Họ tên: Trần Vân Anh
MSV: B20DCCN075

I. Apps

1. catalog

1.1. catalog/models.py

ShoppingCart > catalog > models.py > ...

```python
from django.db import models
from django.contrib.auth.models import User
from db_mongo_connection import db

# Create your models here.
category_collection = db['Category']


class Category:
    def __init__(self, name, slug, description, is_active=True):
        self.name = name
        self.slug = slug
        self.description = description
        self.is_active = is_active
        self.created_at = None  # You can set these values based on your requirements
        self.updated_at = None

    def __str__(self):
        return self.name

```

1.2. catalog/views.py

ShoppingCart > catalog > views.py > ...

```python
from django.shortcuts import render
from .models import category_collection, Category
from django.http import HttpResponse


# Create your views here.
def init(request):
    records = [
        {
            "name": "Thiếu Nhi",
            "slug": "thieu-nhi",
            "description": "Sách thiếu nhi",
        },
        {
            "name": "Khoa Học",
            "slug": "khoa-hoc",
            "description": "Sách khoa học"
        },
        {
            "name": "Kinh Tế",
            "slug": "kinh-te",
            "description": "Sách kinh tế"
        }
    ]

    category_collection.insert_many(records)

    return HttpResponse("New category is added")

```

1.3. catalog/urls.py

ShoppingCart > catalog > urls.py > ...

```python
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
        path('init/', views.init, name='init_catalog'),
]
```

2. product

2.1. product/models.py

ShoppingCart > product > models.py > ...

```python
from django.db import models
from db_mongo_connection import db
from catalog.models import Category

# Create your models here.
product_collection = db['Product']


class Product:
    def __init__(self, name, author, slug, price, image, is_active=True,
                 is_bestseller=False, quantity=0, description='', categories=None):
        self.name = name
        self.author = author
        self.slug = slug
        self.price = price
        self.image = image
        self.is_active = is_active
        self.is_bestseller = is_bestseller
        self.quantity = quantity
        self.description = description
        self.created_at = None  # You can set these values based on your requirements
        self.updated_at = None
        self.categories = categories or []

    def __str__(self):
        return self.name

```

2.2. product/views.py

ShoppingCart > product > views.py > add_product

```python
from django.shortcuts import render
from .models import product_collection
from django.http import HttpResponse
from .utils import parse_product_from_mongo


# Create your views here.
def init(request):
    records = [
        {…
        {…
        {…
        {…
        {…
        {…
    ]
    product_collection.insert_many(records)
    return HttpResponse("New products is added")


def index(request):
    return HttpResponse("<h1>Product App</h1>")

```

```python

def add_product(request):
    records = {
        "name": "ProductTest",
        "price": 20000,
        "digital": True,
        "image": None
    }

    product_collection.insert_one(records)
    return HttpResponse("New product is added")


def get_all_product(request):
    mongo_data = product_collection.find()

    products = [parse_product_from_mongo(data) for data in mongo_data]
    print(products)
    return HttpResponse(products)

```

2.3. product/urls.py

ShoppingCart > product > urls.py > ...

```python
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
    path('', views.index, name='product_index'),
    path('init/', views.init, name='init_products'),
    path('add/', views.add_product, name='product_add'),
    path('get_all/', views.get_all_product, name='product_get_all')
]

```

3. cart

3.1. cart/models.py

ShoppingCart > cart > models.py > ...

```python
from django.db import models
from django.contrib.auth.models import User


class Cart(models.Model):
    id = models.AutoField(primary_key=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
    active = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def add_to_cart(self, product_slug):
        cart_item = CartItem.objects.get_or_create(product_slug=product_slug)
        self.items.add(cart_item)
        self.save()

    def remove_from_cart(self, product_slug):
        cart_item = CartItem.objects.get(product_slug=product_slug)
        self.items.remove(cart_item)
        self.save()

    def remove_all_from_cart(self):
        for item in self.items.all():
            self.items.remove(item)
        self.save()


```

```python

# Create your models here.
class CartItem(models.Model):
    id = models.AutoField(primary_key=True)
    price = models.FloatField(default=0)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    product_slug = models.CharField(max_length=255, default='')
    cart_id = models.ForeignKey(Cart, on_delete=models.CASCADE, null=True,
                                blank=True, related_name='cart_detail')
    quantity = models.IntegerField(null=False, default=0)

```

3.2. cart/views.py

ShoppingCart > cart > views.py > addToCart

```python

def addToCart(request, product_slug):
    try:
        product = product_collection.find({'slug': product_slug})[0]
        cart = Cart.objects.first()
        if not cart:
            cart = Cart(
            )
            cart.save()
        obj, created = CartItem.objects.update_or_create(
            cart_id_id=cart.id,
            product_slug=product_slug,
            defaults={
                'price': product['price'],
            }
        )
        if not created:
            obj.quantity += 1
        else:
            obj.quantity = 1
        obj.save()
        messages.success(request, "Thành công .")
        return redirect(request.META.get('HTTP_REFERER', 'home'))

    except Exception as e:
        return HttpResponse(str(e))

