
# KENDRIYA VIDYALAYA - IIT, CHENNAI

# 2022 – 2023

# COMPUTER SCIENCE PROJECT

# PHOTO SHARE

## *TEAM MEMBERS*

1. Siddid Soni
2. Sreeshanth Ryali

# CERTIFICATE

This is to certify that Sreeshanth, a student of class XII B has successfully completed the investigatory project “Photo Share” under the guidance of Mrs. Nasreen Salma R (PGT Computer Science) during the academic year 2022-2023 in the partial fulfillment of Computer Science Practical Examination conducted by CBSE.

*Sign of Internal Examiner* *Sign of External Examiner*

*Sign of Principal* *School Seal*

# ACKNOWLEDGEMENT

I would like to express deep sense of gratitude to Mrs. Nasreen Salma R, PGT – Computer Science for guiding me through the course of the project.

I would also like to express my sincere thanks to Dr. M. Manickasamy, Principal, KV – IIT, Chennai for his constant motivation which helped me in completion of the project with great ease.

I extend my sincere gratitude to my parents and my friends for their timely help and support for the completion of the project.

# INDEX

# INTRODUCTION

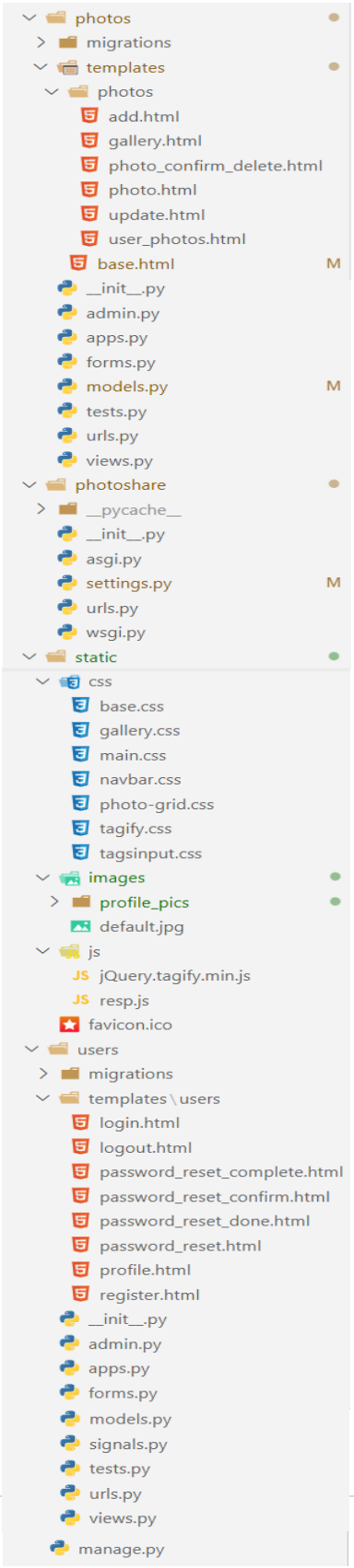
Photo sharing websites allow users to post their images to share with their online buddies or even the entire web, as opposed to people publishing pictures in hard copy and exhibiting to friends over a cup of tea. Users are free to publish anything, regardless of the camera or device that took the photo.

# DATABASE DESIGN

# MODULES USED

- django
- django-crispy-forms
- django-taggit
- pillow
- mysqlclient

# File structure



# SOURCE CODE

## Photoshare>photos>admin.py

```
from django.contrib import admin

# Register your models here.
from .models import Photo

admin.site.register(Photo)
```

## Photoshare>photos>apps.py

```
from django.apps import AppConfig


class PhotosConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'photos'
```

## Photoshare>photos>models.py

```
from django.db import models
from django.contrib.auth.models import User
from django.utils import timezone
from django.urls import reverse
from taggit.managers import TaggableManager


# Create your models here

class Photo(models.Model):
    image=models.ImageField(null=False, blank=False)
    description=models.TextField()
    date_posted=models.DateTimeField(default=timezone.now, null=False,
blank=False)
    auther=models.ForeignKey(User, on_delete=models.SET_NULL, null=True,
blank=True)
    tags = TaggableManager(blank=False)
    is_private=models.BooleanField(default=False,null=False)

    def __str__(self):
        return self.description
```

```
    def get_absolute_url(self):
        return reverse("photo", kwargs={"pk": self.pk})
```

## Photoshare>photos>urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.Gallery.as_view(), name="gallery"),
    path('user/<str:username>/', views.UserGallery.as_view(), name='user-
gallery'),
    path('photo/<int:pk>/update/', views.PhotoUpdateView.as_view(),
name='update'),
    path('photo/<int:pk>/delete/', views.PhotoDeleteView.as_view(),
name='delete'),
    path('photo/<int:pk>/', views.ViewPhoto.as_view(), name="photo"),
    path('add/', views.AddPhoto.as_view(), name="add"),
```

## Photoshare>photos>forms.py

```
from django.forms import ModelForm, Textarea, Select, FileInput,
ValidationError, CheckboxInput
from .models import Photo
from taggit.forms import TagWidget

class Upload(ModelForm):
    image = FileInput(attrs={'class': 'form-select border-secondary' ,'type':
'file', 'required': True})
    class Meta:
        model = Photo
        fields = ['description', 'image', 'tags', 'is_private']
        widgets = {
                    'description': Textarea(
                          attrs={'class': 'txt form-control border-secondary',
                          'placeholder': 'enter description',
                          'id': 'validationTextarea','rows': 5,
                          'required': True}),


                    'image': FileInput(
                        attrs={'class': 'txt form-control border-secondary',
                        'type': 'file', 'required': True}),
```

```
                    'tags': TagWidget(
                        attrs={'class': 'txt form-control border-secondary',
                        'data-role' : "tagsinput"}),
                    'is_private': CheckboxInput(
                        attrs={'class': 'form-check-input',
                        'role': 'switch', 'id': 'flexSwitchCheckDefault'}
                    )
                }

    def clean_image(self):
        data = self.cleaned_data.get("image")
        if data.size > 5242880:
            raise ValidationError("file too large...!(>5MB)")
        return data

    def clean_tags(self):
        data = self.cleaned_data["tags"]
        data.remove(':')
        data.remove('[{')
        data.remove('}]')
        data.remove('value')
        if '{' in data:
            data.remove('{')
            data.remove('}')
        if len(data) < 3:
            raise ValidationError("atleast three tags required!!")
        return data



class Update(ModelForm):
    class Meta:
        model = Photo
        fields = ['description', 'tags', 'is_private']
        widgets = {
                    'description': Textarea(
                        attrs={'class': 'txt form-control border-secondary',
'placeholder': 'enter description', 'id': 'validationTextarea','rows': 5,
'required': True}),
                    'category': Select(
                        attrs={'class': 'txt form-select border-secondary',
'aria-label': "Default select example"}),
                    'tags': TagWidget(
                        attrs={'class': 'txt form-control border-secondary',
'data-role' : "tagsinput"}),
```

```
                    'is_private': CheckboxInput(
                        attrs={'class': 'form-check-input', 'role': 'switch',
'id': 'flexSwitchCheckDefault'}
                    )
                    }

    def clean_tags(self):
        data = self.cleaned_data["tags"]
        data.remove(':')
        data.remove('[{')
        data.remove('}]')
        data.remove('value')
        if '{' in data:
            data.remove('{')
            data.remove('}')
        if len(data) < 3:
            raise ValidationError("atleast three tags required!!")
        return data
```

## Photoshare>photos>views.py

```
import os
from django.http import request
from django.shortcuts import redirect, get_object_or_404
from django.urls.base import reverse
from .models import Photo
from .forms import Upload, Update
from django.contrib.auth.mixins import LoginRequiredMixin, UserPassesTestMixin
from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView
from django.contrib.auth.models import User
from django.urls import reverse_lazy
from taggit.models import Tag
import binascii

# Create your views here.

class ViewPhoto(UserPassesTestMixin, DetailView):
    model = Photo
    template_name = 'photos/photo.html'

    def get_context_data(self, **kwargs):
        context = super(ViewPhoto, self).get_context_data(**kwargs)
        prev_url = self.request.META.get('HTTP_REFERER')
        if prev_url is None:
            prev_url = reverse('gallery')
        if prev_url == self.request.build_absolute_uri('update') + '/':
```

```
            prev_url = reverse('gallery')
        context['prev_url'] = prev_url
        return context

    def test_func(self):
        photo = self.get_object()
        if photo.is_private:
            if self.request.user == photo.auther:
                return True
            else:
                return False
        else:
            return True


class Gallery(ListView):
    model = Photo
    template_name = 'photos/gallery.html'
    context_object_name = 'photos'
    paginate_by = 20

    def get_context_data(self, *, object_list=None, **kwargs):
        context = super(Gallery, self).get_context_data(**kwargs)
        _tag = self.request.GET.get('tag') or ''
        search_tag = self.request.GET.get('tags') or ''
        if _tag:
            tag = get_object_or_404(Tag, slug=_tag)
            context['is_tag'] = True
            context['tag'] = tag.name
        context['search_input'] = search_tag
        return context

    def get_queryset(self):
        _tag = self.request.GET.get('tag') or ''
        _search_tag = self.request.GET.get('tags') or ''
        lst = _search_tag.split()
        if _tag:
            qs = []
            for i in Photo.objects.filter(tags__slug=_tag):
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)

            return qs
```

```
        if _search_tag:
            qs = []
            for i in Photo.objects.filter(tags__name__in=lst).distinct():
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)
            return qs

        else:
            qs = []
            for i in Photo.objects.all():
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)
            return qs


class UserGallery(ListView):
    model = Photo
    template_name = 'photos/user_photos.html'
    context_object_name = 'photos'
    paginate_by = 20

    def get_context_data(self, *, object_list=None, **kwargs):
        context = super(UserGallery, self).get_context_data(**kwargs)
        _tag = self.request.GET.get('tag') or ''
        search_tag = self.request.GET.get('tags') or ''
        if _tag:
            tag = get_object_or_404(Tag, slug=_tag)
            context['is_tag'] = True
            context['tag'] = tag.name
            context['utag'] = 'Tagged by ' + tag.name
        context['search_input'] = search_tag
        return context

    def get_queryset(self):
        user = get_object_or_404(User, username=self.kwargs.get('username'))
        _tag = self.request.GET.get('tag') or ''
        _search_tag = self.request.GET.get('tags') or ''
        lst = _search_tag.split()
        if _tag:
```

