

Федеральное государственное бюджетное образовательное учреждение высшего  
профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)

---

Отчет по домашнему заданию  
по дисциплине «Разработка интернет приложений».  
Вариант 8.

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53

Костенкова Ю.В.

Москва, 2017

# Задание

Разработать веб-сервис на базе технологий: Python, Django, JS, MySQL.

## Код операции

### Urls.py

```
"""dz_v2 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.conf.urls import url
from django.contrib import admin
from dz2 import views
from django.conf.urls.static import static

from django.contrib.staticfiles.urls import staticfiles_urlpatterns

from django.conf import settings
urlpatterns = [

    url(r'^admin/', admin.site.urls),
    url(r'^register/$', views.RegisterFormView.as_view()),
    url(r'^login/$', views.LoginFormView.as_view()),
    url(r'^logout/$', views.LogoutView.as_view()),
    url(r'^groups/$', views.musical_groups_view.as_view()),
    url(r'^group/(?P<id1>\d+)\$', views.group_view.as_view(),
name='group_url'),
    url(r'^create_group/$', views.create_group.as_view()),
    url(r'^add/(?P<id2>\d+)\$', views.add.as_view(), name='add_url'),
]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)+
static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

urlpatterns.append(url(r'^$', views.musical_groups_view.as_view()))
```

### Models.py

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

def avatar_upload_to(instance, filename):
```

```

        return os.path.join('uploads', instance.name +
os.path.splitext(filename)[1])

class Musical_group(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=30)
    style = models.CharField(max_length=50)
    year = models.CharField(max_length=6)
    country = models.CharField(max_length=50)
    description = models.CharField(max_length=255)
    picture = models.ImageField(upload_to=avatar_upload_to)
# Create your models here.

class Group_user(models.Model):
    group_id = models.ForeignKey(Musical_group, on_delete=models.CASCADE)
    user_id = models.ForeignKey(User, on_delete=models.CASCADE)

```

## Views.py

```

from django.shortcuts import render
from django.views.generic.edit import FormView
#from django.contrib.auth.forms import UserCreationForm
# Опять же, спасибо django за готовую форму аутентификации.
#from django.contrib.auth.forms import AuthenticationForm
from dz2.form import UserCreationForm
from dz2.form import AuthenticationForm
# Функция для установки сессионного ключа.
# По нему django будет определять, выполнил ли вход пользователь.
from django.contrib.auth import login
from django.http import HttpResponseRedirect
from django.views.generic.base import View
from django.views.generic.list import ListView
from django.contrib.auth import logout
from dz2 import models
from django.contrib.auth.models import User
from dz2 import form
from django.shortcuts import redirect, reverse
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger

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("/")

class LoginFormView(FormView):
    form_class = AuthenticationForm

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

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

    def form_valid(self, form):
        # Получаем объект пользователя на основе введенных 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)
# Create your views here.

class musical_groups_view(ListView):
    model = models.Musical_group

    def get(self, request):
        d = self.model.objects.values('id', 'name', 'picture')
        paginator = Paginator(d, 5) # Show 25 contacts per page
        page = request.GET.get('page')
        try:
            contacts = paginator.page(page)
        except PageNotAnInteger:
            # If page is not an integer, deliver first page.
            contacts = paginator.page(1)
        except EmptyPage:
            # If page is out of range (e.g. 9999), deliver last page of
            results.
            contacts = paginator.page(paginator.num_pages)
        if (contacts.paginator.num_pages <= 4):
            r = range(1, contacts.paginator.num_pages+1)
        else:
            r = range(contacts.number-1, contacts.number+4)
        return render(request, 'groups.html', {"contacts": contacts, 'user':
request.user, 'range': r})

class group_view(View):
    model = models.Musical_group
    model1 = models.Group_user
    model2 = User
    def get(self, request, id1):

        b = dict(group=self.model.objects.filter(id=id1))

        a = dict(users=self.model1.objects.filter(group_id=id1))

        b.update(a)
        c={'user':request.user}
        b.update(c)
        d={'id':id1}
        b.update(d)
        return render(request, "inform_group.html", b)

class create_group(FormView):
    form_class = form.Musical_group_form
    model = models.Musical_group

```

```

template_name = "create_group.html"
success_url = "/groups/"
def form_valid(self, form):
    # Создаём пользователя, если данные в форму были введены корректно.
    form.save()

