

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



Отчет по лабораторной работе №7
«Работа с формами, авторизация, django admin»
По курсу “Разработка интернет-приложений”

Студент группы ИУ5-51
Жизневский П.И.

Москва 2017

Задание и порядок выполнения

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

Исходники:

Views.py:

```
from django.shortcuts import render
from django.http import HttpResponseRedirect, HttpResponse
from django.views.generic import ListView
from django import forms
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.contrib.auth.mixins import LoginRequiredMixin
from .models import *

class TeacherList(LoginRequiredMixin, ListView):
    context_object_name = 'teachers'
    login_url = '/labs/'
    redirect_field_name = 'redirect_to'
    model = Teacher
    template_name = 'teacher_list.html'

class PulpitList(ListView):
    context_object_name = 'pulpits'
    model = Pulpit
    template_name = 'pulpit_list.html'
```

```

def registration_dumb(request):
    errors = {}
    request.encoding = 'utf-8'
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['uname']='Введите логин'
        elif len(username) < 5:
            errors['uname']='Длина логина должна быть не меньше 5 символов'

        if User.objects.filter(username=username).exists():
            errors['uname']='Такой логин уже занят'

        password = request.POST.get('password')
        if not password:
            errors['psw']='Введите пароль'
        elif len(password) < 8:
            errors['psw']='Длина пароля должна быть не меньше 8 символов'

        password_repeat = request.POST.get('password2')
        if password != password_repeat:
            errors['psw2']='Пароли должны совпадать'

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

        last_name = request.POST.get('last_name')
        if not last_name:
            errors['lname']='Введите фамилию'

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

        if not errors:
            user = User.objects.create_user(username, email, password)
            # usr = Userus()
            # usr.user = user
            # usr.first_name = first_name
            # usr.last_name = last_name
            # usr.save()
            return HttpResponseRedirect('/labs/users')
        else:
            context = {'errors':errors, 'username':username, 'email': email,
                'last_name': last_name, 'first_name': first_name}
            return render(request, 'registration_dumb.html',context)

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

class RegistrationForm(forms.Form):
    username = forms.CharField(min_length=5,label='Логин')
    password = forms.CharField(min_length=8,widget=forms.PasswordInput,
label='Пароль')
    password2 = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Повторите ввод')
    email = forms.EmailField(label='Email')
    last_name = forms.CharField(label='Фамилия')
    first_name = forms.CharField(label='Имя')

```

```

def registration_user(request):
    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:
            data = form.cleaned_data
            User.objects.create_user(data['username'], data['email'],
data['password'])
            # usr = Userus()
            # usr.user = user
            # usr.first_name = data['first_name']
            # usr.last_name = data['last name']
            # usr.save()
            return HttpResponseRedirect('/labs/authorization')
        else:
            form = RegistrationForm()

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

@login_required(login_url='/labs/authorization')
def success_authorization(request):
    return HttpResponseRedirect('/labs')

def success_authorization_dumb(request):
    if request.user.is_authenticated:
        return HttpResponseRedirect('/labs/')
    else:
        return HttpResponseRedirect('/labs/authorization')

def authorization(request):
    errors = {}
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['uname'] = 'Введите логин'
        elif len(username) < 5:
            errors['uname'] = 'Длина логина должна быть не меньше 5 символов'

        password = request.POST.get('password')
        if not password:
            errors['psw'] = 'Введите пароль'
        elif len(password) < 8:
            errors['psw'] = 'Длина пароля должна быть не меньше 8 символов'

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

    if not errors:
        login(request, user)
        #return HttpResponseRedirect('/labs/success_authorization_dumb')

```

```

        return HttpResponseRedirect('/labs/success_authorization')
    else:
        context = {'errors':errors}
        return render(request, 'authorization.html',context)

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

def logout_view(request):
    logout(request)
    return HttpResponseRedirect('/labs/')

class AuthorizationForm(forms.Form):
    pass

def index(request):
    return render(request, 'test.html')

```

Urls.py

```

from django.conf.urls import url, include
from django.contrib import admin
from labapp.views import index
#from labapp.views import home, labapp

```

```

urlpatterns = [
    url(r'^labs/', include('labapp.urls')),
    url(r'^admin/', admin.site.urls),
]

```

Labbapp/urls.py

```

from django.conf.urls import url

from . import views

urlpatterns = [
    url(r'^logout$', views.logout_view, name='logout'),
    url(r'^success_authorization_dumb$', views.success_authorization_dumb,
name='success_authorization_dumb'),
    url(r'^success_authorization$', views.success_authorization,
name='success_authorization'),
    url(r'^authorization/$', views.authorization, name='authorization'),
    url(r'^registration_dumb/$', views.registration_dumb,
name='registration_dumb'),
    url(r'^registration_user/$', views.registration_user,
name='registration_user'),
    url(r'^teachers/$', views.TeacherList.as_view(), name='teacher_list'),
    url(r'^pulpit/$', views.PulpitList.as_view(), name='pulpit_list'),
    # url(r'^reviews/$', views.ReviewList.as_view(), name='review_list'),
    url(r'^$', views.index, name='index'),
]

```

admin.py

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

# Register your models here.
class UserAdmin(admin.ModelAdmin):
    #fields = ('first_name', 'last_name')
    list_display = ('username', 'full_name', 'age', 'has_reviews',)
    list_filter = ('age',)
    search_fields = ['last_name', 'first_name']

    def full_name(self, obj):
        return "{} {}".format(obj.last_name, obj.first_name)

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

@admin.register(Pulpit)
class PulpitAdmin(admin.ModelAdmin):
    empty_value_display = '-empty-'

