

Кафедра «Информатика»

ПРОЕКТ ПРОВЕРЕН

Рецензент

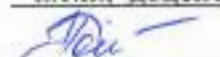
\_\_\_\_\_ / \_\_\_\_\_ /

« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

к.т.н., доцент

 /С.Г.Пономарева/

« 30 » МАЯ 2016 г.

## Базы данных по автосалонам города Нижневартовска

(наименование темы)

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОМУ КВАЛИФИКАЦИОННОМУ ПРОЕКТУ ЮУрГУ-230100. 2016.152.ПЗ ВКР

Консультанты

Экономическая часть


ст. преподаватель

 /А.В.Прокопьев/

« 08 » МАЯ 2016 г.

Безопасность жизнедеятельности


к.п.н., доцент

 /А.Б.Тряпичин/

« 12 » МАЯ 2016 г.

Руководитель проекта


к.п.н. доцент

 /Е.З.Никонова/

« 30 » МАЯ 2016 г.

Автор проекта

студент группы НвФл-423

 /И.В.Бегун/

« 30 » МАЯ 2016 г.

Нормоконтролер

ст. преподаватель

 /Л.Н.Буцик/

« 30 » МАЯ 2016




МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФИЛИАЛ ЮЖНО-УРАЛЬСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА  
В Г. НИЖНЕВАРТОВСКЕ  
КАФЕДРА «ИНФОРМАТИКА»

НАПРАВЛЕНИЕ 230100 « Информатика  
и вычислительная техника»

**УТВЕРЖДАЮ**

Заведующий кафедрой  
К.Т.Н., доцент

 /С.Г.Пономарева /  
« 05 » Февраля 2016 г.

**З А Д А Н И Е**

на выпускную квалификационную работу студента

Бегуна Ивана Викторовича

1. Тема работы

База данных по автосалонам города Нижневартовска

Утверждена приказом по университету от « 15 » Апреля 2016 г. № 661

2. Срок сдачи студентом законченной работы « 30 » МАЯ 2016 г.

3. Исходные данные к работе

1) Необходимо создать интернет-сайт;

2) Разработать рабочую часть проекта с использованием технологий Java Script, PHP, HTML, Dynamic HTML;

3) Разработать дизайн пользовательского интерфейса;

4) Проанализировать имеющиеся программные аналоги;

5) Интерфейс должен быть дружелюбным и понятным посетителю;

4. Содержание пояснительной записки (перечень подлежащих разработке вопросов)

- 1) Анализ предметной области и постановка задачи;
- 2) Описание среды программирования для Web-страниц;
- 3) Разработка интернет-сайта;
- 4) Расчет затрат на разработку;
- 5) Содержание и эксплуатация вычислительной техники;
- 6) Экономический эффект;
- 7) Требования безопасности;

7. Дата выдачи задания « 05 » ФЕВРАЛЯ 2016г.

Задание выдал руководитель Никонова Е.З.

Задание принял к исполнению студент Бегун И.В.

### КАЛЕНДАРНЫЙ ПЛАН

| Наименование этапов выпускной квалификационной работы | Срок выполнения этапа      | Отметки о выполнении этапа |
|---|----------------------------|----------------------------|
| 1. Введение   | 08.04.2016                 | выполнено                  |
| 2. Анализ предметной области и постановка задачи      | 09.04.2016 –<br>11.04.2016 | выполнено                  |
| 3. Описание среды программирования для Web-страниц    | 11.04.2016 –<br>12.04.2016 | выполнено                  |
| 4. Разработка интернет-сайта                          | 13.04.2016 –<br>04.05.2016 | выполнено                  |
| 5. Экономический раздел                               | 05.05.2016 –<br>08.05.2016 | выполнено                  |
| 6. Безопасность жизнедеятельности                     | 09.05.2016 –<br>12.05.2016 | выполнено                  |
| 7. Заключение   | 13.05.2016 –<br>14.05.2016 | выполнено                  |
| 8. Библиографический список                           | 15.05.2016                 | выполнено                  |
| 9. Приложения   | 16.05.2016 –<br>20.05.2016 | выполнено                  |
|   |                            |                            |
|   |                            |                            |
|   |                            |                            |
|   |                            |                            |
|   |                            |                            |
|   |                            |                            |
|   |                            |                            |

Заведующий кафедрой  / С.Г.Пономарева /

Руководитель проекта  / Е.З.Никонова /

Студент  / И.В.Бегун /

## АННОТАЦИЯ

Бегун И.В. База данных по автосалонам города  
Нижневартовска. – Нижневартовск: филиал  
ЮУрГУ, Информатика; 2016, 85 с., 5 ил., 3 табл.,  
библиогр. список – 33 наим., 3 прил.

Предметной областью данной выпускной квалификационной работы является Интернет-сайт на котором будут представлены все имеющиеся автомобили автосалонов г. Нижневартовск для информирования жителей г. Ноябрьск.

В настоящее время информация о предложениях автосалонов г. Нижневартовск доступна в офисе ООО «КАИС + Ноябрьск». Для того, чтобы люди могли получить более полную и актуальную информацию о наличии автомобилей, ООО «КАИС + Ноябрьск» необходимо разработать Интернет-сайт, на котором будут предоставлены все имеющиеся автомобили автосалонов г. Нижневартовск, доступ к которому будет возможен как с персонального компьютера, так и с мобильного устройства.

В рамках ВКР для ООО «КАИС + Ноябрьск» был разработан Интернет-сайт, на котором будут предоставлены все имеющиеся автомобили автосалонов г. Нижневартовск.

Таким образом, целью ВКР является разработка Интернет-сайта для получения и отображения для покупателей необходимой информации от заказчика, а также из дополнительных источников.

|                        |      |                        |                  |          |   |   |      |        |
|------------------------|------|------------------------|------------------|----------|---|---|------|--------|
| <b>230100.2016.152</b> |      |                        |                  |          |   |   |      |        |
| Изм                    | Лист | № докум.               | Подп.            | Дата     | <b>База данных по автосалонам города<br/>Нижневартовска</b> | Лит.  | Лист | Листов |
|                        |      | Разраб. Бегун И.В.     | <i>[Подпись]</i> | 30.05    |   | 33  | 6    | 85     |
|                        |      | Проверил Никонова Е.З. | <i>[Подпись]</i> | 30.05    |   | Филиал ФГБОУ ВПО "ЮУрГУ"<br>(НИУ) в г.Нижневартовске<br>Кафедра «Информатика» |      |        |
|                        |      | Рецензент              |                  |          |   |   |      |        |
|                        |      | И.контр. Буйлушкина    | <i>[Подпись]</i> | 30.05.16 |   |   |      |        |
|                        |      | Утв. Пономарева        | <i>[Подпись]</i> | 30.05.16 |   |   |      |        |

## ОГЛАВЛЕНИЕ

|   |    |
|---|----|
| ВВЕДЕНИЕ.....   | 8  |
| 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ ...   | 9  |
| 1.1 Описание предметной области .....   | 9  |
| 1.2 Цель работы .....   | 9  |
| 1.3 Постановка задачи .....   | 10 |
| 1.4 Задачи, решаемые в ходе работы.....   | 10 |
| 2 ОПИСАНИЕ СРЕДЫ ПРОГРАММИРОВАНИЯ ДЛЯ WEB-СТРАНИЦ .....                                     | 12 |
| 2.1 Формализация и стандартизацию данных и задач .....                                      | 12 |
| 2.2 Характеристика технологии HTML .....  | 12 |
| 2.3 Характеристика технологии php.....  | 14 |
| 3 РАЗРАБОТКА ИНТЕРНЕТ-САЙТА.....  | 16 |
| 3.1 Методика разработки проекта .....   | 16 |
| 3.2 Использование парсинга при создании нового Интернет-ресурса                             | 17 |
| 3.3 Используемая библиотека для парсинга сайтов .....                                       | 19 |
| 3.4 Листинг сайта .....   | 27 |
| 4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ .....  | 31 |
| 4.1 Расчет затрат на разработку .....   | 31 |
| 4.2 Содержание и эксплуатация вычислительной техники .....                                  | 33 |
| 4.3 Экономический эффект .....  | 35 |
| 5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ .....  | 36 |
| 5.1 Требования безопасности.....  | 36 |
| 5.2 Меры безопасности при работе с копировально-множительной техникой .....                 | 38 |
| 5.3 Требования охраны труда при эксплуатации электрооборудования офисными работниками ..... | 40 |
| ЗАКЛЮЧЕНИЕ .....  | 44 |
| БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....  | 45 |
| ПРИЛОЖЕНИЯ .....  | 48 |
| Приложение А. ЛИСТИНГ ФАЙЛА INDEX.PHP .....   | 48 |
| Приложение Б. ЛИСТИНГ СТИЛЕЙ.....   | 56 |
| Приложение В. ГЛАВНАЯ СТРАНИЦА САЙТА.....   | 85 |

## ВВЕДЕНИЕ

ООО «КАИС + Ноябрьск» организация, которая осуществляет продажу автомобилей, сотрудничая со всеми автосалонами города Нижневартовск. ООО «КАИС + Ноябрьск» размещает предложения автосалонов г. Нижневартовск и после того, как покупатели выбирают нужный им автомобиль, связывается в автосалонами и привозит автомобили в г. Ноябрьск.

Для того, чтобы люди могли получить более полную и актуальную информацию о наличии автомобилей, ООО «КАИС + Ноябрьск» необходимо разработать Интернет-сайт, на котором будут предоставлены все имеющиеся автомобили автосалонов г. Нижневартовск.

Интернет-сайт должен быть рассчитан на среднестатистического малоопытного пользователя сети Интернет. Поэтому, наряду с применением графических дизайн-технологий, интерфейс сайта должен быть максимально упрощен и достаточно удобен и информативен.

Интернет-сайт должен носить не только информационно-ознакомительный характер, но и содержать элементы рекламного характера.

С точки зрения системного администратора, в чьи функции входят поддержание работоспособности Интернет сайта, регулярное обновление и др., проект требует к себе минимум затрат по эксплуатации и администрированию, и каких либо проблем, а также необходимость в существенных затратах времени и финансовых средств при администрировании, не возникает.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

## 1.1 Описание предметной области

Предметной областью данной выпускной квалификационной работы (далее – ВКР) является Интернет-сайт на котором будут предоставлены все имеющиеся автомобили автосалонов г. Нижневартовск для информирования жителей г. Ноябрьск.

