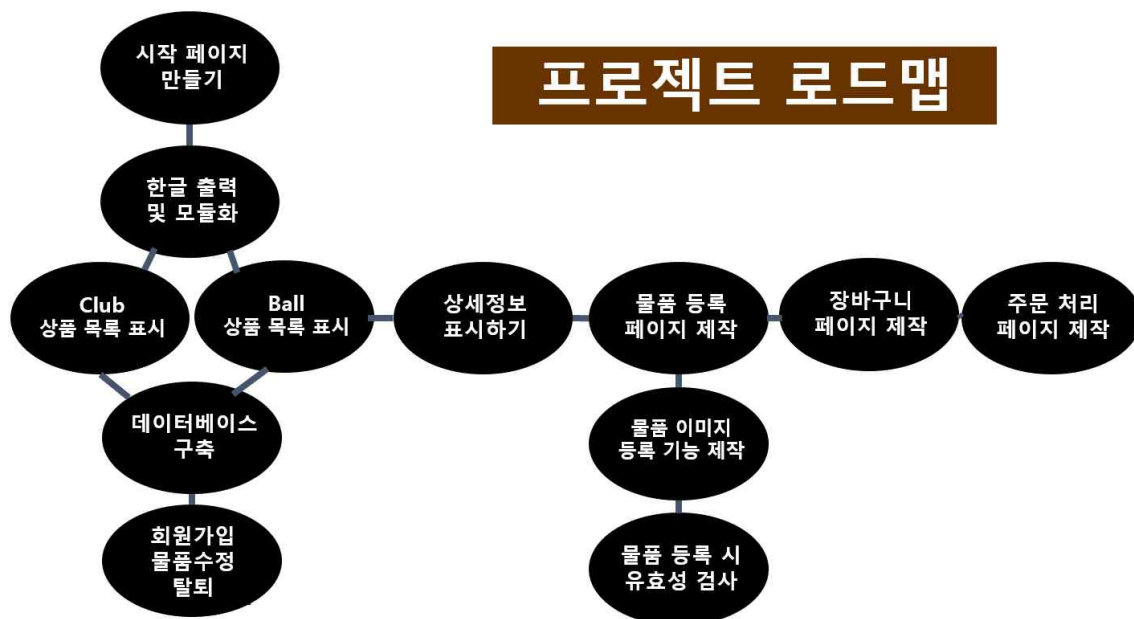
2. 프로젝트 진행 내용

2.1 프로젝트 진행 개요

기본	2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 15, 16
2, 3, 4, 5	기본 홈페이지 구조 완성
6	판매자 물품 등록 페이지 구조 완성
7	판매자 물품 등록 페이지에 이미지 추가 기능 완성
8	판매자 물품 등록 시 js로 유효성 검사 기능 완성
15	데이터베이스 개발환경 구축 완료 (MySQL 8.0.22)

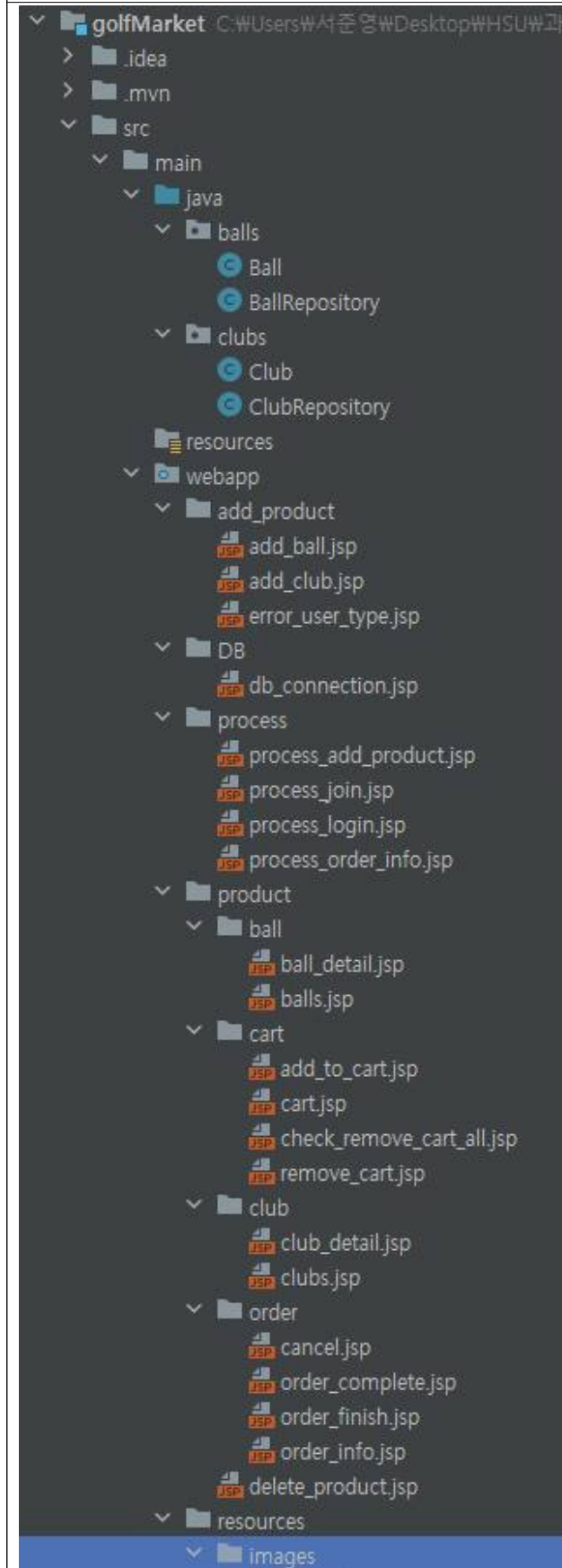
확장	2, 3, 4, 5, 6, 7, 8, 10, 13, 14, 15, 16, 17
2, 3, 4, 5	교재와 달리 두 개의 상품 카테고리(Club, Ball)로 나누어서 물품 확인 가능
6	판매자 등록도 두 개의 카테고리로 구분 가능
13	세션을 활용한 장바구니 기능 완성, JavaScript를 활용해 더 편하게 사용 가능하도록 제작 완료.
14	쿠키를 활용한 주문 처리 페이지 제작 중완료
16	데이터베이스를 활용한 상품 등록, 삭제 기능 및 회원등록, 탈퇴 기능 구현
16	데이터베이스를 활용한 상품 등록, 삭제 권한 기능 구현 판매자 및 관리자만 상품 등록, 삭제 가능

