

# PJT 05

# 인증 페이지 구현

# INDEX

---

- 개요
- 준비사항
- 요구사항
- 도전 과제
- 제출

# 개요

## 프로젝트 개요

- 커뮤니티 웹 서비스의 데이터 구성 및 인증 시스템 구성 단계
- 영화 데이터의 생성, 조회, 수정, 삭제가 가능한 애플리케이션 완성
- 로그인, 로그아웃, 회원가입, 회원탈퇴, 회원정보수정, 비밀번호 변경이 가능한 애플리케이션 완성

## 프로젝트 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application 제작
- Django Model과 ORM에 대한 이해
- Django Authentication System에 대한 이해

# 준비사항

## | 개발도구

- Visual Studio Code
- Google Chrome
- Django 4.2.x



# 요구사항

## 공통 요구사항

- 명시된 요구사항 이외에는 자유롭게 작성해도 무관
- Bootstrap을 이용하여 자유롭게 스타일링 가능
- `.gitignore` 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않음

## | 필수 요구사항 - 프로젝트 및 앱

- 프로젝트 이름
  - mypjt
- 앱 이름
  - movies
  - accounts

## | 필수 요구사항 - Model Class

- Movie 모델 클래스
  - 영화 제목과 줄거리를 저장할 필드 2개 지정
- User 모델 클래스
  - AbstractUser 모델을 상속받는 커스텀 모델을 사용

## | 필수 요구사항 - Form Class

- **Movie** 모델의 데이터 검증, 저장, 에러메시지 등을 모두 관리하기 위해 적절한 **ModelForm**을 사용
- **User** 모델의 데이터 데이터 검증, 저장, 에러메시지 등을 모두 관리하기 위해 적절한 Built-in **Form**과 **ModelForm** 그리고 필요한 경우 커스텀 한 **ModelForm**을 사용

## 필수 요구사항 - view 함수

- movies 앱의 view 함수

함수명	역할	허용 HTTP Method
index	<ul style="list-style-type: none"><li>• 전체 영화 데이터 조회</li></ul>	GET
create	<ul style="list-style-type: none"><li>• 영화 데이터 작성 폼 출력</li><li>• 유효성 검증 및 영화 데이터 저장</li></ul>	GET & POST
detail	<ul style="list-style-type: none"><li>• 단일 영화 데이터 및 댓글 목록 조회</li><li>• 댓글 작성 폼 출력</li></ul>	GET
update	<ul style="list-style-type: none"><li>• 영화 데이터 수정 페이지 조회</li><li>• 유효성 검증 및 영화 데이터 수정</li></ul>	GET & POST
delete	<ul style="list-style-type: none"><li>• 단일 영화 데이터 삭제</li></ul>	POST

## 필수 요구사항 - view 함수

- accounts 앱의 view 함수

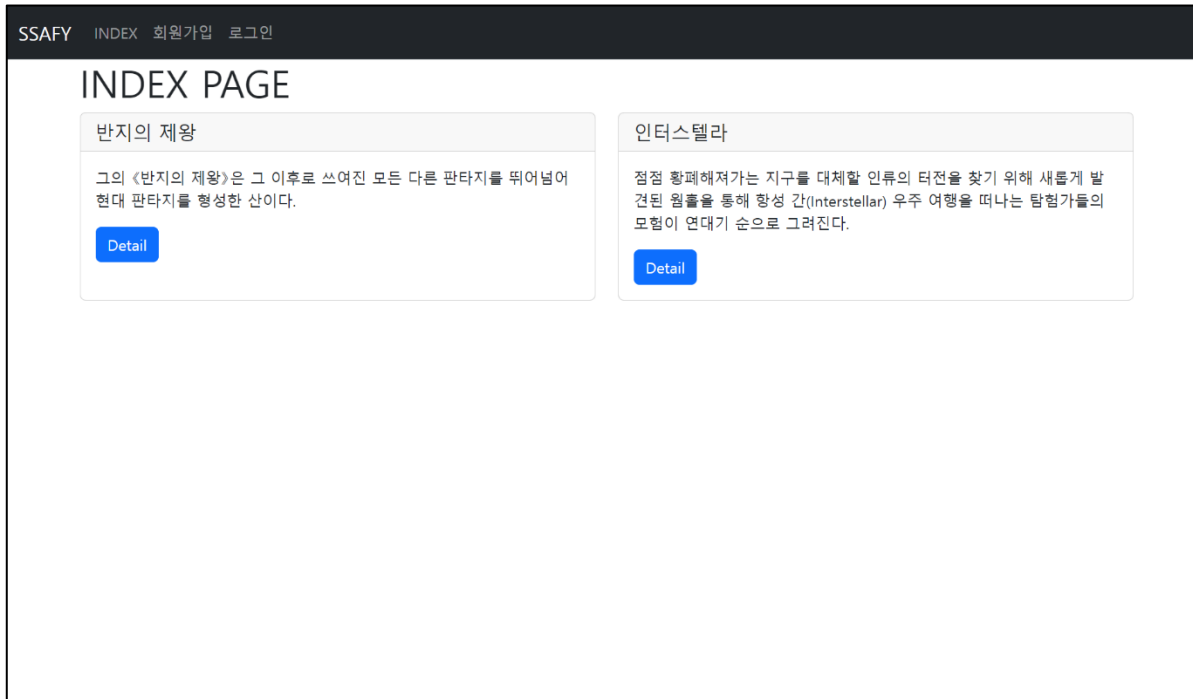
함수명	역할	허용 HTTP Method
login	<ul style="list-style-type: none"><li>로그인 폼 출력</li><li>유효성 검증 및 세션 데이터 저장</li></ul>	GET & POST
logout	<ul style="list-style-type: none"><li>DB와 클라이언트의 쿠키에서 인증된 사용자의 세션 데이터 삭제</li></ul>	POST
signup	<ul style="list-style-type: none"><li>회원가입 폼 출력</li><li>유효성 검증 및 회원 데이터 저장</li></ul>	GET & POST
update	<ul style="list-style-type: none"><li>회원정보수정 폼 출력</li><li>유효성 검증 및 회원 데이터 수정</li></ul>	GET & POST
delete	<ul style="list-style-type: none"><li>단일 회원 데이터 삭제</li></ul>	POST
change_password	<ul style="list-style-type: none"><li>비밀번호 변경 폼 출력</li><li>유효성 검증 및 비밀번호 변경</li></ul>	GET & POST

