**Московский государственный университет им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:

**«Разработка Интернет Приложений»**

Jango – Admin

Формы Авторизации и регистрации

Лабораторная работа 5-8

Исполнитель:

Студент группы РТ5-51

Крутов Т.Ю.

Преподаватель:

Гапанюк Ю. Е.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2017 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. 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

4. При возникновении ошибок в момент отправки формы, введенные

значения в полях ввода, кроме пароля, не должны исчезать.

5. Переписать view регистрации с использованием Django Form, правила

валидации удалить из view, использовать встроенный механизм

валидации полей.

url.py

from django.conf.urls.static import static  
from django.conf import settings  
from django.conf.urls import url, include  
from django.contrib import admin  
from my\_app.views import \*  
  
from django.conf.urls.static import static  
from django.conf import settings  
  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^orders/', include('my\_app.urls')),  
 url(r'^orders/', OrdersView.as\_view(), name='orders\_url'),  
 url(r'^^$', main, name='main\_url'),  
 url(r'^main/', main, name='main\_url'),  
 url(r'^(?P<id>\d+)', phone\_info, name='phone\_info\_url'),  
 url(r'^db/', db, name='db\_url'),  
 url(r'^login/', login, name='login\_url'),  
 url(r'^registration/', registration, name='registration\_url'),  
 url(r'^registration2/', registration2, name='registration2\_url'),  
 url(r'^logout/', logout, name='logout\_url'),  
 url(r'^success/', success, name='success\_url'),  
 url(r'^groups/', Groups.as\_view(), name='groups\_url'),  
 url(r'^teachers/', Teachers.as\_view(), name='teachers\_url'),  
 url(r'^teachersIU', TeacherIUView.as\_view(), name='IU\_url'),  
 url(r'^teachersSM', TeacherSMView.as\_view(), name='SM\_url'),  
 url(r'^teachersRT', TeacherRTView.as\_view(), name='RT\_url'),  
 url(r'^teachersBMT', TeacherBMTView.as\_view(), name='BMT\_url'),

base.html

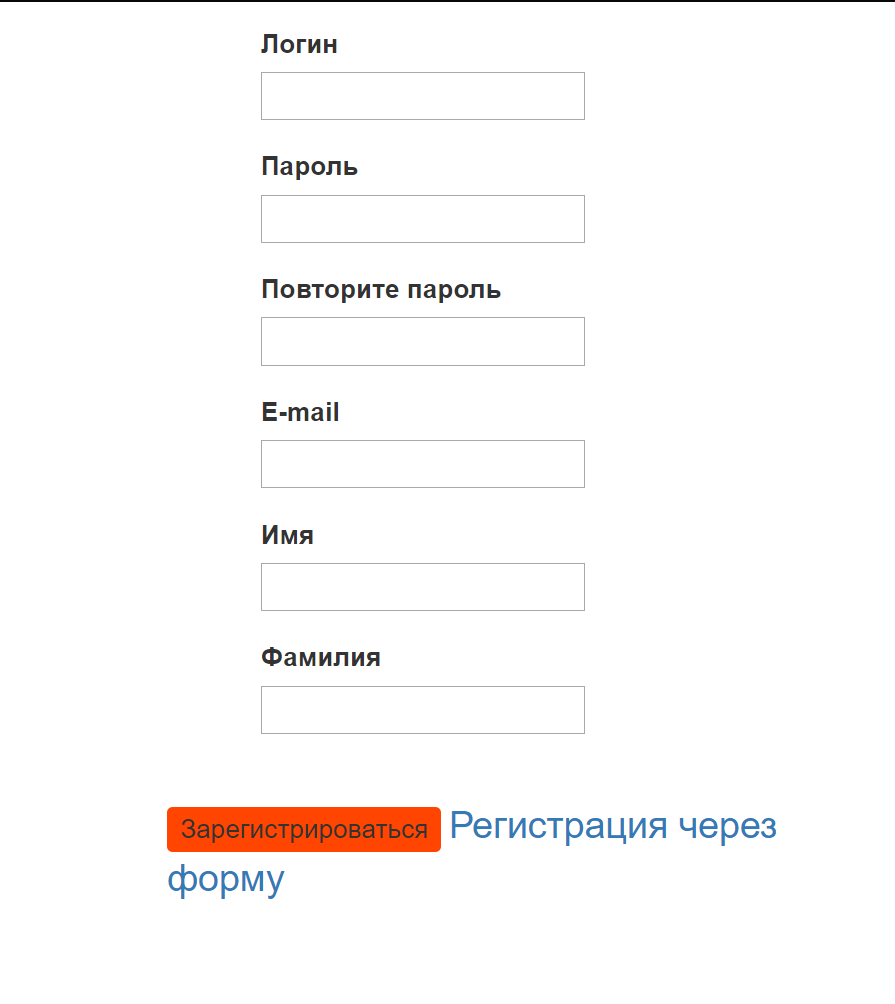
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Преподаватели</title>  
 {% load static %}  
 <link rel="stylesheet" href="{% static 'Bootstrap/css/bootstrap.css' %}">  
 <link rel="stylesheet" href="{% static 'Bootstrap/css/font-awesome.min.css' %}">  
 <link rel="stylesheet" href="{% static 'Bootstrap/css/main.css' %}">  
 <link rel="stylesheet" href="{% static 'Bootstrap/js/bootstrap.min.js' %}">  
 <link rel="icon" href="{% static '/images/logo.png' %}" type="image/x-icon">  
 <link rel="shortcut icon" href="{% static '/images/logo.png' %}" type="image/x-icon">  
{# <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js%22%3E</script>#}  
 <script type="text/javascript" src="{% static 'Bootstrap/js/my.js' %}"></script>  
 <script type="text/javascript" src="{% static 'Bootstrap/js/bootstrap.min.js' %}"></script>  
 <script type="text/javascript" src="{% static 'Bootstrap/js/jquery-3.2.1.min.js' %}"></script>  
</head>  
<body>  
 <div class="navbar navbar-inverse navbar-fixed-top">  
 <div class="container">  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <p1><img src="{% static "images/small\_4072.jpg" %}" width="23" height="25" style="margin: 13px"/></p1>  
 <a class="navbar-brand" href="#">Преподаватели</a>  
 </div>  
 <div class="navbar-collapse collapse">  
 <ul class="nav navbar-nav navbar-right">  
 <li><a href="{% url 'main\_url' %}">На главную </a></li>  
 <li><a href="{% url 'courses\_url' %}">Работа с БД </a></li>  
{# <li><a href="{% url 'orders\_url' %}">Преподаватели</a></li>#}  
 <li><a href="{% url 'db\_url' %}">Преподаватели</a></li>  
 <li>  
 <div class="dropdown">  
 <button class="dropdown-toggle" data-toggle="dropdown">  
 <img src="{% static "images/small\_4072.JPG" %}" width="30" height="30" alt=""/>  
{# <span class="caret"></span>#}  
 </button>  
 <div class="dropdown-content">  
 {% if request.user.is\_authenticated %}  
 <span>Здравствуйте, {{ request.user.username }}</span>  
 <a href="{% url 'logout\_url' %}">Выйти</a>  
 {% else %}  
 <a href="{% url 'login\_url' %}">Вход</a>  
 <a href="{% url 'registration\_url' %}">Регистрация</a>  
 <a href="{% url 'registration2\_url' %}">Регистрация2</a>  
 {% endif %}  
 </div>  
 </div>  
 </li>  
 </ul>  
 </div>  
 </div>  
 </div>  
 {% block body1 %}{% endblock %}  
</body>  
</html>

