

# JSP 프로젝트 결과보고서



3조

20190659 김민지

20200934 이단비

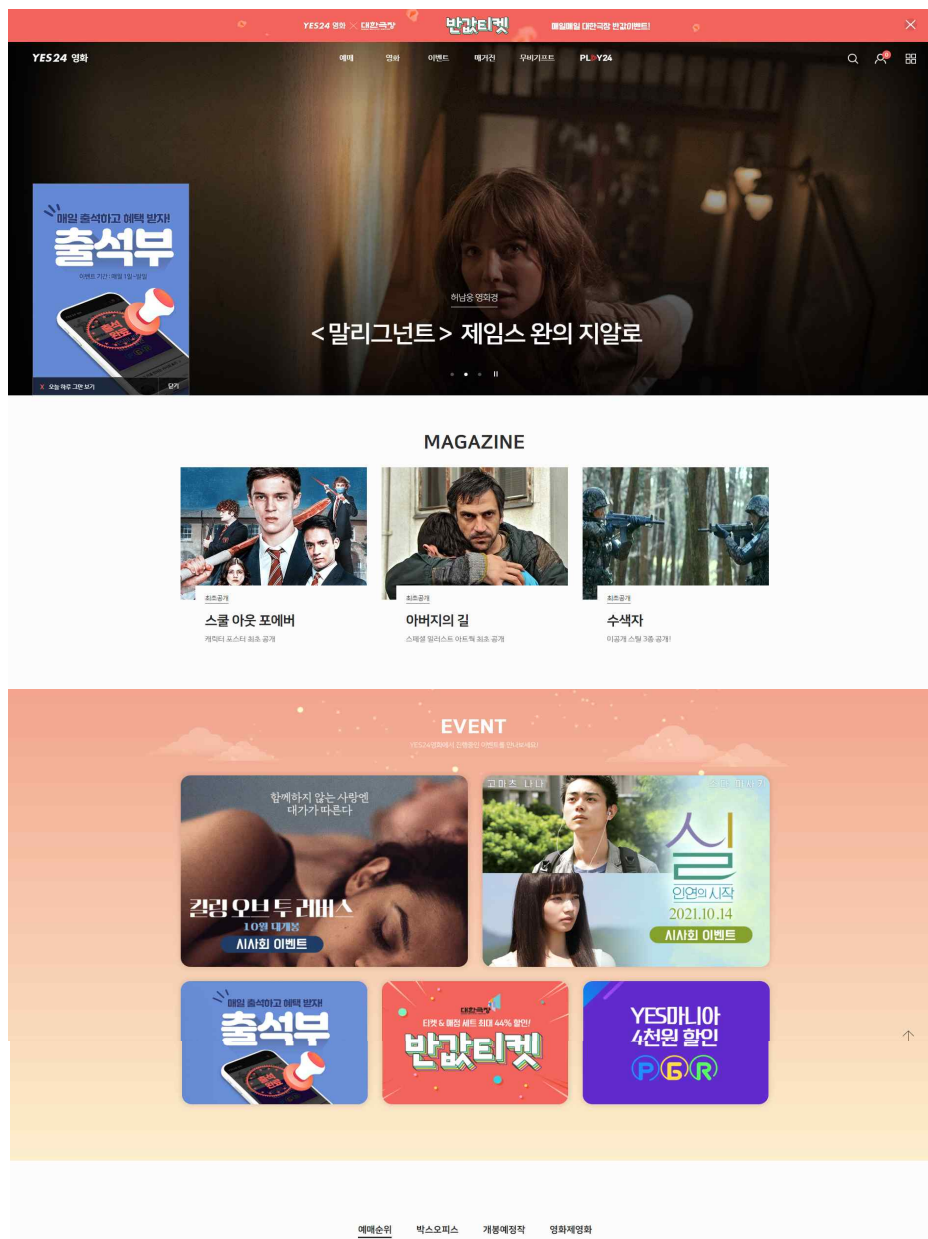
20202888 임시은

## 가. 프로젝트 개요

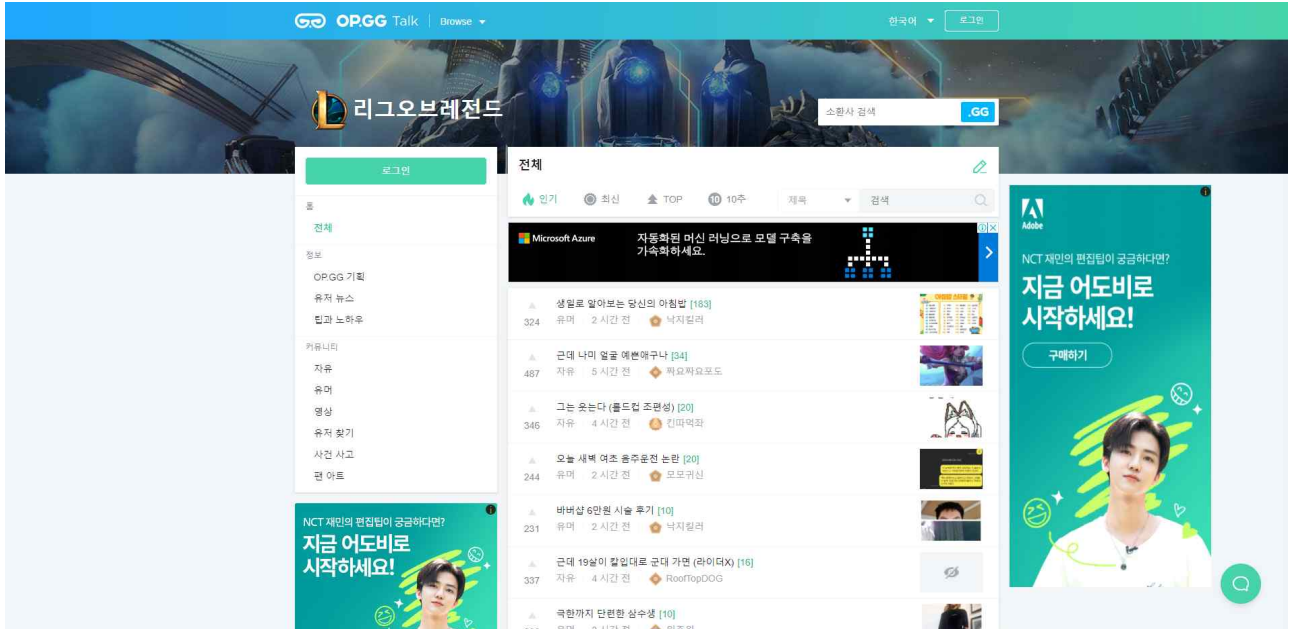
### 1.1. 기획의도

각각의 영화 예매 사이트를 통합하여 상영시간을 한 눈에 비교하고 사이트 내의 회원가입을 간편하게 할 수 있는 기능으로 기존의 영화 예매 사이트를 보완한다. 또한 커뮤니티 기능을 추가해 영화 관람 후 소통할 수 있는 공간을 만들고 싶어 기획하게 되었다.

#### 1.1.1. 벤치마킹



❖ 에스24 영화 : <https://movie.yes24.com/>



❖ 리그오브레전드 전적 검색(커뮤니티) : <https://talk.op.gg/s/lol/all?sort=popular>

### 1.1.2. 주요 고객

영화를 관람하거나 흥미 또는 좋아하는 사람을 대상

### 1.1.3. 주요 기능

- ◆ 각 영화관 사이트(CGV, 롯데시네마, 메가박스 등) 통합하여 예매
- ◆ 예매권/할인권 안내 (카드사 혜택 등)
- ◆ 이벤트 (시사회, 쿠폰 할인, 출석)
- ◆ 영화 뉴스(개봉 정보, 관객 몇백만명 돌파 등 다양한 이슈 뉴스)
- ◆ 극장 할인티켓 세트
- ◆ 영화 소개 (예매 순위, 현재상영작, 박스오피스, 영화제영화)
- ◆ 각각의 특정 영화 소개 (기본정보, 배우/제작진, 동영상, 포토, 관련기사, 평점(리뷰))
- ◆ 영화 커뮤니티 게시판

## 1.2 제작기술

JSP & Servlet, Javascript, HTML, CSS3, SQL, ajax, json

### 1.3 추진단계

[illegible]

## 나. 프로젝트 구현

### 2.1. 기능 구조



① **회원**: 영화 예매, 마이페이지, 커뮤니티 글 작성 및 삭제, 커뮤니티 게시물 조회

② **비회원**: 페이지 열람 가능

③ **관리자**: 모든 페이지의 기능 권한. 공지사항 작성 및 수정, 회원 관리(수정, 삭제), 커뮤니티 게시물 관리 마이페이지 관리, 예매 관리

## 2.2. 메뉴별 권한

구성	세부메뉴	내용	접근 권한		
			비회원	회원	관리자
메인 페이지	레이아웃	회원가입	●	●	●
		로그인, 마이 페이지, 영화 예매		●	●
	예매	예매 정보 (비회원은 예매 열람만 가능)	●	●	●
