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

**Отчет по домашнему заданию**

**По курсу “Разработка интернет-приложений”**

Выполнил:

Студент группы ИУ5-51

Яковлев О.Я.

Вариант 26

Субъект: Пользователь

Отношение: Подписка

Объект: Канал

Views.py

from django.db.models import SET  
from django.shortcuts import render  
from django.http import HttpResponseRedirect  
from django.views.generic import View, DetailView  
from .models import \*  
from django import forms  
from django.contrib.auth import authenticate, login, logout  
from django.contrib.auth.decorators import login\_required  
from datetime import datetime  
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger  
  
  
def main(request):  
 user = request.user  
 par = {  
 'header': 'Main page',  
 'user': user  
  
 }  
 return render(request, 'MainPage.html', context=par)  
  
  
class ChannelForm(forms.ModelForm):  
 class Meta(object):  
 model = Channel  
 fields = ['channel\_name', 'rating', 'type', 'videos', 'picture', 'ChannelId']  
  
 def save(self):  
 channel = Channel()  
 channel.channel\_name = self.cleaned\_data.get('name')  
 channel.rating = self.cleaned\_data.get('price')  
 channel.type = self.cleaned\_data.get('type')  
 channel.videos = self.cleaned\_data.get('quantity')  
 picture = self.cleaned\_data.get('picture')  
 channel.picture = picture  
 channel.save()  
  
  
class RegistrationForm(forms.Form):  
 username = forms.CharField(min\_length=5, label='Логин')  
 password = forms.CharField(min\_length=6, widget=forms.PasswordInput, label='Пароль')  
 password2 = forms.CharField(min\_length=6, widget=forms.PasswordInput, label='Повторите ввод')  
 last\_name = forms.CharField(label='Фамилия')  
 first\_name = forms.CharField(label='Имя')  
 email = forms.EmailField(label='Email')  
  
  
def registration\_1(request):  
 form = RegistrationForm(request.POST or None)  
 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:  
 new\_user = User.objects.create\_user(data['username'], data['email'], data['password'])  
 print(new\_user)  
 user1 = User1()  
 user1.user1 = new\_user  
 user1.email = data['email']  
 user1.last\_name = data['last\_name']  
 user1.first\_name = data['first\_name']  
 user1.save()  
 return HttpResponseRedirect('/login1')  
 else:  
 form = RegistrationForm()  
 return render(request, 'registration\_1.html', {'form': form})  
  
  
def registration\_form(request):  
 errors = {}  
 if request.method == 'POST':  
 last\_name = request.POST.get('last\_name')  
 if not last\_name:  
 errors['last\_name'] = 'Введите Фамилию'  
  
 first\_name = request.POST.get('first\_name')  
 if not first\_name:  
 errors['first\_name'] = 'Введите имя'  
  
 email = request.POST.get('Email')  
 if not email:  
 errors['Email'] = 'Введите Email'  
  
 username = request.POST.get('username')  
 if not username:  
 errors['username'] = 'Введите логин'  
 elif len(username) < 8:  
 errors['username'] = 'Логин должен превышать 5 символов'  
 if User.objects.filter(username=username).exists():  
 errors['username'] = 'Данный логин занят'  
  
 password = request.POST.get('password')  
 if not password:  
 errors['password'] = 'Введите пароль'  
 elif len(password) < 8:  
 errors['password'] = 'Длина пароля должна превышать 6 символов'  
 password\_repeat = request.POST.get('password2')  
 if password != password\_repeat:  
 errors['password\_repeat'] = 'Пароли должны совпадать'  
 print(username, password, "1")  
  
 if not errors:  
 new\_user = User.objects.create\_user(username, email, password)  
 print(new\_user)  
 user1 = User1()  
 user1.user1 = new\_user  
 user1.email = email  
 user1.last\_name = last\_name  
 user1.first\_name = first\_name  
 user1.save()  
 return HttpResponseRedirect('/login2')  
 else:  
 context = {'errors': errors, 'username': username, 'email': email, 'last\_name': last\_name,  
 'first\_name': first\_name, 'password': password, 'password\_repeat': password\_repeat}  
 return render(request, 'registration.html', context)  
 return render(request, 'registration.html', {'errors': errors})  
  
