

## search

### Example social media

#### nav.html

```
<form class="form-inline my-2 my-lg-0" action="{% url 'post_list' %}">

    {% if query %}

        <input class="form-control mr-sm-2" type="search" value="{{ query }}"
placeholder="Search" aria-label="Search" name="q">

    {% else %}

        <input class="form-control mr-sm-2" type="search" placeholder="Search" aria-
label="Search" name="q">

    {% endif %}

    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
</form>
```

#### Urls.py

```
path('', views.post_list,name='post_list'),
```

#### Views.py

```
def post_list(request):

    query=request.GET.get('q')

    if query:
        data=Post.objects.filter(
            Q(title__icontains=query)|
            Q(author__username=query)|
            Q(body__icontains=query)

        )
    else:

        data = Post.published.all()

    context={
        'post': data,
        'query' : query
    }

    return render(request,'blog/post_list.html',context)
```

#### Post\_list.html

```
{% extends 'blog/base.html' %}

{% block content %}

    <div class="row">

        {% for i in post %}
```

## search

```
<div class="col-md-6">

<div class="img-thumbnail">

    <a href="{% url 'post_detail' i.id i.slug%}">
        <i>{{ i.title }}</i>
    </a>

    <small style="float: right">{{ i.created }}</small><br>

    Author : {{ i.author }}<br>
    <p>{{ i.body }}</p>

</div>

</div>
<br>

{% empty %}

<div class="container-fluid">

<p>No Posts Found on search : {{ query }}</p>

</div>

{% endfor %}
</div>

{% endblock %}
```

## Models.py

```
class Post(models.Model):
```

```
    objects = models.Manager() # default Manager
    published = PublishedManager() # custom model manager
```

```
    STATUS_CHOICES=(
        ('draft','Draft'),
        ('published','Published')
    )
```

```
    title = models.CharField(max_length=200)
    slug = models.SlugField(max_length=100)
    author = models.ForeignKey(User, related_name="blog_posts", on_delete =
models.DO_NOTHING,)
    body = models.TextField()
    created = models.DateTimeField(auto_now_add=True) # Automatically set the field to now
when the object is first created.
    updated = models.DateTimeField(auto_now=True) #Automatically set the field to now every
time the object is saved.
    status = models.CharField(max_length=100 ,choices=STATUS_CHOICES, default='draft')

    def _str_(self):
        return self.title
```