# Create Project

1 – Create MyBlog Project

2 - MyBlog>pip install pillow

3 - MyBlog>py manage.py startapp App\_Login

4 - MyBlog>py manage.py startapp App\_Blog

5 – Create templates FOLDER

6 – Create static FOLDER

7 – Create media FOLDER

8 – settings.py

import os.path

from pathlib import Path

# Build paths inside the project like this: BASE\_DIR / 'subdir'.

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

TEMPLATES\_DIR = os.path.join(BASE\_DIR, 'templates')

STATIC\_DIR = os.path.join(BASE\_DIR, 'static')

MEDIA\_DIR = os.path.join(BASE\_DIR, 'media')

# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!

SECRET\_KEY = 'django-insecure-$ko9qccy&i8k+htx=33+&u0%!5qqq%%1swk)^pzv9ref#22&m5'

# 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',

'App\_Login',

'App\_Blog',

]

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 = 'MyBlog.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [BASE\_DIR / 'templates']

,

'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 = 'MyBlog.wsgi.application'

# Database

# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.sqlite3',

'NAME': BASE\_DIR / 'db.sqlite3',

}

}

# Password validation

# https://docs.djangoproject.com/en/3.2/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/3.2/topics/i18n/

LANGUAGE\_CODE = 'en-us'

TIME\_ZONE = 'UTC'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

# Static files (CSS, JavaScript, Images)

# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC\_URL = '/static/'

# Default primary key field type

# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

STATICFILES\_DIRS = [STATIC\_DIR]

MEDIA\_ROOT = MEDIA\_DIR

MEDIA\_URL = '/media/'

# Models

9 – App\_Login => models.py

from django.db import models

from django.contrib.auth.models import User

class UserProfile(models.Model):

user = models.OneToOneField(User, related\_name='user\_profile', on\_delete=models.CASCADE)

profile\_pic = models.ImageField(upload\_to='profile\_pics')

10 - MyBlog>py manage.py migrate

11 - MyBlog>py manage.py makemigrations App\_Login

12 - MyBlog>py manage.py migrate

13 – App\_Blog => models.py

from django.db import models

from django.contrib.auth.models import User

class Blog(models.Model):

author = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='post\_author')

blog\_title = models.CharField(max\_length=264, verbose\_name="Put a Title")

slug = models.SlugField(max\_length=264, unique=True)

blog\_content = models.TextField(verbose\_name="What is on your mind?")

blog\_image = models.ImageField(upload\_to='blog\_images', verbose\_name="Blog Image")

publish\_date = models.DateTimeField(auto\_now\_add=True)

update\_date = models.DateTimeField(auto\_now=True)

def \_\_str\_\_(self):

return self.blog\_title

class Comment(models.Model):

blog = models.ForeignKey(Blog, on\_delete=models.CASCADE, related\_name='blog\_comment')

user = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='user\_comment')

comment = models.TextField()

comment\_date = models.DateTimeField(auto\_now\_add=True)

class Likes(models.Model):

blog = models.ForeignKey(Blog, on\_delete=models.CASCADE, related\_name='liked\_blog')

user = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='liker\_user')

14 - MyBlog>py manage.py migrate

15 - MyBlog>py manage.py makemigrations

16 - MyBlog>py manage.py migrate

17 - MyBlog>py manage.py runserver

18 - MyBlog>py manage.py createsuperuser

Username (leave blank to use 'ihc'): admin

Email address: admin@yahoo.com

Password:123

Password (again):123

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.

19 – App\_Blog => admin.py

from django.contrib import admin

from . models import Blog, Comment, Likes

admin.site.register(Blog)

admin.site.register(Comment)

admin.site.register(Likes)

20 – App\_Login => admin.py

from django.contrib import admin

from . models import UserProfile

admin.site.register(UserProfile)

21 - <http://127.0.0.1:8000/admin/>

22 – MyBlog => urls.py

from django.contrib import admin

from django.urls import path, include

from . import views

urlpatterns = [

path('admin/', admin.site.urls),

path('account/', include('App\_Login.urls')),

path('blog/', include('App\_Blog.urls')),

path('', views.index, name='index')

]

23 – App\_Blog => urls.py

from django.urls import path

from . import views

urlpatterns = [

]

24 – App\_Login => urls.py

from django.urls import path

from . import views

urlpatterns = [

]

25 – MyBlog => views.py

from django.http import HttpResponse

def index(request):

return HttpResponse('Home Page')

26 – In the templates FOLDER -> Creates FOLDERS App\_Blog & App\_Login

# URL, View and Template for Blog

27 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<div class="container-fluid">

<a class="navbar-brand" href="#">Home</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="#">Link</a>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

{% block body\_block %}

{% endblock %}

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

28 – templates => App\_Blog => blog\_list.html

{% extends 'base.html' %}

{% block title %} BlogList {% endblock %}

{% block body\_block %}

<h1>Blog List Home Page</h1>

{% endblock %}

29 – App\_Blog => views.py

from django.shortcuts import render

def blog\_list(request):

return render(request, 'App\_Blog/blog\_list.html', context={})

30 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.blog\_list, name='blog\_list')

