



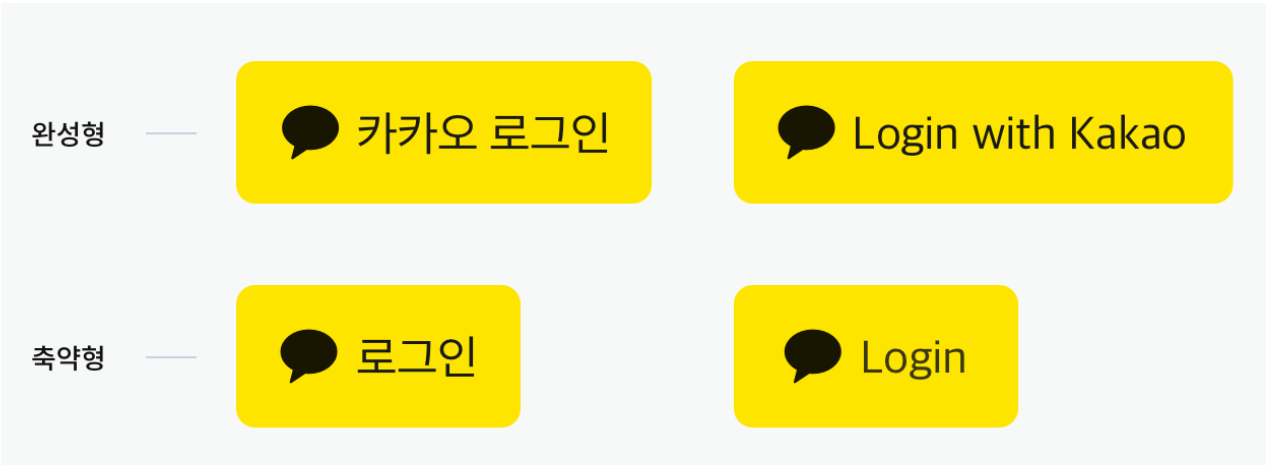
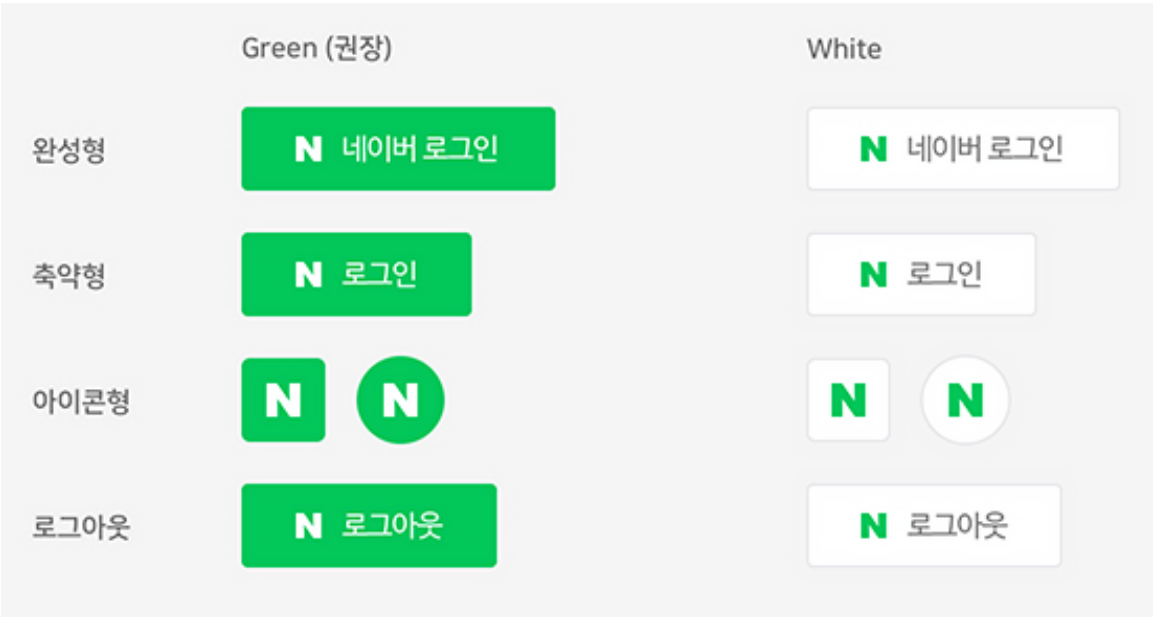
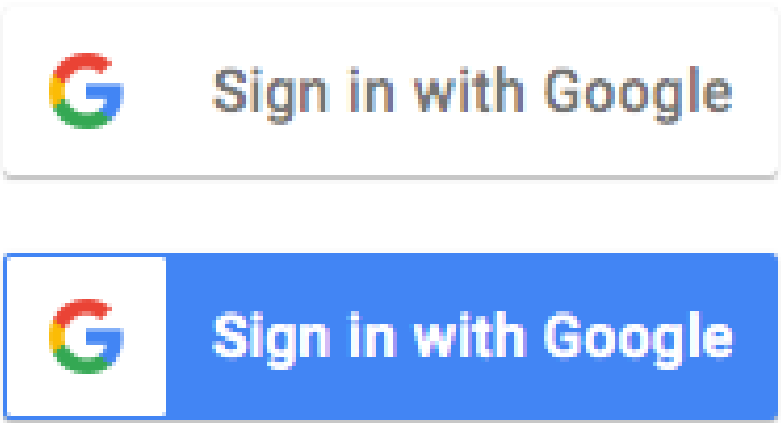
소셜로그인

