

제가 진행했던 프로젝트들입니다!

프로젝트 경험 소개

1) 개인 미니 프로젝트 - 커피 쇼핑몰

2-1) 팀 프로젝트 - 영화 커뮤니티 사이트 (JSP MVC 모델 2)

2-2) 개인 프로젝트 - 영화 커뮤니티 사이트 (Spring MVC 2)

프로젝트 경험 1) 개인 미니 프로젝트

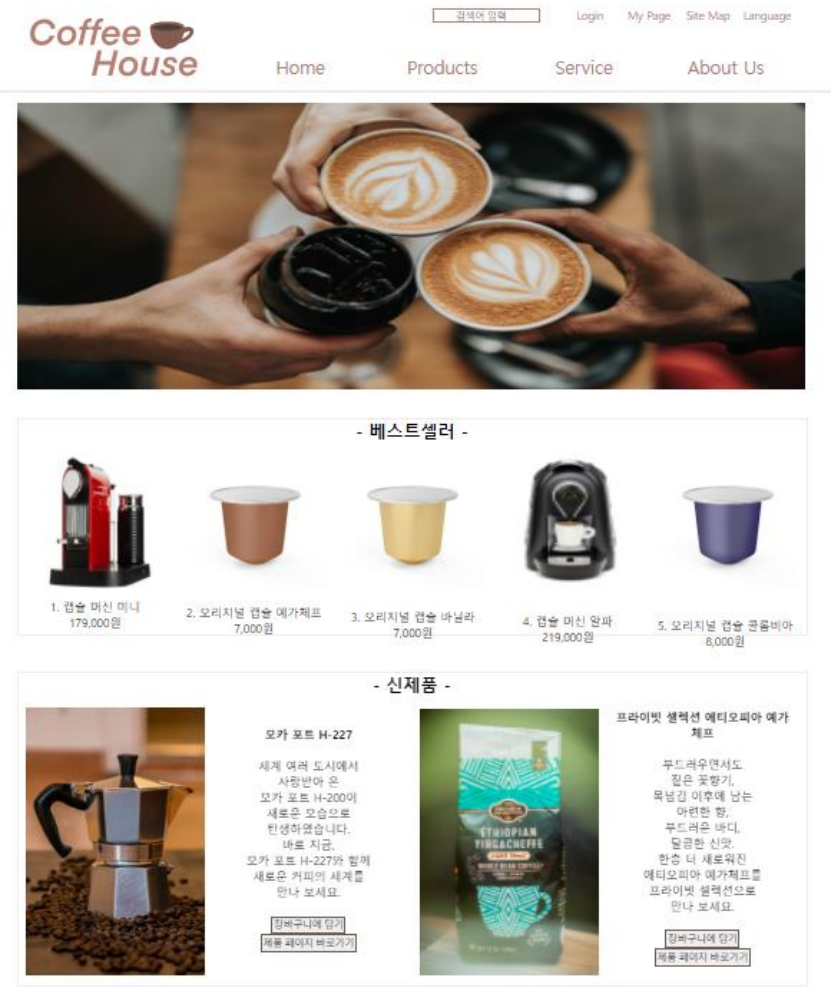
Coffee House 커피 쇼핑몰

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 목표 | 화면구현 및 JSP MVC 모델 2를 능숙하게 사용할 수 있는 것 |
| 개발환경 | 개발 언어 : HTML5, CSS3, JavaScript, Java 8, JSP 라이브러리 : Bootstrap 3.5 웹 서버 : Apache Tomcat v9.0 개발도구 : Eclipse 2021-12 DB : Oracle 11g 운영체제 : Windows 10 기타 도구 : SQL Plus, DBeaver, EditPlus, Procreate |
| 프로젝트 기간 | 화면구현 1 : 2022.01.08 - 2022.01.17 화면구현 2(Bootstrap 적용) : 2022.02.16 - 2022.02.21 JSP 모델 1 : 2022.02.03 - 2022.02.23 JSP 모델 2 : 2022.02.23 - 2022.02.27 |
| 주요 기능 | 회원가입, 로그인, 회원정보 조회 및 수정, 글 작성, 글 수정, 글 삭제, 댓글 조회 |
| 구현 정도 | 60% |

프로젝트 경험 1) 개인 미니 프로젝트

커피 쇼핑몰

화면 구현, JSP 모델1 로그인·게시판 기능 구현을 위해 진행했던 미니 프로젝트입니다.



프로젝트 경험 2-1) 팀 프로젝트 (JSP MVC 모델 2)

Movie So Good 영화 커뮤니티 사이트

| | |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 목표 | 모든 기능들을 온전히 자신의 것으로 만들어 모든 팀원이 능숙한 개발자가 되는 것 |
| 개발환경 | 개발 언어 : HTML5, CSS3, JavaScript, Java 8, JSP 라이브러리 : Bootstrap 3.5 웹 서버 : Apache Tomcat v9.0 개발도구 : Eclipse 2021-12 DB : Oracle 11g 운영체제 : Windows 10, Mac OS 협업 툴 : Zoom 기타 도구 : SQL Plus, DBeaver, eXERD, EditPlus, Procreate |
| 프로젝트 기간 | 2022.01.24 - 2022.04.07 |
| 특수 기능 | 회원 등급제, 장르·카테고리별 게시판, 핫 게시판, 내가 쓴 글 모아 보기, 스크랩 기능, 영화 별점 및 좋아요 기록 |
| 구현 정도 | 100% |

-기여도

| | |
|-----------|-----|
| 기획 | 100 |
| 디자인 | 100 |
| 개발(기능 구현) | 150 |

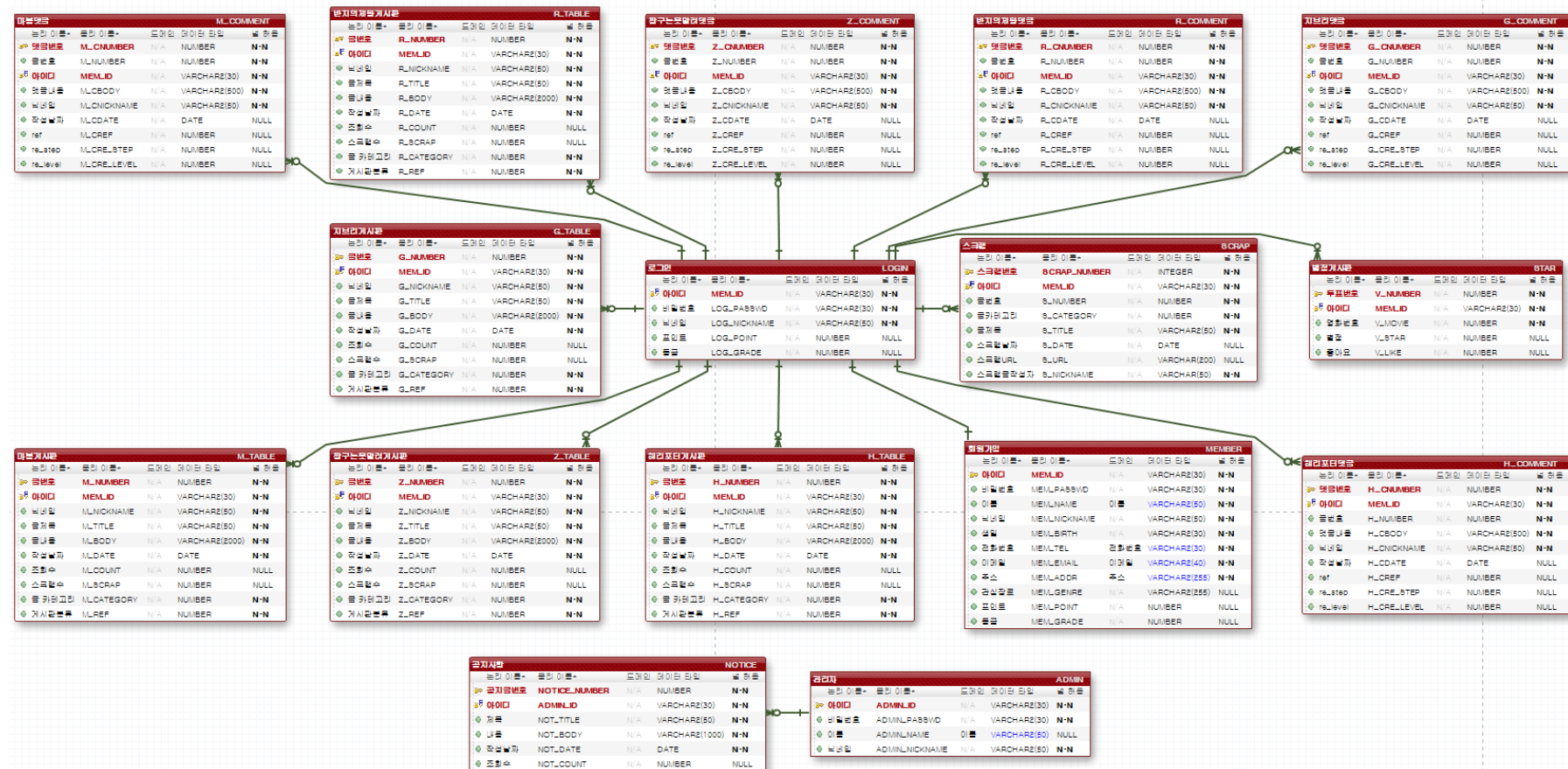
-주요 기여 부분

프로젝트 주제 제시 및 주제 구체화
로고, 게시판 화면 등 전체적인 디자인
모든 특수 기능 구현
오류 발견 및 수정
팀원 도움

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

설계 - 모델링



커뮤니티 사이트 구현을 위한
모델링을 진행하였습니다.

