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

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практической работе №2**

по дисциплине

«**Технология разработки программных приложений**»

**Тема: «Основы работы с Bash Scriptами»**

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКБО-36-22 | Утенков Юрий Юрьевич |
| **Проверил:** | ассистент Петрова А.А. |

МОСКВА 2024 г.

**О Г Л А В Л Е Н И Е**

[Практическая работа № 2 3](#_Toc159705468)

[Выполнение работы 3](#_Toc159705469)

[Часть 1. 5](#_Toc159705470)

[Часть 2. 8](#_Toc159705471)

[Выводы 15](#_Toc159705472)

# Практическая работа № 2

Выполнение работы

**Задание:**

**Часть 1. Написание простейших сценариев**

1. Напишите сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл.2. Напишите сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует.

3. Напишите сценарий, который с помощью цикла прочитает файл и выведет его содержимое.

4. Напишите сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория.

5. Напишите сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе.

6. Напишите сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение.11. Отмените один из коммитов в локальном репозитории.

**Часть 2. Развертка и запуск проекта при помощи Bash Script**

1. Определение зависимостей проекта

Любой проект зависит от ряда библиотек, которые предоставляют тот или иной функционал. Для развертывания приложения необходимо, чтобы данные библиотеки были установлены в том окружении, где предполагается это самое развертывание. На основании этого для начала необходимо определить, какие зависимости имеет проект. По ссылке <https://www.dropbox.com/s/ija7ax3sj6ysb0p/blocknote-master.tar.gz> расположен проект для скачивания. Будет скачан архив с непонятным названием, распаковать его можно при помощи команды tar -xvf имя\_архива имя\_директории\_для\_распаковки. Проект написан на языке программирования Python. Необходимо составить список зависимостей проекта в виде requirements.txt файла. Данный файл содержит в себе список библиотек, которые необходимо установить в окружение для запуска приложения. Подробнее про составление данного файла можно почитать по ссылке <https://semakin.dev/2020/04/requirements_txt/>. Зависимости в Python можно определить по import’ам в файлах, однако некоторые библиотеки включены в стандартную библиотеку языка, поэтому также необходимо будет определить, является ли библиотека внешней или же встроенной в язык.

2. Создание виртуального окружения

Python позволяет создавать так называемое виртуальное окружение. Данное окружение представляет из себя отдельную копию Python с собственным набором библиотек. Оно позволяет работать с проектами, не загрязняя основной интерпретатор ненужными глобально, то есть для всей системы, библиотеками. Подробнее про создание такого рода окружений можно прочитать по ссылке [https://ru.hexlet.io/courses/python-setup- environment](https://ru.hexlet.io/courses/python-setup-%20environment). Необходимо на основании составленного в прошлом шаге списка команд написать скрипт скачивания указанного в прошлом шаге проекта с последующим созданием виртуального окружения и настройкой его под проект, то есть установкой всех необходимых библиотек.

3. Написание скрипта запуска приложения на новой системе

Bash-скрипты позволяют создать с нуля всё необходимое окружение в системе, начиная с установки самого python-a и всего необходимого ПО для запуска приложения и заканчивая запуском самого приложения. Для начала необходимо установить python 3. Сделать это можно при помощи команды sudo apt install python3. Далее необходимо загрузить к себе на машину собственный проект. После этого необходимо воссоздать полученное на прошлом этапе виртуальное окружение со всеми зависимости. Затем необходимо запустить проект из виртуального окружения при помощи следующих команд:

* python manage.py makemigrations
* python manage.py migrate
* python manage.py runserv

**Ход выполнения работы:**

Часть 1.

Подготовка к работе с Bash (рисунок 1).

