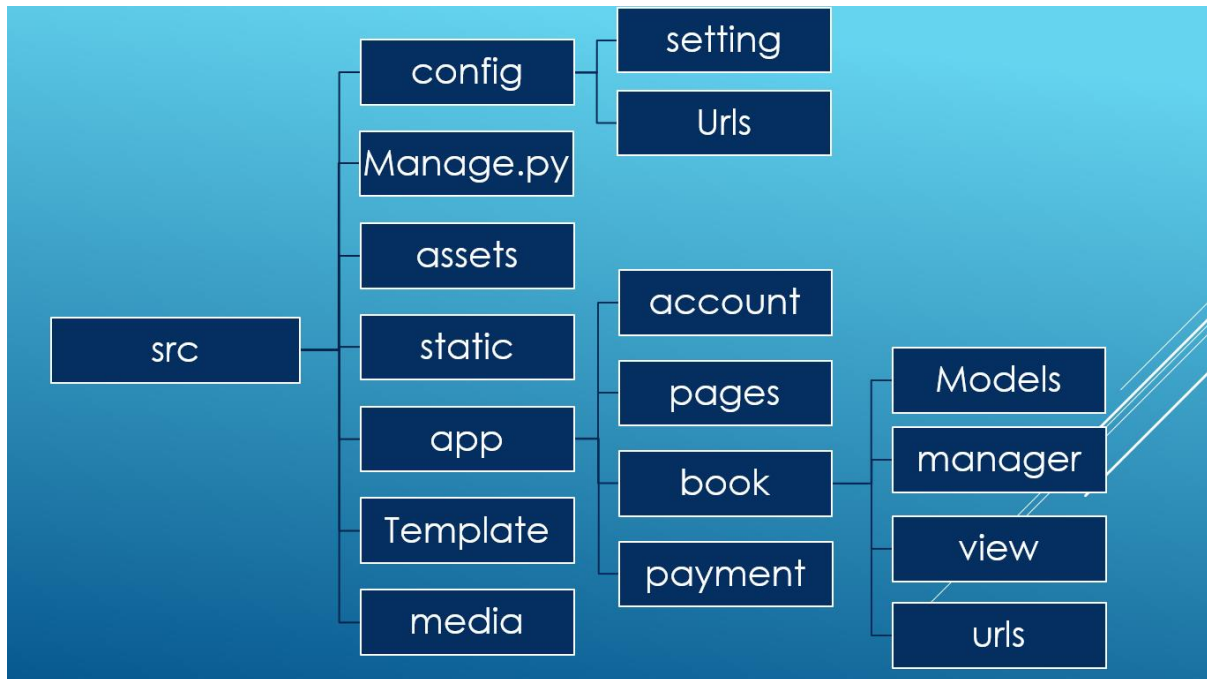


نمودار پیکربندی پروژه:



توضیحات اپ ها و مدل ها :

: pages app

-این اپ برای صفحات استاتیک ساخته شده است و تمپلیت base و home در آن قرار داده شده است.

: accounts app

-این اپ برای مدیریت کاربران (ثبت اطلاعات، ورود، خروج، تغییر پسورد و ..) ساخته شده است. مدهای CustomUser، Customer، AddressModel در آن قرار گرفته است.

-کلاس CustomUser برای ثبت کاربران، طراحی شده است. این کلاس از کلاس AbstractUser جنگو ارث بری میکند. برای هر کاربر، نام کاربری، ایمیل، پسورد، تاریخ ساخت حساب ثبت می شود.

```
class CustomUser(AbstractUser):
```

```
    REQUIRED_FIELDS = [('email'), ]
```

```
    USERNAME_FIELDS = ('email',)
```

```

username = models.CharField('کاربری نام', max_length=50, unique=True, null=True,
blank=True)
email = models.EmailField('ایمیل', blank=False, unique=True)
password = models.CharField('عبور رمز', max_length=50)
register_date = models.DateTimeField(verbose_name='حساب ساخت تاریخ',
auto_now_add=True, blank=True)

