

**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 Г14

Выполнил:
студент группы ИУ5-54

Лялин А.И.

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
Г14	Диски	Библиотека

Ход выполнения работы:

Создание моделей

models.py:

```
from django.db import models

class Disk(models.Model):
    disk_name = models.CharField(max_length=50, verbose_name="Название Диска")
    type = models.CharField(max_length=15, verbose_name="Вид")
    amount = models.DecimalField(max_digits=8, decimal_places=0,
    verbose_name="Количество дисков")

Lib(models.Model):
    lib_name = models.CharField(max_length=50, verbose_name="Название библиотеки")
    lib_id = models.DecimalField(max_digits=10, decimal_places=0,
    verbose_name="ID библиотеки")
```

Создание сериализаторов

serializers.py:

```
from lib.models import Lib
from disk.models import Disk
from rest_framework import serializers

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

```

        fields = ["pk", "id", "name"]
    class DiskSerializer(serializers.ModelSerializer):
class Meta:
    # Модель, которую мы сериализуем
model = Disk
    # Поля, которые мы сериализуем
    fields = ["pk", "disk_name", "type", "amount"]

```

View

view.py:

```

from rest_framework import viewsets
from disk.serializers import DiskSerializer
from disks.serializers import LibSerializer
from disks.models import Disk
from disks.models import Lib
from django.shortcuts import render

class LibViewSet(viewsets.ModelViewSet):
    queryset = Lib.objects.all().order_by('date_modified')
    serializer_class = LibSerializer # Сериализатор для модели
class DiskViewSet(viewsets.ModelViewSet):
    queryset = Disk.objects.all().order_by('date_modified')
    serializer_class = DiskSerializer # Сериализатор для модели
def LibList(request):
    return render(request, 'libs.html', {'data': {
        'libs': Lib.objects.all(),
    }})
def DiskList(request):
    return render(request, 'disks.html', {'data': {
        'disks': Disks.objects.all(),
    }})
def GetLib(request, id):
    return render(request, 'lib.html', {'data': {
'lib': Lib.objects.filter(id=id)[0]
    }})
def GetDisk(request, id):
    return render(request, 'disk.html', {'data': {
'disk': Disk.objects.filter(id=id)[0],
    'libs': Lib.objects.all(),    }})

```

URL

urls.py

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

router = routers.DefaultRouter()
router.register(r'libs', views.LibViewSet) router.register(r'disks',
views.DiskViewSet)

urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
    path('', admin.site.urls),
    path('admin/', include(admin.site.urls)),
    path('rk/lb', views.LibList),
    path('rk/dsk', views.DiskList),
    path('rk/lb/<int:id>', views.GetLib,
name='lib_url'),
    path('rk/dsk/<int:id>', views.GetDisk,
name='disk_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>
```

libs.html

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

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

{% block content %}
    <h2>Список библиотек:</h2>
    <h3>
        <ul>
            {% for lib in data.libs %}
                <li><a href="{% url 'lib_url' lib.id
%}">{{lib.lib_name}}</a></li>
            {% empty %}
                <li>Список пуст</li>
            {% endfor %}
        </ul>
    </h3>
</block>
```

```

        </ul>
    </h3>
{% endblock %}

```

lib.html

```

{% extends 'base.html' %}

{% block title %}{{ data.lib.lib_name }}{% endblock %}

{% block content %}
    <div>Название: {{ data.lib.lib_name }}</div>
{% endblock %}

```

diskproc.html

```

{% extends 'base.html' %}

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

{% block content %}
    <h2>Список дисков:</h2>
<h3><ul>
    {% for disk in data.disks %}
        <li><a href="{% url 'disk_url' disk.id
%}">{{disk.disk_name}}</a></li>
    {% empty %}
        <li>Список пуст</li>
    {% endfor %}
</ul>
</h3> {% endblock %}

```

main.html

```

{% extends 'base.html' %}

{% block title %}{{ data.disk.disk_name }}{% endblock %}
{% block content %}
    <div>Название: {{ data.disk.disk_name }}</div>
    {% for lib in data.lib%}
        {% if data.disk.lib_id == lib.id %}
            <div>Библиотека: {{ lib.lib_name }}</div>
        {% endif %}
    {% endfor %}
    {% endblock %}

```