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

def post(self, request, *args, **kwargs):
    form1 = form.Musical_group_form(request.POST)
    if form1.is_valid() and request.FILES.get('picture'):
        obj = models.Musical_group()
        obj = form1.save(commit=False)
        obj.picture = request.FILES['picture']
        obj.save()
        b = dict(group=self.model.objects.filter(name=obj.name))
        return HttpResponseRedirect("/group/"+str(b['group'][0].id))
    else:
        form1.add_error('picture', 'Добавьте логотип')
    return render(request, 'create_group.html', {'form':form1})

class add(View):
    model = models.Musical_group
    model1 = models.Group_user
    model2 = User
    def get(self, request, id2):
        if request.user.is_authenticated():
            u = self.model2.objects.filter(id=request.user.id)
            a = dict(users=self.model1.objects.filter(group_id=id2))
            b = dict(group=self.model.objects.filter(id=id2))
            ab = dict(users=self.model1.objects.filter(group_id=id2))
            b.update(ab)
            er = 0
            for i in a['users']:
                if i.user_id == request.user:
                    er = 1
                    break
            if er == 1:
                ca = {'uze': "Вы уже вступили в эту группу"}
                b.update(ca)
                k = {'message':''}
                ca = {'error':''}
                b.update(k)
                b.update(ca)
                d = {'id': id2}
                b.update(d)
                return render(request, "inform_group.html", b)
            else:
                a=models.Group_user()

                a.group_id=b['group'][0]
                a.user_id=u[0]
                a.save()

                k = {'message':"Вы успешно вступили в группу"}
                ca = {'error':''}
                b.update(k)
                b.update(ca)
                d = {'id': id2}
                b.update(d)

                return render(request, "inform_group.html", b)

```

```

else:
    b = dict(group=self.model.objects.filter(id=id2))
    a = {'error': "Вы не авторизованны. Войдите или
зарегистрируйтесь"}
    c = {'message':''}
    b.update(a)
    b.update(c)
    d = {'id': id2}
    b.update(d)
    ab = dict(users=self.model.objects.filter(group_id=id2))
    b.update(ab)
    return render(request, "inform_group.html", b)

```

## login.html

```

{% extends 'base.html' %}
{% load static %}
{% block title %}Вход{% endblock %}
{% block body %}
    <nav class="navbar navbar-default color-nav">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="/">
                    
                </a>
            </div>

            <!-- Collect the nav links, forms, and other content for toggling
-->
            <div class="collapse navbar-collapse" id="bs-example-navbar-
collapse-1">
                <ul class="nav navbar-nav navbar-right color-text">
                    <li><a class="color-text" href="/create_group/">Добавить
группу</a></li>
                    {% if user.is_authenticated %}
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle color-text
color-nav" data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">{{ user.username }} <span class="caret"></span></a>
                        <ul class="dropdown-menu color-text color-nav">
                            <li><a href="/logout/">Выйти</a></li>
                        </ul>
                    </li>
                    {% else %}
                    <li class="dropdown">
                        <a href="#" class="dropdown-toggle color-text
color-nav" data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">Вход или регистрация <span class="caret"></span></a>
                        <ul class="dropdown-menu color-text color-nav">
                            <li><a href="/login/">Войти </a></li>
                            <li><a href="/register/">Регистрация</a></li>
                        </ul>
                    </li>
                    {% endif %}
                </ul>
            </div>
        </div>
    </nav>

```

```

        </ul>
    </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div>
    <form class="form-signin" method="POST">
        {% csrf_token %}
        {% if form.non_field_errors %}
            {% for error in form.non_field_errors %}
                <div class="error">
                    {{ error|escape }}
                </div>
            {% endfor %}
        {% endif %}
        <div class="form-group row">
            <label for="example-text-input" class="col-xs-2 col-form-
label">Логин</label>
            <div class="col-xs-10">
                {{ form.username }}
            </div>
        </div>
        <div class="form-group row">
            <label for="example-text-input" class="col-xs-2 col-form-
label">Пароль</label>
            <div class="col-xs-10">
                {{ form.password }}
            </div>
        </div>
        <button class="btn btn-lg btn-primary btn-block"
type="submit">Войти</button>
    </form>
</div>
{% endblock %}

```

## Creat\_group.html

