

사용자 인증기반 웹 서비스 구현

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application 제작
- Django web framework를 통한 데이터 조작
- Authentication(사용자 인증)에 대한 이해

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 회원관리 기능 개발을 위한 단계로, 사용자 인증상태에 따라 데이터의 생성, 조회, 수정, 삭제할 수 있는 기능을 완성합니다. 해당 기능은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다.

A. 프로젝트 구조

pjt06/은 startproject 명령어로 생성되는 project 디렉토리입니다.

community/는 startapp 명령어로 생성되는 application 디렉토리입니다.

아래의 폴더구조는 주요한 폴더와 파일만 명시되어 있습니다.

```
pjt06/  
  settings.py  
  urls.py  
  ...  
  
templates/  
  base.html
```

```

accounts/
  migrations/
  templates/
    accounts/
      *.html
  forms.py
  models.py
  urls.py
  views.py
  ...

community/
  migrations/
  templates/
    community/
      *.html
  forms.py
  models.py
  urls.py
  views.py
  ...

manage.py
...

```

B. Model

정의할 모델 클래스의 이름은 Review이며, 다음과 같은 정보를 저장합니다.

사용자(User) 모델은 django에서 기본 제공하는 모델을 사용합니다.

제시된 정보 외 모든 추가 정보는 선택 사항입니다.

필드명	자료형	설명
movie_title	String(<= 100)	리뷰한 영화의 제목
title	String(<= 100)	리뷰의 제목
content	Text	내용
rank	Integer	리뷰한 영화의 평점

C. Form

Review 모델의 데이터 검증, 저장, 에러메시지, HTML을 모두 관리하기 위해 ModelForm을 사용합니다.

User 모델의 데이터 검증, 저장, 에러메시지, HTML, 인증을 모두 관리하기 위해 django에서 제공하는 ModelForm, Form을 사용합니다.

D. Admin

관리자 페이지에서 Review, User 모델의 데이터 생성, 조회, 수정, 삭제가 가능합니다.

E. URL

URL은 app_name과 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)

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
GET & POST	/<<review_pk>/update/	Form 표시 및 기존 리뷰 수정
GET or POST	/<<review_pk>/delete/	단일 리뷰 삭제

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
 - 사용자의 인증 여부와 관계없이 전체 리뷰 목록 조회로 이동할 수 있는 링크를 표시합니다.
 - 사용자가 인증되어 있지 않다면, 로그인 페이지와 회원가입 페이지로

이동할 수 있는 링크를 표시합니다.

- C. **사용자가 인증되어 있다면**, 새로운 리뷰 작성페이지로 이동할 수 있는 링크와 로그아웃 할 수 있는 링크를 표시합니다.

ii. 신규 사용자 생성(회원가입)

1. 이미 인증되어 있는 사용자가 해당 페이지로 요청을 보낸 경우, 전체 리뷰 목록 페이지로 redirect합니다.
2. django에서 제공하는 UserCreationForm을 사용합니다.
3. HTTP method GET
 - A. 사용자에게 응답으로 제공할 HTML파일은 signup.html입니다.
 - B. form에 작성한 정보는 POST 방식으로 제출(submit)합니다.
 - C. 회원가입을 할 수 있는 form을 표시하며, 다음과 같은 input요소들을 포함해야 합니다.

필드 name	HTML 요소	Type
username	input	text
password	input	password
password_confirmation	input	password

4. HTTP method POST
 - A. 데이터가 유효하다면 전송된 데이터를 데이터베이스에 저장합니다.
 - B. 그 후 사용자를 인증(로그인)하고 전체 리뷰 목록 페이지로 redirect합니다.
 - C. 데이터가 유효하지 않다면 데이터를 작성하는 form을 에러메시지와 함께 사용자 화면에 표시합니다.

iii. 기존 사용자 인증(로그인)

1. 인증되어 있는 사용자는 전체 리뷰 목록 페이지로 redirect합니다.
2. django에서 제공하는 AuthenticationForm을 사용합니다.
3. HTTP method GET
 - A. 사용자에게 응답으로 제공할 HTML파일은 login.html입니다.

