

# [과제 7] Django의 `django.contrib.auth` 를 활용한 로그인 / 로그아웃 기능 실습

## Django `login()` 함수

Django의 `login()` 함수는 사용자를 인증된 상태로 세션에 등록하는 함수입니다. 이 함수는 `django.contrib.auth` 모듈에 포함되어 있고, 로그인 처리를 수동으로 할 때 주로 사용됩니다.

### ✓ 기본 개념

```
from django.contrib.auth import authenticate, login
```

- `authenticate()` : 사용자 인증을 수행 (아이디/비밀번호가 맞는지 확인)
- `login()` : 인증된 사용자를 현재 세션에 로그인 처리함

### ✓ 기본 예제

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login
from django.contrib import messages

def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']

        user = authenticate(request, username=username, password=password)
        # 사용자 인증
        if user is not None:
            login(request, user) # 로그인 (세션 등록)
            return redirect('home') # 로그인 성공 후 이동
        else:
            messages.error(request, '아이디 또는 비밀번호가 틀렸습니다.')
```

```
return render(request, 'login.html')
```

## ✓ 작동 방식

- `login(request, user)` 는 사용자 정보를 세션에 저장하여 `request.user` 가 로그인된 사용자로 인식되게 만듭니다.
- 내부적으로는 Django의 세션 프레임워크를 사용하여 `sessionid` 쿠키를 설정합니다.
- 이로써 다음 요청부터는 `request.user.is_authenticated == True` 로 동작합니다.

## ✓ URL 연결

```
# urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('login/', views.login_view, name='login'),
]
```

## ✓ 템플릿 예시

```
<form method="post">
  {% csrf_token %}
  <input type="text" name="username" placeholder="아이디">
  <input type="password" name="password" placeholder="비밀번호">
  <button type="submit">로그인</button>
</form>
```

## ✓ 관련 함수

함수	설명
<code>authenticate()</code>	유저 인증 처리 (비밀번호 확인 포함)
<code>login()</code>	세션에 사용자 등록 (로그인 처리)
<code>logout()</code>	세션에서 사용자 제거 (로그아웃 처리)

`request.user`

현재 로그인된 사용자 객체

## Django `logout()` 함수

Django에서 `logout()` 함수는 사용자의 인증 세션을 종료(로그아웃)하는 함수입니다. 주로 `django.contrib.auth` 모듈에 포함되어 있습니다.

### logout()

#### ✓ 기본 사용법

```
from django.contrib.auth import logout
from django.shortcuts import redirect

def logout_view(request):
    logout(request) # 세션에서 사용자 정보 제거
    return redirect('login') # 로그아웃 후 로그인 페이지 등으로 리다이렉트
```

#### ✓ 작동 방식

- `logout(request)` 는 현재 요청과 연결된 사용자의 세션 데이터를 삭제합니다.
- 로그아웃 처리가 끝나면, 일반적으로 `redirect()` 를 통해 로그인 화면 또는 홈페이지로 이동합니다.
- `request.user` 는 이후에 `AnonymousUser` 객체로 바뀝니다.

#### ✓ `urls.py` 에 연결

```
from django.urls import path
from . import views

urlpatterns = [
    path('logout/', views.logout_view, name='logout'),
]
```

#### ✓ 템플릿에서 로그아웃 버튼 만들기

```
<a href="{% url 'logout' %}">Logout</a>
```

또는 POST 방식으로 로그아웃할 경우 (보안 권장)

```
<form method="post" action="{% url 'logout' %}">
  {% csrf_token %}
  <button type="submit">Logout</button>
</form>
```

## ✓ 주의사항

- `logout()` 은 클라이언트의 세션 쿠키를 무효화하지만, 실제로 세션 객체는 서버에 남아 있을 수 있습니다 (세션 만료 설정에 따라).
- 로그아웃 시 관련 로그를 남기거나 추가 처리가 필요하다면, `logout_view()` 안에 추가하면 됩니다.

# Django의 `django.contrib.auth` 를 활용한 로그인 / 로그아웃 기능 실습

폼, 뷰, URL, 템플릿까지 모두 포함된 프로젝트로, 기본 `User` 모델을 사용하고 `username` 기반 로그인입니다.

## ✓ 실습 목표

- Django 기본 User 모델 기반 로그인/로그아웃
- 인증 함수: `authenticate`, `login`, `logout` 사용
- 로그인 성공 시 환영 메시지, 로그아웃 기능 포함

## 🔧 1. 프로젝트 및 앱 생성

```
django-admin startproject myproject
cd myproject
python manage.py startapp accounts
```



## 2. settings.py 설정

myproject/settings.py 에서 앱 등록:

```

INSTALLED_APPS = [
    ...
    'accounts',
    'django.contrib.auth',
    'django.contrib.sessions',
    'django.contrib.messages',
]

```



## 프로젝트 구조

```

myproject/
├── accounts/
│   ├── views.py
│   ├── urls.py
│   └── templates/accounts/
│       ├── login.html
│       └── home.html
├── myproject/
│   ├── settings.py
│   ├── urls.py
│   └── manage.py

```



## 3. accounts/views.py : 총 8점

```

from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.contrib import messages

def login_view(request):
    if request.method == 'POST':
        username = request.POST._____ # 여기에 동작코드를 작성하세요

```

요(1점)

```
password = request.POST._____ # 여기에 동작코드를 작성하세
```

요(1점)

```
user = authenticate(request, _____, _____) # 여기에  
동작코드를 작성하세요(총 2점, 각 1점)
```

