Московский государственный университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Домашнее задание по курсу:

«Разработка Интернет Приложений»

Исполнитель:

Студент группы РТ5-51

Житенев В.Г.

Преподаватель:

Гапанюк Ю. Е.

«___»_____

Москва 2017 г.

Задание и порядок выполнения

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

№	Субъект	Отношение	Объект
3	Пользователь	Отзыв	Товар

Исходный код:

settings.py

```
"""
Django settings for untitled project.

Generated by 'django-admin startproject' using Django 2.0.1.

For more information on this file, see
https://docs.djangoproject.com/en/2.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.0/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/2.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '5%j)pj##c5(*sbk8ig(rf3ougd7m2*@2vw&z+zau8qz)66v@h9'

# 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',
    'my_app.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 = 'untitled.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 = 'untitled.wsgi.application'


# Database
# https://docs.djangoproject.com/en/2.0/ref/settings/#databases

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


# Password validation
# https://docs.djangoproject.com/en/2.0/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/2.0/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/2.0/howto/static-files/
```

```
STATIC_URL = '/static/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'my_app/media')
MEDIA_URL = '/media/'
```

urls.py

```
"""My_HW URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/2.0/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.conf.urls import url
from django.contrib import admin
from my_app.views import *
from django.conf.urls.static import static
from django.conf import settings
from django.views.generic.base import RedirectView
urlpatterns = [
    url(r'^admin', admin.site.urls),
    url(r'^$', RedirectView.as_view(url='/main', permanent=False), name='index'),
    url(r'^main', main, name='main_url'),
    url(r'^login', login, name='login_url'),
    url(r'^logout', logout, name='logout_url'),
    url(r'^registration2', registration2, name='registration2_url'),
    url(r'^registration', registration, name='registration_url'),
    url(r'^good_show', good_show, name='good_show_url'),
    url(r'^feedback_add', fb_add, name='feedback_add_url'),
    url(r'^good', GoodList.as_view(), name='good_url'),
    url(r'^good_add', good_add, name='good_add_url'),
    url(r'^shop', ShopList.as_view(), name='shop_url'),
]
if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_URL)
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

admin.py

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


class GoodAdmin(admin.ModelAdmin):
    fields = ('name_good', 'type_good', 'photo_good', 'description_good')
    list_display = ["name_good", "type_good",  "photo_good", "description_good"]  #
Выводит 3 поля
    search_fields = ('id_good','name_good', 'type_good')
    list_filter = ["id_good"]
    list_per_page = 10
    class Meta:
        model = Good


class ShopAdmin(admin.ModelAdmin):
    fields =('name_shop', 'adr_shop')
    list_display = ("name_shop", "adr_shop", "get_good" )  # Выводит 3 поля
    search_fields = ('name_shop', 'adr_shop')
    list_filter = ["id_shop"]
    list_per_page = 10
    class Meta:
```

```
        model = Shop


class FBAdmin(admin.ModelAdmin):
    list_per_page = 10


admin.site.register(Shop, ShopAdmin)
admin.site.register(Good, GoodAdmin)
admin.site.register(Feedback, FBAdmin)
```

apps.py

```
from django.apps import AppConfig


class MyAppConfig(AppConfig):
    name = 'my_app'
    verbose_name='Магазин'
```

forms.py

```
from django import forms
from django.core.exceptions import ValidationError
from my_app.models import *


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 пароль')
    email = forms.EmailField(widget=forms.EmailInput, label='E-mail')
    firstname = forms.CharField(label='Имя')
    surname = forms.CharField(label='Фамилия')

    def clean(self):
        cleaned_data = super(RegistrationForm, self).clean()
        password = cleaned_data.get('password')
        password2 = cleaned_data.get('password2')
        if password and password != password2:
            raise ValidationError("Пароли должны совпадать")
        return cleaned_data

    def clean_username(self):
        username = self.cleaned_data.get('username')
        username_qs = User.objects.filter(username=username)
        if username_qs.exists():
            raise ValidationError("Пользователь с таким именем уже существует")
        return username


class ShopForm(forms.ModelForm):
    class Meta:
        model = Shop
        fields = ["name_shop", "adr_shop"]
        widgets = {'name_shop': forms.Textarea(attrs={'resize': 'none'})}


class GoodForm(forms.ModelForm):
    class Meta:
        model = Good
        fields = ['name_good', 'type_good', 'photo_good', 'description_good']
        widgets = {'name_good': forms.Textarea(attrs={'resize': 'none'})}


class FbForm(forms.ModelForm):
```

```
    class Meta:
        model = Feedback
        exclude = ['']

class Good_add_Form(forms.ModelForm):
    class Meta:
        model = Good
        exclude = ['']
```

models.py

```
from django.db import models
from django.contrib.auth.models import User
# Create your models here.

class Good(models.Model):
    class Meta():
        db_table = "Good"
        verbose_name_plural = "Товары"
        verbose_name = "Товар"
    id_good = models.AutoField(primary_key=True)
    name_good = models.CharField(max_length=50, verbose_name='Название товара')
    type_good = models.CharField(max_length=50, verbose_name='Тип товара')
    photo_good = models.ImageField(null=True, blank=True, verbose_name='Фото Товара')
    description_good = models.CharField(max_length=2055, blank=True,
verbose_name='Описание товара')
    user = models.ManyToManyField(User,  verbose_name='ID отзыва', through="Feedback",
through_fields=('good','fb_users'))

    def __str__(self):
        return '%s %s' % (self.name_good, self.type_good)

    def get_user(self):
        return [{'id': user.id, 'name': user.username} for user in self.user.all()]


class Feedback(models.Model):
    class Meta():
        db_table= "Feedback"
        verbose_name_plural = "Отзывы"
        verbose_name = "Отзыв"
    fb_users = models.ForeignKey(User, on_delete=models.CASCADE,
verbose_name='Пользователь')
    good=models.ForeignKey(Good, on_delete=models.CASCADE)
    information = models.CharField(max_length=2055, blank=True, verbose_name='Отзыв о
товаре')
    photo = models.ImageField(null=True, blank=True, verbose_name='Фото')


    def __str__(self):
        return '%s %s' % (self.fb_users, self.information)

    def get_good_show(self):
        return [{'id_good': good.id_good, 'name_good': good.name_good, 'type_good':
good.type_good, 'photo_good': good.photo_good, 'description_good':
good.description_good} for good in self.good.objects.all()]



class Shop(models.Model):
    class Meta():
        db_table = "Shop"
        verbose_name_plural = "Магазины"
        verbose_name = "Магазин"
    id_shop =  models.AutoField(primary_key=True)
    name_shop = models.CharField(max_length=20, verbose_name='Название магазина')
    adr_shop = models.CharField(max_length=100, verbose_name='Адрес магазина')
    assort_shop = models.ManyToManyField(Good, verbose_name='Ассортимент магазина')
```

```
def get_good(self):
    return ", ".join([f.name_good for f in self.assort_shop.all()])

def __str__(self):
    return self.name_shop

def get_assort(self):
    return ", ".join([s.type_good for s in self.assort_shop.all()])
get_assort.short_description = 'Ассортимент магазина'
```

