

# REPORT

*Hanseo University*



## 영화 추천 및 설명 시스템

교과명 : 데이터베이스  
수강일 : 화요일  
교수명 : 심종익 교수님  
학과명 : 항공소프트웨어공학과  
팀 명 : Sofa  
학 번 : 201701198, 201701227  
이 름 : 박경범, 정규진  
제출일 : 2012.11.28



## <목차>

1. 프로젝트 주제 및 개요 .....	3
1.1 프로젝트 주제 .....	
1.2 프로젝트 개요 .....	
1.3 프로젝트 기대효과 .....	
2. 팀원 개별 역할 .....	3
3. 프로젝트 개발 도구 .....	4
4. 프로젝트 개발 일정 .....	4
5. 서버 시스템 구축 .....	6
6. 기능설명 .....	7
7. DBMS 구성 .....	10
8. 프로그램 코드 .....	13

## 1. 프로젝트 주제 및 개요

---

### 1-1. 프로젝트 주제

---

본 프로젝트는 영화 추천 및 설명을 제공하는 시스템 개발이다. 사용자에게 다양한 영화를 추천함과 동시에 그 영화의 제목, 감독, 개봉일 등에 대한 정보를 제공한다. 깔끔한 UI를 제공하여 다양한 정보를 알아보기 쉬운 것이며, 카테고리별 검색을 통해 원하는 정보를 원활히 찾을 수 있다. 또한, 사용자끼리의 영화정보 추천 및 소통을 할 수 있는 홈페이지이다.

### 1-2. 프로젝트 개요

---

영화 추천 및 설명 프로그램이다. 지속되는 코로나로 영화관에서 영화를 관람하기보다 집에서 영화를 보는 사람들이 증가하였다. 그로 인해 'Netflix', 'Watcha'와 같은 OTT 서비스로 많은 수요가 몰리고 있다. 하지만 OTT 서비스를 통해 많은 콘텐츠를 접할 수 있지만, 너무 많은 선택지로 자신이 원하는 콘텐츠 선택을 고민하게 되는 문제점이 있다. 이런 문제점을 가진 사람들을 위해 다양한 카테고리를 활용한 검색으로 취향에 맞는 영화를 쉽게 찾을 수 있으며, 기재되어 있는 설명을 통해 정보를 받을 수 있으며, 게시판을 이용하여 서로 소통할 수 있다.

### 1-3 프로젝트 기대효과

---

영화정보를 원하는 사람들에게 간편하고 빠르게 자료를 제공할 수 있다. 그와 동시에 알지 못했던 새로운 영화를 알게 되는 효과를 기대할 수 있으며, 서로의 의견을 공유할 수 있고 서로에게 영화를 추천해줄 수 있다.

## 2. 팀원 개별 역할

---

201701198 박경범 - 프로그램 메인 제작 및 프로그램 설계

201701227 정규진 - 프로그램 보조 담당 및 자료수집

## 3. 프로젝트 개발 도구

---

**HTML** - 인터넷 서비스의 하나인 월드 와이드 웹을 통해 볼 수 있는 문서를 만들 때 사용하는 웹 언어의 한 종류이다.

**CSS**- CSS는 HTML로 작성된 문서의 표시 방법을 기술하기 위한 스타일 시트 언어다.

**JavaScript** - 자바스크립트는 크로스 플랫폼, 객체지향 스크립트 언어로 웹페이지의 동작을 담당한다.

**DBMS: MySQL** - MySQL은 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템이다.

## 4. 프로젝트 개발 일정

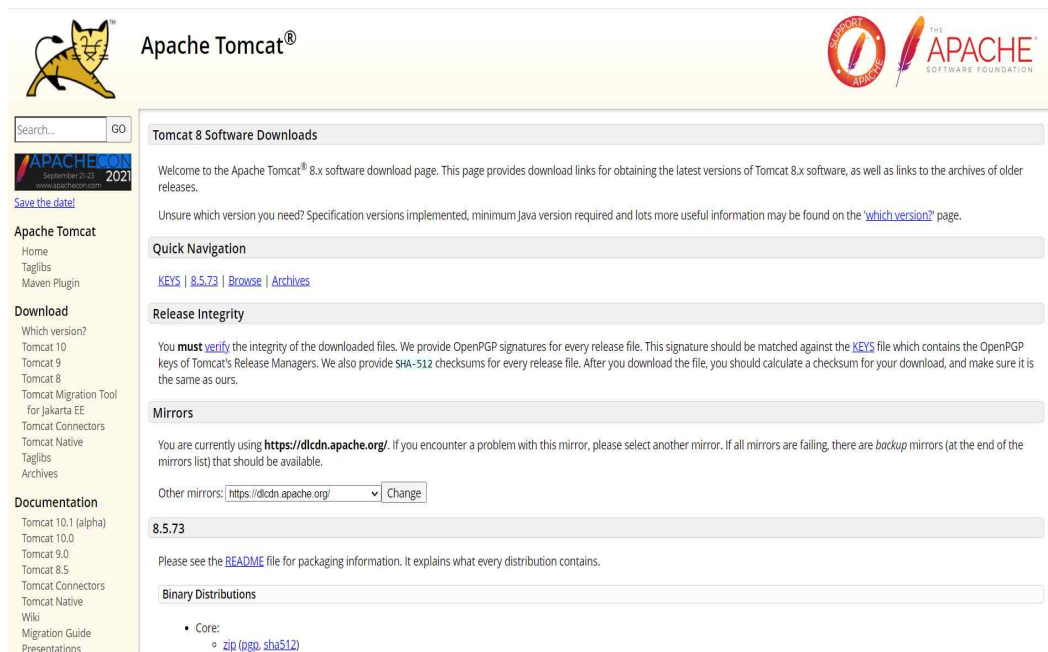
---

	10월			11월				12월
	6주차	7주차	8주차	9주차	10주차	11주차	12주차	13주차
아이디어 구상 및 계획서 작성								
홈페이지 제작								
UI 디자인 및								

