

08 PJT

Axios 비동기 통신을 이용한 웹 사이트 구현

INDEX

- 개요
- 준비사항
- 프로젝트 안내
- 요구사항
- 도전 과제
- 제출

개요

프로젝트 개요

- AJAX와 JSON 데이터를 활용하는 커뮤니티 웹 서비스의 구성
- 장르 별 영화 데이터 조회가 가능한 애플리케이션 완성
- 영화 리뷰의 좋아요가 가능한 애플리케이션 완성
- 유저 간 팔로우가 가능한 애플리케이션 완성
- 알고리즘을 통한 영화 추천이 가능한 애플리케이션 완성

프로젝트 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 애플리케이션 제작
- AJAX와 JSON에 대한 이해
- Many to one relationship(N:1)에 대한 이해
- Many to many relationship(N:M)에 대한 이해
- 추천 알고리즘 설계

준비사항

| 개발도구

- Visual Studio Code
- Google Chrome
- Django 4.2.x

프로젝트 안내

프로젝트 및 앱

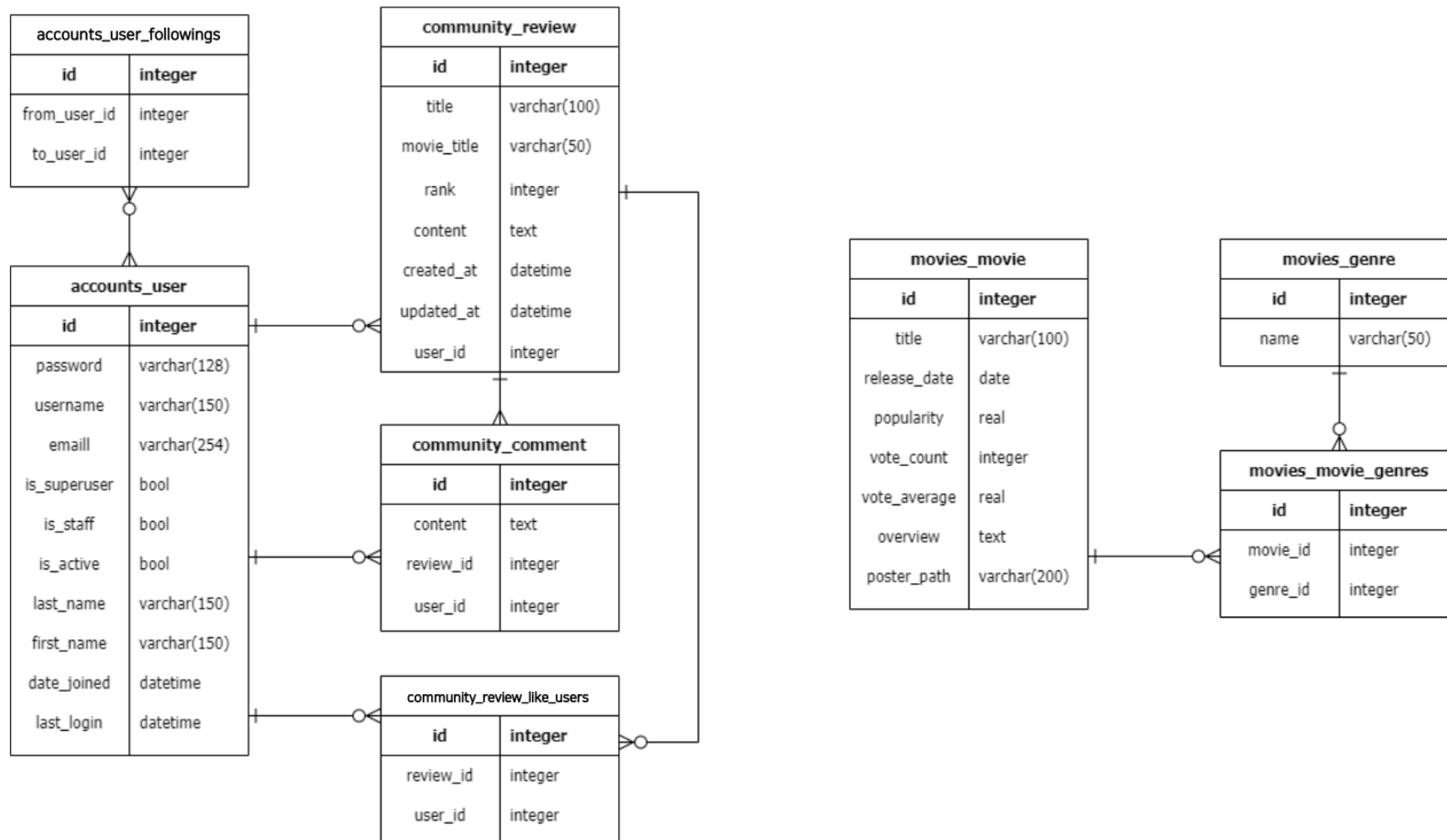
- 프로젝트 이름
 - mypjt
- 앱 이름
 - movies
 - community
 - accounts