  
class LoginForm(forms.Form):  
 username = forms.CharField(label='Логин')  
 password = forms.CharField(widget=forms.PasswordInput, label='Пароль')  
  
  
def log\_in(request):  
 errors = {}  
 if request.method == 'POST':  
 username = request.POST.get('username')  
 if not username:  
 errors['username'] = 'Введите логин'  
 elif len(username) < 5:  
 errors['username'] = 'Слишком короткий логин. Минимальная длина-5 знаков'  
  
 password = request.POST.get('password')  
 if not password:  
 errors['password'] = 'Введите пароль'  
 elif len(password) < 8:  
 errors['password'] = 'Слишком короткий пароль. Минимальная длина-8 знаков'  
  
 user = authenticate(request, username=username, password=password)  
 print(user)  
 if user is None and 'username' not in errors.keys() and 'password' not in errors.keys():  
 errors['login'] = 'Логин или пароль введены неверно'  
  
 if not errors:  
 login(request, user)  
 return HttpResponseRedirect('/channels')  
 else:  
 context = {'errors': errors}  
 return render(request, 'UserLogin.html', context)  
 return render(request, 'UserLogin.html', {'errors':errors})  
  
  
def log\_in1(request):  
 if request.method == 'POST':  
 form = LoginForm(request.POST)  
 print(form)  
 data = form.cleaned\_data  
  
 if form.is\_valid():  
 user = authenticate(request, username=data['username'], password=data['password'])  
 print(len(data['username']), len(data['password']))  
 if user is not None:  
 login(request, user)  
 return HttpResponseRedirect('/channels')  
 else:  
 form.add\_error('username', ['Неверный логин или пароль'])  
 else:  
 form = LoginForm()  
 return render(request, 'UserLogin\_1.html', {'form': form})  
  
  
def logout\_view(request):  
 logout(request)  
 return render(request, 'MainPage.html')  
  
  
@login\_required(login\_url='/login2')  
def logged\_in(request):  
 return render(request, 'items.html')  
  
  
def logged\_in\_view(request):  
 if request.user.is\_authenticated:  
 return render(request, 'items.html')  
 else:  
 return HttpResponseRedirect('/login1')  
  
  
def new\_channel(request):  
 errors = {}  
 if request.method == 'POST':  
 channel\_name = request.POST.get('channel\_name')  
 if not channel\_name:  
 errors['channel\_name'] = 'Введите Название Канала'  
 if Channel.objects.filter(channel\_name=channel\_name).exists():  
 errors['channel\_name'] = 'Данный канал уже существует'  
  
 rating = request.POST.get('rating')  
 if not rating:  
 errors['rating'] = 'Введите рейтинг'  
 if Channel.objects.filter(rating=rating).exists():  
 errors['rating'] = 'Команда с таким значением рейтинга уже существует'  
  
 type = request.POST.get('type')  
 if not type:  
 errors['type'] = 'Введите тип канала'  
  
 videos = request.POST.get('videos')  
 if not videos:  
 errors['videos'] = 'Введите кол-во видео канала'  
  
 picture = request.FILES.get('picture')  
 if not picture:  
 errors['picture'] = 'Загрузите фото'  
 if not errors:  
 channel = Channel(channel\_name=channel\_name, rating=rating, type=type, videos=videos,  
 picture=picture)  
 channel.save()  
 ChannelId = channel.ChannelId  
 return HttpResponseRedirect('/channel/' + str(ChannelId))  
 else:  
 context = {'errors': errors, 'channel\_name': channel\_name, 'rating': rating, 'type': type,  
 'videos': videos, 'picture': picture}  
  
 return render(request, 'new\_item.html', locals())  
  