]

31 – MyBlog => views.py

from django.http import HttpResponse

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

def index(request):

return HttpResponseRedirect(reverse('App\_Blog:blog\_list'))

32 – VISIT -> <http://127.0.0.1:8000> -> Automatic Move TO -> <http://127.0.0.1:8000/blog/>

# User Sign Up

33 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm

def sign\_up(request):

form = UserCreationForm()

registered = False

if request.method == 'POST':

form = UserCreationForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

34 – App\_Login => urls.py

from django.urls import path

from . import views

urlpatterns = [

path('signup/', views.sign\_up, name='signup')

]

35 – templates => App\_Login => singup.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Account {% endblock %}

{% block body\_block %}

{% if registered %}

<div class="alert alert-success">

Account Create Successfully <a href="#">Login Here</a>

</div>

{% else %}

<h1>Create New Account</h1>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-success">Create Account</button>

</form>

<br>

<p>Already have an account? <a href="#">Login Here</a></p>

{% endif %}

{% endblock %}

36 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'signup' %}">Sign Up</a>

</li>

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

37 - MyBlog>pip install django-crispy-forms

38 – settings.py

…

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'App\_Login',

'App\_Blog',

'crispy\_forms',

]

# bootstrap, bootstrap3, bootstrap4, uni-form

CRISPY\_TEMPLATE\_PACK = 'bootstrap4'

…

39 – VISIT -> <http://127.0.0.1:8000/admin/auth/user/>

# Login

40 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

def sign\_up(request):

form = UserCreationForm()

registered = False

if request.method == 'POST':

form = UserCreationForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

41 – App\_Login => ulrs.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

]

42 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

{% if user.is\_authenticated %}

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Logout</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

43 – templates => App\_Login => login.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Login {% endblock %}

{% block body\_block %}

<h2>Login Your Account</h2>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary">Login</button>

</form>

{% endblock %}

# User Profile

44 – App\_Login => forms.py

from django import forms

from django.contrib.auth.forms import UserCreationForm

from django.contrib.auth.models import User

class SignUpForm(UserCreationForm):

email = forms.EmailField(label="Email Address", required=True)

class Meta:

model = User

fields = ('username', 'email', 'password1', 'password2')

45 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

46 – templates => App\_Login => login.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Login {% endblock %}

{% block body\_block %}

<h2>Login Your Account</h2>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary">Login</button>

</form>

<p>Don't have an account? <a href="{% url 'App\_Login:signup' %}">SignUp Here</a></p>

{% endblock %}

47 – templates => App\_Login => profile.html

{% extends 'base.html' %}

{% block title %} Profile {% endblock %}

{% block body\_block %}

<h2>Your Profile</h2>

<p>UserName: {{ user.username }}</p>

<p>Email: {{ user.email }}</p>

<p>Name: {{ user.first\_name }} {{ user.last\_name }}</p>

{% endblock %}

48 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

@login\_required

def profile(request):

return render(request, 'App\_Login/profile.html', context={})

49 – App\_Login => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

path('profile/', views.profile, name='profile'),

]

50 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

{% if user.is\_authenticated %}

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:profile' %}">Profile</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Logout</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# User Change Profile

51 – App\_Login => forms.py

from django import forms

from django.contrib.auth.forms import UserCreationForm, UserChangeForm

from django.contrib.auth.models import User

class SignUpForm(UserCreationForm):

email = forms.EmailField(label="Email Address", required=True)

class Meta:

model = User

fields = ('username', 'email', 'password1', 'password2')

class UserProfileChange(UserChangeForm):

class Meta:

model = User

fields = ('username', 'email', 'first\_name', 'last\_name')

52 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm, UserProfileChange

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

@login\_required

def profile(request):

return render(request, 'App\_Login/profile.html', context={})

@login\_required

def user\_change(request):

current\_user = request.user

form = UserProfileChange(instance=current\_user)

if request.method == 'POST':

form = UserProfileChange(request.POST, instance=current\_user)

if form.is\_valid():

form.save()

form = UserProfileChange(instance=current\_user)

return render(request, 'App\_Login/change\_profile.html', context={'form': form})

53 – App\_Login => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

path('profile/', views.profile, name='profile'),

path('change-profile/', views.user\_change, name='user\_change'),

]

54 – templates => App\_Login => profile.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Change\_User {% endblock %}

{% block body\_block %}

<h2>User Profile Change</h2>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary">Change Profile</button>

</form>

{% endblock %}

55 – templates => profile.html

{% extends 'base.html' %}

{% block title %} Profile {% endblock %}

{% block body\_block %}

<h2>Your Profile</h2>

<p>UserName: {{ user.username }}</p>

<p>Email: {{ user.email }}</p>

<p>Name: {{ user.first\_name }} {{ user.last\_name }}</p>

<br>

<a href="{% url 'App\_Login:user\_change' %}" class="btn btn-warning">Change Profile Info</a>

{% endblock %}

56 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

{% if user.is\_authenticated %}

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:profile' %}">Profile</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Logout</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

# Change Password

57 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm, PasswordChangeForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm, UserProfileChange

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

@login\_required

def profile(request):

return render(request, 'App\_Login/profile.html', context={})

@login\_required

def user\_change(request):

current\_user = request.user

form = UserProfileChange(instance=current\_user)

if request.method == 'POST':

form = UserProfileChange(request.POST, instance=current\_user)

if form.is\_valid():

form.save()

form = UserProfileChange(instance=current\_user)

return render(request, 'App\_Login/change\_profile.html', context={'form': form})

@login\_required

def pass\_change(request):

current\_user = request.user

changed = False

form = PasswordChangeForm(current\_user)

if request.method == 'POST':

form = PasswordChangeForm(current\_user, data=request.POST)

if form.is\_valid():

form.save()

changed = True

return render(request, 'App\_Login/pass\_change.html', context={'form': form, 'changed': changed})

58 – App\_Login => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

path('profile/', views.profile, name='profile'),

path('change-profile/', views.user\_change, name='user\_change'),

path('password/', views.pass\_change, name='pass\_change'),

]

59 – templates => App\_Login => pass\_change.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Password\_Change {% endblock %}

{% block body\_block %}

{% if changed %}

<div class="alert alert-success">

Password Change Successfully

</div>

{% endif %}

<h2>Password Change</h2>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary">Change Password</button>

</form>

{% endblock %}

# User Profile Pic

60 – MyBlog => urls.py

from django.contrib import admin

from django.urls import path, include

from django.conf import settings

from django.contrib.staticfiles.urls import static, staticfiles\_urlpatterns

from . import views

urlpatterns = [

path('admin/', admin.site.urls),

path('account/', include('App\_Login.urls')),

path('blog/', include('App\_Blog.urls')),

path('', views.index, name='index')

]

urlpatterns += staticfiles\_urlpatterns()

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

61 – App\_Login => forms.py

from django import forms

from django.contrib.auth.forms import UserCreationForm, UserChangeForm

from django.contrib.auth.models import User

from App\_Login.models import UserProfile

class SignUpForm(UserCreationForm):

email = forms.EmailField(label="Email Address", required=True)

class Meta:

model = User

fields = ('username', 'email', 'password1', 'password2')

class UserProfileChange(UserChangeForm):

class Meta:

model = User

fields = ('username', 'email', 'first\_name', 'last\_name')

class ProfilePic(forms.ModelForm):

class Meta:

model = UserProfile

fields = ['profile\_pic']

62 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm, PasswordChangeForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm, UserProfileChange, ProfilePic

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

@login\_required

def profile(request):

return render(request, 'App\_Login/profile.html', context={})

@login\_required

def user\_change(request):

current\_user = request.user

form = UserProfileChange(instance=current\_user)

if request.method == 'POST':

form = UserProfileChange(request.POST, instance=current\_user)

if form.is\_valid():

form.save()

form = UserProfileChange(instance=current\_user)

return render(request, 'App\_Login/change\_profile.html', context={'form': form})

@login\_required

def pass\_change(request):

current\_user = request.user

changed = False

form = PasswordChangeForm(current\_user)

if request.method == 'POST':

form = PasswordChangeForm(current\_user, data=request.POST)

if form.is\_valid():

form.save()

changed = True

return render(request, 'App\_Login/pass\_change.html', context={'form': form, 'changed': changed})

@login\_required

def add\_pro\_pic(request):

form = ProfilePic()

return render(request, 'App\_Login/pro\_pic\_add.html', context={'form': form})

63 – App\_Login => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

path('profile/', views.profile, name='profile'),

path('change-profile/', views.user\_change, name='user\_change'),

path('password/', views.pass\_change, name='pass\_change'),

path('ChangeProfilePic/', views.add\_pro\_pic, name='add\_pro\_pic'),

]

64 – templates => App\_Login => profile.html

{% extends 'base.html' %}

{% block title %} Profile {% endblock %}

{% block body\_block %}

<h2>Your Profile</h2>

{% if user.user\_profile %}

<img src="/media/{{ user.user\_profile.profile\_pic }}" alt="" class="rounded-circle">

{% else %}

<img src="/media/profile.png" alt="" class="rounded-circle" width="120px">

{% endif %}

<br>

<a href="{% url 'App\_Login:add\_pro\_pic' %}" class="btn btn-info btn-sm">Change Profile Picture</a>

<br>

<p>UserName: {{ user.username }}</p>

<p>Email: {{ user.email }}</p>

<p>Name: {{ user.first\_name }} {{ user.last\_name }}</p>

<br>

<a href="{% url 'App\_Login:user\_change' %}" class="btn btn-warning">Change Profile Info</a>

{% endblock %}

65 – templates => App\_Login => pro\_pic\_add.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} Profile Pic Update {% endblock %}

{% block body\_block %}

<h2>Add Your Profile Picture</h2>

<form method="post" enctype="multipart/form-data">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary btn-sm">Add Picture</button>

</form>

{% endblock %}

66 - MyBlog>pip install django-cleanup

67 – settings.py

…

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'App\_Login',

'App\_Blog',

'crispy\_forms',

'django\_cleanup.apps.CleanupConfig',

]

…

68 – App\_Login => views.py

from django.shortcuts import render

from django.contrib.auth.forms import UserCreationForm, AuthenticationForm, PasswordChangeForm

from django.contrib.auth import login, authenticate, logout

from django.shortcuts import HttpResponseRedirect

from django.urls import reverse

from django.contrib.auth.decorators import login\_required

from App\_Login.forms import SignUpForm, UserProfileChange, ProfilePic

def sign\_up(request):

form = SignUpForm()

registered = False

if request.method == 'POST':

form = SignUpForm(data=request.POST)

if form.is\_valid():

form.save()

registered = True

dict = {'form':form, 'registered':registered}

return render(request, 'App\_Login/singup.html', context=dict)

def login\_page(request):

form = AuthenticationForm()

if request.method == 'POST':

form = AuthenticationForm(data=request.POST)

if form.is\_valid():

username = form.cleaned\_data.get('username')

password = form.cleaned\_data.get('password')

user = authenticate(username=username, password=password)

if user is not None:

login(request, user)

return HttpResponseRedirect(reverse('index'))

return render(request, 'App\_Login/login.html', context={'form': form})

@login\_required

def logout\_user(request):

logout(request)

return HttpResponseRedirect(reverse('login'))

@login\_required

def profile(request):

return render(request, 'App\_Login/profile.html', context={})

@login\_required

def user\_change(request):

current\_user = request.user

form = UserProfileChange(instance=current\_user)

if request.method == 'POST':

form = UserProfileChange(request.POST, instance=current\_user)

if form.is\_valid():

form.save()

form = UserProfileChange(instance=current\_user)

return render(request, 'App\_Login/change\_profile.html', context={'form': form})

@login\_required

def pass\_change(request):

current\_user = request.user

changed = False

form = PasswordChangeForm(current\_user)

if request.method == 'POST':

form = PasswordChangeForm(current\_user, data=request.POST)

if form.is\_valid():

form.save()

changed = True

return render(request, 'App\_Login/pass\_change.html', context={'form': form, 'changed': changed})

@login\_required

def add\_pro\_pic(request):

form = ProfilePic()

if request.method == 'POST':

form = ProfilePic(request.POST, request.FILES)

if form.is\_valid():

user\_obj = form.save(commit=False)

user\_obj.user = request.user

user\_obj.save()

return HttpResponseRedirect(reverse('App\_Login:profile'))

return render(request, 'App\_Login/pro\_pic\_add.html', context={'form': form})

@login\_required

def change\_pro\_pic(request):

form = ProfilePic(instance=request.user.user\_profile)

if request.method == 'POST':

form = ProfilePic(request.POST, request.FILES, instance=request.user.user\_profile)

if form.is\_valid():

form.save()

return HttpResponseRedirect(reverse('App\_Login:profile'))

return render(request, 'App\_Login/pro\_pic\_add.html', context={'form': form})

69 – App\_Login => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Login'

urlpatterns = [

path('signup/', views.sign\_up, name='signup'),

path('login/', views.login\_page, name='login'),

path('logout/', views.logout\_user, name='logout'),

path('profile/', views.profile, name='profile'),

path('change-profile/', views.user\_change, name='user\_change'),

path('password/', views.pass\_change, name='pass\_change'),

path('ChangeProfilePic/', views.add\_pro\_pic, name='add\_pro\_pic'),

path('Change-picture/', views.change\_pro\_pic, name='change\_pro\_pic'),

]

70 – templates => App\_Login => profile.html

{% extends 'base.html' %}

{% block title %} Profile {% endblock %}

{% block body\_block %}

<h2>Your Profile</h2>

{% if user.user\_profile %}

<img src="/media/{{ user.user\_profile.profile\_pic }}" width="120px" alt="" class="rounded-circle">

<br>

<a href="{% url 'App\_Login:change\_pro\_pic' %}" class="btn btn-info btn-sm">Change Profile Picture</a>

<br>

{% else %}

<img src="/media/profile.png" alt="" class="rounded-circle" width="120px">

{% endif %}

<br>

<a href="{% url 'App\_Login:add\_pro\_pic' %}" class="btn btn-info btn-sm">Add Profile Picture</a>

<br>

<p>UserName: {{ user.username }}</p>

<p>Email: {{ user.email }}</p>

<p>Name: {{ user.first\_name }} {{ user.last\_name }}</p>

<br>

<a href="{% url 'App\_Login:user\_change' %}" class="btn btn-warning btn-sm">Change Profile Info</a>

{% endblock %}

# Blog

71 – templates => App\_Blog => create\_blog.html

{% extends 'base.html' %}

{% block title %} CreateBlog {% endblock %}

{% block body\_block %}

<h1>Create Blog</h1>

{% endblock %}

72 – templates => App\_Blog => blog\_details.html

{% extends 'base.html' %}

{% block title %} BlogDetails {% endblock %}

{% block body\_block %}

<h1>Blog Details</h1>

{% endblock %}

73 – templates => App\_Blog => edit\_blog.html

{% extends 'base.html' %}

{% block title %} EditBlog {% endblock %}

{% block body\_block %}

<h1>Edit Blog</h1>

{% endblock %}

74 – templates => App\_Blog => my\_blogs.html

{% extends 'base.html' %}

{% block title %} MyBlog {% endblock %}

{% block body\_block %}

<h1>My Blog</h1>

{% endblock %}

75 – templates => App\_Blog => blog\_list.html

{% extends 'base.html' %}

{% block title %} BlogList {% endblock %}

{% block body\_block %}

<h1>Blog List Home Page</h1>

{% endblock %}

76 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

{% if user.is\_authenticated %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Blog:create\_blog' %}">Write Blog</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

