

# Big Data Application (01)

## TEAM21 행운의 팀

김규리 (KIM KYURI, 1771113), 문지현 (Moon Jee Hyun, 1771093)

### #1. Team Information

**Team name:** 행운의 팀 (team21)

**Team Leader:** 김규리 (KIM KYURI)

**Members:** 김규리 (KIM KYURI, 1771113), 문지현 (Moon Jee Hyun, 1771093)

### #2. Project Overview & schedules

**Project theme:** Analysis of Movie Data

**Dataset:** <https://www.kaggle.com/tmdb/tmdb-movie-metadata>

#### 개요(description):

캐글의 TMDb 5000 Movie Dataset을 활용하여 영화와 관련된 정보를 조회할 수 있는 웹사이트를 구현한다. 사용 테이블은 총 6개이며 레코드 수의 총합은 224개이다. 사용자는 사이트에 로그인하면 영화 관련 기본 정보를 검색할 수 있으며, 사이트의 추가적인 기능은 다음과 같다.

1. 연도별 영화 수를 알 수 있다.
2. 배우를 검색하면 해당 배우가 나온 영화를 알 수 있다.
3. 장르를 검색하면 해당 장르의 영화를 알 수 있다.
4. 예산별로 영화의 순위를 표시하며,  $(\text{revenue}/\text{budget}) * 100$  지표 정보를 확인한다.
5. 인기도 순서 랭킹으로 영화 정보를 표시한다.

#### 일정 (schedules):

- 기획 (~10.13)
- 데이터셋 수집 및 전처리 (10.14 ~ 10.18)
- 데이터 분석 및 테이블 구현 (10.19 ~ 10.26)
- 웹페이지 프론트 작업 (10.27 ~ 11.03)
- 웹페이지 백엔드 작업(11.04 ~ 11.15)
- 최종 수정 및 완성 (~ 11.19)
- 데모 제작 및 제출 (~ 11.22)

팀원 별 기여도:

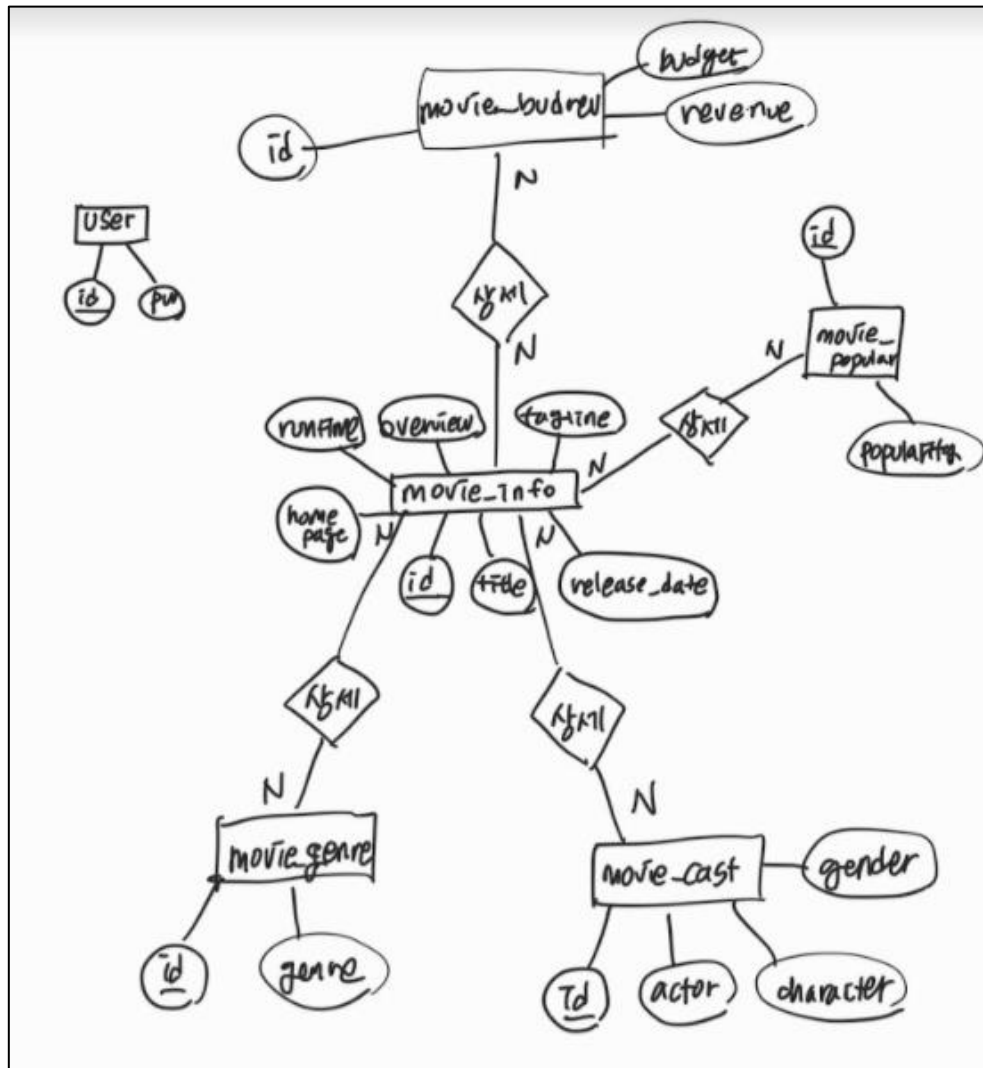
**김규리(KIN KYURI, 1771113): 50**

- 데이터 정제 및 테이블 생성: user , movie\_budrev , movie\_popular
- 회원가입 페이지 작업: signin.php , signin\_action.php
- 개인페이지 작업: myhome.php
- 회원탈퇴 php 코드 작업: signout\_action.php
- 연도 탭 작업: year.php
- 예산 탭 작업: budget.php
- 첫 페이지 작업: index.php
- 홈페이지 전체 검토 및 수정
- 계획서 및 최종보고서 작성

**문지현(Moon, Jee Hyun, 1771093): 50**

- 데이터 정제 및 테이블 생성: Movie\_info , Movie\_cast, Movie\_genre
- 로그인 페이지 작업: login.php, login\_action.php
- 회원수정 페이지 작업: update\_action.php
- 장르별 검색페이지 작업: search\_genre.php
- 배우별 검색페이지 작업: search\_actor.php
- 인기도별(평점) 페이지 작업: popular.php
- 홈페이지 전체 검토 및 수정
- 계획서 및 최종보고서 작성

