



Netscore

프로젝트 팀원 : 강현, 이한얼

목차

1. 프로젝트 기획

2. 데이터 베이스 구성도

3. 프로젝트 Layout

4. 웹사이트 서비스에 대한 설명



프로젝트 기획

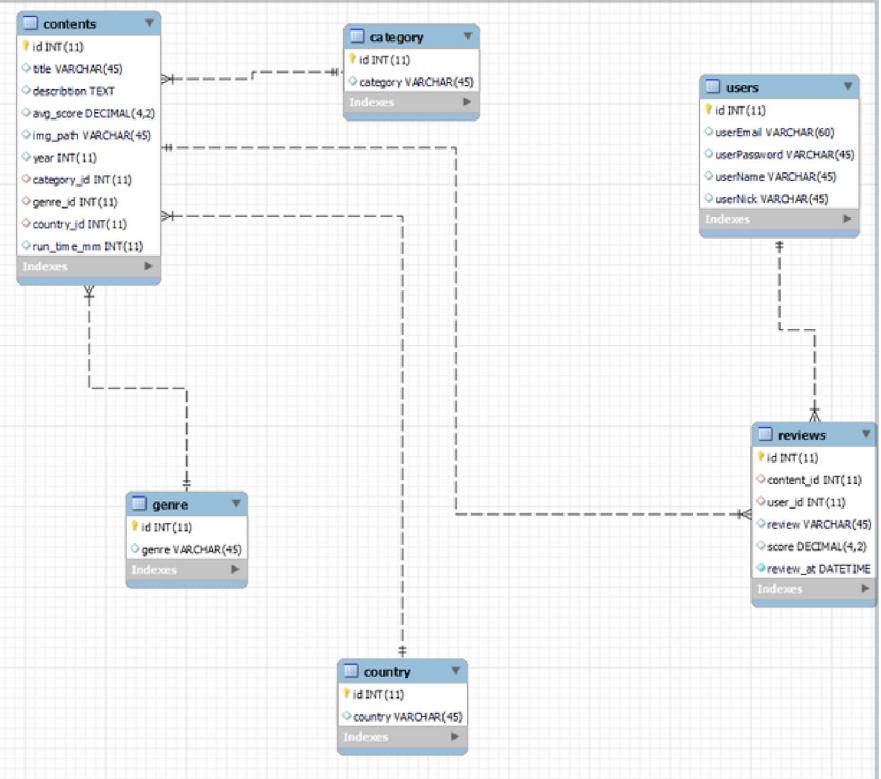
넷플릭스 이용자가 넷플릭스에서 제공되는 컨텐츠의 시청 후기 및 평가 사이트 만들기



데이터 베이스 구성도

주요 테이블:
Users 및 Contents 테이블

Users 테이블과 Contents 테이블에
참조될 컬럼을 FK로 연결하여 DB 구
성



개발 환경

1. 프로그램 툴 : Eclipse IDE 4.25
2. 웹 서버: Apache tomcat 9.0
3. JDK: Amazon Correto jdk 11
4. Jquery 버전 : <https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js>
5. bootstrap 버전 : <https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js>
6. 데이터 베이스 : Mysql Workbench 8.0 CE
7. 운영체제: Windows 10
8. 언어: JAVA , SQL