{% if user.is\_authenticated %}

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

{{ user.username }}

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="{% url 'App\_Login:profile' %}">Your Profile</a></li>

<li><a class="dropdown-item" href=""></a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="{% url 'App\_Login:login' %}">Logout</a></li>

</ul>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

77 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

def blog\_list(request):

return render(request, 'App\_Blog/blog\_list.html', context={})

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = '\_\_all\_\_'

78 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.blog\_list, name='blog\_list'),

path('write/', views.CreateBlog.as\_view(), name='create\_blog'),

]

79 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

def blog\_list(request):

return render(request, 'App\_Blog/blog\_list.html', context={})

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

80 – templates => App\_Blog => create\_blog.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} CreateBlog {% endblock %}

{% block body\_block %}

<h2>Write Your Blog Here:</h2>

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<br>

<button type="submit" class="btn btn-success btn-sm">Publish</button>

</form>

{% endblock %}

# Blog HomePage

81 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

class BlogList(ListView):

context\_object\_name = 'blogs'

model = Blog

template\_name = 'App\_Blog/blog\_list.html'

queryset = Blog.objects.order\_by('-publish\_date')

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

82 – App\_Blog => models.py

from django.db import models

from django.contrib.auth.models import User

class Blog(models.Model):

author = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='post\_author')

blog\_title = models.CharField(max\_length=264, verbose\_name="Put a Title")

slug = models.SlugField(max\_length=264, unique=True)

blog\_content = models.TextField(verbose\_name="What is on your mind?")

blog\_image = models.ImageField(upload\_to='blog\_images', verbose\_name="Blog Image")

publish\_date = models.DateTimeField(auto\_now\_add=True)

update\_date = models.DateTimeField(auto\_now=True)

class Meta:

ordering = ['publish\_date']

def \_\_str\_\_(self):

return self.blog\_title

class Comment(models.Model):

blog = models.ForeignKey(Blog, on\_delete=models.CASCADE, related\_name='blog\_comment')

user = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='user\_comment')

comment = models.TextField()

comment\_date = models.DateTimeField(auto\_now\_add=True)

class Meta:

ordering = ['-comment\_date']

def \_\_str\_\_(self):

return self.comment

class Likes(models.Model):

blog = models.ForeignKey(Blog, on\_delete=models.CASCADE, related\_name='liked\_blog')

user = models.ForeignKey(User, on\_delete=models.CASCADE, related\_name='liker\_user')

83 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.BlogList.as\_view(), name='blog\_list'),

path('write/', views.CreateBlog.as\_view(), name='create\_blog'),

]

84 – templates => App\_Blog => blog\_list.html

{% extends 'base.html' %}

{% load custom\_filters %}

{% block title %} BlogList {% endblock %}

{% block body\_block %}

<h1>Blog List Home Page</h1>

{% for blog in blogs %}

<h2>{{ blog.blog\_title }}</h2>

<h6><i>{{ blog.publish\_date }}</i></h6>

<div class="row">

<div class="col-sm-4">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" title="{{ blog.blog\_title }}" width="100%">

</div>

<div class="com-sm-8">

<p>{{ blog.blog\_content|linebreaks }}</p>

<h6>Posted By: <a href="">{{ blog.author }}</a></h6>

</div>

</div>

{% endfor %}

{% endblock %}

85 – App\_Blog => templatetags => \_\_init\_\_.py & custom\_filters.py

from django import template

register = template.Library()

def range\_filter(value):

return value[0:600] + "...."

register.filter('range\_filter', range\_filter)

86 – templates => App\_Blog => create\_blog.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} CreateBlog {% endblock %}

{% block body\_block %}

<h2>Write Your Blog Here:</h2>

<form method="post" enctype="multipart/form-data">

{{ form|crispy }}

{% csrf\_token %}

<br>

<button type="submit" class="btn btn-success btn-sm">Publish</button>

</form>

{% endblock %}

# Blog details Page

87 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

class BlogList(ListView):

context\_object\_name = 'blogs'

model = Blog

template\_name = 'App\_Blog/blog\_list.html'

queryset = Blog.objects.order\_by('-publish\_date')

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

def blog\_details(request, slug):

blog = Blog.objects.get(slug=slug)

return render(request, 'App\_Blog/blog\_details.html', context={'blog': blog})

