

DB를 활용한 웹 페이지 구현

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application 제작
- Django web framework를 통한 데이터 조작
- ORM(Object Relational Mapping)에 대한 이해
- Django ModelForm을 활용한 사용자 요청 데이터 유효성 검증

2. 준비사항

A. 개발도구 및 라이브러리

- i. Visual Studio Code
- ii. Google Chrome Browser
- iii. Django 3.2+

3. 요구사항

커뮤니티 서비스의 **게시판 기능** 개발을 위한 단계로, 영화 데이터의 생성, 조회, 수정, 삭제 가능한 어플리케이션을 완성합니다. 해당 기능은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다. 아래 기술된 사항들은 필수적으로 구현해야 하는 내용입니다.

Django 프로젝트 이름은 **pjt06**, 앱 이름은 **movies**로 지정합니다.

4. Model

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

필드명	데이터 유형	역할
title	varchar(20)	영화 제목
audience	integer	관객 수
release_date	date	개봉일
genre	varchar(30)	장르
score	float	평점
poster_url	text	포스터 경로
description	text	줄거리

5. URL

URL 패턴	역할
/movies/	전체 영화 목록 페이지 조회
/movies/create/	영화 작성 페이지 조회 & 단일 영화 데이터 저장
/movies/<pk>/	단일 영화 상세 페이지 조회
/movies/<pk>/update/	영화 수정 페이지 조회 & 단일 영화 데이터 수정
/movies/<pk>/delete/	단일 영화 데이터 삭제

- URL의 pk는 영화 데이터의 id입니다.

6. Admin

위에서 정의한 모델 Movie을 Admin site에 등록합니다.

MovieAdmin 클래스를 작성하며 등록 후 Admin site에서 데이터의 생성, 조회, 수정, 삭제가 가능해야 합니다.

7. View

View 함수명	역할	허용 HTTP Method
index	전체 영화 데이터 조회 및 index.html 렌더링	GET
create	create.html 렌더링 & 유효성 검증 및 영화 데이터 저장 후 detail.html 리다이렉트	GET & POST
detail	단일 영화 데이터 조회 및 detail.html 렌더링	GET
update	수정 대상 영화 데이터 조회 및 update.html 렌더링 & 유효성 검증 및 영화 데이터 수정 후 detail.html 리다이렉트	GET & POST
delete	단일 영화 데이터 삭제 및 index.html 리다이렉트	POST

8. Form

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

B. genre 필드

i. HTML select element를 출력해 코미디, 공포 혹은 로맨스 장르 데이터를 선택할 수 있도록 합니다.

C. score 필드

- i. input element의 type은 number로 설정합니다.
- ii. input element attribute 중 step은 0.5, min은 0 그리고 max는 5로 설정합니다.

D. release_date 필드

- i. input element의 type은 date로 설정합니다.

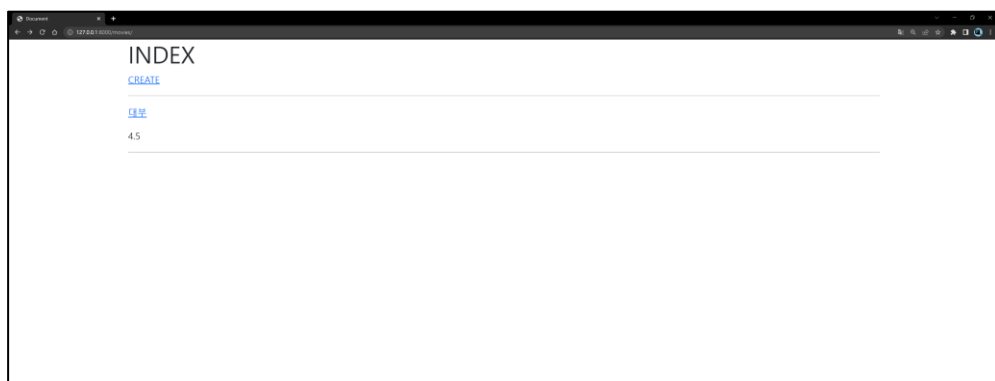
9. Template

A. 공유 템플릿 (base.html)

- i. 모든 템플릿 파일은 base.html을 상속받아 사용합니다.
- ii. bootstrap CDN을 포함합니다.

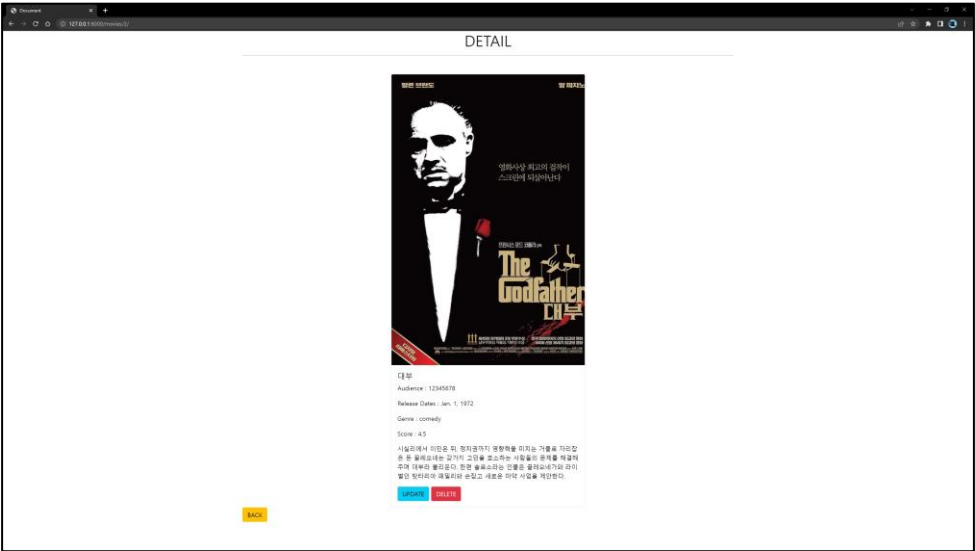
B. 전체 영화 목록 조회 템플릿 (index.html)