역할 분담

이한얼

강현

카테고리 리스트 불러오기

세션 설정

로그인 및 로그아웃

Best TOP 10 리스트

컨텐츠 상세보기

페이지 웹 디자인

리뷰 작성

리뷰 삭제

검색

별점 기능 구현

DB 구현

DB 구현



Layout

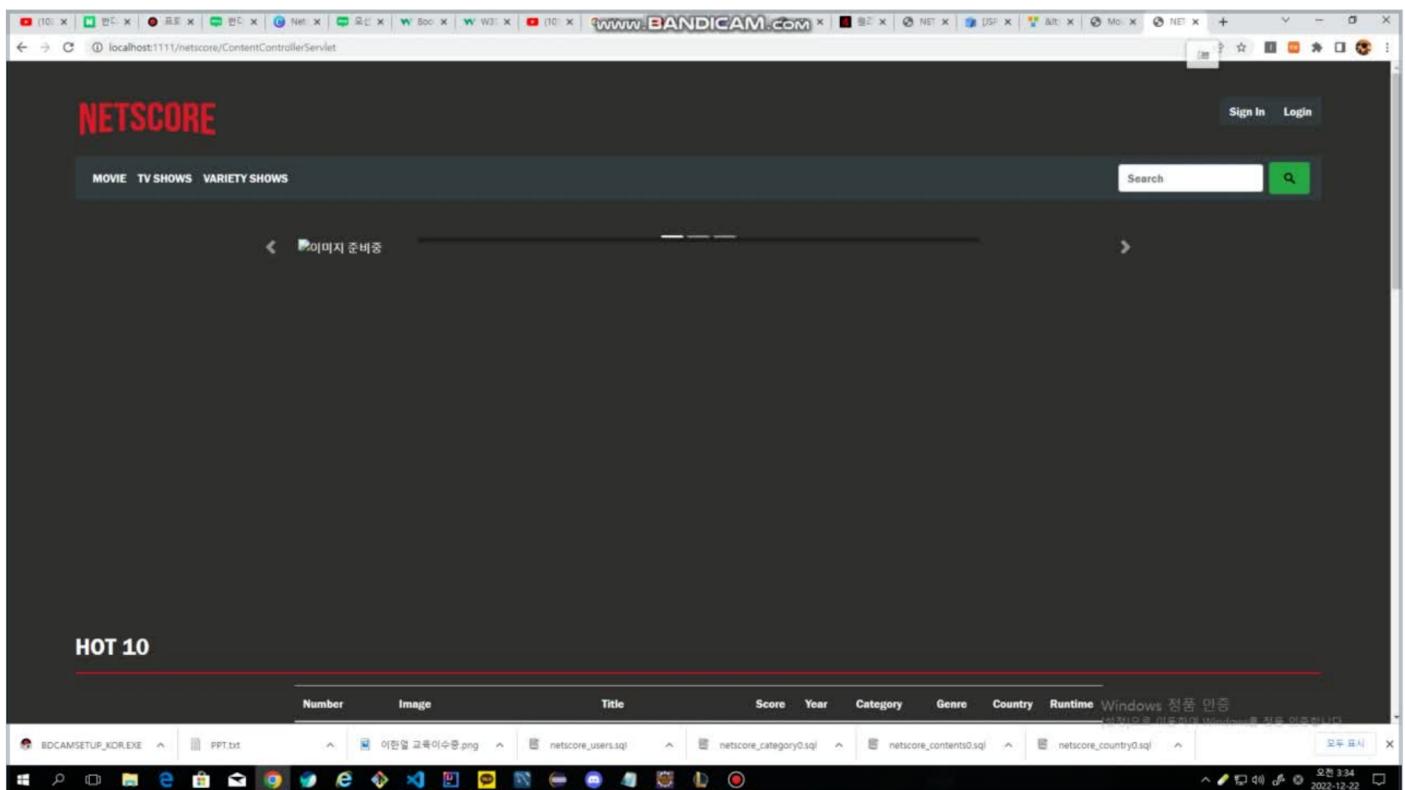
1. 메인 페이지에 공통적으로 적용될 디자인 및 테이블 디자인은 main.css에
입력
2. Slide 기능과 Dropdown 등의 페이지 기능 구성에 추가될 내용은 jquery와
bootStrap을 사용

