

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Кафедра информационных технологий

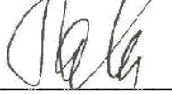
КУРСОВАЯ РАБОТА

**WEB-СЕРВИС ДЛЯ ОПРЕДЕЛЕНИЯ ПАРАМЕТРОВ СБОРКИ
КОМПЬЮТЕРА.**

Работу выполнил _____  _____ В.А. Бура
(подпись, дата) (инициалы, фамилия)

Факультет компьютерных технологий и прикладной математики 3 курс

Направление 09.03.03– «Прикладная информатика»

Научный руководитель, доц.
канд. физ.-мат. наук, доц. _____  _____ О.В.Гаркуша
(подпись, дата) (инициалы, фамилия)

Нормоконтролер, ст. преп. _____  _____ А.В.Харченко
(подпись, дата) (инициалы, фамилия)

Краснодар 2017

СОДЕРЖАНИЕ

Введение.....	3
1 Обзор технологий Web-разработки	4
2 Описание проекта и реализация	10
2.1 База данных.....	10
2.2 Реализация приложения	18
3 Описание интерфейса пользователя.....	28
Заключение	32
Список используемых источников.....	Ошибка! Закладка не определена.

ВВЕДЕНИЕ

На сегодняшний день, многие люди желают собрать персональный компьютер сами, тем самым сэкономив деньги за счет самостоятельной сборки и выбрав определенные характеристики, сделав акцент на задачи, для которых этот компьютер будет использоваться. Но прежде чем идти в магазин, людям необходимо изучить данную сферу, узнать какие характеристики комплектующих влияют на общую производительность. Для того чтобы облегчить процесс изучения и принятия решения было разработано WEB-приложение, ориентированное на поставленную задачу.

Данная курсовая работа посвящена изучению средств разработки Web-приложений, а также проектированию и реализации собственного приложения, которое позволяет на основе выбранных фирм производителей видеокарт и процессоров или по ответам пользователя, на предоставленные вопросы, подобрать оптимальные по стоимости и производительности, комплектующие для сборки персонального компьютера.

Первая глава курсовой работы содержит обзор используемых для создания web-приложения языков программирования, фреймворков, технологий, алгоритмов. Вторая глава посвящена описанию проекта и реализации приложения. Третья глава содержит описание интерфейса пользователя.

1 Обзор технологий web-разработки

Для реализации данной задачи, мной были выбраны следующие языки: HTML[1], CSS, PHP[2], JavaScript[3]. Каждый из них реализует определенные функции приложения. Для удобства необходимо было выбрать такую среду разработки, которая могла бы поддерживать синтаксис всех представленных языков. Например, на эту роль мог бы подойти Eclipse.

Eclipse чаще всего используется для разработки Java приложений. Этот софт поддерживает плагины, которые делают возможной поддержку дополнительных языков программирования. Есть плагины для C/C++, Ruby, PHP и других.. Большие компании, например Google, предоставляют свои средства разработки для этой платформы. Можно легко создавать приложения для Android и App Engine. Eclipse бесплатный и открытый ресурс. Что касается минусов, он показался не очень удобен в использовании с точки зрения интерфейса, так же при стандартной установке отсутствовали встроенные во многих других средах инструменты, ускоряющие написание кода. Следует отметить высокую мощность и, следовательно, высокую требовательность к компьютеру. Для данной курсовой работы не требуется такого большого количества функций, по крайней мере, на данном этапе.

Так же была рассмотрена такая среда как Netbeans. Еще одна среда для Java, но как и Eclipse, может быть расширена для дополнительных языков: PHP, Python, C/C++ и других. Приложение запускается на Linux, Windows и OSX. Netbeans может ускорить процесс разработки десктопных приложений с помощью Drag and Drop конструктора. Негативной стороной является производительность – Netbeans может тормозить на больших проектах. Этот редактор кода бесплатный и открытый ресурс.

После просмотренных выше сред разработки, было принято решение, обратить внимание на более простые программы. Одной из таких стал

NotePad++. Этот мощный и легкий. Хотя название нагадывает расширенный блокнот, это могучий инструмент. Он прост для новичков, но он также подходит для профессионалов. Notepad++ поддерживает каждый из популярных языков программирования. Поддерживает разбиение рабочей среды на два окна редактирования, FTP браузер, макросы и мощные возможности редактирования текста.

После него была установлена программа – Sublime Text 3 это красивый кроссплатформенный редактор. Он быстрый и богат функционалом, для практически каждого языка программирования. Поддерживает несколько выделений, сворачивание кода, макросы, проекты и другое. Также возможно полноэкранный редактирование, которое выглядит превосходно на больших мониторах. Запускается на Linux, Windows и OSX. Этот редактор предоставляется с неограниченным тестовым периодом, но вам нужно купить лицензию за 59\$, которая может быть использована на каждом компьютере. Не смотря на огромную функциональность, данная среда имеет огромную скорость, возможность устанавливать плагины и темы, которые позволяют сделать все еще более удобным, но при этом скорость работы почти никак не изменяется.