```
            qs = []
            for i in Photo.objects.filter(auther=user, tags__slug=_tag).order_by('-
date_posted'):
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)
            return qs

        if _search_tag:
            qs = []
            for i in Photo.objects.filter(auther=user,
tags__name__in=lst).distinct().order_by('-date_posted'):
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)
            return qs

        else:
            qs = []
            for i in Photo.objects.filter(auther=user).order_by('-date_posted'):
                if i.is_private:
                    if i.auther == self.request.user:
                        qs.append(i)
                else:
                    qs.append(i)
            return qs


"""
@login_required
def addPhoto(request):
    if request.method == 'POST':
        form=Upload(request.POST, request.FILES)
        if form.is_valid():
            Photo.auther = request.user
            form.save()
            return redirect('gallery')
    else:
        form=Upload()
    context={
        'form':form
    }
```

```
    return render(request, 'photos/add.html', context)
"""


class AddPhoto(LoginRequiredMixin, CreateView):
    model = Photo
    template_name = 'photos/add.html'
    form_class = Upload
    success_url = reverse_lazy('gallery')

    def form_valid(self, form):
        form.instance.auther = self.request.user
        salt = str(binascii.hexlify(os.urandom(32)))
        lst = []
        for i in salt:
            lst.append(i)
        lst.remove('b')
        lst.remove("'")
        lst.remove("'")
        form.instance.image.name = ''.join(lst) + form.instance.image.name
        return super().form_valid(form)


class PhotoUpdateView(LoginRequiredMixin, UserPassesTestMixin, UpdateView):
    model = Photo
    template_name = 'photos/update.html'
    context_object_name = 'photos'
    form_class = Update

    def form_valid(self, form):
        form.instance.auther = self.request.user
        return super(PhotoUpdateView, self).form_valid(form)

    def test_func(self):
        post = self.get_object()
        if self.request.user == post.auther:
            return True
        return False


class PhotoDeleteView(LoginRequiredMixin, UserPassesTestMixin, DeleteView):
    model = Photo
    success_url = '/'

    def test_func(self):
        post = self.get_object()
```

```
        if self.request.user == post.auther:
            return True
        return False
```

## Photoshare>templates>base.html

```
{% load static %}
<!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>{% block title %}{% endblock %}</title>
    <link rel="icon" href="{% static 'favicon.ico' %}">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU" crossorigin="anonymous">
    <link rel="stylesheet" href="{% static 'css/tagsinput.css' %}" />
    <link rel="stylesheet" href="{% static 'css/navbar.css' %}" />
    <link rel="stylesheet" href="{% static 'css/main.css' %}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Noto+Sans&display=swap"
rel="stylesheet">
    <style> @import
url('https://fonts.googleapis.com/css2?family=Noto+Sans&display=swap'); </style>
    {% block style %}{% endblock %}

</head>
<body>
          <nav class="navbar h-nav">
            {% if user.is_authenticated %}
              <ul class="nav-list">
                  <li><a class="nav-brand" href="{% url 'gallery' %}">Photo
Album</a></li>
                  <li><a class='nav-itm v-hid' href="{% url 'profile' %}">Profile</a></li>
                  <li><a class='nav-itm v-hid' href="{% url 'user-gallery' user.username
%}">My Uploads</a></li>
              </ul>
              <ul class="rightNav">
                <li><a class='nav-itm v-hid' href="{% url 'add' %}">Upload</a></li>
                <li><a class="nav-itm v-hid" href="{% url 'logout' %}">Logout</a></li>
              </ul>
            {% else %}
            <ul class="nav-list">
```

```
            <li><a class="nav-brand" href="{% url 'gallery' %}">Photo Album</a></li>
          </ul>
          <ul class="rightNav">
              <li><a class="nav-itm v-hid" href="{% url 'login' %}">Login</a></li>
              <li><a class="nav-itm v-hid" href="{% url 'register' %}">Register</a></li>
          </ul>
          {% endif %}
          <button id="menu">☰</button>
        </nav>

    {% block content %}
    {% endblock %}
    <!-- JavaScript Bundle with Popper -->
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-/bQdsTh/da6pkI1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ"
crossorigin="anonymous"></script>
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
  <script src="{% static 'js/tagsinput.js' %}"></script>
  <script src="{% static 'js/resp.js' %}"></script>
</body>
</html>
```

## Photoshare>photos>templates>photos>add.html

```
{% load static %}
<!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>Document</title>
  <link rel="icon"href="{% static 'favicon.ico' %}">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU" crossorigin="anonymous">
  <link rel="stylesheet" href="{% static 'css/tagify.css' %}" />
  <link rel="stylesheet" href="{% static 'css/main.css' %}">
```

```
    <style>
        .card {
            background: rgb(33 37 41);
            border: none;
        }

        .txt{
            background: rgb(19, 19, 19);
            color: aliceblue;
        }
        .txt:focus {
            background: rgb(19, 19, 19);
            color: aliceblue;
        }
    </style>
</head>
<body class="m-4">
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-5">
                <a href="{% url 'gallery' %}" class="btn btn-dark my-3">Go Back</a>
                <div class="card border-secondary">
                    <form method="POST" action="" enctype="multipart/form-data">
                        {% csrf_token %}
                        <div class="mb-3 m-3">
                            <legend class="form-label ">Description</legend>
                            {{ form.description }}
                        </div>
                        <div class="mb-3 m-3">
                            <label class="form-label">Upload image</label>
                            {{form.image}}
                            {% for err in form.image.errors %}
                                <small style="color: red;">{{err}}</small>
                            {% endfor %}
                        </div>
                        <div class="mb-3 m-3">
                            <label class="form-label">Tags <small
style='color:red'>*required</small></label>
                            {{form.tags}}
                            {% for err in form.tags.errors %}
                                <small style="color: red;">{{err}}</small>
                            {% endfor %}
                        </div>
                        <div class="mb-3 m-3">
                            <div class="form-check form-switch">
                                {{form.is_private}}
```

```
                                <label for="flexSwitchCheckDefault" class="form-check-
label">make private</label>
                            </div>
                        </div>
                        <button type="submit" class="btn btn-primary m-3 mb-
3">Submit</button>
                    </form>
                </div>
            </div>
        </div>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-/bQdsTh/da6pkI1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ"
crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script src="{% static 'js/jQuery.tagify.min.js' %}"></script>
    <script>
        $('[name=tags]').tagify({delimiters: ","});
    </script>
</body>
</html>
```

## Photoshare>photos>templates>photos>gallery.html

```
{% extends "base.html" %}
{% load static %}
{% block title %}Gallery{% endblock %}
{% block style %}
<style>
    @import url('{% static "css/base.css" %}');
    @import url('{% static "css/photo-grid.css" %}');
    @import url('{% static "css/gallery.css" %}');
</style>
{% endblock %}
{% block content %}
<div class="container my-4">
```

```
        <!-- search bar -->

        <section class="banner">
            <h1>
                Search image
            </h1>
            <form class="search" method="GET" class="row g-3">
                <div class="col-8">
                    <label>
                        <div class="search-bar">
                            <svg class="s-icon"
xmlns="http://www.w3.org/2000/svg" x="0px" y="0px"
                                width="50" height="50"
                                viewBox="0 0 50 50"
                                style=" fill:#fff;"><path d="M 21 3 C
11.621094 3 4 10.621094 4 20 C 4 29.378906 11.621094 37 21 37 C 24.710938 37
28.140625 35.804688 30.9375 33.78125 L 44.09375 46.90625 L 46.90625 44.09375 L
33.90625 31.0625 C 36.460938 28.085938 38 24.222656 38 20 C 38 10.621094
30.378906 3 21 3 Z M 21 5 C 29.296875 5 36 11.703125 36 20 C 36 28.296875
29.296875 35 21 35 C 12.703125 35 6 28.296875 6 20 C 6 11.703125 12.703125 5
21 5 Z"></path>
                            </svg>
                            <input type='text' class="search-box" name='tags'
placeholder="Search Photo" value="{{search_input}}">
                        </div>
                    </label>
                </div>
            </form>
        </section>


        <!-- display photos -->
        {%if is_tag%}
            <a href="{% url 'gallery' %}" class="btn btn-dark my-3 mt-0">Go
Back</a>
        {%endif%}
        {%if is_tag%}
            <h1 class="mb-3">Tagged {{ tag }} ({{ page_obj.paginator.count
}})</h1>
        {%endif%}
        <div class="photo-grid">
            {% for photo in photos %}
                    <a
                        href="{% url 'photo' photo.id %}"
                        class="card"
```