2.2 프로젝트 전체 구성도

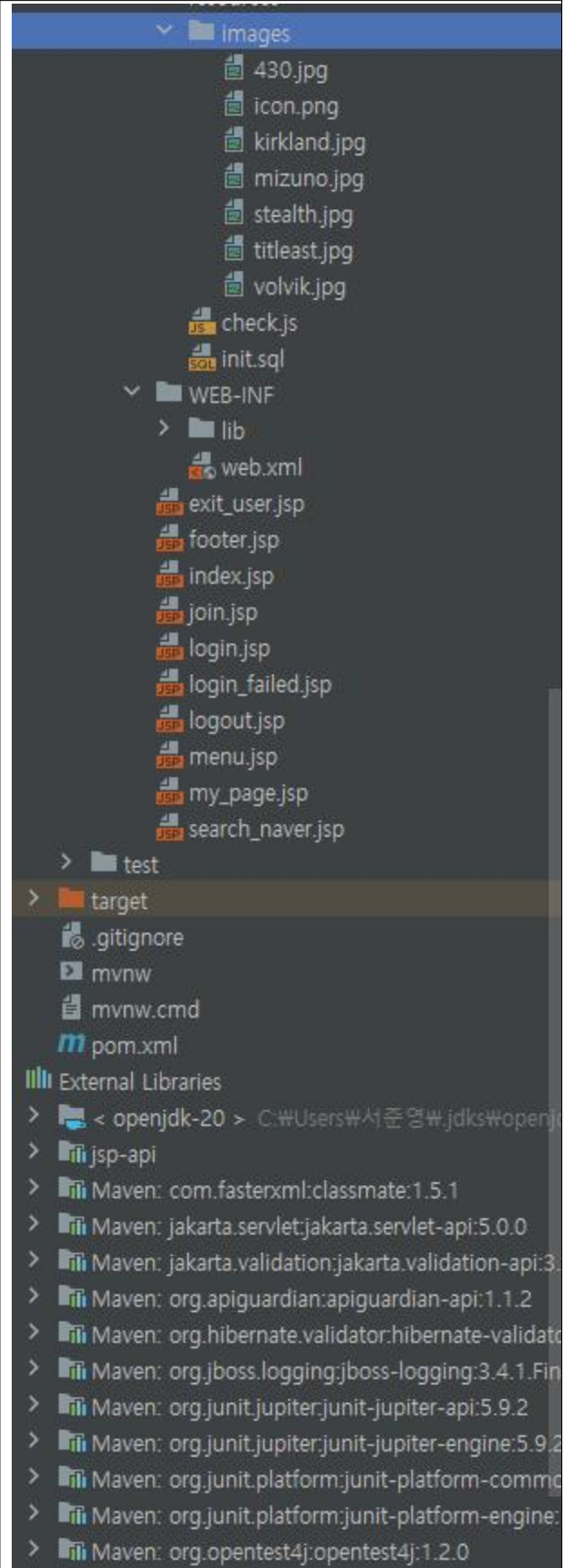


[프로젝트 로드맵]

project explorer(IntelliJ Ultimate) (1)



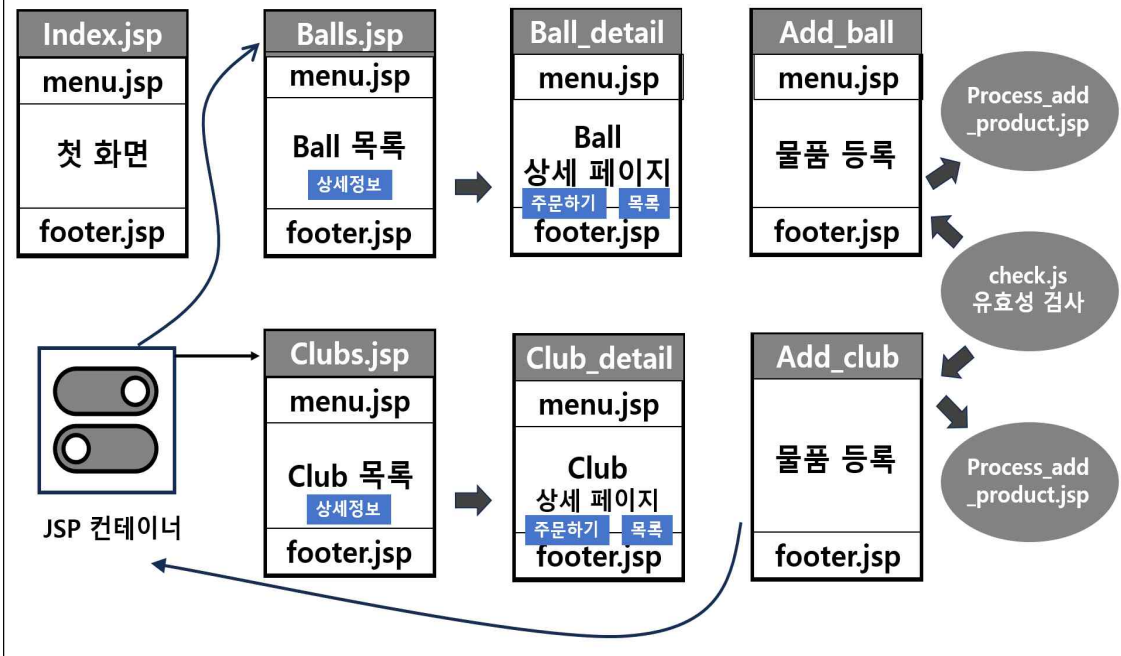
project explorer(IntelliJ Ultimate) (2)



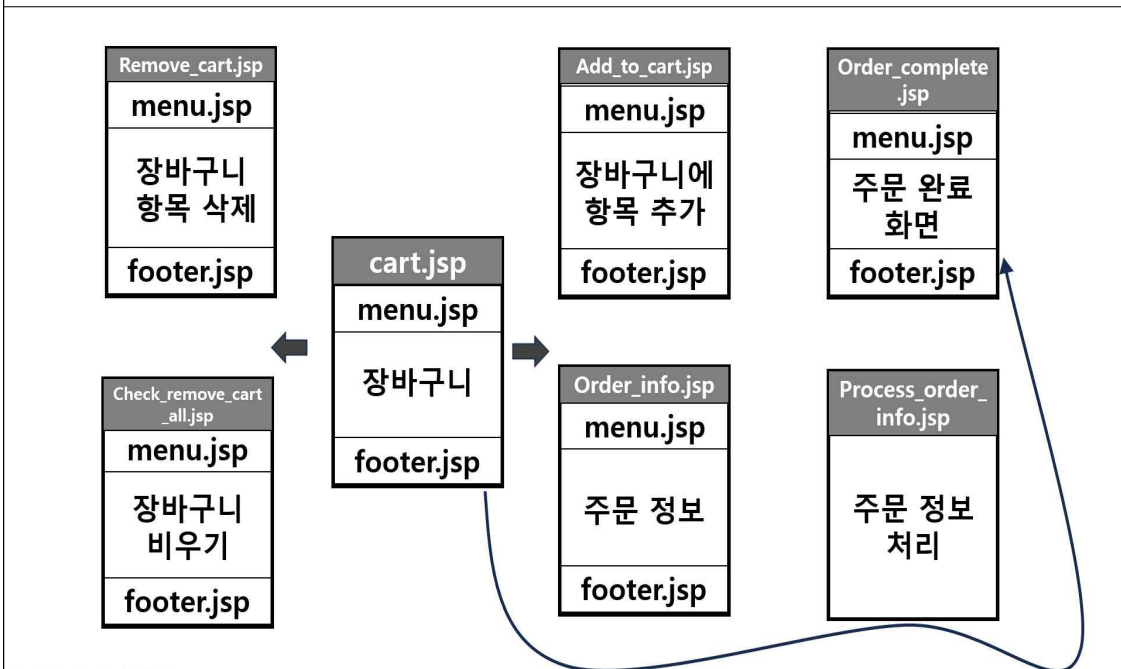
project explorer(IntelliJ Ultimate) (3)

