

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

Отчет по лабораторной работе № 5 по курсу
«Разработка интернет-приложений»
«Шаблонизация»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53

Ореликов М.Г.

(подпись)

"__" _____ 2016 г.

Задание

В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.

- Создать проект
- Реализовать view, в которых генерируются html-страницы
- В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
- Все шаблоны должны расширять базовый шаблон
- Для элементов списка использовать тег `include`
- По нажатии на элемент списка должна открываться страница информации об элементе
- Для верстки необходимо использовать Bootstrap

Исходный код

urls.py

```
from django.conf.urls import url, include

urlpatterns = [
    url(r'^$', include('testApp.views')),
```

views.py

```
from django.shortcuts import render
from django.views import View

class OrdersView(View):
    orders = [
        {'title': 'title1', 'id': 1, 'text': 'text1'},
        {'title': 'title2', 'id': 2, 'text': 'text2'},
        {'title': 'title3', 'id': 3, 'text': 'text3'},
        {'title': 'title4', 'id': 4, 'text': 'text4'},
        {'title': 'title5', 'id': 5, 'text': 'text5'},
        {'title': 'title6', 'id': 6, 'text': 'text6'},
        {'title': 'title7', 'id': 7, 'text': 'text7'},
        {'title': 'title8', 'id': 8, 'text': 'text8'},
        {'title': 'title9', 'id': 9, 'text': 'text9'},
        {'title': 'title10', 'id': 10, 'text': 'text10'},
        {'title': 'title11', 'id': 11, 'text': 'text12'},
    ]
    def get(self, request):
        data = {
            'orders': OrdersView.orders,
            'count' : len(OrdersView.orders),
        }
        return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id,
                'title': OrdersView.orders[int(id)-1].get('title'),
                'text': OrdersView.orders[int(id)-1].get('text'),
            }
        }
        return render(request, 'order.html', data)
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta http-equiv="pragma" content="no-cache">
  {% load static %}
  <link href="{% static 'css/bootstrap.css' %}" rel="stylesheet">
  <link href="{% static 'css/bootstrap.css.map' %}" rel="stylesheet">
  <link href="{% static 'css/bootstrap-theme.css.map' %}" rel="stylesheet">
  <link href="{% static 'css/bootstrap-theme.css' %}" rel="stylesheet">
  <title>{% block title %}{% endblock %}</title>
</head>
<body>
  <nav class="navbar navbar-inverse navbar-static-top">
    <a href="{% url 'main' %}" class="navbar-brand">Фрукты</a>
  </nav>

  {% block body %}Переопределяется в наследниках{% endblock %}
  <hr>
  <footer>
    <p>&copy; BMSTU Orelikov Mike 2016</p>
  </footer>
</body>
</html>
```

orders.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %} Titletitle {% endblock %}
{% block body %}

  <div class="container">
    <br>
    <div class = "row">

      {% for order in orders %}
        <div class="col-lg-4">
          <div class="jumbotron">
            <h2>
              <a href="{% url 'order_url' order.id %}">{% include
"order_element.html" %}</a>
              <p><a class="btn" href="{% url 'order_url' id=order.id
%}">Узнать подробнее >></a></p>
              <button type="button" class="btn btn-default navbar-
btn"><a href="{% url 'order_url' order.id %}">Войти</a></button>
            </h2>
          </div>
        </div>
      {% empty %}
        <div>Пустой список</div>
      {% endfor %}

    </div>
    <br>
  </div>
  {% if count > 2 %}
    Большой список
  {% else %}
```

```
        Список короткий
    {% endif %}

{% endblock %}
```

order_element.html

```
<h1> {{ order.title }} </h1>
```

Результаты выполнения