초기 데이터

- fixture 파일은 제공된 스켈레톤 코드의 movies 앱 내에 위치
- fixture data의 load는 반드시 모델 정의 및 migrate 이후에 직접 진행

```
$ python manage.py migrate  
  
# json 파일은 movies/fixtures/movies/ 에 위치  
$ python manage.py loaddata movies/movies.json
```

모델 관계



URL

- accounts 앱의 URL

URL 패턴	역할
/accounts/signup/	<ul style="list-style-type: none">회원 생성 페이지 조회 & 단일 회원 데이터 생성 (회원가입)
/accounts/login/	<ul style="list-style-type: none">로그인 페이지 조회 & 세션 데이터 생성 및 저장 (로그인)
/accounts/logout/	<ul style="list-style-type: none">세션 데이터 삭제 (로그아웃)
/accounts/profile/<username>/	<ul style="list-style-type: none">사용자 상세 조회 페이지 (프로필 조회)
/accounts/<user_pk>/follow/	<ul style="list-style-type: none">사용자 팔로우 기능

URL

- community 앱의 URL

URL 패턴	역할
/community/	• 전체 리뷰 목록 페이지 조회
/community/create/	• 새로운 리뷰 생성 페이지 조회 & 단일 리뷰 데이터 저장
/community/<review_pk>/	• 단일 리뷰 상세 페이지 조회
/community/<review_pk>/comments/create/	• 단일 댓글 데이터 저장
/community/<review_pk>/like/	• 단일 리뷰 좋아요 기능

URL

- `movies` 앱의 URL

URL 패턴	역할
<code>/movies/</code>	<ul style="list-style-type: none">• 전체 영화 목록 페이지 조회
<code>/movies/filter-genre/</code>	<ul style="list-style-type: none">• 필터링 된 영화 데이터 제공
<code>/movies/recommended/</code>	<ul style="list-style-type: none">• 영화 추천 페이지 조회

view 함수

- accounts 앱의 view 함수

함수명	역할	허용 HTTP Method
signup	• 회원 생성 페이지 조회 & 단일 회원 데이터 생성 (회원가입)	GET & POST
login	• 로그인 페이지 조회 & 세션 데이터 생성 및 저장 (로그인)	GET & POST
logout	• 현재 사용자의 세션 데이터 삭제 (로그아웃)	GET & POST
profile	• 사용자 상세 조회 페이지 (프로필 조회)	GET

view 함수

- community 앱의 view 함수

함수명	역할	허용 HTTP Method
index	• 전체 리뷰 목록 페이지 조회	GET
create	• 새로운 리뷰 생성 페이지 조회 & 단일 리뷰 데이터 저장	GET & POST
detail	• 단일 리뷰 상세 페이지 조회	GET
create_comment	• 단일 댓글 데이터 저장	POST

view 함수

- movies 앱의 view 함수

함수명	역할	허용 HTTP Method
index	• 전체 리뷰 목록 페이지 조회	GET
filter_genre	• 필터링 된 영화 데이터 제공	GET
recommended	• 영화 추천 페이지 조회	GET

요구사항

공통 요구사항

- 명시된 요구사항 이외에는 자유롭게 작성해도 무관
- Bootstrap을 이용하여 자유롭게 스타일링 가능
- `.gitignore` 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않음

| 필수 요구사항

- A. 유저 팔로우 기능
- B. 리뷰 좋아요 기능
- C. 영화 장르 필터링

A. 유저 팔로우 기능

1. accounts 앱의 follow view 함수를 활용
2. 프로필 페이지(accounts/profile.html) 활용
3. 해당 사용자를 팔로우 할 수 있는 버튼, 팔로워 수와 팔로잉 수를 표시
4. 인증된 사용자만 다른 사용자를 팔로우 할 수 있으며,
사용자는 자기 자신을 팔로우 할 수 없음
5. 팔로우 버튼을 클릭하는 경우, AJAX 기술을 이용하여
새로고침 없이 프로필 페이지 화면을 구성 (팔로우 버튼 토글, 팔로워/팔로잉 수)

B. 리뷰 좋아요 기능

1. community 앱의 like view 함수를 활용
2. 전체 리뷰 목록 조회 페이지(`community/index.html`) 활용
3. 좋아요 버튼과 좋아요 개수를 표시
4. 인증된 사용자만 좋아요를 진행 할 수 있음
5. 좋아요 버튼을 클릭하는 경우, AJAX 기술을 이용하여
새로고침 없이 프로필 페이지 화면을 구성 (좋아요 버튼 토글, 좋아요 수)

C. 영화 장르 필터링

1. `movies` 앱의 `filter_genre` view 함수를 활용
2. 전체 영화 목록 조회 페이지(`movies/index.html`) 활용
3. 선택한 장르에 맞는 영화 데이터 목록 필터링 후 출력
 - 장르와 영화는 제공된 `fixtures` 활용
4. 페이지 첫 접속 시에는 전체 영화를 출력
5. 사용자의 인증 여부와 관계없음
6. 장르를 선택하는 경우, AJAX 기술을 이용하여 새로고침 없이 영화 목록 조회 페이지 화면을 재구성

완성 화면 예시

완성 화면 예시 (1/4)

- 유저 팔로우 (accounts/profile.html)

Hello, user01

[내 프로필](#)

Logout

[Movie](#) [Community](#) [New Review](#)

admin의 프로필 페이지

팔로잉 : 0 / 팔로워 : 1

Unfollow

완성 화면 예시 (2/4)

- 리뷰 좋아요 (community/index.html)

Hello, user01

[내 프로필](#)

[Logout](#)

[Movie](#) [Community](#) [New Review](#)

Community

작성자 : [user01](#)

글 번호: 1

글 제목: 리뷰 제목

글 내용: 리뷰 내용

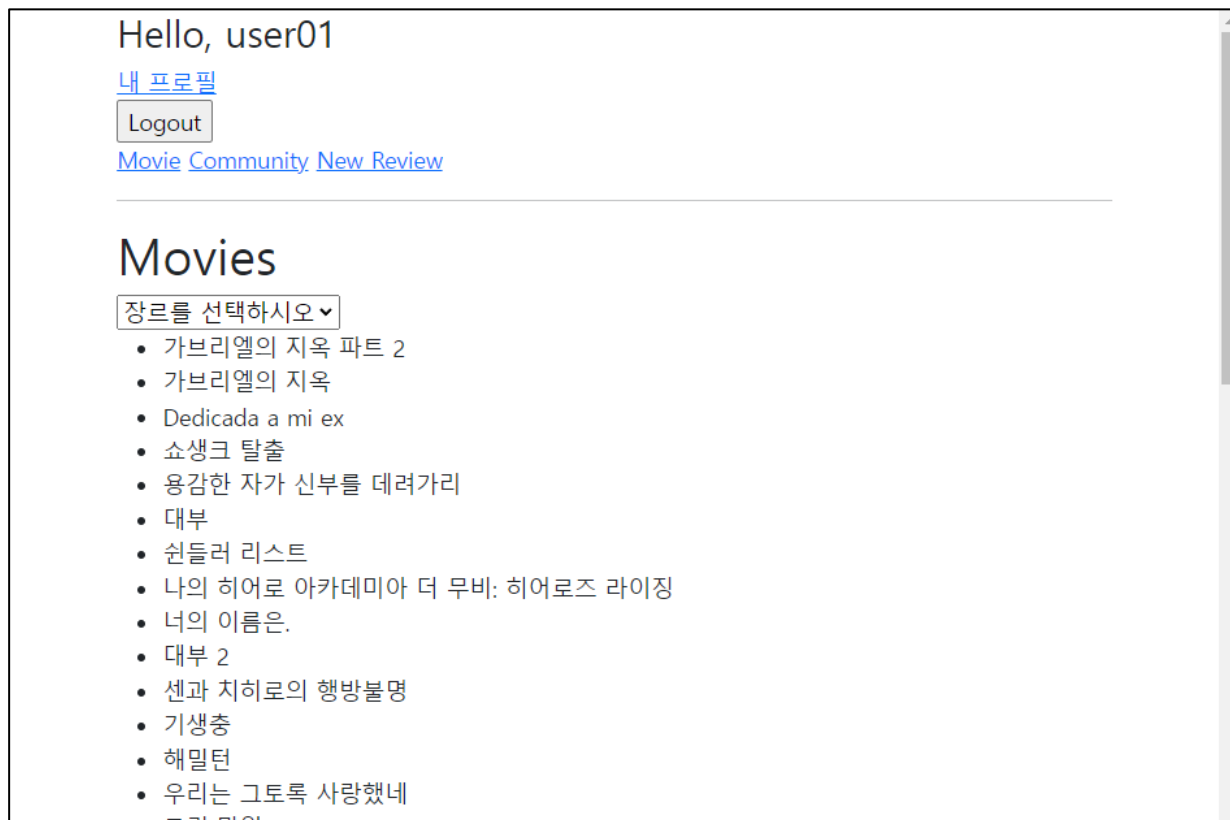
[좋아요 취소](#)

1 명이 이 글을 좋아합니다.

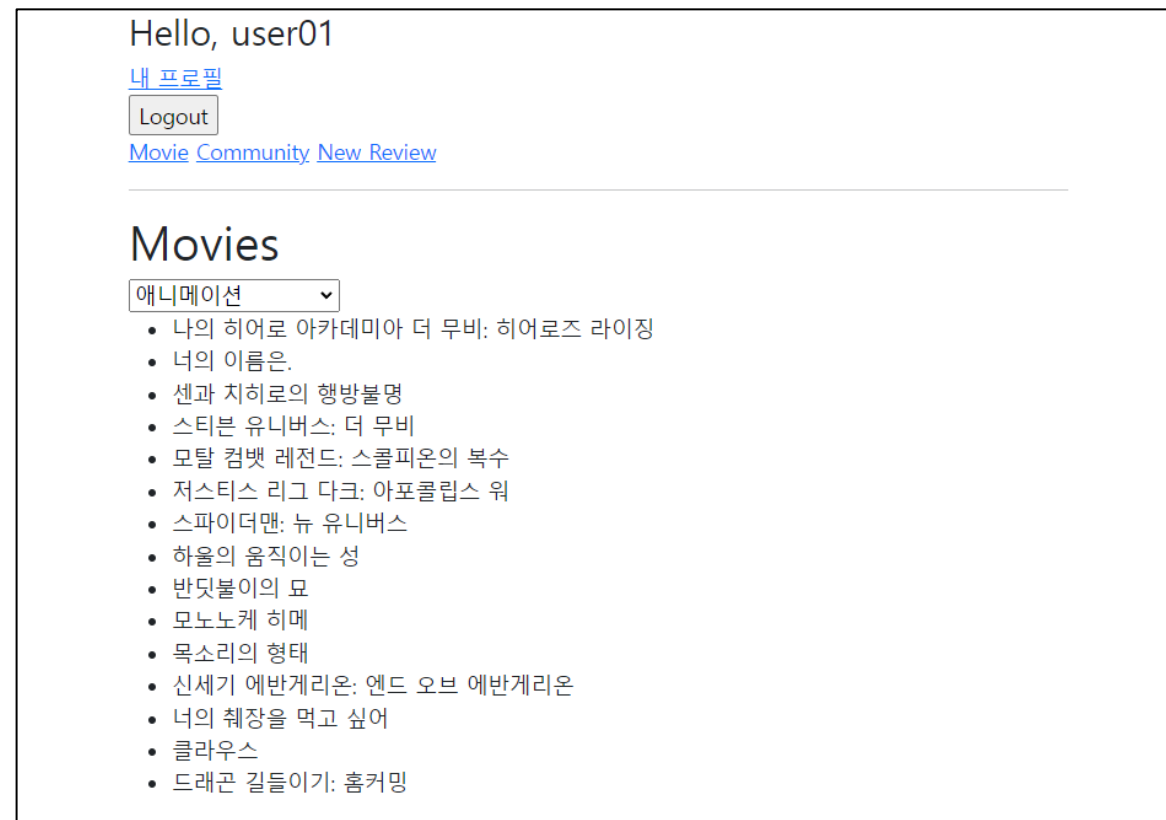
[\[detail\]](#)

완성 화면 예시 (3/4)

- 영화 장르 필터링 (movies/index.html)



장르 선택 전



장르 선택 후

완성 화면 예시 (4/4)

- 기타 페이지

Hello, user01

[내 프로필](#)

[Logout](#)

[Movie Community New Review](#)

CREATE

Title:

Movie title:

Rank:

Content:

[제출](#)

[\[back\]](#)

리뷰 작성 페이지

[Movie Community New Review](#)

DETAIL

1 번째 글

제목: 리뷰 제목

영화 제목: 영화 제목

내용: 리뷰 내용

평점: 5

작성 시각: 2023. 10. 10. 10:10

수정 시각: 2023. 10. 10. 10:10

댓글 목록

1개의 댓글이 있습니다.

user01 - 댓글 1

Content: [\[back\]](#) [제출](#)

리뷰 상세 페이지

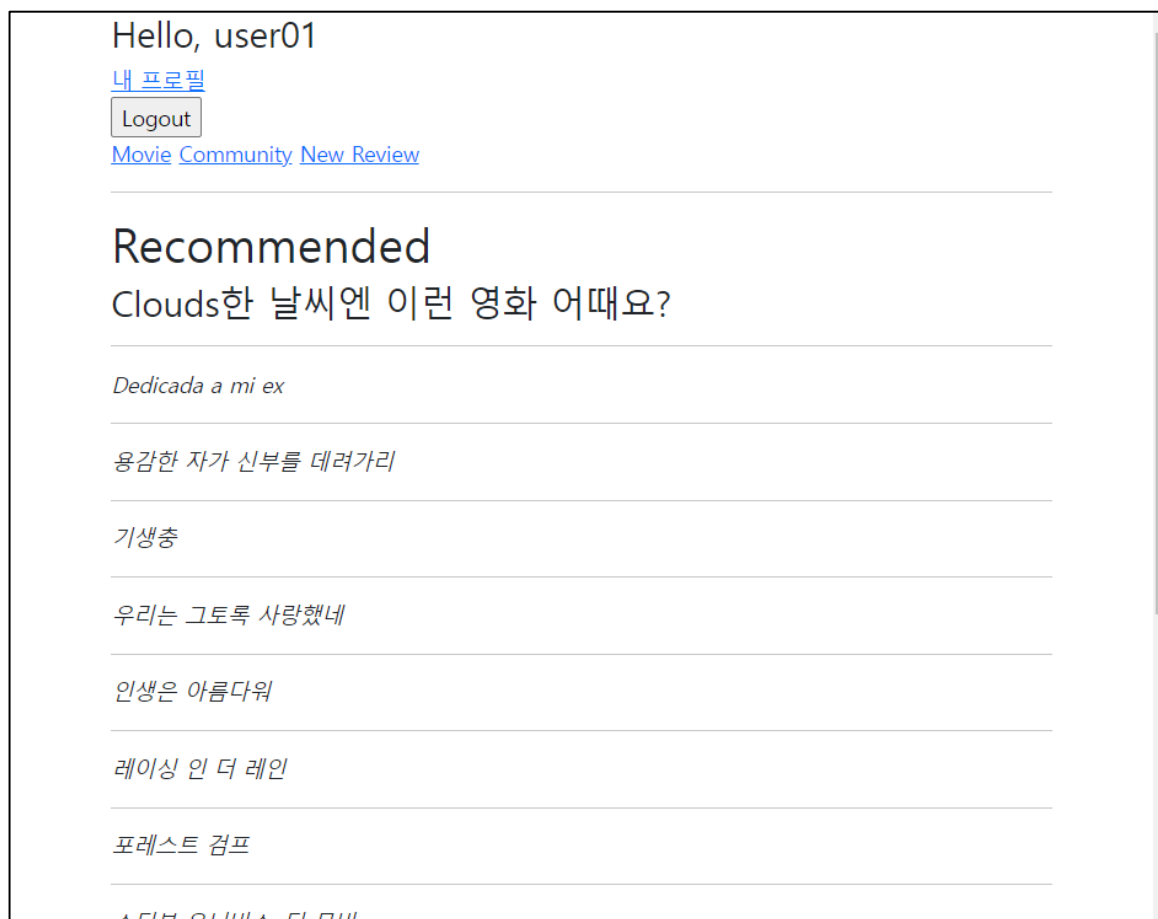
도전 과제

영화 추천 기능 구현 - 요구사항

- movies 앱의 recommended view 함수를 활용
- 추천 영화 페이지(movies/recommended.html) 활용
- 영화를 추천하는 알고리즘은 자유롭게 구상
 - 외부 API 활용 가능
- 구현한 알고리즘에 대한 설명은 README.md에 작성
 - 구현이 어렵다면 아이디어를 상세히 정리해서 작성

완성 화면 예시

- OpenWeatherMap API 활용 예시



제출

제출 시 주의사항

- 제출기한은 금일 18시까지 입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 **README.md** 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 ‘프로젝트 번호 + pjt’로 지정합니다. (ex. **01-pjt**)
- 반드시 각 반 담당 강사님을 Maintainer로 설정해야 합니다.