به نام خدا

مستندات پروژه درس رایانش ابری

موضوع پروژه :

فروشگاه محصولات داکرایز شده،

با استفاده از فریمووورک DJANGO و

پایگاه داده MYSQL

استاد مربوطه : سرکارخانم دکتر هدی طاهری

دانشجو :

محمدسجاد رحمانی

۴۰۰۱۲۷۷۳۳۰

دی ماه ۱۴۰۳

فهرست مطالب

بخش اول معرفی و تنظیمات

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

پیش نیاز های پروژه شامل نصب و راه‌اندازی Docker و Docker copmpose و Git می‌باشد.

تست نصب این ابزار :

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 rquirements.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 فایل‌های داکرکامپوز :

```dockerfile
# 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/
```

فایل های dev. و prod.

```yaml
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:
```

```yaml
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$zhlka5&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 models.py و views.py و urls.py و serializers.py و urls.py در اپ اصلی :

```python
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
```

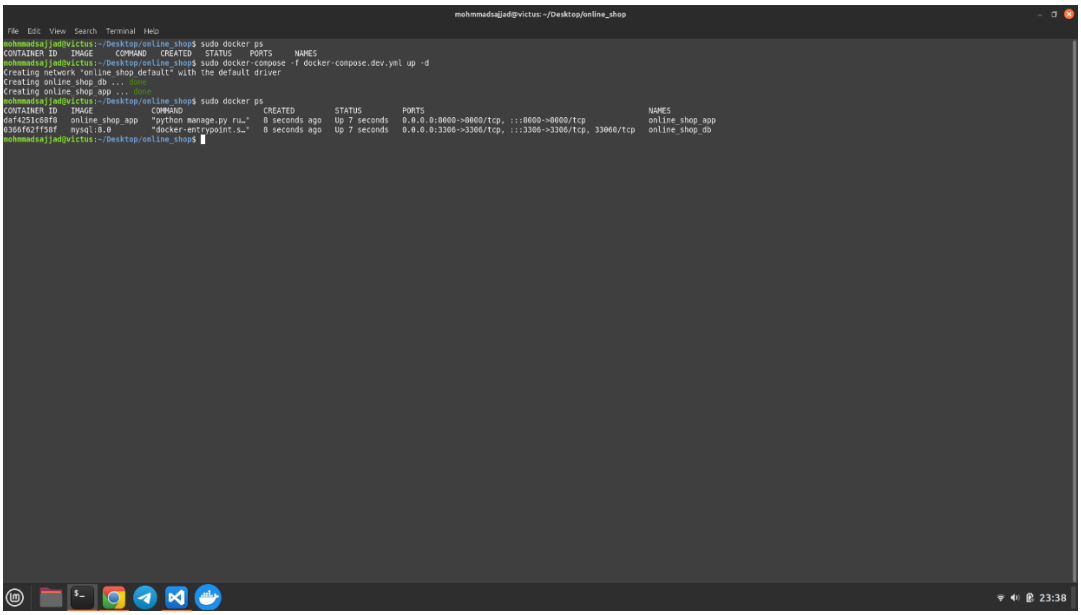
```python
from rest_framework import viewsets
from .models import Product, Category
from .serializers import ProductSerializer, CategorySerializer

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.
```

```python
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)),
]
```

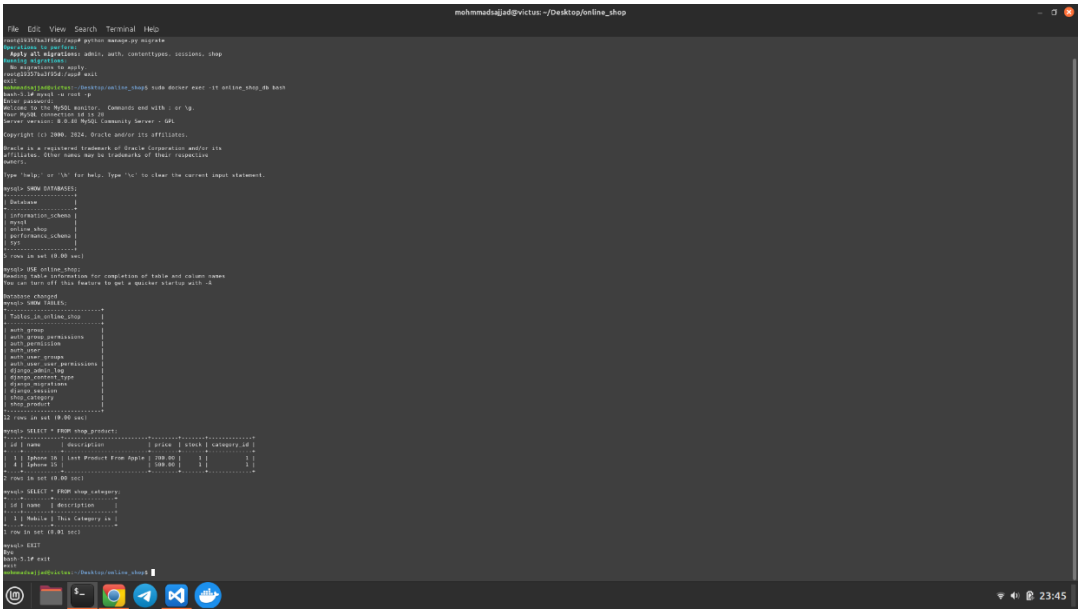
```python
"""
URL configuration for online_shop project.

