

데이터베이스 설계

1. 목표

- 데이터를 생성, 조회, 수정, 삭제 할 수 있는 Web Application 제작
- Python Web Framework를 통한 데이터 조작
- Authentication에 대한 이해
- Database 1:N의 이해와 데이터 관계 설정

2. 준비사항

A. 언어

- i. Python 3.7+
- ii. Django 3.X

B. 도구

- i. vsCode
- ii. Chrome Browser

3. 요구사항

커뮤니티 서비스의 회원관련 기능 개발을 위한 단계로, 모델간의 관계 설정 후 데이터의 생성, 조회, 수정, 삭제 할 수 있는 기능을 완성합니다. 해당 기능은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다.

A. 프로젝트 구조

pjt05/은 startproject 명령어로 생성되는 project 디렉토리입니다.

community/는 startapp 명령어로 생성되는 application 디렉토리입니다.

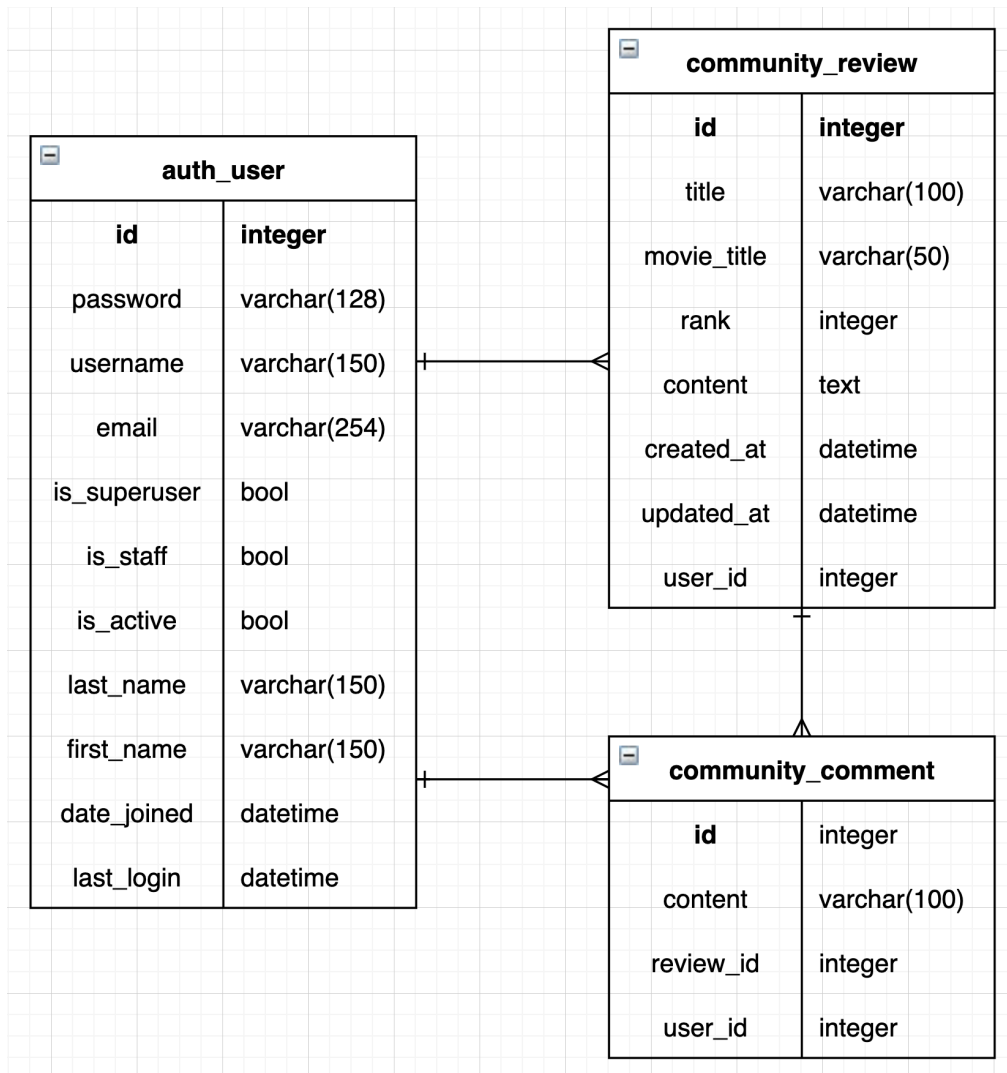
아래의 폴더구조는 주요한 폴더와 파일만 명시되어 있습니다.

```
pjt05/  
    settings.py  
    urls.py  
    accounts/
```

```
migrations/
templates/
    accounts/
        signup.html
        login.html
    forms.py
    models.py
    urls.py
    views.py
community/
    migrations/
    templates/
        community/
            form.html
            review_list.html
            review_detail.html
    forms.py
    models.py
    urls.py
    views.py
templates/
    base.html
manage.py
...
```

