



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 2**  
**по дисциплине**  
**«Технология разработки программных приложений»**

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

Выполнил студент группы ИКБО-13-22

Тринеев П.С

Принял доцент

Петренко А.А.


Практическая работа выполнена

✓ «17» 02 2024 г.

  
(подпись студента)

«Зачтено»

✓ «17» 02 2024 г.

  
(подпись руководителя)

Москва 2024

## Оглавление

Часть 1. Базовые Bash скрипты.....	2
Задание 1. ....	2
Задание 2. ....	3
Задание 3. ....	4
Задание 4. ....	5
Задание 5. ....	6
Часть 2. Развертка и запуск проекта при помощи Bash Script.....	8
1. Определение зависимостей проекта.....	8
2. Создание виртуального окружения.....	9
3. Написание скрипта запуска приложения на новой системе.....	12

## Часть 1. Базовые Bash скрипты.

### Задание 1.

Напишите сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл.



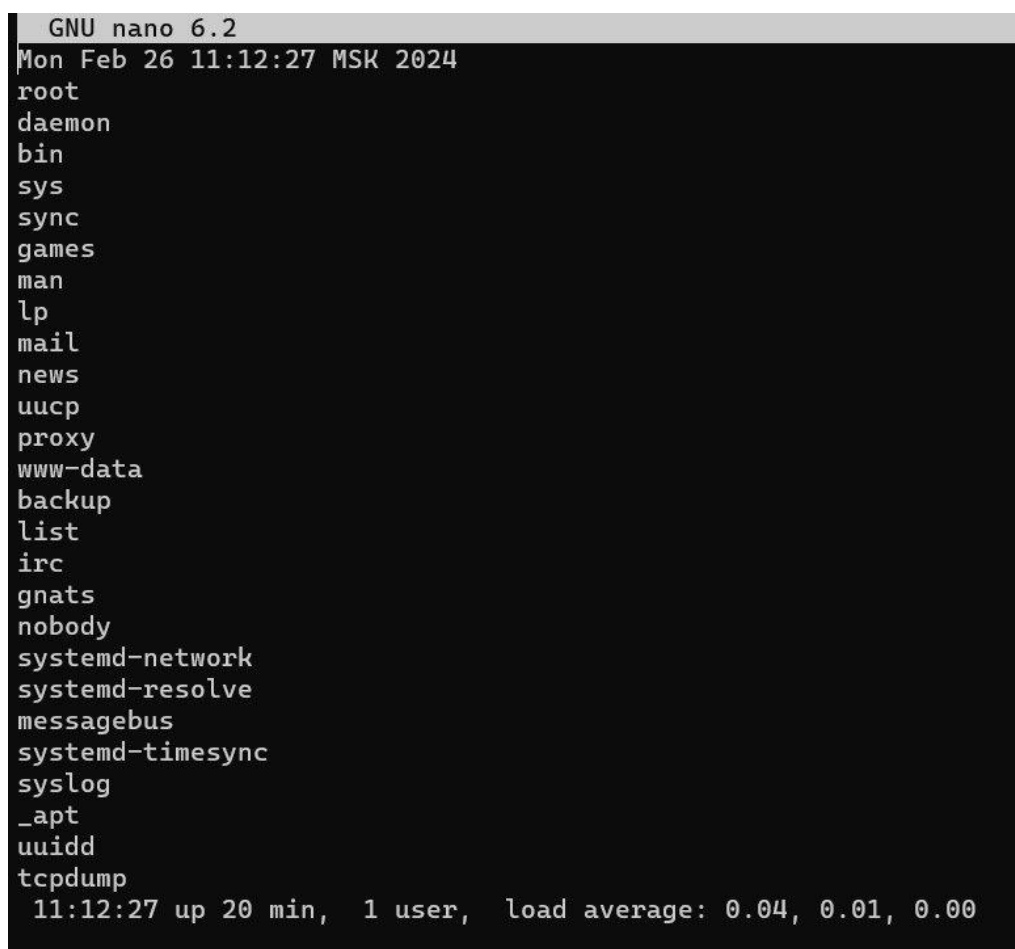
```
GNU nano 6.2
#!/bin/bash
date
sed 's/:.*/' /etc/passwd
uptime
```

