

종합 실습 프로젝트 #02

1. 목표

- 데이터를 생성, 조회, 삭제, 수정할 수 있는 Web Application 제작
- Python Web Framework를 통한 데이터 조작
- Database 1:N Relation에 대한 이해
- 회원 가입 및 로그인 기능을 활용한 영화 평점 사이트 제작

2. 준비 사항

1. Python Web Framework 사용을 위한 환경 설정 - Python 3.7+
2. Python Web Framework - Django

3. 요구 사항

1. 데이터베이스

- ORM을 통해서 작성될 클래스의 이름은 `Movie`, `Rating`이며, 다음과 같은 정보를 저장합니다.

- **Movie**

필드명	자료형	설명
title	String	영화 제목
description	Text	영화 개요
poster	Image	영화 포스터 사진
created_at	DateTime	생성 날짜(자동 생성)
updated_at	DateTime	수정 날짜(자동 생성)
user	Integer(FK)	생성한 User의 id(pk)

- **Rating**

필드명	자료형	설명
score	Float	평점
content	String	한줄평
created_at	DateTime	생성 날짜(자동 생성)
updated_at	DateTime	수정 날짜(자동 생성)
user	Integer(FK)	생성한 User의 id(pk)

2. 페이지(기능)

1. 회원 가입 Form

1. 해당 페이지에 접근하는 URL은 `/accounts/signup/` 입니다.
2. 해당 페이지는 로그인을 하지 않았을 경우에만 접근 가능합니다.
3. 회원 가입을 할 수 있는 Form이 표시 되며, Django에서 기본적으로 제공하는 Form Class를 사용합니다.
4. Form에 작성된 정보는 Form Submit 버튼 클릭 시 `회원 가입` 페이지로 생성 요청(request)과 함께 전송됩니다.
5. 요청을 보내는 방식(method)은 POST를 사용합니다.

2. 회원 가입

1. 해당 페이지에 접근하는 URL은 `/accounts/signup/` 입니다.
2. 해당 페이지는 로그인을 하지 않았을 경우에만 접근 가능합니다.
3. 이전 페이지로부터 전송 받은 데이터를 데이터베이스에 저장하여 회원을 생성합니다.
4. 회원 생성 후, 해당 회원으로 로그인 되도록 합니다.
5. 정상적으로 회원 가입이 된 경우, `영화 목록` 페이지로 Redirect 합니다.

3. 로그인 Form

1. 해당 페이지에 접근하는 URL은 `/accounts/login/` 입니다.
2. 해당 페이지는 로그인을 하지 않았을 경우에만 접근 가능합니다.
3. 로그인을 할 수 있는 Form이 표시 되며, Django에서 기본적으로 제공하는 Form Class를 사용합니다.
4. Form에 작성된 정보는 Form Submit 버튼 클릭 시 `로그인` 페이지로 로그인 요청(request)과 함께 전송됩니다.
5. 요청을 보내는 방식(method)은 POST를 사용합니다.

4. 로그인

1. 해당 페이지에 접근하는 URL은 `/accounts/login/` 입니다.
2. 해당 페이지는 로그인을 하지 않았을 경우에만 접근 가능합니다.
3. 이전 페이지로부터 전송 받은 데이터를 사용하여 로그인을 수행합니다.
4. 정상적으로 로그인이 된 경우, `영화 목록` 페이지로 Redirect 합니다.

5. 로그아웃

1. 해당 페이지에 접근하는 URL은 `/accounts/logout/` 입니다.
2. 해당 페이지는 로그인을 했을 경우에만 접근 가능합니다.
3. 현재 로그인 되어 있는 사용자를 로그아웃 합니다.

4. 정상적으로 로그아웃이 된 경우, **영화 목록** 페이지로 Redirect 합니다.

6. 영화 목록

1. 해당 페이지에 접근하는 URL은 **/movies/** 입니다.
2. 데이터베이스에 존재하는 모든 영화의 목록이 표시 되며, 각 영화의 **title**, **score** 가 표시됩니다.
3. **title** 을 클릭 시, 해당 **영화 정보 조회** 페이지로 이동합니다.
4. 영화 목록 최상단에 **새 영화 등록** 링크가 있으며, 클릭 시 **영화 정보 생성 Form** 페이지로 이동합니다.

7. 영화 정보 생성 Form

1. 해당 페이지에 접근하는 URL은 **/movies/new/** 입니다.
2. 해당 페이지는 로그인을 했을 경우에만 접근 가능합니다.
3. 영화 정보를 작성할 수 있는 Form이 표시 되며, 다음과 같은 input들을 가지고 있습니다.

