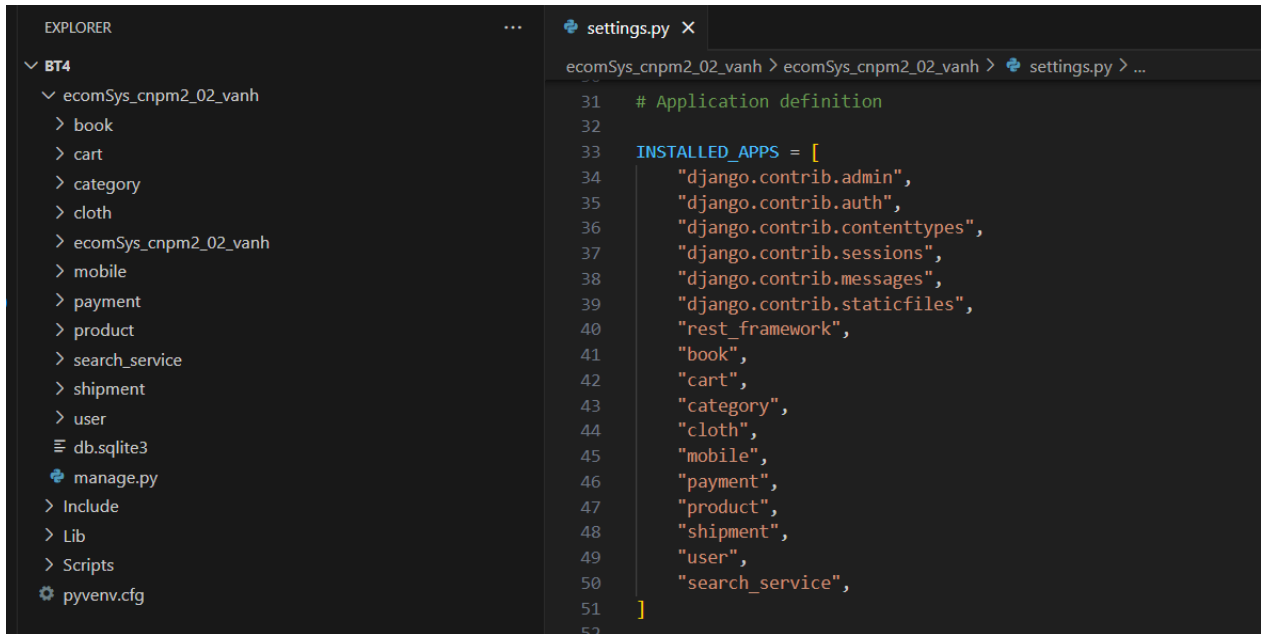


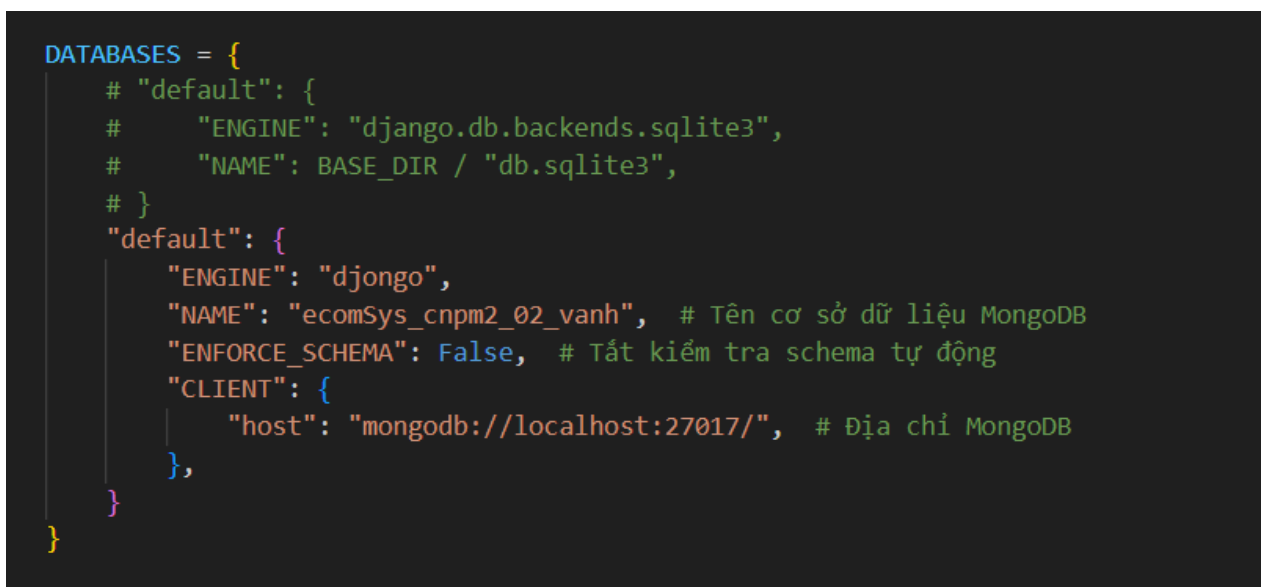
Họ tên: Trần Văn Anh

MSV: B20DCCN075

## 1. 10 apps + design with MongoDB



```
31 # Application definition
32
33 INSTALLED_APPS = [
34     "django.contrib.admin",
35     "django.contrib.auth",
36     "django.contrib.contenttypes",
37     "django.contrib.sessions",
38     "django.contrib.messages",
39     "django.contrib.staticfiles",
40     "rest_framework",
41     "book",
42     "cart",
43     "category",
44     "cloth",
45     "mobile",
46     "payment",
47     "product",
48     "shipment",
49     "user",
50     "search_service",
51 ]
52
```



```
DATABASES = {
    # "default": {
    #     "ENGINE": "django.db.backends.sqlite3",
    #     "NAME": BASE_DIR / "db.sqlite3",
    # }
    "default": {
        "ENGINE": "djongo",
        "NAME": "ecomSys_cnpm2_02_vanh", # Tên cơ sở dữ liệu MongoDB
        "ENFORCE_SCHEMA": False, # Tắt kiểm tra schema tự động
        "CLIENT": {
            "host": "mongodb://localhost:27017/", # Địa chỉ MongoDB
        },
    },
}
```

## 2. Rest API

## ▼ shopping-cart

GET http://127.0.0.1:8000/api/book

POST http://127.0.0.1:8000/api/book

GET http://127.0.0.1:8000/api/book/5

GET http://127.0.0.1:8000/api/cart

GET http://127.0.0.1:8000/api/cart/7

GET http://127.0.0.1:8000/api/search/?query=naruto

POST http://127.0.0.1:8000/api/cart/

```
ecomSys_cnpm2_02_vanh > ecomSys_cnpm2_02_vanh > 📄 urls.py > ...
```

```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path("admin/", admin.site.urls),
6     path("api/book/", include("book.urls")),
7     path("api/cart/", include("cart.urls")),
8     path("api/search/", include("search_service.urls")),
9 ]
```