Так же была рассмотрено такое приложение как PhpStorm. Программа представляет собой интеллектуальный редактор таких языков как: PHP, HTML и JavaScript с возможностями анализа кода на лету, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Авто дополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6, 7.0 и 7.1 (современные и традиционные проекты), включая генераторы, сопрограммы, пространства имен, замыкания, типаж и синтаксис коротких массивов. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов. PhpStorm разработан на основе

платформы IntelliJ IDEA, написанной на Java. Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины.

После рассмотрения всех вышеперечисленных программ, я остановился на последних двух, а именно Sublime Text 3 (На нем осуществлялась верстка сайта, в связи с его большой скоростью работы и удобством) и PHPStorm (Оказалась самой требовательной средой, но по интерфейсу самым удобным, особенно что касалось работы с файлами разных расширений).

Так же для реализации некоторых элементов верстки был использован фреймворк для языка JavaScript – jQuery[4]. Что это за фреймворк и почему было принято решение использовать именно его?

jQuery – фреймворк, разработанный для языка JavaScript, который заметно ускоряет написание кода, позволяя обращаться к различным тэгам, классам, и т.д. с помощью знака '\$', в отличии от 'document.' в JavaScript. Так появляется возможность работать, не обращая внимания на особенности браузера.

К минусам можно приписать внушительный объем, так что целесообразнее использовать его в крупных проектах, для написания более читаемого кода, что необходимо при работе в команде, а также более простой реализации вещей, которые требовали больших затрат энергии и сил без использования данного фреймворка.

Все языки, фреймворки, среды разработки, которые использовались при создании web-приложения описаны выше, ниже приведены языки и фреймворки, которые также рассматривались в качестве инструментов для реализации поставленной задачи.

Python[5] - высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности

разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Для данного языка также существует множество фреймворков, которые упрощают работу, при создании web-приложений. Примером таких фреймворков могут выступать:

Flask[6] является микрофреймворком для создания вебсайтов на языке Python. Одним из проектных решений во Flask является то, что простые задачи должны быть простыми; они не должны занимать много кода, и это не должно ограничивать вас. Поэтому мы сделали несколько вариантов дизайна, некоторые люди могут посчитать это удивительным и не общепринятым.

Например, Flask использует локальные треды внутри объектов, так что вы не должны передавать объекты в пределах одного запроса от функции к функции, оставаясь в безопасном треде. Хотя это очень простой подход и позволяет сэкономить много времени, это также может вызвать некоторые проблемы для очень больших приложений, поскольку изменения в этих локальных тредах-объектов может произойти где угодно в том же треде. Для того, чтобы решить эти проблемы, мы не скрываем от вас локальные треды-объекты, вместо этого охватываем их и предоставляем вам много инструментов, чтобы сделать работу с ними настолько приятным насколько это возможно.

Во Flask много вещей предварительно сконфигурированы, на основе общей базовой конфигурации. Например, шаблоны и статические файлы сохранены в подкаталогах в пределах исходного дерева.

Основная причина почему Flask называется «микрофреймворком» — это идея сохранить ядро простым, но расширяемым. В нем нет абстрактного уровня базы данных, нет валидации форм или всего такого, что уже есть в других библиотеках к которым вы можете обращаться. Однако Flask поддерживает расширения, которые могут добавить необходимую функциональность и имплементирует их так, как будто они уже были встроены изначально. В настоящее время уже есть расширения: формы валидации, поддержка загрузки файлов, различные технологии аутентификации и многие другие.

Django - это высокоуровневая веб-инфраструктура Python, которая позволяет быстро создавать безопасные и поддерживаемые веб-сайты. Построенный опытными разработчиками, Django заботится о многих проблемах веб-разработки, поэтому можно сосредоточиться на написании своего приложения без необходимости изобретать колесо. Это бесплатный и открытый источник, имеет процветающее и активное сообщество, отличную документацию и множество опций для бесплатной и платной поддержки. Основными плюсами данного фреймворка можно выделить:

Гибкость. Django может быть (и был) используемым для создания практически любого типа веб-сайта - от систем управления контентом и вики, до социальных сетей и новостных сайтов. Он может работать с любой клиентской платформой и может доставлять контент практически в любом формате (включая HTML, RSS-каналы, JSON, XML и т. Д.).

Безопасность. Django помогает разработчикам избегать многих распространенных ошибок безопасности, предоставляя инфраструктуру, которая была разработана для «правильного решения», чтобы автоматически

защитить сайт. Например, Django обеспечивает безопасный способ управления учетными записями пользователей и паролями, избегая распространенных ошибок, таких как включение информации о сеансе в файлы cookie, где она уязвима (вместо этого куки-файлы содержат только ключ, а фактические данные хранятся в базе данных) и непосредственно хранят пароли, а не хеш пароля.