```

-برای اطلاعات مشتریان خود، از مدل Customer استفاده میکنیم. هر مشتری ارتباط یک به یک با جدول کاربرها دارد. از هر مشتری نام، نام خانوادگی، جنسیت را ذخیره میکنیم. جنسیت به صورت Choise می باشد که دو انتخاب دارد.

```

class Customer(models.Model):
    Female = 'F'
    Male = 'M'
    first_name = models.CharField(max_length=50, blank=True, null=True)
    last_name = models.CharField(max_length=50, blank=True, null=True)
    GENDER_CHOICES = [(Female, 'زن'), (Male, 'مرد')]
    user = models.OneToOneField(CustomUser, on_delete=models.CASCADE,
primary_key=True)
    gender = models.CharField('جنسیت', choices=GENDER_CHOICES, max_length=1,
blank=True, null=True)

```

-کلاس Address برای ثبت آدرس کاربران می باشد. یک فیلد فارن کی برای ارجاع به مشتری، یک فیلد برای ثبت شهر، یک فیلد برای کد پستی و یک تکست فیلد برای ثبت آدرس دارد.

```

class AddressModel(models.Model):

    customer = models.ForeignKey(Customer, blank=False, on_delete=models.CASCADE,
verbose_name='کاربر')
    city = models.CharField(max_length=50, blank=True, null=True)
    address = models.TextField('آدرس', max_length=300)
    Postal_code = models.IntegerField(max_length=10)

```

: book app

این اپ برای محصول (کتاب) طراحی گردیده و مدل های BookModel, CategoryModel, Author, Store در آن قرار دارد.

- کلاس Category برای ثبت دسته بندی ها به کار می رود و یک فیلد برای نام دسته بندی دارد.

```
class CategoryModel(models.Model):
```

```
    # name of Category
    name = models.CharField('نام', max_length=50)
```

- کلاس Author برای ثبت اطلاعات نویسنده کتاب تعریف شده است و یک فیلد برای ثبت "نام" دارد.

```
class Author(models.Model):
```

```
    name = models.CharField('نام', max_length=150, blank=False)
```

- کلاس Book برای ثبت اطلاعات کتاب تعریف شده است. از هر کتاب اطلاعات نام، نویسنده، قیمت، قیمت با تخفیف، تاریخ ثبت کتاب، دسته بندی و همچنین یک فیلد image برای ثبت تصویر کتاب وجود دارد.

```
class BookModel(models.Model):
```

```
    name = models.CharField('نام', max_length=100)
    author = models.ForeignKey(Author, on_delete=models.DO_NOTHING,
        verbose_name='نویسنده')
    price = models.DecimalField('قیمت', max_digits=18, decimal_places=2)
    price_discount = models.DecimalField('تخفیف با قیمت', max_digits=18, decimal_places=2,
        blank=True, null=True)
    created = models.DateField('کتاب ثبت تاریخ', auto_now_add=True)
    categories = models.ManyToManyField(CategoryModel, blank=False,
        verbose_name='گروه')
    image = models.ImageField(upload_to='book/', null=True, default='0.png')
```

- کلاس Store برای ثبت تعداد محصولات تعریف شده است. یک فیلد فارن کی برای ارجاع به Book و یک فیلد برای تعداد اجناس دارد.

```
class Store(models.Model):
```

```
    product = models.ForeignKey(BookModel, on_delete=models.CASCADE,  
related_name='product', verbose_name='محصولات')
```

```
    Inventory = models.IntegerField('موجودی')
```

:payment app

این اپ برای پرداخت و محاسبات آن طراحی گردیده و مدل های DiscountCodeValue, DiscountCodePercent, DiscountPercent, DiscountValue, Invoice, InvoiceLine در آن قرار دارد.

