Московский государственный технический университет им. Н.Э. Баумана

Домашнее задание по дисциплине “Разработка интернет приложений”

ИСПОЛНИТЕЛЬ:

студент группы РТ5-51
Чечелев Д.С.

"__"_____2018 г.

Москва - 2017 – 2018

Задание

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

20	Пользователь	Оказание	Услуга

Settings py

```
import os

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


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.10/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'sr@rkp34eba+a0q%o&upfe4mgpkgx(c(-tacvydt7d^4y5e=!3'

# 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',
    'Lab5_App',
    'loginsys',
]

MIDDLEWARE = [#MIDDLEWARE_CLASSES = [
    '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 = 'Lab_5.urls'

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

WSGI_APPLICATION = 'Lab_5.wsgi.application'


# Database
# https://docs.djangoproject.com/en/1.10/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/1.10/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/1.10/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/1.10/howto/static-files/

STATIC_URL = '/static/'

MEDIA_ROOT = os.path.join(BASE_DIR, 'media').replace('\\', '/')
MEDIA_URL = '/media/'
STATIC_ROOT = 'C:/Users/user/Desktop/DZ/static'
```

Urls.py

```python
from django.conf.urls import url, include
from django.contrib import admin
from django.conf.urls.static import static
from django.conf import settings
from Lab5_App.views import *

urlpatterns = [
    url(r'^function_view/', function_view),
    url(r'^class_b_view/', ClassBased.as_view()),
    url(r'^allorders/(?P<id>\d+)', order_info, name='orderinfo_url'),
    url(r'^allorders/', AllOrders.as_view()),
    url(r'^admin/', admin.site.urls),
    url(r'^order/', order_add, name='order_url'),
    url(r'^service/(?P<id>\d+)', service_info, name='service_url'),
    url(r'^auth/', include('loginsys.urls')),
    url(r'^$', OrdersList.as_view(), name='main'),
]

if settings.DEBUG:
    urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

```
16  from django.conf.urls import url,include
17
18  from loginsys.views import *
19
20  urlpatterns = [
21                          url(r'^login/', login),
22                          url(r'^logout/', logout),
23                          url(r'^register/', register),
24                          url(r'^register2/', register2),
25  ]
```

Admin.py

```
from django.contrib import admin
from loginsys import models
# Register your models here.


class ServiceClass(admin.ModelAdmin):
    fields = (('id', 'title'), 'price', 'picture')
    list_filter = ('price',)
    list_display = ('id', 'title', 'price', 'picture')
    search_fields = ('title', 'id')
    list_per_page = 10

class UserServiceClass(admin.ModelAdmin):
    fields = (('id', 'service'), 'user', 'description', 'desc_picture')
    list_filter = ('id', 'user')
    list_display = ('id', 'service', 'user', 'description', 'desc_picture')
    search_fields = ('user', 'service')
    list_per_page = 10


admin.site.register(models.Service, ServiceClass)
admin.site.register(models.UserService, UserServiceClass)
```

Forms.py

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



class ImageForm(forms.Form):
    workers = forms.ModelChoiceField(queryset=User.objects.all())
    def label_from_instance(self, obj):
        return self.workers.__str__()


class UserServiceForm(forms.ModelForm):
    class Meta:
        model = UserService
        fields = ["user", "service", "description", "desc_picture"]


class ServiceForm(forms.ModelForm):
    class Meta:
        model = Service
        fields = ["id", "title", "price", "picture"]
```

Models.py

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

```
class Service(models.Model):
    id = models.AutoField(primary_key=True)
    title = models.TextField(max_length=30)
    users = models.ManyToManyField(User, through='UserService')
    price = models.IntegerField(default=500)
    picture = models.ImageField(default='order/images/333.png', upload_to='order/images')

    def __str__(self):
        return self.title


class UserService(models.Model):
    id = models.AutoField(primary_key=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    service = models.ForeignKey(Service, on_delete=models.CASCADE)
    description = models.CharField(max_length=255, default="no desc")
    desc_picture = models.ImageField(default='order/images/333.png',
upload_to='order/images')
```