- External Libraries
 - > < openjdk-20 > C:\Users\서준영\jdk\op
 - > jsp-api
 - > Maven: com.fasterxml:classmate:1.5.1
 - > Maven: jakarta.servlet:jakarta.servlet-api:5.0.0
 - > Maven: jakarta.validation:jakarta.validation-ap
 - > Maven: org.apiguardian:apiguardian-api:1.1.2
 - > Maven: org.hibernate.validator:hibernate-vali
 - > Maven: org.jboss.logging:jboss-logging:3.4.1
 - > Maven: org.junit.jupiter:junit-jupiter-api:5.9.2
 - > Maven: org.junit.jupiter:junit-jupiter-engine:5
 - > Maven: org.junit.platform:junit-platform-corr
 - > Maven: org.junit.platform:junit-platform-engi
 - > Maven: org.opentest4j:opentest4j:1.2.0
 - > mysql-connector-java-8.0.22
- Scratches and Consoles
 - Database Consoles
 - > GoalFMarketDB@localhost
 - console
- Extensions

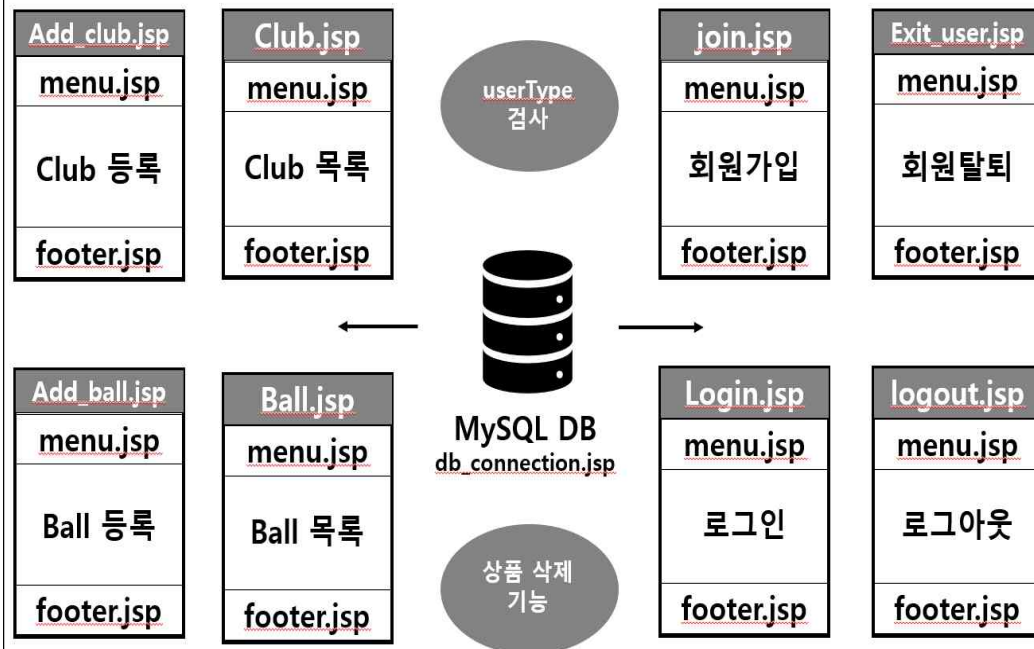
페이지 구성도(1)



페이지 구성도(2)



페이지 구성도(3)



2.3 소개하고 싶은 페이지

process_join.jsp

-회원가입 처리를 위한 페이지이다

-JSP와 JS를 이용해서 중복 가입 방지 처리를 구현했다.

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@include file="../DB/db_connection.jsp"%>
<html>
<head>
<script>
    //ID가 중복인 경우 실행
    //DB에서 ID 값 찾아서 존재하면 (resultSet.next())가능하면
    function already_exist_id(){
        alert("이미 존재하는 ID입니다. 다른 ID를 입력하세요.");
        window.location.href = "../join.jsp"
    }
</script>
</head>
<body>
<%
    PreparedStatement pstmt = null;
    ResultSet resultset = null;
    String sqlQuery = null;

    String id = request.getParameter("id");
    sqlQuery = "SELECT * FROM USER where id=?";
    pstmt = connection.prepareStatement(sqlQuery);
    pstmt.setString(1,id);
    resultset = pstmt.executeQuery();
    if(resultSet.next()){//if절 실행되면 ID가 중복
%>
    <script>
        already_exist_id();
    </script>
<%
    pstmt.close();
    resultset.close();
    connection.close();
```

process_join.jsp(2)

- ID 중복이 없는 경우 정상적으로 가입된다.
- JSP로 오류 페이지를 만들 수도 있었지만, 웹 페이지에서 여러 페이지를 이동하는 것은 피로감을 준다고 생각해 한 페이지 내에서 오류처리를 하도록 구현하고 싶었다. 그래서 JS를 선택했고, 좀 더 복잡했지만 원하는 구현을 해낼 수 있었다. 이 페이지 외에도 상품 등록 페이지, 로그인 처리 페이지 등 다양한 오류 처리를 JS로 구현했다.

```
<%  
    pstmt.close();  
    resultSet.close();  
    connection.close();  
    return; //ID 중복이면 모두 close() 후 JSP 종료  
}  
String pwd = request.getParameter("pwd");  
String type = request.getParameter("type");  
String name = request.getParameter("name");  
String tel = request.getParameter("tel");  
String nation = request.getParameter("nation");  
//항목 총 6개  
//id, pwd, type, name, tel, nation  
sqlQuery="INSERT INTO USER VALUES(?,?,?,?,?,?)";  
pstmt = connection.prepareStatement(sqlQuery);  
  
pstmt.setString(1,id);  
pstmt.setString(2,pwd);  
pstmt.setString(3,type);  
pstmt.setString(4,name);  
pstmt.setString(5,tel);  
pstmt.setString(6,nation);  
  
pstmt.executeUpdate();  
pstmt.close();  
connection.close();  
response.sendRedirect("../index.jsp");  
%>  
</body>  
</html>
```

index.jsp

- GoalF Market이 실행되는 홈 페이지이다.
- 화면 중앙의 네이버 쇼핑 검색은 JS로 구현되었다.

```
<%@ page import="java.util.Calendar" %>
<%@ page import="java.text.SimpleDateFormat" %>
<%@ page contentType="text/html; charset=utf-8" language="java" %>

<html>
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9F0VZLESaAA55NDz0xhy9GkcIds" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9F0VZLESaAA55NDz0xhy9GkcIds">
  <title>GoalF Market</title>
  <script>
    function search() {
      let search = document.getElementById("search").value;
      console.log(search);
      let url = "https://search.shopping.naver.com/search/all?query="+search;
      window.open(url);
    }
  </script>
</head>
<body>
<div class="container py-4">
  <%@include file="menu.jsp"%>
  <!--String welcome = "Welcome to GoalF Market";
  String tagline = "<b>온라인 골프 쇼핑물에 오신 것을 환영합니다!</b>";-->
  <div class="p-5 mb-4 bg-success text-white rounded-3">
    <div class="container-fluid py-5">
      <h1 class="display-5 fw-bold"><%=welcome%></h1>
      <p class="col-md-8 fs-4">Online golf mall</p>
    </div>
  </div>
  <div class="row align-items-md-stretch text-center">
    <div class="col-md-12">
      <div class="h-100 p-5">
        <h3><%=tagline%></h3>
      </div>
    </div>
  </div>
</div>
</body>
</html>
```

index.jsp(2)

- 홈 페이지가 실행될 때 마다 session.getAttribute를 통해 세션 정보를 얻어온다.