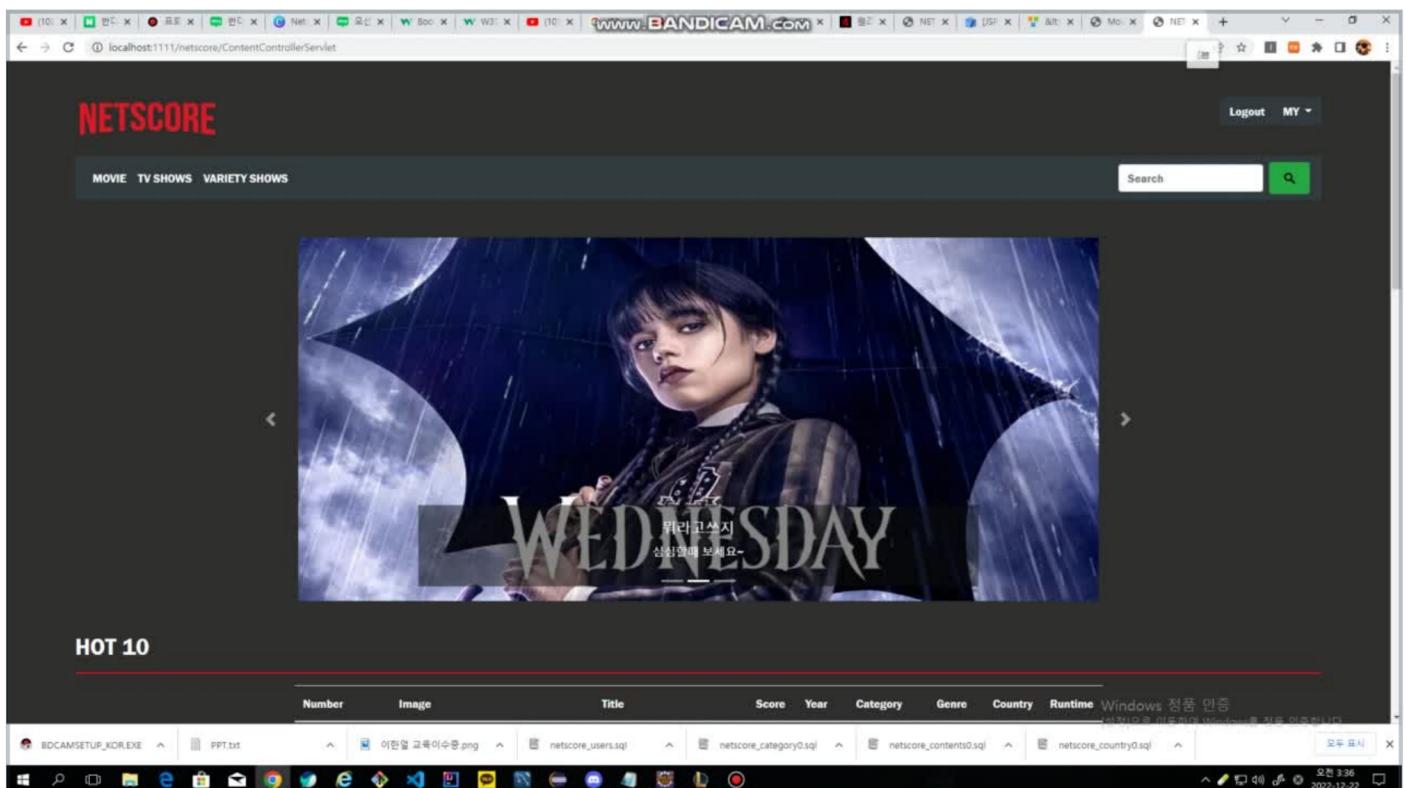


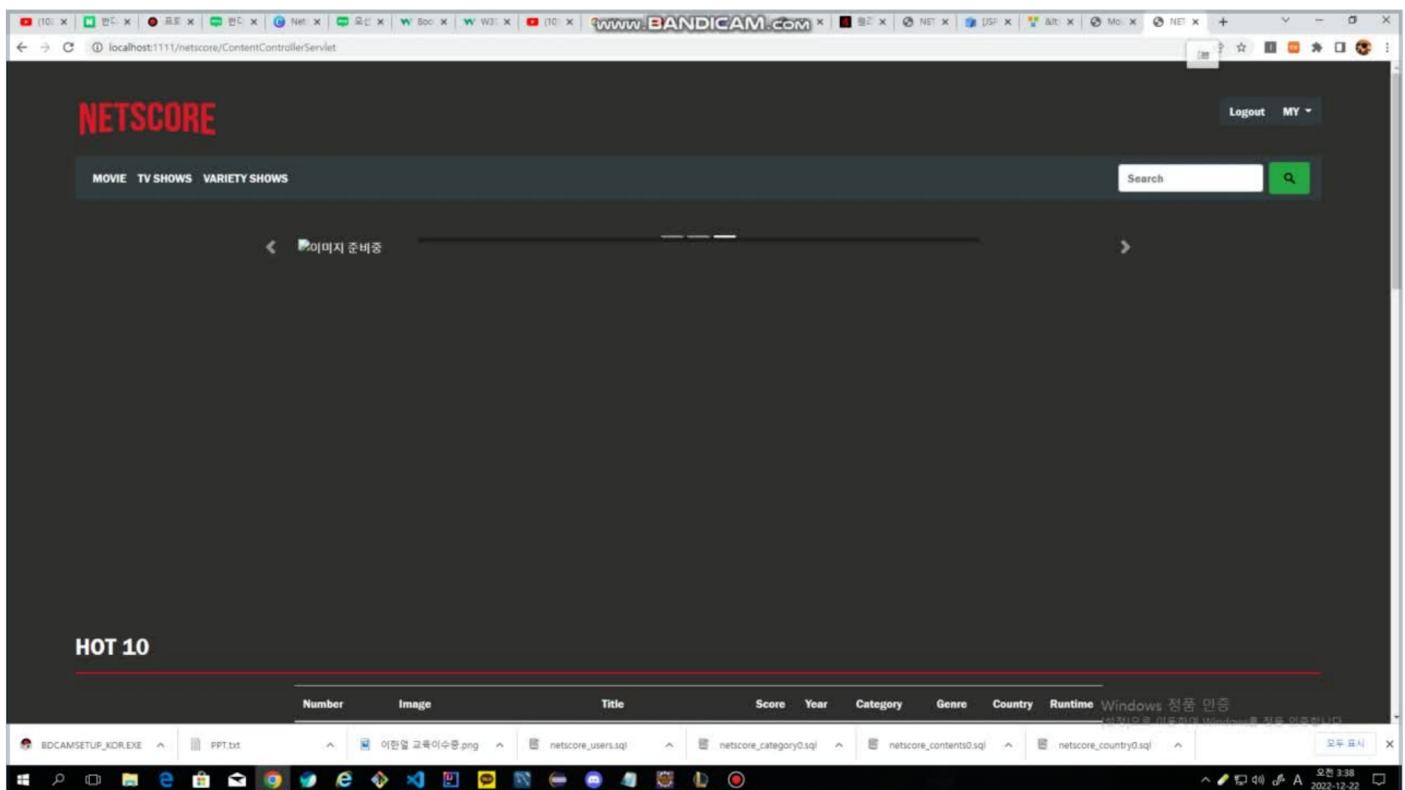
주요 구현 코드

