

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.
5. **Предметная область:** книги и главы (вариант А-24).

Код программы

settings.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/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-*/rh8_+b3k-1wa2%0e1z9rbf#09#xjem649j_)xk0ldigc44k&'

# 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',
    'App.apps.AppConfig',
    'rest_framework'
]

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.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.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

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

# Password validation
# https://docs.djangoproject.com/en/3.2/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/3.2/topics/i18n/

```

```

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'


# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

models.py

```

from django.db import models

# Create your models here.
class Book(models.Model):
    title = models.CharField('Название книги', max_length=200)
    genre = models.CharField('Жанр книги', max_length=200)
    year = models.CharField('Год написания', max_length=4)

class Chapter(models.Model):
    chapter_book = models.ForeignKey(Book, on_delete=models.CASCADE)
    title = models.CharField('Название главы', max_length=500)
    page = models.IntegerField('Страница главы')

```

serializers.py

```

from .models import Book, Chapter
from rest_framework import serializers

class BookSerializer(serializers.ModelSerializer):
    class Meta:
        model = Book
        fields = ['title', 'genre', 'year']

class ChapterSerializer(serializers.ModelSerializer):
    class Meta:
        model = Chapter
        fields = ['chapter_book', 'title', 'page']

```

views.py

```
from django.shortcuts import render
from django.http import Http404
from rest_framework import viewsets
from .serializers import BookSerializer, ChapterSerializer
from .models import Book, Chapter

# Create your views here.
class BookViewSet(viewsets.ModelViewSet):
    queryset = Book.objects.all()
    serializer_class = BookSerializer

class ChapterViewSet(viewsets.ModelViewSet):
    queryset = Chapter.objects.all()
    serializer_class = ChapterSerializer

def BookList(request):
    return render(request, 'book/list.html', {'book_list':
Book.objects.all()})

def ChapterList(request):
    return render(request, 'chapter/list.html', {'chapter_list':
Chapter.objects.all()})

def BookDetail(request, book_id=0):
    try:
        book = Book.objects.get(id=book_id)
    except:
        raise Http404('Книга отсутствует!')
    return render(request, 'book/detail.html', {'book': book})

def ChapterDetail(request, chapter_id=0):
    try:
        chapter = Chapter.objects.get(id=chapter_id)
    except:
        raise Http404('Глава отсутствует!')
    return render(request, 'chapter/detail.html', {'chapter': chapter})
```

urls.py

```
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'books', app_views.BookViewSet)
router.register(r'chapters', app_views.ChapterViewSet)

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(router.urls)),
    path('app/bookslist', app_views.BookList),
    path('app/chapterslist', app_views.ChapterList),
    path('app/bookslist/<int:book_id>', app_views.BookDetail,
name='book_detail'),
    path('app/chapterslist/<int:chapter_id>', app_views.ChapterDetail,
name='chapter_detail')
```

base.html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>PK-2: {% block title %}{% endblock %}</title>
</head>
<body>
  Коротенко Евгений, гр. ИУ5Ц-71Б, РК-2
  <hr>
  {% block content %}{% endblock %}
</body>
</html>
```

book/list.html

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

{% block title %}Books List{% endblock %}

{% block content %}

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

  {% if book_list %}
    {% for book in book_list %}
      <a href="{% url 'book_detail' book.id %}">{{ book.title }} жанр:
    {{ book.genre }}</a><br>
    {% endfor %}
  {% else %}
    Список пустой!
  {% endif %}

{% endblock %}
```

book/detail.html

```
{% extends 'base.html' %}
{% block title %}{{ book.title }}{% endblock %}
{% block content %}

  <center><h1>{{ book.title }}</h1></center><hr><br>

  <table border="1" cellpadding="5">
    <tr>
      <th>Жанр</th>
      <th>{{ book.genre }}</th>
    </tr>
    <tr>
      <th>Год</th>
      <th>{{ book.year }}</th>
    </tr>
  </table>
  <br><hr><br>
  <a href="/app/bookslist">К списку книг</a>
  <br>
  <a href="/app/chapterslist">К списку глав</a>

{% endblock %}
```

chapter/list.html

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

{% block title %}Chapters List{% endblock %}

{% block content %}

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

    {% if chapter_list %}
        {% for chapter in chapter_list %}
            <a href="{% url 'chapter_detail' chapter.id %}">{{ chapter.title
}} стр. {{ chapter.page }}</a><br>
        {% endfor %}
    {% else %}
        Список пустой!
    {% endif %}

{% endblock %}
```

chapter/detail.html

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

{% block title %}{{ chapter.title }}{% endblock %}

{% block content %}

    <center><h1>{{ chapter.title }}</h1></center><hr><br>

    <table border="1" cellpadding="5">
    <tr>
        <th>Страница</th>
        <th>{{ chapter.page }}</th>
    </tr>
    </table>
    <br><hr><br>
    <a href="/app/chapterslist">К списку глав</a>
    <br>
    <a href="/app/bookslist">К списку книг</a>

{% endblock %}
```

Добавление данных через API

Добавление книги

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "title": "Думай и богатей",
    "genre": "Саморазвитие",
    "year": "1937"
  },
  {
    "title": "Белый клык",
    "genre": "Повесть",
    "year": "1906"
  }
]
```

Raw dataHTML form

Название книги	Таинственный остров
Жанр книги	Приключения
Год написания	1875

POST

```
HTTP 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

[
  {
    "title": "Думай и богатей",
    "genre": "Саморазвитие",
    "year": "1937"
  },
  {
    "title": "Белый клык",
    "genre": "Повесть",
    "year": "1906"
  },
  {
    "title": "Таинственный остров",
    "genre": "Приключения",
    "year": "1875"
  }
]
```

Добавление главы

```
{
  "chapter_book": 1,
  "title": "Глава 2: разбогатеи",
  "page": 15
},
{
  "chapter_book": 2,
  "title": "Часть 1",
  "page": 1
},
{
  "chapter_book": 2,
  "title": "Часть 2",
  "page": 33
},
{
  "chapter_book": 2,
  "title": "Часть 3",
  "page": 70
}
]
```

Raw data HTML form

Chapter book

Book object (3)

Название главы

Глава 1: Полет

Страница главы

2

POST

```
{
  "chapter_book": 2,
  "title": "Часть 3",
  "page": 70
},
{
  "chapter_book": 3,
  "title": "Глава 1: Полет",
  "page": 2
}
]
```


Просмотр данных

Список книг /app/bookslist

Коротенко Евгений, гр. ИУ5Ц-71Б, РК-2

Список книг

[Думаи и богатей жанр: Саморазвитие](#)
[Белый клык жанр: Повесть](#)
[Таинственный остров жанр: Приключения](#)

Список глав /app/chapterslist

Коротенко Евгений, гр. ИУ5Ц-71Б, РК-2

Список глав

[Глава 1: подумай стр. 3](#)
[Глава 2: разбогатей стр. 15](#)
[Часть 1 стр. 1](#)
[Часть 2 стр. 33](#)
[Часть 3 стр. 70](#)
[Глава 1: Полет стр. 2](#)

Просмотр книги /app/bookslist/3

Коротенко Евгений, гр. ИУ5Ц-71Б, РК-2

Таинственный остров

Жанр	Приключения
Год	1875

[К списку книг](#)
[К списку глав](#)

Глава 1: Полет