

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

Выполнил:  
студент группы  
ИУ5-53Б

Рыбина Арина

Дата: 28.12.2021

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.
2. 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).

### urls.py

```
from django.contrib import admin
from django.urls import path

from computers import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('master/', views.master),
    path('master/detail-<str:name>/', views.get_inf, name='detail_url'),
]
```

### apps.py

```
from django.apps import AppConfig

class BmstuLabConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'computers'
```

### models.py

```
from typing import Optional

from dataclasses import dataclass


@dataclass
class PCModel:
    photo: str
    name: str
    screen: float
    CPU: str
    memory: str
    price: float


@dataclass
class PCMark:
    name: str
    models: list[PCModel]
```

```
class Database:
    _PC_MARKS = {
        'LENOVO': PCMark(
            name='LENOVO',
            models=[
                PCModel(
                    photo='LENOVO/1.png',
                    name='IdeaPad 314ITL6',
                    screen=14,
                    CPU='Intel Core i3',
                    memory='SSD 512 ГБ',
                    price=46_015,
                ),
                PCModel(
                    photo='LENOVO/2.png',
                    name='V15-ADA',
                    screen=15.6,
                    CPU='AMD Ryzen 3 3250U',
                    memory='SSD 256 ГБ',
                    price=31_561,
                ),
                PCModel(
                    photo='LENOVO/3.png',
                    name='ThinkBook 15 G2-ARE',
                    screen=15.6,
                    CPU='AMD Ryzen 3 4300U',
                    memory='SSD 256 ГБ',
                    price=44_800,
                ),
            ]
        )
    }
```

views.py

```
from dataclasses import asdict

from django.shortcuts import render

from .models import Database


def master(request):
    names = Database.get_names()
    return render(request, 'index.html', {
        'data': {
            'names': names
        }
    })


def get_inf(request, name: str):
    mark = Database.get_mark_by_name(name)
    return render(
        request,
        'details.html',
        {
            'data': asdict(mark),
        },
    )
```

index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Computers</title>
</head>
<body>
  <h5># для этой страницы я не придумала тему..</h5>
  <h1>Here are some computer marks:</h1>
  <ul>
    {% for var in data.names %}
      <li>
        <a href="detail-{{ var }}"> {{ var }}</a>
      </li>
    {% endfor %}
  </ul>
</body>
</html>

```

base.html

```

<!doctype html>
<html lang="en" class="h-100">
<head>
  <meta charset="utf-8">
  {% load static %}
  <link rel="stylesheet" type="text/css" href="{% static 'style.css' %}">
  <title>{% block title %}{% endblock %}</title>
</head>
<body>
  <table class="table_style">
    {% block content %}{% endblock %}
  </table>
  <a href="javascript:history.back()" title="Вернуться на предыдущую страницу"> Back </a>
</body>
</html>

```

detail.html

```

{% extends 'base.html' %}

{% block title %}{{ data.name }}{% endblock %}

{% block content %}

    <tbody>
    <tr>
        <th>Model</th>
        <th>Screen</th>
        <th>CPU</th>
        <th>Memory</th>
        <th>Price</th>
        <th>Photo</th>
    </tr>
    {% for model in data.models %}
        <tr>
            {% include 'block.html' with elem=model %}
        </tr>
    {% endfor %}

    </tbody>

{% endblock %}

```

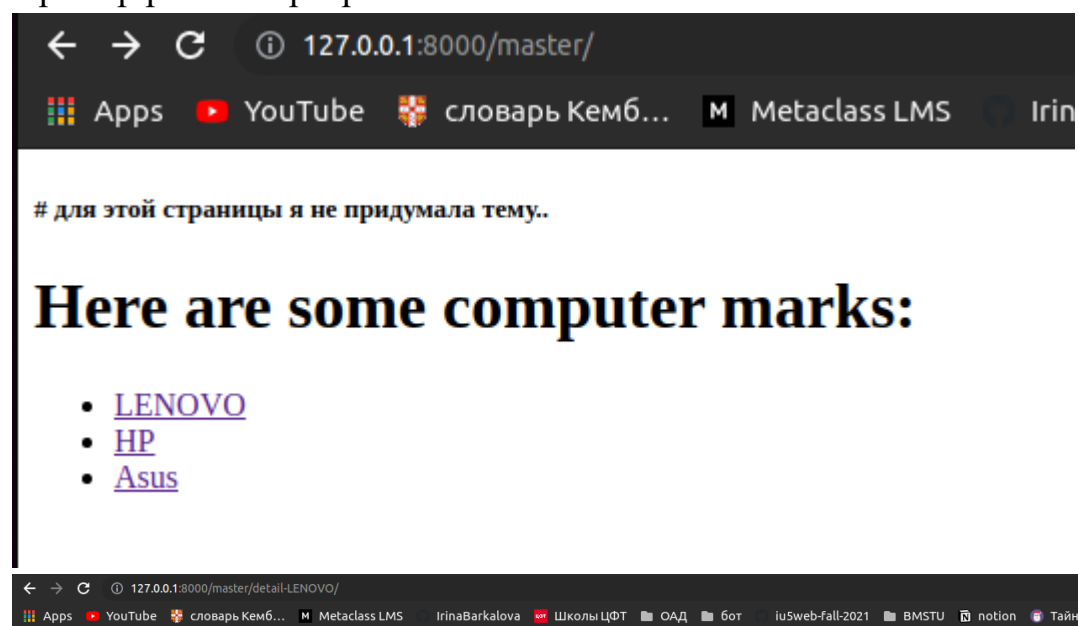
block.html


```

<td>{{ elem.name }}</td>
<td>{{ elem.screen }}</td>
<td>{{ elem.CPU }}</td>
<td>{{ elem.memory }}</td>
<td>{{ elem.price }}</td>
<td>
    {% load static %}
    <img width=120 height=100 src={% static elem.photo %} alt='' />
</td>

```

## Пример работы программы



Model	Screen	CPU	Memory	Price	Photo
IdeaPad 314ITL6	14	Intel Core i3	SSD 512 ГБ	46015	
V15-ADA	15.6	AMD Ryzen 3 3250U	SSD 256 ГБ	31561	
ThinkBook 15 G2-ARE	15.6	AMD Ryzen 3 4300U	SSD 256 ГБ	44800	