Views

```
from django.http import HttpResponse, HttpResponseRedirect
from django.shortcuts import render
from django.utils.decorators import method_decorator
from django.views.generic import View
from django.views.generic import ListView
from datetime import datetime
from django.urls import reverse
from django.contrib import auth
from loginsys.models import *
from Lab5_App.forms import *
from django.core.paginator import Paginator, EmptyPage, PageNotAnInteger
import sqlite3

from django.contrib.auth.decorators import login_required


def function_view(request):
    return HttpResponse('response from func view')


class ClassBased(View):
    def get(self, request):
        return HttpResponse('response from class based view')


def main page(request):
    data = {'services': Service.objects.all()}
    paginate_by = 5

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


class OrdersList(ListView):
    form_class = ServiceForm
    model = Service
    template_name = "orders.html"

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

    def post(self, request):
```

```
        form = self.form_class(request.POST or None, request.FILES or None)
        if form.is_valid():
            instance = form.save(commit=False)
            instance.save()
            return HttpResponseRedirect(reverse('order_url', args=(instance.pk,)))
        return render(request, self.template_name, {'form': form})


class AllOrders(ListView):
    form_class = UserServiceForm
    model = UserService
    template_name = "allorders.html"

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

    def post(self, request):
        form = self.form_class(request.POST or None, request.FILES or None)
        if form.is_valid():
            instance = form.save(commit=False)
            instance.save()
            return HttpResponseRedirect(reverse('order_url', args=(instance.pk,)))
        return render(request, self.template_name, {'form': form})


def order_add(request):
    users = User.objects.all()
    services = Service.objects.all()
    form = UserServiceForm(request.POST or None, request.FILES or None)
    context = {"form": form, "services": services, "users": users}
    if form.is_valid():
        instance = form.save(commit=False)
        instance.save()
        return HttpResponseRedirect(reverse('orderinfo_url', args=(instance.pk, )))
    return render(request, "order.html", context)


def order_info(request, id):
    userservice = UserService.objects.filter(id=id)
    userservice = userservice.first()
    #if request.POST and titleform.is_valid():

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


def service_info(request, id):
    service = Service.objects.filter(id=id)
    service = service.first()
    #if request.POST and titleform.is_valid():
    orders = UserService.objects.filter(service=id)
    return render(request, 'service.html', locals())
```