- B. form에 작성한 정보는 POST 방식으로 제출(submit)합니다.
- C. 로그인 할 수 있는 form을 표시하며, 다음과 같은 input요소들을 포함해야 합니다.

필드 name	HTML 요소	Type
username	input	text
password	input	password

4. HTTP method POST

- A. 데이터가 유효하다면 사용자를 인증(로그인)합니다.
- i. 로그인 하기 전의 페이지 URL이 함께 전송된 경우, 로그인 이후 해당 URL로 redirect합니다.
 - ii. 이전 페이지의 URL이 함께 전송되지 않았다면, 전체 리뷰 목록 조회 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. 인증되어 있는 사용자만 새로운 리뷰를 작성할 수 있습니다.
2. HTTP method GET
 - A. 사용자에게 응답으로 제공할 HTML파일은 form.html입니다.
 - B. form에 작성한 정보는 POST 방식으로 제출(submit)합니다.
 - C. 리뷰를 작성할 수 있는 form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드 name	HTML 요소	Type
movie_title	input	text
title	input	text
rank	input	number
content	textarea	없음

3. HTTP method POST

- A. 데이터가 유효하다면 데이터베이스에 저장하고 상세 조회 페이지로 redirect합니다.
- B. 데이터가 유효하지 않다면 데이터를 작성하는 form을 에러메시지와 함께 사용자 화면에 표시합니다.

vi. 전체 리뷰 목록 조회

- 1. 사용자에게 응답으로 제공할 HTML파일은 index.html입니다.
- 2. index.html은 제목을 포함한 리뷰의 데이터를 자유롭게 표시합니다.
- 3. 제목을 클릭 시 해당 리뷰의 상세 조회 페이지로 이동합니다.

vii. 단일 리뷰 상세 조회

- 1. 표시할 리뷰의 pk는 URL과 함께 전달된 리뷰의 id입니다.
- 2. 사용자에게 응답으로 제공할 HTML파일은 detail.html입니다.
- 3. 리뷰 데이터가 없을 경우, 404 에러 페이지를 표시합니다.
- 4. detail.html에는 적절한 HTML 요소를 사용하여, 조회하는 리뷰의 영화제목, 제목, 내용, 평점을 표시합니다.

viii. 기존 리뷰 수정

- 1. 인증되어 있는 사용자만 기존 리뷰를 수정할 수 있습니다.
- 2. 수정할 리뷰의 pk는 URL과 함께 전달된 리뷰의 id입니다.
- 3. HTTP method GET
 - A. 사용자에게 응답으로 제공할 HTML파일은 form.html입니다.
 - B. form에 작성한 정보는 POST 방식으로 제출(submit)합니다.
 - C. 리뷰를 수정할 수 있는 Form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드 name	HTML 요소	Type
movie_title	input	text
title	input	text
rank	input	number
content	textarea	없음

4. HTTP method POST

- A. 데이터가 유효하다면 데이터베이스에 저장하고 상세 조회 페이지로 redirect합니다.
- B. 데이터가 유효하지 않다면 에러메시지를 포함하여 데이터를 작성하는 Form을 다시 표시합니다.

ix. 리뷰 데이터 삭제

1. 삭제할 리뷰의 pk는 URL과 함께 전달된 리뷰의 id입니다.
2. HTTP method는 GET과 POST 중 선택하여 구성합니다.
3. 삭제가 완료되면 전체 리뷰 목록 조회 페이지로 redirect합니다.

x. 추가적인 Styling

1. Bootstrap을 활용하여 자유롭게 스타일링 합니다.

4. 결과

- A. 제출 기한은 **21.09.17(금) 18:00**까지입니다. 제출기한을 엄수하여 주시기 바랍니다.
- B. 반드시 README.md 파일에 각 단계별 구현 과정, 학습 내용 및 어려웠던 부분을 기록하여 제출합니다.
- C. 위에 명시된 사항은 최소 조건이며 추가적인 기능을 구현할 수 있습니다.
- D. <https://lab.ssafy.com>에 pj06 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
- E. 각 반 담당 교수님을 Maintainer로 설정해야 합니다.