МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Практическая работа

по курсу: «Архитектура бизнес-приложений»

Выполнил:

студент гр. КТБЗ 3-1

Фролов Д. А.

Проверил:

Свиридов А.С.

Таганрог 2025

Вариант 2

Категория книг для рекомендации - SCIENCE\_FICTION

Количество возвращаемых книг - 4

Ход работы:

Сначала создадим рабочую папку WebService и определим структуру каталогов в ней:

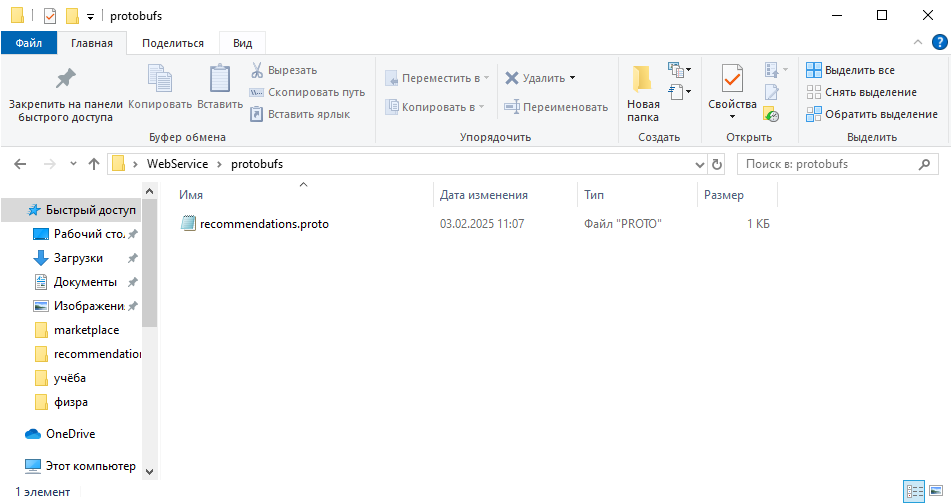


Рис.1 – Структура каталогов в папке protobufs

Каталог protobufs/ будет содержать файл с именем recommendations.proto.

По сформированному описанию API сгенерируем код Python для взаимодействия с ним в каталоге recommendations/.

Во-первых, мы должны установить grpcio-tools. Создадим файл recommendations/requirements.txt и добавим следующее:

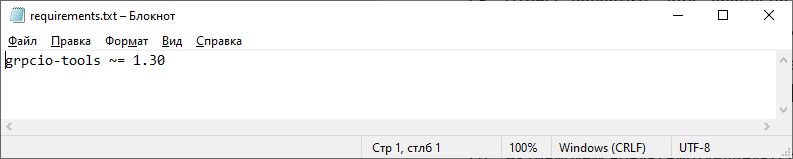


Рис.2 – Установка grpcio-tools

Чтобы запустить код локально запустим терминал командной строки cmd. Дальнейшие действия выполняем в нем, перейдя в рабочую папку проекта

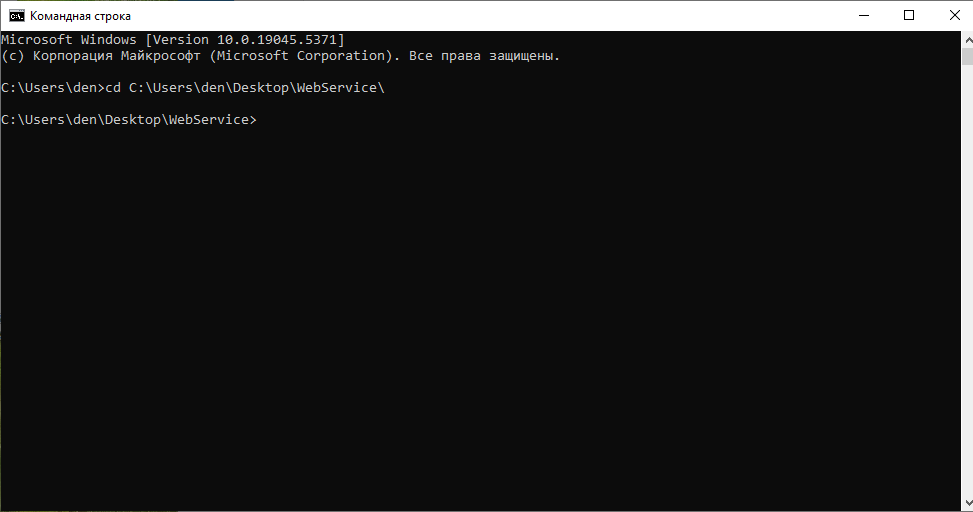


Рис.3 – Рабочая папка проекта

Активируем новую виртуальную среду:

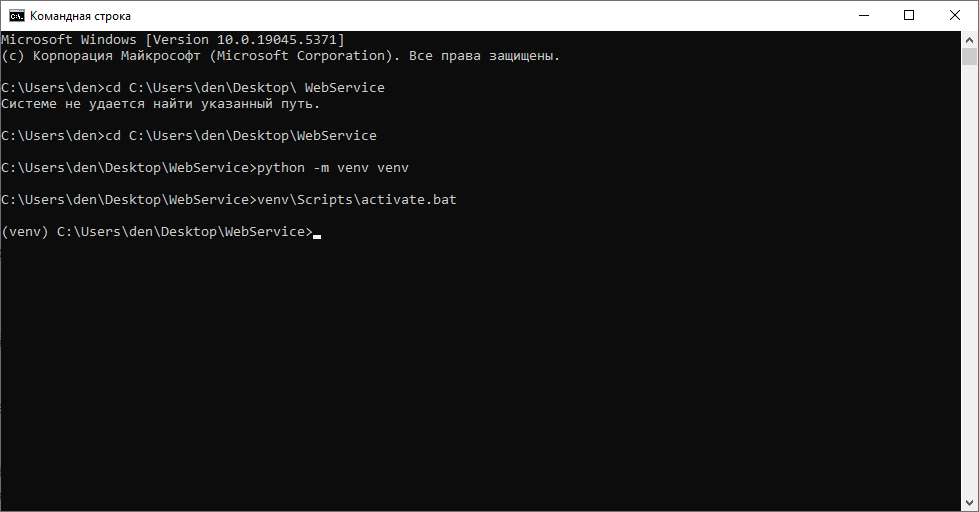


Рис.3 – Активация новой виртуальной среды

Установим в нее зависимости:

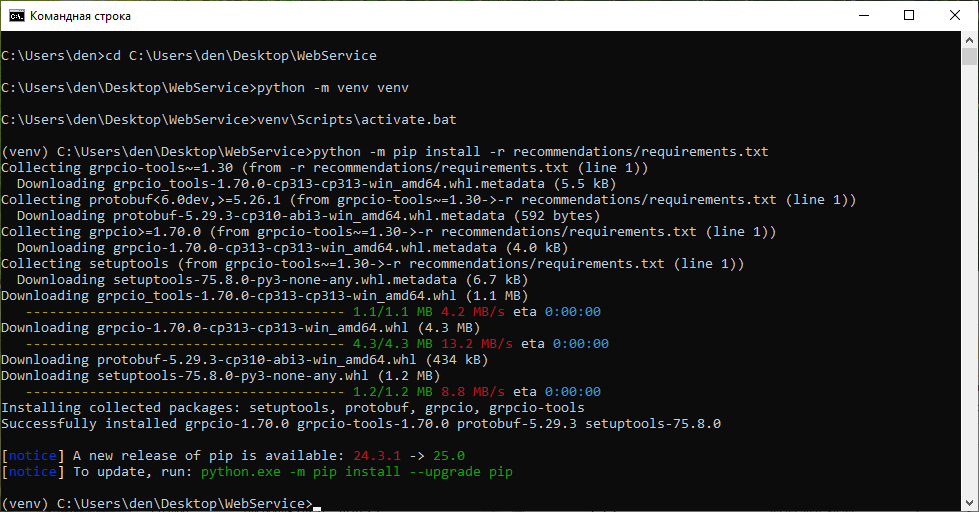


Рис.4 – установка зависимостей

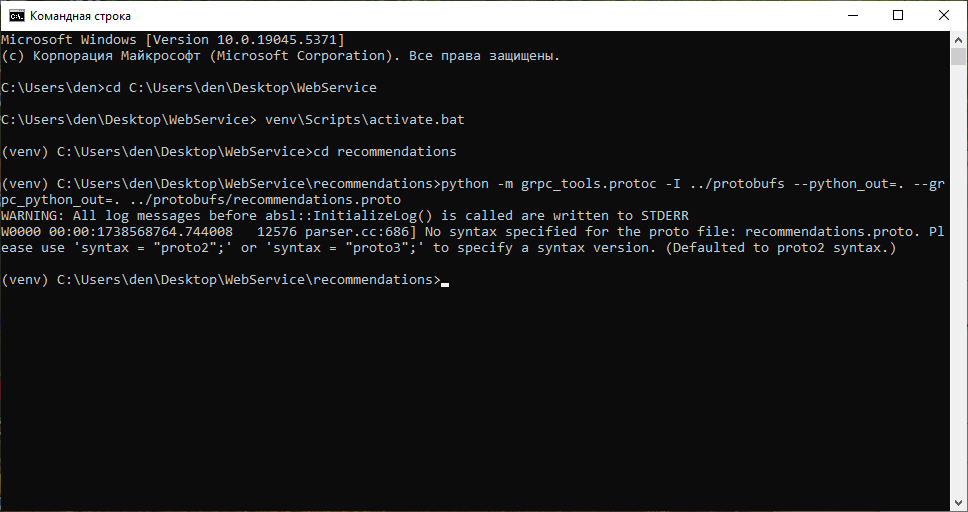
Cгенерируем код Python для API из protobufs, выполнив следующую команду:

Рис.5 - Генерация кода Python для API

Создание сервера RPC.

Создадим файл recommendations.py с кодом сервиса рекомендаций в подпапке проекта recommendations/ и сохраним в нем следующее содержание

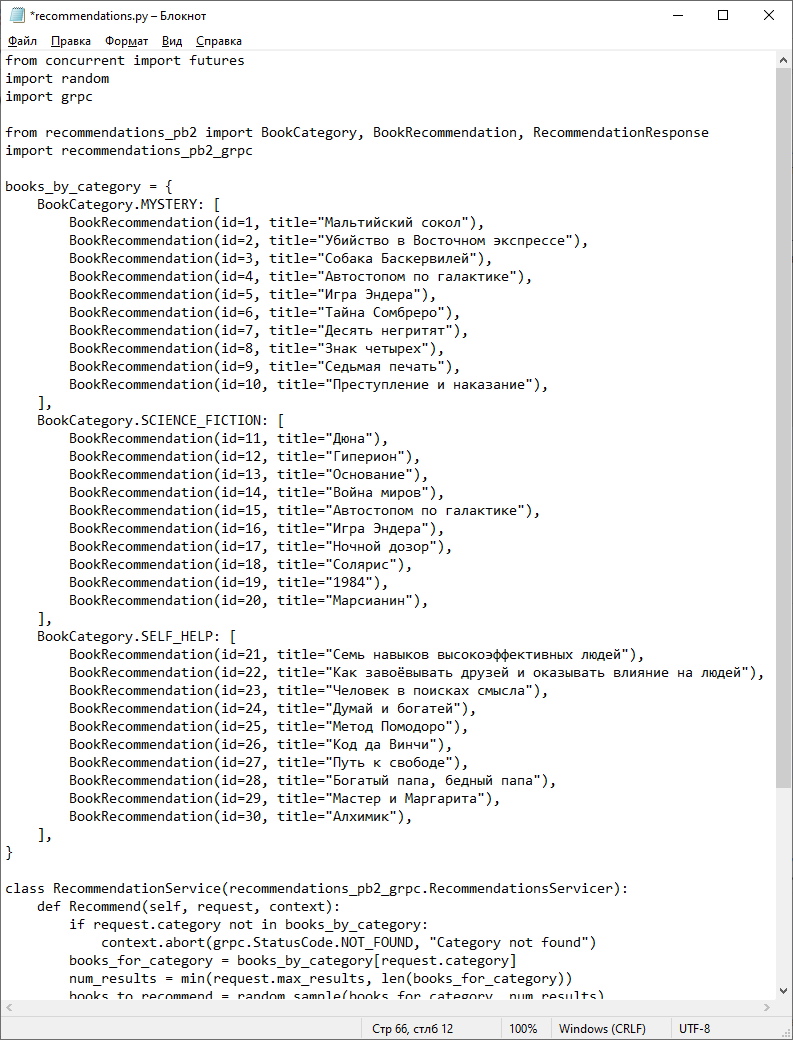


Рис.6 – Содержание файла recommendations.py

Далее, в том же файле создадим класс, реализующий функции микросервиса. Также определим функцию serve() в том же файле: 

Рис.7 – создание класса

Создание микросервиса Marketplace

Создадим новый каталог marketplace/ и создадим в нем файл marketplace.py для микросервиса Marketplace.