  
class ChannelsView(View):  
 def get(self, request):  
 dict\_users = {}  
 channels = Channel.objects.all()  
 channels1 = Channel.objects.all()  
 form = ChannelForm()  
 paginator = Paginator(channels, 4)  
 page = request.GET.get('page')  
 try:  
 channels = paginator.page(page)  
 except PageNotAnInteger:  
 channels = paginator.page(1)  
 except EmptyPage:  
 channels = paginator.page(paginator.num\_pages)  
 return render(request, 'items.html', context={'channels': channels, 'users': dict\_users, 'form': form,  
 'channels1': channels1})  
  
  
class ChannelObject(DetailView):  
 model = Channel  
 context\_object\_name = 'ChannelObject'  
 template\_name = 'obj.html'  
  
  
def new\_subscriber(request, channel):  
 errors = {}  
 if request.method == 'POST':  
  
  
 subc = Subc()  
 uid = request.user.id  
 user = User.objects.get(id=uid)  
 subc.user = user  
 subc\_id = subc.id  
 if Channel.objects.filter(subc\_\_user=user).exists():  
 errors['username'] = 'Вы уже подписанны на данный канал'  
 channel1 = Channel.objects.get(ChannelId=int(channel))  
 date = datetime.today()  
 subc.date = date  
 if not errors:  
 subc.save()  
 subc2 = SubcChannel()  
 subc2.channel\_id = channel1.ChannelId  
 subc2.id = subc\_id  
 subc2.subc\_id = subc.id  
 subc2.user\_id = uid  
 subc2.save()  
 return HttpResponseRedirect('/channel/'+str(channel1.ChannelId)+'/')  
 else:  
 context = {'errors': errors, 'username': user}  
  
 return render(request, 'new\_bet.html', locals())  
  
  
def subcribers\_users(request, channel):  
 subcribers = []  
 channel1 = Channel.objects.get(ChannelId=int(channel))  
 users = User1.objects.all()  
 subc3 = []  
 subc = SubcChannel.objects.filter(channel\_id=channel1.ChannelId).all()  
 for b in subc:  
 subc1 = Subc.objects.filter(id=b.subc\_id).all()  
 if len(subc1) != 0:  
 subc3.append(b.id)  
 for i in subc3:  
 subcribers.append(Subc.objects.get(id=i))  
 return render(request, 'bets.html', context={"subcribers": subcribers, "users": users, 'channel': channel})

settings.py

*"""  
Django settings for untitled8 project.  
  
For more information on this file, see  
  
  
For the full list of settings and their values, see  
  
"""*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\_\_)))  
*# BASE\_DIR = os.path.dirname(os.path.dirname(\_\_file\_\_))  
  
  
# Quick-start development settings - unsuitable for production  
  
  
# SECURITY WARNING: keep the secret key used in production secret!*SECRET\_KEY = '\*-^fly(c-6\*fig9t%of)-&%khbu2s&z895#)$k&eo0nw%$))$r'  
  
*# SECURITY WARNING: don't run with debug turned on in production!*DEBUG = True  
  
ALLOWED\_HOSTS = []  
  
SITE\_ID = 1  
*# Application definition*INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
 'django.contrib.sites',  
 'app.apps.AppConfig',  
  
]  
  
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 = 'untitled8.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 = 'untitled8.wsgi.application'  
  
  
*# Database*DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.sqlite3',  
 'NAME': os.path.join(BASE\_DIR, 'db.sqlite3'),  
 'USER': '',  
 'PASSWORD': '',  
 'HOST': '',  
 'PORT': '',  
 }  
}  
  
  
*# Password validation*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*LANGUAGE\_CODE = 'en-us'  
  
TIME\_ZONE = 'UTC'  
  
