**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *Подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *Дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Разработка web-сервиса конфигурации ПК | | | | | |
|  |  | | | | | |
|  | |  | БР.БИСТ.20.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | М.А. Ревякин |  |  |  | канд. техн. наук, доц.  Е.Б.Тарасов |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Ревякину Михаилу Александровичу | | |
|  |  | | |
| 1. Тема «Разработка web-сервиса конфигурации ПК» утверждена приказом № 203/c от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 1 | 20.05.2022 |
| Описание предметной области | | 10 | 20.05.2022 |
| Проектирование информационной системы | | 20 | 25.05.2022 |
| Руководство пользователя | | 6 | 07.06.2022 |
| Заключение | | 1 | 09.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  Выполнения |
| Презентация PowerPoint | 17 | 07.06.2022 |

7. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование  Раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | Е.Б. Тарасов |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | М.А. Ревякин |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 45 страниц, 27 рисунков, 7 таблиц, 15 источников.

Ключевые слова: *конфигурация ПК, web-сервис, базы данных.*

Выпускная квалификационная работа посвящена разработке web-сервиса конфигурации ПК.

В работе представлена разработка концепции и дизайна для сайта сервиса, а также разметка данного сайта, проработка программной части и баз данных.

Для достижения цели были изучены и смоделированы процессы, выполняемые в ходе конфигурации, рассмотрены аналогичные сайты-конфигураторы и спроектирован и разработан сайт для конфигурации.

**ABSTRACT**

This work contains 45 pages, 27 figures, 7 tables, 15 sources.

Keywords: *PC configuration, web-service, databases.*

The final qualifying work is devoted to the development of a PC configuration web service.

The paper presents the development of the concept and design for the service site, as well as the markup of this site, the development of the software and databases.

To achieve this goal, the processes performed during configuration were studied and modeled, similar configurator sites were considered, and a configuration site was designed and developed.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ПО – программное обеспечение.

ПК – персональный компьютер.

БД – база данных.

СУБД – система управления базами данных.

Hard-lock – жёсткий запрет. В программе: невозможность выполнения какого-либо действия.

Soft-lock – мягкий запрет. В программе: предупреждение о невозможности реализации действия.

HEDT-процессор – High-End Desktop Processor. Процессор наивысшего потребительского уровня. Используется, чаще всего, в серверах или на самых лучших рабочих станциях. В обычных компьютерах не применяется из-за высокой цены и нулевых преимуществ в обычных задачах.

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc42884660)

1 [Описание предметной области](#_Toc42884661) 9

[1.1 Основные сведения о конфигураторах ПК.](#_Toc42884662) 10

[1.2 Существующие конфигураторы ПК](#_Toc42884663) 12

[1.3 Итоги анализа конфигураторов](#_Toc42884663) 17

[1.4 Выводы по разделу](#_Toc42884663) 18

[2 Проектирование информационной системы](#_Toc42884664) 19

[2.1 Основные требования](#_Toc42884665) 19

[2.2 Прототипирование баз данных 2](#_Toc42884665)2

[2.3 Выбор среды разработки 2](#_Toc42884665)6

[2.4 Прототипирование логики и дизайна](#_Toc42884665) 28

[2.5 Моделирование бизнес-процессов информационной системы](#_Toc42884665) 31

[2.6 Общий концепт программного обеспечения](#_Toc42884666) 34

[2.7 Выводы по разделу](#_Toc42884668) 36

[3 Руководство пользователя](#_Toc42884672) 37

[3.1 Назначение и условия применения](#_Toc42884673) 37

[3.2 Подготовка к работе](#_Toc42884674) 37

[3.3 Описание операций](#_Toc42884675) 38

[3.4 Аварийные ситуации](#_Toc42884676) 42

[3.5 Рекомендации по освоению 4](#_Toc42884677)2

[3.6 Вывод по разделу 4](#_Toc42884678)2

[Заключение](#_Toc42884683) 43

[Список использованных источников](#_Toc42884684) 44

**ВВЕДЕНИЕ**

На современном рынке электроники сложилась ситуация, в которой услуги, требующие минимальной квалификации, оцениваются очень высоко, и сборка компьютеров – пример такой услуги. Собранная машина стоит намного дороже суммы стоимостей её комплектующих, и наценка достигает от десяти процентов до нескольких стоимостей. Если для простого пользователя, который берёт себе ПК, подобное может встать в переплату нескольких десятков процентов, и не оказать существенного влияния, то в условиях компьютеризации офиса или тем более крупной компании эти же десятки процентов могут исчисляться уже в миллионах рублей, что станет причиной значительных убытков без какой-либо прибыли. В Интернете существуют приложения, позволяющие получить какие-то общие сведения, однако для человека малознакомого или вообще незнакомого с конфигурацией и сборкой они не являются действительной помощью, а могут привести лишь к ещё большим тратам.

Конфигурация комплектующих – задача многофакторная. Даже не учитывая стоимость комплектующих, как минимум необходимо учитывать цель, для которой собирается компьютер. Как минимум их можно разделить на компьютеры для офиса/мультимедиа, рабочие станции и игровые компьютеры. Существуют ещё серверные машины и особые суперкомпьютеры для научных расчётов, однако их в рамках данной работы и данного конфигуратора собирать не имеет смысла, каждая такая машина собирается индивидуально. В случае же, если учитывается и цена комплектующих, количество разных вариантов возрастает многократно, так как минимальные системные требования к программам не всегда отражают реальную картину и аналоги более старых моделей процессоров или видеокарт могут быть интуитивно непонятны пользователю.

Существует и проблема совместимости комплектующих. Самый очевидный частный случай – несовместимость материнской платы и центрального процессора. На потребительском рынке сложилась ситуация почти полного паритета у процессоров AMD и Intel, однако ни одна плата, которая подходит для какого-либо процессора от Intel, не подойдёт для AMD, обратное также верно. Существуют проблемы и менее очевидные, связанные с несовместимостью платы и типа оперативной памяти, или платы и подключения различных жёстких дисков, как то, например, наличие М.2 слота для подключения сверхбыстрых твердотельных накопителей.

Ещё сложнее проблемы оптимизации, которые могут меняться каждое поколение комплектующих, то есть раз в год-два. Например, проблема ранних процессоров AMD Ryzen, из-за привязки частоты передачи данных между ядрами к частоте оперативной памяти, которая очень сильно сказывалась на производительности компьютера. При медленной памяти процессор с большим трудом работал с многоядерными приложениями. Проблема сохранилась и на новых процессорах той же линейки, однако она сильно сглажена. В то время как на процессорах Intel или AMD других линеек подобной проблемы нет, и частота оперативной памяти влияет намного меньше.

Всё это также перекликается с проблемой энергопотребления. Компьютер это энергоёмкое устройство, разброс потребления у разных конфигураций огромен, от «офисных» 250-300 ватт, до 1200-1500 у лучших рабочих станций. В зависимости от целеполагания при конфигурации машины необходимо также рассчитывать, сколько энергии будет потреблять компьютер, и выбирать блок питания по этому параметру, а также по некоторым другим.

Количество дополнительных и основных разъёмов у разных комплектующих может отличаться. Например: видеокарты и разъёмы PCI Express на них. У офисных маломощных видеокартах этих разъёмов, по сути, разъёмов дополнительного питания, не существует. У карт среднего диапазона может быть один разъём PCIe x6 (шесть контактов) или PCIe x8 (восемь контактов), на более высокоуровневых картах могут быть различные комбинации, вплоть до двух PCIe x8 и одного PCIe x6. Если дополнительного питания не будет, то карта не сумеет запуститься. Питание же подаётся от блока питания.

Также через SATA подключаются дисководы и подобные устройства. Всё это нужно учесть.

Наличие же конфигуратора позволит решить эти и многие другие проблемы, и стать решением как для индивидуальных пользователей, у которых нет желания разбираться в том, как работает их компьютер, так и для компаний, процесс принятия решений снабжения конфигуратор поможет значительно упростить, как может упростить и принятие решений об обновлении машин в случае необходимости.

Цель работы: проектирование и разработка web-сайта конфигуратора

ПК. Для достижения поставленной цели необходимо решить следующие задачи:

– проанализировать все необходимые аспекты конфигурации;

– разработать требования к сайту;

– разработать базу данных с необходимыми комплектующими;

– разработать структуру и дизайн сайта;

– осуществить скриптовую реализацию функциональности сайта.

Объектом исследования выпускной квалификационной работы является технология поддержки в конфигурировании ПК.

Предметом исследования является процесс конфигурации ПК.

Практическая значимость, помимо означенных выше проблем, состоит в том, что разработанный конфигуратор позволит значительно упростить выбор компьютеров без излишних ограничений.

Методы исследования, применяемые в работе:

– анализ;

– сравнение;

– обобщение;

– моделирование.

**1 Описание предметной области**

Вопрос грамотного подхода к конфигурации ПК для любой фирмы, в которой применяются компьютеры (а это более 90% в 2017 году, и почти 94% в 2019 году, тенденция идёт к постепенной полной компьютеризации [1]) – вопрос оптимизации затрат и оптимизации рабочего времени: с одной стороны крайне нежелательна ситуация, когда компьютер на час работы десять минут фактически не работает, а с другой – тратить большую сумму денег на лучший на рынке вариант не имеет никакого смысла, если всё, что требуется от компьютера – офисная работа. Для обычного же пользователя это ничуть не меньший вопрос затрат и оптимизации, просто в меньших масштабах как имеющихся средств, так и количества необходимых машин.

Подобная компетенция существует у довольно ограниченного круга лиц, и даже люди, связанные с IT, особенно с программной частью, могут не знать аппаратную, а тем более – следить за изменениями в аппаратной части. Отдельного сотрудника нанимать также не выход: замена компьютеров требуется раз в три, а иногда и более, лет. Нередки случаи в офисах, когда люди работают на десятилетних и более старших машинах. Если крупная компания это может решить наймом отдельного технического специалиста, в перечне компетенций которого будет и эта, то мелкие и средние компании – не могут. Тем более этого не могут сделать простые пользователи. Эта разработка направлена на создание решения для этого процесса.

Оболочка решения может быть как приложением, так и сайтом. Однако был выбран сайт, и вот почему: пользователей Интернета в России сейчас 124 миллиона, при населении в 146 миллионов, что означает, что почти у каждого россиянина сознательного возраста есть доступ в Интернет. В то же время, большинство программ, которые ранее были только в виде приложений, переходят или в двойной вид: как приложение, так и сайт, либо же полностью забрасывают своё приложение, оставляя только сайт. В конфигураторе нет никаких необходимых функций, которые не мог бы покрыть сайт, а базы данных, которые могут занять существенный объём, не нужно будет загружать на компьютер. Совокупность факторов делает web-оболочку наилучшим решением как с точки зрения оптимальности в использовании, так и с точки зрения оптимальности в проектировании.

* 1. **Основные сведения о конфигураторах ПК**

Конфигураторы ПК – в современности являются довольно редким видом программ, и представлены, в основном, как приложение к магазинам. Их назначение очевидно, однако приоритетные задачи могут быть абсолютно разные у разных конфигураторов: это может быть задача «консультанта», которая состоит в том, чтобы показать характеристики предлагаемых вариантов, и проверить на физическую совместимость комплектующих. Обычно такую применяют в магазинах электроники, для заказов компьютеров через Интернет. Задачей также может являться задача «разделения опыта», когда пользователи Интернета собирают какие-то компьютеры, и они остаются в базе данных, чтобы другой пользователь смог просто взять такой же. То есть конфигуратор должен в любом случае представлять возможности:

– возможность проверки на физическую совместимость комплектующих;

– возможность сохранения/загрузки ранее сделанной конфигурации;

– возможность подробного отображения важных характеристик.

Также конфигуратор должен стирать или минимизировать проблемы конфигурации компьютера, такие как:

– проблема возможных малых знаний пользователя. Иногда индивидуальный пользователь совершенно не разбирается в компьютерах, и у него нет какого-то вспомогательного материала. Для этого иногда мало физической совместимости, а нужна также условно «логическая» совместимость, чтобы никакое комплектующее не вызывало эффект «бутылочного горлышка» для производительности;

– проблема «альтернатив». Почти любому комплектующему есть аналог или из другой линейки, или от другой компании. Даже такие олигополии, как рынок процессоров, обычно имеют аналоги. Однако в зависимости от ситуации на рынке цены на аналоги могут быть несопоставимы. Например, на момент написания работы цена процессора АМД была равна 11 000 рублей, а аналога от Интел – чуть более 20 000 рублей [2]. Также существует возможность полного отсутствия какой-либо детали на рынке, как, например, в 2020-2022 произошло с видеокартами, продолжается и сейчас с процессорами, а началось ещё в 2020, до этого произошло точно такое же, с видеокартами, в 2018 году, и плавно перетекло в текущий кризис. При этом различные карты разных компаний затронула по-разному. Предсказать это, без досконального знания нескольких областей знаний, невозможно. Поэтому конфигуратор должен предлагать максимально широкий список примерно равноценных альтернатив;

– проблема «излишних сущностей». Отсылает к первой проблеме, и является или результатом избыточной попытки решить её, или полным игнорированием любых конфигураторов. Различных решений под различные задачи много. Даже внутри рынка видеокарт их десятки, а внутри рынка мониторов их могут быть тысячи, а кулеров или ОЗУ могут быть десятки тысяч. Если не отсекать лишнее как, например, неподходящее для задачи, или неоптимальное в данном случае, то проблема возможной малой компетенции пользователя сделает конфигуратор просто бесполезным: для пользователей, знающих рынок, в конфигураторе нет пользы, а пользователям, которые не знают ничего о рынке комплектующих, эта проблема создаст ещё больше неудобств, поэтому решение этой проблемы должно быть приоритетным.

Именно эту проблему почти не затрагивают все имеющиеся на момент написания конфигураторы ПК, поэтому на неё нужно обратить повышенное внимание.

* 1. **Существующие конфигураторы ПК.**

Прежде, чем приступать к разработке, стоит проанализировать уже существующие аналоги, и понять их основные проблемы и основные достоинства. Таких сайтов немного, как было указано выше, и сравнивать их можно будет по тому, как они решают вышеописанные проблемы, а также по тому, дают ли они те возможности, которые представлены выше.

*DNS.*

Конфигуратор известной сети DNS [3], стартовый экран которого представлен на рисунке 1.1.

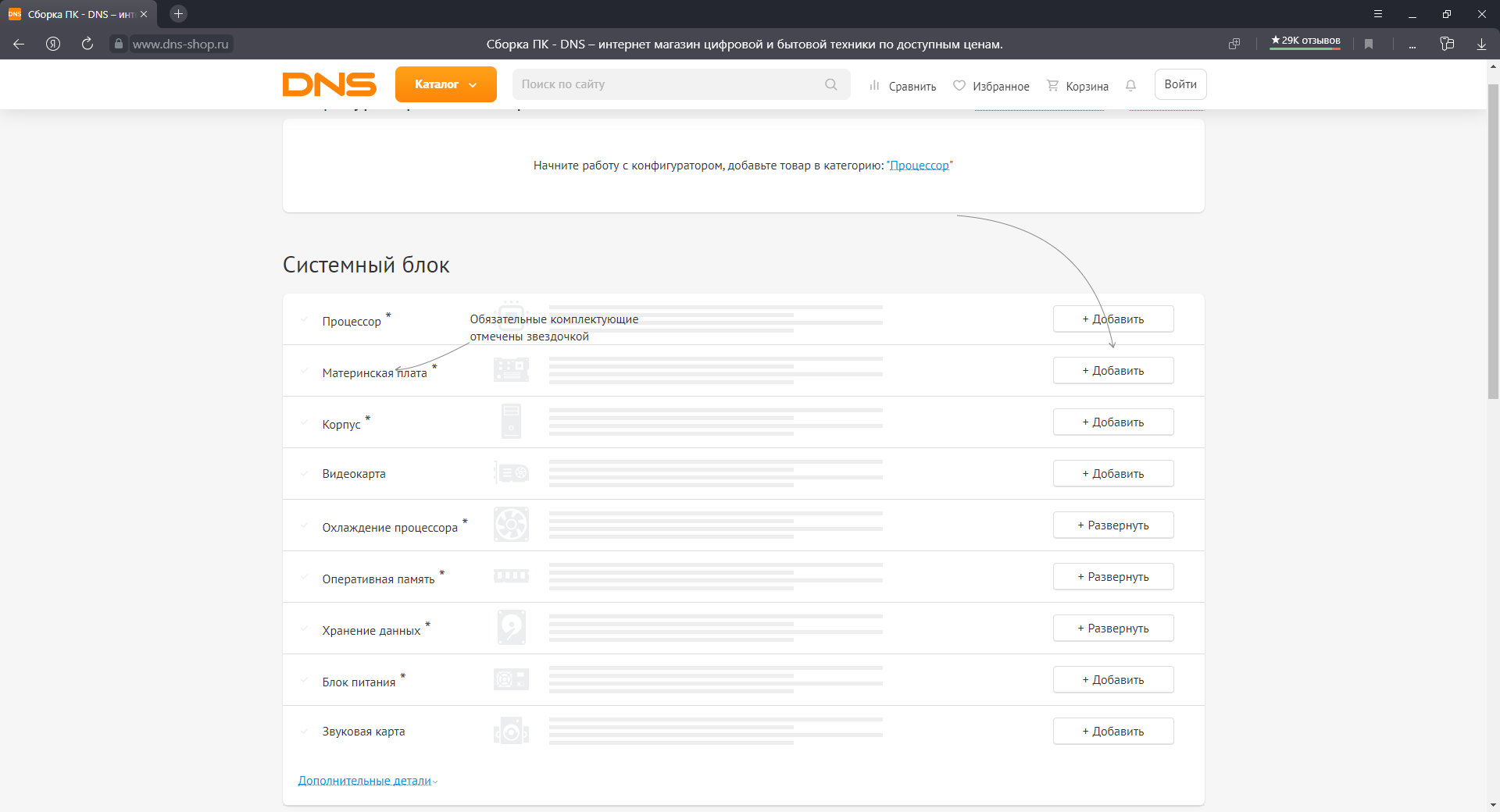


Рисунок 1.1 – Стартовый экран конфигуратора ПК от DNS

Первая проблема этого конфигуратора: необходимо выбрать процессор  
из всего обилия в DNS, и конфигуратор не помогает пользователю, оставляя его с всем каталогом магазина, в котором представлены почти все современные, и даже некоторые устаревшие модели, без разделения их на какие-либо группы. Однако у процессоров демонстрируются характеристики, что позволит сориентироваться более опытному пользователю. Экран выбора процессоров представлен на рисунке 1.2.