Но не смотря на все плюсы Python самого, как языка программирования, так и его фреймворков, Flask и Django, выбор был сделан в сторону PHP. Последний обладает большей скоростью, разница не сильно заметная, но она есть, а так же, этот язык, кажется интуитивно понятным и более простым.

2 Описание проекта и реализация

2.1 База данных

База Данных состоит из 11 таблиц, которые созданы с помощью СУБД MySQL[8]. Был сделан выбор в сторону этой СУБД, т.к. она более простая в использовании, имеется большое наличие документации, в том числе и на русском языке.

Опишем структуру таблиц более подробно.

Для каждого комплектующего создана отдельная таблица со своим уникальным ключом, который передается в таблицу с финальной сборкой.

Название таблицы в базе данных – `CPU`. Ее структура представлена в таблице 1.

Таблица 1-Таблица БД CPU

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы
Model	Varchar(55)	Модель
Frequency	Float	Частота процессора в Mhz
Kol-vo yader	Int(11)	Кол-во физических ядер
Img	Varchar(55)	Путь к директории с изображением процессора
Price	Float	Средняя цена процессора

Таблица `video_card` – предназначена для хранения данных о видеокарте.

Таблица 2 – Таблица БД video_card

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы видеокарты
Seria	Varchar(55)	Серия видеокарты
Model	Varchar(55)	Модель
Memory	Int(11)	Кол-во памяти видеокарты в Гб
Memory_type	Varchar(55)	Тип видеопамяти
Img	Varchar(55)	Путь к директории с изображением видеокарты
Price	Float	Средняя цена видеокарты

Таблица motherboard – предназначена для хранения данных о материнской плате.

Таблица 3 – Таблица БД motherboard

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы материнской платы
Model	Varchar(55)	Модель
Img	Varchar(55)	Путь к директории изображением материнской платы
Price	Float	Средняя цена материнской платы

Таблица motherboard_slots– предназначена для хранения всех имеющихся слотов различных видеокарт.

Таблица 4 – Таблица БД motherboard_slots

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Name_slot	Varchar(55)	Наименование слота

Таблица Соотнош_Мат_Разъем – предназначена для взаимосвязи таблицы 3 и таблицы 4, т.е., хранит соотношение материнской платы, уникальный ключ который передается и различных слотов, уникальный ключ которых так же передается и их количество.

Таблица 5- Таблица БД Соотнош_Мат_Разъем

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Id_mat	Varchar(55)	Передается уникальный ключ материнской платы
Id_slot	Varchar(55)	Передается уникальный ключ слота
Kol	Varchar(55)	Количество слотов с данным ключом

Таблица RAM – предназначена для хранения данных об оперативной памяти. Название таблицы в базе данных – `RAM`.

Таблица 6 - Таблица БД RAM

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ

Таблица 6 - Таблица БД RAM (Продолжение)

Firma	Varchar(55)	Наименование фирмы
Model	Varchar(55)	Модель
Memory	Int(11)	Кол-во памяти в Гб
Img	Varchar(55)	Путь к директории изображением оперативной памяти
Price	Float	Средняя цена оперативной памяти

Таблица power_supply – предназначена для хранения данных о блоке питания.

Таблица 7 – Таблица БД power_supply

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы
Model	Varchar(55)	Модель
Kol_vt	Float	Количество Вт в блоке питания

Таблица 7 – Таблица БД power_supply (Продолжение)

Img	Varchar(55)	Путь к директории изображения блока питания
Price	Float	Средняя цена блока питания

Таблица hard_drive – предназначена для хранения данных о жестком диске.

Таблица 8 – Таблица БД hard_drive

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы
Model	Varchar(55)	Модель
volume	Float	Объем памяти в Мб
Img	Varchar(55)	Путь к директории с изображением
Price	Float	Средняя цена процессора

Таблица body – предназначена для хранения данных о корпусе.

Таблица 9 – Таблица БД body

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Firma	Varchar(55)	Наименование фирмы
Model	Varchar(55)	Модель
Img	Varchar(55)	Путь к директории изображения корпуса
Price	Float	Средняя цена корпуса

Таблица BuildPC – предназначена для хранения данных о финальной сборке.

Таблица 10 – Таблица БД BuildPC

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Body_id	Int(11)	Передается уникальный ключ корпуса

Таблица 10 – Таблица БД BuildPC (Продолжение)

Video_card_id	Int(11)	Передается уникальный ключ видеокарты
CPU_id	Int(11)	Передается уникальный ключ процессора
RAM_id	Int(11)	Передается уникальный ключ оперативной памяти
Power_supply_id	Int(11)	Передается уникальный ключ блока питания
Motherboard_id	Int(11)	Передается уникальный ключ материнской платы
Hard_drive_id	Int(11)	Передается уникальный ключ жесткого диска
Img	Varchar(55)	Путь к директории с изображением корпуса(Используется и для общей сборки)
Description	Text	Описание возможностей сборки для пользователя
Price	Float	Общая средняя цена сборки

Таблица build_position – предназначена для хранения всех имеющихся сборок с их координатами.