login and register views

```
from django.http import HttpResponse, HttpResponseRedirect
from django.shortcuts import render, redirect, render_to_response
from django.template.context_processors import csrf
from django.views.generic import View
from django.contrib.auth.models import User
from datetime import datetime
from django.contrib import auth
from django.contrib.auth.forms import UserCreationForm
from django.forms import EmailField, CharField



def function_view(request):
```

```
    return HttpResponse('response from func view')


class ClassBased(View):
    def get(self, request):
        return HttpResponse('response from class based view')


class CustomUserCreationForm(UserCreationForm):
    email = EmailField(label=("Email adress"), required=True, help_text=("Required."))
    firstname = CharField(label=("firstname"), max_length=20, min_length=1, required=True)
    lastname = CharField(label=("lastname"), max_length=20, min_length=1, required=True)

    class Meta:
        model = User
        fields = ("username", "email",
                  # "password1", "password2",
                  "firstname", "lastname")

    def save(self, commit=True):
        user = super().save(commit=False)
        user.email = self.cleaned_data["email"]
        user.firstname = self.cleaned_data["firstname"]
        user.lastname = self.cleaned_data["lastname"]
        if commit:
            user = User.objects.create_user(username=user.username,
                                            password=user.password,
                                            email=user.email,
                                            last_name=user.lastname,
                                            first_name=user.firstname)
            #user.save()
        return user

def login(request):
    args = {}
    args.update(csrf(request))
    if request.POST:
        username = request.POST.get('username', '')
        password = request.POST.get('password', '')
        user = auth.authenticate(username=username,password=password)
        if user is not None:
            auth.login(request, user)
            return redirect('main')
        else:
            args['login_error'] = "Пользователь не найден"
            return render_to_response('login.html',args)

    else:
        return render_to_response('login.html',args)


def logout(request):
    auth.logout(request)
    return redirect('main')


def register(request):
    args = {}
    #args.update(csrf(request))
    args['form'] = CustomUserCreationForm()
    if request.POST:
        newuser_form = CustomUserCreationForm(request.POST)
        if newuser_form.is_valid():
            #newuser_form.save()
            #newuser = auth.authenticate(username=newuser_form.cleaned_data['username'],
password=newuser_form.cleaned_data['password2'])
            # import pdb
            # pdb.set_trace()
```

```
            #auth.login(request, newuser)
            user = newuser_form.save()
            auth.login(request, user)
            return redirect('main')
        else:
            args['error_messages'] = newuser_form.error_messages
            args['form'] = newuser_form
    return render(request, 'register.html', args)


def register2(request):
    errors = {}
    #errors.update(csrf(request))
    if request.method == 'POST':
        username = request.POST.get('username')
        if not username:
            errors['username'] = 'Введите логин'
        elif len(username) < 5:
            errors['username'] = 'Логин минимум 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'] = 'Пароль не меньше 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.keys():
            user = User.objects.create_user(username=username,
                                            password=password,
                                            email=email,
                                            first_name=firstname,
                                            last_name=surname)
            return HttpResponseRedirect('/')
    return render(request, 'register2.html', errors)
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<meta charset="utf-8">
<head>
    {% load static %}
    <link rel="stylesheet" type="text/css" href="{% static 'bootstrap.css' %}">
    <link rel="stylesheet" type="text/css" href="{% static 'bootstrap-theme.css' %}">
    <script type="text/javascript" src="{% static '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>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
    <div class="navbar navbar-inverse">
        <div class="container">
            <div class="navbar-header">
```

```
                <a class="navbar-brand" href="{% url 'main' %}">DZ</a>
            </div>
        </div>
        <ul class="nav-tabs">
            {% if request.user.username %}
                <li><a style="color: rgba(255,254,246,0.52)" href="/auth/logout/">Выход ({{
request.user.username }})</a></li>
            {% else %}
                <li><a style="color: rgba(255,254,246,0.52)" href="/auth/login">Войти</a></li>
            {% endif %}
        <br><li><a style="color: rgba(255,254,246,0.52); margin-right: 1cm;"
href="/order">Сделать заказ</a></li>
         <li><a style="color: rgba(255,254,246,0.52)" href="/allorders">Список
заказов</a></li>
        </ul>
    </div>
    <div class="container">
        <header>{% block header %}{% endblock %}</header>
        <div>
            {% block body %}{% endblock %}
        </div>
    </div>
    <hr>
    <div class="container">
        <div class="footer">
            <p>&copy Чечелев Денис Сергеевич РТ5-51 2017-2018</p>
        </div>
    </div>
</body>
</html>
```

Orders.html

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

{% block title %}Услуги{% endblock %}

{% block header %}<h1 href="/order/" class="text-left">Заказать услугу</h1>{% endblock %}

{% block body %}
<div id="body" style="text-align: left">
    <div al-app  al-init="count=5" @scroll.debounce-50="($element.scrollTop >
$element.scrollHeight - $element.clientHeight - 150)?count++:0" class="body">
        <div al-repeat="n in count">
    {% if object_list %}
            <table id="services">
                <div class="service">
                    {% for service in object_list %}
                        <a href="/service/{{ service.id }}" class="course-element">
                            <div class="service-photo">
                                {% if service.picture %}
                                    <img class="img-service" src="{{ service.picture.url }}"
width="250" height="250">
                                {% endif %}
                            </div>
                            <span class="service-title">{{ service.title }}</span>
                        </a>
                    {% endfor %}
                </div>
            </table>
{#            {% if is_paginated %}#}
{#                <div class="pagination">#}
{#                    <span class="page-links">#}
{#                        {% if page_obj.has_previous %}#}
{#                            <a class="aquo" href="/?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="/?page={{ page }}" class="page">{{ page
}}</a>#}
{#                                  {% endifequal %}#}
{#                              {% endif %}#}
{#                          {% endfor %}#}
{#                          {% if page_obj.has_next %}#}
{#                              <a class="aquo" href="/?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 %}#}
    {% endif %}
     </div>
</div>
{% endblock %}



