

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Домашнее задание по курсу:

**«Разработка Интернет Приложений»**

Исполнитель:

Студент группы РТ5-51

Макаров А.В.

Преподаватель:

Гапанюк Ю. Е.

« \_\_\_\_ » \_\_\_\_\_



Москва 2017 г.

## Задание и порядок выполнения

Разработать вебсервис на базе технологий: Python, Django, JS, MySQL

№	Субъект	Отношение	Объект
10	Пользователь	Бронь	Концерт

## Исходный код:

### settings.py

```
"""
Django settings for Lab7 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
from django.core.urlresolvers import reverse_lazy

LOGIN_REDIRECT_URL = reverse_lazy('dashboard')
LOGIN_URL = reverse_lazy('login')
LOGOUT_URL = reverse_lazy('logout')

# 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 = 'w-h)wemw%23i5o%2k6d1&6%!uns!hb&(%eg6vv^$&200j(8p9u'

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

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    'myapp'
]

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 = 'Lab7.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 = 'Lab7.wsgi.application'

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

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'userbronkoncert'),
    }
}

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

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        '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

```

"""Lab7 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/1.11/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.conf.urls import url, include
    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
"""
from django import views
from django.conf.urls import url
from django.contrib import admin
from myapp.views import *

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^register/', RegisterFormView.as_view(), name="reg_url"),
    url(r'^login/', LoginFormView.as_view(), name="login_url"),
    url(r'^logout/', LogoutView.as_view(), name="logout_url"),
    url(r'^tableuser/', UserList2.as_view(), name="goods_url"),
    url(r'^edituser/', concerts_add, name="edit_url"),
    url(r'^adduser/', user_add, name="add_posts"),
]

```

## init.py

## admin.py

```

from django.contrib import admin

from .models import *

admin.site.register(User2)
admin.site.register(Concert)

```

## forms.py

```
from django import forms
from myapp.models import *

class UsForm(forms.ModelForm):
    class Meta:
        model = User2
        fields = ["name", "idconcert"]
        widgets = {'information': forms.Textarea(attrs={'resize': 'none'})}

class CoForm(forms.ModelForm):
    class Meta:
        model = Concert
        fields = ["name"]
        widgets = {'information': forms.Textarea(attrs={'resize': 'none'})}
```

## models.py

```
from django.db import models

class User2(models.Model):
    id = models.AutoField(primary_key=True,) # ТИП ПОЛЯ
    name = models.CharField(max_length=255)
    idconcert = models.ForeignKey('Concert')

class Concert(models.Model):
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30)
```

## views.py

```
from django.shortcuts import redirect
from django.core.urlresolvers import reverse
from django.views.generic import ListView, FormView
from myapp.models import *
from myapp.forms import *
from django.contrib.auth.forms import UserCreationForm

# Опять же, спасибо django за готовую форму аутентификации.
from django.contrib.auth.forms import AuthenticationForm

# Функция для установки сессионного ключа.
# По нему django будет определять, выполнил ли вход пользователь.
from django.contrib.auth import login
from django.shortcuts import render, HttpResponseRedirect
from django.views.generic.base import View
from django.contrib.auth import logout

class UserList2(ListView):
    model = User2
    template_name = "user2_list.html"
    context_object_name = "obj_list"

    def get(self, request, *args, **kwargs):
        self.id = request.GET.get('id')
        return super(UserList2, self).get(request, *args, **kwargs)

def concerts(request, course_id=6):
    if request.method == 'POST':
        Concert.objects.get(id_course=course_id).course_users.add(request.user)
    return render(request, "concert.html", {'concerts':
Concert.objects.get(id_course=course_id)}, locals())

def concerts_add(request):
    if request.method == "POST":
        form = CoForm(request.POST)
        if form.is_valid():
            post = form.save()
            post.name = request.POST.get('name')
            post.save()
            return HttpResponseRedirect("/tableuser/")
    else:
        form = CoForm()
    return render(request, 'newedit.html', {'form': form})

def user_add(request):
    Users = User2.objects.all()
    Concerts = Concert.objects.all()
    form = UsForm(request.POST)
    context = {"form": form, "Users": Users, "Concerts": Concerts}
    if form.is_valid():
        post = form.save()
        post.name = request.POST.get('name')
        post.idconcert = request.POST.get('Concert')
        post.save()
        return HttpResponseRedirect("/tableuser/")
    return render(request, 'addpost.html', context)

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("/login/")


class LoginFormView(FormView):
    form_class = AuthenticationForm

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

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

    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 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 = "register.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)

```

## base.html