88 – templates => App\_Blog => blog\_details.html

{% extends 'base.html' %}

{% block title %} BlogDetails {% endblock %}

{% block body\_block %}

<h1>Blog Details</h1>

<div class="row">

<div class="col-sm-6">

<h2>{{ blog.blog\_title }}</h2>

<h4>Posted By: {{ blog.author }}</h4>

<i><h6>Published On: {{ blog.publish\_date }}</h6></i>

<i><h6>Updated On: {{ blog.update\_date }}</h6></i>

</div>

<div class="col-sm-6">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" width="100%">

</div>

{{ blog.blog\_content|linebreaks }}

</div>

{% endblock %}

89 – templates => App\_Blog => blog\_list.html

{% extends 'base.html' %}

{% load custom\_filters %}

{% block title %} BlogList {% endblock %}

{% block body\_block %}

<h1>Blog List Home Page</h1>

{% for blog in blogs %}

<h2>{{ blog.blog\_title }}</h2>

<h6><i>{{ blog.publish\_date }}</i></h6>

<div class="row">

<div class="col-sm-4">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" title="{{ blog.blog\_title }}" width="100%">

</div>

<div class="com-sm-8">

<p>{{ blog.blog\_content|linebreaks }}</p>

<a href="{% url 'App\_Blog:blog\_details' slug=blog.slug %}">Read More!</a>

<h6>Posted By: <a href="">{{ blog.author }}</a></h6>

</div>

</div>

{% endfor %}

{% endblock %}

90 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.BlogList.as\_view(), name='blog\_list'),

path('write/', views.CreateBlog.as\_view(), name='create\_blog'),

path('details/<slug:slug>', views.blog\_details, name='blog\_details'),

]

# Blog Comments

91 – App\_Blog => forms.py

from django import forms

from App\_Blog.models import Blog, Comment

class CommentForm(forms.ModelForm):

class Meta:

model = Comment

fields = ('comment',)

92 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

from App\_Blog.forms import CommentForm

class BlogList(ListView):

context\_object\_name = 'blogs'

model = Blog

template\_name = 'App\_Blog/blog\_list.html'

queryset = Blog.objects.order\_by('-publish\_date')

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

def blog\_details(request, slug):

blog = Blog.objects.get(slug=slug)

comment\_form = CommentForm()

if request.method == 'POST':

comment\_form = CommentForm(request.POST)

if comment\_form.is\_valid():

comment = comment\_form.save(commit=False)

comment.user = request.user

comment.blog = blog

comment.save()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': slug}))

return render(request, 'App\_Blog/blog\_details.html', context={'blog': blog, 'comment\_form': comment\_form})

93 – templates => App\_Blog => blog\_details.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} BlogDetails {% endblock %}

{% block body\_block %}

<h1>Blog Details</h1>

<div class="row">

<div class="col-sm-6">

<h2>{{ blog.blog\_title }}</h2>

<h4>Posted By: {{ blog.author }}</h4>

<i><h6>Published On: {{ blog.publish\_date }}</h6></i>

<i><h6>Updated On: {{ blog.update\_date }}</h6></i>

</div>

<div class="col-sm-6">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" width="100%">

</div>

{{ blog.blog\_content|linebreaks }}

</div>

<div class="row">

<div class="col-sm-6">

<h5>Comments:</h5>

{% for comment in blog.blog\_comment.all %}

<p><a href="#">{{ comment.user }}</a></p>

<b>{{ comment.comment }}</b><br>

<h6><i>{{comment.comment\_date}}</i></h6>

{% endfor %}

</div>

</div>

<div class="col-sm-6">

<form method="post">

{{ comment\_form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary btn-sm">Comment</button>

</form>

</div>

{% endblock %}

# Likes | Unlikes

94 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.3/css/font-awesome.min.css">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

{% if user.is\_authenticated %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Blog:create\_blog' %}">Write Blog</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

{% if user.is\_authenticated %}

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

{{ user.username }}

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="{% url 'App\_Login:profile' %}">Your Profile</a></li>

<li><a class="dropdown-item" href=""></a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="{% url 'App\_Login:login' %}">Logout</a></li>

</ul>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

95 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

from App\_Blog.forms import CommentForm

class BlogList(ListView):

context\_object\_name = 'blogs'

model = Blog

template\_name = 'App\_Blog/blog\_list.html'

queryset = Blog.objects.order\_by('-publish\_date')

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

def blog\_details(request, slug):

blog = Blog.objects.get(slug=slug)

comment\_form = CommentForm()

already\_liked = Likes.objects.filter(blog=blog, user=request.user)

if already\_liked:

liked = True

else:

liked = False

if request.method == 'POST':

comment\_form = CommentForm(request.POST)

if comment\_form.is\_valid():

comment = comment\_form.save(commit=False)

comment.user = request.user

comment.blog = blog

comment.save()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': blog.slug}))

return render(request, 'App\_Blog/blog\_details.html', context={'blog': blog, 'comment\_form': comment\_form, 'liked': liked})

@login\_required()

def liked(request, pk):

blog = Blog.objects.get(pk=pk)

user = request.user

already\_liked = Likes.objects.filter(blog=blog, user=user)

if not already\_liked:

liked\_post = Likes(blog=blog, user=user)

liked\_post.save()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': blog.slug}))

