

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по рубежному контролю №1

Вариант Г16

Выполнила:  
студентка группы ИУ5-54Б  
Носова Элина  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю. Е.  
Подпись и дата:

Москва, 2021 г.

## 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.

Сущности: «Магазин» и «Книга»

Атрибуты для «Магазин»: idstore, name.

Атрибуты для «Книга»: id, title, author, price, idstore.

Models.py

```
from django.db import models

# Create your models here.
class Store(models.Model):
    objects = models.Manager()
    id = models.AutoField(primary_key=True)
    name = models.CharField(max_length=30, blank=True, null=True)

class Book(models.Model):
    objects = models.Manager()
    id = models.AutoField(primary_key=True)
    title = models.CharField(max_length=50, blank=True, null=True)
    author = models.CharField(max_length=50, blank=True, null=True)
    price = models.IntegerField(blank=True, null=True)
    idstore = models.ForeignKey(Store, on_delete=models.CASCADE, null=True)
```

views.py

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.http import HttpResponseNotFound
from bookstore.models import Store
from bookstore.models import Book

def index(request):
    stores = Store.objects.all()
    books = Book.objects.all()
    return render(request, "index.html", {"stores": stores, "books": books})

def create(request):
    if request.method == "POST":
        sb = Store()
        sb.name = request.POST.get("name")
        sb.save()
        return HttpResponseRedirect("/")

def create_book(request):
    if request.method == "POST":
```

```

        b = Book()
        b.title = request.POST.get("title")
        b.author = request.POST.get("author")
        b.price = request.POST.get("price")
        b.idstore = Store.objects.get(id=request.POST.get("idstore"))
        b.save()
    return HttpResponseRedirect("/")

# изменение данных в бд
def edit(request, id):
    try:
        boo = Book.objects.get(id=id)

        if request.method == "POST":
            # boo = Book()
            boo.title = request.POST.get("title")
            boo.author = request.POST.get("author")
            boo.price = request.POST.get("price")
            boo.idstore = Store.objects.get(id=request.POST.get("idstore"))
            boo.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit.html", {"book": boo})
    except Book.DoesNotExist:
        return HttpResponseRedirect("<h2>Book is not found</h2>")

def edit_st(request, id):
    try:
        st = Store.objects.get(id=id)

        if request.method == "POST":
            # st = Store()
            st.name = request.POST.get("name")
            st.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit_store.html", {"store": st})
    except Store.DoesNotExist:
        return HttpResponseRedirect("<h2>Store is not found</h2>")

# удаление данных из бд
def delete(request, id):
    try:
        b = Book.objects.get(id=id)
        b.delete()
        return HttpResponseRedirect("/")
    except Book.DoesNotExist:
        return HttpResponseRedirect("<h2>Book is not found</h2>")

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

```

urls.py

```

from django.shortcuts import render
from django.urls import path
from bookstore import views

```

```
urlpatterns = [
    path('', views.index),
    path('create/', views.create),
    path('create_book/', views.create_book),
    path('edit/<int:id>/', views.edit),
    path('edit_st/<int:id>/', views.edit_st),
    path('delete/<int:id>/', views.delete),
    path('delete_st/<int:id>/', views.delete_st),
]
```

## index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>pk2</title>
</head>
<body class="container">
    <form method="POST" action="create_book/">
        {% csrf_token %}

        <h2>Добавление новой книги</h2>
        <p>
            <label>Введите название</label><br>
            <input type="text" name="title" />
        </p>
        <p>
            <label>Введите автора</label><br>
            <input type="text" name="author" />
        </p>
        <p>
            <label>Введите стоимость</label><br>
            <input type="text" name="price" />
        </p>
        <p>
            <label>Введите id магазина</label><br>
            <input type="text" name="idstore" />
        </p>
        <input type="submit" value="Сохранить" >
    </form>

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

        <h2>Добавление нового магазизана</h2>
        <p>
            <label>Введите название магазина</label><br>
            <input type="text" name="name" />
        </p>
        <input type="submit" value="Сохранить" >

        <h2>Список магазинов</h2>
        <table>
            <tr>

                <th>Название</th>
            </tr>
            {% for st in stores %}
            <tr>
                <td>{{ st.id }}</td>
                <td>{{ st.name }}</td>
                <td><a href="edit_st/{{st.id}}">Изменить</a> | <a
href="delete_st/{{st.id}}">Удалить</a></td>
            </tr>
```

```

        {% endfor %}
    </table>

    {% if books.count > 0 %}
    <h2>Список дисков</h2>
    <table>
        <tr>

            <th>Название</th>
            <th>Ёмкость</th>
            <th>Библиотека</th>
            <th>Библиотека</th>
        </tr>
        {% for b in books %}
        <tr>
            <td>{{ boo.id }}</td>
            <td>{{ b.title }}</td>
            <td>{{ b.author }}</td>
            <td>{{ b.price }}</td>
            <td>{{ boo.idstore }}</td>
            <td><a href="edit/{{b.id}}">Изменить</a> | <a
href="delete/{{b.id}}">Удалить</a></td>
        </tr>
        {% endfor %}
    </table>
    {% endif %}

    {% for st in stores %}
        <h3>Диски в библиотеке {{ st.name }}:</h3>
        {% for b in books %}
            {% if st.id == b.idstore %}
                <li>{{b.name}}</li>
            {% endif %}
        {% endfor %}
    {% endfor %}
</form>
</body>
</html>

```

## edit.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>edit disk</title>
</head>
<body class="container">
    <form method="POST">
        {% csrf_token %}
        <p>
            <p>
                <label>Введите автора</label><br>
                <input type="text" name="author" />
            </p>
            <p>
                <label>Введите стоимость</label><br>
                <input type="text" name="price" />
            </p>
            <p>
                <label>Введите id магазина</label><br>
                <input type="text" name="idstore"/>
            </p>
            <input type="submit" value="Сохранить" >
        </form>

```

```
</body>
</html>
```

## edit\_store.html

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

## Insomnia