В настоящее время информация о предложениях автосалонов г. Нижневартовск доступна в офисе ООО «КАИС + Ноябрьск». Для того, чтобы люди могли получить более полную и актуальную информацию о наличии автомобилей, ООО «КАИС + Ноябрьск» необходимо разработать Интернет-сайт, на котором будут предоставлены все имеющиеся автомобили автосалонов г. Нижневартовск, доступ к которому будет возможен как с персонального компьютера, так и с мобильного устройства.

## 1.2 Цель работы

Целями и задачами данной работы является создание интернет-сайта для ООО «КАИС + Ноябрьск», для привлечения пользователей и снижения нагрузки на менеджеров по продажам. Сайт должен отвечать следующим требованиям:

1. Информация, находящаяся на интернет-сайте должна быть наиболее содержательной и лаконичной, чтобы не перегружать посетителя сайта обилием информации.
2. Информация должна быть тщательно отсортированной по категориям, чтобы пользователю не приходилось тратить время на поиск нужной ему информации.
3. Интерфейс сайта должен быть максимально простым и интуитивно-понятным.



4. Внешний интерфейс должен быть привлекательным, функциональным, но не перегруженным различными элементами управления. Цветовая гамма, а также дизайн-оформление должно соответствовать тематике сайта.
5. Внешний интерфейс, а также текстовое содержимое сайта, должно по возможности носить не только информирующий, но и рекламный характер.
6. Интерфейс администратора должен быть максимально функциональным и не должен содержать ненужных элементов.
7. На сайте, помимо основных элементов, должны присутствовать процедуры регистрации, а также системы защиты от несанкционированного и неавторизированного доступа.

### 1.3 Постановка задачи

В результате создания проекта будет реализована возможность для покупателей иметь актуальную и постоянную о наличии автомобилей в ООО «КАИС + Ноябрьск».

### 1.4 Задачи, решаемые в ходе работы

На стадии разработки интернет-сайта, разработчику было поставлено несколько задач:

1. Разработать функциональную схему информативной части интернет-сайта, собрать и отфильтровать всю предоставленную покупателям информацию. Выделить наиболее приоритетные части и скомпоновать их таким образом, чтобы при размещении информации на сайте, пользователю предоставлялась максимальная свобода в выборе нужной информации по разделам, а также быстрый доступ к нужной информации.

2. Разработать рабочую часть проекта с использованием технологий Java Script, PHP, HTML, Dynamic HTML. Сборка и компоновка отдельных программных модулей, процедур и блоков, представляющих компоненты программного «движка» (основы).
3. Разработка дизайна внешнего интерфейса. Создание дизайнерских шаблонов внешнего интерфейса и согласование с заказчиком о наиболее оптимальном решении, касающемся дизайна. Утверждение окончательного варианта внешнего интерфейса интернет-сайта.
4. Дизайн сайта не должен повторять сайты фирм города Нижневартовск.  
Сборка и настройка модулей и процедур в единое программное целое, а также последующая сборка и тестирование.

## 2 ОПИСАНИЕ СРЕДЫ ПРОГРАММИРОВАНИЯ ДЛЯ WEB-СТРАНИЦ

### 2.1 Формализация и стандартизация данных и задач

Прежде чем приступить к проектированию Веб-сайта, необходимо провести формализацию и стандартизацию данных и задач, являющихся необходимыми для этапов разработки Веб-сайта.

Реализация проекта выполняется в несколько этапов:

1. Предварительные исследования;
2. Разработка технического задания;
3. Создание концепции дизайна сайта;
4. Программирование создание сайта;
5. Тестирование;
6. Устранение недостатков;
7. Сдача проекта заказчику.

Предварительные исследования – это одна из самых важных частей проектирования, поскольку ошибки на этом этапе приводят к значительному снижению ценности выполненной работы. В предварительные исследования входят:

1. Уточнение целей и задач;
2. Изучение бизнеса клиента, определение и анализ целевой аудитории;
3. Анализ решений использованных конкурентами;
4. Составление календарного плана работ;
5. Формирование бюджета и рабочей группы;

Далее следует разработка и утверждение окончательного технического задания, выбор площадки размещения и соответственно формирование требований к аппаратной части проекта.

### 2.2 Характеристика технологии HTML

Разработка и согласование дизайна и разработка структуры сайта оказываются неразделимыми на этапе создания концепции дизайна сайта. Дизайн и структура очень взаимосвязаны, поскольку изменения в объеме информации требуют изменения её представления.

Всемирная паутина World Wide Web (WWW) состоит из Web-страниц, которые создаются с помощью языка разметки гипертекста HTML (HyperText Markup Language). Хотя многие говорят о программировании на этом языке, HTML вовсе не является языком программирования в традиционном понимании. HTML – язык разметки документа. При разработке HTML-документа выполняется разметка текстового документа точно так же, как это делает редактор при помощи красного карандаша. Эти пометки служат для указания формы представления информации, содержащейся в документе.

Каждый раз, когда пользователь открывает Web-страницу, он на самом деле открывает документ, написанный на HTML. Все Web-документы отформатированы с помощью языка HTML, и все гиперссылки, с помощью которых пользователь легко перемещается со страницы на страницу, также организованы средствами языка HTML. И те цветные изображения, которые встречаются на Web-страницах, и заполняемые формы, и бегущие заголовки – все это мы видим благодаря языку HTML.

Специальные программы просмотра HTML-документов, которые часто называют браузерами, служат для интерпретации файлов, размеченных по правилам языка HTML, форматирования их в виде Web-страниц и отображении их содержимого на экране компьютера пользователя. Существует большое количество программ-браузеров, разработанных различными компаниями, однако, на сегодняшний день из всего разнообразия программ явно выделяются две программы-лидера – Google Chrom и Microsoft Internet Explorer.

Таким образом, исходя из вышесказанного можно сделать вывод, что технология HTML является на сегодняшний день практичной и востребованной технологией создания и функционирования динамических, интерактивных интернет - приложений.

### 2.3 Характеристика технологии php

PHP – язык создания сценариев, который давно перерос свое название. Дело в том, что PHP – это аббревиатура от слов Personal Home Page. Первая версия PHP была создана Расмусом Лердорфом в 1994 г. и представляла собой набор инструментов для отслеживания посетителей Web-страницы. Со временем PHP из набора инструментов превратился в полноценный язык программирования, а его название было изменено как рекурсивное образование PHP HyperText Preprocessor (препроцессор гипертекста PHP).

PHP – это серверный язык создания сценариев. Конструкции PHP, вставленные в HTML-текст, выполняются сервером при каждом посещении страницы. Результат их обработки вместе с обычным HTML-текстом передается браузеру.

В настоящее время основной версией PHP является пятая.

Существуют два основных конкурента PHP: Active Server Pages (ASP) компании Microsoft и ColdFusion компании Allaire. По сравнению с ними PHP обладает рядом преимуществ, в числе которых:

**Высокая производительность.** PHP-программы работают быстрее, чем ASP.

**Функциональность.** Разработку PHP-программы можно отделить от собственно разработки Web-страницы, что упростит жизнь и программисту, и дизайнеру.

**Цена.** PHP абсолютно бесплатен.

**Простота в использовании.** Имеющие опыт программирования на распространенных языках найдут синтаксис PHP хорошо знакомым.

**Переносимость.** Один и тот же PHP-код можно использовать как в среде NT, так и на платформах UNIX.

Общие правила построения PHP-программы

PHP-программы состоят из простого текста, поэтому набирать их можно в любом текстовом редакторе. Популярные HTML-редакторы имеют встроенную поддержку для редактирования PHP-программ.

Расширение файлов PHP-программ по умолчанию в PHP4 - .php. На основании этого расширения сервер распознает файл как PHP-программу и запускает интерпретатор.

Исходя из вышеописанного нами для реализации интернет-сайта о наличии автомобилей в ООО «КАИС + Ноябрьск» будет выбран язык php.

### 3 РАЗРАБОТКА ИНТЕРНЕТ-САЙТА

#### 3.1 Методика разработки проекта

Под сбором информации подразумевается получение необходимой информации от заказчика, а также из дополнительных источников.

Следующим этапом разработки являлась селекция и компоновка всей собранной информации. Все полученные данные преобразовывались с помощью логической выборки, в набор упорядоченной по категориям информации. Вся информация была четко отсортирована по категориям, после чего была преобразована в четкую, иерархическую структуру по категориям. Далее были созданы наработки готовых программных шаблонов, содержащих интерактивные компоненты.

Одной из основных и наиболее значимых задач при разработке Веб-сайта является создание оптимального дизайна интерфейса Интернет-сайта. Для достижения оптимального эффекта, было разработано несколько графических дизайн-шаблонов внешнего интерфейса. Далее, после согласования с заказчиком, был отобран наиболее оптимальный вариант.

Интерфейс сайта должен носить не только привлекательный, но и рекламный характер оформления и не содержать лишних элементов. Не должно быть лишней информации, чтобы интерфейс не выглядел перегруженным, и загрузка сайта происходила в максимально короткое время. В связи с этим был выбран наиболее простой, но максимально функциональный вариант внешнего интерфейса. Веб-сайт рассчитан на любого пользователя Интернета. Даже неопытные пользователи могут с успехом просмотреть его содержимое.

Следующим этапом разработки являлось создание пробных экспериментальных программных модулей программной основы (программного движка).

Особенностью данного этапа разработки является активное использование опыта других разработчиков. Для этого была использована

информация, а также готовые демонстрационные программные компоненты, полученные из сети Интернет, путем поиска нужной информации на сайтах разработчиков любителей и разработчиков профессионалов. Далее была также проведена селекция полученных данных и отборка наиболее подходящих из них, после этого, были внимательно изучены методы и принципы, лежащие в основе программных компонентов созданными независимыми разработчиками. После детального изучения принципов работы и применяемых методик программирования, были созданы программные шаблоны, и процедуры для сборки рабочей части программного «движка» Веб-сайта. Далее следует этап разработки, в ходе которого производится тестирование и отладка отдельных программных компонентов и блоков, предназначенных для сборки готового проекта Веб-сайта.

Завершающим этапом разработки является сборка и тестирование готового проекта Веб-сайта. Методикой работы на данном этапе является много платформенное тестирование готового программного продукта. Готовый проект тестировался на работоспособность на локальном компьютере разработчика, на сетевом сервере, а также на различных операционных системах с установленным набором программных компонентов.

