

Django_PJT3

목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web Application 제작
- Python Web Framework를 통한 데이터 조작
- django Modelform을 활용한 HTML과 사용자 요청 데이터 관리
- Authentication 개념의 이해와 활용
- Database 1:N 의 이해와 데이터 관계 설정

준비 사항

- Python/HTML/CSS 코드 작성 및 확인을 위한 환경 구성
- 개발 도구
 - Google Chrome Browser
 - Python 3.7+
 - Django 2.1.15

구현 항목

01. 프로젝트 구조

- `django_pjt3/` 는 `startproject` 명령어로 생성되는 project 디렉토리입니다.
- `accounts/` 와 `community/` 는 `startapp` 명령어로 생성되는 application 디렉토리입니다.

```
django_pjt3/  
  __init__.py  
  settings.py  
  urls.py  
  wsgi.py  
  
accounts/  
  migrations/  
  templates/  
    accounts/  
      signup.html (필수)  
      login.html (필수)  
  __init__.py  
  admin.py  
  app.py  
  models.py  
  test.py  
  urls.py (필수)  
  views.py
```

```

community/
  migrations/
  templates/
    community/
      form.html (필수)
      review_list.html (필수)
      review_detail.html (필수)
  __init__.py
  admin.py
  app.py
  forms.py (필수)
  models.py
  test.py
  urls.py (필수)
  views.py

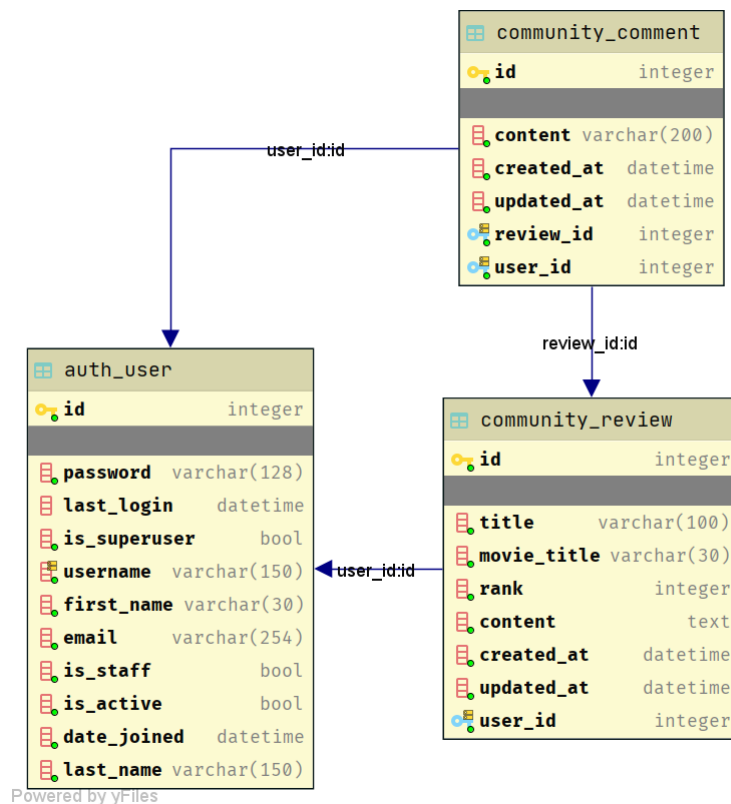
templates/
  base.html (필수)

manage.py

```

02. Model

프로젝트 DB 에서 주요 모델간의 ERD(Entity Relation Diagram)는 아래와 같습니다.



직접 정의할 모델 클래스의 이름은 **Review**, **Comment**이며, 아래와 같은 정보를 저장합니다. 사용자 (**User**) 모델은 django 에서 기본 제공하는 모델을 사용합니다.

Review

필드명	자료형	설명
<code>user</code>	Integer(ForeignKey)	리뷰를 작성한 사용자의 PK
<code>title</code>	String(100자 이하)	리뷰의 제목
<code>movie_title</code>	String(30자 이하)	리뷰하는 영화의 제목
<code>rank</code>	Integer	리뷰의 평점
<code>content</code>	Text	리뷰의 내용
<code>created_at</code>	DateTime	리뷰가 최초 작성된 시간(자동 저장)
<code>updated_at</code>	DateTime	리뷰가 최종 수정된 시간(자동 저장)

Comment

필드명	자료형	설명
<code>user</code>	Integer(ForeignKey)	댓글을 작성한 사용자의 PK
<code>review</code>	Integer(ForeignKey)	댓글이 달린 리뷰의 PK
<code>content</code>	String(200자 이하)	댓글의 내용
<code>created_at</code>	DateTime	댓글이 최초 작성된 시간(자동 저장)
<code>updated_at</code>	DateTime	댓글이 최종 수정된 시간(자동 저장)

03. Form

위에서 정의한 모델들의 데이터 검증, 저장, 에러메시지, HTML을 모두 관리하기 위해 `ModelForm`을 사용합니다.