```
<div class="h-100 p-5">
  <h3><%=tagline%></h3>
  <%
    SimpleDateFormat sdf = new SimpleDateFormat("hh시 mm분 ss초");
    Calendar c = Calendar.getInstance();
    String userType = (String)session.getAttribute("userType");
    int dayNight = c.get(Calendar.AM_PM);
    if(dayNight==0) {
      out.print("현재 접속 시각 : 오전 " + sdf.format(c.getTime())+"<br><br>");
    }
    else{
      out.print("현재 접속 시각 : 오후 " + sdf.format(c.getTime())+"<br><br>");
    }
  %>

  <div class="container">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="p-3 text-primary-emphasis bg-info-subtle border border-primary-subtle rounded-3">
          <div class="input-group mb-3">
            <input type="text" class="form-control text-primary-emphasis" id="search" placeholder="네이버 쇼핑 최저가 검색하기">
            <button class="btn btn-outline-secondary text-primary-emphasis" type="button" onclick="search()" id="button-addon2">검색</button>
          </div>
        </div>
      </div>
    </div>
  </div>

  <%
    out.print("<br><br><br><br>");
  %>
```


index.jsp(3)

- session.getAttribute의 세션 정보는 서버 세션이고, JS를 통한 오류처리에는 활용할 수 없다. 그러므로 클라이언트 측 세션을 활용해야 했다.
- localStorage는 권한 정보가 계속 저장되므로 sessionStorage를 활용했다.
로그인 후에는 반드시 index.jsp를 거쳐 모든 페이지로 이어지도록 구현되었기 때문에 이 페이지에서 서버 세션 정보를 클라이언트 세션 정보와 동기화시켜 다양한 기능을 구현할 수 있었다.

```
<%
    out.print("<br><hr><br><br>");
    if(userType==null) {
        out.print("<b>로그인하세요</b>");
    }
    else if(userType.equals("normal")){
        out.print("<b>현재 접속 권한 : 일반 사용자</b>");
    }
    else if(userType.equals("seller")){
        out.print("<b>현재 접속 권한 : 판매자</b>");
    }
    else if(userType.equals("admin")){
        out.print("<b>현재 접속 권한 : 관리자</b>");
    }
%>

<script>
    sessionStorage.setItem("userType", "<%=userType%>");
</script>
</div>
</div>
</div>

<%@include file="footer.jsp"%>
</div>
</body>
</html>
```

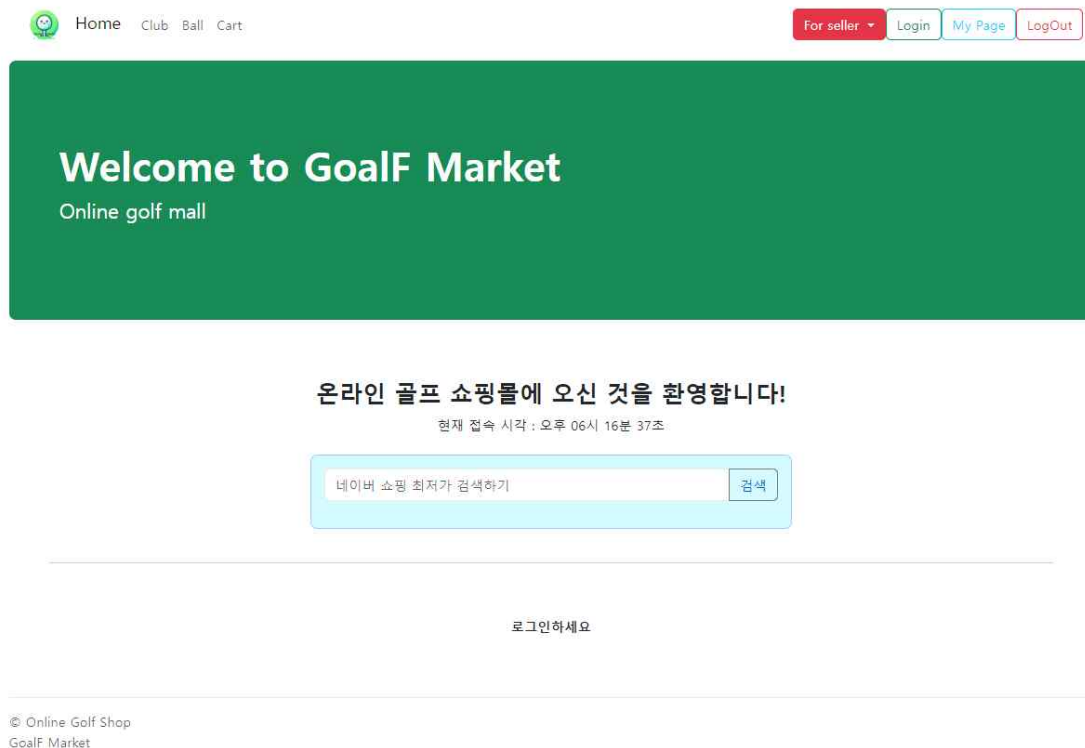
다양한 페이지에서 사용된 로그인 오류 처리 JS 함수이다.

이 함수의 구현이 까다로웠다. 완성하고 보니 간단하지만, 고생했던 부분은 sessionStorage.getItem의 반환값에 있었다. userType 세션이 설정되지 않은 경우(즉 로그인되지 않은 경우) null을 반환한다고 생각해 처음엔 if(!userType)으로 구현했다. 하지만 반환값은 null 이 아닌 “null” 이었다. 문서를 꼼꼼하게 확인하지 않아 생긴 일이었다.