### 3.2 Использование парсинга при создании нового Интернет-ресурса

При создании нового Интернет-ресурса для ООО «КАИС + Ноябрьск» перед вэб-мастером возникает вопрос наполнения сайта контентом. И для того, чтобы максимально заполнить сайт, многие обращаются за помощью в Интернет, где можно найти любую информацию. Но этот метод не так и прост: владельцу сайта приходится столкнуться с рядом значимых проблем:

необходимость размещения большого количества информации. Чем больше информации размещено на отдельном сайте, тем больше времени станет проводить на нем рядовой пользователь. Это вынуждает владельцев ресурса



размещать на нем немислимые объемы контента, из-за чего ручное размещение становится затруднительным;

необходимость в свежей информации. Если большие объемы информации меняются часто, с обновлением информации на сайте не справится даже команда специально обученных сотрудников. В таких случаях смысл ручного обновления контента просто теряется из-за отсутствия физической возможности;

поиск информации для своего сайта с помощью копирования данных других ресурсов. Зачастую парсинг используют для сателлитов и наполнения сайта простыми элементами, например, отзывами о кинофильмах и кулинарными рецептами. Скопированный фрагмент обрабатывается синонимайзером и рерайтером для того, чтобы повысить уникальность полученного материала.

Именно здесь в игру вступает парсинг ресурсов, способный в автоматическом режиме собирать, структурировать и изменять информацию. Парсер – это программа, которая осуществляет синтаксический анализ текста. В отличие от человека, программный парсер способен:

просмотреть и проанализировать интернет-страницы автосалонов г. Нижневартовск, с которыми сотрудничает ООО «КАИС + Ноябрьск» за ограниченный промежуток времени;

дифференцировать технические данные и «человеческую» языковую оболочку для информации;

отобрать нужные данные и отбросить лишнюю «воду»;

собрать конечные данные в простом и понятном виде.

Результаты работы программа-парсер представляет в виде базы данных, простой таблицы, которая обрабатывается соответствующим образом, или файлом формата XML. Но обработка баз данных не относится к парсингу как таковому. Сам результат парсинга – это ясная структура синтаксиса предложения, которую парсер представляет в формате:

1. древа зависимостей;
2. древа составляющих;

3. комбинированным способом.

Принцип работы парсера

Парсер – это программа, разбитая на три подпрограммы, каждая из которых имеет свои функции и задачи, выполняет различные действия:

получение информации в ее в исходном виде. Зачастую парсер просто скачивает код страницы, на которой размещены нужные данные, используя при этом библиотеку с URL для PHP;

выборка и приведение данных к единому формату. С помощью регулярных выражений из скачанного кода извлекаются нужные фрагменты;

компоновка результата. Завершающая часть процесса парсинга, при которой программа компоует и записывает полученные данные в таблицу или файл.

Парсинг давно стал один из обязательных инструментов для работы с сайтом для большинства вэб-мастеров: он позволяет находить актуальную информацию в кратчайшие сроки, анализирует, структурирует ее и подает в удобном для просмотра виде, чем экономит время и силы.

### 3.3 Используемая библиотека для парсинга сайтов

Обычно, учат писать конечный автомат. Суть его в том, что мы перебираем, посимвольно, весь html текст, находим начало тега, и строим дерево документа. Так называемое DOM (Document Object Model)

Сейчас, писать такое самому нет необходимости.

В php, начиная с версии 5, есть встроенные методы работы с деревом документа (класс DOMDocument), но основан он на XML парсере.

А HTML и XML это хоть и очень похожие, но в тоже время абсолютно разные технологии.

К примеру, неременное требование к XML это закрытые теги и отсутствие ошибок.

Отсюда вытекает условие: ошибок в html, который мы парсим с помощью нативных средств php, быть не должно.

К сожалению, на сайтах донорах, ошибки не редки, а значит этот метод отпадает.

Для корректного разбора таких сайтов, на помощь придут php библиотеки PHPQuery, Simple HTML DOM, Zend DOM Query, Nokogiri .

Некоторые из них, после небольших манипуляций скармливают html тому же DOMDocument. Мы не будем их рассматривать.

Мы будем использовать SimpleHTMLDOM.

Скачиваем последнюю версию.

В архиве, две папки (примеры работы и документация) и файл simple\_html\_dom.php.

simple\_html\_dom.php - это и есть вся библиотека, больше ничего для работы не потребуется. Кидаем этот файл в папку с проектом и в своем скрипте просто подгружаем его.

```
1. include 'simple_html_dom.php';
```

Файл подключен и готов к работе.

Для того, чтобы начать разбирать HTML, его сперва нужно получить. В simplehtmlDOM есть методы для удаленной загрузки страниц. После подключения файла библиотеки, нам доступны 2 функции для обработки HTML строк.

```
str_get_html(str) и file_get_html(url)
```

Они делают одно и тоже, преобразуют HTML текст в DOM дерево, различаются лишь источники.

str\_get\_html – на вход получает обычную строку, т.е. если мы получили HTML прибегнув к curl, или file\_get\_contents то мы просто передаем полученный текст этой функции.

```
1 $html = str_get_html('<html><body>Привет!</body></html>');
```

file\_get\_html – сама умеет загружать данные с удаленного URL или из локального файла

```
1 $html = file_get_html('http://toyota-nvartovsk.ru/');
```

Или

```
1 $html = file_get_html('data/test.htm');
```

К сожалению, `file_get_html` загружает страницы обычным `file_get_contents`. Это значит если хостер, выставил в `php.ini` `allow_url_fopen = false` (т.е. запретил удаленно открывать файлы), то загрузить что-то удаленно, не получится. Да и серьезные веб сайты таким способом парсить не стоит, лучше использовать `CURL` с поддержкой `proxy` и `ssl`. Однако для нашего сайта, вполне хватит и `file_get_html`.

```
1 $html = file_get_html('http://toyota-nvartovsk.ru/');
```

в результате, в переменной `$html` будет объект типа `simple_html_dom`.

При больших объемах данных, в библиотеке происходит утечка памяти.

Поэтому после окончания одного цикла надо ее чистить.

Делает это метод `clear`.

```
$html->clear(); // подчищаем за собой  
unset($html);  
}
```

Эти две строчки `$html->clear();` и `unset($html);` лучше писать сразу же после того, как мы создали объект. Иначе скрипт отвалится, забив всю память.

После того, как `html` текст упакован в объект, можно приступить непосредственно к поиску нужных элементов.

Большинство поисковых функций выполняет метод `find(selector, [index])`. Если второй аргумент не задан, метод возвращает массив элементов. Если же задан, то элемент этого массива с индексом `index`.

Пример: скачаем страницу сайта `toyota-nvartovsk.ru`, и выведем все марки автомобилей и их стоимость.

```
<?php
```

```

# Получаем объект для работы с DOM страницы
$html = new simple_html_dom();
$html = file_get_html('http://toyota-nvartovsk.ru/new-cars/');
foreach($html->find('div[class=row] strong, span[class="price"]') as $title)
echo $title->plaintext. '<br>'; $html->clear(); // подчищаем за собой
unset($html);
?>

```

На данной странице, в качестве селектора воспользовались названием тега `<strong>`. Но можно использовать и другие CSS селекторы. Элемент на странице можно найти по его атрибутам. В первую очередь, это название тега, `id` и `class`. Также могут быть использованы и второстепенные атрибуты, к примеру, `span`. Если и этих атрибутов нет, то можно воспользоваться и регулярными выражениями.

Поиск по названию тега

```

1 $html->find('div')
поиск по id
1 $html->find('#preview')
поиск по классу
1 $html->find('.myclass')

```

или комбинированный вариант

```

1 $html->find('#preview div.myclass')

```

в данном случае, сначала найдется элемент с `id= preview` затем в нем найдутся все теги `div`, и уже среди них фильтруются те у которых `class="myclass"`

Если метод `find` ничего не нашел и `index` не задан, то он возвращает пустой массив. Если же `index` задан, то метод возвращает `null`.

Поэтому верным решением будет проверить

```

1 if(count($html->find('#preview div.myclass')))
2 foreach($html->find('#preview div.myclass') as $div)

```

3 echo \$div->innertext;

Поиск по наличию атрибута

1 \$html->find('img [width]'); // найдет нам все изображения у которых задан атрибут ширина

или более конкретный поиск по значению атрибута

1 \$ret = \$html->find('img[width=400px]');// найдет все изображения, у которых задана ширина равная 400px

Такая нотация позволяет искать по двум и более смежным классам

1 \$ret = \$html->find('img[class=active myclass]');//<img class="active myclass"/>

Поиск нескольких тегов

1 \$html->find('a, img, br,span');

Поиск вложенных тегов

1 \$es = \$html->find('ul.myclass li');// найдет все li который является потомком ul(возможно и не прямым)

2 \$es = \$html->find('div.myclass li');// найдет все li в div.myclass

У каждого найденного элемента также есть метод

1 find\$html->find('div.myclass li');//найдет все div.myclass а потом все li лежащие в них

если нам нужно найти все li только первого div'a то мы можем написать так

1 \$html->find('div.myclass',0)->find('li');

Поиск по значению атрибута не ограничивается только равенством. Вот доступные условия

[атрибут] – проверяет есть ли у элемента данный атрибут

[атрибут=величина] – проверяет, есть ли у элемента данный атрибут и равно ли его значение величине.( div[class=myclass] – найдет все div'ы у которых class равен myclass)

[атрибут!=величина] – проверяет, есть ли у элемента данный атрибут и не равно ли его значение величине.( div[class!=myclassok] – найдет все div'ы у которых class не равен myclassok)

[атрибут^=величина] – проверяет, есть ли у элемента данный атрибут и начинается ли его значение с величины ( div[class^=my] – найдет все div'ы у которых class начинается с my, к примеру myclass и myclassok)

[атрибут\$=величина] – проверяет, есть ли у элемента данный атрибут и заканчивается ли его значение величиной( div[class\$=ok] – найдет все div'ы у которых class заканчивается на ok, к примеру myclassok, yok, окно не oki)

[атрибут\*=величина] – проверяет, есть ли у элемента данный атрибут и содержит ли его значение в себе величину, в любом месте(div[class\*=sok] – найдет все div'ы у которых class содержит sok, к примеру myclassok, ysoki, sok)

Обычный текст можно искать как `ter text| $es = $html->find('text'); //`  
найдет все текстовые блоки в html

Комментарии находим по тегу comment

```
1 $es = $html->find('comment');
```

Каждый найденный элемент и сам \$html имеют 5 полей

```
1 $html = str_get_html("<div>foo <b>bar</b></div>");
```

```
2 echo $html; // выведет <div>foo <b>bar</b></div>;
```

```
3 $e = $html->find("div", 0);
```

```
4 echo $e->tag; // Вернет: "div"
```

```
5 echo $e->outertext; // Вернет: <div>foo <b>bar</b></div>
```

```
6 echo $e->innertext; // Вернет: foo <b>bar</b>
```

```
7 echo $e->plaintext; // Вернет: foo bar
```

`$e->tag` Читает или записывает имя тега элемента.

