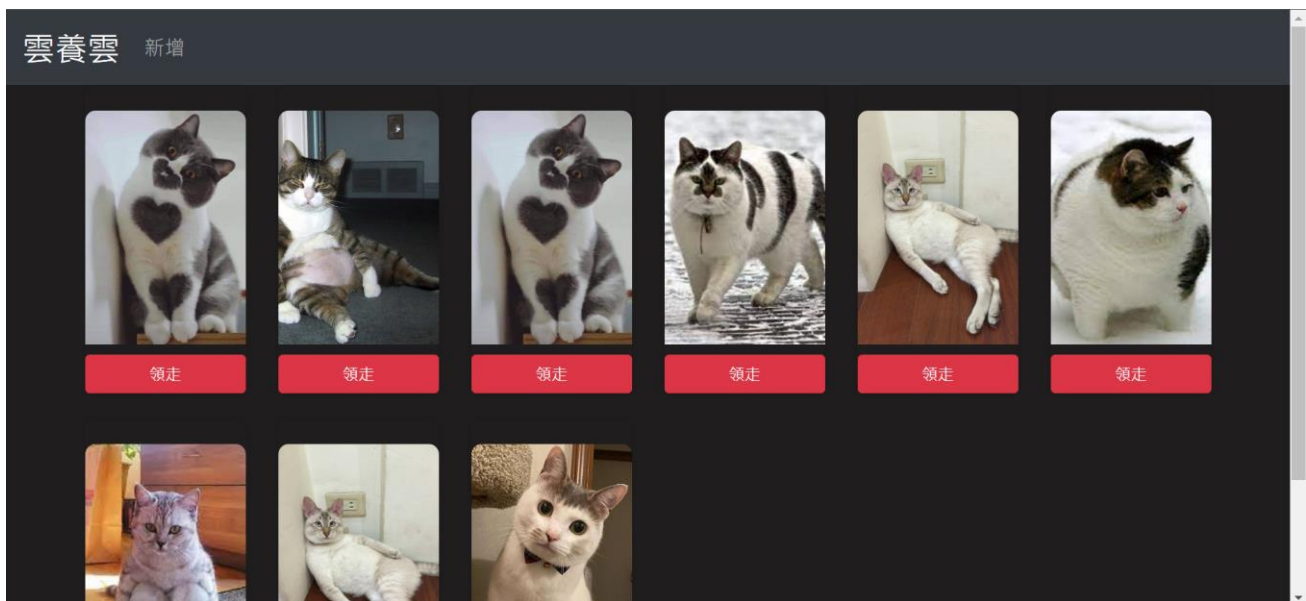


# 期末專案

- 雲養雲圖片共享系統 -



指導老師：許子衡

學生：資工四乙 4A7G0090 尤品淇

## 一、摘要 (Abstract)

當你手機圖片太多，捨不得刪掉圖片，卻又不知道如何是好時候怎麼辦呢？是時候將圖片給需要的人了，上傳你不要的照片！採開放式網站秉持著人性本善，讓大家可以下載自己需要的圖片～

