

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

Отчет по лабораторной работе № 7 по курсу
«Разработка интернет-приложений»
«Авторизация, работа с формами и Django Admin»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53

Ореликов М.Г.

(подпись)

"__" _____ 2016 г.

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'а 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

Листинги файлов проекта

admin.py

```
from django.contrib import admin
from core import models

@admin.register(models.Service)
class Service(admin.ModelAdmin):
    list_display = ('name', 'description', 'category', 'price', 'expensive')
    search_fields = ('name', 'price',)
    list_filter = ('price', 'name')
    actions = ('null_comment',)

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

    def expensive(self, srv):
        return True if int(srv.price) > 17000 else False
```

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 SignupForm(forms.Form):
    username = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
'Username', }),
        min_length=5,
        required=True,
        label='Login'
    )
    first_name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
u'First name', }),
        required=True,
        label='Firstname'
    )
    last_name = forms.CharField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
u'Last name', }),
        required=True,
        label='Lastname'
    )
    email = forms.EmailField(
        widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
'Email', }),
        required=True,
        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':
'Repeat password'}),
        min_length=8,
        label=u'Repeat password'
    )
)
```

```

def clean_username(self):
    username = self.cleaned_data.get('username', '')
    try:
        user = User.objects.get(username=username)
        raise forms.ValidationError('User with such username 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')
    user = User()

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

    return authenticate(username=user.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='Password'
    )

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

        if user is None:
            raise forms.ValidationError('Invalid username or password')

```

models.py

```
from django.db import models
```

```

class Service(models.Model):
    name = models.CharField(max_length=20)
    description = models.CharField(max_length=255)
    category = models.ForeignKey('Service', null=True, blank=True)
    price = models.CharField(max_length=20, null=False)

```

views.py

```
from django.shortcuts import render
from django.contrib.auth.decorators import login_required
from django.views import View
from django.contrib import auth
from django.http import HttpResponseRedirect, Http404
from core.models import Service
from core import forms
```

```
def main(request):
    services = Service.objects.all()

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

class ServiceView(View):
    def get(self, request, id):
        try:
            service = Service.objects.get(id=int(id))
        except:
            raise Http404

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

def login(request):
    redirect = request.GET.get('continue', '/success')
    if request.method == 'POST':
        form = forms.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)
        else:
            form = forms.LoginForm()
    return render(request, 'login.html', {
        'form': form,
    })

def signup(request):
    redirect = request.GET.get('continue', '/success')
    if request.method == 'POST':
        form = forms.SignupForm(request.POST)
        if form.is_valid():
            user = form.save()
            auth.login(request, user)
            return HttpResponseRedirect(redirect)
        else:
            form = forms.SignupForm()
    return render(request, 'signup.html', {
        'form': form,
    })
```

```

@login_required(redirect_field_name='continue')
def login_success(request):
    redirect = request.GET.get('continue', '/')
    return HttpResponseRedirect(redirect)

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

```

urls.py

```

from django.conf.urls import url
from django.contrib import admin
from core import views

urlpatterns = [
    url(r'^$', views.main, name='main'),
    url(r'^logout/$', views.logout, name='logout'),
    url(r'^login/$', views.login, name='login'),
    url(r'^signup/?$', views.signup, name='signup'),
    url(r'^success/?$', views.login_success, name='success'),
    url(r'^service/(?P<id>\d+)\$',
        views.ServiceView.as_view(), name='service'),
    url(r'^admin/', admin.site.urls),
]

```

layout.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    {% load static %}
    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'css/styles.css' %}" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-default">
        <a class="navbar-brand" href="{% url 'main' %}">LAB 7</a>
        {% if request.user.is_authenticated %}
        <a href="{% url 'logout' %}" class="btn btn-default navbar-btn pull-
right">Logout</a>
        {% else %}
        <a href="{% url 'signup' %}" class="btn btn-default navbar-btn pull-
right">Sign up</a>
        <a href="{% url 'login' %}" class="btn btn-default navbar-btn pull-right
login-nav">Log in</a>
        {% endif %}
    </nav>

    <div class="page-header">
        <h1 class="text-center">LAB 7</h1>
    </div>

    <div class="container">
        {% block body %}{% endblock %}
    </div>
    <script src="{% static 'js/bootstrap.min.js' %}"></script>
</body>
</html>

```

login.html

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

{% block title %}
    Login
{% endblock %}

{% block body %}
<div class="wrapper">

    <form class="form-signin" method="post" action="">
        {% csrf_token %}
        <h2 class="form-signin-heading">Please login</h2>
        {% 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 %}
        {% for i in form %}
            {{ i.errors }}
            {{ i }}
        {% endfor %}
        <button class="btn btn-lg btn-primary btn-block"
type="submit">Login</button>
    </form>
</div>
{% endblock %}
```

main.html

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

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

{% block body %}
    <div class="row">
        {% for s in services %}
            <div class="col-xs-4">
                <a href="{% url 'service' id=s.id %}">
                    
                    <h4 class="text-center">{{ s.name }}</h4>
                    <p class="text-center">Price:{{ s.price }} rub.</p>
                </a>
            </div>
        {% endfor %}
    </div>

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

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

{% block body %}
    <div class="row">
        {% for s in services %}
            <div class="col-xs-4">
                <a href="{% url 'service' id=s.id %}">
```

```

        
        <h4 class="text-center">{{ s.name }}</h4>
        <p class="text-center">Price:{{ s.price }} rub.</p>
    </a>
</div>
{% endfor %}
</div>
{% endblock %}

```

service.html

```

{% extends 'layout.html' %}

{% block title %}
    {{ service.name }}
{% endblock %}

{% block body %}
    <div class="text-center">
        <a href="{% url 'main' %}">Back to list</a>
    </div>

    <div class="container">
        <div class="row">
            <div class="col-xs-5 col-xs-offset-1">
                
            </div>
            <div class="col-xs-5 text-center">
                <h2>
                    {{ service.name }}
                </h2>
                <p>Price: {{ service.price }}</p>
                <h4>{{ service.description }}</h4>
            </div>
        </div>
    </div>

{% endblock %}

```

Sigup.html

```

{% extends 'layout.html' %}
{% block title %}
    Sign up
{% endblock %}
{% block body %}
    <div class="wrapper">
        <form class="form-signin" method="post" action="">
            {% csrf_token %}
            <h2 class="form-signin-heading">Please signup</h2>
            {% 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 %}
            {% for field in form %}
                {{ field.errors }}
                {{ field }}
            {% endfor %}
            <button class="btn btn-lg btn-primary btn-block" type="submit">Sign
up</button>
        </form>
    </div>
{% endblock %}

```


Результаты выполнения

LAB 7

Logout

LAB 7

250×250

comp1

Price:16000 rub.

250×250

comp0

Price:18000 rub.

LAB 7

Log in

Sign up

LAB 7

Please login

Username

Password

Login

LAB 7

Log in

Sign up

LAB 7

Please signup

• User with such username already exists

user1

Joe

Joe

joe@gmail.com

Password

Repeat password

Sign up

Select service to change

[ADD SERVICE](#) +Q Action: 0 of 2 selected

<input type="checkbox"/>	NAME	DESCRIPTION	CATEGORY	PRICE	EXPENSIVE
<input type="checkbox"/>	comp1	comp1	-	16000	False
<input type="checkbox"/>	comp0	comp0	-	18000	True

2 services

FILTER

By price

[All](#)
[16000](#)
[18000](#)

By name

[All](#)
[comp0](#)
[comp1](#)