

# 사용자 인증기반 관계형 DB 설계

## 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 Authentication System에 대한 이해
- Database many to one relationship(1:N)에 대한 이해

## 2. 준비사항

### A. 개발도구 및 라이브러리

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

## 3. 요구사항

django 프로젝트 이름은 **pjt07**, 앱 이름은 **movies**와 **accounts**로 지정합니다.  
커뮤니티 서비스의 게시판 기능 개발을 위한 단계로,  
해당 기능들은 향후 커뮤니티 서비스의 필수 기능으로 사용됩니다.  
아래 기술된 사항들은 필수적으로 구현해야 하는 내용입니다.

#### 4. Model

A. 모델 클래스 Movie는 다음과 같은 정보를 저장합니다.

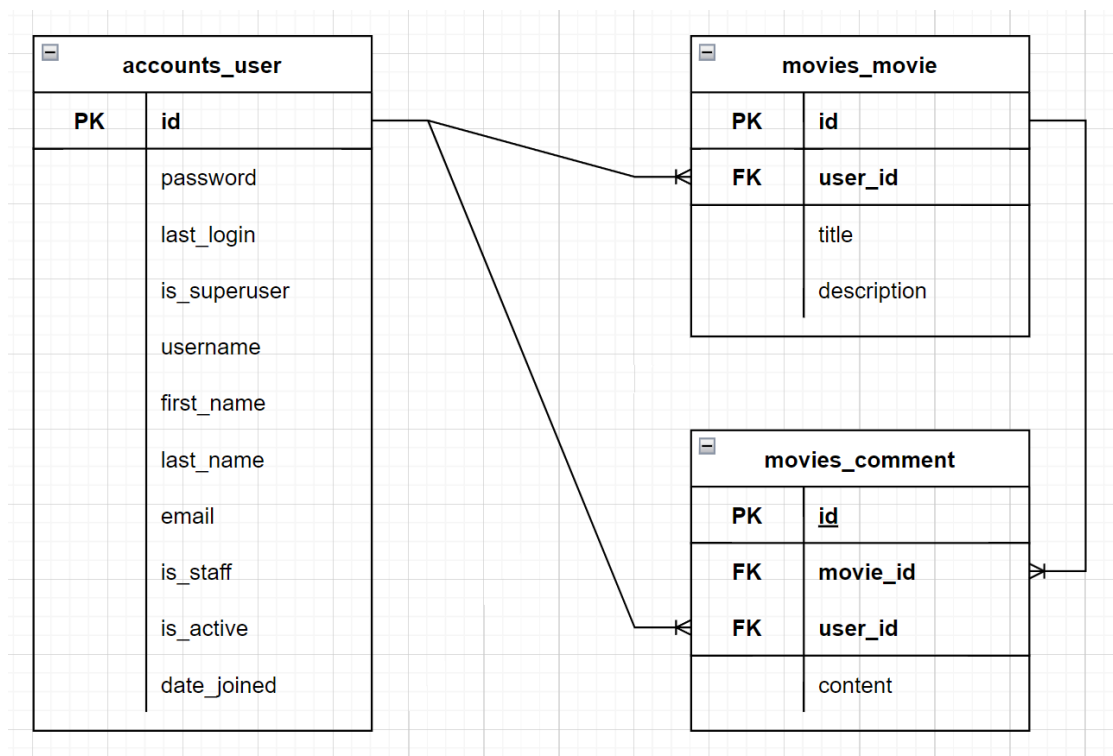
필드명	데이터 유형	역할
title	varchar(20)	영화 제목
description	text	줄거리
user_id	integer	외래 키(User 클래스 참조)

B. 모델 클래스 Comment는 다음과 같은 정보를 저장합니다.

필드명	데이터 유형	역할
content	varchar(100)	영화 제목
movie_id	integer	외래 키(Movie 클래스 참조)
user_id	integer	외래 키(User 클래스 참조)