## 二、介紹 (Introduction)

簡單明瞭的操作介面，上傳你多餘的照片吧！

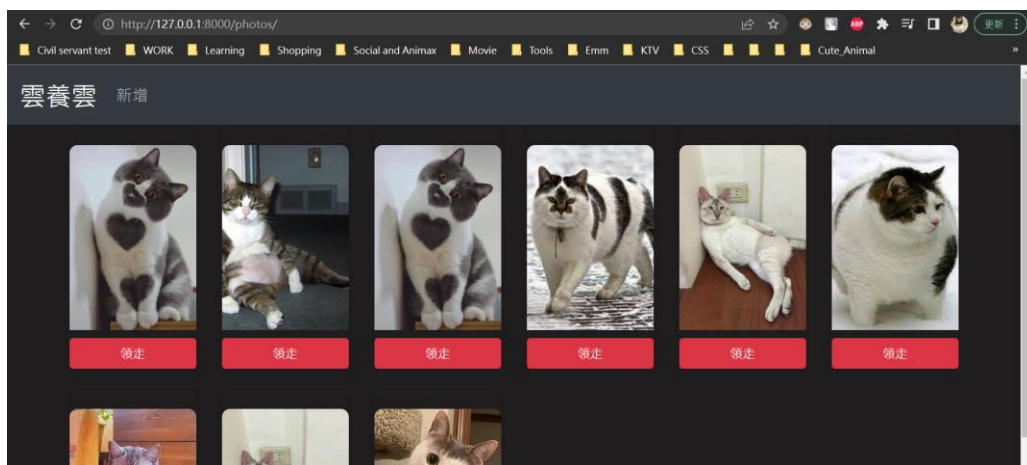
## 三、本文

### (1) 題目動機

手機裡的垃圾圖片太多了，但可能自己不需要的圖片，會是別人珍藏的新圖片。

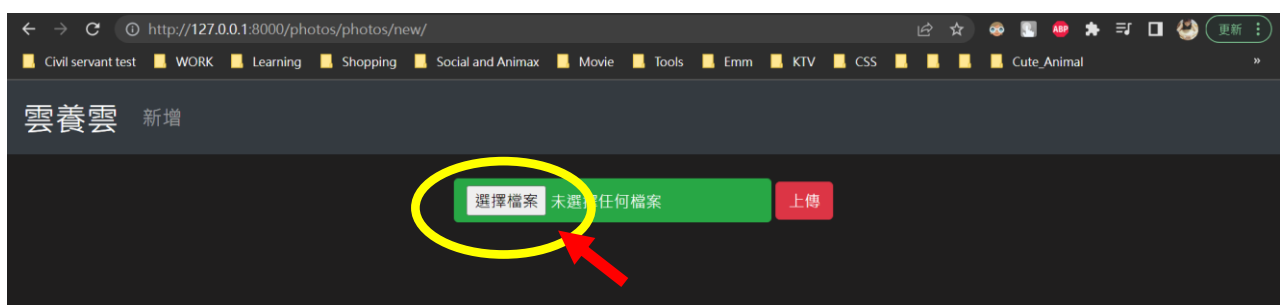
### (2) 畫面說明

#### 1. 主頁面 (<http://127.0.0.1:8000/photos/>)。

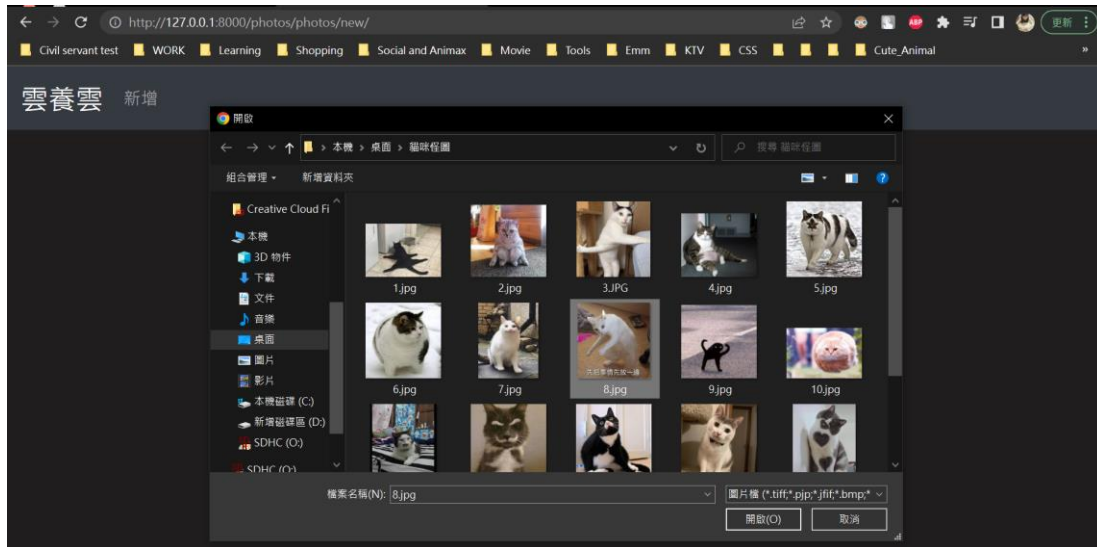