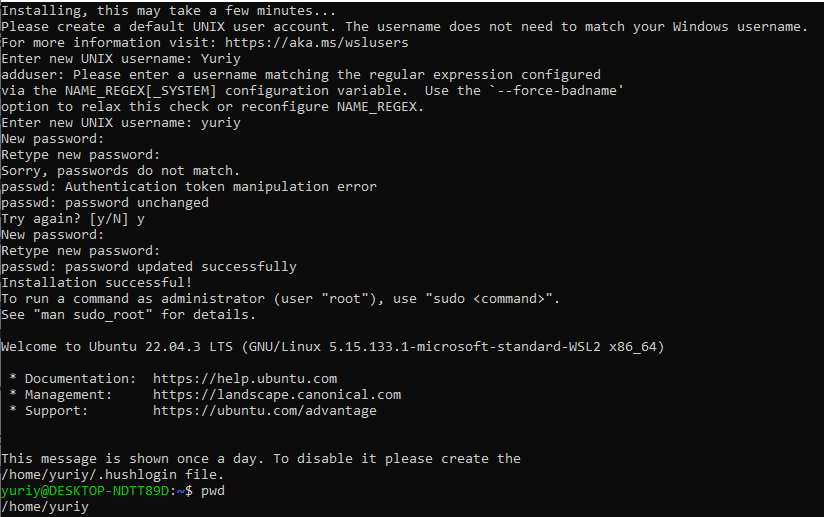


Рисунок 1 – Подготовка к работе с Bash

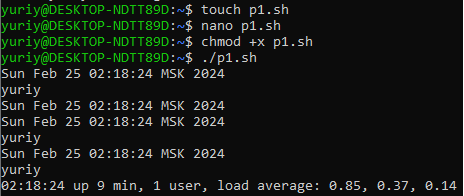


Рисунок 2 – Подготовка для запуска сценария bash и вывод

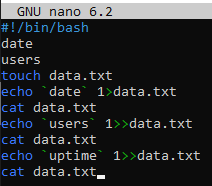


Рисунок 3 – Сценарий, выводящий дату, время, список зарегистрированных пользователей и uptime системы с сохранением в файл

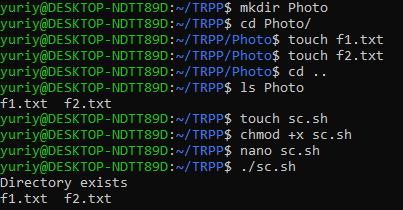


Рисунок 4 – Подготовка для запуска сценария bash и вывод

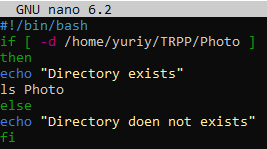


Рисунок 5 – Сценарий, выводящий содержимое каталога или сообщение о том, что его нет

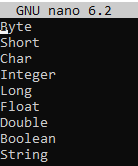


Рисунок 6 – Файл с данными о типах переменных

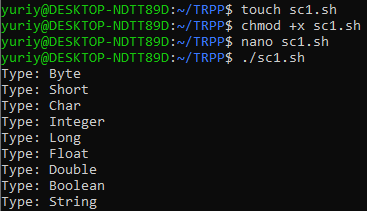


Рисунок 7 – Подготовка для запуска сценария bash и вывод

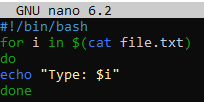


Рисунок 8 – Печать содержимого файла с помощью цикла for

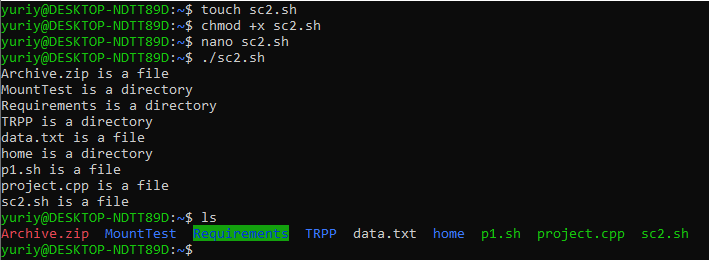


Рисунок 9 – Подготовка для запуска сценария bash и проверка корректности вывода

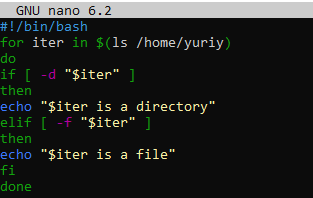


Рисунок 10 – Проход через цикл по каталогу и вывод информации, является ли элемент директорией или файлом

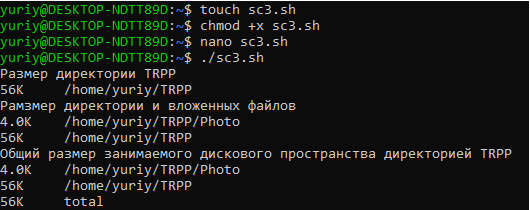


Рисунок 11 – Вывод информации, сколько занято данной директорией памяти

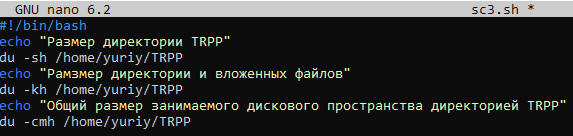


Рисунок 12 – Скрипт, выводящий информацию, сколько занятно данной директорией памяти

