

استارتاپ پرتقالی 🍊

- محدودیت حجم فایل: ۱ مگابایت

حسن یک ایده‌ی استارت‌آپی ناب به ذهنش رسیده. او می‌خواهد تمام پرتقال فروش‌های محل را در یک سامانه ساماندهی کند. او پس از تحقیقات فراوان تصمیم می‌گیرد که برای این کار یک اپ جنگو بنویسد ولی از آنجا که حسن خیلی به جنگو مسلط نیست، از شما درخواست کرده تا به او کمک کنید.

پروژه اولیه

پروژه اولیه را از این [لینک](#) دانلود کنید. ساختار فایل‌های این پروژه به صورت زیر است:

```
orange_seller
├── users
│   ├── apps.py
│   ├── > queryset.py <
│   ├── migrations
│   │   └── __init__.py
│   ├── > models.py <
│   ├── tests.py
│   ├── admin.py
│   └── > views.py <
├── config
│   ├── asgi.py
│   ├── __init__.py
│   ├── > settings.py <
│   ├── > custom_validator.py <
│   ├── urls.py
│   └── wsgi.py
├── manage.py
└── requirements.txt
```

تنظیمات

- موقعیت زمانی باید به وقت ایران باشد.
- فایل‌های استاتیک باید در مسیر /orange_static/ ذخیره شوند.

- تنها میزبان مورد قبول را localhost بگذارید.
- سه validation برای پسورد نیاز است:
 ۱. با ویژگی‌های دیگر کاربر شباهت نداشته باشد. (به طور مثال پسورد برابر با اسم کاربر نباشد).
 ۲. از پسوردهای رایج نباشد. (به طور مثال qwe نباشد).
 ۳. پسورد شما باید تست رجکسی که در تمرین اول سوال سرزمین گوگل نوشتید را پاس کند. برای این کار باید در فایل custom_validator.py یک validator پیاده‌سازی کنید.
- از دو الگوریتم رمزنگاری زیر برای رمز کردن پسورد استفاده کنید:
 ۱. از الگوریتم Argon2 استفاده کنید.
 ۲. از الگوریتم sha1 استفاده کنید.
 ۳. متن خطای ولیدیتور : password is not valid می‌باشد
- به ترتیب پسورد ولیدتورها و رمزنگارهای پسورد دقت کنید.

نکات مهم

- شما تنها مجوز ایجاد تغییرات در فایل‌های زیر را دارید و تمامی تغییرات دیگر شما در فایل‌های پروژه نادیده گرفته خواهند شد.

- config/settings.py
- custom_validator.py

پرتقال دان زادن 🍊

- محدودیت حجم فایل: ۱ مگابایت

حالا که تنظیمات را برای حسن ست کردید. او می‌خواهد تا مدل‌ها را نیز بر اساس خواسته‌ی او پیاده‌سازی کنید. از آنجایی که حسن در ایران زندگی می‌کند شما باید از تاریخ جلالی استفاده کنید. برای اینکار از کتابخانه `django-jalali` کمک بگیرید.

مدل فروشنده Seller

این مدل تمام ویژگی‌های مدل `user` جنگو را دارد. (بدون هیچ رابطه‌ای با این مدل) و نیز ویژگی‌های زیر را داراست:

فیلدهای اضافه‌ای که باید به این مدل اضافه شود.

`birth_date` -> تاریخ تولد شمسی
`national_code` -> کد ملی

متدهای داخلی مدل:

`age` -> سن فروشنده
`is_birthday` -> آیا امروز تولد فروشنده هست؟

توجه: سن فروشنده از تفاضل سال فعلی از سال تولد فروشنده محاسبه می‌شود و نیازی به در نظر گرفتن ماه و روز نیست.

توجه ۲: برای پیاده سازی متدها از `@property` استفاده کنید.

مدل محصولات Product

در این مدل شما باید مشخصات پرتقال‌های فروشندگان را مشخص کنید. فیلدهای مورد نظر برای این مدل به شرح زیر هست؛

weight -> وزن پرتقال
 price -> قیمت آن وزن پرتقال
 seller -> فروشنده پرتقال‌ها

مدل سفارشات Order

در این مدل حسن میخواهد تمامی سفارشات را که هر فروشنده دارند را ثبت کند. فیلدهای مورد نظر برای این مدل به شرح زیر هست:

product -> محصول یا همون پرتقال
 count -> تعداد خرید
 due_date -> زمان تحویل
 created_at -> زمان ثبت سفارش - این فیلد باید به شکل خودکار پر شود.

پروژه اولیه

پروژه اولیه را از این [لینک](#) دانلود کنید. ساختار فایل‌های این پروژه به صورت زیر است:

```
orange_seller
├── users
│   ├── apps.py
│   ├── > queryset.py <
│   ├── migrations
│   │   └── __init__.py
│   ├── > models.py <
│   ├── tests.py
│   ├── admin.py
│   └── > views.py <
├── config
│   ├── asgi.py
│   ├── __init__.py
│   ├── > settings.py <
│   ├── > custom_validator.py <
│   ├── urls.py
│   └── wsgi.py
├── manage.py
└── requirements.txt
```

نکات مهم

- شما تنها مجوز ایجاد تغییرات در فایل زیر را دارید و تمامی تغییرات دیگر شما در فایل‌های پروژه نادیده گرفته خواهند شد.

- `users/models.py`

پیدا کنید پرتقال فروش را 🍊

- محدودیت حجم فایل: ۱ مگابایت

مدل‌ها طبق خواسته‌ی او پیاده‌سازی شد. حالا او از شما خواسته که کوئری‌های زیر را برای او بنویسید و خواسته‌های او را برآورده کنید.

پیاده‌سازی کوئری‌ست‌ها

- تابع `get_sellers_with_more_than_one_product` از شما می‌خواهد تا فروشندگانی که بیشتر از یک محصول را در سیستم ثبت کرده‌اند را برگردانید.
- تابع `get_mean_weight_of_seller_products` با ورودی گرفتن یک فروشنده، می‌خواهد میانگین وزن محصولات او را برگردانید.
- تابع `get_sellers_with_more_than_two_products_or_age_greater_than_30` کوئری را می‌خواهد که فروشندگانی را مشخص کند که بیشتر از دو محصول در سیستم ثبت کرده‌اند یا سن آن‌ها بیشتر از ۳۰ سال است.
- تابع `get_sellers_with_good_price` از شما می‌خواهد تا فروشندگانی خوش حساب را پیدا کنید. یک فروشنده وقتی خوش حساب می‌شود که حداقل یک محصول یا یک درخواست خرید در ۳۰ روز گذشته را تحویل داده باشد و نسبت قیمت به وزن محصول آن کمتر از ۱۰۰۰۰ باشد.
- تابع `get_seller_with_max_price` فروشنده‌ای با گران‌ترین محصول از میان بقیه‌ی فروشندگانش را برمی‌گرداند.
- تابع `sum_of_orders_in_30_days` مجموع تعداد سفارشات که یک فروشنده در ۳۰ روز اخیر تحویل داده را محاسبه می‌کند.

پروژه اولیه

پروژه اولیه را از این لینک دانلود کنید. ساختار فایل‌های این پروژه به صورت زیر است:

```
orange_seller
├── users
└── apps.py
```

```
|   ├── > queryset.py <
|   ├── migrations
|   |   └── __init__.py
|   ├── > models.py <
|   ├── tests.py
|   ├── admin.py
|   └── > views.py <
└── config
    ├── asgi.py
    ├── __init__.py
    ├── > settings.py <
    ├── > custom_validator.py <
    ├── urls.py
    └── wsgi.py
└── manage.py
└── requirements.txt
```

نکات مهم

- شما تنها مجوز ایجاد تغییرات در فایل زیر را دارید و تمامی تغییرات دیگر شما در فایل‌های پروژه نادیده گرفته خواهند شد.

- users/queryset.py

🍊 پرتقال برخط

- محدودیت حجم: ۱ مگابایت

حالا که کوئری‌های پروژه نیز نوشته شده‌اند، حسن می‌خواهد اولین نسخه از برنامه فوق‌العاده خود را در دسترس عموم قرار دهد تا به زودی کنترل بازار پرتقال را در دست بگیرد اما هنوز برنامه او تکمیل نشده است؛ شما باید View های برنامه را نیز برای حسن بنویسید تا کاربران بتوانند از امکانات برنامه او استفاده کنند.

پیاده‌سازی ویوها

ویو صفحه اصلی home_page

در صفحه اصلی لازم است تعداد فروشندگانی که بیشتر از یک محصول در سیستم ثبت کرده‌اند را به کاربر نمایش دهید تا کاربر با دیدن آمار فوق خفن برنامه به استفاده از آن ترغیب شود. برای این منظور از کوئری `get_sellers_with_more_than_one_product` استفاده کنید.

ویو فروشندگان خوش حساب good_price_sellers

در این صفحه لیستی از فروشندگان خوش حساب را برای کاربر نمایش دهید تا کاربر به برنامه اعتماد کند و اقدام به خرید پرتقال از این فروشندگان کند. برای این منظور از کوئری `get_sellers_with_good_price` استفاده کنید.

ویو فروشندگان گران قیمت expensive_sellers

از آنجایی که بعضی کاربران قیمت بالاتر را دلیل بر کیفیت بهتر می‌دانند، در این صفحه برای آنها فروشنده‌های با گران‌قیمت‌ترین کالاها را برگردانید تا نیاز این دسته از کاربران نیز مرتفع شود. برای این منظور از کوئری `get_seller_with_max_price` استفاده کنید.

ویو مجموع تعداد سفارشات یک فروشنده seller_orders_count

حسن بسیار به شفافیت در کسب و کار معتقد است و از این جهت، می‌خواهد آمار فروش هر یک از فروشنده‌ها را در اختیار تمام کاربران سیستم قرار دهد. در این صفحه لازم است آمار تعداد سفارشات قابل تحویل در ۳۰ روز گذشته را برای یک فروشنده خاص برگردانید. برای این منظور از کوئری `sum_of_orders_in_30_days` استفاده کنید. آیدی فروشنده موردنظر در `url` صفحه به شما داده می‌شود. در نظر داشته باشید که اگر هیچ سفارش قابل تحویلی در ۳۰ روز گذشته وجود نداشت باید به فارسی **هیچ** نوشته و برگردانده شود.

پروژه اولیه

پروژه اولیه را از این [لینک](#) دانلود کنید. ساختار فایل‌های این پروژه به صورت زیر است:

```
orange_seller
├── users
│   ├── apps.py
│   ├── > queryset.py <
│   ├── migrations
│   │   └── __init__.py
│   ├── > models.py <
│   ├── tests.py
│   ├── admin.py
│   └── > views.py <
├── config
│   ├── asgi.py
│   ├── __init__.py
│   ├── > settings.py <
│   ├── > custom_validator.py <
│   ├── urls.py
│   └── wsgi.py
├── manage.py
└── requirements.txt
```

نکات مهم

- شما تنها مجوز ایجاد تغییرات در فایل‌های زیر را دارید و تمامی تغییرات دیگر شما در فایل‌های پروژه نادیده گرفته خواهند شد.

- `users/views.py`