```
<script type="text/javascript">
    function check_user_type(){
        let userType = sessionStorage.getItem("userType");
        if(userType==="null"){
            alert("로그인 해주세요.");
            window.location.href="../../login.jsp";
        }
    }
    window.onload=check_user_type;
</script>
```


3. 최종 결과

3.1 프로젝트 실행 결과



GoalF Market에 처음 접속한 화면이다.
화면 하단에서 현재 접속된 계정의 권한을 확인할 수 있다.

사용자의 편리함을 위해 제작된 상단의 NavBar는 GoalF Market 어느 페이지에서나 확인할 수 있다. Club은 골프 채 목록, Ball은 골프 공 목록, Cart는 현재 장바구니에 담아둔 상품을 확인할 수 있도록 구현되었다. Club, Ball 페이지는 로그인하지 않고도 접근할 수 있지만 Cart 페이지는 접근이 제한된다.

우측의 For Seller는 Dropdown Menu로, 판매자가 상품을 등록할 때 사용한다. 판매자 계정이 아닌 경우 접근이 제한된다. Login을 통해 로그인 할 수 있다.



로그인

Login

[로그인](#)[회원가입](#)

화면 우측 상단의 Login 버튼을 클릭해 실행된 로그인 화면이다.
ID와 Password를 입력해 로그인 할 수 있고, 회원가입을 통해 계정을 생성할 수 있다.



회원 가입

Join Us

아이디

test

비밀번호

이름

test

유형

☒ 일반 ☐ 판매자

전화번호

010-1234-5678

국가

Korea

등록하기

회원 가입 화면이다. 일반(구매자) 유형으로 'test' 라는 ID로 새로 회원가입을 진행했다.

localhost:8080 내용:

이미 존재하는 ID입니다. 다른 ID를 입력하세요.

확인

만약 이미 존재하는 ID라면, 에러 알림 창이 출력된다.

로그인

Login

ID
test

Password

로그인

회원가입

방금 회원가입한 test 계정으로 로그인을 진행한다.

Welcome to GoalF Market

Online golf mall

온라인 골프 쇼핑몰에 오신 것을 환영합니다!

현재 접속 시각 : 오후 06시 18분 49초

네이버 쇼핑 최저가 검색하기

검색

현재 접속 권한 : 일반 사용자

화면 하단의 접속 권한이 일반 사용자로 바뀐 것을 확인할 수 있다.

회원정보

My Page

ID : test

사용자 유형 : normal

전화번호 : 010-1234-5678

국가 : Korea

[홈으로 돌아가기](#)

회원탈퇴

로그인 후 우측의 My Page 버튼을 눌러 실행된 화면이다.

회원가입 시 작성했던 정보가 모두 저장되어 있다.

하단의 회원 탈퇴 버튼을 누르면 DB에서 계정이 삭제되고, 복구할 수 없다.

골프 채 목록

Golf Club List



스텔스 1

[TaylorMade] [Driver](#)
최고의 선택. 스텔스 드라이버

500000원

[상세 정보>](#)



Ping 430 Max

[PING] [Driver](#)
최신형 드라이버 핑

700000원

[상세 정보>](#)



Mizuno iron

[Mizuno] [Iron](#)
국민 아이언 미즈노

300000원

[상세 정보>](#)

골프채 정보

Club Info



스텔스 1

최고의 선택, 스텔스 드라이버

클럽 재고번호 ID : **C_000001**

브랜드 : TaylorMade

출시일 : 20-05-05

재고수 : 1

500000원

[주문하기](#)

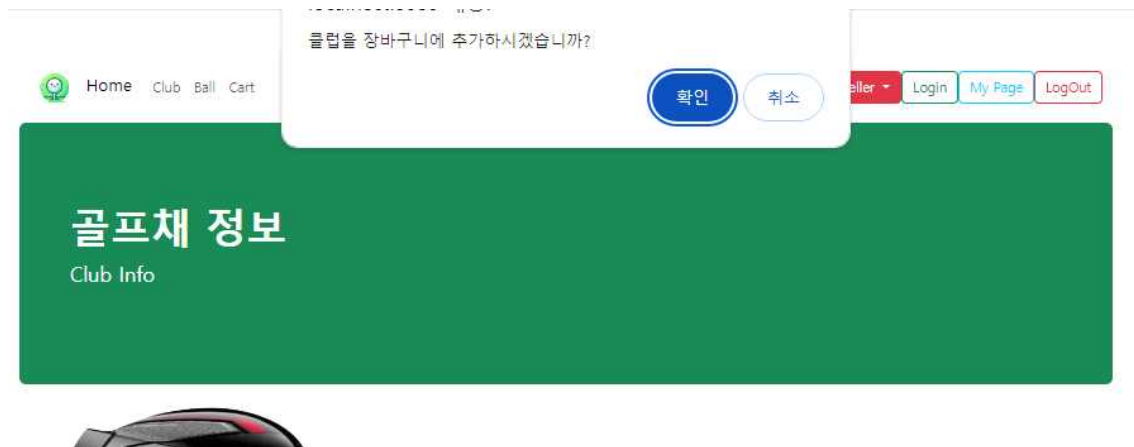
[장바구니 >](#)

[삭제하기](#)

[목록](#)

기존에 등록되어 있는 3개의 골프 채 중, 가장 왼쪽의 스텔스 1의 상세정보 버튼을 클릭한 화면이다. 판매하는 물품의 세부적인 정보를 확인할 수 있다.

주문하기 버튼을 눌러 바로 장바구니에 담을 수 있다.
삭제하기 버튼은 판매자 및 관리자만 접근할 수 있으며,
클릭하면 해당 상품이 DB에서 삭제되어 Club 페이지에 나타나지 않는다.



주문하기 버튼을 눌러 사용자에게 확인을 받는 알림창이 실행된 모습이다. 취소를 누르면 장바구니에 담기지 않고, 아무 일도 일어나지 않는다.



바로 이전의 알림창에서 확인을 클릭하면, 다시 알림창이 출력된다.
사용자의 편리함을 위해 추가된 기능으로 확인을 클릭하면
바로 장바구니 페이지로 이동할 수 있다.

장바구니

Cart

[삭제하기](#)[주문하기](#)

상품	가격	수량	소계	비고
C_000001 - 스텔스1	500000	1	500000	삭제
		총액	500000	

[« 쇼핑 계속하기](#)

확인을 클릭해 바로 장바구니로 이동한 모습이다. 화면 상단의 Cart 버튼을 클릭해서도 이동할 수 있다.

좌측의 삭제하기를 누르면 장바구니에 담겨 있는 모든 상품이 삭제된다. <비고>에 있는 삭제를 누르면 그 행에 해당하는 상품만 삭제된다.

우측의 주문하기를 누르면 장바구니에 있는 모든 상품을 주문하는 페이지로 연결된다.

장바구니

Cart

삭제하기

주문하기

상품	가격	수량	소계	비고
B_000001 - 타이틀리스트	50000	2	100000	삭제
B_000002 - Volvik	7000	1	7000	삭제
C_000001 - 스텔스1	500000	1	500000	삭제
C_000003 - Mizuno Iron	300000	1	300000	삭제
총액			907000	

◀ 쇼핑 계속하기

© Online Golf Shop
Golf Market

테스트를 위해 여러 상품을 장바구니에 추가한 화면이다.

1행의 상품은 장바구니에 두 번 담아 수량이 2개고,
소계에서 100000으로 되어있는 모습을 확인할 수 있다.

localhost:8080 내용:

정말로 삭제할까요? 되돌릴 수 없어요.

확인

취소

좌측의 삭제하기를 누르면 사용자에게 최종 확인을 받는다.
확인을 누르면 장바구니의 모든 상품 항목이 삭제되고 되돌릴 수 없다.

주문 정보

Order Information

성명	<input type="text" value="test"/>
국가명	<input type="text" value="Korea"/>
우편번호	<input type="text" value="12345"/>
주소	<input type="text" value="서울시 성북구 삼선교로 16길 116"/>
수령인 전화번호	<input type="text" value="010-1234-5678"/>
쇼핑 계속하기 주문하기 취소하기	

주문하기를 눌러 주문정보를 입력하는 페이지가 실행된 화면이다.
원하는 정보를 입력할 수 있다. 주문하기를 누르면 구매 영수증을 확인하는 페이지로 이동한다.



Home

Club

Ball

Cart

For seller >

Login

My Page

LogOut

주문 정보

