

**Московский государственный технический университет
им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

**Отчёт по рубежному контролю №2
по курсу «Разработка интернет-приложений»**

Выполнил:

студент группы ИУ5-51Б
Гришин С. В.

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

Подпись и дата:

11.12.2021

Подпись и дата:

11.12.2021

Москва, 2021 г.

Описание задания:

1. Создайте проект Python Django с использованием стандартных средств Django.
2. Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
3. С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
4. Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

№ варианта	Класс 1	Класс 2
11	Программа	Компьютер

Ход выполнения работы:

Создание моделей

models.py:

```
from django.db import models

class Computer(models.Model):
    computer_name = models.CharField(max_length=50, verbose_name="Название компьютера")
    os = models.CharField(max_length=15, verbose_name="Операционная система")
    price = models.DecimalField(max_digits=8, decimal_places=0, verbose_name="Цена компьютера")
    date_modified = models.DateTimeField(auto_now=True, verbose_name="Когда последний раз обновлялась информация о компьютере")

class Program(models.Model):
    program_name = models.CharField(max_length=50, verbose_name="Название программы")
    computer_id = models.DecimalField(max_digits=10, decimal_places=0, verbose_name="ID компьютера")
    memory = models.DecimalField(max_digits=8, decimal_places=0, verbose_name="Занимаемый объем в ГБ")
    date_modified = models.DateTimeField(auto_now=True, verbose_name="Когда последний раз обновлялась информация о программе")
```

Создание сериализаторов

serializers.py:

```
from programs.models import Computer
from programs.models import Program
from rest_framework import serializers

class ComputerSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Computer
        # Поля, которые мы сериализуем
```

```

        fields = ["pk", "computer_name", "os", "price", "date_modified"]

class ProgramSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Program
        # Поля, которые мы сериализуем
        fields = ["pk", "program_name", "computer_id", "memory",
"date_modified"]

```

View

view.py:

```

from rest_framework import viewsets
from programs.serializers import ProgramSerializer
from programs.serializers import ComputerSerializer
from programs.models import Program
from programs.models import Computer
from django.shortcuts import render

class ComputerViewSet(viewsets.ModelViewSet):
    queryset = Computer.objects.all().order_by('date_modified')
    serializer_class = ComputerSerializer # Сериализатор для модели
class ProgramViewSet(viewsets.ModelViewSet):
    queryset = Program.objects.all().order_by('date_modified')
    serializer_class = ProgramSerializer # Сериализатор для модели
def ComputerList(request):
    return render(request, 'computers.html', {'data': {
        'computers': Computer.objects.all(),
    }})
def ProgramList(request):
    return render(request, 'programs.html', {'data': {
        'programs': Program.objects.all(),
    }})

def GetComputer(request, id):
    return render(request, 'computer.html', {'data': {
        'computer': Computer.objects.filter(id=id)[0]
    }})
def GetProgram(request, id):
    return render(request, 'program.html', {'data': {
        'program': Program.objects.filter(id=id)[0],
        'computers': Computer.objects.all(),
    }})

```

URL

urls.py

```

from django.contrib import admin
from programs import views
from django.urls import include, path
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'computers', views.ComputerViewSet)
router.register(r'programs', views.ProgramViewSet)

urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
    path('admin/', admin.site.urls),
    path('rk/comp', views.ComputerList),

```

```

path('rk/prg', views.ProgramList),
path('rk/comp/<int:id>/', views.GetComputer, name='computer_url'),
path('rk/prg/<int:id>/', views.GetProgram, name='program_url')
]

```

Добавление данных с помощью Insomnia

lab6 - My Collection - lab6
Application Edit View Window Tools Help

Dashboard / My Collection

No Environment Cookies

GET http://127.0.0.1:8000/computers/ Send 200 OK 13.8 ms 545 B Just Now

Filter

GET lab6

JSON Auth Query Header Docs

```

1 {
2   "computer_name": "Lenovo",
3   "os": "Windows",
4   "price": 40000
5 }

```

Preview

```

1 [
2   {
3     "pk": 1,
4     "computer_name": "Asus",
5     "os": "Windows",
6     "price": 45000,
7     "date_modified": "2021-12-11T13:42:07.473061Z"
8   },
9   {
10    "pk": 2,
11    "computer_name": "Acer",
12    "os": "Windows",
13    "price": 55000,
14    "date_modified": "2021-12-11T13:42:24.624768Z"
15  },
16  {
17    "pk": 3,
18    "computer_name": "Apple",
19    "os": "MacOS",
20    "price": 80000,
21    "date_modified": "2021-12-11T13:43:30.308448Z"
22  },
23  {
24    "pk": 4,
25    "computer_name": "Dell",
26    "os": "Linux",
27    "price": 30000,
28    "date_modified": "2021-12-11T13:43:54.122430Z"
29  },
30  {
31    "pk": 5,
32    "computer_name": "Lenovo",
33    "os": "Windows",
34    "price": 40000,
35    "date_modified": "2021-12-11T13:44:23.711080Z"
36  }
37 ]

