

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»



Лабораторные работы по курсу:

«Разработка Интернет-приложений»

## Отчет по домашнему заданию

Исполнитель:

Студент группы ИУ5-51

Макаров Д.А.

Преподаватель:

Гапанюк Ю.Е.

«\_\_» \_\_\_\_\_

\_\_\_\_\_

Задание:

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

Код:

Views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect, HttpResponse
from django.views.generic import View
from django.views.generic import ListView
from .models import *
from django import forms
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.views.decorators.csrf import ensure_csrf_cookie
from django.views.generic import DetailView
from django.core.files.storage import FileSystemStorage
from django.core.files import File
from el_pagination.views import AjaxListView
from el_pagination.decorators import page_template
```

```
class CustomerView(ListView):
    model = Customer
    template_name = 'customer_list.html'
```

# форма регистрации

```
class RegistrationForm(forms.Form):
    username = forms.CharField(min_length=5, label='Логин')
    password = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Пароль')
    password2 = forms.CharField(min_length=8, widget=forms.PasswordInput,
label='Повторите ввод')
    email = forms.EmailField(label='Email')
    last_name = forms.CharField(label='Фамилия')
    first_name = forms.CharField(label='Имя')
```

```
class ComputerForm(forms.ModelForm):
    class Meta(object):
        model = Computer
```

```
fields = ['name', 'price', 'pic', 'description', 'quantity', 'type']
```

```
def save(self):
    computer = Computer()
    computer.name = self.cleaned_data.get('name')
    computer.price = self.cleaned_data.get('price')
    computer.type = self.cleaned_data.get('type')
    computer.quantity = self.cleaned_data.get('quantity')
    computer.description = self.cleaned_data.get('description')
    f = self.cleaned_data.get('pic')
    computer.pic = f
    computer.save()
```

```
class AuthorizationForm(forms.Form):
    username = forms.CharField(label='Логин')
    password = forms.CharField(widget=forms.PasswordInput, label='Пароль')
```

```
# регистрация
```

```
def add(request):
    if request.method == 'POST':

        name1 = request.POST.get('name')
        price1 = request.POST.get('price')
        type1 = request.POST.get('type')
        quantity1 = request.POST.get('quantity')
        description1 = request.POST.get('description')
        pic1 = request.FILES.get('pic')
        print(type(price1))
        try:
            int(price1)
            int(quantity1)
        except:
            return render(request, 'add.html', {'error': "Неверно введена цена или количество"})

        else:
            comp = Computer(name=name1, price=price1, type=type1,
quantity=quantity1, description=description1,
pic=pic1)
            comp.save()
```

```
    return HttpResponseRedirect('/item-' + name1)
return render(request, 'add.html', locals())
```

```
def registration(request):
    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:
            data = form.cleaned_data
            user = User.objects.create_user(data['username'], data['email'],
data['password'])
            print(user)
            cust = Customer()
            cust.user = user
            cust.first_name = data['first_name']
            cust.last_name = data['last_name']
            cust.save()
            return HttpResponseRedirect('/authorization/')
        else:
            form = RegistrationForm()

    return render(request, 'registration.html', {'form': form})
```

```
# авторизация django
def authorization(request):
    if request.method == 'POST':
        form = AuthorizationForm(request.POST)
        print(form)
        data = form.cleaned_data
        if form.is_valid():
            user = authenticate(request, username=data['username'],
password=data['password'])
```

```

# user = authenticate(request, username='petrov',password='12345678')
print(len(data['username']), len(data['password']))
if user is not None:
    login(request, user)
    return HttpResponseRedirect('/success_authorization')
else:
    form.add_error('username', ['Неверный логин или пароль'])
    # raise forms.ValidationError('Имя пользователя и пароль не
подходят')
else:
    form = AuthorizationForm()
return render(request, 'authorization.html', {'form': form})

```

```

# успешная авторизация django
@login_required(login_url='/authorization')
def success_authorization(request):
    return HttpResponseRedirect("/")

```

```

# class ItemsView(AjaxListView):
#     template_name = 'labApp/templates/list.html'
#     context_object_name = "list_object"
#     page_template = 'labApp/templates/list_object.html'
@page_template('list_object.html')
def ItemsView(request, template='list.html', extra_context=None):
    dict_customers = {} # код компа - массив покупателей
    data = Computer.objects.all()
    form = ComputerForm()
    for c in data: # по компам
        customers = [] # купили
        orders = Order.objects.all()
        for o in orders: # по заказам
            cur_cust = o.customer.first_name
            # print(cur_cust)# покупатель, сделавший заказ
            for item in o.items.all(): # по элементам заказа
                if item.name == c.name: # если текущий комп
                    if cur_cust not in customers:
                        customers.append(cur_cust) # покупателя в купили
            dict_customers[c.name] = customers # список покупателей для компа
    return render(request, template,
        context={'search': data,

```