#### 2. 新增圖片頁面 (<http://127.0.0.1:8000/photos/photos/new/>)。

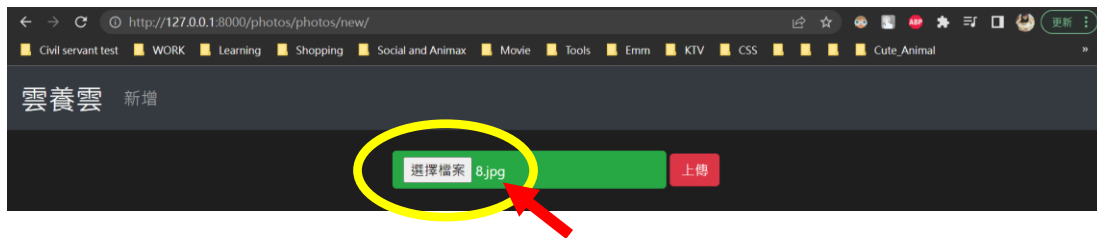
##### a. 點擊選擇檔案按鈕。



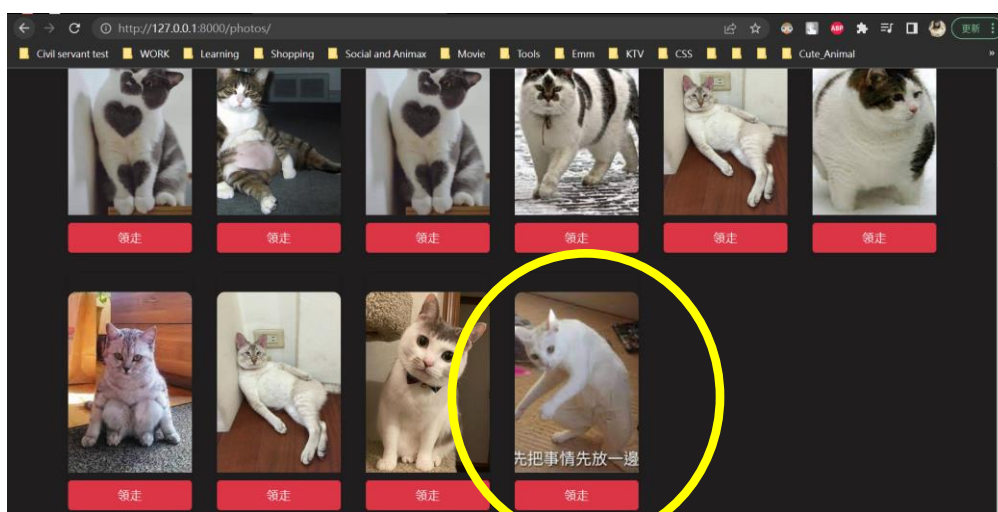
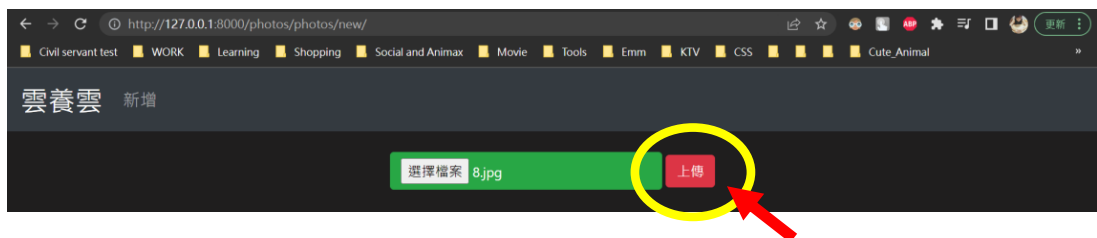
b. 選擇要上傳的圖片。