	영화	영화 정보	●	●	●
	이벤트	시사회 및 이벤트 정보 (비회원은 열람만 가능)	●	●	●
	매거진	뉴스 정보	●	●	●
	마이페이지	예매 내역, 개인정보(등급) 확인		●	●
	커뮤니티	글쓰기, 수정, 삭제 (관리자만 수정 가능)		●	●
		비회원은 관리자 공지사항 및 인기게시물 top5만 열람 가능	●	●	●
관리자	회원 관리	회원 열람, 수정 삭제 회원 등급 조정			●
	게시글 관리	공지사항 등록, 수정 삭제 게시글 등록, 삭제			
	커뮤니티 관리	비속어 사용 및 심한 스포일러 문구 삭제			
	예매 관리	영화 예매 내역 조회			

## 2.3. 메뉴별 권한

### 2.3.1. DataBase명

- movieSite

### 2.3.2. Table명

- adminwrite, communiry, movie, membermovie, mttbl, tickettbl

### 2.3.3. 테이블 상세

테이블명	adminwrite	Table 기술서	작성일	2021/12/06
			작성자	임시은
테이블 설명	adminwrite			

No	Attribute	Data Type	Null	Ky	Default	Description
1	adminRank	int	N			공지사항 게시글 번호
2	adminTitle	varchar(50)	N	PK		공지사항 제목(타이틀)
3	adminID	varchar(20)	N			공지사항 작성자(관리자)
4	adminDate	datetime	N			공지사항 게시글 작성 날짜
5	adminContent	varchar(2048)	N			공지사항 내용
6	adminAvailable	int	N			공지사항 작성 접근 번호

테이블명	community	Table 기술서	작성일	2021/12/06
			작성자	임시은
테이블 설명	community			

No	Attribute	Data Type	Null	Ky	Default	Description
1	cmRank	int	N			커뮤니티 게시글 번호
2	cmTitle	varchar(50)	N	PK		커뮤니티 제목(타이틀)
3	cmID	varchar(20)	N			커뮤니티 작성자(관리자 및 회원)
4	cmDate	datetime	N			커뮤니티 게시글 작성 날짜
5	cmContent	varchar(2048)	N			커뮤니티 내용
6	cmAvailable	int	N			커뮤니티 작성 접근 번호

