



**Министерство науки и высшего образования Российской
Федерации**
Федеральное государственное бюджетное образовательное учреждение высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Отчет по рубежному контролю №2
Курса «Разработка интернет-приложений»
Вариант 27
(Преподаватели, курсы)**

Студент: Яровенко М.В.

Группа: ИУ5Ц-72Б

Преподаватель: Гапанюк Ю.Е.

2021 г.

Оглавление

Задание	3
Текст программы	3
Результат выполнения программы	16

Задание

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

- Создайте проект Python Django с использованием стандартных средств Django.
- Создайте модель Django ORM, содержащую две сущности, связанные отношением один-ко-многим в соответствии с Вашим вариантом из условий рубежного контроля №1.
- С использованием стандартного механизма Django сгенерируйте по модели макет веб-приложения, позволяющий добавлять, редактировать и удалять данные.
- Создайте представление и шаблон, формирующий отчет, который содержит соединение данных из двух таблиц.

Текст программы

Файл models.py

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

```
class Teacher(models.Model):
```

```
    id = models.AutoField('id', primary_key=True)
```

```
    name = models.CharField(max_length=30)
```

```
    family = models.CharField(max_length=30)
```

```
    status = models.CharField(max_length=30)
```

```
    def __str__(self):
```

```
        return f'{self.name}'
```

```
class Meta:
```

```
    managed = False
```

```
    db_table = 'teacher'
```

```

class Kurse(models.Model):
    id = models.AutoField('id', primary_key=True)
    name = models.CharField(max_length=30)
    idt = models.ForeignKey('Teacher', models.DO_NOTHING)

class Meta:
    managed = False
    db_table = 'kurs'

```

Файл views.py

```

from django.shortcuts import render
from django.views.generic.edit import CreateView, UpdateView,
DeleteView
from django import forms
from .models import *
from datetime import datetime

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

def report(request):
    kurses = Kurse.objects.all()
    teachers = Teacher.objects.all()
    params = {'kurses': kurses, 'teachers': teachers, 'date':
datetime.now().strftime("%Y-%m-%d %H:%M:%S")}
    return render(request, 'report.html', params)

```

```
def teacher_list(request):  
    teacher = Teacher.objects.all().values()  
    params = {'entity': 'Teacher', 'objects': teacher}  
    return render(request, 'list.html', params)
```

```
def kurses_list(request):  
    kurses = Kurses.objects.all().values()  
    params = {'entity': 'Kurses', 'objects': kurses}  
    return render(request, 'list.html', params)
```

```
class TeacherCreate(CreateView):  
    model = Teacher  
    fields = ['name', 'family', 'status']  
    success_url = '/teachers'  
    template_name = 'teacher_form.html'
```

```
class TeacherUpdate(UpdateView):  
    model = Teacher  
    fields = ['name', 'family', 'status']  
    pk_url_kwarg = 'id'  
    success_url = '/teachers'  
    template_name = 'teacher_form.html'
```

```
class TeacherDelete>DeleteView):  
    model = Teacher
```

```
pk_url_kwarg = 'id'
success_url = '/teachers'
template_name = 'teacher_delete_form.html'
```

```
class KurseCreate(CreateView):
    model = Kurse
    fields = ['name']
    success_url = '/kurses'
    template_name = 'kurses_form.html'
```

```
class KurseUpdate(UpdateView):
    model = Kurse
    fields = ['name']
    pk_url_kwarg = 'id'
    success_url = '/kurses'
    template_name = 'kurses_form.html'
```

```
class KurseDelete>DeleteView):
    model = Kurse
    pk_url_kwarg = 'id'
    success_url = '/kurses'
    template_name = 'kurses_delete_form.html'
```

Файл urls.py

```
from django.urls import path
from . import views
```

```
urlpatterns = [
```

```

    path("", views.index),
    path('teachers', views.teacher_list),
    path('teacher/create', views.TeacherCreate.as_view()),
    path('teacher/<int:id>/update',          views.TeacherUpdate.as_view(),
name='teacher_update'),
    path('teacher/<int:id>/delete',          views.TeacherDelete.as_view(),
name='teacher_delete'),
    path('kurses', views.kurses_list),
    path('kurses/create', views.KursesCreate.as_view()),
    path('kurses/<int:id>/update',          views.KursesUpdate.as_view(),
name='kurses_update'),
    path('kurses/<int:id>/delete',          views.KursesDelete.as_view(),
name='kurses_delete'),
    path('report', views.report)
]

```

Файл stings.py

```

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-2psfze#(y@m3t2ka$s6chet3_p39ad*31f&-n8y1w@1)%2ymkp'

```

```

# 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',  
    'rk2_app.apps.Rk2AppConfig',  
]
```

```
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 = 'rk2.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        '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',  
            ],  
        },  
    },  
]
```



```
{
    '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 = 'rk2.wsgi.application'
```

```
# Database
```

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

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'rk2',
        'USER': 'dbuser',
        'PASSWORD': 'admin',
        'HOST': 'localhost',
        'PORT': 3306,
```

```

        'OPTIONS': {'charset': 'utf8'},
        'TEST_CHARSET': 'utf8',
    }
}

# 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 = 'en-us'

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'

Файл base.html

<!DOCTYPE html>

<html>

<head>

 <meta charset='utf-8'>

 <title>

 {% block title %} {% endblock %}

 </title>

 <style>

```

a, a:visited{
    color: black
}

p {
    font-size: 1.2rem;
    font-family: sans-serif;
}

</style>
</head>
<body>
<div>
    <a style='margin: 5px;' href='/'>Главная</a>
    <a style='margin: 5px;' href='/kurses'>Курсы</a>
    <a style='margin: 5px;' href='/teachers'>Преподаватели</a>
    <a style='margin: 5px;' href='/report'>Отчёт</a>
</div>
{% block content %} {% endblock %}
</body>
</html>

```

Файл index.html

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

```
{% block title %}
```

Главная страница

```
{% endblock %}
```

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

```
<h2>Главная страница</h2>
```

Студент Яровенко Максим, группа ИУ5Ц-72Б

```
{% endblock % }
```

Файл *kurses_delete_from.html*

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

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

<h2>Удаление записи</h2>

<p>Подтвердите удаление курса <i>{{ object.name }}</i></p>

<form action="" method="POST">

```
{% csrf_token % }
```

<input type="submit" value="Удалить" />

</form>

```
{% endblock % }
```

Файл *kurses_from.html*

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

```
{% block title % }
```

Изменение

```
{% endblock % }
```

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

<h2>Параметры записи</h2>

<form action="" method="post">

```
{% csrf_token % }
```

```

<table>
  {{ form.as_table }}
</table>

<br>

<input type="submit" value="Сохранить" />
</form>

```

```
{% endblock %}
```

Файл list.html

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

```
{% block title %}
```

```
{{ entity }}
```

```
{% endblock %}
```

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

```
<h2>Сущность {{ entity }}</h2>
```

```
<p>Список:</p>
```

```
<ul>
```

```
  {% for elem in objects %}
```

```
    <hr>
```

```
    <li>
```

```
      {% if entity == 'Teacher' %}
```

```
      <p>{{ elem.name }} {{ elem.family }}, {{ elem.status }}</p>
```

```
        <a href='{% url "teacher_update" elem.id %}'>Изменить</a> <br>
```

```
        <a href='{% url "teacher_delete" elem.id %}'>Удалить</a>
```

```
      {% endif %}
```

```

    {% if entity == 'Kurses' %}
    <p> {{ elem.name }} </p><br>
    <a href='{% url "kurses_update" elem.id %}'>Изменить</a> <br>
    <a href='{% url "kurses_delete" elem.id %}'>Удалить</a>
    {% endif %}
  </li>
{% endfor %}
<hr>
</ul>

```

```

{% if entity == 'Teacher' %}
<a href='/teacher/create'>Добавить запись</a>
{% endif %}

```

```

{% if entity == 'Kurses' %}
<a href='/kurses/create'>Добавить запись</a>
{% endif %}
{% endblock %}

```

Файл teacher_delete_from.html

```

{% extends "base.html" %}

```

```

{% block content %}

```

```

<h2>Удаление записи</h2>

```

```

<p>Подтвердите удаления языка {{ object.name }}</p>

```

```

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

```

```
<input type="submit" value="Удалить" />
</form>
{% endblock %}
```

Файл teacher_form.html

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

```
{% block title %}
```

Изменение

```
{% endblock %}
```

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

```
<h2>Параметры записи</h2>
```

```
<form action="" method="post">
```

```
    {% csrf_token %}
```

```
    <table>
```

```
        {{ form.as_table }}
```

```
    </table>
```

```
    <br>
```

```
    <input type="submit" value="Сохранить" />
```

```
</form>
```

```
{% endblock %}
```

Результат выполнения программы

Главная страница веб-сайта:

Главная страница

127.0.0.1:8000

Главная страница

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Главная страница

Студент Яровенко Максим, группа ИУ5Ц-72Б

Список курсов:

Kurses

127.0.0.1:8000

Kurses

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Сущность Kurses

Список:

Аналитическая геометрия

[Изменить](#)
[Удалить](#)

Информатика

[Изменить](#)
[Удалить](#)

Базы данных

[Изменить](#)
[Удалить](#)

[Добавить запись](#)

Список преподавателей:

Teacher

127.0.0.1:8000

Teacher

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Сущность Teacher

Список:

Илья Ильянов, Доцент

[Изменить](#)
[Удалить](#)

Андрей Петров, Профессор

[Изменить](#)
[Удалить](#)

Игорь Солнышко, Лаборант

[Изменить](#)
[Удалить](#)

[Добавить запись](#)

Добавление/изменение записи:

Изменение

127.0.0.1:8000

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Параметры записи

Name:

Family:

Status:

Сохранить

Изменение

+

←

↻

127.0.0.1:8000

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Параметры записи

Name:

Удаление записи:

127.0.0.1:8000/kurses/3/

×

+

←

↻

127.0.0.1:8000

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Удаление записи

Подтвердите удаление курса *Аналитическая геометрия*

127.0.0.1:8000/teacher/3/

×

+

←

↻

127.0.0.1:8000

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Удаление записи

Подтвердите удаления языка Илья

Отчет:

Преподавание

127.0.0.1:8000

Преподавание

[Главная](#) [Курсы](#) [Преподаватели](#) [Отчёт](#)

Список всех курсов и преподавателей:

• Курс: Аналитическая геометрия Преподаватель: Андрей Петров
• Курс: Информатика Преподаватель: Илья Ильинов
• Курс: Базы данных Преподаватель: Игорь Солнышко