- i. 데이터베이스에 존재하는 모든 영화의 목록을 표시합니다.
- ii. 사용자에게 응답으로 제공할 HTML파일은 index.html 입니다.
- iii. index.html은 base.html을 확장합니다.
- iv. index.html에는 적절한 HTML 요소를 사용하여 영화 제목 및 평점을 표시하며, 제목을 클릭 시 해당 영화의 상세 조회 페이지로 이동합니다.



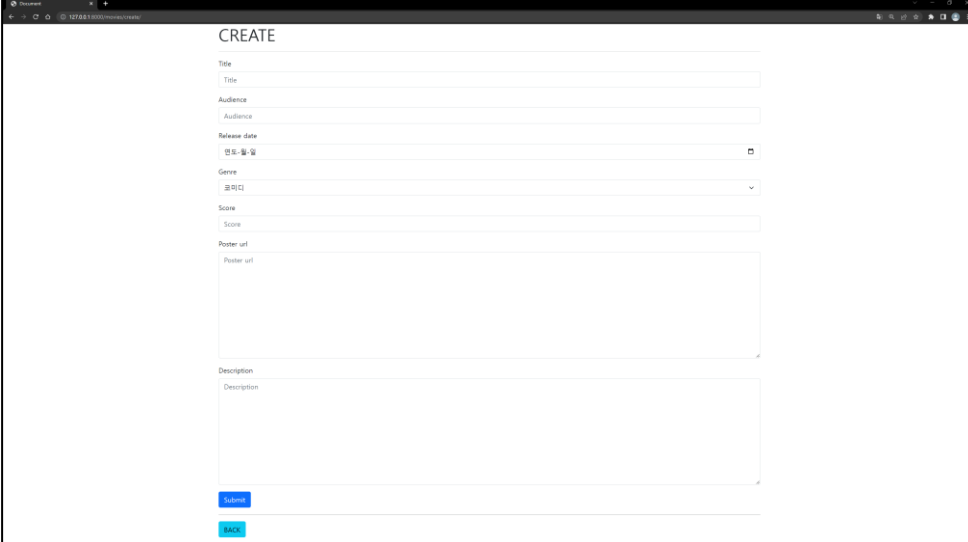
C. 영화 상세 정보 페이지 (detail.html)

- i. 특정 영화의 상세 정보를 표시합니다.
- ii. Bootstrap Card Component를 사용합니다.
- iii. 해당 영화의 수정 및 삭제 버튼을 표시합니다.
- iv. index.html로 이동하는 버튼을 표시합니다.



D. 영화 작성 페이지 (create.html)

- i. 영화 작성 Form을 표시합니다.
- ii. 표시되는 Form은 Movie 모델 클래스에 기반한 ModelForm이어야 합니다.
- iii. Form에 작성한 정보는 제출(submit)시, 영화 데이터를 저장하는 URL로 요청과 함께 전송됩니다.
- iv. index.html로 이동하는 버튼을 표시합니다.



The screenshot shows a web browser window with a URL bar displaying '127.0.0.1:8000/'. The page title is 'CREATE'. The form contains the following fields:

- Title: A text input field.
- Audience: A text input field.
- Release date: A date input field with a calendar icon.
- Genre: A dropdown menu with '코미디' (Comedy) selected.
- Score: A text input field.
- Poster url: A text input field.
- Description: A large text area.

At the bottom of the form, there are two buttons: a blue 'Submit' button and a green 'Cancel' button.

E. 영화 수정 페이지 (update.html)

- i. 영화 수정 Form을 표시합니다.
- ii. 표시되는 Form은 Movie 모델 클래스에 기반한 ModelForm이어야 합니다.
- iii. Form에는 기존 영화 데이터를 출력합니다.
- iv. Form에 작성한 정보는 제출(submit)시, 영화 데이터를 수정하는 URL로 요청과 함께 전송됩니다.
- v. index.html로 이동하는 버튼을 표시합니다.

UPDATE

Title

Audience

Release date

Genre

Score

Poster url

Description

10. 기타

- A. 명세에 작성된 것 이외의 아이디어는 자유롭게 구현합니다.

11. 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 pjt06 프로젝트를 생성하고 업로드하여 제출합니다.
- E. 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.
- F. 제출 후 파일구조는 아래와 같습니다.

pjt06/

movies/

migrations/

templates/

movies/

index.html

detail.html

create.html

update.html

__init__.py

admin.py

apps.py

forms.py

models.py

tests.py

urls.py

views.py

pjt06/

__init__.py

asgi.py

settings.py

urls.py

wsgi.py

templates/

base.html

manage.py

requirements.txt

.gitignore

db.sqlite3

README.md