

Согласовано:  
Гапанюк Ю.Е.

Утверждаю:  
Гапанюк Ю.Е.

"\_\_" \_\_\_\_\_ 2016 г.

«\_\_»\_\_

**Лабораторная работа №7 по курсу**  
**Разработка интернет приложений**

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53  
Бурашников В.В.

\_\_\_\_\_  
"\_" декабря 2016г.

### Задание и порядок выполнения

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

«forms.py»

```
from django import forms
from django.core.exceptions import ValidationError
from django.core.validators import validate_email
from django.contrib.auth.models import User

class LoginForm(forms.Form):
    login = forms.CharField(label='Login')
    password = forms.CharField(label='Password', widget=forms.PasswordInput)

class SignupForm(forms.Form):
    login = forms.CharField(label='Login', min_length=5)
    email = forms.CharField(label='Email')
    first_name = forms.CharField(label='First name')
    last_name = forms.CharField(label='Last name')
    password = forms.CharField(label='Password', min_length=8,
widget=forms.PasswordInput)
    repeat_password = forms.CharField(label='Repeat password',
widget=forms.PasswordInput)

    def clean_login(self):
        login = self.cleaned_data['login']
        if User.objects.filter(username=login):
            raise ValidationError('Этот login уже занят')
        return login

    def clean_email(self):
        email = self.cleaned_data['email']
        validate_email(self.cleaned_data['email'])
        if User.objects.filter(email=email):
            raise ValidationError('Этот email уже зарегистрирован')
        return self.cleaned_data['email']

    def clean(self):
        cleaned_data = super(SignupForm, self).clean()
        if self.cleaned_data.get('password') and
self.cleaned_data.get('repeat_password'):
            if self.cleaned_data['password'] !=
self.cleaned_data['repeat_password']:
                raise ValidationError('Пароли не совпадают')
        return cleaned_data

    def save(self):
```

```

        user = User.objects.create_user(username=self.cleaned_data['login'],
                                         email=self.cleaned_data['email'],
                                         password=self.cleaned_data['password'],

first_name=self.cleaned_data['first_name'],

last_name=self.cleaned_data['last_name'],
        )

    return user

```

## «views.py»

```

from django.shortcuts import render
from django.contrib import auth
from django.http import HttpResponseRedirect
from .forms import LoginForm, SignupForm

def login(request):
    redirect_url = '/'
    if request.method == 'POST':
        redirect_url = '/'
        form = LoginForm(request.POST)
        if form.is_valid():
            user = auth.authenticate(username=form.cleaned_data['login'],
                                    password=form.cleaned_data['password'])

            if user:
                auth.login(request, user)
                return HttpResponseRedirect(redirect_url)
            else:
                form.add_error(None, 'invalid login/password')
        else:
            form = LoginForm()
    return render(request, 'login.html', {'form': form, 'continue':
redirect_url})

def signup(request):
    if request.method == 'POST':
        form = SignupForm(request.POST)
        if form.is_valid():
            user = form.save()
            return HttpResponseRedirect('/')
        else:
            form = SignupForm()
    return render(request, 'signup.html', {'form': form,
                                           'type': 'Registration'})

def index(request):
    return HttpResponseRedirect("Hello")

```

## Шаблоны «login.html»

```
<html>
<head>
  <link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>

<body>

{{ form.non_field_errors }}
<form action="/login/" method="post" class="form-horizontal">
  {% csrf_token %}
  {% for i in form %}
    <div class="form-group">
      <label class="col-sm-2">{{ i.label }}</label>
      <div class="col-sm-10">
        <div>
          {{ i }}
        </div>
        <div>
          {{ i.errors }}
        </div>
      </div>
    </div>
  {% endfor %}

  <div class="form-group">
    <div class="col-sm-offset-2 col-sm-10">
      <input type="submit" value="LOGIN" class="btn btn-
default"/>
    </div>
  </div>
</form>

</body>
</html>
```

## «signup.html»

```
<html>
<head>
{% load static %}
<link rel="stylesheet"
href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
</head>

<body>

{{ form.non_field_errors }}
<form action="/signup/" method="post" class="form-horizontal"
enctype="multipart/form-data">
  {% csrf_token %}
  {% for i in form %}
    <div class="form-group">
      <label class="col-sm-4">{{ i.label }}</label>
      <div class="col-sm-8">
        <div>
          {{ i }}
        </div>
      </div>
    </div>
  {% endfor %}
</form>

</body>
</html>
```

```

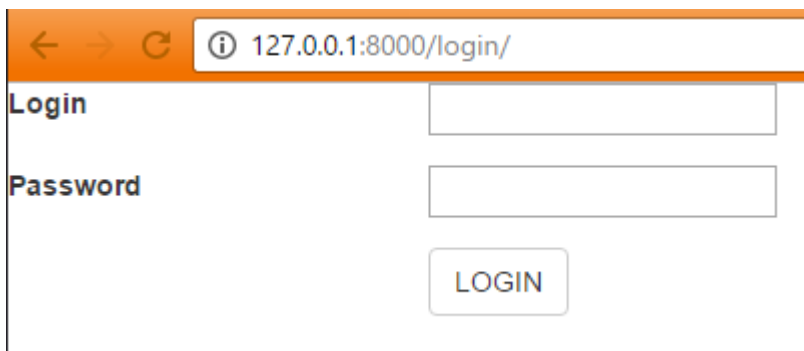
        <div>
            {{ i.errors }}
        </div>

    </div>
</div>
{% endfor %}

<div class="form-group">
    <div class="col-sm-offset-4 col-sm-8">
        <input type="submit" value="SIGNUP" class="btn btn-
default"/>
    </div>
</div>
</form>
</body>
</html>

```

## Login.html



← → ↻ ⓘ 127.0.0.1:8000/login/

**Login**

**Password**

## Signup.html



← → ↻ ⓘ 127.0.0.1:8000/signup/

**Login**

**Email**

**First name**

**Last name**

**Password**

**Repeat password**