</div>
```

Order.html

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

{% block title %}Заказ{% endblock %}

{% block header %}<h1 class="text-left">Заказать услугу</h1>{% endblock %}

{% block body %}
<div class="container">
  <!-- Trigger the modal with a button -->
  <button type="button" class="btn btn-info btn-lg" data-toggle="modal" data-
target="#myModal">Заказ</button>

  <!-- Modal -->
  <div class="modal fade" id="myModal" role="dialog">
    <div class="modal-dialog">

      <!-- Modal content-->
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal">&times;</button>
          <h4 class="modal-title">Добавление нового заказа</h4>
        </div>
        <div class="modal-body">

<div id="body" style="text-align: left">
        <br>
        <form method="POST" enctype="multipart/form-data">
            <div class="add-form">
                {% csrf_token %}
                <div class="order_add">
                    <label for="id_service">Услуга:</label>
                    <div class="data-and-error">
                        <select name="service" required id="id_service">
                            <option value="" selected>Не выбрано</option>
```

```
                            {% for service in services %}
                                <option value="{{ service.id }}">{{ service }}</option>
                            {% endfor %}
                        </select>
                        <h5 id="error_service_order">Выберете услугу</h5>
                    </div>
                </div>
                <div class="order_add">
                    <label for="id_user">Исполнитель:</label>
                    <div class="data-and-error">
                        <select name="user" required id="id_user">
                            <option value="" selected>Не выбрано</option>
                            {% for user in users %}
                                <option value="{{ user.id }}">{{ user }}</option>
                            {% endfor %}
                        </select>
                        <h5 id="error_user_order">Выберете исполнителя</h5>
                    </div>
                </div>
                <div class="order_add">
                    <label for="description">Дополнительная информация:</label>
                    <div class="data-and-error">
                        <textarea name="description" cols="40" rows="10" resize="none"
maxlength="2055"
                                  id="description"></textarea>
                    </div>
                </div>
                 <div class="order_add">
                    <label for="desc_picture">Фото:</label>
                    <div class="data-and-error">
                        <input type="file" name="desc_picture" id="desc_picture"/>
                    </div>
                </div><br>
                <button type="submit" id="btn_add">Добавить заказ</button>
                <br>
            </div>
        </form>
    </div>

</div>
        <div class="modal-footer">
          <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
        </div>
      </div>

    </div>
  </div>

</div>

{% endblock %}
```

Allorders.html

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

{% block title %}Услуги{% endblock %}

{% block header %}<h1 class="text-left">Заказы услуг</h1>{% endblock %}

{% block body %}
<div id="body" style="text-align: left">
    {% if object_list %}
            <table id="services">
                <div class="service">
                    {% for order in object_list %}
```

```
                        <a href="/allorders/{{ order.id }}" class="course-element">
                            <span class="service-title">Заказ № {{ order.id }} - {{
order.service }} - {{ order.user }}</span><br>
                        </a>
                    {% endfor %}
                </div>
            </table>
            {% if is_paginated %}
                <div class="pagination">
                    <span class="page-links">
                        {% if page_obj.has_previous %}
                            <a class="aquo" href="/?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="/?page={{ page }}" class="page">{{ page }}</a>
                                {% endifequal %}
                            {% endif %}
                        {% endfor %}
                        {% if page_obj.has_next %}
                            <a class="aquo" href="/?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 %}
    {% endif %}
{% endblock %}
```

Orderinfo.html

```
{% extends 'base.html' %}
                {% block title %}Услуга № {% endblock %}

                {% block header %}<h1></h1>{% endblock %}

                {% block body %}
                    <form action="" method="post">
                                    <div style="font-weight: 600; font-size: 150%" >{{
userservice.service }}</div>
                                    <br>
                                        Исполнитель: {{ userservice.user }} <br>
                                        Description of the task: {{ userservice.description
}}<br>
                                       {% if userservice.desc_picture %}
                                       <img class="img-service" src="{{
userservice.desc_picture.url }}" width="250" height="250">
                                         {% endif %}
                                </form>