테이블명	member movie	Table 기술서	작성일		2021/12/06	
			작성자		임시은	
테이블 설명	membermovie					
No	Attribute	Data Type	Null	Ky	Default	Description
1	memberid	varchar(20)	N	PK		회원 아이디
2	password	varchar(20)	N			회원 비밀번호
3	name	varchar(20)	N			회원 이름
4	email	varchar(20)	N		‘N’	회원 이메일
5	tel	varchar(20)	N			회원 전화번호
6	memberRank	varchar(10)	N			회원 등급

테이블명	movie	Table 기술서	작성일		2021/12/06	
			작성자		임시은	
테이블 설명	movie (API 이용)					
No	Attribute	Data Type	Null	Ky	Default	Description
1	movieRank	int	N			영화 인기순위
2	movieName	varchar(30)	N			영화 제목
3	movieSalesShare	float	N			영화 예매율
4	movieCd	varchar(30)	N	PK		영화 대표코드
5	salesAmt	int	N			해당일의 매출액
6	audiCnt	int	N			해당일의 관객수
7	audiAcc	int	N			누적관객수
8	rankInten	int	N			전일대비 순위의 증감분
9	rankOldAndNew	varchar(5)	N			랭킹에 신규진입여부를 출력합니다. “OLD” : 기존 , “NEW” : 신규



테이블명	mttbl	Table 기술서	작성일		2021/12/06	
			작성자		임시은	
테이블 설명	mttbl					
No	Attribute	Data Type	Null	Ky	Default	Description
1	mtNum	int	N	PK		영화 구분 인덱스(총 360개)
2	mtDate	varchar(15)	N			데이터베이스 초기화 날짜
3	mtMovie	varchar(30)	N			영화명
4	mtCinema	varchar(30)	N			극장관
5	mtTimeline	varchar(15)	N			해당 영화의 시간
6	mtPeople	int	N			잔여 수
7	mtTotalPeople	int	N			좌석 수

테이블명	tickettbl	Table 기술서	작성일		2021/12/06	
			작성자		임시은	
테이블 설명	tickettbl					
No	Attribute	Data Type	Null	Ky	Default	Description
1	ticketNum	int	N	PK		예매 번호(티켓 번호)
2	ticketMovie	varchar(30)	N			영화명
3	ticketCinema	varchar(30)	N			극장관
4	ticketTimeline	varchar(15)	N			관람 시간
5	ticketDate	varchar(30)	N			영화 관람 날짜
6	ticketReservator	varchar(20)	N			예매
7	ticketPeople	int	N			예매인원수

## 2.4. 구현 화면

### 메인 페이지



NO24 영화

회원가입 로그인

메메 영화 이벤트 매거진 커뮤니티

현재 예매순 >



1) 유쾌이탈자  
예매율: 91.4



2) 엔렌트: 마법의 세계  
예매율: 26.7



3) 연애 빠진 로맨스  
예매율: 18.0



4) 이타널스  
예매율: 7.4



5) 장르만 로맨스  
예매율: 6.6

커뮤니티 인기 게시물

- 영화 거의 다봤는데 추천 좀 해주세용
- 내가 본 영화중에서 best5 추천하들 들어와봐 ㅋㅋ
- 오늘 인기 영화는 이거라미?
- 마들석 나온 영화
- 연인이랄 뽐 영화 추천하들

EVENT



반값 티켓 이벤트



출석부 이벤트

매거진

더보기 >



NO24 영화

초창작인 | 개인영화제작팀 | 이미지출판수업자료 | 영상정보처리기사문장 및 판권행 | 배정가온 | 차용인사 | 광고(광고문리) | 문화경영 | 기업정보  
서울특별시 구로구 장안로 446 | 고객센터 1234-1234  
인도네시아 3호 | 사업가등록번호 123-45-67890 | 개인정보 보호 최고 책임자 3호  
COPYRIGHT © DONOVAN ALL RIGHT RESERVED.

### 권한 별 로그인

회원가입 로그인

관리자 님이 로그인 하셨습니다. 로그인아웃



마이페이지

관리자 페이지

김동양 님이 로그인 하셨습니다. 로그인아웃



마이페이지

## 회원가입 및 로그인 화면

아이디  
  
비밀번호  
  
이름  
  
이메일  
  
전화번호

로그인

아이디  
  
비밀번호

## 로그인 코드

```

<%
String userid = (String)session.getAttribute("memid");
String memLogin = (String)session.getAttribute("memLogin");
String userName = AdminDAO.selectUserName(userid);
if(memLogin==null) {
%>
    <div id="login"><a href="login_form.jsp">로그인</a></div>
    <div id="joinId"><a href="memberRegister.jsp">회원가입</a></div>
%>
} else {
    if(userid.equals("admin")){
%>
        <div id="Login">
            <b>관리자</b> 님이 로그인 하셨습니다.
            <input type="button" value="로그아웃" onclick="location.href='../MovieSite/logoutOk.jsp'"><br>
            <div id="peopleBox">
                <a href="../MovieSite/mypage.jsp">
                     마이페이지
                </a>
                <a style="color:#C85448; margin-left: 15px" href="../MovieSite/admin/adminMember.jsp">관리자 페이지</a>
            </div>
        </div>
%>
    } else {
%>
        <div id="Login">
            <b><%=userName %></b> 님이 로그인 하셨습니다.
            <input type="button" value="로그아웃" onclick="location.href='../MovieSite/logoutOk.jsp'"><br>
            <div id="peopleBox">
                <a href="../MovieSite/mypage.jsp">
                     마이페이지
                </a>
            </div>
        </div>
%>
    }
}
%>