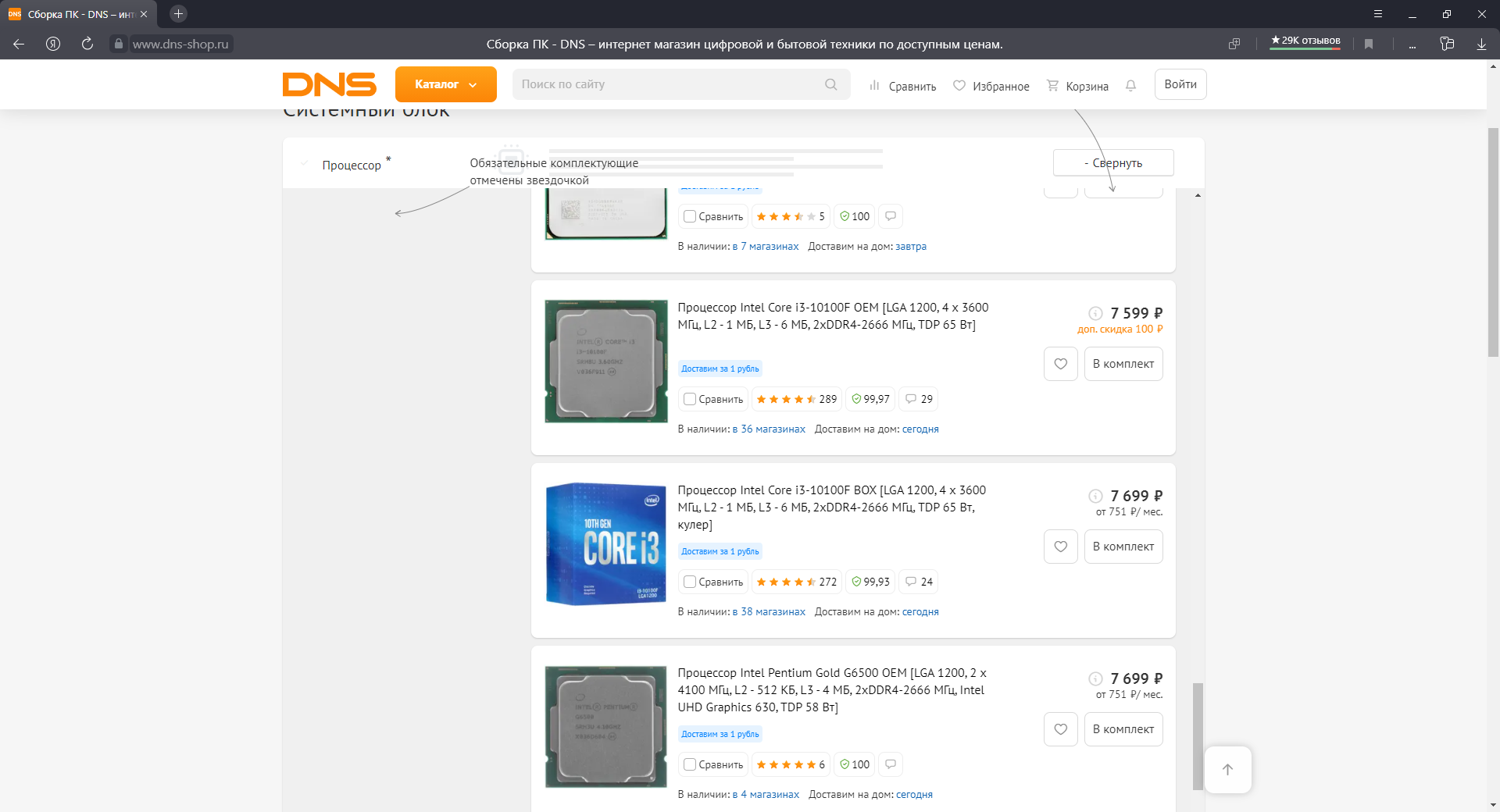


Рисунок 1.2 – Экран выбора процессора у конфигуратора DNS

Далее конфигуратор просто ведёт по простейшему алгоритму совместимых товаров, то есть проверяет физическую совместимость с любым из выбранных процессоров материнской платы, за ней – корпуса, за ним – видеокарты, и так далее. То есть исполняет лишь необходимый минимум, но не мешает, например, взять медленную ОЗУ к первой линейке Ryzen, как приводилось выше. То есть не решает ни проблему излишних сущностей, ни, в достаточной степени, проблему возможных малых знаний.

Из плюсов можно отметить, что, из-за ассоциации с магазином, конфигуратор указывает актуальные на момент сборки цены, а также позволяет подобрать периферию и даже ПО для компьютера. Ещё можно отметить полное соответствие возможностям, перечисленным выше.

*Ситилинк.*

Конфигуратор от Ситилинка [4], так же, как и ДНС, являющегося крупным магазином, представляет собой несколько иную идею. Стартовый экран представлен на рисунке 1.3.

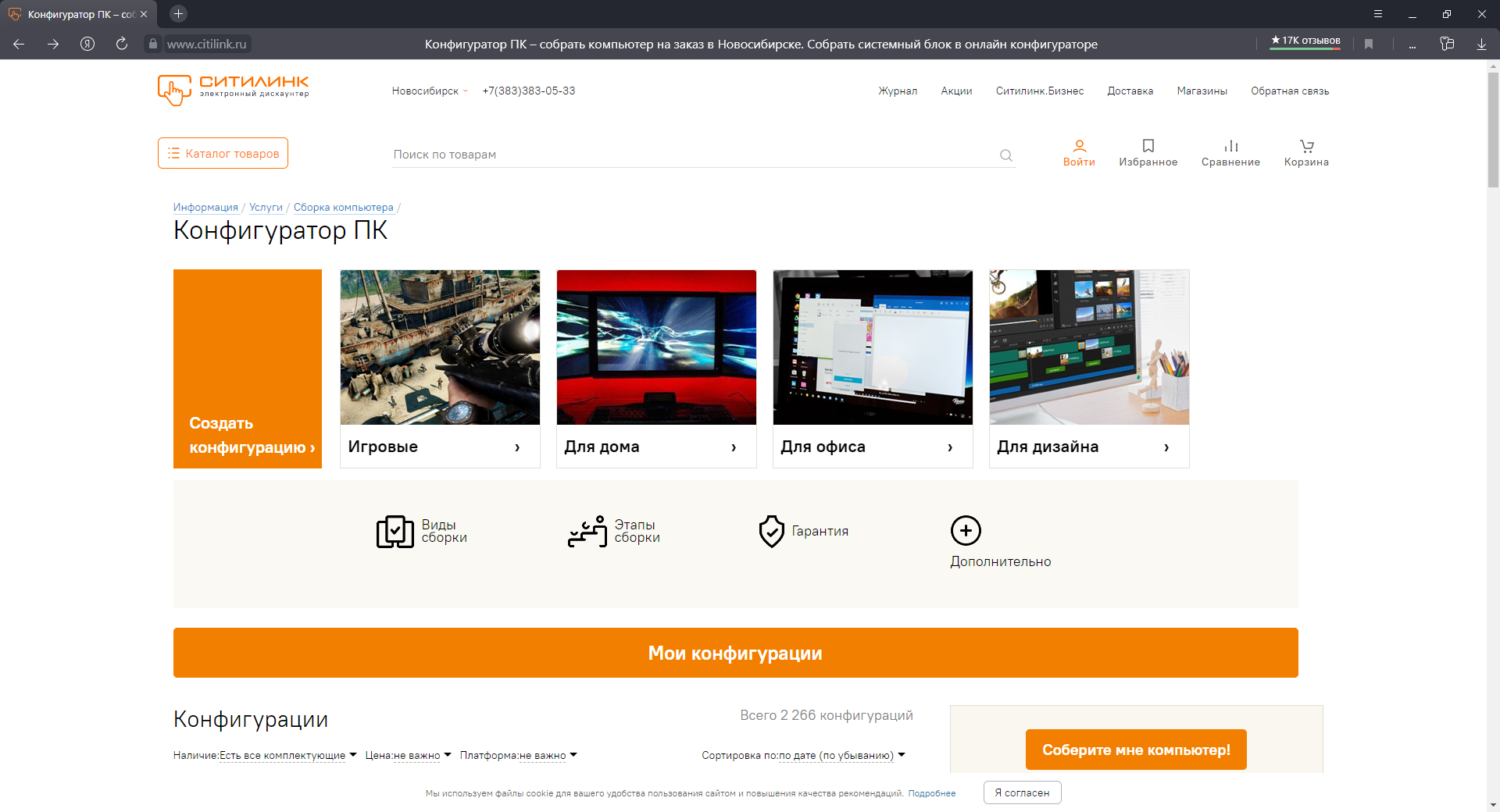


Рисунок 1.3 – Стартовый экран конфигуратора ПК от Ситилинк

Здесь в основу положена идея «разделения опыта». Всё ещё возможно создать свою конфигурацию, однако этот раздел не отличается от DNS, однако на главном экране также представлены пользовательские сборки. У них указывается и назначение, которое «Ситилинк» решил указать четырёх типов, видных на рисунках 1.3 и 1.4. Общая проблема, однако, также находится и в отсутствии модерации, которая позволяет малокомпетентным людям создавать свои сборки крайне сомнительного назначения и качества, и нет никакой рейтинговой системы их оценивания.

На рисунке 1.4 есть примеры пользовательских сборок.

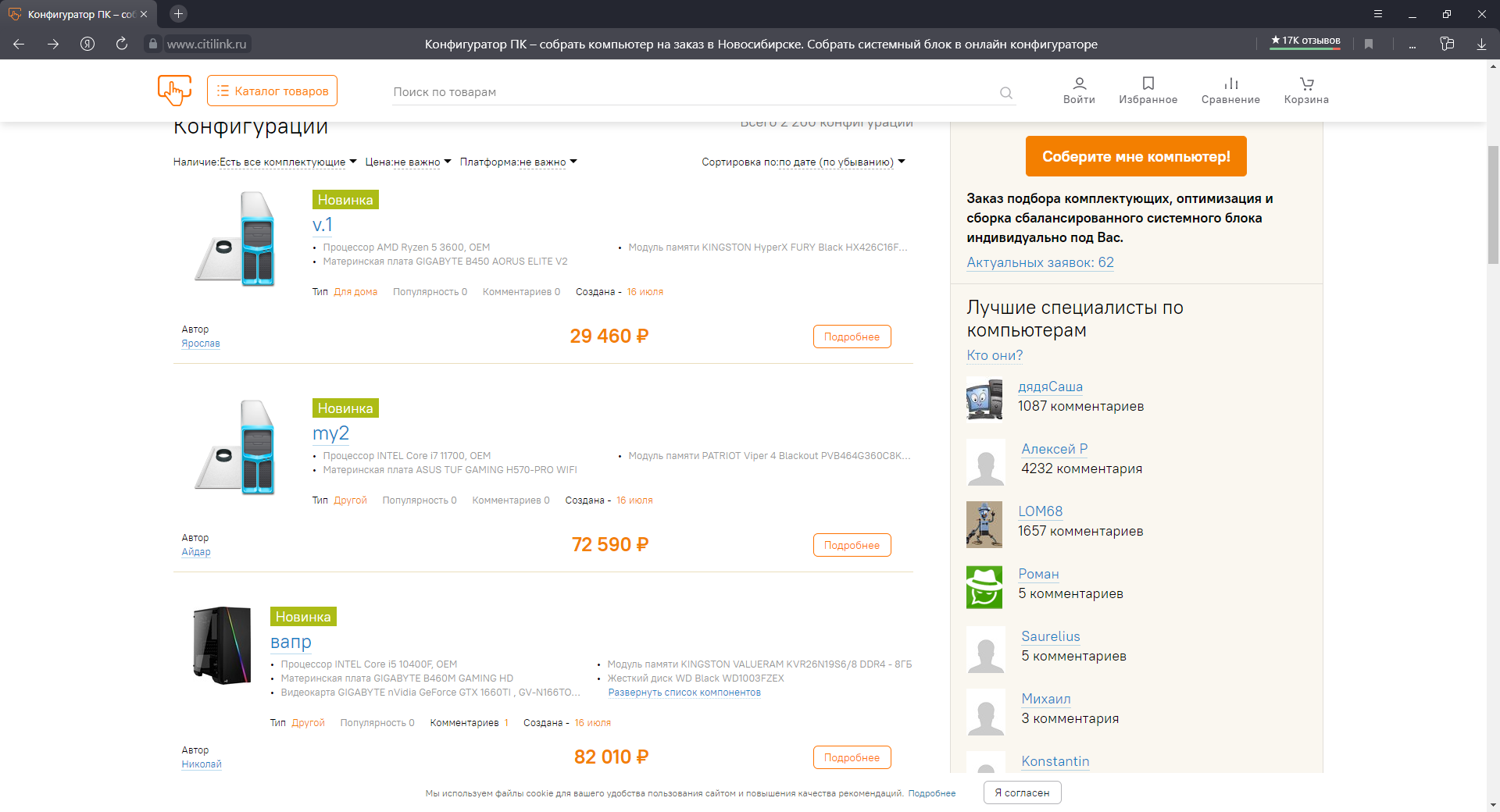


Рисунок 1.4 – Примеры пользовательских сборок на конфигураторе «Ситилинк»

Также существует возможность сборки компьютера командой «Ситилинк», что следует из того, что это – крупный магазин, как и наличие актуальных цен, как и в DNS. Две проблемы конфигуратора «Ситилинк» – отсутствие модерации пользовательских сборок, из-за чего часть попадает явно не в ту категорию, и проблема излишних сущностей, которая решается здесь чуть лучше, чем у DNS, но всё же не решена до конца. Все возможности, однако, в наличии, а также есть возможность поделиться опытом, и сделать несколько сборок, что может привлечь опытных пользователей, которым конфигуратор не нужен.

*Арсенал+.*

Конфигуратор сети «Арсенал» [5] во многом напоминает конфигуратор DNS. Стартовый экран представлен на рисунке 1.5.