views.py

```
from django.core import serializers
from django.http import HttpResponse
from django.shortcuts import render, HttpResponseRedirect
from django.urls import reverse
from django.views.generic import ListView
from my_app.forms import *
from my_app.models import *
from django.contrib.auth.models import User
from django.contrib import auth
from django.contrib.auth import authenticate
from django.contrib.auth.decorators import login_required
from django.shortcuts import get_object_or_404


def main(request):
    return render(request, 'main.html', locals())


class GoodList(ListView):
    form_class = GoodForm
    model = Good
    template_name = "good.html"
    paginate_by = 3

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['feedbacks'] = Feedback.objects.all()
        return context


def good_show(request):
    id = request.GET.get('id')
    if id:
        good = Good.objects.filter(id_good=id)
        good = good.first()
        fb = Feedback.objects.filter(good=id)
        shop = Shop.objects.filter(assort_shop=id)
        if request.method=="POST":
            u = User.objects.get(id=request.user.id)
            good = Good.objects.get(id_good=id)
            feedback = request.POST.get('feedback')
            print(feedback)
            fb=Feedback(fb_users=u, good=good, information=feedback)
            fb.save()
            fb = Feedback.objects.filter(good=id)
        return render(request, "good_show.html", locals())
    else:
        return render(request, "good.html", {'good': Good.objects.all()})


class ShopList(ListView):
    form_class = ShopForm
    model = Shop
    template_name = "shop.html"
    paginate_by = 5


def fb_add(request):
    good = Good.objects.all()
    fb = Feedback.objects.all()
    form = FbForm(request.POST or None, request.FILES or None)
    context = {"form": form, "good": good, "tutors": fb}
    if form.is_valid() and request.method == 'POST':
        instance = form.save(commit=False)
        instance.save()
```

```
        return HttpResponseRedirect(reverse('good_url'))
    return render(request, "feedback_add_form.html", locals())
##################################################################
def good_add(request):
    good  = Good.objects.all()
    form = Good_add_Form(request.POST or None, request.FILES or None)
    if form.is_valid() and request.method == 'POST':
        instance = form.save(commit=False)
        instance.save()
        return HttpResponseRedirect(reverse('good_url'))
    return render(request, "good_add_form.html", locals())


def registration(request):
    errors = {}
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['username'] = 'Введите логин'
        elif len(username) < 5:
            errors['username'] = 'Логин должен превышать 5 символов'
        elif User.objects.filter(username=username).exists():
            errors['username'] = 'Такой логин уже существует'
        password = request.POST.get('password')
        if not password:
            errors['password'] = 'Введите пароль'
        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'
        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 errors:
            # ...
            user = User.objects.create_user(username=username, password=password,
email=email, first_name=firstname,
                                            last_name=surname)
            return HttpResponseRedirect(reverse('login_url'))
    return render(request, 'registration.html', locals())


def login(request):
    error = ""
    username = None
    password = None
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')
        user = authenticate(username=username, password=password)
        if user:
            auth.login(request, user)
            return HttpResponseRedirect(reverse('good_url'))
        else:
            error = "Пользователь не найден"
    return render(request, 'login.html', locals())


@login_required()
def success(request):
    return render(request, reverse('good_url'), locals())
```

```
def logout(request):
    auth.logout(request)
    return HttpResponseRedirect(reverse('main_url'))


def registration2(request):
    form = RegistrationForm(request.POST or None)
    if request.method == 'POST':
        if form.is_valid():
            user = User.objects.create_user(username=request.POST.get('username'),
                                            password=request.POST.get('password'),
                                            email=request.POST.get('email'),
                                            first_name=request.POST.get('firstname'),
                                            last_name=request.POST.get('surname')
                                            )
            return HttpResponseRedirect(reverse('login_url'))
        else:
            print(form.cleaned_data)
    return render(request, 'registration2.html', {'form': form}, locals())
```