`$e->outertext` Читает или записывает весь HTML элемента, включая его самого.

`$e->innertext` Читает или записывает внутренний HTML элемента

`$e->plaintext` Читает или записывает простой текст элемента, это эквивалентно функции `strip_tags($e->innertext)`. Хотя поле доступно для записи, запись в него ничего не даст, и исходный html не изменит

```
1 $html = str_get_html("<div>foo <b>bar</b></div>");
2 $div = $html->find('div',0);
3 $div->plaintext = 'gooo';
4 echo $div->innertext; // вернет <div>foo <b>bar</b></div>
```

Для удаления ненужного элемента из HTML можно затереть его поле `outertext`

```
1 $html = str_get_html("<div>foo <b>bar</b></div>");
2 $b = $html->find('b',0);
3 $b->outertext = "";
4 echo $html->innertext; // вернет <div>foo</div>
```

Тут следует помнить, что хоть элемент и не виден в html, из дерева DOM он никуда не делся

```
1 $html = str_get_html("<div>foo <b>bar</b></div>");
2 $b = $html->find('b',0);
3 $b->outertext = "";
4 echo $html->innertext; // вернет <div>foo</div>, элемент удален из HTML
5 // но
6 echo count($html->find('b')); // вернет 1, в дерево элемент
```

присутствует

при желании мы даже можем вернуть элемент на место

```
1 $b->outertext = '<span>bar</span>';
2 echo $html->innertext; // вернет <div>foo<span>bar</span></div>
```

Для более эффективной навигации по дереву документа доступны методы

`$e->children ( [int $index] )` Возвращает объект N-го прямого потомка, если индекс установлен, в противном случае возвращает массив всех дочерних элементов

`$e->parent()` Возвращает родительский элемент.



`$e->first_child()` Возвращает первый дочерний элемент, или `null`, если ничего не найдено

`$e->last_child()` Возвращает последний дочерний элемент, или `null`, если ничего не найдено

`$e->next_sibling()` Возвращает следующий родственный элемент, или `null`, если ничего не найдено

`$e->prev_sibling()` Возвращает предыдущий родственный элемент, или `null`, если ничего не найдено

```
1 $html = "<div>
```

```
2 <b>bar</b>
```

```
3<b>foo</b>
```

```
4 <span>arg</span>
```

```
5 <div>
```

```
6 <b>tor</b>
```

```
7 </div>
```

```
8 </div>";
```

Все дочерние элементы разные, как-то подобрать к ним селектор проблематично. Поэтому воспользуемся описанными методами.

```
1 $html = str_get_html($html);
```

```
2 $div = $html->find('div',0);
```

```
3 $i = 0;
```

```
4 while($item = $div->children($i++){
```

```
5   echo $item->innertext;
```

```
6 }
```

либо так

```
1 $item = $div->children(0);
```

```
2 echo $item->innertext;
```

```
3 while($item = $item -> next_sibling()){
```

```
4 echo $item->innertext;
5 }
```

Данные методы полезны при разборе таблиц, элементы которых, как правило, структурированы, но не имеют идентифицирующих атрибутов.

Ну и последняя фишка это вызов callback функции на найденный элемент

```
1 function my_callback($element) {
2     if ($element->tag=='span')
3         $element->outertext = '<b>'.$element->innertext. '</b>'; // заменим
```

все span элементы на b

```
4}
```

```
5$html
```

```
str_get_html('<span>bar</span><span>pole</span><span>sushi</span><a>okno</a>'); // Регистрация функции обратного вызова с ее именем
```

```
6$html->set_callback('my_callback'); // вызов функции произойдет при конвертации объекта в строку
```

```
7 echo $html; // на самом деле, при этом вызывается магический метод __toString, он и запускает
```

На экране мы увидим

```
1 <b>bar</b><b>pole</b><b>sushi</b><a>okno</a>
```

Доступ к атрибутам элементов осуществляется напрямую

```
1 foreach($html->find('img') as $img) echo $img->src;
```

```
2//или
```

```
3 echo $html->find('img',0)->src;
```

Таким образом, мы рассмотрели Simple HTML DOM — PHP-библиотека, позволяющая парсить HTML-код с помощью удобных jQuery-подобных селекторов.

### 3.4 Листинг сайта

Код страниц сайта мы будем рассматривать на примере одной страницы, так как остальные будут созданы по аналогии, применяя методы описанные выше к каждому сайту ,в зависимости от исходного кода сайта донора.

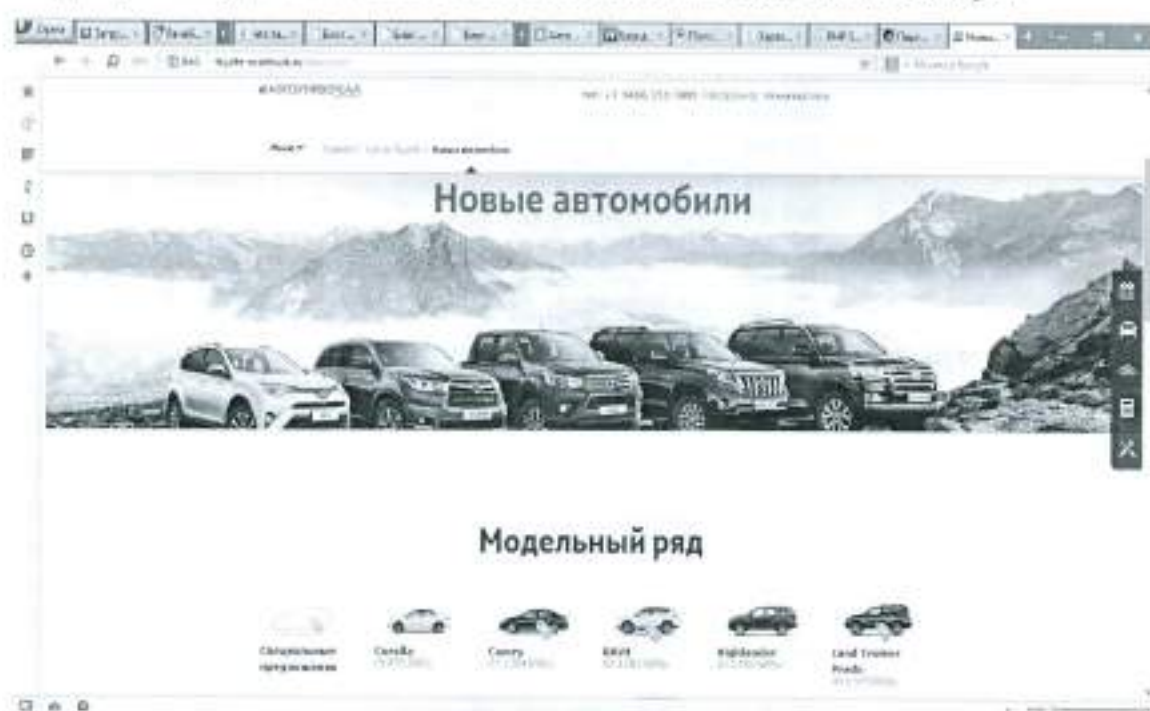


Рисунок 1 – Внешний вид сайта донора

Нам нужно на своем сайте отобразить марку автомобиля и его стоимость.

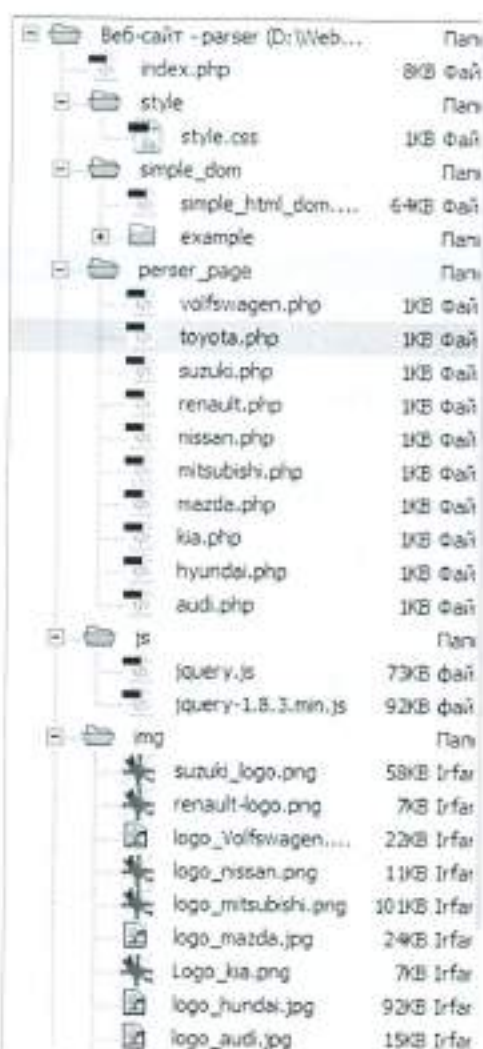
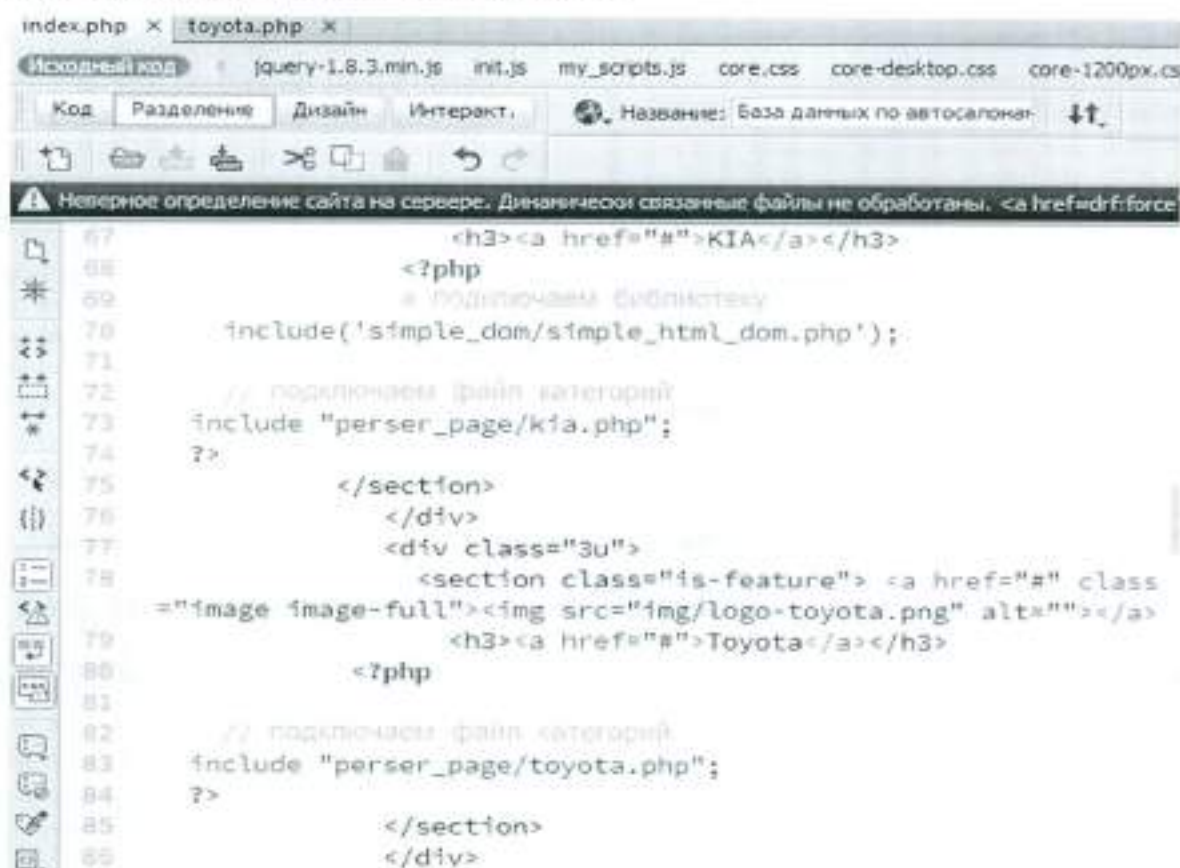


