

Отчет по РК № 2 по курсу
"Разработка Интернет-Приложений"

Выполнил:
Студент группы
ИУ5-55Б
Зубарева А. М.

Москва, МГТУ – 2021

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.

«Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список всех Музыкантов, у которых фамилия заканчивается на «ов», и названия их Оркестров.

«Музыкант» и «Оркестр» связаны соотношением один-ко-многим. Выведите список Оркестр со средней зарплатой Музыкантов в каждом Оркестре, отсортированный по средней зарплате.

№ варианта	Класс 1	Класс 2
5	Музыкант	Оркестр

Текст программы:

rk2/urls.py

```
from django.contrib import admin
from django.urls import path, include
from musicians import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name="main"),
    path('musician/', include([
        path('', views.read_musician, name='read_musician'),
        path('create/', views.create_musician, name="create_musician"),
        path('update/<int:musician_id>/', views.update_musician,
name="update_musician"),
```

```

        path('delete/<int:musician_id>/', views.delete_musician,
name="delete_musician"),
    ])),
    path('orchestra/', include([
        path('', views.read_orchestra, name='read_orchestra'),
        path('create/', views.create_orchestra, name="create_orchestra"),
        path('update/<int:orchestra_id>/',
            views.update_orchestra, name="update_orchestra"),
        path('delete/<int:orchestra_id>/',
            views.delete_orchestra, name="delete_orchestra"),
    ])),
    path('report/', views.report, name="report"),
]

```

musicians/models.py

```

from django.db import models

class Orchestra(models.Model):
    name = models.CharField(max_length=256, verbose_name="Orchestra name")

    def __str__(self):
        return self.name

class Musician (models.Model):
    name = models.CharField(max_length=256, verbose_name="Musician name")
    salary = models.PositiveIntegerField(verbose_name="Musician's salary")
    orchestra = models.ForeignKey(
        Orchestra,
        on_delete=models.SET_DEFAULT,
        null=True,
        default=None,
        related_name="musicians"
    )

    def __str__(self):
        return self.name

```

musicians/views.py

```

from django.db.models.aggregates import Avg
from django.http.response import HttpResponse
from django.shortcuts import get_list_or_404, get_object_or_404, redirect, render
from .models import Orchestra, Musician
from django.db import models

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

# musicians

```

```

def read_musician(request):
    musicians = Musician.objects.all()
    return render(request, 'musician/musician_list.html', {'musicians': musicians})

def create_musician(request):
    if request.method == 'GET':
        orchestras = Orchestra.objects.all()
        return render(request, 'musician/create_musician.html', {"orchestras": orchestras})
    else:
        dto = {}
        for key in request.POST:
            if key in Musician.__dict__:
                dto[key] = request.POST[key]
        dto['orchestra'] = get_object_or_404(
            Orchestra, pk=request.POST['orchestra'])
        new_musician = Musician(**dto)
        new_musician.save()
        return redirect('read_musician')

def update_musician(request, musician_id):
    if request.method == 'GET':
        orchestras = Orchestra.objects.all()
        musician = get_object_or_404(Musician, pk=musician_id)
        return render(request, 'musician/update_musician.html', {"musician": musician,
"orchestras": orchestras})
    else:
        musician = get_object_or_404(Musician, pk=musician_id)
        for key in request.POST:
            if key in musician.__dict__ and key != 'orchestra':
                setattr(musician, key, request.POST[key])
        if 'orchestra' in request.POST:
            setattr(musician, 'orchestra', get_object_or_404(
                Orchestra, pk=request.POST['orchestra']))
        musician.save()
        return redirect('read_musician')

def delete_musician(request, musician_id):
    musician = get_object_or_404(Musician, pk=musician_id)
    musician.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# orchestras

def read_orchestra(request):
    orchestras = Orchestra.objects.all()
    return render(request, 'orchestra/orchestra_list.html', {'orchestras': orchestras})

```

```

def create_orchestra(request):
    if request.method == 'GET':
        return render(request, 'orchestra/create_orchestra.html')
    else:
        new_orchestra = Orchestra(name=request.POST['name'])
        new_orchestra.save()
        return redirect('read_orchestra')

def update_orchestra(request, orchestra_id):
    if request.method == 'GET':
        orchestra = get_object_or_404(Orchestra, pk=orchestra_id)
        return render(request, 'orchestra/update_orchestra.html', {"orchestra": orchestra})
    else:
        orchestra = get_object_or_404(Orchestra, pk=orchestra_id)
        for key in request.POST:
            if key in orchestra.__dict__:
                setattr(orchestra, key, request.POST[key])
        orchestra.save()
        return redirect('read_orchestra')

def delete_orchestra(request, orchestra_id):
    orchestra = get_object_or_404(Orchestra, pk=orchestra_id)
    orchestra.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# REPORT

def report(request):
    # return HttpResponse('♥')
    found_musicians = Musician.objects.filter(name__endswith='ob')
    average_salaries = []
    for orchestra in Orchestra.objects.all():
        average_salaries.append({"orchestra": orchestra, "salary":
round(Musician.objects.filter(
            orchestra=orchestra.pk).aggregate(Avg('salary'))['salary__avg'] or 0)})
    return render(request, 'report.html', {"found_musicians": found_musicians,
"average_salaries": sorted(average_salaries, key=lambda salary: salary['salary'])})

```

templates/inc/base.html

```

{% load static %}

<!doctype html>
<html lang="en" class="h-100">

<head>
    <!-- ===== CHARSET AND VIEWPORT ===== -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- ===== CACHE TURN OFF ===== -->

```

```

<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="expires" content="0">
<!-- ===== COPY RIGHT ===== -->
<meta name="author" content="IU5-55 Vadim Khizhnyakov">
<title>
    IU5-55 Zubareva RK2
</title>

<!-- ===== BOOTSTRAP CORE CSS ===== -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
<!-- ===== SHARED STYLES ===== -->
<link rel="stylesheet" href="{% static 'css/navbar.css' %}">
</head>

<body class="d-flex flex-column h-100">
    <header>
        <!-- Navbar -->
        {% include 'inc/navbar.html' %}
    </header>

    <!-- Begin page content -->
    <main role="main" class="flex-shrink-0">
        <div class="container">
            <div class="row">
                <div class="col-md-12">
                    <!-- Content -->
                    {% block content %}
                    {% endblock content %}
                </div>
            </div>
        </div>
    </main>

    <footer class="footer mt-auto py-3">
        <!-- Footer -->
    </footer>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikTlWxgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</html>

```

templates/inc/navbar.html

```

{% load static %}
<nav class="navbar navbar-expand-sm navbar-dark bg-dark p-3">
    <ul class="navbar-nav nav-tabs ml-auto">
        <li class="nav-item">
            <a href="{% url 'main' %}" class="nav-link">Main Page</a>

```

```

</li>
<li class="nav-item">
  <a href="{% url 'read_musician' %}" class="nav-link">All musicians</a>
</li>
<li class="nav-item">
  <a href="{% url 'create_musician' %}" class="nav-link">Create musician</a>
</li>
<li class="nav-item">
  <a href="{% url 'read_orchestra' %}" class="nav-link">All orchestras</a>
</li>
<li class="nav-item">
  <a href="{% url 'create_orchestra' %}" class="nav-link"
    >Create orchestra</a>
  >
</li>
<li class="nav-item">
  <a href="{% url 'report' %}" class="nav-link">Report</a>
</li>
</ul>
</nav>

```

templates/musician/musician_card.html

```

{% load static %}

<div class="card mb-3">
  <div class="row g-0">
    <div class="col-md-4">
      
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title">{{ musician.name }}</h5>
        <p class="card-text">{{ musician.orchestra.name }}</p>
        <p class="card-text">
          This is a wider card with supporting text below as a natural lead-in to
          additional content. This
          content is a little bit longer.
        </p>
        <p class="card-text"><small class="text-muted">{{ musician.salary
        }}</small></p>
        <a href="{% url 'delete_musician' musician.pk %}" class="btn btn-danger">
          Delete </a>
        <a href="{% url 'update_musician' musician.pk %}" class="btn btn-primary">
          Update </a>
      </div>
    </div>
  </div>
</div>

```

templates/musician/musician_form.html

```

<div class="mb-3">
  <label for="musicianName" class="form-label">Musician Name</label>

```

```

    <input type="text" name="name" value="{{ musician.name }}" required="required"
class="form-control" id="musicianName" />
</div>

<div class="mb-3">
    <label for="musicianSalary" class="form-label">Musician Salary</label>
    <input
        type="number"
        min="0"
        name="salary"
        value="{{ musician.salary }}"
        required="required"
        class="form-control"
        id="musicianSalary"
    />
</div>

<div class="mb-3">
    <label for="musicianOrchestra" class="form-label">Select Orchestra</label>
    <select name="orchestra" class="form-select" aria-label="Default select example"
id="musicianOrchestra">
        {% for orchestra in orchestras %}
        <option value="{{ orchestra.pk }}">{{ orchestra.name }}</option>
        {% endfor %}
    </select>
</div>

