

# 06 PJT

# 관계형 데이터베이스 설계

# INDEX

---

- 관계형 데이터베이스 설계
  - 목표
  - 준비사항
  - 요구사항
  - 제출

# 목표

## 프로젝트 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application 제작
- Django web framework를 사용한 데이터 처리
- Django Model과 ORM에 대한 이해
- Django Authentication System에 대한 이해
- Database many to one relationship(1:N)  
및 many to many relationship(M:N) 에 대한 이해

# 준비사항

## | 개발도구 및 제공사항

- 개발도구
  - Visual Studio Code
  - Google Chrome
  - Django 3.2+
- 제공사항
  - 주어진 스켈레톤코드를 사용합니다.

# 요구사항



## | 공통 요구사항

- 커뮤니티 웹 서비스의 데이터 구성 단계입니다.
- 영화 데이터의 생성, 조회, 수정, 삭제, **좋아요**가 가능한 애플리케이션을 완성합니다.
- 로그인, 로그아웃, 회원가입, 회원탈퇴, 회원정보수정, 비밀번호변경, **팔로우**가 가능한 애플리케이션을 완성합니다.
- 앱 이름은 **movies**와 **accounts**로 지정합니다.
- .gitignore 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.

## | Model

- 정의할 모델 클래스 목록
  - A. Movie
  - B. Comment
  - C. User

## A. Movie

- 정의할 모델 클래스의 이름은 Movie이며, 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. Comment

- 정의할 모델 클래스의 이름은 Comment이며, 다음과 같은 정보를 저장합니다.

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

## C. User

- 정의할 모델 클래스의 이름은 User이며, AbstractUser 모델 클래스를 상속받는 커스텀 모델을 사용합니다.

## | Model

- 생성되는 중계테이블 목록
  - A. movie\_like\_users
  - B. user\_followings

## A. movie\_like\_users

- 생성된 테이블의 이름은 movie\_like\_users이며, 다음과 같은 정보를 저장합니다.

필드명	데이터 유형	역할
movie_id	integer	외래 키(Movie 클래스 참조)
user_id	integer	외래 키(User 클래스 참조)