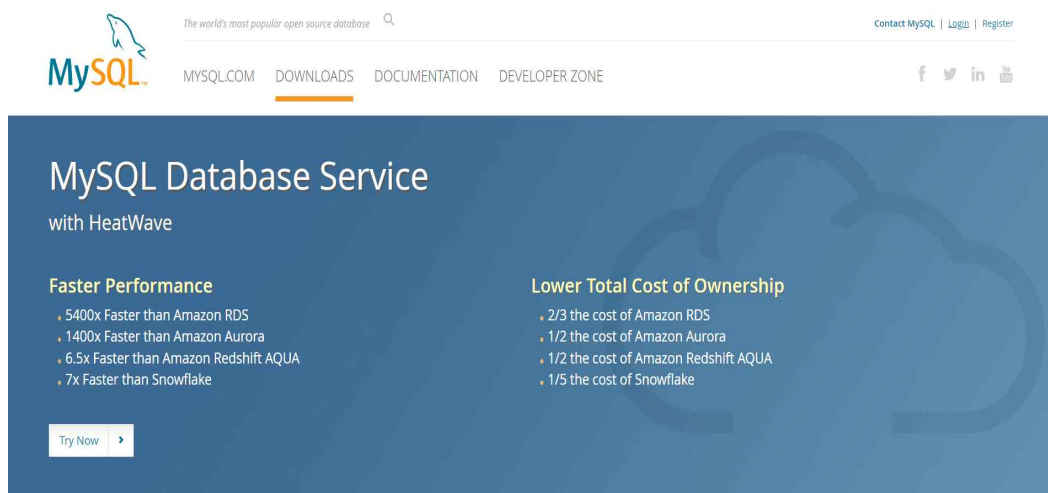
기능구현								
Database 구축								
테스트 및 유지보수								

단계	내용
1단계	전체적인 프로젝트 설계 및 구성도 구체화 및 이를 위한 자료를 조사한다.
2단계	개발도구 조사(HTML, CSS, JavaScript), 데이터베이스(MySQL) 조사 및 설치한다.
3단계	로그인, 회원가입, 로그아웃 기능을 구현하고 메인 페이지 제작 및 개발도구의 구동을 확인한다.
4단계	메인 페이지에 영화 등록을 만든 후 데이터베이스에 등록(insert), 삭제(delete), 검색(search)되도록 한다.
5단계	게시판 기능도 등록(insert), 삭제(delete), 수정이 되도록 한다.
6단계	전체적인 테스트를 진행한다.

## 5. 서버 시스템 구축



서버로는 Tomcat 8.5 버전을 사용하였다.



서버에서 사용할 데이터베이스로 MySQL을 사용하였다.

## 6. 기능설명

영화 추천 사이트   메인   게시판   영화 추천   접속하기 ▼

전체 ▼   최신순 ▼

뽀로로 극장판 눈요정 마을 대모험 김현호   평점7.7

줄거리

장난꾸러기 꼬마팬권 '뽀로로'와 친구들은 우연히 세상에 눈을 뿌리는 '눈요정 마을'에 가게 되고, 조각가가 꿈인 '눈요정 아티'를 만나 즐거운 시간을 보낸다. 그 때, 어디선가 날아든 거대한 용암괴물이 마을을 습격해 눈요정들을 모두 데리고 가는데...

상영 시간: 35분 개봉일: 2014년12월11일 (추천: 0)   추천 삭제

이전   다음

copyright © 2021 SOFA All Rights Reserved.

### ▶ 영화 추천 페이지

영화 추천 페이지에서는 사용자가 자신이 추천하는 영화를 등록하여 다른 사람들에게 소개할 수 있고, 제목을 누르면 해당 영화링크에 해당하는 자세한 사이트로 이동하고, 추천하기 버튼을 눌러 추천, 삭제 버튼을 눌러 글을 삭제하는 기능이 있다. 또한, 장르 및 최신, 추천순을 눌러 자신이 원하는 영화를 쉽고 빠르게 찾을 수 있도록 구현하였다.

### ▶ 영화 사이트, 메인, 영화 추천

상단의 메뉴바에서 영화 추천 사이트를 누르면 메인으로, 메인을 누르면 메인으로, 영화 추천을 누르면 영화 추천으로, 게시판을 누르면 게시판으로 각각에 맞는 페이지로 이동하도록 설정하였다.

## ▶ 게시판



게시판의 글쓰기 버튼을 이용하여 사용자의 글쓰기가 가능하다. 번호, 제목, 작성자, 작성일을 표시하게 제작하였고, 제목을 클릭할 시에 세부 페이지로 들어가게 제작하였다.



본인의 아이디로 세부 페이지에 들어가게 되면 목록, 수정 삭제, 글쓰기 기능을 사용할 수 있다. 본인이 작성한 글이 아닐 경우 수정, 삭제 버튼은 나타나지 않도록 제작하였다.

## ▶ 접속하기

회원가입, 로그인, 로그아웃 기능이 들어가 있으며, 로그인이 상태에서는 드롭다운에 로그아웃만 나타나게 되며, 비 로그인 상태에서는 회원가입과 로그아웃이 나타나도록 제작하였다.



▶ 검색, 등록하기

검색 버튼을 이용하여 영화 장르를 선택하여 검색할 수 있으며, 추천순 및 최신순으로 검색을 할 수 있다.

영화 등록

제목

감독

상영시간

개봉일

평점

장르

영화 주소

줄거리

취소

등록

뽀로로 극장판 눈요정 마을

김현호

35분

2014년 12월 11일

7.7

애니메이션

https://movie.naver.com/movie/bi/mi/basic.naver?code=121132

영건두디가 뽀로로와 조조조와 친구들은 두건이 세상에 눈  
을 뿌리는 '눈요정 마을'에 가게 되고, 조각가가 꿈인 '눈요정  
아티'를 만나 즐거운 시간을 보낸다.  
그 때, 어디선가 날아든 거대한 용암괴물이 마을을 습격해  
눈요정들을 모두 데리고 가는데...

과연, 뽀로로와 친구들은 눈요정들을 무사히 구출하고, 화이  
트 크리스마스를 맞이할 수 있을까요?

취소

등록

검색창 옆의 등록 버튼으로 위와 같은 형식으로 영화를 등록할 수 있다. 등록 버튼을 누르면 Modal을 사용하여 팝업과 같이 영화 등록에 필요한 형식이 나오게 제작하였다.