```

{% extends 'base.html' %}
{% load static %}
{% block title %}Добавление группы{% endblock %}
{% block body %}
    <nav class="navbar navbar-default color-nav">
        <div class="container-fluid">
            <!-- Brand and toggle get grouped for better mobile display -->
            <div class="navbar-header">
                <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-
expanded="false">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand" href="/">
                    
                </a>
            </div>
            <!-- Collect the nav links, forms, and other content for toggling
-->
                <div class="collapse navbar-collapse" id="bs-example-

```

```

navbar-collapse-1">
    <ul class="nav navbar-nav navbar-right color-text">
<li><a class="color-text" href="/create_group/">Добавить группу</a></li>
        {% if user.is_authenticated %}
            <li class="dropdown">
                <a href="#" class="dropdown-toggle color-text
color-nav" data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">{{ user.username }} <span class="caret"></span></a>
                <ul class="dropdown-menu color-text color-nav">
                    <li><a href="/logout/">Выйти</a></li>
                </ul>
            </li>
            {% else %}
            <li class="dropdown">
                <a href="#" class="dropdown-toggle color-text
color-nav" data-toggle="dropdown" role="button" aria-haspopup="true" aria-
expanded="false">Вход или регистрация <span class="caret"></span></a>
                <ul class="dropdown-menu color-text color-nav">
                    <li><a href="/login/"> Войти </a></li>
                    <li><a href="/register/">Регистрация</a></li>
                </ul>
            </li>

            {% endif %}

        </ul>
    </div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>
<div>
    <form class="form-signin" method="POST" enctype="multipart/form-
data">
        {% csrf_token %}
        {% if form.non_field_errors %}
            {% for error in form.non_field_errors %}
                <div class="error">
                    {{ error|escape }}
                </div>
            {% endfor %}
        {% endif %}
        {% if form.name.errors %}
            {% for error in form.name.errors %}
                <div class="error">
                    {{ error|escape }}
                </div>
            {% endfor %}
        {% endif %}
        <div class="form-group row">
            <label for="example-text-input" class="col-xs-2 col-form-
label">Название</label>
            <div class="col-xs-10">
                {{ form.name }}
            </div>
        </div>
        {% if form.style.errors %}
            {% for error in form.style.errors %}
                <div class="error">
                    {{ error|escape }}
                </div>
            {% endfor %}
        {% endif %}
    </form>

```



```

<div class="form-group row">
  <label for="example-text-input" class="col-xs-2 col-form-
label">Стиль</label>
  <div class="col-xs-10">
    {{ form.style }}
  </div>
</div>
{% if form.year.errors %}
  {% for error in form.year.errors %}
    <div class="error">
      {{ error|escape }}
    </div>

  {% endfor %}
{% endif %}
<div class="form-group row">
  <label for="example-text-input" class="col-xs-2 col-form-
label">Год создания</label>
  <div class="col-xs-10">
    {{ form.year }}
  </div>
</div>
{% if form.country.errors %}
  {% for error in form.country.errors %}
    <div class="error">
      {{ error|escape }}
    </div>

  {% endfor %}
{% endif %}
<div class="form-group row">
  <label for="example-text-input" class="col-xs-2 col-form-
label">Страна</label>
  <div class="col-xs-10">
    {{ form.country }}
  </div>
</div>
{% if form.description.errors %}
  {% for error in form.description.errors %}
    <div class="error">
      {{ error|escape }}
    </div>

  {% endfor %}
{% endif %}
<div class="form-group row">
  <label for="example-text-input" class="col-xs-2 col-form-
label">Описание</label>
  <div class="col-xs-10">
    {{ form.description }}
  </div>
</div>
{% if form.picture.errors %}
  {% for error in form.picture.errors %}
    <div class="error">
      {{ error|escape }}
    </div>

