



애견용품 쇼핑몰 프로젝트

강현, 김영준, 손룡, 서수빈

목차



프로젝트 기획



개발환경



프로젝트 개요



마무리

프로젝트 기획



프로젝트 소개

강아지 사료, 간식, 영양제, 의류, 악세서리 등
모든 강아지 용품이 여기에 다 있다!

BOWWOW은 반려동물과 함께하는 더 나은 반려 생활을
꿈꿉니다.

개발환경



프로젝트 일정

기획

01.11~01.13

설계

01.16~01.17

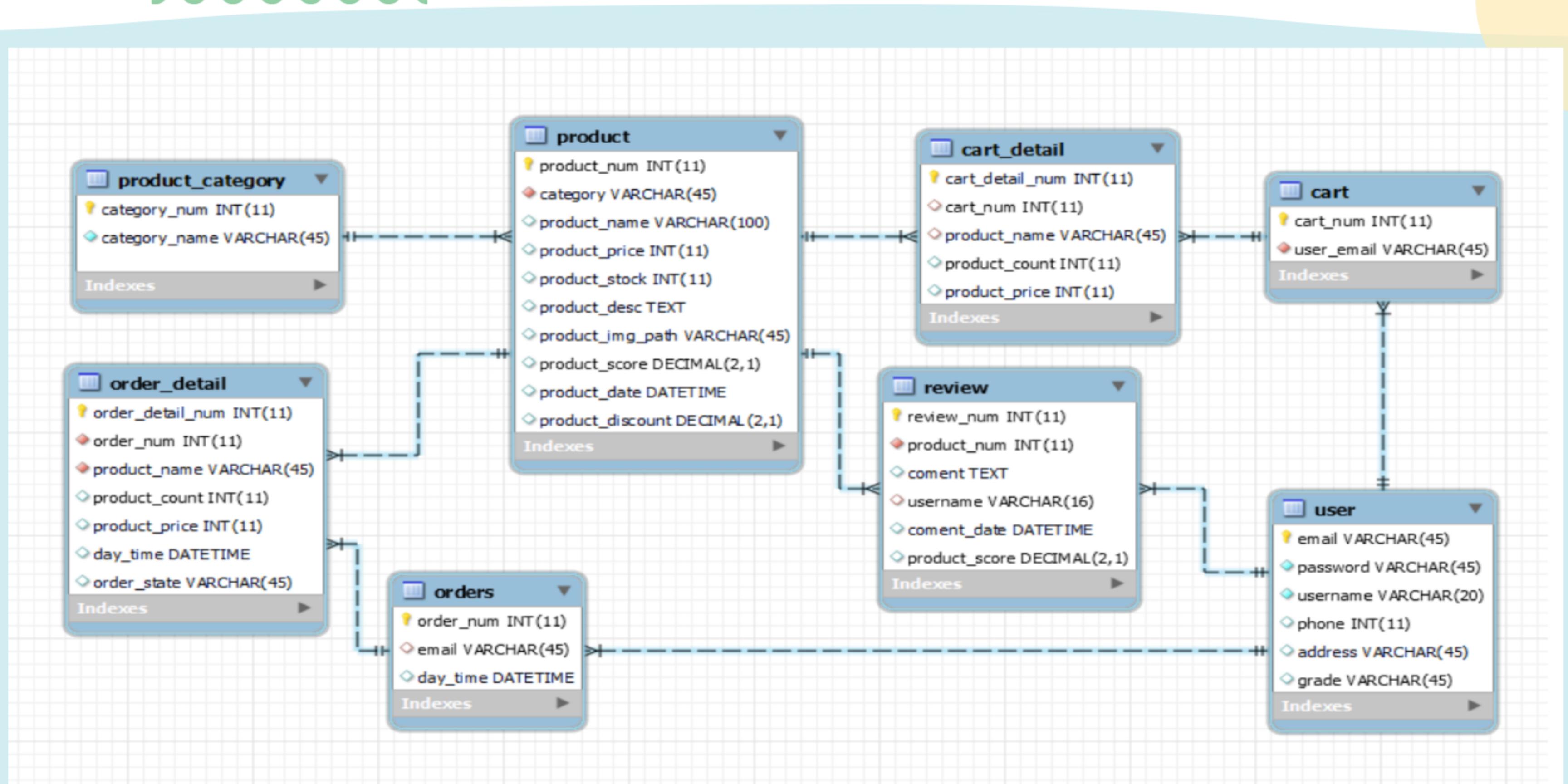
구현 및 테스트

01.18~01.25

발표준비

01.26~01.27

ERD



서비스 설계

회원가입

로그인

제품목록보기

제품상세보기

- 회원정보 수정
- 회원 등급관리(관리자 기능)