c. 選擇後顯示圖片檔名確認。

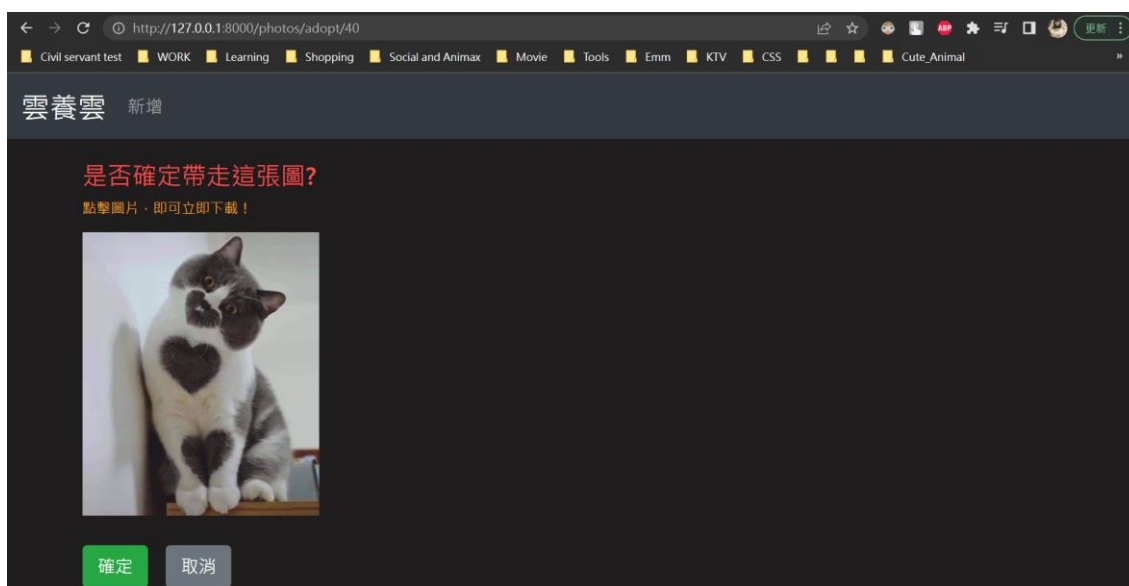


d. 點擊後成功新增圖片，並返回主頁面。

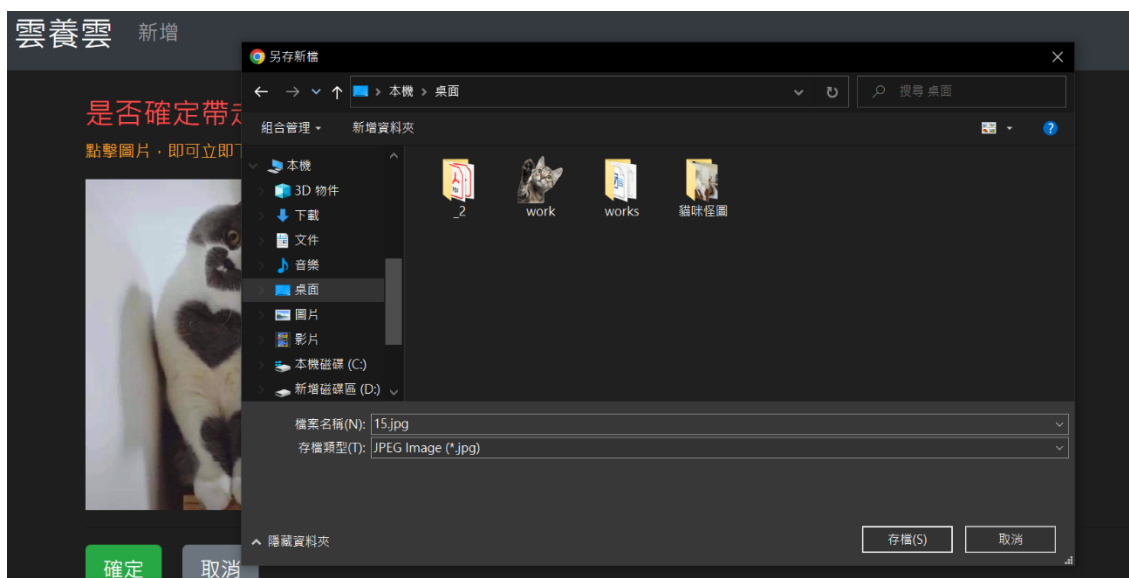


3. 領圖頁面 (<http://127.0.0.1:8000/photos/adopt/40>)。