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용 - 화면

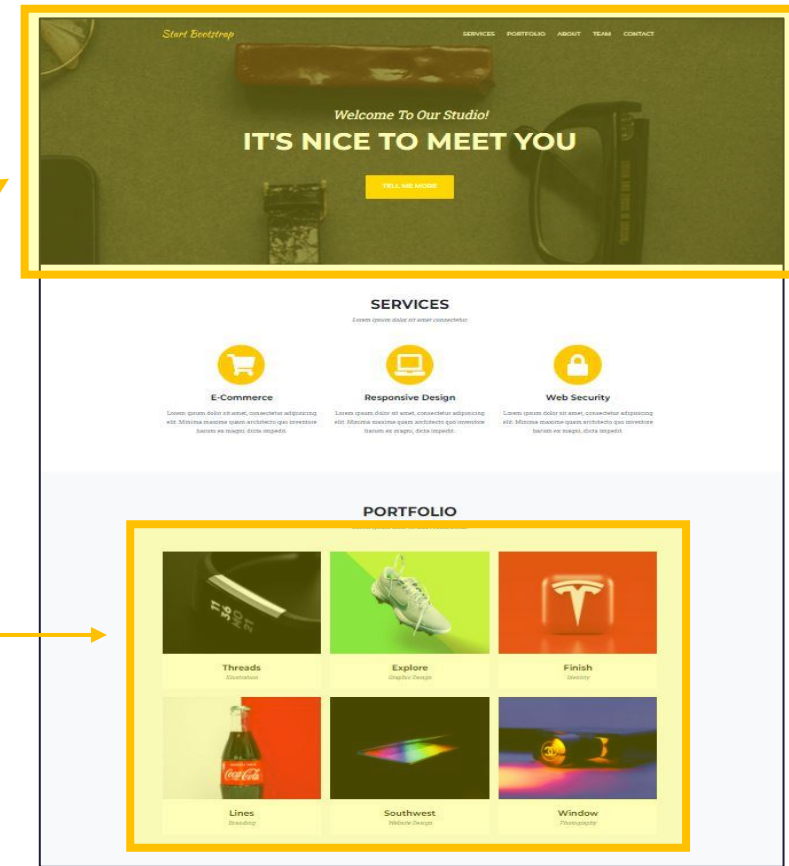
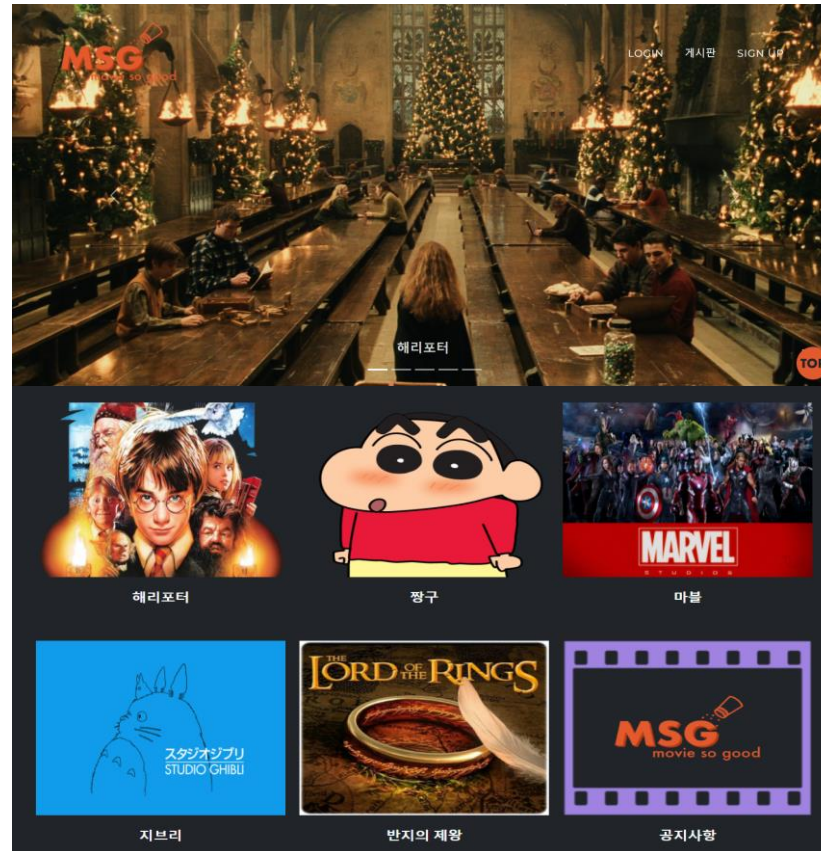
디자인 벤치마킹 - 메인 화면

부트스트랩에서 제공하는 템플릿이

저희가 진행하고자 했던 디자인과 비슷해
소스를 수정하여 메인 화면을 디자인하였습니다.

URL :

<https://startbootstrap.com/previews/agency>



프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용 - 화면

```
<!-- Favicon-->
<link rel="icon" type="image/x-icon" href="assets/favicon2.ico" />
<!-- Font Awesome icons (free version)-->
<script src="https://use.fontawesome.com/releases/v5.15.4/js/all.js" crossorigin="anonymous"></script>
<!-- Google fonts-->
<link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css" />
<link href="https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700" rel="stylesheet" type="text/css" />
<!-- Core theme CSS (includes Bootstrap)-->
<link href="css/styles.css" rel="stylesheet" />
<style>
@import url(css/styles2.css);
</style>
```

부트스트랩에서 제공하는 템플릿의 버전은 5인데, 제가 배웠던 3버전의 코드들과는 조금씩 달랐기 때문에

기본적으로는 5버전의 css 파일(styles.css)을 <link> 태그 안에서 삽입하였고,

styles2.css파일을 생성하여 다르다고 생각되는 필요한 코드들을 3버전에서 옮겼습니다.

그리고 3버전의 css코드를 더 우선해서 적용하기 위해 <style> 태그 안에 @import문을 이용해 삽입하였습니다.

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용 - 특수 기능

장르·카테고리별 게시판

검색결과까지 보여줄 수 있으면서,

자유, 정보 등의 카테고리(h_ref)가 나뉘어지지 않은 상태의

글 목록 출력을 위한 메서드를 작성하였고,

카테고리별로 나뉜 상태의 글 목록과

조회수(h_count)가 50 이상인 인기글 목록 출력을 위해

where조건을 추가한 메서드들 또한 작성하였습니다.

```
public List<HBoardDTO> getBoardArticles(int start, int end, String search, String searchtext) {
    List<HBoardDTO> articleList = null;
    try {
        con = pool.getConnection();
        if (search == null || search == "") {
            sql = "select * from (select h_table.*, rownum as rn from (select * from h_table order by h_number desc) h_table) where rn >= ? and rn <= ?";
        } else {
            if (search.equals("h_title_h_body")) {
                sql = "select * from (select h_table.*, rownum as rn from (select * from h_table "
                    + "where h_title like '%" + searchtext + "%' or h_body like '%" + searchtext
                    + "%') h_table) where rn >= ? and rn <= ?";
            } else {
                sql = "select * from (select h_table.*, rownum as rn from (select * from h_table " + " where "
                    + search + " like '%" + searchtext + "%') h_table) where rn >= ? and rn <= ?";
            }
        }
        System.out.println("getBoardArticles()의 sql=>" + sql);
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, start);
        pstmt.setInt(2, end);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            articleList = new ArrayList(end);
            do {
                HBoardDTO article = this.makeArticleFromResult();
                articleList.add(article);
            } while (rs.next());
        }
    } catch (Exception e) {
        System.out.println("getBoardArticles() 에러 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return articleList;
}
```

```
public List<HBoardDTO> getBoardArticles(int start, int end, String search, String searchtext, String h_ref) {

    where h_ref=" + h_ref
```

```
public List<HBoardDTO> getHotBoardArticles(int start, int end, String search, String searchtext) {

    where h_count > 50
```

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용 - 특수 기능

```
<ul class="nav nav-pills nav-justified">
<li class="col-md-3"><a href="H_List.do?h_ref=1">자유</a></li>
<li class="col-md-3"><a href="H_List.do?h_ref=2">정보</a></li>
<li class="col-md-3"><a href="H_Rate.do">별점</a></li>
<li class="col-md-3"><a href="H_List.do?h_ref=4">HOT</a></li>
</ul>
```

| 자유 | | 정보 | 별점 | HOT | |
|----|----------------------------------------|-------|----------------|-----|--|
| 번호 | 제목 | 작성자 | 작성일 | 조회 | |
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 | |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 25 | |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 | |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 17 | |
| 61 | [아프카병의 죄수] [자유] 61번 글 (2) | 바스드ㄱ | 22.03.31 19:47 | 16 | |