### 3. Search + cart process

#### 3.1. Create book, view book

*a. models.py*

```
ecomSys_cnpm2_02_vanh > book > models.py > ...  
1  from django.db import models  
2  
3  class Book(models.Model):  
4      title = models.CharField(max_length=100)  
5      author = models.CharField(max_length=100)  
6      price = models.DecimalField(max_digits=10, decimal_places=2)  
7  
8      def __str__(self):  
9          return self.title  
10
```

*b. views.py*

```
ecomSys_cnpm2_02_vanh > book > views.py > ...  
1  from rest_framework import generics  
2  from .models import Book  
3  from .serializers import BookSerializer  
4  
5  class BookListCreate(generics.ListCreateAPIView):  
6      queryset = Book.objects.all()  
7      serializer_class = BookSerializer  
8  
9  class BookRetrieve(generics.RetrieveAPIView):  
10     queryset = Book.objects.all()  
11     serializer_class = BookSerializer  
12  
13     class BookList(generics.ListAPIView):  
14         queryset = Book.objects.all()  
15         serializer_class = BookSerializer  
16
```

*c. serializers.py*

```
ecomSys_cnpm2_02_vanh > book > serializers.py > ...
1  from rest_framework import serializers
2  from .models import Book
3
4  class BookSerializer(serializers.ModelSerializer):
5      class Meta:
6          model = Book
7          fields = '__all__'
8
```

*d. urls.py*

```
ecomSys_cnpm2_02_vanh > book > urls.py > ...
1  from django.urls import path
2  from .views import BookListCreate, BookRetrieve, BookList
3
4  urlpatterns = [
5      path('', BookListCreate.as_view(), name='book-list-create'),
6      path('<int:pk>/', BookRetrieve.as_view(), name='book-retrieve'),
7      path('all/', BookRetrieve.as_view(), name='book-list'),
8  ]
9
```

