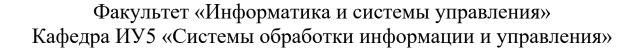
# Московский государственный технический университет им. Н.Э. Баумана



Курс «Разработка интернет приложений»

Рубежный контроль №2

Выполнил:

студент группы ИУ5-52Б Ваганов Даниил Проверил:

преподаватель каф. ИУ5 Гапанюк Ю.Е.

#### 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 DisplayClass(models.Model):
    disp_id = models.AutoField('disp_id', primary_key=True)
    room_number = models.CharField(max length=50)
    square = models.IntegerField(verbose_name='Square')
    def __str__(self):
        return f'{self.room_number}{self.square}'
class Computer(models.Model):
    comp_id = models.AutoField('comp_id', primary_key=True)
    cpu = models.CharField(max_length=80,verbose_name='CPU')
    graphic_card = models.CharField(max_length=80, verbose_name='Graphic
Card<sup>1</sup>)
    disp id = models.ForeignKey('DisplayClass', models.DO NOTHING,
db_column='disp_id', verbose_name='Display Class')
urls.pv:
from django.urls import path
from application import views
urlpatterns = [
    path('', views.index),
    path('class', views.dispclass_list),
    path('class/create', views.DisplayClassCreate.as_view()),
```

```
path('class/<int:disp_id>/update',
views.DisplayClassUpdate.as_view(), name='class_update'),
    path('class/<int:disp_id>/delete',
views.DisplayClassDelete.as_view(), name='class_delete'),
    path('computer', views.computers_list),
    path('computer/create', views.ComputerCreate.as_view()),
    path('computer/<int:comp_id>/update',
views.ComputerUpdate.as_view(), name='computer_update'),
    path('computer/<int:comp_id>/delete',
views.ComputerDelete.as_view(), name='computer_delete'),
    path('report', views.report)
1
views.py:
from django import forms
from django.shortcuts import render
from django.views.generic.edit import CreateView, UpdateView, DeleteView
from .models import Computer, DisplayClass
def index(request):
  return render(request, 'index.html')
def report(request):
  computers = Computer.objects.all()
  params = {'computers': computers}
  return render(request, 'report.html', params)
def dispclass_list(request):
  classes = DisplayClass.objects.all().values()
  params = {'entity': 'DisplayClass', 'objects': classes}
  return render(request, 'list.html', params)
def computers_list(request):
  computers = Computer.objects.all().values()
  params = {'entity': 'Computer', 'objects': computers}
  return render(request, 'list.html', params)
class DisplayClassCreate(CreateView):
  model = DisplayClass
  fields = ['room_number', 'square']
  success url = '/class'
  template_name = 'class_form.html'
class DisplayClassUpdate(UpdateView):
  model = DisplayClass
  fields = ['room_number', 'square']
  pk_url_kwarg = 'disp_id'
  success_url = '/class'
  template name = 'class form.html'
class DisplayClassDelete(DeleteView):
  model = DisplayClass
  pk_url_kwarg = 'disp_id'
  success url = '/class'
```

```
template_name = 'class_delete_form.html'
class ComputerCreate(CreateView):
  model = Computer
  fields = ['cpu', 'graphic_card', 'disp_id']
  success_url = '/computer'
  template name = 'computer form.html'
 def get context data(self, **kwargs):
    context = super(ComputerCreate, self).get context data(**kwargs)
    context['form'].fields['disp_id'] =
forms.ModelChoiceField(queryset=DisplayClass.objects.all(),
empty_label=None, label='Дисплейный класс')
    return context
class ComputerUpdate(UpdateView):
  model = Computer
  fields = ['cpu', 'graphic_card', 'disp_id']
 pk_url_kwarg = 'comp_id'
  success_url = '/computer'
  template_name = 'computer_form.html'
 def get_context_data(self, **kwargs):
    context = super(ComputerUpdate, self).get_context_data(**kwargs)
    context['form'].fields['disp_id'] =
forms.ModelChoiceField(queryset=DisplayClass.objects.all(),
empty_label=None, label='Дисплейный класс')
    return context
class ComputerDelete(DeleteView):
  model = Computer
 pk_url_kwarg = 'comp_id'
  success url = '/computer'
  template_name = 'computer_delete form.html'
base.html:
<!DOCTYPE html>
<html lang="en" class="h-100">
  <head>
    <meta charset="utf-8" />
    <title>{% block title %}RK2{% endblock %}</title>
    <style type="text/css"></style>
  </head>
  <body>
    <div>
      <a style="margin: 5px" href="/">Main Page</a>
      <a style="margin: 5px" href="/computer">Computers</a>
      <a style="margin: 5px" href="/class">Display Classes</a>
      <a style="margin: 5px" href="/report">Report</a>
    </div>
    {% block content %} {% endblock %}
  </body>
</html>
```

