민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ source myvenv/Scripts/activate

bash: myvenv/Scripts/activate: No such file or directory

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ python -m venv myvenv

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ source myvenv/Scripts/activate

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ pip install django

Collecting django

Using cached Django-3.2.3-py3-none-any.whl (7.9 MB)

Collecting pytz

Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)

Collecting sqlparse>=0.2.2

Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)

Collecting asgiref<4,>=3.3.2

Using cached asgiref-3.3.4-py3-none-any.whl (22 kB)

Installing collected packages: sqlparse, pytz, asgiref, django

Successfully installed asgiref-3.3.4 django-3.2.3 pytz-2021.1 sqlparse-0.4.1

WARNING: You are using pip version 21.1.1; however, version 21.1.2 is available.

You should consider upgrading via the 'c:\users\민용진\desktop\멋쟁이사자처럼9기\studywithlikelion\classlion\model실습\myvenv\scripts\python.exe -m pip install --upgrade pip' command.

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ django-admin startproject lionproject

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ ls

lionproject myvenv

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ cd lionproject/

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ ls

lionproject manage.py

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py startapp blog

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$

<models.py>

from django.db import models

# Create your models here.

class Blog(models.Model):

    title = models.CharField(max\_length=200)

    writer = models.CharField(max\_length=100)

    pub\_date = models.DateTimeField()

    body = models.TextField()

    def \_\_str\_\_(self):

        return self.title

class 만들면서 model에서 상속받음

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습 (master)

$ cd lionproject

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ ls

blog lionproject manage.py

(myvenv)

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py makemigrations

Migrations for 'blog':

blog\migrations\0001\_initial.py

- Create model Blog

🡪 앱 내의 migration 폴더를 만들어서 models.py의 변경사항 저장

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py migrate

Operations to perform:

Apply all migrations: admin, auth, blog, contenttypes, sessions

Running migrations:

Applying blog.0001\_initial... OK

* Migration 폴더를 실행시켜 데이터베이스에 적용

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

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

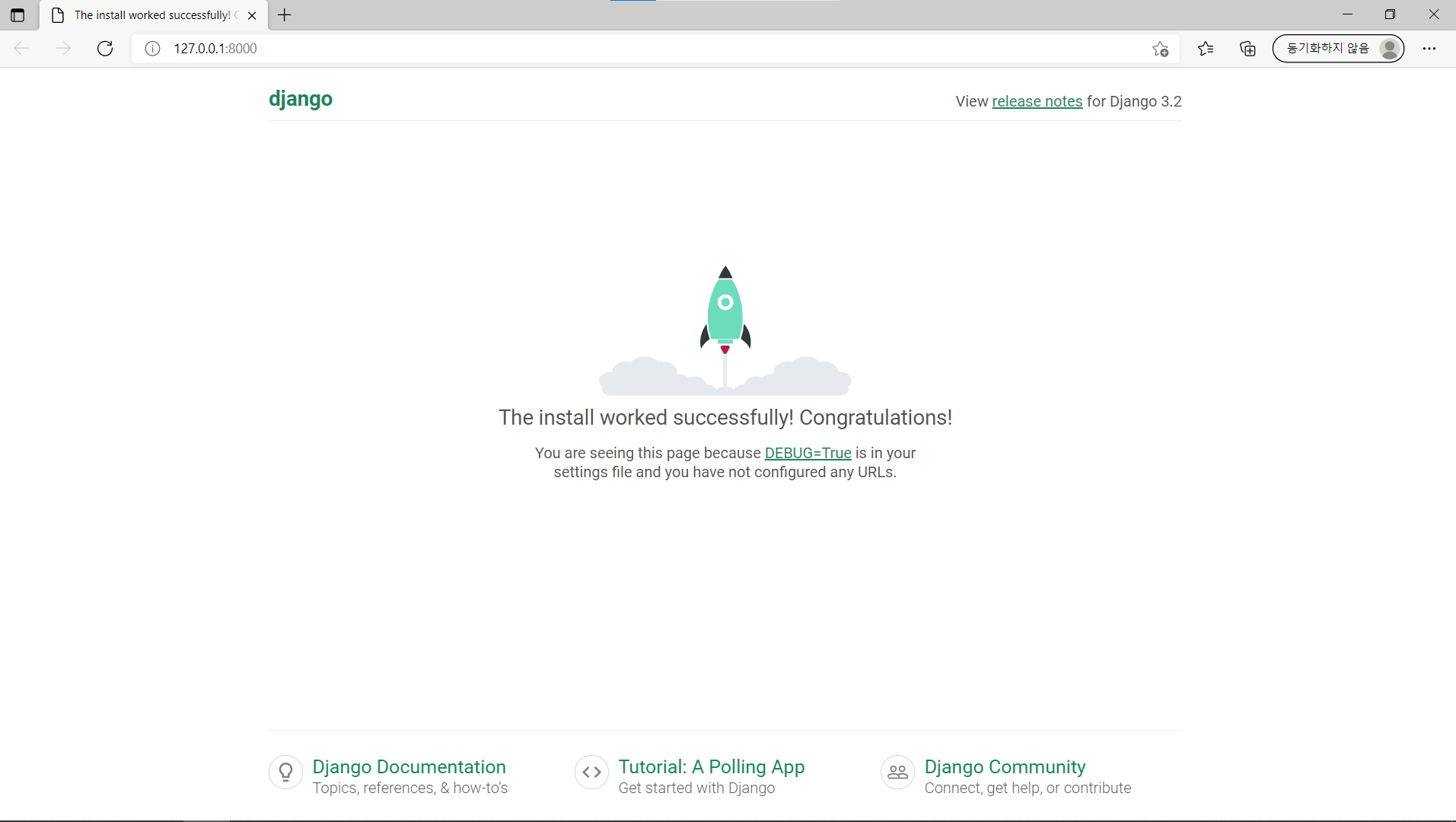
System check identified no issues (0 silenced).