### 3.2. Create cart, view cart

*a. models.py*

```
ecomSys_cnpm2_02_vanh > cart > models.py > ...
1  from django.db import models
2  from book.models import Book
3
4  class CartItem(models.Model):
5      book = models.ForeignKey(Book, on_delete=models.CASCADE)
6      quantity = models.PositiveIntegerField(default=1)
7
8      def __str__(self):
9          return f"{self.quantity} x {self.book.title}"
10
```

*b. views.py*

```
ecomSys_cnpm2_02_vanh > cart > views.py > ...
1  from rest_framework import generics
2  from .models import CartItem
3  from .serializers import CartItemSerializer
4
5  class CartItemCreate(generics.ListCreateAPIView):
6      queryset = CartItem.objects.all()
7      serializer_class = CartItemSerializer
8
9  class CartItemList(generics.ListAPIView):
10     queryset = CartItem.objects.all()
11     serializer_class = CartItemSerializer
12
13  class CartItemRetrieve(generics.RetrieveAPIView):
14     queryset = CartItem.objects.all()
15     serializer_class = CartItemSerializer
16
```

*c. serializers.py*

```
ecomSys_cnpm2_02_vanh > cart > serializers.py > ...
1 from rest_framework import serializers
2 from .models import CartItem
3 from book.models import Book
4 from book.serializers import BookSerializer
5
6
7 class CartItemSerializer(serializers.ModelSerializer):
8     book_id = serializers.IntegerField(write_only=True)
9     book = serializers.IntegerField(
10         source="book.id", read_only=True
11     ) # Trường book để đọc thông tin của sách
12
13     class Meta:
14         model = CartItem
15         fields = [
16             "id",
17             "book_id",
18             "book",
19             "quantity",
20         ]
21
22     def validate_book_id(self, value):
23         try:
24             Book.objects.get(pk=value)
25         except Book.DoesNotExist:
26             raise serializers.ValidationError("Book with this title does not exist.")
27         return value
28
29     def create(self, validated_data):
30         book_id = validated_data.pop("book_id")
31         book = Book.objects.get(pk=book_id)
32         cart_item = CartItem.objects.create(book=book, **validated_data)
33         return cart_item
34
```

#### *d. urls.py*

```
ecomSys_cnpm2_02_vanh > cart > urls.py > ...
1 from django.urls import path
2 from .views import CartItemCreate, CartItemList, CartItemRetrieve
3
4 urlpatterns = [
5     path("", CartItemCreate.as_view(), name="cart-item-create"),
6     path("<int:pk>/", CartItemRetrieve.as_view(), name="cart-item-retrieve"),
7     path("all/", CartItemList.as_view(), name="cart-item-list"),
8 ]
9
```

### 3.3. Search

#### *a. views.py*

```
ecomSys_cnpm2_02_vanh > search_service > views.py > ...
1  from rest_framework import generics
2  from rest_framework.response import Response
3  from book.models import Book
4  from .serializers import BookSearchSerializer
5
6  # http://127.0.0.1:8000/api/search/?query=conan
7
8  class BookSearch(generics.ListAPIView):
9      serializer_class = BookSearchSerializer
10
11     def get_queryset(self):
12         query = self.request.query_params.get('query', None)
13         if query:
14             queryset = Book.objects.filter(title__icontains=query) | Book.objects.filter(author__icontains=query)
15             return queryset
16         else:
17             return Book.objects.none()
18
19     def list(self, request, *args, **kwargs):
20         queryset = self.get_queryset()
21         serializer = self.get_serializer(queryset, many=True)
22         return Response(serializer.data)
23
```

### *b. serializers.py*

```
ecomSys_cnpm2_02_vanh > search_service > serializers.py > ...
1  from rest_framework import serializers
2  from book.models import Book
3
4  class BookSearchSerializer(serializers.ModelSerializer):
5      class Meta:
6          model = Book
7          fields = ['id', 'title', 'author']
8
```

### *c. urls.py*

```
ecomSys_cnpm2_02_vanh > search_service > urls.py > ...
1  from django.urls import path
2  from .views import BookSearch
3
4  urlpatterns = [
5      path('', BookSearch.as_view(), name='book-search'),
6  ]
7
```