ReviewForm

해당 `ModelForm`은 아래의 필드를 검증 및 HTML로 표시합니다.

- `title`
- `movie_title`
- `rank`
- `content`

CommentForm

해당 `ModelForm`은 아래의 필드를 검증 및 HTML로 표시합니다.

- `content`

04. Admin

위에서 정의한 모델 `Review` 와 `Comment` 는 관리자 페이지에서 데이터를 **생성, 조회, 수정, 삭제** 가능해야 합니다. (Admin 페이지에서 테스트 목적으로 데이터를 생성, 조회, 수정, 삭제 할 수 있습니다.)

05. URL

URL 은 app의 이름과 각 URL 패턴별 이름을 설정합니다. View와 Template에서는 **URL을 하드타이핑** 하지 않고 설정한 이름을 사용합니다.

accounts app

accounts app의 모든 URL 패턴은 `/accounts` 로 시작합니다.

HTTP Verb	URL 패턴	설명
<code>GET & POST</code>	<code>/signup/</code>	신규 사용자 생성(회원가입) 및 form 표시
<code>GET & POST</code>	<code>/login/</code>	기존 사용자 인증(로그인) 및 form 표시
<code>GET / POST</code> (선택)	<code>/logout/</code>	인증된 사용자 인증 해제(로그아웃)

community app

community app의 모든 URL 패턴은 `/community` 로 시작합니다.

HTTP Verb	URL 패턴	설명
<code>GET & POST</code>	<code>/create/</code>	신규 리뷰 생성 및 form 표시
<code>GET</code>	<code>/</code>	전체 리뷰 목록 조회 페이지
<code>GET</code>	<code>/<review_pk>/</code>	단일 리뷰 조회와 댓글 전체 조회 및 댓글 작성 form 표시
<code>GET & POST</code>	<code>/<review_pk>/update/</code>	기존 리뷰 수정 및 form 표시
<code>GET / POST</code> (선택)	<code>/<review_pk>/delete/</code>	기존 리뷰 삭제
<code>POST</code>	<code>/<review_pk>/comments/</code>	신규 댓글 생성
<code>GET / POST</code> (선택)	<code>/<review_pk>/comments/<comment_pk>/delete/</code>	기존 댓글 삭제

06. View & Template

필수 조건

- 공유 Template
 - 모든 HTML 파일은 `base.html` 을 확장(**extends**)하여 사용합니다.
 - `base.html` 은 모든 페이지가 공유하는 상단 네비게이션 바를 표시합니다.
 - 네비게이션 바
 - 사용자 인증 여부와 관계 없이 전체 리뷰 목록 조회 페이지와 새로운 리뷰 작성 페이지로 이동할 수 있는 링크를 표시합니다.
 - 사용자가 인증되어 있지 않다면, 로그인 페이지와 회원 가입 페이지로 이동할 수 있는 링크를 표시합니다.
 - 사용자가 인증되어 있다면, 로그아웃할 수 있는 링크를 표시합니다.

accounts app

- 신규 사용자 생성(회원가입)
 - 이미 인증되어있는 사용자는 전체 리뷰 목록 페이지로 **redirect** 합니다.
 - django 에서 기본적으로 사용하는 `UserCreationForm` 을 사용합니다.
 - HTTP method **GET**
 - 사용자에게 응답으로 제공할 HTML 은 `accounts/signup.html` 입니다.
 - 회원가입을 할 수 있는 form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드명	HTML Tag	Type
username	input	text
password	input	password
password_confirmation	input	password

- form 에 작성한 정보는 제출(submit)시, **POST**방식으로 요청과 데이터를 적절한 URL로 전송합니다.
- HTTP method **POST**
 - 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고
 1. 사용자를 인증(로그인)해준 이후
 2. 전체 리뷰 목록 페이지로 **redirect** 합니다.
 - 데이터가 유효하지 않다면 에러 메시지를 포함하여, 데이터를 작성하는 **form** 을 사용자 화면에 표시합니다.
- 기존 사용자 인증(로그인)
 - 이미 인증되어있는 사용자는 전체 리뷰 목록 페이지로 **redirect** 합니다.
 - django 에서 기본적으로 사용하는 `AuthenticationForm` 을 사용합니다.
 - HTTP method **GET**

- 사용자에게 응답으로 제공할 HTML 은 `accounts/login.html` 입니다.
- 로그인할 수 있는 form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.