# 완성 화면 예시

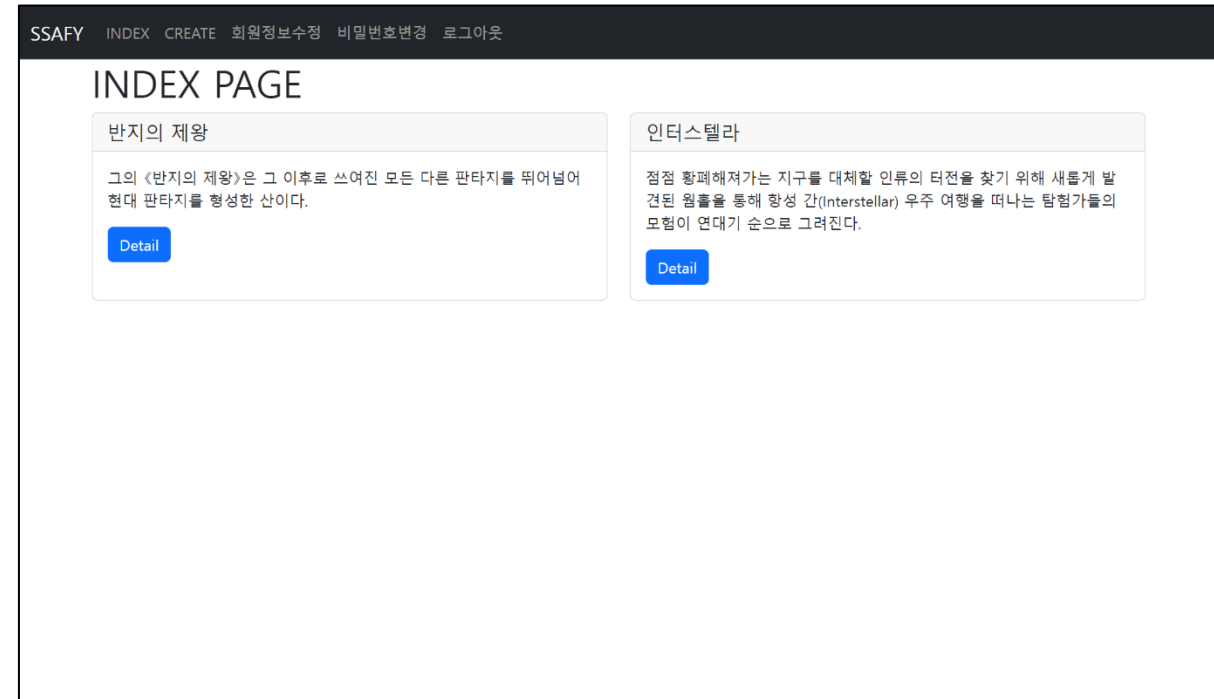


## 완성 화면 예시 (1/8)

- 메인 페이지



로그인 전



로그인 후

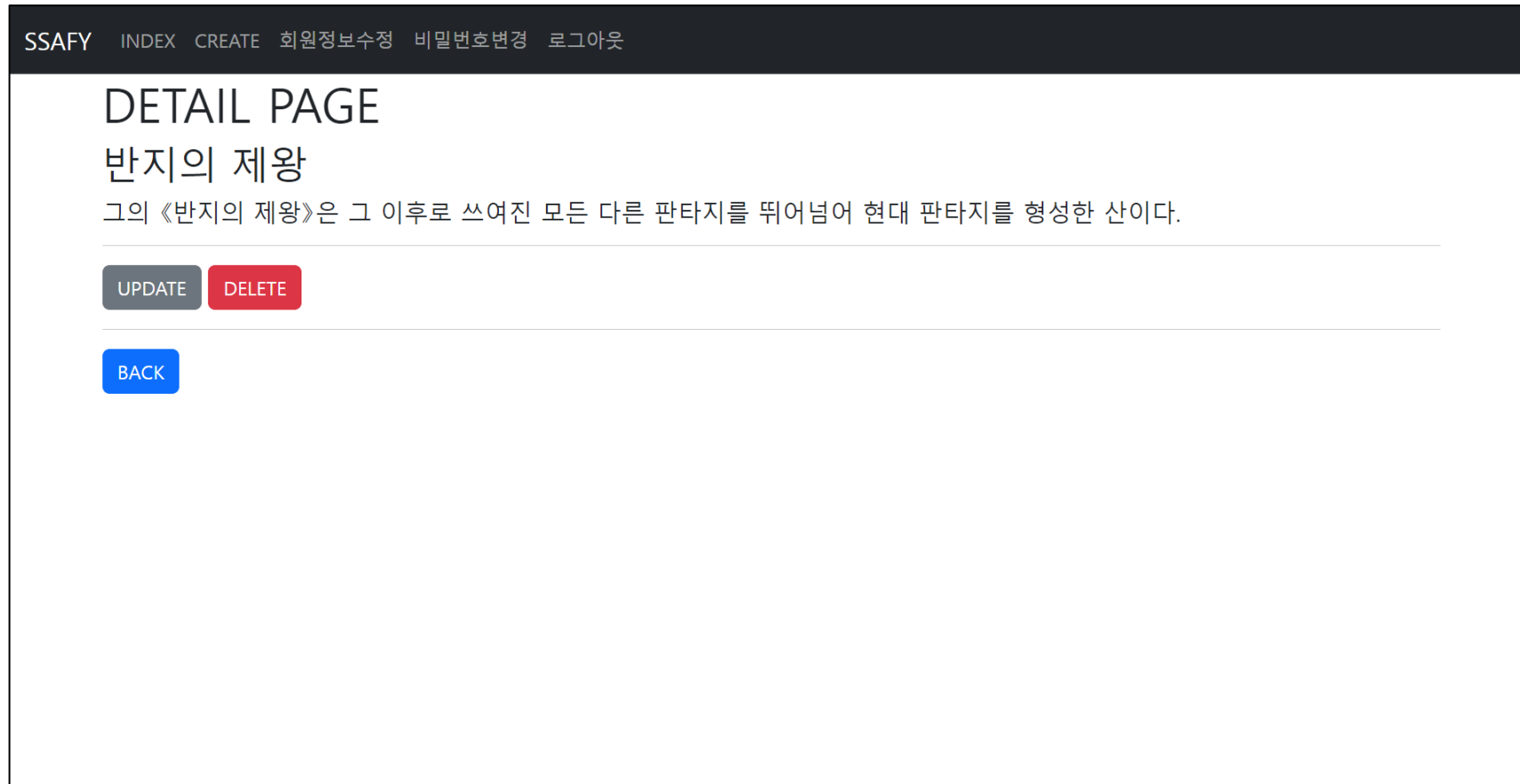
## 완성 화면 예시 (2/8)

- 영화 등록 페이지

The screenshot shows a web interface for creating a new movie entry. At the top, a dark navigation bar contains the text 'SSAFY' followed by links: 'INDEX', 'CREATE', '회원정보수정', '비밀번호변경', and '로그아웃'. Below this, the main content area is titled 'Create'. It features a 'Title:' label next to a text input field. Below the title field is a large, empty rectangular box for the movie description. To the left of this box is the label 'Description:'. At the bottom left of the form, there is a button labeled '제출' (Submit).

