

## هدف از این تمرین: تسلط در نوشتن مدل‌ها | Django 3

شرح تمرین:

فرض کنید یک سایتی داریم و میخواهیم افرادی به عنوان نویسنده در این سایت استخدام کنید. در کنار آن یک سری کاربران عادی نیز در سایتتان ثبت نام میکنند و میتوانند بر روی پست‌ها دیدگاه ثبت کنند.

وظایف نویسندگان ایجاد مطالب علمی، آموزشی و خبری میباشد. پس از ایجاد پست، ادمین‌های سایت بررسی میکنند و در صورتیکه که مطلب مورد تایید بود، وضعیت آن را به حالت قبول تغییر میدهند. برای دیدگاه‌ها هم به همین شکل است.

بازدیدکنندگان سایت میتوانند این پست‌ها را نگاه کنند و کاربران سایت میتوانند برای پست‌ها کامنت (دیدگاه) بنویسند.

در این تمرین از شما میخواهیم که یک مدل Profile ایجاد کنید و آنرا به روش یک به یک به مدل دیفالت User جنگو اتصال کنید تا بتوانیم ویژگی‌های بیشتری را به یوزر اضافه کنیم.

در این مدل نیاز هست فیلد های زیر تعریف شوند (فیلدهای ستاره دار نباید خالی بمانند و حتما باید پر شوند):

- کاربر - OneToOneField (اگر یوزر حذف شد، پروفایل هم حذف شود. یک کلید برای ارتباط معکوس نیز قرار دهید) \*
- زمان عضویت - DateTimeField (پس از ایجاد کاربر، باید این فیلد خودکار پر شود به روز و ساعت آن لحظه)
- سن - PositiveSmallIntegerField \*
- کد ملی - CharField (حداکثر ۱۰ کاراکتر باشد) \*
- شماره موبایل - CharField (حداکثر ۱۱ کاراکتر باشد) \*
- معرفی (bio) - TextField (این فیلد را بصورت اختیاری تعریف کنید. یعنی کاربر میتواند خالی بگذارد)
- جنسیت - CharField (به گونه‌ای تعریف کنید که قابلیت انتخاب بین ۳ گزینه خانم، آقا و سایر را داشته باشد) \*
- آیا نویسنده است؟ - BooleanField ( به صورت پیش فرض عبارت False قرار داده شود) \*
- فایل رزومه - FileField (جهت آپلود رزومه کاربر در سایت) \*

در ادامه از شما می‌خواهیم که یک مدل Post برای پست‌های سایت ایجاد کنید و آنرا طوری به مدل Profile وصل کنید که این امکان فراهم باشد که هر پروفایل بتواند هر چندتا پست داشته باشد و هر پست بتواند متعلق به یک یا چند پروفایل باشد.

در این مدل نیاز هست فیلد های زیر تعریف شوند (فیلدهای ستاره دار نباید خالی بمانند و حتما باید پر شوند):

- نویسنده - ??? (اگر پروفایل حذف شد، این فیلد خالی شود و پست حذف نشود - کلید ارتباط) \*
- زمان ایجاد - DateTimeField (پس از ایجاد پست، باید این فیلد خودکار پر شود به روز و ساعت آن لحظه)
- زمان آخرین ویرایش - DateTimeField (پس از هر تغییر، باید این فیلد خودکار پر شود به روز و ساعت آن لحظه)
- عنوان - CharField (حداکثر ۵۰ کاراکتر باشد) \*
- عکس پست - ImageField (جهت آپلود عکس پست در سایت) \*
- توضیحات خلاصه - CharField (حداکثر ۳۰۰ کاراکتر باشد) \*
- توضیحات کامل - TextField \*
- وضعیت - PositiveSmallIntegerField \*: (به گونه‌ای تعریف کنید که قابلیت انتخاب بین ۳ گزینه پیش‌نویس، در حال انتشار و عدم تایید را داشته باشد)
- دسته‌بندی - PositiveSmallIntegerField \*: (به گونه‌ای تعریف کنید که قابلیت انتخاب بین ۳ گزینه علمی، آموزشی و خبری را داشته باشد)

در نهایت از شما می‌خواهیم که یک مدل Comment برای دیدگاه پست‌های سایت ایجاد کنید و آنرا طوری به مدل Post وصل کنید که این امکان فراهم باشد که هر پست بتواند هر چندتا دیدگاه داشته باشد ولی هر دیدگاه فقط میتواند به یک پست اختصاص داده شود.