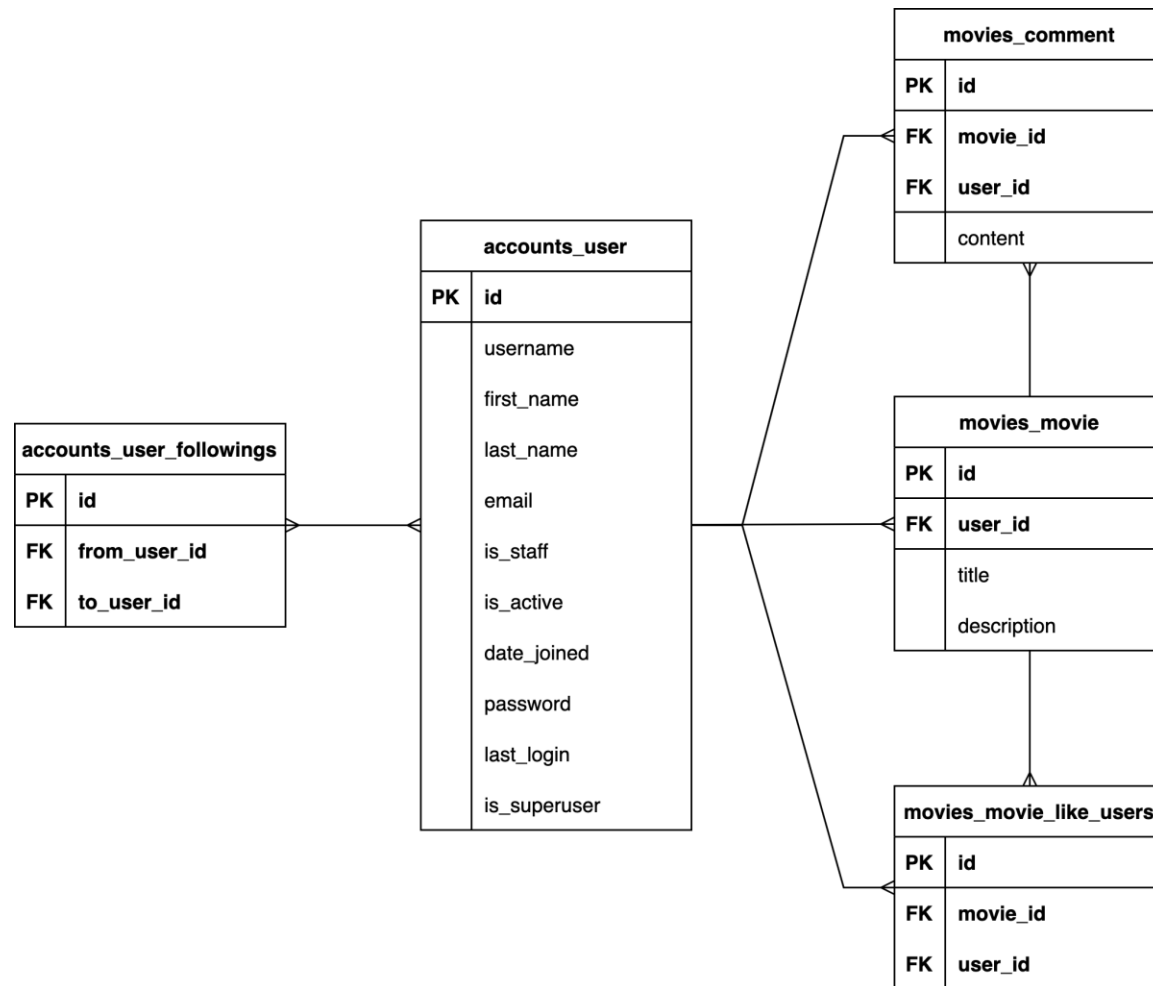
## B. user\_followings

- 생성된 테이블의 이름은 user\_followings이며, 다음과 같은 정보를 저장합니다.

필드명	데이터 유형	역할
user_id	integer	외래 키(User 클래스 참조)
user_id	integer	외래 키(User 클래스 참조)



## ERD (Entity-Relationship Diagram)



## URL (1/2)

- movies 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/movies/	전체 영화 목록 페이지 조회
/movies/create/	새로운 영화 생성 페이지 조회 & 단일 영화 데이터 저장
/movies/<pk>/	단일 영화 상세 페이지 조회
/movies/<pk>/update/	기존 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
/movies/<pk>/likes/	단일 영화 데이터 좋아요 및 좋아요 취소

## URL (2/2)

- accounts 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/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)
/accounts/profile/<username>/	단일 회원 프로필 조회
/accounts/<username>/follow/	단일 회원 팔로우 및 팔로우 취소

## View (1/2)

- movies 앱은 다음 역할을 가지는 view 함수를 가집니다.

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
likes	단일 영화 좋아요(이미 좋아요한 경우 좋아요 취소) 및 index.html 리다이렉트	POST

## View (2/2)

- accounts 앱은 다음 역할을 가지는 view 함수를 가집니다.

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 후 index.html 리다이렉트	GET & POST
update	수정 대상 회원 데이터 조회 및 update.html 렌더링 유효성 검증 및 회원 데이터 수정 후 index.html 리다이렉트	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
profile	단일 회원 데이터 및 작성한 영화, 댓글, 팔로우 수, 팔로잉 수 조회 및 profile.html 렌더링	GET
follow	단일 회원 팔로우(이미 팔로우한 경우 팔로우 취소) 및 profile.html 리다이렉트	POST

## | Admin

- 모델 Movie, Comment, User를 Admin site에 등록합니다.
- Admin site에서 데이터의 생성, 조회, 수정, 삭제가 가능해야 합니다.

## | 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 Form과 ModelForm 그리고 커스텀 ModelForm을 사용합니다.

## | Template

- 공유 템플릿 파일
  - A. base.html
- movies 앱은 다음과 같은 템플릿 파일을 가집니다.
  - B. index.html
  - C. detail.html
  - D. create.html
  - E. update.html



## Template

- accounts 앱은 다음과 같은 템플릿 파일을 가집니다.

F. login.html

G. signup.html

H. update.html

I. change\_password.html

J. profile.html

## A. base.html (1/3)

- “공통 부모 템플릿”
  - 모든 템플릿 파일은 base.html을 상속받아 사용합니다.
  - nav 태그를 사용한 상단 네비게이션 바가 있습니다.
  - 네비게이션 바는 회원의 로그인, 비로그인 상태에 따라 다른 링크를 출력합니다.