C. 모델 클래스 User는 AbstractUser 모델 클래스를 상속받는 커스텀 모델을 사용합니다.

D. ERD



## 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
/movies/<pk>/comments/	단일 댓글 데이터 저장
/movies/<movie_pk>/comments /<comment_pk>/delete/	단일 댓글 데이터 삭제
/accounts/login/	로그인 페이지 조회 & 세션 데이터 생성 및 저장 (로그인)
/accounts/logout/	세션 데이터 삭제 (로그아웃)
/accounts/signup/	회원 생성 페이지 조회 & 단일 회원 데이터 생성 (회원가입)
/accounts/delete/	단일 회원 데이터 삭제 (회원탈퇴)
/accounts/update/	회원 수정 페이지 조회 & 단일 회원 데이터 수정 (회원정보수정)
/accounts/password/	비밀번호 수정 페이지 조회 & 단일 비밀번호 데이터 수정 (비밀번호변경)

## 6. Admin

정의한 모델 클래스 Movie, Comment, User를 Admin site에 등록합니다.

UserAdmin 클래스를 등록 후 Admin site에서

회원 데이터의 생성, 조회, 수정, 삭제가 가능해야 합니다.

## 7. View

### A. movies app

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
comments_create	유효성 검증 및 댓글 데이터 저장 후 detail.html 리다이렉트	POST
comments_delete	단일 댓글 데이터 삭제 및 detail.html 리다이렉트	POST

### B. accounts app

View 함수명	역할	허용 HTTP Method
login	login.html 렌더링 & 회원의 로그인 과정 진행	GET & POST
logout	DB와 클라이언트 쿠키에서 세션 데이터 삭제	POST
signup	signup.html 렌더링 & 유효성 검증 및 회원 데이터 저장	GET & POST
update	수정 대상 회원 데이터 조회 및 update.html 렌더링 & 유효성 검증 및 회원 데이터 수정 후 index.html 렌더링	GET & POST
delete	단일 회원 데이터 삭제 및 index.html 리다이렉트	POST
change_password	change_password.html 렌더링 & 비밀번호 변경 및 index.html 리다이렉트	GET & POST

## 8. Form

- A. Movie 모델과 Comment 모델의 데이터 검증, 저장, 에러메시지, HTML을 모두 관리하기 위해 적절한 ModelForm을 사용합니다.
- B. User 모델의 데이터 검증, 저장, 에러메시지, HTML을 모두 관리하기 위해 적절한 Form, ModelForm 그리고 커스텀 ModelForm을 사용합니다

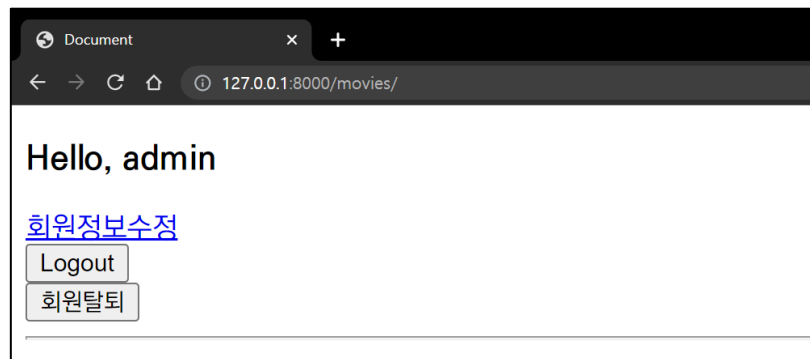
## 9. Template

### A. 공유 템플릿 (base.html)

- i. 모든 템플릿 파일은 base.html을 상속받아 사용합니다.
- ii. nav 태그를 사용한 네비게이션 바가 있습니다.
- iii. 네비게이션 바는 회원의 로그인, 비로그인 상태에 따라 다른 링크를 출력합니다.