Рис. 1. Содержимое файла myscrip



```
root@MSI:~# chmod +x ./myscrip
root@MSI:~# ./myscrip > file.txt
root@MSI:~#
```

Рис. 2. Запуск скрипта myscrip

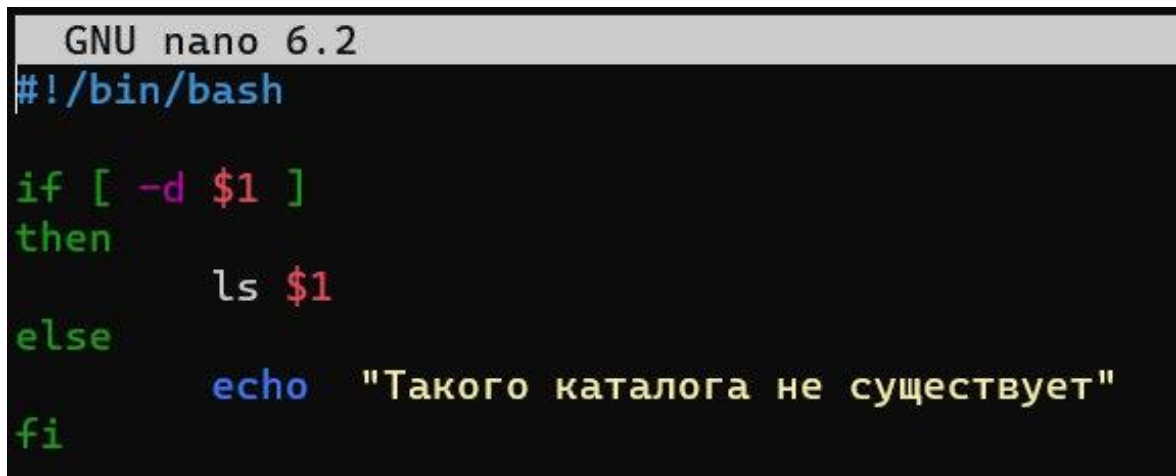


```
GNU nano 6.2
Mon Feb 26 11:12:27 MSK 2024
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
messagebus
systemd-timesync
syslog
_apt
uidd
tcpdump
11:12:27 up 20 min, 1 user, load average: 0.04, 0.01, 0.00
```

Рис. 3. Содержимое файла file.txt

## Задание 2.

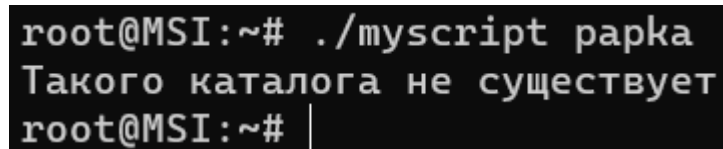
Напишите сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует.



```
GNU nano 6.2
#!/bin/bash

if [ -d $1 ]
then
    ls $1
else
    echo "Такого каталога не существует"
fi
```

Рис. 4. Содержимое файла myscrip

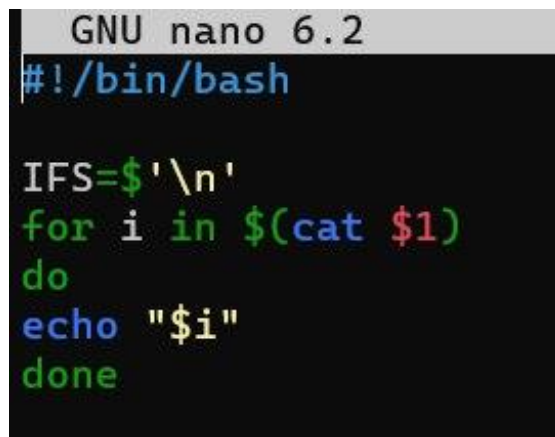


```
root@MSI:~# ./myscrip папка
Такого каталога не существует
root@MSI:~# |
```

