

2023320010 임세빈 Term Project

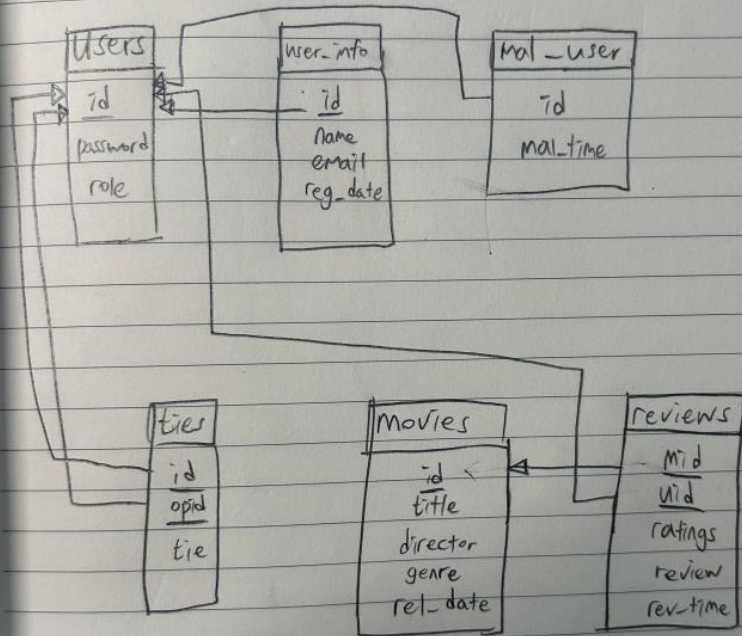
추가 기능은 영화 상세페이지에서 정렬 바꾸기, 유저 이메일 표시&수정, 로그아웃 구현했습니다.

<https://star-mole-d99.notion.site/1450162b84f64c9f9338c4f351058309?pvs=4>

코드 전체 캡처가 어려워 노션 링크 추가로 첨부합니다.

```
postgres=# create database termproject
postgres=# ;
CREATE DATABASE
postgres=# \c termproject;
접속 정보: 데이터베이스="termproject", 사용자="postgres".
termproject=# \i term.sql;
term.sql: No such file or directory
termproject=# \i C:\Desktop\DB\DB_termproject\term.sql;
C:: Permission denied
termproject=# \i C:/Desktop/DB/DB_termproject/term.sql;
psql:C:/Desktop/DB/DB_termproject/term.sql:1: 오류: "ties" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:2: 오류: "reviews" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:3: 오류: "movies" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:4: 오류: "mal_user" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:5: 오류: "user_info" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:6: 오류: "users" 테이블 없음
psql:C:/Desktop/DB/DB_termproject/term.sql:7: 오류: "domain_email" 자료형 없음
CREATE TABLE
CREATE DOMAIN
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
CREATE TABLE
INSERT 0 1
INSERT 0 1
INSERT 0 1
INSERT 0 1
```

Schema diagram



flask 앱 개발을 위해 필요한 모듈을 import하는 부분,

session사용을 위한 secret_key 설정

DB와 연결

```
//tapp.py

from flask import Flask, render_template, request, redirect, url_for, session
import psycopg2
from datetime import datetime

app = Flask(__name__)
app.secret_key = 'sebin'

connect = psycopg2.connect("dbname=termproject user=postgres password=sebin")
cur = connect.cursor()
```

기본 엔드포인트에 대한 라우팅을 수행. signin.html을 render한다.

```
@app.route('/')
def loginmain():
    return render_template("signin.html")
```

signin

users에 id, password가 있는지 확인한다.

users에 있으며 id, password가 공백이 아니면 메인페이지로 이동, 아니면 signin 실패페이지로 이동

```
@app.route('/signin', methods=['POST'])
def signin():
    id = request.form["id"]
    password = request.form["password"]



    cur.execute("SELECT * FROM users WHERE id = %s AND password = %s", (id, password))
    isRegisteredUser = cur.fetchone()

    if isRegisteredUser and len(id)>0 and len(password)>0:
        session['id'] = request.form['id']
        return redirect(url_for('mainpage'))
    else:
        return render_template("signin_fail.html", isRegisteredUser=isRegisteredUser)
```

로그아웃

세션에서 id를 제거하고 login페이지로 이동

Python ▾

 복사  캡션 ...

```
@app.route('/logout')
def logout():
    session.pop('user_id', None)
    return redirect(url_for('loginmain'))
```



메인페이지

정렬 순서를 바꾸기 위한 코드들

current_sort_by 값에 따라 다른 sql문을 실행한다.

sql문은 subquery와 join을 이용하였다.