```
                        style="background-
image:url('{{photo.image.url}}');display: block;"
                    >
                    </a>
            {% empty %}
            <h3>No photos...</h3>
            {% endfor %}
        </div>

    <!-- pagination  -->

    <div class="page" style="margin-top: 10px;">
        {% if is_paginated %}

        {% if page_obj.has_previous %}
            <a class="btn btn-outline-success mb-4" href="?page=1">First</a>
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.previous_page_number }}">Previous</a>
        {% endif %}

        {% for num in page_obj.paginator.page_range %}
            {% if page_obj.number == num %}
            <a class="btn btn-success mb-4" href="?page={{ num }}">{{ num
}}</a>
            {% elif num > page_obj.number|add:'-3' and num <
page_obj.number|add:'3' %}
            <a class="btn btn-outline-success mb-4" href="?page={{ num }}">{{
num }}</a>
            {% endif %}
        {% endfor %}

        {% if page_obj.has_next %}
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.next_page_number }}">Next</a>
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.paginator.num_pages }}">Last</a>
        {% endif %}

        {% endif %}
    </div>
</div>
{% endblock %}
```

Photoshare>photos>templates>photos>photo.html

```
{% extends "base.html" %}
{% block title %}Photo{% endblock %}
{% block content %}
  <div class="container m-8">
        <div class="row justify-content-center">
            <div class="col-md-9">
                <a href="{{ prev_url }}" class="btn btn-dark my-3">Go Back</a>
                {% if object.auther == user %}
                    <a class="btn btn-secondary my-3" href="{% url 'update'
object.id %}">Update</a>
                    <a class="btn btn-danger my-3" href="{% url 'delete'
object.id %}">Delete</a>
                {% endif %}
                <div style='display: flex;justify-content: center'>
                    <div style="max-width: 100%;flex-basis: 80%; justify-
content:center; align-items:center; margin-bottom: 5px;">
                        <img style="max-width: 100%; max-height: 100%;"
src="{{photo.image.url}}" alt="">
                    </div>
                    <div style='padding-left: 50px;flex-basis: 40%;'>
                        <img class='rounded-circle m-1' width='45' height='45'
src="{{ photo.auther.profile.image.url }}">
                        / <a href="{% url 'user-gallery' photo.auther.username
%}">{{ photo.auther.username }}</a>

                        <h4>Description :</h4>
                        <p>
                            {{photo.description}}
                        </p>
                    </div>
                </div>
                {%for tag in photo.tags.all%}
                    <a href="{%url 'gallery'%}?tag={{ tag }}" style='margin:
2px;text-decoration: none;' class="badge badge-tag badge-info">{{tag}}</a>
                {%endfor%}
            </div>
        </div>
    </div>
{% endblock %}
```

Photoshare>photos>templates>photos>update.html

```
{% load static %}
<!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>Document</title>
    <link rel="icon"href="{% static 'favicon.ico' %}">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU"
crossorigin="anonymous">
    <link rel="stylesheet" href="{% static 'css/tagify.css' %}" />
    <link rel="stylesheet" href="{% static 'css/main.css' %}">
    <style>
        .card {
            background: rgb(33 37 41);
            border: none;
        }

        .txt{
            background: rgb(19, 19, 19);
            color: aliceblue;
        }
        .txt:focus {
            background: rgb(19, 19, 19);
            color: aliceblue;
        }
    </style>
</head>
<body class="m-4">
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-5">
                <a href="{% url 'photo' photos.id %}" class="btn btn-dark my-
3">Go Back</a>
                <div class="card border-secondary">
                    <form method="POST" action="" enctype="multipart/form-
data">

                        {% csrf_token %}

                        <div class="mb-3 m-3">
                            <legend class="form-label ">Description</legend>
                            {{ form.description }}
                        </div>
```

```
                        <div class="mb-3 m-3">
                            <label class="form-label">Tags</label>
                            {{form.tags}}
                            {% for err in form.tags.errors %}
                                <small style="color: red;">{{err}}</small>
                            {% endfor %}
                        </div>

                        <div class="mb-3 m-3">
                            <div class="form-check form-switch">
                                {{form.is_private}}
                                <label for="flexSwitchCheckDefault"
class="form-check-label">make private</label>
                            </div>
                            {% for err in form.tags.errors %}
                                <small style="color: red;">{{err}}</small>
                            {% endfor %}
                        </div>

                        <div class="mb-3 m-3">
                            <legend class="form-label">Photo</legend>
                            <img src="{{photos.image.url}}" width="300">
                        </div>
                        <button type="submit" class="btn btn-primary m-3 mb-
3">Update</button>
                    </form>
                </div>
            </div>
        </div>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/js/bootstrap.bundle.min
.js" integrity="sha384-
/bQdsTh/da6pkI1MST/rWKFNjaCP5gBSY4sEBT38Q/9RBh9AH40zEOg7Hlq2THRZ"
crossorigin="anonymous"></script>
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js
" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
```

```
    <script src="{% static 'js/jQuery.tagify.min.js' %}"></script>
    <script>
        $('[name=tags]').tagify({delimiters: ","});
    </script>
</body>
</html>
```

## Photoshare>photos>templates>photos>user_photos.html

```
{% extends "base.html" %}
{% load static %}
{% block title %}Gallery{% endblock %}
{% block style %}
<style>
    @import url('{% static "css/base.css" %}');
    @import url('{% static "css/photo-grid.css" %}');
    @import url('{% static "css/gallery.css" %}');
</style>
{% endblock %}
{% block content %}
<div class="container my-4">
    <section class="banner">
        <h1>
            Search image
        </h1>
        <form class="search" method="GET">
            <div class="col-8">
                <label>
                    <div class="search-bar">
                        <svg class="s-icon" xmlns="http://www.w3.org/2000/svg"
x="0px" y="0px"
                             width="50" height="50"
                             viewBox="0 0 50 50"
                             style=" fill:#fff;"><path d="M 21 3 C 11.621094 3
4 10.621094 4 20 C 4 29.378906 11.621094 37 21 37 C 24.710938 37 28.140625
35.804688 30.9375 33.78125 L 44.09375 46.90625 L 46.90625 44.09375 L 33.90625
31.0625 C 36.460938 28.085938 38 24.222656 38 20 C 38 10.621094 30.378906 3 21
3 Z M 21 5 C 29.296875 5 36 11.703125 36 20 C 36 28.296875 29.296875 35 21 35
C 12.703125 35 6 28.296875 6 20 C 6 11.703125 12.703125 5 21 5 Z"></path>
                        </svg>
                        <input type='text' class="search-box" name='tags'
placeholder="Search Photo" value="{{search_input}}">
                    </div>
                </label>
```

```
            </div>
        </form>
    </section>



    {%if view.kwargs.username == user.username %}
    <h1 class='mb-3'>My photos {{ utag }} ({{page_obj.paginator.count}})</h1>
    {%else%}
    <h1 class="mb-3">Photos by {{ view.kwargs.username }} {{ utag }} ({{
page_obj.paginator.count }})</h1>
    {%endif%}
    <div class="photo-grid">
        {% for photo in photos %}
            <a
                href="{% url 'photo' photo.id %}"
                class="card"
                style="background-image:url('{{photo.image.url}}')"
            ></a>
        {% empty %}
        <h3>No photos...</h3>
        {% endfor %}
    </div>
    <div class="page" style="margin-top: 10px;">
        {% if is_paginated %}

        {% if page_obj.has_previous %}
            <a class="btn btn-outline-success mb-4" href="?page=1">First</a>
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.previous_page_number }}">Previous</a>
        {% endif %}

        {% for num in page_obj.paginator.page_range %}
            {% if page_obj.number == num %}
            <a class="btn btn-success mb-4" href="?page={{ num }}">{{ num
}}</a>
            {% elif num > page_obj.number|add:'-3' and num <
page_obj.number|add:'3' %}
            <a class="btn btn-outline-success mb-4" href="?page={{ num }}">{{
num }}</a>
            {% endif %}
        {% endfor %}

        {% if page_obj.has_next %}
```

```
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.next_page_number }}">Next</a>
            <a class="btn btn-outline-success mb-4" href="?page={{
page_obj.paginator.num_pages }}">Last</a>
        {% endif %}

        {% endif %}
    </div>

</div>


{% endblock %}
```

## Photoshare>photos>templates>photos>photo_confirm_delete.html

```
{% extends "base.html" %}
{% block content %}
<div class='container mt-5'>
    <div class="row justify-content-center">
        <div class="col-md-5">
            <form method="POST">
                {% csrf_token %}
                <fieldset class="form-group">
                    <legend class="border-bottom mb-4">Delete Photo</legend>
                    <h2>Are you sure you want to delete</h2>
                    <img src="{{ object.image.url }}" width='300px'>
                </fieldset>
                <div class="form-group pt-3 mb-3">
                    <button class="btn btn-outline-danger" type="submit">Delete!</button>
                    <a class="btn btn-outline-secondary" href="{% url 'photo' object.id
%}">Cancel</a>
                </div>
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

## Photoshare>users>admin.py

```
from django.contrib import admin
from .models import Profile
# Register your models here.

admin.site.register(Profile)
```

## Photoshare>users>apps.py

```
from django.apps import AppConfig


class UsersConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'users'

    def ready(self):
        import users.signals
```

## Photoshare>users>signals.py

```
from django.db.models.signals import post_save
from django.contrib.auth.models import User
from django.dispatch import receiver
from .models import Profile


@receiver(post_save, sender=User)
def create_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.create(user=instance)

@receiver(post_save, sender=User)
def save_profile(sender, instance, **kwargs):
    instance.profile.save()
```