Рис. 5. Вывод отработки скрипта

### Задание 3.

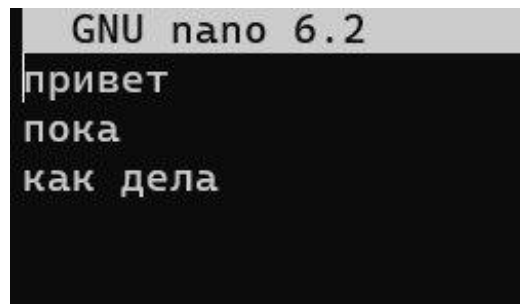
Напишите сценарий, который с помощью цикла прочитает файл и выведет его содержимое.



```
GNU nano 6.2
#!/bin/bash

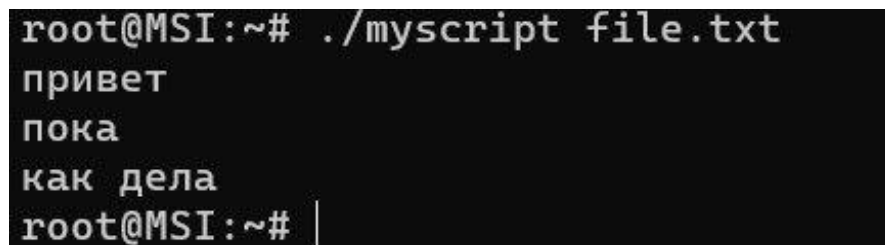
IFS=$'\n'
for i in $(cat $1)
do
echo "$i"
done
```

Рис. 6. Содержимое файла mysript



```
GNU nano 6.2
привет
пока
как дела
```

Рис. 7. Содержание файла file.txt

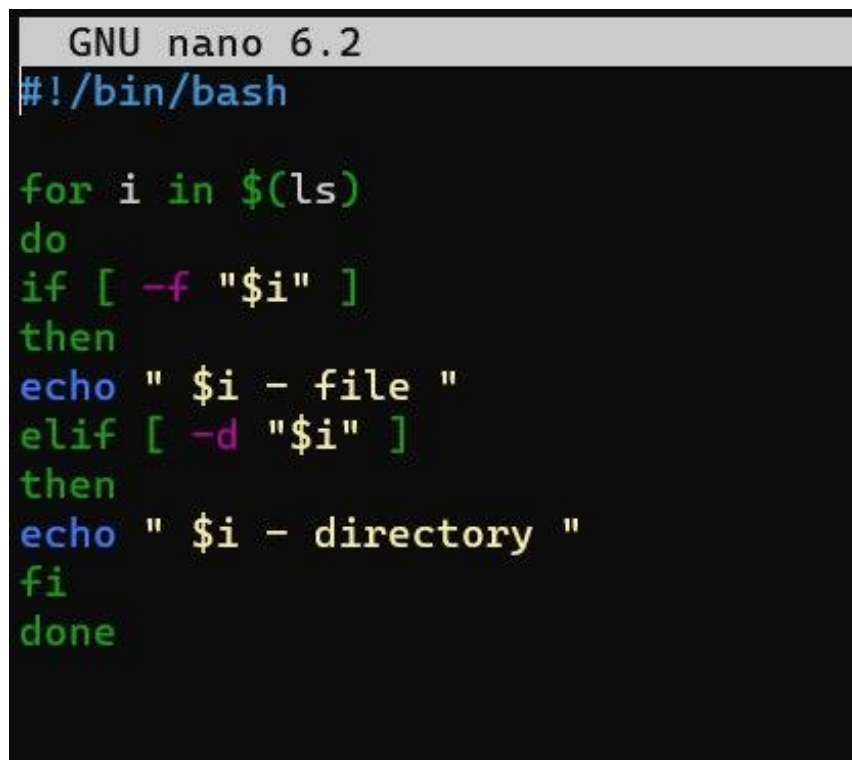


```
root@MSI:~# ./myscript file.txt
привет
пока
как дела
root@MSI:~# |
```

Рис. 8. Вывод отработки скрипта

#### Задание 4.

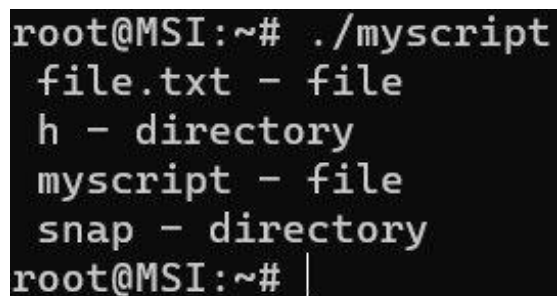
Напишите сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория.



```
GNU nano 6.2
#!/bin/bash

for i in $(ls)
do
if [ -f "$i" ]
then
echo " $i - file "
elif [ -d "$i" ]
then
echo " $i - directory "
fi
done
```

