**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Рубежный контроль №2

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Корчевский Александр |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 г.

**Описание задания:**

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.

Сущности: “Компьютер”, “Дисплейный класс”.

**Текст программы**

***models.py:***

from django.db import models

class BookManager(models.Manager):

def ordered(self):

return self.all().order\_by('title')

def report(self):

books = self.all().order\_by('lib\_id')

return books

class LibManager(models.Manager):

def ordered(self):

return self.all().order\_by('name')

class Book(models.Model):

book\_id = models.AutoField('book\_id', primary\_key=True)

author = models.CharField('Author', max\_length=50)

title = models.CharField('Title', max\_length=50)

pub\_year = models.IntegerField('Publish year')

lib\_id = models.ForeignKey('Library', models.DO\_NOTHING, db\_column='lib\_id')

objects = BookManager()

class Library(models.Model):

lib\_id = models.AutoField('lib\_id', primary\_key=True)

name = models.CharField('Name', max\_length=50)

phone = models.CharField('Phone', max\_length=30)

email = models.CharField('Email', max\_length=50)

objects = LibManager()

def \_\_str\_\_(self):

return str(self.name)

***urls.py:***

from django.contrib import admin

from django.urls import path

from app import views

urlpatterns = [

path('admin/', admin.site.urls),

path('', views.index),

path('book/', views.book\_list),

path('book/create', views.BookCreate.as\_view(), name='book\_create'),

path('book/<int:pk>/update', views.BookUpdate.as\_view(), name='book\_update'),

path('book/<int:pk>/delete', views.BookDelete.as\_view(), name='book\_delete'),

path('library/', views.lib\_list),

path('library/create', views.LibCreate.as\_view(), name='lib\_create'),

path('library/<int:pk>/update', views.LibUpdate.as\_view(), name='lib\_update'),

path('library/<int:pk>/delete', views.LibDelete.as\_view(), name='lib\_delete'),

path('report/', views.report),

]

***views.py:***

from django import forms

from django.shortcuts import render

from .models import Library, Book

from django.views.generic.edit import CreateView, UpdateView, DeleteView

def index(request):

return render(request, 'index.html')

def report(request):

books = Book.objects.report()

return render(request, 'report.html', {'books': books})

def book\_list(request):

books = Book.objects.ordered()

return render(request, 'list.html', {'objs': books, 'name': 'Books'})

def lib\_list(request):

libs = Library.objects.ordered()

return render(request, 'list.html', {'objs': libs, 'name': 'Libraries'})

class BookCreate(CreateView):

model = Book

fields = ['title', 'author', 'pub\_year', 'lib\_id']

success\_url = '/book'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(BookCreate, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Create'

context['entity'] = 'book'

context['form'].fields['lib\_id'] = forms.ModelChoiceField(queryset=Library.objects.all(),

empty\_label=None, label='Library')

return context

class BookUpdate(UpdateView):

model = Book

fields = ['title', 'author', 'pub\_year', 'lib\_id']

success\_url = '/book'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(BookUpdate, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Update'

context['entity'] = 'book'

context['form'].fields['lib\_id'] = forms.ModelChoiceField(queryset=Library.objects.all(),

empty\_label=None, label='Library')

return context

class BookDelete(DeleteView):

model = Book

pk\_url\_kwarg = 'book\_id'

success\_url = '/book'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(BookDelete, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Delete'

context['entity'] = 'book'

context['delete'] = True

return context

class LibCreate(CreateView):

model = Library

fields = ['name', 'phone', 'email']

success\_url = '/library'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(LibCreate, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Create'

context['entity'] = 'library'

return context

class LibUpdate(UpdateView):

model = Library

fields = ['name', 'phone', 'email']

success\_url = '/library'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(LibUpdate, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Update'

context['entity'] = 'library'

return context

class LibDelete(DeleteView):

model = Library

pk\_url\_kwarg = 'lib\_id'

success\_url = '/library'

template\_name = 'form.html'

def get\_context\_data(self, \*\*kwargs):

context = super(LibDelete, self).get\_context\_data(\*\*kwargs)

context['name'] = 'Delete'

context['entity'] = 'library'

context['delete'] = True

return context

***base.html:***

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>RK2</title>

{% load static %}

<link rel="stylesheet" type="text/css" href="{% static '/css/bootstrap.min.css' %}">