-کلاس DiscountCodeValue برای کد تخفیف نقدی است. زمان شروع و پایان برای طول مدت اعتبار کد دارد. کد و مقدار تخفیف از فیلدهای دیگر آن است.

```
class DiscountCodeValue(models.Model):
```

```
    started = models.DateTimeField('شروع زمان', )
```

```
    finished = models.DateTimeField('پایان زمان', )
```

```
    code = models.CharField('تخفیف کد', blank=False, max_length=10)
```

```
    amount = models.FloatField()
```

-کلاس DiscountCodePercent برای کد تخفیف درصدی است. زمان شروع و پایان برای طول مدت اعتبار کد دارد. کد و مقدار درصد تخفیف از فیلدهای دیگر آن است.

```
class DiscountCodePercent(models.Model):
```

```
    started = models.DateTimeField('شروع زمان', )
```

```
    finished = models.DateTimeField('پایان زمان', )
```

```
    code = models.CharField('تخفیف کد', blank=False, max_length=10)
```

```
    percent = models.FloatField()
```

-کلاس DiscountValue برای کد تخفیف نقدی است که مختص به یک کتاب می باشد و با استفاده از کلید فارن کی به کتاب مربوطه ارجاع داده شده است. مقدار تخفیف فیلد دیگر آن می باشد.

```
class DiscountValue(models.Model):
    book = models.ForeignKey('book.BookModel', on_delete=models.CASCADE,
related_name='dis_value',
verbose_name='کتاب نام')

    value = models.DecimalField('مقدار تخفیف', max_digits=8, decimal_places=2)
```

-کلاس DiscountPercent تخفیف درصدی برای یک کتاب می باشد. کتاب مربوطه با استفاده از کلید فارن کی مشخص شده است. فیلد دیگر آن مقدار درصد تخفیف است.

```
class DiscountPercent(models.Model):
    book = models.ForeignKey('book.BookModel', on_delete=models.CASCADE,
related_name='dis_percent',
verbose_name='محصول نام')

    percent = models.IntegerField('تخفیف درصد', )
```

-کلاس Invoice، برای ثبت فاکتورها می باشد. هر فاکتور مختص به یک مشتری می باشد. سفارش در یک تاریخ ثبت میشود و می تواند در تاریخ دیگری تکمیل شود که تاریخ هر دو ثبت می شود. یک قیمت کل دارد و یک قیمت که روی آن تخفیف اعمال شده است. فیلد دیگر، وضعیت سفارش را ثبت میکند که میتواند دو حالت سفارش و یا تکمیل شده داشته باشد. دو فیلد هم دارد که به جدولهای کد تخفیف درصدی و نقدی ارجاع داده شده است.

```
class Invoice(models.Model):
    ORDER = 'O'
    COMPLETE = 'C'
    INVOICE_CHOICES = [(ORDER, 'سفارش'), (COMPLETE, 'ارسال آماده')]

    customer = models.ForeignKey(CustomUser, on_delete=models.DO_NOTHING,
related_name='invoices', verbose_name='کاربر')

    invoice_date = models.DateTimeField('سفارش تاریخ', auto_now_add=True)

    invoice_complete_date = models.DateTimeField('سفارش تکمیل تاریخ', )

    total = models.BigIntegerField('مجموع', )

    total_with_discount = models.BigIntegerField('پرداخت قابل', blank=True, null=True)

    status = models.CharField(choices=INVOICE_CHOICES, max_length=1,
default=ORDER, verbose_name='وضعیت سفارش')

    discount_value = models.ForeignKey(DiscountCodeValue, blank=True, null=True,
on_delete=models.DO_NOTHING,
related_name='value_code_discount')
```

```
discount_percent = models.ForeignKey(DiscountCodePercent, blank=True, null=True,  
on_delete=models.DO_NOTHING,  
related_name='percent_code_discount')
```

-برای ثبت اقلام فاکتورها از جدول InvoiceDetail استفاده میکنیم. یک فیلد که به جدول فاکتور ارجاع داده شده است و فیلد دیگر به کتاب ها (اقلام خریداری شده) ارجاع دارد. یک فیلد که قیمت محصول و تعداد خریداری شده را نمایش می دهد.

```
class InvoiceLine(models.Model):
```

```
    invoice = models.ForeignKey(Invoice, on_delete=models.PROTECT,  
verbose_name='فاکتور شماره')
```

```
    book = models.ForeignKey('book.BookModel', on_delete=models.DO_NOTHING,  
verbose_name='محصول')
```

```
    unit_price = models.IntegerField('واحد قیمت', )
```

```
    quantity = models.IntegerField('تعداد', )
```