                {% endblock %}
```

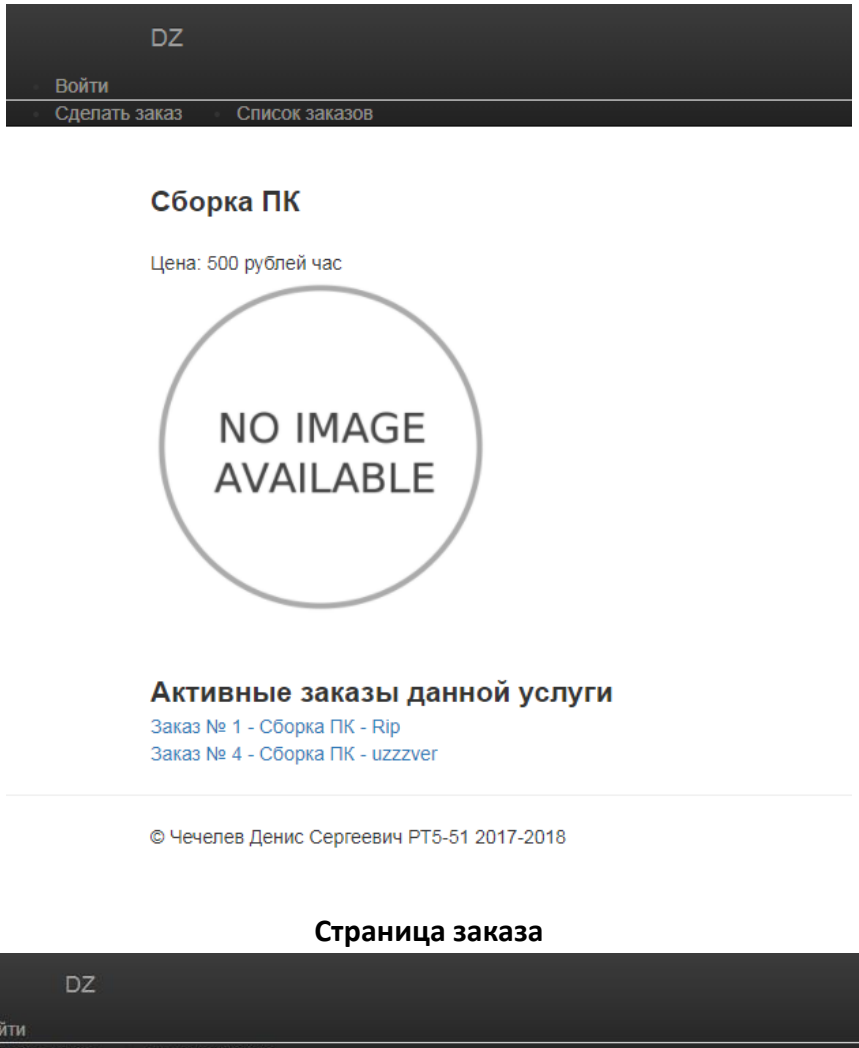
Service.html

```
{% extends 'base.html' %}
                {% block title %}Услуга № {% endblock %}
```

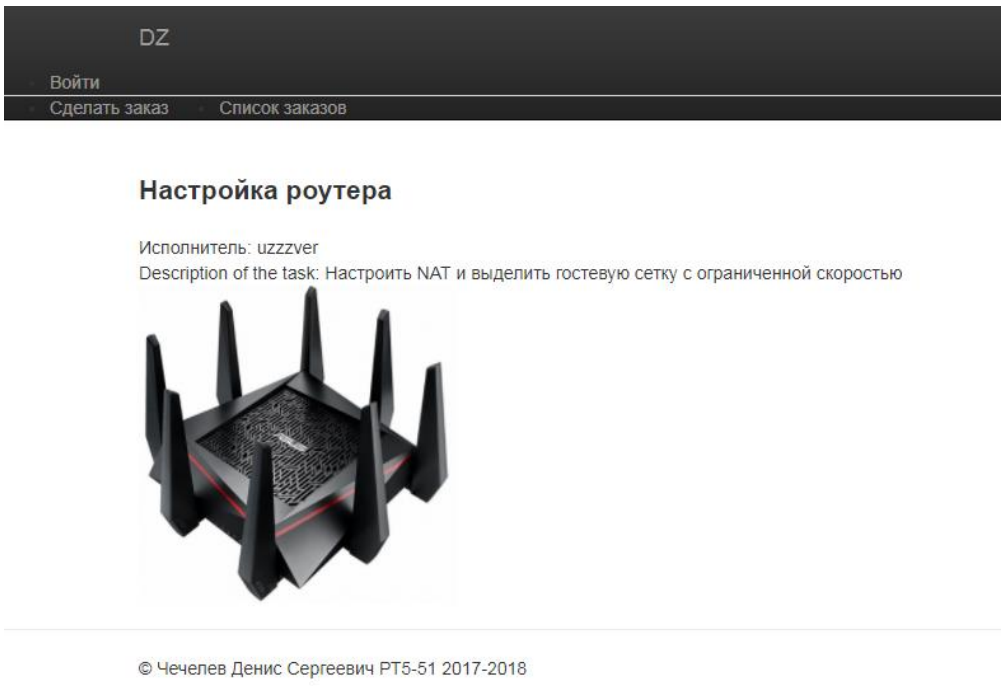
```
                {% block header %}<h1></h1>{% endblock %}

