|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКИЕ РАБОТЫ №11-12**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы** ИКБО-20-21 Сидоров С.Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** ассистент Благирев М.М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена « » 2023 г.

Допущен к работе « » 2023 г.

Москва 2023

**Содержание**

[Практическая работа №11 3](#_Toc153730187)

[Теоретическое введение. 3](#_Toc153730188)

[Задание 4](#_Toc153730189)

[Выполнение задания 5](#_Toc153730190)

[Выводы 7](#_Toc153730191)

[Практическая работа №12 8](#_Toc153730192)

[Теоретическое 8](#_Toc153730193)

[Задание 10](#_Toc153730194)

[Выполнение задания 11](#_Toc153730195)

[Выводы 18](#_Toc153730196)

[Список использованных источников 19](#_Toc153730197)

# Практическая работа №11

## Теоретическое введение.

Python код может выполняться разными способами. Можно написать файл c расширением .py, который бы содержал некий скриптовый код, далее его можно исполнить командой python, например, мы находимся в некотором каталоге где лежит main.py в котором есть код:

Листинг 1 – код примера

**print("Hello world")**

Далее необходимо запустить код командой:

Листинг 2 – запуск кода на Python

D:\RSCHIR> python main.py

Hello world

PS D:\RSCHIR>

Но такой подход обычно используется простыми программами или скриптами и использует глобальную виртуальную среду интерпретатора Python. В этой среде обычно определены пакеты, получаемые с помощью пакетного менеджера python – pip. pip берет пакеты из собственного удаленного репозитория пакетов [PyPi](https://pypi.org/). Эти пакеты представляют собой zip архивы которые содержат в себе код Python некоторую метаинформацию, определяющую зависимости библиотеки и тому подобные вещи.

Соответственно, пакеты, которые будут стоять глобально, могут быть использованы всеми программами, которые выполняются глобально как было описано выше, но проблема в том, что пакеты постоянно обновляются, выходят новые патчи, исправления и, если устанавливать эти обновления глобально, какие-то из проектов могут перестать корректно запускаться, не только из-за несовместимости версий используемых пакетов в конкретных проектах, но и из-за несовместимой версии самого интерпретатора Python.

Поэтому был придуман механизм изоляции приложений, написанных на языке Python – это создание отдельных виртуальных сред интерпретатора Python venv. Такой подход позволяет использовать только те зависимости, которые нужны проекту. Можно сказать, что это своя папка для каждого проекта на Python. Такой подход является наиболее рекомендуемым.

По большому счету пакеты являются библиотеками для языка Python. А pip это способ доставки этих пакетов, по аналогии с Gradle/Maven для Java/Kotlin или npm для Node.js.

## Задание

* + - 1. Вам предлагается написать простое веб приложение Hello world, которое должно получать имя пользователя и выводить на экран, “Hello, {полученное имя}”. Ваша фамилия обязательно должна фигурировать в пути к приложению (localhost:8080/ivanov). Обязательно обработать случай, когда пользователь не передал имя.

## Выполнение задания

Для создания образа, был использован Dockerfile, изображенный на рисунке 1.

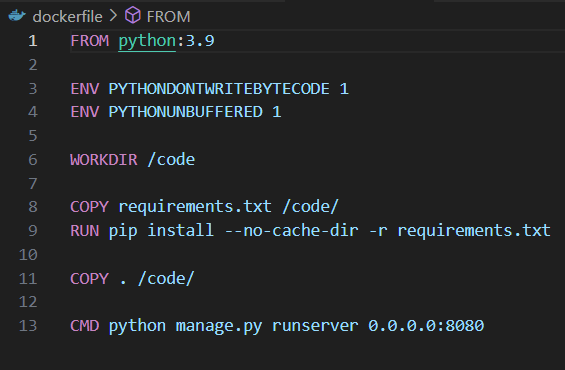


Рисунок 1 – Dockerfile

Содержимое docker-compose.yml показано на рисунке 2.

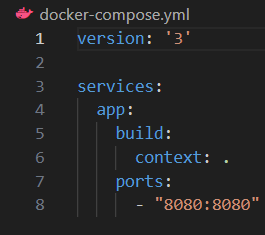


Рисунок 2 – docker-compose.yml

Фрагмент кода urls представлено на рисунке 3.