## ▶ 추천, 삭제하기

영화 추천 사이트 메인 게시판 영화 추천 회원관리 ▾

전체 ▾ 최신순 ▾

내용을 입력하세요.

검색 등록

뽀로로 극장판 눈요정 마을 대모험 김현호

평점7.7

줄거리

장난꾸러기 꼬마평귀 '뽀로로'와 친구들은 우연히 세상에 눈을 뿌리는 '눈요정 마을'에 가게 되고, 조각가가 꿈인 '눈요정 아티'를 만나 즐거운 시간을 보낸다. 그 때, 어디선가 날아든 거대한 용암괴물이 마을을 습격해 눈요정들을 모두 데리고 가는데...

상영 시간: 35분 개봉일: 2014년12월11일 추천: 1

추천 삭제

이전 다음

copyright © 2021 SOFA All Rights Reserved.

우측 하단의 추천 버튼을 누르게 되면 “**추천: 1**”이 된 것을 볼 수 있다. 같은 아이디로 눌렀던 추천 버튼을 한 번 더 누를 시 ‘이미 추천한 글입니다.’라는 문구가 뜬다. 또한, 삭제하기 버튼을 이용하여 게시글을 등록한 아이디로만 등록한 글을 삭제할 수 있게 제작하였다.

## 7. DBMS 구성

### ▶ User

```
mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
userID	varchar(20)	NO	PRI	NULL	
userPassword	varchar(64)	YES		NULL	
userName	varchar(20)	YES		NULL	
userGender	varchar(20)	YES		NULL	
userEmail	varchar(50)	YES		NULL	

로그인, 회원가입의 기능을 담기 위한 테이블이다.

유저 테이블은 userID, userPassword, userName, userGender, userEmail로 구성된다.

타입은 varchar(20), varchar(64), varchar(20), varchar(20), varchar(50) 이다. 괄호의 숫자는 글씨 제한 수를 나타낸다.

userID는 Primary Key로 사용되고, Column들은 각각 ID, 비밀번호, 이름, 성별, 이메일 정보를 저장한다.

## ► Movie

```
mysql> desc movie;
```

Field	Type	Null	Key	Default	Extra
movieTitle	varchar(50)	YES		NULL	
movieDirector	varchar(20)	YES		NULL	
movieTime	varchar(20)	YES		NULL	
movieReleasedate	varchar(20)	YES		NULL	
movieGenre	varchar(20)	YES		NULL	
movieContent	varchar(2048)	YES		NULL	
movieLike	int	YES		NULL	
movieID	int	NO	PRI	NULL	auto_increment
userID	varchar(20)	YES		NULL	
movieURL	varchar(2048)	YES		NULL	
movieRating	varchar(20)	YES		NULL	

영화 등록 정보를 담기 위한 테이블이다.

movie 테이블은 movieTitle, movieDirector, movieTime, movieRealeasedate, movieGenre, movieContent, movieLike, movieID, userID, movieURL, movieRating 으로 구성된다.

타입은 varchar(50), varchar(20), varchar(20), varchar(20), varchar(20), varchar(20), varchar(2048), int, int, varchar(20), varchar(2048), varcahr(20) 이다.

movieID 는 Primary Key로 사용되고, Column들은 각각 제목, 감독, 상영시간, 개봉일, 장르, 줄거리, 추천 수, movieID, 링크, 평점 정보를 저장한다.

## ▶ BBS

---

```
mysql> desc bbs;
```

Field	Type	Null	Key	Default	Extra
bbsID	int	NO	PRI	NULL	
bbsTitle	varchar(50)	YES		NULL	
userID	varchar(20)	YES		NULL	
bbsDate	datetime	YES		NULL	
bbsContent	varchar(2048)	YES		NULL	
bbsAvailable	int	YES		NULL	

게시판 정보를 담기 위한 테이블이다.

BBS 테이블은 bbsID, bbsTitle, userID, bbsData, bbsContent, bbsAvailable로 구성된다.

타입은 int, varchar(50), varchar(20), datetime, varchar(2048), int이다.

bbsID는 Primary Key로 사용되고, Column들은 각각 bbsID, 제목, userID, 등록일, 내용, 삭제 정보를 저장한다.

## ▶ likey

---

```
mysql> desc likey;
```

Field	Type	Null	Key	Default	Extra
userID	varchar(20)	NO	PRI	NULL	
movieID	int	NO	PRI	NULL	

한 명의 유저가 여러 번의 추천 버튼을 누르지 못하도록 제약을 걸기 위해 만든 테이블이다.

likey 테이블은 userID, movieID로 구성되며, 타입은 varchar(20), int이다.

userID와 movieID를 서로 엮어서 Primary Key로 사용하였다.

## 8. 프로그램 코드

▶ movieRecommendation.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="java.io.PrintWriter"%>
<%@ page import="movie.movieIDTO"%>
<%@ page import="movie.movieIDAO"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="java.net.URLEncoder"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport"
    content="width=device-width,initial-scale=1, shrink-to-fit=no">
<title>영화 추천 사이트</title>
<!-- 부트스트랩 CSS 추가하기 -->
<link rel="stylesheet" href="./css/bootstrap.min.css">
<!-- 커스텀 CSS 추가하기 -->
<link rel="stylesheet" href="./css/custom.css">
</head>
<body>
<%
    request.setCharacterEncoding("UTF-8");
    String movieGenre = "전체";
    String searchType = "최신순";
    String search = "";
    int pageNumber = 0;
    if (request.getParameter("movieGenre") != null) {
        movieGenre = request.getParameter("movieGenre");
    }
    if (request.getParameter("searchType") != null) {
        searchType = request.getParameter("searchType");
    }
%>
```

```

    }
    if (request.getParameter("search") != null) {
        search = request.getParameter("search");
    }
    if (request.getParameter("pageNumber") != null) {
        try {
            pageNumber = Integer.parseInt(request.getParameter("pageNumber"));
        } catch (Exception e) {
            out.println("검색 번호 오류");
        }
    }
}
String userID = null;
if (session.getAttribute("userID") != null) {
    userID = (String) session.getAttribute("userID");
}
%>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="index.jsp">영화 추천 사이트</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
        data-target="#navbar">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div id="navbar" class="collapse navbar-collapse">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item"><a class="nav-link" href="main.jsp">메인</a>
            </li>
            <li class="nav-item"><a class="nav-link" href="bbs.jsp">게시판</a>
            </li>
            <li class="nav-item active"><a class="nav-link"
                href="movieRecommendation.jsp">영화 추천</a></li>
        </ul>
        <%
            if(userID == null){
        %>
        <ul class="nav navbar-nav navbar-right">
            <li class="nav-item dropdown alignn right"><a

