

## 2장. 회원관리 - common 앱



# common 앱 생성하기

## ❖ 회원관리를 위한 common app(앱) 생성하기

### 1. blog 프로젝트 하위에 common 앱 생성하기

```
(mysite) > C:\webproject\myblog> python manage.py startapp common
```

### 2. common앱 등록 – config/setting.py

```
INSTALLED_APPS = [  
    'blog',  
    'common',  
    'django.contrib.admin',  
    'django.contrib.auth',
```

### 3. common앱 하위에 urls.py 생성하기

# 로그인 구현하기

## 1. config/urls.py 경로 설정

```
urlpatterns = [  
    path("", views.index),  
    path('admin/', admin.site.urls),  
    path('blog/', include('blog.urls')),  
    path('common/', include('common.urls')),  
]
```

# 로그인 구현하기

## 2. common/urls.py 작성

```
from django.contrib.auth import views as auth_views

app_name = 'common'

urlpatterns = [
    path('login/', auth_views.LoginView.as_view(
        template_name = 'common/login.html'), name='login'),
]
```

# login 구현하기

## 3. login 템플릿 만들기 – templates/common/login.html

```
<h2>로그인</h2>
<form action="" method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">로그인</button>
</form>
```

# login 구현하기

- login 할때 오류 처리 – templates/form\_errors.html

```
{% if form.errors %}
  {% for field in form %}
    {% for error in field.errors %}
      <p><b>{{ field.label }}</b><br>
      <p>{{ error }}</p>
    {% endfor %}
  {% endfor %}
{% endif %}
```

사용자 이름

필수 항목입니다.

비밀번호

필수 항목입니다.

사용자ID

비밀번호

로그인

# 로그인 구현하기

## 5. 로그인, 로그아웃 성공시 이동할 페이지 설정 – config/settings.py

```
# 로그인 시 이동경로  
LOGIN_REDIRECT_URL = '/'  
# 로그아웃 성공시 이동 경로  
LOGOUT_REDIRECT_URL = '/'
```

# 로그아웃 구현하기

[블로그홈](#) | [\(admin\)로그아웃](#) | [회원가입](#) | [글쓰기](#)

## Blog

---

### 대통령 선거일

2021년 12월 26일 6:19 오후

2022년 3월 9일  
투표합시다.





# 로그아웃 구현하기

## 1. URL 매핑 추가 - common/urls.py 작성

```
from django.contrib.auth import views as auth_views

urlpatterns = [
    path('login/', auth_views.LoginView.as_view(
        template_name = 'common/login.html'), name='login'),
    path('logout/', auth_views.LogoutView.as_view(), name='logout'),
]
```

# 로그아웃 구현하기

## 2. 로그인, 로그아웃 메뉴 - navbar.html 작성

```
<nav>
  <ul>
    <li><a href="{% url 'blog:index' %}">Home</a></li>
    <li><a href="{% url 'blog:post_create' %}">글쓰기</a></li>
    {% if user.is_authenticated %}
      <li><a href="{% url 'common:logout' %}">({{ user.username }}님)로그아웃</a></li>
    {% else %}
      <li><a href="{% url 'common:login' %}">로그인</a></li>
    {% endif %}
    <li><a href="{% url 'common:signup' %}">회원가입</a></li>
  </ul>
</nav>
```

# 회원 가입 페이지 만들기

## 회원 가입 폼

[Home](#) [글쓰기](#) [로그인](#) [회원가입](#)

### 회원가입

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

이메일:

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.

# 회원 가입 만들기

## 1. 회원 가입 URL 매핑 – common/urls.py

```
from django.contrib.auth import views as auth_views
app_name = 'common'

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

# 회원 가입 만들기

## 2. UserForm 생성 – common/forms.py

```
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User
from django import forms

class UserForm(UserCreationForm):
    email = forms.EmailField(label='이메일') # 이메일 필드 추가

    class Meta:
        model = User
        fields = ("username", "email") # 튜플 구조
```

# 회원 가입 만들기

## 3. sign up 함수 추가하기 – common/views.py