```

## 회원가입 쿼리

```
public int insertMember(RegisterBean mem) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    int result = 0;
    try {
        conn = DriverManager.getConnection(JDBC_URL, USER, PASS);

        pstmt = conn.prepareStatement("insert into memberMovie values (?, ?, ?, ?, ?, ?)");
        pstmt.setString(1, mem.getMemberid());
        pstmt.setString(2, mem.getPassword());
        pstmt.setString(3, mem.getName());
        pstmt.setString(4, mem.getEmail());
        pstmt.setString(5, mem.getTel());
        if(mem.getMemberid().equals("admin"))
            pstmt.setString(6, "관리자");
        else
            pstmt.setString(6, "GOLD");

        result = pstmt.executeUpdate();
    } catch (Exception ex) {
        System.out.println("Exception" + ex);
    } finally {
        if(rs!=null) try {rs.close();} catch(SQLException e) {}
        if(pstmt!=null) try {pstmt.close();} catch(SQLException e) {}
        if(conn!=null) try{conn.close();} catch(SQLException e) {}
    }
    return result;
}
```

## 로그인 쿼리

```
public int login(String userid, String userpw) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try {
        conn = DriverManager.getConnection(JDBC_URL, USER, PASS);
        String strQuery = "select password from membermovie where memberid = ?";
        pstmt = conn.prepareStatement(strQuery);
        pstmt.setString(1, userid);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            if(rs.getString(1).equals(userpw)) {
                return 1; // 로그인 성공
            }else
                return 0; // 비밀번호 불일치
        }
        return -1; // 아이디가 없음
    } catch (Exception ex) {
        System.out.println("Exception" + ex);
    }
    return -2; // DB 오류
}
```

## 영화진흥원 API 쿼리

```
// DB에 위에서 구한 일별 박스오피스를 저장
public void MovieList(String API) {
    JSONParser parser = new JSONParser();
    try {
        // 파서로 응답 받은 결과 body를 파싱하기
        JSONObject obj = (JSONObject)parser.parse(API);
        // 가장 최상위인 boxOfficeResult 가져오기
        JSONObject obj2 = (JSONObject)obj.get("boxOfficeResult");
        // 그 하위 항목중 dailyBoxOfficeList를 가져오는데 배열이기때문에 JSONArray로 가져옴
        JSONArray dailybox = (JSONArray)obj2.get("dailyBoxOfficeList");
        MovieImgDAO.deleteDB();
        // 배열을 돌면서
        for(int i=0;i<dailybox.size();i++){
            JSONObject tmp = (JSONObject)dailybox.get(i); //인덱스 번호로 접근해서 첫번째부터 가져오기

            int movieRank = Integer.valueOf((String) tmp.get("rank"));
            String movieName = (String)tmp.get("movieNm");
            float movieSalesShare = Float.valueOf((String) tmp.get("salesShare"));
            String movieCd = (String)tmp.get("movieCd");
            int salesAmt = Integer.valueOf((String) tmp.get("salesAmt"));
            int audiCnt = Integer.valueOf((String) tmp.get("audiCnt"));
            int audiAcc = Integer.valueOf((String) tmp.get("audiAcc"));
            int rankInten = Integer.valueOf((String) tmp.get("rankInten"));
            String rankOldAndNew = (String)tmp.get("rankOldAndNew");

            MovieImgDAO.insertDB(movieRank, movieName, movieSalesShare, movieCd, salesAmt, audiCnt, audiAcc, rankInten, rankOldAndNew);
        }
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

## 예매 페이지

영화 선택

예매 TOP10

1 유제아탈자  
2 엔칸토: 마법의 세계  
3 연애 빠진 로맨스  
4 이터널스  
5 장르만 로맨스  
6 돈  
7 프렌치 디스패치  
8 디어 에반 렘슨  
9 강릉  
10 귀멸의 칼날: 나타구오산 편  
TOP10 외 영화 (95)  
영화계 영화 (22)

극장 선택

전체 CGV 롯데시네마 메가박스 기타

MY 극장  
설정된 MY 극장이 없습니다.  
서울(80)  
부산(28)  
경기/인천(144)  
충청/대전(59)  
경북/대구(46)  
경남/울산(39)  
전라/광주(54)  
강원/제주(17)  
CGV 입구정  
메가박스 코엑스  
CGV 강남  
롯데시네마-브로드웨이(신사)  
메가박스 씨티(강남대로)  
이점씨어터  
롯데시네마-도곡  
롯데시네마-강동  
CGV 송파

관람일 선택

2021년 12월

< Today >

12	13	14	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1

시간 선택 > 60분전 예매 / 50분전 취소 가능

2D | [백신패스권] 2관 본관 3층

10:40  
[36/40]

13:00  
[38/40]

15:30  
[40/40]

19:20  
[40/40]

인원 선택 > 최대 4~8명까지 가능

\* 좌석은 선착순으로 배치됩니다.

성인 0 1 2 3 4 5 6 7 8  
청소년 0 1 2 3 4 5 6 7 8

유제아탈자  
CGV 입구정  
2021-12-6 / 10:40  
30000원

예매 정보

다음 단계



## 영화 데이터 삽입 쿼리

