

آماده سازی مدل پست و ترسیم آن در سایت دی بی دیاگرام و ترسیم مدل یوزر برای دسترسی های پرایمری کی

ساخت اپلیکیشن جدید برای پست

```
class post(models.Model):
    # image
    #author
    title = models.CharField(max_length=255)
    content = models.TextField()
    # tag
    # category
    coounted_viwes = models.IntegerField(default=0)
    status = models.BooleanField(default=False)
    published_date = models.DateTimeField(null=True)
    created_date = models.DateTimeField(auto_now_add=True)
    updated_date = models.DateTimeField(auto_now=True)
```

معرفی متا کلاس برای مدل

```
from django.db import models

class post(models.Model):
    title = models.CharField(max_length=255)
    content = models.TextField()
    coounted_viwes = models.IntegerField(default=0)
    status = models.BooleanField(default=False)
    published_date = models.DateTimeField(null=True)
    created_date = models.DateTimeField(auto_now_add=True)
    updated_date = models.DateTimeField(auto_now=True)
    class Meta:
        ordering = ['-created_date']

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

کاستومايز کردن پنل ادمین برای مدل پست و سپس رجیستر کردن با حالت admin.site.register و دکوراتور

```
@admin.register(post)
class PostAdmin(admin.ModelAdmin):
    list_display = ('id', 'title', 'status', 'created_date') # چه فیلد هایی از آن سطر در صفحه نمایش داده شوند
    list_filter = ('status', ) # افزودن یک گزینه فیلتر بر اساس استتوس ها و یا هر فیلد دیگری
    search_fields = ('title', ) # یک سرچ فیلد میگذارد که یک عبارت را مثلا در ستون تایتل ها سرچ میکند
    empty_value_display = '-empty-' # اگر فیلدی خالی بود خالی نشان ندهد و بجای آن عبارتش را نشان بده
    fields = ('title', 'content', 'published_date') # وقتی بر روی یک سطر کلیک میکنیم کدام بخش ها قابل تغییر باشند
    ordering = ('created_date', ) # بر اساس تاریخ ایجاد صفحه ادمین را سورت میکند
```

```
#admin.site.register(post, PostAdmin)
```

حال وقتی مدل را ساختیم چند پست با وضعیت انتشار ترو و فالس ایجاد میکنیم و میخواهیم حالا در صفحه به دو صورت زیر پست های در حال انتشار را ایجاد میکنیم

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  {% for post in posts %}
    {% if post.status == 1 %}
      {{post.title}}
    {% else %}
      {{post.created_date}}
    {% endif %}
  {% endfor %}
</body>
</html>
```

با استفاده از فیلتراسیون مدل

```
def b_home(req):
    posts=post.objects.filter(status=1)
    context = {
        'posts':posts
    }
```

قبل از اینکه پست ها را در صفحه اصلی نمایش دهیم در یک صفحه تستی نمایش می‌دهیم و سپس محتویات را به صفحه اصلی می‌بریم و سپس طبق فرم زیر آنرا می‌سازیم

```
<div class="col-lg-8 posts-list">
  <div class="single-post row">...
</div>
  <div class="single-post row">...
</div>
  <div class="single-post row">...
</div>
  <div class="single-post row">...
</div>
  <div class="single-post row">...
</div>
</div>
<nav class="blog-pagination justify-content-center d-flex">...
</nav>
</div>
```

