

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | **«Информатики и систем управления»** |
| КАФЕДРА | ИУ5 |

Дисциплина «Разработка интернет-приложений»

Отчет по домашнему заданию

|  |  |  |
| --- | --- | --- |
| Студент | группы ИУ5-52Б | Гришин Илья |
| Преподаватель |  | Гапанюк Ю.Е. |

**Цель домашнего задания:** изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.

1. **Стандартное задание:**

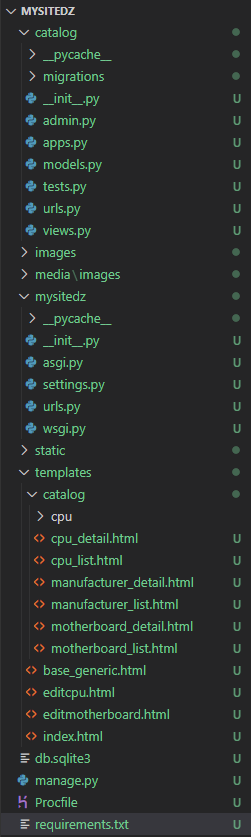
Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

1. Две модели, связанные отношением один-ко-многим.
2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фрейворк по желанию студента.

**Выполненные расширенные задания, добавляющие +1 балл на экзамене:**

1. Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku).
2. Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, <https://c3js.org/>) и развертывание приложения на облачном сервисе (например, heroku).

**2. Структура проекта:**



**3. Текст программы**

catalog/admin.py

from django.contrib import admin

from .models import Manufacturer, CPU, Motherboard

admin.site.register(Motherboard)

admin.site.register(CPU)

admin.site.register(Manufacturer)

catalog/apps.py

from django.apps import AppConfig

class CatalogConfig(AppConfig):

    name = 'catalog'

catalog/models.py

from django.db import models

import datetime

# Create your models here.

class CPU(models.Model):

    YEAR\_CHOICES = []

    for r in range(1980, (datetime.datetime.now().year+1)):

        YEAR\_CHOICES.append((r,r))

    name = models.CharField('Название процессора', max\_length=50, help\_text="Введите название процессора")

    price = models.CharField('Цена процессора', max\_length=20,  null=True, help\_text="Введите цену процессора")

    generation = models.CharField('Поколение процессора', max\_length=50, null=True, help\_text="Введите поколение процессора")

    yearrelease = models.IntegerField('Год релиза', choices=YEAR\_CHOICES, null=True, help\_text="Выберите год релиза процессора")

    socket = models.CharField('Сокет',max\_length=20,  help\_text="Введите сокет")

    numcore = models.IntegerField('Количество ядер', null=True,help\_text="Введите количество ядер")

    numstream = models.IntegerField('Количество потоков', null=True,help\_text="Введите количество потоков")

    cacheL1 = models.CharField('Кэш L1',max\_length=20, null=True, help\_text="Введите объем кэша L1")

    cacheL2 = models.CharField('Кэш L2',max\_length=20, null=True, help\_text="Введите объем кэша L2")

    cacheL3 = models.CharField('Кэш L3',max\_length=20, null=True, help\_text="Введите объем кэша L3")

    basefreq = models.CharField('Базовая частота процессора',max\_length=20, null=True, help\_text="Введите базовую частоту процессора")

    maxfreq = models.CharField('Максимальная частота процессора',max\_length=20, null=True, help\_text="Введите максимальную частоту процессора")

    countrate = models.IntegerField('Количество оценок',null=True)

    rate = models.IntegerField('Общая оценка',null=True)

    avrrate = models.FloatField('Средняя оценка',null=True)

    cpu\_img = models.ImageField('Изображение',upload\_to='images/', blank=True, null=True, help\_text="Загрузите изображение")

    def get\_absolute\_url(self):

        return reverse('cpu-detail', args=[str(self.id)])

    def \_\_str\_\_(self):

          return '%s, %s' % (self.name, self.socket)

    @property

    def cpu\_img\_url(self):

        if self.cpu\_img and hasattr(self.cpu\_img, 'url'):

            return self.cpu\_img.url

    class Meta:

        verbose\_name = 'Процессор'

        verbose\_name\_plural = 'Процессоры'

class Manufacturer(models.Model):

    namecompany = models.CharField('Название производителя', max\_length=100, help\_text="Введите название производителя")

    description = models.TextField('Описание',max\_length=1000, null=True, help\_text="Введите описание")

    urlman = models.URLField('URL', null=True, help\_text="Введите URL")

    manufacturer\_img = models.ImageField('Изображение',upload\_to='images/', blank=True, null=True, help\_text="Загрузите изображение (рекомендуется 1200x300)")

    def get\_absolute\_url(self):

        return reverse('manufacturer-detail', args=[str(self.id)])

    def \_\_str\_\_(self):

        return self.namecompany

    @property

    def manufacturer\_img\_url(self):

        if self.manufacturer\_img and hasattr(self.manufacturer\_img, 'url'):

            return self.manufacturer\_img.url

    class Meta:

        verbose\_name = 'Производитель'

        verbose\_name\_plural = 'Производители'

from django.urls import reverse

class Motherboard(models.Model):

    YEAR\_CHOICES = []

    for r in range(1980, (datetime.datetime.now().year+1)):

        YEAR\_CHOICES.append((r,r))

    title = models.CharField('Название материнской платы', max\_length=50, help\_text="Введите название материнской платы")

    price = models.CharField('Цена материнской платы', max\_length=20, null=True, help\_text="Введите цену материнской платы")

    yearrelease = models.IntegerField('Год релиза', choices=YEAR\_CHOICES, null=True, help\_text="Выберите год релиза материнской платы")

    formfactor = models.CharField('Форм-фактор',max\_length=50, help\_text="Введите форм фактор")

    typememory = models.CharField('Тип памяти',max\_length=20, help\_text="Введите тип памяти")

    socket = models.CharField('Сокет',max\_length=20,  help\_text="Введите сокет")

    countrate = models.IntegerField('Количество оценок',null=True)

    rate = models.IntegerField('Общая оценка',null=True)

    avrrate = models.FloatField('Средняя оценка',null=True)

    motherboard\_img = models.ImageField('Изображение',upload\_to='images/', blank=True, null=True, help\_text="Загрузите изображение")

    cpu = models.ManyToManyField(CPU, help\_text="Выберите подходящий процессор", verbose\_name='Подходящие процессоры')

    manufacturer = models.ForeignKey('Manufacturer', on\_delete=models.SET\_NULL, null=True, verbose\_name='Производитель')

    def \_\_str\_\_(self):

        return self.title

    def get\_absolute\_url(self):

        return reverse('motherboard-detail', args=[str(self.id)])

    @property

    def motherboard\_img\_url(self):

        if self.motherboard\_img and hasattr(self.motherboard\_img, 'url'):

            return self.motherboard\_img.url

    class Meta:

        verbose\_name = 'Материнская плата'

        verbose\_name\_plural = 'Материнские платы'

catalog/urls.py

from django.urls import path

from . import views

from django.conf.urls import url

urlpatterns = [

    path('', views.index, name='index'),

    path('motherboards/', views.MotherboardListView.as\_view(), name='motherboards'),

    path('motherboard/<int:pk>', views.MotherboardDetailView.as\_view(), name='motherboard-detail'),

    path('cpus/', views.CPUListView.as\_view(), name='cpus'),

    path('cpu/<int:pk>',views.CPUDetailView.as\_view(), name='cpu-detail'),

    path('manufacturers/', views.ManufacturerListView.as\_view(), name='manufacturers'),

    path('manufacturer/<int:pk>',views.ManufacturerDetailView.as\_view(), name='manufacturer-detail'),

    # path('create/', views.create),

    path('cpu/editcpu/<int:id>/', views.editcpu),

    path('motherboard/editmotherboard/<int:id>/', views.editmotherboard),

]

catalog/views.py

from django.shortcuts import render

from django.http import HttpResponseRedirect

from django.http import HttpResponseNotFound

from django.shortcuts import get\_object\_or\_404

from django.http import HttpResponse

from .models import Motherboard, CPU, Manufacturer

import django.template.context\_processors

def index(request):

    return render(request, "index.html")

def editcpu(request, id):

    try:

        cpu = CPU.objects.get(id=id)

        if request.method == "POST":

            s = request.POST.get("rate")

            a = s[1:2]

            cpu.rate = cpu.rate+int(a)

            cpu.countrate += 1

            cpu.avrrate = round(cpu.rate/cpu.countrate, 2)

            cpu.save()

            return HttpResponseRedirect("/catalog/cpus")

        else:

            return render(request, "editcpu.html", {"cpu": cpu})

    except CPU.DoesNotExist:

        return HttpResponseNotFound("<h2>CPU not found</h2>")

def editmotherboard(request, id):

    try:

        motherboard = Motherboard.objects.get(id=id)

        if request.method == "POST":

            s = request.POST.get("rate")

            a = s[1:2]

            motherboard.rate = motherboard.rate+int(a)

            motherboard.countrate += 1

            motherboard.avrrate = round(motherboard.rate/motherboard.countrate, 2)

            motherboard.save()

            return HttpResponseRedirect("/catalog/motherboards")

        else:

            return render(request, "editmotherboard.html", {"motherboard": motherboard})

    except Motherboard.DoesNotExist:

        return HttpResponseNotFound("<h2>Motherboard not found</h2>")

from django.views import generic

class MotherboardListView(generic.ListView):

    model = Motherboard

class MotherboardDetailView(generic.DetailView):

    model = Motherboard

    paginate\_by = 3

class CPUListView(generic.ListView):

    model = CPU

    paginate\_by = 2

class CPUDetailView(generic.DetailView):

    model = CPU

class ManufacturerListView(generic.ListView):

    model = Manufacturer

    paginate\_by = 6

class ManufacturerDetailView(generic.DetailView):

    model = Manufacturer

mysitedz/settings.py

from pathlib import Path

import os

BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent

SECRET\_KEY = 'gg)qxyw+szroh\*#-b8-y28\_f1x$ie#3&=e^owrdqtzak6k7g57'

