

12 DataBase

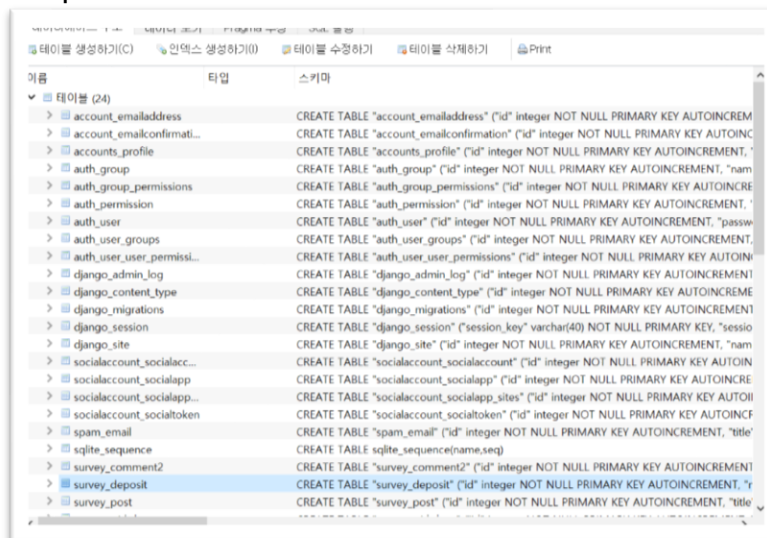
12. DataBase

■ 백엔드

- 서버에서 정보를 저장하고 빼오는 로직
- 상반기 → 서버와 클라이언트가 통신하는 방법
- 하반기 → 정보를 다루는 방법
- 정보는 DataBase에 저장한다. (ex: db.sqlite3)
- Django에서는 models.py 를 이용한다.
- 개념을 만드는 일

■ SQL

- Sqlite3 등 DataBase에서 쓰는 문법



12. DataBase

- 백엔드를 하려면 SQL을 알아야 하는 게 맞다
 - 원래 db를 만들거나 수정하려면 SQL을 알아야 함.
- Models.py
 - Model이라는 클래스를 사용해서 SQL을 몰라도 DataBase를 다룰 수 있도록 함.
 - 장고 개발자들이 파이썬 문법만으로도 DataBase를 다룰 수 있도록 SQL을 연결하여 코딩해둔 것.
 - Django를 쓰면 SQL을 몰라도 된다~ (당장은)

```
models.py

1  from django.db import models
2
3  # Create your models here.
4
5  class Post(models.Model):
6      title = models.CharField(max_length=100)
```

12. DataBase

■ Migration

- Python 문법으로 만든 모델을 실제 db에 적용하는 과정
- Python manage.py makemigrations
 - 몇 번째에 어떻게 코드를 수정했는지 알려주는 파이썬 파일(.py)를 만들어줌.
- Python manage.py migrate
 - makemigrations로 만든 파이썬 파일을 토대로, 가장 최신버전으로 db를 실제 업데이트 하여 적용. 이전 버전으로 되돌아가는 것도 가능.
- Models.py를 수정 → 반드시 migration을 해야 함

```
Anaconda Prompt
(base) C:\Users\송규호\pyworks\django_ex\ycc3>python manage.py makemigrations
Migrations for 'blog':
  blog\migrations\0001_initial.py
    - Create model Post

(base) C:\Users\송규호\pyworks\django_ex\ycc3>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, blog, 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 blog.0001_initial... OK
  Applying sessions.0001_initial... OK

(base) C:\Users\송규호\pyworks\django_ex\ycc3>
```

12. DataBase

■ 만든 모델 불러오기

- 일반적으로 views.py 에 불러와서 사용.
- 어떤 모델을 어떤 함수에서 사용해도 상관 없음.
- 다른 모듈(models.py != views.py)이기 때문에 반드시 import를 하여 사용해야 함.
- Models.py를 수정 → 반드시 migration을 해야 함

■ 모델에 접근하기

- 변수 = 모델.objects.all()
- Ex : post_list = Post.objects.all()

■ 모델 사용하기

- render의 세 번째 인자
- “.html 에서 사용할 수 있는 모델을 넘겨주겠다”
- post_list 이름 바꿔보기

```
1  from django.shortcuts import render
2  from .models import Post
3
4  # Create your views here.
5
6  def list(request):
7      post_list = Post.objects.all()
8      return render(request, 'list.html', {
9          'post_list':post_list,
10     })
11
12
```

12. DataBase

- Django로부터 받은 변수 html에서 사용하기
 - {{ }}
 - Class에서 정의한 변수 사용

```
list.html
1  <a href = "http://127.0.0.1:8000/community/write">good!!</a>
2
3  {{post_list.title}}
```

- 아무것도 안 나온다. 왜? title에 아무것도 안 넣었으니까!