```

ShoppingCart > cart > views.py > addToCart

```python

def viewCart(request):
    try:
        cart = Cart.objects.first()
        if not cart:
            cart = Cart(
            )
            cart.save()

        data = []

        for cart_detail in cart.cart_detail.all():
            product = product_collection.find({'slug': cart_detail.product_slug})[0]
            product['_id'] = str(product['_id'])
            data.append({
                'id': cart_detail.id,
                'price': cart_detail.price,
                'quantity': cart_detail.quantity,
                'product': product,
                'sum_price': cart_detail.price * cart_detail.quantity,
            })
        # cartDetail = cart.cart_detail.all()
        return render(request, "cart.html", get_context({"cart": data}))

    except Exception as e:
        return HttpResponse(str(e))

```

ShoppingCart > cart > views.py > addToCart

```python

def changeQuantity(request):
    try:
        if request.method == 'GET':
            raise Exception('Not support method')
        cart_detail_id = request.POST.get('id')
        type = request.POST.get('type')

        cartDetail = CartItem.objects.get(pk=cart_detail_id)

        if type == '+':
            cartDetail.quantity += 1
            cartDetail.save()

        else:
            if cartDetail.quantity == 1:
                cartDetail.delete()
            else:
                cartDetail.quantity -= 1
                cartDetail.save()

        messages.success(request, "Thành công .")
        return redirect(request.META.get('HTTP_REFERER', 'cart'))
    except Exception as e:
        return HttpResponse(str(e))

```

3.3. cart/urls.py

ShoppingCart > cart > urls.py > ...

```python
from django.contrib import admin
from django.urls import path, include
from . import views

urlpatterns = [
    path('add_to_cart/<str:product_slug>/',views.addToCart, name='addToCart'),
]
```

4. app(checkout+search)

4.1. app/models.py

ShoppingCart > app > models.py > ...

```python
from django.db import models
from django.contrib.auth.models import User
from product.models import Product


class Customer(models.Model):
    user = models.OneToOneField(User, null=True, blank=False,
                                on_delete=models.SET_NULL)  # OneToOneField: One user has one customer
    name = models.CharField(max_length=200, null=True)
    email = models.CharField(max_length=200, null=True)

    def __str__(self):
        return self.name

```

4.2. app/views.py

ShoppingCart > app > views.py > catalog

```python


# Create your views here.
def home(request):
    mongo_data = product_collection.find()
    products = [parse_product_from_mongo(data) for data in mongo_data]

    # products = Product.objects.all()

    context = get_context({'products': products})
    return render(request, 'app/home.html', context)


def checkout(request):
    context = get_context({})
    return render(request, 'app/checkout.html', context)


def catalog(request, category_slug):
    query = {'category': category_slug}
    products_data = product_collection.find(query)

    products = [parse_product_from_mongo(product_data) for product_data in products_data]

    if not products:
        return HttpResponse(product_collection.find())

    context = get_context({'products': products})
    return render(request, 'app/catalog.html', context)

```

```
ShoppingCart > app > views.py > catalog
40  def search(request):
42      if request.method == 'POST':
43          keyword = request.POST.get('keyword')
44
45      product_collection.create_index([('name', 'text'), ('author', 'text'), ('description', 'text')])
46
47      # Truy vấn để tìm kiếm kể cả khi keyword không đầy đủ
48      search_keyword = re.compile(f".*{re.escape(keyword)}.*", re.IGNORECASE)
49      query = {
50          '$or': [
51              {'name': {'$regex': search_keyword}},
52              {'author': {'$regex': search_keyword}},
53              {'description': {'$regex': search_keyword}}
54          ]
55      }
56      products_data = product_collection.find(query)
57      products = [parse_product_from_mongo(product_data) for product_data in products_data]
58
59      if not products:
60          print(f"No products found with keyword: {keyword}")
61
62      context = get_context({'keyword': keyword, 'products': products})
63      return render(request, 'app/search.html', context)
64
65
66  # BASE
67  def get_base_context():
68      mongo_data = category_collection.find()
69      categories = [parse_category_from_mongo(data) for data in mongo_data]
70      context = {'categories': categories}
71      return context
72
73
74  def get_context(dict2):
75      res = {**get_base_context(), **dict2}
76      return res
```

4.3. app/urls.py

```
urlpatterns = [
    path('', views.home, name="home"),
    path('catalog/<str:category_slug>/', views.catalog, name="catalog"),
    path('cart/index', cart.viewCart, name="cart"),
    path('cart/changeQuantity', cart.changeQuantity, name='changeQuantity'),
    path('checkout/', views.checkout, name="checkout"),
    path('search/', views.search, name="search"),
]
```

II. ShoppingCart

1. INSTALLED_APPS

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app',
    'product',
    'catalog',
    'cart'
]
```

2. Database

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'shoppingcart',
        'USER': 'root',
        'PASSWORD': '888888',
        'HOST': 'localhost'
    }
}
```

```
import pymongo

url = 'mongodb://localhost:27017'
client = pymongo.MongoClient(url)
db = client['shoppingcart']
```

3. urlpatterns

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('app.urls')),
    path('product/', include('product.urls')),
    path('category/', include('catalog.urls')),
    path('cart/', include('cart.urls'))
]
```