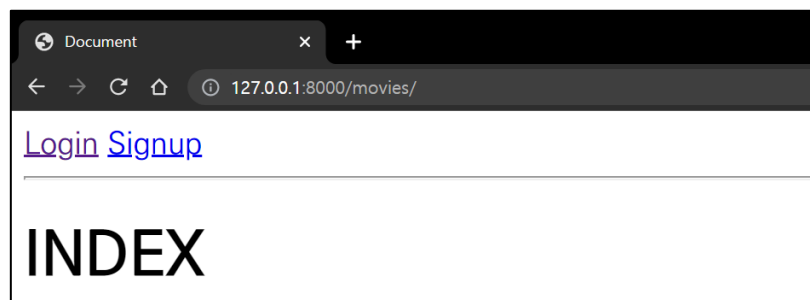
#### 1. 로그인 상태

- A. 회원정보수정
- B. 로그아웃
- C. 회원탈퇴



#### 2. 비로그인 상태

- A. 로그인
- B. 회원가입



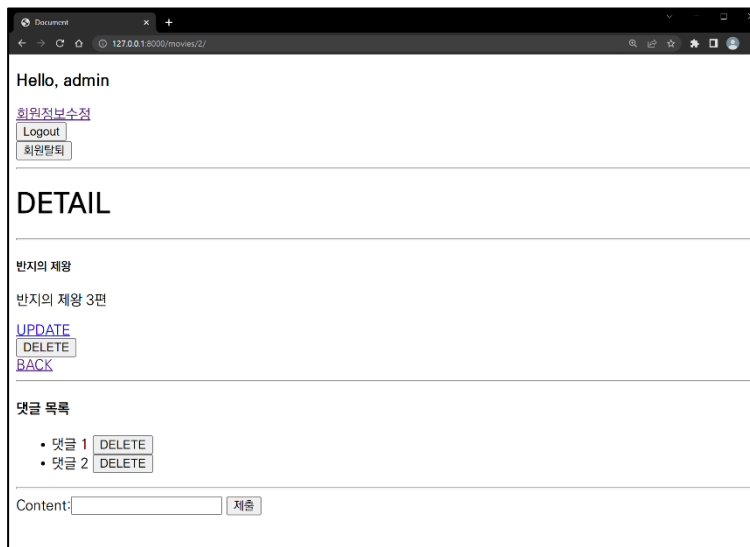
## B. 전체 영화 목록 조회 템플릿 (index.html)

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



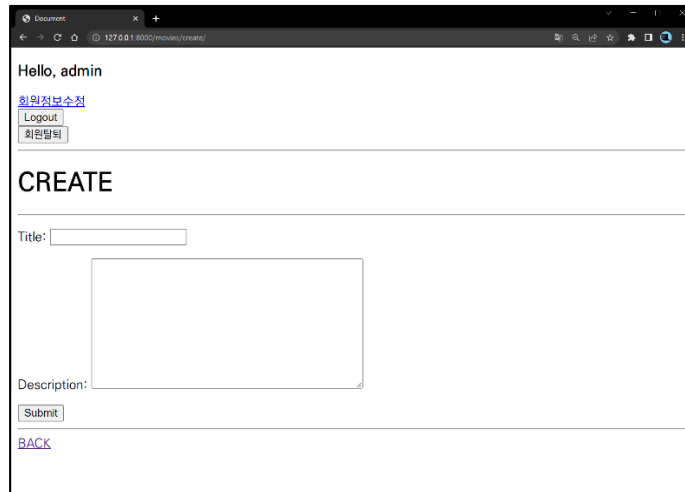
## C. 영화 상세 정보 페이지 (detail.html)

- 특정 영화의 상세 정보를 표시합니다.
- 댓글을 작성할 수 있는 form을 출력합니다.
- 댓글 삭제 버튼은 댓글 작성자에게만 표시합니다.
- 해당 영화의 수정 및 삭제 버튼을 표시합니다.
- 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.