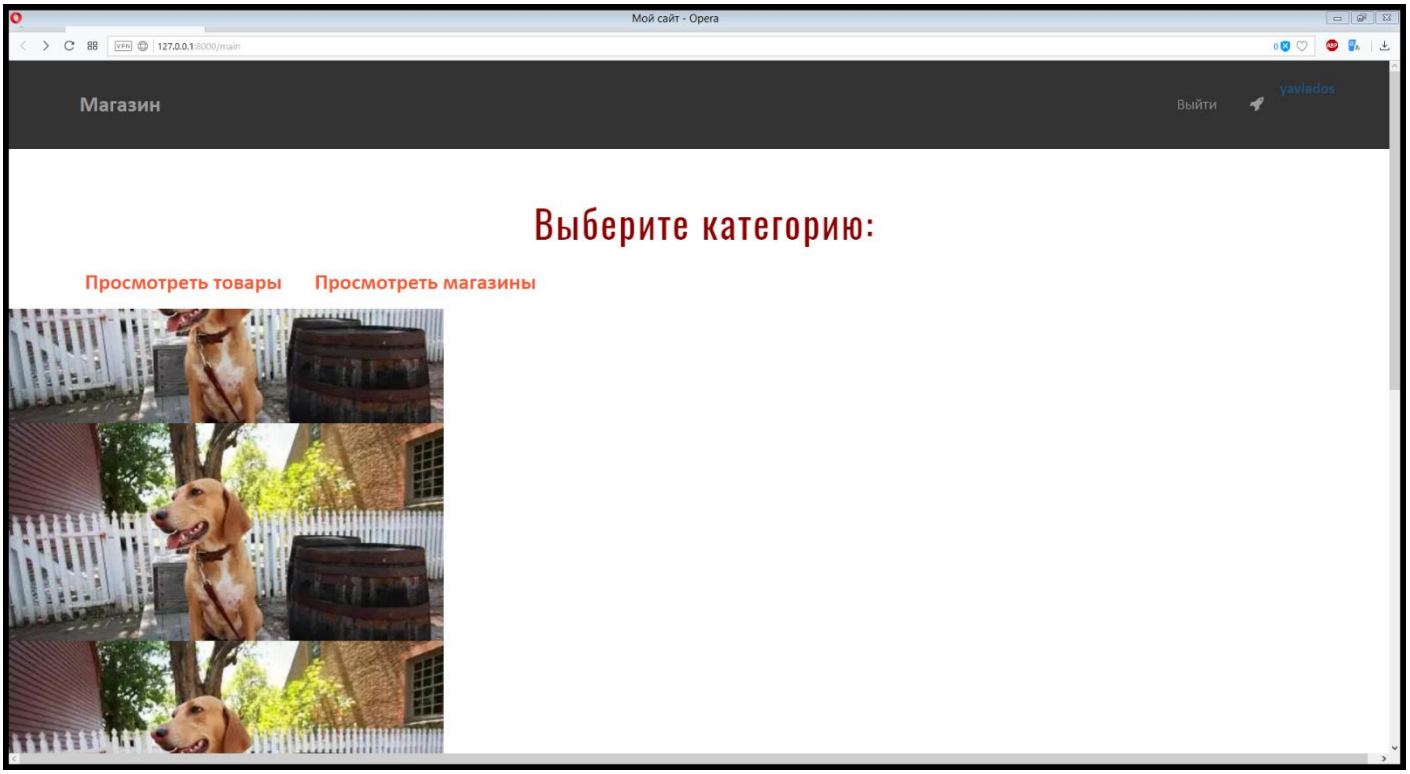
base. html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Мой сайт</title>
    {% load static %}
    <link rel="stylesheet" href="static/Bootstrap/css/bootstrap.css">
    <link rel="stylesheet" href="static/Bootstrap/css/main.css">
    <link rel="stylesheet" href="static/Bootstrap/css/font-awesome.min.css">
    <script type="text/javascript" src="{% static
'Bootstrap/js/alight_0.12.last.min.js' %}"></script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</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">
                </button>
                <a class="navbar-brand" href="{%url 'main_url'  %}">Магазин</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav navbar-right">
                            {% if request.user.is_authenticated %}
                                    <span>{{ request.user.username }}</span>
                                    <li><a href="{% url 'logout_url' %}">Выйти</a></li>
                                {% else %}
                                    <li><a href="{% url 'login_url' %}">Вход</a></li>
                                    <li><a href="{% url 'registration_url'
%}">Регистрация</a></li>
                                {% endif %}
                <li> <li><a href="{%url 'admin:index'  %}"><i class="fa fa-rocket fa-
invert" aria-hidden="true"></i></a></li></li>
                </ul>
            </div>
        </div>
    </div>
    {% block body2 %}<!--Блок для всего остального-->
```

```
    {% endblock %}
    </body>
</html>
```
</html>

main. html

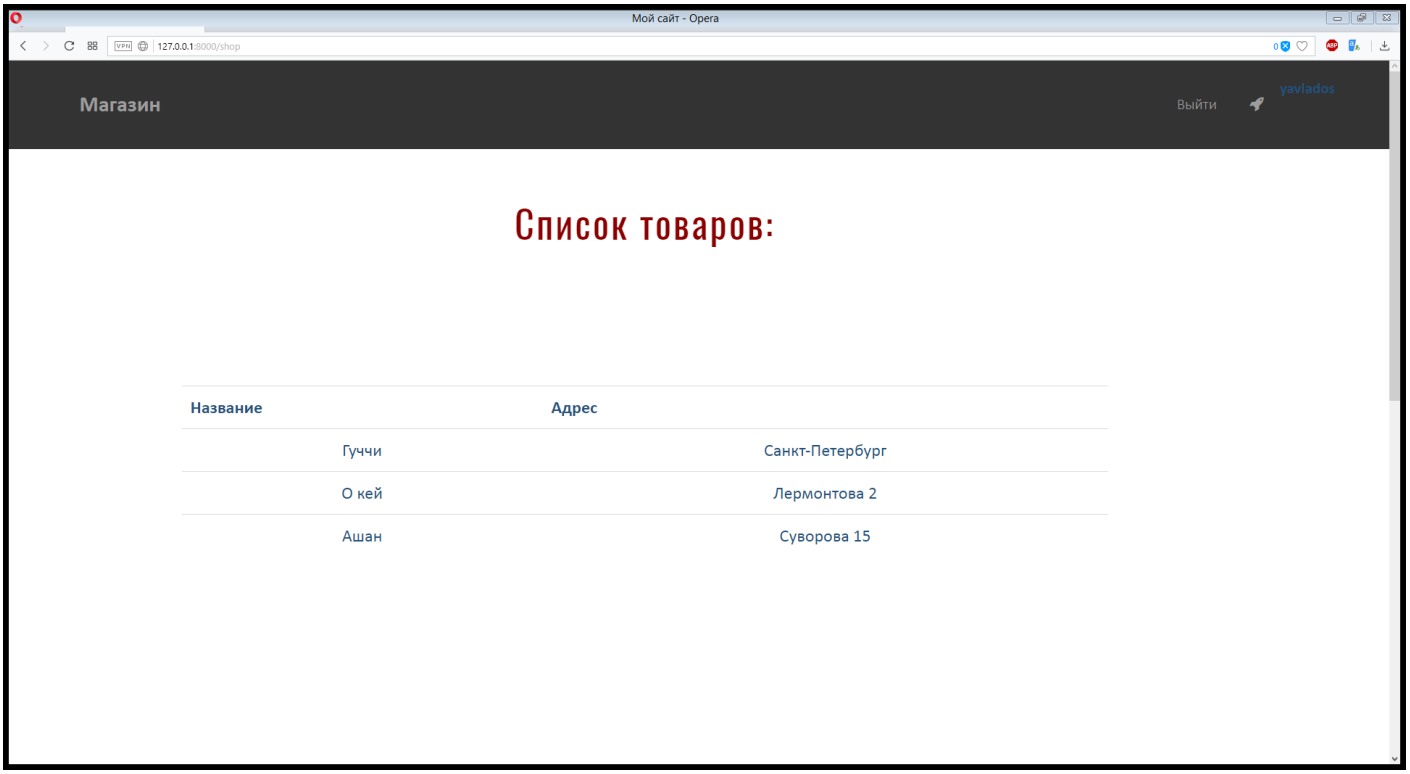
```
<!DOCTYPE html>
{% extends 'base.html' %}
{% block body2 %}
    <div id = "headerwrap">
        <div class="container">
        <div class="row centered">
            <div class="col-lg-8 col-lg-offset-2">
                <h2>Выберите категорию:</h2>
            </div>
            <br>
            <a class="navbar-brand" href="{%url 'good_url'  %}">Просмотреть
товары</a><br>
            <a class="navbar-brand" href="{%url 'shop_url'  %}">Просмотреть
магазины</a>
        </div>
        </div>
    <div al-app al-init="count=6"
        @scroll.debounce-50="($element.scrollTop > $element.scrollHeight -
$element.clientHeight - 150)?count++:0" class="body">
    <div al-repeat="n in count">
            <img :src.tpl="http://lorempixel.com/400/200/animals/{{n}}" />
     </div>
</div>
    </div>
{% endblock %}
```



shop. html

