

**Московский государственный технический университет имени Н.Э. Баумана**

**Кафедра ИУ5  
«Системы обработки информации и управления»**

**Отчет по лабораторной работе №7  
«Авторизация, работа с формами и Django Admin»  
по дисциплине «Разработка Интернет-приложений»**

**Выполнил:  
студент группы ИУ5-53  
Слимов Никита**

**Москва, 2016**

## Задание

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.

- Создайте view, которая возвращает форму для регистрации. **Поля формы:** Логин, Пароль, Повторный ввод пароля, Email, Фамилия, Имя
- Создайте view, которая возвращает форму для авторизации. **Поля формы:** Логин, Пароль
- 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
- При возникновении ошибок в момент отправки формы, введенные значения в полях ввода, кроме пароля, не должны исчезать.
- Переписать view регистрации с использованием Django Form, правила валидации удалить из view, использовать встроенный механизм валидации полей.
- Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.
- Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.
- Реализовать view для выхода из аккаунта.
- Заменить проверку на авторизацию на декоратор login\_required
- Добавить superuser'а через команду manage.py
- Подключить django.contrib.admin и войти в панель администрирования.
- Зарегистрировать все свои модели в django.contrib.admin
- Для выбранной модели настроить страницу администрирования:
- Настроить вывод необходимых полей в списке Добавить фильтры, Добавить поиск, Добавить дополнительное поле в список

## Исходный код

Файл lab/forms.py

```
from django.contrib.auth import authenticate
from django.contrib.auth.hashers import make_password
from django import forms

class SignupForm(forms.Form):
    username = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
'me', }),
        max_length=30, min_length=5, required=True, label='Login'
    )
    first_name = forms.CharField(
```

```

        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
u'Иван', }),
        max_length=30, required=True, label='Firstname'
    )
    last_name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
u'Иванов', }),
        max_length=30, required=True, label='Lastname'
    )
    email = forms.EmailField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
'me@gmail.com', }),
        required=True, max_length=254, label='E-mail'
    )
    password1 = forms.CharField(
        widget=forms.PasswordInput(attrs={'class': 'form-control',
'placeholder': '*****'}),
        min_length=8, label='Password'
    )
    password2 = forms.CharField(
        widget=forms.PasswordInput(attrs={'class': 'form-control',
'placeholder': '*****'}),
        min_length=8, label=u'Repeat password'
    )

    def clean_username(self):
        username = self.cleaned_data.get('username', '')

        try:
            u = User.objects.get(username=username)
            raise forms.ValidationError('Such user already exists')
        except User.DoesNotExist:
            return username

    def clean_password2(self):
        pass1 = self.cleaned_data.get('password1', '')
        pass2 = self.cleaned_data.get('password2', '')

        if pass1 != pass2:
            raise forms.ValidationError('Passwords do not match')

    def save(self):
        data = self.cleaned_data
        password = data.get('password1')
        u = User()

        u.username = data.get('username')
        u.password = make_password(password)
        u.email = data.get('email')
        u.first_name = data.get('first_name')
        u.last_name = data.get('last_name')
        u.is_active = True
        u.is_superuser = False
        u.save()

        return authenticate(username=u.username, password=password)

class SigninForm(forms.Form):
    login = forms.CharField(
        widget=forms.TextInput(
            attrs={'class': 'form-control', 'placeholder': 'me', }

```

```

    ),
    max_length=30,
    label='Login'
)
password = forms.CharField(
    widget=forms.PasswordInput(
        attrs={'class': 'form-control', 'placeholder': '*****', }
    ),
    label='Passsword'
)

def clean(self):
    data = self.cleaned_data
    user = authenticate(username=data.get('login', ''),
password=data.get('password', ''))

    if user is not None:
        if user.is_active:
            data['user'] = user
        else:
            raise forms.ValidationError('The user is not active')
    else:
        raise forms.ValidationError('Wrong username or password')

```

Файл lab/models.py

```

from django.db import models
from django.contrib import admin
from django.contrib.auth.models import User

```

```

class Tutor(models.Model):
    lastname = models.CharField(max_length=50)
    firstname = models.CharField(max_length=50)
    middlename = models.CharField(max_length=50)
    birthday = models.DateField(null=True, blank=True)
    sex = models.BooleanField(default=True)

    def __str__(self):
        return 'Tutor: {} {} {}'.format(self.lastname, self.firstname,
self.middlename)