Таблица 11- Таблица БД build_position

Наименование поля	Тип	Назначение
Id	Int(11)	Уникальный ключ
Build_id	Int(11)	Передается уникальный ключ сборки
x	Int(11)	Координата по оси x
y	Int(11)	Координата по оси Y

2.2 Реализация приложения

Для реализации были использованы язык: php ,js.

В директории сайта имеется файл “index.php” в котором описана верстка сайта, с точки зрения программирования является наиболее интересный подключенный к нему файл “js.js”, в котором реализуется эффект плавного раскрытия списка вопросов, так называемый «аккордеон». Код представлен ниже.

```

window.onload = function(){ // Как только страница загрузилась стартует
//функция
// Аккордеон Реализация
$('.quest-title').click(function () { // По клике на блок с вопросами
    if ( !$(this).hasClass('active')) {
        $('.accor-in').slideDown(300); //Блок открывается
    }
}

```

```

        $(this).addClass('active');// Присвоится класс active
// Так будет понятно в каком состоянии находится сейчас блок
        } else { // Тоже самое, только закрытие
            $('.accor-in').slideUp(300);
            $(this).removeClass('active');// Удаляем класс
        }
    });

};

```

А так же функция контроля максимального и минимального значения, что бы пользователь не мог ввести число большее чем значение на `<input type="range">`

```

// Объявление переменных
var range = $(".range-price");
var priceInput = $(".price-text-input");
var valRange = +range.val();
var valPriceInput = +priceInput.val();

//Изменяем значение ползунка меняется и значение окна ввода
range.change(function() {
    valRange = +range.val();
    priceInput.val(valRange);
});

```

Следующий блок кода отвечает за регулирование и редактирование окна текстового поля. В случае ввода пользователем значения стоимости

меньшее или большее минимального, или максимального значения общей стоимости сборки соответственно, устанавливается минимальное или максимальное значение.

```
//Изменяем значение окна ввода меняется и значение ползунка
priceInput.change(function() {
    valPriceInput = +priceInput.val();
    if (valPriceInput < minPrice) {
        valPriceInput = minPrice;
        priceInput.val(valPriceInput);
    } else if ( valPriceInput > maxPrice) {
        valPriceInput = maxPrice;
        priceInput.val(valPriceInput);
    };
    range.val(valPriceInput);
});
```

По нажатию на одну из клавиш “Отправить” (В зависимости от способа подбора, который выбрал пользователь), данные отправляются методом POST на страницу “result.php”, на которой происходит взаимосвязь с базой данных и обработка данных которые отправил пользователь. Основные строчки кода представлены ниже.

В данной ситуации есть 2 варианта развития событий:

- 1) Пользователь выбрал фирму, которую он предпочитает, и цену, которую считает приемлемой.
- 2) Пользователь ответил на вопросы, и подбор будет происходить исходя из его ответов.

Рассмотрим первый вариант:

```
//      Если была послана форма с Ценой и фирмами.
if ( isset($_POST['firmsAndModels']) ) {

//      Определяем какая серия видеокарты выбрана
$postFirms = $_POST['choose-firms-video'];
if ( $postFirms == 1 ) {
    $seria = 'any';
} else if ( $postFirms == 2 ) {
    $seria = 'NVIDIA';
} else if ( $postFirms == 3 ) {
    $seria = 'AMD';
};
```

После чего те же самые действия проделываем для определения фирмы производителя процессора.

```
$cpuFirm = $_POST['choose-firms-CPU'];
if ( $cpuFirm == 1 ) {
    $cpu = 'any';
} else if ( $cpuFirm == 2 ) {
    $cpu = 'Intel';
} else if ( $cpuFirm == 3 ) {
    $cpu = 'AMD';
};
```

После чего делается SQL-запрос к базе данных, который находит все видеокарты данной серии. Если же был выбран селектор на значении «Любая», то выбираются все имеющиеся видеокарты.

```
if ( $seria != 'any') {  
    $query = $db->query("SELECT `id` FROM `video_card` WHERE  
`seria` = '$seria';");  
} else {  
    $query = $db->query("SELECT `id` FROM `video_card`;");  
}
```

Следующий блок кода, заносит данные об первичном ключе выбранной/ых видеокарт в массив. Это сделано для того, чтобы лишний раз не обращаться к базе данных.

```
$i = 0;  
while($row = $query->fetch()) {  
    $videoList[$i] = $row['id'];  
    $i++;  
}
```

Далее происходит подбор по такой же системе, можно выделить только проверку на существование сборки в данном ценовом диапазоне.