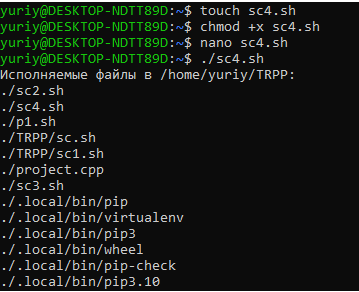


Рисунок 13 – Подготовка для запуска сценария bash и проверка корректности вывода

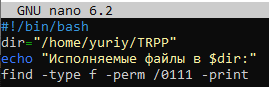


Рисунок 14 – Скрипт, выводящий список всех исполняемых файлов в директории TRPP

Часть 2.

Чтобы дальше работать с python в ubuntu необходимо установить команды pip и pipreqs. Команды pip и pipreqs позволяют записывать зависимости файлов как для всей директории, так и для конкретных файлов. Чтобы установить данные команды, необходимо выдать полные полномочия для определённой папки, в которой будет располагаться проект python.



Рисунок 15 – Выдача полных полномочий для данной папки

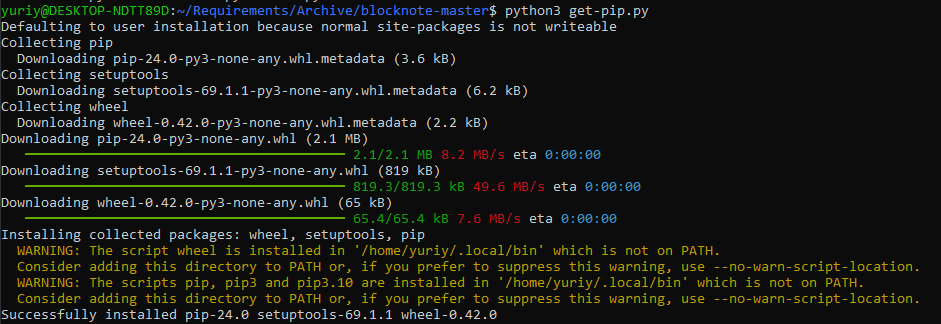


Рисунок 16 – Установка команды pip

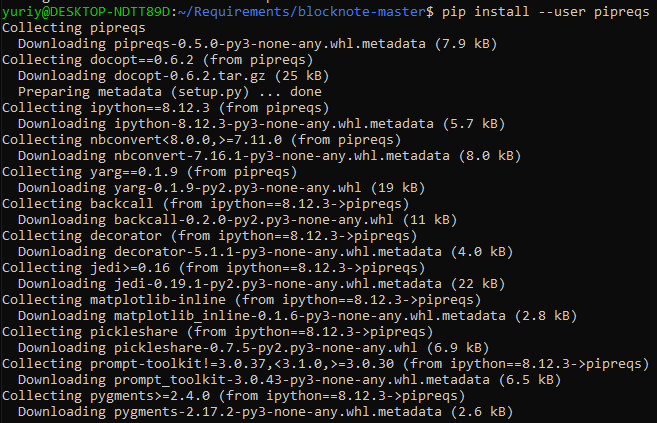


Рисунок 17 – Установка команды pipreqs

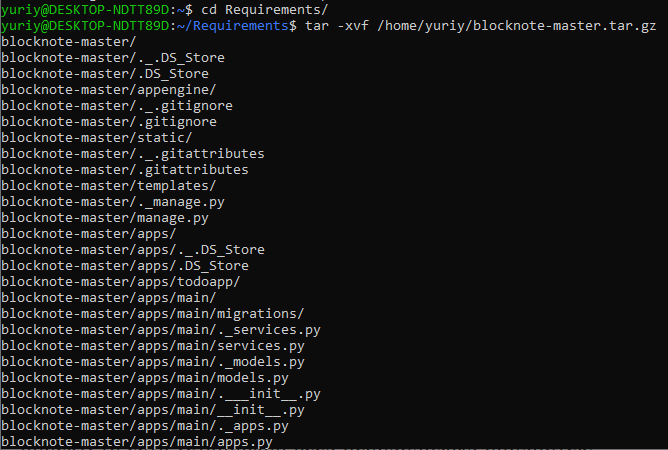


Рисунок 18 – Распаковка данного архива с помощью команды tar