DEBUG = True

ALLOWED\_HOSTS = ['thawing-coast-94757.herokuapp.com', '127.0.0.1']

INSTALLED\_APPS = [

    'django.contrib.admin',

    'django.contrib.auth',

    'django.contrib.contenttypes',

    'django.contrib.sessions',

    'django.contrib.messages',

    'django.contrib.staticfiles',

    'catalog.apps.CatalogConfig',

]

MIDDLEWARE = [

    'django.middleware.security.SecurityMiddleware',

    'django.contrib.sessions.middleware.SessionMiddleware',

    'django.middleware.common.CommonMiddleware',

    'django.middleware.csrf.CsrfViewMiddleware',

    'django.contrib.auth.middleware.AuthenticationMiddleware',

    'django.contrib.messages.middleware.MessageMiddleware',

    'django.middleware.clickjacking.XFrameOptionsMiddleware',

]

ROOT\_URLCONF = 'mysitedz.urls'

TEMPLATES = [

    {

        'BACKEND': 'django.template.backends.django.DjangoTemplates',

        'DIRS': [os.path.join(BASE\_DIR, 'templates')],

        'APP\_DIRS': True,

        'OPTIONS': {

            'context\_processors': [

                'django.template.context\_processors.debug',

                'django.template.context\_processors.request',

                'django.contrib.auth.context\_processors.auth',

                'django.contrib.messages.context\_processors.messages',

            ],

        },

    },

]

WSGI\_APPLICATION = 'mysitedz.wsgi.application'

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.sqlite3',

        'NAME': BASE\_DIR / 'db.sqlite3',

    }

}

AUTH\_PASSWORD\_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',

    },

]

LANGUAGE\_CODE = 'ru'

TIME\_ZONE = 'Europe/Moscow'

USE\_I18N = True

USE\_L10N = True

USE\_TZ = True

STATIC\_URL = '/static/'

STATIC\_ROOT = os.path.join(BASE\_DIR, "staticfiles")

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media/')

MEDIA\_URL = '/media/'

mysitedz/urls.py

from django.urls import path

from django.contrib import admin

urlpatterns = [

    path('admin/', admin.site.urls),

]

from django.urls import include

urlpatterns += [

     path('catalog/', include('catalog.urls')),

]

from django.views.generic import RedirectView

urlpatterns += [

    path('', RedirectView.as\_view(url='/catalog/', permanent=True)),

]

from django.conf import settings

from django.conf.urls.static import static

urlpatterns += static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)

urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

templates/base\_generic.html

<!DOCTYPE html>

<html lang="ru">

<head>

  {% block title %}<title>ДЗ</title>{% endblock %}

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">

  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="anonymous"></script>

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</head>

<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark" aria-label="Ninth navbar example">

        <div class="container-xl">

        {% load static %}

        <a class="navbar-brand" href="{% url 'index' %}">Домашнее задание</a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarsExample07XL" aria-controls="navbarsExample07XL" aria-expanded="false" aria-label="Toggle navigation">

            <span class="navbar-toggler-icon"></span>

        </button>

        <div class="collapse navbar-collapse" id="navbarsExample07XL">

            <ul class="navbar-nav me-auto mb-2 mb-lg-0">

            <li class="nav-item active">

                <a class="nav-link" aria-current="page" href="{% url 'index' %}">Главная</a>

            </li>

            <li class="nav-item active">

                <a class="nav-link" aria-current="page" href="{% url 'motherboards' %}">Материнские платы</a>

            </li>

            <li class="nav-item active">

                <a class="nav-link" aria-current="page" href="{% url 'cpus' %}">Процессоры</a>

            </li>

            <li class="nav-item active">

                <a class="nav-link" aria-current="page" href="{% url 'manufacturers' %}">Производители</a>

            </li>

            </ul>

        </div>

        </div>

    </nav>

    <div class="container-xl mb-4">

    {% block content %}{% endblock %}

    {% block pagination %}

    {% if is\_paginated %}

    <div class="bd-example">

        <nav aria-label="Standard pagination example">

            <nav aria-label="Статьи по Bootstrap 4">

                <ul class="pagination justify-content-center">

            <li class="page-item">

                {% if page\_obj.has\_previous %}

              <a class="page-link" href="{{ request.path }}?page={{ page\_obj.previous\_page\_number }}" aria-label="Previous">

                <span aria-hidden="true">&laquo;</span>

              </a>

              {% endif %}

            </li>

            <button type="button " class="btn btn-light" disabled>Страница {{ page\_obj.number }} из {{ page\_obj.paginator.num\_pages }}</button>

            <li class="page-item">

                {% if page\_obj.has\_next %}

              <a class="page-link" href="{{ request.path }}?page={{ page\_obj.next\_page\_number }}" aria-label="Next">

                <span aria-hidden="true">&raquo;</span>

              </a>

              {% endif %}

            </li>

          </ul>

        </nav>

        </div>

    {% endif %}

  {% endblock %}

  <footer class="text-muted py-5">

    <div class="container">

      <p class="mb-1">Автор работы: Гришин Илья ИУ5-52Б</p>

      <p class="mb-0"><a href="https://github.com/Exepez/RIP-Labs">GitHub</a></p>

    </div>

  </footer>

</body>

</html>

templates/index.html

{% extends "base\_generic.html" %}

{% block content %}

{% load static %}

<h1>Домашнее задание</h1>

<p><strong>Цель домашнего задания:</strong> изучение возможностей создания прототипа веб-приложения на основе базы данных с использованием фреймворка Django.</p>

<h3>Стандартное задание:</h3>

<p>Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:</p>

<ol>

<li>Две модели, связанные отношением один-ко-многим.</li>

<li>Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.</li>

<li>Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.</li>

<li>Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фрейворк по желанию студента.</li>

</ol>

<h3>Выполненные расширенные задания, добавляющие +1 балл на экзамене:</h3>

<ol>

<li>Реализация связи много-ко-многим (с возможностью редактирования данных в пользовательском интерфейсе) и развертывание приложения на облачном сервисе (например, heroku).</li>

<li>Реализация в отчете (пункт 3 стандартного задания) графика на основе данных отчета с использованием библиотек JavaScript (например, <a href="https://c3js.org/" rel="nofollow">https://c3js.org/</a>)   и развертывание приложения на облачном сервисе (например, heroku).</li>

</ol>

{% endblock %}

templates/editcpu.html

{% extends "base\_generic.html" %}

{% block content %}

{% load static %}

<h1>Оценка товара:</h1>

<form method="POST">

    {% csrf\_token %}

<p>

    <h3>Оцениваемый товар: {{cpu.name}}</h3>

</p>

    <div class="container">

        <div class="row">

          <div class="col">

            <label for="exampleFormControlSelect1">Выберите оценку:</label>

            <select class="form-control" id="exampleFormControlSelect1" name="rate">

              <option>(0)</option>

              <option>(1) ✩ </option>

              <option>(2) ✩✩ </option>

              <option>(3) ✩✩✩ </option>

              <option>(4) ✩✩✩✩  </option>

              <option>(5) ✩✩✩✩✩  </option>

            </select>

            <p></p>

            <input type="submit" value="Оценить" >

          </div>

          <div class="col"><h3>Текущая оценка товара:</h3><div id='gaugeArea'>div></div>

          <div class="col"></div>

          <div class="col"></div>

    </div>

</form>

<script src="https://unpkg.com/gauge-chart@latest/dist/bundle.js"></script>

<script>

  let element = document.querySelector('#gaugeArea')

  let chartWidth = 420

  let needleValue = 1

  let str1 = '{{cpu.avrrate}}'

  var k = 2

  var d = 100

  if (str1.length==3)

      { k = 1; d=10}

  let a = str1.substr(2, k)

  let a1 = str1.substr(0, 1)

  var b = Number(a);

  var b1 = Number(a1);

  let gaugeOptions = {

    hasNeedle: true,

    outerNeedle: true,

    needleColor: "black",

    needleStartValue: 0,

    arcColors: ["rgb(224,28,6)","rgb(224,144,6)","rgb(213,224,6)","rgb(112,224,6)","rgb(10,169,17)"],

    arcDelimiters: [20,40,60,80],

    rangeLabel: ["0","5"],

    rangeLabelFontSize: 0,

    arcLabels: ['1', '2', '3', '4'],

    centralLabel: str1,

    arcPadding: 3,

    arcPaddingColor: 'white',

    }

  GaugeChart.gaugeChart(element, chartWidth, gaugeOptions).updateNeedle((b1+b/d)\*20)

</script>

{% endblock %}

templates/editmotherboard.html

{% extends "base\_generic.html" %}

{% block content %}

{% load static %}

<h1>Оценка товара:</h1>

<form method="POST">

    {% csrf\_token %}

<p>

    <h3>Оцениваемый товар: {{motherboard.title}}</h3>

</p>

    <div class="container">

        <div class="row">

          <div class="col">

            <label for="exampleFormControlSelect1">Выберите оценку:</label>

            <select class="form-control" id="exampleFormControlSelect1" name="rate">

              <option>(0)</option>

              <option>(1) ✩ </option>

              <option>(2) ✩✩ </option>

              <option>(3) ✩✩✩ </option>

              <option>(4) ✩✩✩✩  </option>

              <option>(5) ✩✩✩✩✩  </option>

            </select>

            <p></p>

            <input type="submit" value="Оценить" >

          </div>

          <div class="col"><h3>Текущая оценка товара:</h3><div id='gaugeArea'</div></div>

          <div class="col"></div>

          <div class="col"></div>

    </div>

</form>

<script src="https://unpkg.com/gauge-chart@latest/dist/bundle.js"></script>

<script>

  let element = document.querySelector('#gaugeArea')

  let chartWidth = 420

  let needleValue = 1

  let str1 = '{{motherboard.avrrate}}'

  var k = 2

  var d = 100

  if (str1.length==3)

      { k = 1; d=10}

  let a = str1.substr(2, k)

  let a1 = str1.substr(0, 1)

  var b = Number(a);

  var b1 = Number(a1);

  let gaugeOptions = {

    hasNeedle: true,

    outerNeedle: true,

    needleColor: "black",

    needleStartValue: 0,

    arcColors: ["rgb(224,28,6)","rgb(224,144,6)","rgb(213,224,6)","rgb(112,224,6)","rgb(10,169,17)"],

    arcDelimiters: [20,40,60,80],

    rangeLabel: ["0","5"],

    rangeLabelFontSize: 0,

    arcLabels: ['1', '2', '3', '4'],

    centralLabel: str1,

    arcPadding: 3,

    arcPaddingColor: 'white',

    }

  GaugeChart.gaugeChart(element, chartWidth, gaugeOptions).updateNeedle((b1+b/d)\*20)

</script>

{% endblock %}

templates/catalog/cpu\_list.html

{% extends "base\_generic.html" %}

{% block content %}

    <h1>Список процессоров</h1>

    {% if cpu\_list %}

      {% for copy in cpu\_list.all %}

      <p></p>

      <div class="col">

        <div class="card">

          <div class="row g-0">

            <div class="col-md-4">

              <svg class="bd-placeholder-img" width="25%" height="150" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

                <img src="{{ MEDIA\_URL }}{{ copy.cpu\_img\_url }}" width="200" height="200"class="rounded">

              </svg>

            </div>

            <div class="col-md-8">

              <div class="card-body">

                <div class="container">

                  <div class="row">

                    <div class="col">

                      <h5 class="card-title"><a href="{{ copy.get\_absolute\_url }}">{{ copy.name }}</a></h5>

                      <p class="card-text"><small class="text-muted">{{copy.socket}}</small></p>

                    </div>

                    <div class="col">

                      <div class="col-md-8">

                          <div class="card-body">

                            <h5 class="card-text"><class="text-muted">{{copy.price}}</class></h5>

                          </div>

                        </div>

                    </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      {% endfor %}

    {% else %}

      <p>There are no cpus.</p>

    {% endif %}

{% endblock %}

templates/catalog/cpu\_detail.html

{% extends "base\_generic.html" %}

{% block content %}

{% load static %}

  <h1>Процессор {{ cpu.name }}</h1>

<div class="container">

    <div class="row">

      <div class="col">

            <img src="{{ MEDIA\_URL }}{{ cpu.cpu\_img\_url }}" width="400" height="400"class="rounded">

      </div>

      <div class="col">

        <h5 class="card-title">Всего оценок: {{ cpu.countrate}}</h5>

        <div id='gaugeArea'><!-- Plotly chart will be drawn inside this DIV --></div>

     </div>

      <div class="col">

        <div class="col-md-8">

            <div class="card-body">

                  <h1 class="card-title">{{ cpu.price }}</h1>

                  <h3 class="card-title">Оценка товара: {{ cpu.avrrate}} из 5</h3>

                  <button type="button" class="btn btn-warning" onclick="window.location.href = 'editcpu/{{cpu.id}}';">Оценить</button>

            </div>

          </div>

      </div>

</div>

<table class="table">

    <thead>

      <tr>

        <th scope="col"><h2>Характеристики</h2></th>

        <th scope="col"></th>

      </tr>

    </thead>

    <tbody>

     <tr class="table-dark">

        <th colspan="2">Общие параметры</th>

      </tr>

      <tr>

        <td scope="row">Модель</td>

        <td>{{ cpu.name }}</td>

      </tr>

      <tr>

        <td scope="row">Поколение процессоров</td>

        <td>{{ cpu.generation }}</td>

      </tr>

      <tr>

        <td scope="row">Год релиза</td>

        <td>{{ cpu.yearrelease }}</td>

      </tr>

      <tr>

        <td scope="row">Сокет</td>

        <td>{{ cpu.socket }}</td>

      </tr>

      <tr class="table-dark">

        <th colspan="2">Ядро и архитектура</th>

      </tr>

      <tr>

        <td scope="row">Количество ядер</td>

        <td>{{ cpu.numcore }}</td>

      </tr>

      <tr>

        <td scope="row">Максимальное число потоков</td>

        <td>{{ cpu.numstream }}</td>

      </tr>

      <tr>

        <td scope="row">Кэш L1</td>

        <td>{{ cpu.cacheL1 }}</td>

      </tr>

      <tr>

        <td scope="row">Кэш L2</td>

        <td>{{ cpu.cacheL2 }}</td>

      </tr>

      <tr>

        <td scope="row">Кэш L3</td>

        <td>{{ cpu.cacheL3 }}</td>

      </tr>

      <tr>

      <tr class="table-dark">

        <th colspan="2">Частота</th>

      </tr>

      <tr>

        <td scope="row">Базовая частота процессора</td>

        <td>{{ cpu.basefreq }}</td>

      </tr>

      <tr>

        <td scope="row">Максимальная частота в турбо режиме</td>

        <td>{{ cpu.maxfreq }}</td>

      </tr>

    </tbody>

  </table>

  <h4>Подходящие материнские платы:</h4>

  {% for copy in cpu.motherboard\_set.all %}

  <p></p>

  <div class="col">

    <div class="card">

      <div class="row g-0">

        <div class="col-md-4">

          <svg class="bd-placeholder-img" width="25%" height="150" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

            <img src="{{ MEDIA\_URL }}{{ copy.motherboard\_img\_url }}" width="200" height="250"class="rounded">

          </svg>

        </div>

        <div class="col-md-8">

          <div class="card-body">

            <h5 class="card-title"><a href="{% url 'motherboard-detail' copy.pk %}">{{ copy.title }}</a></h5>

            <h5 class="card-text"><class="text-muted">{{copy.price}}</class></h5>

            <p class="card-text"><small class="text-muted">{{copy.socket}}</small></p>

          </div>

        </div>

      </div>

    </div>

  </div>

  <p></p>

  {% endfor %}

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

    <script src='https://cdn.plot.ly/plotly-latest.min.js'></script>

    <script src="https://unpkg.com/gauge-chart@latest/dist/bundle.js"></script>

    <script>

      let element = document.querySelector('#gaugeArea')

      let chartWidth = 420

      let needleValue = 1

      let str1 = '{{cpu.avrrate}}'

      var k = 2

  var d = 100

  if (str1.length==3)

      { k = 1; d=10}

  let a = str1.substr(2, k)

      let a1 = str1.substr(0, 1)

      var b = Number(a);

      var b1 = Number(a1);

      let gaugeOptions = {

        hasNeedle: true,

        outerNeedle: true,

        needleColor: "black",

        needleStartValue: 0,

        arcColors: ["rgb(224,28,6)","rgb(224,144,6)","rgb(213,224,6)","rgb(112,224,6)","rgb(10,169,17)"],

        arcDelimiters: [20,40,60,80],

        rangeLabel: ["0","5"],

        rangeLabelFontSize: 0,

        arcLabels: ['1', '2', '3', '4'],

        centralLabel: str1,

        arcPadding: 3,

        arcPaddingColor: 'white',

        }

      GaugeChart.gaugeChart(element, chartWidth, gaugeOptions).updateNeedle((b1+b/d)\*20)

  </script>

{% endblock %}

templates/catalog/manufacturer\_list.html

{% extends "base\_generic.html" %}

{% block content %}

    <h1>Список производителей</h1>

    {% if manufacturer\_list %}

      {% for copy in manufacturer\_list.all %}

      <p></p>

      <div class="col">

        <div class="card">

          <div class="row g-0">

            <div class="col-md-4">

              <svg class="bd-placeholder-img" width="1%" height="100" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

                <img src="{{ MEDIA\_URL }}{{ copy.manufacturer\_img\_url }}" width="400" height="100"class="rounded">

              </svg>

            </div>

            <div class="col-md-8">

              <div class="card-body">

                <h5 class="card-title"><a href="{{ copy.get\_absolute\_url }}">{{ copy.namecompany }}</a></h5>

              </div>

            </div>

          </div>

        </div>

      </div>

      {% endfor %}

    {% else %}

      <p>There are no manufacturers</p>

    {% endif %}

{% endblock %}

templates/catalog/manufacturer\_detail.html

{% extends "base\_generic.html" %}

{% block content %}

<p></p>

          <div class="col">

            <div class="card">

              <svg class="bd-placeholder-img card-img-top" width="10%" height="1" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image cap" preserveAspectRatio="xMidYMid slice" focusable="false">