June 01, 2021 - 19:52:54

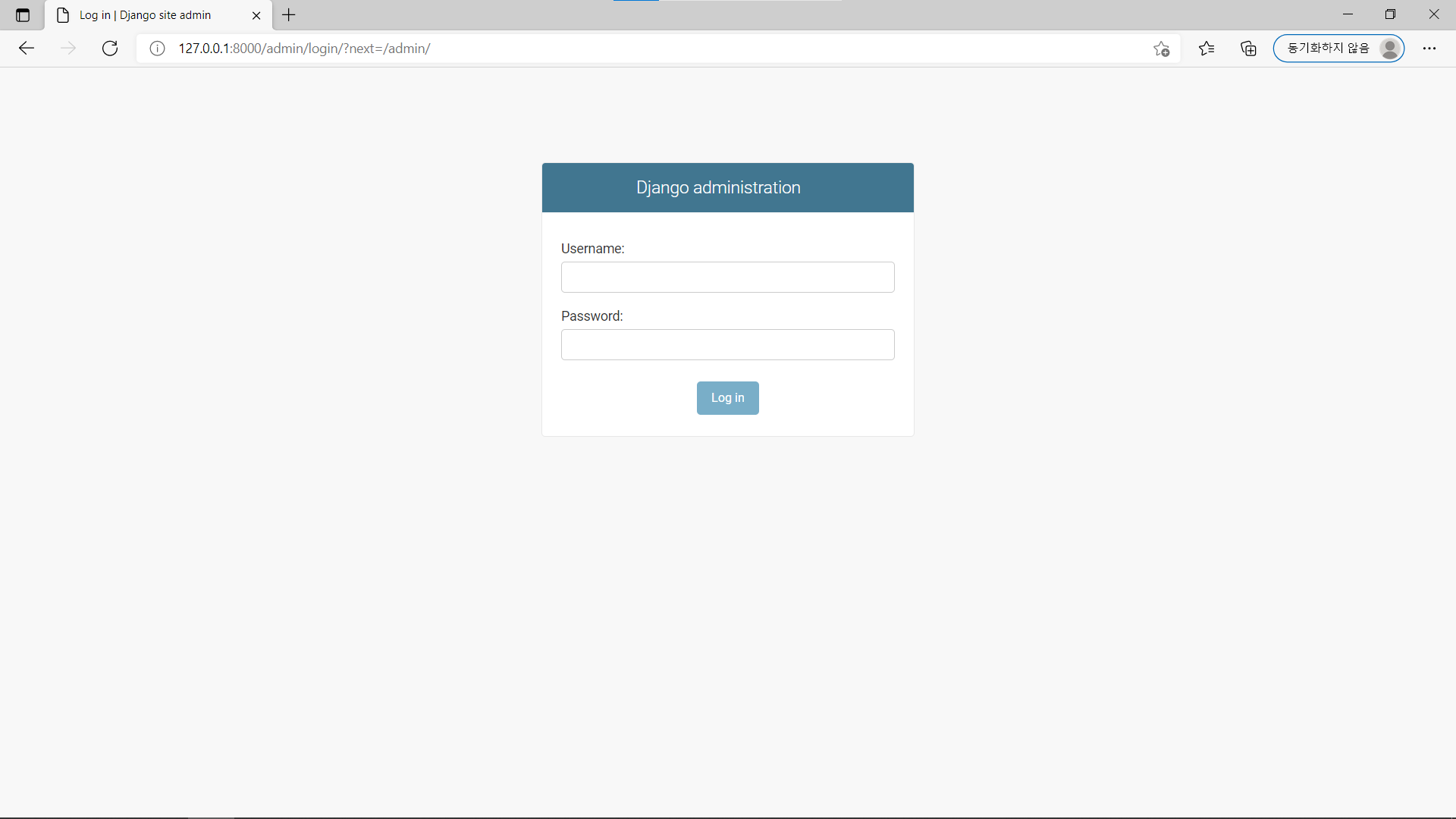
Django version 3.2.3, using settings 'lionproject.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.