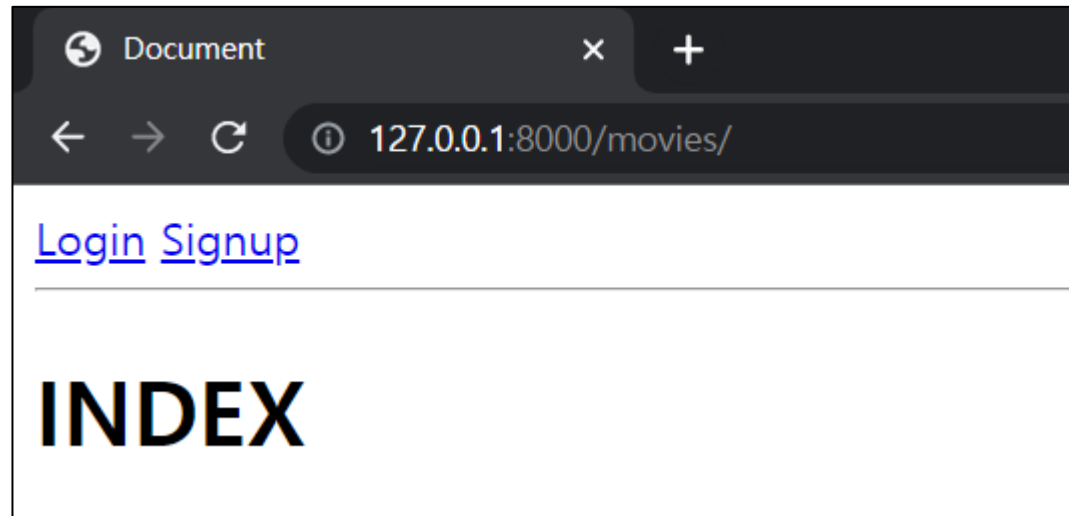
## A. base.html (2/3)

- 로그인 상태
  - A. 회원정보수정
  - B. 로그아웃
  - C. 회원탈퇴
  - D. 내 프로필



## A. base.html (3/3)

- 비로그인 상태
  - A. 로그인
  - B. 회원가입

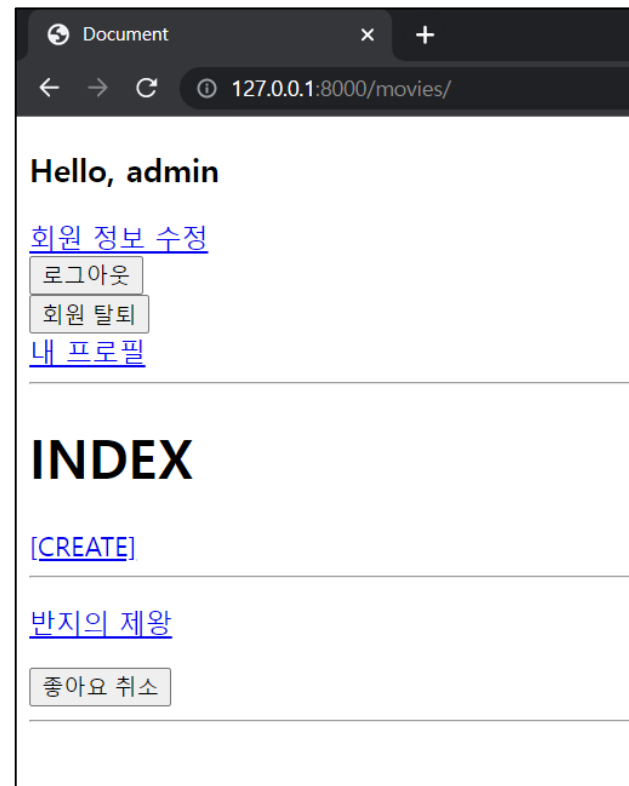
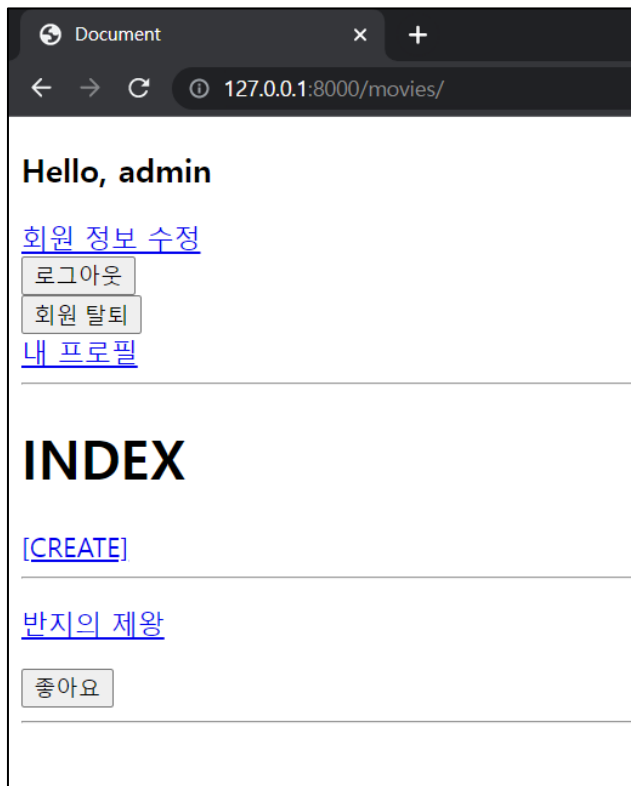


