from django.db import models

from django.contrib.auth.models import User

from django.utils import timezone

from django.utils.text import slugify

# Create your models here.

class Product(models.Model):

CONDITION\_TYPE = (

("New", "New"),

("Used", "Used")

)

# this class will contain all info of a product

name = models.CharField(max\_length=100)

owner = models.ForeignKey(User, on\_delete=models.CASCADE)

description = models.TextField(max\_length=500)

condition = models.CharField(max\_length=100, choices=CONDITION\_TYPE)

category = models.ForeignKey(

'Category', on\_delete=models.SET\_NULL, null=True)

brand = models.ForeignKey('Brand', on\_delete=models.SET\_NULL, null=True)

price = models.DecimalField(max\_digits=8, decimal\_places=2)

image = models.ImageField(upload\_to='main\_product/', blank=True, null=True)

created = models.DateTimeField(default=timezone.now)

slug = models.SlugField(blank=True, null=True)

def save(self, \*args, \*\*kwargs):

if not self.slug and self.name:

self.slug = slugify(self.name)

super(Product, self).save(\*args, \*\*kwargs)

def \_\_str\_\_(self):

return self.name

class ProductImages(models.Model):

# for product images

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

image = models.ImageField(upload\_to='products/', blank=True, null=True)

class Meta:

verbose\_name = 'Product Image'

verbose\_name\_plural = 'Product Images'

def \_\_str\_\_(self):

return self.product.name

class Category(models.Model):

# for product category

category\_name = models.CharField(max\_length=50)

image = models.ImageField(upload\_to='category/', blank=True, null=True)

slug = models.SlugField(blank=True, null=True)

def save(self, \*args, \*\*kwargs):

if not self.slug and self.category\_name:

self.slug = slugify(self.category\_name)

super(Category, self).save(\*args, \*\*kwargs)

class Meta:

verbose\_name = 'category'

verbose\_name\_plural = 'categories'

def \_\_str\_\_(self):

return self.category\_name

class Brand(models.Model):

# for product brand

brand\_name = models.CharField(max\_length=50)

class Meta:

verbose\_name = 'brand'

verbose\_name\_plural = 'brands'

def \_\_str\_\_(self):

return self.brand\_name