همچنین یک ارتباط نیز با پروفایل برقرار کنید همانند ارتباطی که با Post برقرار کردید که مشخص شود این دیدگاه را چه کاربری گذاشته است.

در این مدل نیاز هست فیلد های زیر تعریف شوند (فیلدهای ستاره دار نباید خالی بمانند و حتما باید پر شوند):

- پست - ??? (اگر پست حذف شد، همه‌ی دیدگاه‌های آن پست هم حذف شوند - کلید ارتباط) \*
- کاربر - ??? (اگر پروفایل حذف شد، این فیلد خالی شود و دیدگاه حذف نشود - کلید ارتباط) \*
- زمان ایجاد - DateTimeField (پس از ایجاد دیدگاه، باید این فیلد خودکار پر شود به روز و ساعت آن لحظه)
- متن دیدگاه - TextField \*
- وضعیت - CharFiled \*: (به گونه‌ای تعریف کنید که قابلیت انتخاب بین ۳ گزینه در حال بررسی، مورد قبول و عدم تایید را داشته باشد)

## دستورات پر کاربرد Django:

```
django-admin startproject my_project
```

✱ ایجاد پروژه جدید(همراه با ایجاد یک فولدر):

عبارت my\_project نام دلخواهی است که شما برای پروژه انتخاب میکنید.

این دستور برای شما یک فولدر نیز ایجاد میکند و داخل آن فولدر فایل‌های پروژه را قرار میدهد. اگر بخواهیم مستقیم فایل‌های پروژه داخل همین فولدری که هستیم (دایرکتوری فعلی) ایجاد کنه، انتهای دستور یک نقطه(دات) می‌ذاریم:

```
django-admin startproject config .
```

عبارت config نام پروژه شما همیشه و چون یک فولدر با این نام ایجاد میشه و درونش فایل‌های زیرساختی پروژه ما از جمله settings.py قرار میگیره، معمولا اسمش رو می‌ذارن config اما انتخاب اسم کاملا دلخواه است.

```
python manage.py startapp app_name
```

✱ ایجاد یک اپلیکیشن جدید:

```
python manage.py runserver
```

✱ ران کردن پروژه:

اگر انتهای دستور هیچی ننویسیم، بصورت پیش‌فرض روی پورت ۸۰۰۰ اجرا میشه. میتونیم پورت خاصی رو بهش اختصاص بدیم

```
python manage.py runserver 8001
```

که در اونجا ران بشه بطور مثال:

- اگر بخواهیم سرور را از متوقف کنیم، کافیه کلیدهای ترکیبی Ctrl + C را بزنید.

✱ ایجاد مایگریشن (پس از ایجاد یا ادیت کردن یک یا چند مدل در models.py این دستور را بزنید):

```
python manage.py makemigrations
```

✱ ذخیره جداول مایگریشن در دیتابیس (قبل از اینکار باید ابتدا دستور بالا زده شده باشد) :

```
python manage.py migrate
```

`python manage.py createsuperuser`

✳ ایجاد یک ابرکاربر (ادمین) جدید:

✳ انتقال فایل‌های استاتیک به روت پروژه (برای استفاده از این دستور لازم است که مقدار `STATIC_ROOT` در

`python manage.py collectstatic`

`settings.py` تعریف شده باشد) :

`STATIC_ROOT = os.path.join(BASE_DIR, "static")`

نحوه تعریف متغیر `STATIC_ROOT` :

`import os`

حواستان باشد قبل از آن `os` را ایمپورت کرده باشید:

`python manage.py shell`

✳ ورود به محیط شل جنگو (جهت ارتباط با دیتابیس) :

<https://www.djangoproject.com>

✳ آدرس سایت رسمی جنگو (معرفی جنگو و داکيومنت):

<https://djangopackages.org>

✳ آدرس سایت رسمی پکیج‌های جنگو:

این تمرین متناسب با مثال‌هایی که در دنیای واقعیت و بازار کار با آن روبه‌رو خواهیم شد، طراحی شده است.

با آرزوی موفقیت و پیشرفت شما عزیزان.

مدرس: عرفان اسماعیلی شیرازی