

## Задание

Вариант	Предметная область	
	Класс 1	Класс 2
27	Преподаватель	Учебный курс

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

## settings.py

```
"""
Django settings for RK_2 project.

Generated by 'django-admin startproject' using Django 4.0.

For more information on this file, see
https://docs.djangoproject.com/en/4.0/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/4.0/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure--
!^k&ld=c*(np=p8fp#k^!jk$gd!oiulw3=fqn+nuqud5=1e='

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
```

```

'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'rest_framework',
'App.apps.AppConfig',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'RK_2_ls.urls'

TEMPLATES = [
{
    'BACKEND': 'django.template.backends.django.DjangoTemplates',
    'DIRS': [BASE_DIR / 'templates']
    ,
    'APP_DIRS': True,
    'OPTIONS': {
        'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
            'django.contrib.messages.context_processors.messages',
        ],
    },
},
]

WSGI_APPLICATION = 'RK_2_ls.wsgi.application'

# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
{
    'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',

```

```

    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'ru-ru'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

## models.py

```

from django.db import models

# Create your models here.
class Course(models.Model):
    name = models.CharField('Название курса', max_length=50)

class Teacher(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    firstName = models.CharField('Имя', max_length=50)
    lastName = models.CharField('Фамилия', max_length=50)
    otch = models.CharField('Отчество', max_length=50)

    def fio(self):
        return self.lastName + ' ' + self.firstName + ' ' + self.otch

    def __str__(self):
        return self.fio() + " | Курс: " + self.course.name

```

## serializers.py

```
from rest_framework import serializers
from .models import Course, Teacher

class CourseSerializer(serializers.ModelSerializer):
    class Meta:
        model = Course
        fields = ['id', 'name']

class TeacherSerializer(serializers.ModelSerializer):
    class Meta:
        model = Teacher
        fields = ['id', 'course', 'firstName', 'lastName', 'otch']
```

## views.py

```
from rest_framework import viewsets
from .serializers import CourseSerializer, TeacherSerializer
from .models import Course, Teacher
from django.shortcuts import render
from django.http import Http404

# Create your views here.
class CourseViewSet(viewsets.ModelViewSet):
    queryset = Course.objects.all()
    serializer_class = CourseSerializer

class TeacherViewSet(viewsets.ModelViewSet):
    queryset = Teacher.objects.all()
    serializer_class = TeacherSerializer

def CourseList(request):
    return render(request, 'course/list.html', {'c_list':
Course.objects.all()})

def TeacherList(request):
    return render(request, 'teacher/list.html', {'t_list':
Teacher.objects.all()})

def CourseView(request, course_id):
    try:
        c = Course.objects.get(id=course_id)
    except:
        raise Http404('Курс не найден!')
    return render(request, 'course/detail.html', {'course': c})

def TeacherView(request, teacher_id):
    try:
        t = Teacher.objects.get(id=teacher_id)
    except:
```

```

        raise Http404('Преподаватель не найден!')
    return render(request, 'teacher/detail.html', {'teacher': t})

```

## urls.py

"""RK\_2 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:  
<https://docs.djangoproject.com/en/4.0/topics/http/urls/>

Examples:

Function views

1. Add an import: `from my_app import views`
2. Add a URL to `urlpatterns`: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to `urlpatterns`: `path('', Home.as_view(), name='home')`

Including another `URLconf`

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to `urlpatterns`: `path('blog/', include('blog.urls'))`

```

"""
from django.contrib import admin
from django.urls import path, include
from rest_framework import routers
from App import views as app_views

router = routers.DefaultRouter()
router.register(r'courses', app_views.CourseViewSet)
router.register(r'teachers', app_views.TeacherViewSet)

# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('admin/', admin.site.urls),
    path('app/c_list', app_views.CourseList),
    path('app/t_list', app_views.TeacherList),
    path('app/c/<int:course_id>', app_views.CourseView, name='course_detail'),
    path('app/t/<int:teacher_id>', app_views.TeacherView,
name='teacher_detail')
]

```

## base.html

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}Рубежный контроль № 2{% endblock %}</title>
</head>
<body>

{% block content %}{% endblock %}

<hr>
<center>Шанаурина Е. Г. ИУ5Ц-74Б</center>

</body>
</html>