```
<!DOCTYPE html>
{% extends 'base.html' %}
{% block body2 %}
    <div id="headerwrap">
        <h2>Список товаров:</h2>
    <br><br><br><br>
```

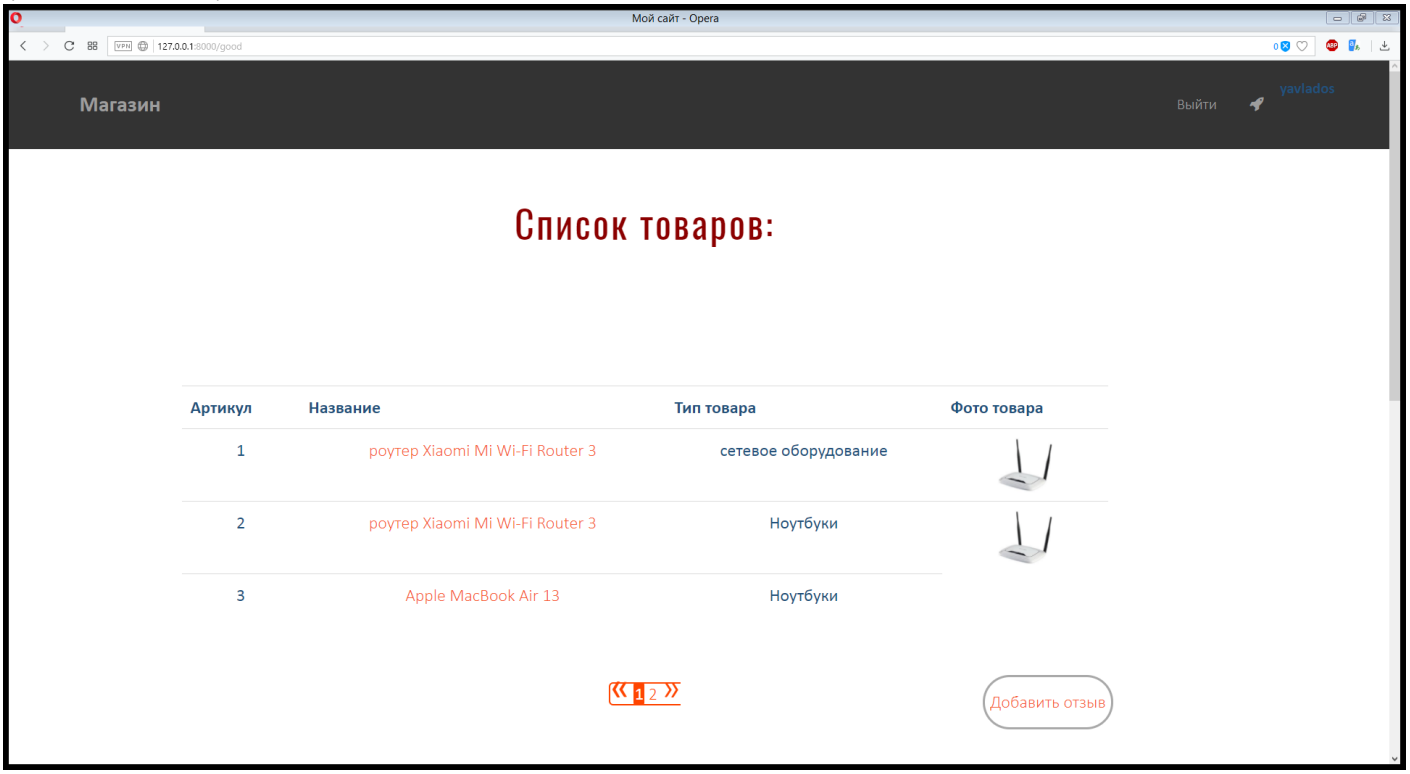
```
      <div class="container">
{% if object_list %}
    <div class ="my-flex-container1">
    <div class ="my-flex-block2">
    <table class="table table-dark">
    <tr>
      <th scope="col">Название</th>
      <th scope="col">Адрес</th>
    </tr>
 {% for shop in object_list %}
     <tr>
         <td>{{ shop.name_shop }}</td>
         <td>{{ shop.adr_shop   }}</td>
     </tr>
 {% endfor %}

    </table>

    {% if is_paginated %}
                <div class="pagination">
                    <span class="page-links">
                        {% if page_obj.has_previous %}
                            <a class="aquo" href="/shop?page={{
page_obj.previous_page_number }}">&laquo;</a>
                        {% else %}
                            <span class="unavailable" href="">&laquo;</span>
                        {% endif %}
                        {% for page in page_obj.paginator.page_range %}
                            {% if page %}
                                {% ifequal page page_obj.number %}
                                    <span class="current-page" href="">{{ page
}}</span>
                                {% else %}
                                    <a href="/shop?page={{ page }}" class="page">{{
page }}</a>
                                {% endifequal %}
                            {% endif %}
                        {% endfor %}
                        {% if page_obj.has_next %}
                            <a class="aquo" href="/shop?page={{
page_obj.next_page_number }}">&raquo;</a>
                        {% else %}
                            <span class="unavailable" href="">&raquo;</span>
                        {% endif %}
                        <span hidden id="per-page">{{ page_obj.paginator.per_page
}}</span>
                    </span>
                </div>
            {% endif %}
{% else %}

            <p>Пока не добавлено ни одного магазина</p>
{% endif %}
    </div>
 </div>

{% endblock %}
```



good.html

```
<!DOCTYPE html>
{% extends 'base.html' %}
{% block body2 %}
    <div id="headerwrap">
        <h2>Список товаров:</h2>
    <br><br><br><br>
      <div class="container">
{% if object_list %}
    <div class ="my-flex-container1">
    <div class ="my-flex-block2">
    <table class="table table-dark">
    <tr>
      <th scope="col">Артикул</th>
      <th scope="col">Название</th>
      <th scope="col">Тип товара</th>
      <th scope="col">Фото товара</th>
    </tr>
 {% for good in object_list %}
     <tr>
         <td>{{ good.id_good }}</td>
         <td><a href={% url "good_show_url"%}?id={{ good.id_good }} > {{
good.name_good}}</a></td>
         <td>{{ good.type_good}}</td>
          {% if good.photo_good %}
               <td> <img class = "img-tovar-table"  src="{{
good.photo_good.url}}"></td>
          {% endif %}
     </tr>
 {% endfor %}

    </table>
        {% if user.is_authenticated  %}
            <a href="{% url 'feedback_add_url' %}">
                <div class="add-button">Добавить отзыв</div>
            </a>
        {% endif %}
    {% if is_paginated %}
                <div class="pagination">
```

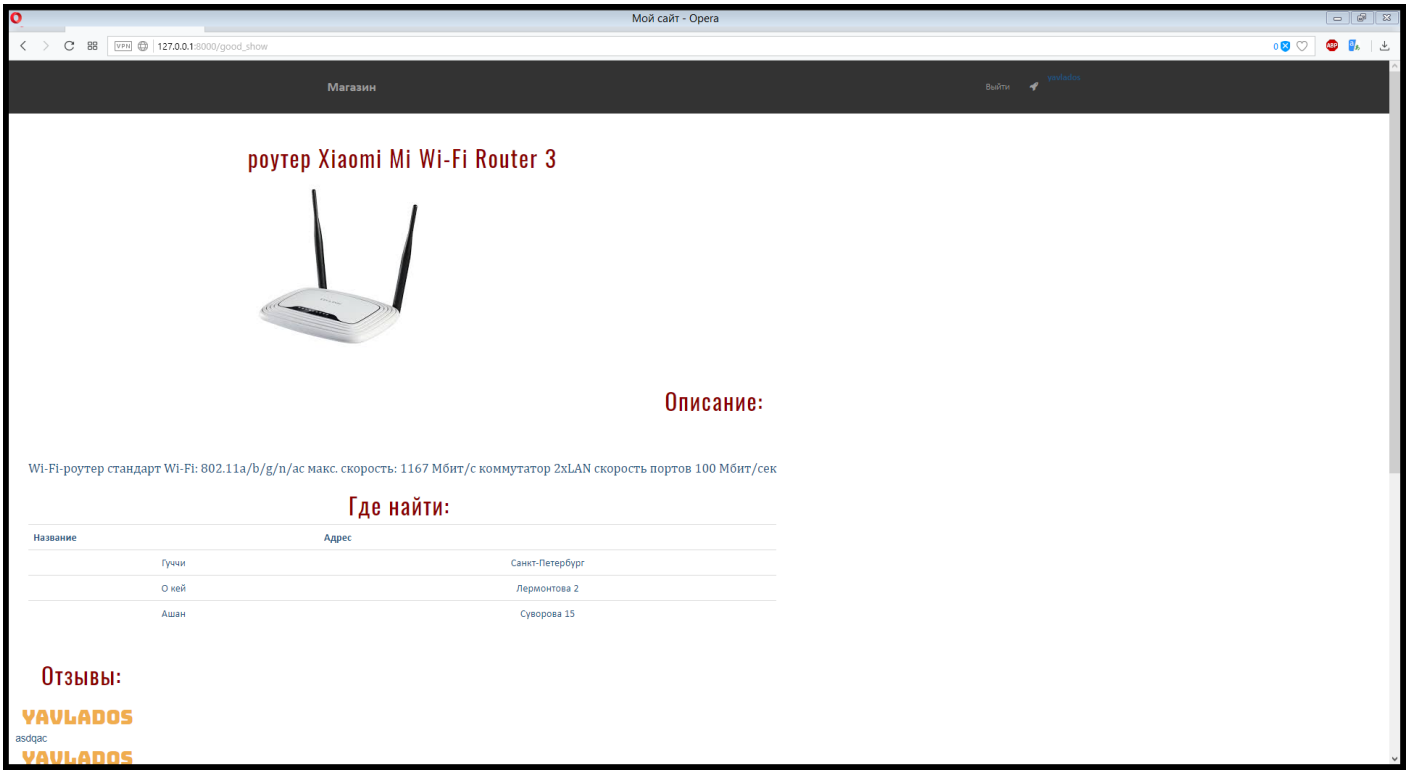
```
                    <span class="page-links">
                        {% if page_obj.has_previous %}
                            <a class="aquo" href="/good?page={{
page_obj.previous_page_number }}">&laquo;</a>
                        {% else %}
                            <span class="unavailable" href="">&laquo;</span>
                        {% endif %}
                        {% for page in page_obj.paginator.page_range %}
                            {% if page %}
                                {% ifequal page page_obj.number %}
                                    <span class="current-page" href="">{{ page
}}</span>
                                {% else %}
                                    <a href="/good?page={{ page }}" class="page">{{
page }}</a>
                                {% endifequal %}
                            {% endif %}
                        {% endfor %}
                        {% if page_obj.has_next %}
                            <a class="aquo" href="/good?page={{
page_obj.next_page_number }}">&raquo;</a>
                        {% else %}
                            <span class="unavailable" href="">&raquo;</span>
                        {% endif %}
                        <span hidden id="per-page">{{ page_obj.paginator.per_page
}}</span>
                    </span>
                </div>
            {% endif %}

