

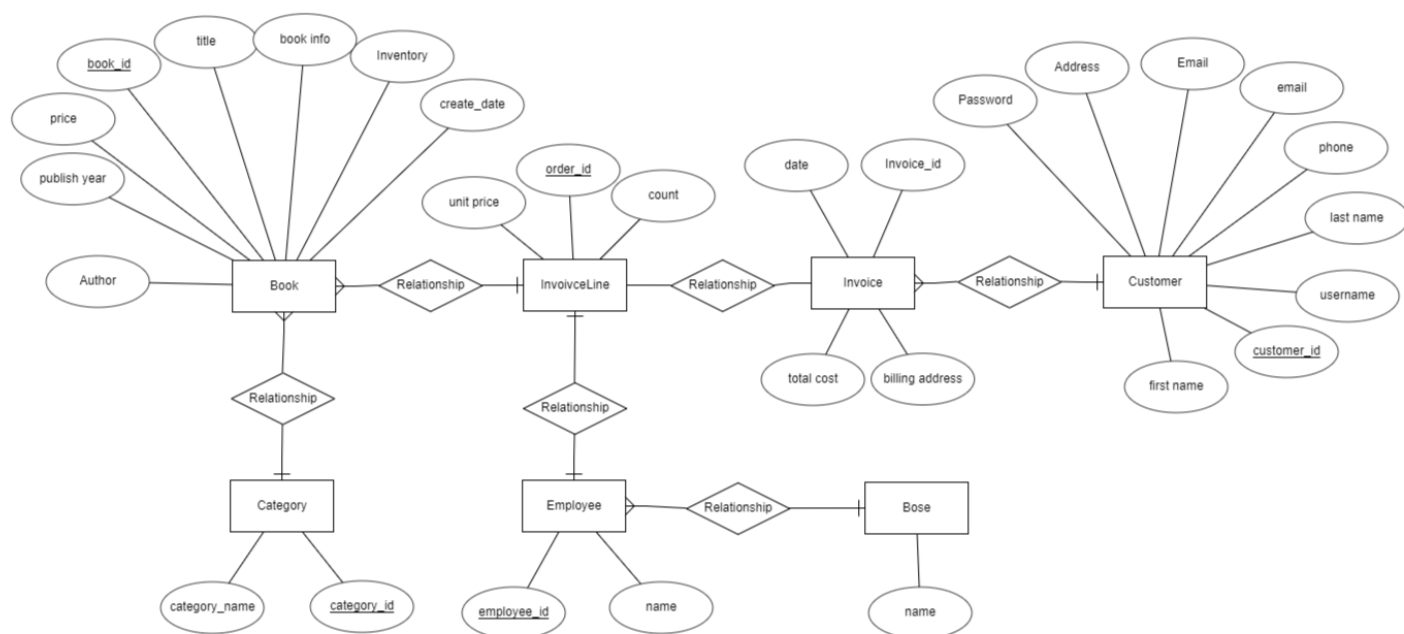
بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

فاطمه خورشیدی

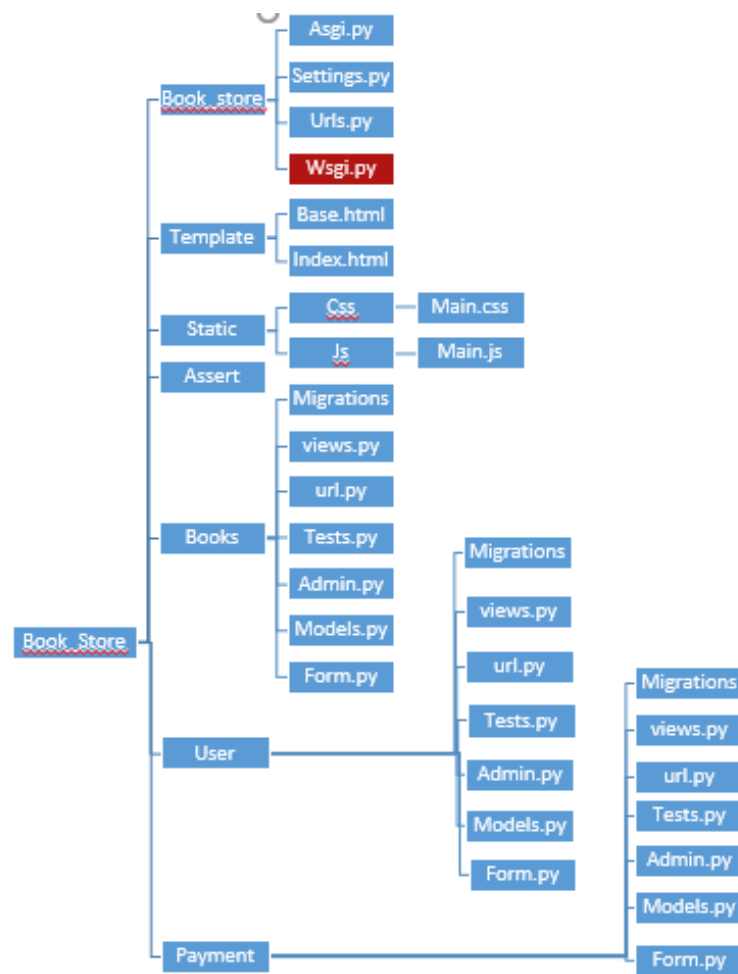
گزارش مرحله اول پروژه پایانی

مکتب 51

در مرحله اول پروژه فروشگاه کتاب شامل ERD پیکر بندی پروژه و توضیح مختصری از app ها می باشد.