### #3. ER 다이어그램

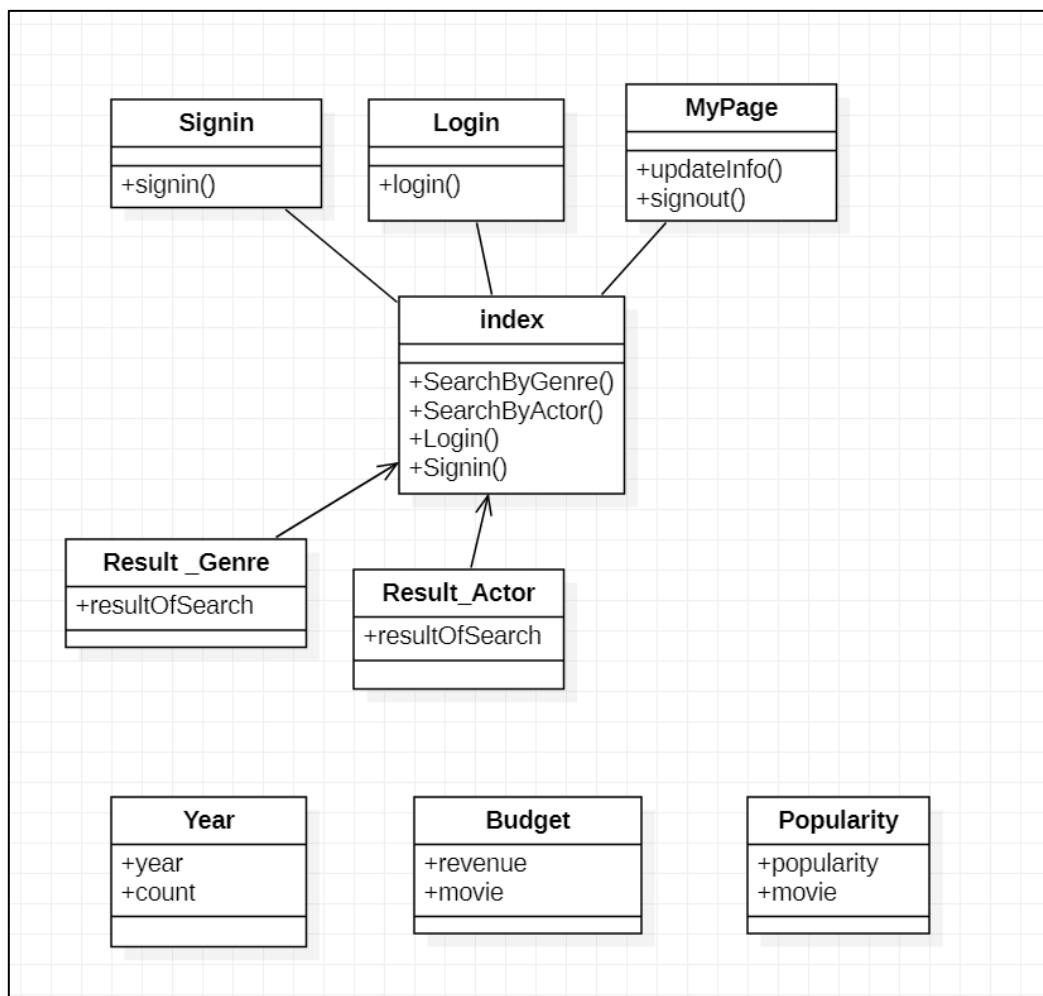


(데이터베이스 및 테이블 생성과 INSERT 작업 쿼리문은 제출파일의 createDB.sql 참고)

1. **User:** id 와 password를 입력하여 로그인할 수 있다. 유저와 영화 테이블에 관한 관계는 추후에 더욱 프로젝트를 발전시켜 연관시킬 수 있을 것이다.
2. **Movie\_info:** 영화의 제목, 내용, 상영 날짜 등 전반적인 데이터를 담고 있는 테이블이다. PK는 영화의 id이다. 또한 해당 PK는 연관된 다른 테이블과 외래 키 관계를 맺고 있다. 사용자가 검색을 통하여 해당 테이블에 존재하는 id를 통해 영화의 제목을 알아낸다. 세부적인 칼럼으로는 위의 ER다이어그램에서 확인할 수 있다.
3. **Movie\_genre:** 영화의 장르에 관한 데이터를 담고 있는 테이블이다. PK는 id에 해당하며 해당 PK를 통해 Movie\_info와 연결된다. 조인하여 movie\_info의 title을 출력하게 된다. 사용자는 장르 탭을 클릭하여 해당 테이블의 장르에 해당하는 랭킹과 title을 알 수 있다.

4. **Movie\_cast:** 영화의 캐스팅에 관한 데이터를 담고 있는 테이블이다. PK는 id에 해당하며 해당 PK를 통해 Movie\_info와 연결된다. 조인하여 movie\_info의 title을 출력하게 된다. 사용자는 검색하고 싶은 배우를 검색하여 배우를 찾을 수 있으며, 이에 따라 title을 알 수 있다.
5. **Movie\_popular:** 영화의 인기도에 관한 데이터를 담고 있는 테이블이다. PK는 id에 해당하며 해당 PK를 통해 Movie\_info와 연결된다. 조인하여 movie\_info의 title을 출력하게 된다. 사용자는 장르 탭을 클릭하여 해당 테이블의 장르에 해당하는 인기도별 랭킹과 title을 알 수 있다.
6. **Movie\_budrev:** 영화의 예산에 관한 데이터를 담고 있는 테이블이다. PK는 id에 해당하며 해당 PK를 통해 Movie\_info와 연결된다. 조인하여 movie\_info의 title을 출력하게 된다. 사용자는 budget 탭을 클릭하여 해당 테이블의 영화에 해당하는 랭킹과 예산, title을 알 수 있다.

#### #4. PHP 코드 다이어그램



## #4-1. 프로젝트 요구사항

(1) Theme of the project: **Analysis of Movie Data**

(2) Database creation scripts and data insert scripts should be created (SQL scripts): **createDB.sql**

(3) INSERT, DELETE, UPDATE, SELECT related functions should be implemented in the Web site.

Insert → **signin.php, signin\_action.php**

update, delete → **update\_action.php, signout\_action.php**

select → **result\_genre.php, result\_actor.php**

(4) Some SELECT queries should include SUM, AVG, etc and also GROUP BY statements: **Year.php**

(5) Provide at least 5 kinds of advanced analysis functions to users for the data:

1st: **search\_genre.php** / 2nd: **search\_actor.php** / 3rd: **year.php** / 4th: **budget.php** / 5th: **popularity.php**

(6) User input should include various types of controls such as text box, buttons, etc:

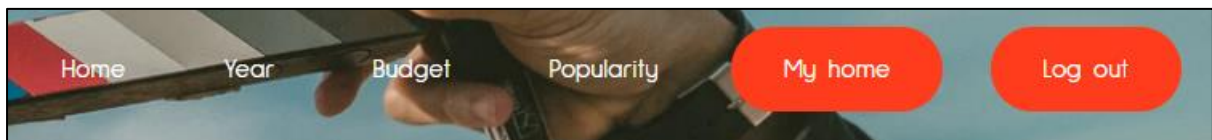
**signin.php** / **search\_genre.php** / **search\_actor.php**