11기 운영진 김민지

멋쟁이사자처럼 대학 11기

소셜로그인

플랫폼 별 제공하는 로그인 가이드가 있음



- 구글: <https://developers.google.com/identity/branding-guidelines?hl=ko>
- 네이버: <https://developers.naver.com/docs/login/bi/bi.md>
- 카카오: <https://developers.kakao.com/docs/latest/ko/kakaologin/design-guide>

소셜로그인

```
(myvenv) C:\Users\kim\Desktop\교육자료>pip install django-allauth
```

2. setting.py 수정 # provider 구글

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.sites',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'blog.apps.BlogConfig',  
    'account.apps.AccountConfig',  
    'allauth',  
    'allauth.account',  
    'allauth.socialaccount',  
    'allauth.socialaccount.providers.google',  
]
```

소셜로그인

3. 맨 밑에 추가

```
140 AUTHENTICATION_BACKENDS = (  
141     # Needed to login by username in Django admin, regardless of 'allauth'  
142     'django.contrib.auth.backends.ModelBackend',  
143  
144     # 'allauth' specific authentication methods, such as login by email  
145     'allauth.account.auth_backends.AuthenticationBackend',  
146 )  
147  
148 SITE_ID = 1  
149 LOGIN_REDIRECT_URL = '/'
```

4. profile 외에 email을 추가적으로 받아오고 싶을 때

```
151 SOCIALACCOUNT_PROVIDERS = {  
152     'google': {  
153         'SCOPE': [  
154             'profile',  
155             'email',  
156         ],  
157         'AUTH_PARAMS': {  
158             'access_type': 'online'  
159         }  
160     }  
161 }
```

소셜로그인

5. 경로 추가

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('blog.urls')),  
    path('account/', include('account.urls')),  
    path('accounts/', include('allauth.urls')),  
    path('', views.index, name="index"),  
] + static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

6. makemigrations

```
(myvenv) C:\Users\kim\Desktop\likelion\백엔드>python manage.py makemigrations
```

오류발생

```
django.core.exceptions.ImproperlyConfigured: Application labels aren't unique, duplicates: account
```

```
'account.apps.AccountConfig',
```

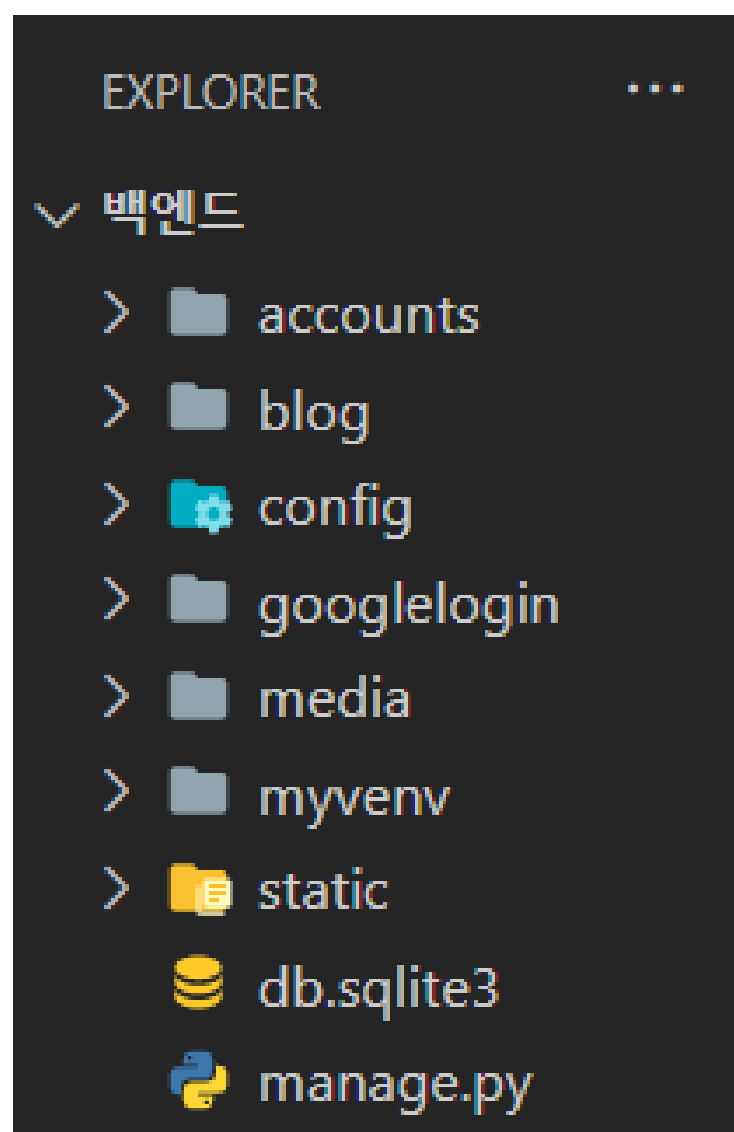
```
'allauth.account',
```

INSTALLED_APPS의 이름이 중복되기 때문

오류 해결

해결 방법: 기존의 회원관리 앱인 account의 이름을 accounts로 변경

1. 폴더를 우클릭, rename, accounts로 변경



2. INSTALLED_APPS에 앱 이름 변경

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.sites',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'blog.apps.BlogConfig',  
    'accounts.apps.AccountsConfig',  
    'allauth',  
    'allauth.account',  
    'allauth.socialaccount',  
    'allauth.socialaccount.providers.google',  
]
```