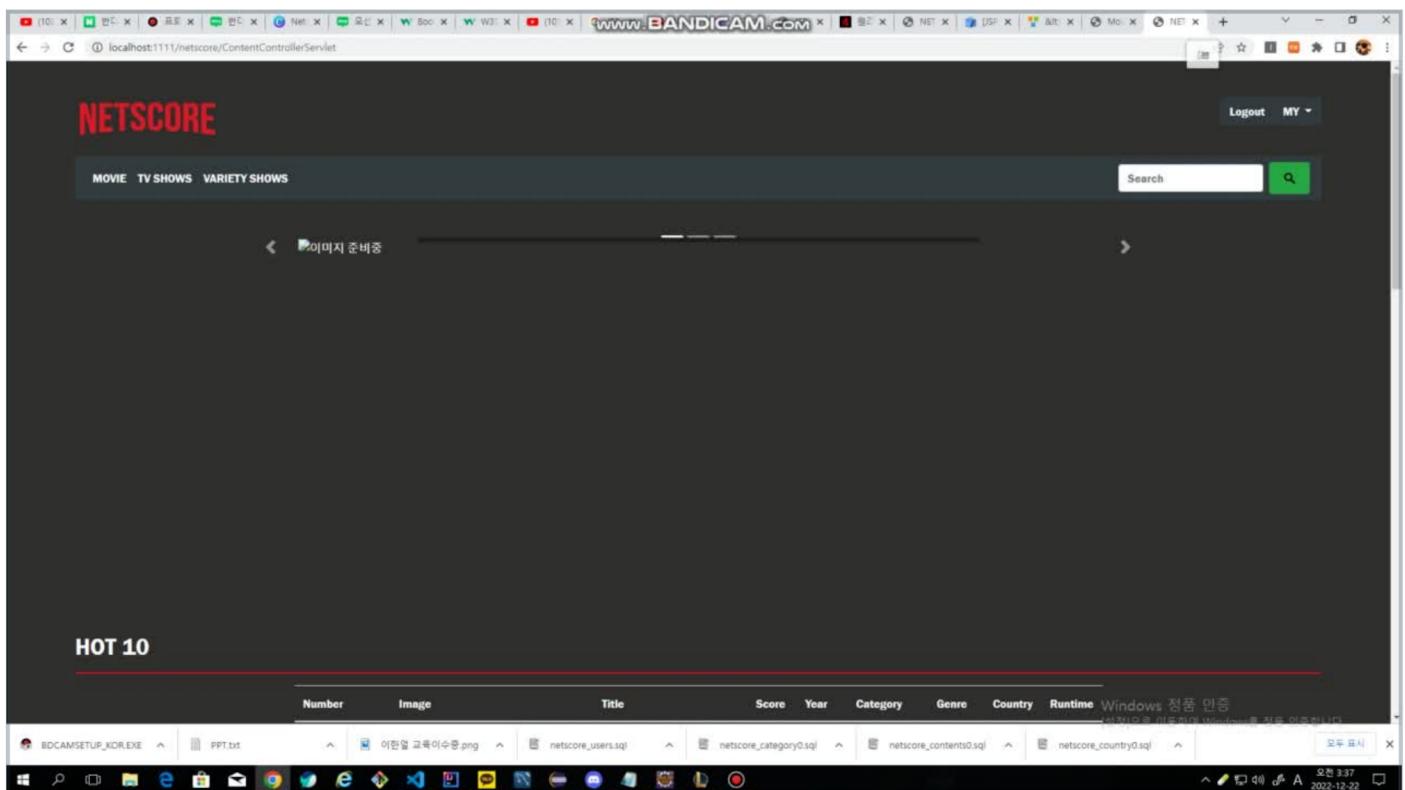
1. 로그인 및 회원가입
2. Best TOP 10
3. 카테고리별 리스트
4. 리뷰 작성 및 삭제
5. 검색 및 로그아웃