```
public int mtInsert(String movieName, int[] mtPeople) {
    int count = 0;
    int zero_count = 0;
    connect();
    String time[] = {"10:40", "13:00", "15:30", "19:20"};
    String place[] = {"CGV 압구정", "메가박스 코엑스", "CGV 강남", "롯데시네마-브로드웨이(신사)",
        "메가박스 씨티(강남대로)", "이름씨어터", "롯데시네마-도곡", "롯데시네마-강동", "CGV 송파"};
    String sql = "INSERT INTO mtTBL VALUES(?, ?, ?, ?, ?, ?, ?)";
    for(int i=0; i<360; i++) {
        if(mtPeople[i]==0) {
            zero_count++;
        }
    }
    if(zero_count == 360) {
        for(int i=0; i<360; i++) {
            mtPeople[i] = 40;
        }
    }
    try {
        String date = getDate().substring(0, 10);
        for(int i=0; i<9; i++) {
            for(int k=0; k<4; k++) {
                pstmt = conn.prepareStatement(sql);
                pstmt.setInt(1, getNext());
                pstmt.setString(2, date);
                pstmt.setString(3, movieName);
                pstmt.setString(4, place[i]);
                pstmt.setString(5, time[k]);
                pstmt.setInt(6, mtPeople[count]);
                pstmt.setInt(7, 40);
                pstmt.executeUpdate();
                count = count + 1;
            }
            count = count + 1;
        }
        return 1;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -1;
}
```

## 고객 예매 데이터 삽입 쿼리

```
public int ticketInsert(String movieName, String Cinema, String Timeline, String Date, String Reservator, String People) {
    connect();
    String sql = "INSERT INTO ticketTBL(ticketMovie, ticketCinema, ticketTimeline, ticketDate, ticketReservator, ticketPeople) VALUES(?, ?, ?, ?, ?, ?)";
    try {
        int tPeople = Integer.valueOf((String) People);

        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, movieName);
        pstmt.setString(2, Cinema);
        pstmt.setString(3, Timeline);
        pstmt.setString(4, Date);
        pstmt.setString(5, Reservator);
        pstmt.setInt(6, tPeople);
        pstmt.executeUpdate();

        return 1;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -1;
}
```

## 각각의 영화 잔여수 업데이트 쿼리

```
public int mtUpdate(String movieName, String Cinema, String Timeline, int People) {
    int mtNum = 0;
    int mtPeople = 0;
    connect();
    String select_mtNum_sql = "select mtNum, mtPeople from mtTBL where mtMovie = ? AND mtCinema = ? AND mtTimeline = ?";
    String sql = "UPDATE mtTBL SET mtPeople=? where mtNum=?";
    try {
        PreparedStatement select_mtNum_pstmt = conn.prepareStatement(select_mtNum_sql);
        select_mtNum_pstmt.setString(1, movieName);
        select_mtNum_pstmt.setString(2, Cinema);
        select_mtNum_pstmt.setString(3, Timeline);
        ResultSet select_mtNum_rs = select_mtNum_pstmt.executeQuery();
        if(select_mtNum_rs.next()) {
            mtNum = select_mtNum_rs.getInt("mtNum");
            mtPeople = select_mtNum_rs.getInt("mtPeople");
        }
        select_mtNum_rs.close();

        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, mtPeople-People);
        pstmt.setInt(2, mtNum);
        pstmt.executeUpdate();

        return 1;
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -1;
}
```

## 영화 페이지



인기상영작 TOP 5



일일 박스오피스 (2021-11-21 (일))

순위	영화	매출액	관객수	누적관객수	전일순위
1	유제이탈자	1,007,101,830원	97,202명	362,420명	-
2	연칸토: 마법의 세계	821,581,830원	83,417명	268,530명	-
3	연애 빠진 로맨스	575,709,290원	53,393명	209,064명	-
4	이터널스	237,713,320원	23,195명	2,979,894명	-
5	장로만 로맨스	177,004,170원	16,686명	468,585명	-
6	눈	166,583,550원	15,104명	1,422,021명	-
7	프렌치 디스패치	44,268,520원	3,986명	62,526명	-
8	디어 에반 헨슨	29,199,980원	2,668명	96,880명	-
9	강릉	18,503,140원	1,726명	302,068명	↑ 2
10	귀멸의 칼날: 나태구모산 편	16,884,040원	1,697명	39,691명	-

## TOP5 인기영화 출력 코드

```

<%>
MovieAPI TestMovie = new MovieAPI();
String API = TestMovie.RequestDayBoxOffice();
TestMovie.MovieList(API);
// 5위까지만 출력
ArrayList<MovieImgVO> selectMovieList = new ArrayList<MovieImgVO>();
selectMovieList = MovieImgDAO.selectMovieRank(5);
for(MovieImgVO movie : selectMovieList) {
    String imgsrc = "movieImgs/" + movie.getMovieCd() + ".jpg";
    String imgalt = "<" + movie.getMovieName() + "> 이미지";

    <div class="item">
        <ul>
            <li><a target="_blank" href="https://search.naver.com/search.naver?where=nexearch&sm=top_hyt&fbm=0&ie=utf8&query=영화 <%= movie.getMovieName() %>">
                <li class="name"><%= movie.getMovieRank() %> <%= movie.getMovieName() %></li>
                <li class="info">예매율: <%= movie.getmovieSalesShare() %></li></a>
            </li>
        </ul>
    </div>
}
    
```



## 일일 박스오피스 출력 코드

```
<section id="daily_box_office">
  <div>
    <div id="dbo_txt">일일 박스오피스 <p>(2021-11-28 (월))</p></div> <!-- MovieAPI의 targetDate 수정할 때마다 같이 수정해야 함 -->
    <section id="table">
      <table>
        <thead>
          <tr><th>순위</th><th>영화</th><th>매출액</th><th>관객수</th><th>누적관객수</th><th>전일순위</th></tr>
        </thead>
        <tbody>
          <%
            ArrayList<MovieImgVO> DailyBoxOffice = new ArrayList<MovieImgVO>();
            DailyBoxOffice = MovieImgDAO.DailyBoxOffice();
            for(MovieImgVO DBO : DailyBoxOffice) {
              String trID = "tr" + DBO.getMovieRank();
              String OldNew = MovieImgDAO.selectRankOldAndNew(DBO.getRankInten(), DBO.getRankOldAndNew());