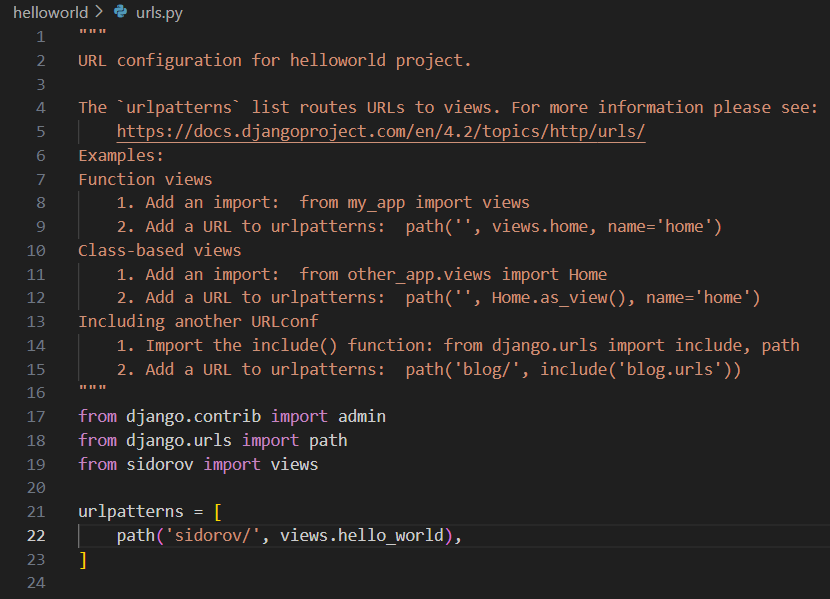


Рисунок 3 – Фрагмент кода urls

Фрагмент кода вывода информации на страницу представлено на рисунке 4.

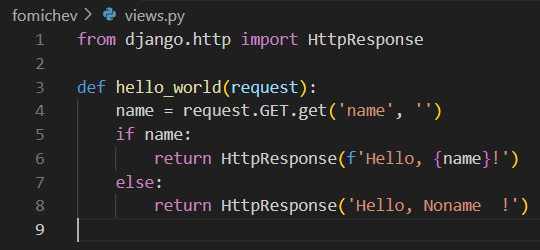


Рисунок 4 – Фрагмент кода views

Результат работы программы представлен на рисунке 5.

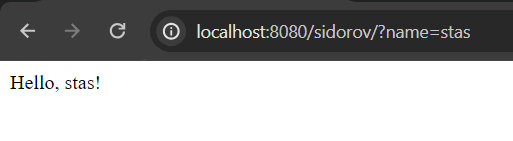


Рисунок 5 – Результат работы программы

## Выводы

Таким образом, было написано простое веб приложение Hello world, которое должно получать имя пользователя и выводить на экран, “Hello, {полученное имя}”.

# Практическая работа №12

Теоретическое **введение.**

Файловая архитектура организации джанго (стандартная) приведена на рисунке 6.

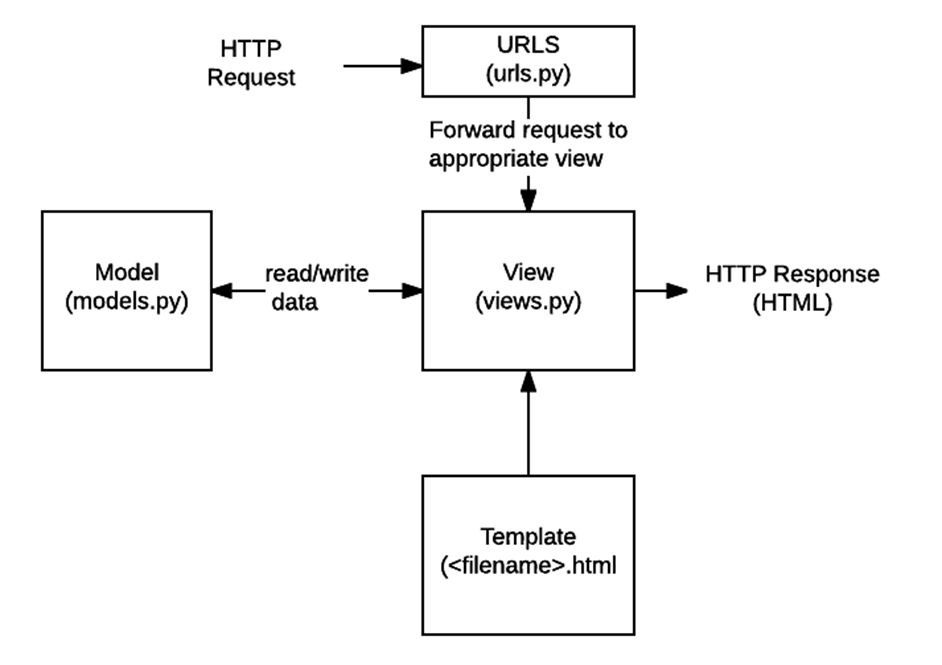


Рисунок 6.  Файловая архитектура организации джанго (стандартная)

