

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:

«Разработка Интернет Приложений»

ЛР7. Авторизация, работа с формами и Django Admin.

Исполнитель:

Студент группы РТ5-51

Чечнев А.А.

Преподаватель:

Гапанюк Ю. Е.

«____»_____



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.

Исходный код:

settings.py:

```
"""
Django settings for test5_full project.

Generated by 'django-admin startproject' using Django 1.11.7.

For more information on this file, see
https://docs.djangoproject.com/en/1.11/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/1.11/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '+glayd^@rojiipq)-qy33nk8i+2un!kjloug(r9l78(f+uw6!4'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['192.168.1.10', '127.0.0.1']

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp.apps.MyappConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
```

```

'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'test5_full.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'test5_full.wsgi.application'

# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'test5_db',
        'USER': 'als',
        'PASSWORD': ' ',
        'PORT': 3306,
        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# Password validation
# https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidato
r',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {

```

```

        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

```

# Internationalization
# https://docs.djangoproject.com/en/1.11/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.11/howto/static-files/

STATIC_URL = '/static/'

```

urls.py

```

from django.conf.urls import url
from django.contrib import admin
from myapp.views import *
# from django.conf.urls import include, url

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^main/', mainView, name="main_url"),
    url(r'^goods/$', GoodsListView.as_view(), name="goods_url"),
    url(r'^info/$', infoView, name="info_url"),
    url(r'^goods/add/$', goodsAddView, name="goods_add_url"),

    url(r'^register/$', RegisterFormView.as_view(), name="reg1"),
    url(r'^reg/$', registrationmy, name="reg_my_url"),
    url(r'^login/$', LoginFormView.as_view(), name="login_url"),
    url(r'^logout/$', LogoutView.as_view(), name="logout_url"),
    url(r'^qqq/$', RegisterFormView.as_view(), name="llogout_url"),

    url(r'^$', LoginFormView.as_view()),

]

```

forms.py

```
from django import forms
from myapp.models import Goods
from django.contrib.auth.forms import UserCreationForm

class GoodsForm(forms.ModelForm):
    class Meta:
        model = Goods
        exclude = []
```

views.py

```
from django.contrib.auth import login, logout
from django.contrib.auth.forms import AuthenticationForm
from django.contrib.auth.models import User
from django.http import HttpResponseRedirect
from django.shortcuts import render
from django.views.generic import ListView
from django.views.generic import View
from django.views.generic.edit import FormView

from .forms import *

def mainView(requests):
    title = "main"
    return render(requests, "main.html", locals())

def infoView(requests):
    title = "main"
    return render(requests, "info.html", locals())

class GoodsListView(ListView):
    model = Goods
    # model = model.objects.filter(good_id='1')
    # list_filter = ["good_id"]
    template_name = "goods.html"
    context_object_name = "obj_list"
    # queryset = "good_id=1"

    def get(self, request, *args, **kwargs):
        self.id = request.GET.get('id')
        return super(GoodsListView, self).get(request, *args, **kwargs)

    def get_queryset(self):
        if self.id:
            return Goods.objects.filter(good_id=self.id)
        else:
            return Goods.objects.all()

def goodsAddView(requests):
    main_text = "Добавить товар:"

    form = GoodsForm(requests.POST or None)
    log_in_is_success = False
```

```

result = ''

if requests.method == "POST" and form.is_valid():
    form.save()

    result = "Товар успешно добавлен!"

return render(requests, "goods_add.html", locals())

class RegisterFormView(FormView):
    form_class = UserCreationForm

    # Ссылка, на которую будет перенаправляться пользователь в случае
    # успешной регистрации.
    # В данном случае указана ссылка на страницу входа для зарегистрированных
    # пользователей.
    success_url = "/login/"

    # Шаблон, который будет использоваться при отображении представления.
    template_name = "log.html"

    def form_valid(self, form):
        # Создаём пользователя, если данные в форму были введены корректно.
        form.save()

        # Вызываем метод базового класса
        return super(RegisterFormView, self).form_valid(form)

class LoginFormView(FormView):
    form_class = AuthenticationForm

    # Аналогично регистрации, только используем шаблон аутентификации.
    template_name = "login.html"

    # В случае успеха перенаправим на главную.
    success_url = "/main/"

    def form_valid(self, form):
        # Получаем объект пользователя на основе введённых в форму данных.
        self.user = form.get_user()

        # Выполняем аутентификацию пользователя.
        login(self.request, self.user)
        return super(LoginFormView, self).form_valid(form)

class LogoutView(View):
    def get(self, request):
        # Выполняем выход для пользователя, запросившего данное
        # представление.
        logout(request)

        # После чего, перенаправляем пользователя на главную страницу.
        return HttpResponseRedirect("/main/")

def registrationmy(request):
    errors = {'username': '', 'password': '', 'password2': '', 'email': '',
'firstname': '', 'surname': ''}
    error_flag = False
    if request.method == 'POST':

```

```

username = request.POST.get('username')
if not username:
    errors['username'] = 'Введите логин'
    error_flag = True
elif len(username) < 5:
    errors['username'] = 'Логин должен превышать 5 символов'
    error_flag = True
elif User.objects.filter(username=username).exists():
    errors['username'] = 'Такой логин уже существует'
    error_flag = True

password = request.POST.get('password')
if not password:
    errors['password'] = 'Введите пароль'
    error_flag = True
elif len(password) < 8:
    errors['password'] = 'Длина пароля должна превышать 8 символов'

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

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

firstname = request.POST.get('firstname')
if not firstname:
    errors['firstname'] = 'Введите имя'
surname = request.POST.get('surname')
if not surname:
    errors['surname'] = 'Введите фамилию'

if not error_flag:
    # ...
    user = User.objects.create_user(username=username,
password=password, email=email, first_name=firstname,
last_name=surname)
    return HttpResponseRedirect('/login/')

return render(request, 'reg_my.html', locals())

```

admin.py

```

from django.contrib import admin
from .models import *

# Register your models here.
class GoodsAdmin(admin.ModelAdmin):
    list_display = ["good_id", "good_name"] # Выводит 2 поля

    # list_display = [field.name for field in Goods._meta.fields]
    # Выводит все поля
    list_filter = ["good_id"]
    # Фильтр по полям

    class Meta:
        model = Goods

```

```

class HorseAdmin(admin.ModelAdmin):
    list_display = ["horse_id", "horse_name"] # Выводит 2 поля

    # list_display = [field.name for field in Goods._meta.fields]
    # Выводит все поля
    # list_filter = ["good_id"]
    # Фильтр по полям

    class Meta:
        model = Horses

class RunAdmin(admin.ModelAdmin):
    # list_display = ["good_id", "good_name"] # Выводит 2 поля

    # list_display = [field.name for field in Goods._meta.fields]
    # Выводит все поля
    # list_filter = ["good_id"]
    # Фильтр по полям

    class Meta:
        model = Run

class RunsAdmin(admin.ModelAdmin):
    list_display = ["runs_run_id", "runs_horse_id"] # Выводит 2 поля

    # list_display = [field.name for field in Goods._meta.fields]
    # Выводит все поля

    class Meta:
        model = Runs

admin.site.register(Goods, GoodsAdmin)
admin.site.register(Horses, HorseAdmin)
admin.site.register(Run, RunAdmin)
admin.site.register(Runs, RunsAdmin)

```

base.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% load static %}

    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{{ title }}</title>
    <link rel="stylesheet" href="{% static 'Bootstrap/css/bootstrap.min.css' %}">
    <link rel="stylesheet" href="{% static 'Bootstrap/css/font-
awesome.min.css' %}">
    <link rel="stylesheet" href="{% static 'Bootstrap/css/main.css' %}">
</head>

```



```

<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#"><i class="fa fa-ellipsis-h"></i>{{
head_text }}</a>
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav navbar-right">

          {% block head_buttons %}{% endblock %}
          {% if request.user.is_authenticated %}
            <li>
              <a href="{% url 'logout_url' %}">Выйти ( {{
request.user.username }} )</a>
            </li>
          {% else %}
            <li> <a href="{% url 'login_url' %}">Вход</a> </li>
            <li> <a href="{% url 'reg_my_url' %}">Регистрация</a>
            <a href="{% url 'registration2_url'
{% #
%}">Регистрация2</a>#}
            </li>
          {% endif %}

        </ul>
      </div>
    </div>
  </div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></scri
pt>
<script src={% static "Bootstrap/js/bootstrap.min.js" %}></script>
<div id = "headerwrap">
  <div class="container">
    <div class="row centered">
      <div class="col-lg-10 col-lg-offset-1">
        {% block body1 %}{% endblock %}

      </div>
    </div>
  </div>
</div>

</body>

</html>