              <tr id="<%=trID %%">
                <td><%= DBO.getMovieRank() %></td>
                <td><%= DBO.getMovieName() %></td>
                <td><fmt:formatNumber value="<%= DBO.getSalesAmt() %%" pattern="#,###,###,###"/>원</td>
                <td><fmt:formatNumber value="<%= DBO.getAudiCnt() %%" pattern="#,###"/>명</td>
                <td><fmt:formatNumber value="<%= DBO.getAudiAcc() %%" pattern="#,###,###"/>명</td>
                <td><%= OldNew %></td>
              </tr>
            }
          %>
        </tbody>
      </table>
    </section>
  </div>
</section>
```

## 이벤트 페이지



전체

이벤트

당첨자발표



## 매거진 페이지

●○○○



## 커뮤니티 페이지

개인정보 유출 방지

더보기 >

공지사항

회원들은 모두 부탁드립니다.

### 자유 게시판

글쓰기

더보기

번호	제목	작성자	등록일
1	영화 거의 다봤는데 추천 좀 해주세용	tlsl	2021-12-03
2	내가 본 영화중에서 best5 추천해줄 들어와봐 ㄱㄱ	1	2021-12-03
3	오늘 인기 영화는 이거라예?	abc	2021-12-03
4	마동석 나온 영화	admin	2021-12-03
5	연인이랑 볼 영화 추천해줄	admin	2021-12-03



## 커뮤니티 게시물 작성 페이지

커뮤니티 글쓰기

글 제목

글 내용

글쓰기

## 커뮤니티 게시물 작성 / 삭제 코드

```
//커뮤니티 글쓰기
else if(action.equals("writecommunity")){
    String cmID =(String)session.getAttribute("memid");
    cmDAO cd = new cmDAO();
    int result = cd.write(community.getCmTitle(), cmID, community.getCmContent());
    if(result==1){
        response.sendRedirect("write_list.jsp");
    }
}
//커뮤니티 글 삭제
}else if(action.equals("cmDelete")){
    String cmID =(String)session.getAttribute("memid");
    String cmRank = request.getParameter("cmRank");
    cmDAO cd = new cmDAO();
    int result = cd.delete(cmRank);
    if(result > 1) {
        response.sendRedirect("write_list.jsp");
    }
} else { %>

    <script>
        alert("삭제 오류입니다.");
    </script>
    response.sendRedirect("write_list.jsp");
}
```

## 커뮤니티 게시글 작성 및 삭제 쿼리

```
//커뮤니티 글쓰기
public int write(String cmTitle, String cmID, String cmContent) {
    connect();
    String SQL = "INSERT INTO COMMUNITY VALUE(?, ?, ?, ?, ?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, getNext()); //인덱스
        pstmt.setString(2, cmTitle); //글 제목
        pstmt.setString(3, cmID); // 작성자 아이디
        pstmt.setString(4, getDate()); // 글 작성 시간
        pstmt.setString(5, cmContent); // 글 내용
        pstmt.setInt(6, 1); // 글 작성 가능 여부
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -1; //데이터베이스 오류
}
```

```
//커뮤니티 글 삭제
public static int cmDelete(String cmRank) {
    connect();
    int result = 1;
    String sql = "DELETE from community where cmRank=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, cmRank);
        result = pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return result;
}
```

## 커뮤니티 게시글 목록 페이지

### 커뮤니티 글 목록

번호	제목	작성자	등록일
1	영화 거의 다봤는데 추천 좀 해주세용	tlsl	2021-12-03
2	내가 본 영화중에서 best5 추천해줄 들어와봐 ㄱㄱ	1	2021-12-03
3	요즘 인기 영화는 이거라예?	abc	2021-12-03
4	마동석 나온 영화	admin	2021-12-03
5	연인이랑 볼 영화 추천해줄	admin	2021-12-03
6	12월 상영예정작	lim	2021-12-06

[더보기](#)[글쓰기](#)

## 커뮤니티 게시글 목록 출력 쿼리

```
// 커뮤니티 글 출력
public static String[] cmWriteEdit(int cmRank) {
    connect();
    String sql = "select * from community where cmRank=?";
    String[] cm_post = new String[6];
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, cmRank);
        ResultSet rs = pstmt.executeQuery();
        while(rs.next()) {
            cm_post[0] = rs.getString("cmRank");
            cm_post[1] = rs.getString("cmTitle");
            cm_post[2] = rs.getString("cmID");
            cm_post[3] = rs.getString("cmDate");
            cm_post[4] = rs.getString("cmContent");
            cm_post[5] = rs.getString("cmAvailable");
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return cm_post;
}
```

## 관리자 페이지



NO24 영화

관리자님이 로그인 하셨습니다. [로그아웃](#)



[마이페이지](#)

[관리자 페이지](#)

[회원 목록](#) [공지사항 작성](#) [공지사항 관리](#) [커뮤니티 작성](#) [예매관리](#)

### 회원 목록

아이디	이름	비밀번호	이메일	전화번호	등급	수정	삭제
1	1	1	1@1	1	GOLD	수정	삭제
abc	에비씨	123	123@naver.com	123	GOLD	수정	삭제
admin	관리자	root	ttlsl13@naver.com	010-1234-5678	관리자	수정	삭제
dong	김동양	1234	im@na	010-1111-1111	GOLD	수정	삭제
ttlsl	사니	123	ttlsl@na	123	VIP	수정	삭제

### NO24 영화

호출작업 | 개인정보처리방침 | 이용약관 | 이용약관 | 영입정보처리방침 | 영입정보처리방침 | 이용약관 | 채용안내 | 광고입마출처 | 출판정보 | 기업정보  
서울특별시 강남구 테헤란로 443 | 고객센터 1234-1234  
02-1234-5678 | 사업자등록번호 123-45-67890 | 개인정보 보호 최고 책임자 3호  
COPYRIGHT© DOONGHANG ALL RIGHT RESERVED.

## 관리자 페이지 - 회원 목록 출력 코드

