

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления» Кафедра «Системы обработки  
информации и управления»



**Отчет**  
**по лабораторной работе №7**  
**Авторизация, работа с формами и Django Admin.**  
**По курсу**  
**“Разработка интернет-приложений”**

**Выполнил:**  
**Студент группы ИУ5-51**  
**Колосков А.С.**

# 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.

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.

6. Во view авторизации реализовать логин при POST запросе. При успешной авторизации должен происходить переход на страницу успешной авторизации.

7.Страница успешной авторизации должна проверять, что пользователь авторизован. Иначе делать перенаправление на страницу авторизации.

8.Реализовать view для выхода из аккаунта.

9.Заменить проверку на авторизацию на декоратор login\_required

10.Добавить superuser'a через команду manage.py

11.Подключить django.contrib.admin и войти в панель администрирования.

12.Зарегистрировать все свои модели в django.contrib.admin

13.Для выбранной модели настроить страницу администрирования:

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

Views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect

from .models import *
from django import forms
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required

def main(request):
    par = {
        'header': 'Main page'
    }
```

```

    return render(request, 'MainPage.html', context=par)

def user_registered(request):
    return render(request, 'registered.html')

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 ввод')
    last_name = forms.CharField(label='Фамилия')
    first_name = forms.CharField(label='Имя')
    email = forms.EmailField(label='Email')

def registration_1(request):
    form = RegistrationForm(request.POST or None)
    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:
            new_user = User.objects.create_user(data['username'],
data['email'], data['password'])
            print(new_user)
            user1 = User1()
            user1.user1 = new_user
            user1.email = data['email']
            user1.last_name = data['last_name']
            user1.first_name = data['first_name']
            user1.save()
            return HttpResponseRedirect('/registered')
        else:
            form = RegistrationForm()
    return render(request, 'registration_1.html', {'form': form})

def registration_form(request):
    errors = {}
    if request.method == 'POST':
        last_name = request.POST.get('last_name')
        if not last_name:
            errors['last_name'] = 'Введите фамилию'

        first_name = request.POST.get('first_name')
        if not first_name:
            errors['first_name'] = 'Введите имя'

        email = request.POST.get('Email')
        if not email:
            errors['Email'] = 'Введите Email'

        username = request.POST.get('username')
        if not username:
            errors['username'] = 'Введите логин'
        elif len(username) < 8:
            errors['username'] = 'Длина логина должна превышать 5 символов'
        if User.objects.filter(username=username).exists():
            errors['username'] = 'Данный логин занят'

```

```

password = request.POST.get('password')
if not password:
    errors['password'] = 'Введите пароль'
elif len(password) < 8:
    errors['password'] = 'Длина пароля должна превышать 6 символов'
password_repeat = request.POST.get('password2')
if password != password_repeat:
    errors['password_repeat'] = 'Пароли не совпадают'
print(username, password, "1")

if not errors:
    new_user = User.objects.create_user(username, email, password)
    print(new_user)
    user1 = User1()
    user1.user1 = new_user
    user1.email = email
    user1.last_name = last_name
    user1.first_name = first_name
    user1.save()
    return HttpResponseRedirect('/registered')
else:
    context = {'errors': errors, 'username': username, 'email': email,
'last_name': last_name,
                'first_name': first_name, 'password': password,
'password_repeat': password_repeat}
    return render(request, 'registration.html', context)
return render(request, 'registration.html', {'errors': errors})

class LoginForm(forms.Form):
    username = forms.CharField(label='Логин')
    password = forms.CharField(widget=forms.PasswordInput, label='Пароль')

def log_in(request):
    errors = {}
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['username'] = 'Введите логин'
        elif len(username) < 5:
            errors['username'] = 'Слишком короткий логин. Минимальная длина-5
знаков'

        password = request.POST.get('password')
        if not password:
            errors['password'] = 'Введите пароль'
        elif len(password) < 8:
            errors['password'] = 'Слишком короткий пароль. Минимальная длина-8
знаков'

        user = authenticate(request, username=username, password=password)
        print(user)
        if user is None and 'username' not in errors.keys() and 'password' not
in errors.keys():
            errors['login'] = 'Логин или пароль введены неверно'