  {% endfor %}
{% endif %}
<div class="form-group row">
  <label class="col-xs-2 col-form-label">Логотип</label>
  <div class="col-xs-10">
    {{ form.picture }}
  </div>
</div>

```

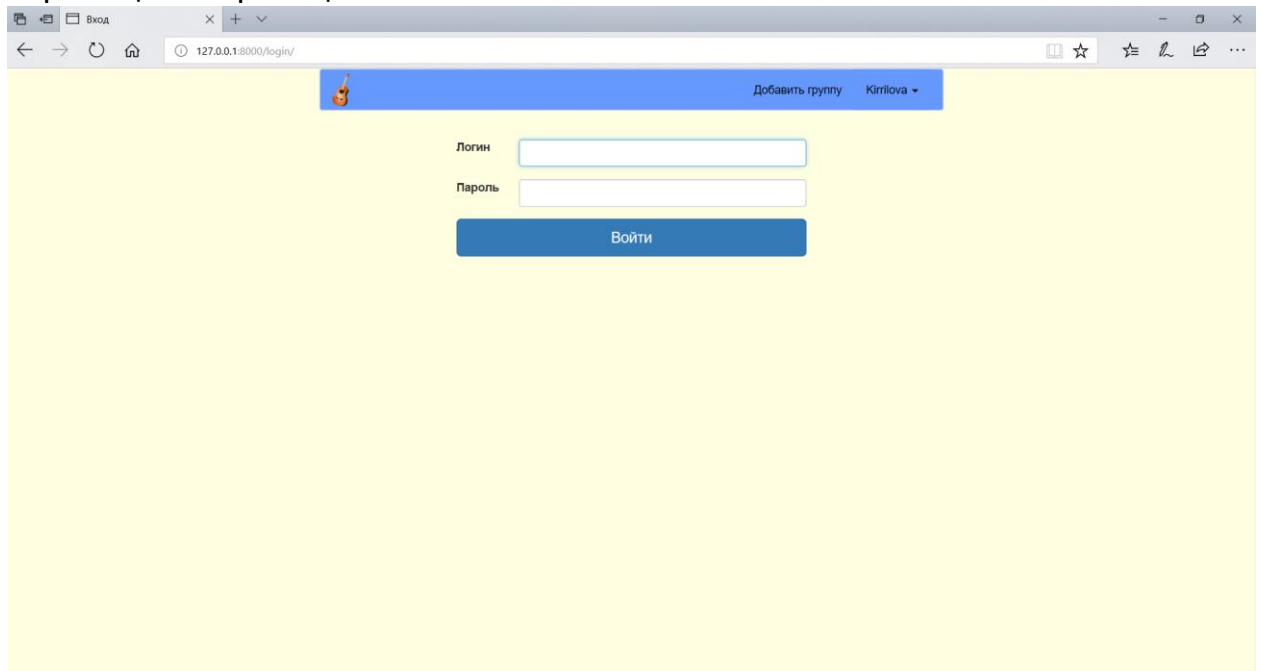
```

        </div>
    </div>
    <button class="btn btn-lg btn-primary btn-block"
type="submit">Создать группу</button>
</form>
</div>
{% endblock %}

