

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.

« ____ » _____

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></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">
              
              {#
                <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>
  {% 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">

```



```

        <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 %}
```

Логин

Пароль

Повторите пароль

E-mail

Имя

Фамилия

[Зарегистрироваться](#) Регистрация через
форму

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>
            </div>
        </div>
    </div>
{% endblock %}
```

```

        </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>
{% endblock %}

```

Логин:

Пароль:

Повторите пароль:

E-mail:

Имя:

Фамилия:

Зарегистрироваться

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 %}

```

Подождите немного и скоро все будет хорошо!

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)

```

← Я 127.0.0.1:8002 Преподаватели

Преподаватели На главную Работа с БД Преподаватели




"ИУ" ФАКУЛЬТЕТ "ИУ"

"ИУ5 Системы обработки информации и управления"

"ИУ1 САУ"

"ИУ8 Информационная безопасность"

ПРЕПОДАВАТЕЛИ



"РТ" ФАКУЛЬТЕТ "РТ"


РТ1

РТ2

РТ4

"РТ5 Системы обработки информации и управления"

ПРЕПОДАВАТЕЛИ




"З" ФАКУЛЬТЕТ "З"

"З1 Ракетные Двигатели"

"З4 Холодильная, криогенная техника, системы кондиционирования и жизнеобеспечения"

ПРЕПОДАВАТЕЛИ



БМТ ФАКУЛЬТЕТ "БМТ"

"БМТ1 Биомедицинские Технические Системы"

"БМТ2 Биомедицинская техника"

"БМТ4 Медико-технический менеджмент"

ПРЕПОДАВАТЕЛИ