3. accounts 폴더의 apps.py 변경

```
class AccountsConfig(AppConfig):  
    default_auto_field = 'django.db.models.BigAutoField'  
    name = 'accounts'
```

오류 해결

4. setting.py 의 AUTH_USER_MODEL 변경

```
30 AUTH_USER_MODEL = 'accounts.CustomUser'
```

5. url 변경

```
✓ urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('blog.urls')),  
    path('account/', include('accounts.urls')),  
    path('accounts/', include('allauth.urls')),  
    path('', views.index, name="index"),  
] + static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

6-1. dbsqlite 지우기

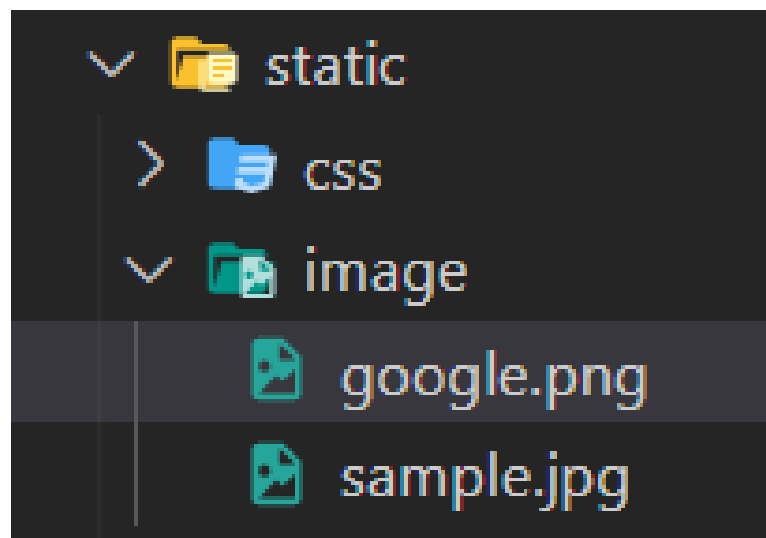
6-2. 데이터를 지우고 싶지 않을 경우 DB수정하기 (인터넷 참고)

7. makemigrations, migrate

8. 서버 열어서 확인

소셜로그인

8. 로그인 아이콘 추가



9. 로그인 html 수정

```
{% block content %}
{% load static %}
{% load socialaccount %}
{% providers_media_js %}

<form action="" method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <input type="submit" value="로그인">
</form>