```

goods.html

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

```

{% block head_buttons %}

<li><a href={% url 'main_url' %}>Домой</a></li>
<li><a href={% url "info_url" %}>Инфо</a></li>
<li class="active"><a href={% url "goods_url"%}>Товары</a></li>
<li><a href="#">Контакты</a></li>
{#
    <li><a href={% url "good numb url" 1%}><i class="fa fa-envelope-o"
aria-hidden="true"></i></a></li>#}

{% endblock %}

{% block body1 %}

<div class="container">
    <div class="row">
        {% if request.user.is_authenticated %}
            <div class="col-lg-1 col-lg-offset-2">
                <table>

                    <caption class="fa-3x"><a href={% url "goods_url" %}>
Товары </a></caption>

                    <tr href="#" data-href-template="#" class="fa-2x">
                        <th>Номер</th>
                        <th>Название</th>
                        <th>Количество</th>
                    </tr>

                    {% for good in obj_list %}
                        <tr data-href-template={% url "goods_url"%}?id={{
good.good_id }}>
                            {#
                                {% url "#" unvier.good_id %}#}
                                <td><a href={% url "goods_url"%}?id={{
good.good_id }} > {{ good.good_id }} </a> </td>
                                <td><h4>{{ good.good_name }}</h4></td>
                                <td><h4>{{ good.good_count }}</h4></td>
                            </tr>
                        {% endfor %}

                    <div class="col-lg-2 col-lg-offset-7">
                        <h2>
                            <a href={% url "goods_add_url" %}>Добавить
товар</a>

                        </h2>
                    </div>
                </table>
            </div>

            {% else %}
                <div class="col-lg-12 col-lg-offset-0">
                    <h2>Чтобы просматривать таблицу, нужно
авторизоваться</h2>
                </div>
            {% endif %}
        </div>
    </div>
{% endblock %}

```

goods_add.html

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

{% block head_buttons %}

    <li><a href={% url 'main_url' %}>Домой</a></li>
    <li><a href={% url "info_url" %}>Инфо</a></li>
    <li><a href={% url "goods_url"%}>Товары</a></li>
    <li><a href="#">Контакты</a></li>
    {#      <li><a href={% url "good_numb_url" 1%}><i class="fa fa-envelope-o"
    aria-hidden="true"></i></a></li>#}

{% endblock %}

{% block body1 %}

    <div class="col-lg-2 col-lg-offset-3">
        <div id = "headerwrap1">
            <h2>{{ main_text }}</h2>
            <br><br>
            <form action="" method="post">{% csrf_token %}
                <h3>
                    {{ form.as_p }}
                </h3>

                <div>
                    <button type="submit" class="btn btn-
primary">Добавить</button>
                    <br><br>
                    {% if result != '' %}
                        <br>
                        <h1>
                            {% csrf_token %}
                            {{ result }}
                        <br><br><br>
                        </h1>
                    {% endif %}
                </div>
            </form>
        </div>

    </div>

{% endblock %}
```

info.html

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

{% block head_buttons %}

    <li><a href={% url 'main_url' %}>Домой</a></li>
    <li class="active"><a href="#">Инфо</a></li>
    <li><a href={% url "goods_url" %}>Товары</a></li>
    <li><a href="#">Контакты</a></li>
    <li><a href="#"><i class="fa fa-envelope-o" aria-
hidden="true"></i></a></li>
{% endblock %}
```

```
{% block body1 %}
    <div class="container">
        <div class="row centered">
            <div class="col-lg-8 col-lg-offset-2">
                <h1>THIS IS INFORMATION</h1>
            </div>
        </div>
    </div>

{% endblock %}
```

main.html

```
{% extends "base.html" %}

{% block head_buttons %}
    <li class="active"><a href="#">Домой</a></li>
    <li><a href={% url "info_url" %}>Инфо</a></li>
    <li><a href={% url "goods_url" %}>Товары</a></li>
    <li><a href="#">Контакты</a></li>
    <li><a href="#"><i class="fa fa-envelope-o" aria-
hidden="true"></i></a></li>

{% endblock %}

{% block body1 %}
    <div id="qwe">
        <h1> {{ main_text }} </h1>
        <div class="col-lg-2 col-lg-offset-9">
            <div id = "headerwrap1">
                <h1>Main uuuu</h1>
            </div>
        </div>
    </div>

{% endblock %}
```