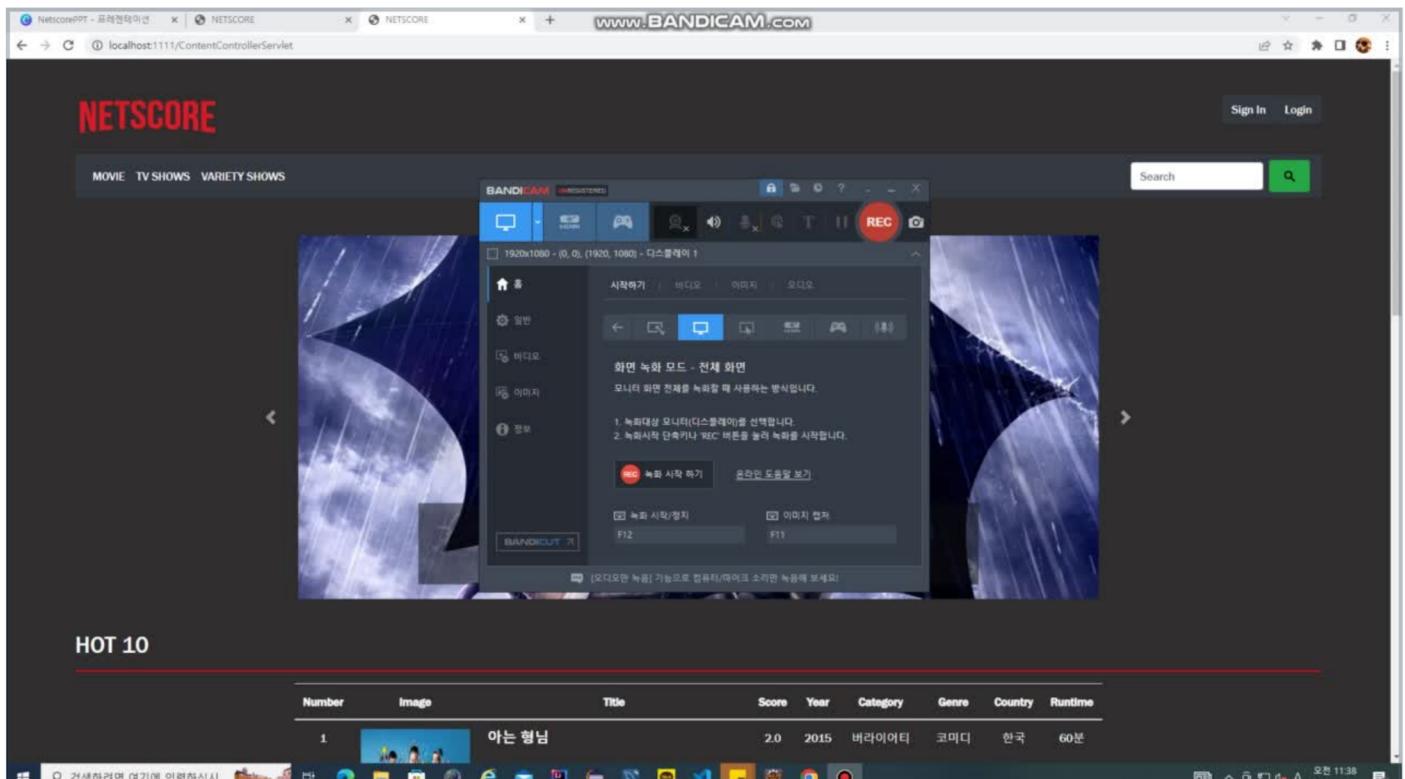
The screenshot shows the homepage of the NETSCORE website. At the top, there is a navigation bar with 'NETSCORE' in red, 'Sign In', and 'Login'. Below the navigation is a search bar with a green search icon. The main content area features a large image of Wednesday Addams from the TV show 'Wednesday'. Overlaid on the image is the title 'WEDNESDAY' in large white letters, with Korean subtitles '뭐라고쓰지' and '심심할때 보세요~' below it. At the bottom left of the image, it says 'Weekly HOT'.