필드명	HTML Tag	Type
username	input	text
password	input	password

- form 에 작성한 정보는 제출(submit)시, **POST**방식으로 요청과 데이터를 적절한 URL로 전송합니다.
- HTTP method **POST**
 - 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고
 1. 사용자를 인증(로그인)해준 이후
 2. 로그인 하기 전, 기존 URL이 함께 넘어왔다면 해당 URL로,
 3. 넘어오지 않았다면 전체 리뷰 목록 페이지로 **redirect** 합니다.
 - 데이터가 유효하지 않다면 에러 메시지를 포함하여, 데이터를 작성하는 form 을 사용자 화면에 표시합니다.
- 인증된 사용자 인증 해제(로그아웃)
 - HTTP method 는 GET 과 POST 중 선택하여 구성합니다.
 - 사용자의 인증을 해제(로그아웃)하고 **전체 리뷰 목록 페이지로 redirect** 합니다.

community app

- 신규 리뷰 생성
 - 인증되지 않은 사용자는 사용자 인증(Login) 페이지로 **redirect** 합니다.
 - HTTP method **GET**
 - 사용자에게 응답으로 제공할 HTML은 `form.html` 입니다.
 - 리뷰를 작성할 수 있는 form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다.
- | 필드명 | HTML Tag | Type |
|-------------|----------|--------|
| title | input | text |
| movie_title | input | text |
| rank | input | number |
| content | textarea | 없음 |
- form에 작성한 정보는 제출(submit)시, **POST**방식으로 요청과 데이터를 적절한 URL로 전송합니다.
 - HTTP method **POST**
 - 기존에 정의한 `ReviewForm` 을 사용하여 데이터를 검증합니다.
 - 데이터가 유효하다면 요청과 함께 전송된 데이터 및 작성자에 대한 정보를 데이터베이스에 저장하고 **단일 리뷰 상세 조회 페이지로 redirect** 합니다.
 - 데이터가 유효하지 않다면 에러 메시지를 포함하여, 데이터를 작성하는 form 을 사용자 화면에 표시합니다.
 - 전체 리뷰 목록 조회

- 데이터베이스에 존재하는 모든 리뷰의 목록을 HTML에 표시합니다.
 - 사용자에게 응답으로 제공할 HTML은 `review_list.html` 입니다.
 - `review_list.html` 은 적절한 HTML 요소를 사용하여 `title` 을 포함한 게시글의 데이터를 자유롭게 표시하며, `title` 을 클릭 시 해당 리뷰의 **상세 조회 페이지**로 이동합니다.
 - **단일 리뷰 상세 조회와 댓글 전체 조회 및 댓글 작성 form 표시**
 - URL 을 통해 함께 전달된 `review_pk` 를 `id(pk)`로 갖는 리뷰의 상세 HTML에 표시해야 합니다.
 - `review_pk` 를 `id(pk)` 로 갖는 리뷰 데이터가 없을 경우, **404 에러 페이지**를 리턴합니다.
 - 사용자에게 응답으로 제공할 HTML은 `review_detail.html` 입니다.
 - **사용자 인증여부에 따라 HTML을 다르게 표시합니다.**
 - 모든 사용자에게 적절한 HTML 요소를 사용하여, 조회하는 리뷰의 `title`, `movie_title`, `rank`, `content`, `created_at`, `updated_at` 을 표시합니다.
 - 모든 사용자에게 적절한 HTML 요소를 사용하여, 리뷰에 달린 모든 댓글의 `content` 와 해당 리뷰를 작성한 사용자의 `username` 을 표시합니다.
 - 인증된 사용자에게만 현재 조회중인 게시글에 댓글을 작성할 수 있는 form을 표시합니다.
 - 리뷰 작성자에게만 현재 조회중인 게시글을 수정, 삭제 할 수 있는 UI를 표시합니다.
 - 댓글 작성자에게만 본인이 작성한 댓글을 삭제할 수 있는 UI를 표시합니다.
 - **기존 리뷰 수정**
 - 인증되지 않은 사용자는 사용자 인증(Login) 페이지로 **redirect** 합니다.
 - 요청을 보낸 사용자와 수정하고자 하는 리뷰의 작성자가 같지 않으면, 단일 리뷰 상세 조회 페이지로 **redirect** 합니다.
 - URL 을 통해 함께 전달된 `review_pk` 를 `id(pk)` 로 갖는 리뷰를 수정합니다.
 - `review_pk` 를 `id(pk)` 로 갖는 리뷰 데이터가 없을 경우, **404 에러 페이지**를 리턴합니다.
 - HTTP method **GET**
 - 사용자에게 응답으로 제공할 HTML은 `form.html` 입니다.
 - 리뷰를 수정할 수 있는 form을 표시하며, 다음과 같은 input 요소들을 포함해야 합니다. 기존의 리뷰 데이터도 함께 제공해야 합니다.
- | 필드명 | HTML Tag | Type |
|-------------|----------|--------|
| title | input | text |
| movie_title | input | text |
| rank | input | number |
| content | textarea | 없음 |
- form에 작성한 정보는 제출(submit) 시, **POST** 방식으로 요청과 데이터를 적절한 URL 로 전송합니다.
 - HTTP method **POST**
 - 기존에 정의한 `ReviewForm` 을 사용하여 데이터를 검증합니다.
 - 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터베이스에 저장하고 **단일 리뷰 상세 조회 페이지**로 **redirect**합니다.
 - 데이터가 유효하지 않다면 에러 메시지를 포함하여, 데이터를 작성하는 **form** 을 사용자 화면에 표시합니다.
 - **기존 리뷰 삭제**
 - HTTP method 는 GET 과 POST 중 선택하여 구성합니다.