Python имеет множество библиотек и фреймворков, созданных специально для анализа данных. NumPy, Pandas, Matplotlib и SciPy предоставляют мощные инструменты для работы с числовыми данными, выполнения операций с массивами, обработки и визуализации данных.

NumPy

NumPy это open-source модуль для python, который предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций. Они объединяются в высокоуровневые пакеты. Они обеспечивают функционал, который можно сравнить с функционалом MatLab. NumPy (Numeric Python) предоставляет базовые методы для манипуляции с большими массивами и матрицами.

SciPy

SciPy (Scientific Python) расширяет функционал numpy огромной коллекцией полезных алгоритмов, таких как минимизация, преобразование Фурье, регрессия, и другие прикладные математические техники.

Matplotlib

Библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой. Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях. Matplotlib написан и поддерживался в основном Джоном Хантером и распространяется на условиях BSD-подобной лицензии.

## Задание

Вам предлагается реализовать простой сервис аналитики, которому бы на вход поступали данные из предыдущих практик (Java и Go), на выходе сервис выдает картинку-диаграмму. Картинки обязательно сохранять в БД(Рекомендуемая – Postgres), и при получении данных которые уже были, использовать график, который уже был сгенерирован.

Необходимо реализовать сервис на основе предыдущей работы с использованием REST.

Сервис обязательно должен работать на Docker контейнере, и написан на Python Django с объединением с предыдущими работами. Приветствуется использование схожих библиотек, описанных в введении для создания картинок и анализа данных.lsZQ

## Выполнение задания

Содержимое Dockerfile и docker-compose изображено на рисунках 7, 8.

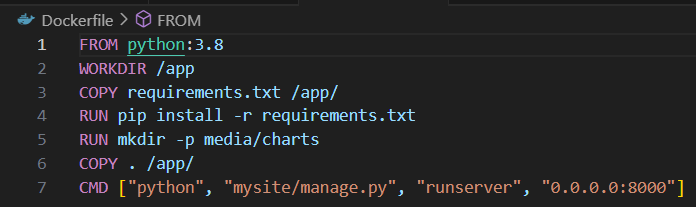


Рисунок 7 – Dockerfile проекта

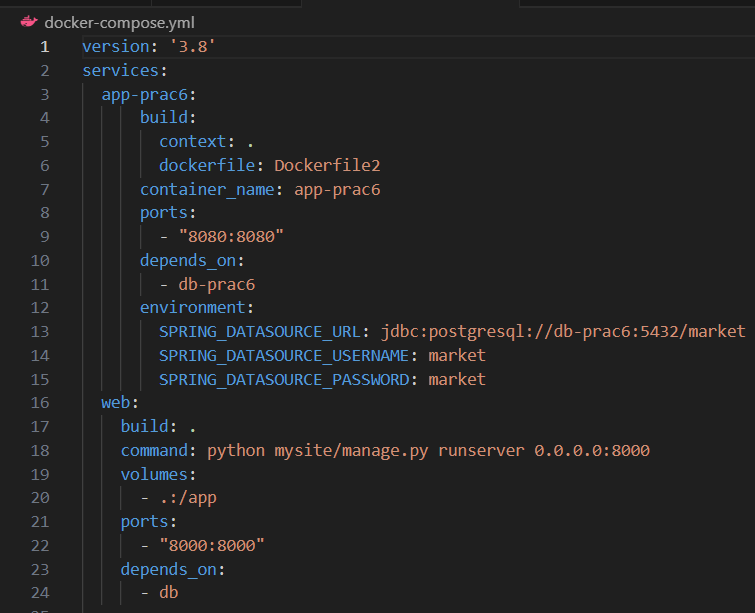


Рисунок 8 – docker-compose проекта

Фрагмент кода manage представлен на рисунке 9.

