

PJT06 웹 페이지 구현

1. Model만들기

명세에 나와있는 것처럼 페이지에 사용할 모델을 생성한다.

```
class Movie(models.Model):
    title = models.CharField(max_length=100)
    title_en = models.CharField(max_length=100)
    audience = models.IntegerField()
    open_date = models.DateTimeField(auto_now_add=True)
    genre = models.CharField(max_length=100)
    watch_grade = models.CharField(max_length=50)
    total_score = models.FloatField()
    poster_url = models.TextField()
    description = models.TextField()
    def get_absolute_url(self):
        return reverse("movie:movie_detail", kwargs={"movie_id": self.id})

class Comment(models.Model):
    movie = models.ForeignKey(Movie, on_delete=models.CASCADE)
    content = models.CharField(max_length=200)
    comment_score = models.CharField(max_length=200)
```

2. main page

기본 html페이지인 base.html을 extends를 사용하여 상속받고 기본 페이지에 들어갈 새 영화등록, DB에 등록되어있는 영화리스트를 출력한다.
load static을 사용하여 static 폴더 안에 있는 내용들을 사용한다.

```
{% extends 'movie/base.html' %}
{% load static %}
{% block title %} movie listt {% endblock %}
{% block body %}


    <h1>This is Movie listt</h1>
<div><a href="{% url 'movie:new_movie' %}">
    <button>새 영화 등록</button></a></div>
{% if movies %}
    <ul>
        {% for movie in movies %}
            <li>
                <a href="{{ movie.get_absolute_url }}">
                    {{ movie.title }}</a> {{ movie.score }}
            </li>
        {% endfor %}
    </ul>
{% endif %}
```

```
{% endblock body %}
```



UNCLE MOVIE WANT "YOU"

새 영화 등록

- [asdasd](#)
- [쥬커](#)
- [엑시트](#)

3. url, views 생성

CRUD 순으로 view를 생성하고 페이지 이동을 위해 `urls.py`에서 `urlpatterns`를 설정해준다.

4. 폼 생성

```
class MovieModelForm(forms.ModelForm):  
    title = forms.CharField(min_length=1, max_length=200)  
  
    class Meta:  
        model = Movie
```

```
fields = '__all__'
```

```
class CommentModelForm(forms.ModelForm):
    content = forms.CharField(min_length=2, max_length=100)
    comment_score = forms.CharField(min_length=2, max_length=100)
    class Meta:
        model = Comment
        fields = ('content', 'comment_score', )
```

5. 새 영화 생성

폼 생성 요청을 받으면 html을 거쳐 새 db를 생성한다.

```
@require_http_methods(['GET', 'POST'])
def new_movie(request):
    if request.method == 'POST':
        form = MovieModelForm(request.POST)
        if form.is_valid():
            movie = form.save()
            return redirect('movie:movie_list')
    else:
        form = MovieModelForm()
    return render(request, 'movie/new_movie.html', {
        'form': form,
    })
```

```
{% extends 'movie/base.html' %}

{% block title %}New Movie{% endblock title %}

{% block body %}
<h1>New Movie</h1>

{% include 'movie/_form.html' %}

{% endblock body %}
```

New Movie

Title: 버드박스

Title en: Bird Box

Audience: 1000000001

Genre: 다큐멘터리

Watch grade: 전체 관람가

Total score: 9.8

https://i.ytimg.com/vi/F8_4LsfVu44/maxresdefault.jpg

Poster url:

안대를 쓰고 리신의 삶을 경험해보는 다큐멘터리 영화.

Description:

제출

영화의 상세 설명을 보기위해 리스트에서 클릭을 통해 view를 거쳐 url을 통해 이동되면 다음과 같이 출력된다.

```
def movie_detail(request, movie_id):
    movie = get_object_or_404(Movie, id=movie_id)
    comments = movie.comment_set.all().order_by('id')
    comment_form = CommentModelForm()
    return render(request, 'movie/movie_detail.html', {
        'movie': movie,
        'comments': comments,
        'comment_form': comment_form,
    })
```

버드박스, Bird Box

관객수 : 1,000,000,001

개봉일 : 2019년 10월 18일 10:04 오전

장르 : 다큐멘터리

관람등급 : 전체 관람가

평점 :



안대를 쓰고 리신의 삶을 경험해보는 다큐멘터리 영화.

목록

수정

삭제

한줄평:

평점:

한줄평

수정을 클릭하면 새 데이터를 쓸 때 본 창이 뜨면서 고칠 수 있다.

조커, jeoker

관객수 : 10,000,000

개봉일 : 2019년 10월 18일 7:35 오전

장르 : 코미디

관람등급 : 12세 이용가

평점 :



조커가 자신의 꿈을 찾아 떠나는 신나는 모험

목록

수정

삭제

한줄평:

평점:

한줄평

Update movie

Title:

Title en:

Audience:

Genre:

Watch grade:

Total score:

Poster url:

Description:

느낀 점.

- db사이의 연동이 힘들다고 생각되었다.