

Московский государственный технический  
университет им. Н. Э. Баумана

Факультет «Радиотехнический»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»  
Отчет по рубежному  
контролю

Выполнил:

студент группы РТ5-51  
Робертс Д.А.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Подпись и дата:

Москва, 2021 г.

# Полученное задание:

## Вариант Е, 15

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.

# Текст программы:

Файл views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from django.http import HttpResponseRedirect
from django.http import HttpResponseNotFound
from main.models import F_Dir
from main.models import Filee

def index(request):
    Dirs = F_Dir.objects.all()
    files = Filee.objects.all()
    return render(request, "index.html", {"Dirs": Dirs, "files": files})

    # сохранение данных в бд
def create(request):
    if request.method == "POST":
        dl = F_Dir()
        dl.name = request.POST.get("name_Dir")
        dl.save()
        return HttpResponseRedirect("/")
def create_file(request):
    if request.method == "POST":
        d = Filee()
        d.name = request.POST.get("name")
        d.capacity = request.POST.get("capacity")
        d.FileeDir = F_Dir.objects.get(name=request.POST.get("FileeDir"))
        d.save()
        return HttpResponseRedirect("/")
    # изменение данных в бд
def edit(request, id):
    try:
        file = Filee.objects.get(id=id)

        if request.method == "POST":
            file.name = request.POST.get("name")
            file.capacity = request.POST.get("capacity")
            file.FileeDir =
F_Dir.objects.get(name=request.POST.get("FileeDir"))
            file.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit.html", {"file": file})
    except Filee.DoesNotExist:
        return HttpResponseNotFound("<h2>file not found</h2>")

def edit_Dir(request, id):
```

```

    try:
        Dir = F_Dir.objects.get(id=id)

        if request.method == "POST":
            Dir.name = request.POST.get("name")
            Dir.save()
            return HttpResponseRedirect("/")
        else:
            return render(request, "edit_Dir.html", {"Dir": Dir})
    except F_Dir.DoesNotExist:
        return HttpResponseRedirect("<h2>Dirrary not found</h2>")
# удаление данных из бд
def delete(request, id):
    try:
        file = Filee.objects.get(id=id)
        file.delete()
        return HttpResponseRedirect("/")
    except Filee.DoesNotExist:
        return HttpResponseRedirect("<h2>file not found</h2>")
def delete_Dir(request, id):
    try:
        Dir = F_Dir.objects.get(id=id)
        Dir.delete()
        return HttpResponseRedirect("/")
    except Filee.DoesNotExist:
        return HttpResponseRedirect("<h2>Dirrary not found</h2>")

```

Файл urls.py

```

from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('create/', views.create),
    path('create_file/', views.create_file),
    path('edit/<int:id>', views.edit),
    path('edit_Dir/<int:id>', views.edit_Dir),
    path('delete/<int:id>', views.delete),
    path('delete_Dir/<int:id>', views.delete_Dir),
]

```

Файл index.html

```

<!DOCTYPE html>

<html>

```

```

<head>

<link rel='stylesheet'
href='https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css' >
  <meta charset="utf-8" />
  <title>рк2</title>
</head >
<body class="container" style="background-color:powderblue; " >
  <div class='group'>
    <form method="POST" action="create_file/" >
      {% csrf_token %}
      <div class="containera" style="float: left;
width: 63%; ">
        <h2>Добавление нового файла</h2>
        <p>
          <label>Введите название файла</label><br>
          <input type="text" name="name" />
        </p>
        <p>
          <label>Введите размер</label><br>
          <input type="text" name="capacity" />
        </p>
        <p>
          <label>Введите название директории</label><br>
          <input type="text" name="FileeDir"/>
        </p>
        <input type="submit" value="Сохранить" >
      </div>
    </form>

    <div class="containerb" style="float: right;
width: 30%; ">
      <form method="POST" action="create/">
        {% csrf_token %}

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

      <h2 >Список директорий</h2>
      <table>
        <tr>
          <th>Id</th>
          <th>Название</th>
        </tr>

```