Python ▾

 복사  캡션 ...

```
@app.route('/main', methods=['GET', 'POST'])
def mainpage():
    #유저 정보 세션
    user_id = session['id']

    current_sort_by1 = 'latest'
    current_sort_by2 = 'latest'

    if request.method == 'POST':
        sort_by1 = request.form.get('sort_by1')
        sort_by2 = request.form.get('sort_by2')

        # sort_by1 값 업데이트
        if sort_by1:
```

영화 상세정보

id를 보고 movies에서 영화 정보를 select해 전달

current_sort_by 에 따라 다른 sql문을 실행해 정렬 순서를 바꾼다.

```
@app.route('/movie_detail/<string:movie_id>', methods=['POST', 'GET'])
def movie_detail(movie_id):
    user_id = session['id']

    cur.execute("SELECT director, genre, rel_date, title, id FROM movies WHERE id = %s", (movie_id,))
    result1 = cur.fetchone()

    current_sort_by = 'latest'

    if request.method == 'POST':
        sort_by = request.form.get('sort_by')
        if sort_by == 'rating':
            current_sort_by = "rating"
        elif sort_by == 'user':
            current_sort_by = "user"
        elif sort_by == 'latest': # 'latest' 경우
            current_sort_by = "latest"

    if current_sort_by == "rating":
        cur.execute("SELECT ratings, uid, review, rev_time FROM reviews WHERE mid = %s ORDER BY ratings DESC")
    elif current_sort_by == "user":
        cur.execute("SELECT ratings, uid, review, rev_time FROM reviews WHERE mid = %s ORDER BY uid")
    elif current_sort_by == "latest":
        cur.execute("SELECT ratings, uid, review, rev_time FROM reviews WHERE mid = %s ORDER BY rev_time DESC")
    result2 = cur.fetchall()

    cur.execute("SELECT TRUNC(AVG(ratings), 1) FROM reviews WHERE mid = %s GROUP BY mid", (movie_id,))
```

영화 상세정보로 리다이렉트

영화 제목으로 검색하는 상황에 일치하는 id를 찾아 상세페이지로 리다이렉트한다.

```
@app.route('/movie_detail_bytitle_redirect/<string:movie_title>')
def movie_detail_bytitle_redirect(movie_title):

    cur.execute("SELECT id FROM movies WHERE title = %s", (movie_title,))
    result = cur.fetchone()
    movie_id = result[0]
    return redirect(url_for('movie_detail', movie_id=movie_id))
```

유저 상세정보

만약 현재 로그인중인 유저 이름을 클릭하면 mypage로 이동하기 위해

result1은 유저가 작성한 리뷰들을,

2,3,4는 유저를 팔로우하는, 유저가 팔로우하는, 유저가 뮤트한 사용자를,

5는 유저의 이메일을 가져온다.

다른 이름을 클릭하면 user_detail로 이동하기 위해

result1은 유저가 작성한 리뷰들을, 5는 유저의 이메일을 가져온다.

```
@app.route('/user_detail/<string:user_id>')
def user_detail(user_id):
    user_id_session = session['id']

    if user_id_session == user_id:
        cur.execute("select uid, ratings, title, review, rev_time FROM user_info, reviews, movies
            WHERE user_info.id = reviews.uid and reviews.mid = movies.id and reviews.uid = %s or
            result1 = cur.fetchall()

        cur.execute("select id from ties where opid = %s and tie = 'follow'", (user_id_session,))
        result2 = cur.fetchall()

        cur.execute("select opid from ties WHERE id = %s and tie = 'follow'", (user_id_session,))
        result3 = cur.fetchall()

        cur.execute("select opid from ties WHERE id = %s and tie = 'mute'", (user_id_session,))
        result4 = cur.fetchall()
```

unfollow, unmute, 둘이 반대 관계입니다. unmute기준으로 설명하겠습니다.

이미 mute관계인지 확인하고 그렇다면 mute관계를 삭제합니다.

그리고 user_detail로 리다이렉트합니다.

```
@app.route('/unfollow_user/<string:user_id>')
def unfollow_user(user_id):
    user_id_session = session['id']

    cur.execute("SELECT * FROM ties WHERE id = %s AND opid = %s AND tie = 'follow'", (user_id_session, user_id_session))
    existing_follow_relationship = cur.fetchone()

    if existing_follow_relationship:
        cur.execute("DELETE FROM ties WHERE id = %s AND opid = %s AND tie = 'follow'", (user_id_session, user_id_session))
        connect.commit()

    return redirect(url_for('user_detail', user_id = user_id_session))

@app.route('/unmute_user/<string:user_id>')
def unmute_user(user_id):
    user_id_session = session['id']

    cur.execute("SELECT * FROM ties WHERE id = %s AND opid = %s AND tie = 'mute'", (user_id_session, user_id_session))
    existing_follow_relationship = cur.fetchone()

    if existing_follow_relationship:
        cur.execute("DELETE FROM ties WHERE id = %s AND opid = %s AND tie = 'mute'", (user_id_session, user_id_session))
        connect.commit()

    return redirect(url_for('user_detail', user_id = user_id_session))
```