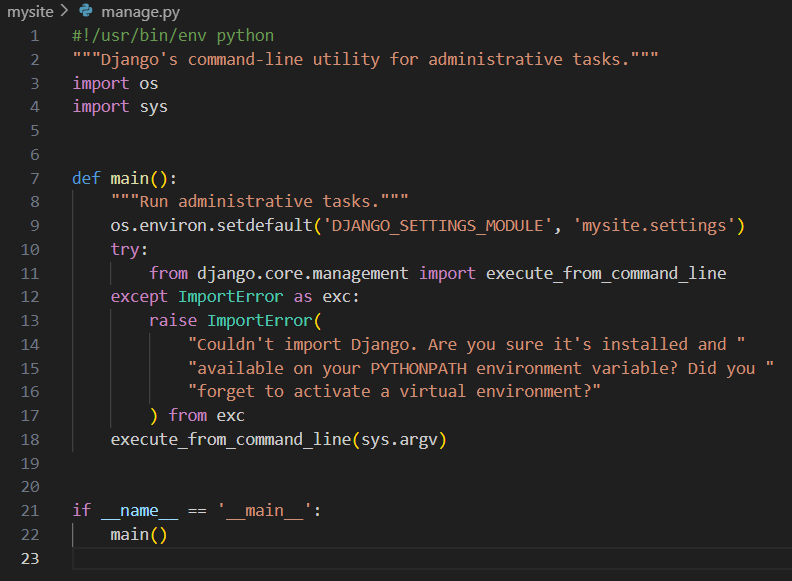


Рисунок 9 – Фрагмент кода manage

Фрагмент кода для отображения диаграмм представлен на рисунке 10 .

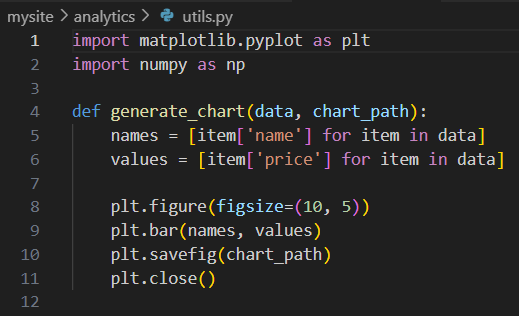


Рисунок 10 – Фрагмент кода для отображения диаграмм

Фрагмент кода для загрузки данных изображен на рисунке 11.

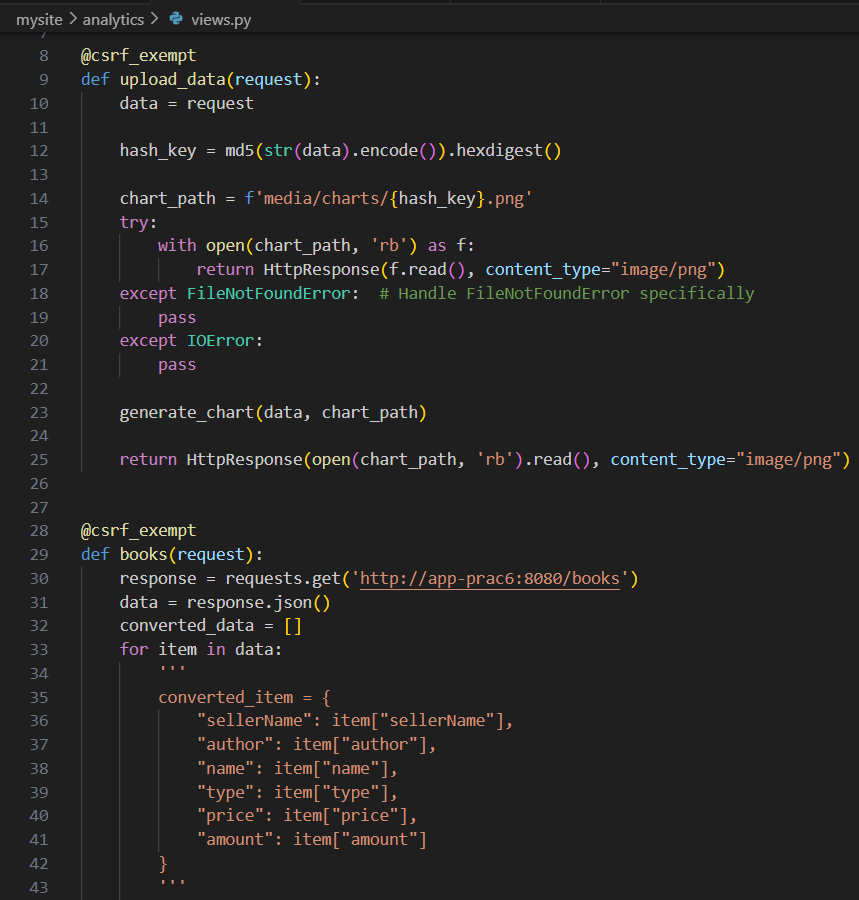


Рисунок 11 – Фрагмент кода обработки данных

Результат работы программы представлен на рисунках 12 – 15.