Откроем файл marketplace/marketplace.py и добавим следующее:

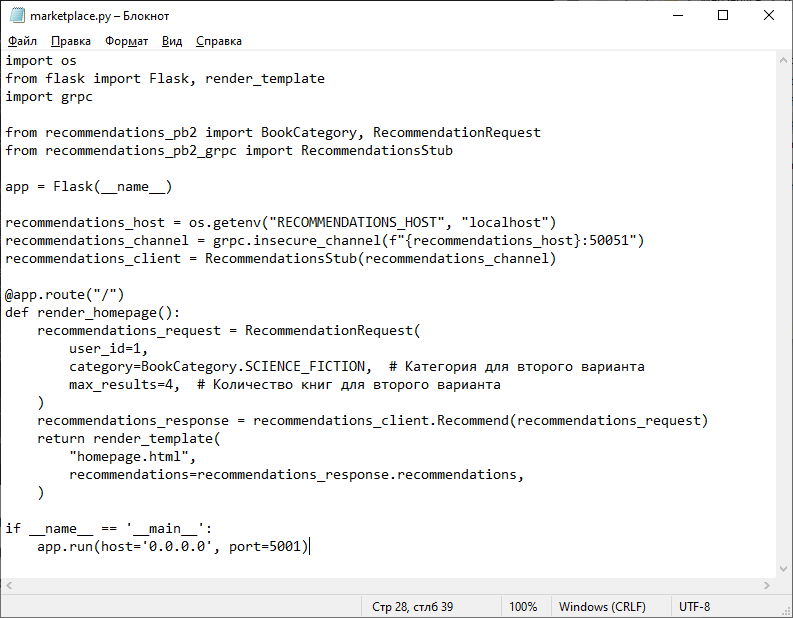


Рис.8 – создание файла Marketplace

Создадим файл homepage.html в каталоге marketplace/templates/ и добавим следующий HTML-код:

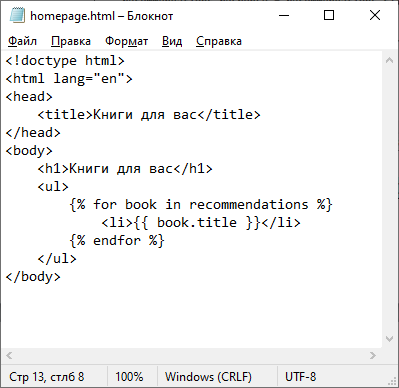


Рис.9 – Cоздание homepage.html

Для запуска кода потребуются следующие зависимости, которые необходимо добавить в marketplace/requirements.txt:

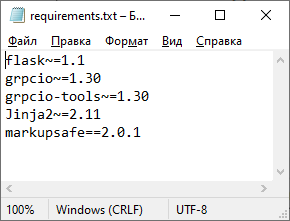


Рис.10 – Добавление зависимостей в marketplace

Теперь, после установки зависимости, нужно также сгенерировать код для protobufs в каталоге marketplace/. Для этого выполним в консоли следующие команды:

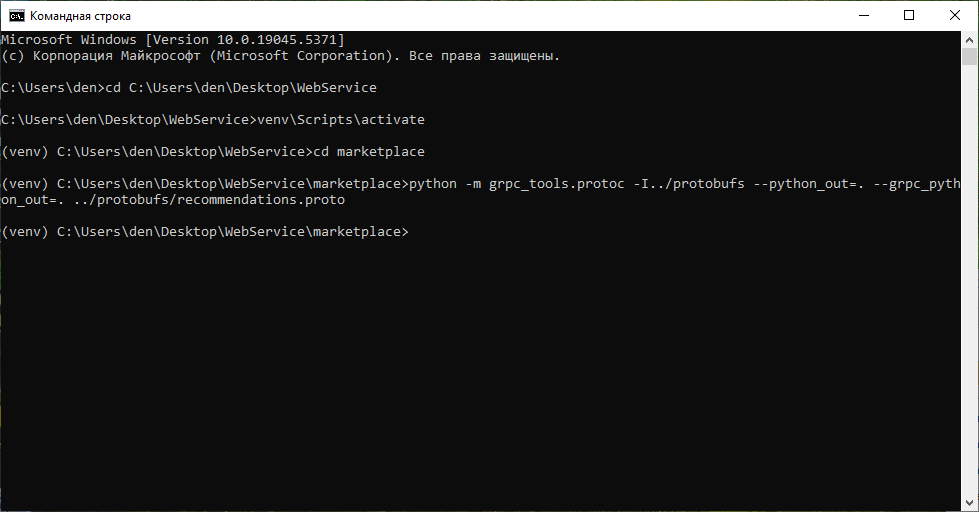


Рис.11 – Генерация кода для protobufs

Упаковка микросервисов Python в Docker.

Создадим файл recommendations/Dockerfile и добавим в него следующее:

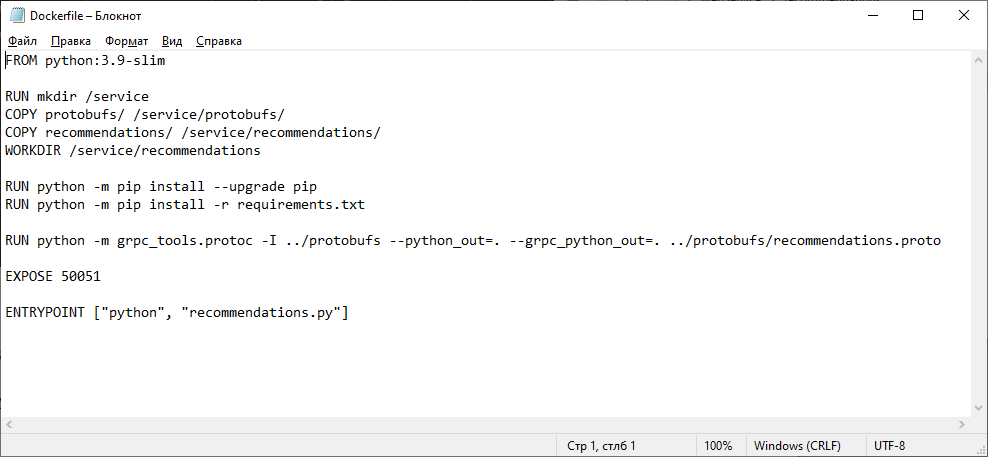


Рис.12 – Создание Dockerfile для recommendations

Чтобы сгенерировать Docker-образ, перейдем в терминале на уровень выше относительно Dockerfile и запустим команду build:

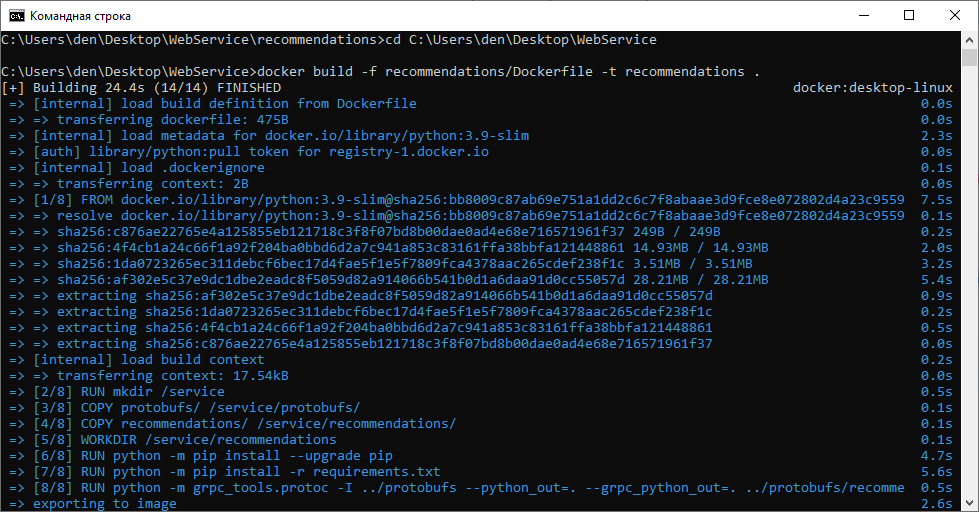


Рис.13 - Генерация Docker-образа

Теперь создадим образ для сервиса Marketplace. Создается аналогичный файл Marketplace/Dockerfile с содержимым:

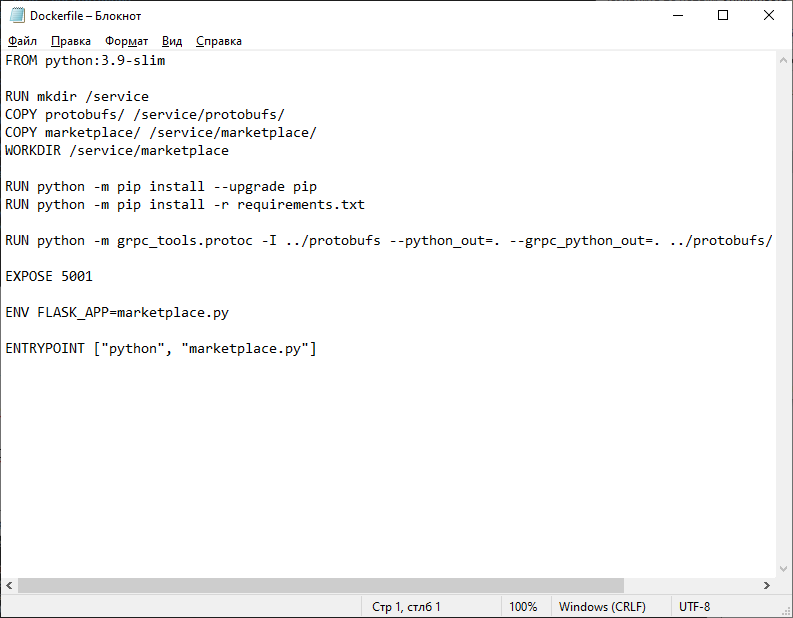


Рис.14 - Создание образа для сервиса Marketplace

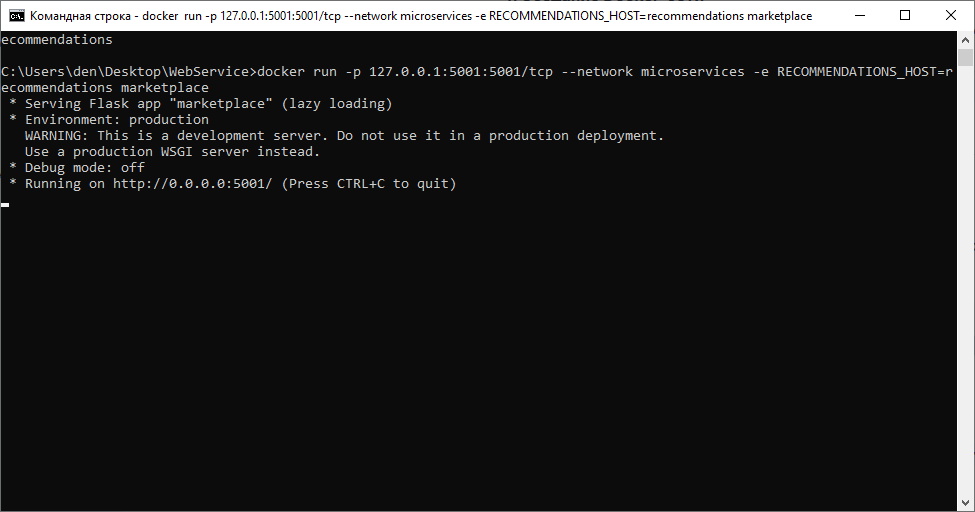
Собираем образ и запускаем контейнер: 

Рис.15 – Запуск контейнера Marketplace

Проверка работы:

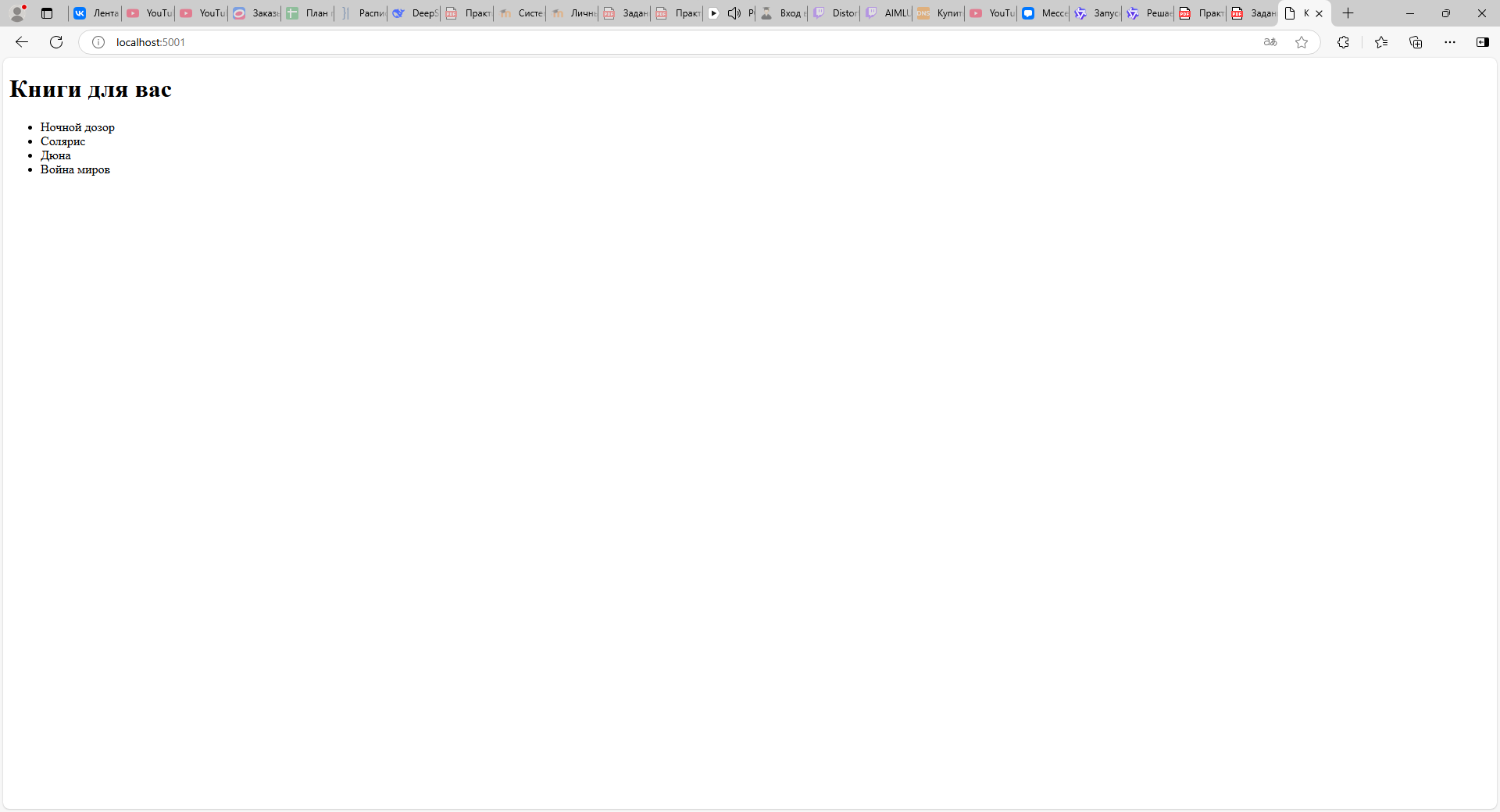


Рис.16 – Проверка работы localhost

Завершение работы.

Для очистки ресурсов сперва необходимо удалить созданные и запущенные контейнеры. Для этого получаем список контейнеров в системе и удаляем контейнеры marketplace и recommendations.

Находим нужные контейнеры и получаем их идентификаторы.