- 카테고리별 제품목록
- 제품 검색
- 할인제품 목록
- VIP 제품 목록(VIP 등급 기능)

- 장바구니 담기
- 구매하기
- 리뷰 등록/수정/삭제



장바구니 내역

- 장바구니 내역 확인
- 장바구니 상품 선택구매
- 장바구니 상품 선택삭제

구매내역

- 구매 내역 확인
- 취소 내역 확인
- 주문 취소하기

pom.xml

pom.xml에서는
Spring Security, Spring MVC, JSP,
JSTL 등을 의존성 주입했고,
Maven(라이브러리를 자동으로 다운
로드하거나 프로젝트를 빌드)의 빌드
정보를 담고 있는 파일입니다

여기서 의존성을 주입할 때 버전이 맞
는지 꼭 확인을 해줘야합니다.

<project> : Maven의 XML 네임스페이스를 지정
<properties> : 프로젝트 관련 속성
<dependencies> : 프로젝트가 의존하는 라이브러리
들의 정보

```
1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <groupId>com.code</groupId>
8   <artifactId>bowwow</artifactId>
9   <version>1.0</version>
10  <packaging>war</packaging>
11
12  <name>bowwow</name>
13  <!-- FIXME change it to the project's website -->
14  <!-- <url>http://www.example.com</url>-->
15
16  <properties>
17    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
18
19    <springframework.version>5.0.2.RELEASE</springframework.version>
20    <springsecurity.version>5.0.0.RELEASE</springsecurity.version>
21    <hibernate.version>5.4.4.Final</hibernate.version>
22    <mysql.connector.version>8.0.17</mysql.connector.version>
23    <c3po.version>0.9.5.2</c3po.version>
24
25    <maven.compiler.source>1.7</maven.compiler.source>
26    <maven.compiler.target>1.7</maven.compiler.target>
27  </properties>
28
29  <dependencies>
30
31  <!-- Spring MVC support -->
32  <dependency>
33    <groupId>org.springframework</groupId>
34    <artifactId>spring-webmvc</artifactId>
35    <version>${springframework.version}</version>
36  </dependency>
37
```

persistence-mysql.properties

스프링이 DB에 접속 할 수 있도록 하는
persistence-mysql.properties파일을 생성

```
1 #jdbc connecting pooling                                jdbc connecting
2
3 jdbc.driver=com.mysql.cj.jdbc.Driver
4 jdbc.url=jdbc:mysql://localhost:3306/shopping?useSSL=false&serverTimezone=UTC
5
6 jdbc.user=springstudent
7 jdbc.password=springstudent
8 #
9 # Connection pool properties
10 #
11
12 connection.pool.initialPoolSize=5                      connection pool 설정
13 connection.pool.minPoolSize=5
14 connection.pool.maxPoolSize=20
15 connection.pool.maxIdleTime=3000
16 #
17 # Hibernate properties
18 #
19 hibernate.dialect=org.hibernate.dialect.MySQLDialect
20 hibernate.show_sql=true
21 hibernate.packagesToScan=com.code.bowwow.entity          hibernate와 mysql connecting
```

MainAppConfig

```
26 @Configuration
27 @EnableWebMvc
28 @EnableTransactionManagement
29 @ComponentScan(basePackages="com.code.bowwow")
30 @PropertySource("classpath:persistence-mysql.properties")
31 public class mainAppConfig implements WebMvcConfigurer {
32
33     @Autowired
34     private Environment env;
35
36     @Bean
37     public ViewResolver viewResolver() {
38         InternalResourceViewResolver viewResolver = new InternalResourceViewResolver();
39         viewResolver.setPrefix("/WEB-INF/view/");
40         viewResolver.setSuffix(".jsp");
41
42         return viewResolver;
43     }
44
45     @Bean
46     public DataSource securityDataSource() {
47         ComboPooledDataSource securityDataSource = new ComboPooledDataSource();
48         try {
49             securityDataSource.setDriverClass(env.getProperty("jdbc.driver"));
50         } catch (PropertyVetoException e) {
51             // TODO Auto-generated catch block
52             e.printStackTrace();
53         }
54         securityDataSource.setJdbcUrl(env.getProperty("jdbc.url"));
55         securityDataSource.setUser(env.getProperty("jdbc.user"));
56         securityDataSource.setPassword(env.getProperty("jdbc.password"));
57         securityDataSource.setInitialPoolSizegetIntProperty("connection.pool.initialPoolSize");
58         securityDataSource.setMinPoolSizegetIntProperty("connection.pool.minPoolSize");
59         securityDataSource.setMaxPoolSizegetIntProperty("connection.pool.maxPoolSize");
60         securityDataSource.setMaxIdleTimegetIntProperty("connection.pool.maxIdleTime");
61
62     return securityDataSource;
63 }
```

MainAppConfig

- @Configuration 은 이 자바파일이 설정 파일임을 나타내는 어노테이션

- @EnableWebMvc 는 SpringMVC를 사용하겠다는 어노테이션

- @PropertySource 어노테이션을 사용해 persistence-mysql.properties파일을 읽어들임

springMvcDispatcherServletInitializer

```
1 package com.code.bowwow.config;
2
3 import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;
4
5 public class springMvcDispatcherServletInitializer extends AbstractAnnotationConfigDispatcherServletInitializer{
6     @Override
7     protected Class<?>[] getRootConfigClasses() {
8         // TODO Auto-generated method stub
9         return null;
10    }
11
12    @Override
13    protected Class<?>[] getServletConfigClasses() {
14        return new Class[] { mainAppConfig.class };
15    }
16
17    @Override
18    protected String[] getServletMappings() {
19        return new String[] { "/" };
20    }
21}
22
```

springMvcDispatcherServletInitializer

AbstractAnnotationConfigDispatcherServletInitializer 클래스는 DispatcherServlet 을 프로젝트 내에서 초기화 시키고 Spring MVC 가 프로젝트 내에서 MVC 관련 코드가 쓰였음을 인지하기 위해 상속받는 것이다.

securityWebApplicationInitializer

```
1 package com.code.bowwow.config;  
2  
3 import org.springframework.security.web.context.AbstractSecurityWebApplicationInitializer;  
4  
5 public class securityWebApplicationInitializer extends AbstractSecurityWebApplicationInitializer {  
6  
7 }  
8
```

securityWebApplicationInitializer

Spring Security를 웹에서 사용하려면 Spring Security Filter를 사용할 수 있게 스프링프로젝트에 등록해줘야 하는데, 이걸 가능하게하는 AbstractSecurityWebApplicationInitializer 를 상속 받게 해야한다.

SecurityConfig

```
import javax.sql.DataSource;

@Configuration
@EnableWebSecurity
public class securityConfig extends WebSecurityConfigurerAdapter {

    @Autowired
    private DataSource securityDataSource;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception{
        auth.jdbcAuthentication()
            .dataSource(securityDataSource)
            .usersByUsernameQuery("SELECT email, password, 'true' AS enabled FROM user WHERE email=?") //username:email , enabled:true
            .authoritiesByUsernameQuery("SELECT email, grade FROM user WHERE email=?") // 권한부여
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests() // HttpServletRequest 객체를 통해 request 를 받겠다는 의미
            // -antMatchers() 는 url 을 매칭시키기 위해 사용 -permitAll() 은 모든 유저가 인증 없이 사용 가능
            .antMatchers("/css/**").permitAll() // css 파일 모든 접근 권한 부여
            .antMatchers("/images/**").permitAll() // 이미지 파일 모든 접근 권한 부여
            .antMatchers("/", "/home", "/signup_form", "/signup").permitAll() // home페이지, 회원가입페이지 로그인없이 접근 가능
            .antMatchers("/vip").hasAnyRole("VIP", "ADMIN")
            .antMatchers("/master_form").hasRole("ADMIN")
            .anyRequest().authenticated()
            .and()
            .formLogin() // 기본값으로 제공되는 로그인 form 말고 자체적으로 제작한 login form 을 쓰겠다는 의미
            .loginPage("/login") // 로그인할 페이지 url
            .loginProcessingUrl("/authenticateTheUser") // 사용자로 부터 입력받은 로그인을 처리할 url
            .permitAll()
            .and()
            .logout().permitAll()
            .and()
            .exceptionHandling().accessDeniedPage("/access-denide");
    }

    CharacterEncodingFilter filter = new CharacterEncodingFilter(); // DB에 한글 인코딩
    filter.setEncoding("UTF-8");
    filter.setForceEncoding(true);
    http.addFilterBefore(filter,CsrfFilter.class);
}
}
```

/vip url은 ADMIN과 VIP role을 가진 사용자만 접근하게 해주어 페이지를 나눴고,
/master_form url은 ADMIN role을 가진 사용자만 접근하게 페이지를 나눴다.

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="security" uri="http://www.springframework.org/security/tags" %>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
4 <html>
5 <body>
6     <div class="container">
7         <c:set var="username" value="${username}" />
8         <!-- 메인 로고 -->
9         <div style="margin:50px auto 30px auto; text-align:center;" >
10            <a href="${pageContext.request.contextPath}"/></a>
11            <br><br>
12            ${username}님 권한이 없습니다
13
14     </div>
15
16
17
18
19 </div>
20 </body>
21 </html>
```

access-denide.jsp



멍멍님 권한이 없습니다

다른 role을 가진 사용자가 접근 했을 시 그냥 접근제한 오류페이지가 나오는 것을 방지하기 위해
exceptionHandling().accessDeniedPage("/access-denide") 를 사용해 access-denide페이지가 나오게
설정 해준다

serviceController 부분설명

"/review_add"

1

```
<!-- 리뷰 입력 -->
<form:form action="review_add" Method="POST">
    <input type="hidden" name="product_num" value="${product_num}" />
    <h4>별점을 선택해주세요</h4>
    <div class="star-rating">
        <input type="radio" id="5-stars" name="product_score" value="5.0" />
        <label for="5-stars" class="star">&#9733;</label>
        <input type="radio" id="4-stars" name="product_score" value="4.0" />
        <label for="4-stars" class="star">&#9733;</label>
        <input type="radio" id="3-stars" name="product_score" value="3.0" />
        <label for="3-stars" class="star">&#9733;</label>
        <input type="radio" id="2-stars" name="product_score" value="2.0" />
        <label for="2-stars" class="star">&#9733;</label>
        <input type="radio" id="1-star" name="product_score" value="1.0" required/>
        <label for="1-star" class="star">&#9733;</label>
    </div>
    <input type="text" class="form-control" name="coment" maxlength='100' placeholder="평점은 최대 100자까지 등록 가능합니다." required/>
    <button type="submit" class="badge badge-pill badge-secondary" onclick="if (!(confirm('등록하시겠습니까'))) return false">리뷰등록</button>
</form:form>
</div>
```

COMMENT

순번	아이디	댓글내용	평점	등록일시
1	admin44		★★★ 3.0	2023-01-22 20:30:40

별점을 선택해주세요



평점은 최대 100자까지 등록 가능합니다.

리뷰등록

detail.jsp 의 리뷰 등록부분.

"/review_add"

<form:form> 태그를 쓴 이유:

Spring Security는 기본적으로 요청에 CSRF가 들어있지 않으면 에러가 뜨는데, 여기서 <form>태그를 쓰면 토큰을 직접 랜덤생성하는 코드를 짜서 넣어줘야하지만 <form:form>는 Spring Security가 자동으로 토큰을 만들어주기 때문에 사용했다

serviceController 부분설명

/review_add ②

```
//리뷰주가
@PostMapping("/review_add")
public String review_add(Principal p, review review, product product) {
    List<user> theUser = bowwowService.getUser(p.getName()); //사용자의 닉네임 받아오기
    review.setUsername(theUser.get(0).getUserName());
    bowwowService.addReview(review); //입력받은 리뷰 및 평점 업데이트
    double averageScore = bowwowService.calAverageScore(product.getProduct_num()); //해당제품의 평점 출력
    bowwowService.avgScore(averageScore, product.getProduct_num()); //출력받은 제품의 평점 업데이트

    return "redirect:/detail?productNum="+review.getProduct_num();
}
```

리뷰 등록하는 부분

```
@Override
public void addReview(review review) {
    Session curSession = sessionFactory.getCurrentSession();
    curSession.saveOrUpdate(review);
}
```

1. addReview 메서드

받아온 값들을 전부 saveOrUpdate하여 review테이블에 넣어준다

```
@Override
public double calAverageScore (int productNum) {
    double score;
    try {
        Session curSession = sessionFactory.getCurrentSession();
        Query<Double> theQuery = curSession.createQuery("select avg(product_score) from review where product_num=:product_num",Double.class);
        theQuery.setParameter("product_num", productNum);

        score = theQuery.getSingleResult();
    }catch(Exception e) {
        score = 0.0;
    }
    return score;
}
```

2. calAverageScore메서드

productNum을 받아와 review 테이블에 있는 product_score의 평균값을
불러오는 메서드이고 초기값을 0.0으로 설정해준다

```
@Override
public void avgScore(double averageScore, int proNum) {
    Session curSession = sessionFactory.getCurrentSession();
    Query theQuery = curSession.createQuery("UPDATE product SET product_score= :avgScore WHERE product_num = :productNum");
    theQuery.setParameter("avgScore", averageScore);
    theQuery.setParameter("productNum", proNum);

    theQuery.executeUpdate();
}
```

3. avgScore메서드

위의 두개 메서드가 실행된 후 받아온 productNum 의 받아온 product_score 를
업데이트 해준다.

Review ~~~~~

Error page



감사합니다