```
        'customers': dict_customers,

        'form': form
    })
```

```
class OneItem(DetailView):
    model = Computer
    context_object_name = 'computer'
    template_name = 'object.html'

    def get_context_data(self, **kwargs):
        context = super(OneItem, self).get_context_data(**kwargs)
        dict_customers = {} # код компа - массив покупателей
        data = Computer.objects.all()

        for c in data: # по компам
            customers = [] # купили
            orders = Order.objects.all()
            for o in orders: # по заказам
                cur_cust = o.customer.first_name
                # покупатель, сделавший заказ
                for item in o.items.all():
                    # по элементам заказа
                    # print(c.name)
                    if item.name == c.name: # если текущий комп
                        if cur_cust not in customers:
                            customers.append(cur_cust) # покупателя в купили
            dict_customers[c.name] = customers
            # print(dict_customers)
        context['customers'] = dict_customers
        # print(context)
        return context

def ord(request, namekomp):
    if request.method == "GET":
        comp = Computer.objects.get(name=namekomp)
        id = request.user.id - 1
        cust = Customer.objects.get(id=id)
        price = 1
        order = Order()
```

```

order.customer = cust
order.price = price
order.total = 1
order.save()
order2 = BelongTO()
order2.quantity = 1
order2.id = id
order2.item_id = comp
order2.order_id = id
order2.save()
return HttpResponseRedirect('/orders/')

```

```

class OrdersView(View):
    def get(self, request):
        empty_orders = []
        computers_in_order = BelongTO.objects.all() # код заказа - компы
        prices = { } # цены
        data = Order.objects.filter(
            customer_id=request.user.id - 1).all() # заказы пользователя
        for o in data:
            computers = BelongTO.objects.filter(
                order_id=o.code).all() # компьютеры заказа
            if len(computers) == 0:
                empty_orders.append(o.code)
            total = 0
            for c in computers:
                cur_comp = Computer.objects.get(name=c.item_id)
                if c.item_id not in prices.keys():
                    prices[c.item_id] = cur_comp.price
                total = total + cur_comp.price * c.quantity
            o.total = total
            o.save()

        return render(request, 'orders.html',
            context={ "data": data,
                      "computers": computers_in_order,
                      "prices": prices,
                      'empty_orders': empty_orders })

def logout_view(request):

```

```

    logout(request)
    return HttpResponseRedirect('/')
admin.py
from django.contrib import admin
from .models import *

# Register your models here.
@admin.register(Customer)
class CustomerAdmin(admin.ModelAdmin):
    list_display = ('user',
                    'customer_name',
                    'email',
                    'first_name',
                    'last_name'
                    )

class BelongTOInline(admin.TabularInline):
    model = BelongTO
    extra = 1
    verbose_name_plural = 'Orders list'

@admin.register(Order)
class OrderAdmin(admin.ModelAdmin):
    def total(self, request):
        total = 0
        items = BelongTO.objects.filter(order_id=request.code)
        for i in items:
            computer = Computer.objects.get(name=i.item_id)
            total += computer.price
        return total

    readonly_fields = ('total',)
    list_display = ('code', 'customer', 'total', 'is_open', 'date',)
    inlines = (BelongTOInline,)

@admin.register(Computer)
class ComputerAdmin(admin.ModelAdmin):
    def orders(self, request):

```



```

orders = []
for s in BelongTO.objects.filter(item_id=request.name):
    orders.append(s.order_id)
return orders

```

```

inline = (BelongTOInline,)
list_display = ('name',
                'price',
                'description',
                'pic',
                'type',
                'quantity',
                'orders')

```

Models.py

```

from django.db import models
from django.contrib.auth.models import User, UserManager
from django.contrib import admin
from django.utils import timezone

```

```

class Customer(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    customer_name = models.CharField(max_length=30)
    email = models.EmailField(max_length=75)
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    objects = UserManager()

    def __unicode__(self):
        return self.customer_name

```

```

class Computer(models.Model):
    PersonalComputer = 'Personal Computer'
    Monoblock = 'Monoblock'
    Laptop = 'Laptop'
    computer_types = {
        (PersonalComputer, 'Personal computer'),
        (Monoblock, 'Monoblock'),
        (Laptop, 'Laptop'),
    }

```

