



КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Студент Тарновский Даниил Романович
фамилия, имя, отчество
 Группа ИУ5-53Б

Преподаватель 21.12.2021 Гапанюк Ю.Е.
подпись, дата
фамилия, и.о.

2021г.

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.

Код:

models.py:

```
from django.db import models

class Lib(models.Model):

    objects = models.Manager()
    name = models.CharField(max_length=50, verbose_name="Название библиотеки")

class Lang(models.Model):

    objects = models.Manager()
    name = models.CharField(max_length=50, verbose_name="Название языка")
    version = models.CharField(max_length=50, verbose_name="Версия языка")
    langLib = models.ForeignKey(Lib, on_delete=models.CASCADE, null=True)
```

urls.py:

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

urlpatterns = [
    path("", views.index),
    path('create/', views.create),
    path('create_lang/', views.create_lang),
    path('edit/<int:id>/', views.edit),
    path('edit_lib/<int:id>/', views.edit_lib),
    path('delete/<int:id>/', views.delete),
    path('delete_lib/<int:id>/', views.delete_lib),
```

```
]
```

views.py:

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from django.http import HttpResponseNotFound
from cd.models import Lang
from cd.models import Lib

def index(request):
    libs = Lib.objects.all()
    langs = Lang.objects.all()
    return render(request, "index.html", {"libs": libs, "langs": langs})

# сохранение данных в бд
def create(request):
    if request.method == "POST":
        dl = Lib()
        dl.name = request.POST.get("name_lib")
        dl.save()
    return HttpResponseRedirect("/")

def create_lang(request):
    if request.method == "POST":
        d = Lang()
        d.name = request.POST.get("name")
        d.version = request.POST.get("version")
        d.langLib = Lib.objects.get(name=request.POST.get("langLib"))
        d.save()
    return HttpResponseRedirect("/")

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

        if request.method == "POST":
            lang.name = request.POST.get("name")
            lang.version = request.POST.get("version")
            lang.Lib = Lib.objects.get(name=request.POST.get("langLib"))
            lang.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit.html", {"lang": lang})
    except Lang.DoesNotExist:
        return HttpResponseNotFound("<h2>Lang not found</h2>")

def edit_lib(request, id):
    try:
```

```

lib = Lib.objects.get(id=id)

if request.method == "POST":
    lib.name = request.POST.get("name")
    lib.save()
    return HttpResponseRedirect("/")
else:
    return render(request, "edit_lib.html", {"lib": lib})
except Lib.DoesNotExist:
    return HttpResponseRedirect("<h2>Library not found</h2>")

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

def delete_lib(request, id):
    try:
        lib = Lib.objects.get(id=id)
        lib.delete()
        return HttpResponseRedirect("/")
    except Lang.DoesNotExist:
        return HttpResponseRedirect("<h2>Library not found</h2>")

```

index.html:

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>рк2</title>
</head>
<body class="container">
    <form method="POST" action="create_lang/">
        { % csrf_token % }

        <h2>Добавление нового языка</h2>
        <p>
            <label>Введите название языка</label><br>
            <input type="text" name="name" />
        </p>
        <p>
            <label>Введите версию</label><br>
            <input type="text" name="version" />
        </p>
        <p>
            <label>Введите название библиотеки</label><br>
            <input type="text" name="langLib"/>
        </p>
        <input type="submit" value="Сохранить" >
    </form>

```

```

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

  <h2>Добавление новой библиотеки</h2>
  <p>
    <label>Введите название библиотеки</label><br>
    <input type="text" name="name_lib" />
  </p>
  <input type="submit" value="Сохранить" >

  <h2>Список библиотек</h2>
  <table>
    <tr>

      <th>Название</th>
    </tr>
    {% for lib in libs %}
    <tr>
      <td>{{ lib.name }}</td>
      <td><a href="edit_lib/{{ lib.id }}">Изменить</a> | <a href="delete_lib/{{ lib.id }}">Удалить</a></td>
    </tr>
    {% endfor %}
  </table>

  {% if langs.count > 0 %}
  <h2>Список языков</h2>
  <table>
    <tr>

      <th>Название</th>
      <th>Версия</th>
      <th>Библиотека</th>
    </tr>
    {% for lang in langs %}
    <tr>

      <td>{{ lang.name }}</td>
      <td>{{ lang.version }}</td>
      <td>{{ lang.langLib.name }}</td>
      <td><a href="edit/{{ lang.id }}">Изменить</a> | <a href="delete/{{ lang.id }}">Удалить</a></td>
    </tr>
    {% endfor %}
  </table>
  {% endif %}

  {% for lib in libs %}
  <h3>Языки с библиотекой {{ lib.name }}:</h3>
  {% for lang in langs %}
    {% if lib.name == lang.langLib.name %}
    <li>{{ lang.name }}</li>
    {% endif %}
  {% endfor %}
  {% endfor %}
</form>
</body>
</html>

```

edit.html:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>edit language</title>
</head>
<body class="container">
  <form method="POST">
    {% csrf_token %}
    <p>
      <label>Введите название языка программирования</label><br>
      <input type="text" name="name" value="{{ lang.name }}" />
    </p>
    <p>
      <label>Введите версию</label><br>
      <input type="text" name="version" value="{{ lang.version }}" />
    </p>
    <p>
      <label>Введите название библиотеки</label><br>
      <input type="text" name="langLib" value="{{ lang.langLib.name }}" />
    </p>
    <input type="submit" value="Сохранить" >
  </form>
</body>
</html>
```

edit_lib.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 библиотеки</label><br>
      <input type="text" name="name" value="{{ lib.name }}" />
    </p>
    <input type="submit" value="Сохранить" >
  </form>
</body>
</html>
```

Демонстрация работы:
Добавим новый язык:

Добавление нового языка

Введите название языка

C++

Введите версию

17

Введите название библиотеки

sfml

Сохранить

Список библиотек

Название

sfml	Изменить Удалить
string	Изменить Удалить
iostream	Изменить Удалить
iomanip	Изменить Удалить
cmath	Изменить Удалить
cstdint	Изменить Удалить
Pandas	Изменить Удалить
NumPy	Изменить Удалить
SciPy	Изменить Удалить

Список языков

Название Версия Библиотека

C++	17	sfml	Изменить Удалить
C++	14	string	Изменить Удалить
python	3	NumPy	Изменить Удалить
python	2	SciPy	Изменить Удалить

Языки с библиотекой sfml:

- C++

Языки с библиотекой string:

- C++

Языки с библиотекой NumPy:

- python

Языки с библиотекой SciPy:

- python

Удалим NumPy

Языки с библиотекой SciPy:

- python

Список библиотек

Название

sfml	Изменить Удалить
string	Изменить Удалить
iostream	Изменить Удалить
iomanip	Изменить Удалить
cmath	Изменить Удалить
cstdint	Изменить Удалить
Pandas	Изменить Удалить
SciPy	Изменить Удалить

Меняем название Pandas

Введите название библиотеки

Список библиотек

Название

sfml	Изменить	Удалить
string	Изменить	Удалить
iostream	Изменить	Удалить
iomanip	Изменить	Удалить
cmath	Изменить	Удалить
cstdint	Изменить	Удалить
Pandass	Изменить	Удалить
SciPy	Изменить	Удалить