



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ (ИУ)

КАФЕДРА _____ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)

О Т Ч Е Т

по лабораторной работе № 4

по дисциплине: Разработка интернет-приложений

на тему: Шаблонизация

Студент ИУ5-53
(Группа)

(Подпись, дата) А.С. Волков
(И.О.Фамилия)

Руководитель

(Подпись, дата) Ю.Е. Гапанюк
(И.О.Фамилия)

2019 г.

1. Задание и порядок выполнения

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

2. Исходные коды

2.1. base.html

```
{% load static %}
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
integrity="sha384-Vkoo8x4CGs03+Hhvx8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
    <link rel="stylesheet" href="{% static 'css/Page1.css' %}">
    <title>{% block title %}{% endblock %}</title>
</head>
<body>
<div>
    <nav class="navbar navbar-expand-md navbar-dark bg-dark mb-4">
    <a class="navbar-brand" href="{% url 'main_url' %}">Магазин</a>
    <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="{% url 'orders_url' %}">Заказы</a>
            </li>
        </ul>
    </div>
</nav>
</div>

<div class="text">
    <h1>{% block title2 %}{% endblock %}<br></h1>
    {% block body %} Меняется {% endblock %}
</div>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-
J6qa4849b1E2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
```

```

integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj003uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
</body>
</html>

```

2.2. orders.html

```

{% extends 'base.html' %}

{% block title %}Заказы{% endblock %}
{% block title2 %}Заказы{% endblock %}

{% block body %}
<ul>
    {% for order in orders %}
    <li><a href="{% url 'order_url' order.id %}">{{ order.title }}</a></li>
    {% empty %}
    <li>
        Нет заказов
    </li>
    {% endfor %}
</ul>
{% endblock %}

```

2.3. order.html

```

{% extends 'base.html' %}

{% block title %}Заказ № {{ order.id }}{% endblock %}
{% block title2 %}Заказ № {{ order.id }}{% endblock %}
{% block body %}
    {% for orderitem in order_dict %}
        {% if orderitem.id == order.id %}
            <li>{{ orderitem.item1 }}</li>
            <li>{{ orderitem.item2 }}</li>
            <li>{{ orderitem.item3 }}</li>
            <li>{{ orderitem.item4 }}</li>
        {% endif %}
    {% endfor %}
{% endblock %}

```

2.4. views.py

```

from django.shortcuts import render
from django.views.generic.base import View

def index(request):
    return render(request, 'index.html')

class OrdersView(View):
    def get(self, request):
        data = {
            'orders':

```

```

        [
            {'title': 'Первый заказ', 'id': '1'},
            {'title': 'Второй заказ', 'id': '2'},
            {'title': 'Третий заказ', 'id': '3'},
        ]
    }
    return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            },
            'order_dict':
                [
                    {'id': '1', 'item1': 'Молоко', 'item2': 'Хлеб', 'item3':
'Кетчуп', 'item4': 'Мясо'},
                    {'id': '2', 'item1': 'Стрепсилс', 'item2': 'Глицин', 'item3':
'СН50Н', 'item4': 'Пластырь'},
                    {'id': '3', 'item1': 'Трансформер', 'item2': 'Лего', 'item3':
'Световой меч', 'item4': 'Пластинин'},
                ]
        }

        return render(request, 'order.html', data)

```

2.5. urls.py

```

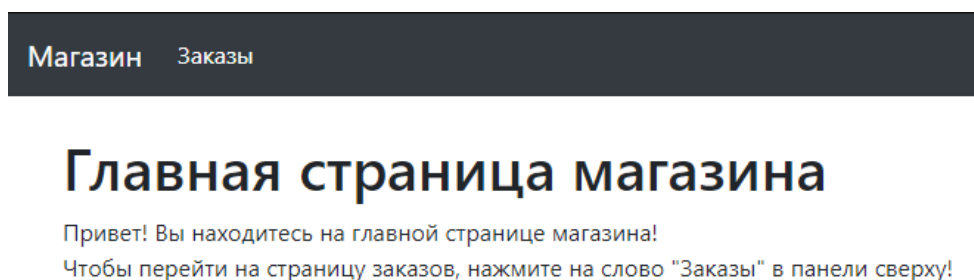
from django.contrib import admin
from django.urls import path
from Page1.views import OrderView, OrdersView, index
from django.conf.urls import url

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', index, name='main_url'),
    url(r'order/(?P<id>\d+)', OrderView.as_view(), name='order_url'),
    url(r'orders/', OrdersView.as_view(), name='orders_url'),
]

```

3. Скриншоты

3.1. Главная страница (index.html)



3.2. Страница со списком заказов (orders.html)

Магазин Заказы

Заказы

- [Первый заказ](#)
- [Второй заказ](#)
- [Третий заказ](#)

3.3. Страницы заказов (order.html)

Магазин Заказы

Заказ № 1

- Молоко
- Хлеб
- Кетчуп
- Мясо

Магазин Заказы

Заказ № 2

- Стрепсилс
- Глицин
- C5H5OH
- Пластырь

Магазин Заказы

Заказ № 3

- Трансформер
- Лего
- Световой меч
- Пластилин