

به نام خدا

مستندات پروژه درس رایانش ابری

موضوع پروژه :

فروشگاه محصولات داکرایز شده،

با استفاده از فریم‌ورک DJANGO و

پایگاه داده MYSQL

استاد مربوطه : سرکار خانم دکتر هدی طاهری

دانشجو :

محمد سجاد رحمانی

۴۰۰۱۲۷۷۳۳۰

دی ماه ۱۴۰۳

۳	بخش اول معرفی و تنظیمات
۱۲	بخش دوم راه اندازی و تست با POST MAN:

بخش اول معرفی و تنظیمات

این پروژه در واقع CRUD محصولات یک فروشگاه می باشد با دسته بندی های متفاوت.

پیش نیاز های پروژه شامل نصب و راه اندازی Docker و Docker compose و Git می باشد.

تست نصب این ابزار :

```
mohammadajjad@victus: ~$ git version
git version 2.43.0
mohammadajjad@victus: ~$ sudo docker version
Client: Docker Engine - Community
Version: 27.4.1
API version: 1.47
Go version: go1.22.10
Git commit: 85d17ea
Built: Tue Dec 17 15:45:42 2024
OS/Arch: linux/amd64
Context: default

Server: Docker Engine - Community
Engine:
Version: 27.4.1
API version: 1.47 (minimum version 1.24)
Go version: go1.22.10
Git commit: c710888
Built: Tue Dec 17 15:45:42 2024
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.7.23
GitCommit: 57170aa6295a39009d861b89c3b3b7b085ca27
runc:
Version: 1.1.14
GitCommit: v1.1.14-0-g2c9f560
docker-init:
Version: 0.19.0
GitCommit: de40ad9
mohammadajjad@victus: ~$ sudo docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
Cython version: 3.12.3
OpenSSL version: OpenSSL 3.0.13 30 Jan 2024
mohammadajjad@victus: ~$
```

ساختار درختی فایل های موجود در این پروژه و محل قرار گیری هر یک از دایرکتوری ها و فایل ها مختلف :

```
online_shop/
├── app/                                # Django app source code
│   ├── manage.py
│   ├── online_shop/                  # Main Django project folder
│   │   ├── __init__.py
│   │   ├── settings.py
│   │   ├── urls.py
│   │   ├── wsgi.py
│   │   └── asgi.py
│   ├── shop/                        # Django app for products
│   │   ├── __init__.py
│   │   ├── admin.py
│   │   ├── apps.py
│   │   ├── models.py
│   │   ├── serializers.py
│   │   ├── tests.py
│   │   ├── urls.py
│   │   └── views.py
├── docker/
│   ├── django/                      # Docker setup for the Django app
│   │   ├── Dockerfile
│   │   └── entrypoint.sh
│   ├── db/                          # Docker setup for the MySQL database
│   │   └── init.sql                 # Optional: SQL script for initializing the database
├── docker-compose.dev.yml           # For development
├── docker-compose.prod.yml         # For production
└── requirements.txt                 # Python dependencies
```

محتویات فایل requirements.txt :

```
asgiref==3.8.1
Django==5.1.4
djangorestframework==3.15.2
mysqlclient==2.2.6
sqlparse==0.5.3
python-decouple-
```

محتویات فایل های داکر فایل و فایل های داکر کامپوز :

```
# Base image
FROM python:3.10-slim

# Environment variables
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

# Install system dependencies
RUN apt-get update && apt-get install -y \
    gcc \
    libmariadb-dev \
    libmariadb-dev-compat \
    default-libmysqlclient-dev \
    pkg-config \
    && apt-get clean

# Set work directory
WORKDIR /app

# Install Python dependencies
COPY requirements.txt /app/
RUN pip install --no-cache-dir -r requirements.txt

# Copy project files
COPY . /app/
```

```
version: '3.9'

services:
  app:
    build:
      context: .
      dockerfile: docker/django/Dockerfile
    container_name: online_shop_app
    command: python manage.py runserver 0.0.0.0:8000
    volumes:
      - ./app:/app
    env_file:
      - .env # Reference the .env file
    ports:
      - "8000:8000"
    depends_on:
      - db

  db:
    image: mysql:8.0
    container_name: online_shop_db
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: online_shop
    ports:
      - "3306:3306"
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

```
version: '3.9'

services:
  app:
    build:
      context: .
      dockerfile: docker/django/Dockerfile
    container_name: online_shop_app
    command: gunicorn online_shop.wsgi:application --bind 0.0.0.0:8000
    ports:
      - "8000:8000"
    env_file:
      - .env # Reference the .env file
    depends_on:
      - db

  db:
    image: mysql:8.0
    container_name: online_shop_db
    environment:
      MYSQL_ROOT_PASSWORD: password
      MYSQL_DATABASE: online_shop
    ports:
      - "3306:3306"
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