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 می‌کنیم :



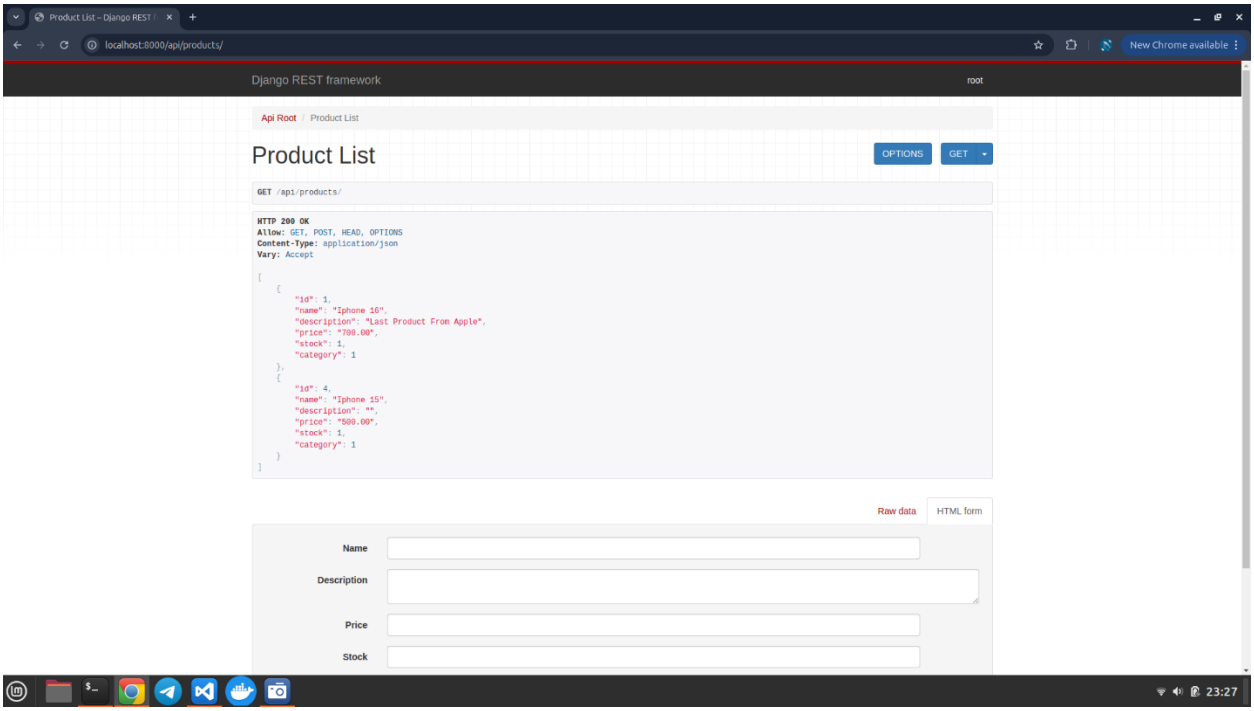
پس از راه اندازی container ها لازم است که مراحل زیر را دنبال کنیم، در اولین راه اندازی داده‌ایی موجود نیست :



