Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



**Отчет** **по лабораторной работе №7**

**«Работа с формами, авторизация, django admin»**

**По курсу** **“Разработка интернет-приложений”**

Студент группы ИУ5-51

Жизневский П.И.

**Москва 2017**

# 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. Создайте view, которая возвращает форму для регистрации.

Поля формы:

* Логин
* Пароль
* Повторный ввод пароля
* Email
* Фамилия
* Имя

2. Создайте view, которая возвращает форму для авторизации.

Поля формы:

* Логин
* Пароль

3. При отправке формы регистрации во view проверять каждый параметр по правилам валидации, если валидация всех полей пройдена, то создавать пользователя и делать перенаправление на страницу логина, а ошибки, если они есть, выводить над формой.

Правила валидации:

* Логин не меньше 5 символов
* Пароль не меньше 8 символов
* Пароли должны совпадать
* Все поля должны быть заполнены
* Логин – уникален для каждого пользователя

1. При возникновении ошибок в момент отправки формы, введенные значения в полях ввода, кроме пароля, не должны исчезать.
2. Переписать view регистрации с использованием Django Form, правила валидации удалить из view, использовать встроенный механизм валидации полей.
3. Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.
4. Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.
5. Реализовать view для выхода из аккаунта.
6. Заменить проверку на авторизацию на декоратор login\_required
7. Добавить superuser’a через комманду manage.py
8. Подключить django.contrib.admin и войти в панель администрирования.
9. Зарегистрировать все свои модели в django.contrib.admin
10. Для выбранной модели настроить страницу администрирования:

* Настроить вывод необходимых полей в списке
* Добавить фильтры
* Добавить поиск
* Добавить дополнительное поле в список

Исходники:

Views.py:

**from** django.shortcuts **import** render  
**from** django.http **import** HttpResponseRedirect, HttpResponse  
**from** django.views.generic **import** ListView  
**from** django **import** forms  
**from** django.contrib.auth **import** authenticate,login, logout  
**from** django.contrib.auth.decorators **import** login\_required  
**from** django.contrib.auth.mixins **import** LoginRequiredMixin  
**from** .models **import** \*  
  
**class** TeacherList(LoginRequiredMixin, ListView):  
 context\_object\_name = **'teachers'** login\_url = **'/labs/'** redirect\_field\_name = **'redirect\_to'** model = Teacher  
 template\_name = **'teacher\_list.html'  
  
class** PulpitList(ListView):  
 context\_object\_name = **'pulpits'** model = Pulpit  
 template\_name = **'pulpit\_list.html'  
  
  
def** registration\_dumb(request):  
 errors = {}  
 request.encoding = **'utf-8'  
 if** request.method == **'POST'**:  
 username = request.POST.get(**'username'**)  
 **if not** username:  
 errors[**'uname'**]=**'Введите логин'  
 elif** len(username) < 5:  
 errors[**'uname'**]=**'Длина логина должна быть не меньше 5 символов'  
  
 if** User.objects.filter(username=username).exists():  
 errors[**'uname'**]=**'Такой логин уже занят'** password = request.POST.get(**'password'**)  
 **if not** password:  
 errors[**'psw'**]=**'Введите пароль'  
 elif** len(password) < 8:  
 errors[**'psw'**]=**'Длина пароля должна быть не меньше 8 символов'** password\_repeat = request.POST.get(**'password2'**)  
 **if** password != password\_repeat:  
 errors[**'psw2'**]=**'Пароли должны совпадать'** email = request.POST.get(**'email'**)  
 **if not** email:  
 errors[**'email'**]=**'Введите email'** last\_name = request.POST.get(**'last\_name'**)  
 **if not** last\_name:  
 errors[**'lname'**]=**'Введите фамилию'** first\_name = request.POST.get(**'first\_name'**)  
 **if not** first\_name:  
 errors[**'fname'**]=**'Введите имя'  
  
 if not** errors:  
 user = User.objects.create\_user(username, email, password)  
 *# usr = Userus()  
 # usr.user = user  
 # usr.first\_name = first\_name  
 # usr.last\_name = last\_name  
 # usr.save()* **return** HttpResponseRedirect(**'/labs/users'**)  
 **else**:  
 context = {**'errors'**:errors, **'username'**:username, **'email'**: email, **'last\_name'**: last\_name, **'first\_name'**: first\_name}  
 **return** render(request, **'registration\_dumb.html'**,context)  
  
 **return** render(request, **'registration\_dumb.html'**,{**'errors'**:errors})  
  
**class** RegistrationForm(forms.Form):  
 username = forms.CharField(min\_length=5,label=**'Логин'**)  
 password = forms.CharField(min\_length=8,widget=forms.PasswordInput, label=**'Пароль'**)  
 password2 = forms.CharField(min\_length=8, widget=forms.PasswordInput, label=**'Повторите ввод'**)  
 email = forms.EmailField(label=**'Email'**)  
 last\_name = forms.CharField(label=**'Фамилия'**)  
 first\_name = forms.CharField(label=**'Имя'**)  
  
**def** registration\_user(request):  
 **if** request.method == **'POST'**:  
 form = RegistrationForm(request.POST)  
 is\_val = form.is\_valid()  
 data = form.cleaned\_data  
 **if** data[**'password'**]!=data[**'password2'**]:  
 is\_val = **False** form.add\_error(**'password2'**,[**'Пароли должны совпадать'**])  
 **if** User.objects.filter(username=data[**'username'**]).exists():  
 form.add\_error(**'username'**,[**'Такой логин уже занят'**])  
 is\_val = **False  
  
 if** is\_val:  
 data = form.cleaned\_data  
 User.objects.create\_user(data[**'username'**], data[**'email'**], data[**'password'**])  
 *# usr = Userus()  
 # usr.user = user  
 # usr.first\_name = data['first\_name']  
 # usr.last\_name = data['last\_name']  
 # usr.save()* **return** HttpResponseRedirect(**'/labs/authorization'**)  
 **else**:  
 form = RegistrationForm()  
  
 **return** render(request, **'registration\_user.html'**,{**'form'**:form})  
  
@login\_required(login\_url=**'/labs/authorization'**)  
**def** success\_authorization(request):  
 **return** HttpResponseRedirect(**'/labs'**)  
  
  
**def** success\_authorization\_dumb(request):  
 **if** request.user.is\_authenticated:  
 **return** HttpResponseRedirect(**'/labs/'**)  
 **else**:  
 **return** HttpResponseRedirect(**'/labs/authorization'**)  
  
  
**def** authorization(request):  
 errors = {}  
 **if** request.method == **'POST'**:  
 username = request.POST.get(**'username'**)  
 **if not** username:  
 errors[**'uname'**]=**'Введите логин'  
 elif** len(username) < 5:  
 errors[**'uname'**]=**'Длина логина должна быть не меньше 5 символов'** password = request.POST.get(**'password'**)  
 **if not** password:  
 errors[**'psw'**]=**'Введите пароль'  
 elif** len(password) < 8:  
 errors[**'psw'**]=**'Длина пароля должна быть не меньше 8 символов'** user = authenticate(request, username=username, password=password)  
 **if** user **is None and 'uname' not in** errors.keys() **and 'psw' not in** errors.keys():  
 errors[**'login'**] = **'Логин или пароль введены неправильно'  
  
 if not** errors:  
 login(request,user)  
 *#return HttpResponseRedirect('/labs/success\_authorization\_dumb')* **return** HttpResponseRedirect(**'/labs/success\_authorization'**)  
 **else**:  
 context = {**'errors'**:errors}  
 **return** render(request, **'authorization.html'**,context)  
  
 **return** render(request, **'authorization.html'**,{**'errors'**:errors})  
  
**def** logout\_view(request):  
 logout(request)  
 **return** HttpResponseRedirect(**'/labs/'**)  
  
**class** AutorizationForm(forms.Form):  
 **pass  
  
def** index(request):  
 **return** render(request, **'test.html'**)

# Urls.py

**from** django.conf.urls **import** url, include  
**from** django.contrib **import** admin  
**from** labapp.views **import** index  
*#from labapp.views import home, labapp*urlpatterns = [  
 url(**r'^labs/'**, include(**'labapp.urls'**)),  
 url(**r'^admin/'**, admin.site.urls),  
]

