**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 |
| Ваганов Даниил |  | 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 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')

disp\_id = models.ForeignKey('DisplayClass', models.DO\_NOTHING, db\_column='disp\_id', verbose\_name='Display Class')

***urls.py:***

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)

]

***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 %}

<li>

<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>

</li>

{% endfor %}

</ul>

{% 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 %}

<table style="text-align: left;">

{{form.as\_table}}

</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>

<p>Are you sure you want to delete computer {{object.comp\_id}}?</p>

<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 %}

<table>

{{form.as\_table}}

</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>

<p>Are you sure you want to delete {{object.room\_number}}?</p>

<form action='' method='post'>

{% csrf\_token %}

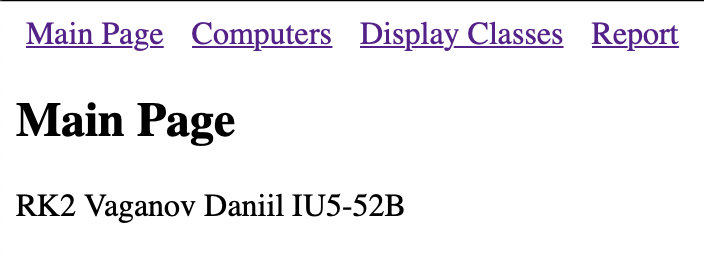
<input type="submit" value="Delete" />

</form>

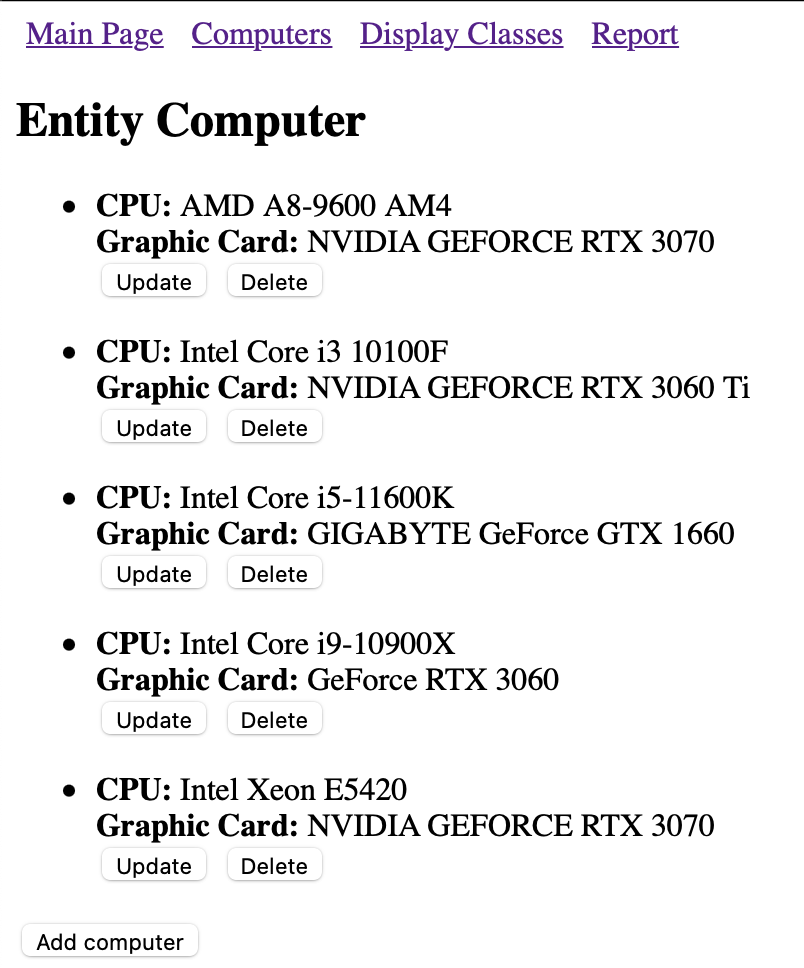
{% endblock %}

**Скриншоты:**

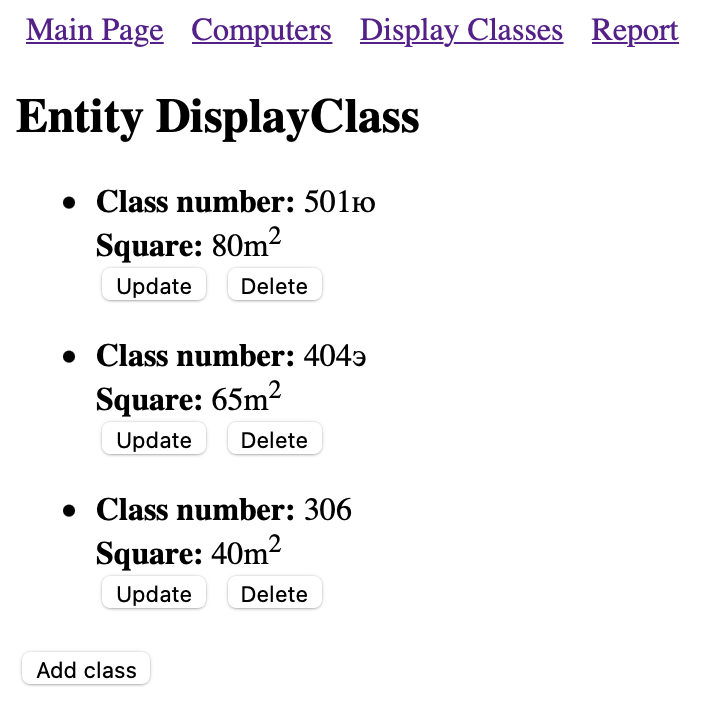
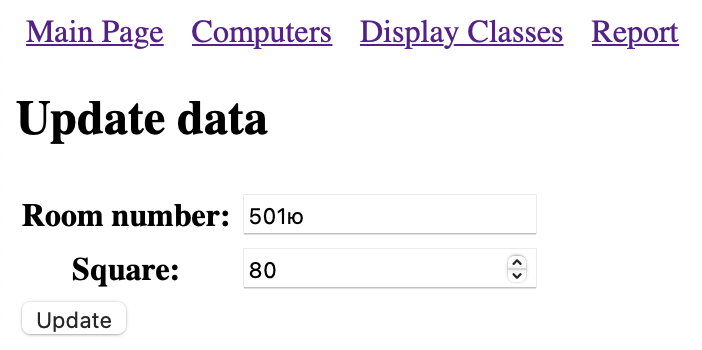
Главная страница:

****

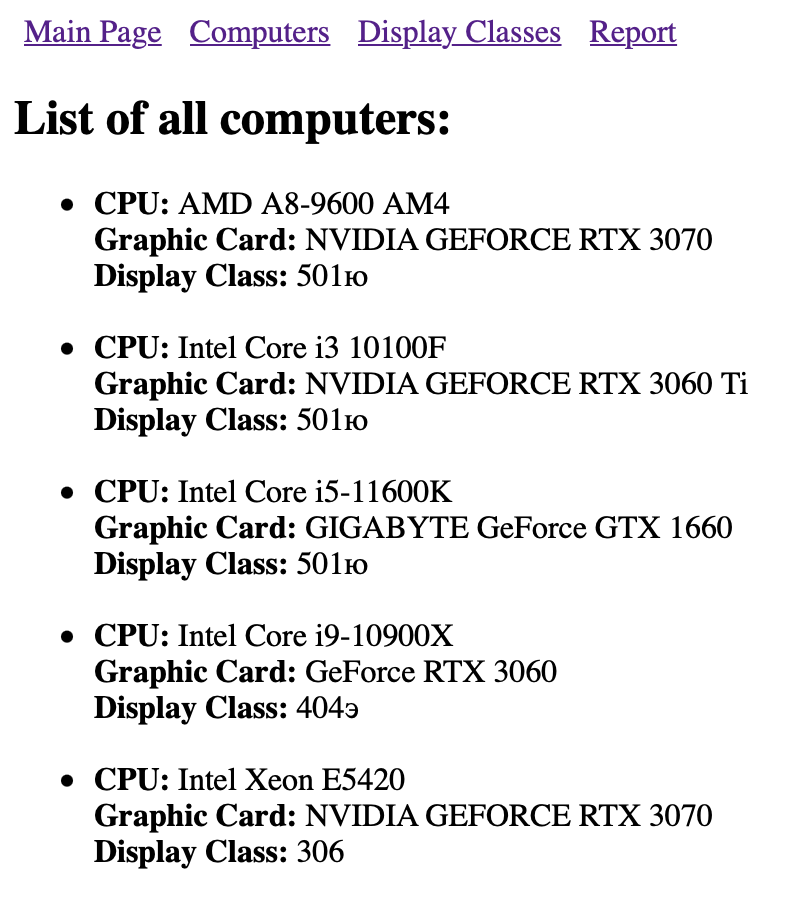
Компьютеры:

** **

Дисплейные классы:

** **

Полная информация о компьютерах:

****