## Photoshare>users>forms.py

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm
from django.forms import FileInput
```

```
from .models import Profile


class UserRegistrationForm(UserCreationForm):
    email=forms.EmailField()

    class Meta:
        model=User
        fields=['username', 'email', 'password1', 'password2']

class UserUpdatefrom(forms.ModelForm):
    email=forms.EmailField()

    class Meta:
        model=User
        fields=['username', 'email']

class ProfileUpdateForm(forms.ModelForm):
    class Meta:
        model=Profile
        fields = ['image']
        widgets = {
            'image': FileInput(
                        attrs={'class': 'form-control border-secondary'
,'type': 'file', 'required': True}),
        }
```

Photoshare>users>models.py

```
from django.db import models
from django.contrib.auth.models import User
from PIL import Image


class Profile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
```

```
    image = models.ImageField(default='default.jpg', upload_to='profile_pics')

    def __str__(self):
        return f'{self.user.username} Profile'

    def save(self, *args, **kwargs):
        super(Profile, self).save(*args, **kwargs)

        img = Image.open(self.image.path)

        if img.height > 300 or img.width > 300:
            output_size = (300, 300)
            img.thumbnail(output_size)
            img.save(self.image.path)
```

## Photoshare>users>urls.py

```
from django.urls import path
from . import views
from django.contrib.auth import views as auth_views
from users import views as user_views


urlpatterns = [
    path('login/', views.UserLogin.as_view(), name='login'),
    path('register/', views.register, name='register'),
    path('profile/', views.profile, name='profile'),
    path('logout/', auth_views.LogoutView.as_view(template_name='users/logout.html'),
name='logout'),
    path('password-reset/',
auth_views.PasswordResetView.as_view(template_name='users/password_reset.html'),
name='password_reset'),
    path('password-reset/done/',
auth_views.PasswordResetDoneView.as_view(template_name='users/password_reset_done.html'),
name='password_reset_done'),
    path('password-reset-confirm/<uidb64>/<token>',
auth_views.PasswordResetConfirmView.as_view(template_name='users/password_reset_confirm.ht
ml'), name='password_reset_confirm'),
    path('password-reset-complete/',
auth_views.PasswordResetCompleteView.as_view(template_name='users/password_reset_complete.
html'), name='password_reset_complete')

]
```

Photoshare>users>views.py

```
from django.shortcuts import render, redirect
from django.views.generic import FormView
from django.contrib.auth.views import LoginView, LogoutView
from .forms import UserRegistrationForm,UserUpdatefrom,ProfileUpdateForm
from django.contrib.auth import login
from django.urls import reverse_lazy
from django.shortcuts import render, redirect
from django.contrib import messages
from django.contrib.auth.decorators import login_required
from .models import Profile

# Create your views here.

class UserLogin(LoginView):
    template_name = 'users/login.html'
    fields = ['email', 'password']

    redirect_authenticated_user = True

    def get_success_url(self):
        return reverse_lazy('gallery')

def register(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            user=form.save()
            if user is not None:
                login(request, user)
            username = form.cleaned_data.get('username')
            messages.success(request, f'Your account has been created! You are
now able to log in')
            return redirect('login')

    else:
        form = UserRegistrationForm()
    return render(request, 'users/register.html', {'form': form})


@login_required
def profile(request):
    if request.method == 'POST':
        u_form = UserUpdatefrom(request.POST, instance=request.user)
```

```
        p_form = ProfileUpdateForm(request.POST,
                                   request.FILES,
                                   instance=request.user.profile)
        if u_form.is_valid() and p_form.is_valid():
            u_form.save()
            p_form.save()
            messages.success(request, f'Your account has been updated!')
            return redirect('profile')

    else:
        u_form = UserUpdatefrom(instance=request.user)
        p_form = ProfileUpdateForm()

    context = {
        'u_form': u_form,
        'p_form': p_form
    }

    return render(request, 'users/profile.html', context)
```

Photoshare>users>templates>users>login.html

```
{% extends "base.html" %}
{% load crispy_forms_tags %}
{% block content %}
    <div class="container mt-5">
        <div class="row justify-content-center">
            <div class="col-md-5">
            <form method="POST">
                {% csrf_token %}
                <fieldset class="form-group">
                    <legend class="border-bottom mb-4">Log In</legend>
                    {{ form|crispy }}
                </fieldset>
                <div class="form-group pt-3">
                    <button class="btn btn-outline-info"
type="submit">Login</button>
                    <small class="text-muted"><a href="{% url 'password_reset'
%}">Forgot password?</a></small>
                </div>
            </form>
            <div class="pt-3">
                <div class="border-top">
                    <small class="text-muted">Need an account? <a class="ml-2"
href="{% url 'register' %}">Sign Up</a></small>
```

```
                </div>
            </div>
        </div>
    </div>
    </div>
{% endblock %}
```

Photoshare>users>templates>users>logout.html

```
{% extends "base.html" %}
{% block content %}
    <div class='container mt-5'>
        <div class="row justify-content-center">
            <div class="col-md-5">
                <h2>You have been logged out</h2>
                <div class="pt-3">
                    <div class="border-top">
                        <small class="text-muted">
                            <a href="{% url 'login' %}">Log In Again</a>
                        </small>
                    </div>
                </div>
            </div>
        </div>
    </div>
{% endblock %}
```

Photoshare>users>templates>users>password_reset_complete.html

```
{% extends "base.html" %}
{% block content %}
<div class='container'>
    <div class="row justify-content-center">
        <div class="col-md-5">
            <div class="alert alert-info">
                Your Password has been set.
            </div>
            <a href="{% url 'login' %}">Sign In Here</a>
        </div>
    </div>
</div>
{% endblock %}
```

Photoshare>users>templates>users>password_reset_confirm.html

```
{% extends "base.html" %}
{% load crispy_forms_tags %}
{% block content %}
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-5">
                <form method="POST">
                    {% csrf_token %}
                    <fieldset class="form-group">
                        <legend class="border-bottom mb-4">Reset
Password</legend>
                        {{ form|crispy }}
                    </fieldset>
                    <div class="form-group pt-3">
                        <button class="btn btn-outline-info"
type="submit">Reset</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
{% endblock %}
```

Photoshare>users>templates>users>password_reset_done.html

```
{% extends "base.html" %}
{% block content %}
<div class='container'>
    <div class="row justify-content-center">
        <div class="col-md-5">
            <div class="alert alert-info">
                An email has been sent to recover your password!
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

Photoshare>users>templates>users>password_reset.html

```
{% extends "base.html" %}
{% load crispy_forms_tags %}
{% block content %}
<div class='container mt-5'>
    <div class="row justify-content-center">
        <div class="col-md-5">
            <div class="content-section">
                <form method="POST">
                    {% csrf_token %}
                    <fieldset class="form-group">
                        <legend class="border-bottom mb-4">Reset
Password</legend>
                        {{ form|crispy }}
                    </fieldset>
                    <div class="form-group pt-3">
                        <button class="btn btn-outline-info"
type="submit">Request Password Reset</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
{% endblock %}
```

## Photoshare>users>templates>users>profile.html

```
{% extends "base.html" %}
{% load crispy_forms_tags %}
{% block content %}
<div class='container mt-3'>
    <div class="row justify-content-center">
        <div class="col-md-5">
            <img class="rounded-circle account-img" height=125 src="{{
user.profile.image.url }}">
                <h2 class="account-heading">{{ user.username }}</h2>
                <p class="text-secondary">{{ user.email }}</p>
            <form method="POST" enctype="multipart/form-data">
                {% csrf_token %}
                <fieldset class="form-group mt-3">
                    <legend class="border-bottom mb-4">Profile Info</legend>
                    {{ u_form|crispy }}
                    <br>
                    {{ p_form|crispy }}
                </fieldset>
```

```
                <a class="btn btn-info mt-3" href="{% url 'password_reset'
%}">Reset Password</a>
                <div class="form-group pt-3">
                    <button class="btn btn-outline-info"
type="submit">Update</button>
                </div>
            </form>
        </div>
    </div>
</div>
{% endblock %}
```

## Photoshare>users>templates>users>register.html

```
{% extends "base.html" %}
{% load crispy_forms_tags %}
{% block content %}
    <div class='container mt-5'>
        <div class="row justify-content-center">
            <div class="col-md-5">
            <form method="POST">
                {% csrf_token %}
                <fieldset class="form-group">
                    <legend class="border-bottom mb-4">Join today</legend>
                    {{ form|crispy }}
                </fieldset>
                <div class="form-group pt-3">
                    <button class="btn btn-outline-info" type="submit">Sign
Up</button>
                </div>
            </form>
            <div class="pt-3">
                <div class="border-top">
                    <small class="text-muted">Already have an account? <a
class="ml-2" href="{% url 'login' %}">Sign In</a></small>
                </div>
            </div>
            </div>
        </div>
    </div>
{% endblock %}
```

## Photoshare>photoshare>asgi.py

```
"""
ASGI config for photoshare project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'photoshare.settings')

application = get_asgi_application()
```

## Photoshare>photoshare>asgi.py

```
"""
Django settings for photoshare project.

Generated by 'django-admin startproject' using Django 3.1.7.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
"""

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'ot!=0r8p-nuilr0kq5b88!51o&ox_uxe2n8xst=n^u_jc=3oax'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['*']


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'photos.apps.PhotosConfig',
    'users.apps.UsersConfig',
    'crispy_forms',
    'taggit',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'photoshare.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
```

```
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'photoshare.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
   'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'photoshare',
        'USER': 'siddid',
        'PASSWORD': 'Root@2022',
        'HOST':'localhost',
        'PORT':'3306',
    },
    'OPTIONS': { 'init_command': 'SET storage_engine=INNODB;' }
}


# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
```

```
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


LOGIN_REDIRECT_URL='gallery'

LOGIN_URL='login'

CRISPY_TEMPLATE_PACK='bootstrap4'

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC_URL = '/static/'
MEDIA_URL = '/images/'

STATICFILES_DIRS = [
    BASE_DIR / 'static'
]

MEDIA_ROOT = BASE_DIR / 'static/images'
STATIC_ROOT = BASE_DIR / 'staticfiles'


EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST= 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = "siddidsoni2@gmail.com"      #enter an spare email id
EMAIL_HOST_PASSWORD = ""           #enter the password
```

## Photoshare>photoshare>urls.py

```
"""photoshare URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
import photos
from django.contrib import admin
from django.urls import path, include

from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('photos.urls')),
    path('',include('users.urls'))
]

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
```

## Photoshare>photoshare>wsgi.py

```
"""
WSGI config for photoshare project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/wsgi/
```

```
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'photoshare.settings')

application = get_wsgi_application()
```

Photoshare>static>css>base.css

```
@import url('https://fonts.googleapis.com/css?family=Noto+Sans&display=swap');

body {
    background: rgb(19, 19, 19);
    color: #fff;
    font-family: 'Noto Sans', sans-serif;
}

.card {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    background: #353535;
    font-size: 3rem;
    color: #fff;
    box-shadow: rgba(3, 8, 20, 0.1) 0px 0.15rem 0.5rem, rgba(2, 8, 20, 0.1)
0px 0.075rem 0.175rem;
    height: 100%;
    width: 100%;
    border-radius: 4px;
    transition: all 500ms;

    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;

 }

.card:hover {
```

```
    box-shadow: rgba(2, 8, 20, 0.1) 0px 0.35em 1.175em, rgba(2, 8, 20, 0.08)
0px 0.175em 0.5em;
    transform: translateY(-3px) scale(1.1);
 }
```

Photoshare>static>css>gallery.css

```
.banner {
    display: flex;
    width: 100%;
    height: 200px;
    align-items: center;
    flex-direction: column;
}

.search {
    display: flex;
    gap: 0.5rem;
}

.search-bar {
    display: flex;
    height: 42px;
    width: 42vw;
    background-color: black;
    border-radius: 20px;
    border: none;
    color: whitesmoke;
    align-items: center;
    margin-top: 10px;
}

.s-icon {
    height: 80%;
}

.search-box{
    text-justify: center;
    height: 90%;
    width: 80%;
    background-color: black;
    border: none;
    color: whitesmoke;
    font-size: large;
```

```
}

.search-box:focus {
    outline: none;
}
```

Photoshare>static>css>main.css

```
body {
    background: rgb(19, 19, 19);
    color: whitesmoke;
}
```

Photoshare>static>css>navbar.css

```
* {
    margin: 0;
    padding: 0;
}


/* navbar */

.navbar {
    display: flex;
    background-color: #212529;
    font-family: cascadia mono;
    height: 65px;

}

/* Left nav */

.nav-list .nav-brand {
    color: white;
    font-size: 20px;
}

.nav-list {
   width: 70%;
   display:flex;
   align-items: center;
   padding-left: 20px;
}
```

```css
.nav-list li{
    padding: 10px 9px;
    list-style: none;
}

.nav-list li a{
    text-decoration: none;
    color: #96989a;
    transition: color 100ms ease-in;
}

/* rightNav */

.rightNav {
    display: flex;
    list-style: none;
    width: 30%;
    align-items: center;
    justify-content: end;
    padding-right: 10px;
}

.rightNav li{
    padding: 8px;
    list-style: none;
}

.rightNav li a{
    text-decoration: none;
    color: #96989a;
    transition: color 100ms ease-in;
}

/* a */

li a:hover{
    color: white;
}

/*button*/

#menu {
    display: none;
    width:52px;
```

```
    height:34px;
    font-size: 22px;
    background-color: #212529;
    color: #96989a;
    border-radius: 7px;
    border: .5px solid #96989a;
    position: absolute;
    right: 3%;
    top:10px;
    cursor: pointer;
}

#menu:active{
    border: 3px solid #96989a;
}

/*media qury*/


@media (max-width:750px) {
    /* .nav-itm {
        display: none;
    } */

    .navbar{
        height: 262px;
        flex-direction: column;
        transition: height 0.3s linear;
        overflow: hidden;
    }

    .nav-list{
        width: 70%;
        flex-direction: column;
        align-items: start;
    }
    .rightNav {
        flex-direction: column;
        align-items: start;
        padding-left: 20px;
        padding-bottom: 10px;
    }
    .h-menu{
        width: 30%;
        justify-content:safe;
```

```
    }

    #menu {
        display: block;
    }

    .h-nav{
        height: 57.2px;
    }


    .anime{
        animation-name: fade;
        animation-duration: 450ms;
    }

}


@keyframes fade {
    0% {opacity:0%}
    25% {opacity:0%}
    50% {opacity:0%}
    75% {opacity:75%}
    100% {opacity:100%}
}
```

Photoshare>static>css>photo-grid.css

```css
.photo-grid {
    display: grid;
    gap: 1rem;

    grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
    grid-auto-rows: 240px;

    overflow: hidden;
}

@media screen and (min-width: 600px) {
    .card-tall {
        grid-row: span 2 / auto;
    }

    .card-wide {
        grid-column: span 2 / auto;
    }
}
```

Photoshare>static>images>default.jpg

Photoshare>static>favicon.ico

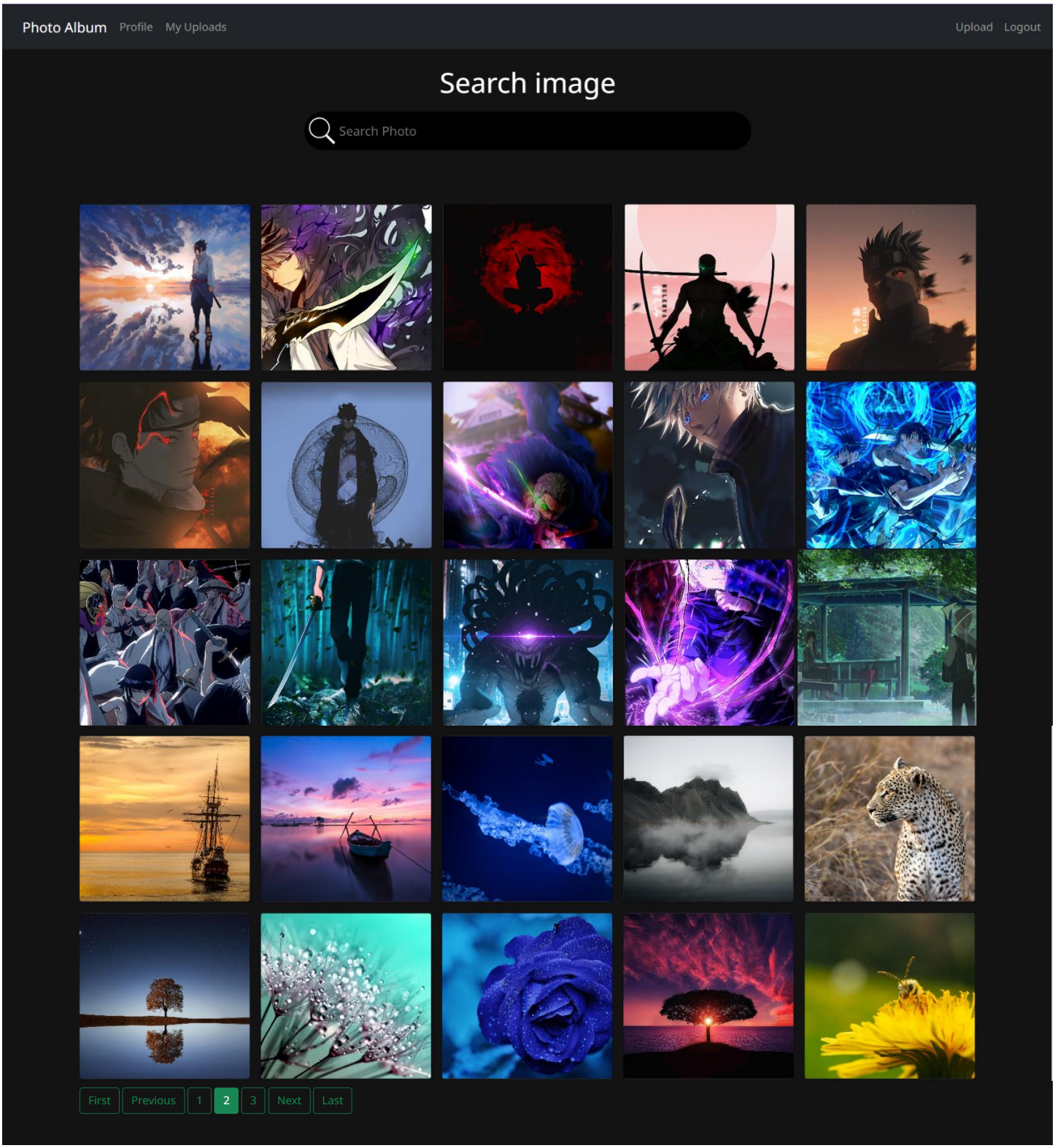
Photoshare>manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'photoshare.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)