{% else %}
            <p>Пока не добавлено ни одного товара</p>
{% endif %}
    </div>
 </div>

{% endblock %}
```



Good_show.html

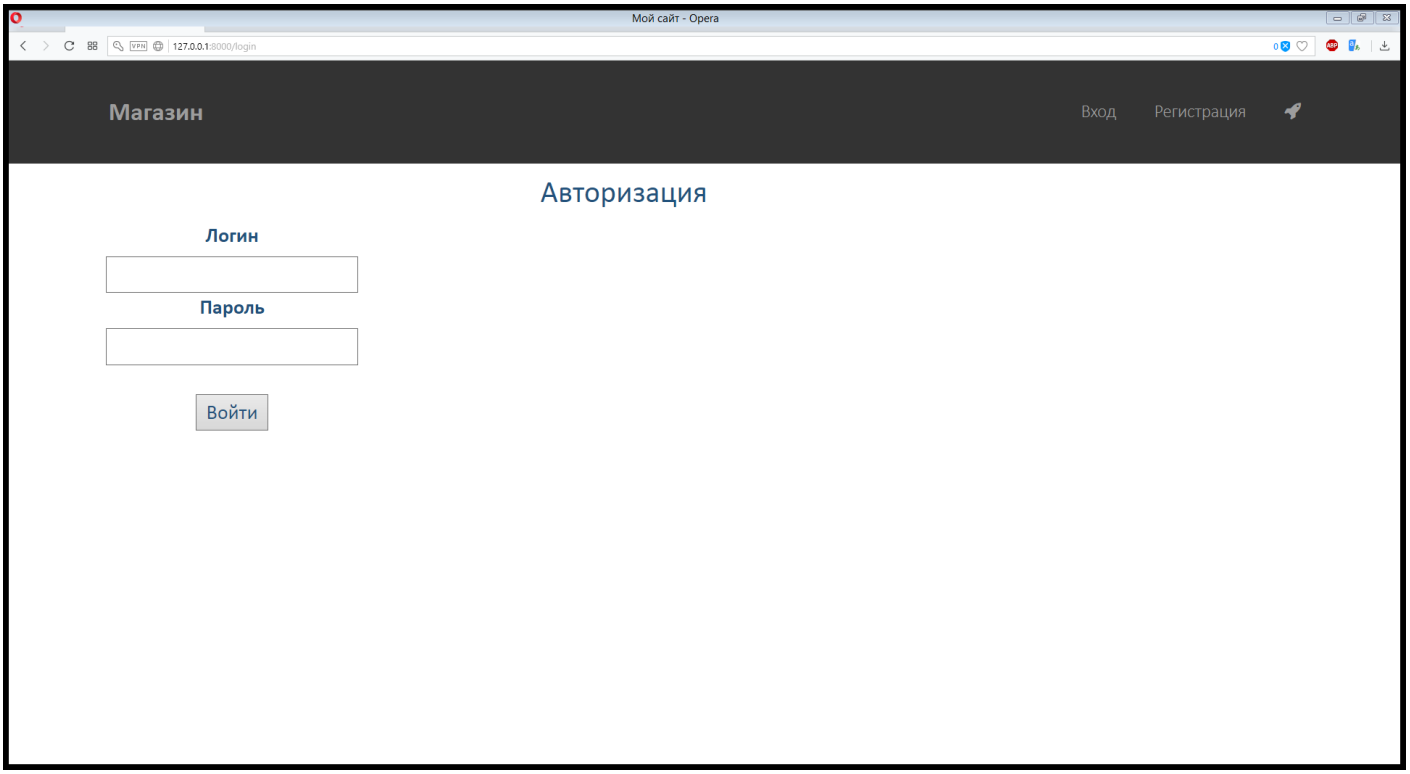
```
<!DOCTYPE html>
{% extends 'base.html' %}
{% block body2 %}
```

