

프레임워크 기반 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
- 관리자 페이지를 통한 데이터 관리

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 **pjt05**, 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/new/	새로운 영화 작성 페이지 조회
/movies/create/	단일 영화 데이터 저장
/movies/<pk>/	단일 영화 상세 페이지 조회
/movies/<pk>/edit/	기존 영화 수정 페이지 조회
/movies/<pk>/update/	단일 영화 데이터 수정
/movies/<pk>/delete/	단일 영화 데이터 삭제

6. Admin

위에서 정의한 모델 Movie를 Admin site에 등록합니다.
MovieAdmin 클래스를 작성하며 등록 후 Admin site에서
데이터의 생성, 조회, 수정, 삭제가 가능해야 합니다.

7. View

View 함수명	역할
index	전체 영화 데이터 조회 및 index.html 렌더링
new	장르 데이터 제공 및 new.html 렌더링
create	새로운 영화 데이터 저장 및 detail.html 리다이렉트
detail	단일 영화 데이터 조회 및 detail.html 렌더링
edit	수정 대상 영화 데이터 조회 및 edit.html 렌더링
update	영화 데이터 수정 및 detail.html 리다이렉트
delete	단일 영화 데이터 삭제 및 index.html 리다이렉트

8. Template

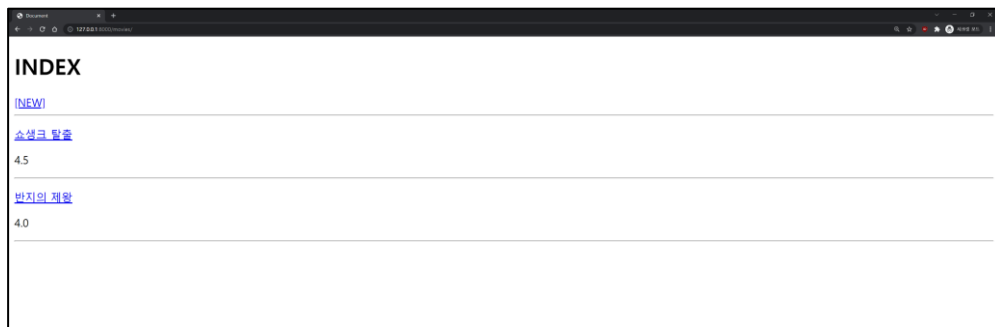
“일부 이미지는 가독성을 위해 확대되었습니다.”

A. 공유 템플릿 (base.html)

- i. 모든 템플릿 파일은 base.html을 상속받아 사용합니다.

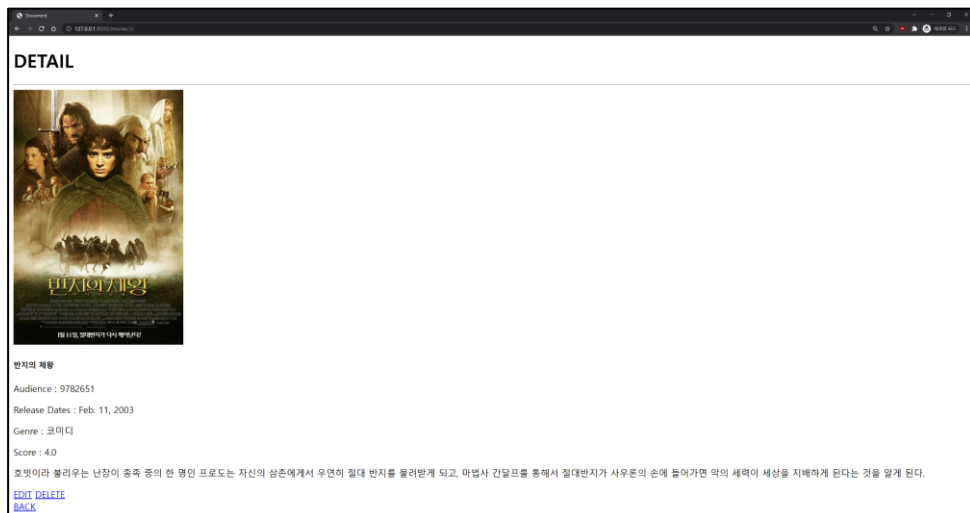
B. 전체 영화 목록 조회 템플릿 (index.html)

- 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.