```

```

                class="nav-link dropdown-toggle" id="dropdown"
                data-toggle="dropdown"> 접속하기</a>
                <div class="dropdown-menu" aria-labelledby="dropdown">
                    <a class="dropdown-item" href="Login.jsp">로그인</a> <a
                        class="dropdown-item" href="userjoin.jsp">회원가입</a>
                </div></li>
            </ul>
        <%
            } else {
        %>
        <ul class="nav navbar-nav navbar-right">
            <li class="nav-item dropdown"><a
                class="nav-link dropdown-toggle" id="dropdown"
                data-toggle="dropdown"> 회원관리</a>
                <div class="dropdown-menu" aria-labelledby="dropdown">
                    <a class="dropdown-item" href="LogoutAction.jsp">로그아웃</a>
                </div></li>
            </ul>
        <%
            }
        %>
    </div>
</nav>

```



```

<section class="container">
  <form method="get" action="/movieRecommendation.jsp"
    class="form-inline mt-3">
    <select name="movieGenre" class="form-control mx-1 mt-2">
      <option value="전체">전체</option>
      <option value="액션">
        <% if(movieGenre.equals("액션")) out.println("selected"); %>>액션</option>
      <option value="애니메이션">
        <% if(movieGenre.equals("애니메이션")) out.println("selected"); %>>애니메이션</option>
      <option value="로맨스">
        <% if(movieGenre.equals("로맨스")) out.println("selected"); %>>로맨스</option>
      <option value="코미디">
        <% if(movieGenre.equals("코미디")) out.println("selected"); %>>코미디</option>
    </select> <select name="searchType" class="form-control mx-1 mt-2">
      <option value="최신순">최신순</option>
      <option value="추천순">
        <% if(searchType.equals("추천순")) out.println("selected"); %>>추천순</option>
    </select> <input type="text" name="search" class="form-control mx-1 mt-2"
      placeholder="내용을 입력하세요.">
    <button type="submit" class="btn btn-primary mx-1 mt-2">검색</button>
    <a class="btn btn-primary mx-1 mt-2" data-toggle="modal"
      href="#registerModal">등록</a>
  </form>
<%

```

```

ArrayList<movie1DTO> movieList = new ArrayList<movie1DTO>();
movieList = new movie1DAO().getList(movieGenre, searchType, search, pageNumber);
if (movieList != null){
  for (int i = 0; i < movieList.size(); i++) {
    if (i == 5)
      break;
    movie1DTO Movie = movieList.get(i);
%>
    <div class="card bg-light mt-3">
      <div class="card-header bg-light">
        <div class="row">
          <a class="col-8 text-left" href="<%= Movie.getMovieURL() %>">
            <%= Movie.getMovieTitle() %>&nbsp;<small>
              <%= Movie.getMovieDirector() %></small>
          </a>
          <div class="col-4 text-right">
            평점<span style="color: green;"><%= Movie.getMovieRating() %></span>
          </div>
        </div>
      </div>
      <div class="card-body">
        <div class="row no-gutters">
          <div class="col">
            <div class="card-block px-2">
              <h4 class="card-title">줄거리</h4>
              <p class="card-text"><%= Movie.getMovieContent() %></p>
            </div>
            <div class="row">
              <div class="col-9 text-left">
                상영 시간<span style="color: black;">:<
                  <%= Movie.getMovieTime() %></span>
                개봉일<span style="color: black;">:<
                  <%= Movie.getMovieReleasedate() %></span>
                <span style="color: green;">(추천: <%= Movie.getMovieLike() %>)

```

```

        </span>
      </div>
    </div>
    <div class="row-3 text-right">
      <a onclick="return confirm('추천하시겠습니까?')"
        href="/likeAction.jsp?movieID=<%=Movie.getMovieID()%>">추천</a>
      <a onclick="return confirm('삭제하시겠습니까?')"
        href="/removeAction.jsp?movieID=<%=Movie.getMovieID()%>">삭제</a>
    </div>
  </div>
</div>
</div>
</div>
<%
}
}
%>
</section>
<ul class="pagination justify-content-center mt-3">
  <li class="page-item">
    <%
    if (pageNumber <= 0) {
%> <a class="page-link disabled">이전</a> <%
    } else {
%> <a class="page-link"
      href="/movieRecommendation.jsp?movieGenre=<%= URLEncoder.encode(movieGenre, "UTF-8") %>
        &searchType=<%=URLEncoder.encode(searchType, "UTF-8")%>&search=
        <%=URLEncoder.encode(search, "UTF-8")%>
        &pageNumber=<%=pageNumber - 1%>">이전</a>
    <%
    }
%>

```

```

  </li>
  <li class="page-item">
    <%
    if (movielist.size() < 6) {
%> <a class="page-link disabled">다음</a> <%
    } else {
%> <a class="page-link"
      href="/movieRecommendation.jsp?movieGenre=<%= URLEncoder.encode(movieGenre, "UTF-8") %>
        &searchType=<%=URLEncoder.encode(searchType, "UTF-8")%>&search=
        <%=URLEncoder.encode(search, "UTF-8")%>
        &pageNumber=<%=pageNumber + 1%>">다음</a>
    <%
    }
%>
  </li>
</ul>

<div class="modal fade" id="registerModal" tabindex="-1" role="dialog"
  aria-labelledby="modal" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="modal">영화 등록</h5>
        <button type="button" class="close" data-dismiss="modal"
          aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form action="/movieAction.jsp" method="post">
          <div class="form-row">
            <div class="form-group col-sm-6">
              <label>제목</label> <input type="text" name="movieTitle"
                class="form-control" maxlength="20">
            </div>