Рисунок 2 – Структура файлов для нашего сайта

Для каждой марки автомобиля и каждого автосалона мы создадим свой файл парсинга.

Рисунок 3 – Файл парсинга

Вставляем на главную страницу сайта.



```
index.php x toyota.php x
Исполнитель: chrome
jquery-1.8.3.min.js init.js my_scripts.js core.css core-desktop.css core-1200px.cs
Код Разделение Дизайн Интеракт. Название: База данных по автосалонам
Неверное определение сайта на сервере. Динамически связанные файлы не обработаны. <a href=drf:force
67 <h3><a href="#">KIA</a></h3>
68 <?php
69 // подключаем библиотеку
70 include('simple_dom/simple_html_dom.php');
71
72 // подключаем файл категорий
73 include "perser_page/kia.php";
74 ?>
75 </section>
76 </div>
77 <div class="3u">
78 <section class="is-feature"> <a href="#" class
79 ="image image-full"></a>
80 <h3><a href="#">Toyota</a></h3>
81 <?php
82 // подключаем файл категорий
83 include "perser_page/toyota.php";
84 ?>
85 </section>
86 </div>
```

Рисунок 4 – Вставка файла парсинга на главную страницу сайта



Рисунок 5 – Результат вставки файла парсинга

## 4 ЭКОНОМИЧЕСКИЙ РАЗДЕЛ

### 4.1 Расчет затрат на разработку

Проект создания Веб-сайта реализуется в том случае, если это экономически целесообразно. Для определения экономического эффекта, необходимо подсчитать:

1. количество и стоимость времени, затраченного на разработку сайта;
2. стоимость обслуживания сайта;
3. возможные финансовые и рекламные дивиденды.

Далее будут проведены все необходимые расчеты и выкладки.

Распределение трудоемкости по стадиям разработки приведено в таблице

№1.

Таблица 1 – Распределение трудоемкости

| <i>Этап разработки</i>     | <i>Описание содержания работ</i>  | <i>Трудоёмкость (час.)</i> |
|----------------------------|---|----------------------------|
| 1                          | 2   | 3                          |
| Постановка задачи          | Предпроектное обследование.<br>Разработка, согласование и утверждение технико-экономического обоснования.   | 14                         |
| Техническое проектирование | Уточнение структуры и формы представления входных и выходных данных. Разработка алгоритма решения задачи. Разработка структуры программы.<br>Разработка плана мероприятий по разработке и внедрению проекта.<br>Разработка пояснительной записки.<br>Согласование и утверждение технического проекта. | 42                         |
| Рабочее проектирование     | Написание программы на языке программирования, первичная отладка, тестирование.<br>Разработка, согласование и утверждение порядка и методики испытаний, корректировка программы.  | 82                         |

Продолжение таблицы 1

| 1                             | 2  | 3         |
|-------------------------------|--|-----------|
| Документирование и внедрение. | Разработка программной документации.<br>Сдача программы и программной документации | 28        |
| ИТОГО:                        |  | 166 часов |

Коэффициент сложности задачи  $c$  – характеризует относительную сложность программы по отношению к так называемой типовой задаче, сложность которой принята равной единице (величина  $c$  лежит в пределах от 0,5 до 1,5). Так как программа не является сложной, чтобы ей присвоить максимальное значение  $c = 1,5$ , но она и не является простой, чтобы присвоить 0,5. Поэтому для моего программного продукта сложность задачи берем 0,6.

$c = 0,6$  – (коэффициент сложности программы)

Коэффициент коррекции программы  $p$  – увеличение объема работ за счет внесения изменений в алгоритм или программу по результатам уточнения постановок (меняется от 0,1 до 0,5). С учетом того, что при разработке интернет-сайта я достаточно хорошо представлял себе конечный результат ПП, возьмем коэффициент равный 0,1

$p = 0,1$  – коэффициент коррекции программы в ходе разработки.

При расчете основной З/П следует учитывать следующие коэффициенты:

Коэффициент увеличения затрат труда, вследствие недостаточного описания задачи. В зависимости от сложности задачи принимается от 1 до 1,5. Так как задача была поставлена в общем виде, то  $b = 1,3$ .

Коэффициент квалификации разработчика  $k$  определяется в зависимости от стажа работы и составляет: для работающих до двух лет – 0,8; от трех лет до пяти лет – 1,0; от пяти до восьми – 1,2; свыше восьми лет – 1,5. Поскольку стажа работы по специальности у меня нет, возьмем  $k = 0,8$ .

Расчёт фонда заработной платы разработчика с учётом всех выше перечисленных коэффициентов выводится по формуле:

$З/П_{осн} = з/п * c * (p+1) * b * k = (15000 \text{ руб.} * 0,6 * (0,1 + 1) * 1,3 * 0,8) = 10296 \text{руб.}$  – в месяц. Где з/п – средняя заработная плата разработчика данного предприятия или фирмы, составляющая 15000руб/месяц.

Поскольку дополнительная заработная плата в виде доплат и премий разработчику программного продукта за время выполнения работ не выплачивалась, то заработная плата основная будет равна заработной плате суммарной.

Страховые отчисления берутся в размере 26% от заработной платы:

$$C_o = (З/П_{осн} / 100 * 26) = 10296 / 100 * 26 = 2676,96 \text{руб.}$$

Суммарные затраты предприятия или фирмы за время разработки программного продукта с учётом страховых отчислений составит:

$$З/П_{общ} = (З/П_{осн} + C_o) * 166/176 = 12235,86 \text{руб.}$$

#### 4.2 Содержание и эксплуатация вычислительной техники

Для расчёта затрат данного вида необходимо иметь значения следующих показателей:

- 1) Стоимость электроэнергии.
- 2) Амортизационные отчисления.
- 3) Техническое обслуживание, или затрат на ремонт.

Стоимость 1 кВт/час электроэнергии составляет: 3руб.

Один компьютер потребляет в среднем 300 Вт в час.

Годовые отчисления затрат электроэнергии потребляемые ПК составят:

$$8 \text{ч.} * 22 \text{дня} * 12 \text{мес.} * 0,3 \text{кВт/ч.} * 3 \text{руб.} = 1900,8 \text{руб.}$$

Амортизационные отчисления зависят от срока полезного использования ПК, а так как данный срок эксплуатации компьютера 3 года, а срок его полезного использования 4 года, то необходимо исчислить амортизационные отчисления.

Стоимость компьютера составляет 25000руб.

Отчисления = стоимость ПК/ срок полезного использования:

$$25000 \text{руб.} / 4 \text{ года} = 6250 \text{руб.}$$



Затраты на ремонт ПК составляют 5% от её стоимости – 1250р.

Действительный фонд работы ПК:

$$\Phi_{\text{ит}} = \Phi_{\text{ном}} - \Phi_{\text{проф}}$$

$$\Phi_{\text{ном}} = 8\text{ч.} * 22\text{дня} * 12\text{мес.} = 2112\text{ч.} - \text{номинальный фонд работы ПК.}$$

$\Phi_{\text{проф}} = 2112/100 * 5 = 105,6\text{ч.}$  – Годовые затраты времени на профилактические работы, 5% от  $\Phi_{\text{ном}}$ .

$$\Phi_{\text{ит}} = 2006\text{ч.}$$

Стоимость машинного часа равна:

$$1900,8\text{руб.} + 1250\text{руб.} / 2006\text{ч.} = 1,5\text{руб.}$$

Содержание и эксплуатация вычислительного комплекса на время написания дипломного проекта составляет (данные из таблицы №1):

$$152\text{час.} * 1,5\text{руб.} = 228\text{руб.}$$

Затраты размещения на платный хостинг оценивается в пределах 600руб. в месяц. Затраты по оплате хостинга в течение года составят:

$$600\text{руб.} * 12\text{мес.} = 7200\text{руб.}$$

Так как компьютер для функционирования этого программного продукта уже имеется, и он удовлетворяет требованиям ПО, то новый покупать не надо. Привлечение дополнительных сотрудников также не требуется.

Затраты на разработку программного продукта приведены в таблице 2.

Таблица 2 – Затраты на разработку ПО

| Наименование расходов                                 | Затраты (руб.) |
|---|----------------|
| Основная заработная плата.                            | 10296,00       |
| Страховые отчисления (26% от фонда заработной платы). | 2676,96        |
| Содержание и эксплуатация вычислительного комплекса   | 228,00         |
| Хостинг   | 7200,00        |
| Всего   | 20400,96       |

### 4.3 Экономический эффект

Рассчитаем экономический эффект, получаемый за счёт роста числа клиентов фирмы ООО «КАИС + Ноябрьск» обусловленного разработкой и использованием сайта.

Месячная прибыль от продажи продукции в фирме составлял 400000руб.

Следствием результата использования созданного сайта является рост числа клиентов и соответственно получение дохода.

За счёт использования сайта при возросшем числе посетителей обеспечен рост получения дохода на 15% (400000руб. \* 0,15 = 60000руб.).

Рассчитаем срок окупаемости созданного сайта:

$$20400,96\text{руб.} / 60000\text{руб.} = 0,34\text{года}$$

Вывод: Если фирма задействует в своей практике разработанный сайт, то дополнительный месячный доход при этом будет определяться в пределах 60000 рублей, а срок окупаемости составит 4 месяца.

