Building Post Management Features in Django

Create a Post – Show a form for the user to enter the title and content.
Display User's Posts – Show a list of posts created by the logged-in user.
Edit a Post – Allow users to update their posts.
Soft Delete a Post – Hide (instead of deleting) the post.

---

1 Create a Post (Form & Button)

Step 1: Create a Form (forms.py)

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

Step 2: Add a Button to Show the Form (post_list.html)

```
<a href="{% url 'create_post' %}">
    <button>Create New Post</button>
</a>
```

Step 3: Create a View to Handle Post Creation (views.py)

```
from django.shortcuts import render, redirect
from .forms import PostForm
from .models import Post


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  # Assign logged-in user as author
```

```
            post.save()
            return redirect('post_list')  # Redirect to post list
    else:
        form = PostForm()
 return render(request, 'create_post.html', {'form': form})
```

Step 4: Create a Template for the Form (create_post.html)

```
<h2>Create a New Post</h2>
<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Save Post</button>
</form>
```

---

2 Display All Posts Created by the User

1: Create a View to Show Posts (views.py)

```
def post_list(request):
    posts = Post.objects.filter(author=request.user, is_deleted=False)  # Show only active posts
    return render(request, 'post_list.html', {'posts': posts})
```

Step 2: Create a Template to Display Posts (post_list.html)

```
<h2>Your Posts</h2>
<a href="{% url 'create_post' %}">
    <button>Create New Post</button>
</a>
<ul>
    {% for post in posts %}
        <li>
            <h3>{{ post.title }}</h3>
```

```
        <p>{{ post.content }}</p>
        <a href="{% url 'edit_post' post.id %}">Edit</a> |
        <a href="{% url 'delete_post' post.id %}">Delete</a>
      </li>
  {% endfor %}
</ul>
```

3Edit a Post

Step 1: Create a View to Handle Editing (views.py)

```
def edit_post(request, post_id):
  post = Post.objects.get(id=post_id, author=request.user)

  if request.method == "POST":
    form = PostForm(request.POST, instance=post)
    if form.is_valid():
      form.save()
      return redirect('post_list')  # Redirect to post list
  else:
    form = PostForm(instance=post)  # Load existing data

  return render(request, 'edit_post.html', {'form': form})
```

Step 2: Create an Edit Template (edit_post.html)

```
html
CopyEdit
<h2>Edit Post</h2>
<form method="POST">
  {% csrf_token %}
```

```
    {{ form.as_p }}
    <button type="submit">Update Post</button>
</form>
```

Soft Delete a Post (Hide Instead of Removing from Database)

Step 1: Modify the Post Model (models.py)

Add a is_deleted field to mark posts as deleted instead of removing them.

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

class Post(models.Model):
    title = models.CharField(max_length=255)
    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)
    is_deleted = models.BooleanField(default=False)  # Soft delete field
```

Step 2: Create a View to Soft Delete a Post (views.py)

```
def delete_post(request, post_id):
    post = Post.objects.get(id=post_id, author=request.user)
    post.is_deleted = True  # Mark post as deleted
    post.save()
    return redirect('post_list')  # Redirect to post list
```

Step 3: Modify the Post List View (views.py)

Ensure only **active posts** are shown.

```
def post_list(request):
    posts = Post.objects.filter(author=request.user, is_deleted=False)  # Exclude deleted posts
```

```
    return render(request, 'post_list.html', {'posts': posts})
```

Final URLs (urls.py)

```
from django.urls import path
from .views import create_post, post_list, edit_post, delete_post
urlpatterns = [
    path('create/', create_post, name='create_post'),
    path('', post_list, name='post_list'),
    path('edit/<int:post_id>/', edit_post, name='edit_post'),
    path('delete/<int:post_id>/', delete_post, name='delete_post'),
]
```