

**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 Г19

Выполнила:
студент группы ИУ5-54

Светашева Ю.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.

| № варианта | Класс 1 | Класс 2 |
|------------|---------|---------------|
| Г19 | Деталь | Производитель |

Ход выполнения работы:

Создание моделей

models.py:

```
from django.db import models

class Detal(models.Model):
    detal_name = models.CharField(max_length=50, verbose_name="Название детали")
    detal_price = models.DecimalField(max_digits=8, decimal_places=2, verbose_name="Цена детали")
    detal_type = models.CharField(max_length=50, verbose_name="Тип детали")

class Proizvoditel(models.Model):
    proizvoditel_name = models.CharField(max_length=50, verbose_name="Имя производителя")
```

Создание сериализаторов

serializers.py:

```
from rest_framework import serializers

from stocks.models import Detal, Proizvoditel

class DetalSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Detal
        # Поля, которые мы сериализуем
        fields = ["pk", "detal_name", "detal_price", "detal_type"]

class ProizvoditelSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = Proizvoditel
        # Поля, которые мы сериализуем
        fields = ["pk", "proizvoditel_name"]
```

View

view.py:

```
from rest_framework import viewsets

from stocks.models import Proizvoditel, Detal
from stocks.serializers import ProizvoditelSerializer, DetalSerializer

class ProizvoditelViewSet(viewsets.ModelViewSet):
    queryset = Proizvoditel.objects.all().order_by('date_modified')
    serializer_class = ProizvoditelSerializer # Сериализатор для модели

class DetalViewSet(viewsets.ModelViewSet):
    queryset = Detal.objects.all().order_by('date_modified')
    serializer_class = DetalSerializer # Сериализатор для модели

def ProizvoditelList(request):
    return render(request, 'proizvoditel.html', {'data': {
        'proizvoditel': Proizvoditel.objects.all(),
    }})

def DetalList(request):
    return render(request, 'detal.html', {'data': {
        'detal': Detal.objects.all(),
    }})

def GetProizvoditel(request, id):
    return render(request, 'proizvoditel.html', {'data': {
        'proizvoditel': Proizvoditel.objects.filter(id=id)[0]
    }})
```

```
def GetDetal(request, id):
    return render(request, 'detal.html', {'data': {
        'detal': Detal.objects.filter(id=id)[0],
        'proizvoditel': Proizvoditel.objects.all(),
    }})
```

URL

urls.py

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

router = routers.DefaultRouter()
router.register(r'proizvoditels', views.ProizvoditelViewSet)
router.register(r'detals', views.DetalViewSet)

urlpatterns = [
    path('', include(router.urls)),
    path('', api-auth/namespace='rest_framework'),
    path('', admin.site.urls),
    path('admin/', include(admin.site.urls)),
    path('rk/proizvoditel', views.ProizvoditelList),
    path('rk/detal', views.DetalList),
    path('rk/proizvoditel/<int:id>', views.GerProizvoditel,
         name='proizvoditel_url'),
    path('rk/detal/<int:id>', views.GetDetal,
         name='detal_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://mir-s3-
cdn-
f.behance.net/project_modules/fs/359d6223647503.5
63266a8b1f39.png); background-repeat: no-
repeat">
    {% block content %}{% endblock %}
</body>
</html>
```

proizvoditel.html

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

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

{% block content %}
    <h2>Список производителей:</h2>
    <h3>
```

```

        <ul>
        {% for proizvoditel in data.proizvoditels %}
            <li><a href="{% url 'proizvoditel_url' proizvoditel.id
%}">{{proizvoditel.proizvoditel_name}}</a></li>
        {% empty %}
            <li>Список пуст</li>
        {% endfor %}
        </ul>

    </h3>
{% endblock %}

```

proizvoditel.html

```

{% extends 'base.html' %}

{% block title %}{{ data.proizvoditel.proizvoditel_name }}{% endblock %}

{% block content %}
    <div>ФИО: {{ data.proizvoditel.proizvoditel_name }}</div>
{% endblock %}

```

detal.html

```

{% extends 'base.html' %}

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

{% block content %}
    <h2>Список деталей:</h2>
<h3><ul>
    {% for detal in data.detals %}
        <li><a href="{% url 'detal_url' detal.id
%}">{{detal.detal_name}}</a></li>
    {% empty %}
        <li>Список пуст</li>
    {% endfor %}
</ul>
</h3> {% endblock %}

```

main.html

```

{% extends 'base.html' %}

{% block title %}{{ data.proizvoditel.proizvoditel_name }}{%
endblock %}
{% block content %}
    <div>Название: {{ data.detal.detal_name }}</div>
    {% for proizvoditel in data.proizvoditel %}
        {% if data.detals.detal_id == proizvoditel.id %}
            <div>Производитель: {{ proizvoditel.proizvoditel_name }}</div>
        {% endif %}
    {% endfor %}
    <div>Деталь: {{ data.detal_name }}</div> {% endblock %}

```