```

<!DOCTYPE html>
<html lang="ru">
{% load static %}
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
alpha.6/css/bootstrap.min.css" integrity="sha384-
rwoIResjU2yc3z8GV/NPeZWAv56rSmLldC3R/AZzGRnGxQQKnKkoFVhFQhNUwEyJ"
crossorigin="anonymous">
<link rel="stylesheet" href="{% static 'css/bootstrap.css' %}">
<head class="navbar">
    {% block scripts %}{% endblock %}
</head>
<body>
<nav class="navbar-toggler-right" role="navigation">
    <meta charset="UTF-8">
    {% block head_buttons %}{% endblock %}
</nav>

```



```

        <div>
            {% block body1 %}Переопределяется в наследниках{% endblock
    %}

        </div>
    </ul>
    <script src="https://code.jquery.com/jquery-3.1.1.slim.min.js"
    integrity="sha384-
    A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k90mg/a/EheAdgtzNs3hpfag6Ed950n"
    crossorigin="anonymous"></script>
    <script
    src="https://cdnjs.cloudflare.com/ajax/libs/tether/1.4.0/js/tether.min.js"
    integrity="sha384-
    DzttdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bBrb"
    crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0-
    alpha.6/js/bootstrap.min.js" integrity="sha384-
    vBWWzLZJ8ea9aCX4pEW3rVHjgjt7zpkNpZk+02D9phzyeVKE+jo0ieGizqPLForn"
    crossorigin="anonymous"></script>
</body>
</html>

```

### userlist.html

```

% extends 'base.html' %}
{% block head_buttons %}
    <div class="row">
        {% if request.user.is_authenticated %}
            <a href={% url 'logout_url' %}>ВЫЙТИ</a>
        {% else %}
            <a href={% url 'login_url' %}>ВОЙТИ</a>
            <ul></ul>
            <a href={% url 'reg_url' %}>регистрация</a>
        {% endif %}
    </div>
{% endblock %}
{% block body1 %}
    {% if request.user.is_authenticated %}
        <hr>
        <table>
            <caption class="fa-3x"><a href={% url "edit_url"
%}>Добавить пользователя </a></caption>
            <tr href="#" data-href-template="#" class="fa-2x">
                <th>Номер</th>
                <th>Пользователь</th>
                <th>Название концерта</th>
            </tr>

            {% for user in obj_list %}
                <tr data-href-template={% url "goods_url"%}?id={{
user.id }}>
                    <td><a href={% url "goods_url"%}?id={{ user }} >
                        {{ user.id }} </a> </td>
                    <td><h4>{{ user.name }}</h4></td>
                    <td><h4>{{ user.idconcert.name }}</h4></td>
                </tr>
            {% endfor %}
        </table>

        {% else %}
            <hr>
            <div>
                <h2>Чтобы просматривать таблицу, нужно
авторизоваться</h2>
            </div>

```

```
{ % endif % }
{ % endblock % }
```

ВЫЙТИ

Номер	Пользователь	Название концерта
1	sdvghgfg	Чайковский
4	Андрей	Чайковский
5	Николай	Питон
6	Куцкен	Чайковский
12	sdfasdfndsrftghjukjghtr	Чкалов

Добавить пользователя

## neweit.html

```
{% extends 'base.html' %}
{% block head_buttons %}
    <div class="row">
        {% if request.user.is_authenticated %}
            <a href={% url 'logout_url' %}>ВЫЙТИ</a>
        {% else %}
            <a href={% url 'login_url' %}>ВОЙТИ</a>
        <ul></ul>
        <a href={% url 'reg_url' %}>регистрация</a>
        {% endif %}
    </div>
{% endblock %}
{% block scripts %}

{% endblock %}
```

Название концерта:

Добавить концерт

Ссылка

## addpost.html

```
{% extends 'base.html' %}
{% block head_buttons %}
    <div class="row">
        {% if request.user.is_authenticated %}
            <a href={% url 'logout_url' %}>ВЫЙТИ</a>
        {% else %}
            <a href={% url 'login_url' %}>ВОЙТИ</a>
        <ul></ul>
        <a href={% url 'reg_url' %}>регистрация</a>
        {% endif %}
    </div>
```