```

```

class SexListFilter(admin.SimpleListFilter):
    title = 'Sex'
    parameter_name = 'sex'

    def lookups(self, request, model_admin):
        return [
            (True, 'Male'),
            (False, 'Female')
        ]

    def queryset(self, request, queryset):
        if self.value() is not None:
            return queryset.filter(sex=self.value())
        return queryset

```

```

@admin.register(Tutor)
class TutorAdmin(admin.ModelAdmin):
    list_display = ('lastname', 'firstname', 'middlename', 'birthday',

```

```

'sex_str', )
search_fields = ('lastname',)
list_filter = (SexListFilter,)

def get_ordering(self, request):
    return ['birthday']

def sex_str(self, obj):
    return 'Male' if obj.sex else 'Female'

sex_str.short_description = 'Sex'

class Course(models.Model):
    name = models.CharField(max_length=100)
    full_name = models.CharField(max_length=255)
    tutor = models.ForeignKey(Tutor)

    def __str__(self):
        return 'Course: {}'.format(self.name)

@admin.register(Course)
class CourseAdmin(admin.ModelAdmin):
    list_display = ('name', 'tutor', 'my_field',)
    readonly_fields = ('my_field',)
    search_fields = ('name', 'full_name',)

    def my_field(self, obj):
        return "Some custom value for course {}".format(obj.name)

    def get_ordering(self, request):
        return ['-name']

    def get_list_filter(self, request):
        return ['tutor']

```

Файл lab/views.py

```

from django.contrib.auth.decorators import login_required
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.views import View
from django.contrib import auth

from lab.models import Tutor, Course
from lab.forms import SignupForm, SigninForm

def main(request):
    tutors = Tutor.objects.all()

    return render(request, 'main.html', {
        'tutors': tutors
    })

def signup(request):
    if request.user.is_authenticated():
        return HttpResponseRedirect('/')

```

```

    if request.method == "POST":
        form = SignupForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth.login(request, user)
            return HttpResponseRedirect('/')
    else:
        form = SignupForm()

    return render(request, 'signup.html', {
        'form': form
    })

def signin(request):
    redirect = request.GET.get('continue', '/success')
    if request.method == "POST":
        form = SigninForm(request.POST)

        if form.is_valid():
            auth.login(request, form.cleaned_data['user'])
            return HttpResponseRedirect(redirect)
    else:
        form = SigninForm()
    return render(request, 'signin.html', {
        'form': form
    })

@login_required(redirect_field_name='continue')
def login_success(request):
    return render(request, 'success.html')

def logout(request):
    redirect = request.GET.get('continue', '/')
    auth.logout(request)
    return HttpResponseRedirect(redirect)

```

```

class TutorView(View):
    def get(self, request, id):
        tutor = Tutor.objects.get(id=int(id))

        courses = Course.objects.filter(tutor=tutor).all()

        return render(request, 'tutor.html', {
            'tutor': tutor,
            'courses': courses
        })

```

Файл templates/layout.html

```

{% load static %}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1,
shrink-to-fit=no">
    <meta name="description" content="">

```

```

<meta name="author" content="">

<title>{% block title %}{% endblock %}</title>

<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
<link href="{% static 'css/bootstrap-theme.min.css' %}" rel="stylesheet">
<link href="{% static 'css/my.css' %}" rel="stylesheet">
</head>

<body>
<nav class="navbar navbar-default navbar-static-top">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand" href="{% url 'main' %}">Project</a>
    </div>
  </div>
</nav>

<div class="jumbotron">
  <div class="container">
    <h1 class="display-3">{% block title_visible %}Hello, world!{%
endblock %}</h1>
  </div>
  <p class="container">
    {% if request.user.is_authenticated %}
      You are authorized as {{ request.user.last_name }} {{
request.user.first_name }}. <a href="{% url 'logout' %}?continue={{
request.path }}">Logout</a>
    {% else %}
      <a href="{% url 'signin' %}">Sign in</a> or
      <a href="{% url 'signup' %}">Sign up</a>
    {% endif %}
  </p>
</div>

<div class="container">
  {% block body %}{% endblock %}

  <hr>

  <footer>
    <p>&copy; BMSTU 2016</p>
  </footer>
</div>

<script src="{% static 'js/jquery-3.1.1.slim.min.js' %}"></script>
<script src="{% static 'js/tether.min.js' %}"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>
</body>
</html>

```

Файл templates/main.html

```

{% extends 'layout.html' %}

{% block title %}
  Main page
{% endblock %}

{% block title_visible %}
  Main
{% endblock %}

```

```
{% block body %}
    {% for t in tutors %}
        {% if forloop.counter|divisibleby:2 %}
            <div class="row">
                {% endif %}
            <div class="col-md-6">
                {% include 'tutor_short.html' with tutor=t %}
            </div>
            {% if forloop.counter|divisibleby:2 or forloop.last %}
                </div>
            {% endif %}

            {% empty %}
                The list is empty.
            {% endfor %}

{% endblock %}
```

Файл templates/signin.html

```
{% extends 'layout.html' %}

{% block title %}
    Sign in
{% endblock %}

{% block title_visible %}
    Sign in form
{% endblock %}

{% block body %}
    <div class="clearfix">
        <div class="signin-form-error">
            {% for e in form.non_field_errors %}
                <div class="alert alert-danger" role="alert">
                    <span class="glyphicon glyphicon-exclamation-sign" aria-
hidden="true"></span>
                    {{ e }}
                </div>
            {% endfor %}
        </div>
        <div class="signin-form clearfix">
            <form class="form-horizontal" method="post" action="">
                {% csrf_token %}
                <div class="form-group clearfix">
                    <label for="login" class="col-sm-2 control-label">{{
form.login.label }}</label>
                    <div class="col-sm-5"> {{ form.login }} </div>
                </div>
                <div class="form-group clearfix">
                    <label for="password" class="col-sm-2 control-label">{{
form.password.label }}</label>
                    <div class="col-sm-5"> {{ form.password }} </div>
                </div>
                <div class="form-group">
                    <div class="col-sm-offset-2 col-sm-2">
                        <button type="submit" class="btn btn-primary">Sign
in</button>
                    </div>
                    <div class="col-sm-5">
```



```

                                <a href="{% url 'signup' %}" class="btn btn-default"
style="float: right">Sign up</a>
                                </div>
                            </div>
                        </form>
                    </div>
                </div>
            </div>
        {% endblock %}

Файл templates/signup.html
{% extends 'layout.html' %}

{% block title %}
    Sign up
{% endblock %}

{% block title_visible %}
    Sign up form
{% endblock %}

{% block body %}
    <div class="clearfix">
        {% for e in form.non_field_errors %}
            <div class="form-error">
                <div class="alert alert-danger" role="alert">
                    <span class="glyphicon glyphicon-exclamation-sign" aria-
hidden="true"></span>
                    {{ e }}
                </div>
            </div>
        {% endfor %}
        <div class="signup-form">
            <form class="form-horizontal" method="post" action="">
                {% csrf_token %}
                {% for field in form %}
                    <div class="clearfix form-group{% if field.errors %} has-
error{% endif %}">
                        <label for="login" class="col-sm-3 control-label">{{
field.label }}</label>
                        <div class="col-sm-6">
                            {{ field }}
                            {% if field.errors %} {% for e in field.errors
%}{{ e }}{% endfor %}
                            {% endif %}
                        </div>
                    </div>
                {% endfor %}
                <div class="form-group">
                    <div class="col-sm-offset-3 col-sm-6">
                        <button type="submit" class="btn btn-primary">Sign
up!</button>
                    </div>
                </div>
            </form>
        </div>
    </div>
{% endblock %}

```

Файл templates/success.html

```
{% extends 'layout.html' %}

{% block title %}
    Success
{% endblock %}

{% block title_visible %}
    Success
{% endblock %}

{% block body %}
    You successfully signed in.
{% endblock %}
```

