

■ 로그인 / 로그아웃

● 프로젝트 / 앱 생성

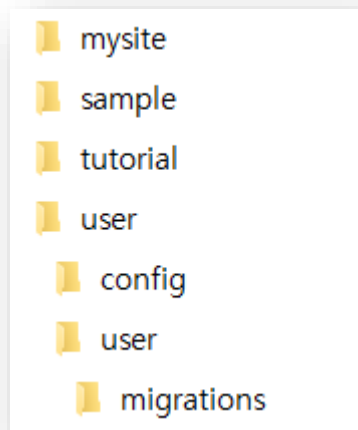
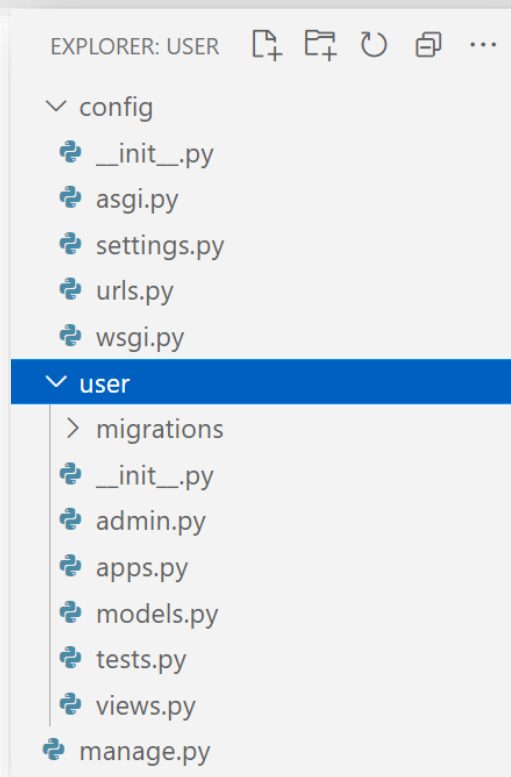
```
C:\WdevWdjango>mkdir user
```

```
C:\WdevWdjango>cd user
```

```
C:\WdevWdjango\user>django-admin startproject config .
```

```
C:\WdevWdjango\user>django-admin startapp user
```

```
C:\WdevWdjango\user>
```



■ 로그인

● 작성 절차

1. settings.py app 등록
2. makemigrations & migrate 실행
3. urls.py 주소 생성
4. 로그인 템플릿 작성
5. 로그인 후 이동할 페이지 URL 지정

■ 로그인

● 1. settings.py app 등록

```
INSTALLED_APPS = [  
    'user', # user.apps.UserConfig  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

■ 로그인

● 2. makemigrations & migrate 실행

- python manage.py makemigrations
- python manage.py migrate

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\dev\django\user> python manage.py makemigrations
No changes detected
PS C:\dev\django\user> █
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\dev\django\user> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
PS C:\dev\django\user> █
```

■ 로그인

● 3. urls.py 주소 생성

```
from django.contrib import admin
from django.urls import path
from django.contrib.auth import views as auth_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path(
        'login/', auth_views.LoginView.as_view(),
        name='login'
    )
]
```

■ 로그인

● 결과 확인

← → ↺ ⓘ 127.0.0.1:8000/login/ ☆ ⚙ ⌵

TemplateDoesNotExist at /login/

registration/login.html

Request Method: GET
Request URL: http://127.0.0.1:8000/login/
Django Version: 3.2.3
Exception Type: TemplateDoesNotExist
Exception Value: registration/login.html
Exception Location: C:\Users\GGoReb\miniconda3\lib\site-packages\django\template\loader.py, line 47, in select_template
Python Executable: C:\Users\GGoReb\miniconda3\python.exe
Python Version: 3.8.5
Python Path: ['C:\\dev\\django\\user', 'C:\\Users\\GGoReb\\miniconda3\\python38.zip', 'C:\\Users\\GGoReb\\miniconda3\\DLLs', 'C:\\Users\\GGoReb\\miniconda3\\lib', 'C:\\Users\\GGoReb\\miniconda3', 'C:\\Users\\GGoReb\\miniconda3\\lib\\site-packages', 'C:\\Users\\GGoReb\\miniconda3\\lib\\site-packages\\win32', 'C:\\Users\\GGoReb\\miniconda3\\lib\\site-packages\\win32\\lib', 'C:\\Users\\GGoReb\\miniconda3\\lib\\site-packages\\Pythonwin']
Server time: Tue, 15 Jun 2021 04:02:21 +0000