- 인증되지 않은 사용자는 사용자 인증(Login) 페이지로 **redirect** 합니다.
- 요청을 보낸 사용자와 삭제하고자 하는 리뷰의 작성자가 같지 않으면, 단일 리뷰 상세 조회 페이지로 **redirect** 합니다.
- URL 을 통해 함께 전달된 `review_pk` 를 `id(pk)` 로 갖는 리뷰를 삭제합니다.
- `review_pk` 를 `id(pk)` 로 갖는 리뷰 데이터가 없을 경우, **404 에러 페이지**를 리턴합니다.
- 삭제가 완료되면 **전체 리뷰 목록 조회 페이지**로 **redirect** 합니다.
- 신규 댓글 생성
 - 인증되지 않은 사용자는 사용자 인증(Login) 페이지로 **redirect** 합니다.
 - 신규 댓글 생성용 form은 **단일 리뷰 상세 조회페이지**에서 표시합니다.
 - 기존에 정의한 `CommentForm` 을 사용하여 데이터를 검증합니다.
 - 데이터가 유효하다면 요청과 함께 전송된 데이터를 데이터 베이스에 저장하고 **단일 리뷰 상세 조회 페이지**로 **redirect**합니다.
 - 데이터가 유효하지 않다면 **단일 리뷰 상세 조회페이지**로 **redirect** 합니다.
- 기존 댓글 삭제
 - 인증되지 않은 사용자는 사용자 인증(Login) 페이지로 **redirect** 합니다.
 - 기존 댓글 삭제용 UI는 **단일 리뷰 상세 조회페이지**에서 표시합니다.
 - URL 을 통해 함께 전달된 `comment_pk` 를 `id(pk)` 로 갖는 댓글을 삭제합니다.
 - `comment_pk` 를 `id(pk)` 로 갖는 댓글 데이터가 없을 경우, **404 에러 페이지**를 리턴합니다.
 - 요청을 보낸 사용자와 삭제하고자 하는 댓글의 작성자가 같지 않으면, 단일 리뷰 상세 조회 페이지로 **redirect** 합니다.
 - 삭제가 완료되면 **단일 리뷰 상세 조회 페이지**로 **redirect** 합니다.

선택 조건

- Bootstrap을 활용하여 자유롭게 추가적으로 스타일링 합니다.
- message framework 를 활용하여 상황에 따라 다른 메시지를 추가로 표시합니다.

제출 안내

1. 제출 기한은 **20.04.22(수) 23:59** 까지 입니다. 제출기한을 엄수하여 주시기 바랍니다.
2. 반드시 `README.md` 파일에 구현 과정 및 페어프로그래밍에서 느낀점을 기록하여 제출해야 합니다.
3. <https://lab.ssafy.com> 에 `django_pjt3` 프로젝트를 생성하시고 해당 프로젝트에 업로드하여 제출합니다.
4. 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.
5. 결과물은 아래와 같은 파일/폴더 구조로 이루어져 있어야 합니다.

```
django_pjt3/
  README.md
  django_pjt3/
    accounts/
    community/
    templates/
    db.sqlite3
    manage.py
```