```
if user is not None:
```

```
_____ # 여기에 동작코드를 작성하세요(1점)
```

```
return _____ # 여기에 동작코드를 작성하세요(1점)
```

```
else:
```

```
messages.error(request, '아이디 또는 비밀번호가 올바르지 않습니다.')
```

```
return render(request, _____) # 여기에 동작코드를 작성하세요  
(1점)
```

```
@login_required
```

```
def home_view(request):
```

```
    return render(request, 'accounts/home.html', {'user': request.user})
```

```
def logout_view(request):
```

```
    _____ # 여기에 동작코드를 작성하세요(1점)
```

```
    return redirect('login')
```

#### 4. **accounts/urls.py** : 총 3점

```
from django.urls import path
```

```
from . import views
```

```
urlpatterns = [
```

```
    path('login/', _____, name='login'), # 여기에 동작코드를 작성하세  
요(1점)
```

```
    path('logout/', _____, name='logout'), # 여기에 동작코드를 작성하  
세요(1점)
```

```
    path('home/', _____, name='home'), # 여기에 동작코드를 작성하세  
요(1점)
```

```
]
```

## 5. myproject/urls.py 수정

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('accounts.urls')),
]
```

## 6. 템플릿 만들기

**templates/accounts/login.html** : 총 1점

```
<h2>로그인</h2>
<form method="post">
    _____ # 여기에 동작코드를 작성하세요(1점)
    <input type="text" name="username" placeholder="아이디" required><br>
    <input type="password" name="password" placeholder="비밀번호" required><br>
    <button type="submit">로그인</button>
</form>

{% for message in messages %}
    <p style="color:red;">{{ message }}</p>
{% endfor %}
```

**templates/accounts/home.html** : 총 1점

```
<h2>환영합니다, {{ user.username }}님!</h2>
<p><a href="_____ ">로그아웃</a></p> # 여기에 동작코드를 작성하세요(1점)
```

## 7. 마이그레이션 및 관리자 계정 생성

```
python manage.py makemigrations
python manage.py migrate

# 슈퍼유저 생성 (테스트용 로그인 계정)
python manage.py createsuperuser
```

## 8. 실행 및 테스트

```
python manage.py runserver
```

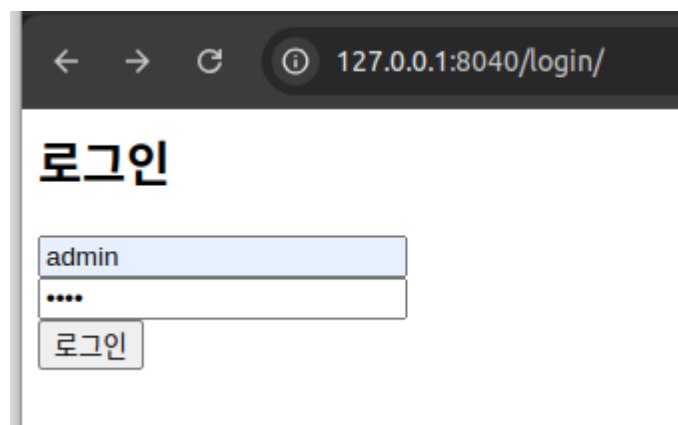
브라우저에서 아래 테스트:

- `http://localhost:8000/login/` → 로그인
- `http://localhost:8000/home/` → 로그인 후 환영 메시지
- `http://localhost:8000/logout/` → 로그아웃

## 요약 정리

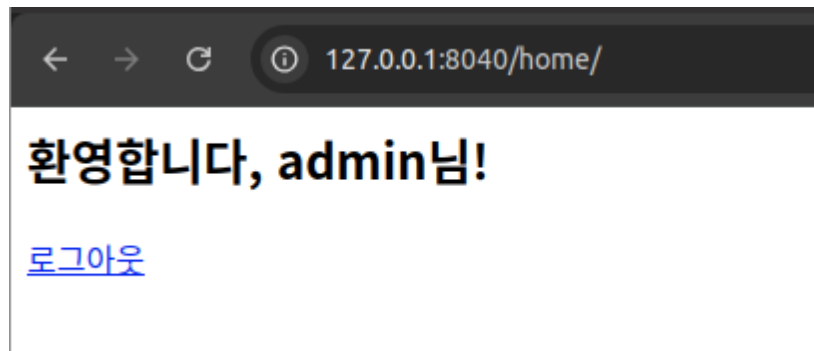
기능	구현
로그인	<code>authenticate()</code> + <code>login()</code>
로그아웃	<code>logout()</code>
로그인 확인	<code>@login_required</code> , <code>request.user.is_authenticated</code>
메시지 출력	<code>django.contrib.messages</code> 사용

### 1. 로그인화면 캡처 (2점)





## 2. 실행화면 (login후 base.html 보여 줌) 캡처 (2점)



## 3. 로그아웃후 화면 캡처 (2점)



- 제출
  - 제출 화면 1,2,3 (각 2점, 6점) 캡처 후 노션이나 hwp, 워드 등을 사용해서 제출 바랍니다.
    - 깃허브 코드 및 사이트 (1점) 제출바랍니다.
  - 가상강의실 제출
- 일정
  - 과제 7 총점 : 총 20점
    - 제출 화면 1,2,3 (각 2점, 6점), 깃허브 코드 및 사이트 (1점) : 총 7점
    - 코드 완성 : 13점
      - `accounts/views.py` : 총 8점
      - `accounts/urls.py` : 총 3점

- `templates/accounts/login.html` : 총 1점
  - `templates/accounts/home.html` : 총 1점
- 5월 14일 (수) 13:30분까지
  - 늦게 제출은 인정하지 않습니다.