<link rel="stylesheet" type="text/css" href="{% static '/css/main.css' %}">

</head>

<body>

<nav>

<a href="/">Main page</a>

<a href="/book">Books</a>

<a href="/library">Libraries</a>

<a href="/report">Report</a>

</nav>

<div class="content">

{% block content %}{% endblock %}

</div>

<script src="{% static 'js/bootstrap.bundle.min.js'%}"></script>

</body>

</html>

***index.html:***

{% extends 'base.html' %}

{% block content %}

<h2>Main page</h2>

<h3>RK2 Korchevsky Alexandr IU5-52B</h3>

{% endblock %}

***report.html:***

{% extends 'base.html' %}

{% block content %}

<h2>Report</h2>

<ul>

{% for book in books %}

<li>

<dvi>

<b>Title: </b>{{ book.title }}<br>

<b>Author: </b>{{ book.author}}<br>

<b>Publish year: </b>{{ book.pub\_year }}<br>

<b>Library: </b>{{ book.lib\_id.name }}<br>

</dvi>

</li>

<br>

{% endfor %}

</ul>

{% endblock %}

***form.html:***

{% extends 'base.html' %}

{% load bootstrap4 %}

{% block content %}

<h2>{{ name }} data</h2>

{% if delete %}

<p>Are u sure u wanna delete {{ entity }} {{ object.title }}?</p>

{% endif %}

<form class="any-form" action="" method="post">

{% csrf\_token %}

{% if not delete %}

{% bootstrap\_form form %}

{% endif %}

<input type="submit" value="{{ name }}"/>

</form>

{% endblock %}

***list.html:***

{% extends 'base.html' %}

{% block content %}

<div class="list-header">

<h2>{{ name }}</h2>

{% if name == 'Books' %}

<button class="create"><a href="{% url 'book\_create' %}">Create</a></button>

{% else %}

<button class="create"><a href="{% url 'lib\_create' %}">Create</a></button>

{% endif %}

</div>

<ul>

{% for obj in objs %}

<li>

{% if name == 'Books' %}

<div class="elem"><b>Title: </b>{{ obj.title }}</div>

<div class="elem"><b>Author: </b>{{ obj.author }}</div>

<div class="elem"><b>Publish year: </b>{{ obj.pub\_year }}</div>

<div class="button-wrapper">

<button class="update"><a href="{% url 'book\_update' obj.book\_id %}">Update</a></button>

<button class="delete"><a href="{% url 'book\_delete' obj.book\_id %}">Delete</a></button>

</div>

{% else %}

<div class="elem"><b>Name: </b>{{ obj.name }}</div><br>

<div class="elem"><b>Email: </b>{{ obj.email}}</div><br>

<div class="elem"><b>Phone: </b>{{ obj.phone }}</div><br>

<div class="button-wrapper">

<button class="update"><a href="{% url 'lib\_update' obj.lib\_id %}">Update</a></button>

<button class="delete"><a href="{% url 'lib\_delete' obj.lib\_id %}">Delete</a></button>

</div>

{% endif %}

<br>

</li>

{% endfor %}

</ul>

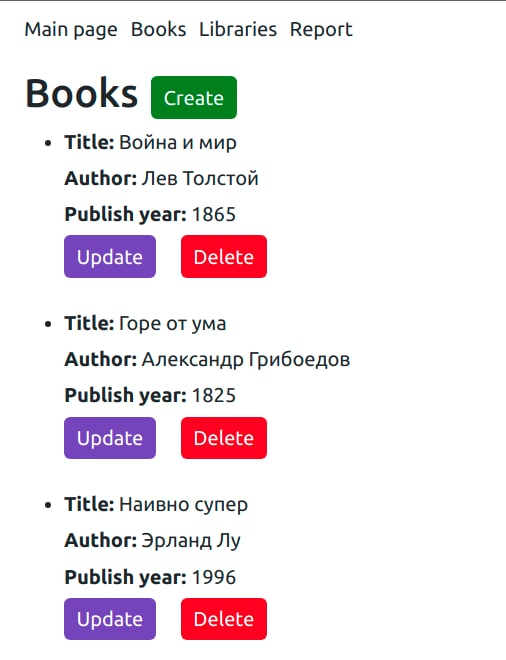
{% endblock %}

**Скриншоты:**

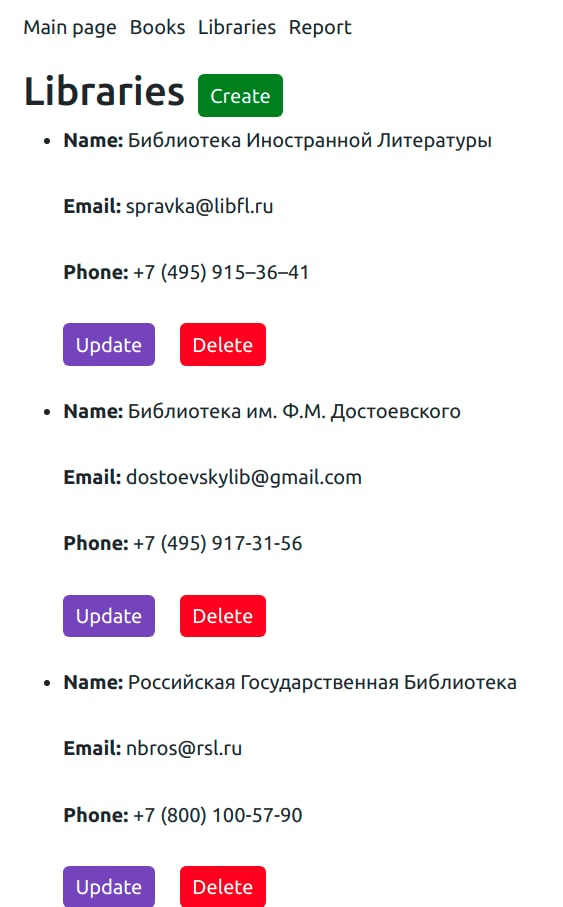
Главная страница:

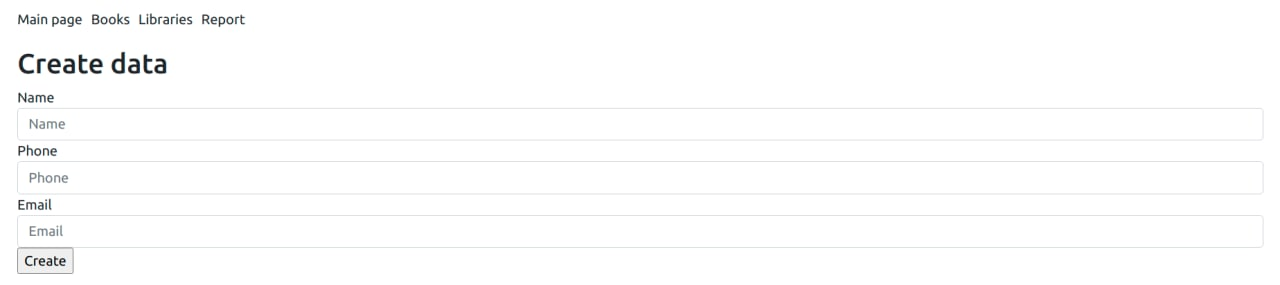


Книги:





Библиотеки:



Отчет:

