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

**Институт информационных технологий (ИИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИППО)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Технология разработки программных приложений»

**Практическое задание № 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | ИКБО-16-22 Трусов Е.С. | (подпись) | |
| Ассистент | Герасимов С.С. | (подпись) | |
| Отчет представлен | «\_\_\_»\_\_\_\_\_\_\_\_2023г. | |  | |

Москва 2023 г.

Задание

**Часть 1**

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.

**Часть 2**

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.

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 всѐ 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 runserver

Выполнение заданий

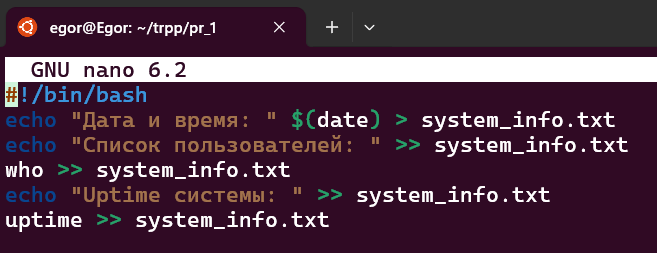


Рисунок 1 – Скрипт для вывода даты, времени, имени пользователя и uptime

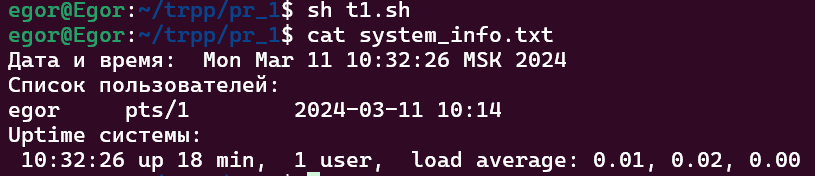


Рисунок 2 – Результат выполнения первого скрипта

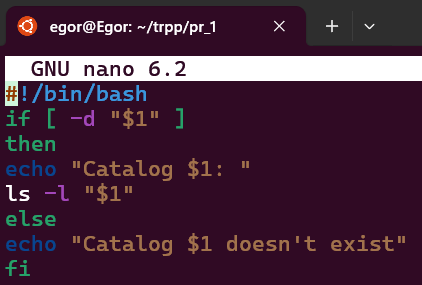


Рисунок 3 – Скрипт для вывода содержимого каталога

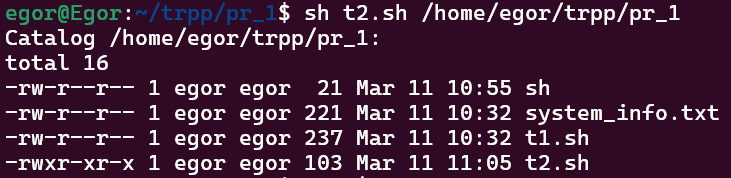


Рисунок 4 – Результат выполнения второго скрипта

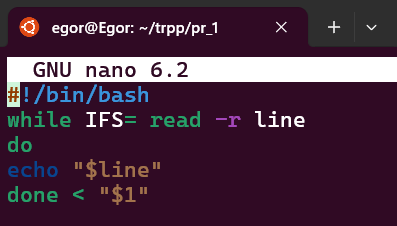


Рисунок 5 – Скрипт для чтения и вывода содержимого файла

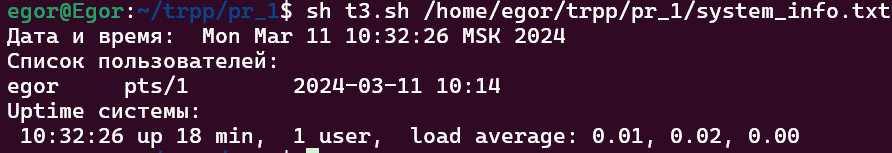


Рисунок 6 – Результат выполнения третьего скрипта



