

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

Отчёт по рубежному контролю №2 по курсу «Разработка интернет
приложений».

Вариант А4.

Выполнил:
Анцифров Н. С.
студент группы ИУ5-51Б

Проверил:
Гапанюк Ю. Е.

Подпись и дата:

Подпись и дата:

Москва, 2021 г.

Содержание:

1. Задание рубежного контроля.....	2
2. Описание объектов классов.....	3
3. Текст программы.....	3
4. Выполнение программы.....	5

1. Задание рубежного контроля.

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.

Вариант задания:

Вариант предметной области – вариант 4.

№ варианта	Класс 1	Класс 2
4	Компьютер	Дисплейный класс

2. Описание объектов классов.

Класс «Компьютер» включает в себя следующие объекты:

- cname – наименование производителя ПК,
- cser – серийный номер конкретной единицы,
- cclassid – ID дисплейного класса, где располагается компьютер.

Класс «Дисплейный класс» включает в себя следующие объекты:

- dnumb – номер помещения дисплейного класса.

3. Текст программы.

urls.py

```
from django.contrib import admin
from django.urls import path
from rk2_app.views import CompDisp

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', CompDisp, name='CompDisps'),
]
```

models.py

```
from django.db import models

# Create your models here.

from django.db import models

class Disp(models.Model):
    dnumb = models.DecimalField(max_digits=5, decimal_places=0,
verbose_name="Номер помещения дисплейного класса")

class Comp(models.Model):
    cname = models.CharField(max_length=20, verbose_name="Наименование
производителя ПК")
    cser = models.CharField(max_length=6, verbose_name="Серийный номер
конкретной единицы")
    cclassid = models.ForeignKey(Disp, on_delete=models.CASCADE,
verbose_name="ID дисплейного класса, где
располагается компьютер")

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

admin.py

```

from django.contrib import admin

# Register your models here.

from .models import Comp, Disp

admin.site.register(Comp)
admin.site.register(Disp)

```

views.py

```

from django.shortcuts import render
from .models import Comp, Disp

# Create your views here.

def CompDisp(request):
    comps = Comp.objects.all()
    disps = Disp.objects.all()
    context = {
        'comps': comps,
        'disps': disps
    }
    return render(request, 'open.html', context)

```

open.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Компьютеры в дисплейных классах</title>
</head>
<body>
<h1>PK2</h1>
<h2>Список компьютеров в дисплейных классах</h2>
    {%for item in disps%}
    <ul>
        <li>Список компьютеров в аудитории {{ item.dnumb }}<br>
            <ol>
                {%for its in comps%}
                {%if its.cclassid == item%}
                <li>{{ its.cname }} {{ its.cser }} <br>
                </li>
                {% endif %}
                {% endfor %}
            </ol>
        </li>
    </ul>
    {% endfor %}
</body>
</html>

```

4. Выполнение программы.

После написания кода и осуществления миграций создадим суперпользователя для доступа к панели администратора и перейдем в неё:

The first screenshot shows the Django administration site's main interface. The header includes the title 'Django administration' and user information 'WELCOME, FIRRY' with links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The left sidebar lists site administration categories: 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'RK2_APP' (Comps, Disps). The 'Comps' link is highlighted. The right sidebar shows 'Recent actions' and 'My actions' (None available).

The second screenshot shows the 'Add comp' form. The breadcrumb trail is 'Home > Rk2_App > Comps > Add comp'. The left sidebar is the same as the first screenshot. The form fields are: 'Наименование производителя ПК:' with value 'HP', 'Серийный номер конкретной единицы:' with value '1AAA01', and 'ID дисплейного класса, где располагается компьютер:' with a dropdown showing 'Disp object (1)'. At the bottom are buttons for 'Save and add another', 'Save and continue editing', and 'SAVE'.

The third screenshot shows the 'Select comp to change' view. The breadcrumb trail is 'Home > Rk2_App > Comps'. The left sidebar is the same. The main area shows a list of computer components with checkboxes: 'COMP', '5EEE05', '4DD04', '3CCC03', '2BBB02', and '1AAA01'. Below the list, it says '5 comps'. At the top right is an 'ADD COMP +' button. The 'Action:' dropdown is set to '-----' and 'Go' is visible, with '0 of 5 selected'.

The fourth screenshot shows the 'Select disp to change' view. The breadcrumb trail is 'Home > Rk2_App > Disps'. The left sidebar is the same. The main area shows a list of display objects with checkboxes: 'DISP', 'Disp object (5)', 'Disp object (4)', 'Disp object (3)', 'Disp object (2)', and 'Disp object (1)'. Below the list, it says '5 disps'. At the top right is an 'ADD DISP +' button. The 'Action:' dropdown is set to '-----' and 'Go' is visible, with '0 of 5 selected'.

Зайдем на сервер и проверим вывод:

← → ↻ ⓘ 127.0.0.1:8000

PK2

Список компьютеров в дисплейных классах

- Список компьютеров в аудитории 404
 1. HP 1AAA01
 2. DELL 2BBB02
- Список компьютеров в аудитории 113
 1. TOSHIBA 4DDD04
- Список компьютеров в аудитории 41
- Список компьютеров в аудитории 362
 1. DELL 3CCC03
- Список компьютеров в аудитории 511
 1. ASUS 5EEE05