Login.html

{% extends 'base.html' %}  
  
{% block body1 %}  
 <div id="body1">  
 <div class="container">  
 <div class="row centered">  
 <div class="col-lg-8 col-lg-offset-2">  
 <h6><br>Авторизация<br><br></h6>  
 </div>  
 <form method="POST">  
 <div class="auth\_block"style="width: 352px">  
 {% csrf\_token %}  
 <div class="auth\_blank">  
 <label class="login" style="width: 60px">  
 Логин  
 </label>  
 <div class="form-block">  
 <input type="text" name="username">  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label class="login" style="width: 70px">  
 Пароль  
 </label>  
 <div class="form-block">  
 <input type="password" name="password">  
 </div>  
 </div>  
 {% if error != "" %}  
 <p>{{ error }}</p>  
 {% endif %}  
 <br>  
 <button type="submit">Войти</button>  
 </div>  
 </form>  
 </div>  
 </div>  
 </div>  
{% endblock %}

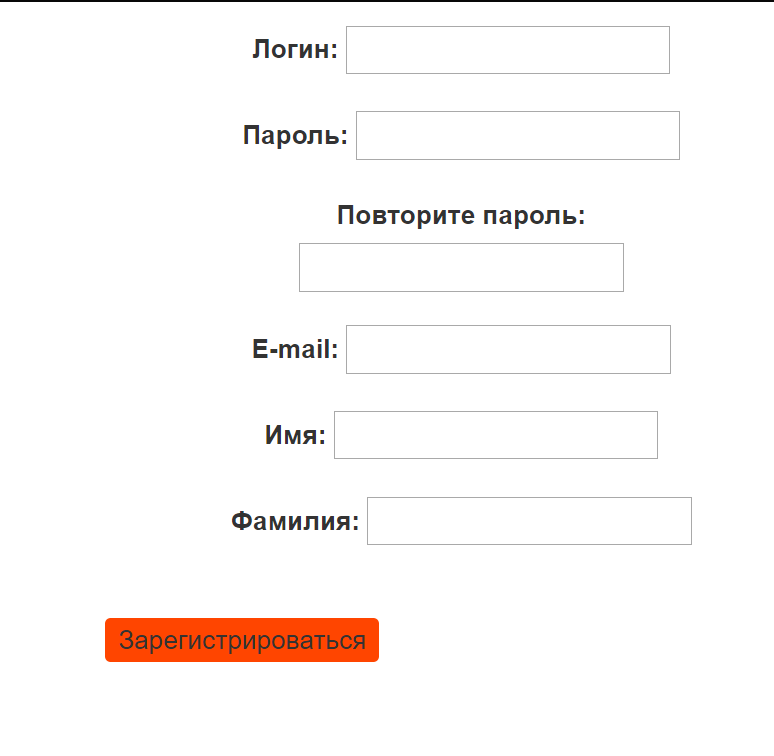
Registration.html

{% extends 'base.html' %}  
  
{% block body1 %}  
 <div id="body1">  
 <div class="container">  
 <div class="row centered">  
 <div class="col-lg-8 col-lg-offset-2">  
 <h5><br>Регистрация<br><br></h5>  
 </div>  
 <form method="POST">  
 <div class="auth\_block" style="width: 452px">  
 {% csrf\_token %}  
 <div class="auth\_blank">  
 <label>  
 Логин  
 </label>  
 <div class="form-block">  
 <input type="text" name="username" value="{{ username }}">  
 {% if errors.username != '' %}  
 <div class="form-error">{{ errors.username }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label>  
 Пароль  
 </label>  
 <div class="form-block">  
 <input type="password" name="password">  
 {% if errors.password != '' %}  
 <div class="form-error">{{ errors.password }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label>  
 Повторите пароль  
 </label>  
 <div class="form-block">  
 <input type="password" name="password2">  
 {% if errors.password2 != '' %}  
 <div class="form-error">{{ errors.password2 }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label>  
 E-mail  
 </label>  
 <div class="form-block">  
 <input type="email" name="email" value="{{ email }}">  
 {% if errors.email != '' %}  
 <div class="form-error">{{ errors.email }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label>  
 Имя  
 </label>  
 <div class="form-block">  
 <input type="text" name="firstname" value="{{ firstname }}">  
 {% if errors.firstname != '' %}  
 <div class="form-error">{{ errors.firstname }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <div class="auth\_blank">  
 <label>  
 Фамилия  
 </label>  
 <div class="form-block">  
 <input type="text" name="surname" value="{{ surname }}">  
 {% if errors.surname != '' %}  
 <div class="form-error">{{ errors.surname }}</div>  
 {% endif %}  
 </div>  
 </div>  
 <br>  
 <button type="submit">Зарегистрироваться</button>  
 <a style=" font-size: 20px; font-weight: 200; " href="{% url 'registration2\_url' %}">Регистрация через форму</a>  
 </div>  
 </form>  
 </div>  
 </div>  
  
  
 </div>  
{% endblock %}



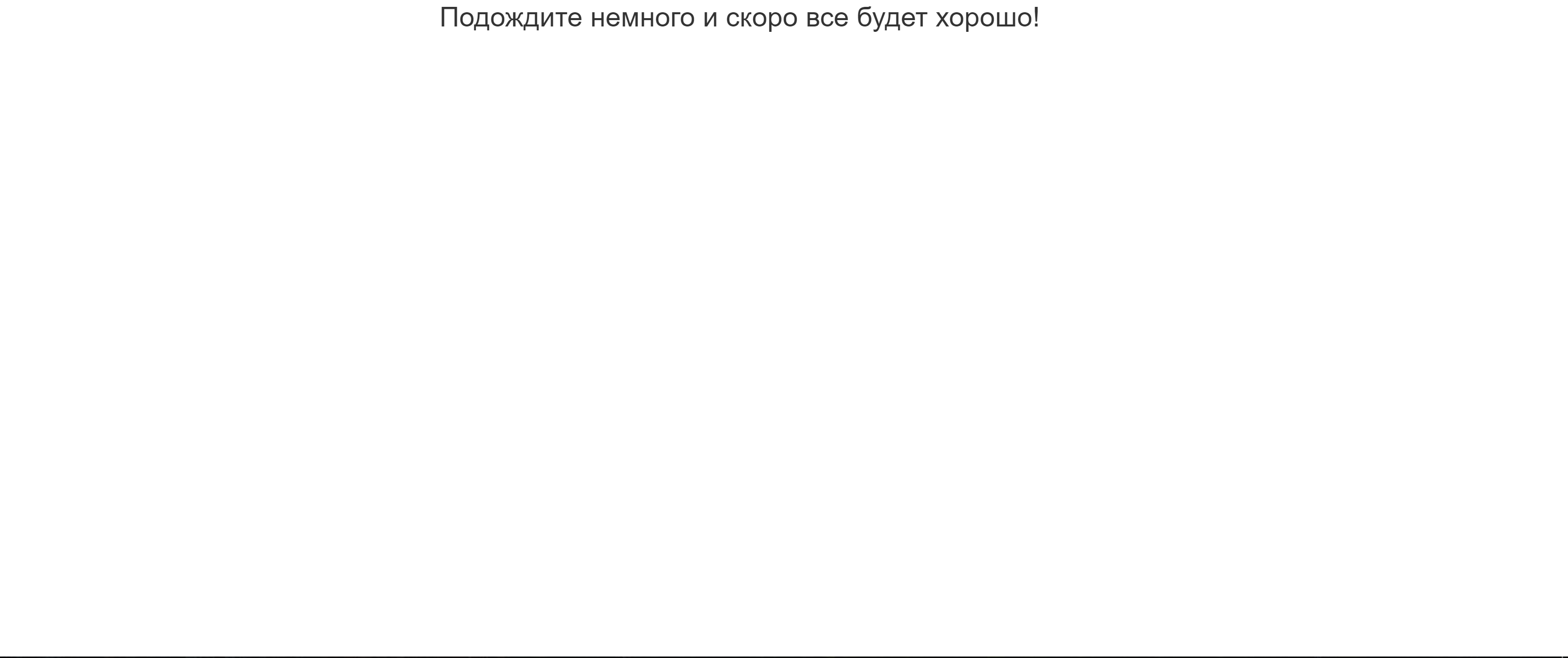
Registration2.html

{% extends 'base.html' %}  
  
{% block body1 %}  
 <div id="body1">  
 <div class="container">  
 <div class="row centered">  
 <div class="col-lg-8 col-lg-offset-2">  
 <h5><br>Регистрация (форма)<br><br></h5>  
 </div>  
 <form method= "POST">  
 <div class="auth\_block" style="width: 452px">  
 {% csrf\_token %}  
 {{ form.as\_p }}  
 <br>  
 <button type= "submit"> Зарегистрироваться </button>  
 </div>  
 </form>  
 </div>  
 </div>  
  
 </div>  
{% endblock %}



Success.html

{% extends 'base.html' %}  
  
{% block body1 %}  
 <div id="body1">  
 <div class="container">  
 <div class="row centered">  
 <div class="col-lg-8 col-lg-offset-2">  
 <h6><br>Авторизация прошла успешно!<br></h6>  
 <h2>Подождите немного и скоро все будет хорошо!<br></h2>  
 {# <a style=" font-size: 20px; font-weight: 200; " href="{% url 'x0\_url' %}">Либо нажмите на данную надпись</a> #}  