if __name__ == '__main__':
    main()
```

# OUTPUT SNAPSHOTS



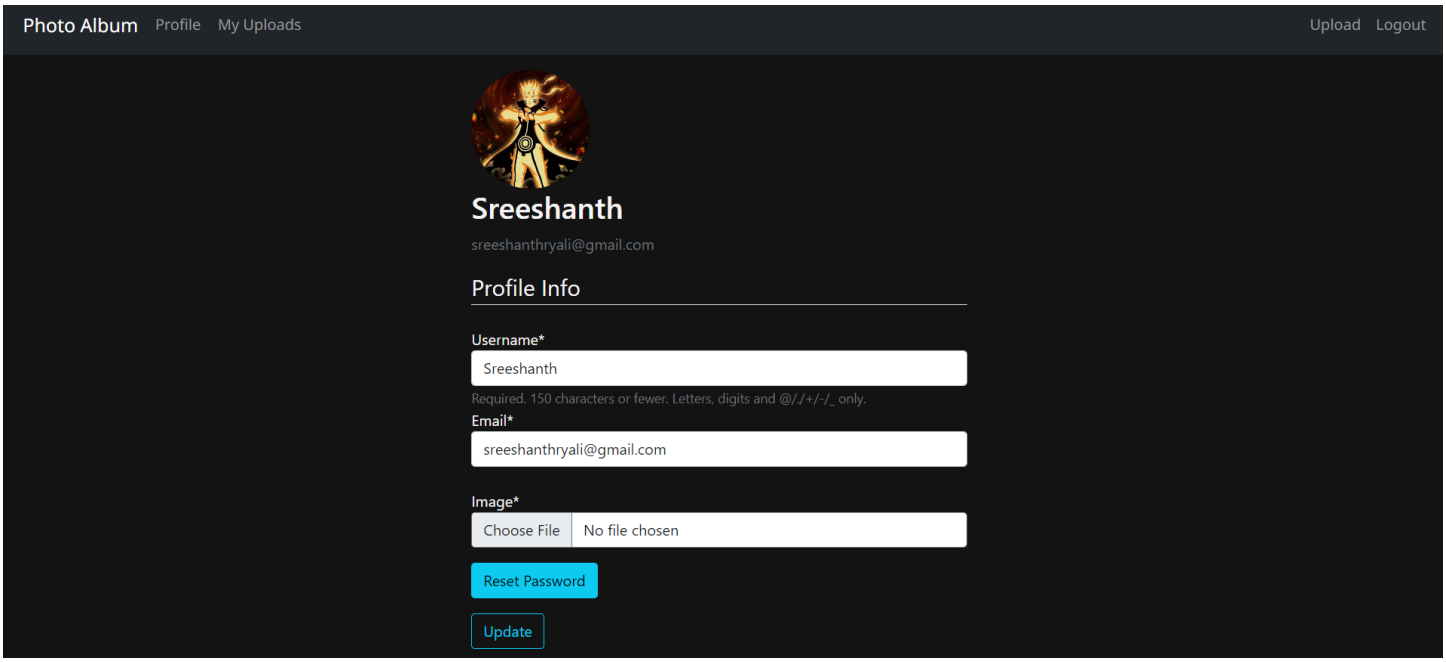


Photo Album
Profile
My Uploads
Upload
Logout
Sreeshanth
sreeshanthryali@gmail.com
Profile Info
Username*
Sreeshanth
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Email*
sreeshanthryali@gmail.com
Image*
Choose File
No file chosen
Reset Password
Update

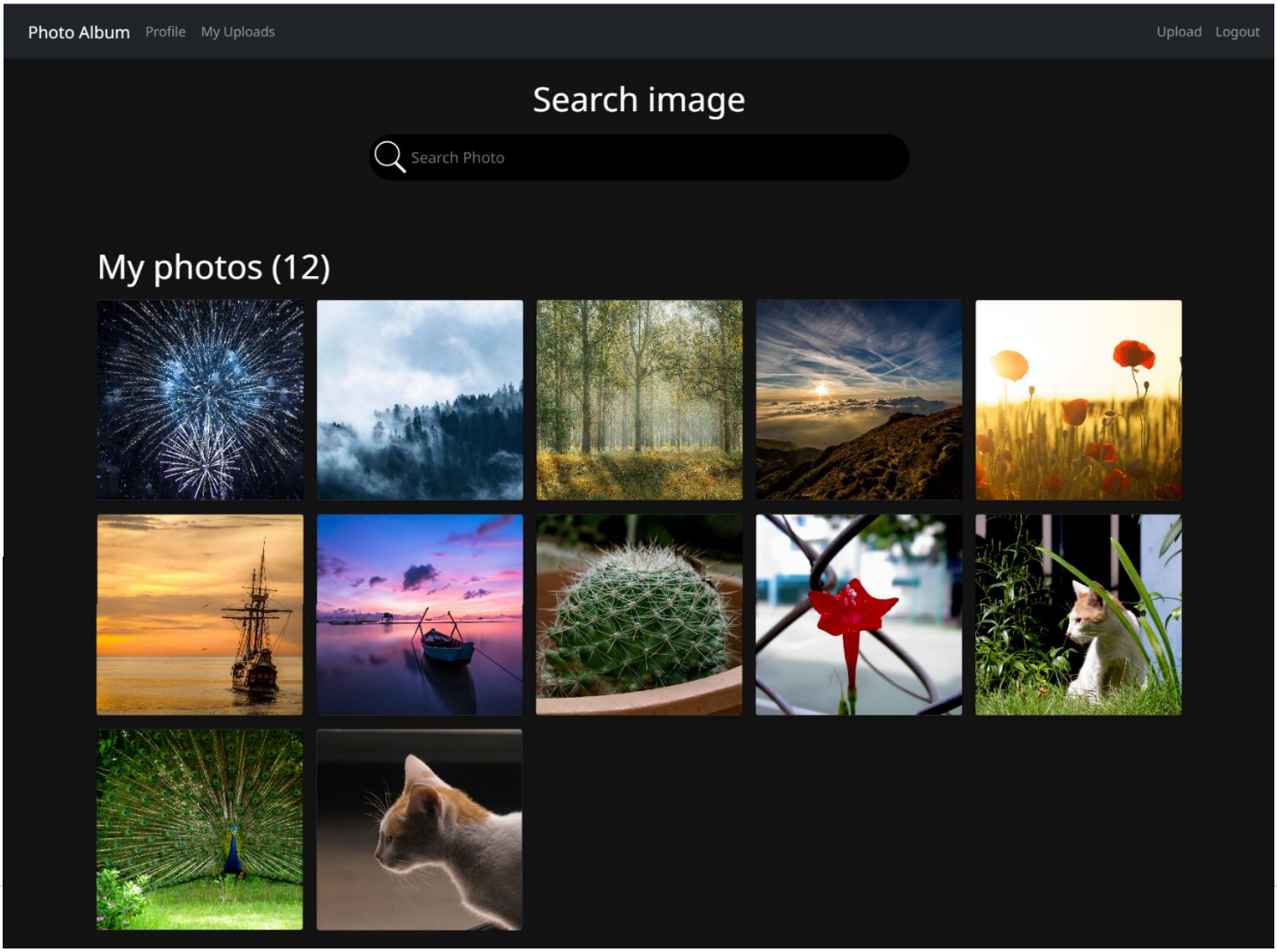


Photo Album
Profile
My Uploads
Upload
Logout
Search image
Search Photo
My photos (12)

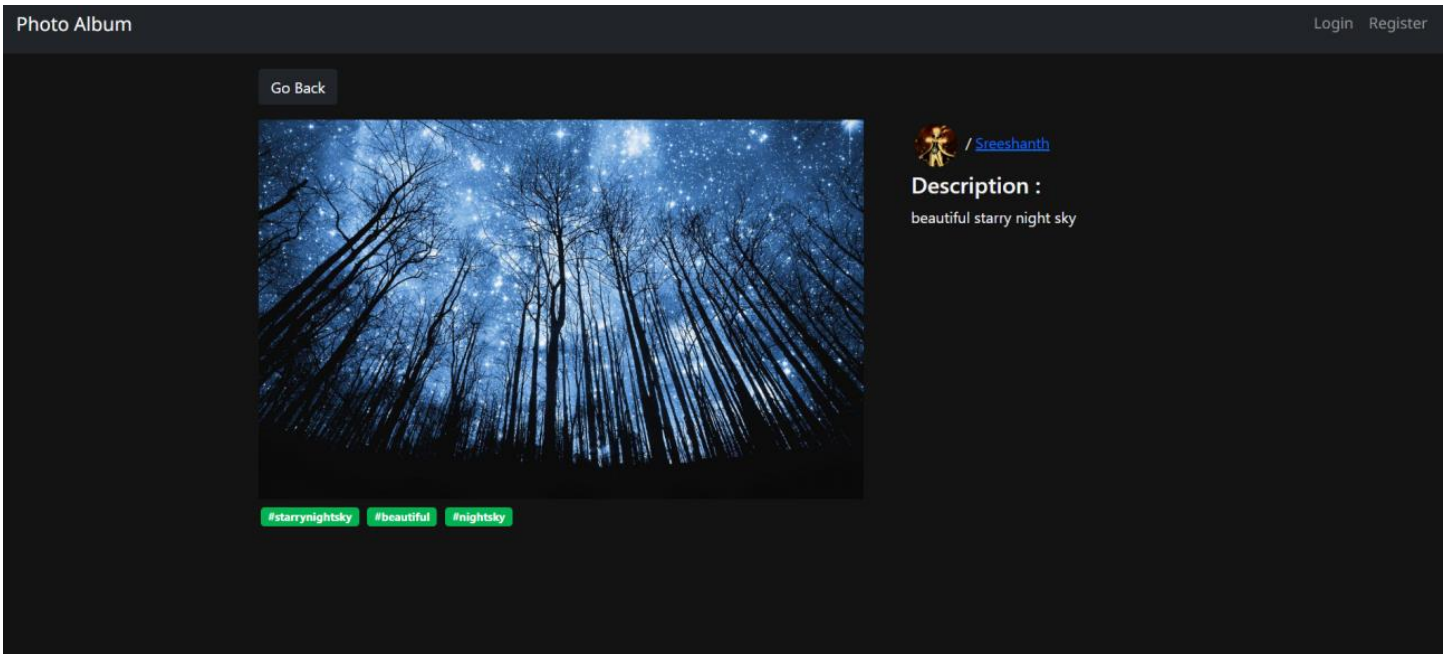


Photo Album
Login
Register
Go Back
/ Sreeshanth
Description :
beautiful starry night sky
#starrynightsky
#beautiful
#nightsky

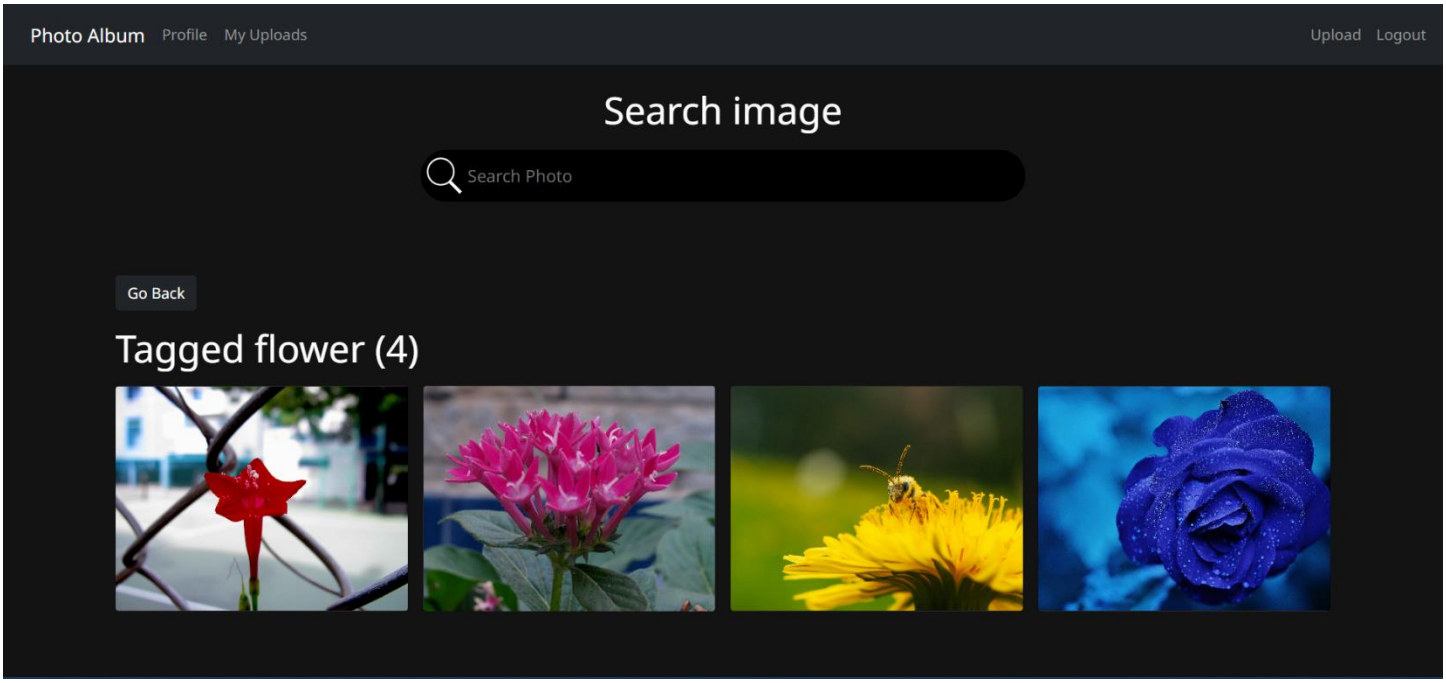
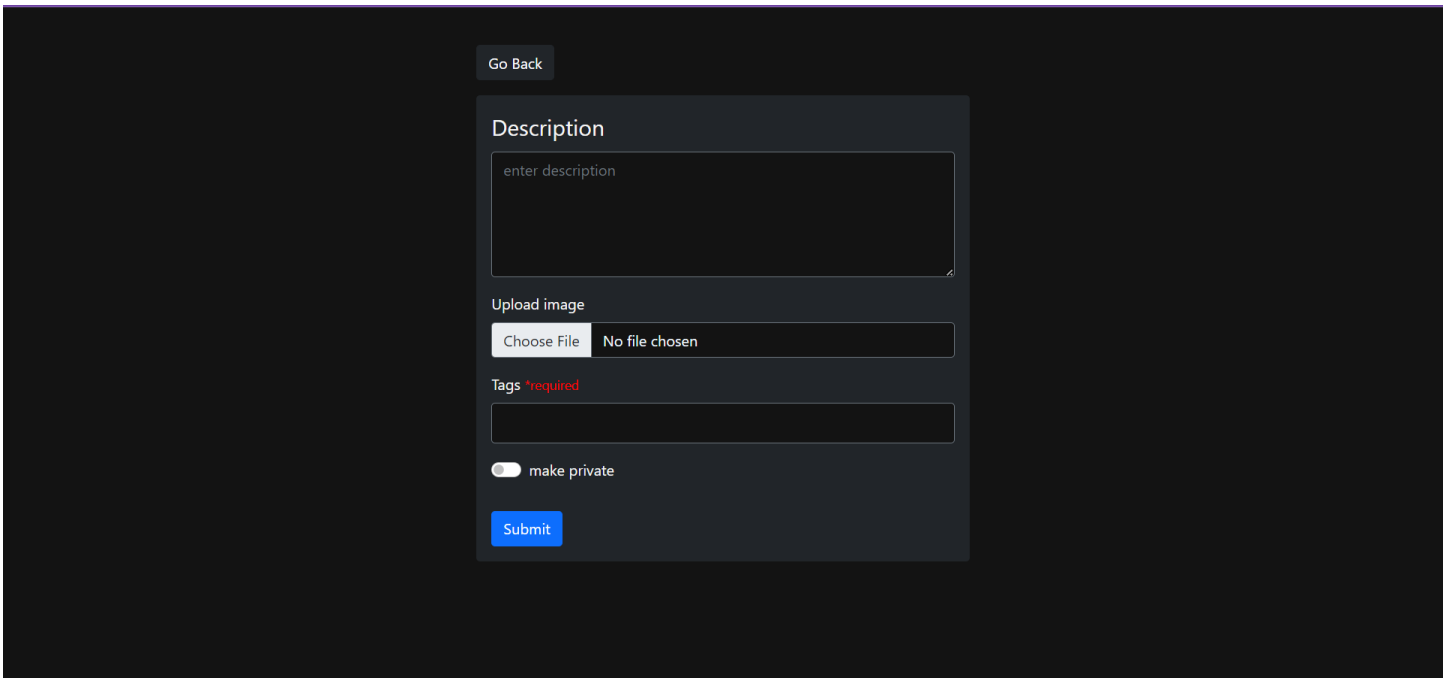


Photo Album
Profile
My Uploads
Upload
Logout
Search image
Search Photo
Go Back