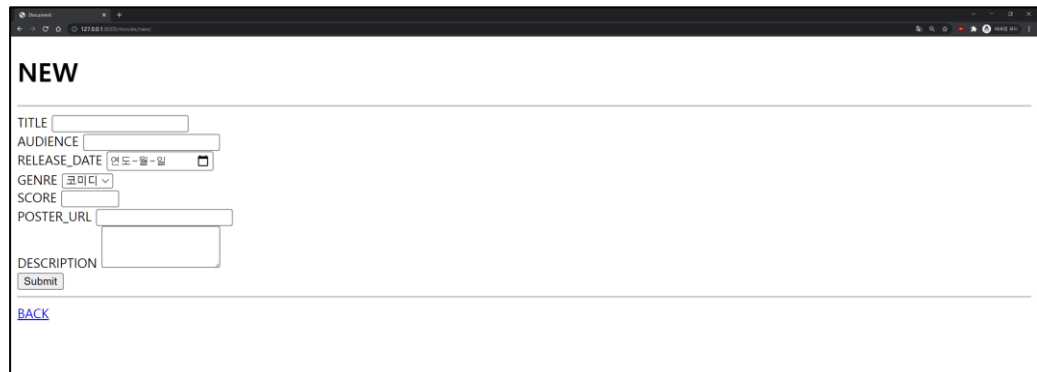
C. 영화 상세 정보 페이지 (detail.html)

- i. 특정 영화의 상세 정보를 표시합니다.
- ii. 해당 영화의 수정 및 삭제 버튼을 표시합니다.
- iii. index.html로 이동하는 버튼을 표시합니다.



D. 영화 작성 페이지 (new.html)

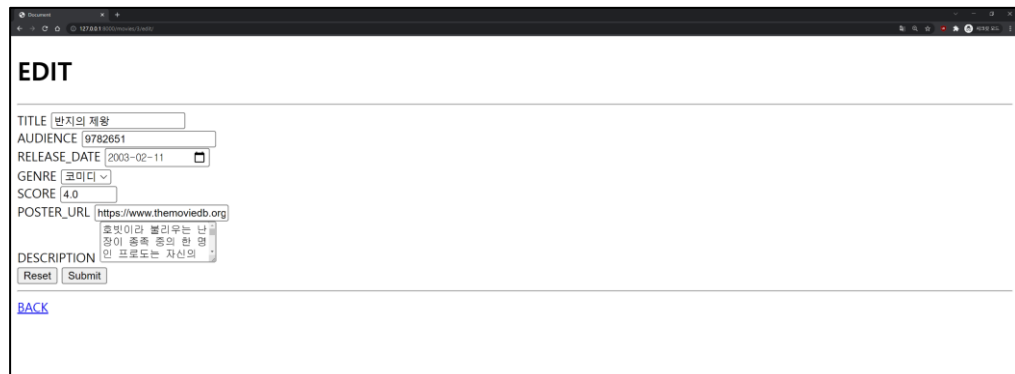
- 영화 작성 Form을 표시합니다.
- Form에 작성한 정보는 제출(submit)시, 영화 데이터를 저장하는 URL로 요청과 함께 전송됩니다.
- index.html로 이동하는 버튼을 표시합니다.



The screenshot shows a web browser window displaying a form titled "NEW". The form contains the following fields: "TITLE" (text input), "AUDIENCE" (text input), "RELEASE_DATE" (date picker set to "연도-월-일"), "GENRE" (dropdown menu with "코미디" selected), "SCORE" (text input), and "POSTER_URL" (text input). Below these fields is a "DESCRIPTION" label and a "Submit" button. At the bottom of the form, there is a blue link labeled "BACK".

E. 영화 수정 페이지 (edit.html)

- 영화 수정 Form을 표시합니다.
- Form에는 기존 영화 데이터를 출력합니다.
- Form에 작성한 정보는 제출(submit)시, 영화 데이터를 수정하는 URL로 요청과 함께 전송됩니다.
- index.html로 이동하는 버튼을 표시합니다.



The screenshot shows a web browser window displaying a form titled "EDIT". The form contains the following fields: "TITLE" (text input with "반지의 제왕"), "AUDIENCE" (text input with "9782651"), "RELEASE_DATE" (date picker set to "2003-02-11"), "GENRE" (dropdown menu with "코미디" selected), "SCORE" (text input with "4.0"), and "POSTER_URL" (text input with "https://www.themoviedb.org"). Below these fields is a "DESCRIPTION" label and a text area containing "호빗이라 불리우는 난장이 종족 중의 한 명인 프롭도는 자신의". At the bottom of the form, there are "Reset" and "Submit" buttons. At the bottom of the page, there is a blue link labeled "BACK".

9. 기타

- A. 명세에 작성된 것 이외의 아이디어는 자유롭게 구현합니다.

10. 제출

- A. 제출기한은 금일 18:00까지입니다. 제출기한을 지키어 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 pjt05 프로젝트를 생성하고 업로드하여 제출합니다.
- E. 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.
- F. 제출 후 파일구조는 아래 예시와 유사합니다.

```
pjt05/  
  movies/  
    migrations/  
    templates/  
      movies/  
        index.html  
        detail.html  
        new.html  
        edit.html  
      __init__.py  
      admin.py  
      apps.py  
      models.py  
      tests.py  
      urls.py  
      views.py  
pjt05/  
  __init__.py  
  asgi.py  
  settings.py  
  urls.py  
  wsgi.py  
templates/  
  base.html  
manage.py  
requirements.txt  
.gitignore  
db.sqlite3  
README.md
```