```

## course/list.html

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

{% block title %}РК №2{% endblock %}

{% block content %}

    <center><h1>Список курсов</h1></center><hr><br>

    {% if c_list %}
        {% for c in c_list %}
            <a href="{% url 'course_detail' c.id %}">{{ c.name }}</a><br>
        {% endfor %}
    {% else %}
        Список пуст
    {% endif %}

{% endblock %}
```

## course/detail.html

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

{% block title %}{{ course.name }}{% endblock %}

{% block content %}

    <center><h1>{{ course.name }}</h1></center><hr><br>

    <table border="1" cellpadding="5">
    <tr>
        <th>Номер курса</th>
        <th>{{ course.name }}</th>
    </tr>
    </table>
    <br><hr><br>
    <a href="/App/c_list">Назад</a>

{% endblock %}
```

## teacher/list.html

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

{% block title %}РК №2{% endblock %}

{% block content %}

    <center><h1>Список преподавателей</h1></center><hr><br>

    {% if t_list %}
        {% for t in t_list %}
            <a href="{% url 'teacher_detail' t.id %}">{{ t.fio }}</a><br>
        {% endfor %}
    {% else %}
        Список пуст
    {% endif %}

{% endblock %}
```

```
{% endblock %}
```

## teacher/detail.html

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

```
{% block title %}{{ group.number }}{% endblock %}
```

```
{% block content %}
```

```
    <center><h1>{{ teacher.fio }}</h1></center><hr><br>
```

```
    <table border="1" cellpadding="5">
```

```
    <tr>
```

```
        <th>Курс</th>
```

```
        <th>{{ teacher.course.name }}</th>
```

```
    </tr>
```

```
    <tr>
```

```
        <th>Фамилия</th>
```

```
        <th>{{ teacher.lastName }}</th>
```

```
    </tr>
```

```
    <tr>
```

```
        <th>Имя</th>
```

```
        <th>{{ teacher.firstName }}</th>
```

```
    </tr>
```

```
    <tr>
```

```
        <th>Отчество</th>
```

```
        <th>{{ teacher.otch }}</th>
```

```
    </tr>
```

```
    </table>
```

```
    <br><hr><br>
```

```
    <a href="/app/g_list">Назад</a>
```

```
{% endblock %}
```

## Добавление, изменение и удаление данных через API

### Добавление курса

POST ▾	http://127.0.0.1:8000/courses/	Send	201 Created	97.3 ms	24 B	Just Now ▾		
JSON ▾	Auth ▾	Query	Header 1	Docs	Preview ▾	Header 10	Cookie	Timeline
<pre>1 { 2   "name": "РИП" 3 }</pre>				<pre>1 { 2   "id": 1, 3   "name": "РИП" 4 }</pre>				

### Удаление курса

DELETE ▾	http://127.0.0.1:8000/courses/1	Send	204 No Content	97.9 ms	0 B	Just Now ▾		
Body ▾	Auth ▾	Query	Header	Docs	Preview ▾	Header 9	Cookie	Timeline
				No body returned for response				

### Список курсов

GET ▾	http://127.0.0.1:8000/courses/	Send	200 OK	8.23 ms	240 B	Just Now ▾		
Body ▾	Auth ▾	Query	Header	Docs	Preview ▾	Header 10	Cookie	Timeline
				<pre>1 [ 2   { 3     "id": 2, 4     "name": "Разработка интернет-приложений" 5   }, 6   { 7     "id": 3, 8     "name": "ИНО" 9   }, 10  { 11    "id": 4, 12    "name": "Русский язык" 13  }, 14  { 15    "id": 5, 16    "name": "Инженерная графика" 17  }, 18  { 19    "id": 6, 20    "name": "Информатика" 21  } 22 ]</pre>				

### Изменение курса

PUT ▾	http://127.0.0.1:8000/courses/3/	Send	200 OK	97.3 ms	47 B	Just Now ▾		
JSON ▾	Auth ▾	Query	Header 1	Docs	Preview ▾	Header 10	Cookie	Timeline
<pre>1 { 2   "name": "АНГЛИЙСКИЙ ЯЗЫК" 3 }</pre>				<pre>1 { 2   "id": 3, 3   "name": "АНГЛИЙСКИЙ ЯЗЫК" 4 }</pre>				



## Добавление преподавателя

POST http://127.0.0.1:8000/groups/ **Send** 201 Created 97.4 ms 43 B

JSON Auth Query Header 1 Docs Preview Header 10 Cookie Timeline

```
1 {
2   "course": 3,
3   "number": "Грынна"}

```

```
1 {
2   "id": 1,
3   "course": 3,
4   "number": "Грынна"
5 }
```

## Изменение преподавателя

PUT http://127.0.0.1:8000/teachers/2/ **Send** 200 OK 102 ms 94 B Just Now

JSON Auth Query Header 1 Docs Preview Header 10 Cookie Timeline

```
1 {
2   "course": 4,
3   "firstName": "Иванов",
4   "lastName": "Иван",
5   "otch": "Иванович"
6 }

```

```
1 {
2   "id": 2,
3   "course": 4,
4   "firstName": "Иванов",
5   "lastName": "Иван",
6   "otch": "Иванович"
7 }

```

## Удаление преподавателя

DELETE http://127.0.0.1:8000/teachers/2/ **Send** 204 No Content 82.3 ms 0 B Just Now

Body Auth Query Header Docs Preview Header 9 Cookie Timeline

No body returned for response

## Список преподавателей

GET http://127.0.0.1:8000/teachers/ **Send** 200 OK 8.87 ms 528 B Just Now

Body Auth Query Header Docs Preview Header 10 Cookie Timeline

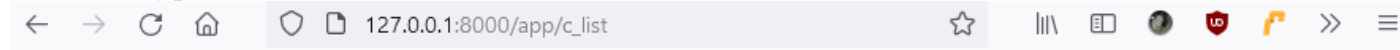
```
1 [
2   {
3     "id": 1,
4     "course": 2,
5     "firstName": "Юрий",
6     "lastName": "Гапанюк",
7     "otch": "Евгеньевич"
8   },
9   {
10    "id": 3,
11    "course": 5,
12    "firstName": "Иванов",
13    "lastName": "Иван",
14    "otch": "Иванович"
15  },
16  {
17    "id": 4,
18    "course": 6,
19    "firstName": "Костиков",
20    "lastName": "Сергей",
21    "otch": "Владимирович"
22  },
23  {
24    "id": 5,
25    "course": 3,
26    "firstName": "Тряскин",
27    "lastName": "Никита",
28    "otch": "Сергеевич"
29  },
30  {
31    "id": 6,
32    "course": 4,
33    "firstName": "Белединова",
34    "lastName": "Оксана",
35    "otch": "Александровна"
36  }
37 ]

```



Select a body type from above

## Список курсов



# Список курсов

---

[Разработка интернет-приложений](#)

[Английский язык](#)

[Русский язык](#)

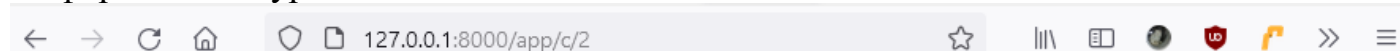
[Инженерная графика](#)

[Информатика](#)

---

Шанаурина Е. Г. ИУ5Ц-74Б

## Информация о курсе



# Разработка интернет-приложений

---

Название курса	Разработка интернет-приложений
----------------	--------------------------------

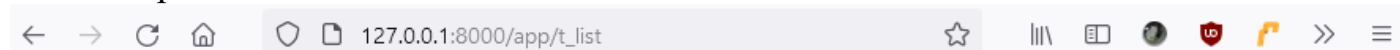
---

[Назад](#)

---

Шанаурина Е. Г. ИУ5Ц-74Б

## Список преподавателей



# Список преподавателей

---

[Гапанюк Юрий Евгеньевич](#)

[Иван Иванов Иванович](#)

[Сергей Костиков Владимирович](#)

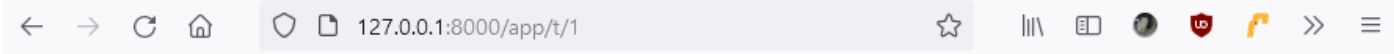
[Никита Тряскин Сергеевич](#)

[Оксана Белединова Александровна](#)

---

Шанаурина Е. Г. ИУ5Ц-74Б

Информация о преподавателе. Соединение двух таблиц.



## Гапанюк Юрий Евгеньевич

---

Курс	Разработка интернет-приложений
Фамилия	Гапанюк
Имя	Юрий
Отчество	Евгеньевич

---

[Назад](#)

---

Шанаурина Е. Г. ИУ5Ц-74Б