

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по РК №2

Выполнил:

студент группы ИУ5-54

Кравцов А.Н.

Подпись и дата:

Проверил:

Преподаватель каф. ИУ5

Гапанюк Ю.Е.

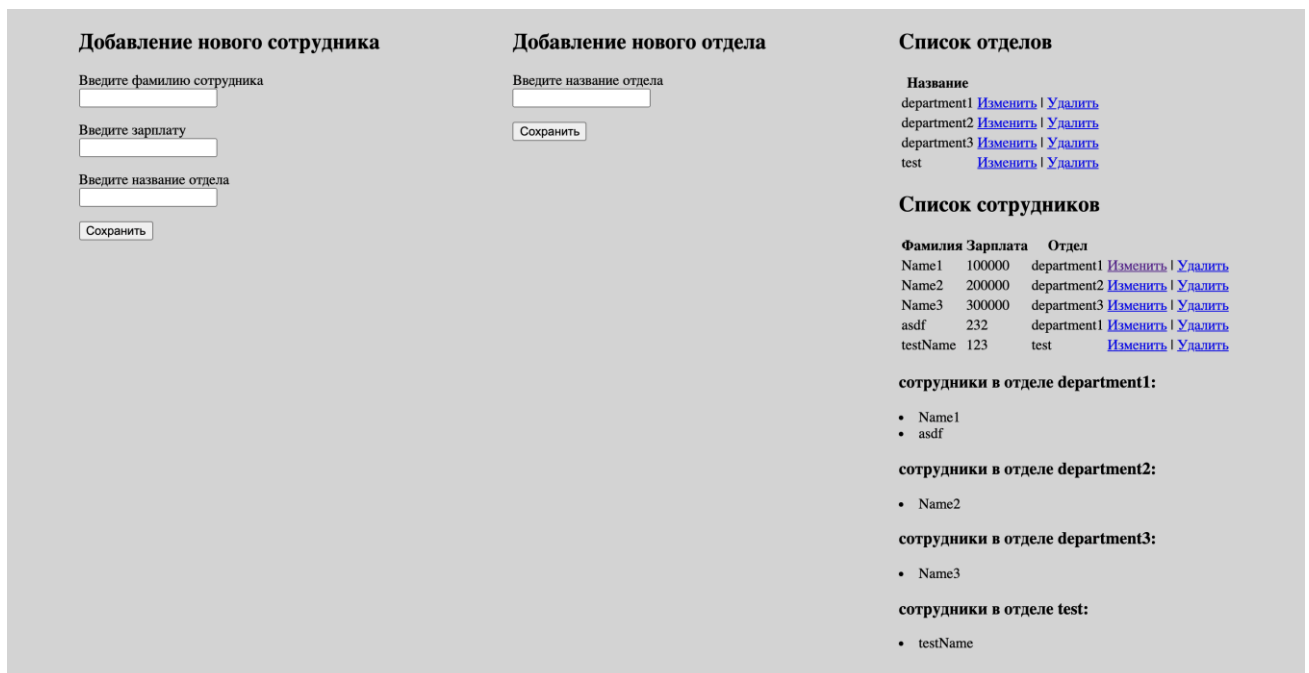
Подпись и дата:

Москва 2021 г.

## Задание.

Задание:

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.



**Добавление нового сотрудника**

Введите фамилию сотрудника

Введите зарплату

Введите название отдела

**Добавление нового отдела**

Введите название отдела

**Список отделов**

Название  
department1 [Изменить](#) | [Удалить](#)  
department2 [Изменить](#) | [Удалить](#)  
department3 [Изменить](#) | [Удалить](#)  
test [Изменить](#) | [Удалить](#)

**Список сотрудников**

Фамилия	Зарплата	Отдел	
Name1	100000	department1	<a href="#">Изменить</a>   <a href="#">Удалить</a>
Name2	200000	department2	<a href="#">Изменить</a>   <a href="#">Удалить</a>
Name3	300000	department3	<a href="#">Изменить</a>   <a href="#">Удалить</a>
asdf	232	department1	<a href="#">Изменить</a>   <a href="#">Удалить</a>
testName	123	test	<a href="#">Изменить</a>   <a href="#">Удалить</a>

**сотрудники в отделе department1:**

- Name1
- asdf

**сотрудники в отделе department2:**

- Name2

**сотрудники в отделе department3:**

- Name3

**сотрудники в отделе test:**

- testName

Листинг

Models.py

```
from django.db import models
```

```
class department(models.Model):  
    objects = models.Manager()
```

```
name = models.CharField(max_length=50, verbose_name="Название отдела")
```

```
class employee(models.Model):  
    objects = models.Manager()  
    name = models.CharField(max_length=50, verbose_name="Фамилия сотрудника")  
    salary = models.IntegerField(verbose_name="Зарплата")  
    departmentID = models.ForeignKey(department, on_delete=models.CASCADE, null=True)
```

