

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 3.8+
- ii. Django 3.1.X

B. 도구

- i. vsCode
- ii. Chrome Browser

3. 요구사항

커뮤니티 서비스의 게시판 기능 개발을 위한 단계로, 영화 데이터의 CRUD를 하는 로직을 완성합니다. 해당 기능은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다.

A. 프로젝트 구조

pjt05/은 startproject 명령어로 생성되는 project 디렉토리입니다.
movies/는 startapp 명령어로 생성되는 application 디렉토리입니다.

```
pjt05/  
  __init__.py  
  asgi.py  
  settings.py  
  urls.py  
  wsgi.py  
  templates/
```

```
base.html
movies/
  migrations/
  templates/
    movies/
      *.html
  __init__.py
  admin.py
  apps.py
  forms.py
  models.py
  test.py
  urls.py
  views.py

manage.py
...
```

B. Model

정의할 모델 클래스의 이름은 `Movie`이며, 다음과 같은 정보를 저장합니다.

필드명	자료형	설명
<code>title</code>	<code>String(<= 100)</code>	제목
<code>overview</code>	<code>Text</code>	줄거리
<code>poster_path</code>	<code>String(<= 500)</code>	포스터 경로

C. Form

`Movie`의 데이터 검증, 저장, 에러메세지, HTML을 모두 관리하기 위해 `ModelForm`을 사용합니다. 해당 `ModelForm`은 `title`, `overview`, `poster_path` 필드를 검증 및 HTML로 표시합니다.

D. Admin

위에서 정의한 모델 `Movie`는 관리자 페이지에서 데이터의 생성, 조회, 수정, 삭제 가능해야 합니다.

E. URL

HTTP verb	URL 패턴	설명
GET & POST	/movies/create/	Form 표시 및 영화 데이터 생성
GET	/movies/	전체 영화 목록 조회
GET	/movies/<pk>/	단일 영화 상세 조회
GET & POST	/movies/<pk>/update/	Form 표시 및 영화 데이터 수정
POST	/movies/<pk>/delete/	단일 영화 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. 네비게이션 바는 전체 영화 목록 조회 페이지와 새로운 영화 작성 페이지로 이동 할 수 있는 링크를 포함합니다.

ii. 전체 영화 목록 조회

1. 데이터베이스에 존재하는 모든 영화의 목록을 표시합니다.
2. 사용자에게 응답으로 제공할 HTML파일은 index.html 입니다.
3. index.html은 base.html을 확장합니다.
4. index.html에는 적절한 HTML요소를 사용하여 title과 포스터 이미지를 표시하며, title을 클릭 시 해당 영화의 상세 조회 페이지로 이동합니다.

iii. 새로운 영화 작성 (GET)

1. 사용자에게 응답으로 제공할 HTML파일은 create.html or form.html입니다.
2. create.html은 base.html을 확장합니다.
3. 영화를 작성할 수 있는 Form을 표시하며, ModelForm을 이용하여 다음과 같은 input 요소들을 포함해야 합니다.

필드명	HTML Tag	Type
title	input	text
overview	textarea	없음
poster_path	input	text

4. Form에 작성한 정보는 제출(submit)시, 사용자 제출 데이터를 저장하는 URL로 요청과 함께 전송됩니다.

iv. 새로운 영화 작성 (POST)

1. ModelForm을 이용하여 요청과 함께 전송된 데이터를 데이터베이스에 저장합니다.
2. 저장이 완료되면 아래의 동작 중 하나를 선택하여 수행합니다.
 - A. 전체 영화 목록 조회 페이지로 Redirect합니다.
 - B. 저장한 영화의 상세 조회 페이지로 Redirect합니다.

v. 단일 영화 상세 조회

1. URL을 통해 함께 전달된 pk에 해당하는 영화 상세정보를 HTML에 표시해야 합니다.
2. 사용자에게 응답으로 제공할 HTML은 detail.html입니다.
3. detail.html은 base.html을 확장합니다.
4. detail.html에는 적절한 HTML 요소를 사용하여, 조회하는 영화의 title, overview, poster_path를 표시합니다.

vi. 영화 데이터 수정 (GET)

1. 사용자에게 응답으로 제공할 HTML파일은 update.html or form.html입니다.
2. URL을 통해 함께 전달된 pk를 pk(id)로 갖는 영화를 수정합니다.
3. 영화를 수정할 수 있는 Form과 기존의 영화 데이터를 표시하며, ModelForm을 이용하여 다음과 같은 input 요소들을 포함해야 합니다.

필드명	HTML Tag	Type
title	input	text
overview	textarea	없음
poster_path	input	text

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

vii. 영화 데이터 수정 (POST)

1. ModelForm을 이용하여 데이터를 검증하고 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고 상세 조회 페이지로 Redirect합니다.
2. 데이터가 유효하지 않다면 에러 메시지를 포함하여 데이터를 작성하는 Form을 표시합니다.

viii. 영화 데이터 삭제

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

ix. 추가적인 Styling

1. Bootstrap을 활용하여 자유롭게 스타일링 합니다.

4. 결과

- A. 제출 기한은 **21.03.19(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 pjt05 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
- E. 각 반 담당 교수님을 Maintainer로 설정해야 합니다.