**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка Интернет-Приложений»

Отчет по лабораторной работе №4

«Создание прототипа веб-приложения с использованием фреймворка Django»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-53 |  | Юрий Евгеньевич Гапанюк |
| Миронова Александра |  |  |
| Подпись и дата:  20.12.21 |  | Подпись и дата: |

Москва, 2021 г.

**Задание:**

1. Создайте прототип веб-приложения с использованием фреймворка Django:

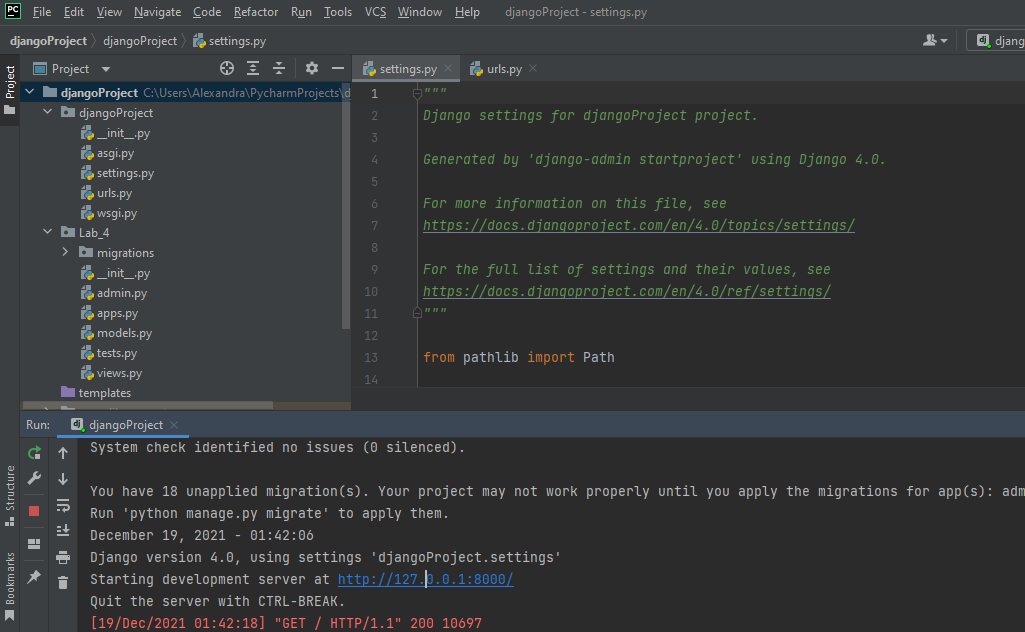
* Создайте виртуальное окружение.
* Установите в него Django.
* Создайте проект и приложение Django.

1. Создайте представления и шаблоны (по желанию можно использовать модели), реализующие концепцию master/detail со следующей функциональностью:

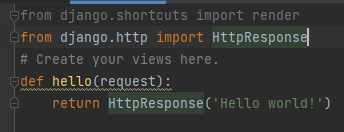
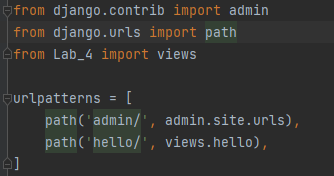
* На странице master в виде списка HTML выводится информация о трех объектах (например, о трех сортах мороженого). Каждая строка списка представляет собой гиперссылку, при нажатии на которую происходит переход к странице detail.
* Страница detail содержит детальное описание объекта (сорта мороженого), фотографию, гиперссылку на master-страницу.
* Фотография относится к статическому содержимому сайта.
* Страница detail должна выводить данные с использованием таблицы HTML.
* Шаблон страницы detail получает от представления данные о детальном объекте с использованием контекста.
* НЕОБЯЗАТЕЛЬНЫЙ ПУНКТ. По желанию можно использовать верстку с применением Bootstrap (или аналогичного фреймворка), а также представления на основе классов (class-based views).

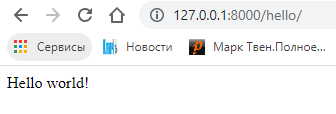
**Подготовка к выполнению программы:**

В программе PyCharm был создан проект Lab4

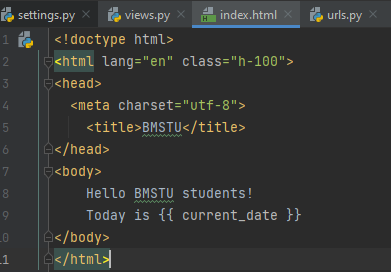
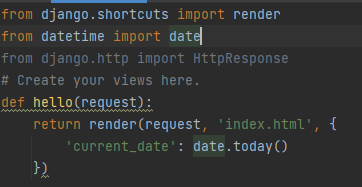


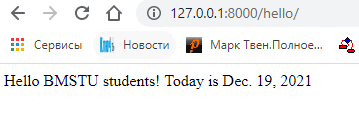
Опробована работа с View и Urls



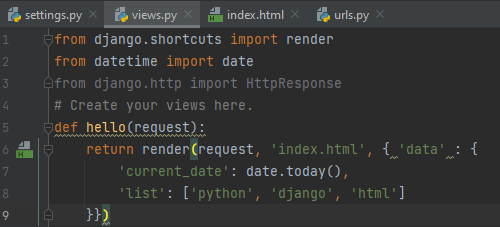
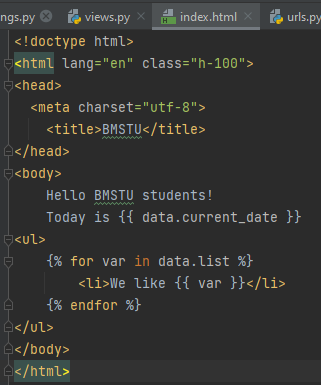


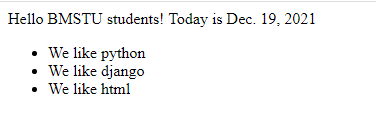
Было опробовано создание шаблона с переменными





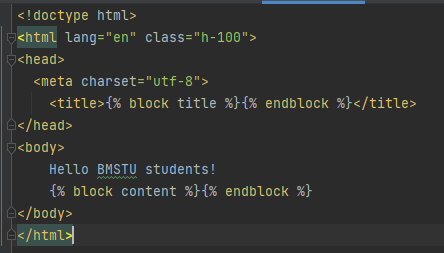
Было опробовано создание шаблона с тэгами

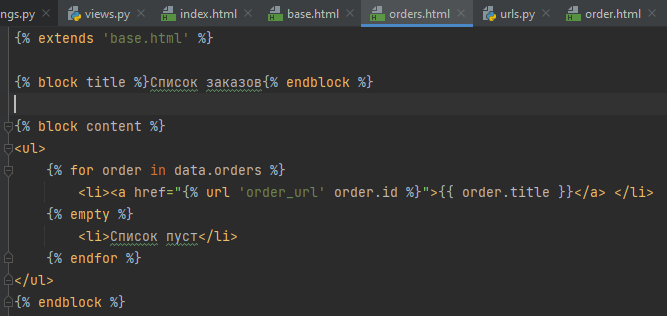


Было опробовано создание наследования шаблона

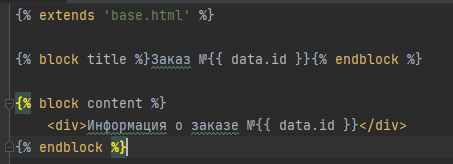
Для этого были созданы файл родитель base.html



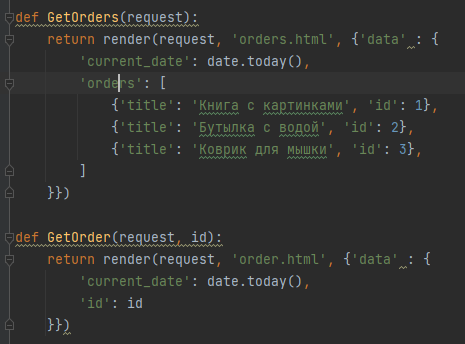
И дочерние файлы orders.html



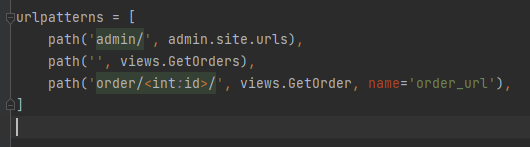
order.html



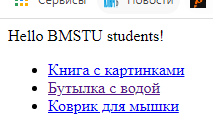
В файл Views.py были добавлены методы



И изменен urls.py

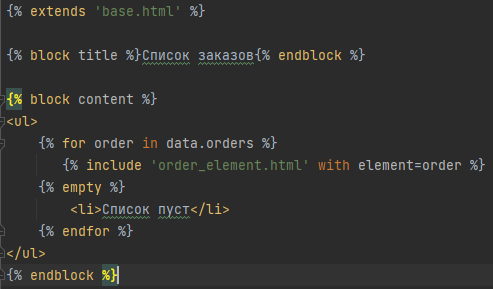


Результат:



Использование тега Include в шаблонизации

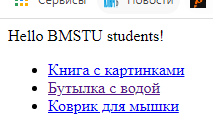
Для этого был изменен файл orders.html



И добавлен файл order\_element.html



Результат:



**Выполнение задания лабораторной работы**

**Текст программы:**

**views.py**

from django.shortcuts import render

def GetIcecreams(request):

return render(request, 'master.html', {'data': {'icecreams':

[

{'title': 'Пломбир', 'id': 1},

{'title': 'Шоколадное мороженное', 'id': 2},

{'title': 'Клубничное мороженное', 'id': 3},

]

}})

def GetIcecream(request, id):

return render(request, 'detail.html', {'data': {

'id': id,

'icecream\_info':

[

{'title': 'Пломбир',

'ingridients': 'Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Ванилин',

'standart': 'ГОСТ 119-41',

'price': '72 р.',

'id': 1},

{'title': 'Шоколадное мороженное',

'ingridients': 'Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Какао',

'standart': 'ГОСТ 31457-2012',

'price': '68 р.',

'id': 2},

{'title': 'Клубничное мороженное',

'ingridients': 'Сливки 33%, Молоко, Яичный желток, Сахарная пудра, Клубника',

'standart': 'ГОСТ 31457-2012',

'price': '62 р.',

'id': 3},

]

}})

**urls.py**

from django.contrib import admin

from django.urls import path

from Lab\_4 import views

urlpatterns = [

path('admin/', admin.site.urls),

path('', views.GetIcecreams, name='home\_url'),

path('details/<int:id>', views.GetIcecream, name='icecream\_url'),

]

**base.html**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>{% block title %}{% endblock %}</title>

</head>

<body>

{% block content %}{% endblock %}

</body>

</html>

**master.html**

{% extends 'base.html' %}

{% block title %}{% endblock %}

{% block content %}

<ul>

<h3>Виды мороженного</h3>

{% for icecream in data.icecreams%}

<li><a href="{% url 'icecream\_url' icecream.id %}">{{ icecream.title }}</a> </li>

{% empty %}

<li>Список пуст</li>

{% endfor %}

</ul>

{% endblock %}

**detail.html**

{% extends 'base.html' %}

{% load static %}

{% block title %}Вид: {% endblock %}

{% block content %}

<ul>

<div style="text-align: center">

{% if data.id == 1 %}

<img src="{% static 'images/plombir.jpg' %}" style="width:300px">

{% elif data.id == 2 %}

<img src="{% static 'images/chockolate.jpg' %}" style="width:300px">

{% else %}

<img src="{% static 'images/strawberry.jpg' %}" style="width:300px">

{% endif %}

</div>

<table border="1" width="100%" cellpadding="5">

<tr>

<th> Название</th>

<th> Ингридиенты </th>

<th> Стандарт </th>

<th> Цена </th>

</tr>

<tr>

{% for icecream in data.icecream\_info %}

{% if icecream.id == data.id %}

<td>{{ icecream.title }}</td>

<td>{{ icecream.ingridients }}</td>

<td> {{ icecream.standart }}</td>

<td>{{ icecream.price }}</td>

{% endif %}

{% endfor %}

</tr>

</table>

<div style="text-align: center">

<li><a href="{% url 'home\_url' %}">{{ 'Home' }}</a> </li>

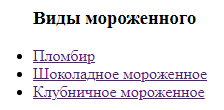
</div>

</ul>

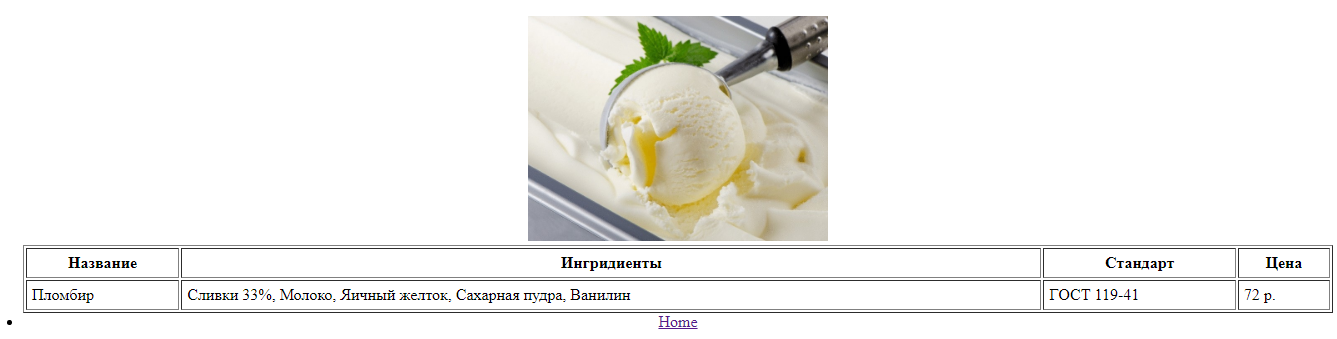
{% endblock %}

**Экранные формы с примерами выполнения программы:**

1) Страница с видами мороженного, содержащая гиперссылки на страницы с подробным описанием каждого



2) Страница с подробным описанием мороженного Пломбир



3) Страница с подробным описанием Шоколадного мороженного



4) Страница с подробным описанием Клубничного мороженного