Рис. 9. Содержимое файла myscrip




```
root@MSI:~# ./myscrip
file.txt - file
h - directory
myscrip - file
snap - directory
root@MSI:~# |
```

Рис. 10. Вывод отработки скрипта

### Задание 5.

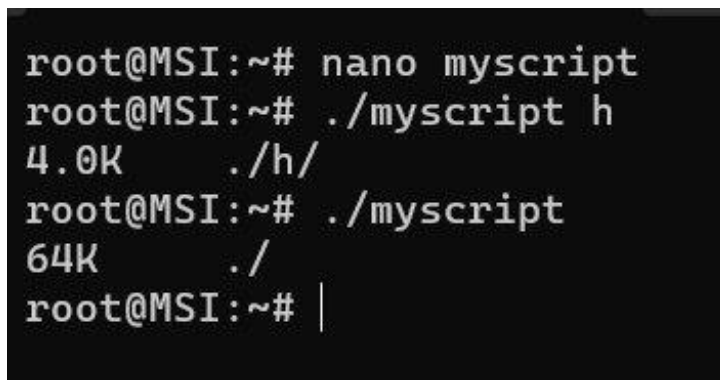
Напишите сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе.



```
GNU nano 6.2
#!/bin/bash

du -sh ./${1}/
```

Рис. 11. Содержимое файла myscript

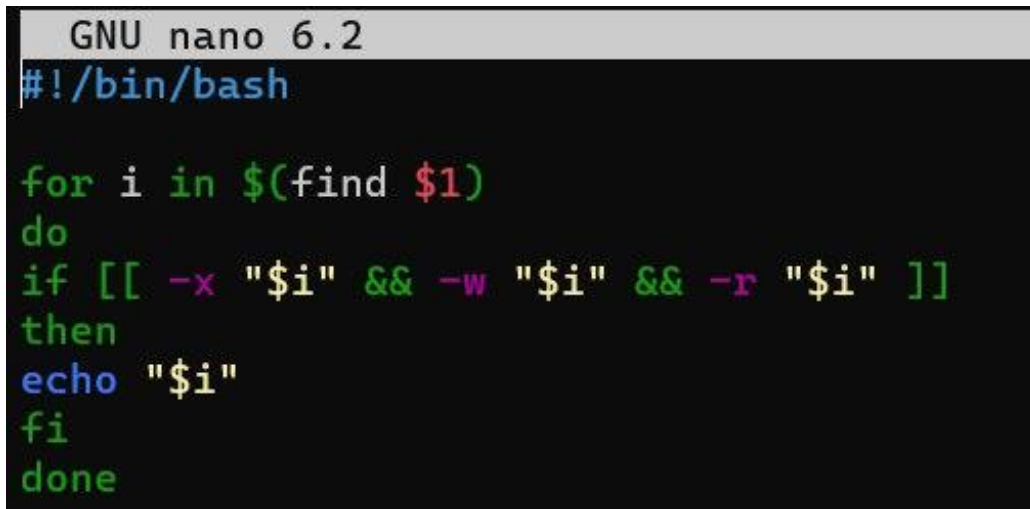


```
root@MSI:~# nano myscript
root@MSI:~# ./myscript h
4.0K    ./h/
root@MSI:~# ./myscript
64K     ./
root@MSI:~# |
```