/admin 치면



민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py createsuperuser

Username: sss

Email address: ss@likelion.org

Password:

Password (again):

This password is too short. It must contain at least 8 characters.

This password is too common.

This password is entirely numeric.

Bypass password validation and create user anyway? [y/N]: y

Superuser created successfully.

민선아@DESKTOP-0MVMCF2 MINGW64 ~/Desktop/멋쟁이사자처럼9기/studywithLikeLion/Classlion/Model실습/lionproject (master)

$ python manage.py runserver

Watching for file changes with StatReloader

Performing system checks...

System check identified no issues (0 silenced).

June 02, 2021 - 00:02:21

Django version 3.2.3, using settings 'lionproject.settings'

Starting development server at http://127.0.0.1:8000/

Quit the server with CTRL-BREAK.

[02/Jun/2021 00:02:25] "GET / HTTP/1.1" 200 10697

[02/Jun/2021 00:02:34] "GET /admin/ HTTP/1.1" 302 0

[02/Jun/2021 00:02:34] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 2214

[02/Jun/2021 00:02:34] "GET /admin/ HTTP/1.1" 302 0

[02/Jun/2021 00:02:34] "GET /admin/login/?next=/admin/ HTTP/1.1" 200 2214

[02/Jun/2021 00:02:39] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0

[02/Jun/2021 00:02:39] "GET /admin/ HTTP/1.1" 200 3324

[02/Jun/2021 00:02:40] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 380

[02/Jun/2021 00:02:40] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331

[02/Jun/2021 00:02:40] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380

<admin.py>

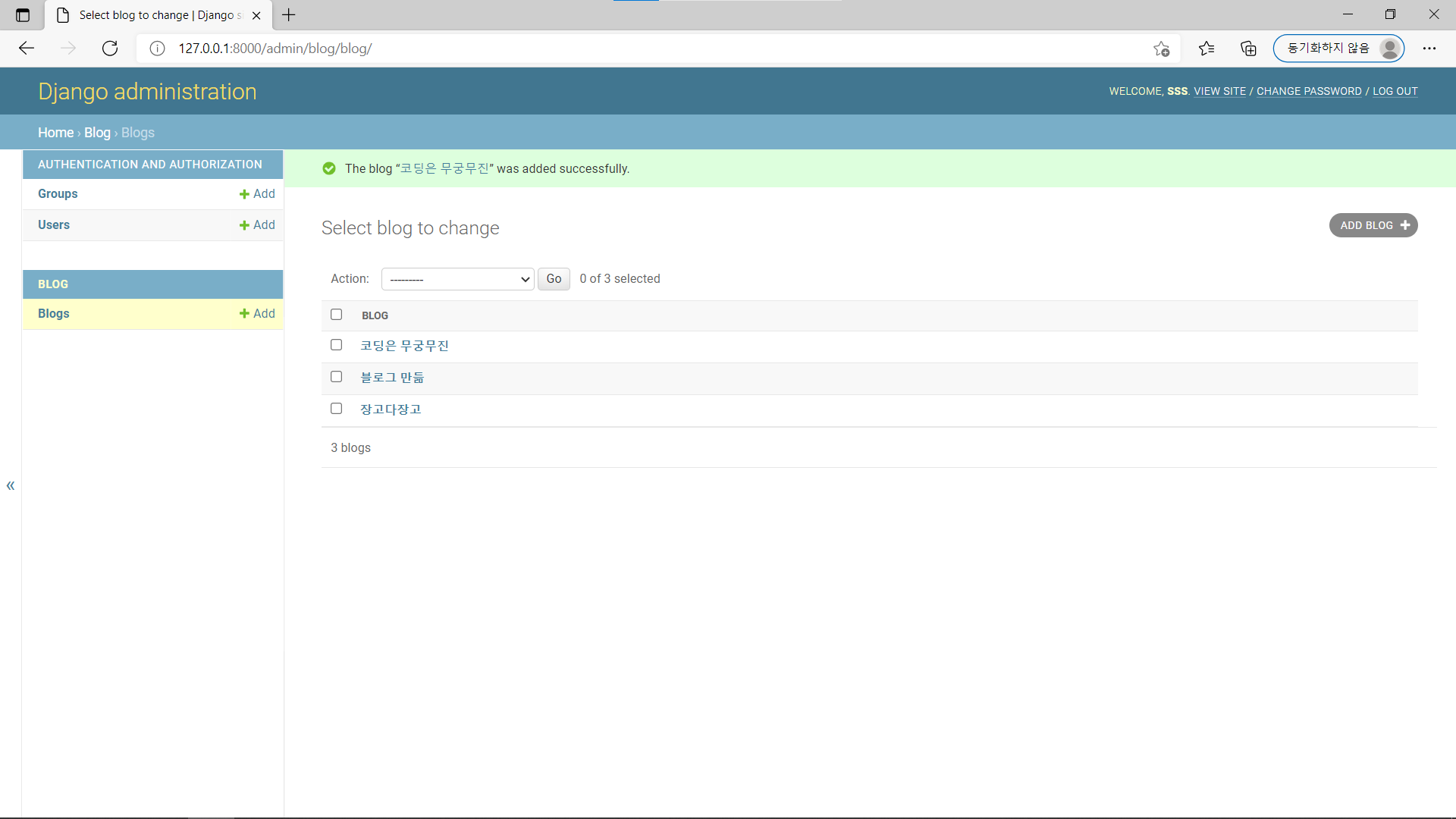
