

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



“Разработка интернет приложений”

Лабораторная работа №7 «Авторизация, работа с формами и Django Admin»

Студент группы ИУ5-51 _____ Руднев И.В.

Преподаватель _____ Гапанюк Ю.Е.

Москва 2016

1. Задание

Научиться обрабатывать веб-формы на стороне приложения, освоить инструменты, которые предоставляет Django по работе с формами. Также освоить инструменты Django по работе с авторизацией и реализовать простейшую авторизацию. Познакомиться с инструментом администрирования Django.

2. Порядок выполнения

Создать нужные формы в файле `forms.py`

```
#!/usr/bin/env python
#-*-coding: utf-8-*-
from django import forms
from django.contrib.auth.models import User
from models import Service

class LoginForm(forms.Form):
    username = forms.CharField(widget=forms.TextInput(attrs={'class': 'input
inputlogin'}), label=u'Логин', max_length = 30)
    password = forms.CharField(widget=forms.PasswordInput(attrs={'class': '
input inputlogin'}), label=u'Пароль', max_length = 30)
    redirect = forms.CharField(widget=forms.HiddenInput, label='')

class SignupForm(forms.Form):
    username = forms.CharField(widget=forms.TextInput(attrs={'class': 'input
inputlogin'}), label=u'Логин', max_length = 30)
    email = forms.EmailField(widget=forms.TextInput(attrs={'class': 'input
inputlogin'}), label='E-mail', max_length = 30)
    password = forms.CharField(widget=forms.PasswordInput(attrs={'class': '
input inputlogin'}), label=u'Пароль', max_length = 30)
    repeat_password = forms.CharField(widget=forms.PasswordInput(attrs={'class': '
input inputlogin'}), label=u'Повторите
пароль', max_length = 30)
    redirect = forms.CharField(widget=forms.HiddenInput, label='')

    def clean_username(self):
        username = self.cleaned_data.get('username')
        if User.objects.filter(username=username).exists():
            self._errors["username"] = self.error_class([u'Пользователь с таким
именем уже существует'])
            del self.cleaned_data["username"]
        if len(username) < 5:
            self._errors["username"] = self.error_class([u'Длина логина должна
быть не менее 5 символов'])
            del self.cleaned_data["username"]
        return username

    def clean(self):
        cleaned_data = super(SignupForm, self).clean()
        password = self.cleaned_data.get('password', '')
        repeat_password = self.cleaned_data.get('repeat_password', '')
        if password != repeat_password:
            self._errors["password"] = self.error_class([u'Пароли не
совпадают'])
        elif len(password) < 8:
            self._errors["password"] = self.error_class([u'Введите пароль длиной
не менее 8 символов'])
        return cleaned_data
```

```

# Форма добавления услуги
class ServiceForm(forms.ModelForm):
    class Meta:
        model = Service
        fields = ['title', 'text', 'price']
        widgets = {
            'title': forms.TextInput(attrs={'class': 'input fieldname'}),
            'text': forms.Textarea(attrs={'class': 'input fieldname'}),
            'price': forms.TextInput(attrs={'class': 'input fieldname'})
        }
        labels = {
            'title': u'Название',
            'text': u'Описание',
            'price': u'Цена'
        }

    def __init__(self, *args, **kwargs):
        self.author = kwargs.pop('author', None)
        super(ServiceForm, self).__init__(*args, **kwargs)

    def save(self, commit=True):
        obj = super(ServiceForm, self).save(commit=False)
        obj.author = self.author
        if commit:
            obj.save()
        return obj

```

Добавить недостающие методы в файл **views.py**

```

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

    if request.POST:
        form = SignupForm(request.POST)
        if form.is_valid():
            redirect = form.cleaned_data.get('redirect', '/')
            username = form.cleaned_data.get('username')
            email = form.cleaned_data.get('email')
            password = form.cleaned_data.get('password')
            User.objects.create_user(username=username, email=email,
password=password)
            us = authenticate(username=username, password=password)
            auth_login(request, us)
            return HttpResponseRedirect(redirect)
        else:
            return render(request, 'signup.html', {'form': form })
    else:
        form = SignupForm(initial={'redirect': request.GET.get('next', '/')})
        return render(request, 'signup.html', {'form': form })