```

        {% for Dir in Dirs %}
        <tr>
            <td>{{ Dir.id }}</td>
            <td>{{ Dir.name }}</td>
            <td><a href="edit_Dir/{{Dir.id}}">Изменить</a> | <a
href="delete_Dir/{{Dir.id}}">Удалить</a></td>
        </tr>
        {% endfor %}
    </table>
    {% if files.count > 0 %}
    </containerb>
</div>
<h2>Список файлов</h2>
<table>
    <tr>
        <th>Id</th>
        <th>Название</th>
        <th>размер</th>
        <th>Директория</th>
    </tr>
    {% for file in files %}
    <tr>
        <td>{{ file.id }}</td>
        <td>{{ file.name }}</td>
        <td>{{ file.capacity }}</td>
        <td>{{ file.FileeDir.name }}</td>
        <td><a href="edit/{{file.id}}">Изменить</a> | <a
href="delete/{{file.id}}">Удалить</a></td>
    </tr>
    {% endfor %}
</table>
    {% endif %}

    {% for Dir in Dirs %}
    <h3>Файлы в директории {{ Dir.name }}:</h3>
    {% for file in files %}
    {% if Dir.name == file.FileeDir.name %}
    <li>{{file.name}}</li>
    {% endif %}
    {% endfor %}
    {% endfor %}
</form>

</body>

</html>

```

Файл admin.py

```
from django.contrib import admin
```

```
from .models import F_Dir
from .models import Filee

admin.site.register(F_Dir)
admin.site.register(Filee)
```

Файл edit\_dir.html

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

Файл edit.html

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

```

        <p>
            <label>Введите название библиотеки</label>
            <br>
            <input type="text" name="FileeDir"
value="{{files.FileeDir.name}}" />
        </p>
        <input type="submit" value="Сохранить" >
    </form>
</body>
</html>

```

Файл models.py

```

from django.db import models

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

class Filee(models.Model):
    objects = models.Manager()
    name = models.CharField (max_length=50, verbose_name="Название файла")
    capacity = models.CharField (max_length=50, verbose_name="Размер файла")
    FileeDir = models.ForeignKey (F_Dir, on_delete=models.CASCADE, null=True)

```



# Экранные формы с примерами выполнения программы:

### Добавление нового файла

Введите название файла

Введите размер

Введите название директории

### Добавление новой директории

Введите название директории

### Список директорий

Id	Название	
5	Study	<a href="#">Изменить</a>   <a href="#">Удалить</a>
6	Game	<a href="#">Изменить</a>   <a href="#">Удалить</a>
7	Work	<a href="#">Изменить</a>   <a href="#">Удалить</a>
8	Tasks	<a href="#">Изменить</a>   <a href="#">Удалить</a>

### Список файлов

Id	Название	размер	Директория	
2	VS_Code	18000	Study	<a href="#">Изменить</a>   <a href="#">Удалить</a>
3	Doka 2	12000	Game	<a href="#">Изменить</a>   <a href="#">Удалить</a>
4	Titanic.ipynb	128	Study	<a href="#">Изменить</a>   <a href="#">Удалить</a>
5	Отчёт.exc	256	Work	<a href="#">Изменить</a>   <a href="#">Удалить</a>
6	Wow	12321	Game	<a href="#">Изменить</a>   <a href="#">Удалить</a>
7	Project	432	Work	<a href="#">Изменить</a>   <a href="#">Удалить</a>
8	Update.rat	1024	Tasks	<a href="#">Изменить</a>   <a href="#">Удалить</a>

**Файлы в директории Study:**

- VS\_Code
- Titanic.ipynb

**Файлы в директории Game:**

- Doka 2
- Wow

**Файлы в директории Work:**

- Отчёт.exc
- Project

**Файлы в директории Tasks:**

- Update.rat

При нажатии на надпись «Изменить» у строки файла происходит переход на следующую страницу:

←

↺

🌐

127.0.0.1:8000

edit files

Введите название файла

Введите размер файла

Введите название директории

При аналогичном переходе для изменения директории происходит переход на следующую страницу:

←

↺

🌐 127.0.0.1:8000

edit Dir

Введите название директории

Study

Сохранить

Админу доступно изменение бд

Django administration

Home > Main > F\_ dirs

AUTHENTICATION AND AUTHORIZATION

Groups

+ Add

Users

+ Add

MAIN

F\_ dirs

+ Add

Filees

+ Add

Select f\_ dir to change

Action:

▼

Go

0 of 4 selected

☐ F\_ DIR

☐ F\_Dir object (8)

☐ F\_Dir object (7)

☐ F\_Dir object (6)

☐ F\_Dir object (5)

4 f\_ dirs