필드명	HTML Tag	Type
title	input	text
description	textarea	text
poster	input	file

4. Form에 작성된 정보는 Form Submit 버튼 클릭 시 **영화 정보 생성** 페이지로 생성 요청 (request)과 함께 전송됩니다.
5. 요청을 보내는 방식(method)은 POST를 사용합니다.

8. 영화 정보 생성

1. 해당 페이지에 접근하는 URL은 **/movies/new/** 입니다.
2. 해당 페이지는 로그인을 했을 경우에만 접근 가능합니다.
3. 이전 페이지로부터 전송 받은 데이터와 현재 로그인 된 사용자의 정보를 함께 데이터베이스에 저장합니다.
4. 해당 페이지에서 저장한 영화 정보를 조회하는 **영화 정보 조회** 페이지로 Redirect 합니다.

9. 영화 정보 조회

1. 해당 페이지에 접근하는 URL은 **/movies/1/**, **/movies/2/** 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.
2. 해당 Primary Key를 가진 영화의 **모든 정보**가 표시됩니다.
3. 영화 정보의 최하단에는 **목록**, **수정**, **삭제** 링크가 있으며, 클릭 시 각각 **영화 목록**, 해당 **영화 정보 수정 Form**, 해당 **영화 정보 삭제** 페이지로 이동합니다.

10. 영화 정보 수정 Form

1. 해당 페이지에 접근하는 URL은 **/movies/1/edit/**, **/movies/2/edit/** 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.
2. 해당 페이지는 로그인을 한 상태에서 현재 로그인한 사용자와 생성한 사용자가 같을 경우에만 접근 가능합니다.

- 해당 Primary Key를 가진 영화 정보를 수정할 수 있는 Form이 표시되며, 정보가 입력된 채로 다음과 같은 input들을 가지고 있습니다.

필드명	HTML Tag	Type
title	input	text
description	textarea	text
poster	input	file

- Form에 작성된 정보는 Form Submit 버튼 클릭 시 **영화 정보 수정** 페이지로 수정 요청(request)과 함께 전송됩니다.
- 요청을 보내는 방식(method)은 POST를 사용합니다.

11. 영화 정보 수정

- 해당 페이지에 접근하는 URL은 `/movies/1/edit/`, `/movies/2/edit/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.
- 해당 페이지는 로그인한 상태에서 현재 로그인한 사용자와 생성한 사용자가 같을 경우에만 접근 가능합니다.
- 해당 Primary Key를 가진 영화 정보를 이전 페이지로부터 전송 받은 데이터로 변경하여 데이터베이스에 저장합니다.
- 해당 페이지에서 수정한 영화 정보를 조회하는 **영화 정보 조회** 페이지로 Redirect 합니다.

12. 영화 정보 삭제

- 해당 페이지에 접근하는 URL은 `/movies/1/delete/`, `/movies/2/delete/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.
- 해당 페이지는 로그인한 상태에서 현재 로그인한 사용자와 생성한 사용자가 같을 경우에만 접근 가능합니다.
- 해당 Primary Key를 가진 영화 정보를 데이터베이스에서 삭제합니다.
- 영화 정보 목록** 페이지로 Redirect 합니다.

13. 평점 생성 Form

- 영화 정보 조회** 페이지에서 평점 정보를 생성합니다.
- 평점을 작성할 수 있는 Form이 표시되며, 다음과 같은 input들을 가지고 있습니다.

필드명	HTML Tag	Type
score	input	number
content	textarea	text

- 해당 Form은 로그인한 경우에만 표시됩니다.
- Form에 작성된 정보는 Form Submit 버튼 클릭 시 **평점 생성** 페이지로 생성 요청(request)과 함께 전송됩니다.
- 요청을 보내는 방식(method)은 POST를 사용합니다.

9. 평점 생성

1. 해당 페이지에 접근하는 URL은 `/movies/1/ratings/new/`, `/movies/2/ratings/new/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 영화 정보의 Primary Key가 들어갑니다.
 2. 해당 페이지는 로그인을 했을 경우에만 접근 가능합니다.
 3. 이전 페이지로부터 전송 받은 데이터와 현재 로그인 된 사용자의 정보를 함께 데이터베이스에 저장합니다.
 4. 해당 페이지에서 저장한 평점의 영화 정보를 조회하는 `영화 정보 조회` 페이지로 Redirect 합니다.
10. 평점 삭제
1. 해당 페이지에 접근하는 URL은 `/movies/1/ratings/1/delete/`, `/movies/1/ratings/2/delete/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분의 첫번째는 데이터베이스에 저장된 영화 정보의 Primary Key, 두번째는 평점의 Primary Key가 들어갑니다.
 2. 해당 페이지는 로그인을 한 상태에서 현재 로그인한 사용자와 생성한 사용자가 같을 경우에만 접근 가능합니다.
 3. 두번째의 Primary Key를 가진 평점을 데이터베이스에서 삭제합니다.
 4. 첫번째의 Primary Key를 가진 영화 정보를 조회하는 `영화 정보 조회` 페이지로 Redirect 합니다.