        if not errors:
            login(request, user)
            return HttpResponseRedirect('/logged_in')
        else:
            context = {'errors': errors}
            return render(request, 'UserLogin.html', context)
    return render(request, 'UserLogin.html', {'errors': errors})

def log_in1(request):

```

```

if request.method == 'POST':
    form = LoginForm(request.POST)
    print(form)
    data = form.cleaned_data

    if form.is_valid():
        user = authenticate(request, username=data['username'],
password=data['password'])
        print(len(data['username']), len(data['password']))
        if user is not None:
            login(request, user)
            return HttpResponseRedirect('/logged_in')
        else:
            form.add_error('username', ['Неверный логин или пароль'])
    else:
        form = LoginForm()
return render(request, 'UserLogin_1.html', {'form': form})

```

```

def logout_view(request):
    logout(request)
    return render(request, 'MainPage.html')

```

```

@login_required(login_url='/login2')
def logged_in(request):
    return render(request, 'loggedin.html')

```

```

def logged_in_view(request):
    if request.user.is_authenticated:
        return render(request, 'loggedin.html')
    else:
        return HttpResponseRedirect('/login1')

```

Urls.py

```

from django.conf.urls import url
from app.views import *
from django.contrib import admin
urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^$', main),
    url(r'^registration$', registration_form, name='registration'),
    url(r'^registration_1$', registration_1, name='registration_1'),
    url(r'^login1$', log_in, name='login1'),
    url(r'^login2$', log_in1, name='login2'),
    url(r'^logout$', logout_view, name='logout'),
    url(r'^logged_in$', logged_in_view, name='logged_in'),
    url(r'^logged_in_1$', logged_in, name='logged_in_1'),
    url(r'^registered$', user_registered, name='registered'),
]

```

admin.py

```

from django.contrib import admin
from .models import *
# Register your models here.

```

```

@admin.register(User1)
class UserAdmin(admin.ModelAdmin):
    empty_value_display = 'null'
    list_display = ('last_name', 'first_name', 'email', 'phone', 'passport',
'birthday')
    list_filter = ('last_name',)
    search_fields = ('last_name', 'first_name', 'email')

```

```

class BetTeam(admin.TabularInline):

```

```
model = BetTeam
extra = 1
```

```
@admin.register(Team)
class TeamAdmin(admin.ModelAdmin):
    empty_value_display = 'null'
    list_display = ('team_name', 'rating', 'sport', 'number_of_players')
    list_filter = ('team_name',)
    search_fields = ['team_name', 'sport']
    inlines = (BetTeam,)
```

```
def bets(self, request):
    bets = []
    for s in BetTeam.objects.filter(team=request.name):
        bets.append(s)
    return bets
```

```
@admin.register(Bet)
class BetAdmin(admin.ModelAdmin):
    empty_value_display = 'null'

    def username(self, obj):
        return "{}".format(obj.user)

    inlines = (BetTeam, )
    list_display = ('id', 'username', 'date', 'amount')
    list_filter = ('id',)
    search_fields = ['username', 'date', 'amount']
```

models.py

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

*# Create your models here.*

```
class User1(models.Model):
    user1 = models.OneToOneField(User, on_delete=models.CASCADE)
    first_name = models.CharField(max_length=20, null=True)
    last_name = models.CharField(max_length=20, null=True)
    phone = models.PositiveIntegerField(unique=True, null=True)
    email = models.EmailField(max_length=50, unique=True)
    birthday = models.DateField(blank=True, null=True)
    passport = models.PositiveIntegerField(unique=True, null=True)

    def __str__(self):
        return str(self.last_name)

class Team(models.Model):
    TeamId = models.AutoField(primary_key=True)
    team_name = models.CharField(max_length=30, unique=True, null=True)
    rating = models.PositiveSmallIntegerField(null=True, unique=True)
    sport = models.CharField(max_length=30, null=True)
    number_of_players = models.PositiveSmallIntegerField(null=True)
    picture = models.ImageField(upload_to='pics/', null=True, blank=True)

    def __str__(self):
        return str(self.team_name)
```

```
class Bet(models.Model):
    user = models.ForeignKey(User1, on_delete=models.CASCADE)
    team = models.ManyToManyField(Team, through="BetTeam")
    date = models.DateField()
    amount = models.FloatField(null=True)
```

