

알고리즘을 적용한 서버 구성

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web Application 제작
- AJAX 통신과 JSON 구조에 대한 이해
- Database 1:N, M:N 관계의 이해와 데이터 관계 설정
- 영화 추천 알고리즘 설계

2. 준비사항

A. 언어

- i. Python 3.9+
- ii. Django 3.2+

B. 도구

- i. Visual Studio Code
- ii. Chrome Browser

3. 요구사항

커뮤니티 서비스의 상세 기능 개발을 위한 단계로, 비동기 통신(AJAX)을 활용하여 사용자의 UI/UX를 개선합니다.

A. 프로젝트 구조

pjt09/은 startproject 명령어로 생성되는 project 디렉토리입니다.

community/는 startapp 명령어로 생성되는 application 디렉토리입니다.

아래의 폴더구조는 주요한 폴더와 파일만 명시되어 있습니다.

```
pjt09/  
  settings.py  
  urls.py
```

```
templates/  
  base.html
```

```
accounts/  
  migrations/  
  templates/  
  forms.py  
  models.py  
  urls.py  
  views.py  
  ...
```

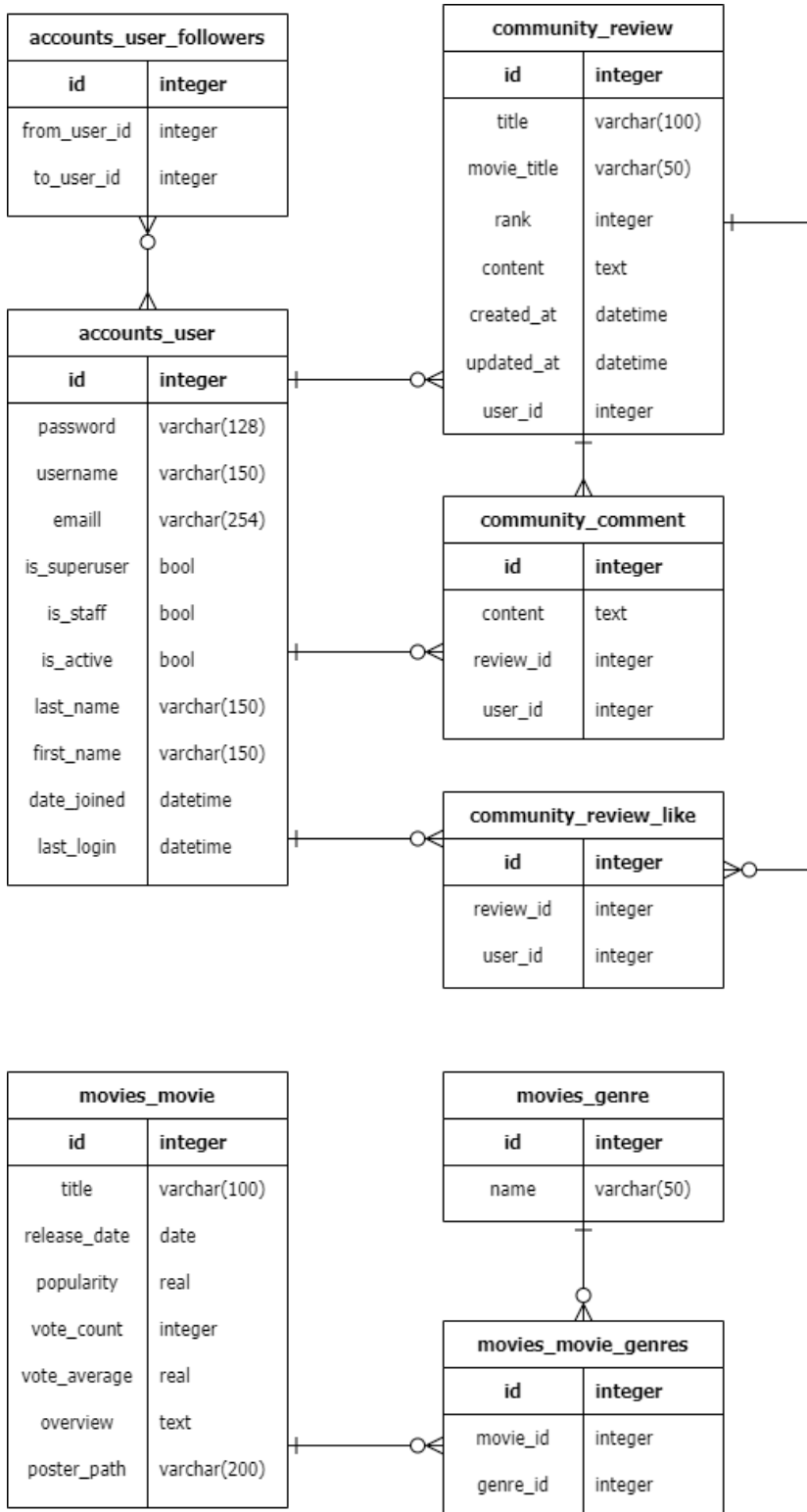
```
community/  
  migrations/  
  templates/  
  forms.py  
  models.py  
  urls.py  
  views.py  
  ...
```

```
movies/  
  fixtures/  
  migrations/  
  templates/  
  forms.py  
  models.py  
  urls.py  
  views.py  
  ...
```

```
manage.py  
...
```

B. Model

데이터베이스에서 모델의 ERD(Entity Relation Diagram)는 아래와 같습니다.