 <meta http-equiv="refresh" content="3; url=/main/">  
 </div>  
 </div>  
 </div>  
 </div>  
{% endblock %}



View.py

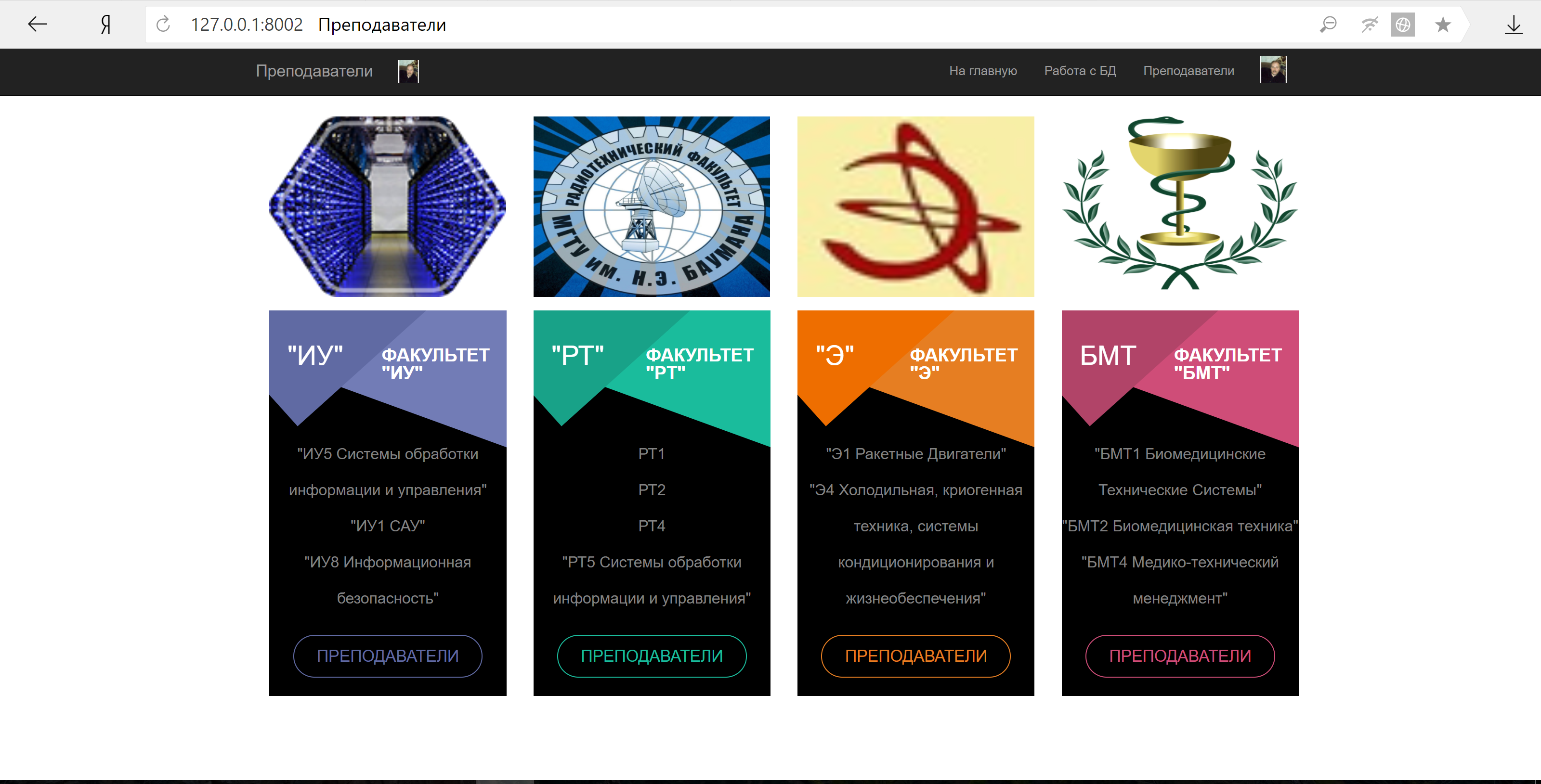
from datetime import datetime  
from .models import \*  
from my\_app.registration import \*  
from django.shortcuts import render, HttpResponseRedirect  
from django.views import View  
from django.views.generic import View, ListView  
from my\_app.forms import RegistrForm  
from datetime import datetime  
from django.contrib.auth.models import User  
from django.contrib import auth  
from django.contrib.auth import authenticate  
from django.contrib.auth.decorators import login\_required  
from django.core import serializers  
from django.http import HttpResponse  
from django.shortcuts import render, HttpResponseRedirect  
from django.urls import reverse  
# Create your views here.  
  
  
  
# Create your views here.  
  
  
class OrdersView(View):  
 def get(self, request):  
 variable = 'Django'  
 today\_date = datetime.now()  
 data = {  
 'orders': [  
 {'title': 'Первый заказ', 'id': 1},  
 {'title': 'Второй заказ', 'id': 2},  
 {'title': 'Третий заказ', 'id': 3}  
 ]  
 }  
 return render(request, 'orders.html', locals())  
  
  
class OrderView(View):  
 def get(self, request, id):  
 variable = 'Django'  
 today\_date = datetime.now()  
 data = {  
 'order': {  
 'id': id  
 }  
 }  
 return render(request, 'order.html', locals())  
  
  
def main(request):  
 return render(request, 'main.html', locals())  
  
  
def db(request):  
 return render(request, 'db.html', locals())  
  
  
class BookView(ListView):  
 model = Book  
 template\_name = 'books.html'  
  
  
class WriterView(ListView):  
 model = Writer  
 template\_name = 'writer.html'  
  
  
def IU\_info(request, id):  
 name = ['Факультет ИУ', 'Факультет РТ', 'Факультет Э',’Факультет БМТ’]  
 IU\_info = 'Ведущий факультет по подготовке кадров в области разработки программного обеспечения, электронной и микросистемной техники для предприятий ракетно-космической отрасли, государственных служб и частных компаний.'  
 RT\_info = 'Факультет был создан приказом министров высшего и среднего специального образования и радиопромышленности в мае 1956 года как филиал вечернего факультета училища по подготовке инженеров для предприятия радиопромышленности. '  
 EN\_info = ' Факультет включает 10 специализированных кафедр и проводит подготовку высококвалифицированных инженеров исследователей и разработчиков для передовых научно - исследовательских институтов, институтов Российской Академии наук, конструкторских бюро и научно - производственных объединений, занимающихся прогнозированием развития, разработкой, моделированием и созданием новых типов экологически чистых ракетно - космических двигателей и установок, энергетических систем, нетрадиционных источников энергии, поршневых и газотурбинных транспортных двигателей, систем криогеники и кондиционирования, вакуумного оборудования и пневмоавтоматики, физико - энергетических и плазменных энергетических установок, гидромашин, гидроприводов и гидропневмоавтоматики, комбинированных экобиозащитных систем.'  
 BMT\_info = 'Единственный в стране факультет «Биомедицинская техника» основан в 1998 году и осуществляет подготовку инженеров по двум специальностям: «Биотехнические системы и устройства» (19.05) «Инженерное дело в медико-биологической практике» (19.06) '  
 info = [IU\_info, RT\_info, EN\_info, BMT\_info]  
 data1 = {'phone': {'id': id}}  
 data2 = {'phones': [{'id': '1', 'phone\_name': 'Факультет ИУ', 'info': IU\_info },  
 {'id': '2', 'phone\_name': 'Факультет РТ', 'info': RT\_info},  
 {'id': '3', 'phone\_name': 'Факультет Э', 'info': EN\_info},  
 {'id': '4', 'phone\_name': 'Факультет БМТ', 'info': BMT\_info}]}  
 return render(request, 'phone\_info.html', locals())  
  
  
def login(request):  
 error = ""  
 username = None  
 password = None  
 if request.method == 'POST':  
 username = request.POST.get('username')  
 password = request.POST.get('password')  
 user = authenticate(username=username, password=password)  
 if user:  
 auth.login(request, user)  
 return HttpResponseRedirect('/success/')  
 else:  
 error = "Попробуй ещё раз"  
 return render(request, 'login.html', locals())  
  