장르·카테고리별 게시판

자유, 정보, HOT 탭에 번호를 부여했고

글 목록 액션 클래스에서 각 탭의 h_ref 파라미터 값을 받아와

Null이라면 기본 메서드,

1이나 2라면 자유나 정보 카테고리를 출력해주는 href메서드, 4라면 인기글 목록을 출력해주는 hot 메서드를 호출할 수 있게 하였습니다.

```
HBoardDAO dbPro=new HBoardDAO();
```

```
String h_ref=request.getParameter("h_ref");
int count=0;
if (h_ref==null) {
    count=dbPro.getArticleSearchCount(search, searchText);
}else if (Integer.parseInt(h_ref)==1 || Integer.parseInt(h_ref)==2) {
    count=dbPro.getArticleSearchCount(search, searchText, h_ref);
}else if (Integer.parseInt(h_ref)==4) {
    count=dbPro.getHotArticleSearchCount(search, searchText);
}
```

```
Hashtable<String, Integer> pgList=dbPro.pgList(pageNum, count);
```

```
List<HBoardDTO> articleList=null;
```

```
if (count > 0) {
    System.out.println(pgList.get("startRow")+"", "+pgList.get("endRow"));
    if (h_ref==null){
        articleList=dbPro.getBoardArticles(pgList.get("startRow"), pgList.get("endRow"), search, searchText);
    }else if (Integer.parseInt(h_ref)==1 || Integer.parseInt(h_ref)==2) {
        articleList=dbPro.getBoardArticles(pgList.get("startRow"), pgList.get("endRow"), search, searchText, h_ref);
    }else if (Integer.parseInt(h_ref)==4) {
        articleList=dbPro.getHotBoardArticles(pgList.get("startRow"), pgList.get("endRow"), search, searchText);
    }
    System.out.println("H_ListAction의 articleList=>"+articleList);
}else {
    articleList=Collections.EMPTY_LIST;
}
```

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용 - 특수 기능

```
public int getCommentCountForList(int h_number) throws Exception {
```

```
    int x = 0;
```

각 글에 해당하는 댓글 수를 구하는 메서드

```
    con = pool.getConnection();
    System.out.println("con=>" + con);
    sql = "select count(*) from h_comment where h_number=? and h_cbody!='삭제된 댓글입니다.&nbsp;';";
    pstmt = con.prepareStatement(sql);
    pstmt.setInt(1, h_number);
    rs = pstmt.executeQuery();
    if (rs.next()) {
        x = rs.getInt(1);
    }
    pool.freeConnection(con, pstmt, rs);
    return x;
}
```

```
private HBoardDTO makeArticleFromResult() throws Exception {
```

글 번호에 따라서 구해진

댓글 수를 담는 메서드

```
    HBoardDTO article = new HBoardDTO();
    HBoardDAO hboard = new HBoardDAO();
    article.setH_number(rs.getInt("h_number"));
    article.setMem_id(rs.getString("mem_id"));
    article.setH_nickname(rs.getString("h_nickname"));
    article.setH_title(rs.getString("h_title"));
    article.setH_body(rs.getString("h_body"));
    article.setH_date(rs.getTimestamp("h_date"));
    article.setH_count(rs.getInt("h_count"));
    article.setH_scrap(rs.getInt("h_scrap"));
    article.setH_category(rs.getInt("h_category"));
    article.setH_ref(rs.getInt("h_ref"));
    article.setH_comments(hboard.getCommentCountForList(rs.getInt("h_number")));
```

```
    return article;
```

```
}
```

JSP MVC 모델 2에서만 구현 가능했던 댓글 수 출력 기능입니다.

```
public List<HBoardDTO> getBoardArticles(int start, int end, String search, String searchText) {
    List<HBoardDTO> articleList = null;
    try {
        con = pool.getConnection();
        if (search == null || search == "") {
            sql = "select * from (select h_table.*, rownum as rn from (select * from h_table order by h_number desc) h_table) where rn >= ? and rn <= ?";
        } else { // 제목+본문
            if (search.equals("h_title_h_body")) { // 제목+본문
                sql = "select * from (select h_table.*, rownum as rn from (select * from h_table "
                    + "where h_title like '%" + searchText + "%' or h_body like '%" + searchText
                    + "%' ) h_table) where rn >= ? and rn <= ?";
            } else {
                sql = "select * from (select h_table.*, rownum as rn from (select * from h_table " + " where "
                    + search + " like '%" + searchText + "%' ) h_table) where rn >= ? and rn <= ?";
            }
        }
        System.out.println("getBoardArticles()의 sql=>" + sql);
        // -----
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, start); // mysql은 start-1
        pstmt.setInt(2, end); // end-1??
        rs = pstmt.executeQuery();
        if (rs.next()) {
            articleList = new ArrayList(end);
            do {
                HBoardDTO article = this.makeArticleFromResult();
                articleList.add(article);
            } while (rs.next());
        }
    } catch (Exception e) {
        System.out.println("getBoardArticles() 에러 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return articleList;
}
```

글 목록을 구하는 메서드에서

구한 댓글 수를 담는 메서드를 사용

```
<td width="51%" align="left">
    <c:choose>
        <c:when test="${article.h_ref==1}">
            <font style="color:gray;">[자유]&nbsp;&nbsp;&nbsp;</font>
        </c:when>
        <c:when test="${article.h_ref==2}">
            <font style="color:gray;">[정보]&nbsp;&nbsp;&nbsp;</font>
        </c:when>
    </c:choose>
    <a href="H_Content.do?h_number=${article.h_number}&pageNum=${pgList.currentPage}">
        ${article.h_title}&nbsp;&nbsp;&nbsp;({${article.h_comments}})
    </a>
</td>
```

글 목록 페이지에 댓글 수를 출력

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

구현 내용

- 특수 기능

| 공지사항 | | | | |
|------|----------------------------------------|-------|----------------|----|
| 번호 | 제목 | 작성자 | 작성일 | 조회 |
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 26 |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 18 |

자유

정보

별첨

HOT

| 번호 | | 제목 | 작성자 | 작성일 | 조회 |
|----|---------------|----------------------------------------|-------|----------------|----|
| 4 | | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 25 |
| 2 | | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 17 |
| 61 | [아프카반의 죄수] | [자유] 61번 글 (2) | 바스드그 | 22.03.31 19:47 | 16 |
| 60 | [불의 잔] | [자유] ㅇ (1) | 다노르 | 22.03.30 03:50 | 14 |
| 59 | [불의 잔] | [자유] asdf (13) | 관리자 | 22.03.25 16:57 | 70 |
| 58 | [비밀의 방] | [자유] qwer (4) | 바스드그 | 22.03.23 20:14 | 31 |
| 57 | [불의 잔] | [자유] 포인트 메서드 수정 후 글쓰기 테스트 (2) | 레호너 | 22.03.23 19:18 | 24 |
| 56 | [불의 잔] | [자유] 포인트, 등급 관련해서 (1) | 크레타프 | 22.03.23 02:48 | 8 |
| 55 | [죽음의 성물 - 2부] | [자유] 다노르 (1) | 크레타프 | 22.03.23 02:08 | 8 |
| 54 | [죽음의 성물 - 1부] | [자유] 바스드그 (0) | 크레타프 | 22.03.23 02:08 | 6 |
| 53 | [불의 잔] | [자유] 별점을 매겨야 좋아요를 누를 수 있게 했음 (1) | 크레타프 | 22.03.23 02:03 | 13 |
| 52 | [아프카반의 죄수] | [자유] 별점 좋아요 문제 하나 (0) | 레호너 | 22.03.23 01:53 | 5 |

모든 글

1234567

글쓰기

제목검색

앞 슬라이드의 메서드들을 이용해
각 댓글 수가 함께 출력되는
모든 장르 게시판과 공지사항 게시판의 목록들을 구현하였습니다.

| 자유 | 정보 | 별첨 | HOT | |
|----|----------------------------------------|-------|----------------|----|
| 번호 | 제목 | 작성자 | 작성일 | 조회 |
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 26 |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 18 |
| 9 | [아프카반] [자유] adf (1) | 바스드그 | 22.03.31 19:26 | 3 |
| 8 | [아프카반의 죄수] [자유] 1234 (0) | 바스드그 | 22.03.17 02:39 | 7 |
| 7 | [자유] 문헌 합니다 (0) | 예전미연 | 22.03.17 02:18 | 2 |
| 6 | [불의 잔] [자유] 댓글 테스트 m (4) | 바스드그 | 22.03.17 00:58 | 9 |
| 5 | [불의 잔] [자유] 스크랩-m2 (0) | 바스드그 | 22.03.16 22:59 | 8 |
| 4 | [아프카반의 죄수] [자유] werw (0) | 레호너 | 22.03.15 22:33 | 2 |
| 3 | [불의 잔] [자유] rerere (0) | 레호너 | 22.03.15 22:33 | 1 |
| 2 | [아프카반] [자유] 마블 글 2 (0) | 바스드그 | 22.03.15 00:25 | 9 |
| 1 | [아프카반] [자유] 마블 글 1 (0) | 바스드그 | 22.03.15 00:25 | 7 |

1

글쓰기

제목검색

자유

정보

별첨

HOT

| 번호 | 제목 | 작성자 | 작성일 | 조회 |
|----|----------------------------------------|-------|----------------|----|
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 26 |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 18 |
| 8 | [아프카의 죄수] [자유] dfg (2) | 바스드그 | 22.03.31 19:16 | 7 |
| 7 | [영화 리뷰] [자유] 1234 (0) | 바스드그 | 22.03.17 02:39 | 5 |
| 6 | [아프카의 죄수] [자유] 댓글 테스트 g (2) | 바스드그 | 22.03.17 00:58 | 9 |
| 5 | [아프카의 죄수] [자유] 스크랩-g2 (0) | 바스드그 | 22.03.16 22:59 | 3 |
| 4 | [불의 잔] [자유] rew (0) | 레호너 | 22.03.15 22:34 | 2 |
| 3 | [아프카의 죄수] [정보] erer (0) | 바스드그 | 22.03.15 16:16 | 0 |
| 2 | [아프카의 죄수] [자유] 1234 (0) | 바스드그 | 22.03.15 16:16 | 0 |
| 1 | [영화 리뷰] [자유] 지브라 글 1 (0) | 바스드그 | 22.03.15 01:57 | 12 |