@login\_required

def unliked(request, pk):

blog = Blog.objects.get(pk=pk)

user = request.user

already\_liked = Likes.objects.filter(blog=blog, user=user)

already\_liked.delete()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': slug}))

96 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.BlogList.as\_view(), name='blog\_list'),

path('write/', views.CreateBlog.as\_view(), name='create\_blog'),

path('details/<slug:slug>', views.blog\_details, name='blog\_details'),

path('liked/<pk>', views.liked, name='liked\_post'),

path('unliked/<pk>', views.unliked, name='unliked\_post'),

]

97 – templates => App\_Blog => blog\_details.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} BlogDetails {% endblock %}

{% block body\_block %}

<h1>Blog Details</h1>

<div class="row">

<div class="col-sm-6">

<h2>{{ blog.blog\_title }}</h2>

<h4>Posted By: {{ blog.author }}</h4>

<i><h6>Published On: {{ blog.publish\_date }}</h6></i>

<i><h6>Updated On: {{ blog.update\_date }}</h6></i>

<h6>{{ blog.liked\_blog.count }} Likes | {{ blog.blog\_comment.count }} Comments</h6>

</div>

<div class="col-sm-6">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" width="100%">

</div>

{{ blog.blog\_content|linebreaks }}

</div>

<hr>

{% if not liked %}

<a href="{% url 'App\_Blog:liked\_post' pk=blog.pk %}">

<i class="fa fa-thumbs-o-up fa-2x"></i>

</a>

{% else %}

<a href="{% url 'App\_Blog:unliked\_post' pk=blog.pk %}">

<i class="fa fa-thumbs-o-down fa-2x"></i>

</a>

{% endif %}

<div class="row">

<div class="col-sm-6">

<h5>Comments:</h5>

{% for comment in blog.blog\_comment.all %}

<p><a href="#">{{ comment.user }}</a></p>

<b>{{ comment.comment }}</b><br>

<h6><i>{{comment.comment\_date}}</i></h6>

{% endfor %}

</div>

</div>

<div class="col-sm-6">

<form method="post">

{{ comment\_form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-primary btn-sm">Comment</button>

</form>

</div>

{% endblock %}

# My Blog Page

98 – App\_Blog => views.py

from django.shortcuts import render, HttpResponseRedirect

from django.views.generic import CreateView, UpdateView, ListView, DetailView, TemplateView, DeleteView

from App\_Blog.models import Blog, Comment, Likes

from django.urls import reverse, reverse\_lazy

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.mixins import LoginRequiredMixin

import uuid

from App\_Blog.forms import CommentForm

class MyBlogs(LoginRequiredMixin, TemplateView):

template\_name = 'App\_Blog/my\_blogs.html'

class UpdateBlog(LoginRequiredMixin, UpdateView):

model = Blog

fields = ('blog\_title', 'blog\_content', 'blog\_image')

template\_name = 'App\_Blog/edit\_blog.html'

def get\_success\_url(self, \*\*kwargs):

return reverse\_lazy('App\_Blog:blog\_details', kwargs={'slug': self.object.slug})

class BlogList(ListView):

context\_object\_name = 'blogs'

model = Blog

template\_name = 'App\_Blog/blog\_list.html'

class CreateBlog(LoginRequiredMixin, CreateView):

model = Blog

template\_name = 'App\_Blog/create\_blog.html'

fields = ('blog\_title', 'blog\_content', 'blog\_image')

def form\_valid(self, form):

blog\_obj = form.save(commit=False)

blog\_obj.author = self.request.user

title = blog\_obj.blog\_title

blog\_obj.slug = title.replace(" ", "-") + "-" + str(uuid.uuid4())

blog\_obj.save()

return HttpResponseRedirect(reverse('index'))

def blog\_details(request, slug):

blog = Blog.objects.get(slug=slug)

comment\_form = CommentForm()

already\_liked = Likes.objects.filter(blog=blog, user=request.user)

if already\_liked:

liked = True

else:

liked = False

if request.method == 'POST':

comment\_form = CommentForm(request.POST)

if comment\_form.is\_valid():

comment = comment\_form.save(commit=False)

comment.user = request.user

comment.blog = blog

comment.save()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug':slug}))

return render(request, 'App\_Blog/blog\_details.html', context={'blog': blog, 'comment\_form': comment\_form, 'liked': liked})

@login\_required

def liked(request, pk):

blog = Blog.objects.get(pk=pk)

user = request.user

already\_liked = Likes.objects.filter(blog=blog, user=user)

if not already\_liked:

liked\_post = Likes(blog=blog, user=user)

liked\_post.save()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': blog.slug}))

@login\_required

def unliked(request, pk):

blog = Blog.objects.get(pk=pk)

user = request.user

already\_liked = Likes.objects.filter(blog=blog, user=user)

already\_liked.delete()

return HttpResponseRedirect(reverse('App\_Blog:blog\_details', kwargs={'slug': blog.slug}))