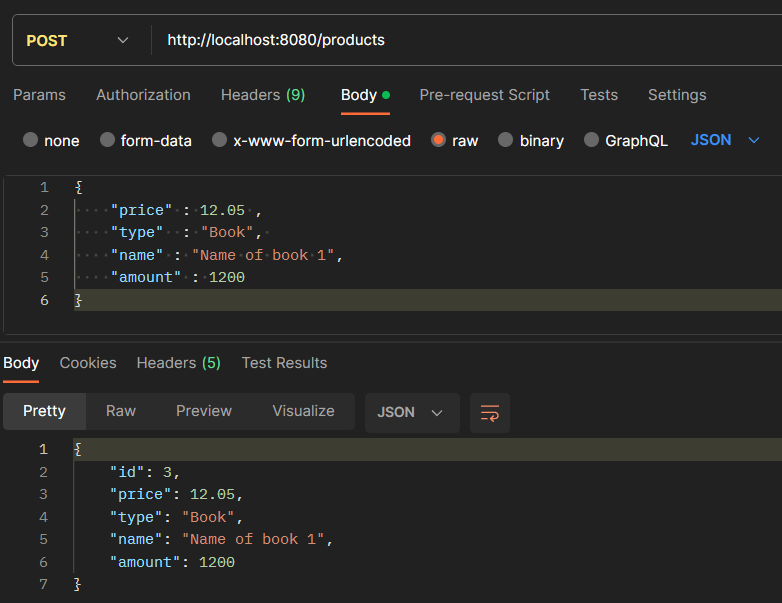


Рисунок 12 – Добавление продукта

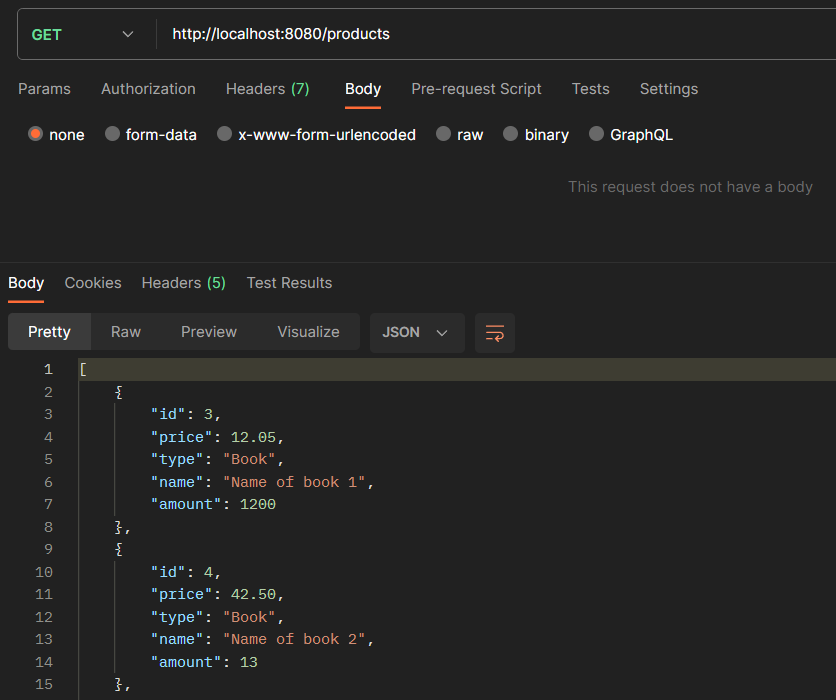


Рисунок 13 – Вывод всех продуктов

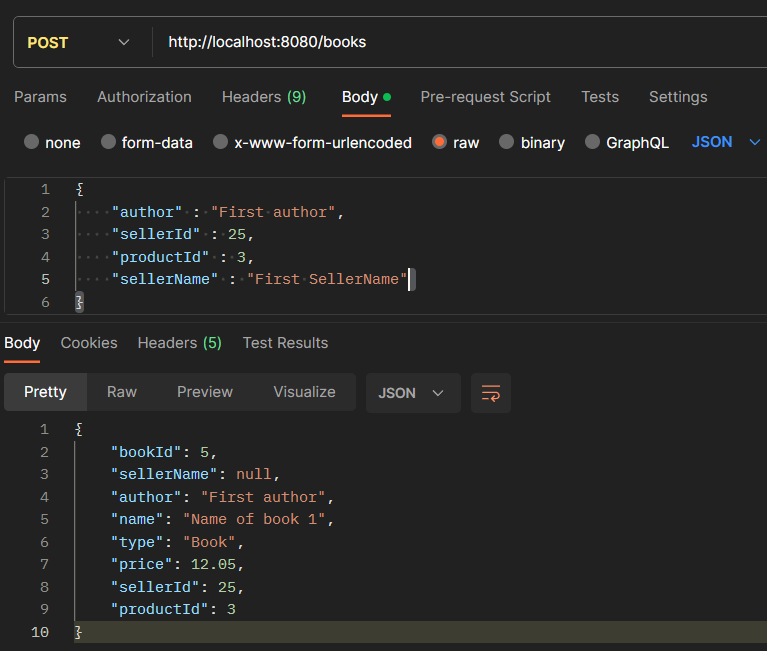