JSP MVC 구성

M : Contents

M: ContentsDbUtil

C : ContentControllerServlet

V : WebContents.jsp



Viewer의 JSP 목록

```
netcore
  Deployment Descriptor: netcore
  JAX-WS Web Services
  Java Resources
  build
  WebContent
    css
    images
    META-INF
    WEB-INF
      Headline.jsp
      index.html
      JoinForm.jsp
      List.jsp
      LoginAction.jsp
      LoginForm.jsp
      LogoutAction.jsp
      Main.jsp
      Review.jsp
```

Model 및 Controller 목록

```
netcore
  Deployment Descriptor: netcore
  JAX-WS Web Services
  Java Resources
    src
      com.web.jdbc
        ContentControllerServlet.java
        ContentDbUtil.java
        Contents.java
        testServlet.java
    Libraries
    build
    WebContent
```

MVC Model 예시

```
6   //-----contents values-----
7   //minutes
8
9
10 public Contents(int content_id, String title, String describtion, float avg_score, String img_path, int year, int category_id,
11   int genre_id, int country_id, int run_time) {
12   super();
13   this.content_id = content_id;
14   this.title = title;
15   this.describtion = describtion;
16   this.avg_score = avg_score;
17   this.img_path = img_path;
18   this.year = year;
19   this.category_id = category_id;
20   this.genre_id = genre_id;
21   this.country_id = country_id;
22   this.run_time = run_time;
23 }
24 private int content_id;
25
26 private String title;
27
28 private String describtion;
29
30 private float avg_score;
31
32 private String img_path;
33
34 private int year;
35
36 private int category_id;      // 1.movie 2.series 3.variety
```

MVC Model 예시

DB의 contents의 LIST를 불러오는 메소드

```
public List<Contents> getContents(String sortBy) throws Exception {
    List<Contents> contents = new ArrayList<>();
    StringBuider stb = new StringBuider("SELECT * FROM contents");
    stb.append(sortBy);
    String sql = stb.toString();
    try {
        conn = dataSource.getConnection();
        mySt = conn.createStatement();
        myRs = mySt.executeQuery(sql);

        while(myRs.next()) {
            int content_id = myRs.getInt("id");
            String title = myRs.getString("title");
            String describtion = myRs.getString("describtion");
            float avg_score = myRs.getFloat("avg_score");
            String img_path = myRs.getString("img_path");
            int year = myRs.getInt("year");
            int category_id = myRs.getInt("category_id");
            int genre_id = myRs.getInt("genre_id");
            int country_id = myRs.getInt("country_id");
            int run_time = myRs.getInt("run_time_mm");
            Contents tempContents = new Contents(content_id, title, describtion, avg_score, img_path, year, category_id, genre_id);
            contents.add(tempContents);
        }
        return contents;
    }finally {
        close(conn, mySt, myRs);
    }
}
```

MVC Viewer 예시

Headline.jsp 파일에 화면 이동 시 공통적으로 표현될 부분을 정리하여 각 화면에 includ 하는 방식을 선택

```
① *ContentControllerServlet.java ② *Contents.java ③ ContentDbUtil.java ④ Headline.jsp X
it src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
it src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundle.min.js"><script type="text/css" rel="stylesheet" href="css/Main.css">
>NETSCORE</title>

headline -->
:klass="head-line">
!-- Go main page btn-->
| href="ContentControllerServlet">
| --- User's btn -->

| if(session.getAttribute("userInfo") == null){
|   <div class="btn-group" style="float:right;">
|     <a href="JoinForm.jsp"><button type="button" class="btn btn-dark">Sign In</button></a>
|     <a href="LoginForm.jsp"><button type="button" class="btn btn-dark">Login</button></a>
|   </div>
| }else if(session.getAttribute("userInfo") != null){
|   <div class="btn-group" style="float:right;">
|     <a href="LogoutAction.jsp"><button type="button" class="btn btn-dark">Logout</button>
|     <div class="btn-group">
|       <button type="button" class="btn btn-dark dropdown-toggle" data-toggle="dropdown" MY
|       </button>
|       <div class="dropdown-menu">
|         <a class="dropdown-item" href="LogoutAction.jsp">Logout</a>
|       </div>
|     </div>
|   </div>
| }</div>
```