```



```

        <div class="form-group col-sm-6">
            <label>감독</label> <input type="text" name="movieDirector"
                class="form-control" maxlength="20">
        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-sm-6">
            <label>상영시간</label> <input type="text" name="movieTime"
                class="form-control" maxlength="20">
        </div>
        <div class="form-group col-sm-6">
            <label>개봉일</label> <input type="text" name="movieReleasedate"
                class="form-control" maxlength="20">
        </div>
    </div>
    <div class="form-row">
        <div class="form-group col-sm-6" class="form-control">
            <label>장르</label> <select name="movieGenre" class="form-control">
                <option value="액션" selected>액션</option>
                <option value="애니메이션">애니메이션</option>
                <option value="로맨스">로맨스</option>
                <option value="코미디">코미디</option>
            </select>
        </div>
        <div class="form-group col-sm-6">
            <label>평점</label> <input type="text" name="movieRating"
                class="form-control" maxlength="20">
        </div>
    </div>
    <div class="form-group">
        <label>영화 주소</label>
        <textarea name="movieURL" class="form-control" maxlength="1024"
            style="height: 180px;"></textarea>
    </div>

```

```

    <div class="form-group">
        <label>줄거리</label>
        <textarea name="movieContent" class="form-control"
            maxlength="2048" style="height: 180px;"></textarea>
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-secondary"
            data-dismiss="modal">취소</button>
        <button type="submit" class="btn btn-primary">등록</button>
    </div>
</form>
</div>
</div>
</div>
<div>
<div class="bg-dark mt-4 p-3 text-center" style="color: #FFFFFF;">
    copyright &copy; 2021 SOFA All Rights Reserved. </div>
<!-- 제이쿼리 자바스크립트 추가하기 -->
<script src="./js/jquery.min.js"></script>
<!-- 파파 자바스크립트 추가하기 -->
<script src="./js/popper.js"></script>
<!-- 부트스트랩 자바스크립트 추가하기 -->
<script src="./js/bootstrap.min.js"></script>
</body>
</html>

```

메인 페이지 화면의 코드이며 영화 등록, 삭제, 검색, 추천 등의 기능들을 담고 있다.

▶ movieAction.jsp

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="movie.movieIDAO" %>
<%@ page import="movie.movieIDTO" %>
<%@ page import="java.io.PrintWriter"%>
<%
    request.setCharacterEncoding("UTF-8");
    String userID = null;
    if (session.getAttribute("userID") != null) {
        userID = (String) session.getAttribute("userID");
    }
    if (userID == null) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('로그인을 하세요. ')");
        script.println("location.href = 'login.jsp'");
        script.println("</script>");
    } else {
        request.setCharacterEncoding("UTF-8");
        String movieTitle = null;
        String movieDirector = null;
        String movieTime = null;
        String movieReleasedate = null;
        String movieGenre = null;
        String movieContent = null;
        String movieURL = null;
        String movieRating = null;

        if (request.getParameter("movieTitle") != null) {
            movieTitle = request.getParameter("movieTitle");
        }
        if (request.getParameter("movieDirector") != null) {
            movieDirector = request.getParameter("movieDirector");
        }
        if (request.getParameter("movieTime") != null) {
            movieTime = request.getParameter("movieTime");
        }
    }
}
```

```

if (request.getParameter("movieReleasedate") != null) {
    movieReleasedate = request.getParameter("movieReleasedate");
}
if (request.getParameter("movieGenre") != null) {
    movieGenre = request.getParameter("movieGenre");
}
if (request.getParameter("movieContent") != null) {
    movieContent = request.getParameter("movieContent");
}
if (request.getParameter("movieURL") != null) {
    movieURL = request.getParameter("movieURL");
}
if (request.getParameter("movieRating") != null) {
    movieContent = request.getParameter("movieRating");
}

if (movieTitle == null || movieDirector == null || movieTime == null ||
    movieReleasedate == null || movieGenre == null
    || movieContent == null) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('입력이 안된 사항이 있습니다.');"");
    script.println("history.back();"");
    script.println("</script>");
} else {
    movie1DAO Movie1DAO = new movie1DAO();
    int result = Movie1DAO.write(
        new movie1DTO(movieTitle, movieDirector, movieTime, movieReleasedate,
            movieGenre, movieContent, 0, 0, userID,
            movieURL, movieRating));

    if (result == -1) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('게시글 등록에 실패하였습니다.');"");
        script.println("history.back();"");
        script.println("</script>");
    } else {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("location.href = 'movieRecommendation.jsp'");
        script.println("</script>");
    }
}
}
}
}

```

%>

영화 등록을 위해 사용되는 jsp 파일이다.

비 로그인 상태에서 글 등록 시 로그인 창으로 돌려보내고, movieRecommendation.jsp에서 입력된 파라미터를 받아 정보가 입력되지 않은 사항이 있으면 ‘입력이 안 된 사항이 있습니다’라는 문구를 띄워주며 전 화면으로 보낸다. 또한, 등록 실패 시에도 문구를 띄우고 전 화면으로 이동한다.

► removeAction.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="movie.movie1DAO"%>
<%@ page import="user.UserDAO"%>
<%@ page import="likey.LikeyDTO"%>
<%@ page import="java.io.PrintWriter"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Movie-Movie</title>
</head>
<body>
<%
    String userID = null;
    if (session.getAttribute("userID") != null) {
        userID = (String) session.getAttribute("userID");
    }
    if (userID == null) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('로그인을 하세요. ')");
        script.println("location.href = 'login.jsp'");
        script.println("</script>");

    } else {

        request.setCharacterEncoding("UTF-8");
        String movieID = null;
        if (request.getParameter("movieID") != null) {
            movieID = request.getParameter("movieID");
        }
        movie1DAO movieDAO = new movie1DAO();

        if (userID.equals(movieDAO.getUserID(movieID))) {
            int result = new movie1DAO().delete(movieID);
            if (result == 1) {
                PrintWriter script = response.getWriter();
                script.println("<script>");
                script.println("alert('삭제되었습니다. ')");
                script.println("location.href = 'movieRecommendation.jsp'");
                script.println("</script>");

            } else {
                PrintWriter script = response.getWriter();
                script.println("<script>");
                script.println("alert('데이터베이스 오류!!')");
                script.println("history.back()");
                script.println("</script>");

            }
        } else {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('자신의 글만 삭제됩니다. ')");
            script.println("history.back()");
            script.println("</script>");

        }

    }
    %>
</body>
</html>
```



영화 등록을 삭제하기 위한 jsp 파일이다.