Рисунок 14 – Добавление книги

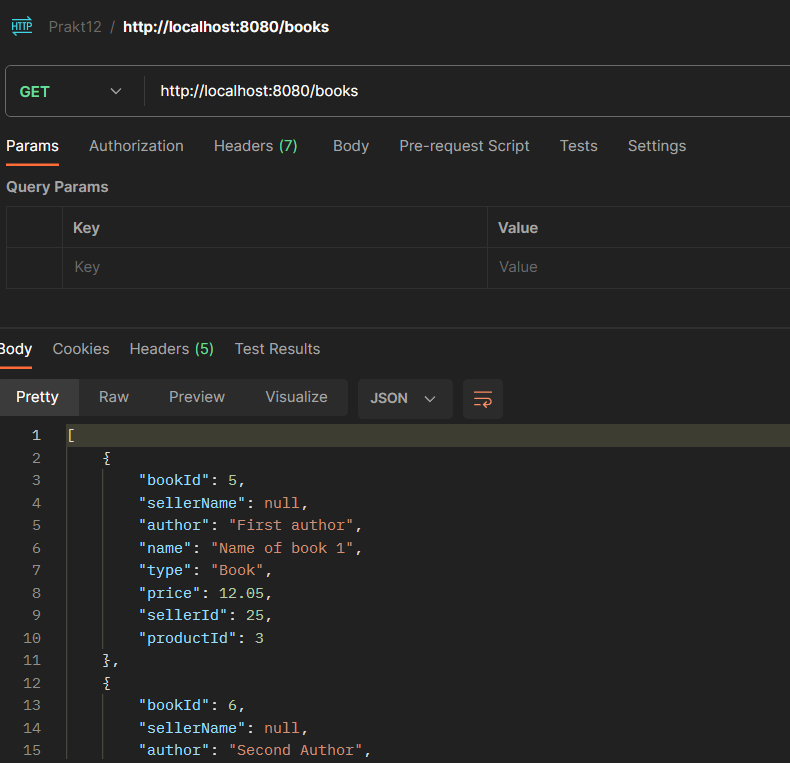


Рисунок 15 – Вывод всех книг

Вывод графика представлен на рисунке 16

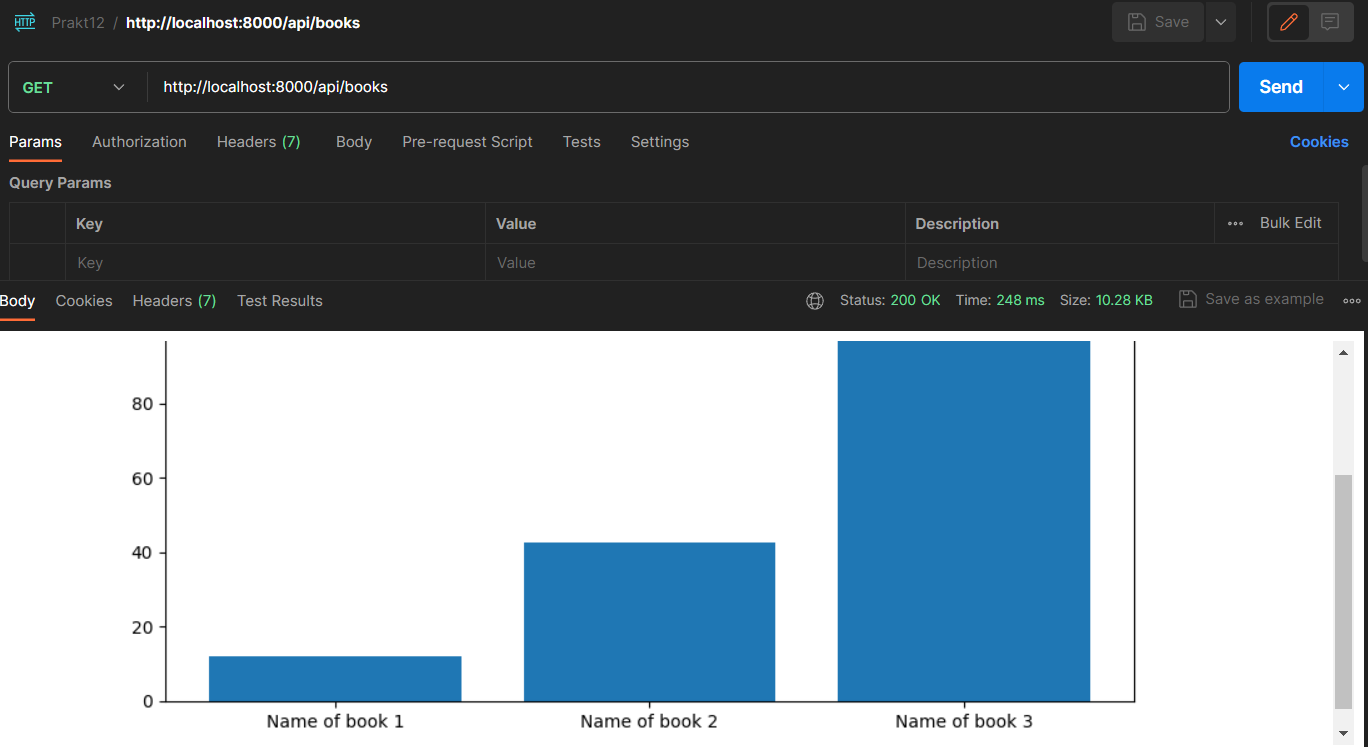


