Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический» Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю №2

Вариант Е19

Выполнил:

студент группы РТ5-51Б Сысоев Александр

доцент каф. ИУ5 Гапанюк Ю. Е.

Проверил:

ПОЛУЧЕННОЕ ЗАДАНИЕ

Рубежный контроль представляет собой разработку веб-приложения с использованием фреймворка Django. Веб-приложение должно выполнять следующие функции:

- 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.

исходный код

Файл models.py:

```
from django.db import models

class Manufacturers (models.Model):
    name_man = models.CharField('Имя производителя', max_length=200)

class Details (models.Model):
    name_det = models.CharField('Наименование детали', max_length=200)
    cost_det = models.IntegerField('Цена детали', default=0)
    mans = models.ForeignKey(Manufacturers, models.DO NOTHING)
```

Файл views.py:

```
from django.shortcuts import render
from rest_framework import viewsets
from Factory.models import Manufacturers, Details
from Factory.serializers import ManSerializer, DetSerializer

def index(request):
    return render(request, 'index.html')

class ManViewSet(viewsets.ModelViewSet):
    queryset = Manufacturers.objects.all().order_by("name_man")
    serializer_class = ManSerializer
```

```
class DetViewSet(viewsets.ModelViewSet):
    queryset = Details.objects.all().order_by("name_det")
    serializer_class = DetSerializer

def report(request):
    return render(request, 'report.html', {'data': {'details':
    Details.objects.select related('mans')}})
```

Файл urls.py:

```
from django.contrib import admin
from django.urls import include, path
from Factory.views import ManViewSet, DetViewSet
import Factory.views
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'Manufacturer', ManViewSet)
router.register(r'Detail', DetViewSet)

urlpatterns = [
path('', include(router.urls)),
path('admin/', admin.site.urls),
path('report/', Factory.views.report),
path('api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
]
```

Файл Factory/urls.py:

```
from django.urls import path

from . import views

urlpatterns = [
path('', views.index),
path('Manufacturer/', views.manufacturers),
path('Detail/', views.details),
path('Manufacturer/<int:id>/', views.manufacturer),
path('Program/<int:id>/', views.detail),
]
```

Файл report.html:

```
{% block title %} Рубежный контроль №2. Сысоев Александр РТ5-51Б. Вариант E19 {% endblock %} {% block body %} 
<div>
<div>

<thead>

ID
Наименование детали

Наименование детали
Цена детали
Производитель

Производитель
```

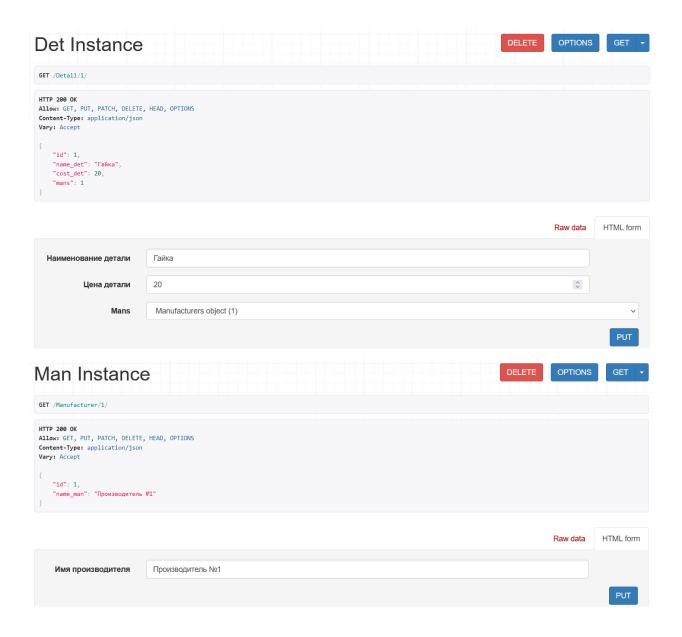
```
{% for det in data.details %}

{{ det.id }}
{{ det.name_det }}
{{ det.ost_det }}
{{ det.mans.name_man }}

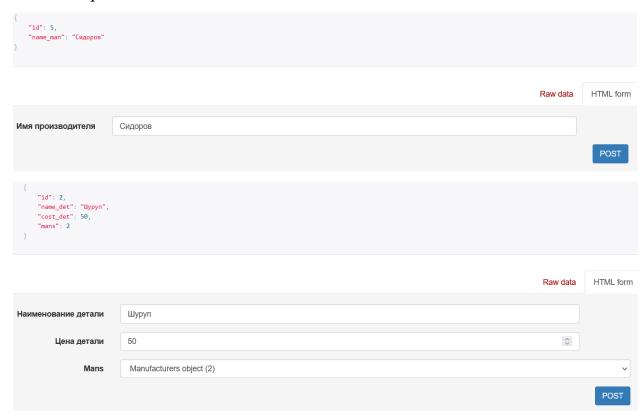
{% endfor %}
```

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

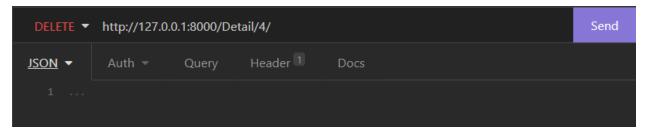
GET-запросы:



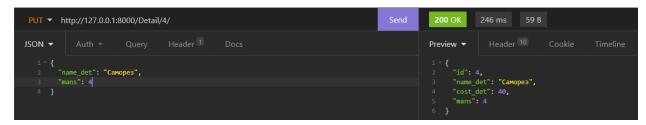
POST-запросы:



DELETE-запрос:



PUT-запрос:



Отчёт, соединяющий данные из 2-х таблиц:

Рубежный контроль №2. Сысоев Александр РТ5-51Б. Вариант Е19

ID	Наименование детали	Цена детали	Производитель
1	Гайка	20	Производитель №1
2	Шуруп	50	Производитель №2
3	Гвоздь	10	Иванов
4	Саморез	40	Петров
5	Штуцер	100	Сидоров
6	Болт	60	Производитель №1