```

Beautify JSON

\$.store.books[*].author

lab6 - My Collection - lab6
Application Edit View Window Tools Help

Dashboard / My Collection

No Environment Cookies

GET http://127.0.0.1:8000/programs/ Send 200 OK 10.1 ms 1076 B Just Now

Filter

GET lab6

JSON Auth Query Header Docs

```

1 {
2   "program_name": "MATLAB",
3   "computer_id": 2,
4   "memory": 6
5 }

```

Preview

```

1 [
2   {
3     "pk": 1,
4     "program_name": "Visual Studio",
5     "computer_id": "1",
6     "memory": "30",
7     "date_modified": "2021-12-11T13:46:34.941706Z"
8   },
9   {
10    "pk": 2,
11    "program_name": "Visual Studio",
12    "computer_id": "3",
13    "memory": "30",
14    "date_modified": "2021-12-11T13:47:06.008680Z"
15  },
16  {
17    "pk": 3,
18    "program_name": "Qlick Sense",
19    "computer_id": "2",
20    "memory": "3",
21    "date_modified": "2021-12-11T13:47:28.794907Z"
22  },
23  {
24    "pk": 4,
25    "program_name": "SQL Server Management Studio",
26    "computer_id": "1",
27    "memory": "1",
28    "date_modified": "2021-12-11T13:47:55.588334Z"
29  },
30  {
31    "pk": 5,
32    "program_name": "Microsoft Office 2010",
33    "computer_id": "1",
34    "memory": "9",
35    "date_modified": "2021-12-11T13:48:13.715912Z"
36  },
37  {
38    "pk": 6,
39    "program_name": "Adobe Acrobat Reader",
40    "computer_id": "4"
41  }
42 ]

```

Beautify JSON

\$.store.books[*].author

base.html

```

<!doctype html>
<html lang="en" class="h-100">
<head>
  <meta charset="utf-8">
  <title>{% block title %}{% endblock %}</title>
</head>

```

```
<body style="background-image: url(https://phonoteka.org/uploads/posts/2021-03/1616520072_8-p-foni-dlya-prezentatsii-krasivie-strogie-8.jpg);
background-repeat: no-repeat">
    {% block content %}{% endblock %}
</body>
</html>
```

computers.html

```
{% extends 'base.html' %}

{% block title %}Рубежный контроль{% endblock %}

{% block content %}
    <h2>Список компьютеров:</h2>
    <h3>
        <ul>
            {% for computer in data.computers %}
                <li><a href="{% url 'computer_url' computer.id
%}">{{ computer.computer_name }}</a></li>
            {% empty %}
                <li>Список пуст</li>
            {% endfor %}
        </ul>
    </h3>
{% endblock %}
```

computer.html

```
{% extends 'base.html' %}

{% block title %}{{ data.computer.computer_name }}{% endblock %}

{% block content %}
    <div>Название: {{ data.computer.computer_name }}</div>
    <div>Операционная система: {{ data.computer.os }} </div>
    <div>Стоимость: {{ data.computer.price }} руб.</div>
{% endblock %}
```

programs.html

```
{% extends 'base.html' %}

{% block title %}Рубежный контроль{% endblock %}

{% block content %}
    <h2>Список программ:</h2>
    <h3><ul>
        {% for program in data.programs %}
            <li><a href="{% url 'program_url' program.id
%}">{{ program.program_name }}</a></li>
        {% empty %}
            <li>Список пуст</li>
        {% endfor %}
    </ul>
</h3>
{% endblock %}
```

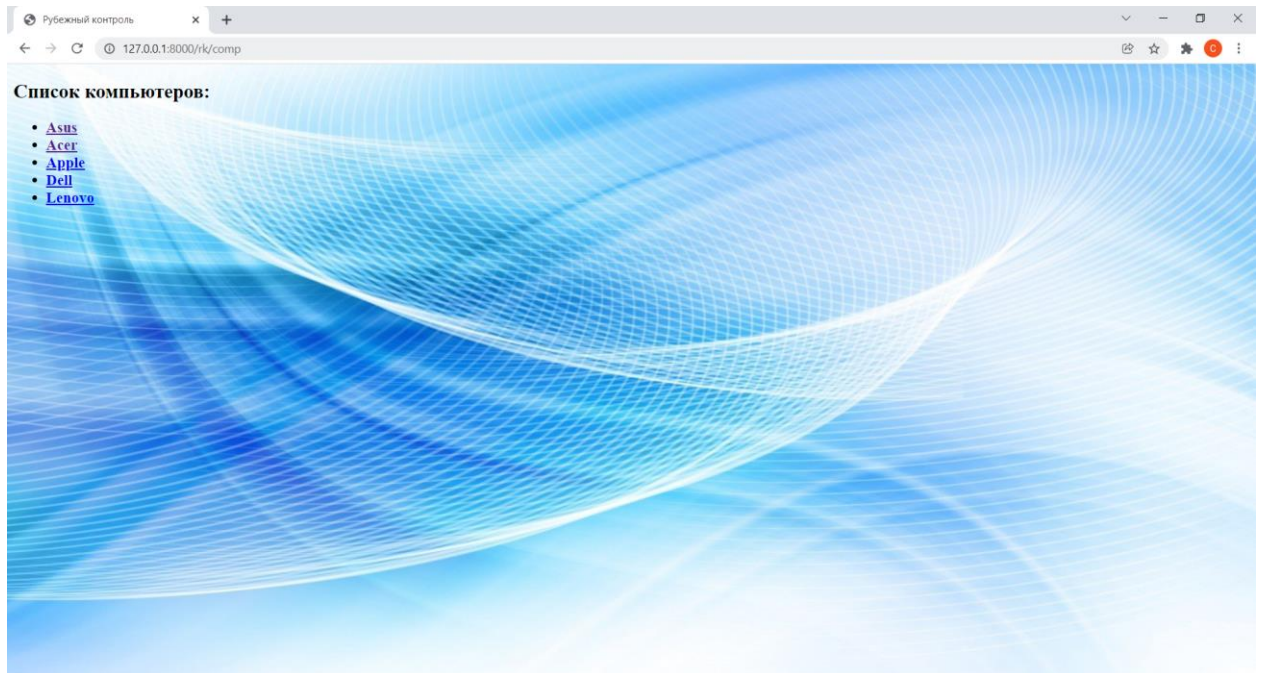
program.html

```
{% extends 'base.html' %}
```

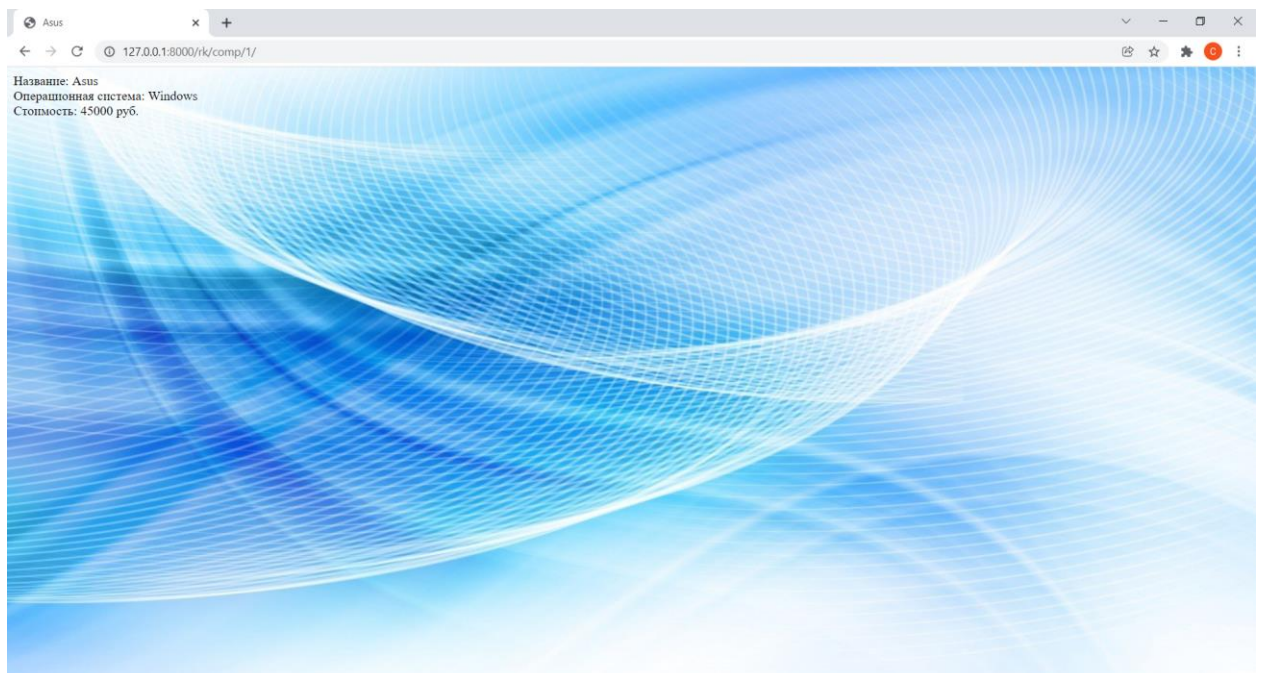
```
{% block title %}{{ data.program.program_name }}{% endblock %}

{% block content %}
<div>Название: {{ data.program.program_name }}</div>
  {% for computer in data.computers %}
    {% if data.program.computer_id == computer.id %}
      <div>Компьютер: {{ computer.computer_name }}</div>
    {% endif %}
  {% endfor %}
  <div>Занимаемое место: {{ data.program.memory }} ГБ</div>
{% endblock %}
```