비 로그인 상태로 글을 삭제하려고 할 때 로그인 창으로 보내고 자신의 아이디와 게시글의 정보가 맞지 않으면 '자신의 글만 삭제됩니다.'라는 문구를 띄워 그 전 화면으로 되돌려 보낸다.

#### ▶ likeAction.jsp

---

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="movie.movie1DAO"%>
<%@ page import="user.UserDAO"%>
<%@ page import="likey.LikeyDAO"%>
<%@ page import="java.io.PrintWriter"%>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Movie-Movie</title>
</head>
<body>
<%
    String userID = null;
    if (session.getAttribute("userID") != null) {
        userID = (String) session.getAttribute("userID");
    }
    if (userID == null) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('로그인을 하세요.')");
        script.println("location.href = 'login.jsp'");
        script.println("</script>");

    } else {

        request.setCharacterEncoding("UTF-8");
        String movieID = null;
        if (request.getParameter("movieID") != null) {
            movieID = request.getParameter("movieID");
        }
        movie1DAO movieDAO = new movie1DAO();
        likeyDAO likeydao = new likeyDAO();
```

```

        likeyDAO likeydao = new likeyDAO();
        int result = new likeyDAO().like(userID, movieID);
        if (result == 1) {
            result = new movie1DAO().like(movieID);
            if (result == 1) {
                PrintWriter script = response.getWriter();
                script.println("<script>");
                script.println("alert('추천이 완료되었습니다.')");
                script.println("location.href = 'movieRecommendation.jsp'");
                script.println("</script>");

            } else {
                PrintWriter script = response.getWriter();
                script.println("<script>");
                script.println("alert('데이터베이스 오류!!')");
                script.println("history.back();");
                script.println("</script>");
            }
        } else {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('이미 추천한 글입니다.')");
            script.println("history.back();");
            script.println("</script>");
        }
    }
}
%>
</body>
</html>

```

추천을 위한 jsp 파일이다.

비 로그인 상태로 글을 삭제하려고 할 때 로그인 창으로 보내고, 이미 추천을 누른 글이라면 '이미 추천한 글입니다.'라는 문구를 띄워 그 전 화면으로 되돌려 보낸다.

► bbs.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="java.io.PrintWriter" %>
<%@ page import="bbs.BbsDAO" %>
<%@ page import="bbs.Bbs" %>
<%@ page import="java.util.ArrayList" %>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport" content="width=device, initial-scale=1">
<link rel="stylesheet" href="css/bootstrap.css">
<link rel="stylesheet" href="css/custom.css">
<title>Movie-Movie</title>
<style type="text/css">
    a, a:hover {
        color: #000000;
        text-decoration: none;
    }
</style>
</head>
<body>
    <%
        String userID = null;
        if(session.getAttribute("userID") != null) {
            userID = (String) session.getAttribute("userID");
        }
        int pageNumber = 1;
        if(request.getParameter("pageNumber") != null) {
            pageNumber = Integer.parseInt(request.getParameter("pageNumber"));
        }
    %>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="index.jsp">영화 추천 사이트</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div id="navbar" class="collapse navbar-collapse">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item">
                    <a class="nav-link" href="main.jsp">메인</a>
                </li>
                <li class="nav-item active">
                    <a class="nav-link" href="bbs.jsp">게시판</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="movieRecommendation.jsp">영화 추천</a>
                </li>
            </ul>
            <%
                if(userID == null){
            %>
            <ul class="nav navbar-nav navbar-right">
                <li class="nav-item dropdown alignn right">
                    <a class="nav-link dropdown-toggle" id="dropdown" data-toggle="dropdown">
                        접속하기</a>
                    <div class="dropdown-menu" aria-labelledby="dropdown">
                        <a class="dropdown-item" href="login.jsp">로그인</a>
                        <a class="dropdown-item" href="join.jsp">회원가입</a>
                    </div>
                </li>
            </ul>
            <%
                } else {
            %>
            <ul class="nav navbar-nav navbar-right">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" id="dropdown" data-toggle="dropdown">
                        회원관리</a>
                    <div class="dropdown-menu" aria-labelledby="dropdown">
                        <a class="dropdown-item" href="logoutAction.jsp">로그아웃</a>
                    </div>
                </li>
            </ul>
            <%
                }
            %>
        </div>
    </nav>
```





► update.jsp

```
<div class="container">
  <div class="row">
    <form method="post" action="updateAction.jsp?bbsID=<%= bbsID %>">
      <table class="table table-striped" style="text-align: center;
        border: 1px solid #dddddd">
        <thead>
          <tr>
            <th colspan="2" style="background-color: #eeeeee;
              text-align: center;">게시판 글수정 양식</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td><input type="text" class="form-control" placeholder="글 제목"
              name="bbsTitle" maxlength="50" value="<%= bbs.getBbsTitle() %>"></td>
          </tr>
          <tr>
            <td><textarea class="form-control" placeholder="글 내용"
              name="bbsContent" maxlength="2048" style="height: 350px;">
              <%= bbs.getBbsContent() %></textarea></td>
          </tr>
        </tbody>
      </table>
      <input type="submit" class="btn btn-primary pull-right" value="수정">
    </form>
  </div>
</div>
<footer class="bg-dark mt-4 p-3 text-center" style="color: #FFFFFF;">
  copyright &copy; 2021 SOFA All Rights Reserved.
</footer>
<script src="https://code.jquery.com/jquery-3.1.1.min.js"></script>
<script src="js/bootstrap.js"></script>
</body>
</html>
```

등록한 게시글의 수정을 위한 jsp 파일이다. 작성했던 게시글이 그대로 나타나게 하여 수정하기 쉽도록 제작하였다.