```

def __str__(self):
    return str(self.id)

class BetTeam(models.Model):
    team = models.ForeignKey(Team, on_delete=models.CASCADE)
    bet = models.ForeignKey(Bet, on_delete=models.CASCADE)
    id = models.AutoField(primary_key=True)

def __str__(self):
    return str(self.id)

base.html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-
theme.min.css">
</body>
    <nav class="navbar navbar-inverse navbar-static-top">
        <div class="container-fluid">
            <div class="navbar-header">
                <a class="navbar-brand" href="/">Главная</a>
            </div>
            <div class="collapse navbar-collapse" id="myNavbar">
                <ul class="nav navbar-nav">
                    <li><a href="/registration">Регистрация</a></li>
                    <li><a href="/login1">Вход</a></li>
                    <li><a href="/logout">Выход</a></li>
                    <li><a href="/registration_1">Регистрация1</a></li>
                    <li><a href="/login2">Вход1</a></li>
                </ul>
            </div>
            <div>
                {% block content %}{% endblock %}
            </div>
        </div>
    </nav>
</body>
</html>

Loggedin.html
{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block content %}
    <div class="jumbotron">
        <div class="container">
            <div class="page-header">
                <h1>ВЫ ВЫПОЛНИЛИ ВХОД</h1>
            </div>
        </div>
    </div>
{% endblock %}

Userlogin_1.html
{% extends 'base.html' %}
{% block title %}login{% endblock %}
{% block content %}
    <div class="jumbotron">
        <div class="container">
            <form method="POST" class="form-horizontal">
                <h1 style="text-align:center">Вход</h1>

```

```

        {% csrf_token %}
        {% for f in form %}
            {% for error in f.errors %}
                <div class="alert " role="alert">{{ f.errors }}</div>
            {% endfor %}
        {% endfor %}
        {% for f in form %}
            <div class="form-group">
                <label for="{{ f.id_for_label }}" class="col-sm-3
control-label">{{ f.label }}:</label>
                <div class="col-sm-9">
                    {{ f }}
                </div>
            </div>
        {% endfor %}
        <div class="form-group">
            <div class="col-sm-offset-3 col-sm-9">
                <button style="text-align:center" class="col-sm-1 btn-
primary control-label" type="submit">Войти</button>
                <input style="text-align:center" class="col-sm-3 col-
sm-offset-1 btn-primary control-label"
                    type="button" value="Зарегистрироваться"
onClick='location.href="/registration_1"'>
            </div>
        </div>
    </form>
</div>
</div>
{% endblock %}

```

MainPage.html

```

{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block content %}
{% endblock %}

```

Registered.html

```

{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block content %}
    <div class="jumbotron">
        <div class="container">
            <div class="page-header">
                <h1>Пользователь зарегистрирован!</h1>
            </div>
        </div>
    </div>
{% endblock %}

```

Registration.html

```

{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block content %}
    <div class="jumbotron">
        <div class="container">
            <div class="page-header">
                <h1>Регистрация</h1>
                {% for err in errors.values %}
                    <div class="alert alert-danger" >{{ err }}</div>
                {% endfor %}
            </div>
            <form method="POST" class="form-horizontal">{% csrf_token %}
                <div class="col-sm-offset-3 form-group">
                    <label class="col-sm-3 control-label">Фамилия:</label>
                    <div class="col-sm-8">
                        <input type="text" name="last_name">
                    </div>
                </div>
            </form>

```