```
    <div id = "headerwrap">
        <h2>{{ good.name_good }}</h2>
    <div class ="my-flex-container2">
        <div class = "tovar-photo">
         {% if good.photo_good %}
                <img class = "img-tovar"  src="{{ good.photo_good.url}}">
         {% endif %}
        </div>
    </div>
    <div class="my-flex-container">
        <div class="my-flex-block1"><h3>Описание:  </h3></div>
        <div class="my-flex-block">
            <p>{{ good.description_good }}</p>
            <h3>Где найти:</h3>
                <table class="table table-dark">
            <tr>
              <th scope="col">Название</th>
              <th scope="col">Адрес</th>
            </tr>
            {% for sh in shop %}
                <tr>
                <td>{{ sh.name_shop }}</td>
                <td>{{ sh.adr_shop }}</td>
                </tr>
            {% endfor %}
                </table>
        </div>
    </div>
    <div class="comment-label">Отзывы:</div>
            {% for feedback in fb %}
                <div class ="username">{{ feedback.fb_users.username  }}</div>
                <div class ="comment">{{ feedback.information }}</div>
                {%  if feedback.photo %}
                    <img src="{{ feedback.photo.url }}">
                {% endif %}
            {% endfor %}
            {% if user.is_authenticated  %}
                <div class = "my-flex-container1">
                <div class="my-flex-block2">
    <h3>{{ request.user.username }}, не желаете оставить отзыв?</h3>
        <form method="POST">
            {% csrf_token %}
        <textarea name="feedback" cols="40" rows="10" resize="none" maxlength="2055"
                  id="id_information"></textarea>
        <button type="submit" id="btn_add">Добавить отзыв</button>
        </form>
                    </div>
                    </div>
            {% endif %}
    </div>
{% endblock %}
```



login.html

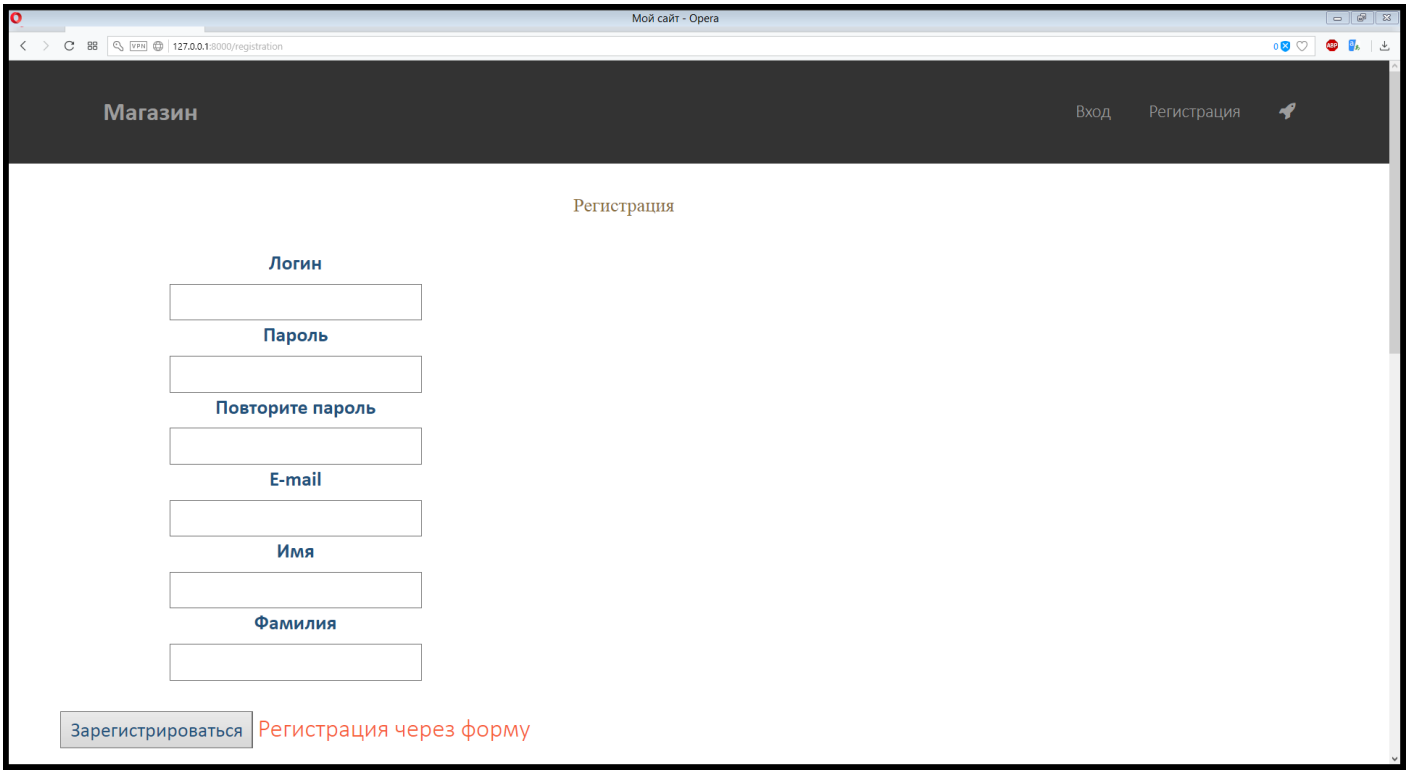
```
<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}
{% block body2 %}
    <div id="body1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h3>Авторизация<br></h3>
                </div>
            <form method="POST">
            <div class="auth_block"style="width: 352px">
                {% csrf_token %}
                <div class="auth_blank">
                    <label class="login" style="width: 60px">
                        Логин
                    </label>
                    <div class="form-block">
                        <input type="text" name="username">
                    </div>
                </div>
                <div class="auth_blank">
                    <label class="login" style="width: 70px">
                        Пароль
                    </label>
                    <div class="form-block">
                        <input type="password" name="password">
                    </div>
                </div>
                {% if error != "" %}
                    <p>{{ error }}</p>
                {% endif %}
                <br>
                <button type="submit">Войти</button>
            </div>
            </form>
            </div>
        </div>
    </div>
{% endblock %}
```



registration.html

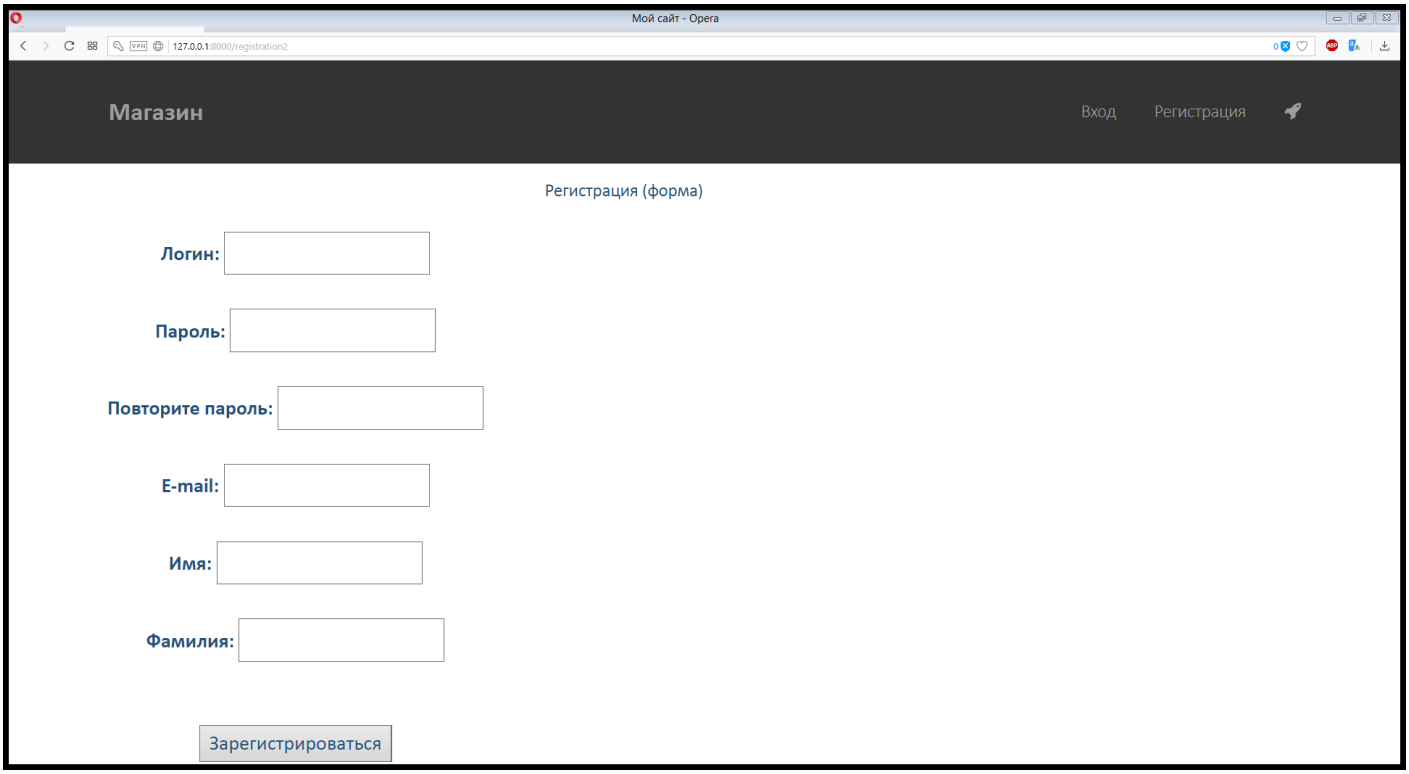
```
<!DOCTYPE html>
{% extends 'base.html' %}
{% load static %}
{% block body2 %}
 <div id="headerwrap">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h5><br>Регистрация<br><br></h5>
                </div>
            <form method="POST">
            <div class="auth_block" style="width: 452px">
                {% csrf_token %}
                <div class="auth_blank">
                    <label>
                        Логин
                    </label>
                    <div class="form-block">
                        <input type="text" name="username" value="{{ username }}">
                        {% if errors.username != '' %}
                            <div class="form-error">{{ errors.username }}</div>
                        {% endif %}
                    </div>
                </div>
                <div class="auth_blank">
                    <label>
                        Пароль
                    </label>
                    <div class="form-block">
                        <input type="password" name="password">
                        {% if errors.password != '' %}
                            <div class="form-error">{{ errors.password }}</div>
                        {% endif %}
                    </div>
                </div>
                <div class="auth_blank">
                    <label>
                        Повторите пароль
                    </label>
                    <div class="form-block">
                        <input type="password" name="password2">
```