► updateAction.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="bbs.BbsDAO" %>
<%@ page import="bbs.Bbs" %>
<%@ page import="java.io.PrintWriter"%>
<% request.setCharacterEncoding("UTF-8"); %>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Movie-Movie</title>
</head>
<body>
<%
    String userID = null;
    if(session.getAttribute("userID") != null) {
        userID = (String) session.getAttribute("userID");
    }
    if(userID == null) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('로그인을 하세요.')");
        script.println("location.href = 'login.jsp'");
        script.println("</script>");
    }

    int bbsID = 0;
    if(request.getParameter("bbsID") != null) {
        bbsID = Integer.parseInt(request.getParameter("bbsID"));
    }
    if(bbsID == 0) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('유효하지 않은 글입니다.')");
        script.println("location href='bbs.jsp'");
        script.println("</script>");
    }
%>
```

```

Bbs bbs = new BbsDAO().getBbs(bbsID);
if(!userID.equals(bbs.getUserID())) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('권한이 없습니다.');"");
    script.println("location href='bbs.jsp'");
    script.println("</script>");
} else {
    if(request.getParameter("bbsTitle") == null || request.getParameter("bbsContent") == null
        || request.getParameter("bbsTitle").equals("") || request.getParameter("bbsContent").equals("")){
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('입력이 안 된 사항이 있습니다.');"");
        script.println("history.back();"");
        script.println("</script>");
    } else{
        BbsDAO bbsDAO = new BbsDAO();
        int result = bbsDAO.update(bbsID, request.getParameter("bbsTitle"), request.getParameter("bbsContent"));
        if(result == -1){
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글 수정에 실패했습니다.');"");
            script.println("history.back();"");
            script.println("</script>");
        }
        else{
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("location.href = 'bbs.jsp'");
            script.println("</script>");
        }
    }
}

%>
</body>
</html>

```

수정된 사항을 다시 DB에 저장하며, 자신의 글이 아닌 경우 게시글 수정을 할 수 없으며, 내용이 비어있는 경우 '입력이 안 된 사항이 있습니다'라는 문구를 띄워주어 수정할 수 있도록 해주는 jsp 파일이다.

▶ deleteAction.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="bbs.BbsDAO" %>
<%@ page import="bbs.Bbs" %>
<%@ page import="java.io.PrintWriter"%>
<% request.setCharacterEncoding("UTF-8"); %>

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Movie-Movie</title>
</head>
<body>
    <%
        String userID = null;
        if(session.getAttribute("userID") != null) {
            userID = (String) session.getAttribute("userID");
        }
        if(userID == null) {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('로그인을 하세요. ')");
            script.println("location.href = 'login.jsp'");
            script.println("</script>");
        }

        int bbsID = 0;
        if(request.getParameter("bbsID") != null) {
            bbsID = Integer.parseInt(request.getParameter("bbsID"));
        }
        if(bbsID == 0) {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('유효하지 않은 글입니다. ')");
            script.println("location href='bbs.jsp'");
            script.println("</script>");
        }
    %>
</body>
</html>
```

```

    }
    Bbs bbs = new BbsDAO().getBbs(bbsID);
    if(!userID.equals(bbs.getUserID())) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('권한이 없습니다.')");
        script.println("location href='bbs.jsp'");
        script.println("</script>");
    } else {
        BbsDAO bbsDAO = new BbsDAO();
        int result = bbsDAO.delete(bbsID);
        if(result == -1){
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글 삭제에 실패했습니다.')");
            script.println("history.back()");
            script.println("</script>");
        }
        else{
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("location.href = 'bbs.jsp'");
            script.println("</script>");
        }
    }
}

%>
</body>
</html>

```

게시판 글 삭제를 위한 jsp 파일이다.



► movie1DAO.java

---

```
package movie;

import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.util.ArrayList;
import java.sql.ResultSet;

public class movie1DAO {
    private Connection conn;
    private ResultSet rs;
    private PreparedStatement pstmt;

    public movie1DAO() {
        try {
            String dbURL = "jdbc:mysql://localhost:3306/MOVIERECOMMEND";
            String dbID = "root";
            String dbPassword = "as56824312@";
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
public int write(movie1DTO Movie1DTO) {
    PreparedStatement pstmt = null;
    try {
        String SQL = "INSERT INTO MOVIE VALUES (?, ?, ?, ?, ?, ?, ?, 0, NULL, ?, ?, ?)";
        pstmt = conn.prepareStatement(SQL);
        pstmt.setString(1, Movie1DTO.getMovieTitle());
        pstmt.setString(2, Movie1DTO.getMovieDirector());
        pstmt.setString(3, Movie1DTO.getMovieTime());
        pstmt.setString(4, Movie1DTO.getMovieReleasedate());
        pstmt.setString(5, Movie1DTO.getMovieGenre());
        pstmt.setString(6, Movie1DTO.getMovieContent());
        pstmt.setString(7, Movie1DTO.getUserID());
        pstmt.setString(8, Movie1DTO.getMovieURL());
        pstmt.setString(9, Movie1DTO.getMovieRating());
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1; // 데이터베이스 오류
}
```

```

public ArrayList<movieDTO> getList(String movieGenre, String searchType, String search, int pageNumber) {
    if (movieGenre.equals("전체")) {
        movieGenre = "";
    }
    ArrayList<movieDTO> movieList = new ArrayList<>();
    PreparedStatement pstmt = null;
    String SQL = "";
    try {
        if (searchType.equals("직접검색")) {
            SQL = "SELECT * FROM MOVIE WHERE movieGenre LIKE ? AND CONCAT"
                + "(movieTitle, movieDirector) LIKE ? ORDER BY movieID DESC LIMIT "
                + pageNumber * 5 + ", " + pageNumber * 5 + 6;
        } else if (searchType.equals("추천순")) {
            SQL = "SELECT * FROM MOVIE WHERE movieGenre LIKE ? AND CONCAT"
                + "(movieTitle, movieDirector) LIKE ? ORDER BY movieLike DESC LIMIT "
                + pageNumber * 5 + ", " + pageNumber * 5 + 6;
        }
        pstmt = conn.prepareStatement(SQL);
        pstmt.setString(1, "%" + movieGenre + "%");
        pstmt.setString(2, "%" + search + "%");
        rs = pstmt.executeQuery();
        movieList = new ArrayList<movieDTO>();
        while (rs.next()) {
            movieDTO movie = new movieDTO(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4),
                rs.getString(5), rs.getString(6), rs.getInt(7), rs.getInt(8), rs.getString(9),
                rs.getString(10), rs.getString(11));
            movieList.add(movie);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return movieList;
}

