

۱۵ نمره

مهلت ارسال: تا ۳۰ روز پس از اتمام دوره

فاز ۲ پروژه پایانی

## هدف از این تمرین: تسلط در شخصی سازی پنل ادمین | Django 3

شرح تمرین:

پیشنیاز این تمرین، تمرین شماره ۱ است. شما میبایست تمرین شماره ۱ را انجام داده باشید و کدهای این بخش را در ادامه‌ی پروژه‌ای که در تمرین ۱ ایجاد کردید اضافه کنید. در این تمرین از شما میخواهیم که پنل ادمین جنگو را شخصی سازی کنید.

✓ جهت تغییر ظاهر پنل ادمین جنگو از یکی از پکیج‌هایی که در سایت <https://djangopackages.org> قرار دارد استفاده کنید. مثال:

DJANGO JET, DJANGO-SUIT, DJANGO-ADMIN-BOOTSTRAPPED, DJANGO-ADMIN-INTERFACE

✓ شما میبایست تمام قسمت‌های این پنل را به فارسی ترجمه کنید و هیچ کلمه‌ی انگلیسی و یا تاریخ میلادی دیده نشود.

نکته: پس از اتمام پیاده سازی این بخش، برای هر یک از مدل‌هایی که در فاز ۱ پیاده‌سازی کردید، حداقل ۵ آجکت معنی‌دار در دیتابیس ایجاد کنید.

## شخصی سازی ادمین مدل Profile:

- در صفحه‌ی مدیریت پروفایل‌ها، به‌ترتیب از سمت راست فیلدهای زیر نمایش داده شوند:  
نام کامل کاربر - شماره موبایل - کدملی - جنسیت - آیا نویسنده است؟ - تاریخ عضویت  
نکات: نمونه صحیح نمایش فیلد تاریخ: ۱۶:۰۰ - ۹۹/۱۲/۰۳
- یک باکس جهت فیلتر کردن پروفایل‌ها به‌ترتیب بر اساس فیلدهای زیر ایجاد کنید:  
زمان عضویت - جنسیت - آیا نویسنده است؟
- یک باکس اینپوت جهت جستجو کردن بین پروفایل‌ها بر اساس فیلدهای زیر ایجاد کنید:  
نام کاربری کاربر - نام کوچک کاربر - نام خانوادگی کاربر - شماره موبایل - کدملی
- ترتیب نمایش پروفایل‌ها باید به‌صورت زیر باشد:  
نام خانوادگی کاربر - نام کوچک کاربر - سن  
نکات: نام خانوادگی و نام کوچک کاربر میبایست براساس حروف الفبا مرتب شوند و اگر چند کاربر نام خانوادگی یکسانی داشتند، براساس نام کوچک آنها مرتب کنید و اگر نام کوچک آنها نیز برابر بود براساس سن آنها به این‌صورت که هرکدام که بزرگتر بود، بالاتر نمایش داده شود، مرتب کنید.
- قابلیت sort کردن از طریق کلیک روی فیلدها را فقط برای فیلدهای زیر فعال کنید:  
نام کامل کاربر - تاریخ عضویت
- قابلیت ویرایش فیلد پروفایل‌ها (قبل از این وارد آن پروفایل بشیم) برای فیلدهای زیر فعال کنید:  
شماره موبایل - کدملی - جنسیت - آیا نویسنده است؟

## شخصی سازی ادمین مدل Post:

- در صفحه‌ی مدیریت پست‌ها، به‌ترتیب از سمت راست فیلدهای زیر نمایش داده شوند:  
**عکس پست - عنوان پست - نام کامل نویسنده - وضعیت - دسته‌بندی - زمان آخریش ویرایش - زمان ایجاد**  
نکات: عکس پست میبایست مربعی با سایز ۱۰۰×۱۰۰ باشد - نمونه صحیح نمایش فیلد تاریخ: ۱۶:۰۰ - ۹۹/۱۲/۰۳
- یک باکس جهت فیلتر کردن پست‌ها به‌ترتیب بر اساس فیلدهای زیر ایجاد کنید:  
**وضعیت - دسته‌بندی - نویسنده - تاریخ ایجاد**
- یک باکس اینپوت جهت جستجو کردن بین پست‌ها بر اساس فیلدهای زیر ایجاد کنید:  
**عنوان پست - توضیحات خلاصه - توضیحات کامل**
- ترتیب نمایش پست‌ها باید بر اساس **تاریخ ایجاد** باشد.  
نکات: این مرتب‌سازی را در کلاس Meta در مدل Post بنویسید که در کوئری‌های ما نیز این ترتیب اعمال شود.
- قابلیت ویرایش فیلد پست‌ها (قبل از این وارد آن پست بشیم) برای فیلدهای زیر فعال کنید:  
**عنوان پست - وضعیت - دسته‌بندی**
- یک کلاس inline از مدل Post ایجاد کنید و آن را به ادمین Profile متصل کنید تا بتوانیم با وارد شدن به صفحه هر پروفایل، پست‌های مربوط به آن پروفایل را مشاهده و بتوانیم یک یا چند پست جدید ایجاد کنیم.
- نکات: بصورت پیش‌فرض ۳ آبجکت اضافی خالی جهت ایجاد Post در پروفایل کاربر نمایش داده میشود که میبایست طوری تعریف کنید کلاس inline را که هیچ آبجکت اضافی جهت ایجاد نمایش ندهد. اگر خواستیم خودمان اضافه کنیم.

## شخصی سازی ادمین مدل Comment:

- در صفحه‌ی مدیریت کامنت‌ها، به‌ترتیب از سمت راست فیلدهای زیر نمایش داده شوند:

عنوان پست - نام کامل کاربر (فردی که دیدگاه گذاشته است) - وضعیت - زمان ایجاد

نکات: نمونه صحیح نمایش فیلد تاریخ: ۱۶:۰۰ - ۹۹/۱۲/۰۳

- یک باکس جهت فیلتر کردن کامنت‌ها به‌ترتیب بر اساس فیلدهای زیر ایجاد کنید:

وضعیت - زمان ایجاد

- یک باکس اینپوت جهت جستجو کردن بین کامنت‌ها بر اساس فیلدهای زیر ایجاد کنید:

نام خانوادگی کاربر - نام کوچک کاربر - عنوان پست

- ترتیب نمایش کامنت‌ها باید بر اساس تاریخ ایجاد باشد.

نکات: این مرتب‌سازی را در کلاس Meta در مدل Comment بنویسید که در کوئری‌های ما نیز این ترتیب اعمال شود.

- قابلیت ویرایش فیلد کامنت‌ها (قبل از این وارد آن کامنت بشیم) را فقط برای فیلد وضعیت فعال کنید.

- دسترسی فیلدهای زیر را read only کنید. فقط قابل مشاهده باشند و امکان ویرایش آن فراهم نباشد.

پست - کاربر - متن دیدگاه (موقع ایجاد ۵ آبجکت تستی خواسته شده، این قسمت را پاک کنید و بعد از آن اضافه کنید)

- یک کلاس inline از مدل Comment ایجاد کنید و آن را به ادمین Profile و Post متصل کنید تا بتوانیم با

وارد شدن به صفحه هر پروفایل کامنت‌هایی که آن کاربر گذاشته را ببینیم. همچنین با وارد شدن به صفحه

هر پست، کامنت‌های مربوط به آن پست را ببینیم و بتوانیم یک یا چند کامنت جدید ایجاد کنیم.

نکات: بصورت پیش‌فرض ۳ آبجکت اضافی خالی جهت ایجاد Post در پروفایل کاربر نمایش داده میشود که میبایست

طوری تعریف کنید کلاس inline را که یک آبجکت اضافی جهت ایجاد نشان دهد. اگر خواستیم خودمان اضافه‌تر کنیم.

## 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 کاملاً 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 صنعت بازار کار با آن روبه‌رو خواهیم شد، طراحی شده است.

با آرزوی موفقیت و پیشرفت شما عزیزان.

مدرس: عرفان اسماعیلی شیرازی