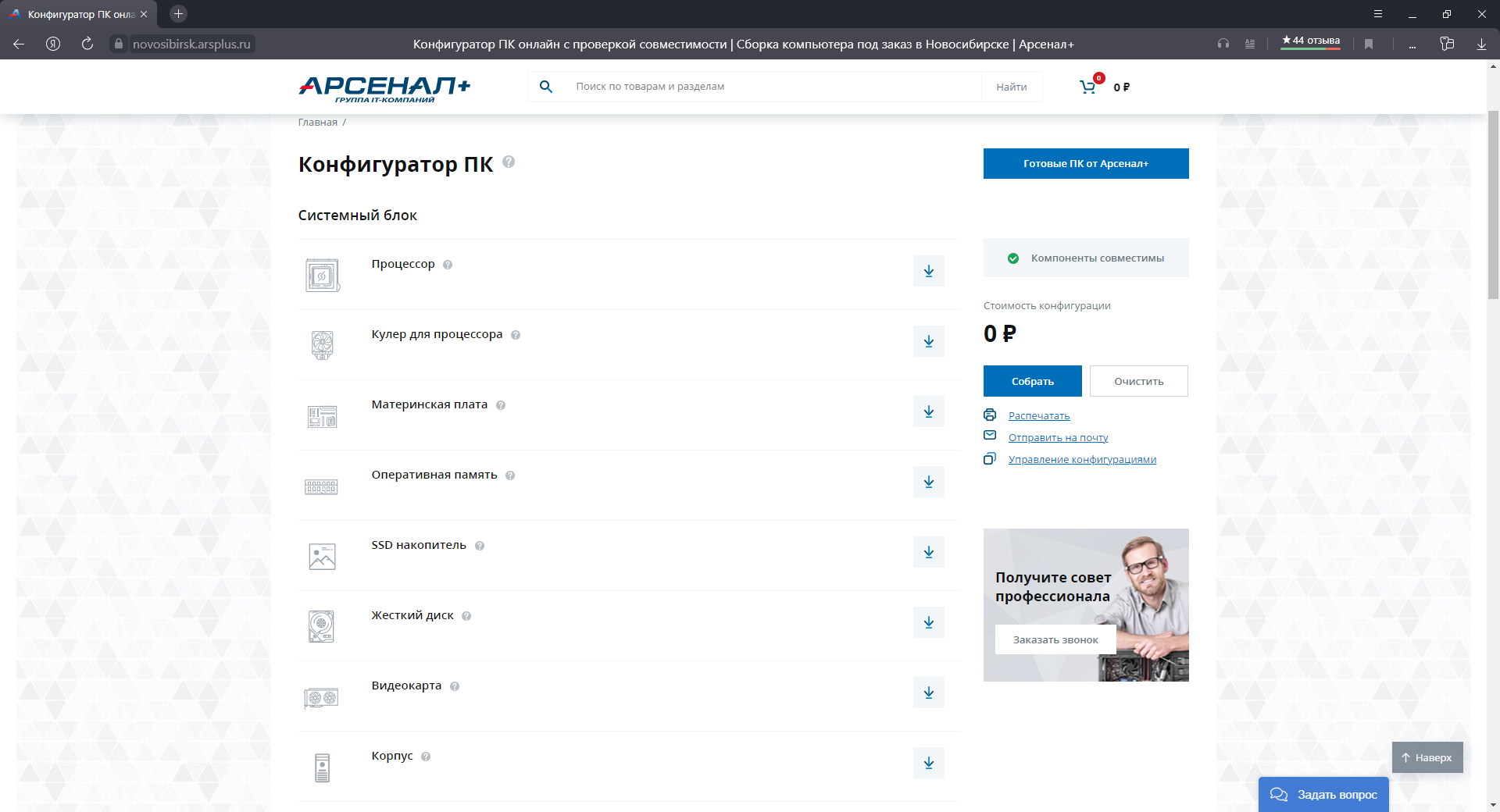


Рисунок 1.5 – Стартовый экран конфигуратора «Арсенал+»

От DNS, однако, «Арсенал» отличают уже готовые решения, которые, впрочем, не являются на самом деле их готовыми решениями, а являются готовыми решениями изначальных поставщиков. Проблема же с решениями изначальных поставщиков в многократной наценке: на этапе закупки комплектующих, на этапе сборки, на этапе закупки готовой сборки, на этапе покупки в магазине.

Ещё одно отличие состоит в том, что здесь нет Hard-lock на выбор несовместимых элементов, а Soft-lock работает не всегда. Например, можно выбрать процессор HEDT-уровня, для которого не подходят стандартные кулеры, но выбрать стандартный кулер. Конфигуратор должен, мягко или жёстко, запрещать подобные действия, однако на рисунке 1.6 показано, что подобное действие не является для конфигуратора ошибочным.

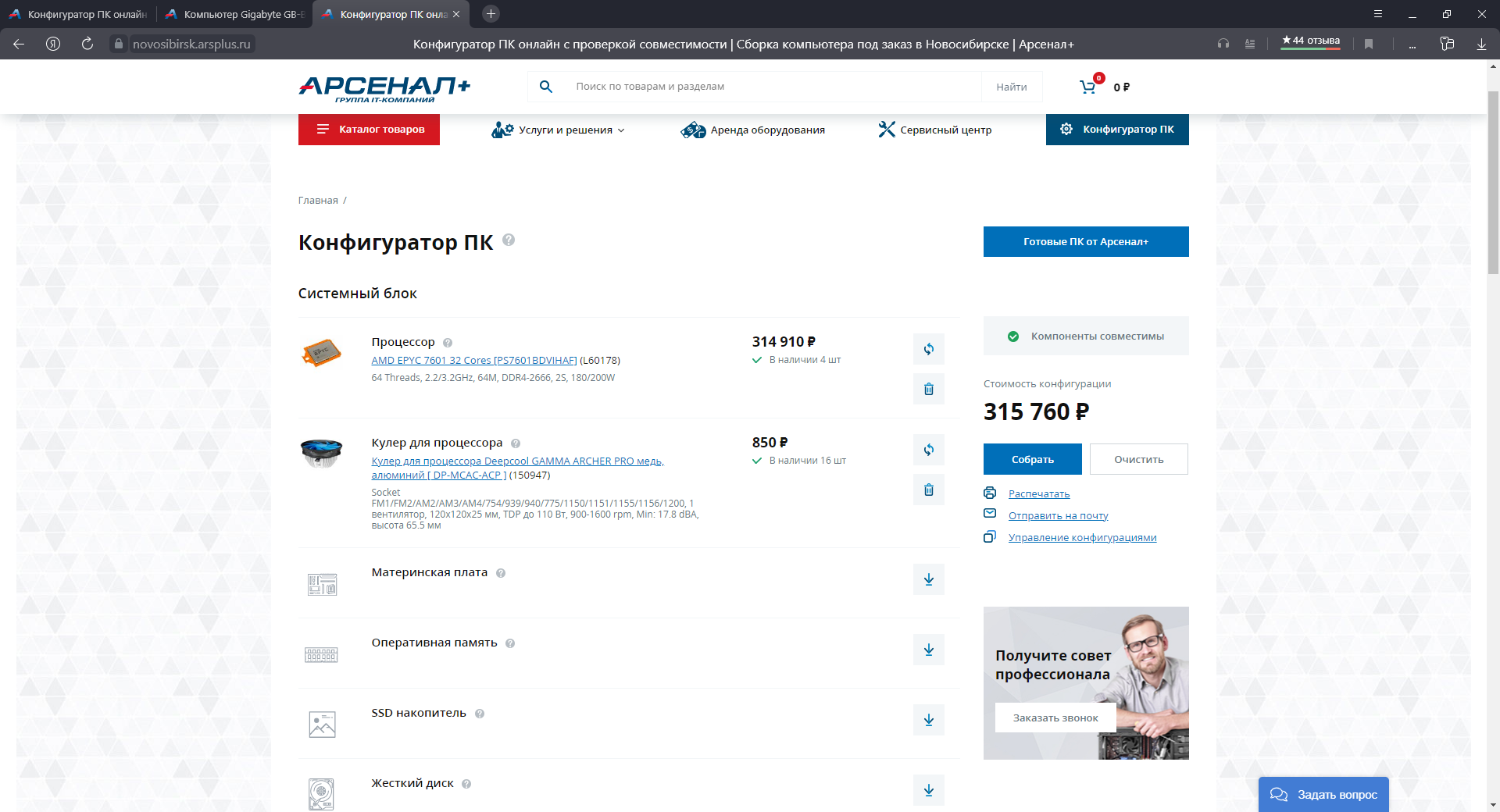


Рисунок 1.6 – Сочетание HEDT-процессора и стандартного кулера считается «совместимым»

Стоит отметить отдельно, что у процессора в этом магазине есть графа «сокет», то есть разъём, а у кулера есть список поддерживаемых сокетов. И сокета HEDT (SP3) у этого кулера в списке нет. Однако, даже если бы этот кулер было возможно поставить на этот процессор, то его фактической производительности не хватило бы на то, чтобы охладить процессор.

В этом конфигураторе не решается ни одна из проблем как логической, так и физической совместимости, также нет пользовательских сборок, которые могли бы сгладить этот недостаток.

* 1. **Итоги анализа конфигураторов**

Из проанализированных конфигураторов [6] только два могут вообще рассматриваться как инструмент помощи пользователям без нужной компетенции. Конфигуратор «Арсенал+», по причине отсутствия даже минимальной защиты от некорректной сборки, пусть такая и является маловероятной в связи с ценой, не может являться адекватным выбором. Оба оставшихся конфигуратора представлены крупными сетями, а, следовательно, оба имеют много общих черт, как то: наличие актуальных цен, выбор только из тех позиций, которые в наличии или заказаны магазином. Однако у конфигуратора «Ситилинк» есть разделение сборок на типы, хотя из-за отсутствия модерации работа этого разделения не всегда корректна. Также у «Ситилинк» есть пользовательские сборки, что является хорошей идеей при грамотной реализации.

Однако проблему лишних сущностей не решает ни один из них, и, следовательно, ни один не является исчерпывающим все потребности, поставленные в работе выше.

* 1. **Выводы по разделу**

В первом разделе рассмотрены основные понятия и термины, относящиеся к конфигураторам и конфигурации персональных компьютеров, описаны основные проблемы и необходимые возможности конфигуратора. Произведён анализ и сравнение трёх разных конфигураторов ПК, сформированы достоинства и недостатки каждого.

Исходя из этого анализа конфигураторов сделан вывод, что ни одна из готовых систем полностью не удовлетворяет всем потребностям, означенным в работе. Сделан вывод о необходимости создания системы помощи в конфигурации ПК в Web-оболочке, так, чтобы она наилучшим образом подходила под требования.

Разработанная система должна позволить собрать компьютер даже человеку с нулевыми или крайне недостаточными знаниями о рынке комплектующих, иметь гибкость, достаточную для её администрирования, с целью внесения изменений, таких, как добавление свежих, недавно вышедших, комплектующих, либо удаления из базы данных старых комплектующих, вышедших из продажи как в крупных магазинах, так и в маленьких, в которых определённые виды комплектующих задерживаются дольше, чем в сетях.

**2 Проектирование информационной системы**

Так как итогом процесса должен быть сконфигурированный ПК, то все процессы должны быть направлены именно на то, чтобы получить как можно более соответствующий заданным критериям компьютер. Необходимо перенести компетенцию конфигурации ПК на программную основу. Для этого, изначально, нужно разработать концепт проекта. Концепт необходим как примерный план логики и требований к программе. Разделить концепт, в этом случае, можно на несколько частей:

– сбор основных требований;

– прототипирование баз данных;

– описание логики.

Этот концепт и требуется разработать. Более того, этот концепт, во многом, определит и бизнес-процессы, особенно бизнес-процесс администрации сайта. Дизайн сайта было решено не включать в концепт из-за простоты и возможности его разработки и без изначального концепта. Можно оговориться, что сайт, так как он играет, по факту, вспомогательную роль, должен быть минималистичен, удобен в использовании, а также максимально стандартен. Чуть более подробно эту тему раскрывают основные требования.

**2.1 Основные требования**

Требования также можно разделить на две части. Условно можно назвать их «технической» и «логической». С технической стороны всё более просто: сайт, очевидно, должен запускаться на любом распространённом браузере. Таковых, по исследованию сайта Statcounter, проводившегося с мая 2021 по май 2022, шесть. Рисунок 2.1 является иллюстрацией результатов исследования.

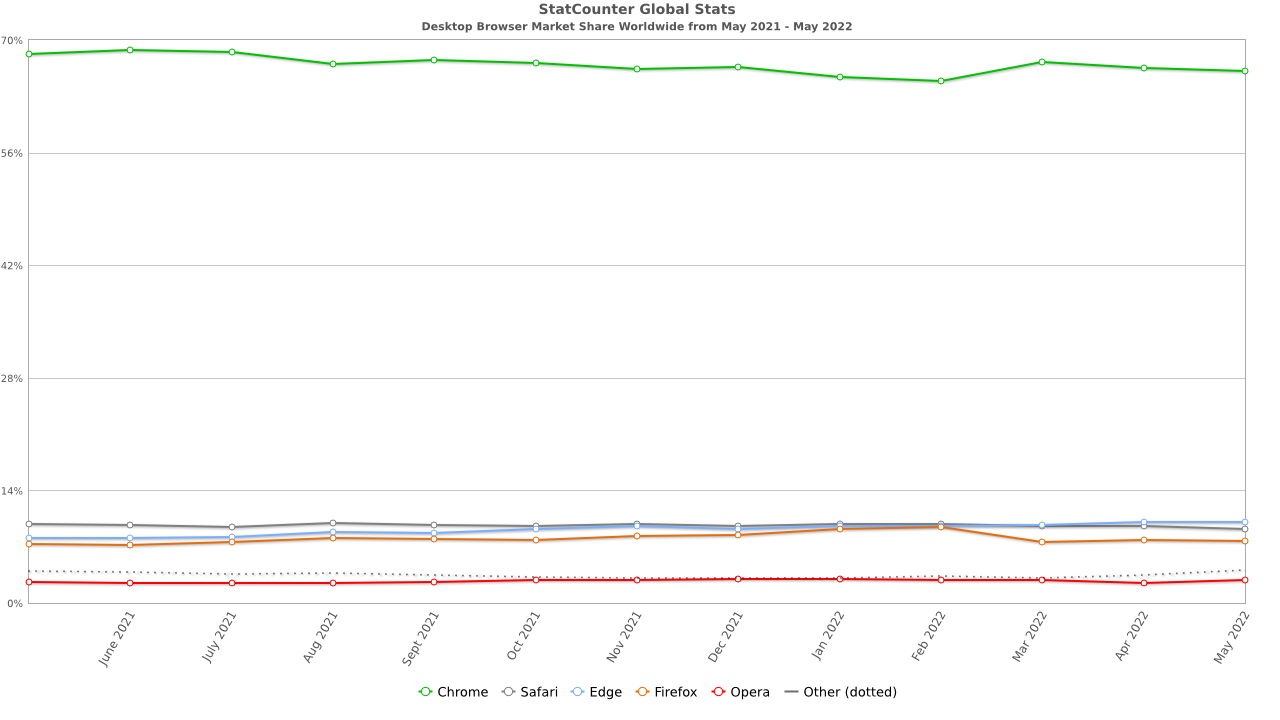


Рисунок 2.1 – Исследование Statcounter

Для более наглядного рассмотрения современного (на май 2022 года) момента на сайте прилагаются и подробные численные описания распространённости браузеров на этот период, которые представлены на рисунке 2.2.



Рисунок 2.2 – Распределение наиболее популярных браузеров

Однако здесь представлены мировые тренды распространённости браузеров, которые, хотя и нужно учитывать, но в рамках данного этапа работы они являются больше ориентировочными, а не строго обязательными, так как основное использование сайта предполагается, всё-таки, в российском регионе. На рисунках 2.3 и 2.4 представлены соответствующие рисункам 2.1 и 2.2 значения аналогичных исследований для России за тот же период.

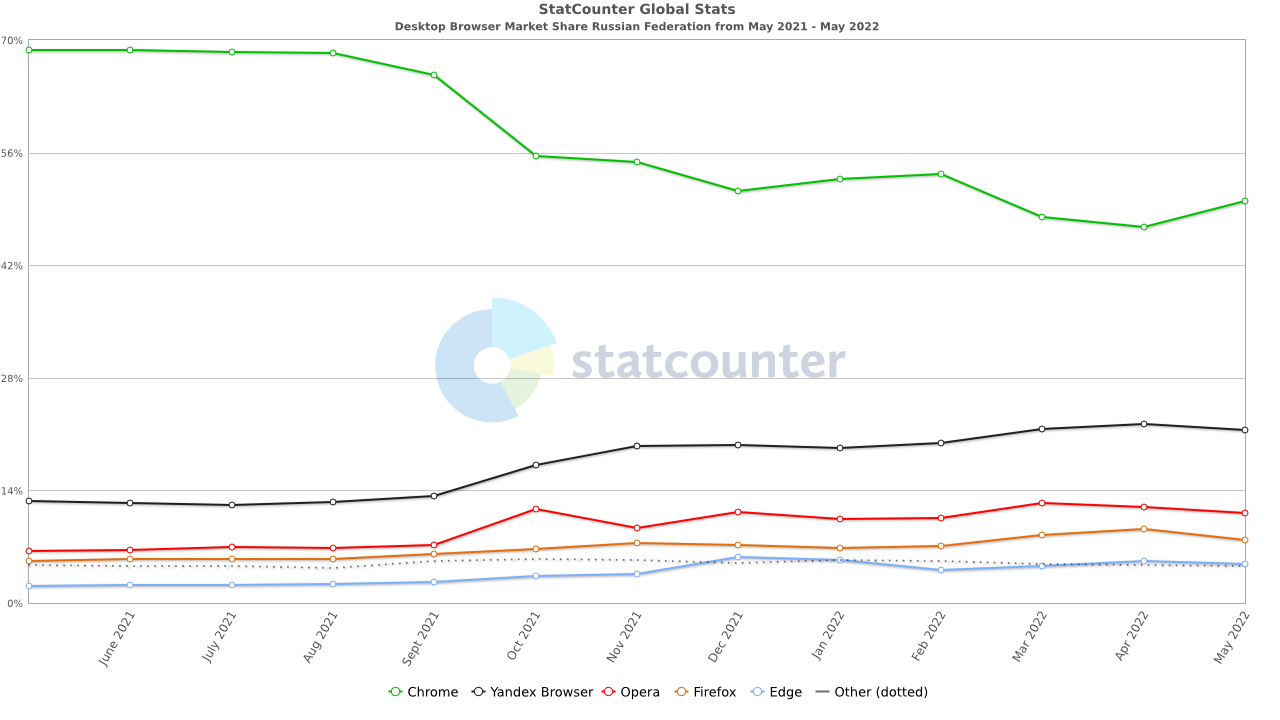


Рисунок 2.3 – Популярность браузеров в России, исследование Statcounter

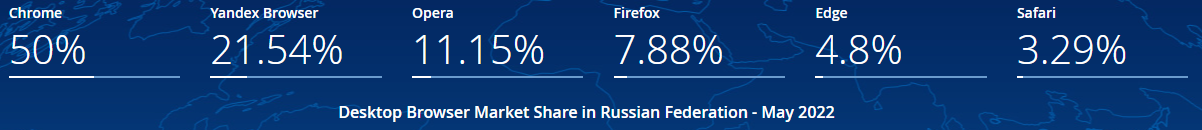


Рисунок 2.4 – Распределение наиболее популярных браузеров в России

Нужно оговориться, что IE и Edge – браузер от одних и тех же разработчиков, и их суть похожа. Более того, IE неизбежно уйдёт окончательно, так как Edge это попытка заместить сомнительный браузер, предустановленный в Windows ранее. Также стоит отметить, что Яндекс.Браузер (российская разработка) и Google Chrome основаны на одном движке, Chromium, и почти гарантирован запуск на одном того сайта, что корректно работал на другом. Следовательно, лист выглядит так:

– Chrome (Yandex);

– Opera;

– Firefox;

– Edge (IE);

– Safari.

