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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю № 2

Вариант Г7

|  |  |
| --- | --- |
| Выполнил: | Проверил: |
| студент группы ИУ5-54 | преподаватель каф. ИУ5 |
| Каримов А.Т. | Гапанюк Ю.Е. |
| Подпись и дата: | 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.

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| Г7 | Микропроцессор | Компьютер |

**Ход выполнения работы:**

**Создание моделей**

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?") |

**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 = [   |  | | --- | | api-auth/ |   path('', include(router.urls)), path('', include('rest\_framework.urls',  namespace='rest\_framework')), | | | | | |
| path(' path(' | |  | | --- | | admin/ | | ', admin.site.urls), | | | |
| rk/comp | | ', views.ComputerList), | | |
| path(' path(' path(' ] | |  | | --- | | rk/prg | | ', views.ProgramList), | | | |
| |  | | --- | | rk/comp/<int:id>/ | | | | | ', views.GetComputer, name='computer\_url'),  , views.GetProgram, name='program\_url') |
| |  | | --- | | rk/prg/<int:id>/ | | | | ' |
|  | | |

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 %} |