

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

Ордена трудового Красного Знамени

**Федеральное государственное бюджетное образовательное учреждение высшего
образования**

**«Московский технический университет связи и информатики»
Кафедра «Сетевые информационные технологии и сервисы»**



Отчет по лабораторной работе

по дисциплине

«Прикладные информационные системы»

на тему:

«Начало работы с Python»

Выполнил студент 4 курса

Группы: УБСТ 2203

Музыченко Г. Н.

Москва, 2025

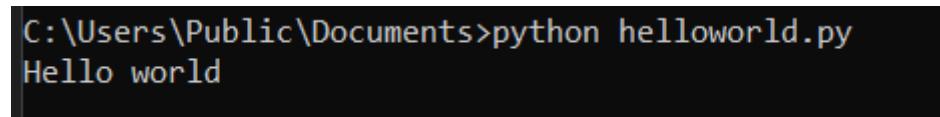
Ход работы

1. Установка Python

После установки Python создадим helloworld.py и добавим туда строчку:

```
print ("Hello World")
```

Результат открытия файла в командной строке (рис. 1).



```
C:\Users\Public\Documents>python helloworld.py
Hello world
```

Рисунок 1 – Вывод командной строки

Создадим файл mygroup.py и добавим туда следующий код:

```
groupmates = [
    {
        "name": "Александр",
        "surname": "Иванов",
        "exams": ["Информатика", "ЭЭиС", "Web"],
        "marks": [4, 3, 5]
    },
    {
        "name": "Иван",
        "surname": "Петров",
        "exams": ["История", "АиГ", "КТП"],
        "marks": [4, 4, 4]
    },
    {
        "name": "Максим",
        "surname": "Перов",
        "exams": ["Математика", "АиГ", "КТП"],
        "marks": [4, 4, 4]
    },
    {
        "name": "Алексей",
        "surname": "Роднищев",
        "exams": ["Web", "Математика", "КТП"],
        "marks": [4, 3, 4]
    },
    {
        "name": "Дарья",
        "surname": "Малышева",
        "exams": ["Информатика", "Математика", "Web"],
        "marks": [5, 5, 4]
    }
]

def print_students(students):
    print("Имя".ljust(15), "Фамилия".ljust(10), "Экзамены".ljust(30),
          "Оценки".ljust(20))
    for student in students:
        print(student["name"].ljust(15), student["surname"].ljust(10),
              str(student["exams"]).ljust(30), str(student["marks"]).ljust(20))

print_students(groupmates)
```

Переменная `groupmates` содержит данные студентов, функция `print_students` содержит цикл, который выводит данные в командную строку в виде отформатированной таблицы. Результат ниже (рис. 2)

Имя	Фамилия	Экзамены	Оценки
Александр	Иванов	['Информатика', 'ЭЭиС', 'Web']	[4, 3, 5]
Иван	Петров	['История', 'АиГ', 'КТП']	[4, 4, 4]
Максим	Перов	['Математика', 'АиГ', 'КТП']	[4, 4, 4]
Алексей	Роднищев	['Web', 'Математика', 'КТП']	[4, 3, 4]
Дарья	Малышева	['Информатика', 'Математика', 'Web']	[5, 5, 4]

Рисунок 2 – Вывод командной строки

Добавим функцию для фильтрации списка студентов по среднему баллу, введенному с клавиатуры:

```
def filter_avgscore(students, score_threshold):
    result = []

    for student in students:
        avg = sum(student["marks"]) / len(student["marks"])

        if avg > score_threshold:
            print(student ["name"], "- прошел фильтр!")
            result.append(student)

    return result

input_score = float(input("Введите средний балл"))
filtered_students = filter_avgscore(groupmates, input_score)

print("\n Студенты со средним балом выше ", input_score)
print_students(filtered_students)
```

```
C:\Users\Public\Documents>python mygroup.py
Имя          Фамилия    Экзамены          Оценки
Александр    Иванов     ['Информатика', 'ЭЭиС', 'Web'] [4, 3, 5]
Иван         Петров     ['История', 'АиГ', 'КТП']   [4, 4, 4]
Максим       Перов      ['Математика', 'АиГ', 'КТП'] [4, 4, 4]
Алексей      Роднищев  ['Web', 'Математика', 'КТП'] [4, 3, 4]
Дарья        Малышева  ['Информатика', 'Математика', 'Web'] [5, 5, 4]
Ведите средний балл 3.5
Александр - прошел фильтр!
Иван - прошел фильтр!
Максим - прошел фильтр!
Алексей - прошел фильтр!
Дарья - прошел фильтр!

Студенты со средним балом выше 3.5
Имя          Фамилия    Экзамены          Оценки
Александр    Иванов     ['Информатика', 'ЭЭиС', 'Web'] [4, 3, 5]
Иван         Петров     ['История', 'АиГ', 'КТП']   [4, 4, 4]
Максим       Перов      ['Математика', 'АиГ', 'КТП'] [4, 4, 4]
Алексей      Роднищев  ['Web', 'Математика', 'КТП'] [4, 3, 4]
Дарья        Малышева  ['Информатика', 'Математика', 'Web'] [5, 5, 4]
```

Рисунок 3 – Вывод командной строки

2. Установка Django

Запустим проект с именем lab_project

```
C:\Users\Public\Documents\lab_project>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you
run 'python manage.py migrate' to apply them.
November 18, 2025 - 20:33:31
Django version 5.2.8, using settings 'lab_project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Рисунок 4 – Вывод командной строки

После установки прейдем по адресу <http://127.0.0.1:8000/>, где будет расположена стартовая страница (рис. 5)



Рисунок 5 – Стартовая страница

3. Настройка сервера

После успешного запуска сервера, меняем содержимое settings.py путем редактирования параметра NAME (рис. 6)

```
 DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db_lab_project'),
    }
}
```

Рисунок 6 – Содержимое файла

Выполняем: `python manage.py migrate` (рис. 7)