99 – App\_Blog => urls.py

from django.urls import path

from . import views

app\_name = 'App\_Blog'

urlpatterns = [

path('', views.BlogList.as\_view(), name='blog\_list'),

path('write/', views.CreateBlog.as\_view(), name='create\_blog'),

path('details/<slug:slug>', views.blog\_details, name='blog\_details'),

path('liked/<pk>', views.liked, name='liked\_post'),

path('unliked/<pk>/', views.unliked, name='unliked\_post'),

path('my-blog/', views.MyBlogs.as\_view(), name='my\_blogs'),

path('edit-blog/<pk>/', views.UpdateBlog.as\_view(), name='edit\_blog'),

]

100 – templates => base.html

<!doctype html>

<html lang="en">

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.3/css/font-awesome.min.css">

<title>

{% block title %}

Blog Page

{% endblock %}

</title>

</head>

<body>

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">My Blog</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-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 me-auto mb-2 mb-lg-0">

<li class="nav-item">

<a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a>

</li>

{% if user.is\_authenticated %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Blog:create\_blog' %}">Write Blog</a>

</li>

{% else %}

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:signup' %}">Sign Up</a>

</li>

<li class="nav-item">

<a class="nav-link" href="{% url 'App\_Login:login' %}">Login</a>

</li>

{% endif %}

{% if user.is\_authenticated %}

<li class="nav-item dropdown">

<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

{{ user.username }}

</a>

<ul class="dropdown-menu" aria-labelledby="navbarDropdown">

<li><a class="dropdown-item" href="{% url 'App\_Login:profile' %}">Your Profile</a></li>

<li><a class="dropdown-item" href="{% url 'App\_Blog:my\_blogs' %}">My Blog</a></li>

<li><hr class="dropdown-divider"></li>

<li><a class="dropdown-item" href="{% url 'App\_Login:login' %}">Logout</a></li>

</ul>

</li>

{% endif %}

</ul>

<form class="d-flex">

<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">

<button class="btn btn-outline-success" type="submit">Search</button>

</form>

</div>

</div>

</nav>

<div class="container mt-3">

{% block body\_block %}

{% endblock %}

</div>

<!-- Optional JavaScript; choose one of the two! -->

<!-- Option 1: Bootstrap Bundle with Popper -->

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>

<!-- Option 2: Separate Popper and Bootstrap JS -->

<!--

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js" integrity="sha384-IQsoLXl5PILFhosVNubq5LC7Qb9DXgDA9i+tQ8Zj3iwWAwPtgFTxbJ8NT4GN1R8p" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.min.js" integrity="sha384-cVKIPhGWiC2Al4u+LWgxfKTRIcfu0JTxR+EQDz/bgldoEyl4H0zUF0QKbrJ0EcQF" crossorigin="anonymous"></script>

-->

</body>

</html>

101 – templates => App\_Blog => blog\_list.html

{% extends 'base.html' %}

{% load custom\_filters %}

{% block title %} BlogList {% endblock %}

{% block body\_block %}

<h1>Blog List Home Page</h1>

{% for blog in blogs %}

<h2>{{ blog.blog\_title }}</h2>

<h6><i>{{ blog.publish\_date }}</i></h6>

<div class="row">

<div class="col-sm-4">

<img src="/media/{{ blog.blog\_image }}" alt="{{ blog.blog\_title }}" title="{{ blog.blog\_title }}" width="100%">

</div>

<div class="com-sm-8">

<p>{{ blog.blog\_content|linebreaks }}

<a href="{% url 'App\_Blog:blog\_details' slug=blog.slug %}">Read More!</a>

</p>

<h6>Posted By: <a href="">{{ blog.author }}</a></h6>

<h6>{{ blog.liked\_blog.count }} Likes | {{ blog.blog\_comment.count }} Comments</h6>

</div>

</div>

{% endfor %}

{% endblock %}

102 – templates => App\_Blog => my\_blogs.html

{% extends 'base.html' %}

{% load custom\_filters %}

{% block title %} My Blog {% endblock %}

{% block body\_block %}

{% for blog in user.post\_author.all %}

<h3>{{ blog.blog\_title }}</h3>

<h6>{{ blog.publish\_date }}</h6>

<div class="row">

<div class="col-sm-4">

<img src="/media/{{blog.blog\_image}}" width="100%">

</div>

<div class="col-sm-8">

<p>{{ blog.blog\_content|range\_filter|linebreaks }}

<a href="{% url 'App\_Blog:blog\_details' slug=blog.slug %}">Read More</a>

</p>

<h6>{{ blog.liked\_blog.count }} Likes | {{ blog.blog\_comment.count }} Comments</h6>

</div>

</div>

<hr>

<a href="{% url 'App\_Blog:edit\_blog' pk=blog.pk %}" class="btn btn-info btn-sm">Edit</a>

{% endfor %}

{% endblock %}

103 – templates => App\_Blog => edit\_blog.html

{% extends 'base.html' %}

{% load crispy\_forms\_tags %}

{% block title %} EditBlog {% endblock %}

{% block body\_block %}

<h1>Edit Blog</h1>

{% if blog.author == user %}

<form method="post">

{{ form|crispy }}

{% csrf\_token %}

<button type="submit" class="btn btn-info btn-sm">Update</button>

</form>

{% else %}

<p>You are not allowed to vie this page</p>

{% endif %}

{% endblock %}