Так в следующими строчками кода задается диапазон цены, выбранной пользователем. Где EPS – заранее заданная константа, которая определяет величину интервала между минимальной и максимальной ценой.

```
$price = $_POST['pred_price'];  
$minPrice = $price - EPS;
```

```
$maxPrice = $price + EPS;
```

И уже используя найденный интервал, используем его для нахождения сборки с заданной ценой, делаем это с помощью следующего запроса.

```
$query = $db->query("SELECT * FROM `BuildPC` WHERE  
`Summary` < $maxPrice AND `Summary` > $minPrice;");
```

Так как таблица к которой мы обращались содержит номера ключей комплектующих, необходимо записать интересующие ключи в массив, как было показано ранее.

После произведенных действий, необходимо соотнести все условия и найти такую сборку, которая бы всем им удовлетворяла. Для этого необходимо пройтись по всем 3 условиям. Для этого был использован цикл `while`.

```
// проверка на наличие такой сборки  
$i = 0;  
$build = array();  
while( $i <= count($priceList) - 1 ) {  
    $j = 0;  
    while ( $j <= count($cpuList) - 1 ) {  
  
        $l = 0;  
        while ( $l <= count($videoList) - 1 ) {
```

После чего необходимо записать условие, что данный процессор, установлен в сборку, которая находится в выбранном ценовом диапазоне. Аналогичные условия и для видеокарты.

```
if ( $cpuList[$j] == $priceList[$i]['CPU_id'] and $videoList[$i] ==  
$priceList[$i]['video_card_id'] ) {
```

После чего закрываются все циклы, увеличивается значение переменных, отвечающие за номер шага. И так пока не будут просмотрены все варианты.

Во втором случае код будет выглядеть так, т.е. если была послана форма с вопросами:

```
} else if ( isset($_POST['quest']) ){
```

Необходимо установить координаты. В данном приложении координатами являются ответы на вопросы, где номер вопросы – это Ось, а номер ответа – числовое значение.

```
$x1 = $_POST['forWhat'];
```

```
$y1 = $_POST['cost'];
```

Вариант просматривать координаты сразу через SQL-запрос, является не самым удобным. Проще собрать все координаты сразу одним запросом:

```
$query = $db->query("SELECT `build_id`,`x`,`y` FROM  
`build_position`");
```

Запишем их в массив:

```
$i = 0;
```



```

while($row = $query->fetch()) {
    $pos[$i]['build_id'] = $row['build_id'];
    $pos[$i]['x'] = $row['x'];
    $pos[$i]['y'] = $row['y'];
    $i++;
};

```

И работая с массивом, можно легко найти минимальное значение расстояния между точками. Для этого достаточно сначала присвоить минимальное значение первому элементу:

```

$min = sqrt( pow(($x1-$pos[0]['x']),2) + pow(($y1-$pos[0]['y']),2) );
$resPosX = $pos[0]['x'];
$resPosY = $pos[0]['y'];
$buildId = $pos[0]['build_id'];

```

После чего достаточно в цикле просмотреть все оставшиеся и, в случае необходимости, пере присвоить и само значение, и координаты, которые ему соответствуют. Пример реализации показан следующим циклом.

```

$i = 1;
while ( $i <= count($pos) - 1) {
    $length = sqrt( pow(($x1-$pos[$i]['x']),2) + pow(($y1-$pos[$i]['y']),2) );
    if ( $length < $min) {
        $min = $length;
        $resPosX = $pos[$i]['x'];
        $resPosY = $pos[$i]['y'];
        $buildId = $pos[$i]['build_id'];
    }
    $i++;
}

```

Далее достаточно просто выбрать из базы данных первичный ключ сборки, который был найден в предыдущем блоке кода. Делает выборку следующий запрос:

```
$query = $db->query("SELECT * FROM `BuildPC` WHERE `id` = '$buildId'");
```

Можно заметить, что для обоих случаев наименование переменных одинаково, это требуется для того, чтобы не вводить лишние условия и проверки, а также не акцентировать внимание на том, какой способ подбора был выбран. Далее происходит вывод информации с базы данных, все делается практически одинаковыми запросами. Код представлен ниже.

Например, программа ничего не смогла найти по заданным значениям, тогда будет выведено советующее сообщение. Код описан ниже.

```
<?php if (empty($build)) { ?>
    <h3 class="norm-title result center" style="color: #D2515C;" >К
    сожалению по данным требованиям мы ничего не смогли собрать</h3>
    <?php }
```

Если была найдена хоть одна сборка, то начинается целый блок кода, который сначала выдает сообщение:

```
else {
    ?>
    <h3 class="norm-title result center">Подобранная сборка ПК</h3>
```

Далее, характеристики каждого комплектующего выводятся на экран. Для этого необходимо исполнить следующий запрос, который выводит характеристики процессора.

```
<?php $cpuQ = $db->query("SELECT * FROM `CPU` WHERE `id` = " .
$build[$i]['CPU_id'] . "");
```

```
while($row = $cpuQ->fetch()) { //Расшифровка  
    ?>
```

Для остальных будут меняться только названия переменных и таблиц.