## 5 БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

### 5.1 Требования безопасности

Требования безопасности, направленные на предотвращение неблагоприятного влияния на здоровье человека вредных факторов производственной среды и трудового процесса при работе с персональными электронно-вычислительными машинами (ПЭВМ) содержатся в СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы», утвержденного Постановлением Главного санитарного врача Российской Федерации от 03.06.2003 г. №118 (в редакции от 03.09.2010 г.).

При работе с ПЭВМ (компьютерами) необходимо соблюдать следующие меры безопасности и охраны труда:

- эксплуатация ПЭВМ должна осуществляться в помещениях с естественным и искусственным освещением;
- оконные проемы должны быть оборудованы регулируемыми устройствами типа жалюзи, занавесей, внешних козырьков, позволяющих исключить прямую блескость, создаваемую солнечными лучами;
- площадь на одно рабочее место пользователей, работающих с видеодисплейными терминалами (мониторами) на базе электронно-лучевой трубки должна составлять не менее 6 квадратных метров, с мониторами на базе плоских дискретных экранов (жидкокристаллические, плазменные) – не менее 4,5 квадратных метров;
- светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов;
- расстояние от глаз до экрана видеодисплейного терминала должно находиться в пределах 600-700 мм, но не ближе 500 мм;

- в помещении с ПЭВМ должна производиться ежедневная влажная уборка пола и мебели. Помещения с работающими ПЭВМ необходимо проветривать после каждого часа работы;

- женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3 часов за рабочую смену) при условии соблюдения гигиенических требований;

- продолжительность непрерывной работы с видеодисплейным терминалом без регламентированного перерыва не должна превышать 1 час;

- для предупреждения преждевременной утомляемости рекомендуется организовывать рабочую смену путём чередования работ с использованием ПЭВМ и без неё;

- при работе с текстовой информацией рекомендуется выбирать наиболее физиологичный режим представления черных символов на белом фоне;

- если работник во время перерыва в работе с ПЭВМ вынужден находиться в непосредственной близости от него (менее 2 метров), то необходимо отключить питание монитора.

В зависимости от категории трудовой деятельности и уровня нагрузки за рабочую смену при работе с ПЭВМ устанавливается суммарное время регламентированных перерывов, приведенное в таблице 3:

Таблица 3 – Суммарное время регламентированных перерывов

| Категория работы с ПЭВМ | Уровень нагрузки за рабочую смену при видах работ с ПЭВМ |                             |                 | Суммарное время регламентированных перерывов при 8-часовой смене, мин. |
|-------------------------|--|-----------------------------|-----------------|--|
|                         | группа А, количество знаков                              | группа Б, количество знаков | группа В, часов |  |
| I                       | до 20 000  | до 15 000                   | до 2            | 50   |
| II                      | до 40 000  | до 30 000                   | до 4            | 70   |
| III                     | до 60 000  | до 40 000                   | до 6            | 90   |

Виды трудовой деятельности с использованием ПЭВМ разделяются на 3 группы:

- группа А – работа по считыванию информации с экрана ВДТ с предварительным запросом;
- группа Б – работа по вводу информации;
- группа В – творческая работа в режиме диалога с ПЭВМ.

Для видов трудовой деятельности устанавливается 3 категории тяжести и напряженности работы с ПЭВМ, которые определяются:

- для группы А – по суммарному числу считываемых знаков за рабочую смену, но не более 60 000 знаков за смену;
- для группы Б – по суммарному числу считываемых или вводимых знаков за рабочую смену, но не более 40 000 знаков за смену;
- для группы В – по суммарному времени непосредственной работы с ПЭВМ за рабочую смену, но не более 6 часов за смену.

## 5.2 Меры безопасности при работе с копировально-множительной техникой

Требования безопасности, направленные на предотвращение неблагоприятного влияния на здоровье человека вредных факторов производственной среды и трудового процесса при работе с копировально-множительной техникой, содержатся в СанПиН 2.2.2.1332-03 «Гигиенические требования к организации работы на копировально-множительной технике», утвержденного Постановлением Главного санитарного врача Российской Федерации от 30.05.2009 г. №107 (в редакции от 07.09.2010 г. №120).

При эксплуатации копировально-множительной техники должны соблюдаться следующие основные меры безопасности:

- в помещениях копировально-множительных работ предусматривается естественное и искусственное освещение;

- не допускается размещение копировально-множительных участков в подвальных помещениях любых типов зданий;

- площадь на одно рабочее место с копировально-множительной техникой должна составлять не менее 6 квадратных метров при объеме не менее 15 кубических метров;

- в светильниках общего и местного освещения помещений с копировально-множительной техникой предусматривается светорассеивающая арматура.

При работе на копировально-множительной технике запрещается:

- производить чистку копировально-множительной техники, не отключив ее от электрической сети;

- опираться на стекло оригиналодержателя, класть на него какие-либо вещи помимо оригинала;

- работать на копировально-множительной технике с треснувшим стеклом оригиналодержателя;

- вынимать застрявшие листы бумаги, не отключив копировально-множительную технику от электрической сети.

Расстановка копировально-множительной техники должна производиться с учётом обеспечения свободного доступа ко всем частям механизмов как для обслуживания, так и для ремонта, при этом:

- расстояние от стены или колонны до краёв машины или аппарата (с учётом конструкции вентсистем) составляет не менее 0,6 м, а со стороны зоны обслуживания – не менее 1,0 м;

- между станками и машинами предусматриваются места для размещения сырья, полуфабрикатов и готовой продукции. Минимальные размеры проходов – не менее 0,6 м.

При применении аппаратов копировально-множительной техники настольного типа, а также единичных стационарных копировально-множительных аппаратов, используемых периодически, для нужд самой

организации, допускается их установка в помещениях, где производятся другие виды работ, с соблюдением требований СанПиН 2.2.2.1332-03.

К работе оператора копировальных и множительных машин не допускаются лица моложе 18 лет, беременные женщины и имеющие медицинские противопоказания.

### 5.3 Требования охраны труда при эксплуатации электрооборудования офисными работниками

Требования безопасности при эксплуатации электрооборудования регламентируются следующими нормативными актами:

- Правилами устройства электроустановок (издание шестое с отдельными разделами и главами в издании седьмом), утвержденными Главтехуправлением, Госэнергонадзором Минэнерго СССР 05.10.1979 г.;

- Правилами технической эксплуатации электроустановок потребителей, утверждёнными Приказом Минэнерго России от 13.01.2003 г. №6;

- Межотраслевыми правилами охраны труда (правилами безопасности) при эксплуатации электроустановок (ПОТ РМ 016-2001), утвержденными Постановлением Минтруда России от 05.01.2001 г. №3.

При эксплуатации офисного электрооборудования необходимо выполнять следующие требования:

- применяемое электрооборудование должно быть заводского изготовления и соответствовать требованиям государственных стандартов и технических условий (что подтверждается в документах завода-изготовителя);

- при эксплуатации и обслуживании электрооборудования необходимо соблюдать требования:

1. настоящей инструкции;
2. паспорта и руководства (инструкции) по эксплуатации электрооборудования, разработанных заводом-изготовителем электрооборудования (при их наличии);

3. Работники при эксплуатации электрооборудования могут производить простейшие операции по его обслуживанию:

4. - подключение и отключение разъемов ПЭВМ и оргтехники (принтеров, факсов, копировальных аппаратов);

5. - установку и удаление бумаги в печатающие и копирующие устройства (в предусмотренные лотки для бумаги);

6. - выемку, установку, замену картриджей в печатающих и копирующих устройствах;

7. - выемку застрявшей бумаги в печатающих и копирующих устройствах;

8. - удаление пыли и загрязнений.

Работы по ремонту электрооборудования должны выполняться специально обученным обслуживающим персоналом (в том числе представителями сторонних организаций);

- электрооборудование, имеющее контакты для подключения заземления, должно быть заземлено, а помещения, где размещаются рабочие места с ПЭВМ (компьютерами), должны быть оборудованы защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации оборудования;

- все крышки и защитные панели должны находиться на своих местах (при отсутствии крышки или защитной панели эксплуатация электрооборудования не допускается);

- при работе с электрооборудованием не допускать попадания влаги на поверхность электрооборудования, а также запрещается работать на электрооборудовании влажными руками;

- вентиляционные отверстия электрооборудования не должны быть перекрыты находящимися вплотную стенами, мебелью, посторонними предметами;



- выдергивание штепсельной вилки электроприбора необходимо осуществлять за корпус штепсельной вилки, при необходимости придерживая другой рукой корпус штепсельной розетки;
- подключение и отключение разъемов компьютеров и оргтехники должно производиться при отключенном питании (за исключением подключения и отключения USB-устройств);
- установка и удаление бумаги осуществляется в лотки (установленные места) печатающих и копирующих устройств;
- выемка, установка, замена картриджей в печатающих и копирующих устройствах, а также выемка застрявшей бумаги должны осуществляться при отключенном электрооборудовании;
- удаление пыли с электрооборудования должно производиться в отключенном от электрической цепи состоянии;
- перед использованием электроприборов необходимо проверить надёжность крепления электророзетки, свериться с номиналом используемого напряжения;
- в помещениях, в которых используется напряжение двух и более номиналов, на всех штепсельных розетках должны быть надписи с указанием номинального напряжения;
- корпуса штепсельных розеток и выключателей не должны содержать трещин, оплавлений и других дефектов, способных снизить защитные свойства или нарушить надёжность контакта;
- недопустимо использовать штепсельные разъёмы в случае существенного нагревания штепсельной розетки или вилки электроприбора при эксплуатации;
- кабели (шнуры) электропитания не должны содержать повреждений изоляции, сильных изгибов и скручиваний;
- неэлектротехническому персоналу, выполняющему работы, при которых может возникнуть опасность поражения электрическим током, присваивается группа I по электробезопасности.

Перечень должностей и профессий, требующих присвоения персоналу I группы по электробезопасности, определяет руководитель Потребителя. Персоналу, усвоившему требования по электробезопасности, относящиеся к его производственной деятельности, присваивается группа I с оформлением в журнале установленной формы; удостоверение не выдается.

Присвоение группы I производится путем проведения инструктажа, который, как правило, должен завершаться проверкой знаний в форме устного опроса и (при необходимости) проверкой приобретенных навыков безопасных способов работы или оказания первой помощи при поражении электрическим током. Присвоение I группы по электробезопасности проводит работник из числа электротехнического персонала данного Потребителя с группой по электробезопасности не ниже III.

