



**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 КАФЕДРА	«Информатики и систем управления»
	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 фреймворк 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:

▼ MYSITEDZ	
▼ catalog	●
> __pycache__	●
> migrations	●
✚ __init__.py	U
✚ admin.py	U
✚ apps.py	U
✚ models.py	U
✚ tests.py	U
✚ urls.py	U
✚ views.py	U
> images	●
> media\images	●
▼ mysitedz	●
> __pycache__	●
✚ __init__.py	U
✚ asgi.py	U
✚ settings.py	U
✚ urls.py	U
✚ wsgi.py	U
> static	●
▼ templates	●
▼ catalog	●
> cpu	
✧ cpu_detail.html	U
✧ cpu_list.html	U
✧ manufacturer_detail.html	U
✧ manufacturer_list.html	U
✧ motherboard_detail.html	U
✧ motherboard_list.html	U
✧ base_generic.html	U
✧ editcpu.html	U
✧ editmotherboard.html	U
✧ index.html	U
≡ db.sqlite3	U
✚ manage.py	U
🔗 Procfile	U
≡ requirements.txt	U

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 HttpResponseRedirect("<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": mother
board})
    except Motherboard.DoesNotExist:
        return HttpResponseRedirect("<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="an
onymous">
    <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="an
onymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper
.min.js" integrity="sha384-
U02eT0CpHqdsJQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="an
onymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.mi
n.js" integrity="sha384-
```

```
JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/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 фреймворк 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'</di
v></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"
>

```

```

</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"><small class="text-muted">{{copy.price}}</small></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">
      
    </div>
  </div>
</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"
>
          
        </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="an
onymous"></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(112,22
4,6)","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"
>
                                
                            </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>
            </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">
                
            </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">
                    
                </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>
```

```

        </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"
>
                                
                            </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>
            </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">
            
        </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}}';">0
            ценить</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></td>
        </tr>
    </tbody>
</table>

</div>
</div>
</div>
```

```

        <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"
>
                    
                </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/t7LECLbyPA2x65KgF800JFdroafW" crossorigin="an
onymous"></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(112,22
4,6)","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/

Домашнее задание Главная Материнские платы Процессоры Производители

Домашнее задание

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

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 фреймворк 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).

Автор работы: Гришин Илья ИУ5-52Б

[GitHub](#)

Список материнских плат



[Z490 AORUS XTREME WF](#)

LGA 1200

129 999 ₽



[Z490 AQUA](#)

LGA 1200

79 999 ₽



[TUF GAMING X570-PRO](#)

AM4

22 499 ₽

Материнская плата Z490 AORUS XTREME WF



Всего оценок: 5



129 999 ₽

Оценка товара: 3,6 из 5

Оценить

Характеристики

Общие параметры

Производитель	GIGABYTE
Модель	Z490 AORUS XTREME WF
Год релиза	2020

Форм-фактор и сокет

Форм-фактор	E-ATX
Сокет	LGA 1200

Память

Тип поддерживаемой памяти	DDR4
---------------------------	------

Подходящие процессоры:



[Intel Core i7-10700K](#)

34 999 ₽

LGA 1200



[Intel Core i9-10900K](#)

46 999 ₽

LGA 1200

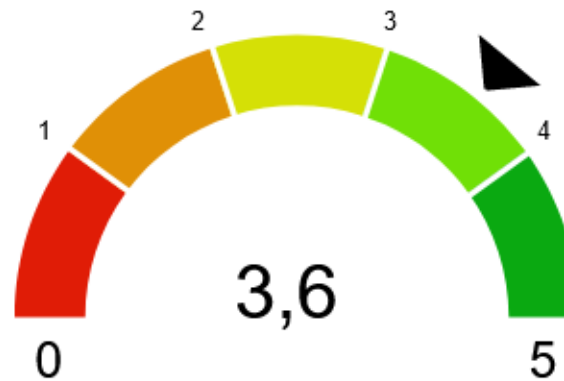
Оценка товара:

Оцениваемый товар: Z490 AORUS XTREME WF

Выберите оценку:

Оценить

Текущая оценка товара:



Автор работы: Гришин Илья ИУ5-52Б

[GitHub](#)

Список процессоров



[Intel Core i7-10700K](#)

LGA 1200

34 999 ₽



[Intel Core i9-10900k](#)

LGA 1200

46 999 ₽

Процессор Intel Core i7-10700K



Всего оценок: 26



34 999 Р

Оценка товара:
3,15 из 5

Оценить

Характеристики

Общие параметры	
Модель	Intel Core i7-10700K
Поколение процессоров	Intel 10-е поколение
Год релиза	2020
Сокет	LGA 1200
Ядро и архитектура	
Количество ядер	8
Максимальное число потоков	16
Кэш L1	512 КБ
Кэш L2	2 МБ
Кэш L3	16 МБ
Частота	
Базовая частота процессора	3800 МГц
Максимальная частота в турбо режиме	5100 МГц

Подходящие материнские платы:

	Z490 AORUS XTREME WF 129 999 Р LGA 1200
	Z490 AQUA 79 999 Р LGA 1200

Оценка товара:

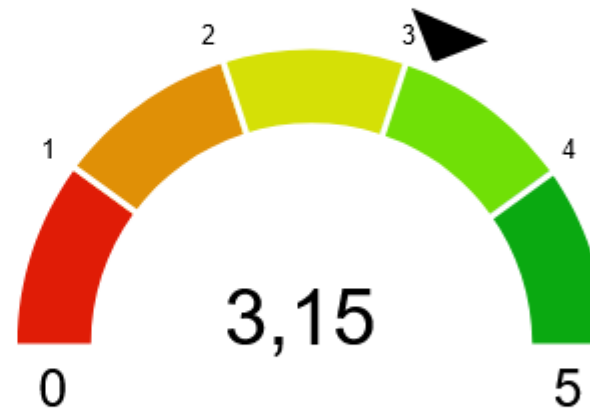
Оцениваемый товар: Intel Core i7-10700K

Выберите оценку:

(0)

Оценить

Текущая оценка товара:



Автор работы: Гришин Илья ИУ5-52Б

[GitHub](#)

Список производителей

GIGABYTE™

[GIGABYTE](#)

ASRock

[ASRock](#)

ASUS®

Inspiring Innovation • Persistent Perfection

[ASUS](#)

Автор работы: Гришин Илья ИУ5-52Б

[GitHub](#)

GIGABYTE™

GIGABYTE

Gigabyte Technology Co., Ltd. — тайваньская компания по производству системных плат, видеокарт и другой компьютерной техники.

[Перейти на сайт компании](#)

Товары производителя:



[Z490 AORUS XTREME WF](#)

129 999 ₽

LGA 1200

Автор работы: Гришин Илья ИУ5-52Б

[GitHub](#)

Администрирование Django

Администрирование сайта

CATALOG		
Материнские платы	+ Добавить	✎ Изменить
Производители	+ Добавить	✎ Изменить
Процессоры	+ Добавить	✎ Изменить
ПОЛЬЗОВАТЕЛИ И ГРУППЫ		
Группы	+ Добавить	✎ Изменить
Пользователи	+ Добавить	✎ Изменить

Последние действия

Мои действия

- [✎ Z490 AORUS XTREME WF](#)
Материнская плата
- [✎ Z490 AQUA](#)
Материнская плата
- [✎ TUF GAMING X570-PRO](#)
Материнская плата
- [✎ Intel Core i7-10700K, LGA 1200](#)
Процессор
- [✎ Intel Core i7-10700K, LGA 1200](#)
Процессор
- [✎ Intel Core i9-10900k, LGA 1200](#)
Процессор
- [✎ AMD Ryzen 7 3800X, AM4](#)
Процессор
- [✎ Intel Core i7-10700K, LGA 1200](#)
Процессор
- [✎ Intel Core i9-10900k, LGA 1200](#)
Процессор
- [✎ AMD Ryzen 7 3800X, AM4](#)
Процессор

admin/catalog/motherboard/

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало · Catalog · Материнские платы

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Выберите Материнская плата для изменения

ДОБАВИТЬ МАТЕРИНСКАЯ ПЛАТА +

Действие: Выполнить Выбрано 0 объектов из 3

☐ МАТЕРИНСКАЯ ПЛАТА

☐ TUF GAMING X570-PRO

☐ Z490 AQUA

☐ Z490 AORUS XTREME WF

3 Материнские платы

admin/catalog/manufacture/

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало · Catalog · Производители

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Выберите Производитель для изменения

ДОБАВИТЬ ПРОИЗВОДИТЕЛЬ +

Действие: Выполнить Выбрано 0 объектов из 3

☐ ПРОИЗВОДИТЕЛЬ

☐ ASUS

☐ ASRock

☐ GIGABYTE

3 Производители

admin/catalog/cpu/

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. [ОТКРЫТЬ САЙТ](#) / [ИЗМЕНИТЬ ПАРОЛЬ](#) / [ВЫЙТИ](#)

Начало · Catalog · Процессоры

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Выберите Процессор для изменения

ДОБАВИТЬ ПРОЦЕССОР +

Действие: Выполнить Выбрано 0 объектов из 3

☐ ПРОЦЕССОР

☐ AMD Ryzen 7 3800X, AM4

☐ Intel Core i9-10900K, LGA 1200

☐ Intel Core i7-10700K, LGA 1200

3 Процессоры

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало · Catalog · Материнские платы · TUF GAMING X570-PRO

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

ИСТОРИЯ

СМОТРЕТЬ НА САЙТЕ >

Изменить Материнская плата

Название материнской платы:

TUF GAMING X570-PRO

Введите название материнской платы

Цена материнской платы:

22 499 P

Введите цену материнской платы

Год релиза:

2020

Выберите год релиза материнской платы

Форм-фактор:

Standard-ATX

Введите форм фактор

Тип памяти:

DDR4

Введите тип памяти

Сокет:

AM4

Введите сокет

Количество оценок:

0

Общая оценка:

0

Средняя оценка:

0.0

Изображение:

На данный момент: images/TUF-GAMING-X570-PRO.jpg

Очистить

Изменить: Обзор...

Файл не выбран.

Загрузите изображение

Подходящие процессоры:

Intel Core i7-10700K, LGA 1200

Intel Core i9-10900K, LGA 1200

AMD Ryzen 7 3800X, AM4

Выберите подходящий процессор Удерживайте "Control" (или "Command" на Mac), чтобы выбрать несколько значений.

Производитель:

ASUS

Удалить

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, ADMIN. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало · Catalog · Производители · ASUS

ИСТОРИЯ СМОТРЕТЬ НА САЙТЕ

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Изменить Производитель

Название производителя: ASUS

Введите название производителя

Описание:

AsusTek Computer Inc. — расположенная на Тайване транснациональная компания, специализирующаяся на компьютерной электронике (как комплектующие, так и готовые продукты).

Введите описание

URL:

Сейчас: <https://www.asus.com/ru/>

Изменить:

Введите URL

Изображение:

На данный момент: [images/1200px-ASUS_Corporate_Logo_JpS2Bvc.jpg](#) ☐ Очистить

Изменить: Файл не выбран.

Загрузите изображение (рекомендуется 1200x300)

Удалить

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ

Администрирование Django

Начало / Catalog / Процессоры / AMD Ryzen 7 3800X, AM4

CATALOG

Материнские платы + Добавить

Производители + Добавить

Процессоры + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

ИСТОРИЯ

СМОТРЕТЬ НА CARTE >

Изменить Процессор

Название процессора: AMD Ryzen 7 3800X

Введите название процессора

Цена процессора: 32 399 Р

Введите цену процессора

Поколение процессора: AMD RYZEN 3000

Введите поколение процессора

Год релиза: 2019

Выберите год релиза процессора

Сокет: AM4

Введите сокет

Количество ядер: 8

Введите количество ядер

Количество потоков: 16

Введите количество потоков

Кэш L1: 512 КБ

Введите объем кэша L1

Кэш L2: 4 МБ

Введите объем кэша L2

Кэш L3: 32 МБ

Введите объем кэша L3

Базовая частота процессора: 3900 МГц

Введите базовую частоту процессора

Максимальная частота процессора: 4500 МГц

Введите максимальную частоту процессора

Количество оценок: 2

Общая оценка: 7

Средняя оценка: 3.5

Изображение: На данный момент: images/ryzen-7-3800x.jpg Очистить

Изменить: Обзор... Файл не выбран.

Загрузите изображение

Удалить

Сохранить и добавить другой объект

Сохранить и продолжить редактирование

СОХРАНИТЬ