# Labbapp/urls.py

**from** django.conf.urls **import** url  
  
**from** . **import** views  
  
urlpatterns = [  
 url(**r'^logout$'**, views.logout\_view, name=**'logout'**),  
 url(**r'^success\_authorization\_dumb$'**, views.success\_authorization\_dumb, name=**'success\_authorization\_dumb'**),  
 url(**r'^success\_authorization$'**, views.success\_authorization, name=**'success\_authorization'**),  
 url(**r'^authorization/$'**, views.authorization, name=**'authorization'**),  
 url(**r'^registration\_dumb/$'**, views.registration\_dumb, name=**'registration\_dumb'**),  
 url(**r'^registration\_user/$'**,views.registration\_user, name=**'registration\_user'**),  
 url(**r'^teachers/$'**,views.TeacherList.as\_view(),name=**'teacher\_list'**),  
 url(**r'^pulpit/$'**,views.PulpitList.as\_view(),name=**'pulpit\_list'**),  
 *# url(r'^reviews/$',views.ReviewList.as\_view(),name='review\_list'),* url(**r'^$'**, views.index, name=**'index'**),  
]

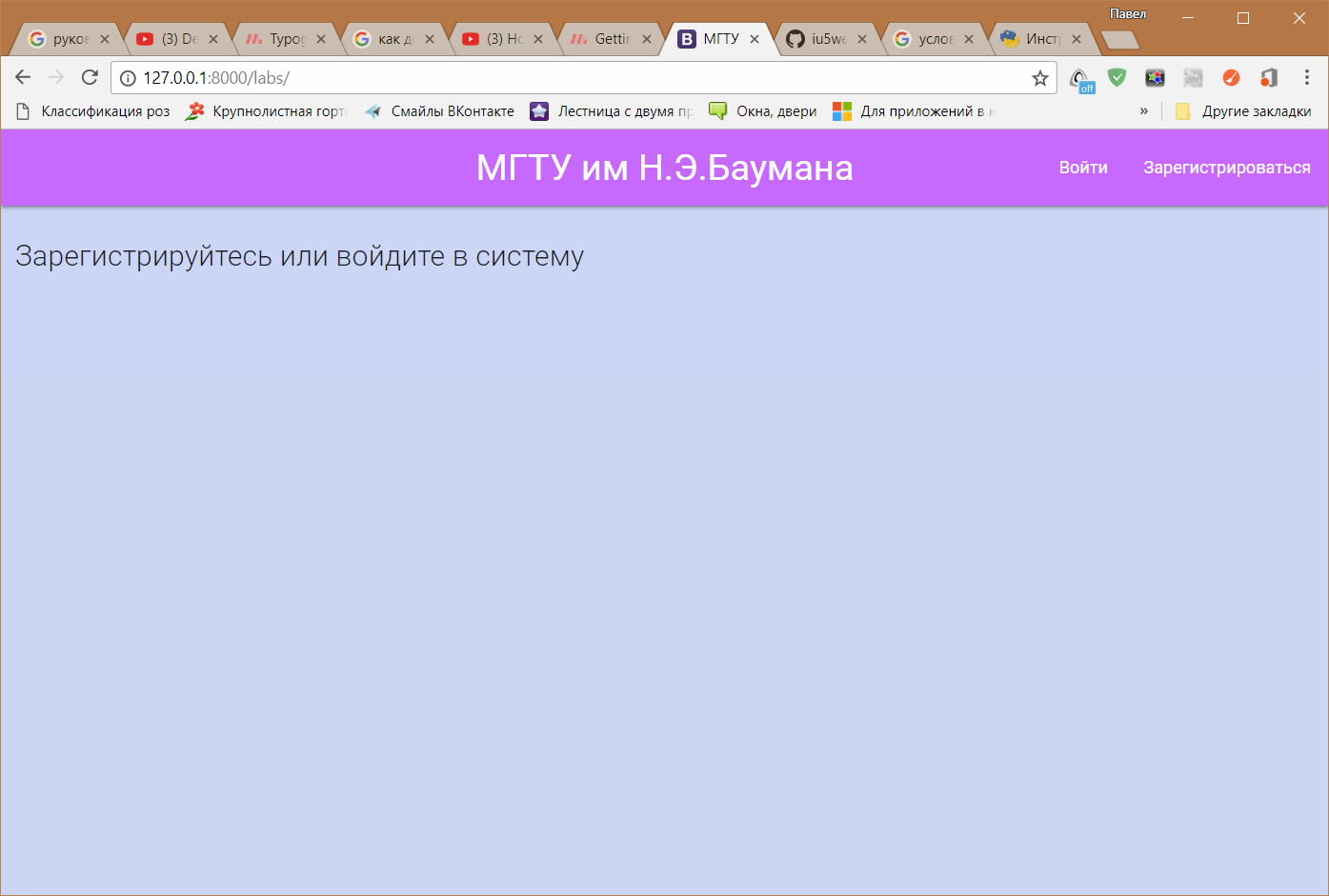
# admin.py

**from** django.contrib **import** admin  
**from** .models **import** \*  
  
*# Register your models here.***class** UserAdmin(admin.ModelAdmin):  
 *#fields = ('first\_name', 'last\_name')* list\_display = (**'username'**,**'full\_name'**,**'age'**,**'has\_reviews'**,)  
 list\_filter = (**'age'**,)  
 search\_fields = [**'last\_name'**, **'first\_name'**]  
  
 **def** full\_name(self, obj):  
 **return "{} {}"**.format(obj.last\_name, obj.first\_name)  
  
 **def** username(self, obj):  
 **return "{}"**.format(obj.user.username)  
@admin.register(Pulpit)  
**class** PulpitAdmin(admin.ModelAdmin):  
 empty\_value\_display = **'-empty-'**@admin.register(Teacher)  
**class** PulpitAdmin(admin.ModelAdmin):  
 empty\_value\_display = **'-empty-'**

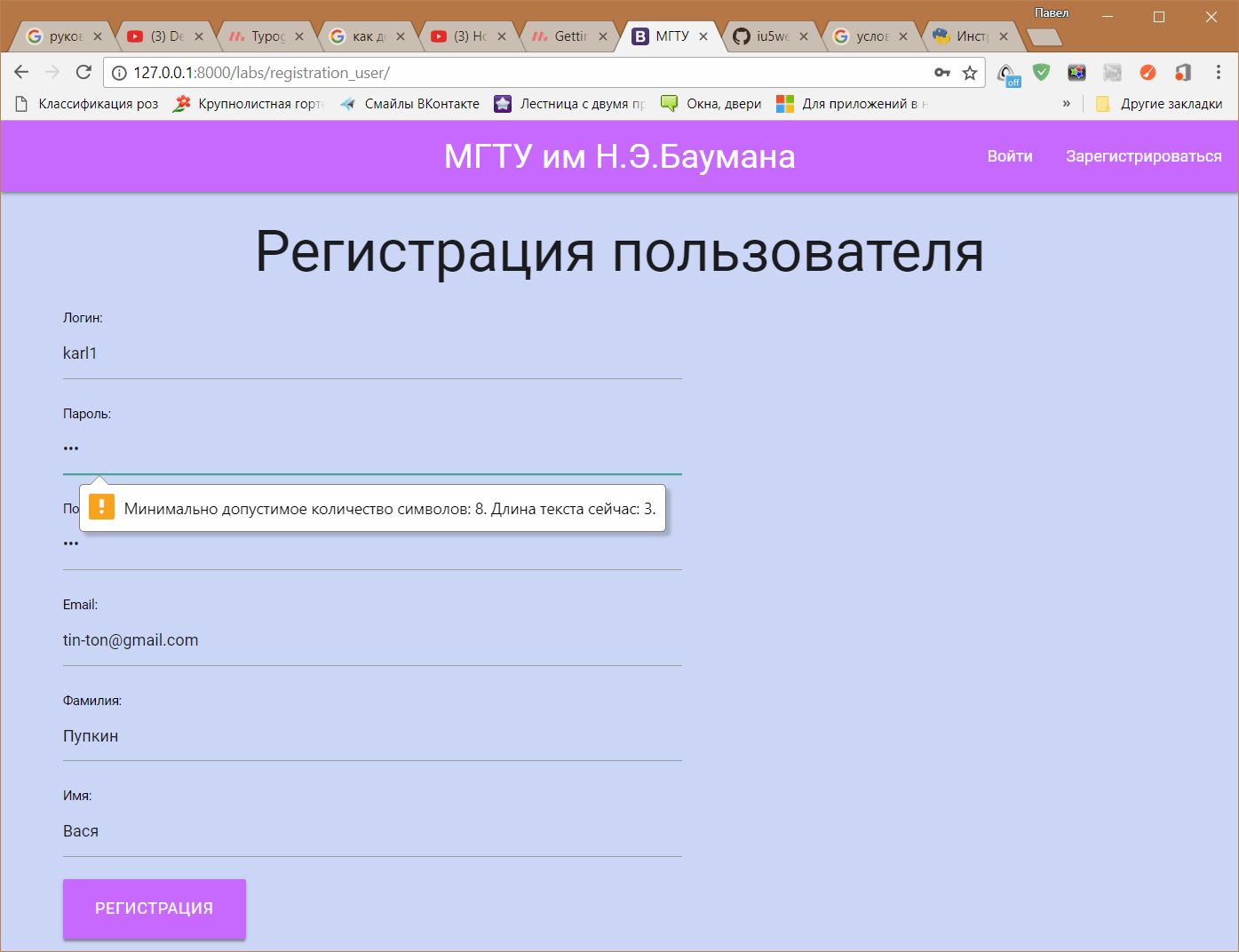
models.py

**from** django.db **import** models  
**from** django.contrib.auth.models **import** User  
**from** django.contrib **import** admin  
  
*# Create your models here.***class** Teacher(models.Model):  
 name = models.CharField(max\_length=30)  
 second\_name = models.CharField(max\_length=30)  
 third\_name = models.CharField(max\_length=30)  
 phone = models.BigIntegerField()  
 mail = models.CharField(max\_length=30)  
  
**class** Pulpit(models.Model):  
 name = models.CharField(max\_length=5)  
 year = models.IntegerField()  
 objects = models.Manager()  
  
**class** Membership(models.Model):  
 teacher = models.ForeignKey(Teacher, on\_delete=models.CASCADE)  
 pulpit = models.ForeignKey(Pulpit, on\_delete=models.CASCADE)

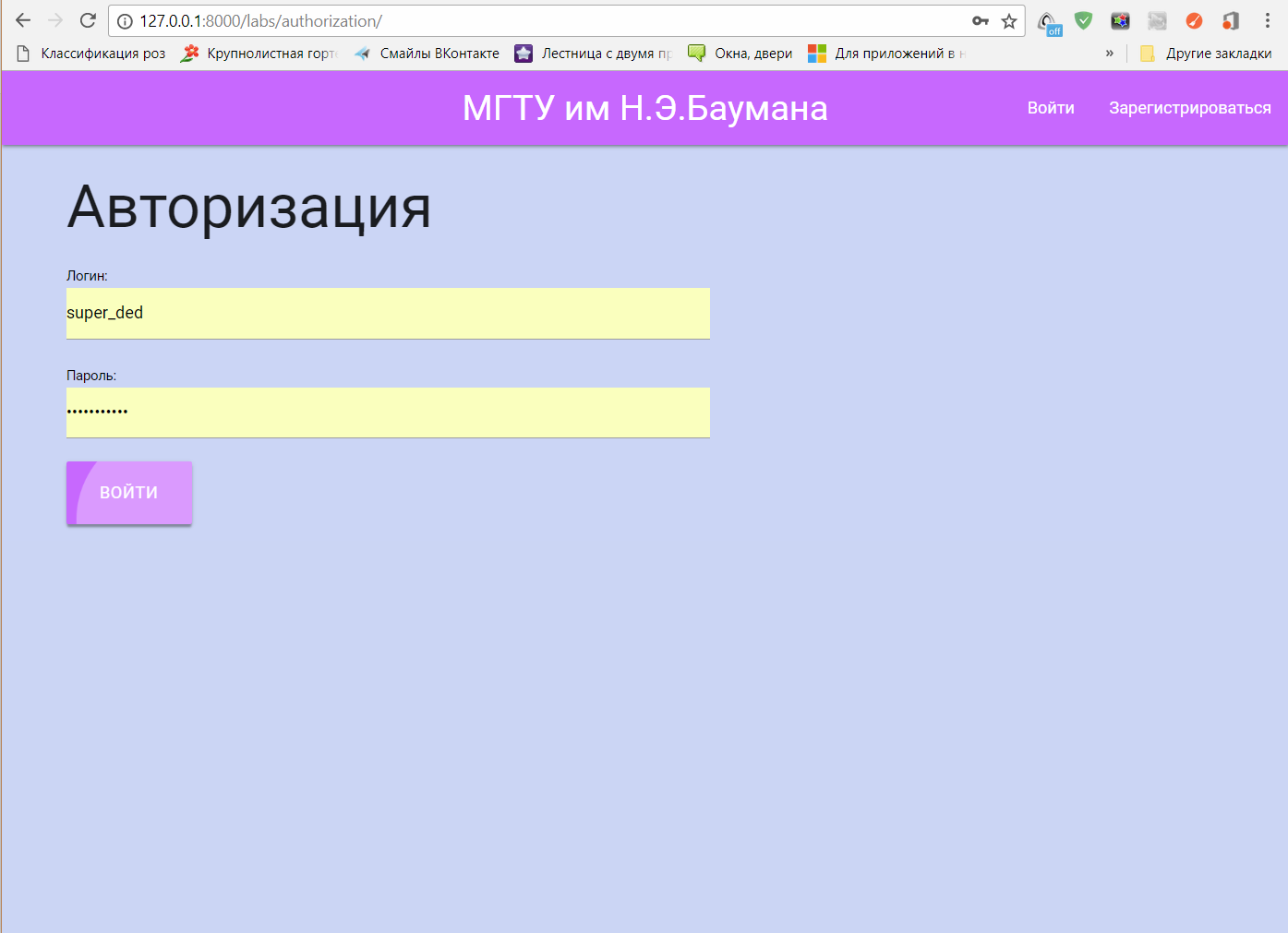
# **Стартовая страница для неавторизованных пользователей**



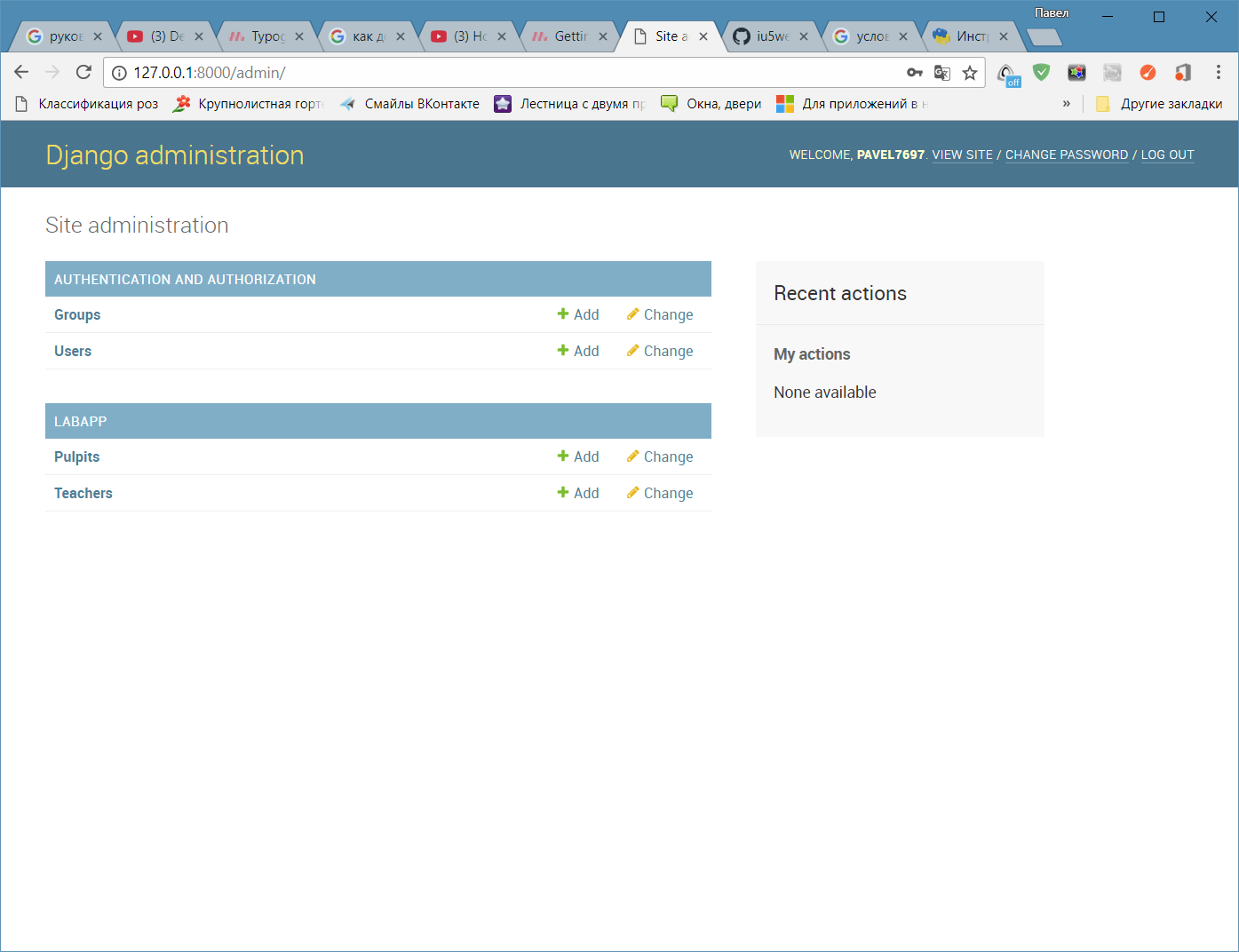
# Страница регистрации



# Форма входа



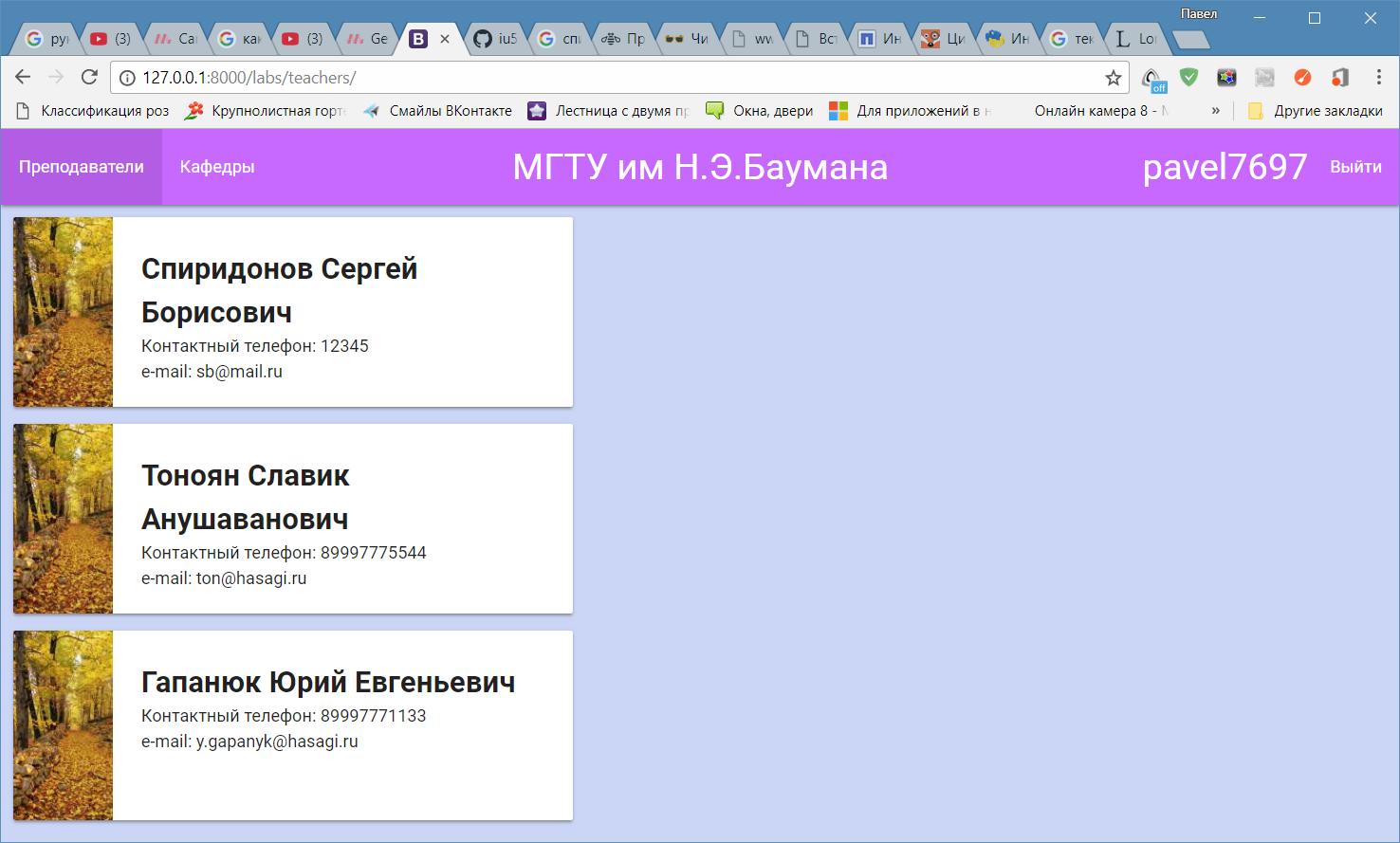
# Меню администратора



# Стартовая страница для авторизованных пользователей



# Список преподавателей



Список кафедр