```
index.html:
{% extends 'base.html' %}
{% block title %}
Main Page
{% endblock %}
{% block content%}
<h2>Main Page</h2>
<span>RK2</span>
<span>Vaganov Daniil IU5-52B</span>
{% endblock %}
report.html:
{% extends 'base.html' %}
{% block title %}Computers{% endblock %}
{%block content %}
<h2>List of all computers:</h2>
ul>
  {% for comp in computers %}
    <
      <div>
        <b>CPU:</b> {{comp.cpu}}<br>
        <b>Graphic Card:</b> {{comp.graphic_card}}<br>
        <b>Display Class:</b> {{comp.disp id.room number}}<br><br>
      </div>
    {% endfor %}
{% endblock %}
computer form.html:
{% extends 'base.html' %}
{% block title %}Update{% endblock %}
{%block content %}
<h2>Update computer's properties</h2>
<form action='' method='post'>
  {% csrf_token %}
  {{form.as_table}}
  <input type="submit" value="Update"/>
</form>
{% endblock %}
computer delete form.html:
{% extends 'base.html' %}
{% block title %}Update{% endblock %}
{%block content %}
<h2>Delete data</h2>
Are you sure you want to delete computer {{object.comp_id}}?<form action='' method='post'>
  {% csrf token %}
```

```
<input type="submit" value="Delete" />
</form>
{% endblock %}
class form.html:
{% extends 'base.html' %}
{% block title %}Update{% endblock %}
{%block content %}
<h2>Update data</h2>
<form action='' method='post'>
  {% csrf_token %}
  {{form.as_table}}
  <input type="submit" value="Update"/>
</form>
{% endblock %}
class_delete_form.html:
{% extends 'base.html' %}
{% block title %}Update{% endblock %}
{%block content %}
<h2>Delete data</h2>
Are you sure you want to delete {{object.room_number}}?
<form action='' method='post'>
  {% csrf_token %}
  <input type="submit" value="Delete" />
</form>
{% endblock %}
```

#### Скриншоты:

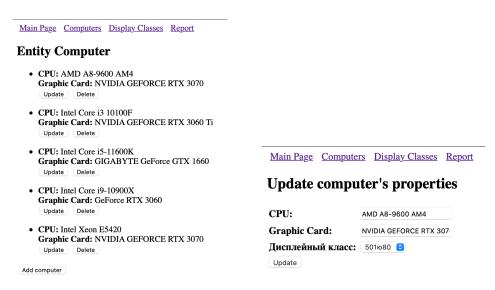
#### Главная страница:

Main Page Computers Display Classes Report

## **Main Page**

RK2 Vaganov Daniil IU5-52B

#### Компьютеры:



#### Дисплейные классы:

Main Page Computers Display Classes Report

#### **Entity DisplayClass**

Class number: 501ю	
Square: $80\text{m}^2$	
Update Delete	
Class number: 4049	Main
Square: 65m <sup>2</sup>	
Update Delete	Upd
Class number: 306	-
Square: $40\text{m}^2$	Room
Update Delete	
	Sq
class	Updat

Main Page Computers Display Classes Report

### Update data

<b>Room number:</b>	501ю	
Square:	80	•
Update		

#### Полная информация о компьютерах:

Main Page Computers Display Classes Report

## List of all computers:

• **CPU:** AMD A8-9600 AM4

**Graphic Card:** NVIDIA GEFORCE RTX 3070

Display Class: 501ю

• **CPU:** Intel Core i3 10100F

Graphic Card: NVIDIA GEFORCE RTX 3060 Ti

Display Class: 501ю

• **CPU:** Intel Core i5-11600K

**Graphic Card:** GIGABYTE GeForce GTX 1660

Display Class: 501ю

• CPU: Intel Core i9-10900X

**Graphic Card:** GeForce RTX 3060

Display Class: 4049

• **CPU:** Intel Xeon E5420

**Graphic Card:** NVIDIA GEFORCE RTX 3070

**Display Class: 306**