```
C:\Users\Public\Documents\lab_project>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
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 sessions.0001_initial... OK
```

Рисунок 7 – Вывод командной строки

Заходим в интерфейс под суперпользователем (рис. 8)

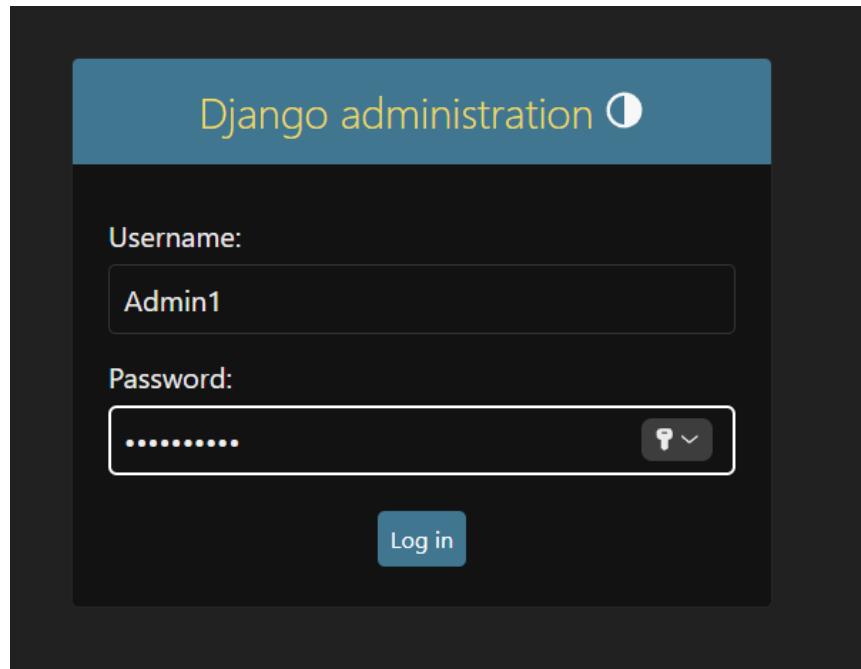


Рисунок 8 – Интерфейс Django

Создадим пользователя Mike, пароль: QVV-Z9K-Gmy-5y5, установив галочку Superuser Status и Staff status.

И пользователя Piter, пароль: 6Tz-eZ4-uiR-N7Z без включения в Staff и без прав суперпользователя (рис. 9).

Action:	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Admin1	-	-	-	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Mike	-	-	-	<input checked="" type="checkbox"/>
<input type="checkbox"/>	Piter	-	-	-	<input type="checkbox"/>
3 users					

Рисунок 9 – Интерфейс Django

Убрав галочку Active, мы можем забанить пользователя (рис. 10)

Permissions

Active
Designates whether this user should be treated as active. Unselect this instead of deleting accounts.

Staff status
Designates whether the user can log into this admin site.

Superuser status
Designates that this user has all permissions without explicitly assigning them.

Рисунок 10 – Интерфейс Django

4. Загрузка проекта на Git

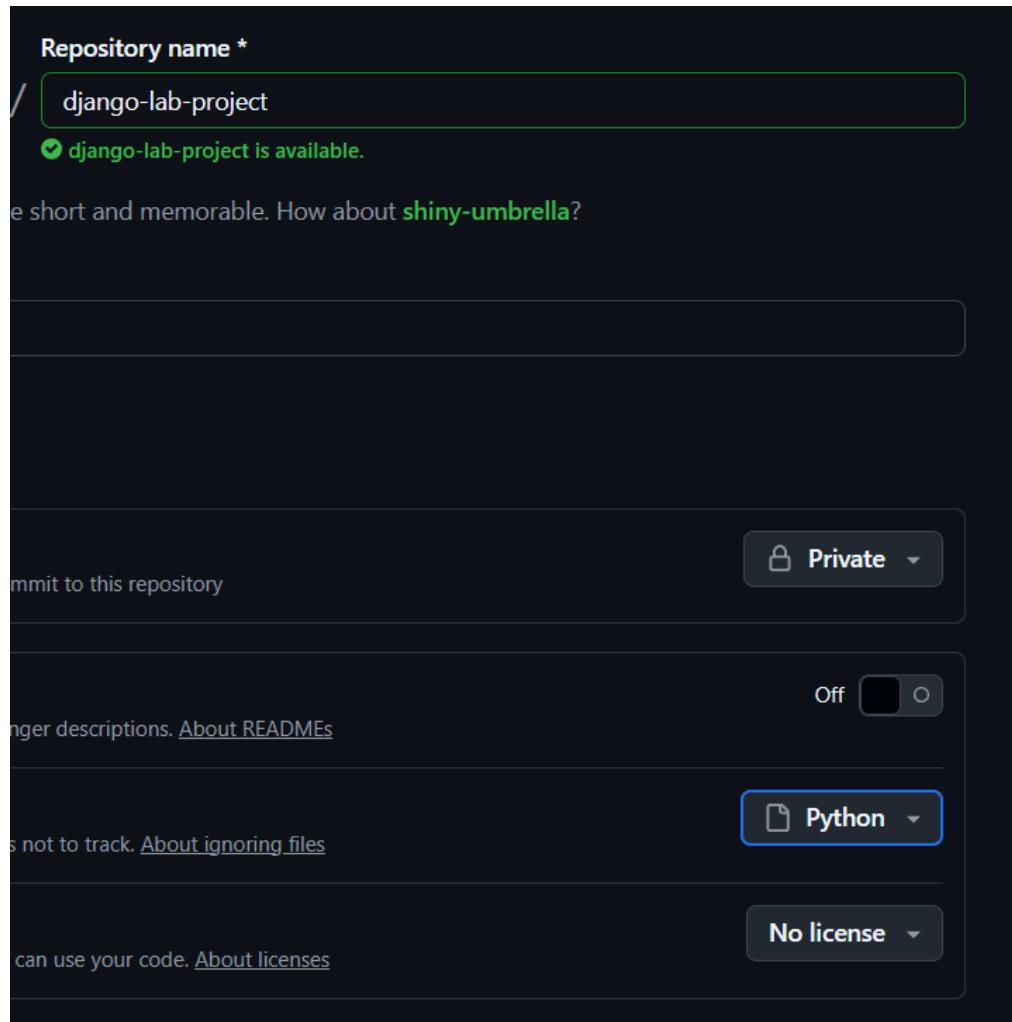


Рисунок 11 – Страница Git