                <img src="{{ MEDIA\_URL }}{{ manufacturer.manufacturer\_img\_url }}">

            </svg>

              <div class="card-body">

                <h5 class="card-title">{{ manufacturer.namecompany }}</h5>

                <p class="card-text">{{ manufacturer.description }}</p>

                <a href="{{ manufacturer.urlman }}" class="btn btn-primary">Перейти на сайт компании</a>

              </div>

            </div>

          </div><p></p>

    <h4>Товары производителя:</h4>

    {% for copy in manufacturer.motherboard\_set.all %}

    <p></p>

    <div class="col">

      <div class="card">

        <div class="row g-0">

          <div class="col-md-4">

            <svg class="bd-placeholder-img" width="25%" height="150" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

              <img src="{{ MEDIA\_URL }}{{ copy.motherboard\_img\_url }}" width="200" height="250"class="rounded">

            </svg>

          </div>

          <div class="col-md-8">

            <div class="card-body">

              <h5 class="card-title"><a href="{% url 'motherboard-detail' copy.pk %}">{{ copy.title }}</a></h5>

              <h5 class="card-text"><class="text-muted">{{copy.price}}</class></h5>

              <p class="card-text"><small class="text-muted">{{copy.socket}}</small></p>

            </div>

          </div>

        </div>

      </div>

    </div>

    {% endfor %}

{% endblock %}

templates/catalog/motherboard\_list.html

{% extends "base\_generic.html" %}

{% block content %}

    <h1>Список производителей</h1>

    {% if manufacturer\_list %}

      {% for copy in manufacturer\_list.all %}

      <p></p>

      <div class="col">

        <div class="card">

          <div class="row g-0">

            <div class="col-md-4">

              <svg class="bd-placeholder-img" width="1%" height="100" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

                <img src="{{ MEDIA\_URL }}{{ copy.manufacturer\_img\_url }}" width="400" height="100"class="rounded">

              </svg>

            </div>

            <div class="col-md-8">

              <div class="card-body">

                <h5 class="card-title"><a href="{{ copy.get\_absolute\_url }}">{{ copy.namecompany }}</a></h5>

              </div>

            </div>

          </div>

        </div>

      </div>

      {% endfor %}

    {% else %}

      <p>There are no manufacturers</p>

    {% endif %}

{% endblock %}

templates/catalog/motherboard\_detail.html

{% extends "base\_generic.html" %}

{% load static %}

{% block content %}

<h1>Материнская плата {{ motherboard.title }}</h1>

<div class="container">

    <div class="row">

      <div class="col">

            <img src="{{ MEDIA\_URL }}{{ motherboard.motherboard\_img\_url }}" width="300" height="375"class="rounded">

      </div>

      <div class="col">

        <div class="col-md-8">

            <div class="card-body">

                  <h5 class="card-title">Всего оценок: {{ motherboard.countrate}}</h5>

                  <div id='gaugeArea'><!-- Plotly chart will be drawn inside this DIV --></div>

            </div>

          </div>

      </div>

      <div class="col">

        <p></p>

        <h1 class="card-title">{{ motherboard.price }}</h1>

        <h3 class="card-title">Оценка товара: {{ motherboard.avrrate}} из 5</h3>

        <button type="button" class="btn btn-warning" onclick="window.location.href = 'editmotherboard/{{motherboard.id}}';">Оценить</button>

  </div>

</div>

<table class="table">

    <thead>

      <tr>

        <th scope="col"><h2>Характеристики</h2></th>

        <th scope="col"></th>

      </tr>

    </thead>

    <tbody>

     <tr class="table-dark">

        <th colspan="2">Общие параметры</th>

      </tr>

      <tr>

        <td scope="row">Производитель</td>

        <td><a href="{% url 'manufacturer-detail' motherboard.manufacturer.pk %}">{{ motherboard.manufacturer }}</a></td>

      </tr>

      <tr>

        <td scope="row">Модель</td>

        <td>{{ motherboard.title }}</td>

      </tr>

      <tr>

        <td scope="row">Год релиза</td>

        <td>{{ motherboard.yearrelease }}</td>

      </tr>

      <tr class="table-dark">

        <th colspan="2">Форм-фактор и сокет</th>

      </tr>

      <tr>

        <td scope="row">Форм-фактор</td>

        <td>{{ motherboard.formfactor }}</td>

      </tr>

      <tr>

        <td scope="row">Сокет</td>

        <td>{{ motherboard.socket }}</td>

      </tr>

      </tr>

      <tr class="table-dark">

        <th colspan="2">Память</th>

      </tr>

      <tr>

        <td scope="row">Тип поддерживаемой памяти</td>

        <td>{{ motherboard.typememory }}</td>

      </tr>

    </tbody>

  </table>

  <h4>Подходящие процессоры:</h4>

  {% for copy in motherboard.cpu.all %}

  <p></p>

  <div class="col">

    <div class="card">

      <div class="row g-0">

        <div class="col-md-4">

          <svg class="bd-placeholder-img" width="25%" height="150" xmlns="http://www.w3.org/2000/svg" role="img" aria-label="Placeholder: Image" preserveAspectRatio="xMidYMid slice" focusable="false">

            <img src="{{ MEDIA\_URL }}{{ copy.cpu\_img\_url }}" width="200" height="200"class="rounded">

          </svg>

        </div>

        <div class="col-md-8">

          <div class="card-body">

            <h5 class="card-title"><a href="{% url 'cpu-detail' copy.pk %}">{{ copy.name }}</a></h5>

            <h5 class="card-text"><class="text-muted">{{copy.price}}</class></h5>

            <p class="card-text"><small class="text-muted">{{copy.socket}}</small></p>

          </div>

        </div>

      </div>

    </div>

  </div>

  <p></p>

  {% endfor %}

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/js/bootstrap.bundle.min.js" integrity="sha384-ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW" crossorigin="anonymous"></script>

    <script src='https://cdn.plot.ly/plotly-latest.min.js'></script>

    <script src="https://unpkg.com/gauge-chart@latest/dist/bundle.js"></script>

    <script>

      let element = document.querySelector('#gaugeArea')

      let chartWidth = 420

      let needleValue = 1

      let str1 = '{{motherboard.avrrate}}'

      var k = 2

  var d = 100

  if (str1.length==3)

      { k = 1; d=10}

  let a = str1.substr(2, k)

      let a1 = str1.substr(0, 1)

      var b = Number(a);

      var b1 = Number(a1);

      let gaugeOptions = {

        hasNeedle: true,

        outerNeedle: true,

        needleColor: "black",

        needleStartValue: 0,

        arcColors: ["rgb(224,28,6)","rgb(224,144,6)","rgb(213,224,6)","rgb(112,224,6)","rgb(10,169,17)"],

        arcDelimiters: [20,40,60,80],

        rangeLabel: ["0","5"],

        rangeLabelFontSize: 0,

        arcLabels: ['1', '2', '3', '4'],

        centralLabel: str1,

        arcPadding: 3,

        arcPaddingColor: 'white',

        }

      GaugeChart.gaugeChart(element, chartWidth, gaugeOptions).updateNeedle((b1+b/d)\*20)

  </script>

{% endblock %}

Procfile

web: PYTHONPATH=$(pwd)'/src' waitress-serve --port=$PORT mysitedz.wsgi:application