MVC Viewer 예시

Main.jsp 파일에 linclud 활용한 예

```
web.xml *ContentControllerServlet.java *Contents.java ContentDbUtil.java Headline.jsp Main.jsp X
19= <body>
20
21     <!-- Headline -->
22     <%@ include file="Headline.jsp" %>
23 <%--    <jsp:include page="Headline.jsp"/>    --%>
```

MVC Viewer 예시

```
ContentControllerServlet.java  *Contents.java  ContentDbUtil.java  Headline.jsp X
```

```
"navbar navbar-expand-lg navbar-dark bg-dark" style="margin:20px 0;">>  
  <div class="collapse navbar-collapse" id="navb">  
    <ul class="navbar-nav mr-auto">  
      <li url var="movie" value="/ContentControllerServlet">  
        <c:param name="command" value="LIST"/>  
        <c:param name="category_id" value="1"/>  
      :url>  
      <li url var="series" value="/ContentControllerServlet">  
        <c:param name="command" value="LIST"/>  
        <c:param name="category_id" value="2"/>  
      :url>  
      <li url var="variety" value="/ContentControllerServlet">  
        <c:param name="command" value="LIST"/>  
        <c:param name="category_id" value="3"/>  
      :url>  
        <a class="nav-item active">  
          <a class="nav-link" href="#">MOVIE</a>  
        </a>  
        <a class="nav-item active">  
          <a class="nav-link" href="#">TV SHOWS</a>  
        </a>  
        <a class="nav-item active">  
          <a class="nav-link" href="#">VARIETY SHOWS</a>
```

category_id 파라미터에 VALUE값을 입력하여
하나의 페이지에서 카테고리별 리스트가 표출 가능하도록 설정

MVC Controller 예시

```
ContentControllerServlet.java × Contents.java ContentDbUtil.java
see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("utf-8");
    try{
        String theCommand = request.getParameter("command");

        if(theCommand == null) {
            theCommand="MAIN";
        }
        switch(theCommand) {
        case "MAIN":
            listPopular(request, response);
            break;
        case "LIST":
            listContents(request, response);
            break;
        case "SEARCH":
            searchContents(request, response);
            break;
        case "LOAD":                                //load review page
            loadReviews(request, response);
            break;
        case "DELETE":                               //delete review & refresh
            deleteReviews(request, response);
            break;
        }
    }catch(Exception e) {
        throw new ServletException(e);
    }
}
```

MVC Controller 예시

```
web.xml *ContentControllerServlet.java *Contents.java ContentDbUtil.java
67
68    }
69
70    // contents list ordered by category(1.movie, 2.series, 3.variety)
71    private void listContents(HttpServletRequest request, HttpServletResponse response) throws Exception {
72        String theCategory_id = request.getParameter("category_id");
73        try {
74            ListContents contents = contentDBUtil.getContents(" WHERE category_id = "+theCategory_id);
75            request.setAttribute("contents_list", contents);
76        }catch(Exception e) {
77            e.printStackTrace();
78        }
79        RequestDispatcher dispatcher = request.getRequestDispatcher("/List.jsp");
80        dispatcher.forward(request, response);
81    }
82
83    private void deleteReview(HttpServletRequest request, HttpServletResponse response) throws Exception {
84        int theReview_id = Integer.parseInt(request.getParameter("review_id"));
85        int theContent_id = Integer.parseInt(request.getParameter("content_id"));
86        contentDBUtil.deleteReview(theReview_id);
87        response.sendRedirect(request.getContextPath() + "/ContentControllerServlet?command=LOAD&content_id=" + theContent_id);
88    }
89
90
91
92    // shows content's inform and reviews
93    private void loadReviews(HttpServletRequest request, HttpServletResponse response) throws Exception {
94        String theContent_id = request.getParameter("content_id");
95        try {
96            ListContents content_info = contentDBUtil.getContents(" WHERE id = " + theContent_id);
97            request.setAttribute("content_info", content_info);
98            ListContentReviews = contentDBUtil.getReviews(" WHERE content_id = " + theContent_id);
99            request.setAttribute("content_reviews", reviews);
100       }catch(Exception e) {
101           e.printStackTrace();
102       }
103   }
```