```
def signup(request):  
    # 회원 가입  
    if request.method == "POST":  
        form = UserForm(request.POST)  
        if form.is_valid():  
            form.save() #db에 저장  
            return redirect('blog:index')  
    else:  
        form = UserForm()  
    context = {'form': form}  
    return render(request, 'common/signup.html', context)
```

# 회원 가입 만들기

## 4. 템플릿 만들기 – templates/common/signup.html

```
<li><a href="{% url 'common:signup' %}">회원가입</a></li>
</ul>
</nav>
</header>
<section>
  <h2>회원가입</h2>
  <form action="" method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">가입</button>
    <button type="reset">취소</button>
  </form>
</section>
</div>
```

# 회원 가입 만들기

- ✓ 인증된 사용자만 글쓰기 가능

```
@login_required(login_url='common:login')
def post_create(request):
    # 포스트 등록
    if request.method == "POST":
        # 글과 이미지 각각 가져옴
        form = PostForm(request.POST, request.FILES)
```

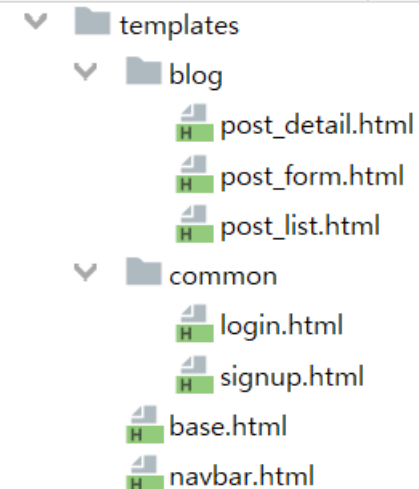


# 표준 템플릿 만들기

- 표준 템플릿 만들기 – base.html

```
<!DOCTYPE html>
<html lang="ko">
{% load static %}
<head>
    <meta charset="UTF-8">
    <title>{% block title %} {% endblock %}</title>
    <link rel="stylesheet" href="{% static 'post/css/style.css' %}">
</head>
<body>
    <div id="container">
        <!-- 네비게이션 파일 포함 -->
        {% include 'navbar.html' %}

        {% block content %}
        {% endblock %}
    </div>
</body>
</html>
```



# 네비게이션 파일 분리

- 네비게이션 파일 – navbar.html

```
<header>
  <nav>
    <ul>
      <li><a href="{% url 'blog:index' %}">Home</a></li>
      <li><a href="{% url 'blog:post_create' %}">글쓰기</a></li>
      {% if user.is_authenticated %}
        <li><a href="{% url 'common:logout' %}">({{ user.username }}님)로그아웃</a></li>
      {% else %}
        <li><a href="{% url 'common:login' %}">로그인</a></li>
      {% endif %}
      <li><a href="{% url 'common:signup' %}">회원가입</a></li>
    </ul>
  </nav>
</header>
```

# 네비게이션 파일 분리

- {% extends 'base.html' %} - 템플릿 파일 상속(확장)

```
{% extends 'base.html' %}
{% block title %}Blog - 전체 보기{% endblock %}
{% block content %}
<section>
  <h2>Blog</h2>
  {% if post_list %}
    {% for post in post_list %}
      <hr>
```

post\_list.html

```
{% extends 'base.html' %}
{% block title %} Blog - {{ post.title }} {% endblock %}
{% block content %}
<section>
  <h2>{{ post.title }}</h2>
```

post\_detail.html

# 작성자 추가하기

[블로그홈](#) | [\(today\)로그아웃](#) | [회원가입](#) | [글쓰기](#)

## Blog

### 인공지능 스피커 - 프렌즈

작성자:today (작성일:2021년 12월 26일 6:43 오후)

인공지능 스피커 - 프렌즈  
네이버가 만든 제품



# 작성자 추가하기

- author 필드 추가하기

1. Post 모델에 author 추가 – User의 외래키

```
from django.contrib.auth.models import User

class Post(models.Model):
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    photo = models.ImageField(upload_to="blog/images/%Y/%m/%d/",
                              null=True, blank=True)
    create_date = models.DateTimeField()
```

# 작성자 추가하기

## 2. 변경된 DB 적용을 위한 makemigrations