```
<%
// 우선 멤버 리스트를 받기 위한 객체 틀 생성
ArrayList<RegisterBean> MemberList = new ArrayList<RegisterBean>();

// DB 접근 하기위해 만들었던 AdminDAO 객체 생성
AdminDAO memberDao = new AdminDAO();

// DB 접근 객체에서 사용자들만 불러오는 함수 요청하여 미리 만들어 둔 멤버 리스트 담기
MemberList = memberDao.selectUserDB();

// MemberList안에서 RegisterBean 필드를 뺀 member를 ArrayList가 끝날때까지 반복시킨다
for(RegisterBean member : MemberList) {
%>
<tr>
<td><div><%=member.getMemberid() %></div></td>
<td><div><%=member.getName() %></div></td>
<td><div><%=member.getPassword() %></div></td>
<td><div><%=member.getEmail() %></div></td>
<td><div><%=member.getTel() %></div></td>
<td><div><%=member.getMemberRank() %></div>
<td class="td_a" onClick="location.href='<%=request.getContextPath() %>/../MovieSite/admin/edit_form.jsp?userid=<%=member.getMemberid() %>'>수정</td>
<td class="td_a" onClick="deleteDB('<%=member.getMemberid() %>')>삭제</td>
</tr>
%>
} %>
</table>
```

## 관리자 페이지 - 공지사항 게시글 작성 페이지

공지사항 글쓰기

글 제목

글 내용

글쓰기

## 관리자 페이지 - 공지사항 관리 페이지

### 글 목록

번호	제목	작성자	작성일	내용	쓰기 권한	수정	삭제
2	개인정보 유출 방지	admin	2021-12-06 13:54:12	저희 사이트는 회원가입 외 따로 개인정보를 받지 않습니다. 문자와 전화같이 개인적으로 연락을 따로 드리지 않으니 개인정보 유출에 유의하시기 바랍니다.	1	수정	삭제
1	회원들은 꼭 독 부탁드립니다.	admin	2021-11-30 16:11:26	안녕하세요. NO24 사이트 관리자입니다. 여러분들께 몇 가지의 말씀을 드리고자 공지사항을 남기게되었습니다. 1. 타인을 비판하는 글은 자제 부탁드립니다. 2. 악성댓글은 가차 없이 삭제되오니 한번 더 생각하고 글을 작성부탁드립니다.	1	수정	삭제

## 관리자 페이지 - 공지사항 관리 코드

```
<%
String action = request.getParameter("action");

if(action.equals("writeAdmin")){
    String adminID = (String)session.getAttribute("memid");
    cmDAO cd = new cmDAO();
    int result = cd.adminWrite(communityadmin.getAdminTitle(), adminID, communityadmin.getAdminContent());
    if(result == 1) {
        response.sendRedirect("notice_list_ex.jsp");
    }
} else if(action.equals("updatecommunity")){
    String adminRank = request.getParameter("adminRank");
    int Rank = Integer.parseInt(adminRank);
    String adminTitle = request.getParameter("adminTitle");
    String adminContent = request.getParameter("adminContent");
    cmDAO cd = new cmDAO();
    int result = cd.adminUpdate(Rank, adminTitle, adminContent);
    if (result > 0){
        <script>
            alert("수정되었습니다.");
            document.location.href="notice_list_ex.jsp";
        </script>
    }
} else {
    <script>
        alert("수정 오류입니다.");
        document.location.href="notice_list_ex.jsp";
    </script>
}
}
```



## 관리자 페이지 - 공지사항 관리 쿼리

```
// 관리자 공지사항 글쓰기
public int adminWrite(String adminTitle, String adminID, String adminContent) {
    connect();
    String SQL = "INSERT INTO AdminWrite VALUE(?, ?, ?, ?, ?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        pstmt.setInt(1, getNext2()); //인덱스
        pstmt.setString(2, adminTitle); //글 제목
        pstmt.setString(3, adminID); // 작성자 아이디
        pstmt.setString(4, getDate()); // 글 작성 시간
        pstmt.setString(5, adminContent); // 글 내용
        pstmt.setInt(6, 1); // 글 작성 가능 여부
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -1; //데이터베이스 오류
}

// 관리자 공지사항 삭제
public int adminDelete(int adminRank) {
    connect();
    int result = 1;
    String sql = "DELETE from AdminWrite where adminRank=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, adminRank);
        result = pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return result;
}

// 관리자 공지사항 수정
public int adminUpdate(int adminRank, String adminTitle, String adminContent) {
    connect();
    int result = 0;
    String sql = "UPDATE adminwrite SET adminTitle=?, adminDate=?, adminContent=? WHERE adminRank=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, adminTitle);
        pstmt.setString(2, getDate());
        pstmt.setString(3, adminContent);
        pstmt.setInt(4, adminRank);
        result = pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return result;
}
```



// 글 목록 가져오기