Order Info

영수증

배송 주소

성명 : test

우편번호 : 12345

주소 : 서울시 성북구 삼선교로 16길 116 (Korea)

전화번호 : 010-1234-5678

배송일: 2024-06-08

상품	#	가격	소계	
B_000001 - 타이틀리스트	50000	2	100000	삭제
B_000002 - Volvik	7000	1	7000	삭제
C_000001 - 스틸스1	500000	1	500000	삭제
C_000003 - Mizuno Iron	300000	1	300000	삭제
총액:			907000	

[이전](#)[주문 완료](#)[취소](#)

주문하기를 클릭해 실행된 페이지이다. 사용자의 최종 확인을 받는 역할을 한다.
우측의 배송일은 현재 시간 + 2일로 설정되어 있다. 실제와 같은 배송기간 구현을 위함이다.

삭제 버튼을 눌러 원하지 않는 항목을 삭제하거나, 취소를 눌러 주문을 취소할 수 있다.
주문 완료 버튼을 클릭하면 최종적으로 주문이 완료된다.

주문 완료

Order Completed

주문해주셔서 감사합니다.

주문은 2024-06-08 에 배송될 예정입니다!!

주문번호 : D82E5370273817249352134E505983AB

배송지 : 서울시 성북구 삼선교로 16길 116

수령인 전화번호 : 010-1234-5678

[#홈으로 돌아가기](#)

주문 완료를 클릭해 최종적으로 주문이 완료된 화면이다.
주문번호는 세션의 id값으로 설정되어 있다.

권한 오류

판매자 및 관리자만 상품 등록 페이지에 접근할 수 있습니다.

[돌아가기](#)

© Online Golf Shop
GoalF Market

로그아웃하지 않고(구매자 계정으로) 화면 상단의 For Seller 메뉴를 클릭한 모습이다.
로그인하지 않은 상태로 클릭해도 결과는 동일하다.

회원 가입

Join Us

아이디	<input type="text" value="seller"/>
비밀번호	<input type="password" value="...."/>
이름	<input type="text" value="판매자"/>
유형	<input type="radio"/> 일반 <input checked="" type="radio"/> 판매자
전화번호	<input type="text" value="010-0000-0000"/>
국가	<input type="text" value="USA"/>
<input type="button" value="등록하기"/>	

© Online Golf Shop
GoalF Market

판매자 계정인 seller로 새롭게 회원가입을 진행한다.

로그인

Login

ID

seleer

Password

....

로그인

회원가입

판매자 계정으로 로그인을 진행한다. seller가 아닌 seeler를 입력했다.
올바르지 않은 ID 또는 비밀번호인 경우 오류 페이지로 이동한다.

로그인 실패

아이디 또는 비밀번호를 확인해주세요.

◀돌아가기

ID가 틀렸기 때문에 오류 페이지가 실행된 화면이다.

볼 등록

Addition Ball

상품코드	<input type="text" value="B_"/>
상품명	<input type="text"/>
가격	<input type="text"/>
브랜드	<input type="text"/>
출시일	<input type="text"/>
설명	<input type="text" value="100자 이상 적어주세요"/>
재고수	<input type="text"/>
로스트볼 여부	<input type="radio"/> 신품 <input type="radio"/> 로스트볼
카테고리	<input type="radio"/> 화이트 볼 <input type="radio"/> 컬러 볼
이미지	<input type="button" value="파일 선택"/> 선택된 파일 없음

[등록하기](#)

다시 판매자 계정으로 정상적으로 로그인을 진행한 후, For Seller 메뉴를 클릭 후 Ball 상품 등록 페이지로 이동한 화면이다. Club의 경우 C_, Ball의 경우 B_로 상품 코드가 시작되고, 내부적인 상품 구분 처리를 위해 사용된다.
이 값은 등록 페이지에 자동으로 입력되어져 있다.

볼 등록

Addition Ball

상품코드	<input type="text" value="B_000004"/>
상품명	<input type="text" value="Test"/>
가격	<input type="text" value="9999"/>
브랜드	<input type="text" value="Test"/>
출시일	<input type="text" value="Test"/>
설명	<input type="text" value="테스트를 위한 상품 등록"/>
재고수	<input type="text" value="9999"/>
로스트볼 여부	<input type="radio"/> 신품 <input checked="" type="radio"/> 로스트볼
카테고리	<input checked="" type="radio"/> 화이트 볼 <input type="radio"/> 컬러 볼
이미지	<input type="text" value="파일 선택"/> 캡처.PNG
<input type="button" value="등록하기"/>	

테스트를 위해 각 항목을 모두 입력했다. 이미지 파일은 판매자가 원하는 대로 정할 수 있고, 서버의 리소스 폴더에 저장되므로 모든 사용자가 로드할 수 있다.



titleist

[타이틀리스트] **whiteBall**

타이틀리스트 볼 최고의 반발력

50000원

[상세 정보>](#)



Volvik color ball

[볼빅] **ColorBall**

볼빅 컬러 볼

7000원

[상세 정보>](#)



CostCo ball

[kirkland] **whiteBall**

코스트코 언폴 볼

3000원

[상세 정보>](#)



Test

[Test] **white**

테스트를 위한 상품 등록

9999원

[상세 정보>](#)

정상적으로 등록을 마친 후 Ball 버튼을 클릭한 화면이다. 기존에 등록되어있던 3개의 볼 외에 방금 전 등록했던 Test Ball을 확인할 수 있다.

골프 볼 정보

Ball Info



Test

테스트를 위한 상품 등록

볼 재고번호 ID : **B-000004**

브랜드 : Test

출시일 : Test

재고수 : 9999

9999원

[주문하기](#)

[장바구니](#)

[삭제하기](#)

[목록](#)

모든 상품은 등록시 DB에 등록되며, DB의 값을 바탕으로 볼 목록 페이지와 볼 상세정보 페이지가 구현된다. Club 페이지와 동일한 방식으로 상품을 주문, 삭제 할 수 있다.

골프 볼 목록

Golf Ball List



titleist

[타이틀리스트] [whiteBall](#)

타이틀리스트 볼, 최고의 반발력

50000원

[상세 정보>](#)



Volvik color ball

[볼빅] [ColorBall](#)

볼빅 컬러 볼

7000원

[상세 정보>](#)



CostCo ball

