7. media 파일 관리

media: file 또는 imagefield를 통해 저장한 모든 파일

1) 파일 처리 순서

httprequest.FILES를 통해 파일 전달 -> 뷰, 폼을 통해 유효성 검사 수행

-> FileField/imageFiled에 경로 저장 -> MEDIA\_ROOT 경로에 파일 저장

2) file field 모델링

settings.py: Media 파일 관련값 설정

MEDIA\_URL = '/media/'   # 각 media 파일에 대한 url prefix

MEDIA\_ROOT = os.path.join(BASE\_DIR,'media') # 파일 필드를 통한 저장시, 파일을 저장할 root 경로

app이름/urls.py: 파일 불러올 때의 urls.py 설정

from django.conf import settings

from django.conf.urls.static import static

if settings.DEBUG:

    urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

$ pip install pillow: 라이브러리 설치

app이름/admin.py: admin에서 사진 미리보기 및 크기 지정

    from django.utils.safestring import mark\_safe

    # 사진 미리보기

    def photo\_tag(self, post):

        if post.photo:

            return mark\_safe( f'<img src="{post.photo.url}" style = "width:75px;" />')

        return None

프로젝트명/utils.py: 파일 생성 후 uuid를 통해 경로와 고유한 파일명을 반환하는 모듈 생성

import os

from uuid import uuid4

from django.utils import timezone

# uuid를 통한 파일명 정하기

def uuid\_name\_upload\_to(instance, filename):

    app\_label = instance.\_\_class\_\_.\_meta.app\_label  # 앱 별로

    cls\_name = instance.\_\_class\_\_.\_\_name\_\_.lower()  # 모델 별로

    ymd\_path = timezone.now().strftime('%Y/%m/%d')  # 업로드 하는 년/월/일 뱌럴

    uuid\_name = uuid4().hex

    extension = os.path.splitext(filename)[-1].lower()  # 확장자 추출 및 소문자 변환

    return '/'.join([

        app\_label,

        cls\_name,

        ymd\_path,

        uuid\_name[:2],

        uuid\_name + extension,

    ])

models.py: Imagefield 추가 및 업로드 경로에 uuid 모듈 적용

    from myproject.utils import uuid\_name\_upload\_to

    # uuid를 통한 중복방지 파일명 지정

    photo = models.ImageField(blank = True, upload\_to=uuid\_name\_upload\_to, max\_length=255)

$ python manage.py makemigrations [앱이름]

$ python manage.py migrate [앱이름]

**8. static 파일관리**

static: html, css, image 등의 정적인 파일