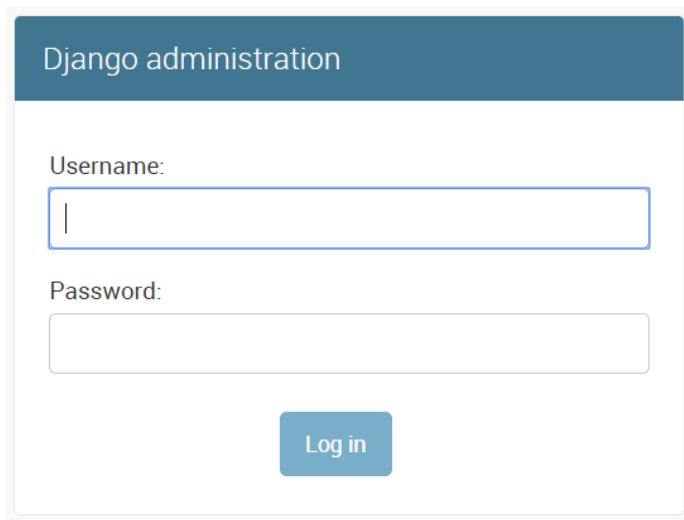
13 관리자 모드

13. 관리자 모드

- `urls.py`
 - `'admin/'`

```
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('blog/', include('blog.urls')),
22     path('community/', include('community.urls')),
23 ]
```

- Go to `127.0.0.1:8000/admin`

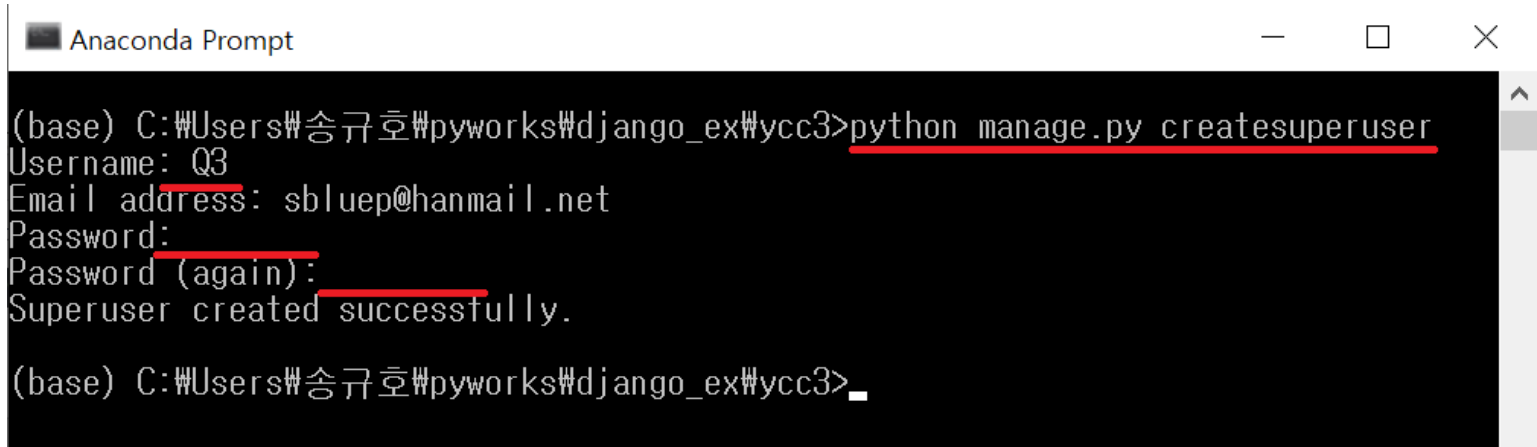


The image shows the Django administration login interface. It has a dark blue header with the text "Django administration". Below the header, there are two input fields: "Username:" and "Password:". The "Username:" field is currently empty with a cursor. Below the "Password:" field is a "Log in" button.