Помимо этого технического требования, которое, очевидно, включает и возможность запуска означенных браузеров на компьютере (телефоне), нужно также осуществить возможность масштабирования, чтобы сайт без проблем запускался на любом устройстве, будь то компьютер, смартфон или планшет, и не был слишком большим и неудобным, либо же слишком маленьким. И, разумеется, необходимо обеспечить приемлемое время доступа к базе данных и приемлемое время обработки запросов.

С логической точки зрения всё несколько сложнее: необходимо осуществить такой алгоритм, и такие критерии оценивания, чтобы компьютер, сконфигурированный программой, можно было бы охарактеризовать как «подходящий», что включает в себя: отсутствие избыточности, полную совместимость обоих видов, которые была означены в первом разделе, достаточность мощности для решения задач, задекларированных в опросе, что предполагает также корректность вопросов опроса. Это, на концептуальном уровне, все требования, которые предъявляются к сайту и программе на нём.

**2.2 Прототипирование баз данных**

Самая сложная часть программы – базы данных. Очень сложно отличить процессор, нужный для игр, от процессора, нужного для графики. Ведь для обеих целей нужны мощные многоядерные процессоры. Таких узких мест достаточно много, и для некоторых из деталей будет необходимо ввести эвристические константы.

Первое, что для этого нужно: обширная база данных всех находящихся в свободной продаже комплектующих, а также тех, что уже вышли из продажи, однако могут оставаться в разных магазинах, то есть ещё одно поколение назад, от находящихся в свободной продаже. Для начала – необходимо определить все сущности. Очевидно, что внутри этой работы невозможно перечислить абсолютно все комплектующие, поэтому придётся допускать некоторый уровень абстракции на менее критических элементах, как то блок питания, где можно оставить только мощность, или оперативная память, где достаточно оставить только форм-фактор и объём, но основные и самые критичные детали, такие как процессор или видеокарта, должны быть перенесены максимально полно. Предполагаемые необходимые сущности и атрибуты вынесены в таблицах 2.1-2.3.

Таблица 2.1 – Сущность «процессор» и её атрибуты

|  |  |
| --- | --- |
| Процессоры | |
| ID | Int (primary key) |
| Модель | Text |
| Количество ядер | Int |
| Количество потоков | Int |
| Назначение | Int |
| Рекомендуемая цена($) | Int |
| Разъём | Text |
| Тепловыделение (W) | Int |
| Тип поддерживаемой оперативной памяти | Int |

Как можно заметить, помимо чисто технических и экономических характеристик, добавлена строка «назначение». Она является частично искусственной, однако собой заменяет довольно большое количество вычисляемых данных. Как то – операции на такт, сложный расчёт ядер, потоков, и операций на такт, и так далее. Атрибут «назначение» является трёхвариантным: 0, 1 и 2. Так, для простоты, указываются офисное, игровое и рабочее предназначение соответственно. Если между офисным и всеми остальными разница очевидна, то вот между игровым и рабочим она несколько размыта: игровой процессор имеет меньше ядер, чем рабочий, ввиду использования малого количества ядер играми (от 1 до 10 в самом лучшем представителе), однако каждое ядро должно, во-первых, иметь высокую тактовую частоту, а во-вторых – высокую плотность операций на такт. Конечно, быстрые ядра не помешают и рабочим процессорам, однако в выборе между большим количеством ядер и мощности отдельного ядра для рабочих процессоров следует выбирать многоядерность.

Таблица 2.2 – Сущность «видеокарта» и её атрибуты

|  |  |
| --- | --- |
| Видеокарты | |
| ID | Int (primary key) |
| Модель | Text |
| Тепловыделение (W) | Int |
| Стандарт PCI (e) | Int |
| Рекомендуемая цена | Int |
| Уровень производительности | Int |

Здесь также есть эвристический атрибут «уровень производительности», который необходим для разбиения видеокарт на крупные группы по производительности. Это, опять-таки, необходимо для сокращения сложных расчётов, как и в случае с процессорным «назначением».

Здесь предлагается взять величины от 0 до 9. На девятом уровне будет современный флагман RTX 3090, и все аналогичные ему, если таковые будут. На восьмом – RTX 3080/ti, и все близкие к ним, и так далее. На нулевом уровне будут самые маломощные из включенных в рассмотрение карт. Видеокарты ещё низшего уровня нет смысла включать в БД.

В последующих версиях эту переменную будет необходимо заменить на математическую формулу, высчитывающую примерно то же самое, здесь «Уровень производительности» является некоторым математическим допущением, которое можно исправить в будущем, путём внесения изменений в архитектуру программы.

Таблица 2.3 – Сущность «чипсет» и её атрибуты

|  |  |
| --- | --- |
| Чипсет | |
| ID | Int (primary key) |
| Модель | Text |
| Сокет | Text |
| Поддерживаемая компания | Text |
| Поддерживаемое поколение | Text |

С чипсетами ситуация чуть более сложная: как один чипсет может поддерживать несколько поколений, так и несколько чипсетов гарантированно поддерживают одно поколение, в зависимости от уровня материнской платы, следовательно, необходимо сделать некоторое «сочетание», которым и стала сущность «чипсет». Она соединяет в себе расчётные компоненты и обращение к базам данных, так как нет общего случая даже для одного процессора, и нужны некоторые дополнительные данные.

Теперь, для удобства представления, всё это необходимо объединить в некоторое условное сочетание, которое будет называться «компьютер», который, в сущности, представляет собой запрос, который делает сайт через код к базе данных комплектующих. Суммарное представление показано на таблице 2.4.

Таблица 2.4 – Условная сущность «компьютер» и её атрибуты

|  |  |
| --- | --- |
| Компьютер | |
| Процессор | Int |
| Видеокарта | Int |
| Чипсет | Int |

Для более наглядного представления имеет смысл сделать ER-диаграмму, отражающую примерную структуру сущностей базы данных, что представлено на рисунке 2.5.

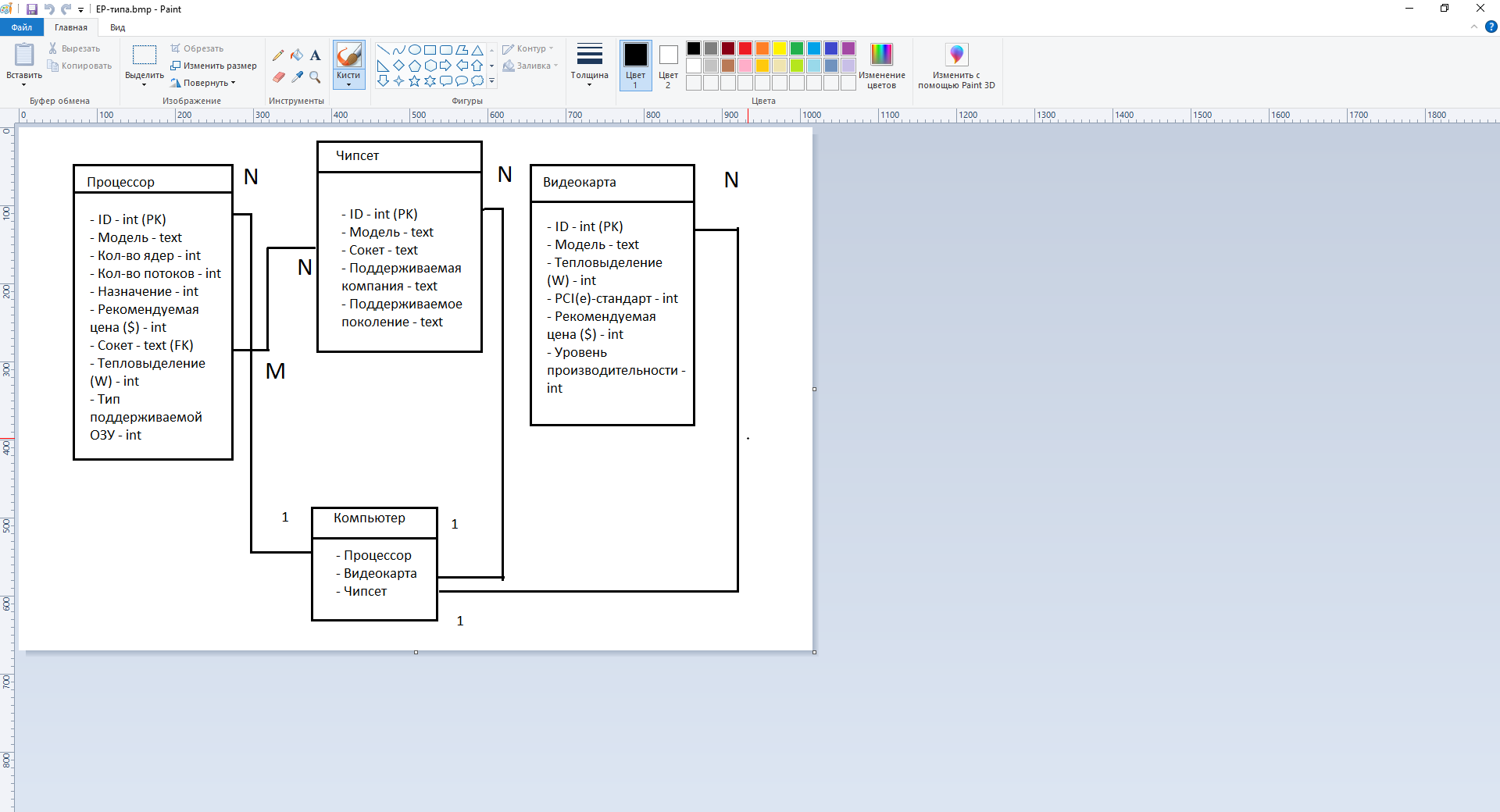


Рисунок 2.5 – ER-диаграмма БД

Некоторые другие, безусловно, необходимые элементы компьютера, такие, как оперативная память или кулер, не имеет смысла помещать в БД, ввиду с одной стороны излишней сложности в учёте каждой модели каждой компании, для оперативной памяти это более 1000 активных моделей, друг от друга аналоги отличаются или абсолютно незначительно, или крайне мало, поэтому достаточно высчитывания рекомендуемого объёма, который будет определяться внутри логики программы, а также не имеет смысла размещать жёсткий диск, так как под разные стили требуется абсолютно разный жёсткий диск, и рекомендовать что-то тут сложно, а разъём у жёстких дисков (не учитывая твердотельные) всегда одинаковый. Сама рекомендация наличия/отсутствия SSD (твердотельных накопителей) будет выводиться из расчётных характеристик, как и рекомендация по рассеиванию тепла от кулера.

По сущности – у всех этих сущностей такая же проблема, как и у оперативной памяти, из-за чего их не имеет смысла включать в БД.

**2.3 Выбор среды разработки**

В первую очередь необходимо вновь указать, какие языки были избраны для реализации программы. Это Javascript, как язык логики, это HTML как язык разметки сайтов. Также будет необходима база данных, а, следовательно, и какая-то СУБД.

Для начала нужно сформулировать критерии, по которым будет отбираться среда разработки. В первую очередь – свободное ли распространение у этой среды? Современность требует осторожного подхода к любым платным и несвободно распространяемым программам и тем более средам. Второе – системные требования. Стационарный ПК может быть любой мощности, но для максимально удобной работы среду должно быть возможно запустить и на самом слабом ноутбуке. Третье – желательно, чтобы среда разработки имела большой срок функционирования и популярность, ввиду большей вероятности найти решение возможных проблем при разработке.

Самый простой выбор для HTML. HTML можно писать даже в стандартном «блокноте», что, конечно, очень удобно в разрезе использования где угодно, но имеет ряд ограничений в вопросе удобства. Автозаполнения и подсветки в «блокноте» нет. Можно рассмотреть варианты, указанные в таблице 1, в неё вошли исключительно те программы, в которых есть основной перечень удобств, вроде подсветки кода, а также обладающие популярностью у разработчиков.

HTML-среды требуют крайне мало ресурсов, так как, в сущности, не имеют компилятора, в роли которого выступает браузер, а также не имеют компоновщика, однако для офисных ноутбуков программирование даже в самых лёгких средах может быть сопряжено с проблемами. Впрочем, если требования меньше, чем возможный минимум железа для ноутбука, можно ограничиться отметкой о том, что среда разработки запустится в любом случае, выбор таковой представлен на таблице 2.5.

Таблица 2.5 – Сравнение HTML-сред разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Название среды | Notepad++ | Brackets | Sublime |
| Системные требования | Минимальные. | Минимальные | Минимальные |
| Свободное распространение | + | + | - |
| Дата выпуска | 2003 год | 2014 (поддержка прекращена в 2021) | 2008 |
| Расширение | +, обширные | + | + |
| Поддержка ОС | Windows, ReactOS | Win, Linux, MacOS | Win, Linux, MacOS |
| Функционал | Десятки языков «из коробки», сотни в пакетах | Исключительно веб-языки | Десятки языков «из коробки», ещё несколько десятков в пакетах |

Исходя из сравнительного анализа был выбран Notepad++, как программа с наибольшей историей, свободно распространяемая, а также с максимумом расширений и минимумом сложностей с переносом на другую машину.

В Notepad++ можно также программировать и на Javascript, так что скриптовую часть логики можно написать внутри того же Notepad++.

К выбору СУБД можно предъявить все те же требования, что и к прошлым средам разработки, результаты показаны в таблице 6. Однако прямые системные требования не могут быть применены, так как всё зависит от размера базы данных, поэтому в таблице 2.6 не будет указания системных требований.

Таблица 2.6 – Сравнение СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | MySQL | MongoDB | Firebase |
| SQL-запросы | + | NoSQL | NoSQL |
| Соединение с JS | + | + | + |
| Поддержка ОС | Win | Win | Win |
| Интерфейс | Привычный для разработчиков БД на SQL | Классический NoSQL | Свой |
| Свободное распространение | + | -, распространение остановлено в России | -, хост на облаке |

Была выбрана СУБД MySQL, как свободно распространяемая СУБД с SQL-запросами, то есть подобная любой другой SQL-СУБД.

**2.4 Прототипирование логики и дизайна**

Для работы программы необходим старт, то есть – результаты опроса, который представлен на сайте. Список вопросов представлен на рисунке 2.6.

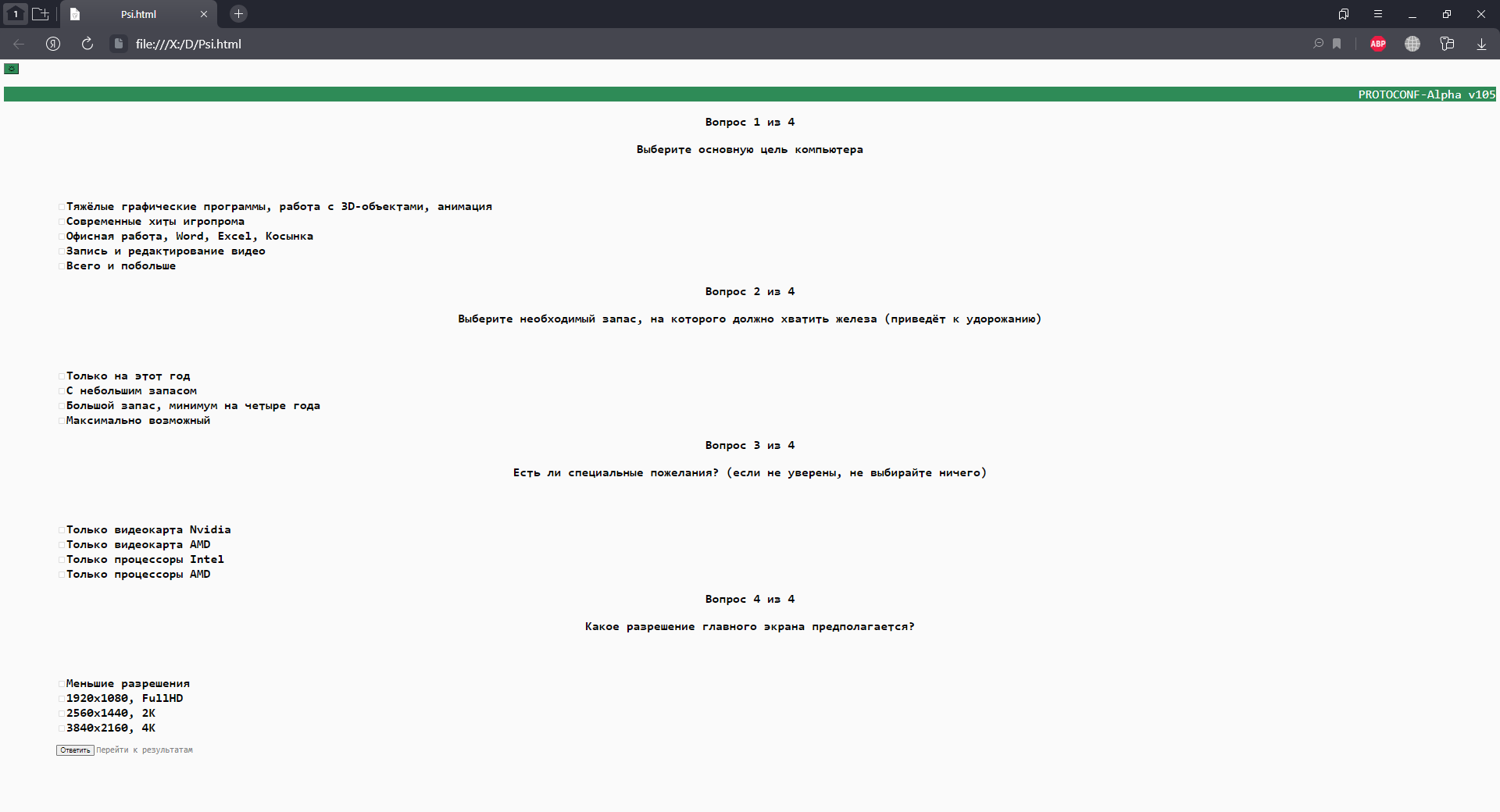


Рисунок 2.6 – Все вопросы

На рабочем прототипе вопросы будут выводиться по одному, и изменения в переменных будут происходить после каждого ответа. Каждый вопрос по-разному влияет на разные переменные. Их предполагается, пока что, четыре. На основе переменных, отвечающих за назначение и расчётную мощность компьютера, из базы данных будет сгенерирован компьютер, и выведен в готовую форму, которая представлена на рисунке 2.7.

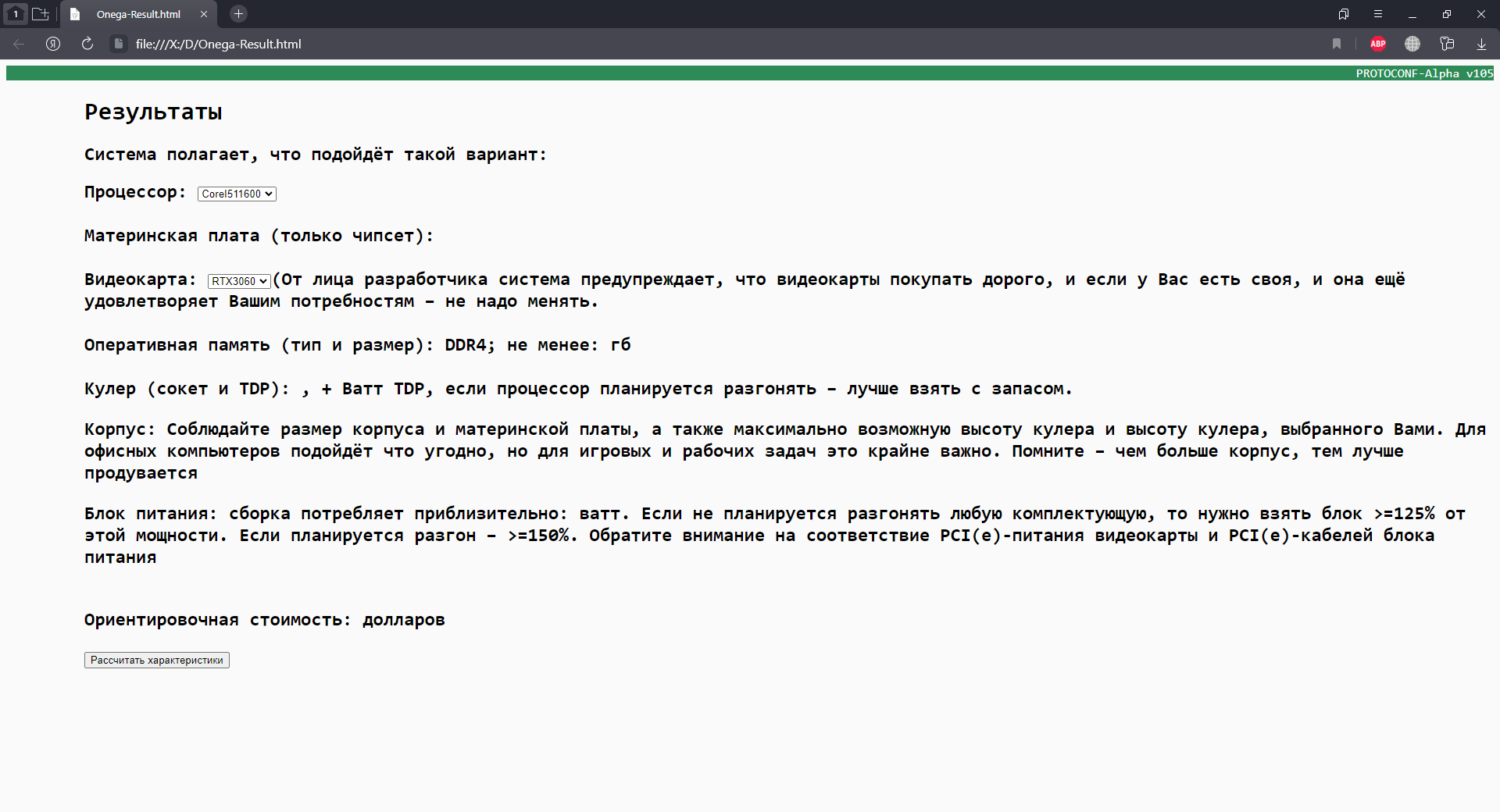


Рисунок 2.7 – Пустая форма предоставления результатов

В программе также предполагается необходимость сохранения полученной конфигурации, которую можно сделать двояко: либо через личный кабинет, и сохранение непосредственно там, что, в данной работе, неоправданно, так как никаких личных сведений в процессе конфигурации не раскрывается, а сложность создания личного кабинета радикально усложнит сайт и работу с ним, либо через генерацию некоторого кода, который будет собирать компьютер по коду, который является шифром для базы данных. Разумеется, для этого необходимо и главное меню, которое представлено на рисунке 2.8.

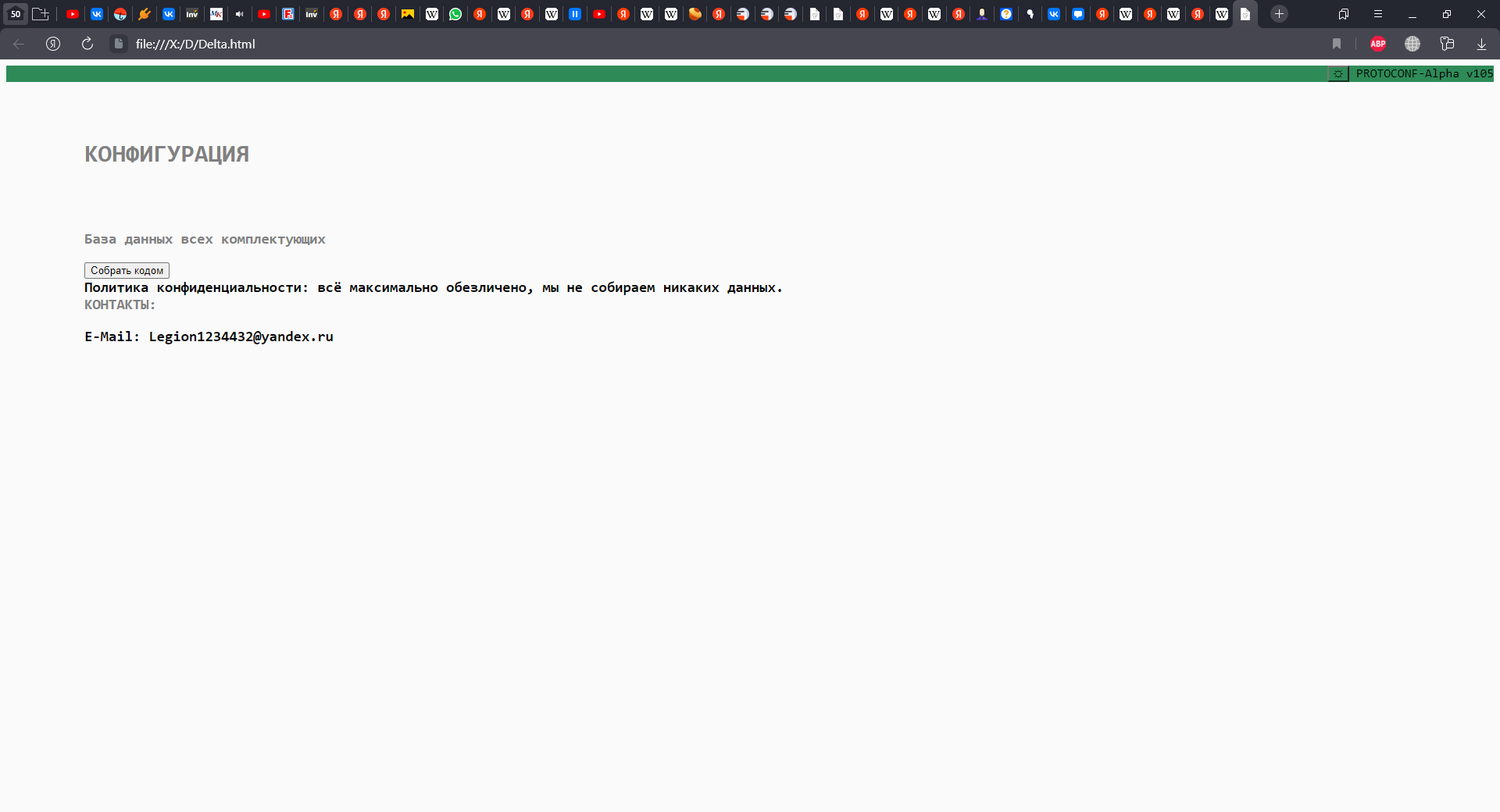


Рисунок 2.8 – Главное меню

На рисунках выше представлен также и прототип дизайна сайта. Сайт, как и было указано выше, минималистичен, и как можно менее нагружен. На рисунке 2.9 показано окошко для ввода кода.

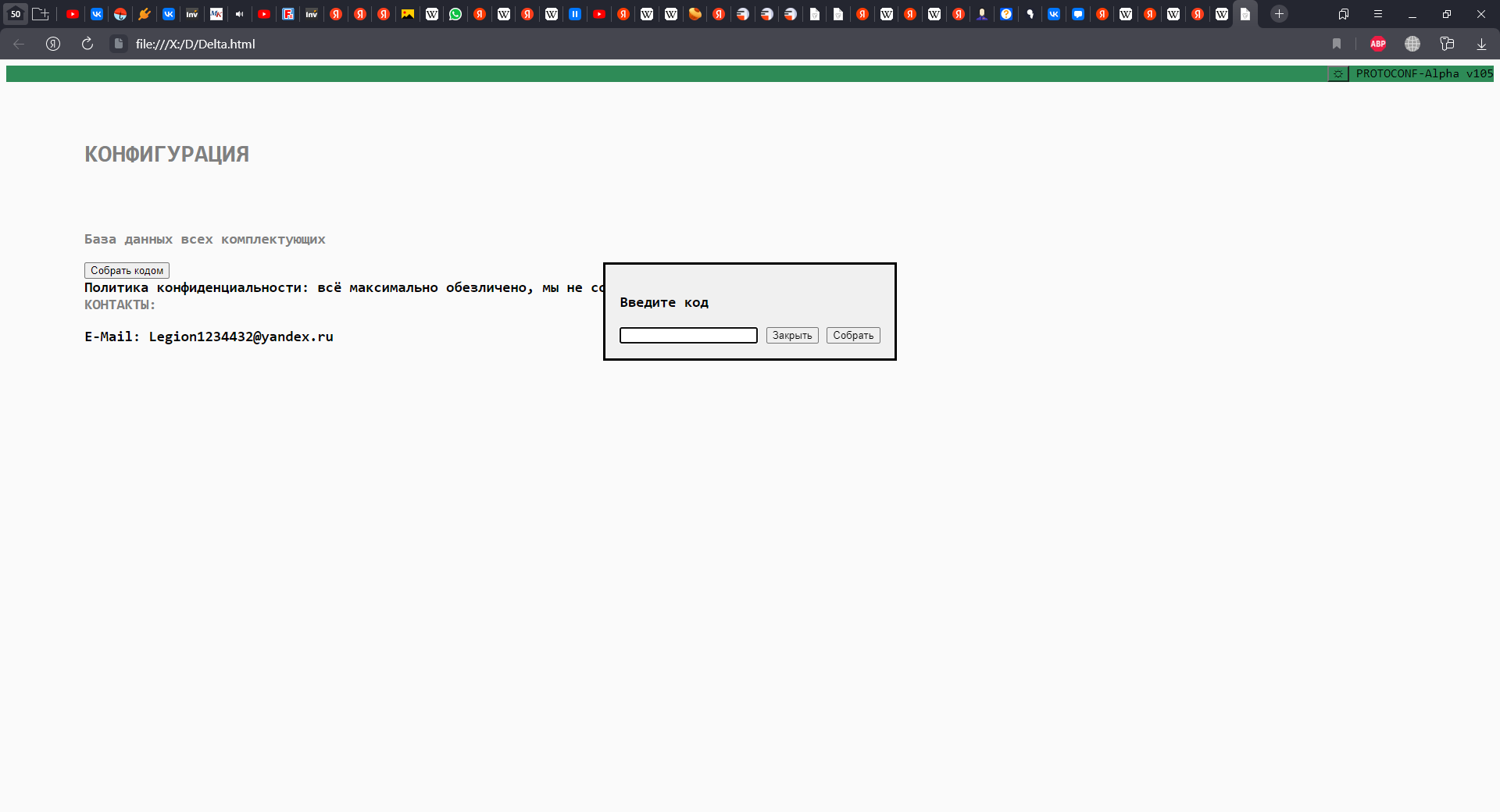


Рисунок 2.9 – Диалоговое окно для ввода кода конфигурации

Реализация кода конфигурации достигается через ID каждой комплектующей, и проведения всё тех же расчётов, которые помогли собрать ПК в прошлый раз. Так как влияние случая полностью исключено – компьютер получится аналогичным.

Также остались экраны, которые плохо передают насыщенный чёрный цвет, и, хотя большинство экранов с этим справляются, необходимо охватить как можно более широкий спектр пользователей. Различные исследования по-разному описывают полезность тёмного дизайна [7, 8, 9], однако по результатам опросов на различных веб-ресурсах, с довольно высокой достоверностью можно говорить об определённом превосходстве тёмного дизайна над светлым в количестве использований (52.2% используют постоянно, 24,8% используют время от времени тёмный дизайн по результатам Habr.com [10], 79% используют тёмный дизайн по результатам 4pda.to [11]), следовательно, дизайном по умолчанию должна бы стать тема, оформленная в тёмных тонах, но пока принято применять светлые тона для темы по умолчанию, поэтому и в этой программе будет применён похожее решение.

На рисунке 2.10 показан вариант с тёмными тонами, как альтернативного дизайна.

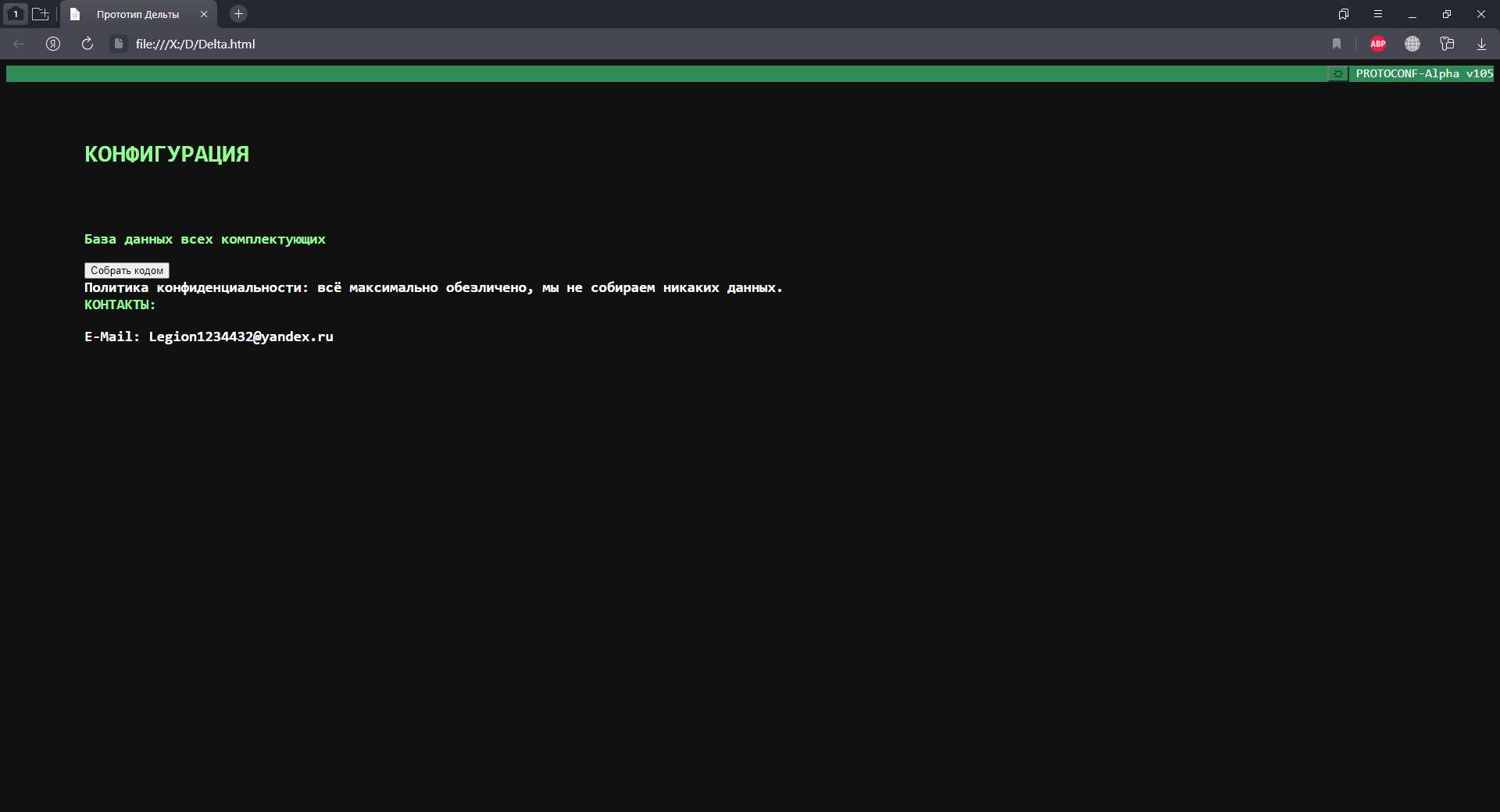


Рисунок 2.10 – Альтернативный дизайн

Такой дизайн является менее традиционным, хотя большинство использующих тёмный дизайн перешли на него из-за резкого света белых страниц.

**2.5 Моделирование бизнес-процессов информационной системы**

Бизнес-процессы ИС описывают основной функционал как самой ИС, так и окружения [12], как минимум – процессы администратора, который неизбежно должен (по крайней мере – на этом этапе развития) вручную пополнять базу данных по мере выхода новых комплектующих.

Для большего удобства понимания важнейших процессов в ИС логично применить нотацию языка UML [12]. Unified Modeling Language – язык для описания в объектно-ориентированном моделировании, которым можно описывать бизнес-процессы.

Для наглядности и дополнительного обоснования актуальности создания ИС можно проиллюстрировать это бизнес-процессами условной компании «А», у которой возникла потребность докупить/обновить компьютерный парк, что отражено на рисунке 2.11.

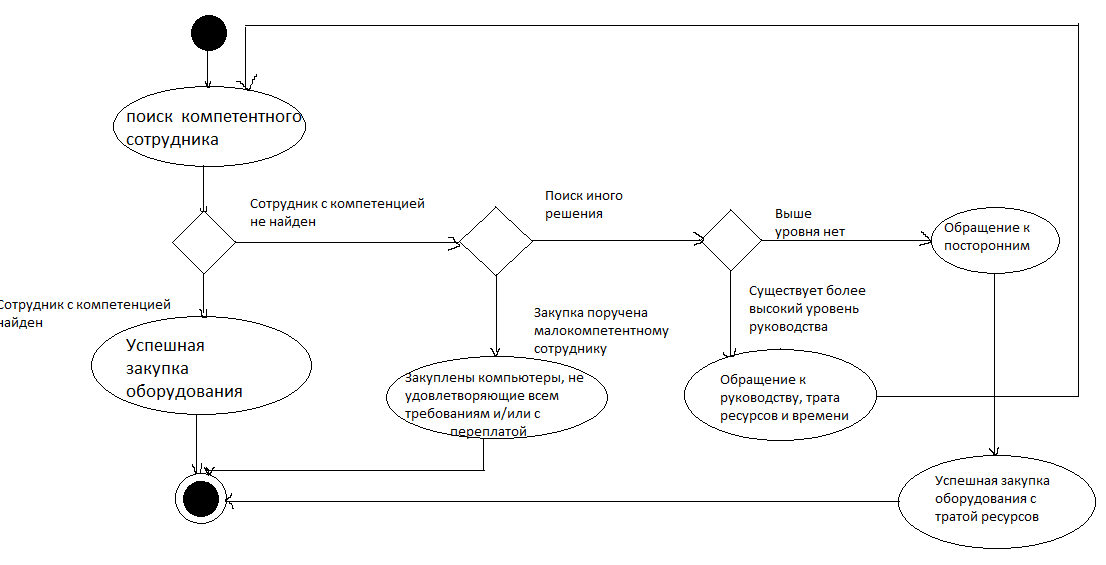


Рисунок 2.11 – Традиционный вариант закупки

Для пользования разработанной системой не нужна компетенция, поэтому любой сотрудник или руководитель не имеют ограничений компетенции, и на рисунке 2.12 представлена такая же активность с наличием подобной системы, то есть то, как «должно быть».

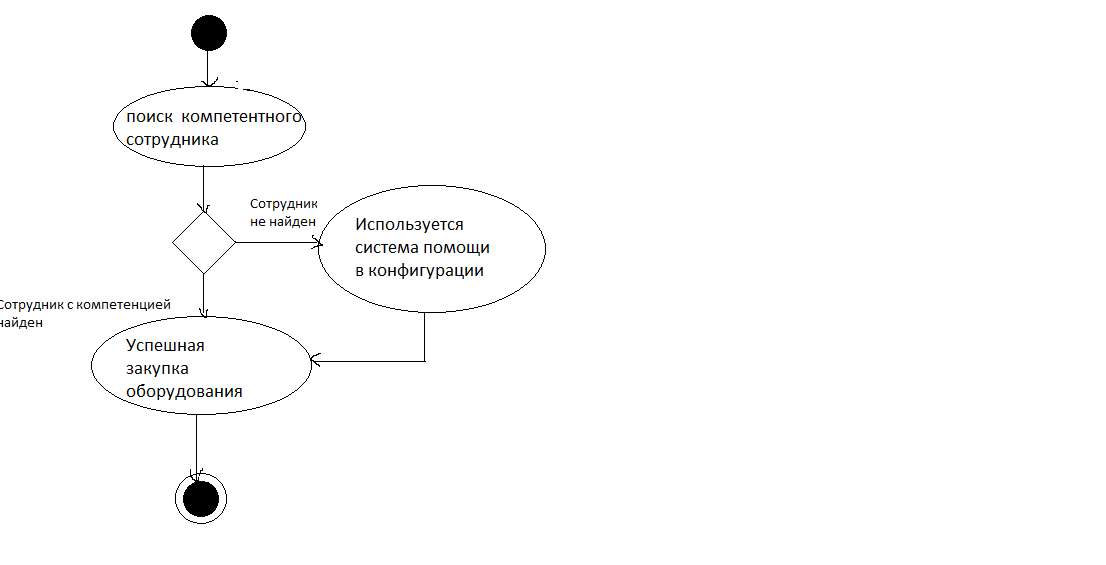


Рисунок 2.12 – Вариант закупки компьютеров с помощью системы помощи в конфигурации

Как можно заметить – схема очень сильно сократилась, достигнута не только экономия ресурсов временных и материальных, но и управленческих, с этой системой возникшую проблему, по крайней мере, в вопросах компетенции, точно можно будет решить на любом уровне, если на нём есть хотя бы один пользователь ПК. Однако для того, чтобы это работало, необходим труд администратора, и его use case представлен на рисунке 2.13.

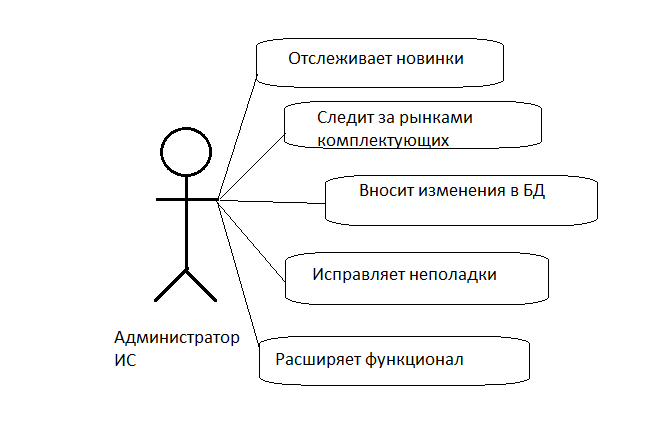


Рисунок 2.13 – Диаграмма use case администратора ИС

Необходимо разъяснить два момента: отслеживание администрацией рынка комплектующих нужно для того, чтобы вовремя убрать из базы данных те комплектующие, которые сняты с продажи, и более, вероятно, никогда не появятся. Выдача такого «отсутствующего» результата не приведёт к тем изменениям, что есть между рисунками 2.10 и 2.11, а, наоборот, добавит ещё один пункт к рисунку 2.10, и выльется в дополнительные затраты времени сотрудника и материальные затраты компании.

Расширение функционала – необязательная черта, однако в реалиях этого прототипа очевидно, что можно добавить больше функционала. Например: реализовать простейшие, обучающие подключению комплектующих к компьютеру, пояснения.

**2.6** **Общий концепт программного обеспечения**

Исходя из построенных концепций программного обеспечения уже можно создать полноценный рабочий прототип. Для более наглядного вида разработанной концепции принято применять диаграмму классов [13] (class-diagram), которая, для этого ПО, представлена на рисунке 2.14.

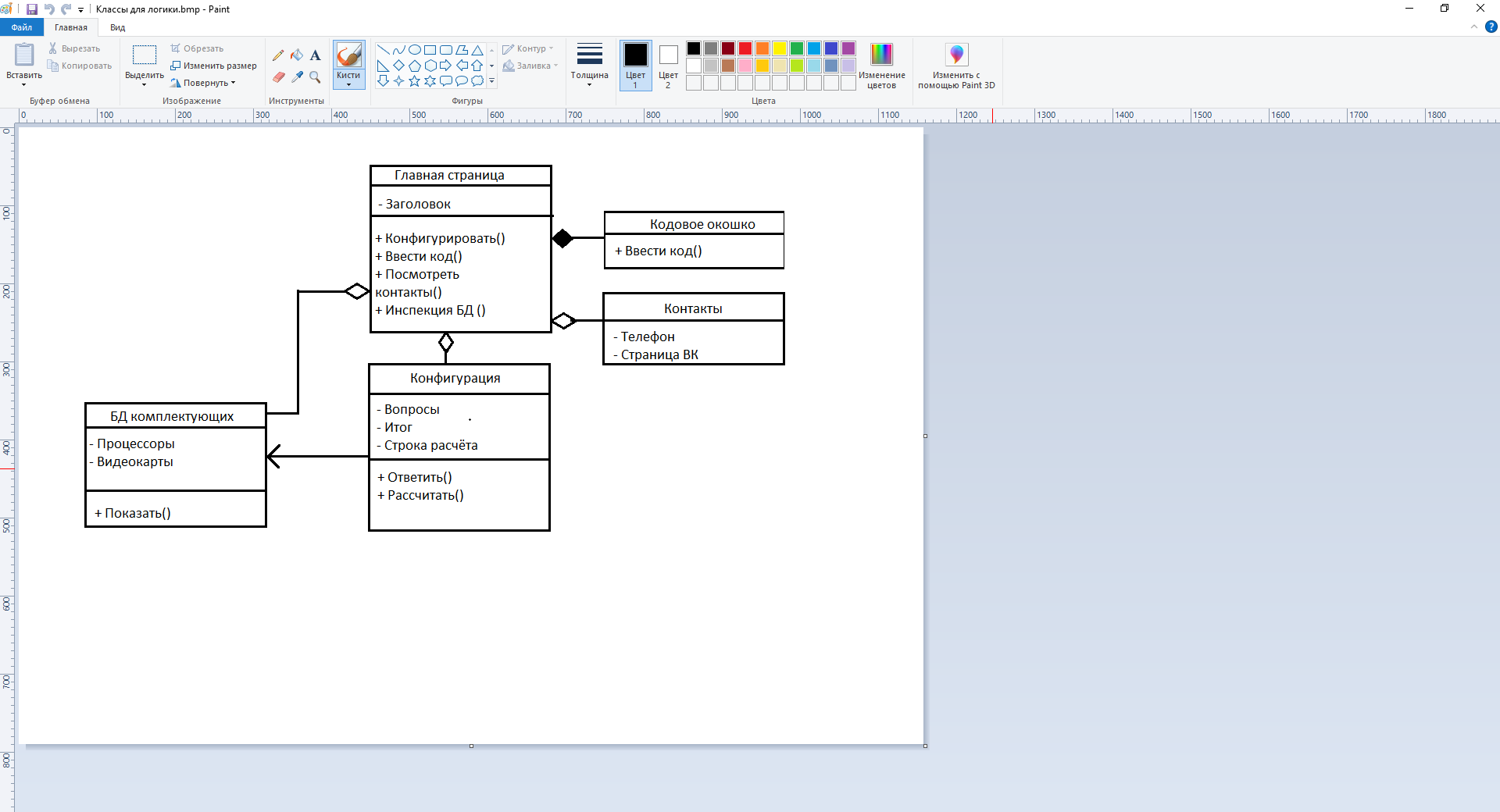


Рисунок 2.14 – Диаграмма классов конфигуратора ПК

Для рассмотрения логики необходимо привести перечень методов, и примерные результаты их работы, что представлено в таблице 2.7.

Таблица 2.7 – Перечень методов

|  |  |  |
| --- | --- | --- |
| Окно (класс) | Метод | Описание метода |
| Главная страница | Конфигурировать | Метод конфигурации запускается после опроса, на основании переменных для выбора из БД, составляет запросы к БД, грузит из БД значения. |

Продолжение таблицы 2.7

|  |  |  |
| --- | --- | --- |
| Окно (класс) | Метод | Описание метода |
| – | Ввести код | Метод ввода кода запускается по нажатию одноимённой кнопки, и показывает окно, в котором можно ввести код, по которому метод «Конфигурировать» соберёт конфигурацию |
| – | Посмотреть контакты | Открывает полное окно контактов с разработчиками |
| – | Инспектировать БД | Открывает окно полной базы данных |
| Кодовое окошко | Ввести код | Передаёт код в метод «Конфигурировать», что позволяет сделать аналог сохранения и загрузки, уточнить конфигурацию, исключив лишние варианты |
| Конфигурация | Ответить | Считывает выбранный вариант ответа, а также не позволяет пропустить те вопросы, ответы на которые необходимы |
| – | Рассчитать | Рассчитывает второстепенные переменные, не указанные в базе данных |
| БД комплектующих | Показать | Загружает всю базу данных, и выводит в два раздела: для процессоров и для видеокарт |

Этот перечень достаточно полно отражает все методы, и объясняет их работу.

**2.7 Выводы по разделу**

В рамках этого раздела были сформулированы критерии выбора языка и сред разработки, обоснован выбор языка разработки, был проведён анализ и выбор сред разработки, разработаны основные элементы информационной системы: размечен сайт, разработана база данных, обоснован концепт дизайна сайта, продемонстрирована его реализация, разработана логика программы [14].

Описание бизнес-процессов позволяет ещё раз, теперь уже графически, показать актуальность и оптимальность выбранной системы, выделить, возможно, избыточные или недостающие операции, определить зоны ответственности в рамках процесса и/или системы.

Модель «как есть» позволила просчитать варианты того, как развиваются события сейчас, а модель «как будет» показала экономию всех возможных видов ресурсов, как временного и материального, как самых очевидных, так и чуть менее очевидного административного ресурса в ряде случаев.

Таким образом, можно ещё раз сделать вывод, что разработка подобной системы необходима, так как существует ниша и потребность. А ввиду простейшего интерфейса и минимальной терминологии она подходит сотруднику с почти любым уровнем квалификации, единственное требование: уметь пользоваться ПК. На следующих версиях имеет смысл перенести программу на мобильные телефоны, что достигается простым изменением разметки.

После завершения этого раздела нужно создать руководство пользователя, хотя программа и является простой, однако не существует абсолютно интуитивно понятной для всех программы, поэтому создание руководства пользователя с демонстрацией нормальной работы необходимо.

# 

# **3 Руководство пользователя**

ПО разработано для конфигурации пользовательских ПК для любого пользовательского назначения, будь то офисное, рабочее или игровое. ПО не учитывает серверные машины, однако внутри пользовательских компьютеров позволяет проверять физическую, жёсткую совместимость различных комплектующих, а также логическую, мягкую совместимость. ПО создано для малокомпетентных в сфере конфигурации ПК пользователей, как следствие – пользователю не нужно никакой компетенции, кроме элементарных.