Document

127.0.0.1:8000/movies/create/

Hello, admin

[회원정보수정](#)

[Logout](#)

[회원탈퇴](#)

---

## CREATE

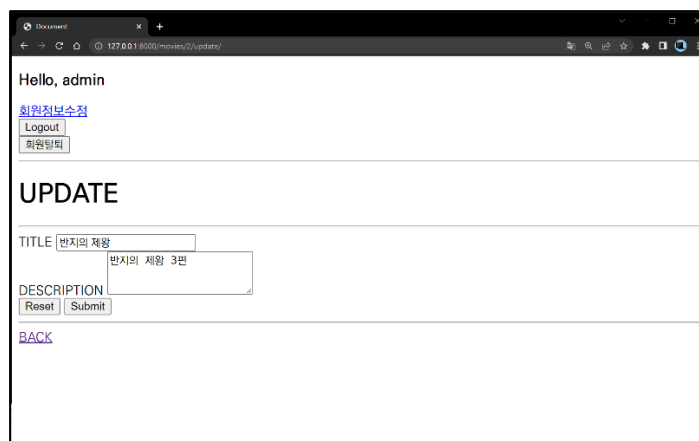
Title:

Description:

[BACK](#)

#### 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.



Document

127.0.0.1:8000/movies/2/update/

Hello, admin

[회원정보수정](#)

[Logout](#)

[회원탈퇴](#)

---

## UPDATE

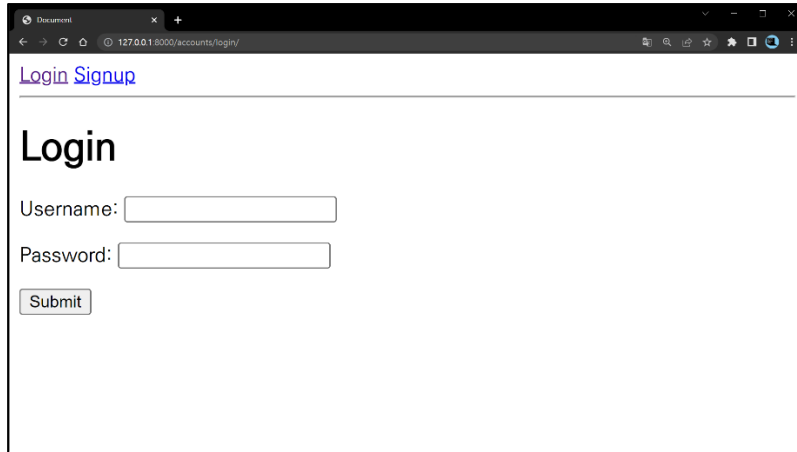
TITLE

DESCRIPTION

[BACK](#)

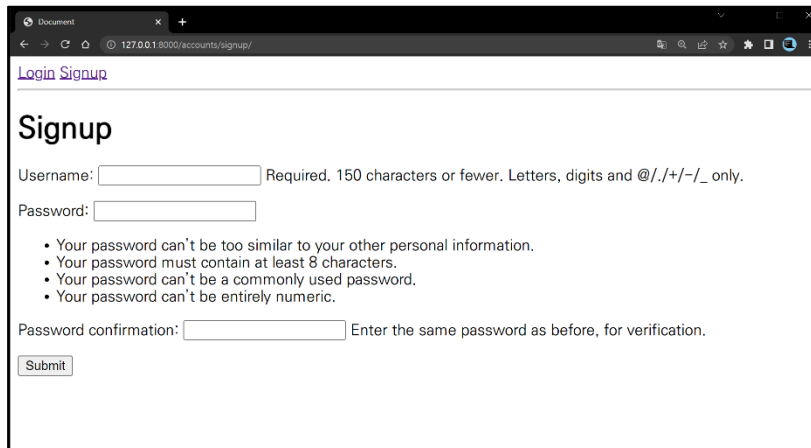
## F. 로그인 페이지 (login.html)

- 로그인 Form을 표시합니다.
- Form에 작성한 정보는 제출(submit)시, 로그인 URL로 요청과 함께 전송됩니다.