Рис. 12. Вывод отработки скрипта

### Задание 6.

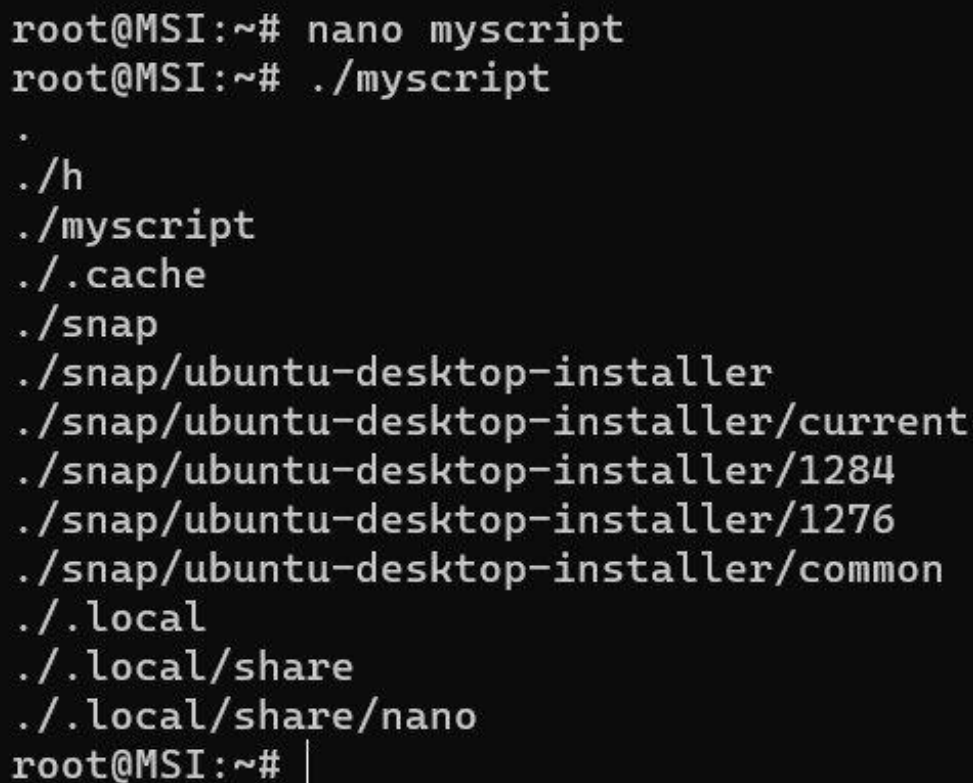
Напишите сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение.



```
GNU nano 6.2
#!/bin/bash

for i in $(find $1)
do
if [[ -x "$i" && -w "$i" && -r "$i" ]]
then
echo "$i"
fi
done
```

Рис. 13. Содержимое файла mysript



```
root@MSI:~# nano mysript
root@MSI:~# ./myscript
.
./h
./myscript
./.cache
./snap
./snap/ubuntu-desktop-installer
./snap/ubuntu-desktop-installer/current
./snap/ubuntu-desktop-installer/1284
./snap/ubuntu-desktop-installer/1276
./snap/ubuntu-desktop-installer/common
./.local
./.local/share
./.local/share/nano
root@MSI:~# |
```

Рис. 14. Вывод отработки скрипта



## 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. 11 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/](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.

```
root@MSI:~# ls
blocknote-master.tar.gz catalog file.txt myscript snap
root@MSI:~# tar -xvf blocknote-master.tar.gz
```

Рис. 15. Распаковка архива `blocknote-master.tar.gz`

```
#!/bin/bash
echo -n > requirements.txt
for file in $(find -name '*.py')
do
    grep -ohP '(?<=^from )\w+|(?<=^import )\w+' $file >> requirements.txt
done
sort -u -o requirements.txt requirements.txt
```

Рис. 16. Скрипт `myscript`

```
root@MSI:~# ./myscript
root@MSI:~# cat requirements.txt
ckeditor
django
dotenv
main
os
pathlib
simple_history
sys
time
tinymce
```

Рис. 17. Содержимое файла requirements.txt

## 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-setupenvironment>.

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.

```
GNU nano 6.2                                mysc
#!/bin/bash

cd blocknote-master

echo -n > requirements.txt
for i in $(find -name '*.py')
do
    grep -ohP '(?<=^from )\w+|(?<=^import )\w+' $i >> requirements.txt
done

sort -u -o requirements.txt requirements.txt

apt install python3
apt install python3-pip
apt install python3-virtualenv

virtualenv venv
source venv/bin/activate

for i in $(cat requirements.txt)
do
    pip install $i
done

deactivate

cd ..
```