## B. index.html (1/2)

- “전체 영화 목록 조회 페이지”
  - 데이터베이스에 존재하는 모든 영화의 목록을 표시합니다.
  - 적절한 HTML 요소를 사용하여 영화 제목을 표시하며,  
제목 클릭 시 해당 영화의 상세 조회 페이지(detail.html)로 이동합니다.
  - 적절한 HTML 요소를 사용하여 좋아요 기능을 구현하며  
좋아요를 클릭 시 좋아요 취소로, 좋아요 취소 클릭 시 좋아요로 텍스트가 변합니다.

## B. index.html (2/2)

- “전체 영화 목록 조회 페이지”



## C. detail.html (1/2)

- “영화 상세 정보 페이지”
  - 특정 영화의 상세 정보를 표시합니다.
  - 댓글을 작성할 수 있는 form을 표시합니다.
  - 댓글 삭제 버튼은 댓글 작성자에게만 표시합니다.
  - 해당 영화의 수정 및 삭제 버튼을 표시합니다.
  - 전체 영화 목록 조회 페이지(index.html)로 이동하는 링크를 표시합니다.

## C. detail.html (2/3)

- “영화 상세 정보 페이지”





## C. detail.html (3/3)

- “영화 상세 정보 페이지”



## | D. create.html (1/2)

- “영화 생성 페이지”
  - 특정 영화를 생성하기 위한 HTML form 요소를 표시합니다.
  - 표시되는 form은 Movie 모델 클래스에 기반한 ModelForm이어야 합니다.
  - 작성한 정보는 제출(submit)시 단일 영화 데이터를 저장하는 URL로 요청과 함께 전송됩니다.
  - 전체 영화 목록 조회 페이지(index.html)로 이동하는 링크를 표시합니다.

## D. create.html (2/2)

- “영화 생성 페이지”

Document x +

← → ↻ 127.0.0.1:8000/movies/create/

Hello, admin

[회원 정보 수정](#)

[로그아웃](#)

[회원 탈퇴](#)

[내 프로필](#)

---

## CREATE

Title:

Description:

[BACK](#)

## | E. update.html (1/2)

- “영화 수정 페이지”
  - 특정 영화를 수정하기 위한 HTML form 요소를 표시합니다.
  - 표시되는 form은 Movie 모델 클래스에 기반한 ModelForm이어야 합니다.
  - HTML input 요소에는 기존 데이터를 출력합니다.
  - Reset 버튼은 사용자의 모든 입력을 초기 값으로 재설정 합니다.
  - 작성한 정보는 제출(submit)시 단일 영화 데이터를 수정하는 URL로 요청과 함께 전송됩니다.
  - 영화 상세 정보 페이지(detail.html)로 이동하는 링크를 표시합니다.