Tagged flower (4)


Go Back
Description
enter description
Upload image
Choose File
No file chosen
Tags *required
make private
Submit

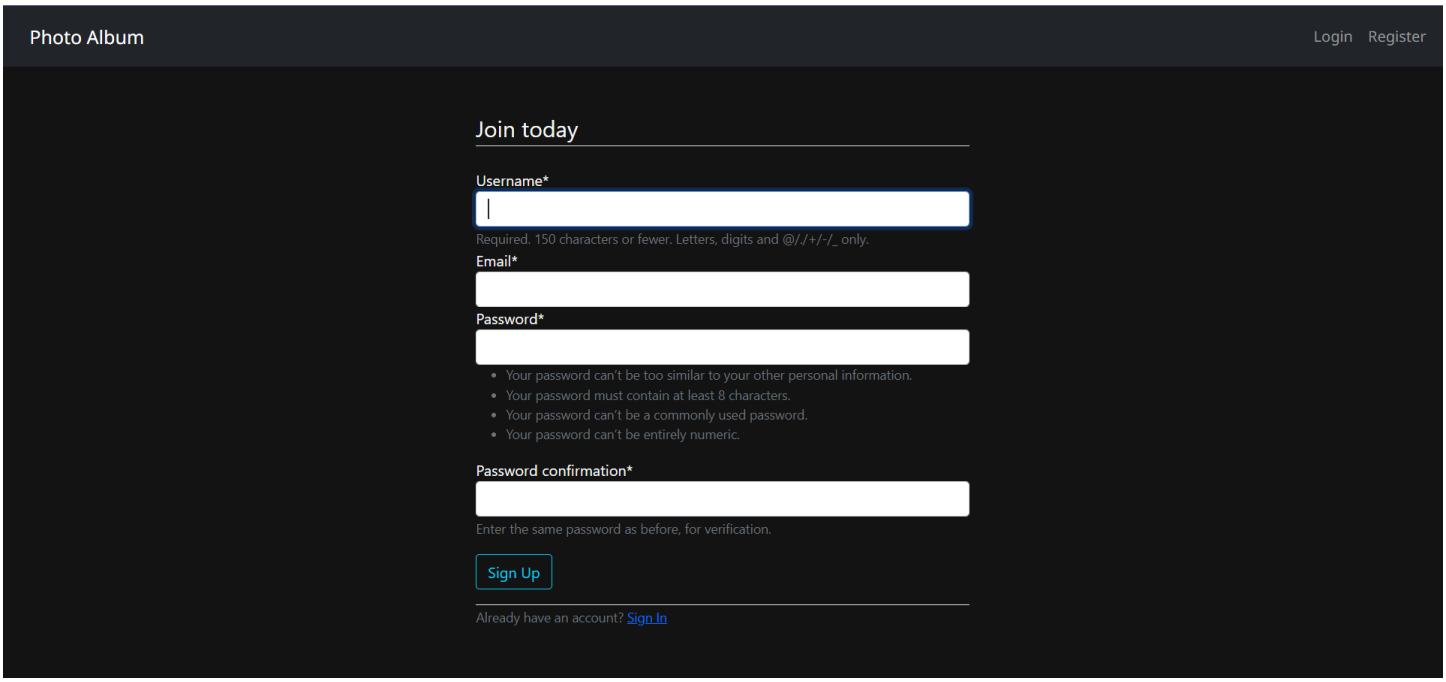
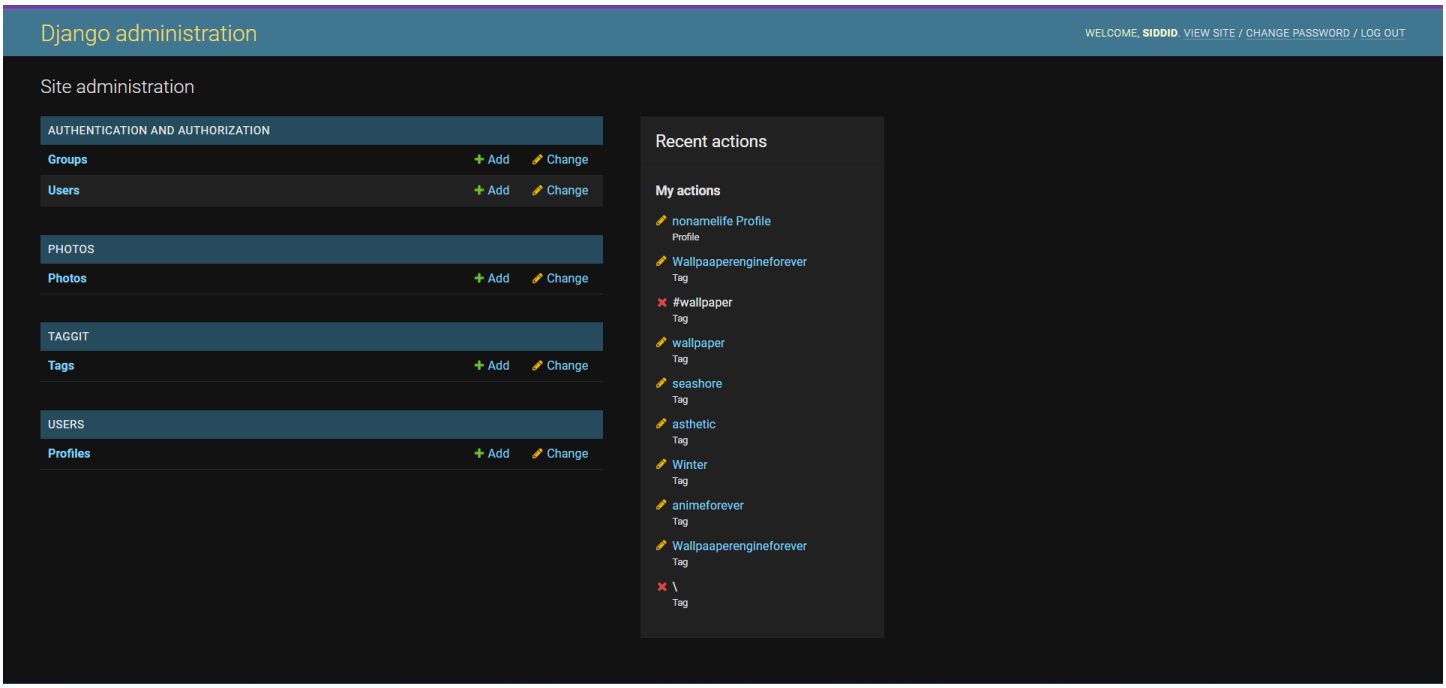


Photo Album
Login
Register
Join today
Username*
Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.
Email*
Password*
Your password can't be too similar to your other personal information.
Your password must contain at least 8 characters.
Your password can't be a commonly used password.
Your password can't be entirely numeric.
Password confirmation*
Enter the same password as before, for verification.
Sign Up
Already have an account? Sign In


Django administration
WELCOME, SIDDID. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
Users + Add Change
PHOTOS
Photos + Add Change
TAGGIT
Tags + Add Change
USERS