```
{% endblock %}
{% block scripts %}

{% endblock %}
```

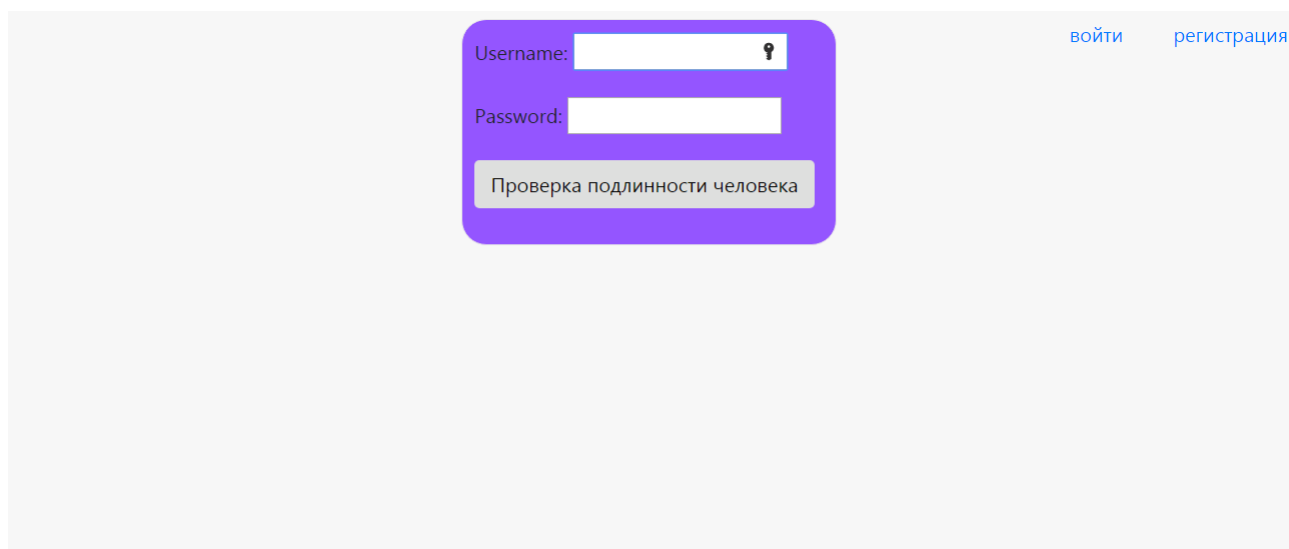
```
{% block body1 %}
<hr>
<form method='post'>
    {% csrf_token %}
    <div class="concert_add">
        {% csrf_token %}
        <div class="user_add" enctype="multipart/form-data">
            <label for="id_name">Имя зрителя:</label>
            <div class="data-and-error">
                <input type="text" name="name" maxlength="100"
required id="id_name"/>
            </div>
        </div>

        <div class="user_add">
            <label for="id_subject">Категория:</label>
            <div class="data-and-error">
                <select name="idconcert">
                    <option value="" selected>Не выбрано</option>
                    {% for subject in Concerts %}
                        <option value="{{ subject.id }}">{{
subject.name }}</option>
                    {% endfor %}
                </select>
                <h5 id="error_subject_course">Выберете концерт</h5>
            </div>
        </div>
        <button type="submit" id="btn_add">Бронь</button>
    </form>

{% endblock %}
```

## login.html

```
{% extends 'base.html' %}
{% block head_buttons %}
<title>Авторизация</title>
<style type="text/css">
body {
padding-top: 40px;
padding-bottom: 40px;
background-color: #f5f5f5;}
.form-signin {
max-width: 300px;
padding: 10px 10px 29px;
margin: 0 auto 20px;
background-color: #9556ff;
border: 1px solid #e5e5e5;
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
border-radius: 20px;
}
</style>
<div class="row">
<a href={% url 'login_url' %}>войти</a>
<ul></ul>
<a href={% url "reg_url" %}>регистрация</a>
</div>
{% endblock %}
{% block body1 %}
<form class="form-signin" method="post">
{% csrf_token %}
<!-- as p для того, чтобы каждый элемент формы был с новой строки -->
{{ form.as_p }}
<button
class="btn btn-large"
type="submit">
Проверка подлинности человека
</button>
</form>
{% endblock %}
```



Username:

Password:

[войти](#) [регистрация](#)

## registration.html

```
{% extends 'base.html' %}
{% block head_buttons %}
<title>Регистрация</title>
<style type="text/css">
  body {
    padding-top: 40px;
    padding-bottom: 40px;
    background-color: #f5f5f5;
  }
  .form-signin {
    max-width: 500px;
    padding: 15px 15px 29px;
    margin: 0 auto 20px;
    background-color: #9556ff;
    border: 1px solid #e3e5ce;
    -webkit-border-radius: 20px;
    -moz-border-radius: 20px;
    border-radius: 20px;
  }
</style>
<div class="row">
  <a href={% url 'login_url' %}>войти</a>
  <ul></ul>
  <a href={% url "reg_url" %}>регистрация</a>
</div>
{% endblock %}
{% block body1 %}
<form class="form-signin" method="post">
  {% csrf_token %}
  <!-- as p для того, чтобы каждый элемент формы был с новой строки -->
  {{ form.as_p }}
  <button
    class="btn btn-large"
    type="submit"
  >
    Зарегистрироваться
  </button>
</form>
{% endblock %}
```

[войти](#)[регистрация](#)

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

[Зарегистрироваться](#)