                {% block body %}
                    <form action="" method="post">
                                    <div style="font-weight: 600; font-size: 150%" >{{
service.title }}</div>
                                    <br>
                                        Цена: {{ service.price }} рублей час<br>
                                       {% if service.picture %}
                                       <img class="img-service" src="{{ service.picture.url }}"
width="250" height="250">
                                       {% endif %}<br><br><br>
                                    <div style="font-weight: 600; font-size: 150%" >Активные
заказы данной услуги</div>
                                   {% for order in orders %}
                                       <a href="/allorders/{{ order.id }}" class="course-
element">
                                           <span class="service-title">Заказ № {{ order.id }} -
{{ order.service }} - {{ order.user }}</span><br>
                                       </a>
                                   {% endfor %}
                            </form>
                {% endblock %}
```

Список услуг

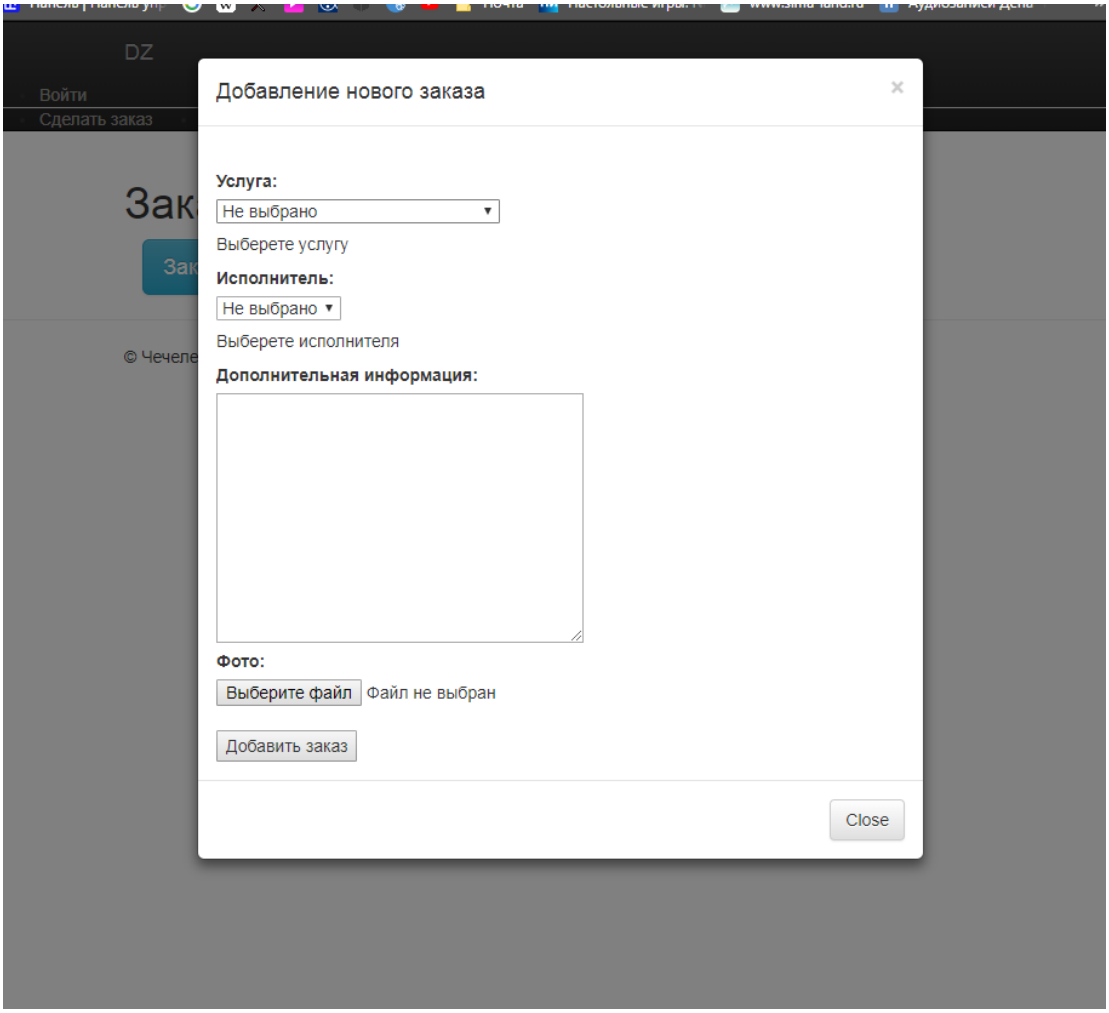
Страница услуги



Страница заказа



Страница добавления заказа

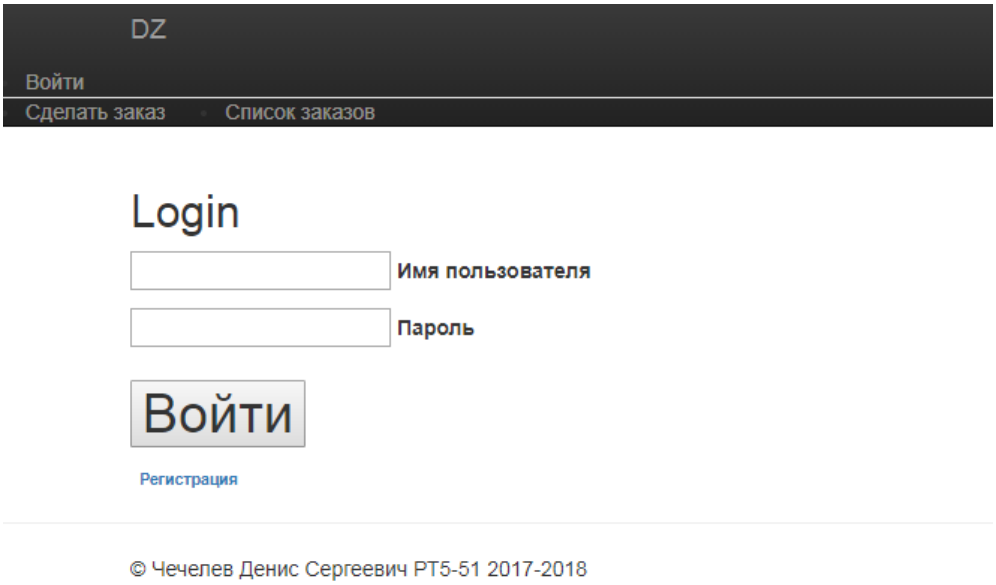


Список заказов

Заказы услуг

Заказ № 1 - Сборка ПК - Rip
Заказ № 2 - Восстановление жесткого диска - Rip
Заказ № 3 - Я что то нажал и не включается - Rip
Заказ № 4 - Сборка ПК - uzzzver
Заказ № 5 - Настройка роутера - uzzzver
Заказ № 8 - Замена комплектующих - uzzzver
Заказ № 9 - Восстановление жесткого диска - uzzzver

Авторизация



Регистрация (их две осталось после лаб 7)