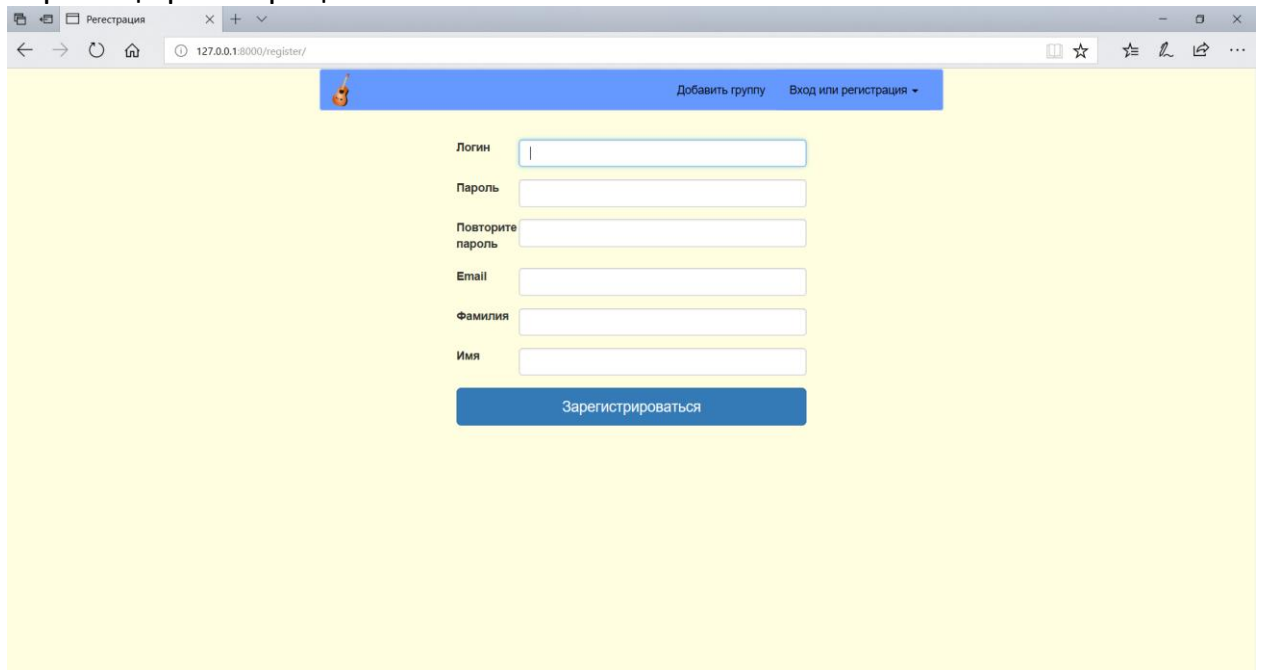
```

## Результаты работы

### Страница авторизации



### Страница регистрации



### Страница создания объекта

Добавление группы

Добавить группу Вход или регистрация

Название

Стиль

Год создания

Страна

Описание

Логотип  Обзор...

Создать группу

## Страница со списком объектов

Музыкальные группы

Добавить группу Вход или регистрация

Битлз

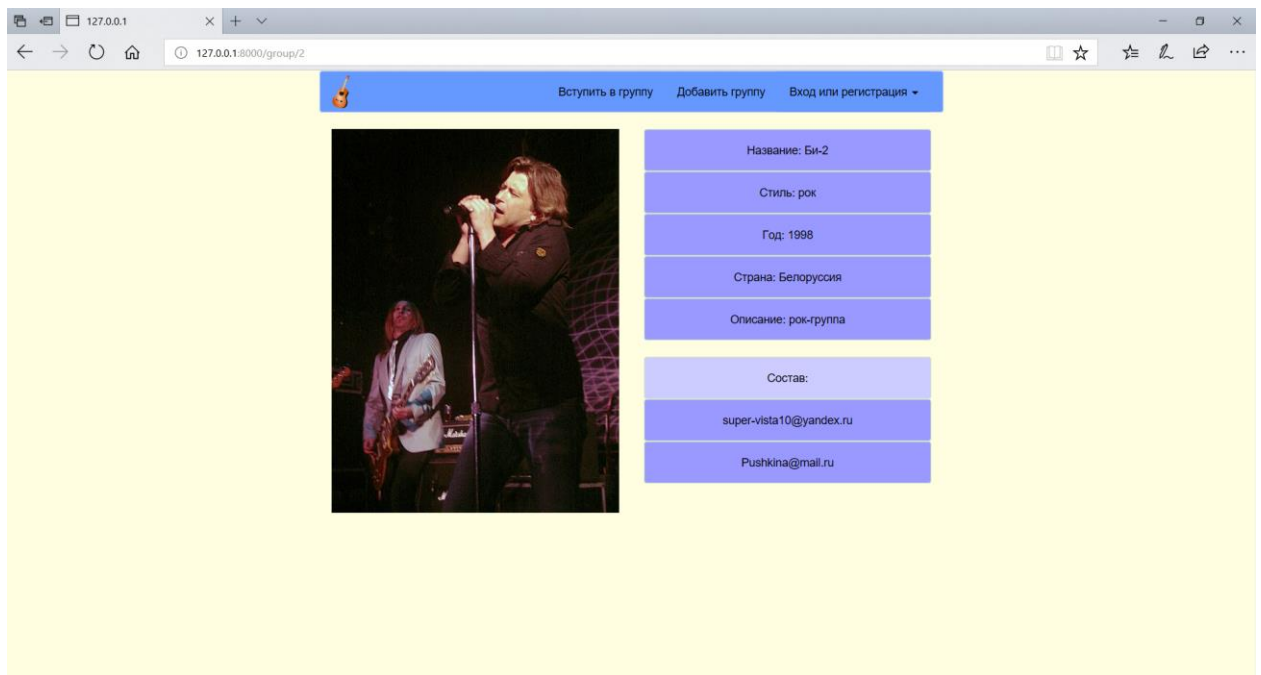
Би-2

Ленинград

Пятница

Нирвана

## Страница с описанием объекта



## Страница администрирования