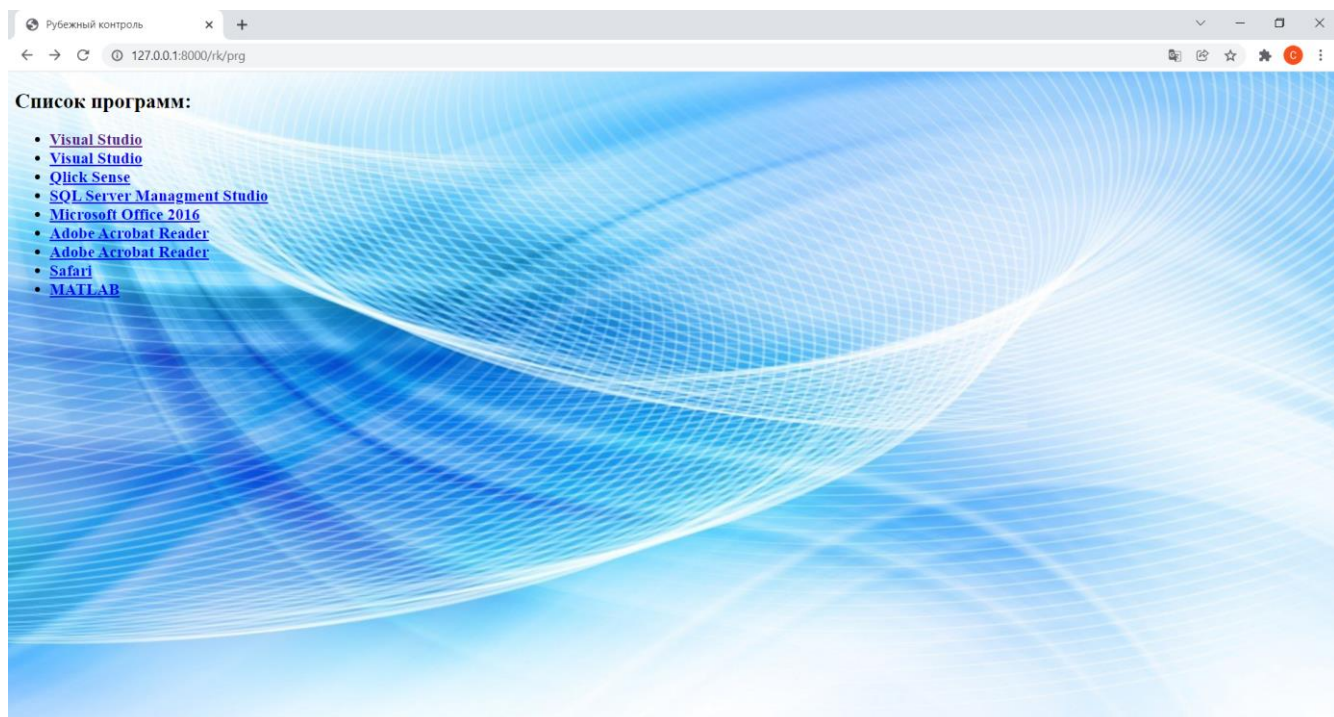
Список компьютеров



Информация о компьютере



Список программ



Информация о программе (соединение двух таблиц)