Рисунок 16 – Вывод графика

Изменение данных изображено на рисунке 17

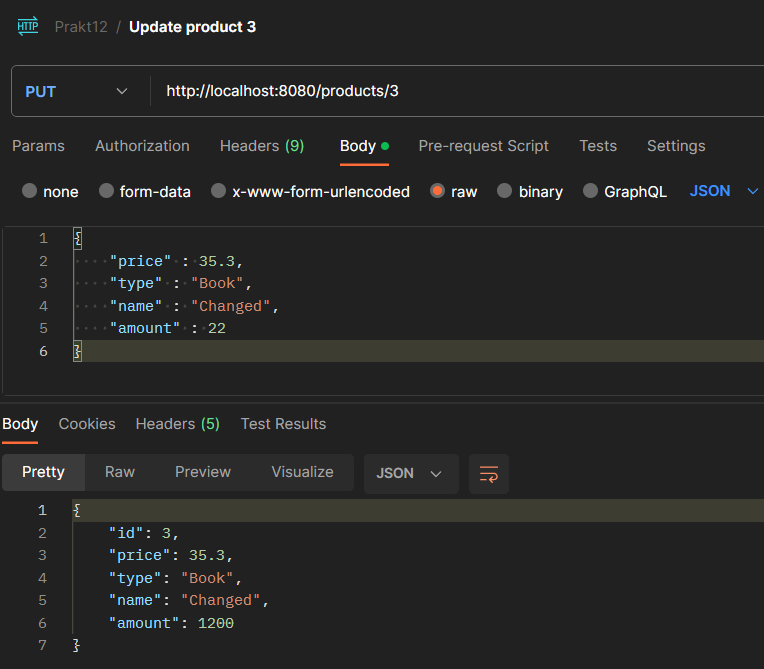


Рисунок 17 – Изменение данных

На рисунке 18 изображен новый график.