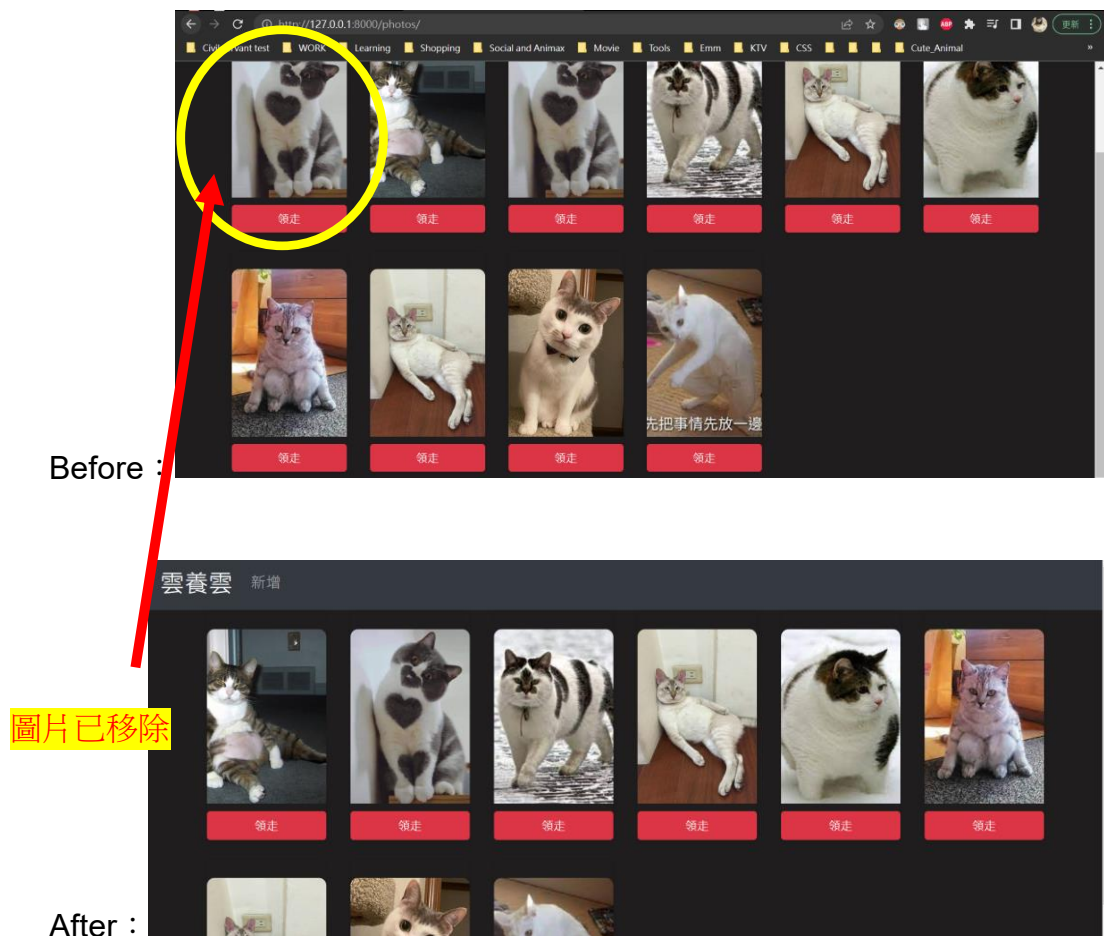
a. 與使用者確認是否要帶走這張圖。



b. 點擊貓咪圖片任何一處，皆可以下載圖片。



c. 點擊後，原本的圖片就會從動態資料庫內移除，找到新歸屬。

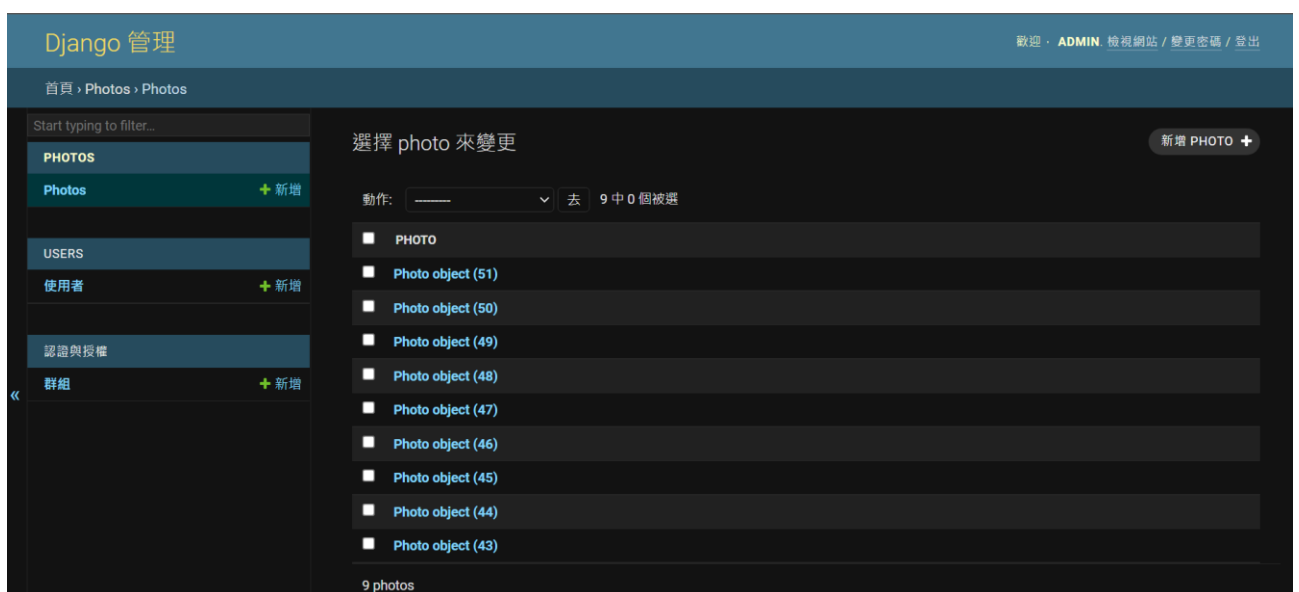