<a href="{% provider_login_url 'google' %}"></a>

{% endblock %}
```


소셜로그인

10. site 추가

SITES

Sites

+ Add

✎ Change

기존의 example.com 수정

Change site

http://127.0.0.1:8000

Domain name:

http://127.0.0.1:8000

Display name:

http://127.0.0.1:8000

ADD SITE한다면 setting.py의 SITE_ID = 1를 2로 변경

소셜로그인

11. social applications 등록

SOCIAL ACCOUNTS		
Social accounts	+ Add	✎ Change
Social application tokens	+ Add	✎ Change
Social applications	+ Add	✎ Change

Add social application

Provider:

Google ▾

Name:

google

Client id:

App ID, or consumer key

Secret key:

API secret, client secret, or consumer secret

Key:


Key


provider google 선택, name 지정

구글 클라우드

12. client id, secret key 만들기

<http://console.cloud.google.com/apis/dashboard>






리소스, 문서, 제품 등 검색(/)


API

API 및 서비스

← 웹 애플리케이션의 클라이언트 ID

 삭제

새 프로젝트




projects 할당량이 7개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)

[MANAGE QUOTAS](#)


프로젝트 이름 *

myblog



프로젝트 ID: myblog-390608입니다. 나중에 변경할 수 없습니다. [수정](#)

위치 *

 조직 없음

[찾아보기](#)

상위 조직 또는 폴더

만들기

취소

구글 클라우드

동의 화면 구성 - 필수 요소만 넣고 넘기기

Google Cloud

myblog

리소스, 문서, 제품 등 검색(/)

검색

민지

API API 및 서비스

사용 설정된 API 및 서비스

라이브러리

사용자 인증 정보

OAuth 동의 화면

페이지 사용 동의

사용자 인증 정보

+ 사용자 인증 정보 만들기

삭제

삭제된 사용자 인증 정보 복원

사용 설정한 API에 액세스하려면 사용자 인증 정보를 만드세요. 자세히 알아보기

앱리케이션에 대한 정보를 포함하여 OAuth 동의 화면을 구성해야 합니다.

동의 화면 구성

API 키

이름

생성일

↓

제한사항

작업

표시할 API 키가 없습니다.

OAuth 2.0 클라이언트 ID

이름

생성일

↓

유형

클라이언트 ID

작업

표시할 OAuth 클라이언트가 없습니다.

서비스 계정

이메일

이름

↑

작업

표시할 서비스 계정이 없습니다.

서비스 계정 관리

구글 클라우드

1

nyblog ▾

리소스, 문서, 제품 등 검색(/)

사용자 인증 정보

+ 사용자 인증 정보 만들기

삭제

삭제된 사용자 인증 정보 복원

사용 설정한 API에 액세스하려면

API 키

☐ 이름

표시할 API 키가 없습니다.

OAuth 2.0 클라이언트 ID

☐ 이름

생성일 ↓

유형

API 키

할당량과 액세스 권한을 확인하기 위해 간단한 API 키로 프로젝트를 확인합니다.

OAuth 클라이언트 ID

앱에서 사용자 데이터에 액세스할 수 있도록 사용자 동의를 요청합니다.

서비스 계정

로봇 계정을 사용하여 서버 간의 앱 수준 인증을 사용 설정합니다.

사용자 인증 정보 선택 도움말

사용할 사용자 인증 정보의 유형을 결정할 수 있도록 몇 가지 질문을 합니다.

2

← OAuth 클라이언트 ID 만들기

클라이언트 ID는 Google OAuth 서버에서 단일 앱을 식별하는 데 사용됩니다. 앱이 여러 플랫폼에서 실행되는 경우 각각 자체 클라이언트 ID가 있어야 합니다. 자세한 내용은 [OAuth 2.0 설정](#)을 참조하세요. OAuth 클라이언트 유형을 [자세히 알아보세요](#).

애플리케이션 유형 *
웹 애플리케이션

이름 *
blog

OAuth 2.0 클라이언트의 이름입니다. 이 이름은 콘솔에서 클라이언트를 식별하는 용도로만 사용되며 최종 사용자에게 표시되지 않습니다.

3

승인된 자바스크립트 원본 ?

브라우저 요청에 사용

URI 1 *
http://127.0.0.1:8000

+ URI 추가

승인된 리디렉션 URI ?

웹 서버의 요청에 사용

URI 1 *
http://127.0.0.1:8000

URI 2 *
http://127.0.0.1:8000/accounts/google/login/callback/

+ URI 추가

참고: 설정이 적용되는 데 5분에서 몇 시간이 걸릴 수 있습니다.


저장

취소

구글 클라우드

OAuth 클라이언트 생성됨

API 및 서비스의 사용자 인증 정보에서 언제든지 클라이언트 ID와 보안 비밀에 액세스할 수 있습니다.


 OAuth 액세스는 [OAuth 동의 화면](#)에 나열된 [테스트 사용자](#)로 제한됩니다.

클라이언트 ID	570494671045-5uevdrqmo5rkl0qh5kdv0mnar5nqopae.apps.googleusercontent.com
클라이언트 보안 비밀번호	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
생성일	2023년 6월 22일 PM 5시 45분 0초 GMT+9
상태	✔ 사용 설정됨

📄 JSON 다운로드

Sites:

Available sites ⓘ



Chosen sites ⓘ

<http://127.0.0.1:8000>



소셜로그인

13. 중간 페이지 생략

Menu:

- [Sign In](#)
- [Sign Up](#)

Sign In Via Google

You are about to sign in using a third party account from Google.

Continue

```
148  SITE_ID = 1
149
150  SOCIALACCOUNT_LOGIN_ON_GET=True
151  LOGIN_REDIRECT_URL = '/'
152
153  SOCIALACCOUNT_PROVIDERS = {
```