  
def registration(request):  
 errors = {'username': '', 'password': '', 'password2': '', 'email': '', 'firstname': '', 'surname': ''}  
 error\_flag = False  
 if request.method == 'POST':  
 username = request.POST.get('username')  
 if not username:  
 errors['username'] = 'Введите логин'  
 error\_flag = True  
 elif len(username) < 5:  
 errors['username'] = 'Логин должен превышать 5 символов'  
 error\_flag = True  
 elif User.objects.filter(username=username).exists():  
 errors['username'] = 'Такой логин уже существует'  
 error\_flag = True  
 password = request.POST.get('password')  
 if not password:  
 errors['password'] = 'Введите пароль'  
 error\_flag = True  
 elif len(password) < 8:  
 errors['password'] = 'Длина пароля должна превышать 8 символов'  
 password\_repeat = request.POST.get('password2')  
 if password != password\_repeat:  
 errors['password2'] = 'Пароли должны совпадать'  
 error\_flag = True  
 email = request.POST.get('email')  
 if not email:  
 errors['email'] = 'Введите e-mail'  
 firstname = request.POST.get('firstname')  
 if not firstname:  
 errors['firstname'] = 'Введите имя'  
 surname = request.POST.get('surname')  
 if not surname:  
 errors['surname'] = 'Введите фамилию'  
 if not error\_flag:  
 user = User.objects.create\_user(username=username, password=password, email=email, first\_name=firstname,  
 last\_name=surname)  
 return HttpResponseRedirect('/login/')  
 return render(request, 'registration.html', locals())  
  
  
@login\_required()  
def success(request):  
 # if not request.user.is\_authenticated:  
 # return HttpResponseRedirect('/login/')  
 return render(request, 'success.html', locals())  
  
  
def registration2(request):  
 form = RegistrationForm(request.POST or None)  
 if request.method == 'POST':  
 if form.is\_valid():  
 user = User.objects.create\_user(username=request.POST.get('username'),  
 email=request.POST.get('email'),  
 password=request.POST.get('password'),  
 first\_name=request.POST.get('firstname'),  
 last\_name=request.POST.get('surname'))  
 return HttpResponseRedirect('/login/')  
 return render(request, 'registration2.html', {'form': form})  
  
  
def logout(request):  
 auth.logout(request)  
 return HttpResponseRedirect('/main/')  
  
  
class TeacherIUView(ListView):  
 model = teacherIU  
 template\_name = 'IU.html'  
  
  
class TeacherSMView(ListView):  
 model = teacherSM  
 template\_name = 'SM.html'  
  
  
class TeacherBMTView(ListView):  
 model = teacherBMT  
 template\_name = 'BMT.html'  
  
class TeacherRTView(ListView):  
 model = teacherRT  
 template\_name = 'RT.html'  
  
  
class Teachers(ListView):  
 model = teachers  
 template\_name = 'teachers.html'  
  
class Groups(ListView):  
 model = groups  
 template\_name = 'groups.html'

registration.py

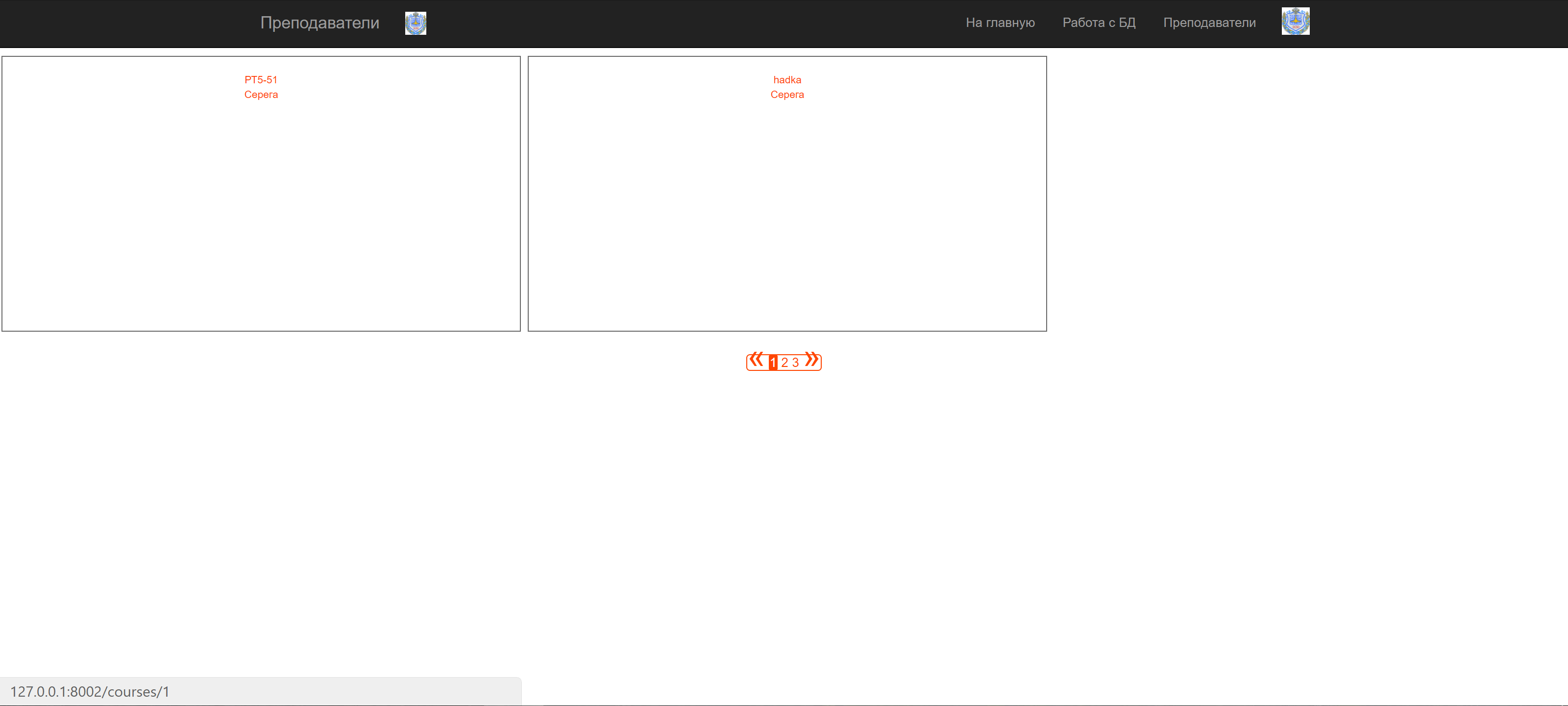
from django import forms  
from django.contrib.auth.forms import UserCreationForm  
  