```

name = models.CharField(max_length=255, unique=True, primary_key=True)
price = models.IntegerField(null=True, default=0)
description = models.TextField(
    max_length=500, default='No description yet')
type = models.CharField(
    max_length=30, choices=computer_types, default=PersonalComputer)
# pic = models.ImageField(upload_to='media/', default='media/ts.jpg')
quantity = models.IntegerField(null=True, default=0)
pic = models.ImageField(upload_to='media/', null=True, blank=True)

def __str__(self):
    return self.name

```

```

class Order(models.Model):
    date = models.DateTimeField(default=timezone.now)
    items = models.ManyToManyField(Computer, through='BelongTo')
    code = models.AutoField(max_length=6, unique=True, primary_key=True)
    customer = models.ForeignKey(Customer, on_delete=models.CASCADE)
    total = models.DecimalField(
        decimal_places=2, max_digits=10, unique=False, default=0.0)
    is_open = models.BooleanField(default=True)

    def __str__(self):
        return str(self.code)

```

```

class BelongTO(models.Model):
    item = models.ForeignKey(Computer, on_delete=models.CASCADE)
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    quantity = models.IntegerField(null=True)
    id = models.CharField(unique=True, primary_key=True, max_length=255)

    def __str__(self):
        return self.id

```

urls.py

```

"""lab5 URL Configuration

```

The `urlpatterns` list routes URLs to views. For more information please see:  
<https://docs.djangoproject.com/en/1.11/topics/http/urls/>  
 Examples:

## Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `url(r'^$', views.home, name='home')`

## Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `url(r'^$', Home.as_view(), name='home')`

## Including another URLconf

1. Import the `include()` function: `from django.conf.urls import url, include`
2. Add a URL to `urlpatterns`: `url(r'^blog/', include('blog.urls'))`

"""

```
from django.conf.urls import url, include
from django.contrib import admin
from labApp.views import *
from django.conf.urls.static import static
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.conf import settings
```

```
urlpatterns = [
    url(r'^customers/$', CustomerView.as_view(), name='customers_url'),
    url(r'^item-(?P<pk>[A-Za-z0-9- ]+)$',
        OneItem.as_view(), name='item_view'),
    url(r'^registration/$', registration, name='registration'),
    url(r'^add/$', add, name='add'),
    url(r'^authorization/$', authorization, name='authorization'),
    url(r'^logout/$', logout_view, name='logout'),
    url(r'^orders/$', OrdersView.as_view(), name='orders'),
    url(r'^$', ItemsView),
    url(r'^success_authorization$', success_authorization,
name='success_authorization'),
    url(r'^admin/', include(admin.site.urls)),
    url(r'^ord/(?P<namekomp>[A-Za-z0-9- ]+)$', ord, name='ord', )]
urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

## list\_object.html

```
{% load static %}
{% load el_pagination_tags %}

{% paginate search %}
{% for item in search %}
    <section class="panel panel-search">
```

```

{#      <div class="panel-heading bg-primary">#}
        <div class="panel-title">
            <h2>
                <div class="title">
                    <a href="/item-{{ item.name }}">{{ item.name }}</a> <small> {{
item.type }}</small>
                </div>
            </h2>
        </div>
{#      </div>#}
        <div class="panel-body ">
            <div class="row">
                <div class="col-lg-4">
                    
                </div>
                <div class="col-lg-4">
                    <h4>{{ item.description }}</h4>
<h4>Have already bought such users as:</h4>
                    {% for key, value in customers.items %}
                        {% if key == item.name %}
                            {% for v in value %}
                                <h5>{{ v }}</h5>
                            {% endfor %}
                        {% endif %}
                    {% endfor %}
                    <h4>Available: </h4>
                    <h4 id="available{{ item.name }}">{{ item.quantity }}</h4>

                </div>
            </div>
        </div>
        <div class="panel-footer">
            <div class="row">
                <div class="col-md-offset-4 col-md-2">
                    <h4>Price: {{ item.price }}</h4>
                </div>
            </div>
        </div>
    </section>
{% empty %}
    Finish
{% endfor %}

```

```
{% show_more % }
```

Object.html

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

```
{% csrf_token % }
```

```
{% block title % }Products{% endblock % }
```

```
{% block content % }
```

```
<div class="container-fluid">
```

```
<div class="blog-header">
```

```
<h1 style="font-family: Lobster; text-align: center">{{ computer.name }}</h1>
```

```
</div>
```

```
<div class="row">
```

```
<br>
```

```
<div class="col-lg-4">
```

```
{% load staticfiles % }
```

```