Файл templates/tutor.html

```
{% extends 'layout.html' %}

{% block title %}
    Tutor {{ tutor.lastname }} {{ tutor.firstname }} {{ tutor.middlename }}
{% endblock %}

{% block title_visible %}
    Tutor {{ tutor.lastname }} {{ tutor.firstname }} {{ tutor.middlename }}
{% endblock %}

{% block body %}
    Tutor {{ tutor.lastname }} {{ tutor.firstname }} {{ tutor.middlename }}
    <br/>
    {{ tutor.birthday }}
    {{ tutor.sex }}

    <br/>
    <br/>

    Courses:
    <ul>
        {% for c in courses %}
            <li>{{ c.name }} ({{ c.full_name }})</li>
        {% endfor %}
    </ul>
{% endblock %}
```

Файл templates/tutor\_short.html

```
{% with id=tutor.id %}
<h2><a href="{% url 'tutor' id=id %}">{{ tutor.lastname }}</a></h2>
<p>{{ tutor.lastname }} {{ tutor.firstname }} {{ tutor.middlename }}</p>
<p><a class="btn btn-secondary" href="{% url 'tutor' id=tutor.id %}"
role="button"> &rarr; </a></p>
{% endwith %}
```

Результат

Select tutor to change | Django

Nikita Slimov

127.0.0.1:8000/admin/lab/tutor/?

Django administration

WELCOME, ADMIN. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Lab > Tutors

Select tutor to change

ADD TUTOR +

Search

Action: 

-----

Go

 0 of 10 selected

<input type="checkbox"/>	LASTNAME	FIRSTNAME	MIDDLENAME	BIRTHDAY	SEX
<input type="checkbox"/>	Lastname 7	Firstname 7	Middlename 7	Sept. 1, 1951	Male
<input type="checkbox"/>	Lastname 8	Firstname 8	Middlename 8	Sept. 1, 1966	Male
<input type="checkbox"/>	Lastname 2	Firstname 2	Middlename 2	Sept. 1, 1968	Female
<input type="checkbox"/>	Lastname 3	Firstname 3	Middlename 3	Sept. 1, 1969	Male
<input type="checkbox"/>	Lastname 1	Firstname 1	Middlename 1	Sept. 1, 1976	Female
<input type="checkbox"/>	Lastname 0	Firstname 0	Middlename 0	Sept. 1, 1980	Male
<input type="checkbox"/>	Lastname 9	Firstname 9	Middlename 9	Sept. 1, 1984	Male
<input type="checkbox"/>	Lastname 5	Firstname 5	Middlename 5	Sept. 1, 1987	Male
<input type="checkbox"/>	Lastname 6	Firstname 6	Middlename 6	Sept. 1, 1992	Male
<input type="checkbox"/>	Lastname 4	Firstname 4	Middlename 4	Sept. 1, 1999	Male

FILTER

By Sex

All

Male

Female

Select tutor to change | Django

Nikita Slimov

127.0.0.1:8000/admin/lab/tutor/?sex=False

Django administration

WELCOME, **ADMIN** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Lab > Tutors

Select tutor to change

ADD TUTOR +

Q |

Search

2 results (10 total)

Action: 

-----

Go

 0 of 2 selected

<input type="checkbox"/>	LASTNAME	FIRSTNAME	MIDDLENAME	BIRTHDAY	SEX
<input checked="" type="checkbox"/>	Lastname 2	Firstname 2	Middlename 2	Sept. 1, 1968	Female
<input type="checkbox"/>	Lastname 1	Firstname 1	Middlename 1	Sept. 1, 1976	Female

2 tutors

FILTER

By Sex

All

Male

Female

Sign in

Nikita Slimov

127.0.0.1:8000/signin

Project

Sign in form

[Sign in](#) or [Sign up](#)

Login

admin

Passsword

\*\*\*\*\*

Sign in

Sign up

© BMSTU 2016

Sign in

127.0.0.1:8000/signin?continue=/success

Project

# Sign in form

[Sign in](#) or [Sign up](#)

Wrong username or password

Login

Password

© BMSTU 2016

Sign up

127.0.0.1:8000/signup?continue=/

Project

# Sign up form

[Sign in](#) or [Sign up](#)

Login

Firstname

Lastname

E-mail

Password

Repeat password

Sign up

Nikita Slimov

127.0.0.1:8000/signup

Sign up form

[Sign in](#) or [Sign up](#)

Login

admin

Such user already exists

Firstname

Me

Lastname

Me

E-mail

me@me.ru

Password

\*\*\*\*\*

Repeat password

\*\*\*\*\*

Sign up!

© BMSTU 2016