1

글쓰기

제목검색

자유

정보

별첨

HOT

| 번호 | 제목 | 작성자 | 작성일 | 조회 |
|----|----------------------------------------|-------|----------------|----|
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 26 |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 18 |
| 8 | [영화 리뷰] [정보] DFG (0) | 바스드그 | 22.03.31 17:47 | 6 |
| 7 | [영화 리뷰] [자유] 다른 게시판도 포맷트 적용 되는지 (1) | 다노르 | 22.03.19 01:59 | 7 |
| 6 | [영화 리뷰] [자유] 1234 (0) | 바스드그 | 22.03.17 02:39 | 2 |
| 5 | [영화 리뷰] [정보] 댓글 테스트 z (2) | 바스드그 | 22.03.17 00:58 | 17 |
| 4 | [자유] 스크랩-z1 (1) | 바스드그 | 22.03.16 22:58 | 11 |
| 3 | [영화 리뷰] [자유] scrap (0) | 레호너 | 22.03.16 02:44 | 4 |
| 2 | [영화 리뷰] [자유] qwerqwe (0) | 레호너 | 22.03.15 22:33 | 6 |
| 1 | [영화 리뷰] [자유] 랑구 글 1 (0) | 바스드그 | 22.03.10 16:48 | 12 |

1

글쓰기

제목검색

자유

정보

별첨

HOT

| 번호 | 제목 | 작성자 | 작성일 | 조회 |
|----|----------------------------------------|-------|----------------|----|
| 4 | [공지] < MSG : movie so good > | admin | 22.03.17 01:51 | 12 |
| 3 | [공지] < MSG : movie so good > 이용수칙 안내 | admin | 22.03.11 01:50 | 26 |
| 2 | [공지] < MSG : movie so good > SNS 오픈 안내 | admin | 22.03.11 01:50 | 2 |
| 1 | [공지] < MSG:movie so good > 그랜드 오픈 안내 | admin | 22.03.09 03:58 | 18 |
| 5 | [영화 리뷰] [자유] 1234 (0) | 바스드그 | 22.03.17 02:39 | 4 |
| 4 | [영화 리뷰] [자유] 댓글 테스트 r (1) | 바스드그 | 22.03.17 00:58 | 12 |
| 3 | [영화 리뷰] [자유] 스크랩-r2 (0) | 바스드그 | 22.03.16 22:59 | 4 |
| 2 | [영화 리뷰] [자유] 스크랩-r1 (0) | 바스드그 | 22.03.16 22:59 | 3 |
| 1 | [영화 리뷰] [자유] weewe (0) | 레호너 | 22.03.15 22:34 | 4 |

1

글쓰기

제목

검색

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능

```
if (insert > 0) {
    String sql2 = "update login set log_point=log_point+10 where mem_id=?";
    pstmt = con.prepareStatement(sql2);
    pstmt.setString(1, article.getMem_id());

    int pointup = pstmt.executeUpdate();
    con.commit();

    if (pointup > 0) {
        String sql3 = "select log_point, log_grade from login where mem_id=?";
        pstmt = con.prepareStatement(sql3);
        pstmt.setString(1, article.getMem_id());
        rs = pstmt.executeQuery();
        if (rs.next()) {
            point = rs.getInt(1);
            grade = rs.getInt(2);

            String sql4 = "";
            if (point < 100) {
                sql4 = "update login set log_grade=1 where mem_id=?";
            } else if (point >= 100 && point < 500) {
                sql4 = "update login set log_grade=2 where mem_id=?";
            } else if (point >= 500 && point < 2000) {
                sql4 = "update login set log_grade=3 where mem_id=?";
            } else if (point >= 2000 && point < 10000) {
                sql4 = "update login set log_grade=4 where mem_id=?";
            } else if (point >= 10000) {
                sql4 = "update login set log_grade=5 where mem_id=?";
            }
            pstmt = con.prepareStatement(sql4);
            pstmt.setString(1, article.getMem_id());

            int gradeup = pstmt.executeUpdate();
            con.commit();
        }
    }
}
```

회원 등급제

회원의 포인트에 따라 등급을 5개로 나누었으며

글 작성시 10포인트 적립, 삭제시 10포인트 회수되게 하였고,

댓글의 경우 5포인트 적립, 5포인트 회수되게 하였습니다.

또한 포인트 변동시 등급도 바로 변경될 수 있게 하였습니다.

```
if (delete1 > 0) {
    String sql4 = "update login set log_point=log_point-10 where mem_id=?";
    pstmt = con.prepareStatement(sql4);
    pstmt.setString(1, h_mem_id);

    int pointdown = pstmt.executeUpdate();
    con.commit();

    if (pointdown > 0) {
        String sql5 = "select log_point, log_grade from login where mem_id=?";
        pstmt = con.prepareStatement(sql5);
        pstmt.setString(1, h_mem_id);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            point = rs.getInt(1);
            grade = rs.getInt(2);

            if (point < 100) {
                sql = "update login set log_grade=1 where mem_id=?";
            } else if (point >= 100 && point < 500) {
                sql = "update login set log_grade=2 where mem_id=?";
            } else if (point >= 500 && point < 2000) {
                sql = "update login set log_grade=3 where mem_id=?";
            } else if (point >= 2000 && point < 10000) {
                sql = "update login set log_grade=4 where mem_id=?";
            } else if (point >= 10000) {
                sql = "update login set log_grade=5 where mem_id=?";
            }
            System.out.println("sql=>" + sql);
            pstmt = con.prepareStatement(sql);
            pstmt.setString(1, h_mem_id);

            int gradeup = pstmt.executeUpdate();
            con.commit();
        }
    }
}
```

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능

```
private BoardDTO makeHArticleFromResult() throws Exception {

    BoardDTO article = new BoardDTO();

    article.setNumber(rs.getInt("h_number"));
    article.setMem_id(rs.getString("mem_id"));
    article.setNickname(rs.getString("h_nickname"));
    article.setTitle(rs.getString("h_title"));
    article.setBody(rs.getString("h_body"));
    article.setDate(rs.getTimestamp("h_date"));
    article.setCount(rs.getInt("h_count"));
    article.setScrap(rs.getInt("h_scrap"));
    article.setCategory(rs.getInt("h_category"));
    article.setRef(rs.getInt("h_ref"));

    HBoardDAO hboard=new HBoardDAO();
    article.setComments(hboard.getCommentCountForList(rs.getInt("h_number")));

    return article;
}
```

•
•
•

이러한 메서드를 장르별로 모두 만들어 글 목록 메서드의 길이를 줄였습니다.

내가 쓴 글 모아 보기 기능

매개변수로 넘어온 장르의 값에 따라 그 장르별 글의 수와 목록을 구하고 답는 메서드를 작성하였습니다.

```
public List<BoardDTO> getMyArticles(int start, int end, String init, String mem_id) {
    List<BoardDTO> articleList = null;
    try {
        con = pool.getConnection();
        sql = "select * from (select " + init + "_table.*, rownum as rn from (select * from "
            + init + "_table where mem_id='" + mem_id + "' order by " + init + "_number desc) "
            + init + "_table) where rn >= ? and rn <= ?";
        pstmt = con.prepareStatement(sql);
        pstmt.setInt(1, start);
        pstmt.setInt(2, end);
        rs = pstmt.executeQuery();
        if (rs.next()) {
            articleList = new ArrayList(end);
            if (init.equals("h")) {
                do {
                    BoardDTO article = this.makeHArticleFromResult();
                    articleList.add(article);
                } while (rs.next());
            } else if (init.equals("z")) {
                do {
                    BoardDTO article = this.makeZArticleFromResult();
                    articleList.add(article);
                } while (rs.next());
            } else if (init.equals("m")) {
                do {
                    BoardDTO article = this.makeMArticleFromResult();
                    articleList.add(article);
                } while (rs.next());
            } else if (init.equals("g")) {
                do {
                    BoardDTO article = this.makeGArticleFromResult();
                    articleList.add(article);
                } while (rs.next());
            } else if (init.equals("r")) {
                do {
                    BoardDTO article = this.makeRArticleFromResult();
                    articleList.add(article);
                } while (rs.next());
            }
        }
    } catch (Exception e) {
        System.out.println("getMyArticles() 예외 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return articleList;
}
```

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능

내가 쓴 글 모아 보기 기능

각 장르 탭에 이니셜로 값을 주고,

액션클래스에서 장르 파라미터 값을 받아온 후 메서드를 호출해

각 장르의 글의 수와 목록이 출력되도록 하였습니다.

```
<ul class="nav nav-pills nav-justified">
<li class="col-md-1"></li>
<li class="col-md-2"><a href="MyArticles.do?genre=h">해리포터</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=z">짱구</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=m">마블</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=g">지브리</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=r">반지의 제왕</a></li>
<li class="col-md-1"></li>
</ul>
```

```
String genre=request.getParameter("genre");
```

```
int count=0;
List<BoardDTO> articleList=null;
```

```
BoardDAO dbPro=new BoardDAO();
```

```
if (genre==null) {
    count=0;
}else {
    count=dbPro.getMyArticleCount(genre, mem_id);
}
```

```
Hashtable<String, Integer> pgList=dbPro.pgList(pageNum, count);
```

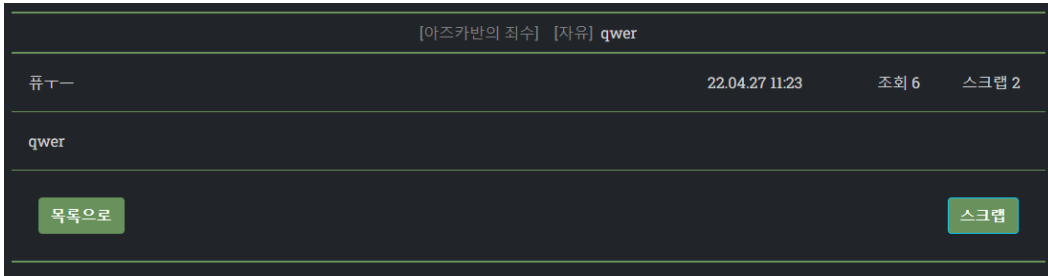
```
if (count > 0) {
    System.out.println(pgList.get("startRow")+" ", "+pgList.get("endRow"));
    articleList=dbPro.getMyArticles(pgList.get("startRow"), pgList.get("endRow"), genre, mem_id);
}else {
    articleList=Collections.EMPTY_LIST;
}
```

| 내가 쓴 글 | | | | | |
|-------------------------------------------|---------------|-------------------|------|----------------|----|
| | | | | | |
| 해리포터 짱구 마블 지브리 반지의 제왕 | | | | | |
| 번호 | 제목 | | 작성자 | 작성일 | 조회 |
| 7 | [아이언맨] | [자유] 마블 글 (0) | ㅂㅈㄷㄱ | 22.04.20 17:34 | 12 |
| 6 | [아이언맨] | [자유] sdfrr (0) | ㅂㅈㄷㄱ | 22.03.31 19:26 | 17 |
| 5 | [가디언즈 오브 갤럭시] | [자유] 1234 (0) | ㅂㅈㄷㄱ | 22.03.17 02:39 | 9 |
| 4 | [블랙팬서] | [자유] 댓글 테스트 m (0) | ㅂㅈㄷㄱ | 22.03.17 00:58 | 9 |
| 3 | [블랙팬서] | [자유] 스크랩-m2 (0) | ㅂㅈㄷㄱ | 22.03.16 22:59 | 9 |
| 2 | [어벤져스] | [자유] 마블 글 2 (0) | ㅂㅈㄷㄱ | 22.03.15 00:25 | 9 |
| 1 | [어벤져스] | [자유] 마블 글 1 (0) | ㅂㅈㄷㄱ | 22.03.15 00:25 | 7 |

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능



'Scrap.do?s_number=\${article.h_number}&s_category=\${article.h_category}&s_title=\${article.h_title}&s_nickname=\${article.h_nickname}'

```
public String makeScrapUrl(int s_category, int s_number) {
    String scrapurl="";
    if (s_category>=10 && s_category<20) {
        scrapurl="H_Content.do?h_number="+s_number;
    }else if (s_category>=20 && s_category<30) {
        scrapurl="Z_Content.do?z_number="+s_number;
    }else if (s_category>=30 && s_category<40) {
        scrapurl="M_Content.do?m_number="+s_number;
    }else if (s_category>=40 && s_category<50) {
        scrapurl="G_Content.do?g_number="+s_number;
    }else if (s_category>=50 && s_category<60) {
        scrapurl="R_Content.do?r_number="+s_number;
    }
    return scrapurl;
}
```

```
String scrapurl="";
int s_category=Integer.parseInt(request.getParameter("s_category"));
int s_number=Integer.parseInt(request.getParameter("s_number"));
```

```
BoardDAO board=new BoardDAO();
scrapurl=board.makeScrapUrl(s_category, s_number);
System.out.println("scrapurl=>" +scrapurl);
```

```
int scrap_number=0;
if (request.getParameter("scrap_number")!=null) {
    scrap_number=Integer.parseInt(request.getParameter("scrap_number"));
    System.out.println("scrap_number!=null일 때=>" +scrap_number);
}else {
    System.out.println("scrap_number==null일 때=>" +scrap_number);
}

article.setScrap_number(scrap_number);
article.setMem_id(mem_id);
article.setS_number(Integer.parseInt(request.getParameter("s_number")));
article.setS_category(Integer.parseInt(request.getParameter("s_category")));
article.setS_title(request.getParameter("s_title"));
article.setS_date(new Timestamp(System.currentTimeMillis()));
article.setS_url(scrapurl);
article.setS_nickname(request.getParameter("s_nickname"));
```

board.scrap(article);

```
int s_category=article.getS_category();
String init=this.makeInit(s_category);
if (scrap > 0) {
    sql="update "+init+"_table set "+init+"_scrap="+init+"_scrap+1 where "+init+"_number=?";
    pstmt=con.prepareStatement(sql);
    pstmt.setInt(1, article.getS_number());
    int update=pstmt.executeUpdate();
    con.commit();
}
```

<meta http-equiv="Refresh" content="0;url=\${scrapurl}">

스크랩 기능

타 회원의 글에만 노출되는 스크랩 버튼을 클릭하면

해당 글의 번호, 카테고리, 제목, 작성자 정보가 전송되고,

액션 클래스에서 카테고리 조건에 따라 글의 url을 생성하는 메서드를 호출한 뒤

DTO객체 board에 필요한 정보들을 저장하고 스크랩 테이블에 insert하는 메서

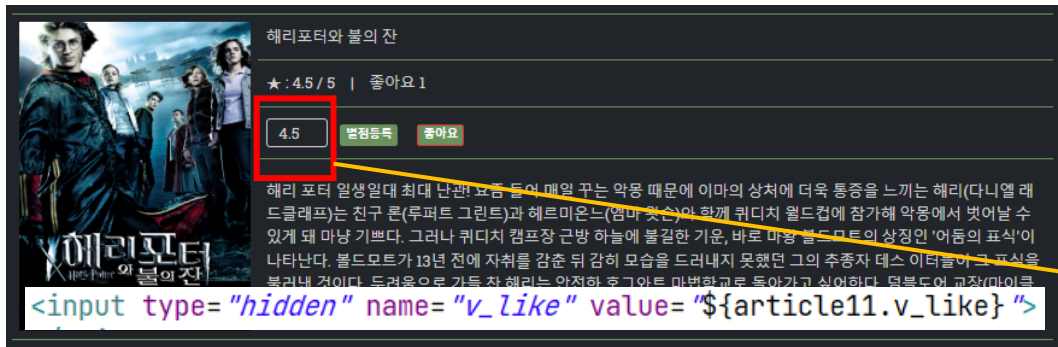
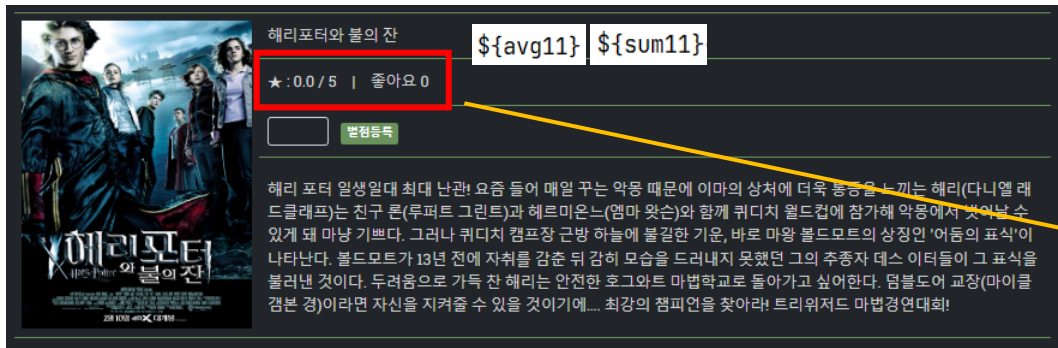
드를 호출해 저장하고, 해당 글의 스크랩 수를 1 증가시킵니다.

스크랩이 완료되고 스크랩된 글인 s_url로 돌아갈 수 있게 하였습니다.