Рис. 18. Скрипт myscript

```
(venv) root@MSI:~/blocknote-master# pip freeze
alabaster==0.7.16
asgiref==3.7.2
asyncio==3.4.3
Babel==2.14.0
boltons==23.1.1
certifi==2024.2.2
cffi==1.16.0
charset-normalizer==3.3.2
configparser==6.0.1
cryptography==42.0.5
dataclasses==0.6
DateTime==5.4
dis==1.0.1
Django==5.0.2
docutils==0.20.1
functools==1.0.2
Genshi==0.7.7
idna==3.6
imagesize==1.4.1
importlib==1.0.4
ipaddress==1.0.23
Jinja2==3.1.3
lxml==5.1.0
MarkupSafe==2.1.5
packaging==23.2
pathlib==1.0.1
pyparsing==2.21
Pygments==2.17.2
pytz==2024.1
railroad==0.5.0
requests==2.31.0
six==1.16.0
snowballstemmer==2.2.0
Sphinx==7.2.6
sphinxcontrib-applehelp==1.0.8
sphinxcontrib-devhelp==1.0.6
sphinxcontrib-htmlhelp==2.0.5
sphinxcontrib-jsmath==1.0.1
```

Рис. 19. Список всех 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-а 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.

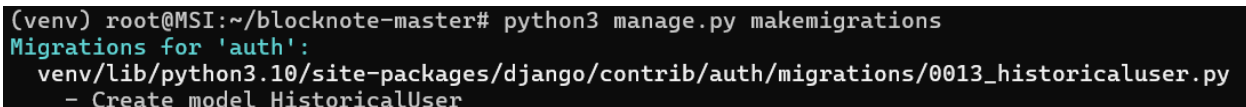
После этого необходимо воссоздать полученное на прошлом этапе виртуальное окружение со всеми зависимостями.

Затем необходимо запустить проект из виртуального окружения при помощи следующих команд:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py runserver
```



```
(venv) root@MSI:~/blocknote-master# python3 manage.py makemigrations
Migrations for 'auth':
  venv/lib/python3.10/site-packages/django/contrib/auth/migrations/0013_historicaluser.py
  - Create model HistoricalUser
```

Рис. 20. Выполнение команды `python manage.py makemigrations`

```
(venv) root@MSI:~/blocknote-master# python3 manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions, todoapp
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying auth.0013_historicaluser... OK
  Applying main.0001_initial... OK
  Applying sessions.0001_initial... OK
  Applying todoapp.0001_initial... OK
  Applying todoapp.0002_auto_20210316_1013... OK
(venv) root@MSI:~/blocknote-master# python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

Рис. 21. Выполнение команды python manage.py migrate

```
(venv) root@MSI:~/blocknote-master# python3 manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 02, 2024 - 12:05:38
Django version 3.1.6, using settings 'appengine.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[02/Mar/2024 12:05:45] "GET / HTTP/1.1" 302 0
[02/Mar/2024 12:05:45] "GET /login/?next=/ HTTP/1.1" 200 2608
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:05:45] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /docs/5.0/assets/img/favicons/manifest.json
Not Found: /docs/5.0/assets/img/favicons/favicon.ico
[02/Mar/2024 12:05:46] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
[02/Mar/2024 12:05:46] "GET /docs/5.0/assets/img/favicons/favicon.ico HTTP/1.1" 404 3846
Not Found: /docs/5.0/assets/img/favicons/favicon-32x32.png
[02/Mar/2024 12:05:46] "GET /docs/5.0/assets/img/favicons/favicon-32x32.png HTTP/1.1" 404 3864
Not Found: /docs/5.0/assets/img/favicons/favicon-16x16.png
[02/Mar/2024 12:05:46] "GET /docs/5.0/assets/img/favicons/favicon-16x16.png HTTP/1.1" 404 3864
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:05:47] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /signup/
[02/Mar/2024 12:05:48] "GET /signup/ HTTP/1.1" 404 3747
Not Found: /favicon.ico
[02/Mar/2024 12:05:48] "GET /favicon.ico HTTP/1.1" 404 3759
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:05:51] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:05:59] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:06:15] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:06:47] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
Not Found: /docs/5.0/assets/img/favicons/manifest.json
[02/Mar/2024 12:07:51] "GET /docs/5.0/assets/img/favicons/manifest.json HTTP/1.1" 404 3852
```

Рис. 22. Выполнение команды python manage.py runserver

## **Вывод.**

Было освоена работа с Bash Scriptами.