

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

Факультет «Информатика и системы управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю № 2

Выполнил:
студент группы ИУ5-52

Веревкина Д.В.

Подпись и дата:

Проверил:
преподаватель каф.
ИУ5

Гапанюк Ю.Е.

Подпись и дата:

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.

Ход работы

models.py

```
from django.db import models
class
Microproc(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 микропроцессора")

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?")
```

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.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(),
    }})

```

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-include('rest_framework.urls',
auth/namespace='rest_framework')),
        , admin.site.urls),
    path('admin/', admin.site.urls),
path('rk/comp', views.ComputerList),
        , views.ProgramList),
    path('rk/prg', views.ProgramList),
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')

```

proc.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 %}

```

compproc.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 %}
```

main.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 %}
```