4. 後臺可以看到目前圖片與歷史有多少圖片上去，又被帶走了。(看編號)

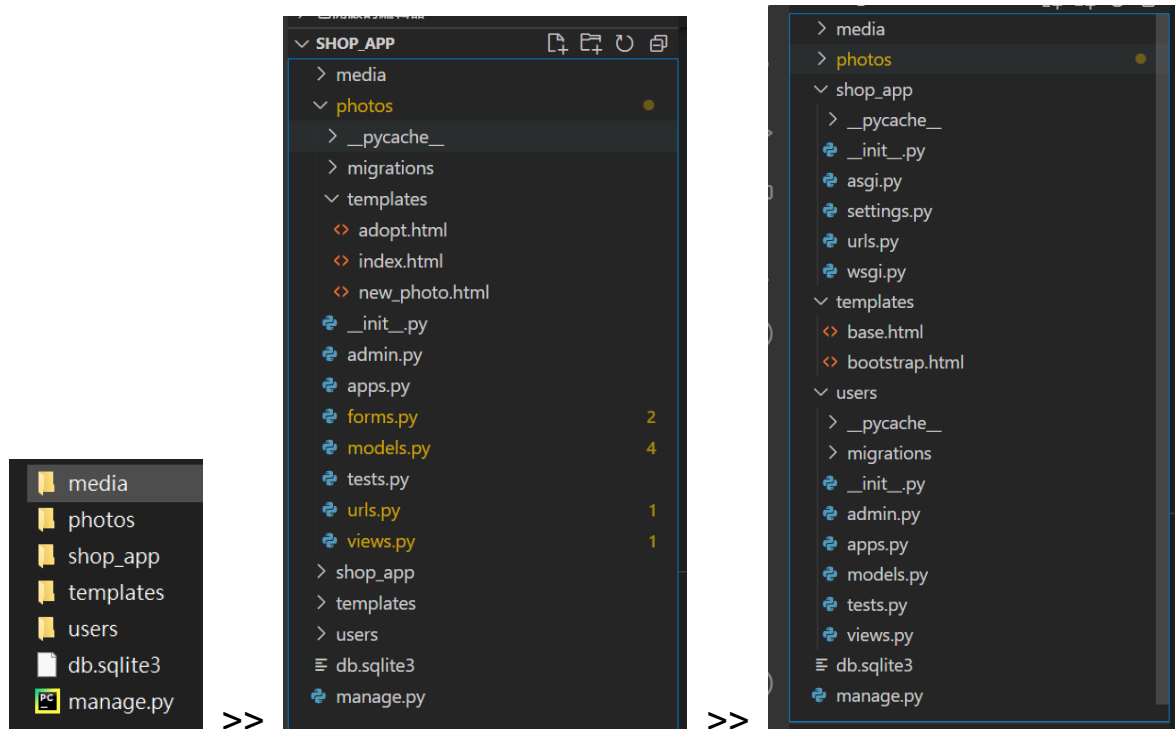
使用者名稱：admin

電子信箱：[4A7G0090@stust.edu.tw](mailto:4A7G0090@stust.edu.tw)