```

templates/musician/musician_list.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All Musicians</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-column">
    {% for musician in musicians %} {% include 'musician/musician_card.html' with
musician=musician %} {% endfor %}
</div>
{% endblock content %}

```

templates/musician/create_musician.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New Musician</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_musician' %}" method="POST" class="form">
        {% csrf_token %} {% include 'musician/musician_form.html' with musician=musician %}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>

```



```

    </form>
</div>
{% endblock content %}

```

templates/musician/update_musician.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Musician</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_musician' musician.pk %}" method="POST" class="form">
        {% csrf_token %} {% include 'musician/musician_form.html' with musician=musician %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}

```

templates/orchestra/orchestra_card.html

templates/orchestra/orchestra_form.html

```

{% load static %}

<div class="card m-3" style="width: 18rem">
    
    <div class="card-body">
        <h5 class="card-title">{{ orchestra.name }}</h5>
        <p class="card-text">
            Some quick example text to build on the card title and make up the bulk of the
            card's content.
        </p>
        <a href="{% url 'delete_orchestra' orchestra.pk %}" class="btn btn-danger"> Delete
    </a>
        <a href="{% url 'update_orchestra' orchestra.pk %}" class="btn btn-primary"> Update
    </a>
    </div>
</div>

```

templates/orchestra/orchestra_list.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All Orchestras</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-wrap">

```

```
    {% for orchestra in orchestras %} {% include 'orchestra/orchestra_card.html' with
orchestra=orchestra %} {% endfor %}
</div>
{% endblock content %}
```

templates/orchestra/create_orchestra.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New Orchestra</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_orchestra' %}" method="POST" class="form">
        {% csrf_token %} {% include 'orchestra/orchestra_form.html' with orchestra=orchestra
%}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}
```

templates/orchestra/update_orchestra.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Orchestra</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_orchestra' orchestra.pk %}" method="POST" class="form">
        {% csrf_token %} {% include 'orchestra/orchestra_form.html' with orchestra=orchestra
%}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}
```

templates/report.html

```
{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Report Page</h1>
</div>
<!-- Page Content -->
<div>
    <h1>Musicians with name ends with 'ов'</h1>
    {% for musician in found_musicians %} {% include 'musician/musician_card.html' with
musician=musician %} {% endfor %}
</div>

<div>
```

```

<h1>Average salary</h1>
<div class="d-flex flex-wrap">
    {% for avg_salary in average_salaries %}
        <div class="d-flex flex-column align-center" >
            <h3 class="text-center">Average salary {{ avg_salary.salary }}</h3>
            {% include 'orchestra/orchestra_card.html' with orchestra=avg_salary.orchestra %}
        </div>
    {% endfor %}
</div>
{% endblock content %}

```

templates/index.html

```

{% extends 'inc/base.html' %}

{% load static %}

{% block content %}
<div class="container">
    <h1 class="text-center mt-4">Main Page</h1>
</div>
<!-- Page Content -->
<div>
    
</div>
{% endblock content %}

```

Результат работы:

Django administration

Home › Musicians › Orchestras

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

MUSICIANS

Musicians [+ Add](#)

Orchestras [+ Add](#)

Select orchestra to change

Action: 0 of 2 selected

☐ ORCHESTRA

☐ Paris

☐ Moscow

2 orchestras

Django administration

WELCOME **ROOT** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home › Musicians › Musicians

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

MUSICIANS

Musicians [+ Add](#)

Orchestras [+ Add](#)

Select musician to change

[ADD MUSICIAN](#) [+](#)

Action: 0 of 2 selected

☐ MUSICIAN

☐ Иванов

☐ Смыслов

2 musicians

Main Page



All Musicians



Смыслов Максим

Paris

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

100000

Delete

Update



Иванов

Moscow

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

1524

Delete

Update



Иванова

Moscow

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

1234

Delete

Update



Сидорова

Moscow

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

1345

Delete

Update

New Musician

Musician Name

Сидорова

Musician Salary

1345

Select Orchestra

Moscow

Create

Report Page

Musicians with name ends with 'ОВ'



Смыслов

100000

Delete Update



Иванов

1524

Delete Update

Average salary

Average salary 34193.0 Average salary 1524.0



Moscow

Delete Update



Paris

Delete Update

Report Page

Musicians with name ends with 'OB'



Иванов

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

1524

Delete Update

Average salary

Average salary 0

Average salary 1368

Average salary 100000



test

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete Update



Moscow

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete Update



Paris

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete Update