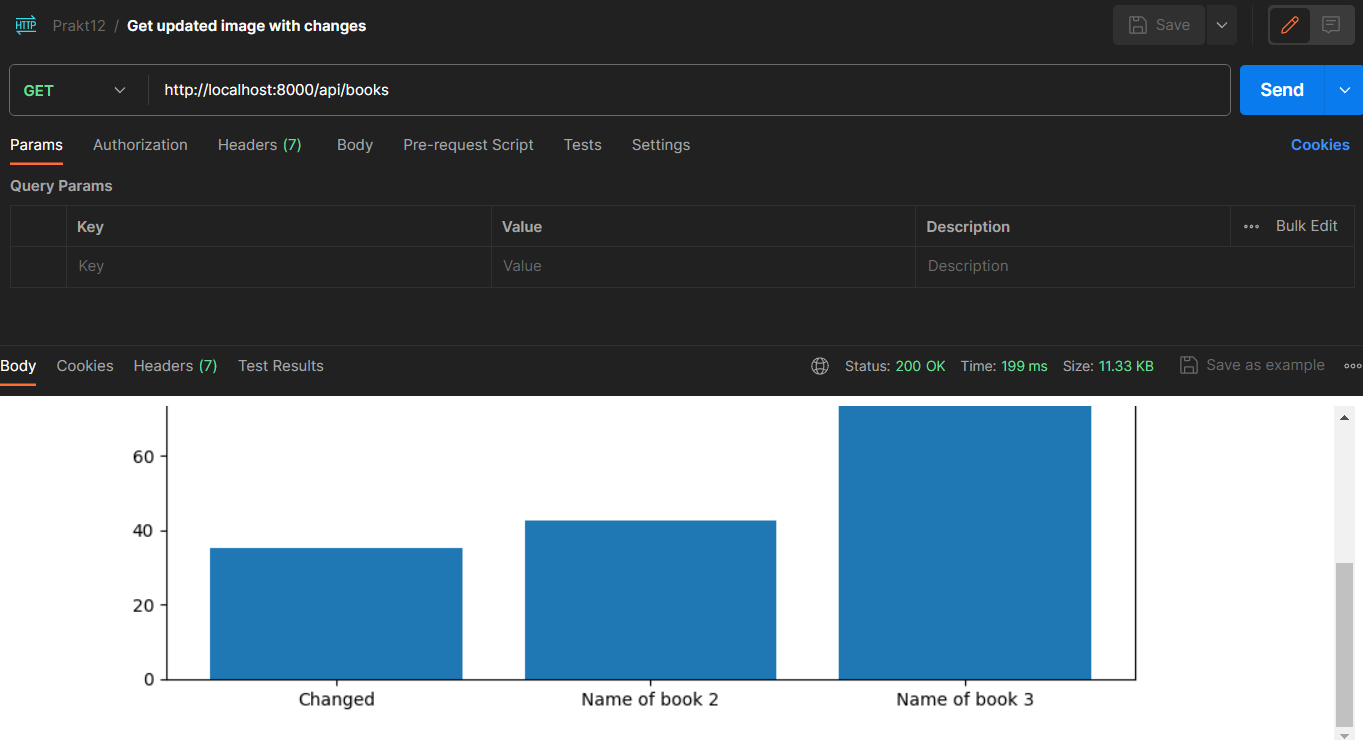


Рисунок 18 – Измененный график

## Выводы

В результате выполнения данной работы был реализован простой сервис аналитики, которому бы на вход поступали данные из предыдущих практик (Java и Go), на выходе сервис выдает картинку-диаграмму.

## Список использованных источников

1. Оф. документация Django - https://docs.djangoproject.com/en/4.2/
2. RESTful на Django - https://habr.com/ru/articles/722054/
3. Views в Django - https://ru.hexlet.io/courses/python-django-basics/lessons/views/theory\_unit
4. Установка Python - https://devpractice.ru/python-lesson-1-install/
5. Docker - https://habr.com/ru/articles/253877/
6. Docker compose - https://habr.com/ru/companies/ruvds/articles/450312/
7. https://habr.com/ru/articles/352678/ - NumPy
8. https://pythonworld.ru/novosti-mira-python/scientific-graphics-in-python.html - Matplotlib
9. https://django.fun/ru/articles/tutorials/sozdajte-rest-api-za-30-minut-s-pomoshyu-django-rest-framework/ - REST на Django
10. https://habr.com/ru/articles/729610/ - Posgres, Django, Docker
11. https://blog.skillfactory.ru/glossary/pandas/ - Pandas