```
                    {% if errors.password2 != '' %}
                        <div class="form-error">{{ errors.password2 }}</div>
                    {% endif %}
                </div>
            </div>
            <div class="auth_blank">
                <label>
                    E-mail
                </label>
                <div class="form-block">
                    <input type="email" name="email" value="{{ email }}">
                    {% if errors.email != '' %}
                        <div class="form-error">{{ errors.email }}</div>
                    {% endif %}
                </div>
            </div>
            <div class="auth_blank">
                <label>
                    Имя
                </label>
                <div class="form-block">
                    <input type="text" name="firstname" value="{{ firstname }}">
                    {% if errors.firstname != '' %}
                        <div class="form-error">{{ errors.firstname }}</div>
                    {% endif %}
                </div>
            </div>
            <div class="auth_blank">
                <label>
                    Фамилия
                </label>
                <div class="form-block">
                    <input type="text" name="surname" value="{{ surname }}">
                    {% if errors.surname != '' %}
                        <div class="form-error">{{ errors.surname }}</div>
                    {% endif %}
                </div>
            </div>
            <br>
            <button type="submit">Зарегистрироваться</button>
            <a style=" font-size: 20px; font-weight: 200; "  href="{% url
'registration2_url' %}">Регистрация через форму</a>
        </div>
    </form>
        </div>
    </div>
</div>
{% endblock %}
```



Registration2.html

```
<!DOCTYPE html>
{% extends 'base.html' %}
{% block body2 %}
 <div id="body1">
        <div class="container">
            <div class="row centered">
                <div class="col-lg-8 col-lg-offset-2">
                    <h5><br>Регистрация (форма)<br><br></h5>
                </div>
              <form method= "POST">
          <div class="auth_block" style="width: 452px">
                {% csrf_token %}
                {{ form.as_p }}
              <br>
              <button type= "submit"> Зарегистрироваться </button>
          </div>
    </form>
            </div>
        </div>
    </div>
{% endblock %}
```



Feedback_add_form.html

```
{% extends 'base.html' %}
{% block body2 %}
     <div id = "headerwrap">
        <div class="my-flex-container1">
            <div class="my-flex-block">
                 <h3>Уважаемый, {{ request.user.username }}, не желаете оставить
отзыв?</h3>
              <form method="POST" enctype="multipart/form-data">
              {% csrf_token %}
                  {{ form.as_p }}
                    <button type="submit" id="btn_add">Добавить отзыв</button>
              </form>
                {% if user.is_superuser %}
            <a href="#course-add-modal" data-toggle="modal">
                <div class="add-button-tovar" style="color: orangered; bottom: 10%;
border-color: orangered">&plus;</div>
            </a>
    {% endif %}
             <div id="course-add-modal" class="modal fade">
            <div class="modal-dialog">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-
hidden="true">×</button>
                        <h2 class="modal-title">Добавление нового отзыва</h2>
                    </div>
                    <div class="modal-body">
                        <form method="POST" enctype="multipart/form-data">
                            <div class="add-form container">
                                {% csrf_token %}
                                    {{ form.as_p }}
                            </div>
                            <div class="modal-footer">
                                <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыть</button>
                            </div>
                             <button type="submit" id="btn_add">Добавить отзыв</button>
                            </form>
                </div>
            </div>
```

```
        </div>
            </div>
        </div>
    </div>
    </div>
{% endblock %}
```