```
public ArrayList<cmAdBean> adminWriteList() {
    int num = 1;
    int rank;
    int count = 0;
    int[] reset_rank = new int[1000];
    // 위에 만들어놓은 connect()로 DB와 연결하기
    connect();
    // SQL 구문을 변수에 미리 집어 넣기
    String dropSQL = "alter table `adminwrite` drop primary key";
    String alterSQL = "ALTER TABLE `AdminWrite` MODIFY COLUMN `adminRank` Int(20) PRIMARY KEY";
    String SQL = "SELECT * FROM adminwrite";
    String SQL_DESC = "SELECT * FROM adminwrite ORDER BY adminRank DESC";
    // ArrayList<cmDAO>의 형태로 반환하기 위해서 list 객체 만들어주기
    ArrayList<cmAdBean> list = new ArrayList<cmAdBean>();
    try {
        // SQL 명령어를 준비시켜놓기
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        // SQL 명령어 실행(SELECT * FROM adminwrite)
        ResultSet rs = pstmt.executeQuery();

        PreparedStatement dropPSTMT = conn.prepareStatement(dropSQL);
        try {
            dropPSTMT.executeUpdate();
        } catch (Exception e) {
            PreparedStatement alterPSTMT = conn.prepareStatement(alterSQL);
            alterPSTMT.executeUpdate();
            alterPSTMT.close();
            dropPSTMT.executeUpdate();
        } finally {
            dropPSTMT.close();
        }

        // 가져온 데이터를 하나씩 순차적으로 검색
        while(rs.next()) {
            // 사용자 객체를 만들
            cmAdBean cAB = new cmAdBean();
            // 그 객체의 id를 rs(조회한 데이터)의 컬럼명 adminRank를 불러와서 설정
            rank = rs.getInt("adminRank");
            while(num < rank) {
                reset_rank[num-1] = num;
                num = num + 1;
            }
            if(num == rank) {
                reset_rank[num-1] = rank;
                num = num + 1;
            }
            updateAdminRank(reset_rank[count], rs.getInt("adminRank"));
            count = count + 1;
            // 멤버 리스트에 멤버 담기
        }
        PreparedStatement alterPSTMT = conn.prepareStatement(alterSQL);
        try {
            alterPSTMT.executeUpdate();
        } catch (Exception e) {
            dropPSTMT.executeUpdate();
            dropPSTMT.close();
            alterPSTMT.executeUpdate();
        } finally {
            alterPSTMT.close();
        }
        rs.close();

        PreparedStatement pstmt2 = conn.prepareStatement(SQL_DESC);
        ResultSet rs2 = pstmt2.executeQuery();
        while(rs2.next()) {
            cmAdBean cAB = new cmAdBean();
            cAB.setAdminRank(rs2.getInt("adminRank"));
            cAB.setAdminTitle(rs2.getString("adminTitle"));
            cAB.setAdminID(rs2.getString("adminID"));
            cAB.setAdminDate(rs2.getString("adminDate"));
            cAB.setAdminContent(rs2.getString("adminContent"));
            cAB.setAdminAvailable(rs2.getInt("adminAvailable"));
            // 멤버 리스트에 멤버 담기
            list.add(cAB);
        }
        rs2.close();
    }
}
```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
    finally {
        disconnect();
    }
    return list;
}

```

## 관리자 페이지 - 예매 목록 관리 페이지

### 예매 목록

\* 좌석은 선착순으로 배치됩니다.

예매번호	예매자	영화명	극장관	영화시간	관람인원수
1	admin	유체이탈자	CGV 압구정	10:40	4명
2	tsl	유체이탈자	CGV 압구정	13:00	2명
3	tsl	장르만 로맨스	메가박스 씨티(강남대로)	13:00	7명
4	admin	유체이탈자	메가박스 코엑스	10:40	3명
5	admin	연애 빠진 로맨스	메가박스 코엑스	10:40	2명

## 관리자 페이지 - 예매 목록 쿼리

```

public ArrayList<ticketTblVO> ticketList() {
    // 위에 만들어놓은 connect()로 DB와 연결하기
    connect();
    // SQL 구문을 변수에 미리 집어 넣기
    String SQL = "SELECT * FROM tickettbl";

    // ArrayList<cmDAO>의 형태로 반환하기 위해서 list 객체 만들어주기
    ArrayList<ticketTblVO> list = new ArrayList<ticketTblVO>();
    try {
        // SQL 명령어를 준비시켜주기
        PreparedStatement pstmt = conn.prepareStatement(SQL);
        ResultSet rs = pstmt.executeQuery();

        // 가져온 데이터를 하나씩 순차적으로 검색
        while(rs.next()) {
            // 사용자 객체를 만들
            ticketTblVO tT = new ticketTblVO();

            tT.setTicketNum(rs.getInt("ticketNum"));
            tT.setTicketMovie(rs.getString("ticketMovie"));
            tT.setTicketCinema(rs.getString("ticketCinema"));
            tT.setTicketTimeline(rs.getString("ticketTimeline"));
            tT.setTicketDate(rs.getString("ticketDate"));
            tT.setTicketReservator(rs.getString("ticketReservator"));
            tT.setTicketPeoPle(rs.getInt("ticketPeople"));
            // 멤버 리스트에 멤버 담기
            list.add(tT);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    finally {
        disconnect();
    }
    return list;
}

```

## 다. 결론 및 향후 발전 방안

영화 예매 사이트로 하나의 영화관 사이트가 아닌 여러 영화관의 영화 예매를 한 곳의 사이트에서 예매할 수 있다. 실시간으로 잔여 수를 표시해 주며, 최대 8명까지 예매 가능하다. 커뮤니티 페이지에서는 회원들만 게시글을 작성할 수 있으며, 공지사항 게시글은 관리자만 작성 가능하다. 커뮤니티의 원활한 활동을 위해 관리자는 회원들의 글을 삭제하고 관리할 수 있다. 현재까지 구현한 사이트에서 예매 취소, 리뷰 관리 등 더 많은 기능들을 추가하면 영화 예매 사이트 및 회원들의 다양한 커뮤니티 사이트로 발전 가능할 것으로 보인다.