B. Model

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



직접 정의할 모델 클래스의 이름은 Review와 Comment이며, 다음과 같은 정보를 저장합니다. 사용자(User) 모델은 django에서 기본 제공하는 모델을 사용합니다.

C. Form

Review, Comment의 데이터 검증, 저장, 에러메세지, HTML을 모두 관리하기 위해 ModelForm을 사용합니다.

User의 데이터 검증, 저장, 에러메세지, HTML을 모두 관리하기 위해 django에서 제공하는 ModelForm, Form을 사용합니다.

D. Admin

Review, Comment는 관리자 페이지에서 데이터의 생성, 조회, 수정, 삭제 가능해야 합니다. (Admin 페이지에서 테스트 목적으로 데이터를 생성, 조회, 수정, 삭제할 수 있습니다.

E. URL

URL은 app_name과 name을 설정하여 적용합니다.

i. accounts app

accounts app의 모든 URL 패턴은 /accounts로 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 / POST	/logout/	인증된 사용자 인증 해제(로그아웃)

ii. community app

community app의 모든 URL 패턴은 /community로 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/	댓글 생성

F. View & Template

i. 공유 템플릿 생성 및 사용

- 모든 HTML파일은 base.html을 확장(extends)하여 사용합니다.
- base.html은 모든 페이지가 공유하는 상단 네비게이션 바를 표시합니다.
- 네비게이션 바
 - 전체 리뷰 목록 조회 페이지와 새로운 리뷰 작성 페이지로 이동할 수 있는 링크를 포함합니다.
 - 사용자가 인증되어 있지 않다면, 로그인 페이지와 회원 가입 페이지

로 이동 할 수 있는 링크를 포함합니다.

C. 사용자가 인증되어 있다면, 로그아웃 할 수 있는 링크를 표시합니다.

ii. 신규 사용자 생성(회원가입)

1. 이미 인증되어 있는 사용자는 전체 리뷰 목록 페이지로 redirect합니다.

2. django에서 제공하는 UserCreationForm을 사용합니다.

3. HTTP method GET

A. 사용자에게 응답으로 제공할 HTML은 accounts/signup.html입니다.

B. 회원가입을 할 수 있는 form을 표시하며, 다음과 같은 input요소들을 포함해야 합니다.

필드명	Tag	Type
username	input	text
password	input	password
password_confirmation	input	password

C. form에 작성한 정보는 제출(submit)시, POST 방식으로 요청과 데이터를 적절한 URL로 전송합니다.

4. HTTP method POST

A. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장합니다. 그 후 사용자를 인증(로그인)하고 전체 리뷰 목록 페이지로 redirect합니다.

B. 데이터가 유효하지 않다면 에러 메시지를 포함하여, 데이터를 작성하는 form을 사용자 화면에 표시합니다.

iii. 기존 사용자 인증(로그인)

1. 이미 인증되어있는 사용자는 전체 리뷰 목록 페이지로 redirect합니다.

2. django에서 제공하는 AuthenticationForm을 사용합니다.

3. HTTP method GET

A. 사용자에게 응답으로 제공할 HTML은 accounts/login.html입니다.

B. 로그인 할 수 있는 form을 표시하며, 다음과 같은 input요소들을 포함해야 합니다.

C. form에 작성한 정보는 제출(submit)시, POST방식으로 요청과 데이터를 적절한 URL로 전송합니다.

4. HTTP method POST

A. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장합니다. 그 후 사용자를 인증(로그인)합니다.

i. 로그인 하기 전, 기존 URL이 함께 넘어왔다면 해당 URL로 redirect합니다.

ii. 기존 URL이 넘어오지 않았다면 전체 리뷰 목록 페이지로 redirect합니다.

iv. 인증된 사용자 인증해제(로그아웃)

1. HTTP method는 GET과 POST 중 선택하여 구성합니다.

2. 사용자의 인증을 해제(로그아웃)하고 전체 리뷰 목록 페이지로 redirect합니다.

v. 새로운 리뷰 작성

1. 이미 인증되어있는 사용자는 전체 리뷰 목록 페이지로 redirect합니다.

2. HTTP method GET

A. 사용자에게 응답으로 제공할 HTML파일은 form.html입니다.

B. 리뷰를 작성할 수 있는 Form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드명	Tag	Type
title	input	text
movie_title	input	text
rank	input	number
content	textarea	없음

C. Form에 작성한 정보는 제출(submit)시, POST 방식으로 요청과 데이터를 적절한 URL로 전송합니다.

3. HTTP method POST

- A. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장합니다. 그 후 생성된 상세 리뷰 조회 페이지로 redirect합니다.
- B. 데이터가 유효하지 않다면 에러 메시지를 포함하여 데이터를 작성하는 Form을 표시합니다.

vi. 전체 리뷰 목록 조회

- 1. 사용자에게 응답으로 제공할 HTML은 review_list.html입니다.
- 2. review_list.html은 적절한 HTML요소를 사용하여 title을 포함한 게시글의 데이터를 자유롭게 표시하며, title을 클릭 시 해당 리뷰의 상세 조회 페이지로 이동합니다.

vii. 단일 리뷰 상세 조회

- 1. URL을 통해 함께 전달된 review_pk를 id(pk)로 갖는 리뷰의 상세정보를 HTML에 표시해야 합니다.
- 2. 사용자에게 응답으로 제공할 HTML은 review_detail.html입니다.
- 3. review_pk를 id(pk)로 갖는 리뷰 데이터가 없을 경우, 404 에러 페이지를 표시합니다.
- 4. review_detail.html에는 적절한 HTML 요소를 사용하여, 조회하는 리뷰의 title, movie_title, content, rank, created_at, updated_at을 표시합니다.

viii. 댓글 생성

- 1. 단일 리뷰 상세 조회 페이지에서 댓글을 작성할 수 있는 Form을 표시합니다.
- 2. Form에 작성한 정보는 제출(submit)시, POST 방식으로 요청과 데이터를 적절한 URL로 전송합니다.
- 3. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장합니다. 그 후 댓글을 작성하던 단일 리뷰 상세 페이지로 redirect합니다.

ix. 추가적인 Styling

- 1. Bootstrap을 활용하여 자유롭게 스타일링 합니다.

4. 결과

- A. 제출 기한은 **20.10.07(수) 18:00** 까지 입니다. 제출기한을 엄수하여 주시기 바랍니다.
- B. 반드시 README.md 파일에 각 단계별 상세 구현 과정, 학습 내용 및 어려웠던 부분을 기록하여 제출해야 합니다.
- C. 위에 명시된 사항은 최소 조건이며 추가적인 정보를 수정 할 수 있습니다.
- D. <https://lab.ssafy.com>에 pjt05 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
- E. 각 반 담당 교수님을 Maintainer로 설정해야 합니다.