[kirkland] [whiteBall](#)

코스트코 번들 볼

3000원

[상세 정보>](#)

현재 로그인 된 계정이 판매자 계정이므로, 삭제하기 버튼을 클릭해 물품을 삭제했다.
일반 계정(구매자)의 경우 접근 권한 오류 페이지로 이동한다.

회원정보

My Page

ID : **seller**

사용자 유형 : **seller**

전화번호 : **010-0000-0000**

국가 : **USA**

[홈으로 돌아가기](#) »

[회원탈퇴](#)

판매자 계정으로 로그인 된 상태에서 My Page를 클릭한 화면이다.
가입 시 입력한 정보가 모두 포함되어 있다.

localhost:8080 내용:

로그인 되어있지 않습니다.

확인

localhost:8080 내용:

로그인 후 마이페이지에 접속할 수 있습니다.

확인

localhost:8080 내용:

이미 로그인되어 있습니다.

확인

각각 로그인 되어 있지 않은 상태로 Logout, MyPage를 클릭하고, 로그인 된 상태로 Login 버튼을 클릭해 실행된 오류이다. 확인을 누르면 로그인 페이지 또는 홈 페이지로 알맞게 이동된다.

Welcome to GoalF Market

Online golf mall

온라인 골프 쇼핑물에 오신 것을 환영합니다!

현재 접속 시각 : 오후 06시 15분 09초

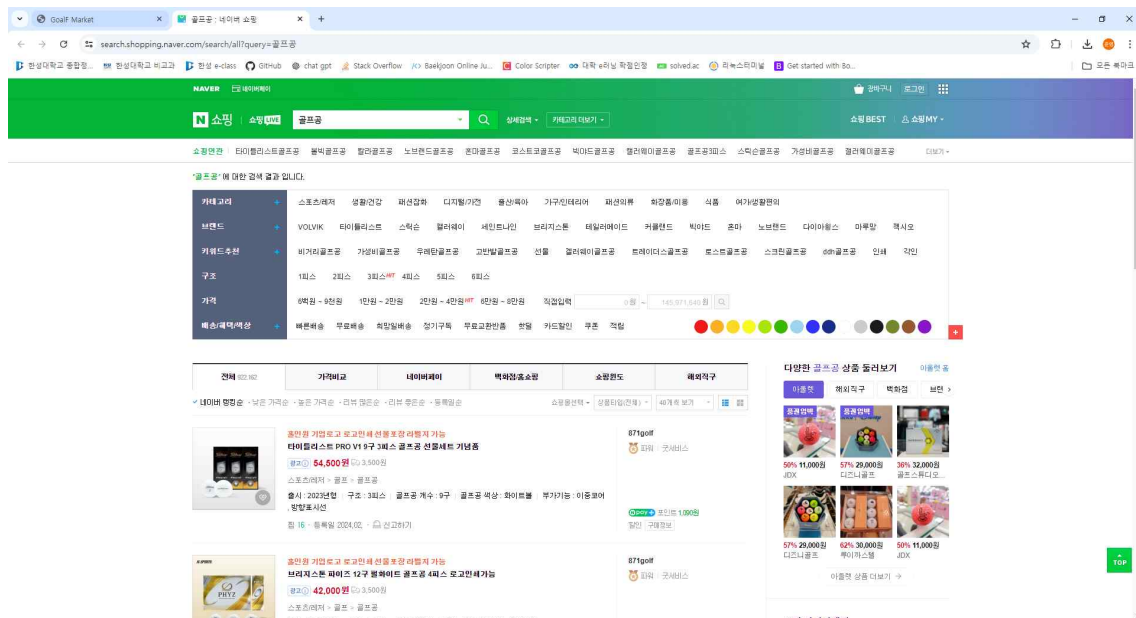
현재 접속 권한 : 판매자

© Online Golf Shop
GoalF Market

세 가지 오류 중 마지막 오류(로그인 된 상태로 다시 Login 클릭)의 확인 버튼을 클릭한 화면이다. 홈 페이지로 이동되고, 현재 계정이 로그아웃되지 않는 것도 확인할 수 있다.

홈페이지 중앙의 검색 창은 네이버 쇼핑으로 연결되는 검색창이다. 내용을 입력하고 우측의 검색 버튼을 클릭하면 네이버 쇼핑에서 동일한 내용으로 검색한다.

로그인하지 않고도 이용할 수 있는 기능이다.



검색 버튼을 눌러 네이버 쇼핑 홈페이지가 실행된 모습이다. GoalF Market에서 검색한 “골프공” 항목을 자동으로 검색하도록 구현됐다. 이 기능은 사용자가 최저가를 비교해 쇼핑할 수 있도록 제작되었다.

	id	pwd	type	name	tel	nation
▶	a	b	NULL	NULL	c	d
	buyer	buyer	normal	ss	123123	why
	s	1234	normal	s	test	test
	seller	test	seller	판매자	010-0000-0000	USA
	test	test	normal	test	010-1234-5678	Korea
	testseller	testpwd	seller	testseller	test	test
*	NULL	NULL	NULL	NULL	NULL	NULL

테스트 후 MySQL WorkBench에서 SELECT * FROM USER; SQL Query를 실행한 결과이다. 테스트를 위해 회원가입한 seller와 test 계정을 확인할 수 있다.

	ball_id	ball_price	ball_name	ball_brand	ball_description	ball_stock	ball_category	ball_releaseDate	ball_isLosted	ball_fileName
▶	B_000001	50000	titleast	타이틀리스트	타이틀리스트 볼, 최고의 반발력	50	whiteBall	20-05-05	0	titleast.jpg
	B_000002	7000	Volvik color ball	볼빅	볼빅 컬러 볼	30	ColorBall	22-01-15	1	volvik.jpg
	B_000003	3000	CostCo ball	kirkland	크스트코 번들 볼	1000	whiteBall	24-03-01	0	kirkland.jpg
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

테스트 후 MySQL WorkBench에서 SELECT * FROM BALL; SQL Query를 실행한 결과이다. 테스트를 위해 생성했던 test Ball을 삭제해서 더 이상 DB에서 조회할 수 없는 것을 확인할 수 있다.

3.2 프로젝트 특이사항

<사용 라이브러리, IDE, 프레임워크 버전 등 명세>

사용 IDE : IntelliJ IDEA Ultimate - 2024.1.1.

구현을 위해 사용된 언어 및 프레임워크: HTML, CSS(Bootstrap), JavaScript, Java, JSP, MySQL, Maven

사용 라이브러리 및 버전:

MySQL - 8.0.22

MySQL JDBC - 8.0.22

cos - 22.0.5

fasterXML - 1.5.1

jakarta.servlet - 5.0.0

jakarta.validation 3.0.0

apiguardian - 1.1.2

hibernate validator - 7.0.5 final

jdk - 20.0.1

maven compiler - 11

maven war plugin - 3.3.2

J unit - 5.9.2

서버 : Apache Tomcat 10.1.19

프로젝트를 진행하며 프로젝트 환경 구축이 가장 까다로웠다. 여러 프로젝트를 진행할 때마다 느끼지만, 환경 구축이 가장 까다롭고 가장 꼼꼼하게 잘 준비해야 한다. 교재에선 Eclipse Enterprise IDE로 프로젝트를 진행했지만, 나는 Eclipse가 싫어 IntelliJ로 진행하다 보니 교재의 환경 구축과는 다른 부분이 많았다.

직접 검색하고 공부해가며 환경을 구축했다. 처음 IntelliJ를 설치하고 Jakarta EE를 라이브러리로 포함시키는 것부터 어려웠다. Eclipse Enterprise에선 Dynamic web project 클릭 한번으로도 다 진행되었지만 IntelliJ에선 그렇지 않았다. Project Structure에서 직접 Maven에서 추가해야 했다. 여기까진 IDE의 기능으로 쉽게 할 수 있었지만, 문제는 cos였다. MultipartRequest를 위해 반드시 필요한데, 잘 사용되지 않는 라이브러리라 Maven에 없었다. 가지고 있는 cos.jar 파일을 라이브러리에 직접 추가하면, 컴파일되어 동작은 하지만 IDE가 이를 인식하지 못해 자동완성 기능도 사용할 수 없고, 알 수 없는 문장이라며 빨간 밑줄을 계속 그어댔다. 이 문제를 해결하기 위해선 pom.xml에 외부 라이브러리 경로(PC 내 절대경로)를 Dependency로 추가하고, 빌드 플러그인도 따로 설정해줘야 했다. 그 외에도 패킷, 아티팩트, WAR, WAR_exploded 등 공부해야 할 것이 많았다.

이 정도까지 스스로 공부하고 문제를 해결하다 보니 자신감이 많이 붙었다. 교재를 따라가지 않고 내가 원하는 대로, 만들고 싶은 웹 앱을 만들기로 결심했다. 상품의 종류를 늘리고, 회원이 DB에서 처리하고, 페이지 내에서만 오류처리를 해서 서버 부하를 줄이는 등 원하는 기능은 대부분 구현해 낼 수 있었다.

처음 어려웠던 구현에서의 난관은 MultiPartRequest를 통한 이미지 업로드였다. 이미지를 업로드한 뒤 페이지에선 이미지가 로드되지 않지만, 서버 PC의 리소스 폴더엔 잘 저장되는 것이었다. 디버깅 로그를 추가하고, 온갖 문서를 다 뒤져봐도 소용이 없었다.