Присвоение I группы по электробезопасности проводится с периодичностью не реже 1 раза в год.

## ЗАКЛЮЧЕНИЕ

Данная ВКР заключалась в разработке информационного ресурса в виде Веб-сайта.

Веб-сайт успешно справляется со своими задачами, при минимальных затратах на эксплуатацию.

После завершения работы над программой было еще раз произведено полное тестирование всей программы.

Получившийся в результате всех этих действий, программный продукт (Веб-сайт) удовлетворяет всем предъявленным заказчиком требованиям, полностью работоспособен, не требует ни каких доработок, и готов к эксплуатации.

Экономическая оценка эффективности показывает, что данное приложение экономически выгодно. Срок окупаемости составляет 4 месяца.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Астелс, Дэвид; Миллер Гранвилл; Новак, Мирослав, Практическое руководство по экстремальному программированию, Пер. с англ. - М.: Издательский дом "Вильямс", 2008. - 320 с.: ил. - Парал. тит. Англ
2. Б. Я. Советов, А. И. Водяхо, В. А. Дубенецкий Архитектура информационных систем: учебник для студ. учреждений высш. проф. образования. – М. : Издательский центр «Академия», 2012. – 288 с.
3. Балдин К. В. Информационные системы в экономике [Текст] / К. В. Балдин, В. Б. Уткин. - М.: Академия, 2012. – 288 с.
4. Балдин К. В. Информационные системы в экономике [Текст] / К. В. Балдин, В. Б. Уткин. - М.: Дашков и Ко, 2009. – 396 с.
5. Вигерс Карл, Разработка требований к программному обеспечению, Пер, с англ. - М.:Издательско-торговый дом "Русская Редакция", 2008. - 576с.: ил
6. Гвоздева Т. В., Б. А. Баллод, Проектирование информационных систем, М, Издательство: Феникс, 2009 г., 512 стр.
7. Голицына О. Л., И. И. Попов, Н. В. Максимов, Т. Л. Партыка, Информационные технологии, М, Издательство Инфра-М, 2009 г., 608 стр.
8. Дейв Крейн, Эрик Паскарелло, Даррен Джеймс. AJAX в действии: Учебник – М.: Вильямс, 2007. 450 – 490 с.
9. Дэвид Флэнаган. JavaScript. Подробное руководство: Учебник – М.: Символ Плюс, 2008. 243 – 249 с.
10. Емельянова Н. З., Партыка Т. Л., И. И. Попов, Проектирование информационных систем, М, Издательство: Форум, 2009 г., 432 стр.
11. Емельянова Н. З., Т. Л. Партыка, И. И. Попов, М, Издательство Форум, 2007 г., , 416 стр.

12. Ефимов Е. Н. Информационные системы и технологии в экономике [Текст]: Учебник для вузов / Е. Н. Ефимов, Е. В. Ефимова, Г. М. Лапицкая. – Ростов на Дону: Издательский центр "МарТ", 2012. – 288 с.
13. Избачков Ю. Информационные системы [Текст]: Учебник для вузов / Ю. Избачков, В. Петров, А. Васильев, И. Телина. – СПб.: Питер, 2010. – 544 с.
14. Илющечкин В. М. , Основы использования и проектирования баз данных, М, Издательство Юрайт, 2010 г., 224 стр.
15. Информационные системы в экономике: учебник для вузов / Под ред. Г.А. Титоренко. – М.: ЮНИТИ-ДАНА, 2008. – 463 с.
16. Колесников С.Н. Как организовывать проект внедрения / С.Н. Колесников [Электронный ресурс]. – Режим доступа: <http://citforum.ru/cfin/articles/organize.shtml>
17. Косиненко Н. С. Информационные системы и технологии в экономике [Текст]. / Н. С. Косиненко, И. Г. Фризен. – М.: Дашков и Ко, 2012. – 304 с.
18. Котляров В. П., Т. В. Коликова, Основы тестирования программного обеспечения, Издательства: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2009 г., 288 стр.
19. Кристиан Дари, Богдан Бринзаре, Филип Черчез-Тоза, Михай Бусика. AJAX и PHP. Разработка динамических веб-приложений: Учебник – М.: Символ Плюс, 2007, 289 стр.
20. Кузин А. В., С. В. Левонисова, Базы данных, М, Издательство: Академия, 2008 г., 320 стр.
21. Кузнецов С. Д., Основы баз данных, М, Издательства: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2007 г., 488 стр.
22. Леонтьев В. Новейшая энциклопедия. Компьютер и Интернет 2015. – М.: ОлмаМедиаГрупп, 2014. – 960 с.

23. Мирошников Б. Сетевой фактор. Интернет и общество. – М.: Кучково поле, 2015. – 288 с.
24. Молчанов А. Ю., Системное программное обеспечение, М, Издательство: Питер, 2010 г., 400 стр.
25. Незнанов А. А., Программирование и алгоритмизация, М, Издательство: Академия, 2010 г., 304 стр.
26. Пирогов В. Ю., Информационные системы и базы данных. М, Организация и проектирование, Издательство: БХВ-Петербург, 2009 г. 528 стр.
27. Пирогов В.Ю. Информационные системы и базы данных. Организация и проектирование. – СПб.: БХВ-Петербург, 2009. – 528 с.
28. Ракитов А., Информация, наука, технология в глобальных исторических изменениях. – М.: Директ-Медиа, 2014. - 105 с.
29. Робачевский А. Интернет изнутри. Экосистема глобальной Сети. – М.: Альпина Паблишер, 2015. – 223 с.
30. Романов А.Н., Одинцов Б.Е. Информационные системы в экономике (лекции, упражнения и задачи): Учеб. пособие. – М.: ИНФРА-М, 2012. – 477 с.
31. Скрипкин К. Г. Экономическая эффективность информационных систем / К. Г. Скрипкин. – М.: ДМК Пресс. – 2012. – 256 с.
32. Федеральный закон Российской Федерации от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации»
33. Федорова Г. Н. Информационные системы [Текст] / Г. Н. Федорова. - М.: Академия, 2011. – 208 с.

## ЛИСТИНГ ФАЙЛА INDEX.PHP

```
<!DOCTYPE HTML>
<html>
<head>
<title>База данных по автосалонам города Нижневартовска</title>
<meta charset="utf-8">
<link
href="http://fonts.googleapis.com/css?family=Open+Sans:400,700|Open+Sans+Conde
nsed:700" rel="stylesheet">
<script src="js/jquery-1.8.3.min.js"></script>
<script
src="css/5grid/init.js?use=mobile,desktop,1000px&mobileUI=1&mobileUI.t
heme=none&mobileUI.titleBarOverlaid=1"></script>
<script type="text/javascript" src="js/my_scripts.js"></script>
<noscript>
<link rel="stylesheet" href="css/5grid/core.css">
<link rel="stylesheet" href="css/5grid/core-desktop.css">
<link rel="stylesheet" href="css/5grid/core-1200px.css">
<link rel="stylesheet" href="css/5grid/core-noscript.css">
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/style-desktop.css">
</noscript>
<!--[if lte IE 9]>
<link rel="stylesheet" href="css/ie9.css">
<![endif]-->
<!--[if lte IE 8]>
<link rel="stylesheet" href="css/ie8.css">
<![endif]-->
```

```

<!--[if lte IE 7]>
<link rel="stylesheet" href="css/ie7.css">
<![endif]-->
</head>
<body class="homepage">
<header id="header">
  <div class="logo">
    <div>
      <h1><a>База данных по автосалонам города
Нижевартовска</a></h1>
      <span class="byline"></span> </div>
    </div>
  </header>
<nav id="nav" class="mobileUI-site-nav">
  <ul>
    <li class="current_page_item"><a href="index.php">Главная</a></li>
    <li><a href="">О компании</a></li>
    <li><a href="">Услуги</a></li>
    <li><a href="">Галерея работ</a></li>
    <li><a href="">Оформить заказ</a></li>
  </ul>
</nav>
<div id="main-wrapper">
  <div id="main" class="5grid-layout">
    <div class="row">
      <div class="12u">
        <section class="is-highlight">
          <header>
            <h2>База данных по автосалонам</h2>

```



```
<h2>города Нижневартовска</h2>
```

```
<span class="byline">Бесплатный многоканальный телефон:
```

```
8-800-000-00-00</span></header>
```

```
</section>
```

```
</div>
```

```
</div>
```

```
<div class="row">
```

```
<div class="12u">
```

```
<section class="is-features">
```

```
<h2 class="major"><span>Модельный ряд автомобилей</span></h2>
```

```
<div class="5grid">
```

```
<div class="row">
```

```
<div class="3u">
```

```
<section class="is-feature"> <a href="#" class="image image-  
full"></a>
```

```
<h3><a href="#">KIA</a></h3>
```

```
<?php
```

```
# подключаем библиотеку
```

```
include('simple_dom/simple_html_dom.php');
```

```
// подключаем файл категорий
```

```
include "perser_page/kia.php";
```

```
?>
```

```
</section>
```

```
</div>
```

```
<div class="3u">
```

```
<section class="is-feature"> <a href="#" class="image image-  
full"></a>
```

```

<h3><a href="#">Toyota</a></h3>
    <?php

        // подключаем файл категорий
include "perser_page/toyota.php";
?>

    </section>
</div>
<div class="3u">
    <section class="is-feature"> <a href="#" class="image image-
full"></a>
        <h3><a href="#">Hyundai</a></h3>
    <?php

        // подключаем файл категорий
include "perser_page/hyundai.php";
?>

    </section>
</div>
<div class="3u">
    <section class="is-feature"> <a href="#" class="image image-
full"></a>
        <h3><a href="#">Mitsubishi</a></h3>
    <?php

        // подключаем файл категорий
include "perser_page/mitsubishi.php";
?>

    </section>

