**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Системы обработки информации и управления»

Кафедра ИУ5. Курс «РИП»

Отчет по лабораторным работам №5

## «Шаблонизация»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53 |  | преподаватель каф. ИУ5 |
| Белков А.Д. |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2017 г.

**Задание и порядок выполнения ЛР №5**

1. В этой ЛР вы создадите Django-проект, покажете пользователю статичную   
   страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги,   
   наследование шаблонов.
2. Создать проект
3. Реализовать view, в которых генерируются html-страницы
4. В шаблонах должны быть использованы рассмотренные конструкции:   
   переменные, вложенные значения, циклы, условия
5. Все шаблоны должны расширять базовый шаблон
6. Для элементов списка использовать тег include
7. По нажатии на элемент списка должна открываться страница информации об   
   элементе
8. Для верстки необходимо использовать Bootstrap

**Код программы**

**urls.py**

from lab\_app.views import ProductView, ListProductView  
  
urlpatterns = [  
 url(r'^admin/', admin.site.urls),  
 url(r'^product/(?P<product\_id>\d+)', ProductView.as\_view()),  
 url(r'^$', ListProductView.as\_view(),  
 name='products-list'),  
]

**base-template.html**

<!DOCTYPE html>  
<html lang="en">  
 <head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <!-- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these tags -->  
 <meta name="description" content="">  
 <meta name="author" content="">  
  
 <title>{% block title %}{% endblock %}</title>  
  
 <!-- Bootstrap core CSS -->  
 <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">  
  
 <!-- Custom styles for this template -->  
 <style>  
 body {  
 padding-top: 60px;  
 background-color: white;  
 }  
 .jumbotron {  
 margin: 20px -30px;  
 padding: 20px 20px;  
 }  
 </style>  
  
 </head>  
  
 <body>  
  
 <nav class="navbar navbar-inverse navbar-fixed-top">  
 <div class="container-fluid">  
 <!-- Brand and toggle get grouped for better mobile display -->  
 <div class="navbar-header">  
 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">  
 <span class="sr-only">Toggle navigation</span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 <span class="icon-bar"></span>  
 </button>  
 <a class="navbar-brand" href="/">Lab #5</a>  
 </div>  
  
 <!-- Collect the nav links, forms, and other content for toggling -->  
 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">  
  
 {% block navbar-links %}  
  
 {% endblock %}  
  
 </div><!-- /.navbar-collapse -->  
 </div><!-- /.container-fluid -->  
 </nav>  
  
 <div class="container">  
  
 <div class="starter-template">  
 {% block body %}  
  
 {% endblock %}  
 </div>  
  
 </div><!-- /.container -->  
  
 <!-- Bootstrap core JavaScript  
 ================================================== -->  
 <!-- Placed at the end of the document so the pages load faster -->  
 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>  
 <script src="{% static 'js/bootstrap.min.js' %}"></script>  
 </body>  
</html>

**product.html**

{% extends 'base\_template.html' %}  
  
{% block title %} {% endblock %}  
  
{% block navbar-links %}  
  
 <ul class="nav navbar-nav">  
 <li><a href="/">< back to products</a></li>  
 </ul>  
  
{% endblock %}  
  
{% block body %}  
  
 <h1>{{ product.name }}</h1>  
 <div class="media">  
 <div class="media-left">  
 <a href="#">  
 <img class="media-object" src="{{ product.image\_url }}" alt="{{ product.name }}" height=500>  
 </a>  
 </div>  
 <div class="media-body">  
 <div class="jumbotron">  
 <h2>{{ product.seller }}</h2>  
 </div>  
 <div class="jumbotron">  
 <p>{{ product.description }}</p>  
 </div>  
 <p><a class="btn btn-lg btn-success" role="button">Buy</a></p>  
 </div>  
 </div>  
  
  
{% endblock %}

**product-list.html**

{% extends 'base\_template.html' %}  
  
{% block title %} {% endblock %}  
  
{% block navbar-links %} {% endblock %}  
  
{% block body %}  
  
 <h1>Products</h1>  
 <ul>  
 {% for product **in** object\_list %}  
  
 <div class="jumbotron">  
 <div class="media-left media-middle">  
 <a href="#">  
 <img class="media-object" src="{{ product.image\_url }}" alt="{{ product.name }}" width=200>  
 </a>  
 </div>  
 <div class="media-body">  
 <h2>{{ product.name }}</h2>  
 <p><span class="label label-default">{{ product.seller }}</span></p>  
 <p>{{ product.description }}</p>  
 <p><a class="btn btn-primary btn-lg" href="product/{{ product.id }}" role="button">Learn more</a></p>  
 </div>  
 </div>  
  