## 완성 화면 예시 (3/8)

- 영화 상세 페이지



## 완성 화면 예시 (4/8)

- 영화 수정 페이지

[SSAFY](#) [INDEX](#) [CREATE](#) [회원정보수정](#) [비밀번호변경](#) [로그아웃](#)

### Update

Title:

Description: 

그의 《반지의 제왕》은 그 이후로 쓰여진 모든 다른 판타지를 뛰어넘어 현대 판타지를 형성한 산이다.

## 완성 화면 예시 (5/8)

- 로그인 페이지

SSAFY INDEX 회원가입 로그인

### LOGIN PAGE

사용자 이름:

비밀번호:

## 완성 화면 예시 (6/8)

- 회원가입 페이지

SSAFY INDEX 회원가입 로그인

### SIGNUP PAGE

사용자 이름:  150자 이하 문자, 숫자 그리고 @/./+/\_만 가능합니다.

이메일 주소:

이름:

성:

비밀번호:

- 다른 개인 정보와 유사한 비밀번호는 사용할 수 없습니다.
- 비밀번호는 최소 8자 이상이어야 합니다.
- 통상적으로 자주 사용되는 비밀번호는 사용할 수 없습니다.
- 숫자로만 이루어진 비밀번호는 사용할 수 없습니다.

비밀번호 확인:  확인을 위해 이전과 동일한 비밀번호를 입력하세요.

## 완성 화면 예시 (7/8)

- 회원 정보 수정 페이지
  - Django가 기본적으로 제공하는 비밀번호 변경 링크를 클릭하면 비밀번호 페이지로 이동

SSAFY INDEX CREATE 회원정보수정 비밀번호변경 로그아웃

### UPDATE PAGE

이메일 주소:

이름:

성:

사용자 이름:  150자 이하 문자, 숫자 그리고 @/./+/-/\_만 가능합니다.

비밀번호:

비밀번호가 설정되지 않습니다.  
원본 비밀번호는 저장되지 않으므로, 해당 사용자의 비밀번호를 확인할 수 없습니다. 다만 이 [폼](#)을 사용하여 비밀번호를 변경할 수 있습니다.

## 완성 화면 예시 (8/8)

- 비밀번호 변경 페이지

[SSAFY](#) [INDEX](#) [CREATE](#) [회원정보수정](#) [비밀번호변경](#) [로그아웃](#)

### PASSWORD CHANGE

기존 비밀번호:

새 비밀번호:

- 다른 개인 정보와 유사한 비밀번호는 사용할 수 없습니다.
- 비밀번호는 최소 8자 이상이어야 합니다.
- 통상적으로 자주 사용되는 비밀번호는 사용할 수 없습니다.
- 숫자로만 이루어진 비밀번호는 사용할 수 없습니다.

새 비밀번호 (확인):



# 도전 과제

## 도전 과제 안내

- 생성형 AI 도구를 활용하여 도전과제 요구사항 해결하기
- 생성형 AI를 통해 코드 생성, 아이디어 구상, 문제 해결 방법 탐색 등 다양한 방식으로 활용 가능
- 선택한 생성형 AI 서비스는 자유롭게 결정
- 최종 결과물은 AI 생성 내용을 바탕으로 직접 수정 및 개선하여 적용하기

## AI 활용 주의사항

- AI 도구는 보조 수단으로 활용하되, 능동적인 자세로 학습에 임할 것
- 최종적인 이해와 적용은 자기 주도적 학습을 통해 이루어지며,  
배운 내용을 스스로 기록하고 정리하며 학습 효과를 높일 것

# 개인 프로필 페이지

## | 사용자 별 개인 프로필 페이지 만들기

- User 모델과 OneToOne 관계를 활용하여 풍부한 사용자 프로필 기능을 구현
- 신규 Profile 모델에 추가 필드를 구현 (나이, 직업, 관심사, 프로필 사진 등)
- 참고
  - Django Signals

# 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 점 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- <https://lab.ssafy.com/>에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjt'로 지정합니다. (ex. 01-pjt)
- 반드시 각 반 담당 강사님을 Maintainer로 설정해야 합니다.