Примерно так выглядит реализация данного приложения.

3 Описание интерфейса пользователя

Главное окно имеет вид, представленный на рисунке 1.

На всех страницах у пользователя есть навигация с помощью двух ссылок вверху, а именно «Главная» и «О проекте». Вторую мы рассмотрим чуть позже.

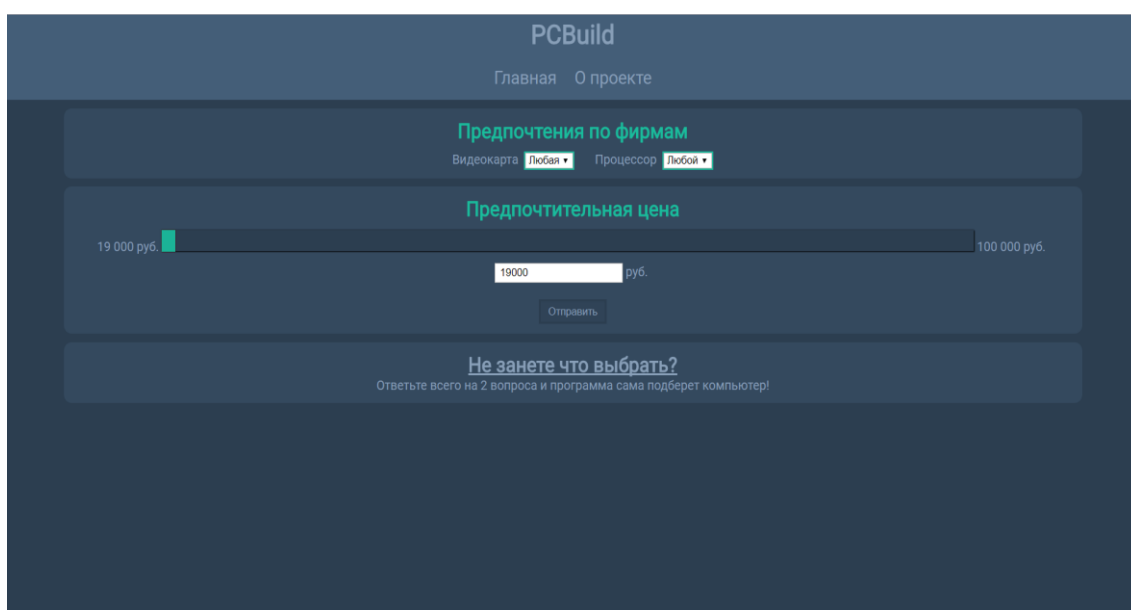


Рисунок 1 – Главное окно

На данной странице пользователю доступно несколько способов, с помощью которых программа будет подбирать самую оптимальную сборку комплектующих для персонального компьютера:

- 1) Самому выбрать интересующие его фирмы процессора и серии видеокарт, для этого присутствует два выпадающих списка, которые находятся в блоке с заголовком «Предпочтения по фирмам», при нажатии на них будет раскрыт весь список фирм или серий, в зависимости от

выбранного списка. А также имеется возможность выбрать оптимальную цену, которая изначально равна цене минимальной сборке, которая присутствует в нашей базе данных. Указать цену возможно с помощью ползунка, который находится сразу под заголовком «Предпочтительная цена», где требуется с помощью зажатой левой кнопки мыши, переместить ползунок, в, интересующий пользователя, ценовой диапазон. Вторым вариантом, просто вписать приемлемую для посетителя сайта цену в окно для ввода, которое располагается немного ниже ползунка. После того, как пользователь указал всё, что считал необходимым, ему требуется нажать на кнопку «Отправить», находящуюся ниже текстового окна.

2) Вторым способом, если пользователь не имеет никаких предпочтений, связанных с фирмой процессора или серии видеокарты. Для этого на сайте предусмотрена опросная система, которая подберет компьютер, в зависимости от того, чего именно хочет пользователь. Для того чтобы перейти к этому способу, посетителю сайта необходимо нажать на блок, в заголовке которого находится надпись: «Не знаете, что выбрать?». После чего раскроется блок с вопросами и вариантами ответа на них. Данный блок представлен на рисунке 2.



The image shows a dark-themed web form titled "Не знаете что выбрать?" (Don't know what to choose?). Below the title is a subtitle: "Ответьте всего на 2 вопроса и программа сама подберет компьютер!" (Answer just 2 questions and the program will choose the computer for you!). The form contains two sections:

- Для чего вам компьютер?** (For what do you need a computer?) with radio button options:
 - Играть в игры (Play games)
 - Работа с требовательными программами (Work with demanding programs)
 - Работа с офисными приложениями (Work with office applications)
 - Серфинг интернета (Internet surfing)
 - Работа с большим количеством окон (Work with a large number of windows)
 - Универсальная машина (Universal machine)
- Установите желательную стоимость** (Set the desired price) with radio button options for price ranges:
 - до 20.000р. (up to 20,000 rubles)
 - 20.000-30.000 (20,000-30,000)
 - 30.000-40.000 (30,000-40,000)
 - 40.000-50.000 (40,000-50,000)
 - 50.000-60.000 (50,000-60,000)
 - 60.000-80.000 (60,000-80,000)
 - от 80.000р. (from 80,000 rubles)

At the bottom of the form is a button labeled "Отправить" (Send).