  
class RegistrationForm(forms.Form):  
 username = forms.CharField(min\_length=5, label='Логин')  
 password = forms.CharField(min\_length=6, widget=forms.PasswordInput, label='Пароль')  
 password2 = forms.CharField(min\_length=6, widget=forms.PasswordInput, label='Повторите пароль')  
 email = forms.EmailField(widget=forms.EmailInput, label='E-mail')  
 firstname = forms.CharField(label='Имя')  
 surname = forms.CharField(label='Фамилия')

admin.py

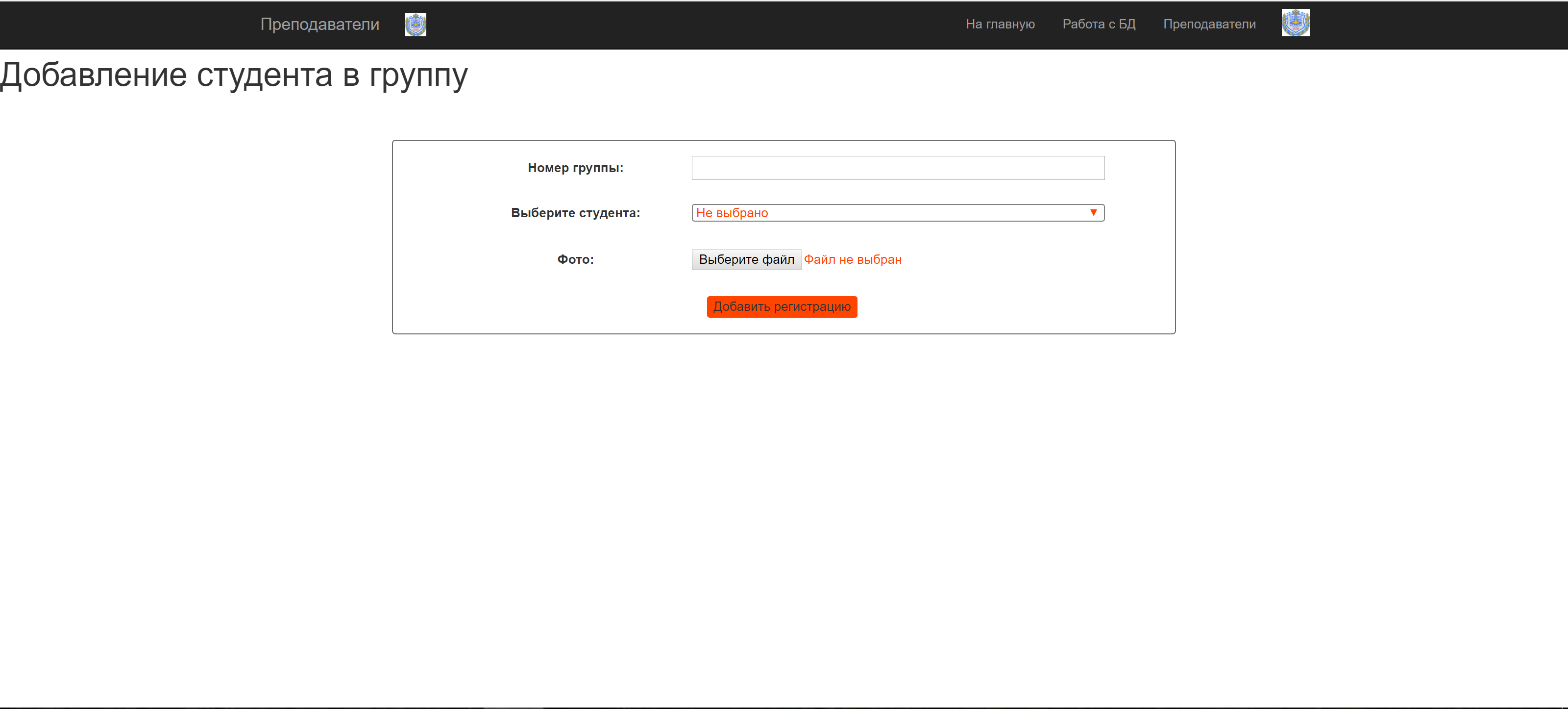
from django.contrib import admin  
from.models import \*  
  