결론적으로 문제는, apache tomcat 폴더 하위의 webapps 폴더에 GoalFMarket 프로젝트의 배포 폴더가 있지 않았다는 것이었다. 학년별로 진행했던 프로젝트를 정리하기 위해 tomcat 서버 폴더가 아닌 다른 폴더로 지정해뒀는데, 이것이 화근이었다. 그래서 tomcat 서버는 마치 정적 리소스를 로드하듯이, 서버가 처음 시작될 때 모두 포함했던 것이었다. 이 문제는 우연히 Server Restart를 할 때만 이미지가 로드되는 것을 눈치채고 해결할 수 있었다. 한참을 고생했던 문제였지만 서버의 동작 방식에 대해 많은 공부를 할 수 있었다.

두 번째로 Cookie의 문제도 있었다. 교재에선 JSP 파일들을 모두 한 디렉토리에 두기 때문에 쿠키가 계속 유효하지만, GoalFMarket 프로젝트에선 디렉토리를 상세하게 나누어 진행했다. 그 때문에 쿠키가 계속 유효하지 않아 주문처리 페이지가 제대로 동작하지 않았다. 한참을 검색하고 공부해 Cookie의 setPath 메서드와 하위 URL에선 쿠키가 유효하다는 사실을 배울 수 있었다.

마지막으로는 세션이었다. sessionStorage를 제대로 알지 못해 무작정 JSP의(서버의) session으로만 모든 문제를 해결하려고 했다. 하지만 웹 페이지의 사용성을 위해 한 페이지 내에서 이동하지 않고 오류처리를 구현하고 싶었고, 결국 JavaScript에서는 세션을 어떻게 사

용할 수 있을지를 고민하게 되었다. 그렇게 찾아낸 것이 sessionStorage였다. 하지만 sessionStorage.getItem("userType")과 같이 사용해도 원하는 결과를 얻어낼 수는 없었다. 서버측 세션과 클라이언트측 세션은 서로 다른 것이었기 때문이다. 그 외에도 반환형을 제대로 보지 못해 한참을 헤맸었다. 이 문제를 해결하며 웹과 세션에 대해 한층 더 깊이 알 수 있었고, 세션을 활용해 회원 권한 처리를 구현하며 세션을 활용해 제작할 수 있을 만한 여러 아이디어들이 떠오르기도 했다.

그 외에도 어려운 문제들은 많았다. navBar에서 웹앱의 루트 디렉토리를 어떻게 설정할 것인지, 부트스트랩을 활용한 Dropdown 메뉴는 왜 동작하지 않는지, 상품이 Ball인지 Club인지를 어떻게 구분할 것인지 등이었다. 그 때 마다 포기하지 않고 끝까지 문제를 해결해 냈다. 문제를 해결할 때 마다 스스로도 지식이 늘어가는 것이 느껴졌다. 문제를 마주할 때 마다 괴롭긴 했지만 얻어가는 것이 많았다.

4. 프로젝트 진행 소감

완벽한 웹 앱이라고 부를 수는 없겠지만 어느 정도 수준의 구현은 완료한 것 같아 뿌듯하다. 원하는 기능 구현과 원하는 결과물 제작을 위해선 구현 실력은 기본이고, 전체적인 큰 그림의 이해와 함께 기술의 원리를 이해하는 것 또한 매우 중요하다는 것을 많이 배울 수 있었다. 단순히 코드를 잘 짤지거나 복잡한 코딩을 잘 해낸다 수준의 문제가 아니었다. 어떤 함수나 기술을 배울 때 깊이, 꼼꼼히 공부해야 한다는 것을 알게 되었다. 예를 들어 세션에서의 문제처럼, 세션을 이해하지 못하거나 서버 측, 클라이언트 측의 세션이 분리되어 있다는 사실을 공부하지 않았다면 아무리 코딩을 잘해도 문제는 해결할 수 없었을 것이다.

이번 프로젝트를 통해 문제를 마주하면 외면하거나 도망치지 말고, 끝까지 붙잡고 노력하면 해결할 수 있다는 자신감과, 해결을 통해 많은 것을 배울 수 있다는 사실을 깨닫게 되었다. 앞으로도 어떤 문제를 마주하면 외면하거나 누구에게 도움을 청해 쉽게 해결하기보다는, 스스로 부딪혀 많은 지식을 배우는, 뿌리깊은 프로그래머가 되기 위해 노력해야겠다.

5. 참고 문헌

- [1] 쉽게 배우는 JSP 웹 프로그래밍, 송미영, 한빛아카데미, 2023
- [2] 명품 HTML5+CSS3+Javascript 웹 프로그래밍, 황기태, 생능출판사, 2022
- [3] 부트스트랩, <https://getbootstrap.com/docs/5.3/components/buttons/>
GoalFMarket의 디자인을 위해 활용했다.
- [4] 골마켓, <https://www.golmarket.co.kr>
- [5] 개인 블로그, <https://blog.naver.com/hack1s/40097085809>
DefaultRenamePolicy() 사용을 위해 참조했지만 문제가 해결되진 않음.
- [6] 개인 블로그, <https://hajoung56.tistory.com/108>
크롬 브라우저에서 쿠키 값이 정상적으로 설정되는지 확인하기 위해 참조했지만 문제가 해결되진 않음.
- [7] 개인 블로그,
<https://programmers.tistory.com/entry/Cookie-%EC%98%B5%EC%85%98-Path>
Cookie.setPath와 개발자 도구에서 쿠키 확인(디버깅)하는 방법을 검색했다.
- [8] DBMS & NoSQL
<https://www.lesstif.com/dbms/mysql-jdbc-the-server-time-zone-value-kst-is-unrecognized-or-represents-more-than-one-time-zone-100204548.html>
timezone SQL Exception 문제해결을 위해 검색했다.
- [9] JavaScript Info, <https://ko.javascript.info/localstorage>
sessionStorage의 특성과 사용법을 공부했다.
- [10] 개인 블로그,
<https://doosicee.tistory.com/entry/Maven%EC%9D%98-%EC%84%A4%EC%A0%95%ED%8C%8C%EC%9D%BC-Pomxml%EC%9D%84-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90>
maven과 pom.xml, 의존성 관리, 라이브러리 추가 등 다양한 부분을 공부했다.

[11] 개인 블로그,

<https://velog.io/@leyuri/IntelliJ-%EC%9B%B9-%EC%96%B4%ED%94%8C%EB%A6%AC%EC%BC%80%EC%9D%B4%EC%85%98-%EB%B0%B0%ED%8F%AC-%EC%9C%84%ED%95%9C-%ED%8C%A8%ED%82%A4%EC%A7%80-%EC%9C%A0%ED%98%95%EA%B3%BC-war-warexploaed-%EC%B0%A8%EC%9D%B4>

war, war_exploded에 대해 공부했다.

[12] 개인 블로그,

<https://velog.io/@sunk-dev/IntelliJ-Tomcat-%ED%99%98%EA%B2%BD-%EC%84%A4%EC%A0%95>

IntelliJ와 tomcat서버 연동을 위한 기본 설정을 공부했다.

[13] 인프런 질문답변,

<https://www.inflearn.com/questions/23403/artifact%EB%8A%94-%EB%AC%B4%EC%97%87%EC%9D%B8%EA%B0%80%EC%9A%94>

아티팩트란 무엇인가에 대해 공부했다.