USE\_I18N = True  
  
USE\_L10N = True  
  
USE\_TZ = True  
  
  
*# Static files (CSS, JavaScript, Images)*STATIC\_URL = '/static/'  
MEDIA\_URL = '/pics/'  
*# MEDIA\_ROOT = ':A/Users/olegy\_000/Desktop/РИП/ДЗ/untitled8/'*STATICFILES\_DIRS = [  
 os.path.join(BASE\_DIR, "static"),  
 '/var/www/static/',  
]

urls.py

from django.conf.urls import url  
from app.views import \*  
from django.conf.urls.static import static  
from django.conf import settings  
from django.contrib import admin  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^$', main),  
 url(r'^registration$', registration\_form, name='registration'),  
 url(r'^registration\_1$', registration\_1, name='registration\_1'),  
 url(r'^login1$', log\_in, name='login1'),  
 url(r'^login2$', log\_in1, name='login2'),  
 url(r'^logout$', logout\_view, name='logout'),  
 url(r'^logged\_in$', logged\_in\_view, name='logged\_in'),  
 url(r'^logged\_in\_1$', logged\_in, name='logged\_in\_1'),  
 url(r'^new\_channel$', new\_channel, name='new\_channel'),  
 url(r'^channels$', ChannelsView.as\_view(), name='channels'),  
 url(r'^channel/(?P<pk>\d+)/$', ChannelObject.as\_view(), name='ChannelObject'),  
 url(r'^new\_subscriber/(?P<channel>\d+)/$', new\_subscriber, name='new\_subscriber'),  
 url(r'^subscribers/(?P<channel>\d+)/$', subcribers\_users, name='subscribers'),  
]  
urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

admin.py

from django.contrib import admin  
from .models import \*  
from django.contrib.auth.models import User  
*# Register your models here.*@admin.register(User1)  
class UserAdmin(admin.ModelAdmin):  
 empty\_value\_display = 'null'  
 list\_display = ('last\_name', 'first\_name', 'email')  
 list\_filter = ('last\_name',)  
 search\_fields = ['last\_name', 'first\_name', 'email']  
  
  
class SubcChannel(admin.TabularInline):  
 model = SubcChannel  
 extra = 1  
  
  
@admin.register(Channel)  
class ChannelAdmin(admin.ModelAdmin):  
 empty\_value\_display = 'null'  
 list\_display = ('channel\_name','rating','type','videos')  
 list\_filter = ('channel\_name',)  
 search\_fields = ['channel\_name','type']  
  
 inlines = (SubcChannel,)  
  
 def channels(self, request):  
 channels = []  
 for s in SubcChannel.objects.filter(team=request.name):  
 channels.append(s)  
 return channels  
  
  
@admin.register(Subc)  
class SubcAdmin(admin.ModelAdmin):  
 empty\_value\_display = 'null'  
  
 def user\_last\_name(self, obj):  
 return "{}".format(obj.user)  
  
 inlines = (SubcChannel,)  
  
 list\_display = ('id', 'user\_last\_name', 'date')  
 list\_filter = ('id',)  
 search\_fields = ['user\_last\_name', 'date']

models.py

from django.db import models  
from django.contrib.auth.models import User  
  
*# Create your models here.*class User1(models.Model):  
 user1 = models.OneToOneField(User, on\_delete=models.CASCADE)  
 first\_name = models.CharField(max\_length=20, null=True)  
 last\_name = models.CharField(max\_length=20, null=True)  
 phone = models.PositiveIntegerField(unique=True, null=True)  
 email = models.EmailField(max\_length=50, unique=True)  
 birthday = models.DateField(blank=True, null=True)  
 passport = models.PositiveIntegerField(unique=True, null=True)  
  
 def \_\_str\_\_(self):  
 return str(self.last\_name)  
  
  
class Channel(models.Model):  
 ChannelId = models.AutoField(primary\_key=True)  
 channel\_name = models.CharField(max\_length=30, unique=True, null=True)  
 rating = models.PositiveSmallIntegerField(null=True, unique=True)  
 type = models.CharField(max\_length=30, null=True)  
 videos = models.PositiveSmallIntegerField(null=True)  
 picture = models.ImageField(upload\_to='pics/', null=True, blank=True)  
  
 def \_\_str\_\_(self):  
 return str(self.channel\_name)  
  
  
class Subc(models.Model):  
 user = models.ForeignKey(User, on\_delete=models.CASCADE)  
 channel = models.ManyToManyField(Channel, through="SubcChannel")  
 date = models.DateField()  
 amount = models.FloatField(null=True)  
 id = models.AutoField(primary\_key=True)  
 def \_\_str\_\_(self):  
 return str(self.id)  
  