 {% endfor %}  
 </ul>  
  
{% endblock %}

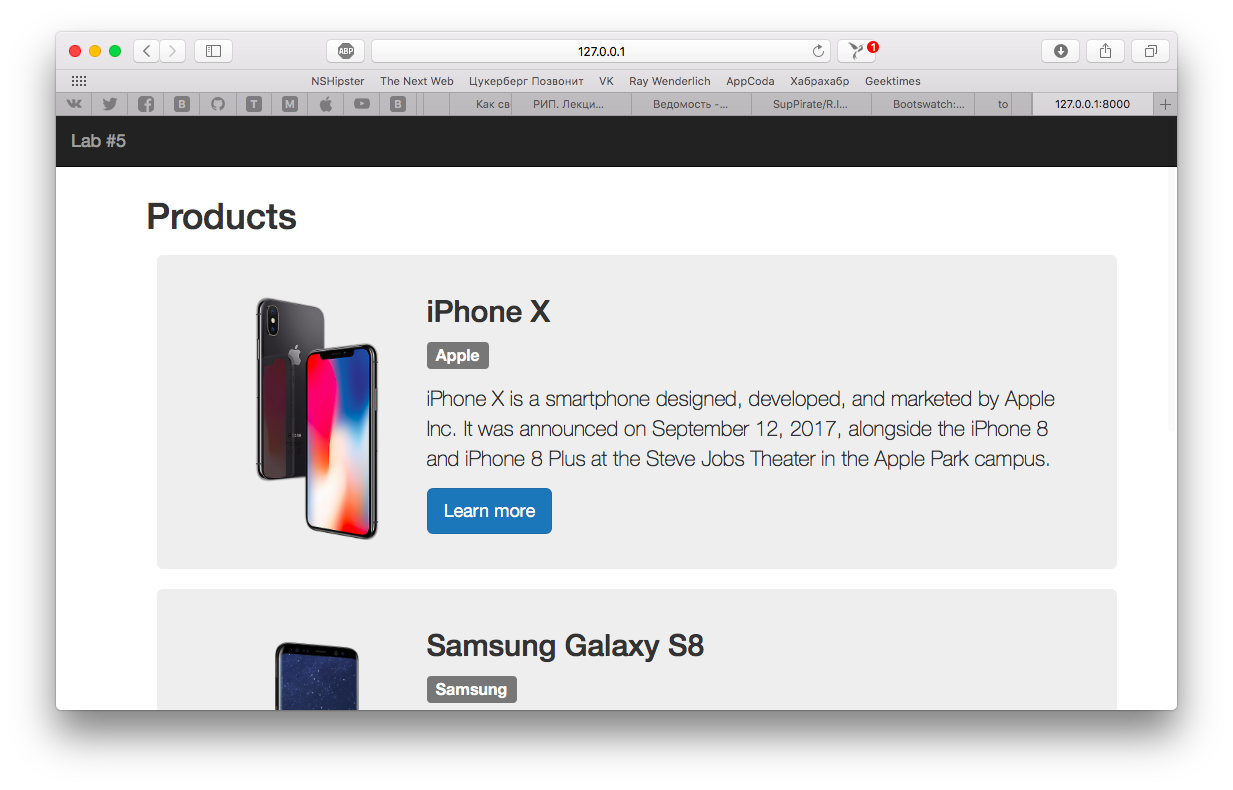
**views.py**

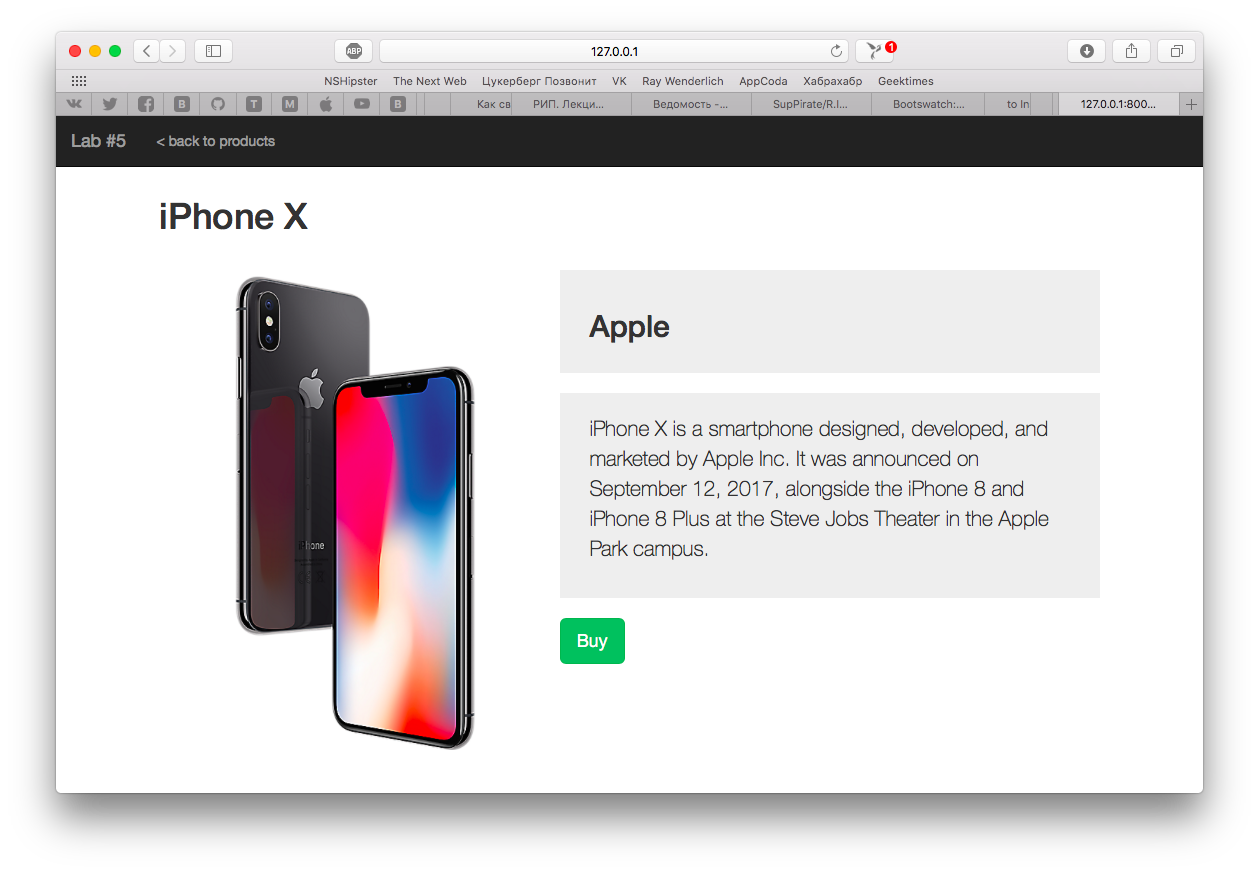
# -\*- coding: utf-8 -\*-  
from \_\_future\_\_ import unicode\_literals  
  
from django.shortcuts import render  
  
from django.http import HttpResponse  
from django.views.generic import View, ListView  
  
from lab\_app.models import Product  
  
  
def function\_view(request):  
 return HttpResponse('response from function view')  
  
  
class ListProductView(ListView):  
  
 model = Product  
 template\_name = 'product\_list.html'  
  
  
class ProductView(View):  
 def get(self, request, product\_id):  
 product = Product.objects.get(id=product\_id)  
 return render(request, 'product.html', {"product": product})

**models.py**

# -\*- coding: utf-8 -\*-  
from \_\_future\_\_ import unicode\_literals  
  
from django.db import models  
  
  
# Create your models here.  
  
class Product(models.Model):  
 name = models.CharField(  
 max\_length=255,  
 )  
 description = models.CharField(  
 max\_length=1000,  
 )  
 seller = models.CharField(  
 max\_length=255,  
 )  
 image\_url = models.URLField(  
 max\_length=255  
 )  
  
 def \_\_str\_\_(self):  
 return ' '.join([  
 self.name,  
 ' from ',  
 self.seller,  
 ])

**Скришоты выполнения**

****

****