리뷰 작성

html에서 리뷰 정보들을 받아오고 review에 이미 사용자가 이 영화에 리뷰를 작성했는지 확인
이미 작성했다면 리뷰를 삭제

새 리뷰를 작성

```
@app.route('/submit_review', methods=['POST'])
def submit_review():
    if request.method == 'POST':
        mid = request.args.get('mid')
        uid = request.args.get('uid')
        ratings = int(request.form['rating'])
        review = request.form['review']
        rev_time = datetime.now()

        cur.execute("SELECT * FROM reviews WHERE mid = %s AND uid = %s", (mid, uid))
        existing_review = cur.fetchone()

        if existing_review:
            cur.execute("DELETE FROM reviews WHERE mid = %s AND uid = %s", (mid, uid))
            connect.commit()

        # 새 리뷰
        cur.execute("INSERT INTO reviews (mid, uid, ratings, review, rev_time) VALUES (%s, %s, %s, %s, %s)",
                    (mid, uid, ratings, review, rev_time))
        connect.commit()

        return redirect(url_for('movie_detail', movie_id=mid))
```

admin이 영화를 추가하는 부분

html에서 영화 정보를 받아오고 movies에 데이터를 삽입한다.

끝나면 다시 mypage_admin으로 리다이렉트

```
@app.route('/add_movie', methods=['GET', 'POST'])
def add_movie():
    user_id_session = session['id']

    if request.method == 'POST':
        cur.execute("SELECT COUNT(*) FROM movies")
        row_count = cur.fetchone()[0]

        id = row_count + 1
        title = request.form['title']
        director = request.form['director']
        genre = request.form['genre']
        release_date = request.form['release_date']

        # movies 테이블에 데이터 삽입
        cur.execute("INSERT INTO movies (id, title, director, genre, rel_date) VALUES (%s, %s, %s, %s, %s)"
                    (id, title, director, genre, release_date))
        connect.commit()

    cur.execute("select uid, ratings, title, review, rev_time FROM user_info, reviews, movies
        WHERE user_info.id = reviews.uid and reviews.mid = movies.id and reviews.uid = %s" % user_id_session)
    result1 = cur.fetchall()

    return render_template("mypage_admin.html", user_detail = result1, user_id = user_id_session)
```

이메일을 업데이트

user_info 의 email을 update한다.

```
@app.route('/update_email', methods=['POST'])
def update_email():
    user_id = session.get('id')

    new_email = request.form.get('email')

    cur.execute("UPDATE user_info SET email = %s WHERE id = %s", (new_email, user_id))
    connect.commit()

    return redirect(url_for('user_detail', user_id=user_id))
```

앱 실행

default port로 실행이 안 돼서 8000번 포트로 실행했다.

개발을 위해 debug옵션을 켜줬다.

```
if __name__ == '__main__':
    app.run(port=8000, debug=True)
```

//html

signin.html

처음으로 보이는 signin 페이지

id와 password를 입력하고 signin 또는 signup버튼을 누른다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>signin</title>
  </head>
  <body>
    <form action="/signin" method="post">
      <label for="user_id">ID:</label><br />
      <input id="user_id" type="text" name="id" value="" /><br />

      <label for="user_pwd">PASSWORD:</label><br />
      <input id="user_pwd" type="password" name="password" value="" /><br />
      <br />

      <button type="submit" name="send" value="sign in" formaction="/signin">
        sign in
      </button>
      <button type="submit" name="send" value="sign up" formaction="/signup">
        sign up
      </button>
    </form>
  </body>
</html>
```

signin_fail.html

signin 실패 페이지

signin실패 이유를 보여주는 텍스트가 추가된 signin 페이지이다.

HTML ▾

📄 복사 캡션 ...

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>signin_fail</title>
  </head>
  <body>
    <form action="/signinin" method="post">
      <label for="user_id">ID:</label><br />
      <input id="user_id" type="text" name="id" value="" /><br />

      <label for="user_pwd">PASSWORD:</label><br />
      <input id="user_pwd" type="password" name="password" value="" /><br />
      <br />

      <button type="submit" name="send" value="login" formaction="/signin">
        sign in
      </button>
      <button type="submit" name="send" value="sign up" formaction="/signup">
        sign up
      </button>
    </form>

    <p style="color: red">
      등록되지 않은 사용자이거나 id 또는 password를 입력하지 않았습니다.
    </p>
  </body>
```

signup_fail.html

signup 실패 페이지

signup 실패 이유를 보여주는 텍스트가 추가된 signin페이지이다

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>signup_fail</title>
  </head>
  <body>
    <form action="/signin" method="post">
      <label for="user_id">ID:</label><br />
      <input id="user_id" type="text" name="id" value="" /><br />