Рисунок 7 – Скрипт для вывода списка файлов и директорий с указанием типа

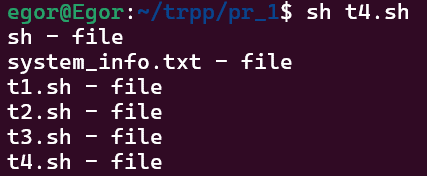


Рисунок 8 – Результат выполнения четвёртого скрипта

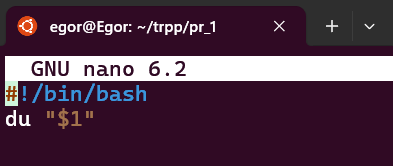


Рисунок 9 – Скрипт для полсчёта объема диска



Рисунок 10 – Результат выполнения пятого скрипта

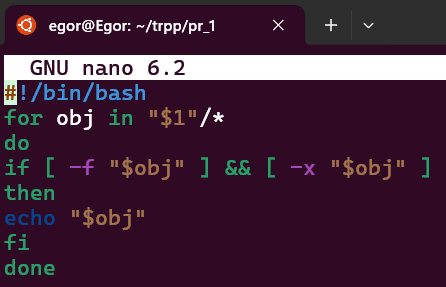


Рисунок 11 – Скрипт для вывода списка исполняемых файлов



Рисунок 12 – Результат выполнения шестого скрипта

Часть 2

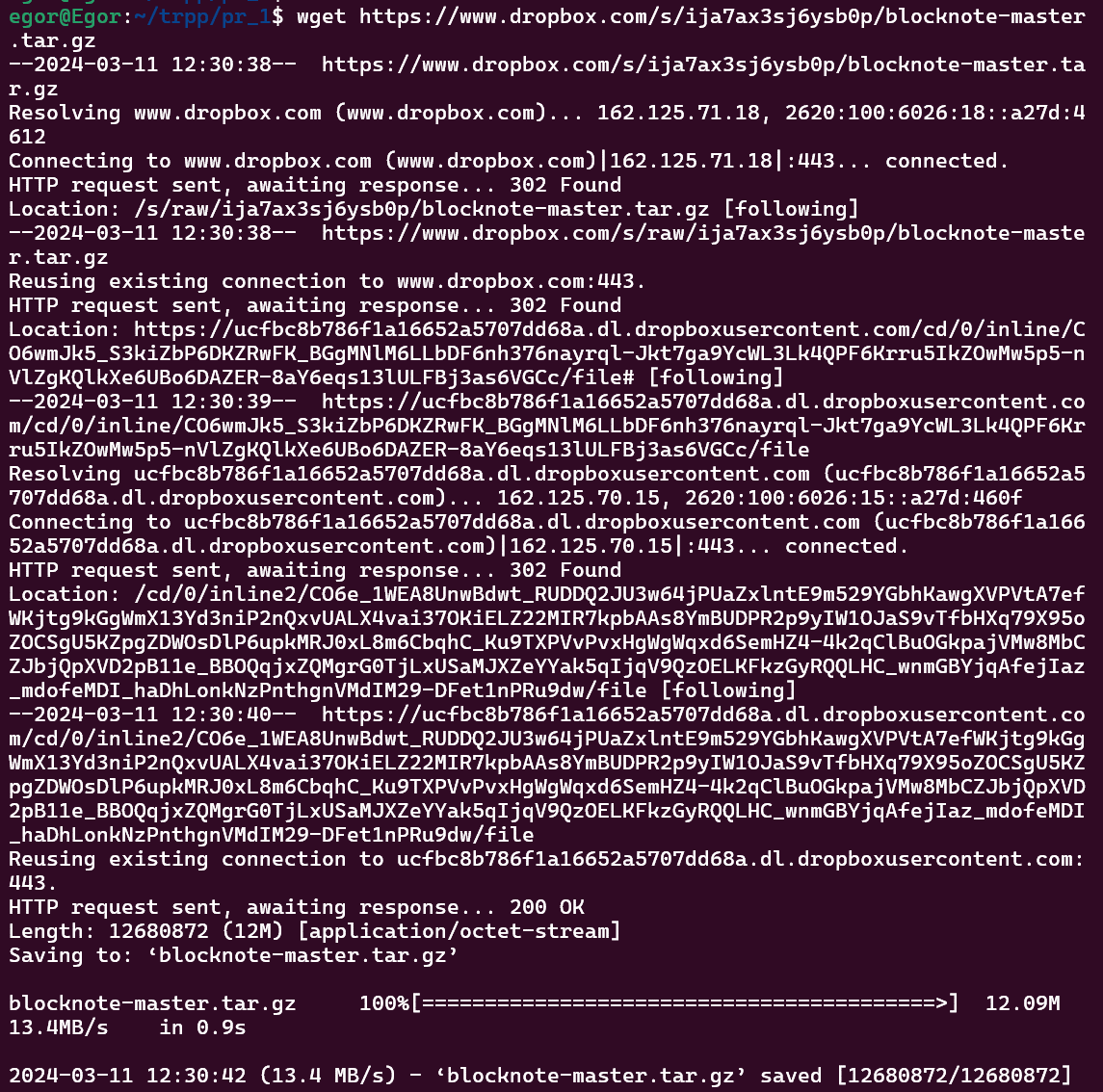


Рисунок 13 – Установка архива



Рисунок 14 – Разархивация



Рисунок 16 – Файл requirements.txt с зависимости

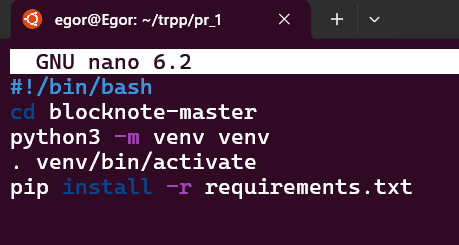


Рисунок 17 – Скрипт setup

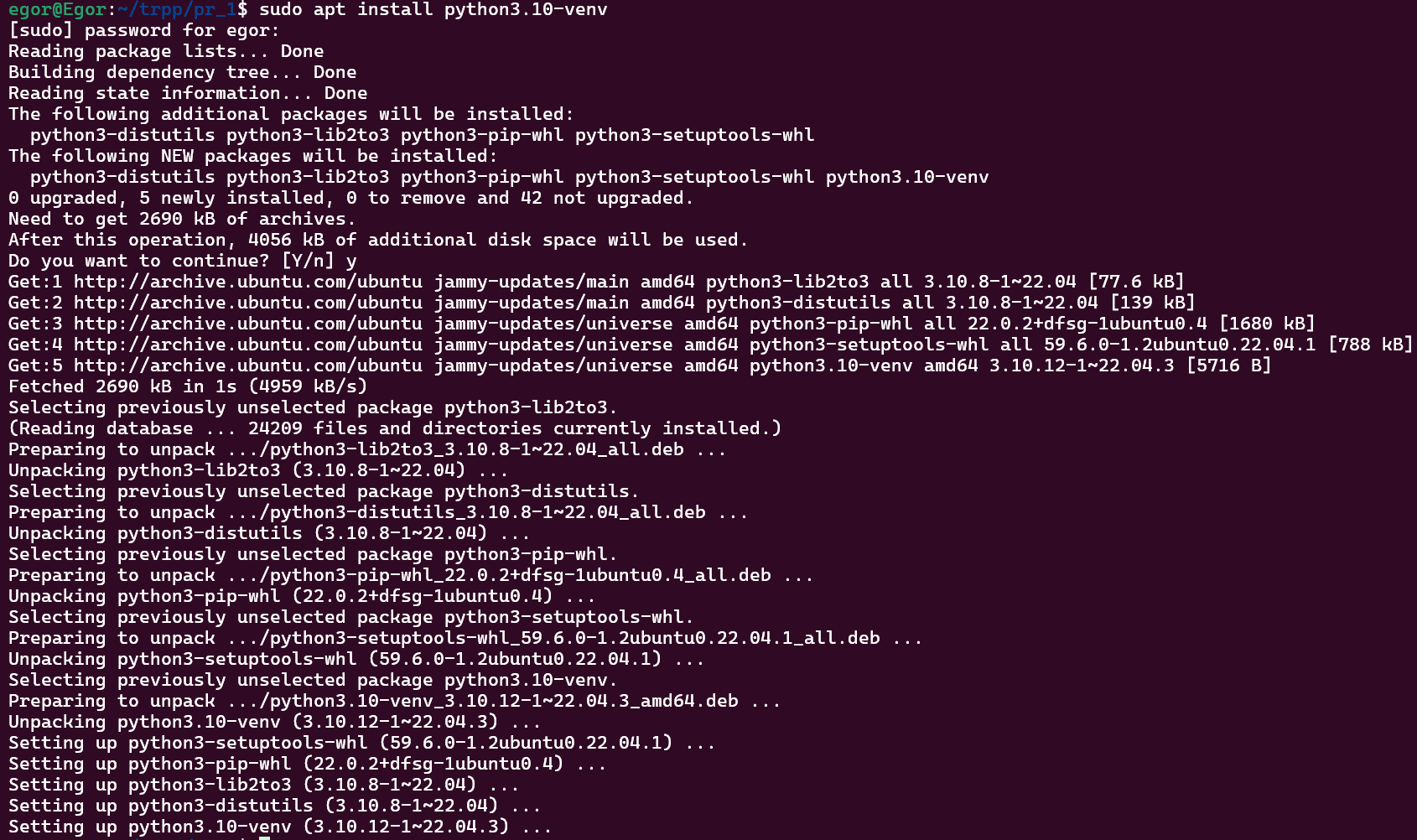


Рисунок 18 – Установка новой версии питона

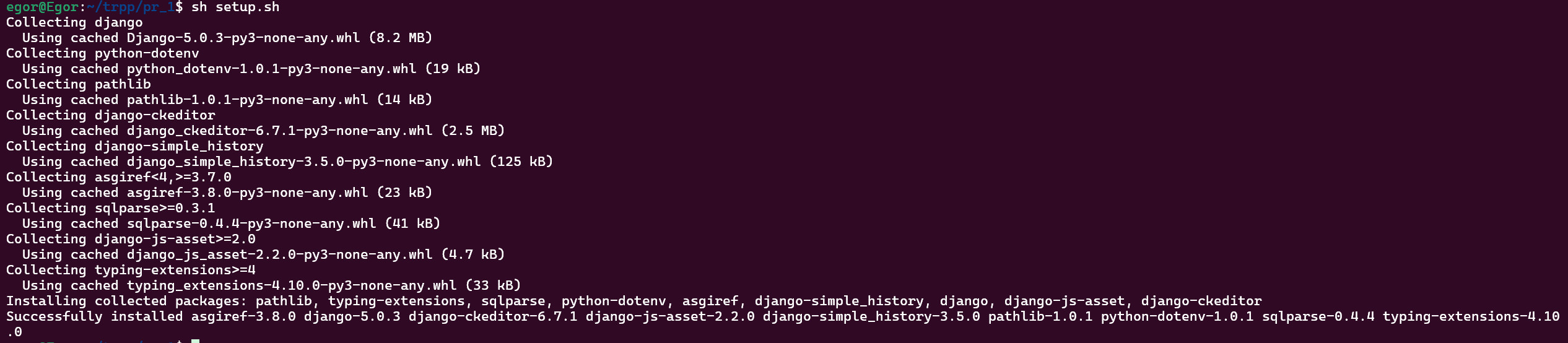


Рисунок 19 – Запуск скрипта setup

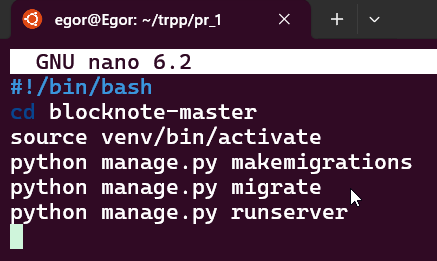