```

<div class="col-sm-offset-3 form-group">
  <label class="col-sm-3 control-label">Имя:</label>
  <div class="col-sm-8">
    <input type="text" name="first_name">
  </div>
</div>
<div class="col-sm-offset-3 form-group">
  <label class="col-sm-3 control-label">Email:</label>
  <div class="col-sm-8">
    <input type="text" name="Email">
  </div>
</div>
<div class="col-sm-offset-3 form-group">
  <label class="col-sm-3 control-label">Логин:</label>
  <div class="col-sm-8">
    <input type="text" name="username">
  </div>
</div>
<div class="col-sm-offset-3 form-group">
  <label class="col-sm-3 control-label">Пароль:</label>
  <div class="col-sm-8">
    <input type="password" name="password">
  </div>
</div>
<div class="col-sm-offset-3 form-group">
  <label class="col-sm-3 control-label">Повторите
ВВОД:</label>
  <div class="col-sm-8">
    <input type="password" name="password2">
  </div>
</div>
<button style="text-align:center" class="col-sm-offset-3 col-
sm-2 btn btn-primary control-label" type="submit">Зарегистрировать</button>
</form>
</div>
</div>
{% endblock %}
Registration_1.html
{% extends 'base.html' %}
{% block title %}registration{% endblock %}
{% block content %}
  <div class="main_block_elem">
    <div class="jumbotron">
      <div class="container">
        <form method="POST" class="form-horizontal">
          <h1>Регистрация </h1>
          {% csrf_token %}
          {% for field in form %}
            {% for error in field.errors %}
              <div class="alert alert-danger role="alert">{{
field.errors }}</div>
            {% endfor %}
          {% endfor %}
          {% for field in form %}
            <div class="form-group">
              <label for="{{ field.id_for_label }}" class="col-
sm-3 control-label">{{ field.label }}:</label>
              <div class="col-sm-9">
                {{ field }}
              </div>
            </div>
          {% endfor %}
          <div class="form-group">
            <div class="col-sm-offset-3 col-sm-9">
              <button type="submit" class="btn btn-
primary">Зарегистрировать</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
{% endblock %}

```

```

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

Userlogin.html
{% extends 'base.html' %}
{% block title %}{% endblock %}
{% block content %}
    <div class="jumbotron">
        <div class="container">
            <form method="POST" class="form-horizontal">
                <h1 style="text-align:center">Вход</h1>
                {% csrf_token %}
                {% for err in errors.values %}
                    <div class="alert alert-danger" >{{ err }}</div>
                {% endfor %}
                <div class="col-sm-offset-3 form-group">
                    <label class="col-sm-3 control-label">Логин:</label>
                    <div class="col-sm-8">
                        <input type="text" name="username">
                    </div>
                </div>
                <div class="col-sm-offset-3 form-group">
                    <label class="col-sm-3 control-label">Пароль:</label>
                    <div class="col-sm-8">
                        <input type="password" name="password">
                    </div>
                </div>
                <div class="form-group">
                    <div class="col-sm-offset-3 col-sm-9">
                        <button style="text-align:center" class="col-sm-1 btn
btn-primary control-label" type="submit">Войти</button>
                        <input style="text-align:center" class="col-sm-3 btn
col-sm-offset-1 btn-primary control-label" type="button"
value="Зарегистрироваться" onClick='location.href="/registration_1"'>
                    </div>
                </div>
            </form>
        </div>
    </div>
{% endblock %}

```

Результат

[Главная](#)
[Регистрация](#)
[Вход](#)
[Выход](#)
[Регистрация1](#)
[Вход1](#)

# Регистрация

Фамилия:

Имя:

Email:

Логин:

Пароль:

Повторите ввод:

Зарегистрировать

# Вход

Логин:

Пароль:

Войти

Зарегистрироваться

## Django administration

### Site administration

APP		
Bets	<a href="#">+ Add</a>	<a href="#">Change</a>
Teams	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>
AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">Change</a>
Users	<a href="#">+ Add</a>	<a href="#">Change</a>

### Recent actions

#### My actions

- [+ 1](#)  
Bet
- [+ Соболева](#)  
User1

Home > App > Teams

### Select team to change

Action:  0 of 8 selected

<input type="checkbox"/>	TEAM NAME	RATING	SPORT	NUMBER OF PLAYERS
<input type="checkbox"/>	Liverpool	9	футбол	11
<input type="checkbox"/>	FCB	8	футбол	11
<input type="checkbox"/>	Manchester	5	футбол	11
<input type="checkbox"/>	Спартак	7	футбол	11
<input type="checkbox"/>	Динамо	4	футбол	11
<input type="checkbox"/>	Локомотив	6	футбол	11
<input type="checkbox"/>	Tottenham Hotspur	2	футбол	11
<input type="checkbox"/>	ЦСКА	1	футбол	11

8 teams