Password：a1234567



### (3) 實作程式架構



採 MVC 架構，”上傳”按鈕(Controller)按下後，會傳送至模型(Model)並存入資料庫(DB)，存入後回到主頁面(View)，主頁面(View)再從模型(Model)中取出資料庫(DB)呈現，故主頁面(View)會新增剛剛所上傳的新圖片，同理於領走圖片中進行刪除資料庫圖片同理。

### (4) 程式輸出

1. 於 models.py 中建立 Photo 資料庫並新增 Image 圖片欄位，參數 upload\_to 設定圖片上傳後所存放的路徑，參數 blank 和 null 為 False 代表欄位是不允許空值。

```
models.py 4 X  urls.py 1  views.py 1  adopt.html  new_photo.html  inc
photos > models.py > Photo
1  from django.db import models
2  from django.utils import timezone
3
4  class Photo(models.Model):
5      image = models.ImageField(upload_to='images/', blank=False, null=False)
6
7      upload_date = models.DateField(default=timezone.now)
```



2. 於 form.py 中建立可以上傳圖片的表單，設定 HTTP5 的輸入類型。

```
forms.py 2 X
photos > forms.py > UploadModelForm > Meta
1 from django import forms
2 from .models import Photo
3
4 class UploadModelForm(forms.ModelForm):
5     class Meta:
6         model = Photo
7         fields = ('image',)
8         widgets = {
9             'image': forms.FileInput(attrs={'class': 'form-control-file'}),
10            'form': forms.DateInput
11        }
```

3. 新增 photos/urls.py 的檔案，加入 photos 本地路由設定。  
(adopt 為領走圖片畫面的路由指向 views.py 處理 adopt 的類別，photos/new 為新增圖片畫面的路由指向 views.py 處理 new\_photo 的類別。)

```
urls.py 1 X
photos > urls.py > ...
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name='index'),
6     path('adopt/<str:pk>', views.adopt, name='adopt'),
7     path('photos/new/', views.new_photo, name='new_photo')
8 ]
```

4. 修改 photos/views.py 檔案，建立相關 view 處理類別。
1. adopt 類別部分：點擊按鈕後會使用 POST 請求方法呼叫檢視 adopt 函式(View Function)，接著利用傳送過來的 POST 資料取得 pk 的 id 編號，找出 photos 內物件，通過驗證就把資料從資料庫中刪除，最後返回主頁面。
  2. New\_photo 類別部分：點擊按鈕後會使用 POST 請求方法呼叫檢視 new\_photo 函式(View Function)，接著利用傳送過來的 POST 資料及檔案來建立資料模型表單(ModelForm)物件，通過驗證就把資料儲存至資料庫中，最後返回主頁面。

```
views.py 1 X
photos > views.py > adopt
1 from django.shortcuts import render, redirect
2 from .forms import UploadModelForm
3 from .models import Photo
4
5 def index(request):
6     photos = Photo.objects.all()
7     form = UploadModelForm()
8
9     context = {
10         'photos': photos,
11         'form': form
12     }
13
14     return render(request, 'index.html', context)
15
16 def adopt(request, pk):
17     photos = Photo.objects.get(id=pk)
18
19     if request.method == "POST":
20         photos.delete()
21         return redirect('/photos')
22
23     context = {
24         'photos': photos
25     }
26
27     return render(request, 'adopt.html', context)
28
29 def new_photo(request):
30     photos = Photo.objects.all()
31     form = UploadModelForm()
32
33     if request.method == "POST":
34         form = UploadModelForm(request.POST, request.FILES)
35         if form.is_valid():
36             form.save()
37             return redirect('/photos')
38
39     context = {
40         'photos': photos,
41         'form': form
42     }
43
44     return render(request, 'new_photo.html', context)
```

5. 修改 `settings.py` 設定，`MEDIA` 兩行程式代表上傳圖片後，就會自動建立 `media` 的資料夾，並在底下建立資料模型中所定義的路徑(`image/`)，來存放圖片。

```

3
4  ✓ INSTALLED_APPS = [
5      'django.contrib.admin',
6      'django.contrib.auth',
7      'django.contrib.contenttypes',
8      'django.contrib.sessions',
9      'django.contrib.messages',
10     'django.contrib.staticfiles',
11     'users',
12     'photos',
13 ]
14
15 AUTH_USER_MODEL = 'users.CustomUser'
16
17 STATIC_URL = '/static/'
18 STATIC_ROOT = os.path.join(BASE_DIR, 'static')
19
20 MEDIA_URL = '/media/'
21 MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
22

