|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_

**О Т Ч Е Т**

**по лабораторной работе №6**

по дисциплине: Разработка интернет-приложений\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

на тему: Работа с формами, авторизация и модуль администрирования в Django. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_ИУ5-53Б\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Труфанов В.А.\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Гапанюк Ю.Е.\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2019 г.*

**Задание и порядок выполнения**

Основная цель данной лабораторной работы – научиться обрабатывать веб-формы на стороне приложения, освоить инструменты, которые предоставляет Django, по работе с формами. Также в этой лабораторной работе вы освоите инструменты Django по работе с авторизацией и реализуете простейшую авторизацию. Напоследок, вы познакомитесь с инструментом администрирования Django – как в несколько строчек кода сделать панель администратора сайта.

1. **Выполнение задания и исходный код проекта**

Создайте view, которая возвращает форму для регистрации**:**

class MyFormView(FormControlsStylerMixin, View):

template\_name = None

form\_class = None

form = None

def render\_form(self, request, \*args, \*\*kwargs):

return render(request, self.template\_name, {'form': self.form})

def get(self, request, \*args, \*\*kwargs):

self.form = self.get\_form(request, \*args, \*\*kwargs)

return self.render\_form(request, \*args, \*\*kwargs)

def post(self, request, \*args, \*\*kwargs):

self.form = self.post\_form(request, \*args, \*\*kwargs)

if self.form.is\_valid():

self.save\_form(request, \*args, \*\*kwargs)

self.form = self.get\_form(request, \*args, \*\*kwargs)

if self.form.is\_bound:

self.form = self.set\_validation\_attrs(self.form)

return self.render\_form(request, \*args, \*\*kwargs)

def get\_form(self, request, \*args, \*\*kwargs):

return self.form\_class()

def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST)

def save\_form(self, request, \*args, \*\*kwargs):

self.form.save()

class SignUpUserFormView(MyFormView):

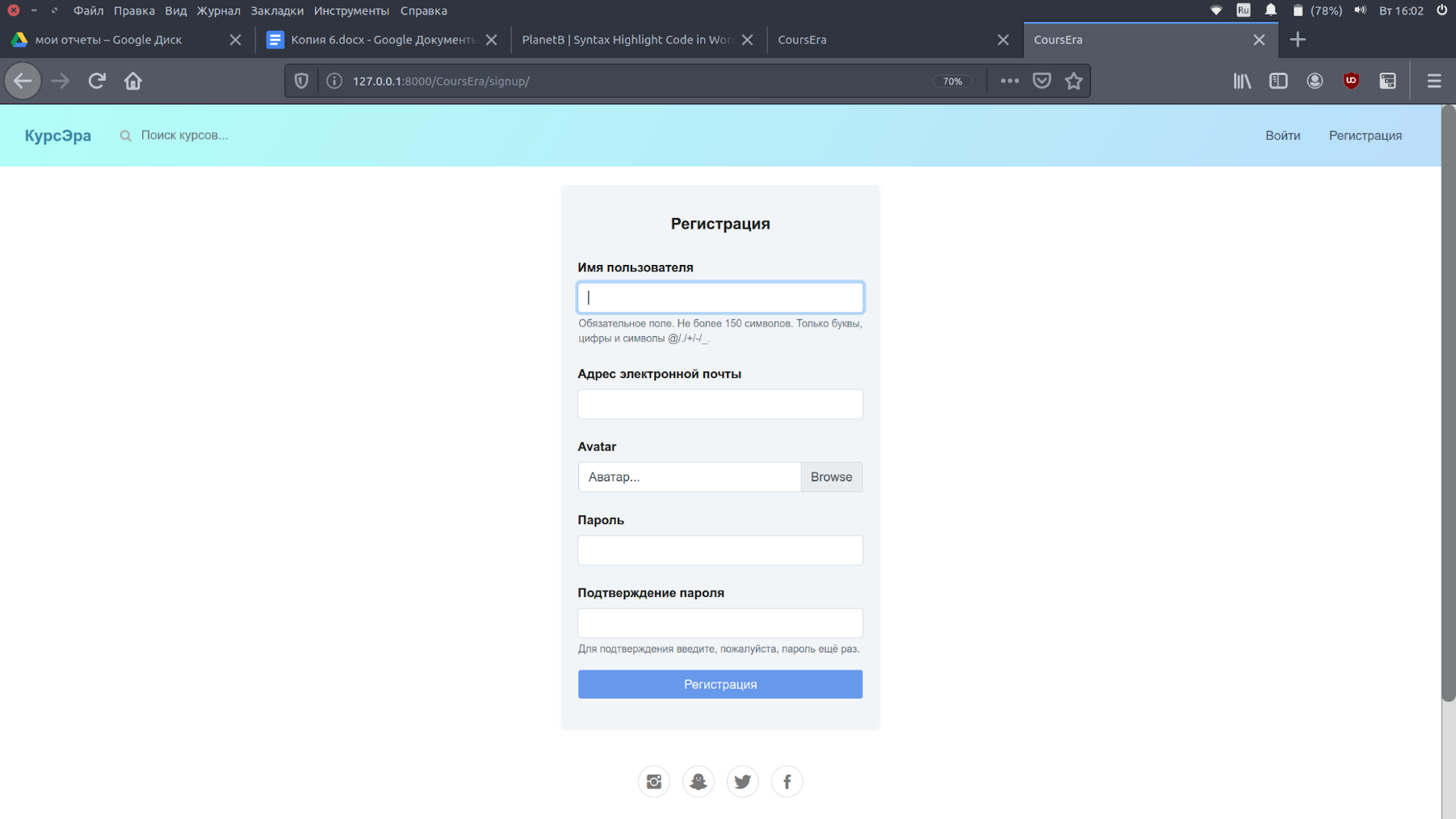
def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST, request.FILES)

def save\_form(self, request, \*args, \*\*kwargs):

messages.success(request, 'Регистрация успешна завершена!')

super().save\_form(request, \*args, \*\*kwargs)



Создайте view, которая возвращает форму для авторизации:

class MyLoginView(FormControlsStylerMixin, LoginView):

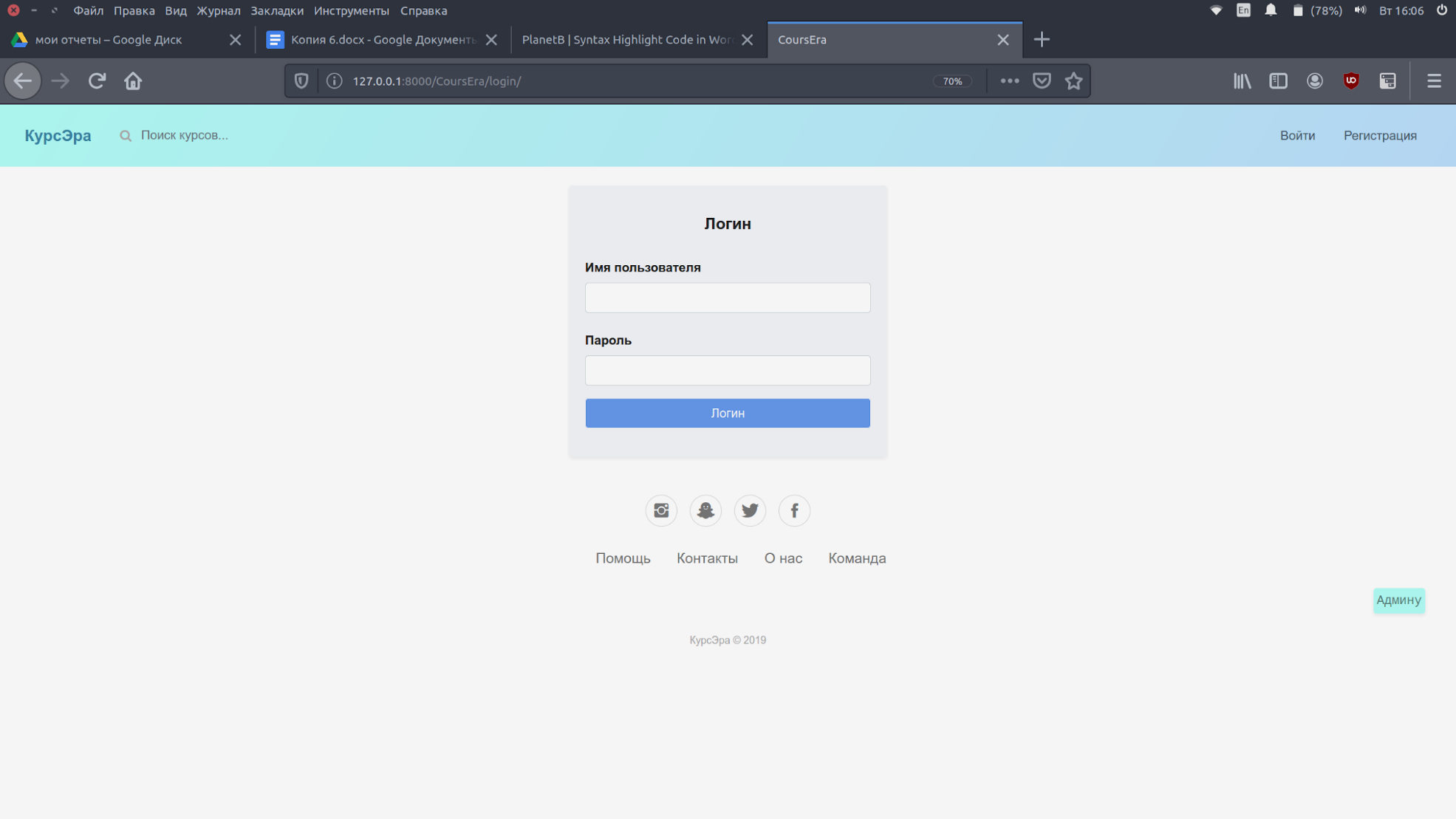
def get\_form(self, form\_class=None):

form = super().get\_form(form\_class)

if form.is\_bound:

form = self.set\_validation\_attrs(form)

return form



**Декоратор @login\_required:**

path('unenroll/<int:pk>/', login\_required(views.CourseUnenrollView.as\_view()), name='course-unenroll'),

**Добавить superuser’a через команду manage.py**

createsuperuser

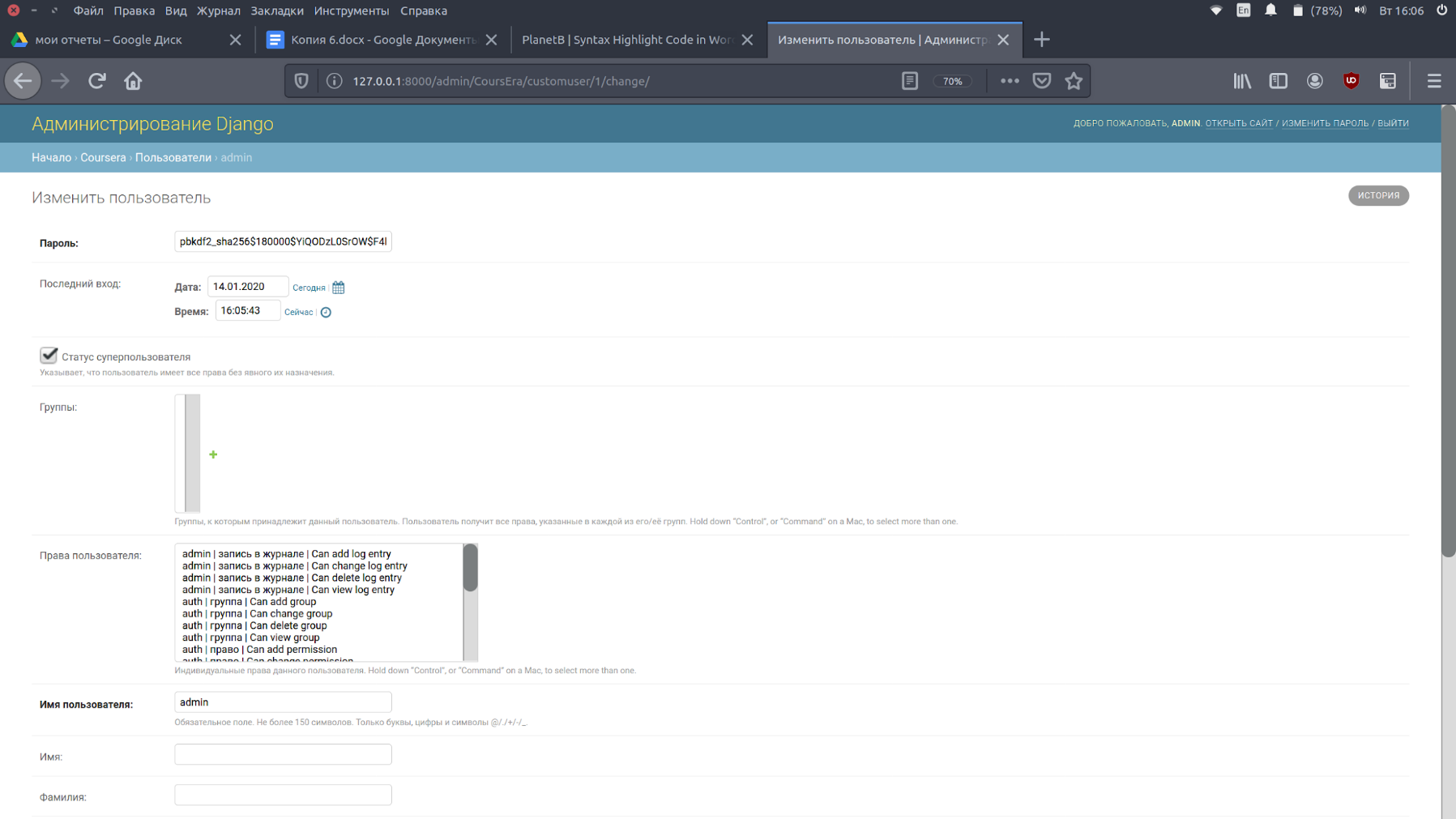
->Username: admin

->Email address: admin@gmail.com

->Password: \*\*\*\*\*\*\*\*\*\*

->Password (again): \*\*\*\*\*\*\*\*\*\*

->Superuser created successfully.



**Настройка админ-панели:**

from django.contrib import admin

from .models import Course, CustomUser

admin.site.register(Course)

admin.site.register(CustomUser)