  
class SubcChannel(models.Model):  
 channel = models.ForeignKey(Channel, on\_delete=models.CASCADE)  
 subc = models.ForeignKey(Subc, on\_delete=models.CASCADE)  
 id = models.AutoField( primary\_key=True)  
 def \_\_str\_\_(self):  
 return str(self.id)

base.html

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="UTF-8">  
 <title>{% block title %}{% endblock %}</title>  
 {% load static %}  
 <link rel="stylesheet" type="text/css" href={% static "/css/bootstrap-theme.min.css.css" %} />  
 <link rel="stylesheet" type="text/css" href={% static "/css/bootstrap.min.css" %} />  
 <link rel="stylesheet" type="text/css" href={% static "/css/style.css" %} />  
<body>  
 <nav class="navbar navbar-inverse navbar-static-top">  
 <div class="container-fluid">  
 <div class="navbar-header">  
 <a class="navbar-brand" href="/">Главная</a>  
 </div>  
 <div class="collapse navbar-collapse" id="myNavbar">  
 <ul class="nav navbar-nav">  
 <li><a href="/registration">Регистрация</a></li>  
 <li><a href="/login1">Вход</a></li>  
 <li><a href="/logout">Выход</a></li>  
 {% block userlog %}​{% endblock %}  
*{# <li><a href="/registration\_1">Регистрация1</a></li>#}  
{# <li><a href="/ulogin\_1">Вход1</a></li>#}* </ul>  
 </div>  
 <div>  
 {% block content %}​{% endblock %}  
 </div>  
 </div>  
 </nav>  
</body>  
</html>

UserLogin.html

{% extends 'base.html' %}  
{% block title %}{% endblock %}  
{% block userlog %}  
 {% if user.is\_authenticated %}  
 <p class="text-primary">User: <span class="glyphicon glyphicon-user "></span>{{ user.username }}</p>  
 {% endif %}  
{% endblock %}  
{% block content %}  
 <div class="jumbotron">  
 <div class="container">  
 <form method="POST" class="form-horizontal">  
 <h1 style="text-align:center">Вход</h1>  
 {% csrf\_token %}  
 {% for err in errors.values %}  
 <div class="alert alert-danger" >{{ err }}</div>  
 {% endfor %}  
 <div class="col-sm-offset-3 form-group">  
 <label class="col-sm-3 control-label">Логин:</label>  
 <div class="col-sm-8">  
 <input type="text" name="username">  
 </div>  
 </div>  
 <div class="col-sm-offset-3 form-group">  
 <label class="col-sm-3 control-label">Пароль:</label>  
 <div class="col-sm-8">  
 <input type="password" name="password">  
 </div>  
 </div>  
 <div class="form-group">  
 <div class="col-sm-offset-3 col-sm-9">  
 <button style="text-align:center" class="col-sm-1 btn btn-primary control-label" type="submit">Войти​</button>  
 <input style="text-align:center" class="col-sm-3 btn col-sm-offset-1 btn-primary control-label" type="button" value="Зарегистрироваться" onClick='*location*.href="/registration\_1"'>  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
{% endblock %}

Registration\_1.html

{% extends 'base.html' %}  
{% block title %}registration{% endblock %}  
{% block content %}  
 <div class="main\_block\_elem">  
 <div class="jumbotron">  
 <div class="container">  
 <form method="POST" class="form-horizontal">  
 <h1>Регистрация </h1>  
 {% csrf\_token %}  
 {% for field in form %}  
 {% for error in field.errors %}  
 <div class="alert alert-danger" role="alert">{{ field.errors }}</div>  
 {% endfor %}  
 {% endfor %}  
 {% for field in form %}  
 <div class="form-group">  
 <label for="{{ field.id\_for\_label }}" class="col-sm-3 control-label">{{ field.label }}:</label>  
 <div class="col-sm-9">  
 {{ field }}  
 </div>  
 </div>  
 {% endfor %}  
 <div class="form-group">  
 <div class="col-sm-offset-3 col-sm-9">  
 <button type="submit" class="btn btn-primary">Зарегистрировать</button>  
 </div>  
 </div>  
 </form>  
 </div>  
 </div>  
 </div>  
{% endblock %}