class TeacherAdminIU(admin.ModelAdmin):  
 fields = ('teacher\_fio', 'teacher\_department', 'teacher\_f')  
 list\_filter = ('teacher\_fio', 'teacher\_department', 'teacher\_f' )  
 list\_display = ('teacher\_fio', 'teacher\_department', 'teacher\_f')  
 search\_fields = ('teacher\_fio', 'teacher\_department', 'teacher\_f')  
 list\_per\_page = 10  
  
class TeacherAdminRT(admin.ModelAdmin):  
 fields = ('teacher\_fio', 'teacher\_department')  
 list\_filter = ('teacher\_fio', 'teacher\_department')  
 list\_display = ('teacher\_fio', 'teacher\_department')  
 search\_fields = ('teacher\_fio', 'teacher\_department')  
 list\_per\_page = 10  
  
class TeacherAdminSM(admin.ModelAdmin):  
 fields = ('teacher\_fio', 'teacher\_department')  
 list\_filter = ('teacher\_fio', 'teacher\_department')  
 list\_display = ('teacher\_fio', 'teacher\_department')  
 search\_fields = ('teacher\_fio', 'teacher\_department')  
 list\_per\_page = 10  
  
class TeacherAdminBMT(admin.ModelAdmin):  
 fields = ('teacher\_fio', 'teacher\_department')  
 list\_filter = ('teacher\_fio', 'teacher\_department')  
 list\_display = ('teacher\_fio', 'teacher\_department')  
 search\_fields = ('teacher\_fio', 'teacher\_department')  
 list\_per\_page = 10  
  
class TeacherAdmin(admin.ModelAdmin):  
 fields = ('teacher\_fio', 'teacher\_department', 'group\_num')  
 list\_filter = ('teacher\_fio', 'teacher\_department', ('group\_num', admin.RelatedOnlyFieldListFilter))  
 list\_display = ('teacher\_fio', 'teacher\_department')  
 search\_fields = ('teacher\_fio', 'teacher\_department', 'group\_num')  
  
class GroupsAdmin(admin.ModelAdmin):  
 fields = ('group\_number', 'stud')  
 list\_filter = ('group\_number', 'stud')  
 list\_display = ('group\_number', 'stud')  
 search\_fields = ('group\_number', 'stud')  
  
class StudentsAdmin(admin.ModelAdmin):  
 fields = ( 'student\_fio', 'group\_number')  
 list\_filter = ( 'student\_fio', 'group\_number')  
 list\_display = ( 'student\_fio', 'group\_number')  
 search\_fields = ('student\_fio', 'group\_number')  
  
admin.site.register(Book)  
admin.site.register(Writer)  
  
admin.site.register(teachers, TeacherAdmin)  
admin.site.register(groups, GroupsAdmin)  
admin.site.register(students, StudentsAdmin)  
admin.site.register(teacherBMT, TeacherAdminBMT)  
admin.site.register(teacherRT, TeacherAdminRT)  
admin.site.register(teacherSM, TeacherAdminSM)  
admin.site.register(teacherIU, TeacherAdminIU)



Работа с базой данных и пагинация



Добавление нового объекта



Реализация связи много ко многим

class students(models.Model):  
 id\_students = models.AutoField(primary\_key=True)  
 student\_fio = models.CharField(max\_length=100, blank=False, null=False, verbose\_name='ФИО Студента')  
 group\_number = models.CharField(max\_length=100, verbose\_name='Номер группы')  
  
 def \_\_str\_\_(self):  
 return '%s' % (self.student\_fio)  
  
 # def natural\_key(self):  
 # return '%s %s %s' % (self.hotell, self.kod\_nomera, self.patronymic)  
  
 class Meta:  
 verbose\_name\_plural = "Студенты"  
 verbose\_name = "Студент"  
  
class groups(models.Model):  
 id\_groups = models.AutoField(primary\_key=True)  
 group\_number = models.CharField(max\_length=45, verbose\_name='Номер группы')  
 stud = models.ForeignKey(students, on\_delete=models.CASCADE, verbose\_name='Студент')  
  
 def \_\_str\_\_(self):  
 return self.group\_number  
  
 class Meta:  
 verbose\_name\_plural = "Группы"  
 verbose\_name = "Группа"  
  
  
class teachers(models.Model):  
 id\_teacher = models.AutoField(primary\_key=True)  
 teacher\_fio = models.CharField(max\_length=100, blank=False, null=False, verbose\_name='ФИО преподавателя')  
 teacher\_department = models.CharField(max\_length=100, verbose\_name='Кафедра')  
 group\_num = models.ManyToManyField(groups, verbose\_name='Номер группы')  
  
 def \_\_str\_\_(self):  
 return self.teacher\_fio  
  
 class Meta:  
 verbose\_name\_plural = "Преподаватели"  
 verbose\_name = "Преподаватель"  
  
 def get\_group\_num(self):  
 return [r.group\_number for r in self.group\_num.all()]  
 get\_group\_num.short\_description = 'Номер группы'