```

```

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

    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            user = authenticate(username=username, password=password)
            redirect = form.cleaned_data.get('redirect', '/')
            if user:
                auth_login(request, user)
                if redirect == '/login/' or redirect == '/signup/':
                    redirect = '/'
                return HttpResponseRedirect(redirect)
            else:
                error = u'Неверная связка логин-пароль'
                form.redirect = redirect
                return render(request, 'login.html', {'form': form, 'error':
error})
        else:
            return render(request, 'login.html', {'form': form})

    else:
        form = LoginForm(initial={'redirect': request.GET.get('next', '/')})
        return render(request, 'login.html', {'form': form})

def logout(request):
    auth_logout(request)
    return HttpResponseRedirect('/')

def create_service (request):
    if not request.user.is_authenticated():
        return HttpResponseRedirect('/login/?next=' + request.path)
    if request.POST:
        user=request.user
        form = ServiceForm(request.POST, author=user)
        if form.is_valid():
            service = form.save()
            return HttpResponseRedirect('/service/' + str(service.id))
        else:
            return render(request, 'new_service.html', { 'form' : form })
    else:
        form = ServiceForm()
        return render(request, 'new_service.html', { 'form' : form })

```

Добавить шаблон **login.html**

```

{% extends "base.html" %}
{% block content %}

<div class="content contentask" style="padding-left: 50px">
<h1>Авторизация</h1>

```

```

<span class="error">{{ error }}</span>

<form name="login" method="post" action="/login/">
    {% csrf_token %}
    {% for field in form %}
        {% if field.errors %}
            <span class="error">Поле обязательно</span>
        {% endif %}
        <div>
            <label class = "fieldname" for="{{
field.id_for_label }}"> {{ field.label }} </label>
            <div {% if field.errors %}class="error_input"{%
endif %}>{{ field }}</div>
            </div>
        {% endfor %}
        <input class="button" type="submit" value="Войти">
    </form>
</div>

{% endblock %}

```

Signup.html

```

{% extends "base.html" %}
{% block content %}
    <div class="content contentask" style="padding-left: 50px">
        <h1>Регистрация</h1>
        <form action="/signup/" class="form_question" method="POST"
        enctype="multipart/form-data">
            {% csrf_token %}

            {% for field in form %}
                <span class="error"> {{ field.errors }}</span>
                <div>
                    <label class = "fieldname" for="{{ field.id_for_label
}}"> {{ field.label }} </label>
                    <div {% if field.errors %}class="error_input"{% endif
%}> {{ field }} </div>
                </div>
            {% endfor %}

            <div>
                <input class="button" type="submit" style="margin-top:
15px" value="Создать аккаунт">
            </div>

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

```

New_service.html

```

{% extends "base.html" %}
{% block content %}
    <div class="login-form">
        <h1 style="size: 18px">Создать услугу </h1>
        <form action="/newservice/" class="form_question"
        method="POST">

            {% csrf_token %}

            {% for field in form.visible_fields %}

```

```

        {% if field.errors %}
            <span class="error">Поле обязательно</span>
        {% endif %}
        <div>
            <label style="color: #1b6d85" for="{{
field.id_for_label }}"> {{ field.label }} </label>
            {{ field }}
        </div>
    {% endfor %}
    <input class="button" type="submit" value="Отправить">

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

```

Настроить страницу администрирования для возможности фильтрации, поиска и т.д в файле **admin.py**:

```

from django.contrib import admin
from django.contrib.admin import DateFieldListFilter

# Register your models here.

from django.contrib import admin
from models import Service

@admin.register(Service)
class AuthorAdmin(admin.ModelAdmin):
    list_display = ('title', 'price')
    search_fields = ('title')
    list_filter = (
        ('added_at', DateFieldListFilter),
    )

```

Скриншоты:

My_APP

Search

Создать услугу

Создать аккаунт Войти

Регистрация

Логин

E-mail

Пароль

Повторите пароль

Создать аккаунт

Авторизация

Логин

madsunrise

Пароль

Войти

Создать услугу

Название

Описание

Цена

Отправить

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

✎ Change

Users

+ Add

✎ Change

MY_APP

Services

+ Add

✎ Change

Recent actions

My actions

None available

Select service to change

ADD SERVICE +

Q		Search
Action:	Delete selected services ▾	Go
<input type="checkbox"/>	TITLE	PRICE
<input type="checkbox"/>	David Nelson	966.00
<input type="checkbox"/>	James Ford	290.00
<input type="checkbox"/>	Theresa Carter	374.00
<input type="checkbox"/>	Sara Harris	115.00
<input type="checkbox"/>	Mary Salinas	217.00
<input type="checkbox"/>	Mark Neal	82.00
<input type="checkbox"/>	Linda Fischer	601.00
<input type="checkbox"/>	Renee Lin	728.00
<input type="checkbox"/>	Sarah Horn	524.00
<input type="checkbox"/>	William Hernandez	883.00
<input type="checkbox"/>	Rebecca Rose	156.00
<input type="checkbox"/>	Benjamin Turner	989.00

FILTER

By added at

Any date
Today
Past 7 days
This month
This year