## G. 회원가입 페이지 (signup.html)

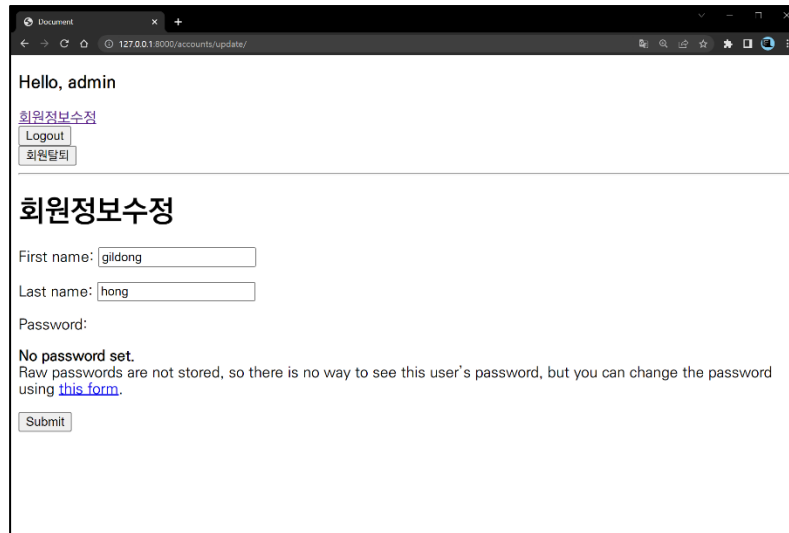
- 회원가입 Form을 표시합니다.
- Form에 작성한 정보는 제출(submit)시, 회원가입 URL로 요청과 함께 전송됩니다.





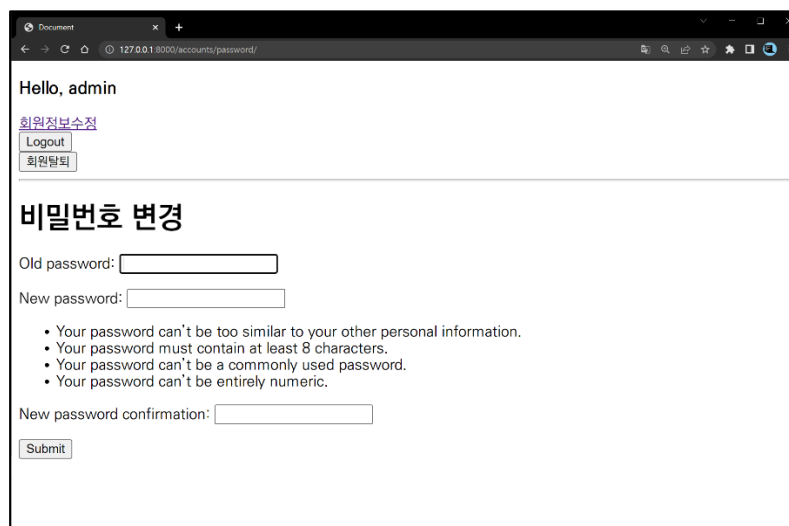
## H. 회원정보 수정 페이지 (update.html)

- 회원정보 수정 Form을 표시합니다.
- Form에 작성한 정보는 제출(submit)시, 회원정보수정 URL로 요청과 함께 전송됩니다.



## I. 비밀번호 변경 페이지 (change\_password.html)

- 비밀번호 변경 Form을 표시합니다.
- Form에 작성한 정보는 제출(submit)시, 비밀번호 변경 URL로 요청과 함께 전송됩니다.



## 10. 필수 요구사항

- A. 내장 User 모델은 반드시 커스텀 User 모델로 대체해서 사용합니다.
- B. 로그인 한 회원만 본인이 작성한 영화 게시글을 작성, 수정 및 삭제할 수 있습니다.
- C. 로그인 한 회원만 본인이 작성한 댓글을 작성 및 삭제할 수 있습니다.
- D. 비밀번호 변경 직후 로그인 상태를 유지해야 합니다.

## 11. 기타

- A. 명세에 작성된 것 이외의 아이디어는 자유롭게 구현합니다.

## 12. 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 pjt07 프로젝트를 생성하고 업로드하여 제출합니다.
- E. 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.