      <label for="user_pwd">PASSWORD:</label><br />
      <input id="user_pwd" type="password" name="password" value="" /><br />
      <br />

      <button type="submit" name="send" value="login" formaction="/signin">
        sign in
      </button>
      <button type="submit" name="send" value="sign up" formaction="/signup">
        sign up
      </button>
    </form>

    <p style="color: red">
      이미 존재하는 id이거나 id 또는 password를 입력하지 않았습니다.
    </p>
  </body>
```

main.html

메인 페이지

현재 로그인중인 ID와 logout, mainpage버튼은 모든 창에 다 있다.

영화 리스트를 보여준다. latest, genre, ratings 순서로 정렬을 바꿀 수 있다.

리뷰들을 보여준다. latest, title, followers 순서로 정렬을 바꿀 수 있다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>main</title>
  </head>

  <body>
    <div style="display: flex; align-items: center;">
      <p style="margin-right: 10px;">ID : </p>
      <form action="{{ url_for('user_detail', user_id=user_id) }}" method="get">
        <button type="submit">{{ user_id }}</button>
      </form>
    </div>

    <form action="{{ url_for('logout') }}" method="get">
      <button type="submit">logout</button>
    </form>

    <form action="{{ url_for('mainpage') }}" method="get">
      <button type="submit">mainpage</button>
    </form>
```

movie_detail.html

영화 상세정보 페이지

영화 제목, 감독, 장르, 개봉일, 리뷰들을 보여준다.

별점과 리뷰를 추가할 수 있다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>movie_detail{{movie_id}}</title>
  </head>
  <body>
    <div style="display: flex; align-items: center">
      <p style="margin-right: 10px">ID :</p>
      <form action="{{ url_for('user_detail', user_id=user_id) }}" method="get">
        <button type="submit">{{ user_id }}</button>
      </form>
    </div>
    <form action="{{ url_for('logout') }}" method="get">
      <button type="submit">logout</button>
    </form>
    <form action="{{ url_for('mainpage') }}" method="get">
      <button type="submit">mainpage</button>
    </form>

    <h1>{{movie_detail[3]}}</h1>
    <table border="1">
      <thead>
        <th>director</th>
        <th>genre</th>
        <th>release date</th>
```


user_detail.html

유저 상세정보 페이지

유저의 이름, 이메일주소를 보여준다.

follow와 mute를 할 수 있다.

유저가 작성한 리뷰들을 보여준다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>user_detail</title>
  </head>
  <body>
    <div style="display: flex; align-items: center;">
      <p style="margin-right: 10px;">ID : </p>
      <form action="{{ url_for('user_detail', user_id=user_id) }}" method="get">
        <button type="submit">{{ user_id }}</button>
      </form>
    </div>
    <form action="{{ url_for('logout') }}" method="get">
      <button type="submit">logout</button>
    </form>
    <form action="{{ url_for('mainpage') }}" method="get">
      <button type="submit">mainpage</button>
    </form>

    <h1>{{user_detail[0][0]}}</h1>
    <p>email: {{ email[0][0] }} </p>
```




mypage.html

유저이름'mypage, 이메일을 보여준다. 이메일을 수정할 수 있다.

게시한 리뷰들과

나를 팔로우하는, 내가 팔로우중인, 뮤트중인 유저들을 보여준다.

HTML ▾

 복사  캡션 

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>user_detail</title>
  </head>
  <body>
    <div style="display: flex; align-items: center;">
      <p style="margin-right: 10px;">ID : </p>
      <form action="{{ url_for('user_detail', user_id=user_id) }}" method="get">
        <button type="submit">{{ user_id }}</button>
      </form>
    </div>
    <form action="{{ url_for('logout') }}" method="get">
      <button type="submit">logout</button>
    </form>
    <form action="{{ url_for('mainpage') }}" method="get">
      <button type="submit">mainpage</button>
    </form>

    <h1>{{ user_id }}'s mypage</h1>

    <h3>[Reviews]</h3>
    <table border="1">
```

mypage_admin.html

admin만 접속 가능하다

작성한 리뷰를 보여준다.

제목, 감독, 장르, 개봉일을 선택해 영화를 추가할 수 있다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>user_detail</title>
  </head>
  <body>
    <div style="display: flex; align-items: center;">
      <p style="margin-right: 10px;">ID : </p>
      <form action="{{ url_for('user_detail', user_id=user_id) }}" method="get">
        <button type="submit">{{ user_id }}</button>
      </form>
    </div>
    <form action="{{ url_for('logout') }}" method="get">
      <button type="submit">logout</button>
    </form>
    <form action="{{ url_for('mainpage') }}" method="get">
      <button type="submit">mainpage</button>
    </form>

    <h1>{{ user_id }}'s mypage</h1>

    <h3>[Reviews]</h3>
    <table border="1">
```