Template-loader postmortem

Django tried loading these templates, in this order:

Using engine django:

- django.template.loaders.app_directories.Loader:

- LoginView는 registration 디렉토리 내의 login.html을 기본 경로 및 파일로 사용

- template_name 옵션으로 경로 및 파일 변경 가능

■ 로그인

● 3. urls.py 주소 생성 (수정)

```
from django.contrib import admin
from django.urls import path
from django.contrib.auth import views as auth_views

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

■ 로그인

● 4. 로그인 템플릿 작성 - login.html

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>

<div class="container my-3">
  <form method="post" class="post-form" action="{% url 'login' %}">
    {% csrf_token %}
    {% include "form_errors.html" %}
    <div class="form-group">
      <label for="username">사용자ID</label>
      <input type="text" class="form-control" name="username" id="username"
        value="{% form.username.value|default_if_none:'' %}">
    </div>
    <div class="form-group">
      <label for="password">비밀번호</label>
      <input type="password" class="form-control" name="password"
        id="password" value="{% form.password.value|default_if_none:'' %}">
    </div>
    <button type="submit" class="btn btn-primary">로그인</button>
  </form>
</div>
```


■ 로그인

● 4. 로그인 템플릿 작성 - form-errors.html

```
{% if form.errors %}
```

```
    {% for field in form %}
```

```
        {% for error in field.errors %}
```

```
            <!-- 필드 오류를 출력한다. -->
```

```
            <div class="alert alert-danger">
```

```
                <strong>{{ field.label }}</strong>
```

```
                {{ error }}
```

```
            </div>
```

```
        {% endfor %}
```

```
    {% endfor %}
```

○ 폼 오류 2가지

- 필드 오류 : 입력값이 누락되었거나 형식에 맞지 않음
- 넌필드 오류 : 입력값과 관계없이 발생한 오류

```
    {% for error in form.non_field_errors %}
```

```
        <!-- 넌필드 오류를 출력한다. -->
```

```
        <div class="alert alert-danger">
```

```
            <strong>{{ error }}</strong>
```

```
        </div>
```

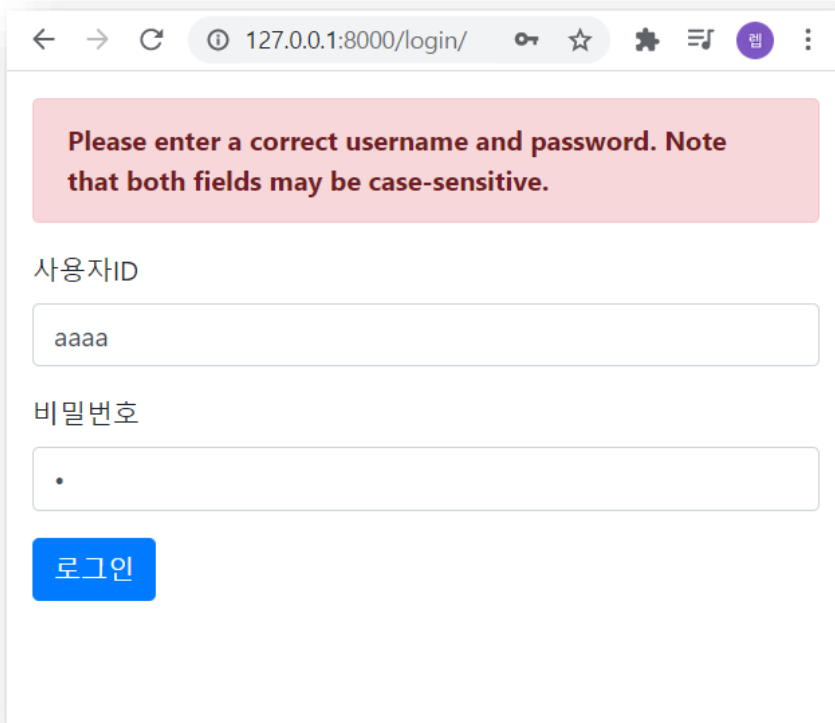
```
    {% endfor %}
```

```
{% endif %}
```