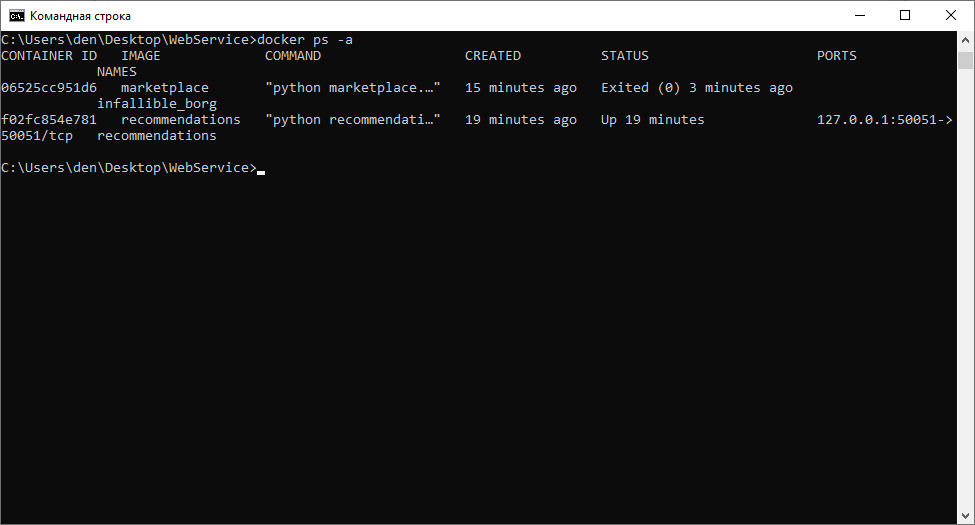


Рис.17 – Нахождение нужных контейнеров

Удаляем оба контейнера по их ID командой docker rm и проверяем на наличие контейнеров



Рис.18 – Проверка на наличие контейнеров

Далее необходимо удалить образы Docker для обоих сервисов. Для этого необходимо получить список образов Docker в системе командой docker images:

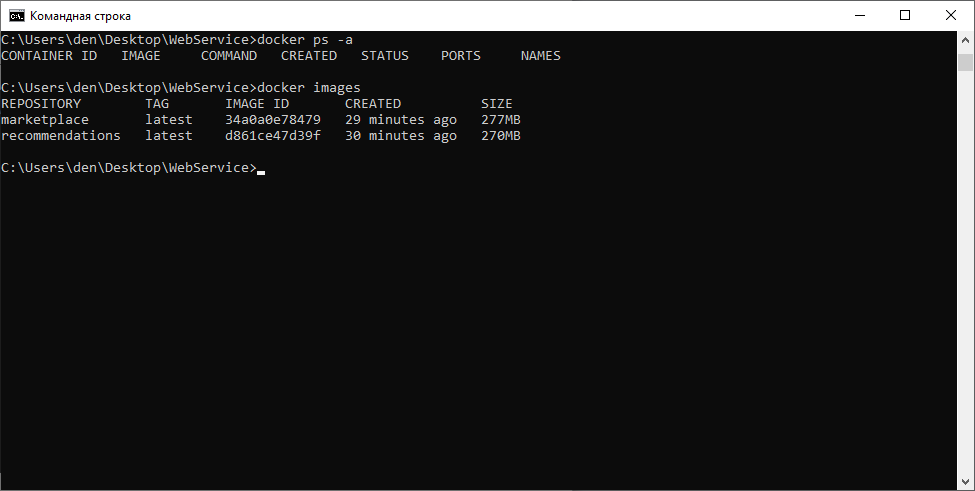


Рис.19 – Отображение образов Docker

Находим образы созданных ранее сервисов и удаляем их по имени командой docker image rm:

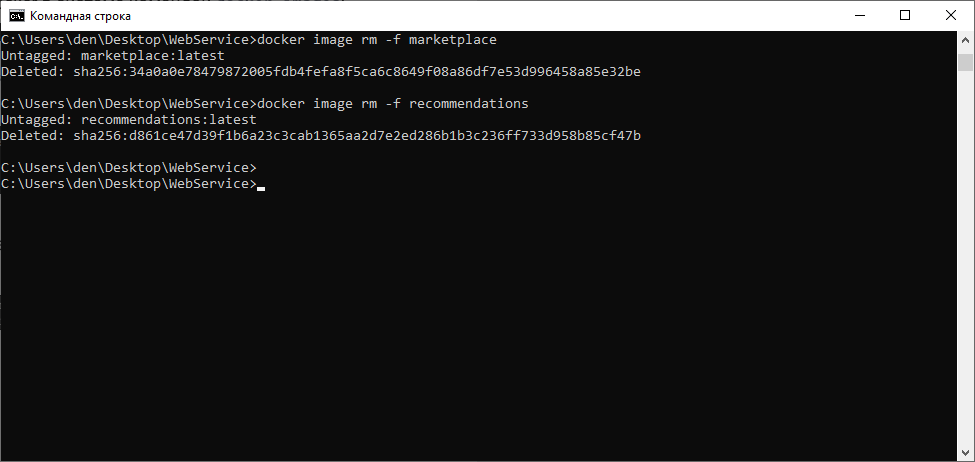


Рис.20 – Удаление образов