- 모델링 > migrate 이후 제공된 `movies/fixtures/movies.json`을 활용하여 데이터를 입력하고 시작합니다.

C. URL

URL은 app_name과 pattern name을 설정하여 적용합니다.

i. accounts app

accounts app의 모든 URL 패턴은 accounts/로 시작합니다.

HTTP verb	URL 패턴	설명
GET & POST	signup/	Form 표시 및 신규 사용자 생성(회원가입)
GET & POST	login/	Form 표시 및 기존 사용자 인증(로그인)
GET or POST	logout/	인증된 사용자 인증 해제(로그아웃)
GET	<username>/	사용자 상세 조회 페이지
POST	<user_pk>/follow/	팔로우 추가/팔로우 취소

ii. community app

community app의 모든 URL 패턴은 community/로 시작합니다.

HTTP verb	URL 패턴	설명
GET & POST	create/	Form 표시 및 신규 리뷰 생성
GET	/	전체 리뷰 목록 조회 페이지
GET	<review_pk>/	단일 리뷰 상세 조회 페이지
POST	<review_pk>/comments/create/	댓글 생성
POST	<review_pk>/like/	좋아요 기능

iii. movies app

movies app의 모든 URL 패턴은 movies/로 시작합니다.

HTTP verb	URL 패턴	설명
GET	/	전체 영화 목록 조회 페이지
GET	<movie_pk>/	단일 영화 상세 조회 페이지
GET	recommended/	추천 영화 조회 페이지

D. View & Template

i. 스켈레톤 코드 활용

1. 제공된 스켈레톤 코드를 활용하여 프로젝트를 구성합니다. 필요한 경우, 모델링과 UI등을 자유롭게 개선/수정할 수 있습니다.

ii. 유저 팔로우 기능

1. 페이지에 팔로워 수와 팔로잉 수를 표시합니다.
2. 프로필 페이지에는 해당 프로필 페이지의 사용자를 팔로우 할 수 있는 버튼을 표시합니다.
3. 인증(로그인)된 사용자만 다른 사용자를 팔로우 할 수 있으며, 사용자는 자기 자신을 팔로우 할 수 없습니다.
4. 팔로우 버튼을 클릭하는 경우, AJAX통신을 이용하여 서버에서 JSON데이터를 받아와 상황에 맞게 HTML화면을 구성합니다.

iii. 리뷰 좋아요 기능

1. 전체 리뷰 목록 조회 페이지에 **좋아요** 버튼과 좋아요 개수를 표시합니다.
2. 이미 좋아요 버튼을 누른 경우 취소 버튼을 표시합니다.
3. 인증(로그인)된 사용자만 리뷰에 좋아요 할 수 있습니다.
4. 좋아요 버튼을 클릭하는 경우, AJAX통신을 이용하여 서버에서 JSON데이터를 받아와 상황에 맞게 HTML화면을 구성합니다.

iv. movies app

1. 사용자의 인증 여부와 관계없이 전체 영화 목록 조회 페이지에서 적절한 UI를 활용하여 영화 목록을 제공합니다. (사용자에게 제공할 HTML은 index.html입니다.)
2. 사용자의 인증 여부와 관계없이 단일 영화 상세 조회 페이지에서 적절한 UI를 활용하여 영화 정보를 제공합니다. (사용자에게 제공할 HTML은 detail.html입니다.)

v. 영화 추천 기능

1. 사용자가 인증되어 있다면, 적절한 알고리즘을 활용하여 10개의 영화를 추천하여 제공합니다.
2. 영화를 추천하는 알고리즘은 자유롭게 구상합니다.
3. 구현한 알고리즘에 대해 README.md에 상세히 작성합니다.

vi. 추가적인 Styling

1. Bootstrap을 활용하여 자유롭게 스타일링 합니다.

4. 제출

- A. 제출기한은 금일 18:00까지입니다. 제출기한을 지켜 주시기 바랍니다.
- B. 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출해야 합니다.
- C. 위에 명시된 사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- D. <https://lab.ssafy.com>에 pjt09 프로젝트를 생성하고 업로드하여 제출합니다.
- E. 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.