```

```
</div>
```

```
<div class="col-lg-4">
```

```
<h4>{{ computer.name }} is a modern {{ computer.type }}.</h4>
```

```
<br>
```

```
<h5>{{ computer.description }}</h5>
```

```
<br>
```

```
<h4>Have already bought such users as:</h4>
```

```
{% for key, value in customers.items % }
```

```
{% if key == computer.name % }
```

```
{% for v in value % }
```

```
<h5>{{ v }}</h5>
```

```
{% endfor % }
```

```
{% endif % }
```

```
{% endfor % }
```

```
Available: <span id="available">{{ computer.quantity }}</span></h4>
```

```
<br>
```

```
<h4>Price: {{ computer.price }}</h4>
```

```
</div>
```

```
<div class="col-lg-3 blog-sidebar">
```

```
<section class="panel panel-search">
```

```

        { % if user.is_authenticated % }

        <a role="button" class="btn btn-success btn-lg btn-block"
href="/">Start page</a>
        <a role="button" class="btn btn-danger btn-lg btn-block" href="{ %
url 'ord' computer.name % }">Order
        it!</a>

        <a role="button" class="btn btn-success btn-lg btn-block" href="{ %
url 'orders' % }">Orders</a>

        { % endif % }

```

```

</section>

```

```

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

```

## List.html

```

{ % extends 'base.html' % }
{ % csrf_token % }
{ % block title % }Products{ % endblock % }
{ % block content % }
    <div class="container-fluid">
        <div class="blog-header">
            { % if user.is_authenticated % }
            <h1 class="blog-title col-lg-offset-3"><b>Search results</b></h1>
            </div>
            <div class="row">
                <div class="col-lg-8 blog-main">
                    { % include 'list_object.html' with item=item % }
                    { % block js % }
                        { { block.super } }
                        <script src="http://code.jquery.com/jquery-latest.js"></script>
                        <script src="/static/js/el-pagination.js"></script>
                        <script>$.endlessPaginate({
paginateOnScroll: true,
paginateOnScrollMargin: 20

```

```

});</script>
    {% endblock % }
    <div id="loading_block"></div>
</div>
<div class="col-lg-3 blog-sidebar">
    <section class="panel panel-search">

        <a role="button" class="btn btn-success btn-lg btn-block"
href="/">Start page</a>
        <a role="button" class="btn btn-success btn-lg btn-block"
href="{% url 'add' %}">Add new computer</a>

        <a role="button" class="btn btn-success btn-lg btn-block"
href="{% url 'orders' %}">Orders</a>

    {% else % }
    <h1 class="blog-title col-lg-offset-3"><b>Добро пожаловать. Для
использования всего функционала авторизуйтесь</b></h1>
    {% endif % }

</section>

</div>
</div>
</div>
{% endblock % }
Authorization.html
{% extends 'base.html' % }

{% block title % }authorization{% endblock % }

{% block content % }
<div class="main_block_elem">
<form method="POST" class="form-horizontal">
    <h3 style="text-align:center">Авторизация пользователя</h3>
    {% 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">Войти</button>
        </div>
    </div>
</form>
</div>
{% endblock % }
Registration

```

```

{% extends 'base.html' % }

```

```

{% block title % }registration{% endblock % }

```

```

{% block content % }
<div class="main_block_elem">
<form method="POST" class="form-horizontal">
    <h3 style="text-align:center">Регистрация пользователя</h3>
    {% 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>
{% endblock %}
```

## Add new

Неверно введена цена или количество

Name

Price

Type Personal Computer ▾

Quantity

Description

Picture

Choose File No file chosen

ADD

HW

 test3

Выйти

## Your orders

Code	Date	Active
7	22 декабря 2017 г. 1:52	Yes

# Add new

Name

name

Price

price

Type 

Personal Computer ▼

Quantity

quantity

Description

description

Picture

Choose File

No file chosen

ADD

## NewPC



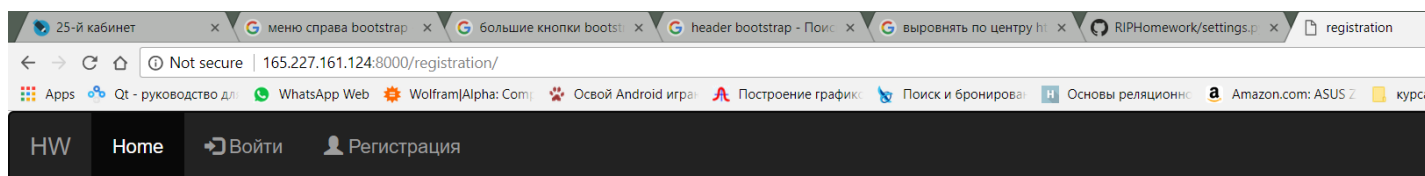
NewPC is a modern Personal Computer.

text

Available: 1

Price: 100


- Start page
- Order it!
- Orders



## Регистрация пользователя

Логин:

Пароль:

 Please lengthen this text to 8 characters or more (you are currently using 1 character).

Email:

Фамилия:

Имя:



## Авторизация пользователя

Логин:

Пароль:

