

Отчет по РК № 2 по курсу
"Разработка Интернет-Приложений"

Выполнил:
Студент группы
ИУ5-55Б
Хижняков В.М.

Москва, МГТУ – 2021

Задание:

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

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.

№ варианта	Класс 1	Класс 2
16	Книга	Книжный магазин

Текст программы:

rk2/urls.py

```
from django.contrib import admin
from django.urls import path, include
from books import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.index, name="main"),
    path('book/', include([
        path('', views.read_book, name='read_book'),
        path('create/', views.create_book, name="create_book"),
        path('update/<int:book_id>', views.update_book, name="update_book"),
        path('delete/<int:book_id>', views.delete_book, name="delete_book"),
    ])),
    path('store/', include([
        path('', views.read_store, name='read_store'),
        path('create/', views.create_store, name="create_store"),
        path('update/<int:store_id>',
            views.update_store, name="update_store"),
        path('delete/<int:store_id>',
            views.delete_store, name="delete_store"),
    ])),
    path('report/', views.report, name="report"),
]
```

books/models.py

```
from django.db import models

class BookStore(models.Model):
    name = models.CharField(max_length=256, verbose_name="Store name")

    def __str__(self):
        return self.name

class Book (models.Model):
    name = models.CharField(max_length=256, verbose_name="Book name")
    cost = models.PositiveIntegerField(verbose_name="Book cost")
    store = models.ForeignKey(
        BookStore,
        on_delete=models.SET_DEFAULT,
        null=True,
        default=None,
        related_name="books"
    )

    def __str__(self):
        return self.name
```

books/views.py

```
from django.db.models.aggregates import Avg
from django.http.response import HttpResponseRedirect
from django.shortcuts import get_list_or_404, get_object_or_404, redirect, render
from .models import BookStore, Book
from django.db import models

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

# BOOKS

def read_book(request):
    books = Book.objects.all()
    return render(request, 'book/book_list.html', {'books': books})

def create_book(request):
    if request.method == 'GET':
        stores = BookStore.objects.all()
        return render(request, 'book/create_book.html', {"stores": stores})
    else:
        dto = {}
        for key in request.POST:
            if key in Book.__dict__:
```

```

        dto[key] = request.POST[key]

        dto['store'] = get_object_or_404(BookStore, pk=request.POST['store'])
        new_book = Book(**dto)
        new_book.save()
        return redirect('read_book')

def update_book(request, book_id):
    if request.method == 'GET':
        stores = BookStore.objects.all()
        book = get_object_or_404(Book, pk=book_id)
        return render(request, 'book/update_book.html', {"book": book, "stores": stores})
    else:
        book = get_object_or_404(Book, pk=book_id)
        for key in request.POST:
            if key in book.__dict__ and key != 'store':
                setattr(book, key, request.POST[key])
        if 'store' in request.POST:
            setattr(book, 'store', get_object_or_404(
                BookStore, pk=request.POST['store']))
        book.save()
        return redirect('read_book')

def delete_book(request, book_id):
    book = get_object_or_404(Book, pk=book_id)
    book.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# STORES

def read_store(request):
    stores = BookStore.objects.all()
    return render(request, 'store/store_list.html', {'stores': stores})

def create_store(request):
    if request.method == 'GET':
        return render(request, 'store/create_store.html')
    else:
        new_store = BookStore(name=request.POST['name'])
        new_store.save()
        return redirect('read_store')

def update_store(request, store_id):
    if request.method == 'GET':
        store = get_object_or_404(BookStore, pk=store_id)
        return render(request, 'store/update_store.html', {"store": store})
    else:
        store = get_object_or_404(BookStore, pk=store_id)
        for key in request.POST:
            if key in store.__dict__:
                setattr(store, key, request.POST[key])

```

```

        store.save()
        return redirect('read_store')

def delete_store(request, store_id):
    store = get_object_or_404(BookStore, pk=store_id)
    store.delete()
    return redirect(request.META.get('HTTP_REFERER'))

# REPORT

def report(request):
    expensive_books = Book.objects.filter(cost__gt=1000)
    avg_prices = []
    for store in BookStore.objects.all():
        avg_prices.append({"store": store, "price": Book.objects.filter(
            store=store.pk).aggregate(Avg('cost'))['cost__avg']})
    return render(request, 'report.html', {"expensive_books": expensive_books, "avg_prices":
avg_prices})

```

templates/inc/base.html

```

{% load static %}

<!doctype html>
<html lang="en" class="h-100">

<head>
    <!-- ===== CHARSET AND VIEWPORT ===== -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <!-- ===== CACHE TURN OFF ===== -->
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <!-- ===== COPY RIGHT ===== -->
    <meta name="author" content="IU5-55 Vadim Khizhnyakov">
    <title>
        IU5-55 Khizhnuakov RK2
    </title>

    <!-- ===== BOOTSTRAP CORE CSS ===== -->
    <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <!-- ===== SHARED STYLES ===== -->
    <link rel="stylesheet" href="{% static 'css/navbar.css' %}">
</head>

<body class="d-flex flex-column h-100">
    <header>
        <!-- Navbar -->

```

```

    {% include 'inc/navbar.html' %}
</header>

<!-- Begin page content -->
<main role="main" class="flex-shrink-0">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <!-- Content -->
                {% block content %}
                {% endblock content %}
            </div>
        </div>
    </div>
</main>

<footer class="footer mt-auto py-3">
    <!-- Footer -->
</footer>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</html>

```

templates/inc/navbar.html

```

{% load static %}
<nav class="navbar navbar-expand-sm navbar-dark bg-dark p-3">
    <a class="navbar-brand ml-5" href="{% url 'main' %}">Khizhnyakov</a>

    <ul class="navbar-nav ml-auto">
        <li class="nav-item">
            <a href="{% url 'main' %}" class="nav-link">Main Page</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'read_book' %}" class="nav-link">All books</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'create_book' %}" class="nav-link">Create book</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'read_store' %}" class="nav-link">All stores</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'create_store' %}" class="nav-link">Create store</a>
        </li>
        <li class="nav-item">
            <a href="{% url 'report' %}" class="nav-link">Report</a>
        </li>
    </ul>
</nav>

```

templates/book/book_card.html

```
{% load static %}

<div class="card mb-3">
  <div class="row g-0">
    <div class="col-md-4">
      
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title">{{ book.name }}</h5>
        <p class="card-text">{{ book.store.name }}</p>
        <p class="card-text">
          This is a wider card with supporting text below as a natural lead-in to
additional content. This
          content is a little bit longer.
        </p>
        <p class="card-text"><small class="text-muted">{{ book.cost }}</small></p>
        <a href="{% url 'delete_book' book.pk %}" class="btn btn-danger"> Delete
</a>
        <a href="{% url 'update_book' book.pk %}" class="btn btn-primary"> Update
</a>
      </div>
    </div>
  </div>
</div>
```

templates/book/book_form.html

```
<div class="mb-3">
  <label for="bookName" class="form-label">Book Name</label>
  <input type="text" name="name" value="{{ book.name }}" required="required"
class="form-control" id="bookName" />
</div>

<div class="mb-3">
  <label for="bookCost" class="form-label">Book Cost</label>
  <input
    type="number"
    min="0"
    name="cost"
    value="{{ book.cost }}"
    required="required"
    class="form-control"
    id="bookCost"
  />
</div>

<div class="mb-3">
  <label for="bookStore" class="form-label">Select Store</label>
  <select name="store" class="form-select" aria-label="Default select example"
id="bookStore">
    {% for store in stores %}
```

```

        <option value="{{ store.pk }}">{{ store.name }}</option>
    {% endfor %}
</select>
</div>

```

templates/book/book_list.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">All Books</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-column">
    {% for book in books %} {% include 'book/book_card.html' with book=book %} {% endfor %}
</div>
{% endblock content %}

```

templates/book/create_book.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">New Book</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'create_book' %}" method="POST" class="form">
        {% csrf_token %} {% include 'book/book_form.html' with book=book %}
        <button type="submit" class="btn btn-primary">Create</button>
    </form>
</div>
{% endblock content %}

```

templates/book/update_book.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Book</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_book' book.pk %}" method="POST" class="form">
        {% csrf_token %} {% include 'book/book_form.html' with book=book %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}

```

templates/store/store_card.html

```

{% load static %}

```



```

<div class="card m-3" style="width: 18rem">
  
  <div class="card-body">
    <h5 class="card-title">{{ store.name }}</h5>
    <p class="card-text">
      Some quick example text to build on the card title and make up the bulk of the
      card's content.
    </p>
    <a href="{% url 'delete_store' store.pk %}" class="btn btn-danger"> Delete </a>
    <a href="{% url 'update_store' store.pk %}" class="btn btn-primary"> Update </a>
  </div>
</div>

```

templates/store/store_form.html

```

<div class="mb-3">
  <label for="storeName" class="form-label">Store Name</label>
  <input
    type="text"
    name="name"
    value="{{ store.name }}"
    required="required"
    class="form-control"
    id="storeName"
  />
</div>

```

templates/store/store_list.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
  <h1 class="text-center mt-4">All Stores</h1>
</div>
<!-- Page Content -->
<div class="d-flex flex-wrap">
  {% for store in stores %} {% include 'store/store_card.html' with store=store %} {%
endfor %}
</div>
{% endblock content %}

```

templates/store/create_store.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
  <h1 class="text-center mt-4">New Store</h1>
</div>
<!-- Page Content -->
<div>
  <form action="{% url 'create_store' %}" method="POST" class="form">
    {% csrf_token %} {% include 'store/store_form.html' with store=store %}
    <button type="submit" class="btn btn-primary">Create</button>
  </form>
</div>

```

```

    </form>
</div>
{% endblock content %}

```

templates/store/update_store.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Update Store</h1>
</div>
<!-- Page Content -->
<div>
    <form action="{% url 'update_store' store.pk %}" method="POST" class="form">
        {% csrf_token %} {% include 'store/store_form.html' with store=store %}
        <button type="submit" class="btn btn-primary">Save</button>
    </form>
</div>
{% endblock content %}

```

templates/report.html

```

{% extends 'inc/base.html' %} {% load static %} {% block content %}
<div class="container">
    <h1 class="text-center mt-4">Report Page</h1>
</div>
<!-- Page Content -->
<div>
    <h1>Expensive books</h1>
    {% for book in expensive_books %} {% include 'book/book_card.html' with book=book %}
{% endfor %}
</div>

<div>
    <h1>Average costs</h1>
    <div class="d-flex flex-wrap">
        {% for avg_price in avg_prices %}
        <div class="d-flex flex-column align-center" >
            <h3 class="text-center">Average cost {{ avg_price.price }}</h3>
            {% include 'store/store_card.html' with store=avg_price.store %}
        </div>
        {% endfor %}
    </div>
</div>
{% endblock content %}

```

templates/index.html

```

{% extends 'inc/base.html' %}

{% load static %}

{% block content %}

```


```
<div class="container">
  <h1 class="text-center mt-4">Main Page</h1>
</div>
<!-- Page Content -->
<div>
  
</div>
{% endblock content %}
```

Результат работы:

Khizhnyakov

[Main Page](#)[All books](#)[Create book](#)[All stores](#)[Create store](#)[Report](#)


Main Page



Khizhnyakov

[Main Page](#)[All books](#)[Create book](#)[All stores](#)[Create store](#)[Report](#)

All Books



latte


bloom-n-brew

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

123

Delete

Update



capuchino


brrrew

This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

123123

Delete

Update



work pls

brrrew

KhizhnyakovMain PageAll booksCreate bookAll storesCreate storeReport

New Book

Book Name

Book Cost

Select Store

brrrew

Create

KhizhnyakovMain PageAll booksCreate bookAll storesCreate storeReport

Update Book

Book Name

latte

Book Cost

123


Select Store

brrrew

Save

KhizhnyakovMain PageAll booksCreate bookAll storesCreate storeReport

All Stores




brrrew

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete

Update



bloom-n-brew

Some quick example text to build on the card title and make up the bulk of the card's content.

Delete

Update

New Store

Store Name

Create



Update Store

Store Name

Save

Report Page

Expensive books

	<div>capuchino</div> <div>brrrew</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>123123</div> <div>DeleteUpdate</div>
	<div>work pls</div> <div>brrrew</div> <div>This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.</div> <div>123123</div> <div>DeleteUpdate</div>

Average costs

Average cost 61872.75

Average cost 345.0



brrrew

Some quick example text to build on the card title and make up the bulk of the card's content.

DeleteUpdate



bloom-n-brew

Some quick example text to build on the card title and make up the bulk of the card's content.

DeleteUpdate