

**Лабораторная работа №7 по дисциплине  
“Разработка интернет приложений”  
Вариант 5**

ИСПОЛНИТЕЛЬ:  
студент группы РТ5-51  
Гузилов.А.В

"\_\_" \_\_\_\_\_ 2016 г.

## 1. 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
2. Создайте view, которая возвращает форму для авторизации
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
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'а через команду `manage.py`
11. Подключить `django.contrib.admin` и войти в панель администрирования.
12. Зарегистрировать все свои модели в `django.contrib.admin`

## 2. Листинги файлов проекта

### Файл forms.py:

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

class RegistrationForm(forms.Form):
    username = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control',
        'placeholder': 'Username', }),
        max_length=30, min_length=5, required=True, label='Login'
    )
    first_name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control',
        'placeholder': u'Имя', }),
        max_length=30, required=True, label='Firstname'
    )
    last_name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control',
        'placeholder': u'Фамилия', }),
        max_length=30, required=True, label='Lastname'
    )
    email = forms.EmailField(
        widget=forms.TextInput(attrs={'class': 'form-control',
        'placeholder': 'E-mail', }),
        required=True, max_length=254, label='E-mail'
    )
    password1 = forms.CharField(
        widget=forms.PasswordInput(attrs={'class': 'form-control',
        'placeholder': 'Password'}),
        min_length=8, label='Password'
    )
    password2 = forms.CharField(
        widget=forms.PasswordInput(attrs={'class': 'form-control',
        'placeholder': 'Password'}),
        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 LoginForm(forms.Form):
    login = forms.CharField(
        widget=forms.TextInput(
            attrs={'class': 'form-control', 'placeholder': 'Username', }
        ),
        max_length=30,
        label='Login'
    )
    password = forms.CharField(
```

```

        widget=forms.PasswordInput(
            attrs={'class': 'form-control', 'placeholder': 'Password', }
        ),
        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('not active')
    else:
        raise forms.ValidationError('Wrong username or password')

```

### **Файл views.py:**

```

from django.shortcuts import render
from django.contrib import auth
from django.http.response import HttpResponseRedirect
from django.views import View
from login.models import *
from login.forms import *

```

```

def registration(request):

    if request.user.is_authenticated():
        return render(request, 'loggedin.html')

    if request.method == "POST":
        form = RegistrationForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth.login(request, user)
            return render(request, 'registered.html')
    else:
        form = RegistrationForm()

```

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

```
def login(request):
    if request.method == "POST":
        form = LoginForm(request.POST)

        if form.is_valid():
            auth.login(request, form.cleaned_data['user'])
            return render(request, 'loggedin.html')
    else:
        form = LoginForm()
    return render(request, 'login.html', {
        'form': form
    })
```

```
def loggedin(request):
    return render(request, 'loggedin.html')
```

### **Файл templates/registration.html:**

```
<html>
  <form action = 'registration' method = "POST">
    {% for e in form.non_field_errors %}
      {{ e }}
    {% endfor %}
    {% csrf_token %}
    {% for field in form %}
      <p>{{ field.label }}</p>
      <p>{{ field }}</p>
      {% if field.errors %}
        {% for e in field.errors %}
          {{ e }}
        {% endfor %}
      {% endif %}
    {% endfor %}
    <button>Sign up!</button>
  </form>
</html>
```

### Файл templates/registered.html:

```
<html>
  <body>
    <h1> You are successfully registered</h1>
  </body>
</html>
```

### Файл templates/login.html:

```
<html>
<body>
  <form action = "login" method="POST">
    {% for e in form.non_field_errors %}
      {{ e }}
    {% endfor %}
    {% csrf_token %}
    {{ form.login.label }}
    {{ form.login }}
    {{ form.password.label }}
    {{ form.password }}

  <button> Log In </button>
</form>

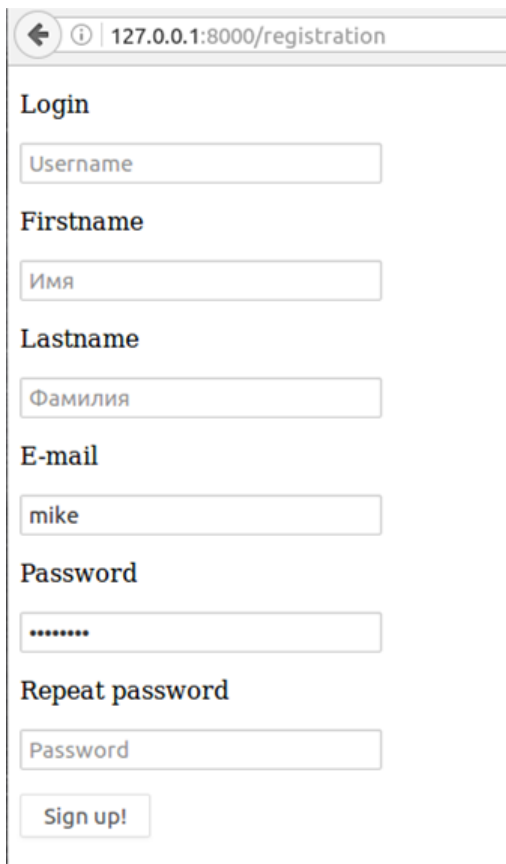
</body>
</html>
```

### Файл templates/loggedin.html:

```
<html>
  <body>
    <h1>You are successfully logged in</h1>
  </body>
</html>
```

## Скриншоты выполнения лабораторной работы:

### Форма регистрации



127.0.0.1:8000/registration

Login

Username

Firstname

Имя

Lastname

Фамилия

E-mail

mike

Password

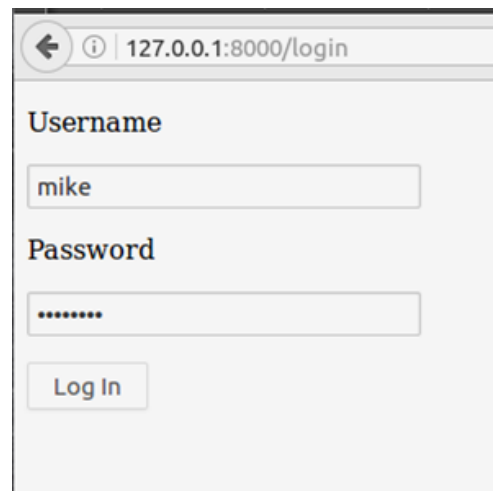
\*\*\*\*\*

Repeat password

Password

Sign up!

### Форма авторизации



127.0.0.1:8000/login

Username

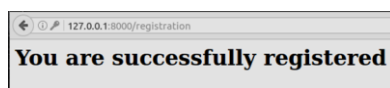
mike

Password

\*\*\*\*\*

Log In

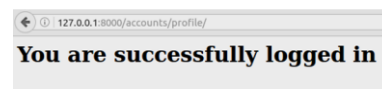
### При успешной регистрации



127.0.0.1:8000/registration

You are successfully registered

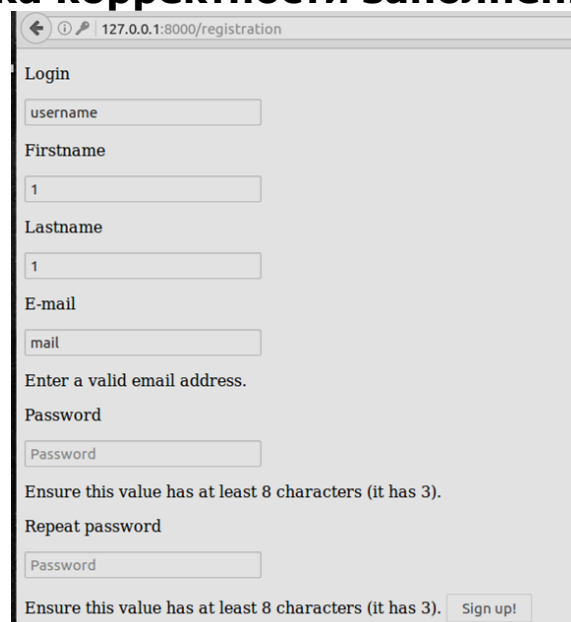
### При успешной авторизации



127.0.0.1:8000/accounts/profile/

You are successfully logged in

### Проверка корректности заполнения полей



127.0.0.1:8000/registration

Login

username

Firstname

1

Lastname

1

E-mail

mail

Enter a valid email address.

Password

Password

Ensure this value has at least 8 characters (it has 3).

Repeat password

Password

Ensure this value has at least 8 characters (it has 3).

Sign up!



# Панель управления Django