requirements.txt

asgiref==3.3.1

Django==3.1.4

gunicorn==20.0.4

Pillow==8.1.0

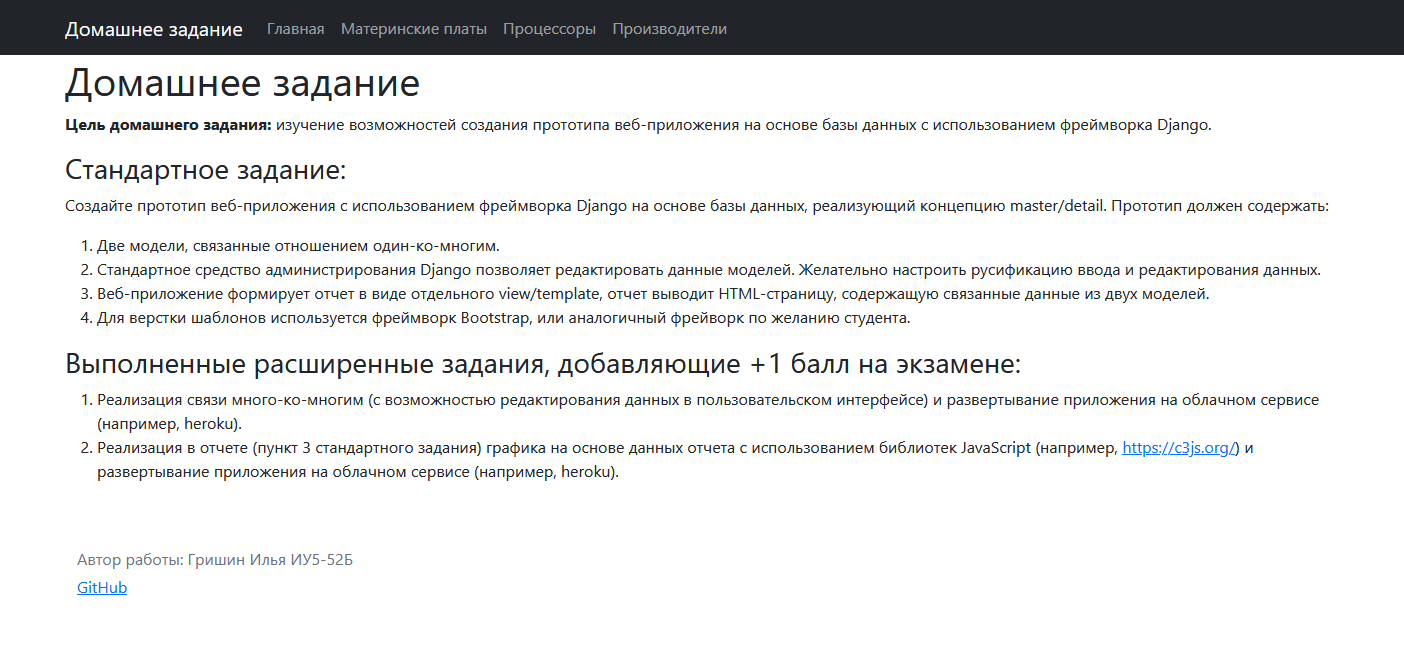
pytz==2020.5

sqlparse==0.4.1

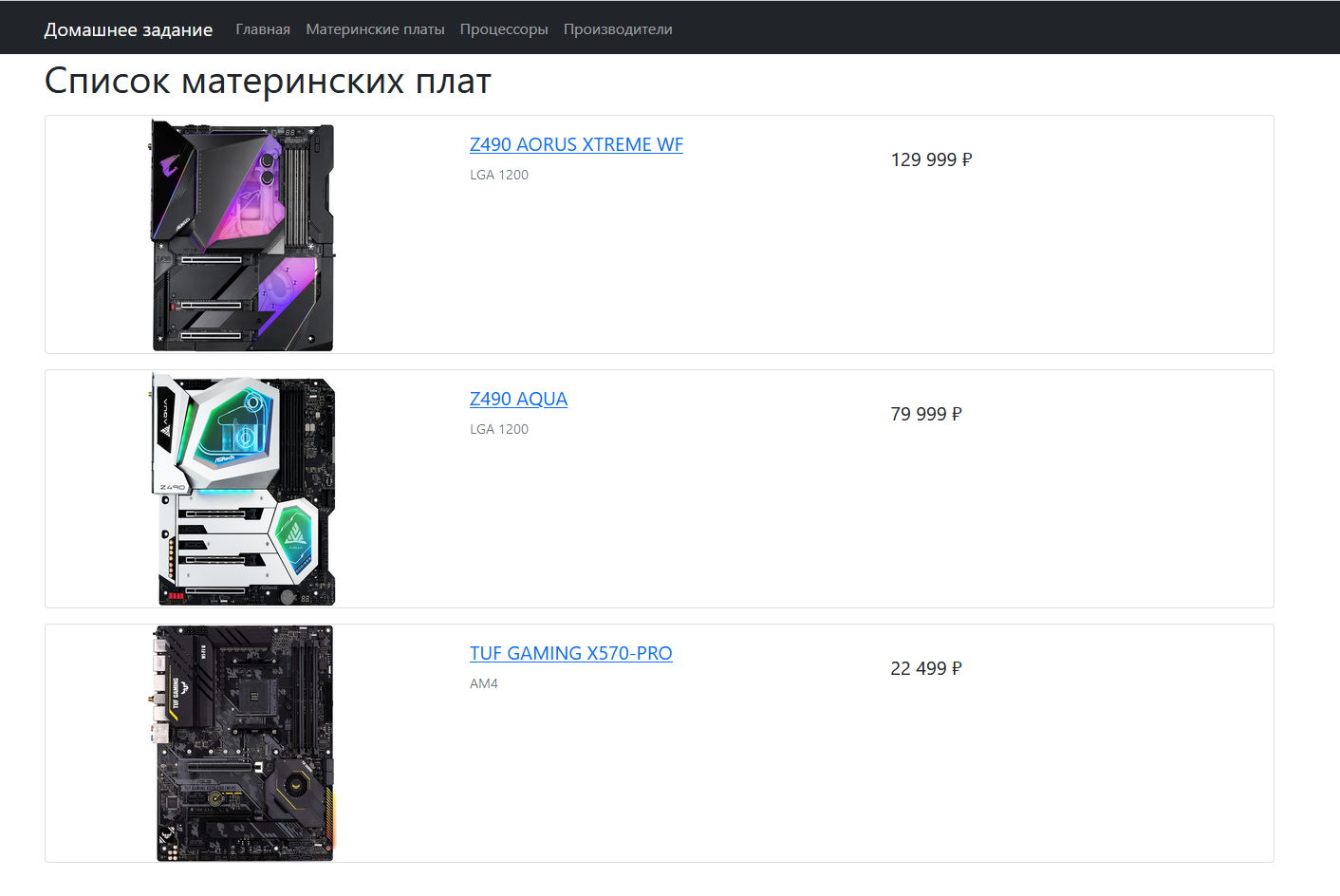
waitress==1.4.4

**4. Экранные формы с примерами выполнения программы**

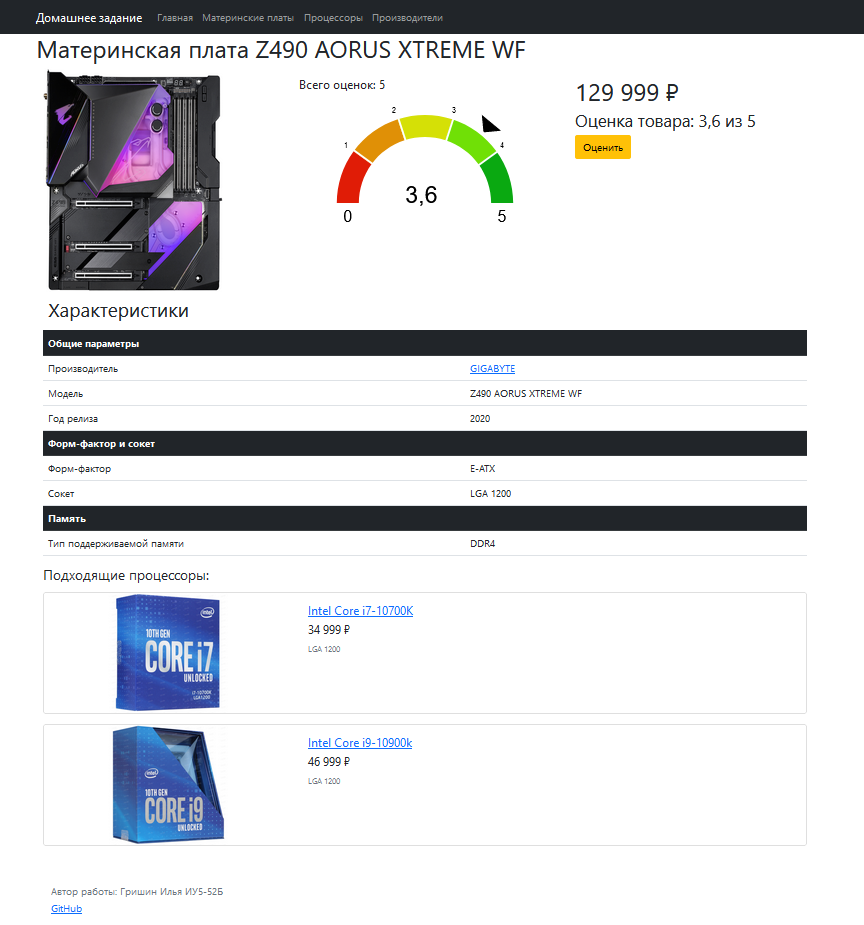
catalog/



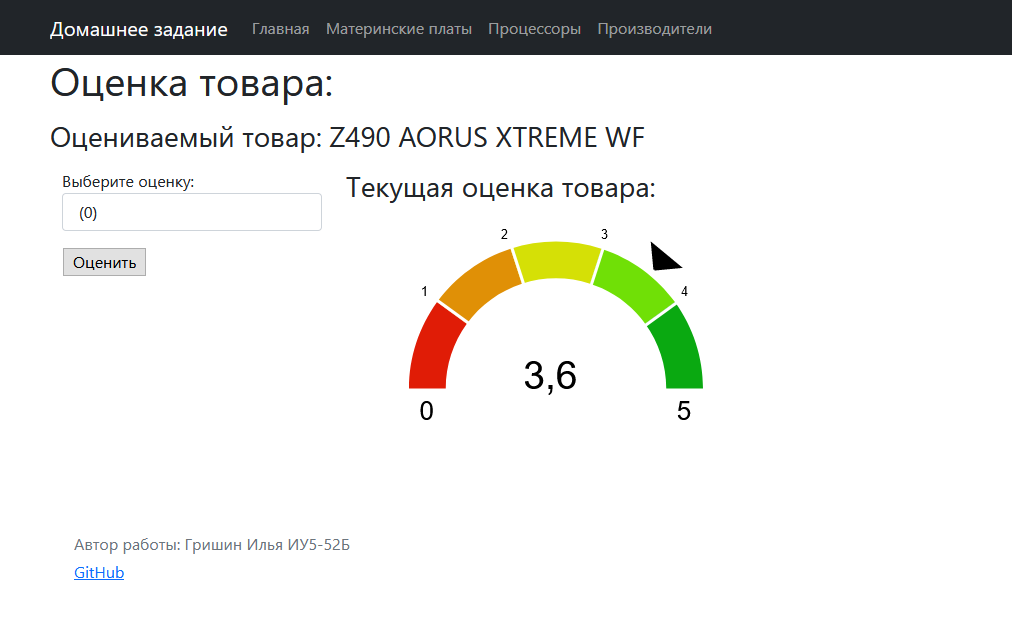
catalog/motherboards/



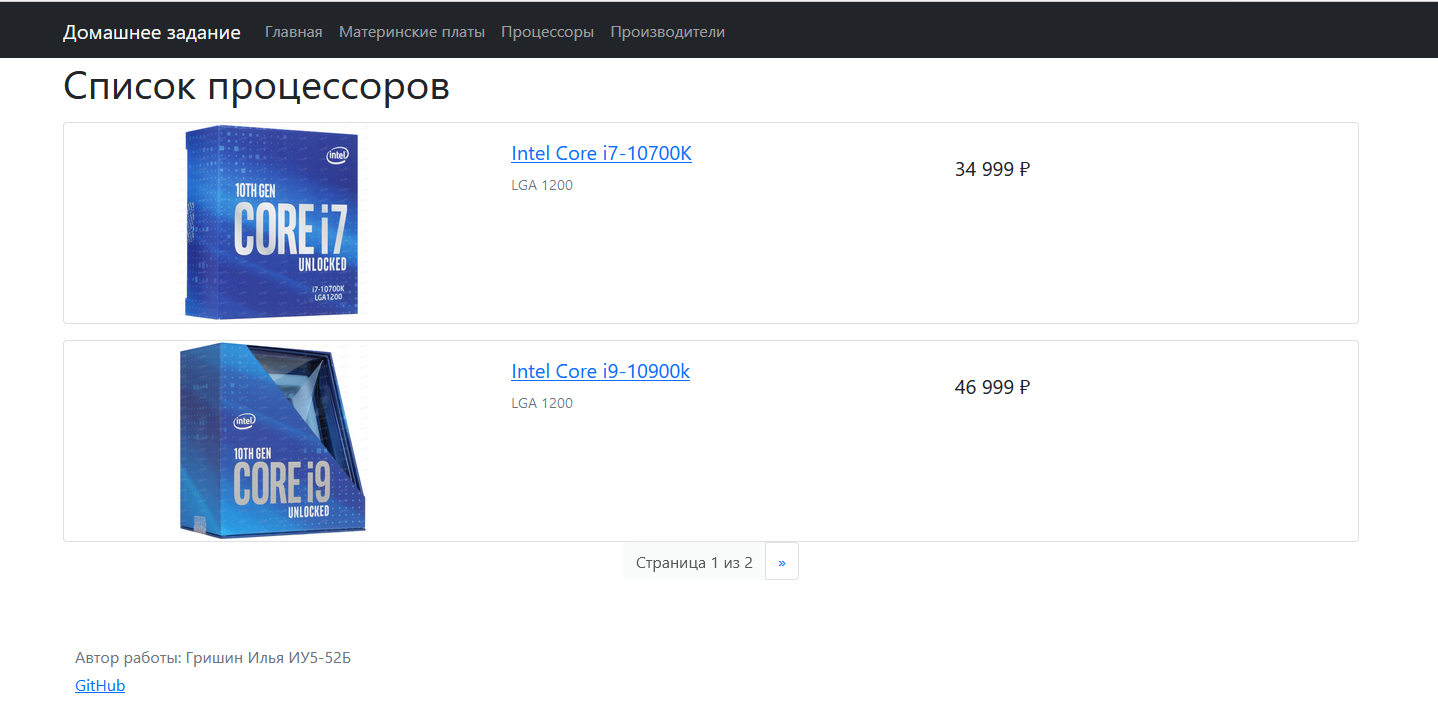
catalog/motherboard/1/



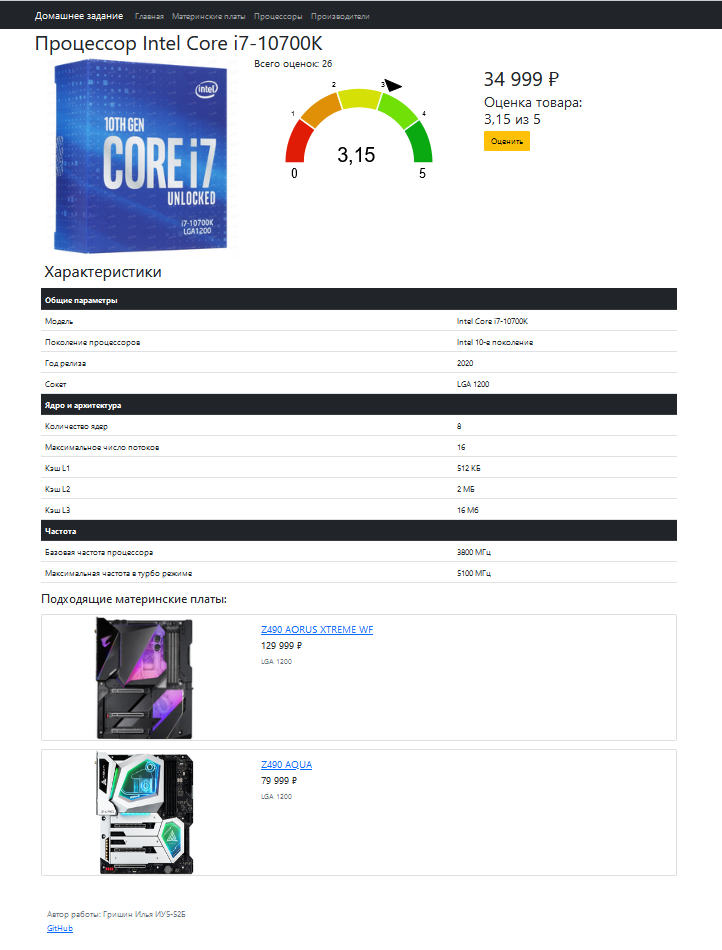
catalog/motherboard /editmotherboard/1/



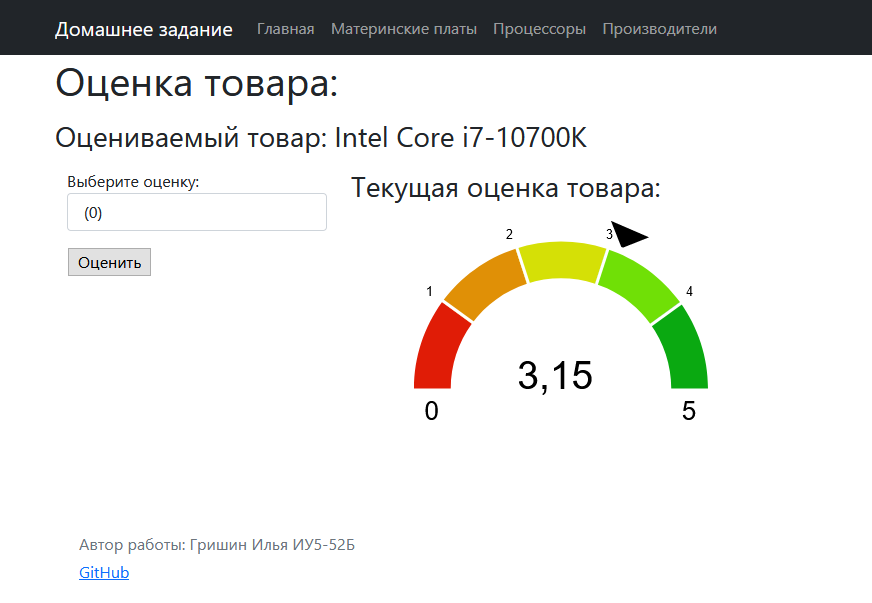
catalog/cpus/



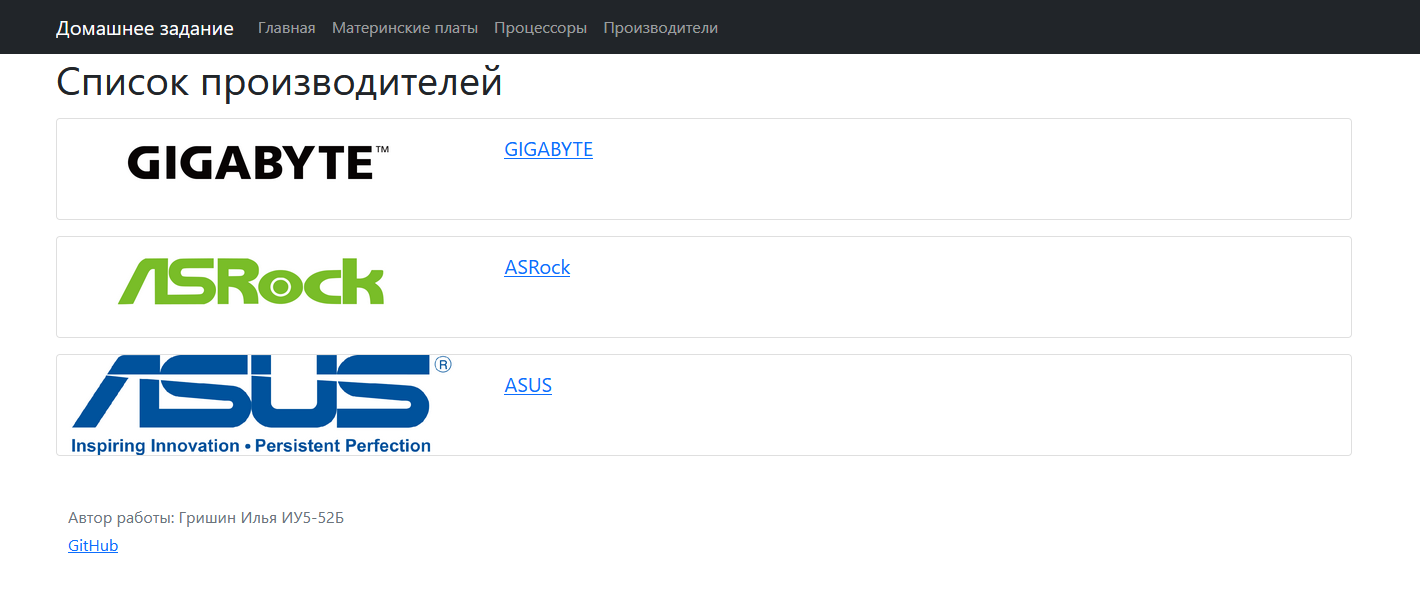
catalog/cpu/1



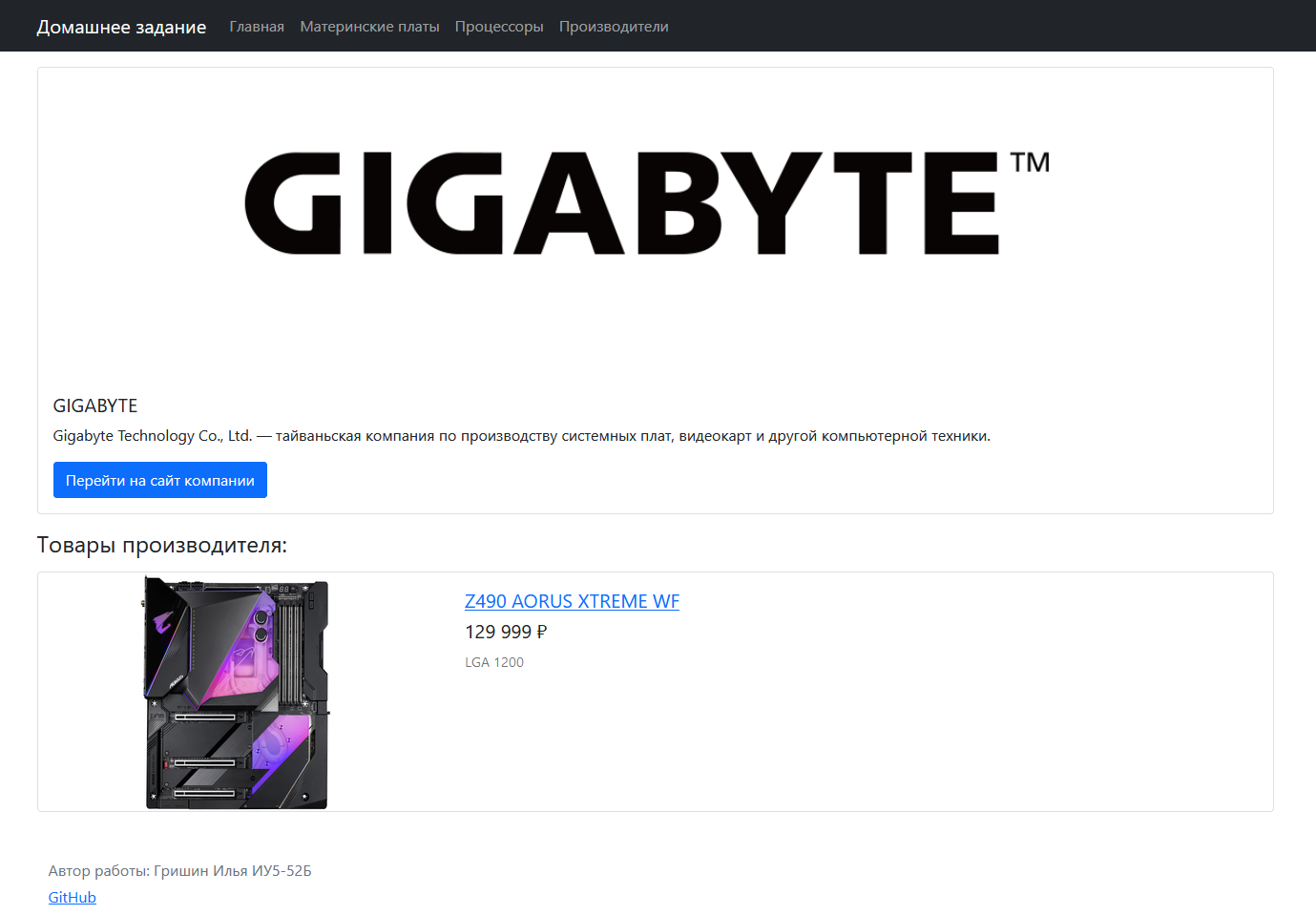
catalog/cpu/editcpu/1/



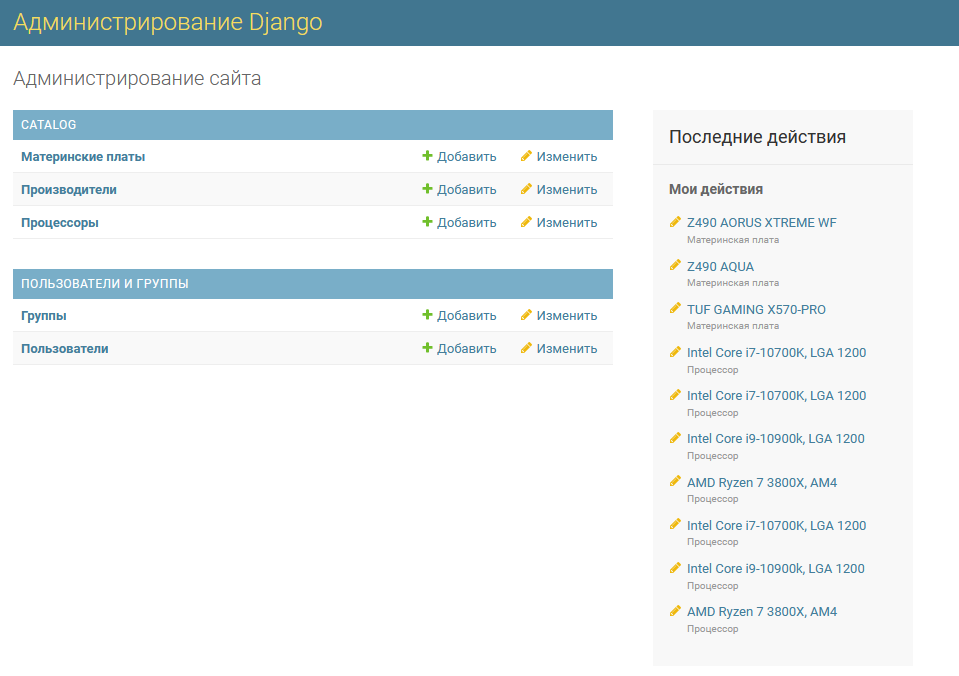
catalog/manufacturers/



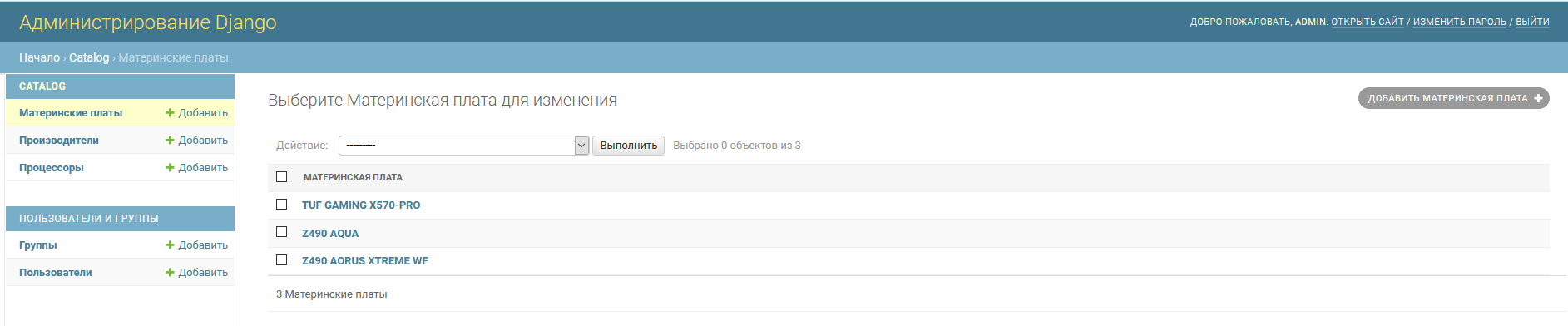
catalog/manufacturer/1



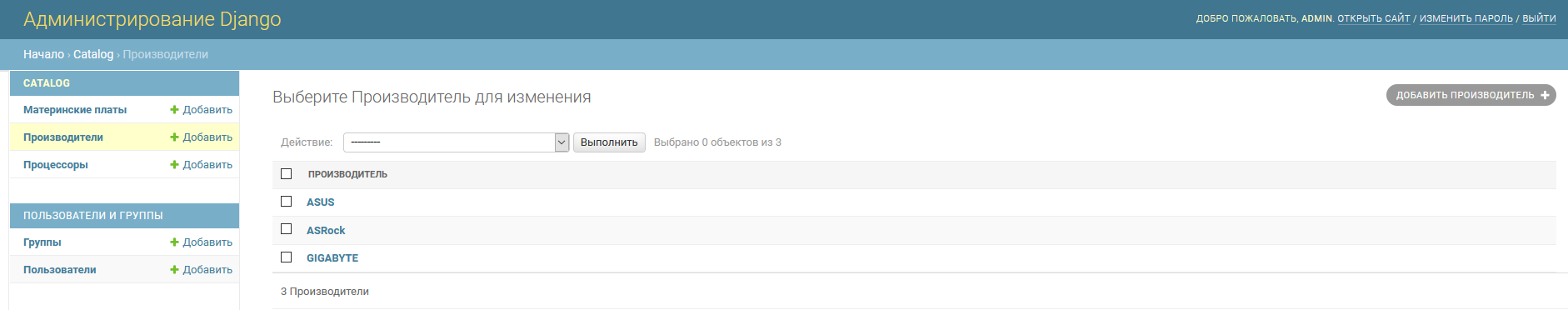
admin/



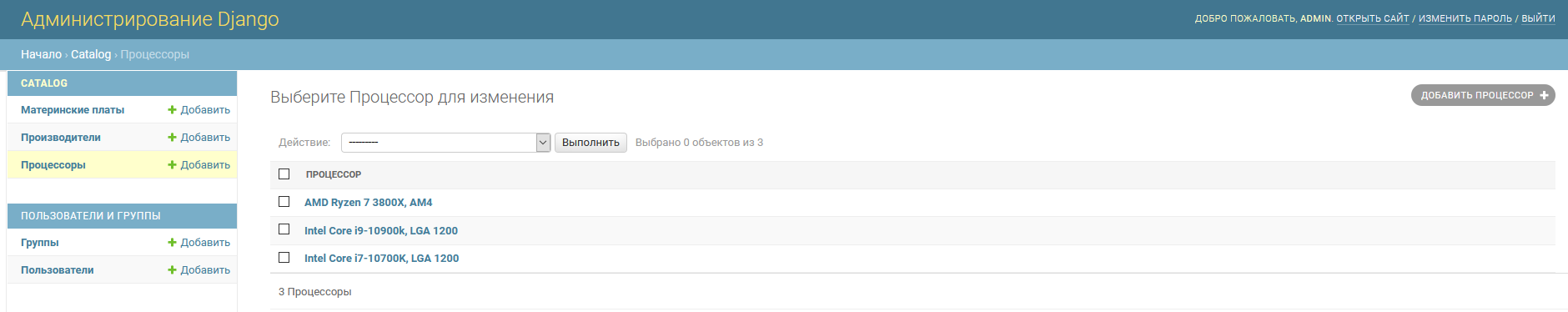
admin/catalog/motherboard/



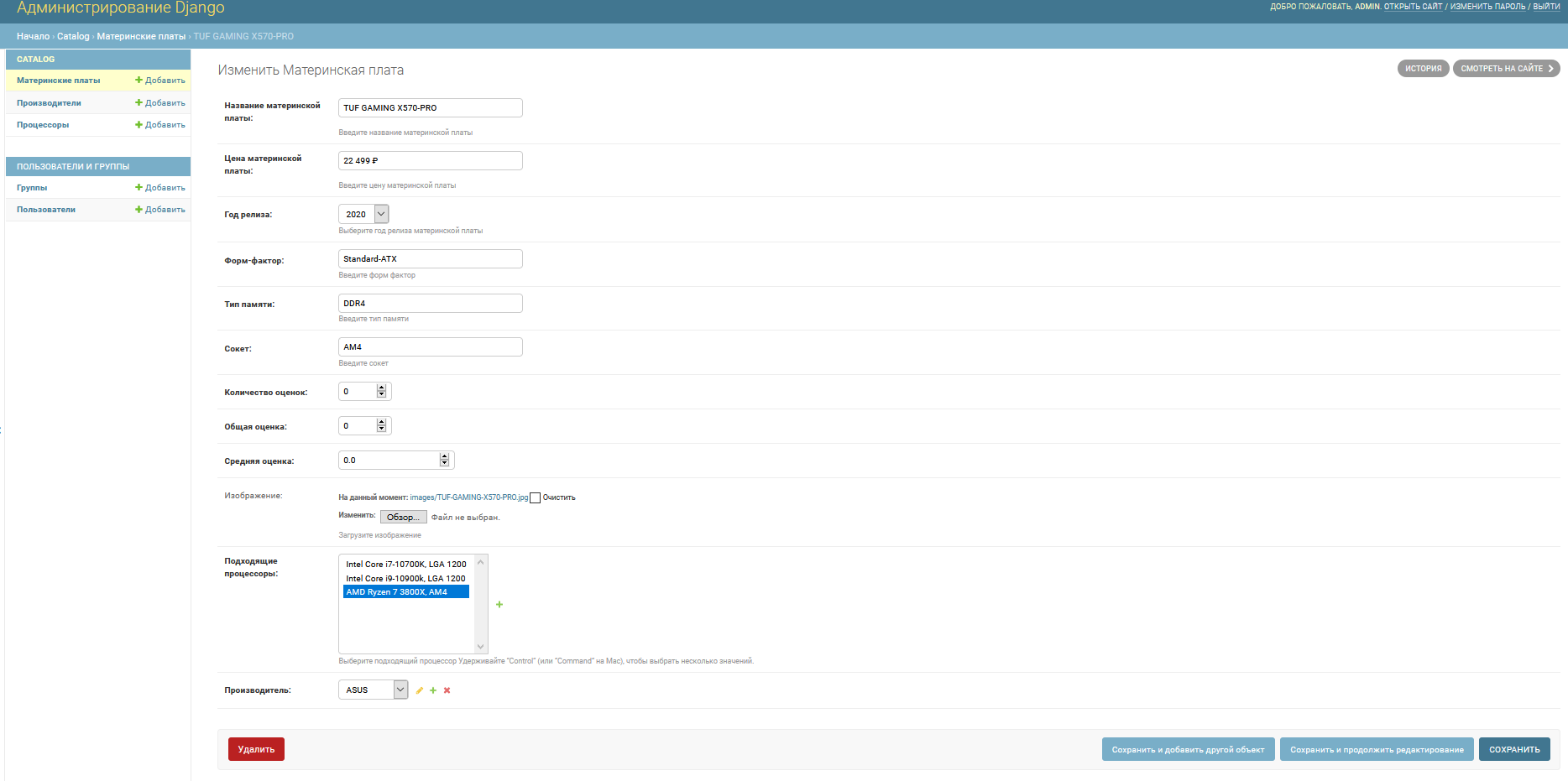
admin/catalog/manufacturer/



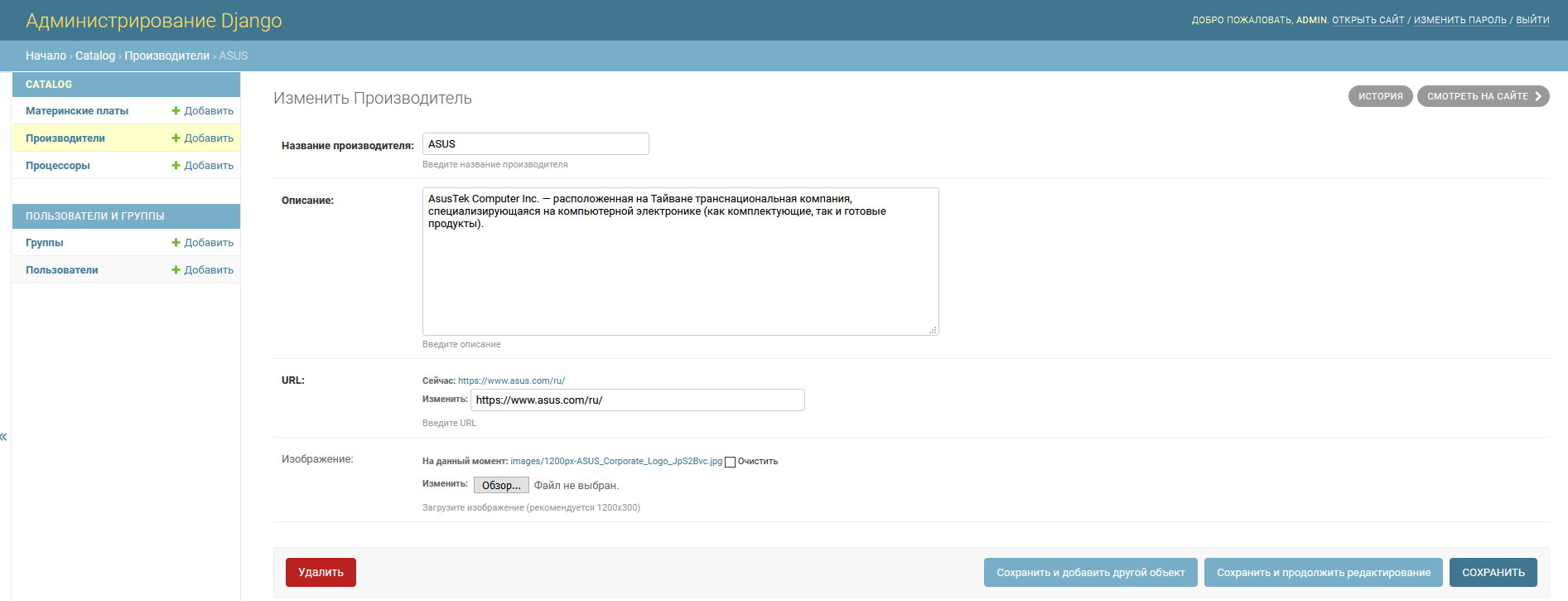
admin/catalog/cpu/



admin/catalog/motherboard/3/change/



admin/catalog/manufacturer/3/change/



admin/catalog/cpu/3/change/