* 1. **Назначение и условия применения**

Как и указано выше – назначение этого ПО состоит в сборке ПК, в нём автоматизирован подбор основных комплектующих и дополнительные расчёты, которые позволяют выбрать второстепенные комплектующие. То есть, если разбить по операциям:

– получить конфигурацию;

– вернуть ранее полученную конфигурацию через код;

– посмотреть все компоненты в БД;

– узнать контакты Администрации.

* 1. **Подготовка к** **работе**

Программа развёрнута на сайте, нужно просто зайти на сайт, программа уже будет работать. Проверить работоспособность программы пользователю можно только пройдя опрос. Если опрос работает, и выдаёт корректный результат – программа работает.

* 1. **Описание операций**

Описывать операции в отрыве от работы реальной программы не имеет смысла [15]. Поэтому имеет смысл начать с главной страницы сайта, которая представлена на рисунке 3.1.

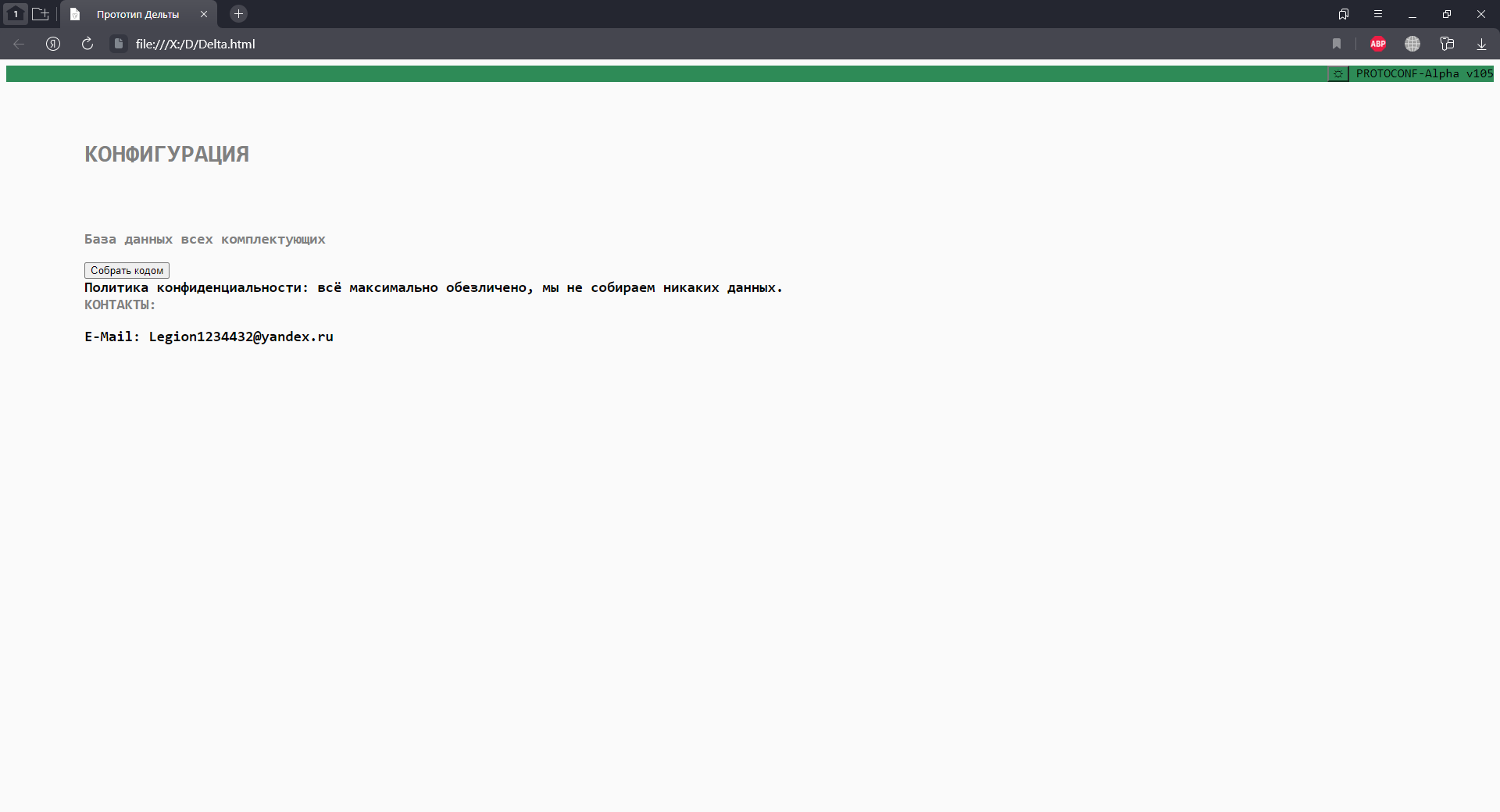


Рисунок 3.1 – Главная страница сайта

На странице представлена возможность пройти опрос, и получить конфигурацию, зайти в БД всех комплектующих, собрать с помощью кода, а также отображаются контакты.

В первую очередь будут рассмотрены основные функции. А именно – конфигурация, третий вопрос которой представлен на рисунке 3.2.

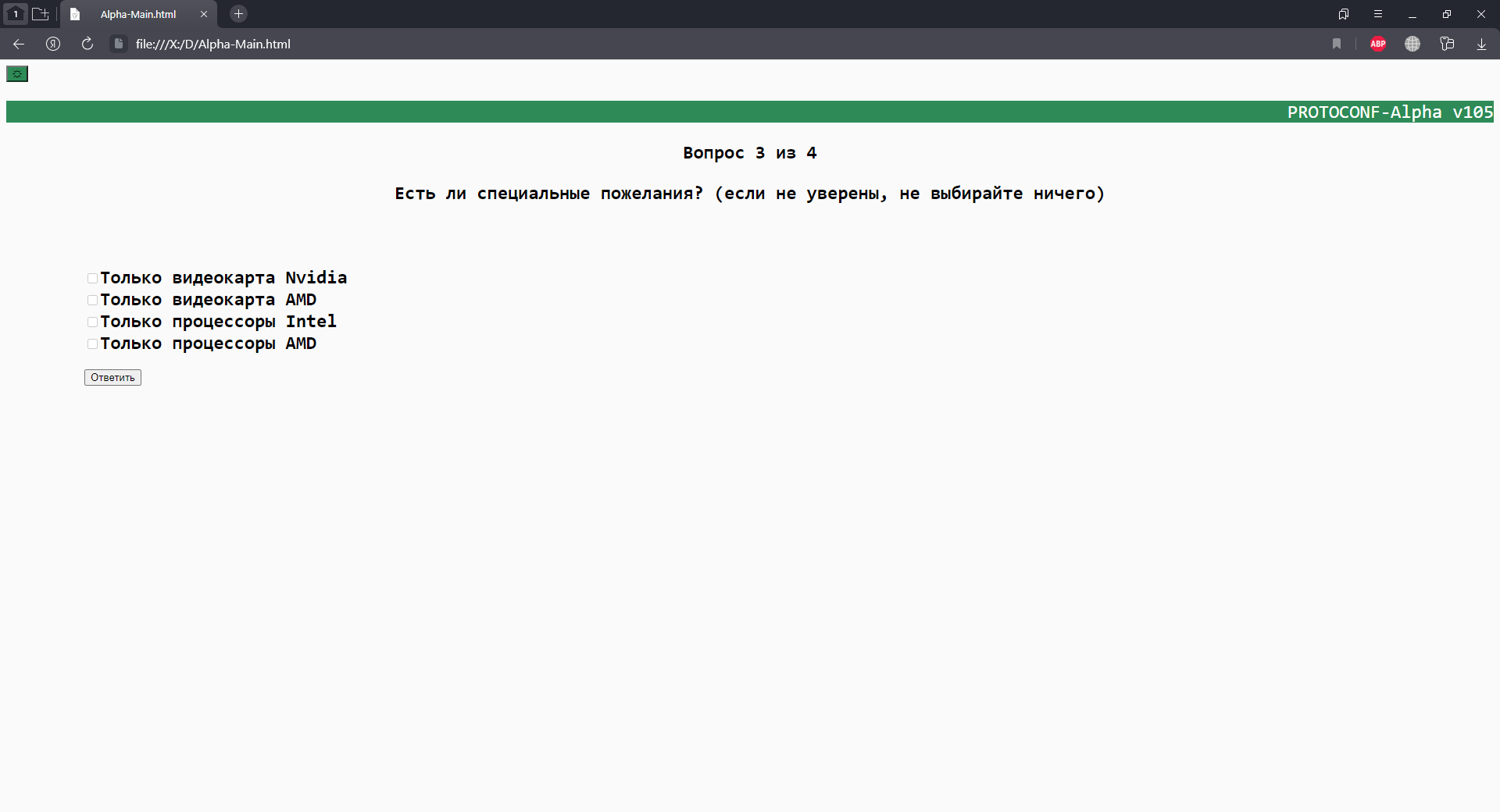


Рисунок 3.2 – Вопрос из опросного листа

В случае неполучения ответа на любой вопрос, кроме третьего, который может быть пустым, система выведет сообщение об обязательности вопроса, как на рисунке 3.3.

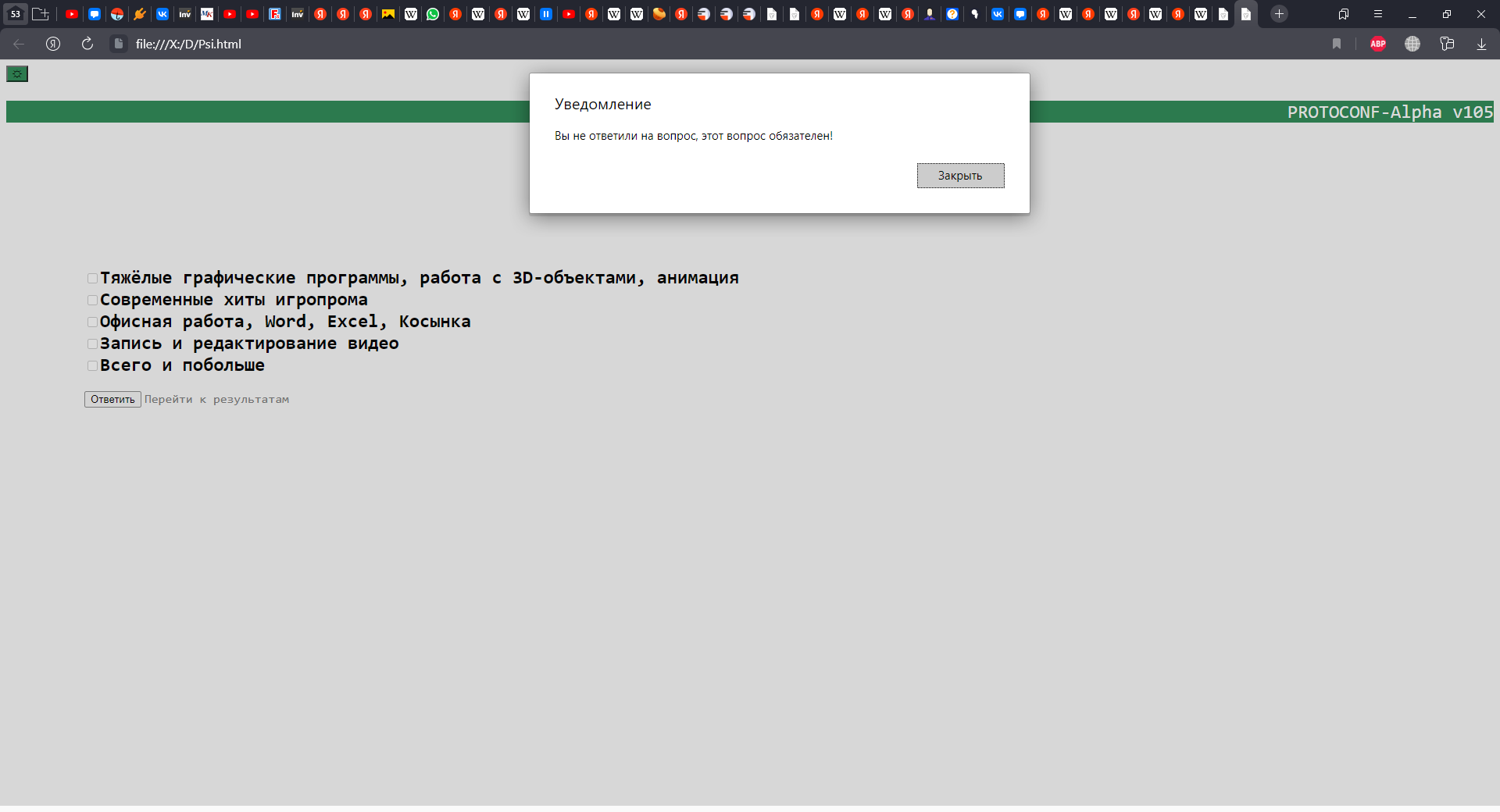


Рисунок 3.3 – Пример уведомления об обязательности вопроса

База данных всех комплектующих представлена на рисунке 3.4. Её задача – помогать Администрации быстро оценивать, нет ли чего-то устаревшего, и не нужно ли что-либо добавить в базу данных.

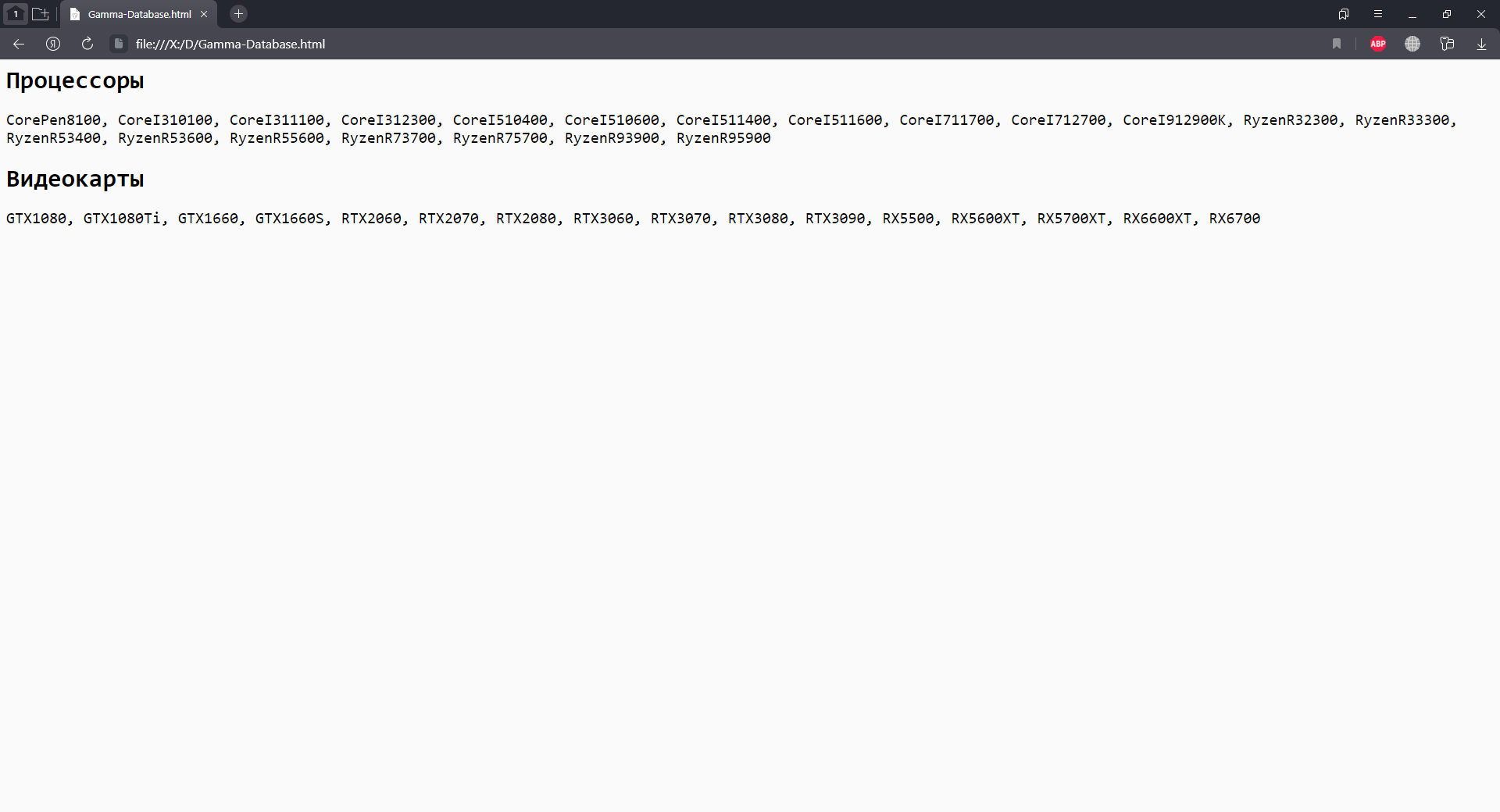


Рисунок 3.4 – База данных всех комплектующих

Сборка с помощью кода осуществляется путём открытия диалогового окна, какое показано на рисунке 3.5.