from django.contrib import admin

from .models import Blog

# Register your models here.

admin.site.register(Blog)

* Admin 사이트에 등록해줘야함
* 다시 웹 페이지에서 새로고침



<models.py>

    def \_\_str\_\_(self):

        return self.title

글 제목 보이게 override 하는 것

CRUD = Create, Read, Update, Delete

**READ**

from django.shortcuts import render

from .models import Blog

# Create your views here.

def home(request):

    blogs = Blog.objects.all()

    return render(request, 'home.html',{'blogs':blogs})

<urls.py>

from django.contrib import admin

from django.urls import path

from blog.views import \*

urlpatterns = [

    path('admin/', admin.site.urls),

    path('',home,name="home")

]

<home.html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>BLOG</title>

</head>

<body>

    <h1>Blog Project</h1>

    {{blogs}}

</body>

</html>

Views.py에 만들어놨던 key값 {{blogs}} 사용

Views.py urls.py home.html templates파일 다 만들었으면

Python manage.py runserver 🡪 runserver 하면 됨!!

<home.html> css로 조금 꾸미는 과정

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>BLOG</title>

    <style>

        body{text-align: center;}

    </style>

</head>

<body>

    <h1>Blog Project</h1>

    <div class="container">

    {% for blog in blogs %}

        <div>

        <h3> {{blog.title}}</h3>

        {{blog.writer}}

        {{blog.body}}

        <br>

        </div>

    {% endfor %}

    </div>

</body>

</html>

<models.py>

본문이 너무 길 때 첫화면에 뜨지 않게 summary하는 함수 기능

    def summary(self):

        return self.body[:100]

100번째 index까지 슬래싱 slashing

<Path Converter>

Detail 상세페이지 만들기

Details.html