Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра математической кибернетики и информационных технологий

Лабораторная работа №3

по дисциплине

«Кроссплатформенные технологии программирования»

Выполнил студент группы

БСТ1702

Панкратов В.И.

Вариант №14

Москва

2019

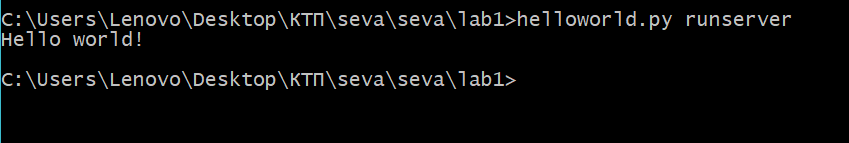
**Выполнено в среде IDLE Python 2.7.15**

**Задание 1:** Выполнить простейшую программу с помощью командной строки.

**Код:**

Print (“Hello,world!”)

**Результат выполнения:**



**Задание 2:** Форматированный вывод списка студентов в виде таблицы

**Код:**

#coding:utf-8

groupmates = [

{

"name": u"Полина",

"group": "1702",

"age": 19,

"marks": [5,4,4,5,5]

},

{

"name": u"Татьяна",

"group": "1703",

"age": 18,

"marks": [5,4,4,5,3]

},

{

"name": u"Анастасия",

"group": "1701",

"age": 19,

"marks": [5,5,5,5,5]

},

{

"name": u"Екатерина",

"group": "1702",

"age": 19,

"marks": [5,5,4,5,5]

}

]

def print\_students(students):

print u"Имя студента".ljust(15), \

u"Группа".ljust(8), \

u"Возраст".ljust(8), \

u"Оценки".ljust(20)

for student in students:

print \

student["name"].ljust(15), \

student["group"].ljust(8), \

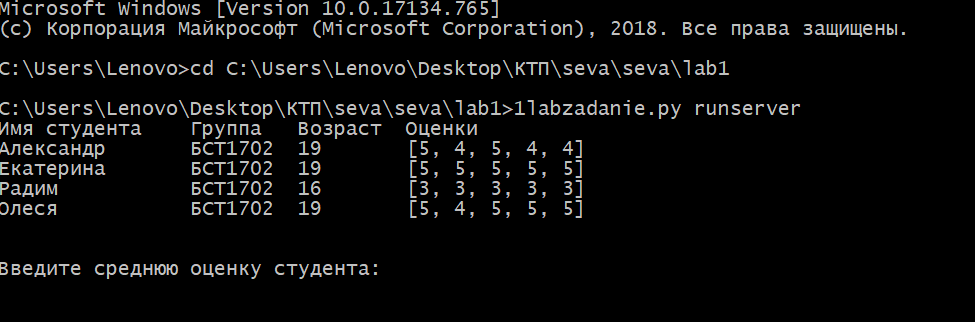
str(student["age"]).ljust(8), \

str(student["marks"]).ljust(20)

print "\n"

print\_students(groupmates)

**Результат выполнения:**



**Задание 3:** Напишите функцию фильтрации студентов по средней оценке (так чтобы функция возвращала всех студентов выше заданного в параметрах

функции среднего балла).

Примерная схема работы функции: создание пустого массива, куда будут добавляться все студенты, прошедшие фильтрацию; запуск цикла, в каждой итерации которого необходимо считать среднюю оценку текущего студента и сравнивать с тем значением, что передано в качестве параметра.

**Код:**

Код из задания 2

def filtration(students,p):

print u"Студенты, средняя оценка которых выше заданного параметра"

b=[]

for i in students:

a=len(i["marks"])

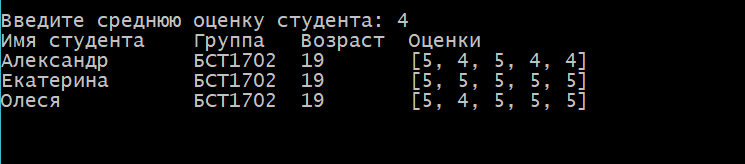
m=sum(i["marks"])

if (m/a)>=p:

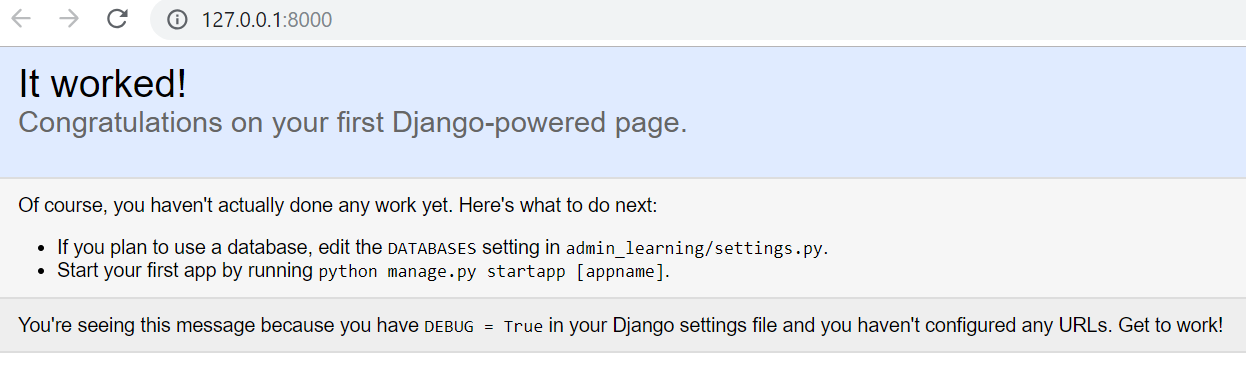
b.append(i)

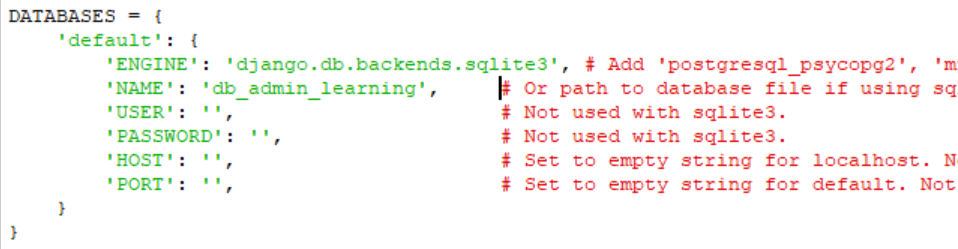
print\_students(b)

**Результат выполнения:**



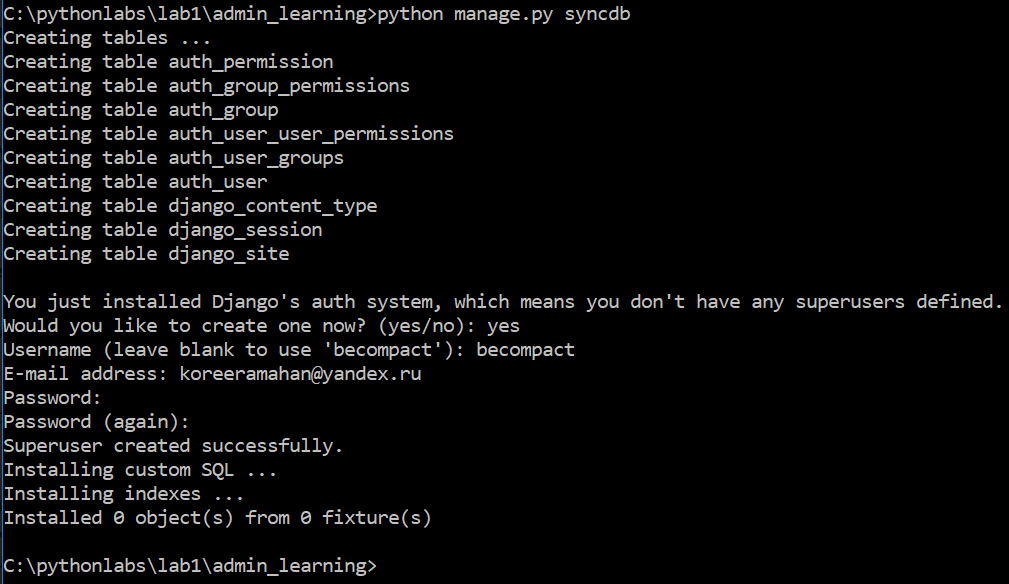
**Посещение страницы http://127.0.0.1:8000/ с помощью браузера.**





Теперь в качестве базы данных подключена sqlite3, а название новосозданной базы будет “db\_admin\_learning”.

**Исследование административного интерфейса django.**



**После раскомментирования в settings.py:**

INSTALLED\_APPS = (

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.sites',

'django.contrib.messages',

'django.contrib.staticfiles',

# Uncomment the next line to enable the admin:

**'django.contrib.admin',**

# Uncomment the next line to enable admin documentation:

# 'django.contrib.admindocs',

)

**После раскомментирования в urls.py:**

from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:

**from django.contrib import admin**

**admin.autodiscover()**

urlpatterns = patterns('',

# Examples:

# url(r'^$', 'admin\_learning.views.home', name='home'),

# url(r'^admin\_learning/', include('admin\_learning.foo.urls')),

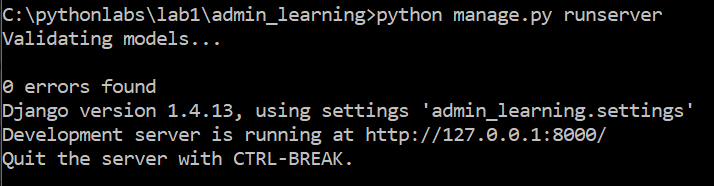
# Uncomment the admin/doc line below to enable admin documentation:

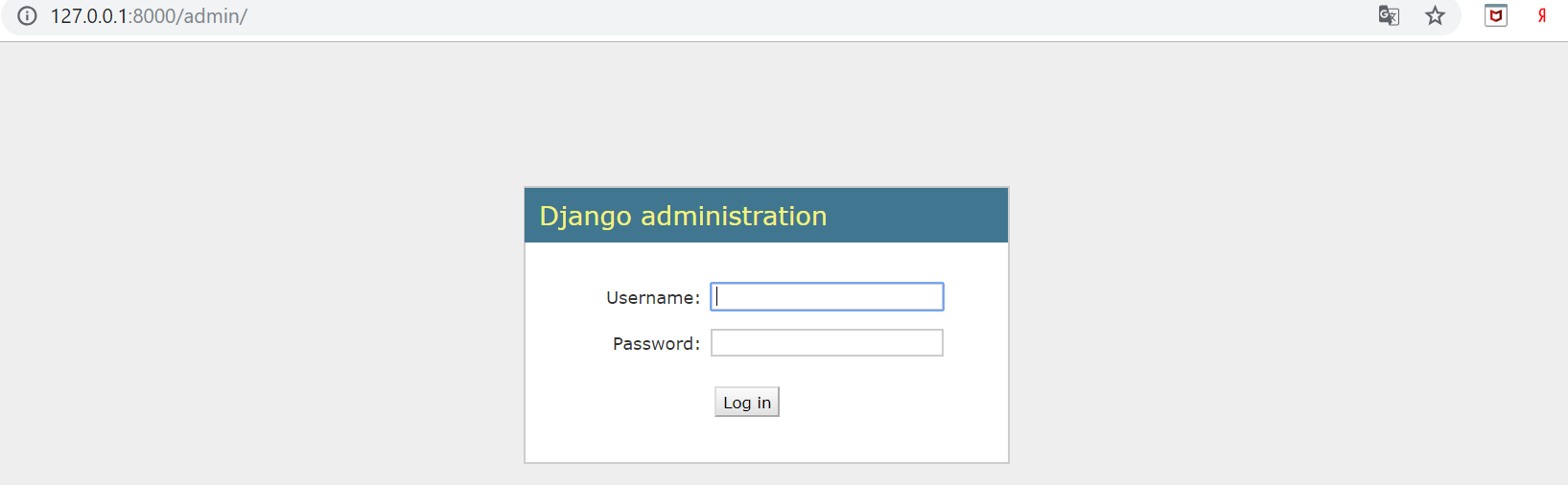
# url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

# Uncomment the next line to enable the admin:

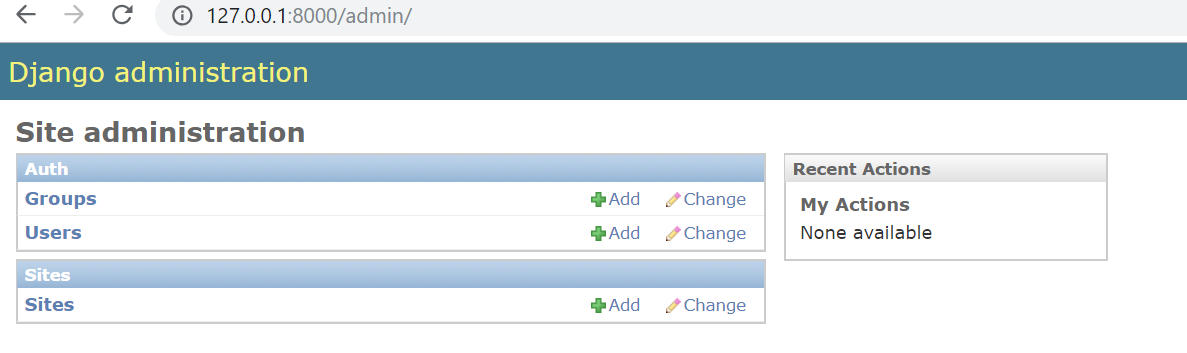
**url(r'^admin/', include(admin.site.urls)),**

**Запуск сервера и переход по адресу http://127.0.0.1:8000/admin/:**

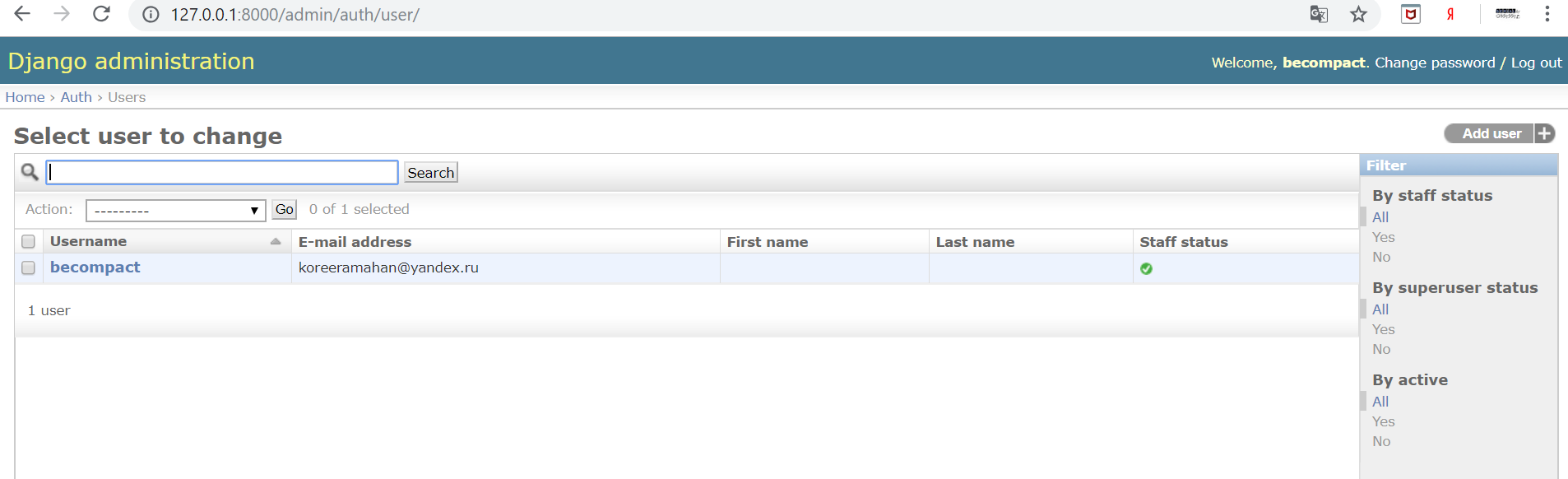




**Скриншот страницы административного интерфейса Django:**



**Список всех зарегистрированных пользователей на сайте:**

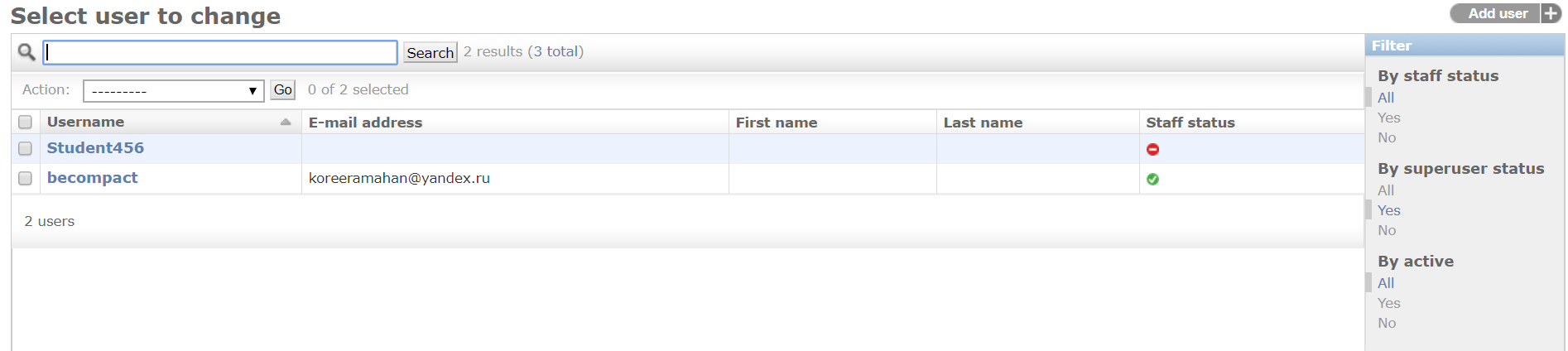


**Изучение интерфейса.**

**Список после добавления пользователей с правами и без прав суперюзера:**



**Список всех пользователей с правами суперюзера:**



**Список пользователей без прав суперюзера:**



**Список пользователей, которые были «забанены» (неактивные):**