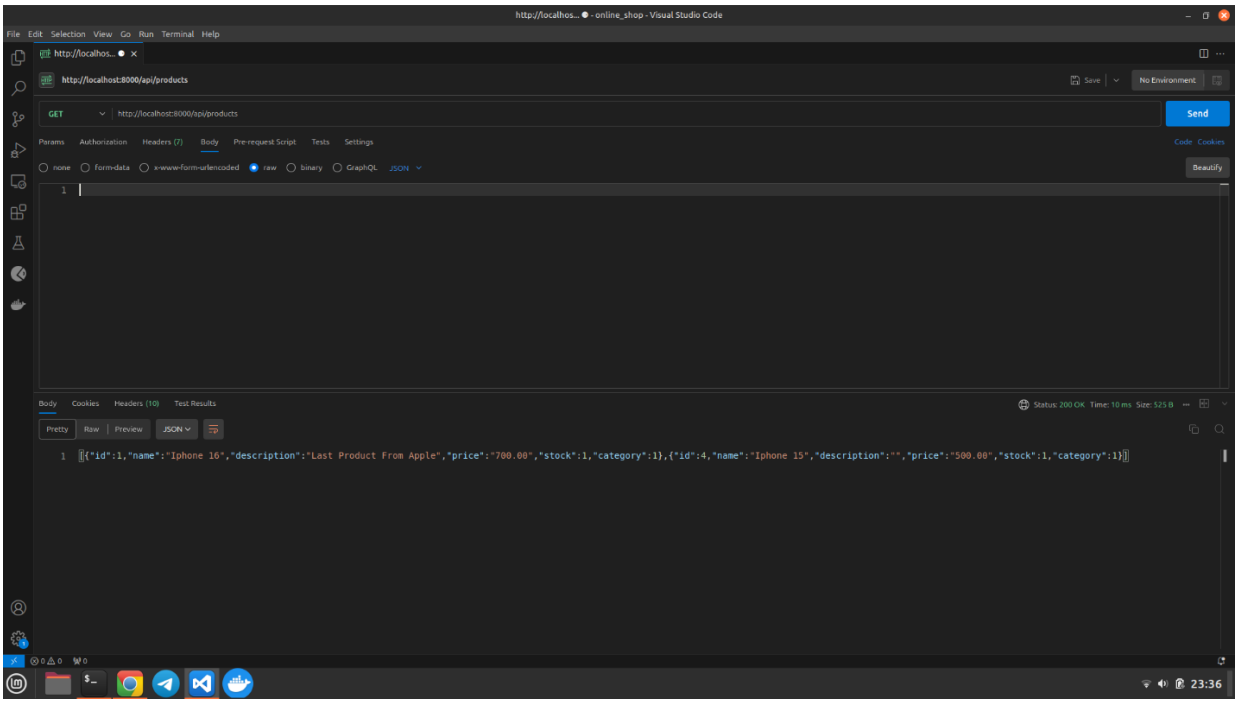
برای استفاده از python manage.py migrate باید وارد bash کانتینر اصلی شویم و باید دستور docker exec -it online_shop_app bash را وارد کنیم.

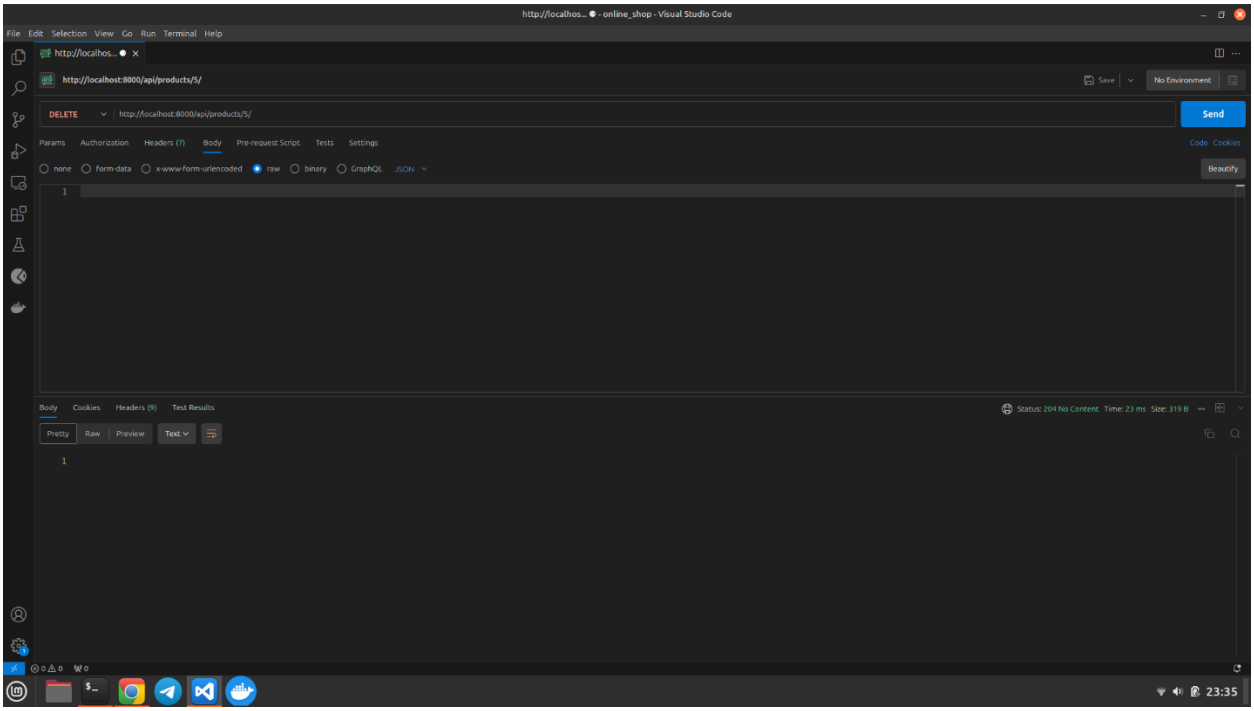
پس از انجام مراحل بالا پروژه بر روی Port 8000 سیستم لوکال اجرا می‌شود.

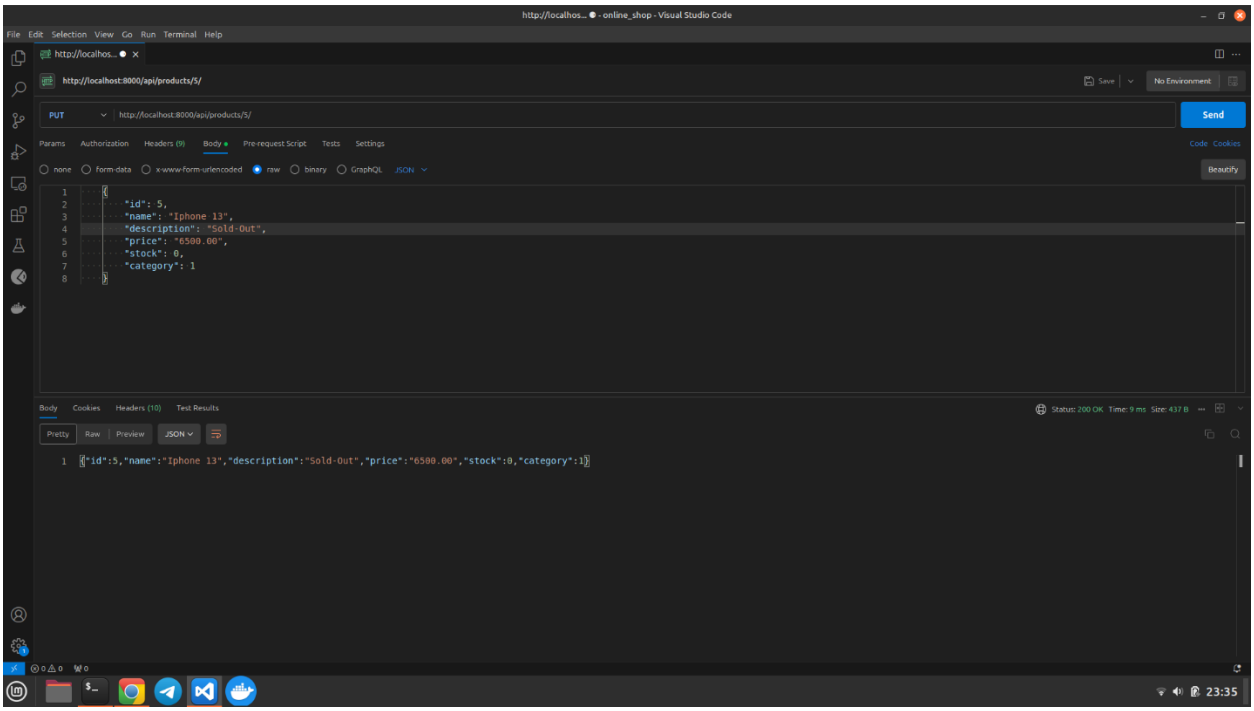
تصویر مربوط به اجرای پروژه :

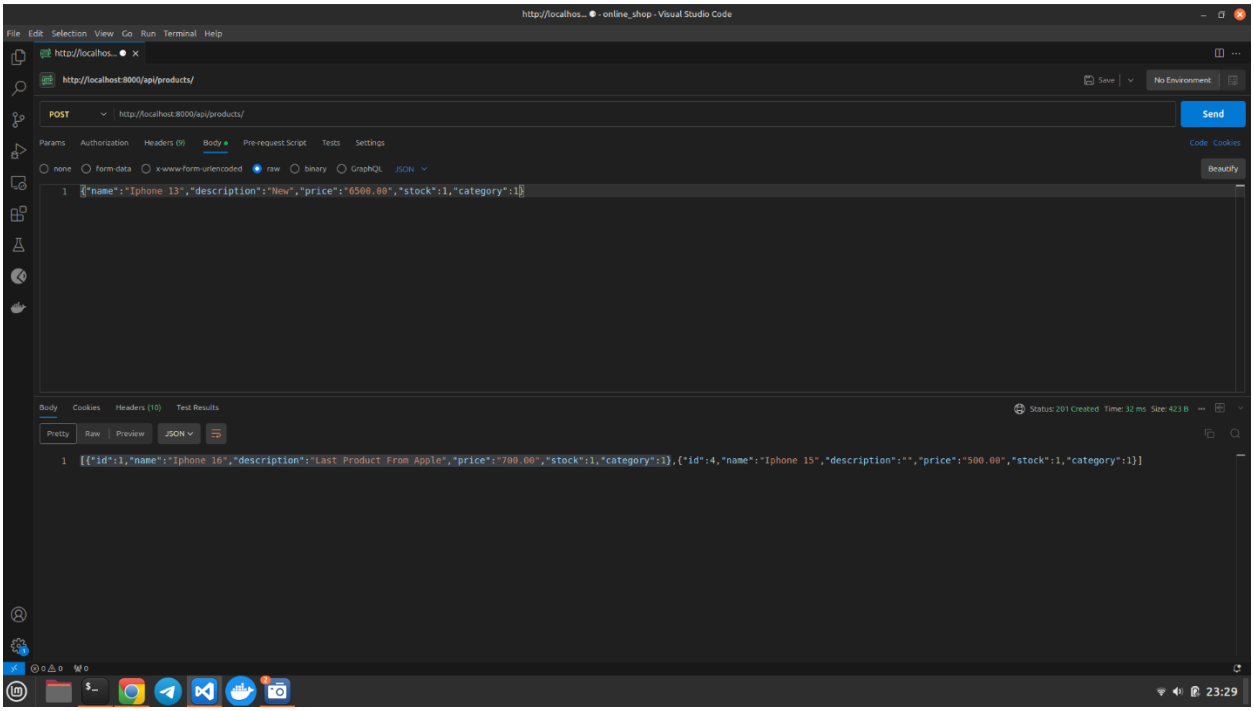


تصاویر مربوط به تست افزونه POST MAN در VS CODE :






http://localhos... - online_shop - Visual Studio Code
File Edit Selection View Go Run Terminal Help
http://localhos...
http://localhost:8000/api/products/5/
Save
No Environment
PUT
http://localhost:8000/api/products/5/
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Code Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{
"id": 5,
"name": "Iphone 13",
"description": "Sold-Out",
"price": "6500.00",
"stock": 0,
"category": 1
}
Body Cookies Headers (10) Test Results
Status: 200 OK Time: 9 ms Size: 437 B
Pretty Raw Preview JSON
{"id":5,"name":"Iphone 13","description":"Sold-Out","price":"6500.00","stock":0,"category":1}
23:35


http://localhos... - online_shop - Visual Studio Code
File Edit Selection View Go Run Terminal Help
http://localhos...
http://localhost:8000/api/products/
Save
No Environment
POST
http://localhost:8000/api/products/
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Code Cookies
none form-data x-www-form-urlencoded raw binary GraphQL JSON
Beautify
{"name":"Iphone 13","description":"New","price":"6500.00","stock":1,"category":1}
Body Cookies Headers (10) Test Results
Status: 201 Created Time: 32 ms Size: 423 B
Pretty Raw Preview JSON
[{"id":1,"name":"Iphone 16","description":"Last Product From Apple","price":"700.00","stock":1,"category":1},{"id":4,"name":"Iphone 15","description":"","price":"500.00","stock":1,"category":1}]
23:29