

Create a functionality or button to display Post Creation form where User/Author can fill the title and content to create the post.

```
1  ### 📁 models.py
2  from django.db import models
3  from django.contrib.auth.models import User
4
5  class Post(models.Model):
6      author = models.ForeignKey(User, on_delete=models.CASCADE)
7      title = models.CharField(max_length=200)
8      content = models.TextField()
9      created_at = models.DateTimeField(auto_now_add=True)
10     updated_at = models.DateTimeField(auto_now=True)
11     is_deleted = models.BooleanField(default=False) # Soft delete flag
12
13     def __str__(self):
14         return self.title
15
```

```
1  ### 📁 forms.py
2  from django import forms
3  from .models import Post
4
5  class PostForm(forms.ModelForm):
6      class Meta:
7          model = Post
8          fields = ['title', 'content']
9          widgets = {
10              'title': forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Enter post title'}),
11              'content': forms.Textarea(attrs={'class': 'form-control', 'placeholder': 'Write your content here...'}),
12          }
13
```

```
1  ### 🌐 urls.py
2  from django.urls import path
3  from . import views
4
5  urlpatterns = [
6      path('', views.post_list, name='post_list'),
7      path('create/', views.create_post, name='create_post'),
8      path('edit/<int:post_id>/', views.edit_post, name='edit_post'),
9      path('delete/<int:post_id>/', views.delete_post, name='delete_post'),
10 ]
11
```

```

1  # Create a new post
2  @login_required
3  def create_post(request):
4      if request.method == 'POST':
5          form = PostForm(request.POST)
6          if form.is_valid():
7              post = form.save(commit=False)
8              post.author = request.user
9              post.save()
10             return redirect('post_list')
11         else:
12             form = PostForm()
13         return render(request, 'blog/create_post.html', {'form': form})

```

To display all the Post create by the user.

```

1  ### 🚀 views.py
2  from django.shortcuts import render, redirect, get_object_or_404
3  from .models import Post
4  from .forms import PostForm
5  from django.contrib.auth.decorators import login_required
6
7  # Display all posts
8  @login_required
9  def post_list(request):
10     posts = Post.objects.filter(author=request.user, is_deleted=False).order_by('-created_at')
11     return render(request, 'blog/post_list.html', {'posts': posts})

```

```

1  ###  templates/blog/post_list.html
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <title>My Posts</title>
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
8  </head>
9  <body class="container mt-5">
10     <h2>My Posts</h2>
11     <a href="{% url 'create_post' %}" class="btn btn-success mb-3">+ Create New Post</a>
12     <ul class="list-group">
13         {% for post in posts %}
14             <li class="list-group-item">
15                 <h4>{{ post.title }}</h4>
16                 <p>{{ post.content }}</p>
17                 <small>Created at: {{ post.created_at }}</small>
18                 <br>
19                 <a href="{% url 'edit_post' post.id %}" class="btn btn-primary btn-sm">Edit</a>
20                 <a href="{% url 'delete_post' post.id %}" class="btn btn-danger btn-sm">Delete</a>
21             </li>
22             {% empty %}
23                 <li class="list-group-item">No posts found.</li>
24             {% endfor %}
25     </ul>
26 </body>
27 </html>

```

Functionality to Edit the Post

```

1  # Edit an existing post
2  @login_required
3  def edit_post(request, post_id):
4      post = get_object_or_404(Post, id=post_id, author=request.user, is_deleted=False)
5      if request.method == 'POST':
6          form = PostForm(request.POST, instance=post)
7          if form.is_valid():
8              form.save()
9              return redirect('post_list')
10     else:
11         form = PostForm(instance=post)
12     return render(request, 'blog/edit_post.html', {'form': form})

```

```

1
2 ### 📄 templates/blog/edit_post.html
3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="UTF-8">
7   <title>Edit Post</title>
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
9 </head>
10 <body class="container mt-5">
11   <h2>Edit Post</h2>
12   <form method="POST">
13     {% csrf_token %}
14     {{ form.as_p }}
15     <button type="submit" class="btn btn-primary">Update Post</button>
16   </form>
17 </body>
18 </html>
19

```

Functionality to Delete the Post [Soft delete]

```

1 # Soft delete a post
2 @login_required
3 def delete_post(request, post_id):
4     post = get_object_or_404(Post, id=post_id, author=request.user)
5     post.is_deleted = True # Soft delete
6     post.save()
7     return redirect('post_list')

```