■ 로그인

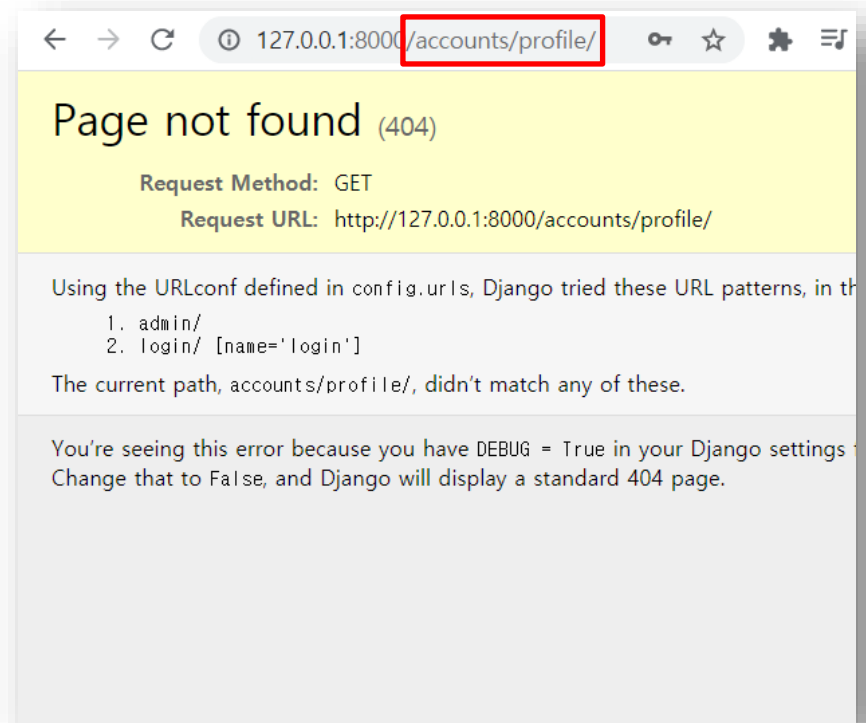
● 결과 확인

존재하지 않는 사용자 정보로 로그인



A screenshot of a web browser showing a login page. The address bar displays '127.0.0.1:8000/login/'. A pink error message box at the top states: 'Please enter a correct username and password. Note that both fields may be case-sensitive.' Below this, there are two input fields: '사용자ID' (Username) containing 'aaaa' and '비밀번호' (Password) containing a single dot. A blue button labeled '로그인' (Login) is at the bottom.

admin 계정으로 로그인



A screenshot of a web browser showing a 404 error page. The address bar displays '127.0.0.1:8000/accounts/profile/' with the path portion highlighted by a red box. The page has a yellow header with the text 'Page not found (404)'. Below this, it shows 'Request Method: GET' and 'Request URL: http://127.0.0.1:8000/accounts/profile/'. The main content area explains that Django tried URL patterns 'admin/' and 'login/ [name='login']' but they didn't match the current path 'accounts/profile/'. A footer note mentions that the error is visible because 'DEBUG = True' in the Django settings.

■ 로그인

● 5. 로그인 후 이동할 URL 지정 - settings.py

...

```
LOGIN_REDIRECT_URL = '/login/'
```

login.html

```
...
</form>

{% if user.is_authenticated %}
    <p>{{ user.username }} 님 반갑습니다.</p>
{% else %}
    <p>현재 상태 : Guest</p>
{% endif %}
</div>
```

로그인

현재 상태 : Guest

로그인

admin 님 반갑습니다.

