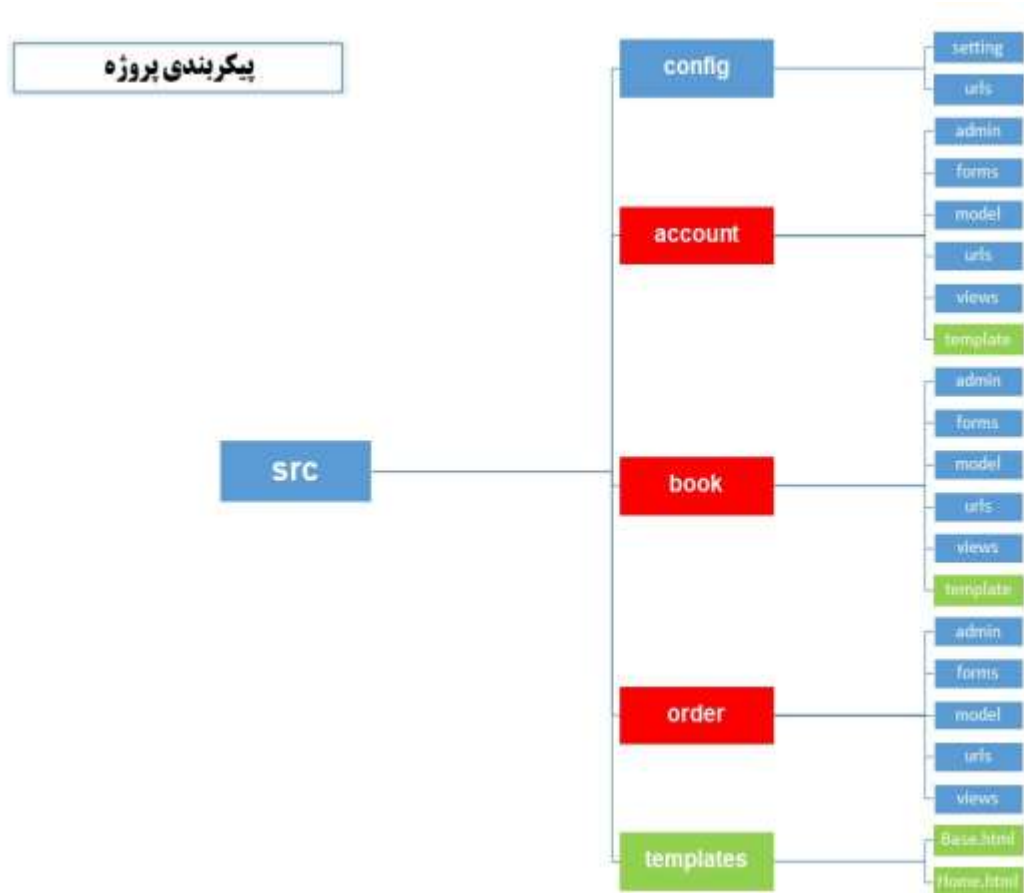


بنام خدا

پروژه فروشگاه آنلاین کتاب – ۱۴۰۰/۰۵/۲۰



: config

تنظیمات اصلی پروژه شامل فایل های **setting** و **urls**

اپ های ایجاد شده در پروژه:

: accounts

برای ایجاد مدل های مربوط به مشتریان و کارمندان و مدیریت کاربران

در این اپ مدل **Staff** (شامل کارمندان و مدیر) از مدل **AbstractUser** ارث بری کرده است.

:book

در این اپ مدلی برای کتاب ها و مدلی برای دسته بندی موضوعات مختلف ایجاد شده است که رابطه این دو جدول بصورت many-to-many می باشد.

۱ – مدل **Book** شامل فیلدهای زیر می باشد:

title: عنوان کتاب

Author: نویسنده کتاب

Category: دسته ای که این کتاب به آن تعلق دارد

Price: قیمت تاب

Quantity: موجودی فروشگاه از این کتاب

Created_by: کاربری که این کتاب را به پایگاه داده اضافه کرده است. که در اینجا از `get_user_model()` جهت ارجاع به مدل کاربران و شناسایی کاربر استفاده شده است.

Created_time: زمان ایجاد این رکورد در جدول

Update_time: زمان بروزرسانی این رکورد از جدول

۲ – مدل **Category** شامل فیلدهای زیر می باشد:

Name: نام دسته

Created_by: کاربری که این دسته را ایجاد کرده است

Created_time: زمان ایجاد این رکورد در جدول

Update_time: زمان بروزرسانی این رکورد از جدول

:order

در این اپ که برای مدیریت بخش مربوط به سفارشات و سبد خرید می باشد، مدل های زیر ایجاد شده است.

