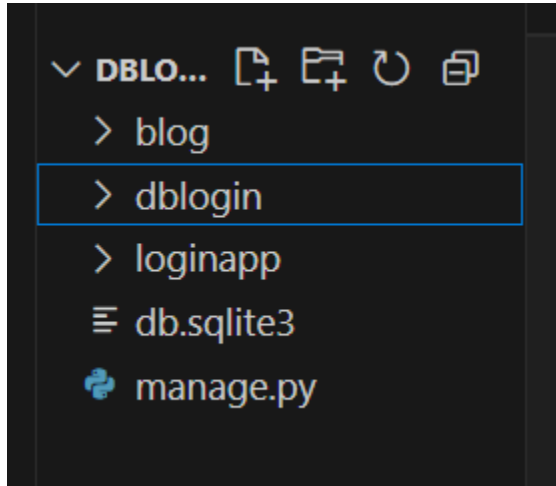


Creating simple blog posting Application with login and logout:-

Project Structure :-



urls.py (of project folder dblogin):-

```
from django.urls import path,include
from loginapp import views

urlpatterns = [
    path('', include('blog.urls')),
    path('signup/', views.signup_view, name='signup'),
    path('login/', views.login_view, name='login'),
    path('logout/', views.logout_view, name='logout'),
]
```

models.py file of blog app:-

```
from django.db import models
from django.contrib.auth.models import User
class Post(models.Model):
    title = models.CharField(max_length=200)
    content = models.TextField()
    author = models.ForeignKey(User, on_delete=models.CASCADE)
```

```
created_at = models.DateTimeField(auto_now_add=True)
updated_at = models.DateTimeField(auto_now=True)
def __str__(self):
    return self.title
```

forms.py file:-

```
from django import forms
from .models import Post
class PostForm(forms.ModelForm):
    class Meta:
        model = Post
        fields = ['title', 'content']
```

urls.py of blog app:-

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('post/new/', views.create_post, name='create_post'),
    path('post/<int:pk>/edit/', views.update_post, name='update_post'),
    path('post/<int:pk>/delete/', views.delete_post,
name='delete_post'),
]
```

views.py of blog app:-

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Post
from .forms import PostForm
from django.contrib.auth.decorators import login_required

from django.shortcuts import render, redirect, get_object_or_404
from django.contrib.auth.decorators import login_required
from .models import Post
from .forms import PostForm

def home(request):
    posts = Post.objects.all()
    return render(request, 'home.html', {'posts': posts})
```

```

@login_required
def create_post(request):
    if request.method == 'POST':
        form = PostForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.author = request.user
            post.save()
            return redirect('home')
    else:
        form = PostForm()
    return render(request, 'post_form.html', {'form': form})

```

```

@login_required
def update_post(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if request.method == 'POST':
        form = PostForm(request.POST, instance=post)
        if form.is_valid():
            form.save()
            return redirect('home')
    else:
        form = PostForm(instance=post)
    return render(request, 'post_form.html', {'form': form})

```

```

@login_required
def delete_post(request, pk):
    post = get_object_or_404(Post, pk=pk)
    if request.method == 'POST':
        post.delete()
        return redirect('home')
    return render(request, 'post_confirm_delete.html', {'post': post})

```

blog/templates files:-

home.html :-

```

<!DOCTYPE html>
<html>

```

```

<head>
  <title>Blog Home</title>
</head>
<body>
  <h1>All Posts</h1>

  {% if user.is_authenticated %}
    <a href="{% url 'create_post' %}">Create New Post</a>
  {% endif %}

  <ul>
    {% for post in posts %}
      <li>
        <h2>{{ post.title }}</h2>
        <p>{{ post.content }}</p>
        <p><em>By {{ post.author }}</em></p>

        {% if user == post.author %}
          <a href="{% url 'update_post' post.pk %}">Edit</a> |
          <a href="{% url 'delete_post' post.pk %}">Delete</a>
        {% endif %}
      </li>
      {% empty %}
        <li>No posts yet.</li>
      {% endfor %}
    </ul>
</body>
</html>

```

post_form.html file :-

```

<!DOCTYPE html>
<html>
<head>
  <title>{% if form.instance.pk %}Edit{% else %}Create{% endif %} Post</title>
</head>
<body>
  <h1>{% if form.instance.pk %}Edit{% else %}Create{% endif %} Post</h1>

  <form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save</button>
  </form>
</body>
</html>

```

```
</form>

  <a href="{% url 'home' %}">Back to Home</a>
</body>
</html>
```

post_confirm_delete.html :-

```
<!DOCTYPE html>
<html>
<head>
  <title>Delete Post</title>
</head>
<body>
  <h1>Are you sure you want to delete this post?</h1>
  <h2>{{ post.title }}</h2>
  <p>{{ post.content }}</p>

  <form method="POST">
    {% csrf_token %}
    <button type="submit">Yes, delete</button>
    <a href="{% url 'home' %}">Cancel</a>
  </form>
</body>
</html>
```

Loginapp (loginapp folder files with code):-

forms.py file:-

```
# forms.py
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm

class SignupForm(UserCreationForm):
    email = forms.EmailField(required=True)

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

views.py file :-

```
# views.py
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib import messages
from .forms import SignupForm
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth.decorators import login_required

def signup_view(request):
    if request.method == 'POST':
        form = SignupForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Account created successfully!')
            return redirect('login')
    else:
        form = SignupForm()
    return render(request, 'signup.html', {'form': form})
```

```

def login_view(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            user = authenticate(request, username=username, password=password)
            if user is not None:
                login(request, user)
                return redirect('home')
            else:
                messages.error(request, 'Invalid username or password.')
        else:
            messages.error(request, 'Invalid form.')
    else:
        form = AuthenticationForm()
    return render(request, 'login.html', {'form': form})

def logout_view(request):
    logout(request)
    return redirect('login')

@login_required
def home(request):
    return render(request, 'home.html')

```

Loginapp/templates :-

login.html file :-

```

<h2>Login</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Login</button>
</form>
<a href="{% url 'signup' %}">Don't have an account? Sign up</a>

```

signup.html file :-

```
<h2>Sign Up</h2>
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Sign Up</button>
</form>
<a href="{% url 'login' %}">Already have an account? Login</a>
```

Home1.html file code:-

```
<h2>Welcome, {{ request.user.username }}!</h2>
<a href="{% url 'logout' %}">Logout</a>
```