

# Django\_PJT2

---

## 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
- Object Relational Mapping에 대한 이해
- HTTP 요청 메소드와 응답 상태코드에 대한 이해

## 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_pjt2/` 는 `startproject` 명령어로 생성되는 project 디렉토리입니다.
- `community/` 는 `startapp` 명령어로 생성되는 application 디렉토리입니다.

```
django_pjt2/  
  __init__.py  
  settings.py  
  urls.py  
  wsgi.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

정의할 모델 클래스의 이름은 `Review`이며, 다음과 같은 정보를 저장합니다.

### Review

필드명	자료형	설명
<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)

---

## 03. Form

위에서 정의한 모델 `Review`의 데이터 검증, 저장, 에러메시지, HTML을 모두 관리하기 위해 `ModelForm`을 사용합니다. 해당 `ModelForm`은 아래의 필드를 검증 및 HTML로 표시합니다.

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

## 04. Admin

위에서 정의한 모델 `Review`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 하드타이핑하지 않고 설정한 이름을 사용합니다.**

HTTP Verb	URL 패턴	설명
GET & POST	/community/create/	신규 리뷰 생성 및 Form 표시
GET	/community/	전체 리뷰 목록 조회 페이지
GET	/community/<review_pk>/	단일 리뷰 상세 조회 페이지
GET & POST	/community/<review_pk>/update/	기존 리뷰 수정 및 Form 표시
GET / POST (선택)	/community/<review_pk>/delete/	기존 리뷰 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 바**를 표시합니다.
  - 네비게이션 바는 **전체 리뷰 목록 조회 페이지**와 **새로운 리뷰 작성 페이지**로 이동할 수 있는 링크를 포함합니다.
- 새로운 리뷰 생성
  - 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 URL 로 전송합니다.
- HTTP method **POST**
  - 기존에 정의한 ModelForm을 사용하여 **데이터를 검증**합니다.
  - **데이터가 유효하다면** 요청과 함께 전송된 데이터를 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 조회

- 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.
- `review_detail.html` 에는 적절한 HTML 요소를 사용하여, 조회하는 리뷰의 `title`, `movie_title`, `rank`, `content`, `created_at`, `updated_at` 을 표시합니다.
- `review_detail.html` 에는 사용자가 현재 조회중인 게시글을 **수정, 삭제** 할 수 있는 UI가 함께 제공되어야 합니다.

- 기존 리뷰 수정

- 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 URL로 전송합니다.
- HTTP method **POST**
  - 기존에 정의한 ModelForm을 사용하여 **데이터를 검증**합니다.
  - **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

- 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.
- HTTP method 는 GET 과 POST 중 선택하여 구성합니다.
- 삭제가 완료되면 **전체 리뷰 목록 조회 페이지**로 **Redirect** 합니다.

- 추가적인 Styling

- Bootstrap을 활용하여 자유롭게 스타일링 합니다.

## 선택 조건

- Bootstrap을 활용하여 스타일링 할 때, CDN 이 아닌 `static/` 폴더를 활용합니다.
- 

## 제출 안내

---

1. 제출 기한은 20.04.08(수) 20:00 까지 입니다. 제출기한을 엄수하여 주시기 바랍니다.
2. 반드시 `README.md` 파일에 각 단계별 상세 구현 과정 및 어려웠던 부분을 기록하여 제출해야 합니다.
3. <https://lab.ssafy.com> 에 `django_pjt2` 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_pjt2/  
  README.md  
  django_pjt2/  
    ...  
  
  community/  
    ...  
  
  templates/  
    ...  
  
  db.sqlite3  
  manage.py
```