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능



영화 별점 및 좋아요 기록 기능

회원들이 입력한 별점의 평균과 좋아요의 수를 조회할 수 있게 하였고,
회원이 별점을 입력한 상태라면 본인이 입력한 별점이 출력되게 하였습니다.

```
public double avgStar(int v_movie) {
    double x=0;
    try {
        con=pool.getConnection();
        con.setAutoCommit(false);
        sql="select round(avg(v_star), 2) from star where v_movie=?";
        pstmt=con.prepareStatement(sql);
        pstmt.setInt(1, v_movie);
        rs=pstmt.executeQuery();
        if (rs.next()) {
            x=rs.getDouble(1);
        }
        con.commit();
    } catch (Exception e) {
        System.out.println("avgStar() 에러 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return x;
}
```

```
public int countLikes(int v_movie) {
    int x=0;
    try {
        con=pool.getConnection();
        sql="select sum(v_like) from star where v_movie=?";
        pstmt=con.prepareStatement(sql);
        pstmt.setInt(1, v_movie);
        rs=pstmt.executeQuery();
        if (rs.next()) {
            x=rs.getInt(1);
        }
    } catch (Exception e) {
        System.out.println("countLikes() 에러 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return x;
}
```

```
BoardDAO avgPro=new BoardDAO();
double avg11=avgPro.avgStar(11);
:
:
```

```
BoardDAO sumPro=new BoardDAO();
int sum11=sumPro.countLikes(11);
:
:
```

```
public BoardDTO getMyRate(String mem_id, int v_movie) {
    BoardDTO article=null;
    try {
        con=pool.getConnection();
        con.setAutoCommit(false);
        sql="select * from star where mem_id=? and v_movie=?";
        pstmt=con.prepareStatement(sql);
        pstmt.setString(1, mem_id);
        pstmt.setInt(2, v_movie);
        rs=pstmt.executeQuery();
        if(rs.next()) {
            article=makeRateArticleFromResult();
            con.commit();
            System.out.println("getMyRate()의 article=>" + article);
        }
    } catch (Exception e) {
        System.out.println("getMyRate() 에러 발생=>" + e);
    } finally {
        pool.freeConnection(con, pstmt, rs);
    }
    return article;
}
```

```
BoardDTO article11=dbPro.getMyRate(mem_id, 11);
:
:
```

프로젝트 경험 2-1

영화 커뮤니티 사이트 (JSP MVC 모델 2)

5. 구현 내용 - 특수 기능

```
public void rate(BoardDTO article) {
    int v_number=article.getV_number();
    int number=0;
    try {
        con=pool.getConnection();
        con.setAutoCommit(false);
        sql="select * from star where mem_id=? and v_movie=?";
        pstmt=con.prepareStatement(sql);
        pstmt.setString(1, article.getMem_id());
        pstmt.setInt(2, article.getV_movie());
        rs=pstmt.executeQuery();
        if (rs.next()) {
            sql="update star set v_star=?, v_like=? where mem_id=? and v_movie=?";
            pstmt=con.prepareStatement(sql);
            pstmt.setDouble(1, article.getV_star());
            pstmt.setInt(2, article.getV_like());
            pstmt.setString(3, article.getMem_id());
            pstmt.setInt(4, article.getV_movie());
            int update=pstmt.executeUpdate();
            con.commit();
        }else {
            sql="select max(v_number) from star";
            pstmt=con.prepareStatement(sql);
            rs=pstmt.executeQuery();
            if (rs.next()) {
                number=rs.getInt(1)+1;
            }else {
                number=1;
            }
            con.commit();
            sql="insert into star(v_number, mem_id, v_movie, v_star, v_like) ";
            sql+=" values(?, ?, ?, ?, ?)";
            pstmt=con.prepareStatement(sql);
            pstmt.setInt(1, number);
            pstmt.setString(2, article.getMem_id());
            pstmt.setInt(3, article.getV_movie());
            pstmt.setDouble(4, article.getV_star());
            pstmt.setInt(5, article.getV_like());
            int insert=pstmt.executeUpdate();
            con.commit();
        }
    } catch (Exception e) {
        System.out.println("rate() 메서드 에러 발생=>" +e);
    } finally {
    }
}
```

영화 별점 및 좋아요 기록 기능

별점 계산 오류를 방지하기 위해 별점을 먼저 입력해야 좋아요를 입력할 수 있는 버튼이 뜨게 하였습니다.

별점 테이블에서 회원의 아이디가 포함된 데이터를 검색하여 결과가 없다면 입력을, 있다면 수정을 할 수 있게 하였습니다.

그리고 좋아요 버튼이 눌러 있지 않을 때 누르면 1, 좋아요 버튼이 눌러 있을 때 누르면 0이 입력되게 하였습니다.

```
<c:if test="${article11.v_star!=null}">
</c:choose>
<c:when test="${article11.v_like==1}">
    <form role="form" class="rankForm" name="rankForm2" method="post" action="H_Rate.do" onsubmit="return writeSave()">
        <input type="hidden" name="v_number" value="${v_number}">
        <input type="hidden" name="mem_id" value="${mem_id}">
        <input type="hidden" name="v_movie" value="11">
        <input type="hidden" name="v_like" value="0">
        <c:if test="${article11.v_star!=null}">
            <input type="hidden" name="v_star" value="${article11.v_star}">
        </c:if>
        <button type="submit" class="btn btn-danger btn-xs">좋아요</button>
    </form>
</c:when>
<c:when test="${article11.v_like==0 || article11.v_like==null}">
    <form role="form" class="rankForm" name="rankForm2" method="post" action="H_Rate.do" onsubmit="return writeSave()">
        <input type="hidden" name="v_number" value="${v_number}">
        <input type="hidden" name="mem_id" value="${mem_id}">
        <input type="hidden" name="v_movie" value="11">
        <input type="hidden" name="v_like" value="1">
        <c:if test="${article11.v_star!=null}">
            <input type="hidden" name="v_star" value="${article11.v_star}">
        </c:if>
        <button type="submit" class="btn btn-default btn-xs">좋아요</button>
    </form>
</c:when>
</c:choose>
</c:if>
```

프로젝트 경험 2-2) 개인 프로젝트 (Spring MVC 2)

Movie So Good 영화 커뮤니티 사이트

| | |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 목표 | 스프링 기능을 모두 익혀 JSP MVC 2 팀 프로젝트를 스프링으로 완전히 변환하는 것 |
| 개발환경 | 개발 언어 : HTML5, CSS3, JavaScript, Java 8, JSP 프레임워크 : Spring Framework, MyBatis 라이브러리 : Bootstrap 3.5 웹 서버 : Apache Tomcat v9.0 개발도구 : Eclipse 2021-12 DB : Oracle 11g 운영체제 : Windows 10 기타 도구 : SQL Plus, DBeaver, eXERD, EditPlus, Procreate |
| 프로젝트 기간 | 2022.03.30 - 2022.04.20 |
| 특수 기능 | 회원 등급제, 장르·카테고리별 게시판, 핫 게시판, 내가 쓴 글 모아 보기, 스크랩 기능, 별점 및 좋아요 등록 게시판 |
| 구현 정도 | 99% |

프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

장르·카테고리별 게시판

검색결과까지 보여줄 수 있으면서,

자유, 정보 등의 카테고리(h_ref)가 나뉘어지지 않은 상태의 글 목록

출력을 위한 메서드인


selectList_H sql문을 작성하였고,

카테고리별로 나뉜 상태의 글 목록과

조회수(h_count)가 50 이상인 인기글 목록 출력을 위해

where조건을 추가한 sql문들 또한 작성하였습니다.

```
<select id="selectList_H" parameterType="map" resultType="HBoardDTO">
  select *
    from (select h_table.*, rownum rn
          from (select * from h_table
                <where>
                  <if test="search=='h_title' and searchText!=null">
                    h_title like '%' || #{searchText} || '%'
                  </if>
                  <if test="search=='h_body' and searchText!=null">
                    h_body like '%' || #{searchText} || '%'
                  </if>
                  <if test="search=='h_nickname' and searchText!=null">
                    h_nickname like '%' || #{searchText} || '%'
                  </if>
                  <if test="search=='h_all' and searchText!=null">
                    h_title like '%' || #{searchText} || '%' or
                    h_body like '%' || #{searchText} || '%' or
                    h_nickname like '%' || #{searchText} || '%'
                  </if>
                </where>
              order by h_number desc) h_table)
  <![CDATA[
    where rn >=#{start} AND rn <=#{end}
  ]]>
</select>
```



```
id="selectListHref_H" order by h_number desc) h_table where h_ref=#{h_ref})
```

```
id="selectListHot_H" order by h_number desc) h_table where h_count > 50)
```

프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

```
<ul class="nav nav-pills nav-justified">
<li class="col-md-3"><a href="H_List.do?h_ref=1">자유</a></li>
<li class="col-md-3"><a href="H_List.do?h_ref=2">정보</a></li>
<li class="col-md-3"><a href="H_Rate.do">별점</a></li>
<li class="col-md-3"><a href="H_List.do?h_ref=4">HOT</a></li>
</ul>
```

| 자유 | | 정보 | | 별점 | HOT | |
|-----|---------|------------------------------------|--------------------|-------|----------------|----|
| 번호 | | 제목 | | 작성자 | 작성일 | 조회 |
| 3 | | [공지] <MSG:movie so good> 이용수칙 안내 | | admin | 22.04.20 16:06 | 2 |
| 2 | | [공지] <MSG:movie so good> SNS 오픈 안내 | | admin | 22.04.20 16:06 | 1 |
| 1 | | [공지] <MSG:movie so good> 그랜드 오픈 안내 | | admin | 22.04.20 16:05 | 1 |
| 133 | [비밀의 방] | [자유] | 1234 (0) | 관리자 | 22.04.25 14:50 | 15 |
| 132 | [문의란] | [자유] | MSG_Spring 테스트 (0) | 박지민 | 22.04.20 22:43 | 11 |