(7) Use PHP sessions to store and retrieve information: **index.php**

(8) Use at least one transaction in the database for some part of the project: **signin.php**

## #4-2. PHP 다이어그램 및 페이지 기능 설명

### 1. Index.php (요구사항 7 만족)



인덱스를 포함한 모든 페이지의 상단에는 로그인 창으로 이동하는 버튼이 있다. 만일 사용자가 로그인을 한 상태라면 자신의 정보를 수정하는 Myhome 페이지 혹은 logout을 할 수 있는 버튼이 활성화된다. (요구사항 7 만족)

```
<?php
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21');

if(isset($_SESSION['userid'])) {
    ?>

    <li class = "add-list">
        <a href="myhome.php">My home</a>
    </li>

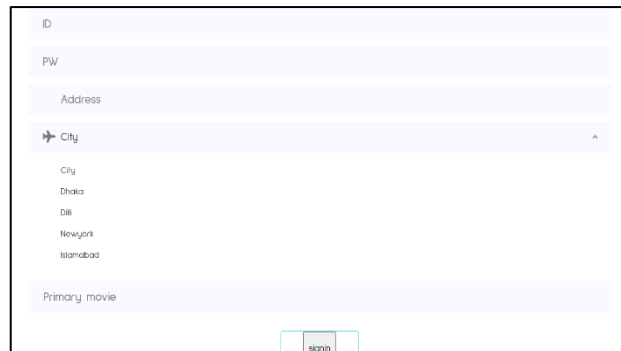
    <li class = "add-list">
        <a href="logout_action.php">Log out</a>
    </li>
}
else {
    ?>

    <li class = "add-list">
        <a href="login.php">Login</a>
    </li>

    <li class = "add-list">
        <a href="signin.php">Signin</a>
    </li>
}
?>
```

## 2. Signin.php, signin\_action.php (요구사항 3, 6, 8 만족)

사용자는 Signin 페이지에서 자신의 개인 정보를 입력한다. 이때, 텍스트 창뿐만 아니라 드롭다운 박스 형식의 정보도 입력한다. (요구사항 6 만족)



A web form for signing in. It contains the following fields: ID (text), PW (password), Address (text), City (dropdown menu with options: City, Dhaka, Dli, Newyork, Islamabad), and Primary movie (text). A 'signin' button is located at the bottom right.

사용자에게 입력 받은 insert를 통해 pw값과 id값을 저장한다. 이 때, php – mysql 트랜잭션을 사용하였다.

(요구사항 3, 8 만족)

```
$id=$_POST['id'];
$pw=$_POST['pw'];

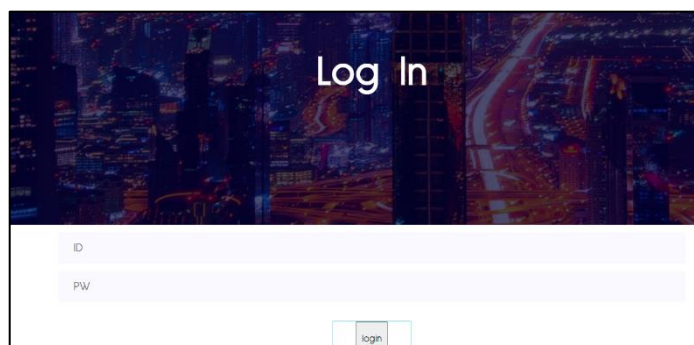
// 입력받은 데이터를 DB에 저장
$query = "insert into user (id, pw) values ('$id','$pw')";

$result = $connect->query($query);

// 저장이 됐다면 (result = true) 가입 완료
if($result) {
    ?>
    <script>
        alert('가입 되었습니다.');
```

## 3. Login.php

사용자는 해당페이지를 통해 로그인을 진행할 수 있다.



A login page with a background image of a city at night. The text 'Log In' is displayed in the top right. Below the text is a form with two fields: ID (text) and PW (password). A 'login' button is located at the bottom right.

#### 4. Mypage (update, delete 수행)

로그인한 사용자는 해당 페이지를 통해 자신의 정보를 수정하거나 탈퇴할 수 있다.