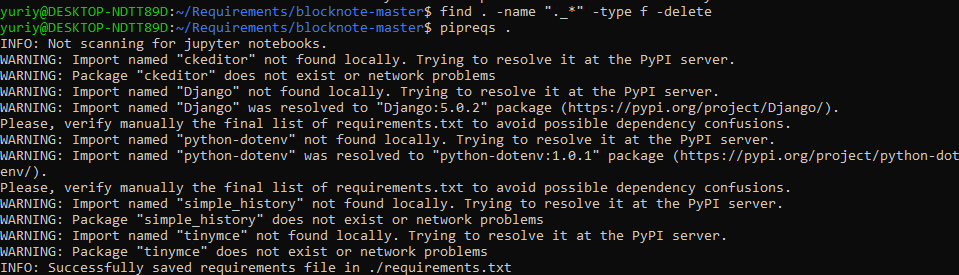


Рисунок 19 – Удаление файлов байткода python и запись зависимостей проекта в файл

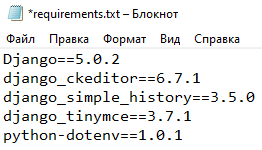


Рисунок 20 – Зависимости данного проекта в файле

Чтобы работать дальше, необходимо создать виртуальное окружение для python, необходимо установить команду venv, которая позволит создать отдельный интерпретатор python’а для разрешения конфликтов, связанных с одинаковым названием переменных, функций, методов и т.д.

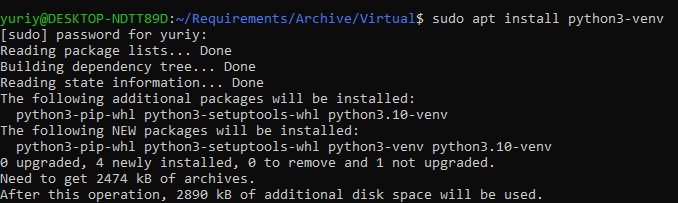


Рисунок 21 – Установка команды venv

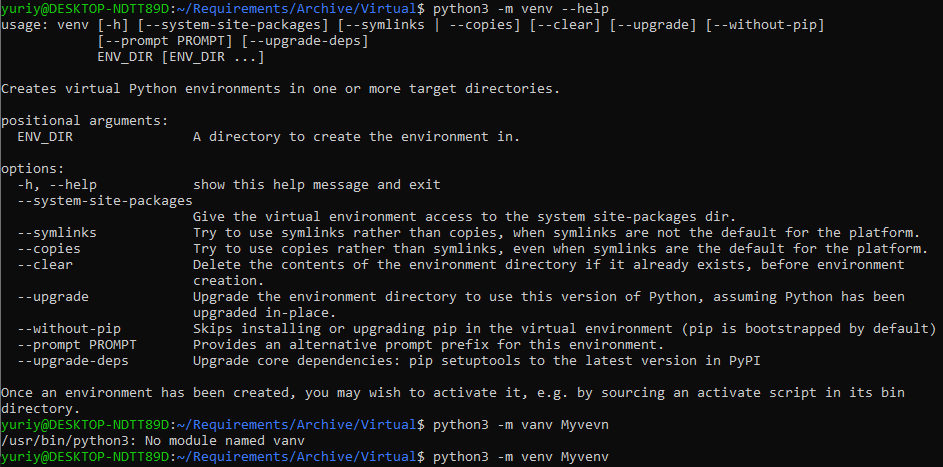


Рисунок 22 – Проверка работоспособности команды venv и создание виртуального окружения Myvenv

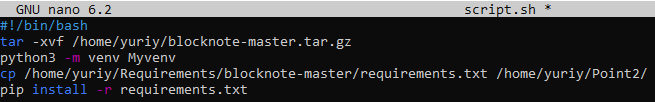


Рисунок 23 – Скрипт, распаковывающий архив, создающий виртуальное окружение и загружающий все необходимые для проекта зависимости

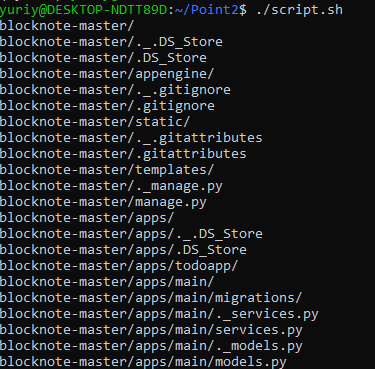


Рисунок 24 – Запуск скрипта

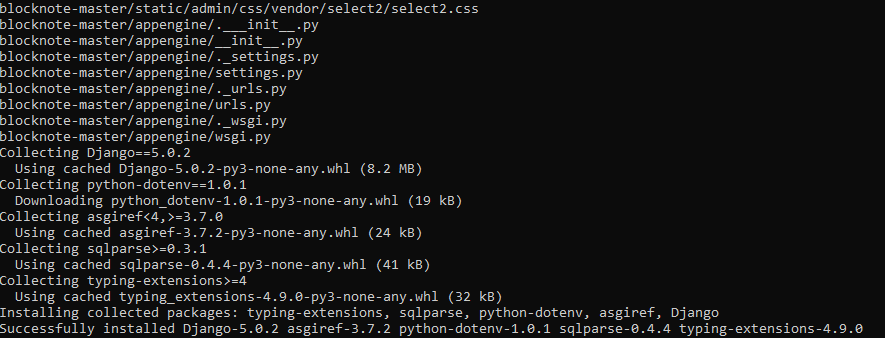


Рисунок 25 – Результат выполнения скрипта

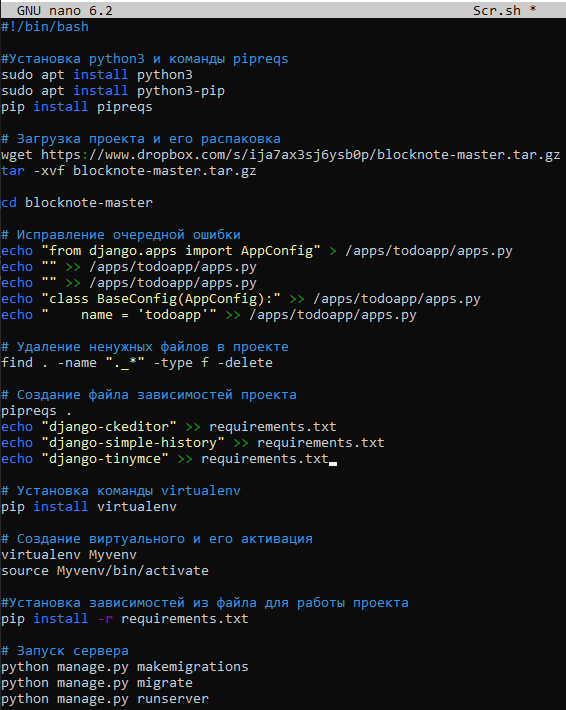


Рисунок 26 – Скрипт запуска приложения

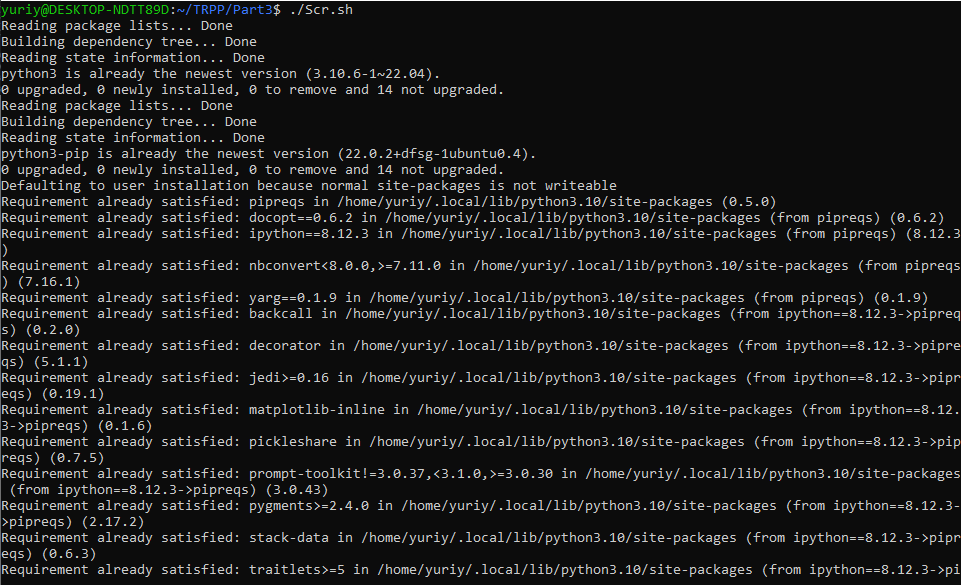