@admin.register(Teacher)
class PulpitAdmin(admin.ModelAdmin):
    empty_value_display = '-empty-'
```

models.py

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

# Create your models here.
class Teacher(models.Model):
    name = models.CharField(max_length=30)
    second_name = models.CharField(max_length=30)
    third_name = models.CharField(max_length=30)
    phone = models.BigIntegerField()
    mail = models.CharField(max_length=30)

class Pulpit(models.Model):
    name = models.CharField(max_length=5)
    year = models.IntegerField()
    objects = models.Manager()

class Membership(models.Model):
    teacher = models.ForeignKey(Teacher, on_delete=models.CASCADE)
    pulpit = models.ForeignKey(Pulpit, on_delete=models.CASCADE)
```

Стартовая страница для неавторизованных пользователей

← → ↻ 127.0.0.1:8000/labs/ ☆

Классификация роз Крупнолистная горш Смайлы ВКонтакте Лестница с двумя пр Окна, двери Для приложений в н » Другие закладки

МГТУ им Н.Э.Баумана Войти Зарегистрироваться

Зарегистрируйтесь или войдите в систему

Страница регистрации

Классификация роз Крупнолистная горш Смайлы ВКонтакте Лестница с двумя пр Окна, двери Для приложений в н » Другие закладки

МГТУ им Н.Э.Баумана Войти Зарегистрироваться

Регистрация пользователя

Логин:
karl1

Пароль:
...

Показано сообщение: Минимально допустимое количество символов: 8. Длина текста сейчас: 3.

Email:
tin-ton@gmail.com

Фамилия:
Пупкин

Имя:
Вася

РЕГИСТРАЦИЯ

Форма входа

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/labs/authorization/`. The browser's bookmark bar contains several items, including "Классификация роз", "Крупнолистная горш", "Смайлы ВКонтакте", "Лестница с двумя п", "Окна, двери", and "Для приложений в". The page has a purple header with the text "МГТУ им Н.Э.Баумана" and links for "Войти" and "Зарегистрироваться". The main content area is light blue and features the heading "Авторизация". Below the heading, there are two input fields: "Логин:" with the value "super_ded" and "Пароль:" with masked characters ".....". A purple button labeled "ВОЙТИ" is positioned below the password field.

127.0.0.1:8000/labs/authorization/

Классификация роз Крупнолистная горш Смайлы ВКонтакте Лестница с двумя п Окна, двери Для приложений в Другие закладки

МГТУ им Н.Э.Баумана Войти Зарегистрироваться

Авторизация

Логин:

super_ded

Пароль:

.....

ВОЙТИ

Меню администратора

The screenshot shows the Django administration interface in a web browser window with the address bar displaying `127.0.0.1:8000/admin/`. The browser's bookmark bar is the same as in the previous screenshot. The page has a dark blue header with the text "Django administration" and a welcome message "WELCOME, PAVEL7697" followed by links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". The main content area is white and features the heading "Site administration". Below the heading, there are two sections: "AUTHENTICATION AND AUTHORIZATION" and "LABAPP". Each section contains a table with columns for the object name, a "+ Add" link, and a "Change" link. The "AUTHENTICATION AND AUTHORIZATION" section has two rows: "Groups" and "Users". The "LABAPP" section has two rows: "Pulpits" and "Teachers". On the right side of the page, there is a "Recent actions" section with a "My actions" subsection, which currently shows "None available".

127.0.0.1:8000/admin/

Классификация роз Крупнолистная горш Смайлы ВКонтакте Лестница с двумя п Окна, двери Для приложений в Другие закладки

Django administration

WELCOME, PAVEL7697 [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

LABAPP		
Pulpits	+ Add	Change
Teachers	+ Add	Change

Recent actions

My actions

None available

Стартовая страница для авторизованных пользователей

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/labs/'. The browser's bookmark bar contains several links: 'Классификация роз', 'Крупнолистная гор...', 'Смайлы ВКонтакте', 'Лестница с двумя п...', 'Окна, двери', 'Для приложений в...', 'Онлайн камера 8 -', and 'Другие закладки'. The website's header is purple and contains the text 'МГТУ им Н.Э.Баумана' in the center, 'pavel7697' on the right, and 'Выйти' next to it. On the left side of the header, there are links for 'Преподаватели' and 'Кафедры'. Below the header, the main content area has a light blue background. It starts with the heading 'Свежие новости' (Latest News). Below this heading, there are three identical green rectangular boxes, each containing a paragraph of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.'

Список преподавателей


The screenshot shows the same web browser window as before, but the address bar now displays '127.0.0.1:8000/labs/teachers/'. The website's header remains the same. The main content area has a light blue background and displays a list of three teachers. Each teacher's entry consists of a small square image of a forest path with autumn leaves, followed by their name in bold, their last name in bold, and their contact information. The first teacher is 'Спиридонов Сергей Борисович' with contact number '12345' and email 'sb@mail.ru'. The second teacher is 'Тоноян Славик Анушаванович' with contact number '89997775544' and email 'ton@hasagi.ru'. The third teacher is 'Гапанюк Юрий Евгеньевич' with contact number '89997771133' and email 'y.gapanyk@hasagi.ru'.

Список кафедр


← → ↻ 127.0.0.1:8000/labs/pulpit/ ☆

Классификация роз Крупнолистная горт Смайлы ВКонтакте Лестница с двумя п Окна, двери Для приложений в Онлайн камера 8 - Другие закладки


Преподаватели Кафедры МГТУ им Н.Э.Баумана pavel7697 Выйти



Название: RT
Год основания: 1890



Название: IU
Год основания: 1968



Название: MT