## E. update.html (2/2)

- “영화 수정 페이지”

Document x +

← → ↻ 127.0.0.1:8000/movies/1/update/

Hello, admin

[회원 정보 수정](#)

로그아웃

회원 탈퇴

[내 프로필](#)

---

## UPDATE

---

TITLE 반지의 제왕

DESCRIPTION 반지의 제왕 3편

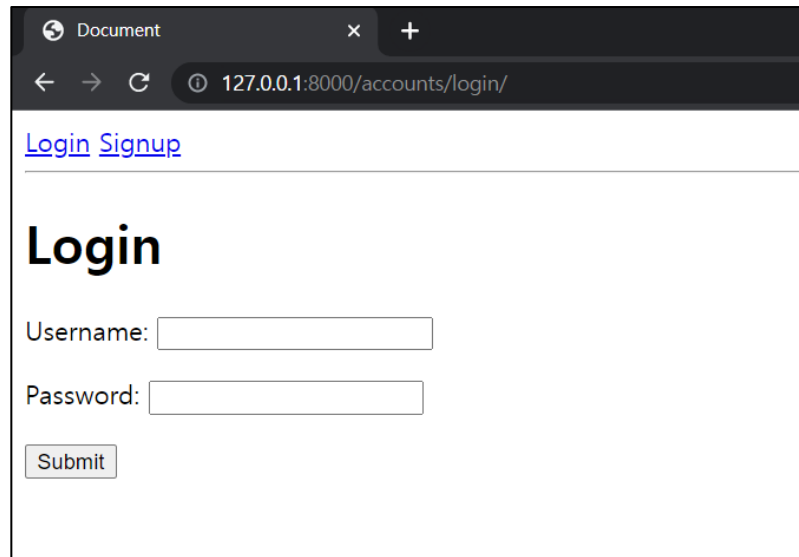
Reset Submit

---

[BACK](#)

## F. login.html

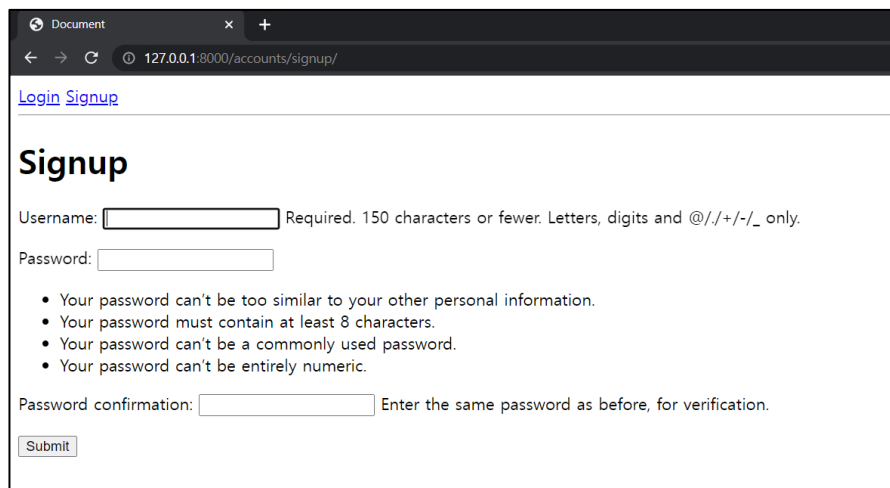
- “로그인 페이지”
  - 로그인을 위한 HTML form 요소를 표시합니다.
  - 작성한 정보는 제출(submit)시 로그인 URL로 요청과 함께 전송됩니다.



The screenshot shows a web browser window with a single tab titled 'Document'. The address bar displays the URL '127.0.0.1:8000/accounts/login/'. Below the address bar, there are two links: 'Login' and 'Signup'. The main content area features a heading 'Login' in a large, bold font. Underneath the heading, there are two input fields: 'Username:' followed by a text input box, and 'Password:' followed by a text input box. At the bottom of the form, there is a 'Submit' button.

## G. signup.html

- “회원가입 페이지”
  - 회원가입을 위한 HTML form 요소를 표시합니다.
  - 작성한 정보는 제출(submit)시 회원가입 URL로 요청과 함께 전송됩니다.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/accounts/signup/". The page has a header with links for "Login" and "Signup". The main content area is titled "Signup" and contains a form with the following fields and instructions:

- Username:** A text input field followed by the instruction: "Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only."
- Password:** A password input field.
- Password requirements:** A bulleted list of rules:
  - Your password can't be too similar to your other personal information.
  - Your password must contain at least 8 characters.
  - Your password can't be a commonly used password.
  - Your password can't be entirely numeric.
- Password confirmation:** A text input field followed by the instruction: "Enter the same password as before, for verification."
- Submit:** A button at the bottom of the form.

## H. update.html

- “회원정보수정 페이지”
  - 회원정보수정을 위한 HTML form 요소를 표시합니다.
  - 작성한 정보는 제출(submit)시 회원정보수정 URL로 요청과 함께 전송됩니다.