Рисунок 27 – Запуск данного скрипта

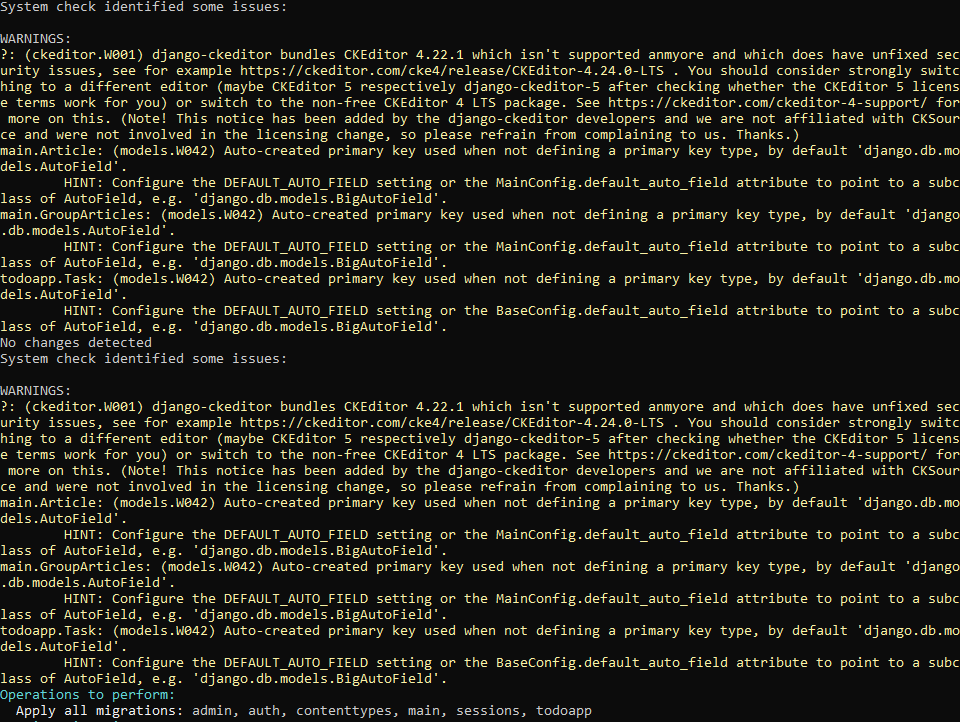


Рисунок 28 – Запуск сервера

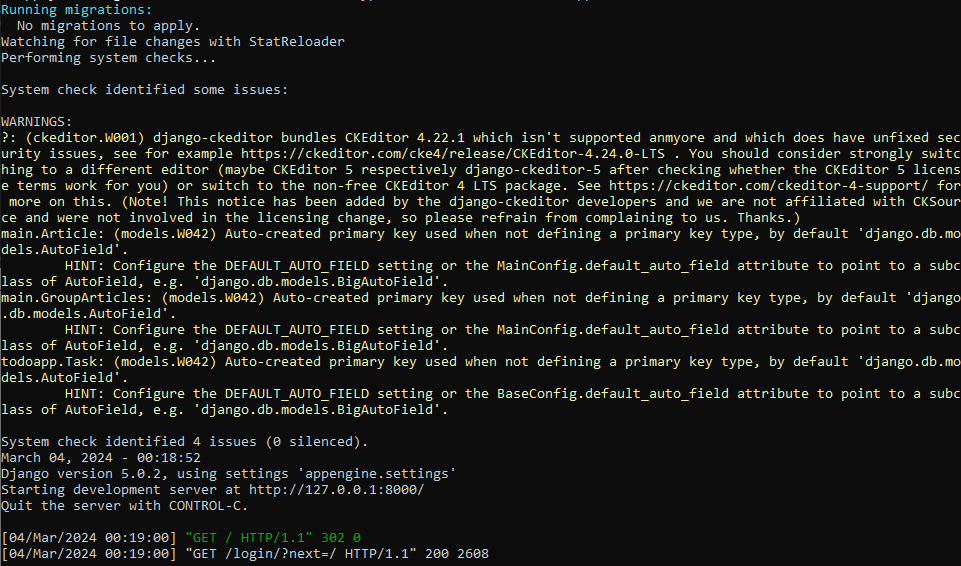


Рисунок 29 – Запуск сервера

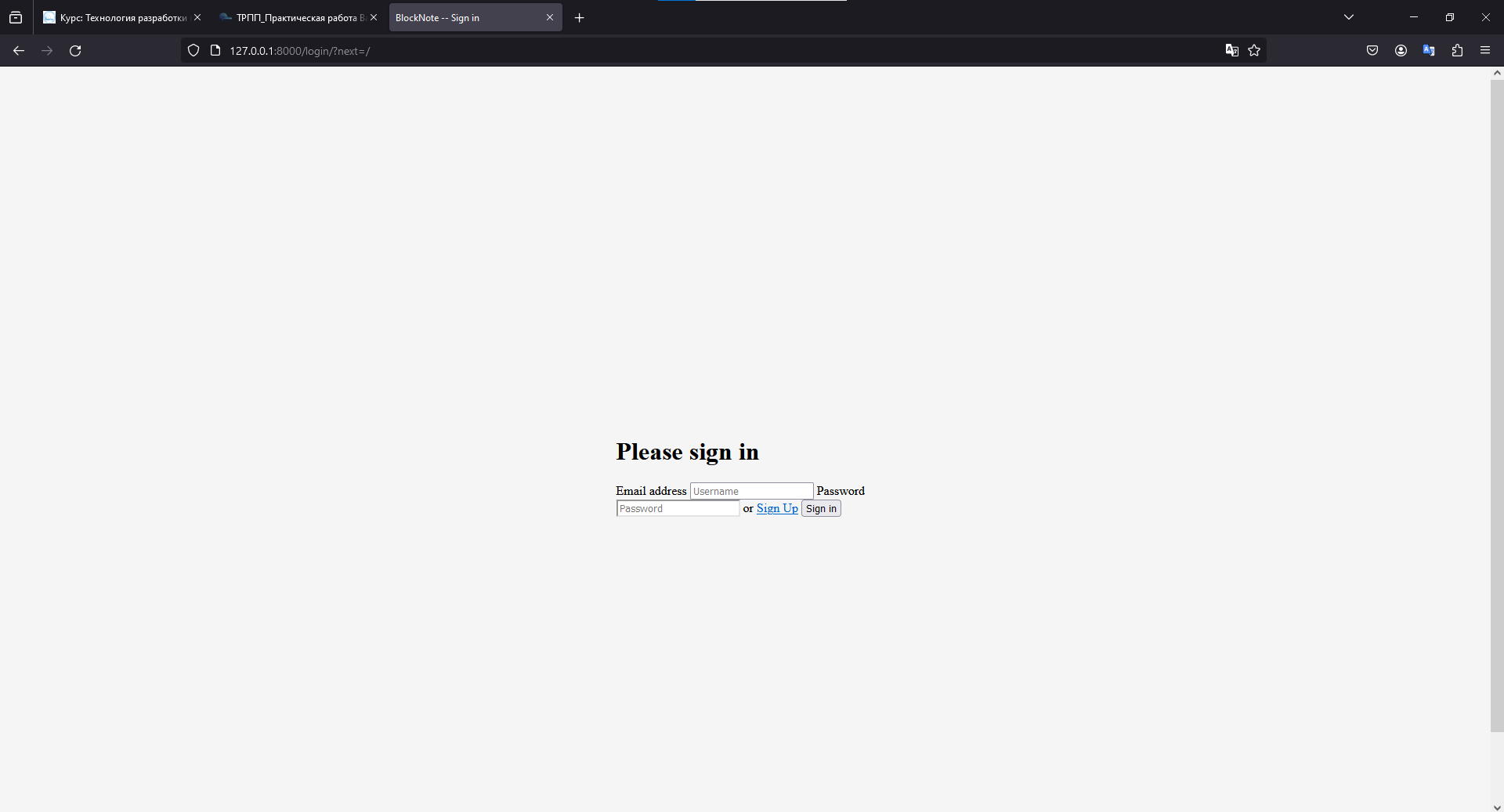


Рисунок 30 – Результат запуска сервера

# Выводы

В ходе выполнения данной работы я расширил свои знания в области написания скриптов bash, установки с помощью него зависимостей, проектов, файлов. Научился работать с зависимостями различных проектов, отслеживать встроенные и внешние зависимости, устанавливать их для корректной работы проекта.