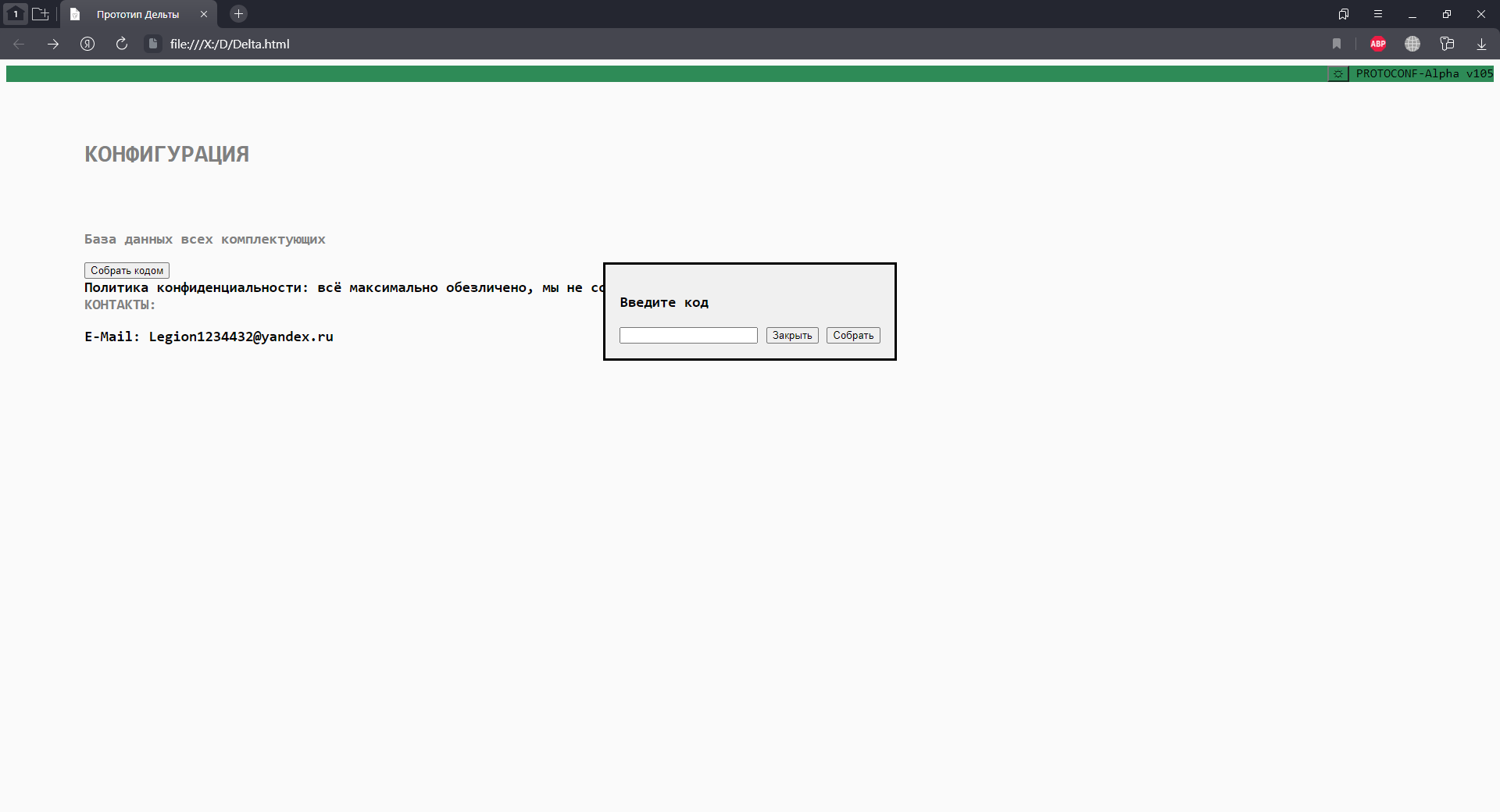


Рисунок 3.5 – Диалоговое окно кодовой загрузки

В случае ошибки кода система выведет предупреждение о том, что код неверен, как показано на рисунке 3.6.

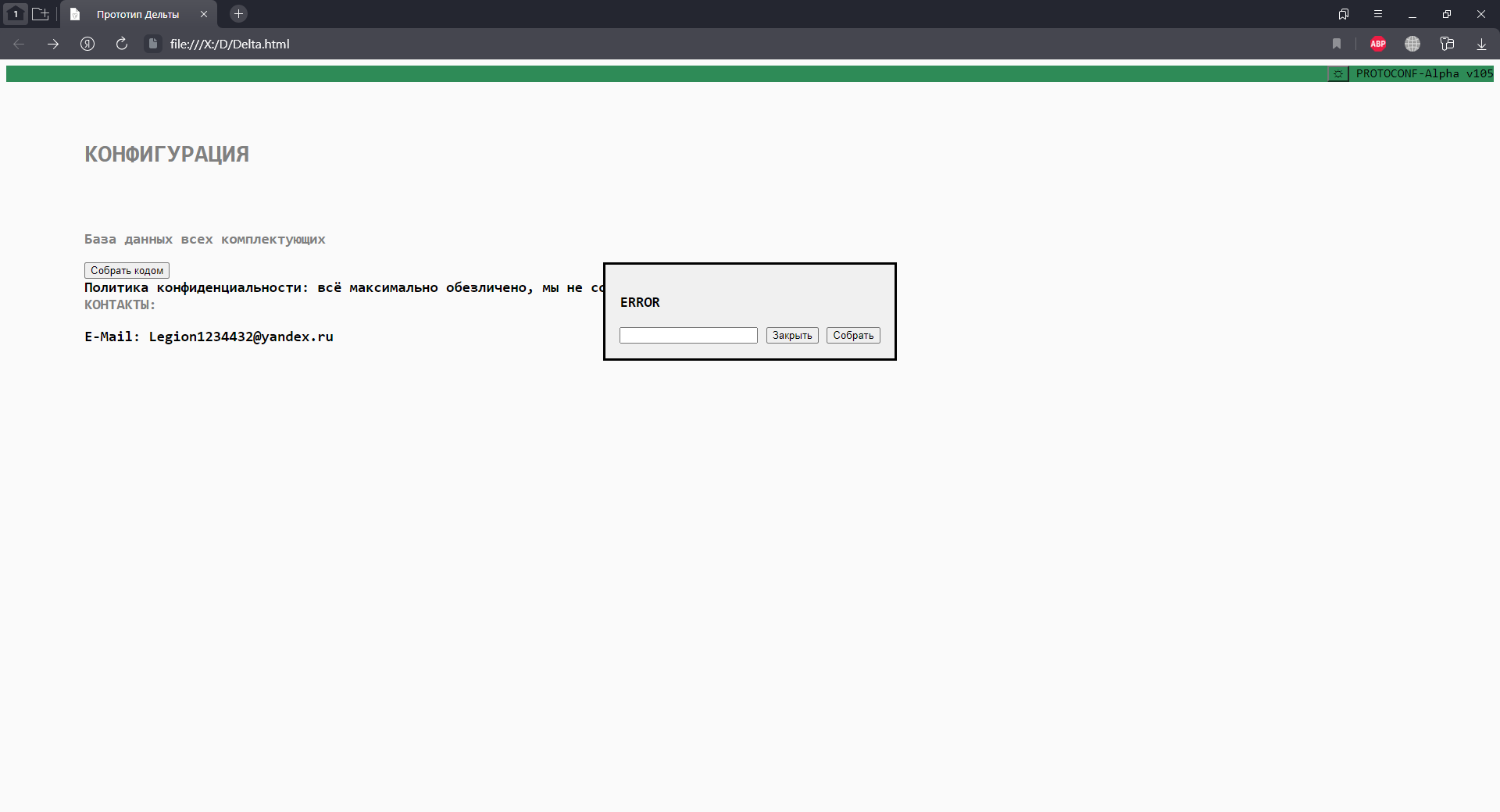


Рисунок 3.6 – Диалоговое окно загрузки при ошибочном введении кода

Итоговый экран представлен на рисунке 3.7. Расчётные данные берутся частично из базы данных (сущность «чипсет», например), частично рассчитываются из ответов. Также, по рекомендованным ценам и расчётным показателям вычисляется приблизительная стоимость конфигурации. Ввиду отсутствия постоянной выборки на любом из сайтов невозможна подгрузка цен, проблема решается ориентировочными значениями.

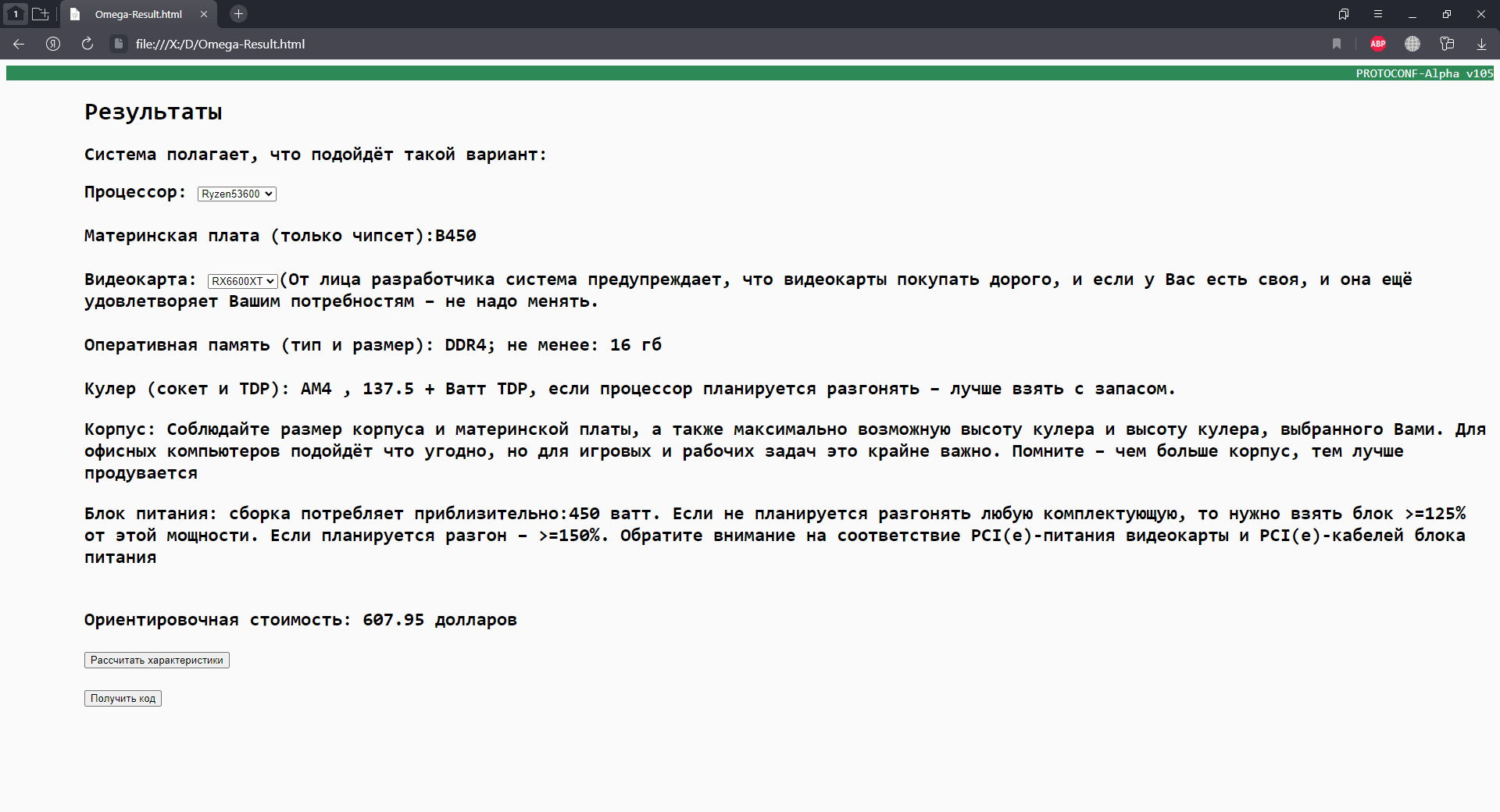


Рисунок 3.7 – Экран конфигурации

* 1. **Аварийные ситуации**

В случае единичной ошибки на стороне пользователя может помочь обновление страницы и повторение действий, либо же, если пользователь использует какой-нибудь экзотический браузер, может помочь смена браузера на более классический.

Если же причина аварийной ситуации на стороне сервера, то программа, даже если работает, может работать некорректно, не имеет смысла пользоваться программой до устранения неполадок на сервере и/или в архитектуре программы. Для более быстрого решения проблемы можно воспользоваться страницей контактов, и связаться с администрацией сайта для уточнения проблемы.

* 1. **Рекомендации по освоению**

Программа создана для пользователей с различной компетентностью, и единственная компетенция, которая должна быть у пользователя: базовый уровень пользования ПК. Никакие более существенные компетенции не требуются.

* 1. **Вывод по разделу**

В разделе было создано руководство пользователя, показана нормальная работа приложения, созданы варианты решения гипотетических проблем, показаны некоторые ошибки, которые могут иметь распространение, продемонстрирована функция сохранения с помощью кода, показано назначение программы и составлены рекомендации по освоению.

# **ЗАКЛЮЧЕНИЕ**

Данная выпускная квалификационная работа направлена на разработку прототипа web-конфигуратора ПК.

В ВКР были разработаны, обоснованы, описаны и наполнены следующие разделы:

– главная страница;

– окно конфигурации;

– опросный лист;

– база данных комплектующих;

– контакты.

Разработка информационной системы происходила в несколько этапов, так как это сложный и трудоёмкий процесс, соответственно на создание ИС было затрачено большое количество времени.

В работе были описаны методологии UML, на ней разработаны бизнес процессы, необходимые для работы информационной системы. Для разработки ИС были рассмотрены аналоги конфигураторов и ресурсы для создания концептуального проекта, а также среды разработки базы данных, проведено их сравнение.

Итогом работы является готовый прототип конфигуратора, уже способный конфигурировать ПК в пределах своей базы данных, которая исчисляется несколькими десятками комплектующих обоих видов, обладающий потенциалом расширения.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1 Федеральная служба государственной статистики. Исследование «информационные и коммуникационные технологии», 2019 г. (дата обращения: 06.06.2022).

2 Global Digital 2022. Статистика интернета и соцсетей на 2022. [Электронный ресурс] URL: https://www.sostav.ru/publication/we-are-social-i-hootsuite-52472.html (дата обращения: 15.06.2022).

3 Конфигуратор DNS [Электронный ресурс] URL: https://www.dns-shop.ru/configurator/ (дата обращения: 15.06.2022).

4 Конфигуратор Citilink. [Электронный ресурс] URL: https://www.citilink.ru/configurator/?utm\_source=yandex&utm\_medium=cpc&utm\_campaign=cities-srch-svs-configurator&utm\_content=%7Cc%3A45052954%7Cg%3A3912670146%7Cb%3A7840349350%7Ck%3A17887977830%7Cst%3Asearch%7Ca%3Ano%7Cs%3Anone%7Ct%3Apremium%7Cp%3A1%7Cr%3A%7Cdev%3Adesktop&utm\_term=konfigurator\_kompyuter&utm\_param1=all&yclid=4027266082733055636 (дата обращения: 15.06.2022).

5 Конфигуратор Арсенал+ [Электронный ресурс] URL: https://novosibirsk.arsplus.ru/build\_computer/ (дата обращения: 15.06.2022).

6 Определение конфигурации ПК. [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Конфигурация\_компьютера (дата обращения: 09.06.2022).

7 Ссылка на исследование связи усталости глаз и позитивной полярности дисплея (достигается при светлом оформлении). [Электронный ресурс] URL: https://www.semanticscholar.org/paper/Positive-display-polarity-is-advantageous-for-both-Piepenbrock-Mayr/1605c4f7d561f6cca0fcce6062f788562ccbae6f?p2df (дата обращения: 05.06.2022).

8 Ссылка на исследование связи усталости глаза и различных причин. [Электронный ресурс] URL: https://www.mayoclinic.org/diseases-conditions/eyestrain/symptoms-causes/syc-20372397 (дата обращения: 27.05.2022).

9 Ссылка на исследование позитивной связи тёмного оформления и энергоэффективности. [Электронный ресурс] URL: https://onezero.medium.com/dark-mode-isnt-the-answer-to-our-screen-problems-1b9db4ef06fb (дата обращения: 27.05.2022).

10 Ссылка на другой опрос про предпочтение различных тем оформления (Habr). [Электронный ресурс] URL: https://habr.com/ru/post/473962/ (дата обращения: 05.06.2021).

11 Ссылка на опрос про предпочтение различных тем оформления (4pda). [Электронный ресурс] URL: https://4pda.to/2021/08/30/389874/opros\_polzuetes\_li\_vy\_tyomnoj\_temoj\_na\_smartfone/ (дата обращения: 06.06.2021).

12 Определение языка UML. [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/UML (дата обращения: 08.06.2022).

13 Материал по диаграммам классов. [Электронный ресурс] URL: https://prog-cpp.ru/uml-classes/ (дата обращения: 01.06.2022).

14 Паттерны: MVC vs MVP vs MVVM // Хабр [Электронный ресурс] URL: <https://habr.com/ru/post/215605/> (дата обращения 06.06.2022).

15 Куликов, С. Тестирование программного обеспечения. Третье издание. Базовый курс. EPAM Systems, 2020. – 314 с. : ил.; ).