## Views.py

```
from django.shortcuts import render  
from django.http import HttpResponseRedirect  
from django.http import HttpResponseNotFound  
from rk.models import employee  
from rk.models import department
```

```
def index(request):  
    departments = department.objects.all()  
    emps = employee.objects.all()  
    return render(request, "index.html", {"departments": departments, "emps": emps})
```

```
def create(request):  
    if request.method == "POST":  
        dl = department()  
        dl.name = request.POST.get("name_department")  
        dl.save()  
    return HttpResponseRedirect("/")
```

```
def create_emp(request):  
    if request.method == "POST":  
        d = employee()  
        d.name = request.POST.get("name")  
        d.salary = request.POST.get("salary")  
        d.departmentID = department.objects.get(name=request.POST.get("departmentID"))  
        d.save()  
    return HttpResponseRedirect("/")
```

```
def edit(request, id):  
    try:  
        emp = employee.objects.get(id=id)
```

```
    if request.method == "POST":  
        emp.name = request.POST.get("name")  
        emp.salary = request.POST.get("salary")
```

```

emp.departmentID = department.objects.get(name=request.POST.get("departmentID"))
emp.save()
return HttpResponseRedirect("/")
else:
return render(request, "edit.html", {"emp": emp})
except employee.DoesNotExist:
return HttpResponseRedirect("<h2>emp not found</h2>")

```

```

def edit_department(request, id):
try:
dep = department.objects.get(id=id)

```

```

if request.method == "POST":
dep.name = request.POST.get("name")
dep.save()
return HttpResponseRedirect("/")
else:
return render(request, "edit_department.html", {"department": dep})
except department.DoesNotExist:
return HttpResponseRedirect("<h2>departmentrary not found</h2>")

```

```

def delete(request, id):
try:
emp = employee.objects.get(id=id)
emp.delete()
return HttpResponseRedirect("/")
except employee.DoesNotExist:
return HttpResponseRedirect("<h2>emp not found</h2>")

```

```

def delete_department(request, id):
try:
dep = department.objects.get(id=id)
dep.delete()
return HttpResponseRedirect("/")
except employee.DoesNotExist:
return HttpResponseRedirect("<h2>departmentrary not found</h2>")

```

## Urls.py

```

from django.urls import path
from rk import views

```

```

urlpatterns = [
path("", views.index),
path('create/', views.create),

```

```

path('create_emp/', views.create_emp),
path('edit/<int:id>/', views.edit),
path('edit_department/<int:id>/', views.edit_department),
path('delete/<int:id>/', views.delete),
path('delete_department/<int:id>/', views.delete_department),
]

```

## Index.html

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>рк2</title>
</head>
<body class="container" style="display: flex; justify-content: space-around;">
<form method="POST" action="create_emp/">
{% csrf_token %}

<h2>Добавление нового сотрудника</h2>
<p>
<label>Введите фамилию сотрудника</label><br>
<input type="text" name="name" />
</p>
<p>
<label>Введите зарплату</label><br>
<input type="text" name="salary" />
</p>
<p>
<label>Введите название отдела</label><br>
<input type="text" name="departmentID" />
</p>
<input type="submit" value="Сохранить" >
</form>

<form method="POST" action="create/">
{% csrf_token %}

<h2>Добавление нового отдела</h2>
<p>
<label>Введите название отдела</label><br>
<input type="text" name="name_department" />
</p>
<input type="submit" value="Сохранить" >
</form>
<div>
<h2>Список отделов</h2>
<table>
<tr>
<th>Название</th>

```

```
{% for department in departments %}
<tr>
<td>{{ department.name }}</td>
<td><a href="edit_department/{{department.id}}">Изменить</a> | <a
href="delete_department/{{department.id}}">Удалить</a></td>
</tr>
{% endfor %}
</table>
```

```
{% if emps.count > 0 %}
<h2>Список сотрудников</h2>
<table>
<tr>
```

```
<th>Фамилия</th>
<th>Зарплата</th>
<th>Отдел</th>
</tr>
{% for emp in emps %}
<tr>
```

```
<td>{{ emp.name }}</td>
<td>{{ emp.sallary }}</td>
<td>{{ emp.departmentID.name }}</td>
<td><a href="edit/{{emp.id}}">Изменить</a> | <a href="delete/{{emp.id}}">Удалить</a></td>
</tr>
{% endfor %}
</table>
{% endif %}
```

```
{% for department in departments %}
<h3>сотрудники в отделе {{ department.name }}:</h3>
{% for emp in emps %}
{% if department.name == emp.departmentID.name %}
<li>{{emp.name}}</li>
{% endif %}
{% endfor %}
{% endfor %}
</div>
</body>
</html>
```

Edi\_emp.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>edit disk</title>
</head>
```

```
<body class="container">
<form method="POST">
{% csrf_token %}
<p>
<label>Введите фамилию сотрудника</label><br>
<input type="text" name="name" value="{{disk.name}}" />
</p>
<p>
<label>Введите зарплату</label><br>
<input type="text" name="salary" value="{{disk.salary}}" />
</p>
<p>
<label>Введите название отдела</label><br>
<input type="text" name="departmentID" value="{{disk.departmentID.name}}" />
</p>
<input type="submit" value="Сохранить" >
</form>
</body>
</html>
```

### Edit\_department.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>edit departments</title>
</head>
<body class="container">
<form method="POST">
{% csrf_token %}
<p>
<label>Введите название отдела</label><br>
<input type="text" name="name" value="{{departments.name}}" />
</p>
<input type="submit" value="Сохранить" >
</form>
</body>
</html>
```