РАБОЧАЯ ДОКУМЕНТАЦИЯ ПО ЗАПУСКУ (Windows OS)

1. Сначала надо убедиться, что версия питона выше, чем 3.9 (иначе match работать не будет);
2. После того, как мы удостоверились, что версия питона выше, чем 3.9 (в идеале 3.11), скачиваем всё из файлика с зависимостями, если на что-то ругается питон, то надо доустановить самостоятельно;
3. Для стабильно работы нам надо поставить виртуальное окружение (.venv), для этого нам надо установить его. Создание виртуального окружения с помощью venv очень просто. Перейдите в директорию вашего проекта и выполните следующую команду для Windows OS:

|  |
| --- |
| python -m venv myenv |

1. Если работа открывается на Windows OS, то надо перейти в папку с проектом, в терминале PyCharm и пишем данную команду:

|  |
| --- |
| .\.venv\Scripts\activate.ps1 |

1. Если работа открывается на Linux, то надо перейти в папку с проектом, в терминале пишем данную команду:

|  |
| --- |
| sudo apt install python3-venv |

1. Перейдите в директорию вашего проекта и выполните следующую команду для Linux:

|  |
| --- |
| source myenv/bin/activate |

1. После мы должны увидеть такую картину

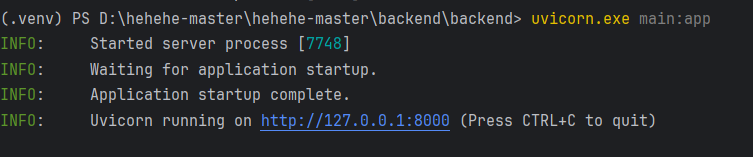


Слева от пути появилась (.venv)

1. Теперь пишем команду активации бэкенда:

|  |
| --- |
| uvicorn.exe main:app |

1. Если запуск был успешным, то увидим такую картину



На этом активация бэкенда Завершена!

Чтобы активировать фронтенд надо в терминале, если Windows OS, то лучше в PowerShell, прописать путь:

|  |
| --- |
| cd D:\hehehe-master\hehehe-master\nginx-1.27.1 |

После пишем просто:

|  |
| --- |
| .\nginx.exe |

Всё, фронтенд теперь тоже активен, хорошей работы!

Также у Вас есть файлик с базой данных, её надо импортировать.

**Импорт**

Чтобы импортировать БД, создайте в оболочке MySQL новую базу данных, которая будет служить в качестве места назначения импорта:

CREATE DATABASE newdatabase;

Затем войдите в оболочку MySQL и введите следующую команду в командную строку:

mysql -u [имя\_пользователя] -p newdatabase < [имя\_бд].sql

Данная команда импортирует выбранную базу данных в созданную БД в MySQL.

Пароль от базы данных:

|  |
| --- |
| admin |

Имя пользователя:

|  |
| --- |
| root |

Имя базы данных:

|  |
| --- |
| restaurant1.sql |

РАБОЧАЯ ДОКУМЕНТАЦИЯ ПО ЗАПУСКУ (Linux OS, желательно Debian)

1) Извлечь папку hack из архиве в / (путь к папке должен выглядеть следующим образом /hack ) и перейти в неё

2) Установить [Docker.io](https://docker.io/) и Docker-compose

|  |
| --- |
| apt install docker.io docker-compose |

3) Выполнить сборку:

|  |
| --- |
| docker-compose build |

4) Запустить контейнеры:

|  |
| --- |
| docker-compose up -d |

ВАЖНО! Frontend работает на порте 8080

ПОДСКАЗКИ И ВЕКТОРЫ АТАКИ

1. При открытии сайта мы видим страничку с меню, списком команды и т.д. Первое, что нужно сделать – исследовать сайт в целом
2. После исследования замечаем следующее:
   1. Имеется несколько форм регистрации (вход в лк и панель администратора)
   2. Имеется несколько полей для ввода данных и отправки их на сервер
3. Для входа в личный кабинет сотрудников можно сделать следующее:
   1. Sql-инъекции в бд с разведкой её структуры и получением таблиц и их полей
   2. Чуть более детальное исследование страниц в меню блюд:
      1. Имеется единственное название блюда на английском
      2. При входе в его описание видим много текста, среди которого подчёркнутое слово «флаг» и единственное в тексте слово латиницей – Aurora. Это и есть наш логин в ЛК с флагом. Пароль подбирается брутфорсом.
4. Для входа в панель администратора используется классический брутфорс с логином admin (через dirsearch подглядеть не выйдет).
5. Третий флаг спрятан с использованием нативной уязвимости языка php – eval(). В поле «Адрес доставки» в сервисе доставки можно спокойно использовать reverse-shell код и с его помощью получить полный доступ к серверу.