```

6. 建立樣版基底檔案 `photos/templates/base.html`，新增 `css` 樣式與 `nav` 設定

```

1  <!doctype html>
2  <html>
3  <head>
4      <!-- Required meta tags -->
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width,
7
8      <!-- Bootstrap CSS -->
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
10
11      <title>雲養雲</title>
12
13  <style>
14      p {
15          color: #ffa32a;
16      }
17      h3 {
18          color: rgb(252,82,82);
19      }
20      .flex{
21          display:flex;
22          width:100%;
23      }
24
25      .flex div{
26          margin:auto;
27      }
28  </style>

```

```
</style>
</head>
<body style="background-color: #f1f3f4;">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <a class="navbar-brand" href="{% url 'index' %}" style="font-size: 32px;">雲雲雲</a>
  <button class="navbar-toggler" type="button"
    data-toggle="collapse" data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item">
        <a class="nav-link" href="{% url 'new_photo' %}" style="font-size: 28px;">新增</a>
      </li>
    </ul>
  </div>
</nav>

{% block upload %}
{% endblock %}
```

7. 建立樣版檔案 `photos/templates/adopt.html` (左圖)和 `new_photo.html` (右圖)，`Adopt.html` 的 `photos.image.url` 跟操作者確認圖片是否要這張，且可以點擊外層包裹的圖片連結進行下載，`New_photo.html` 的 `form.image` 是從先前的表單 `form.py` 中設定好的 `HTTP5` 性質可以直接建立物件。

adopt.html X

photos > templates > adopt.html > ...  
1 {% extends 'base.html' %}  
2  
3 {% block content %}  
4  
5 <br>  
6 <h3>是否確定帶走這張圖?</h3>  
7 <p>點擊圖片，即可立即下載！</p>  
8  
9 <form action="" method="POST">  
10 {% csrf\_token %}  
11 <div class="center">  
12 <a href="{% photos.image.url %}" download="圖片">  
13   
14 </a>  
15 <br/>  
16 <hr/>  
17 <input class="btn btn-success btn-lg" type="submit" value="確定" style="margin-right: .8em;"/>  
18 <a class="btn btn-secondary btn-lg" href="{% url 'index' %}">取消</a>  
19 </form>  
20  
21 {% endblock %}

new\_photo.html X

photos > templates > new\_photo.html > form  
1 {% extends 'base.html' %}  
2  
3 {% block upload %}  
4  
5 <br>  
6 <form action="" method="POST" enctype="multipart/form-data">  
7 {% csrf\_token %}  
8 <div class="flex">  
9 <div>  
10 <a href="" class="btn btn-success">{{ form.image }}</a>  
11 <input type="submit" class="btn btn-danger" value="上傳">  
12 </div>  
13 </div>  
14 </form>  
15  
16 {% endblock %}



8. 建立樣版檔案 `photos/templates/index.html`，顯示資料庫內目前有的圖片，從 `photos` 陣列中一個一個取出來顯示。

```
index.html X
photos > templates > index.html > ...
1  {% extends 'base.html' %}
2
3  <br />
4
5  {% block content %}
6
7  <div class="row">
8      {% for photo in photos %}
9
10         {% if photo.image %}
11             <div class="col-md-2">
12                 <div class="card mb-4 shadow-sm" style="border-radius: 10px; background-color: #fff;">
13                     <br>
14                     
19                     <a href="{% url 'adopt' photo.id %}" class="btn btn-danger">領走</a>
20                 </div>
21             </div>
22         {% endif %}
23     {% endfor %}
24 </div>
25
26 {% endblock %}
```

## 四、結論 (Conclusions) + 學習心得

過程中更加了解了什麼是 Django 架構，也學習到了上傳圖片的方法，對於資料庫操作、刪除跟 MVC 架構也更加地了解了。額外還有了解到了很多時候出問題都再 Setting 或路由沒有設定好，而導致一直出現問題。

## 五、參考資料 (References)

[1] [Django 教學 9]6 個步驟搞懂 Django 上傳圖片的功能

<https://www.learncodewithmike.com/2020/04/django-image-upload.html>