#### update\_action.php – UPDATE 기능

```
session_start();
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");
$id = $_SESSION['userid'];
$pw = $_POST['pw'];

$query = "update user set id='$id',pw='$pw'";
$result = $connect->query($query);
```

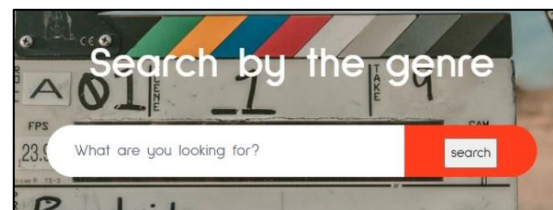
#### signinout\_action.php – DELETE 기능

```
session_start();
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");
$id = $_SESSION['userid'];

$query = "delete from user where id=$id";
$result = $connect->query($id);
```

#### 5. Result\_genre – search\_genre.php (1<sup>st</sup> advanced function, 요구사항 6 만족)

첫번째 고급분석 기능으로, 사용자가 장르를 검색하면 movie\_genre 테이블에 해당 영화들이 나온다. 또한, id값이 FK가 되어 movie\_info의 title이 나온다. (요구사항 6 만족)



```
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");
$find = $_POST['genre'];
$query = "select * , count(*) from movie_genre where name = '$find'";
$result = $connect->query($query);
$total = mysqli_num_rows($result);
```

index	title	genre
20	Avatar	Action

## 6. Result\_actor – search\_actor.php (2<sup>nd</sup> advanced function, 요구사항 6 만족)

두번째 고급분석 기능으로, 사용자가 배우의 이름을 검색하면 movie\_cast 테이블에 해당 배우의 이름이 나온다. 또한, id값이 FK가 되어 movie\_info의 title 이 나온다. (요구사항 6 만족)

### Search by Actor

```
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");
$find = $_POST['actor'];
$query = "select * from movie_cast where actor = '$find'";
$result = $connect->query($query);
$total = mysqli_num_rows($result);
$constTotal = $total;
```

Index	title	character	gender
1	Pirates of the Caribbean: At World's End	Captain Jack Sparrow	Man
2	Pirates of the Caribbean: Dead Man's Chest	Captain Jack Sparrow	Man
3	The Lone Ranger	Tonto	Man
4	Pirates of the Caribbean: On Stranger Tides	Captain Jack Sparrow	Man

## 7. Year.php (3<sup>rd</sup> advanced function, 요구사항 4 만족)

3번째 고급분석 기능으로, 사용자는 year 탭을 통해 연도별로 개봉된 영화를 확인할 수 있다. 여기서 Count와 연도별로 Group by 함수가 포함된 쿼리를 사용하였다.

(요구사항 4 만족)

```
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");
$query = "SELECT year(release_date) as yeardate, COUNT(release_date) AS cnt FROM movie_info GROUP BY Year(release_date) desc;";
$result = $connect->query($query);
$total = mysqli_num_rows($result);
```

index	year	count
11	2016	1
10	2015	2
9	2014	1
8	2013	2



## 8. Budget.php (4<sup>th</sup> advanced function)

4번째 고급분석 기능으로, 사용자는 영화들의 예산을 파악할 수 있다. 이때, budget대비 revenue를 통해 우리 팀이 자체적으로 계산한 수입 공식을 통해 income에 대한 내용도 확인할 수 있다. 수입 공식은 다음과 같다.  $income = (revenue/budget) * 100$

```
<tr class="even">
  <td width = "50" align = "center"><?php echo (21-$total)?></td>
  <td width = "500" align = "center">
    <?php echo $rows['movie_name']?></td>
    <td width = "200" align = "center"><?php echo $rows['budget']?></td>
    <td width = "200" align = "center"><?php echo $rows['revenue']?></td>
    <td width = "200" align = "center"><?php echo ($rows['revenue']/$rows['budget'])*100 ?></td>
  </tr>
<?php
  $total--;
}
?>
```

rank	title	budget	revenue	income
1	Avatar	237000000	2787965087	1176.3565767932
2	The Avengers	220000000	1519557910	690.70814090909
3	Avengers: Age of Ultron	280000000	1405403694	501.92989071429

## 9. Popularity.php (5<sup>th</sup> advanced function)

5번째 고급분석 기능으로, 인기도별 랭킹확인이 가능하다.

```
<?php
  while($rows = mysqli_fetch_assoc($result)){ //DB에 저장된 데이터 수 (열 기준)
?>
<tr class="even">
  <td width = "50" align = "center"><?php echo ($all-$total+1)?></td>
  <td width = "500" align = "center">
    <?php echo $rows['movie_name']?></td>
    <td width = "200" align = "center"><?php echo $rows['popularity']?></td>
  </tr>
<?php
  $total--;
}
?>
```

rank	title	popularity
1	Batman v Superman: Dawn of Justice	155.790452
2	Avatar	150.437577
3	Pirates of the Caribbean: Dead Man's Chest	145.847379
4	The Avengers	144.448633
5	Pirates of the Caribbean: At World's End	139.082615