Рисунок 2 – Блок с вопросами и вариантами ответов на них

В блоке имеется 2 вопроса, с разными вариантами ответов. После выбора ответов на вопросы необходимо нажать на кнопку «Отправить», которая располагается ниже.

После нажатия на эту кнопку, вне зависимости от выбора способа подбора, пользователь попадает на новую страницу, вид которой зависит от того, смогла ли программа найти подходящие комплектующие. К примеру, если этого не произошло, то страницу будет выглядеть, как представлено на рисунке 3 и сообщает пользователю о том, что по его запросу нет сведений.

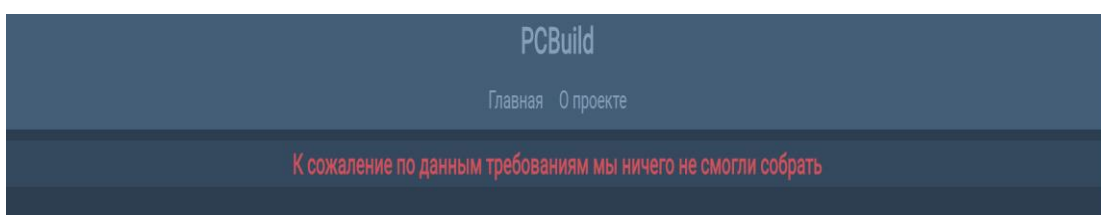


Рисунок 3 – Сборок не найден

Второй вариант того, как может выглядеть страница, представлен на рисунке 4. На рисунке показано, что в базе данных нашлась одна сборка, и комплектация этой сборки расписана на странице.

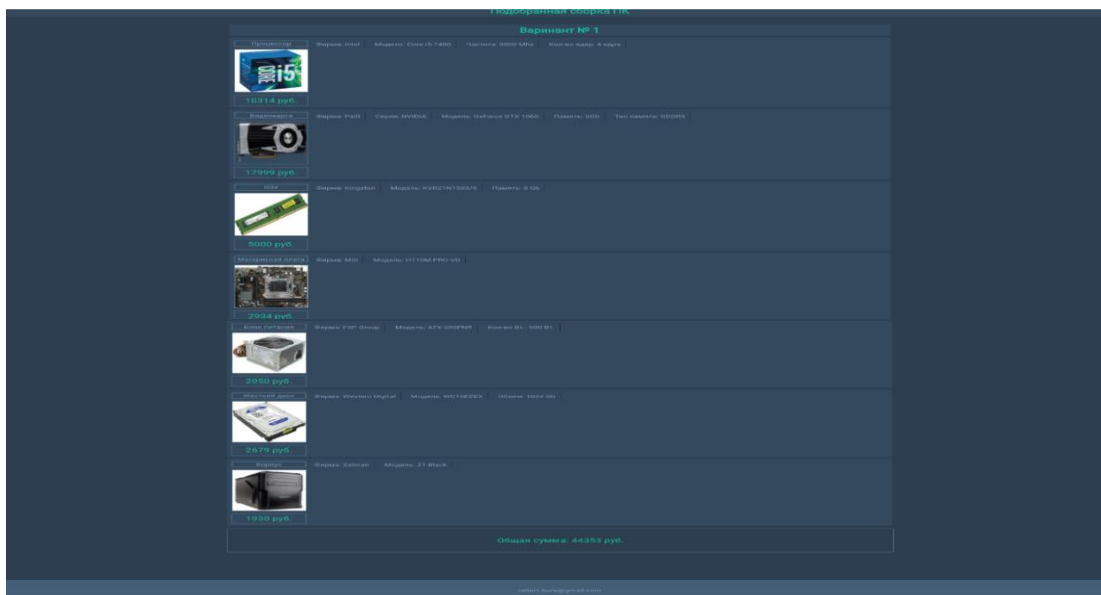


Рисунок 4 – Найдена одна сборка

На рисунке 5 представлен вариант результатов поиска =, когда найдено несколько сборок. Сборки перечисляются последовательно с полным описанием всех комплектующих, с их картинками, характеристиками и ценой. В конце сборки пишется общая сумма всех комплектующих.