شکل 1: مدل ERD پروژه



شکل 2: پیکر بندی پروژه

```

class Invoice(models.Model):
    customer = models.ForeignKey('Books', on_delete=models.DO_NOTHING, related_name='invoices')
    invoice_date = models.DateTimeField()
    billing_address = models.ForeignKey('Address', on_delete=models.CASCADE)
    total = models.BigIntegerField()

    def order(self):
        pass
        # if state = 'سفارش'
        # else:
        #     state = 'نید'

```

شکل 3: مدل 1

مدل شماره 1 صورت حساب مشتری می باشد که دارای فیلد های تاریخ و قیمت نهایی و کلید های خارجی مشتری و آدرس می باشد.

```

class InvoiceLine(models.Model):
    invoice = models.ForeignKey(Invoice, on_delete=models.PROTECT)
    book = models.ForeignKey('Books', on_delete=models.DO_NOTHING)
    unit_price = models.IntegerField()
    count = models.IntegerField()

```

شکل 4: مدل 2

مدل شماره 2 دارای attribute های قیمت واحد و تعداد کتاب های خریداری شده و همچنین کلید های خارجی صورت حساب و کتاب می باشد.

```

# مدل تخفیف
class Discount(models.Model):
    name = models.CharField(max_length=100)
    code = models.CharField(max_length=30,
                            blank=True, null=False,
                            )
    is_active = models.BooleanField(default=True)
    valid_from = models.DateTimeField(default=datetime.now)
    valid_until = models.DateTimeField(blank=True, null=True)

    num_uses = models.IntegerField(_('Number of times already used'),
                                   default=0)

    objects = DiscountBaseManager()

    def __unicode__(self):
        return self.get_name()

    def get_name(self):
        return self.name

```

شکل 5: مدل شماره 3

مدل شماره 3 دارای فیلدهای تاریخ و نام و سایر attribute ها است که به صورت اولیه تعریف شده اند تا در مراحل بعدی گسترش یابند.

```

# مدل سبد خرید
class Cart(models.Model):
    customer = models.OneToOneField('Customer', null=True, on_delete=models.CASCADE)
    books = models.ManyToManyField('Book')

```

شکل 6: مدل شماره 4

مدل شماره 4 سبد خرید است که فیلدهای مشتری و کتاب را دارد که بصورت کلید خارجی تعریف شده است.

```
class Book(models.Model):
    title = models.CharField(max_length=200)
    author = models.CharField(max_length=40)
    category = models.ForeignKey('Category', on_delete=models.DO_NOTHING)
    price = models.DecimalField(max_digits=5, decimal_places=2)
    publish_year = models.DateTimeField('publish_year', null=True)
    image = models.ImageField(upload_to='books/', blank=True)
    book_info = models.TextField('book information', null=True, blank=True)
    inventory = models.IntegerField()
    # create at...
    created = models.DateTimeField(auto_now_add=True)
    # update at...
    updated = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f'{self.title} '

class Category(models.Model):
    name = models.CharField(max_length=200)
```

شکل 7: مدل شماره 5 و 6

در تصویر بالا مدل های کتاب و دسته بندی کتاب قرار دارد که فیلدهای آن تعریف شده است.

```

class Customer(models.Model):
    first_name = models.CharField(max_length=40, blank=True, null=True)
    last_name = models.CharField(max_length=20, blank=True, null=True)
    username = models.CharField(max_length=80)
    password = models.CharField(max_length=10)
    address = models.ForeignKey('Address', on_delete=models.CASCADE)
    phone = models.CharField(max_length=24, blank=True, null=True)
    email = models.EmailField()
    # create at...
    created = models.DateTimeField(auto_now_add=True)
    # update at...
    updated = models.DateTimeField(auto_now=True)

    @property
    def full_name(self):
        return f'{self.first_name} {self.last_name}'

    def status(self):
        # آیا مشتری لاگین است یا خیر
        pass

```

شکل 8: مدل مشتری

```

class Address(models.Model):
    class meta:
        abstract = True
    address = models.CharField(max_length=70)
    city = models.CharField(max_length=40)
    state = models.CharField(max_length=40, blank=True, null=True)
    country = models.CharField(max_length=40, blank=True, null=True)
    postal_code = models.CharField(max_length=10)
    # create at...
    created = models.DateTimeField(auto_now_add=True)
    # update at...
    updated = models.DateTimeField(auto_now=True)

    def __str__(self):
        return f'{self.address} in {self.city} city'

class Boss(models.Model):
    name = models.CharField(max_length=40)

```

شکل 9: مدل آدرس

بدلیل اینکه هر کاربر باید یک آدرس در پروفایل خود داشته باشد یک کلاس جدا برای آدرس در نظر گرفته شده که کلاس مشتری از آن ارث ببرد و فیلدهای کلاس آدرس را نیز داشته باشد.