TRAINING DAY17 REPORT

12 JULY 2025

Implementing a Search Bar in an Existing Django Blog Application

A **search bar** in Django allows users to query the database dynamically using a **GET request**. Instead of displaying all posts, the system filters and displays only those that match the user's search input.

The **key concepts** involved are:

- Handling GET requests in Django views.
- Using **query parameters** (request.GET.get()).
- Applying **ORM filters** (__icontains) for case-insensitive searches.
- Dynamically displaying filtered results in templates.

Implementation Steps

1. Existing Blog Model:

Assuming we already have a Post model defined as follows:

blogapi > blog > models.py > ...

```python
# blog/models.py
from django.db import models
from django.contrib.auth.models import User

class Post(models.Model):
    title = models.CharField(max_length=200)
    author = models.ForeignKey(User, on_delete=models.CASCADE)
    content = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.title

```

This model is already migrated and has sample data available.

2. Creating the Search View:

We modify or extend the existing home view to include search functionality.

blogapi > blog > views.py > ...

```python
# blog/views.py
from django.shortcuts import render
from .models import Post

def home(request):
    query = request.GET.get('q')   # Get search query from URL
    if query:
        posts = Post.objects.filter(
            title__icontains=query
        ) | Post.objects.filter(
            content__icontains=query
        )
    else:
        posts = Post.objects.all().order_by('-created_at')

    context = {
        'posts': posts,
        'query': query
    }
    return render(request, 'blog/home.html', context)

```

3. Updating the Template:

We add a search bar to the homepage template.

templates > home.html > ...

```
{% extends "base.html" %}

{% block title %}Blog Name{% endblock %}

{% block content %}

  <h1>Welcome to My Django Blog</h1>

  <form method="GET" action="{% url 'home' %}">
    <input type="text" name="q" placeholder="Search posts..." value="{{ query }}">
    <button type="submit">Search</button>
  </form>

  <hr>

  {% if posts %}
    {% for post in posts %}
      <h2>{{ post.title }}</h2>
      <p>{{ post.content|truncatewords:30 }}</p>
      <small>By {{ post.author.username }} | {{ post.created_at|date:"M d, Y" }}</small>
      <hr>
    {% endfor %}
  {% else %}
    <p>No posts found for your search.</p>
  {% endif %}


{% endblock %}

```

4. URL Configuration:

Ensure the homepage view is linked properly.

blogapi > blog > urls.py > ...

```
# blog/urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
]

```

And in the main project URLs:

```
# blogproject/urls.py
from django.contrib import admin
from django.urls import path, include

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

```