Obj.html

{% **extends 'base.html'** %}  
{% **csrf\_token** %}  
{% **block title** %}Channel{% **endblock** %}  
{% **block userlog** %}<**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>​{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div**>  
 <**div class="col-lg-4 center-pos"**>  
 <**img src="**{{**ChannelObject**.**picture**.**url** }}**"**>  
 </**div**>  
 <**div class="col-lg-3"**>  
 <**div class="blog-header title\_padding"**>  
 <**h1**>{{**ChannelObject**.**channel\_name** }}</**h1**>  
 </**div**>  
 <**h4**>Тип канала: {{**ChannelObject**.**type** }}</**h4**>  
 <**h4**>Кол-во видео: {{ **ChannelObject**.**videos** }}</**h4**>  
 <**h4**>Рейтинг: {{ **ChannelObject**.**rating** }}</**h4**>  
 <**br**>  
 </**div**>  
 <**div class="col-lg-3 "**>  
 {% **if user**.**is\_authenticated** %}  
 <**a role="button" class="btn btn-default btn-lg btn-block" href="/channels" style="margin-top**: 28**px**;**"**>Назад к списку</**a**>  
 <**a role="button" class="btn btn-default btn-lg btn-block"  
 onClick='*location***.href=**"/subscribers/**{{ **ChannelObject**.**ChannelId**}}**"'**>Просмотр подписчиков</**a**>  
 <**a role="button" class="btn btn-default btn-lg btn-block"  
 onClick='*location***.href=**"/new\_subscriber/**{{ **ChannelObject**.**ChannelId**}}**"'**>Подписаться</**a**>  
 {{ }}  
 {% **endif** %}  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

MainPage.html

{% **extends 'base.html'** %}  
{% **block title** %}{% **endblock** %}  
{% **block userlog** %}  
 {% **if user**.**is\_authenticated** %}  
 <**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>  
 {% **endif** %}  
{% **endblock** %}  
{% **block content** %}  
{% **if user**.**is\_authenticated** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="col-md-3"**>  
 <**a href="/channels"**>Просмотр списка каналов</**a**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **else** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="col-md-6"**>  
 <**p**>Для просмотра списка каналов выполните вход</**p**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endif** %}  
  
{% **endblock** %}

Registration.html

{% **extends 'base.html'** %}  
{% **block title** %}{% **endblock** %}  
{% **block userlog** %}  
 {% **if user**.**is\_authenticated** %}  
 <**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>  
 {% **endif** %}  
{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="page-header"**>  
 <**h1**>Регистрация</**h1**>  
 {% **for err in errors**.**values** %}  
 <**div class="alert alert-danger"** >{{ **err** }}</**div**>  
 {% **endfor** %}  
 </**div**>  
 <**form method="POST" class="form-horizontal"**>{% **csrf\_token** %}  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Фамилия:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="text" name="last\_name"**>  
 </**div**>  
 </**div**>  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Имя:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="text" name="first\_name"**>  
 </**div**>  
 </**div**>  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Email:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="text" name="Email"**>  
 </**div**>  
 </**div**>  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Логин:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="text" name="username"**>  
 </**div**>  
 </**div**>  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Пароль:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="password" name="password"**>  
 </**div**>  
 </**div**>  
 <**div class="col-sm-offset-3 form-group"**>  
 <**label class="col-sm-3 control-label"**>Повторите ввод:</**label**>  
 <**div class="col-sm-8"**>  
 <**input type="password" name="password2"**>  
 </**div**>  
 </**div**>  
  
 <**button style="text-align**:**center" class="col-sm-offset-3 col-sm-2 btn btn-primary control-label" type="submit"**>Зарегистрировать​</**button**>  
  