```
Please select a fix:
  1) Provide a one-off default now (will be set on all existing
  2) Quit, and let me add a default in models.py
Select an option: 1
Please enter the default value now, as valid Python
The datetime and django.utils.timezone modules are available, s
Type 'exit' to exit this prompt
```

>>> 1 → 1은 superuser의 id 값을 의미함

## 3. DB 적용을 위한 migrate 명령 실행

# 작성자 추가하기

## 4. post\_create()에 form.author 추가 – blog/views.py

```
@login_required(login_url='common:login')
def post_create(request):
    # 포스트 등록
    if request.method == "POST":
        # 글과 이미지 각각 가져옴
        form = PostForm(request.POST, request.FILES)
        if form.is_valid():
            post = form.save(commit=False) #가져장
            post.pub_date = timezone.now() #등록일
            post.author = request.user #작성자
            post.save() #db에 저장
            return redirect('blog:index')
        else:
            form = PostForm()
            context = {'form': form}
            return render(request, 'blog/post_form.html', context)
```

# 작성자 추가하기

```
{% for post in post_list %}
```

post\_list.html

```
<hr>
```

```
<h3><a href="{% url 'blog:detail' post.id %}">{{ post.title }}</a></h3>
```

```
<h3><a href="blog/{{ post.id }}">{{ post.title }}</a></h3>-->
```

```
<h5>작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})</h5>
```

```
<p style="white-space:pre-line; line-height:2em">{{ post.content }}</p>
```

```
{% if post.photo %}
```

```

```

```
{% endif %}
```

```
{% endfor %}
```

```
<h2>{{ post.title }}</h2>
```

post\_detail.html

```
<h5>작성자 : {{ post.author }} (작성일 : {{ post.pub_date }})</h5>
```

```
<p style="white-space:pre-line; line-height:2em">{{ post.content }}</p>
```

```
<p>{{ post.content }}</p>
```



# login 구현하기

## 3. login 템플릿 만들기 – templates/common/login.html

```
<form action="{% url 'common:login' %}" method="post">
  {% csrf_token %}
  <fieldset style="width: 300px;">
    <legend>로그인</legend>
    {% include 'form_errors.html' %}
    <p>
      <label for="username">아이디</label>
      <input type="text" name="username" id="username"
        value="{% form.username.value|default_if_none:'' %}">
    </p>
    <p>
      <label for="password">비밀번호</label>
      <input type="password" name="password" id="password"
        value="{% form.password.value|default_if_none:'' %}">
    </p>
    <button type="submit">로그인</button>
  </fieldset>
</form>
```

# 회원 가입 만들기

## 4. 템플릿 만들기 – templates/common/signup.html

```
<form method="post">
  {% csrf_token %}
  <fieldset style="width: 300px;">
    <legend>회원 가입</legend>
    {% include 'form_errors.html' %}
    <p>
      <label for="username">아이디</label>
      <input type="text" name="username" id="username"
        value="{{ form.username.value|default_if_none:'' }}">
    </p>
    <p>
      <label for="password1">비밀번호</label>
      <input type="password" name="password1" id="password1"
        value="{{ form.password1.value|default_if_none:'' }}">
    </p>
```









# 회원 가입 만들기

## 3. 템플릿 만들기 – templates/common/signup.html

```
<p>
    <label for="password2">비밀번호 확인</label>
    <input type="password" name="password2" id="password2"
        value="{{ form.password2.value|default_if_none:'' }}">
</p>
<p>
    <label for="email">이메일</label>
    <input type="email" name="email" id="email"
        value="{{ form.email.value|default_if_none:'' }}">
</p>
<button type="submit">가입</button>
<button type="reset">취소</button>
</fieldset>
</form>
```

# 깃으로 버전(소스) 관리하기

## ➤ 깃에 업로드하기

	sugu2100 common 앱 생성	
	blog	common 앱 생성
	common	common 앱 생성
	media/blog/images/2021/12/26	common 앱 생성
	myblog	common 앱 생성
	templates	common 앱 생성
	.gitignore	common 앱 생성
	manage.py	blog site start