تصاویر مربوط به settings.py :

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'shop',  
]
```

```
SECRET_KEY = config('SECRET_KEY')  
DEBUG = config('DEBUG', default=False, cast=bool)  
ALLOWED_HOSTS = config('ALLOWED_HOSTS', default='').split(',')  
  
DATABASES = {  
    'default': {  
        'ENGINE': config('DB_ENGINE'),  
        'NAME': config('DB_NAME'),  
        'USER': config('DB_USER'),  
        'PASSWORD': config('DB_PASSWORD'),  
        'HOST': config('DB_HOST'),  
        'PORT': config('DB_PORT'),  
    }  
}  
  
LANGUAGE_CODE = config('LANGUAGE_CODE', default='en-us')  
TIME_ZONE = config('TIME_ZONE', default='UTC')
```


محتوای فایل .env

```
# Django settings
SECRET_KEY=django-insecure-ifgd)l+!x+a+qgkfpk*x@i=@!6$zh1ka5&8v4zj5m_uefb-iri
DEBUG=True
ALLOWED_HOSTS=localhost,127.0.0.1,0.0.0.0

# Database configuration
DB_ENGINE=django.db.backends.mysql
DB_NAME=online_shop
DB_USER=root
DB_PASSWORD=password
DB_HOST=db
DB_PORT=3306

# Application-specific settings
TIME_ZONE=UTC
LANGUAGE_CODE=en-us
```

تصاویر مربوط به فایل‌های `serializers.py` و `urls.py` و `views.py` و `models.py` و
و `urls.py` در اپ اصلی :

```
from django.db import models

# Create your models here.

class Category(models.Model):
    name = models.CharField(max_length=255, unique=True)
    description = models.TextField(blank=True)

    def __str__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    stock = models.PositiveIntegerField()
    category = models.ForeignKey(Category, related_name='products', on_delete=models.CASCADE)

    def __str__(self):
        return self.name
```

```

from rest_framework import viewsets
from .models import Product, Category
from .serializers import ProductSerializer, CategorySerializer

class CategoryViewSet(viewsets.ModelViewSet):
    queryset = Category.objects.all()
    serializer_class = CategorySerializer

class ProductViewSet(viewsets.ModelViewSet):
    queryset = Product.objects.all()
    serializer_class = ProductSerializer
from django.shortcuts import render

# Create your views here.

```

```

from django.urls import path, include
from rest_framework.routers import DefaultRouter
from .views import ProductViewSet, CategoryViewSet

router = DefaultRouter()
router.register(r'categories', CategoryViewSet)
router.register(r'products', ProductViewSet)

urlpatterns = [
    path('', include(router.urls)),
]

```

```

"""
URL configuration for online_shop project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls), # Default admin site
    path('api/', include('shop.urls')), # Include the URLs from your app
]

```

بخش دوم راه اندازی و تست با POST MAN :

برای ساختن و راه اندازی container ها دستورات زیر را وارد می کنیم :

```
The Edit View Search Terminal Help
mohammedajid@victus: ~/Desktop/online_shop

mohammedajid@victus:~/Desktop/online_shop$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
mohammedajid@victus:~/Desktop/online_shop$ sudo docker compose -f docker-compose.dev.yml up -d
Creating network "online_shop_default" with the default driver
Creating online_shop_db ... done
Creating online_shop_app ... done
mohammedajid@victus:~/Desktop/online_shop$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
da4231c8f8b   online_shop_app   "python manage.py ru..."   8 seconds ago   Up 7 seconds   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp   online_shop_app
036af62f1f5f   mysql:8.0       "docker-entrypoint.s..."   8 seconds ago   Up 7 seconds   0.0.0.0:3306->3306/tcp, :::3306->3306/tcp   online_shop_db
mohammedajid@victus:~/Desktop/online_shop$
```

پس از راه اندازی container ها لازم است که مراحل زیر را دنبال کنیم، در اولین راه اندازی داده‌ایی موجود نیست :