 </**form**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

New\_bet.html

{% **extends "base.html"** %}  
{% **block title** %}Add new{% **endblock** %}  
{% **block userlog** %}<**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>​{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="col-md-8"**>  
 <**h3**>Подписаться на канал</**h3**>  
 {% **for err in errors**.**values** %}  
 <**div class="alert alert-danger"** >{{ **err** }}</**div**>  
 {% **endfor** %}  
 <**form method="POST" enctype="multipart/form-data"**>  
 {% **csrf\_token** %}  
  
 <**button type="submit" class="btn btn-default btn-lg btn-block"** >Подписаться</**button**>  
 <**a role="button" class="btn btn-default btn-lg btn-block" href="/channels"**>Назад к списку</**a**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

UserLogin\_1.html

{% **extends 'base.html'** %}  
{% **block title** %}login{% **endblock** %}  
{% **block userlog** %}  
 {% **if user**.**is\_authenticated** %}  
 <**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>  
 {% **endif** %}  
{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**form method="POST" class="form-horizontal"**>  
 <**h1 style="text-align**:**center"**>Вход</**h1**>  
 {% **csrf\_token** %}  
 {% **for f in form** %}  
 {% **for error in f**.**errors** %}  
 <**div class="alert " role="alert"**>{{ **f**.**errors** }}</**div**>  
 {% **endfor** %}  
 {% **endfor** %}  
 {% **for f in form** %}  
 <**div class="form-group"**>  
 <**label for="**{{ **f**.**id\_for\_label** }}**" class="col-sm-3 control-label"**>{{ **f**.**label** }}:</**label**>  
 <**div class="col-sm-9"**>  
 {{ **f** }}  
 </**div**>  
 </**div**>  
 {% **endfor** %}  
 <**div class="form-group"**>  
 <**div class="col-sm-offset-3 col-sm-9"**>  
 <**button style="text-align**:**center" class="col-sm-1 btn-primary control-label" type="submit"**>Войти</**button**>  
 <**input style="text-align**:**center" class="col-sm-3 col-sm-offset-1 btn-primary control-label"  
 type="button" value="Зарегистрироваться" onClick='*location***.href=**"/registration\_1"'**>  
 </**div**>  
 </**div**>  
 </**form**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

Items.html

{% **extends 'base.html'** %}  
{% **csrf\_token** %}  
{% **block title** %}Channels{% **endblock** %}  
{% **block userlog** %}<**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>​{% **endblock** %}  
{% **block content** %}  
  
  
 <**div class="jumbotron"**>  
 <**div class="container-fluid"**>  
 <**div class="blog-header"**>  
 <**h1 class="col-lg-offset-4"**>Список каналов</**h1**>  
 </**div**>  
 <**div class="row"**>  
 <**div class="col-md-9"**>  
 {% **include 'item.html' with item**=**item** %}  
 </**div**>  
 <**div class="col-md-offset-9"**>  
*{# {% if user.is\_superuser %}#}* <**a role="button" class="btn btn-default btn-lg btn-block"  
 onClick='*location***.href=**"/new\_channel"'**>Добавить канал</**a**>  
*{# {% endif %}#}* </**div**>  
 </**div**>  
 </**div**>  
 <**div class="pagination"**>  
 <**span class="step-links"**>  
 {% **if channels**.**has\_previous** %}  
 <**a href="?page=**{{ **channels**.**previous\_page\_number** }}**"**>previous</**a**>  
 {% **endif** %}  
  
 <**span class="current"**>  
 Page {{ **channels**.**number** }} of {{ **channels**.**paginator**.**num\_pages** }}.  
 </**span**>  
 {% **if channels**.**has\_next** %}  
 <**a href="?page=**{{ **channels**.**next\_page\_number** }}**"**>next</**a**>  
 {% **endif** %}  
 </**span**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