Рисунок 20 – Скрипт run

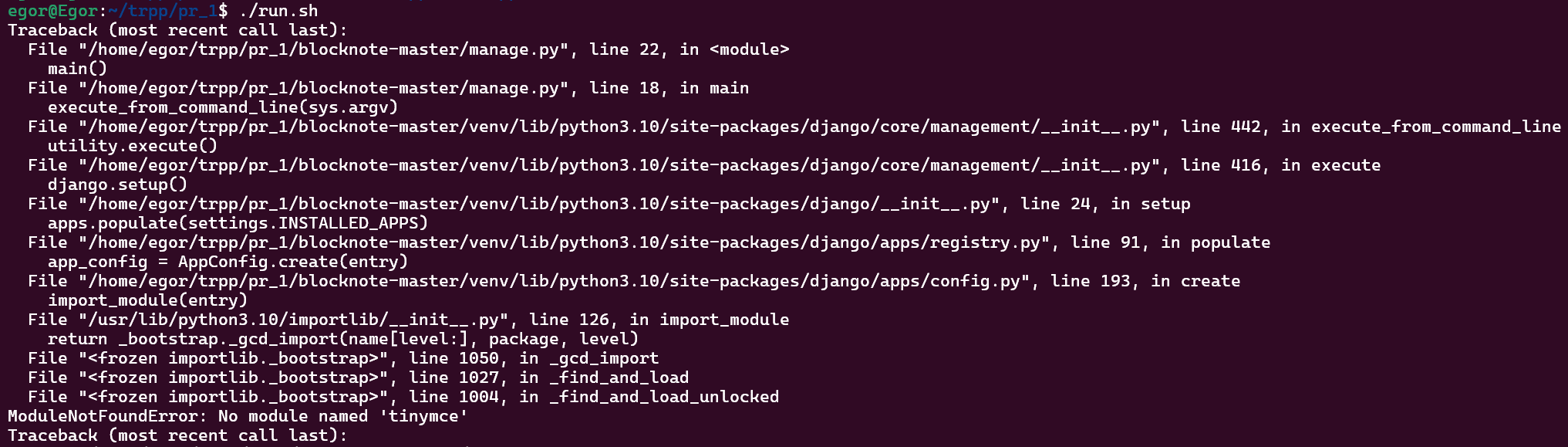


Рисунок 21 – Запуск скрипта run

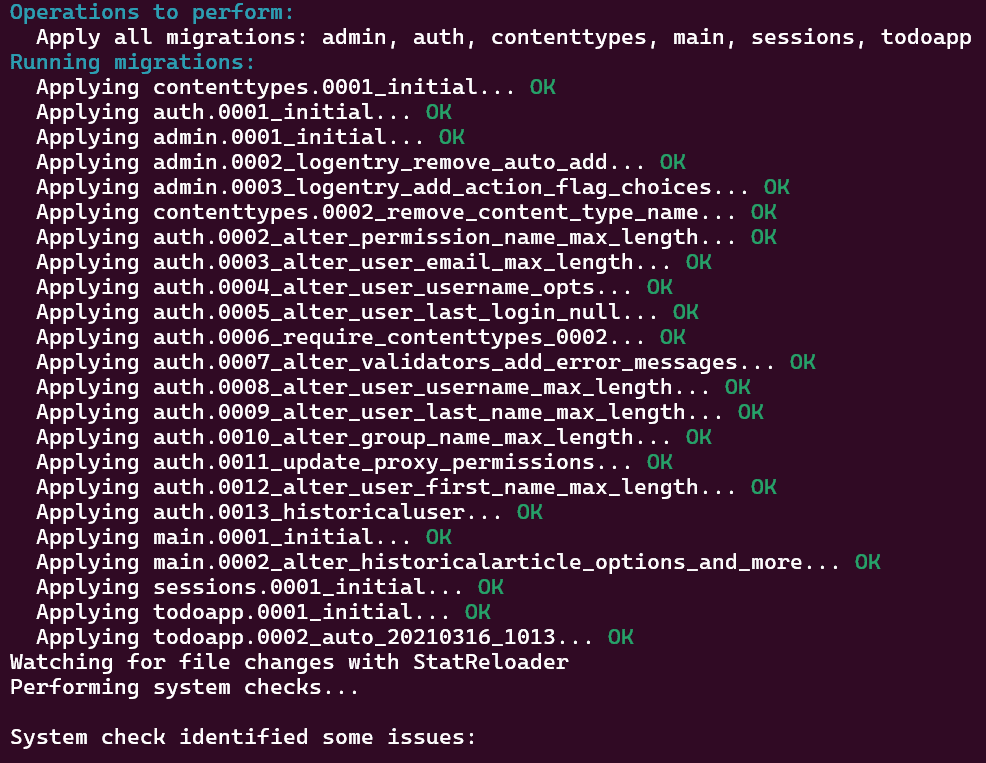


Рисунок 22 – Запуск системы

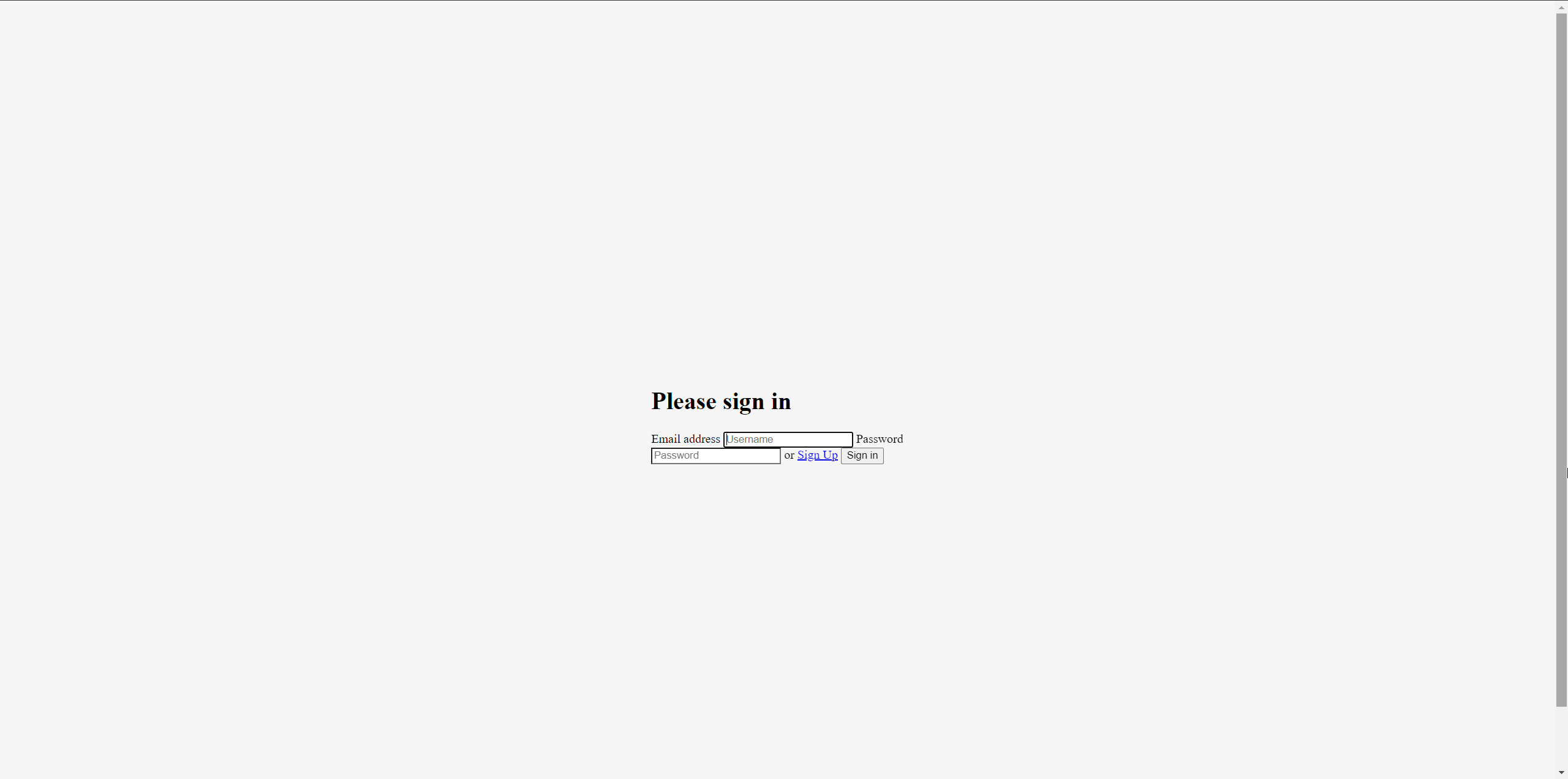


Рисунок 23 – Результат запуска

# Вывод

В ходе выполнения данной работы были изучены основы работы с Bash Scriptами. Были написаны скрипты, выполняющие разные задачи, а также помогающие создать окружение, необходимое для запуска приложения.