Document

127.0.0.1:8000/accounts/update/

Hello, admin

[회원 정보 수정](#)

[로그아웃](#)

[회원 탈퇴](#)

[내 프로필](#)

---

### 회원정보수정

First name:

Last name:

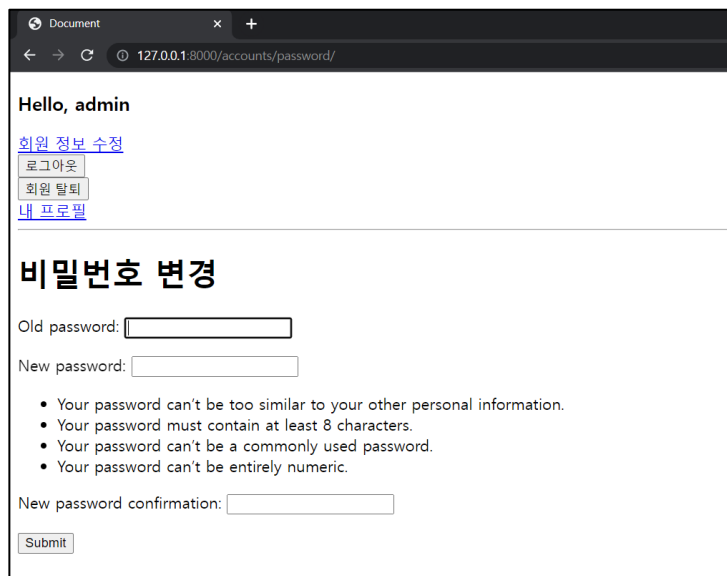
Password:

**No password set.**  
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using [this form](#).



## I. change\_password.html

- “비밀번호변경 페이지”
  - 비밀번호변경 을 위한 HTML form 요소를 표시합니다.
  - 작성한 정보는 제출(submit)시 비밀번호변경 URL로 요청과 함께 전송됩니다.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/accounts/password/". The page content includes a greeting "Hello, admin", a list of links ("회원 정보 수정", "로그아웃", "회원 탈퇴", "내 프로필"), and a section titled "비밀번호 변경". This section contains input fields for "Old password:", "New password:", and "New password confirmation:", followed by a list of password requirements and a "Submit" button.

Document x +  
← → ↻ 127.0.0.1:8000/accounts/password/  
Hello, admin  
[회원 정보 수정](#)  
[로그아웃](#)  
[회원 탈퇴](#)  
[내 프로필](#)  
  
**비밀번호 변경**  
Old password:   
New password:   

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

New password confirmation:

## J. 추가 필수 요구사항

- 로그인 한 회원만 영화 정보를 생성, 수정, 삭제할 수 있습니다.
- 본인이 작성한 영화 정보에 대해서만 수정, 삭제할 수 있습니다.
- 로그인 한 회원만 댓글을 생성, 삭제할 수 있습니다.
- 본인이 작성한 댓글에 대해서만 삭제할 수 있습니다.
- 비밀번호 변경 직후 로그인 상태를 유지해야 합니다.
- 본인을 팔로우할 수 없습니다.

## K. 추가과제

- 팔로우, 팔로잉, 좋아요 수를 적절한 페이지에서 출력합니다.
  - 참고) [DTL filter](#), [QuerySet API](#)
- 싫어요 기능을 구현합니다.
  - 참고) 좋아요 기능과의 차이점은 무엇일까요?
- 대댓글 기능을 구현합니다.
  - 참고) 뒷 페이지에 계속
- CSS, bootstrap등을 활용하여 페이지를 꾸밉니다.

## K. 추가과제 - 대댓글

- 아래의 참고사항은 예시이며, 다른 방식으로 구현해도 좋습니다.
- 대댓글은 댓글이며, 다른 댓글을 참조합니다.
- 즉, 댓글과 대댓글의 차이점은 참조하는 댓글의 유무입니다.
  - 참조하는 댓글이 있다면 대댓글  
참조하는 댓글이 없다면 그냥 댓글
    - 참고) [null=True](#)
- 참고) [retrieving-specific-objects](#)

# 제출

## 제출 시 주의사항

- 제출기한은 금일 18시까지 입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjт'로 지정합니다. (ex. **01\_pjt**)
- 반드시 각 반 담당 교수님을 Maintainer로 설정해야 합니다.