Item.html

{% **for item in channels** %}  
 <**section class="panel "**>  
 <**div class="panel-body "**>  
 <**div class="row"**>  
 <**div class="col-lg-4 center-pos"**>  
 <**img src="**{{**item**.**picture**.**url**}}**"**>  
 </**div**>  
 <**div class="col-lg-6"**>  
 <**div class="panel-title"**>  
 <**h2**>  
 <**div class="title title\_padding"**>  
 <**a href="/channel/**{{**item**.**ChannelId** }}**/"**>{{**item**.**channel\_name**}}</**a**>  
 </**div**>  
 </**h2**>  
 </**div**>  
 <**h4**>Вид канала:{{**item**.**type**}}</**h4**>  
 <**h4**>Количество видео:{{**item**.**videos** }}</**h4**>  
 <**h4**>Рейтинг:{{**item**.**rating** }}</**h4**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**section**>  
{% **empty** %}  
 ПУСТОЙ СПИСОК  
{% **endfor** %}

Bets.html

{% **extends 'base.html'** %}  
{% **csrf\_token** %}  
{% **block title** %}Subcribers{% **endblock** %}  
{% **block userlog** %}<**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>​{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="blog-header"**>  
 <**h1 class="blog-title col-lg-offset-3"**>Подписки на канал</**h1**>  
 </**div**>  
 <**table id ='orders' class="table table-striped table-hover "**>  
 <**thead**>  
 <**tr**>  
 <**th**>ID Подписки</**th**>  
 <**th**>Дата</**th**>  
 <**th**>Пользователь</**th**>  
 </**tr**>  
 </**thead**>  
 <**tbody**>  
 {% **for subc in subcribers** %}  
 <**tr**>  
 <**td**>{{ **subc**.**id**}}</**td**>  
 <**td**>{{ **subc**.**date**}}</**td**>  
 <**td**>  
 {% **for u in users** %}  
 {% **if u**.**id** == **subc**.**user**.**id** %}  
 {{ **u**.**last\_name** }}  
 {{ **u**.**first\_name** }}  
 {% **endif** %}  
 {% **endfor** %}  
 </**td**>  
 </**tr**>  
 {% **endfor** %}  
 </**tbody**>  
 </**table**>  
 <**a role="button" class="btn btn-default btn-lg btn-block" href="/channel/**{{ **channel** }}**"**>Назад к каналу</**a**>  
 <**a role="button" class="btn btn-default btn-lg btn-block" href="/channels"**>Назад к списку</**a**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

New\_item.html

{% **extends "base.html"** %}  
{% **block title** %}Add new{% **endblock** %}  
{% **block userlog** %}<**p class="text-primary"**>User: <**span class="glyphicon glyphicon-user "**></**span**>{{ **user**.**username** }}</**p**>​{% **endblock** %}  
{% **block content** %}  
 <**div class="jumbotron"**>  
 <**div class="container"**>  
 <**div class="page-header"**>  
 <**h3**>Добавить канал</**h3**>  
 {% **for err in errors**.**values** %}  
 <**div class="alert alert-danger"** >{{ **err** }}</**div**>  
 {% **endfor** %}  
 </**div**>  
 <**div class="col-md-8"**>  
 <**form method="POST" enctype="multipart/form-data"** >  
 {% **csrf\_token** %}  
 <**div class="form-group"**>  
 <**label**>Название</**label**>  
 <**input type="text" name="channel\_name" class="form-control" placeholder="название"**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Рейтинг</**label**>  
 <**input type="text" name="rating" class="form-control" placeholder="рейтинг"**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Тип канала</**label**>  
 <**input type="text" name="type" class="form-control" placeholder="Тип канала"**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Кол-во видео</**label**>  
 <**input type="text" name="videos" class="form-control" placeholder="кол-во видео"**>  
 </**div**>  
 <**div class="form-group"**>  
 <**label**>Picture</**label**>  
 <**input type="file" name="picture"** >  
 </**div**>  
 <**button type="submit" class="btn btn-default btn-lg btn-block"**>Добавить</**button**>  
 <**a role="button" class="btn btn-default btn-lg btn-block" href="/channels"**>Назад к списку</**a**>  
 </**form**>  
 </**div**>  
 </**div**>  
 </**div**>  
{% **endblock** %}

