Profiles + Add Change
Recent actions
My actions
nonamelife Profile
Profile
Wallpaaperengineforever
Tag
#wallpaper
Tag
wallpaper
Tag
seashore
Tag
asthetic
Tag
Winter
Tag
animeforever
Tag
Wallpaaperengineforever
Tag
\
Tag

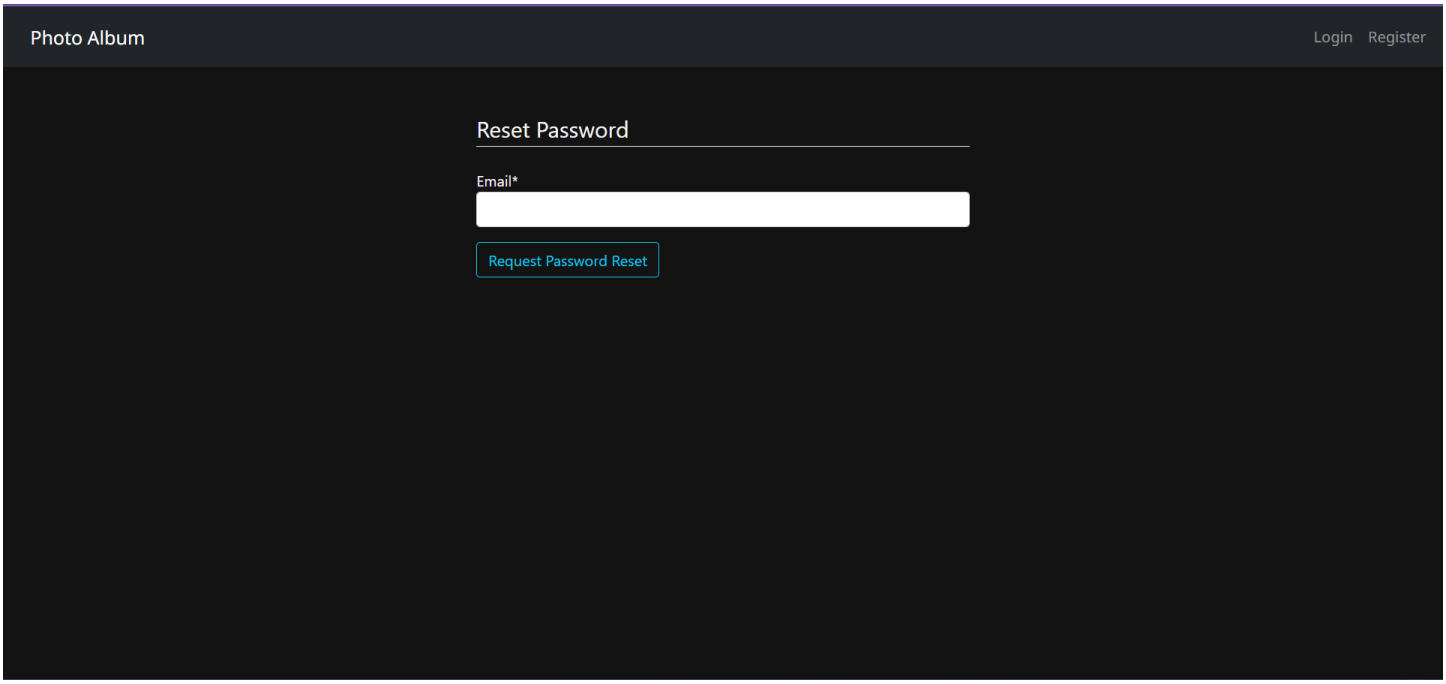


Photo Album
Login Register
Reset Password
Email*
Request Password Reset

# CONCLUSION

This project added value to our knowledge. We had a lot of fun creating the project like how to handle faults and errors, what to do when the code doesn’t produce the desired results, etc. We intend to develop more features, including the ability to add friends, a system for voting pictures, and much more.

# BIBILOGRAPHY


- https://www.youtube.com/playlist?list=PL-51WBLyFTg2vW-_6XBoUpE7vpmoR3ztO

- https://www.youtube.com/playlist?list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p