```

```

public int like(String movieID) {
    String SQL = "UPDATE MOVIE SET movieLike = movieLike +1 WHERE movieID = ?";

    try {
        pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, Integer.parseInt(movieID));
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (conn != null)
                conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (pstmt != null)
                pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (rs != null)
                rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return -1;
}

```

```

public int delete(String movieID) {
    String SQL = "DELETE FROM MOVIE WHERE movieID = ?";
    try {
        pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, Integer.parseInt(movieID));
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (conn != null)
                conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (pstmt != null)
                pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (rs != null)
                rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return -1;
}

```

```

public String getUserID(String movieID) {
    String SQL = "SELECT userID FROM MOVIE WHERE movieID = ?";
    String result = null;
    try {
        pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, Integer.parseInt(movieID));
        rs = pstmt.executeQuery();
        if (rs.next()) {
            result = rs.getString(1);
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        try {
            if (conn != null)
                conn.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (pstmt != null)
                pstmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        try {
            if (rs != null)
                rs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return result;
}
}

```



movie1DTO에서 변수를 받아오고 최신순, 추천순을 위한 쿼리문, 추천, 삭제, UserID를 위한 쿼리문이 담겨 있는 java 파일이다.

▶ movie1DTO.java

---

```
public class movie1DTO {  
    String movieTitle;  
    String movieDirector;  
    String movieTime;  
    String movieReleasedate;  
    String movieGenre;  
    String movieContent;  
    int movieLike;  
    int movieID;  
    String userID;  
    String movieURL;  
    String movieRating;  
}
```

movie table의 Column값을 담기 위한 java 파일이다.

## ▶ likeyDAO.java

```
package likey;

import java.sql.Connection;

public class likeyDAO {
    private Connection conn;
    private PreparedStatement pstmt;
    private ResultSet rs;

    public likeyDAO() {
        try {
            String dbURL = "jdbc:mysql://localhost:3306/MOVIERECOMMEND";
            String dbID = "root";
            String dbPassword = "as56824312@";
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public int like(String userID, String movieID) {
        String SQL = "INSERT INTO LIKEY VALUES (?, ?)";
        try {
            pstmt = conn.prepareStatement(SQL);
            pstmt.setString(1, userID);
            pstmt.setString(2, movieID);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {if(conn !=null) conn.close();} catch (Exception e) {e.printStackTrace();}
            try {if(pstmt !=null) pstmt.close();} catch (Exception e) {e.printStackTrace();}
            try {if(rs !=null) rs.close();} catch (Exception e) {e.printStackTrace();}
        }
        return -1; // 추천 중복
    }
}
```

추천을 위한 쿼리문이 담겨 있는 java 파일이다.

## ▶ likeyDTO.java

```
String userID;
int movieID;
```

likey table의 Column값을 저장하기 위한 java 파일이다.

► BbsDAO.java

```
package bbs;

import java.sql.Connection;

public class BbsDAO {
    private Connection conn;
    private ResultSet rs;

    public BbsDAO() {
        try {
            String dbURL = "jdbc:mysql://localhost:3306/MOVIERECOMMEND";
            String dbID = "root";
            String dbPassword = "as56824312@";
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public String getDate() {
        String SQL = "SELECT NOW()";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                return rs.getString(1);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ""; // 데이터베이스 오류
    }

    public int write(String bbsTitle, String userID, String bbsContent) {
        String SQL = "INSERT INTO BBS VALUES (?, ?, ?, ?, ?, ?)";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setInt(1, getNext());
            pstmt.setString(2, bbsTitle);
            pstmt.setString(3, userID);
            pstmt.setString(4, getDate());
            pstmt.setString(5, bbsContent);
            pstmt.setInt(6, 1);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1; // 데이터베이스 오류
    }

    public ArrayList<Bbs> getList(int pageNumber){
        String SQL = "SELECT * FROM BBS WHERE bbsID < ? AND bbsAvailable = 1 ORDER BY bbsID DESC LIMIT 10;";
        ArrayList<Bbs> list = new ArrayList<Bbs>();
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setInt(1, getNext() - (pageNumber - 1) * 10);
            rs = pstmt.executeQuery();
            while(rs.next()) {
                Bbs bbs = new Bbs();
                bbs.setBbsID(rs.getInt(1));
                bbs.setBbsTitle(rs.getString(2));
                bbs.setUserID(rs.getString(3));
                bbs.setBbsDate(rs.getString(4));
                bbs.setBbsContent(rs.getString(5));
                bbs.setBbsAvailable(rs.getInt(6));
                list.add(bbs);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```

        return list;
    }

    public boolean nextPage(int pageNumber) {
        String SQL = "SELECT * FROM BBS WHERE bbsID < ? AND bbsAvailable = 1;";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setInt(1, getNext() - (pageNumber - 1) * 10);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                return true;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }

    public Bbs getBbs(int bbsID) {
        String SQL = "SELECT * FROM BBS WHERE bbsID = ?;";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setInt(1, bbsID);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                Bbs bbs = new Bbs();
                bbs.setBbsID(rs.getInt(1));
                bbs.setBbsTitle(rs.getString(2));
                bbs.setUserID(rs.getString(3));
                bbs.setBbsDate(rs.getString(4));
                bbs.setBbsContent(rs.getString(5));
                bbs.setBbsAvailable(rs.getInt(6));
                return bbs;
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public int update(int bbsID, String bbsTitle, String bbsContent) {
        String SQL = "UPDATE BBS SET bbsTitle = ?, bbsContent = ? WHERE bbsID = ?;";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setString(1, bbsTitle);
            pstmt.setString(2, bbsContent);
            pstmt.setInt(3, bbsID);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1; // 데이터베이스 오류
    }

    public int delete(int bbsID) {
        String SQL = "UPDATE BBS SET bbsAvailable = 0 WHERE bbsID = ?;";
        try {
            PreparedStatement pstmt = conn.prepareStatement(SQL);
            pstmt.setInt(1, bbsID);
            return pstmt.executeUpdate();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return -1; // 데이터베이스 오류
    }
}

```

게시판에서 사용되는 다양한 기능을 수행하기 위한 java 파일이다.

▶ BbsDAO.java

---

```
private int bbsID;  
private String bbsTitle;  
private String userID;  
private String bbsDate;  
private String bbsContent;  
private int bbsAvailable;
```

bbs table의 Column값을 담기 위한 java 파일이다.