13. 관리자 모드

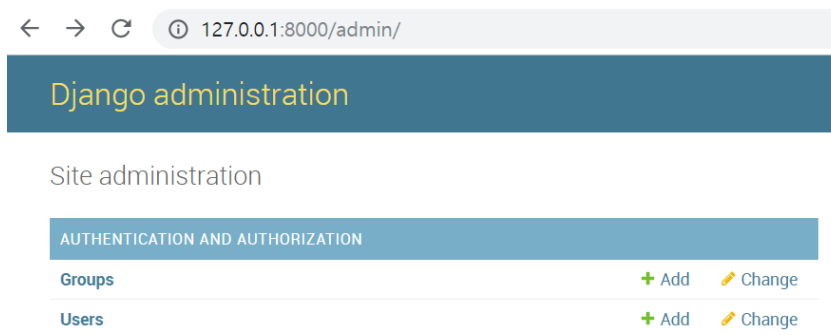
■ 관리자 아이디 만들기

- python manage.py createsuperuser



```
Anaconda Prompt
(base) C:\Users\송규호\pyworks\django_ex\ycc3>python manage.py createsuperuser
Username: Q3
Email address: sbluep@hanmail.net
Password:
Password (again):
Superuser created successfully.
(base) C:\Users\송규호\pyworks\django_ex\ycc3>_
```

■ 로그인



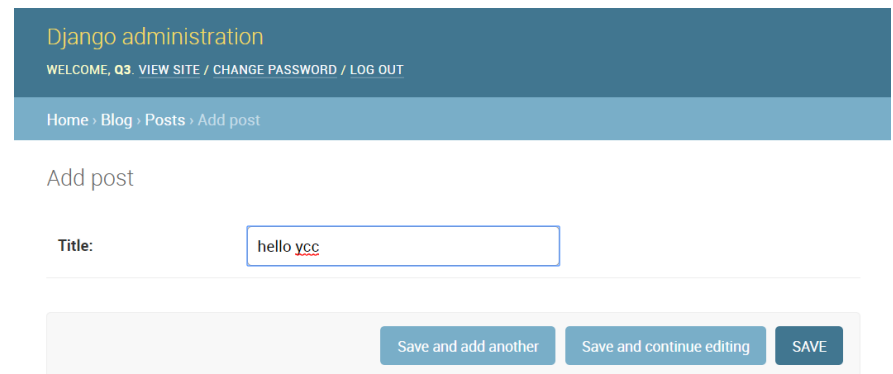
13. 관리자 모드

■ admin.py

```
admin.py

1  from django.contrib import admin
2  from .models import Post
3
4  # Register your models here.
5
6  @admin.register(Post)
7  class PostAdmin(admin.ModelAdmin):
8      list_display = ('title',)
```

■ 관리자 페이지 새로고침 → Add → SAVE → 127.0.0.1:8000/blog/list

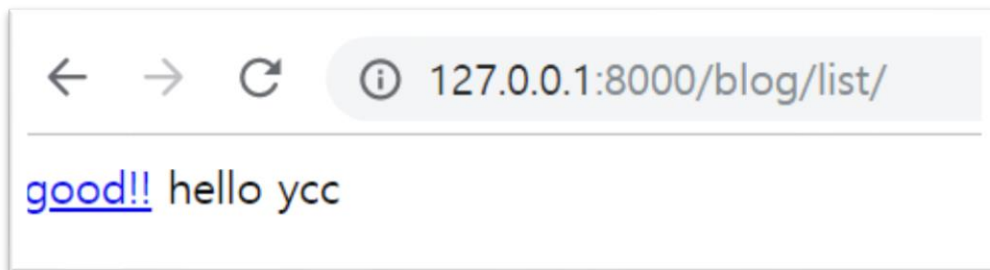


13. 관리자 모드

- 여전히 그대로
- List.html

```
admin.py  list.html
1  <a href = "http://127.0.0.1:8000/community/write">good!!</a>
2
3  {{post_list.0.title}}
```

- 127.0.0.1:8000/blog/list



13. 관리자 모드

■ 연습문제 1

- 관리자페이지에서 hello Yonsei 라는 타이틀의 포스트 하나 더 작성.
- 127.0.0.1:8000/blog/list에서 hello Yonsei 보이게 하기.

13. 관리자 모드

- 장고 템플릿 for문
 - 반복작업 간소화

```
3  {{post_list.0.title}}
4  {{post_list.1.title}}
5  {{post_list.2.title}}
6  {{post_list.3.title}}
7  {{post_list.4.title}}
```



```
11 {% for post in post_list %}
12     {{post.title}}
13 {% endfor %}
```

13. 관리자 모드

■ 연습문제 2

- models.py에서 Post 클래스의 title 밑에, content 라는 모델 하나 더 만들기
- Title과 Content를 127.0.0.1:8000/blog/list 에서 볼 수 있게 하기