DB 연동 방식

Mysql에 생성된 DB를 Resource로 설정 후 계정 정보와 리턴값 및 driverClass를 Mysql로 설정

```
web.xml    X server.xml    X context.xml    testServlet.java
Bind to grammar/schema...
└─ <Context>
?
  └─ <Resource name="jdbc/netscore"
    auth="Container" type="javax.sql.DataSource"
    maxActive="20" maxIdle="5" maxWait="10000"
    username="webstudent" password="webstudent"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://localhost:3306/netscore?useSSL=false&serv
  >
</Context>
|
```

DB 연동 방식

```
web.xml    server.xml    context.xml    testServlet.java    ContentControllerServlet.java ×
1 package com.web.jdbc;
2
3+ import java.io.IOException;[]
14
15+ /**
16  * Servlet implementation class ContentControllerServlet
17  */
18 @WebServlet("/ContentControllerServlet")
19 public class ContentControllerServlet extends HttpServlet {
20     private static final long serialVersionUID = 1L;
21
22     private ContentDbUtil contentDbUtil;
23
24+     @Resource(name = "jdbc/netscore")
25     private DataSource dataSource;
26
27
28+     @Override
29     public void init() throws ServletException {
```

로그인의 세션 설정

```
web.xml  LogoutAction.jsp  LoginAction.jsp  ContentControllerServlet.java ×
...
31
32
33④ private void checkUser(HttpServletRequest request, HttpServletResponse response) throws Exception {
34     String checkEmail = request.getParameter("userEmail");
35     String checkPassword = request.getParameter("userPassword");
36     List<Contents> user = contentDbUtil.getUser(checkEmail);
37     try {
38         if(user.isEmpty()) {                                // no matching email
39             request.setAttribute("loginCheck", "wrong_email");
40         }else if(!user.get(0).getUserPassword().equals(checkPassword)){    // wrong password
41             request.setAttribute("loginCheck", "wrong_password");
42         }else if(user.get(0).getUserPassword().equals(checkPassword)) {    // match password
43             request.setAttribute("loginCheck", user);
44         }else {
45             request.setAttribute("loginCheck", "wrong_something");
46         }
47     }catch(Exception e) {
48         e.printStackTrace();
49     }
50     RequestDispatcher dispatcher = request.getRequestDispatcher("/LoginAction.jsp");
51     dispatcher.forward(request, response);
52 }
```

로그인의 세션 연동

PrintWriter를 import하여 서블릿에서 설정해준 파라미터 값으로 조건문 설정 후

파라미터 값으로 로그인과 세션 설정 과정

```
12
13@ <%
14     PrintWriter script = response.getWriter();
15     if(request.getAttribute("loginCheck") == "wrong_email"){
16         script.println("<script>");
17         script.println("alert('THERE IS NO MATCHING EMAIL')");
18         script.println("history.back()");
19         script.println("</script>");
20     }else if(request.getAttribute("loginCheck") == "wrong_password"){
21         script.println("<script>");
22         script.println("alert('MIGHT BE WRONG PASSWORD')");
23         script.println("history.back()");
24         script.println("</script>");
25     }else if(request.getAttribute("loginCheck") == "wrong_something"){
26         script.println("<script>");
27         script.println("alert('TRY AGAIN')");
28         script.println("history.back()");
29         script.println("</script>");
30     }else if(request.getAttribute("loginCheck") == "same_email"){
31         script.println("<script>");
32         script.println("alert('The email is already exists')");
33         script.println("history.back()");
34         script.println("</script>");
35     }else{
```

로그아웃의 세션 설정

|Logout시 세션도 종료되도록 session.invalidate() 입력 후 팝업 생성 설정

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <%@ page import="java.util.* , java.io.PrintWriter, com.web.jdbc.*" %>
4 <!DOCTYPE html>
5<html>
6<head>
7 <meta charset="UTF-8">
8 <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.1/dist/jquery.slim.min.js"></script>
9 <title>NETSCORE</title>
10 </head>
11<body>
12
13 <%session.invalidate();%>
14
15<script>
16 alert('See you agin');
17 location.href = 'ContentControllerServlet';
18 </script>
19
```

감사합니다.

강현, 이한얼