```
The Edit View Search Terminal Help
mohammedajid@victus: ~/Desktop/online_shop

mohammedajid@victus:~/Desktop/online_shop$ sudo docker exec -it online_shop_db bash
mysql: [Warning] Using a password on the command line interface can be insecure.
mysql> use online_shop;
mysql> apply all migrations;
mysql> exit;

mohammedajid@victus:~/Desktop/online_shop$ sudo docker exec -it online_shop_app bash
bash: /bin/bash: file not found
docker exec: error: exec of /bin/bash failed: exec: "/bin/bash": executable file not found in $PATH
You may need to run the Docker Desktop, Compose and with -- or --up
Your MySQL connection id is 16
Server Version: 8.0.30 MySQL Community Server - GPL
Copyright (c) 2000, 2024 Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help; type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| online_shop |
| performance_schema |
| sys |
+-----+
1 row in set (0.00 sec)

mysql> USE online_shop;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_online_shop |
+-----+
| auth_group |
| auth_group_permissions |
| auth_permission |
| auth_user |
| auth_user_groups |
| auth_user_permissions |
| django_admin_log |
| django_content_type |
| django_migrations |
| django_sessions |
| shop_category |
| shop_product |
| shop_promotion |
+-----+
13 rows in set (0.00 sec)

mysql> SELECT * FROM shop_product;
+-----+
| id | name | description | price | stock | category_id |
+-----+
| 1 | Iphone 15 | Last Product From Apple | 550.00 | 1 | 1 |
| 2 | Iphone 14 | Last Product From Apple | 350.00 | 1 | 1 |
+-----+
2 rows in set (0.00 sec)

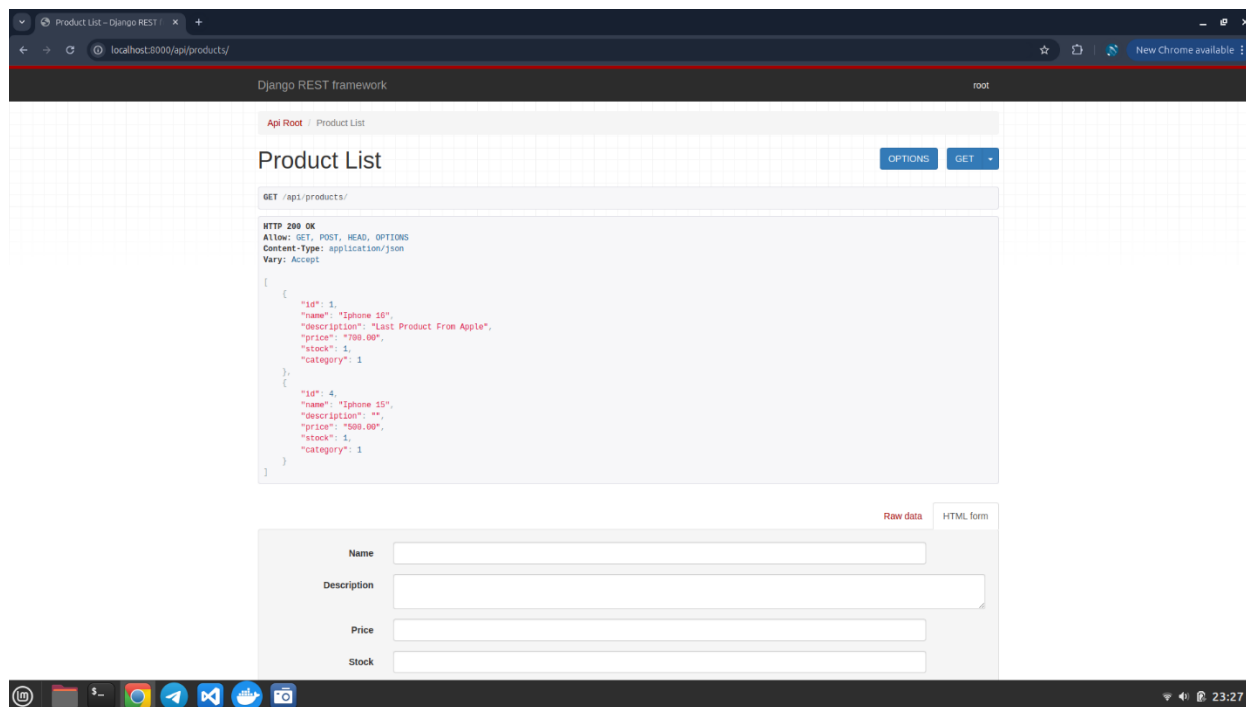
mysql> SELECT * FROM shop_category;
+-----+
| id | name | description |
+-----+
| 1 | Mobile | This Category is |
+-----+
1 row in set (0.00 sec)

mysql> EXIT;
Bye, I'll exit
mohammedajid@victus:~/Desktop/online_shop$
```

برای استفاده از `python manage.py migrate` باید وارد `bash` کانتینر اصلی شویم و باید دستور `docker exec -it online_shop_app bash` را وارد کنیم.

پس از انجام مراحل بالا پروژه بر روی `Port 8000` سیستم لوکال اجرا می‌شود.

تصویر مربوط به اجرای پروژه :



تصاویر مربوط به تست افزونه POST MAN در VS CODE :