■ 로그아웃

● 작성 절차

1. urls.py 주소 생성
2. 로그인 후 이동할 페이지 URL 지정

■ 로그아웃

● 1. urls.py 주소 생성

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

■ 로그아웃

● 2. 로그인 후 이동할 URL 지정 - settings.py

...

```
LOGIN_REDIRECT_URL = '/login/'  
LOGOUT_REDIRECT_URL = '/login/'
```

login.html

```
...  
</form>  
  
{% if user.is_authenticated %}  
  <p>  
    <a class="nav-link" href="{% url 'logout' %}">  
      {{ user.username }} 님 반갑습니다. (로그아웃)  
    </a>  
  </p>  
{% else %}  
  <p>  
    현재 상태 : Guest  
  </p>  
{% endif %}  
</div>
```

로그인

admin 님 반갑습니다. (로그아웃)

■ 회원가입

● 작성 절차

1. urls.py 주소 생성
2. 회원가입 템플릿 작성
3. form 작성 - 유효성 검사용
4. view 작성 - 회원가입 로직

■ 회원가입

● 1. urls.py 주소 생성 - urls.py

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


■ 회원가입

● 2. 회원가입 템플릿 작성 - signup.html

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js">
</script>

<div class="container my-3">
  <div class="row my-3">
    <div class="col-4">
      <h4>계정생성</h4>
    </div>
    <div class="col-8 text-right">
      <span>또는 <a href="{% url 'login' %}">로그인 하세요.</a></span>
    </div>
  </div>
</div>
```

■ 회원가입

● 2. 회원가입 템플릿 작성 - signup.html

```
<form method="post" class="post-form">
  {% csrf_token %}
  {% include "form_errors.html" %}
  <div class="form-group">
    <label for="username">사용자 이름</label>
    <input type="text" class="form-control" name="username" id="username"
      value="{% form.username.value|default_if_none:'' %}">
  </div>
  <div class="form-group">
    <label for="password1">비밀번호</label>
    <input type="password" class="form-control" name="password1"
      id="password1" value="{% form.password1.value|default_if_none:'' %}">
  </div>
  <div class="form-group">
    <label for="password2">비밀번호 확인</label>
    <input type="password" class="form-control" name="password2"
      id="password2" value="{% form.password2.value|default_if_none:'' %}">
  </div>
  <div class="form-group">
    <label for="email">이메일</label>
    <input type="text" class="form-control" name="email" id="email"
      value="{% form.email.value|default_if_none:'' %}">
  </div>
  <button type="submit" class="btn btn-primary">생성하기</button>
</form>
</div>
```

■ 회원가입

● 3. form 작성 - forms.py

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

class UserForm(UserCreationForm):
    email = forms.EmailField(label="이메일")

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

■ 회원가입

● 4. view 작성 - 회원가입 로직

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

from .forms import UserForm

def signup(request):
    if request.method == "POST":
        form = UserForm(request.POST)
        if form.is_valid():
            form.save()

            username = form.cleaned_data.get('username')
            raw_password = form.cleaned_data.get('password1')

            user = authenticate(username=username, password=raw_password)
            login(request, user)

            return redirect('index')
    else:
        form = UserForm()
    return render(request, 'signup.html', {'form': form})
```

■ 회원가입

● 결과 확인

http://127.0.0.1:8000/signup/

계정생성 [또는 로그인 하세요.](#)

사용자 이름

비밀번호

비밀번호 확인

이메일

[생성하기](#)

생성하기

계정생성 [또는 로그인 하세요.](#)

사용자 이름

비밀번호

비밀번호 확인

이메일

[생성하기](#)

8자 이상
숫자만 X
연속숫자 X
...

사용자ID

비밀번호

[로그인](#)

[user](#) 님 반갑습니다. (로그아웃)