۱ – مدل **PercentageOff** بعنوان جدولی شامل کتابهایی که دارای تخفیف درصدی هستند و فیلدهای زیر را دارا می باشد:

book: کتاب

category: کدام گروه از کتاب ها

value: مقداری تخفیف اعمال شده. چون این تخفیف باید بصورت درصد بر روی قیمت کتاب اعمال شود، از کاربر یک عدد بین ۰ تا ۱۰۰ دریافت می شود. برای کنترل این بازه عدد از گزینه **validators** که دارای دو مقدار کمیه و بیشینه می باشد استفاده شده است. در هنگام اعمال این مقدار باید آن را تقسین بر ۱۰۰ کنیم.

`validators=[MinValueValidator(0), MaxValueValidator(100)]`

created_by: این تخفیف توسط چه کسی ایجاد شده است

created_time: زمان ایجاد تخفیف

expired_time: زمان انقضای تخفیف

۲ – مدل **CashOff** برای نگهداری کتابهایی که دارای تخفیف نقدی هستند و شامل فیلدهای زیر میباشد:

book: کتاب

category: کدام گروه از کتاب ها

value: مقدار تخفیف اعمال شده

created_by: این تخفیف توسط چه کسی ایجاد شده است

created_time: زمان ایجاد

expired_time: زمان انقضای تخفیف

۳ – مدل **DiscountCode** که یک جدول جهت نگهداری کدهای تخفیف و مشخصات مربوط به آن ها می باشد. این مدل تخفیف تنها روی سبد خرید مشتری اعمال می شود. این جدول شامل فیلدهای زیر است:

discount_code: کد تخفیف

value: درصد تخفیف که بصورت یک عدد بین ۰ تا ۱۰۰ دریافت می شود

created_by: این تخفیف توسط چه کسی ایجاد شده است

valid_from: از چه زمانی این کد تخفیف فال است

valid_to: زمان انقضای این تخفیف

active: آیا این کد فعال است یا خیر

این کلاس شامل متود **create_code()** می باشد که یک کد شش رقمی شامل اعداد و حروف بصورت رندوم تولید می کند.

```
def create_code(self):
    """
    این متود یک کد 6 کاراکتری شامل اعداد و حروف بصورت رندوم بعنوان مد تخفیف تولید می کند
    """
    self.discount_code = ''.join(random.choice(string.ascii_uppercase + string.digits) for _ in range(5))
    return self.discount_code
```

۴ – مدل **ShoppingBasket** که سبد خرید مشتری می باشد، به این صورت که به هر مشتری یک سبد خرید تعلق دارد که می تواند در هر بار بازدید از فروشگاه اقلام مورد نظر خود را جهت سفارش در این سبد خرید قرار دهد. این سبد موارد زیر را شامل می شود:

customer: مشتری

order_date: تاریخ سفارش

discount: میزان تخفیف تعلق گرفته به این سبد خرید

متود `get_total_price()` برای محاسبه قیمت کل سبد خرید بعد از اعمال کد تخفیف ایجاد شده است.

```
def get_total_price(self):  
    """  
    این متود براساس میزان تخفیف این سبد قیمت نهایی آن را محاسبه می کند  
    :return:  
    """  
    total_price = sum(item.get_cost() for item in self.items.all())  
    if self.discount:  
        discount_price = (self.discount / 100) * total_price  
        return int(total_price - discount_price)  
    return total_price
```

۵ - مدل **OrderItems** که شامل جزئیات هر قلم از سفارشات موجود در سبد خرید را شامل می شود و فیلدهای زیر را دارد:

shop_basket: کدام سبد خرید

book: کتاب سفارش داده شده

book_price: قیمت فروش کتاب

book_quantity: تعداد خریداری شده

status: وضعیت سفارش پرداخت شده یا سفارش داده شده

همچنین این کلاس دارای **property** بنام `sell_price` است که در آن قیمت فروش کتاب پس اعمال تخفیف های نقدی و درصدی در صورت وجود محاسبه می گردد و در متود `get_cost()` که قیمت تمام شده این ردیف از سبد خرید را با توجه به تعداد خریداری شده است محاسبه می کند، استفاده می گردد.

```
@property  
def sell_price(self):  
    """  
    قیمت نهایی کتاب پس از اعمال تخفیف ها  
    :return:  
    """  
    final_price = (book.price - self.cash_off) - ((self.percent_off/100) * book.price)  
    return final_price
```

```
def get_cost(self):  
    """  
    :return: قیمت نهایی این ردیف از سبد کالا را با توجه به قیمت اصلی کتاب و تعداد خریداری شده محاسبه می کند  
    """  
    return self.sell_price * self.book_quantity
```