**Министерство цифрового развития, связи и массовых коммуникаций**

**Российской Федерации**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное**

**Образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра: "Математическая кибернетика и информационные технологии"

**Курсовая работа на тему:**

**«Создание проекта Интернет-Магазина»**

Выполнил студент

группы БСС1902:

Зайка А. И.

**Москва 2021 г.**

**Задание:**

Создание проекта интернет-магазина на базе: Python, Django, MySQL, JS

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'])

подходят')

else:

# 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

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

inlines = (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'^$', views.home, name='home') Class-based views
3. Add an import: from other\_app.views import Home
4. Add a URL to urlpatterns: url(r'^$', Home.as\_view(), name='home') Including another URLconf
5. Import the include() function: from django.conf.urls import url, include
6. 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">

<img src="{{ item.pic }}" width="90%"/>

</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 %}

<img src="{{ computer.pic }}" width="100%" border="10%"/>

</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 sr[c="ht](http://code.jquery.com/jquery-latest.js)tp:[//code.jquery.com/jquery-latest.js](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 %}