장르·카테고리별 게시판

자유, 정보, HOT 탭에 번호를 부여했고

글 목록 컨트롤러에서 각 탭의 h_ref 파라미터 값을 받아와

Null이라면 기본 메서드,

1이나 2라면 자유나 정보 카테고리를 출력해주는 _href메서드, 4라면 인기글 목

록을 출력해주는 _hot 메서드를 호출할 수 있게 하였습니다.

```
Map<String, Object> map=new HashMap<String, Object>();
```

```
String h_ref=request.getParameter("h_ref");
map.put("h_ref", h_ref);
map.put("search", search);
map.put("searchtext", searchtext);
```

```
int count=0;
if (h_ref==null) {
    count=hBoardDao.getRowCount(map);
}else if (Integer.parseInt(h_ref)==1 || Integer.parseInt(h_ref)==2) {
    count=hBoardDao.getRowCount_href(map);
}else if (Integer.parseInt(h_ref)==4) {
    count=hBoardDao.getRowCount_hot(map);
}
```

```
Hashtable<String, Integer> pgList=hBoardDao.pageList(currentPage, count);
```

```
map.put("start", pgList.get("startRow"));
map.put("end", pgList.get("endRow"));
```

```
List<HBoardDTO> articleList=null;
if (count > 0) {
    if (h_ref==null) {
        articleList=hBoardDao.list(map);
    }else if (Integer.parseInt(h_ref)==1 || Integer.parseInt(h_ref)==2) {
        articleList=hBoardDao.list_href(map);
    }else if (Integer.parseInt(h_ref)==4) {
        articleList=hBoardDao.list_hot(map);
    }
}else {
    articleList=Collections.EMPTY_LIST;
}
```


프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

```
<update id="pointupA_H" parameterType="String">
    update login set log_point=log_point+10 where mem_id=#{mem_id}
</update>
<update id="pointdownA_H" parameterType="String">
    update login set log_point=log_point-10 where mem_id=#{mem_id}
</update>
<update id="pointupC_H" parameterType="String">
    update login set log_point=log_point+5 where mem_id=#{mem_id}
</update>
<update id="pointdownC_H" parameterType="String">
    update login set log_point=log_point-5 where mem_id=#{mem_id}
</update>

<update id="changeGrade1_H" parameterType="String">
    update login set log_grade=1 where mem_id=#{mem_id}
</update>
<update id="changeGrade2_H" parameterType="String">
    update login set log_grade=2 where mem_id=#{mem_id}
</update>
<update id="changeGrade3_H" parameterType="String">
    update login set log_grade=3 where mem_id=#{mem_id}
</update>
<update id="changeGrade4_H" parameterType="String">
    update login set log_grade=4 where mem_id=#{mem_id}
</update>
<update id="changeGrade5_H" parameterType="String">
    update login set log_grade=5 where mem_id=#{mem_id}
</update>
```

회원 등급제

회원의 포인트에 따라 등급을 5개로 나누었으며

글 작성시 10포인트 적립, 삭제시 10포인트 회수되게 하였고,

댓글의 경우 5포인트 적립, 5포인트 회수되게 하였습니다.

또한 포인트 변동시 등급도 바로 변경될 수 있게 하였고

등급을 지체없이 반영시키기 위해 세션에 다시 등록하였습니다.

```
if (insert > 0) {
    String mem_id=com.getMem_id();
    System.out.println("H_WriteController의 mem_id=>" + mem_id);
    int pointup=hBoardDao.pointupA(mem_id);
    System.out.println("글 작성시 포인트 적립 성공 여부(pointup)=>" + pointup);
    int point=hBoardDao.getPoint(mem_id);

    if (pointup > 0) {
        System.out.println("등급 변동 전 현재 등급=>" + hBoardDao.getGrade(mem_id));
        int gradeup=0;
        if (point < 100) {
            gradeup=hBoardDao.changeGrade1(mem_id);
        } else if (point >= 100 && point < 500) {
            gradeup=hBoardDao.changeGrade2(mem_id);
        } else if (point >= 500 && point < 2000) {
            gradeup=hBoardDao.changeGrade3(mem_id);
        } else if (point >= 2000 && point < 10000) {
            gradeup=hBoardDao.changeGrade4(mem_id);
        } else if (point >= 10000) {
            gradeup=hBoardDao.changeGrade5(mem_id);
        }
        System.out.println("글 작성 후 등급 변동 여부(gradeup)=>" + gradeup);
        System.out.println("등급 변동 후 현재 등급=>" + hBoardDao.getGrade(mem_id));
    }
    MemberDTO mcom=memberDao.getNPG(mem_id);
    int log_grade=mcom.getLog_grade();
    System.out.println("log_grade=>" + log_grade);

    MemberDTO mcom2=memberDao.selectPG(mem_id);
    memberDao.syncPG(mcom2);

    HttpSession session=request.getSession();
    session.setAttribute("mem_grade", log_grade);
}
```


프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

```
<select id="getMyArticleCountH" parameterType="map" resultType="Integer">
    select count(*) from h_table where mem_id=#{mem_id}
</select>
```

•
•
•

```
public int getMyArticleCount(Map<String, Object> map, String genre) {
    int count=0;
    if (genre!=null) {
        if (genre.equals("h")) {
            count=getSqlSession().selectOne("getMyArticleCountH", map);
        }else if (genre.equals("z")) {
            count=getSqlSession().selectOne("getMyArticleCountZ", map);
        }else if (genre.equals("m")) {
            count=getSqlSession().selectOne("getMyArticleCountM", map);
        }else if (genre.equals("g")) {
            count=getSqlSession().selectOne("getMyArticleCountG", map);
        }else if (genre.equals("r")) {
            count=getSqlSession().selectOne("getMyArticleCountR", map);
        }
    }else {
        count=0;
    }
    return count;
}
```

내가 쓴 글 모아보기 기능

매개변수로 넘어온 장르의 값에 따라 그 장르별 글의 수와 목록을 구하는 sql문을 실행하는 메서드를 작성하였습니다.

```
<select id="getMyArticlesH" parameterType="map" resultType="HBoardDTO">
    select *
        from (select h_table.*, rownum rn
                from (select * from h_table where mem_id=#{mem_id}
                      order by h_number desc) h_table)

    <![CDATA[
        where rn >= #{start} and rn <= #{end}
    ]]>
</select>
```

•
•
•

```
public List getMyArticles(Map<String, Object> map, String genre) {
    List list=null;
    if (genre!=null) {
        if (genre.equals("h")) {
            list=getSqlSession().selectList("getMyArticlesH", map);
        }else if (genre.equals("z")) {
            list=getSqlSession().selectList("getMyArticlesZ", map);
        }else if (genre.equals("m")) {
            list=getSqlSession().selectList("getMyArticlesM", map);
        }else if (genre.equals("g")) {
            list=getSqlSession().selectList("getMyArticlesG", map);
        }else if (genre.equals("r")) {
            list=getSqlSession().selectList("getMyArticlesR", map);
        }
    }else {
        System.out.println("x");
        list=getSqlSession().selectList("getMyArticles", map);
    }
    return list;
}
```

프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

```
<ul class="nav nav-pills nav-justified">
<li class="col-md-1"></li>
<li class="col-md-2"><a href="MyArticles.do?genre=h">해리포터</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=z">짱구</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=m">마블</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=g">지브리</a></li>
<li class="col-md-2"><a href="MyArticles.do?genre=r">반지의 제왕</a></li>
<li class="col-md-1"></li>
</ul>
```

| 내가 쓴 글 | | | | | |
|----------------------------------------|---------------|-------------------|------|----------------|----|
| | | | | | |
| 해리포터 짱구 마블 지브리 반지의 제왕 | | | | | |
| 번호 | 제목 | | 작성자 | 작성일 | 조회 |
| 7 | [아이언맨] | [자유] 마블 글 (0) | ㅂㅈㄷㄱ | 22.04.20 17:34 | 12 |
| 6 | [아이언맨] | [자유] sdfrr (0) | ㅂㅈㄷㄱ | 22.03.31 19:26 | 17 |
| 5 | [가디언즈 오브 갤럭시] | [자유] 1234 (0) | ㅂㅈㄷㄱ | 22.03.17 02:39 | 9 |
| 4 | [블랙팬서] | [자유] 댓글 테스트 m (0) | ㅂㅈㄷㄱ | 22.03.17 00:58 | 9 |
| 3 | [블랙팬서] | [자유] 스크랩-m2 (0) | ㅂㅈㄷㄱ | 22.03.16 22:59 | 9 |
| 2 | [어벤져스] | [자유] 마블 글 2 (0) | ㅂㅈㄷㄱ | 22.03.15 00:25 | 9 |
| 1 | [어벤져스] | [자유] 마블 글 1 (0) | ㅂㅈㄷㄱ | 22.03.15 00:25 | 7 |

내가 쓴 글 모아보기 기능

각 장르 탭에 이니셜로 값을 주고,

Controller 클래스에서 장르 파라미터 값을 받아온 후 메서드를 호출해

각 장르의 글의 수와 목록이 출력되도록 하였습니다.

```
Map<String, Object> map=new HashMap<String, Object>();
```

```
String genre=request.getParameter("genre");
HttpSession session=request.getSession();
String mem_id=(String)session.getAttribute("idKey");
```

```
map.put("genre", genre);
map.put("mem_id", mem_id);
```

```
int count=0;
if (genre==null) {
    count=0;
}else {
    count=boardDao.getMyArticleCount(map, genre);
}
```

```
Hashtable<String, Integer> pgList=boardDao.pgList(currentPage, count);
```

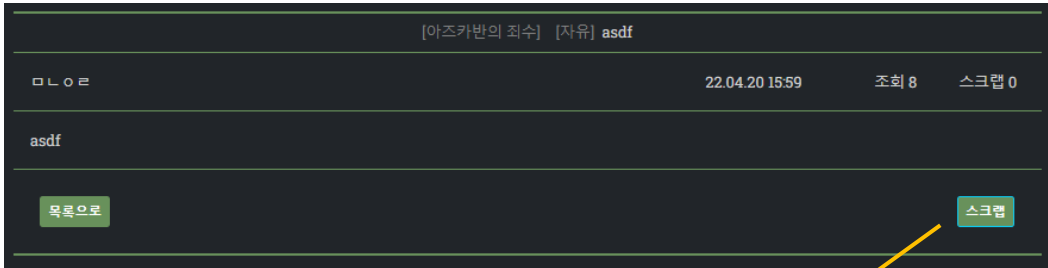
```
map.put("start", pgList.get("startRow"));
map.put("end", pgList.get("endRow"));
```

```
List articleList=null;
if (count > 0) {
    articleList=boardDao.getMyArticles(map, genre);
}else {
    articleList=Collections.EMPTY_LIST;
}
```

프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능



`'Scrap.do?s_number=${article.h_number}&s_category=${article.h_category}&s_title=${article.h_title}&s_nickname=${article.h_nickname}'`

```
public String makeInit(Integer s_category) {
    String init="";

    if (s_category>=10 && s_category<20) {
        init="h";
    }else if (s_category>=20 && s_category<30) {
        init="z";
    }else if (s_category>=30 && s_category<40) {
        init="m";
    }else if (s_category>=40 && s_category<50) {
        init="g";
    }else if (s_category>=50 && s_category<60) {
        init="r";
    }

    return init;
}

public String makeScrapUrl(String init, Integer s_number) {
    String s_url=init.toUpperCase()+"_Content.do?" +init+"_number="+s_number;
    return s_url;
}
```

스크랩 기능

타 회원의 글에만 노출되는 스크랩 버튼을 클릭하면

해당 글의 번호, 카테고리, 제목, 작성자 정보가 전송되고,
카테고리 조건에 따라 장르의 종류를 찾아내는 메서드를 호출합니다.

그리고 구한 장르 값을 매개변수로 넣어 url을 생성하는 메서드를 호출합니다.

DTO객체 board에 필요한 정보들을 저장하고 스크랩 테이블에 insert하는 메서드를 호출해 저장하고, 해당 글의 스크랩 수를 1 증가시킵니다.

스크랩이 완료되고 스크랩된 글인 s_url로 돌아갈 수 있게 하였습니다.

```
int scrap_number=boardDao.getNewScrapNum()+1;
HttpSession session=request.getSession();
String mem_id=(String)session.getAttribute("idKey");

int s_number=Integer.parseInt(request.getParameter("s_number"));
int s_category=Integer.parseInt(request.getParameter("s_category"));
System.out.println("s_category=>" +s_category);
String s_title=request.getParameter("s_title");
String s_nickname=request.getParameter("s_nickname");

String init=boardDao.makeInit(s_category);
board.setInit(init);

String s_url=boardDao.makeScrapUrl(init, s_number);
System.out.println("생성된 s_url=>" +s_url);

board.setScrap_number(scrap_number);
board.setMem_id(mem_id);
board.setS_number(s_number);
board.setS_category(s_category);
board.setS_title(s_title);
board.setS_url(s_url);
board.setS_nickname(s_nickname);

int scrap=boardDao.scrap(board);
System.out.println("스크랩 성공 여부(scrap)=>" +scrap);

int scrapup=boardDao.scrapUp(board);

request.setAttribute("s_url", s_url);
return "redirect:/" +s_url;
```

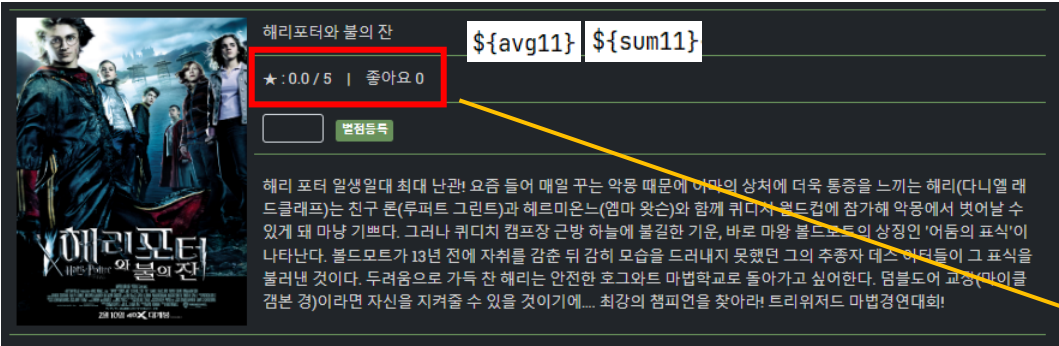
프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

영화 별점 및 좋아요 기록 기능

회원들이 입력한 별점의 평균과 좋아요의 수를 조회할 수 있게 하였고,
회원이 별점을 입력한 상태라면 본인이 입력한 별점이 출력되게 하였습니다.

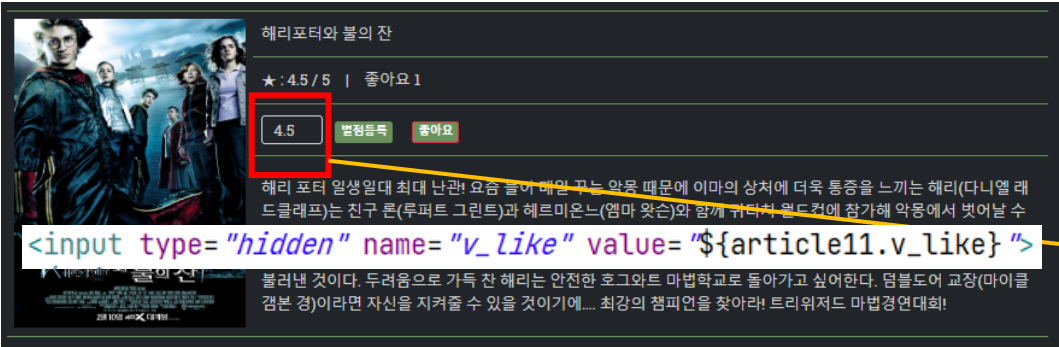


```
<select id="avgStar" parameterType="Integer" resultType="Double">
    select round(avg(v_star), 2) from star where v_movie=#{v_movie}
</select>

<select id="countLikes" parameterType="Integer" resultType="Integer">
    select sum(v_like) from star where v_movie=#{v_movie}
</select>
```

```
double avg11=boardDao.avgStar(11);
int sum11=boardDao.countLikes(11);

:
:
:
```



```
<input type="hidden" name="v_like" value="${article11.v_like}">
```

```
<select id="getMyRate" parameterType="map" resultType="BoardDTO">
    select * from star where mem_id=#{mem_id} and v_movie=#{v_movie}
</select>
```

```
Map<String, Object> map11=new HashMap<String, Object>();
map11.put("mem_id", mem_id);
map11.put("v_movie", 11);
BoardDTO article11=boardDao.getMyRate(map11);

:
:
```

프로젝트 경험 2-2

영화 커뮤니티 사이트 (Spring MVC 2)

5. 구현 내용 - 특수 기능

영화 별점 및 좋아요 기록 기능

별점 계산 오류를 방지하기 위해 별점을 먼저 입력해야 좋아요를 입력할 수 있는 버튼이 뜨게 하였습니다.

별점 테이블에서 회원의 아이디가 포함된 데이터를 검색하여 결과가 없다면 입력을, 있다면 수정을 할 수 있게 하였습니다.

그리고 좋아요 버튼이 눌러 있지 않을 때 누르면 1, 좋아요 버튼이 눌러 있을 때 누르면 0이 입력되게 하였습니다.

```
<c:if test="${article11.v_star!=null}">
</c:if>
<c:when test="${article11.v_like==1}">
    <form role="form" class="rankForm" name="rankForm2" method="post" action="H_Rate.do" onsubmit="return writeSave()">
        <input type="hidden" name="v_number" value="${v_number}">
        <input type="hidden" name="mem_id" value="${mem_id}">
        <input type="hidden" name="v_movie" value="11">
        <input type="hidden" name="v_like" value="0">
        <c:if test="${article11.v_star!=null}">
            <input type="hidden" name="v_star" value="${article11.v_star}">
        </c:if>
        <button type="submit" class="btn btn-danger btn-xs">좋아요</button>
    </form>
</c:when>
<c:when test="${article11.v_like==0 || article11.v_like==null}">
    <form role="form" class="rankForm" name="rankForm2" method="post" action="H_Rate.do" onsubmit="return writeSave()">
        <input type="hidden" name="v_number" value="${v_number}">
        <input type="hidden" name="mem_id" value="${mem_id}">
        <input type="hidden" name="v_movie" value="11">
        <input type="hidden" name="v_like" value="1">
        <c:if test="${article11.v_star!=null}">
            <input type="hidden" name="v_star" value="${article11.v_star}">
        </c:if>
        <button type="submit" class="btn btn-default btn-xs">좋아요</button>
    </form>
</c:when>
</c:choose>
</c:if>
```

```
HttpSession session=request.getSession();
String mem_id=(String)session.getAttribute("idKey");
System.out.println("H_RateController submit()의 mem_id=>"+mem_id);

int update=0;
int insert=0;

Map<String, Object> mapF=new HashMap<String, Object>();
mapF.put("mem_id", mem_id);
System.out.println("com.getV_movie()=>"+com.getV_movie());
mapF.put("v_movie", com.getV_movie());
int find=boardDao.findRate(mapF);

if(find > 0) {
    update=boardDao.updateRate(com);
}else {
    int maxnum=boardDao.getMaxNum()+1;
    com.setV_number(maxnum);
    insert=boardDao.insertRate(com);
}

return "redirect:/H_Rate.do";
```

읽어주셔서 감사합니다.