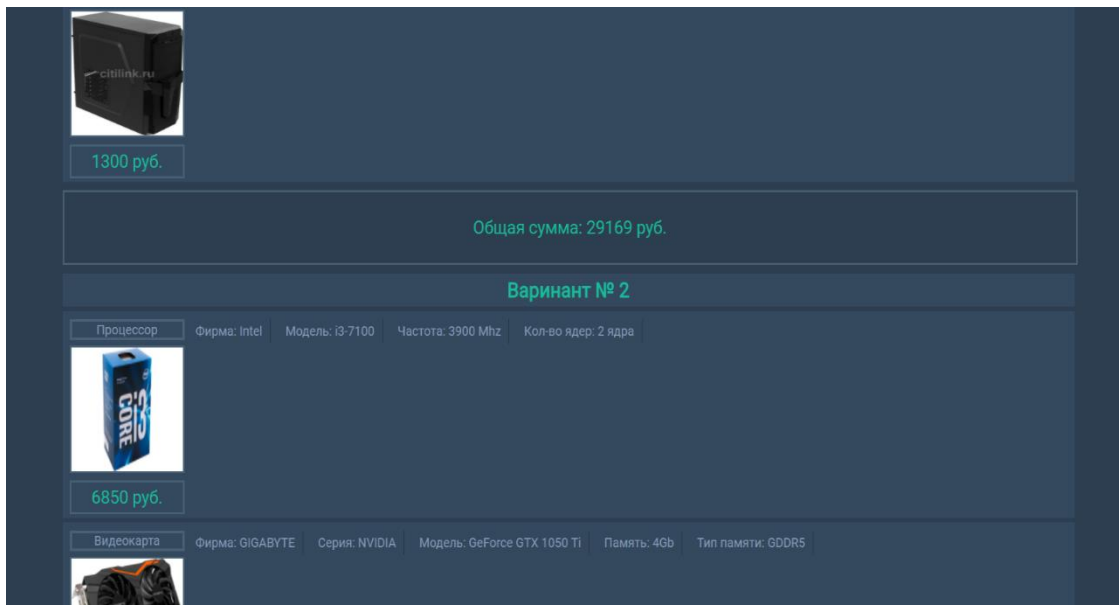


Рисунок 5 – Разделение между двумя сборками, удовлетворяющие заданным характеристикам

Также существует, просто информационная страница, на которую можно попасть по нажатию на ссылку «О проекте», о которой говорится в начале главы. Она представлена на рисунке 6 и содержит информацию об этом проекте.

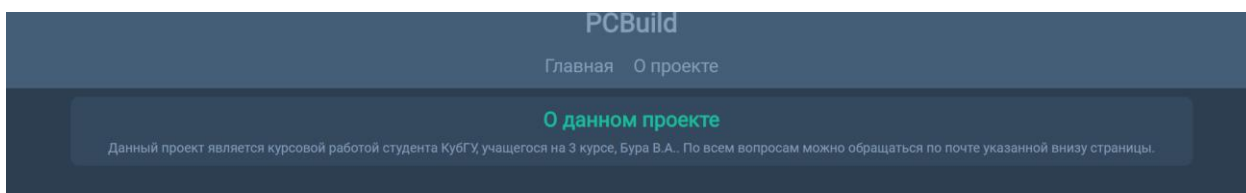


Рисунок 6 – О данном проекте

Отсюда можно вернуться на главную страницу кликнув по ссылке «Главная»

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы были изучены этапы создания web-приложений, получены навыки работы с языком разметки HTML и каскадной таблицей стилей CSS, с помощью которых были созданы каркасы страниц. Так же были выработаны навыки работы с СУБД MySQL, создание таблиц, первичных ключей. Был изучен язык программирования PHP, и язык SQL-запросов.

На основе полученных знаний было создано web-приложение, которое позволяет пользователю подобрать сборку персонального компьютера двумя способами, либо по его предпочтениям касательно фирм производителей процессора и видеокарт, либо сборка будет подобрана по результатам ответов на представленные вопросы. При втором варианте все сборки, представляются в виде объектов в n -мерном пространстве, где n – это количество вопросов. И самая близкая точка к той что «создал» пользователь – является сборка, наиболее подходящая под его ответы.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Htmlbook.ru | Для тех кто делает сайты. URL: <http://htmlbook.ru/html> (дата обращения 12.12.2017)
2. PHP.net. URL: <http://php.net/manual/ru/index.php> (дата обращения 12.12.2017)
3. Современный учебник JavaScript. URL: <https://learn.javascript.ru/> (дата обращения 12.12.2017)
4. jQuery API Documentation. URL: <https://api.jquery.com/> (Дата обращения 12.12.2017)
5. 3.6.4rc1 Documentation. URL: <https://docs.python.org/3/> (Дата обращения 12.12.2017)
6. Welcome | Flask (A Python Microframework). URL: <http://flask.pocoo.org/docs/0.12/> (Дата обращения 12.12.2017)
7. Веб-документация MDN. URL: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/> (Дата обращения 12.12.2017)
8. PHPClub - клуб разработчиков PHP. URL: <https://phpclub.ru/mysql/doc/> (Дата обращения 12.12.2017)