

프레임워크 기반 웹페이지 구현

1. 목표

- 데이터를 생성, 조회, 수정, 삭제 할 수 있는 Web Application 제작
- Python Web Framework를 통한 데이터 조작
- Object Relational Mapping에 대한 이해
- django ModelForm을 활용한 HTML과 사용자 요청 데이터 관리

2. 준비사항

A. 언어

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

B. 도구

- i. vsCode
- ii. Chrome Browser

3. 요구사항

커뮤니티 서비스의 게시판 기능 개발을 위한 단계로, 모델폼을 활용하여 데이터의 생성, 조회, 수정, 삭제를 하는 로직을 완성합니다. 해당 기능은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다.

A. 프로젝트 구조

pjt04/은 startproject 명령어로 생성되는 project 디렉토리입니다.

community/는 startapp 명령어로 생성되는 application 디렉토리입니다.

```
pjt04/  
  __init__.py  
  asgi.py  
  settings.py  
  urls.py  
  wsgi.py
```

```

community/
  migrations/
  templates/
    community/
      form.html
      review_list.html
      review_detail.html
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  test.py
  urls.py
  views.py

templates/
  base.html

manage.py
...

```

B. Model

정의할 모델 클래스의 이름은 Review이며, 다음과 같은 정보를 저장합니다.

필드명	자료형	설명
title	String(<= 100)	리뷰의 제목
movie_title	String(<= 50)	리뷰하는 영화의 제목
content	Text	리뷰의 내용
rank	Integer	리뷰의 평점
created_at	DateTime	리뷰가 최초 작성된 시간(자동 저장)
updated_at	DateTime	리뷰가 최종 수정된 시간(자동 저장)

C. Form

Review의 데이터 검증, 저장, 에러메세지, HTML을 모두 관리하기 위해 ModelForm을 사용합니다. 해당 ModelForm은 title, movie_title, content, rank 필드를 검증 및 HTML로 표시합니다.

D. Admin

Review는 관리자 페이지에서 데이터의 생성, 조회, 수정, 삭제 가능해야 합니다. (Admin 페이지에서 테스트 목적으로 데이터를 생성, 조회, 수정, 삭제 할 수 있습니다.

E. URL

URL은 app_name과 name을 설정하여 적용합니다.

HTTP verb	URL 패턴	설명
GET & POST	/community/create/	Form 표시 및 신규 리뷰 생성
GET	/community/	전체 리뷰 목록 조회 페이지
GET	/community/<review_pk>/	단일 리뷰 상세 조회 페이지
GET & POST	/community/<review_pk>/update/	Form 표시 및 기존 리뷰 수정
GET / POST	/community/<review_pk>/delete/	기존 리뷰 삭제

F. View & Template

i. 공유 템플릿 생성 및 사용

1. 모든 HTML파일은 base.html을 확장(extends)하여 사용합니다.
2. base.html은 모든 페이지가 공유하는 상단 네비게이션 바를 표시합니다.
3. 네이게이션 바는 전체 리뷰 목록 조회 페이지와 새로운 리뷰 작성 페이지로 이동 할 수 있는 링크를 포함합니다.

ii. 새로운 리뷰 작성(GET)

1. 사용자에게 응답으로 제공할 HTML파일은 form.html입니다.
2. 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

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

iii. 새로운 리뷰 작성(POST)

1. 기존에 정의한 ModelForm을 사용하여 데이터를 검증합니다.
2. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고 단일 리뷰 상세 조회 페이지로 Redirect합니다.
3. 데이터가 유효하지 않다면 에러 메시지를 포함하여 데이터를 작성하는 Form을 표시합니다.

iv. 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.

v. 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_detail.html에는 적절한 HTML 요소를 사용하여, 조회하는 리뷰의 title, movie_title, content, rank, created_at, updated_at을 표시합니다.
4. 리뷰를 수정, 삭제 할 수 있는 UI가 함께 제공되어야 합니다.

vi. 기존 리뷰 수정(GET)

1. 사용자에게 응답으로 제공할 HTML파일은 form.html입니다.
2. URL을 통해 함께 전달된 review_pk를 id(pk)로 갖는 리뷰를 수정합니다.
3. 리뷰를 수정 할 수 있는 Form과 기존의 리뷰 데이터를 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드명	Tag	Type
title	input	text
movie_title	input	text

rank	input	number
content	textarea	없음

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

vii. 기존 리뷰 수정(POST)

1. 기존에 정의한 ModelForm을 사용하여 데이터를 검증합니다.
2. 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고 단일 리뷰 상세 조회 페이지로 Redirect합니다.
3. 데이터가 유효하지 않다면 에러 메시지를 포함하여 데이터를 작성하는 Form을 표시합니다.

viii. 기존 리뷰 삭제

1. URL을 통해 함께 전달된 review_pk를 id(pk)로 갖는 리뷰를 삭제합니다.
2. HTTP method는 GET과 POST 중 선택하여 구성합니다.
3. 삭제가 완료되면 전체 리뷰 목록 조회 페이지로 Redirect합니다.

ix. 추가적인 Styling

1. Bootstrap을 활용하여 자유롭게 스타일링 합니다.

4. 결과

- A. 제출 기한은 **20.09.18(금) 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 pjt04 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
- E. 각 반 담당 교수님을 Maintainer로 설정해야 합니다.