```

```
</div>
```

```
<div class="3u">
```

```
<section class="is-feature"> <a href="#" class="image image-  
full"></a>
```

```
<h3><a href="#">Nissan</a></h3>
```

```
<?php
```

```
// подключаем файл категорий
```

```
include "perser_page/nissan.php";
```

```
?>
```

```
</section>
```

```
</div>
```

```
<div class="3u">
```

```
<section class="is-feature"> <a href="#" class="image image-  
full"></a>
```

```
<h3><a href="#">Audi</a></h3>
```

```
<?php
```

```
// подключаем файл категорий
```

```
include "perser_page/audi.php";
```

```
?>
```

```
</section>
```

```
</div>
```

```
<div class="3u">
```

```
<section class="is-feature"> <a href="#" class="image image-  
full"></a>
```

```
<h3><a href="#">Renault</a></h3>
```

```
<?php
```

```
// подключаем файл категорий
include "perser_page/renault.php";
?>

    </section>
</div>
<div class="3u">
    <section class="is-feature"> <a href="#" class="image image-
full"></a>
    <h3><a href="#">Volkswagen</a></h3>
    <?php

// подключаем файл категорий
include "perser_page/volkswagen.php";
?>

    </section>
</div>
<div class="3u">
    <section class="is-feature"> <a href="#" class="image image-
full"></a>
    <h3><a href="#">Mazda</a></h3>
    <?php

// подключаем файл категорий
include "perser_page/mazda.php";
?>

    </section>
</div>
<div class="3u">
```

```

<section class="is-feature"> <a href="#" class="image image-full"></a>
    <h3><a href="#">Suzuki</a></h3>
    <?php

        // подключаем файл категорий
include "perser_page/suzuki.php";
?>

    </section>
</div>
</div>
<footer id="footer" class="5grid-layout">
    <div class="row">
    <div class="12u">
        <section>
            <h2 class="major"><span>Мы в соцсетях</span></h2>
            <ul class="contact">
                <li><a href="#" class="facebook">Facebook</a></li>
                <li><a href="#" class="twitter">Twitter</a></li>
                <li><a href="#" class="rss">RSS</a></li>
                <li><a href="#" class="dribbble">Dribbble</a></li>
                <li><a href="#" class="linkedin">LinkedIn</a></li>
                <li><a href="#" class="googleplus">Google+</a></li>
            </ul>
        </section>
    </div>
</div>
<div class="row">

```

<div id="copyright"> &copy; 2016 База данных по автосалонам города  
Нижневартовска </div>

</div>

</footer>

</body>

</html>

## ЛИСТИНГ СТИЛЕЙ

```
/* Basic */
/*****
*****/

*
{
    -moz-box-sizing: border-box;
    -webkit-box-sizing: border-box;
    -o-box-sizing: border-box;
    -ms-box-sizing: border-box;
    box-sizing: border-box;
}

body
{
    font-family: 'Open Sans', sans-serif;
    font-weight: 400;
    background: #BEBEBE;
    color: #6b7770;
    font-size: 11.5pt;
    line-height: 2em;
}

h1, h2, h3, h4, h5, h6, h7
{
    font-family: 'Open Sans Condensed', sans-serif;
    font-weight: 700;
    text-transform: uppercase;
```

```
color: #6b7770;
}

h1 a, h2 a, h3 a, h4 a, h5 a, h6 a, h7 a
{
    text-decoration: none;
    color: inherit;
}

h2.major
{
    position: relative;
    border-top: solid 5px #e7eae8;
}

h2.major span
{
    background: #fff;
    position: relative;
    display: inline-block;
}

#footer h2.major span
{
    background: #f2f5f3;
}
```



```
a
{
    color: #006376;
    text-decoration: underline;
}

a:hover
{
    text-decoration: none;
}

form
{
}

form input,
form select,
form textarea
{
    -webkit-appearance: none;
}

br.clear
{
    clear: both;
}

p, ul, ol, dl, table
{
```

```
        margin-bottom: 3em;
    }

    section,
    article
    {
        margin-bottom: 6em;
    }

    section > :last-child,
    article > :last-child
    {
        margin-bottom: 0;
    }

    section:last-child,
    article:last-child
    {
        margin-bottom: 0;
    }

    .byline
    {
        display: block;
        font-family: 'Open Sans Condensed', sans-serif;
        font-weight: 700;
        text-transform: uppercase;
        color: #c1cac5;
    }
```

```

.image
{
    position: relative;
    display: inline-block;
    border-radius: 8px;
}

.image img
{
    display: block;
    width: 100%;
    border-radius: 8px;
}

```

/\*  
 Note: The following style adds the textured/pixelated look to  
 content  
 images. I added this to help mask the distortion introduced  
 when a browser  
 resizes an image, but if you don't want this to happen just  
 remove or comment  
 out the entire style.  
 \*/

```

.image:before
{
    content: "";
    position: absolute;
    left: 0;

```

```
top: 0;

        width: 100%;
        height: 100%;
        background: url('images/overlay.png');
        opacity: 0.2;
        border-radius: 8px;
    }
```

```
.image-full
{
    display: block;
    width: 100%;
    margin: 0 0 2em 0;
}
```

```
.image-left
{
    float: left;
    margin: 0 2em 2em 0;
}
```

```
.image-centered
{
    display: block;
    margin: 0 0 2em 0;
}
```

```
.image-centered img
{
```

```
margin: 0 auto;
    width: auto;
}

.button
{
    display: inline-block;
    font-family: 'Open Sans Condensed', sans-serif;
    font-weight: 700;
    text-transform: uppercase;
    text-decoration: none;
    background: #e3daa8;
    color: #fff;
    border-radius: 8px;
    outline: 0;
    -moz-transition: background-color .2s ease-in-out;
    -webkit-transition: background-color .2s ease-in-out;
    -o-transition: background-color .2s ease-in-out;
    -ms-transition: background-color .2s ease-in-out;
    transition: background-color .2s ease-in-out;
}

.button:hover
{
    background: #ebe3b2;
}

.button-alt
{
```

```
background: #c1cac5;
    }

    .button-alt:hover
    {
        background: #ccd5d0;
    }

.actions
{
}

ul.style1
{
    list-style: disc;
    padding-left: 1.5em;
}

ul.style2
{
}

ul.style2 > li
{
    border-top: solid 2px #e7eae8;
    padding-top: 1.75em;
    margin-top: 1.75em;
}
```

```
ul.style2 > li:first-child
```

```
{  
    border-top: 0;  
    padding-top: 0;  
    margin-top: 0;  
}
```

```
ul.contact
```

```
{  
}
```

```
ul.contact li
```

```
{  
    position: relative;  
    display: inline-block;  
    margin: 0 0.25em 0 0.25em;  
    top: 0;  
    -moz-transition: top .15s ease-in-out;  
    -webkit-transition: top .15s ease-in-out;  
    -o-transition: top .15s ease-in-out;  
    -ms-transition: top .15s ease-in-out;  
    transition: top .15s ease-in-out;  
}
```

```
ul.contact li a
```

```
{  
    display: block;  
    position: relative;  
    background: #c1cac5 url('images/icons-contact.png');
```

```

width: 48px;
height: 48px;
text-indent: -9999em;
border-radius: 6px;
outline: 0;
-moz-transition: background-color .2s ease-in-out;
-webkit-transition: background-color .2s ease-in-out;
-o-transition: background-color .2s ease-in-out;
-ms-transition: background-color .2s ease-in-out;
transition: background-color .2s ease-in-out;
}

ul.contact li a.facebook {
background-position: 0px 0px; }
ul.contact li:hover a.facebook {
background-color: #3C5A98; }
ul.contact li a.twitter {
background-position: -48px 0px; }
ul.contact li:hover a.twitter {
background-color: #2DAAE4; }
ul.contact li a.rss {
background-position: -96px 0px; }
ul.contact li:hover a.rss {
background-color: #F2600B; }
ul.contact li a.dribbble {
background-position: -144px 0px; }
ul.contact li:hover a.dribbble {
background-color: #C4376B; }

```



```
ul.contact li a.linkedin {
background-position: -192px 0px; }
ul.contact li:hover a.linkedin {
background-color: #006599; }
ul.contact li a.tumblr {
background-position: -240px 0px; }
ul.contact li:hover a.tumblr {
background-color: #31516A; }
ul.contact li a.googleplus {
background-position: -288px 0px; }
ul.contact li:hover a.googleplus { background-
color: #DA2713; }

ul.contact li:hover
{
top: -5px;
}

ul.contact li:active
{
top: 0;
}

ul.special
{
}

ul.special li
{
```

```
position: relative;
    display: inline-block;
    margin: 0 1.5em 0 1.5em;
    top: 0;
    -moz-transition: top .15s ease-in-out;
    -webkit-transition: top .15s ease-in-out;
    -o-transition: top .15s ease-in-out;
    -ms-transition: top .15s ease-in-out;
    transition: top .15s ease-in-out;
}

ul.special li:hover
{
    top: -10px;
}

ul.special li:hover a
{
    background: #b9d2ce;
}

ul.special a
{
    display: block;
    position: relative;
    background: #c1cac5;
    width: 12em;
    height: 12em;
    border-radius: 6em;
```

```
text-indent: -9999em;

outline: 0;
-moz-transition: background-color .2s ease-in-out;
-webkit-transition: background-color .2s ease-in-out;
-o-transition: background-color .2s ease-in-out;
-ms-transition: background-color .2s ease-in-out;
transition: background-color .2s ease-in-out;
}

ul.special a:before
{
    content: "";
    position: absolute;
    width: 128px;
    height: 128px;
    left: 50%;
    top: 50%;
    margin-top: -64px;
    margin-left: -64px;
    background: url('images/icons-special.png');
}

ul.special a.battery:before
{
    background-position: 0px 0px;
}

ul.special a.tablet:before
{
```

```
background-position: -128px 0px;
    }

    ul.special a.flask:before
    {
        background-position: -256px 0px;
    }

    ul.special a.chart:before
    {
        background-position: -384px 0px;
    }

ul.meta
{
    font-size: 0.85em;
    color: #a1aaa5;
}

ul.meta li
{
    position: relative;
    display: inline-block;
    padding-left: 24px;
    margin-left: 1em;
}

ul.meta li a
{
```

```
color: inherit;
    outline: 0;
}

ul.meta li:first-child
{
    margin-left: 0;
}

ul.meta li:before
{
    content: ";
    display: block;
    position: absolute;
    width: 16px;
    height: 16px;
    left: 0;
    top: 50%;
    margin-top: -8px;
    background: url('images/icons-meta.png');
}

ul.meta li.timestamp:before
{
    background-position: 0px 0px;
}

ul.meta li.comments:before
{
```

```
background-position: -16px 0px;  
    }
```

```
/*  
*****  
*****/  
/* Section + Article Types */  
/*  
*****  
*****/
```

```
.is-highlight  
{  
    text-align: center;  
}
```

```
.is-features  
{  
}
```

```
.is-feature  
{  
}
```

```
.is-blog  
{  
}
```

```
.is-post  
{  
}
```

```
background-position: -16px 0px;  
    }
```

```
/*  
*****  
******/
```

```
/* Section + Article Types */
```

```
/*  
*****  
******/
```

```
.is-highlight  
{  
    text-align: center;  
}
```

```
.is-features  
{  
}
```

```
.is-feature  
{  
}
```

```
.is-blog  
{  
}
```

```
.is-post  
{  
}
```

```
.is-post-summary
```

```
{  
}
```

```
.is-page-content
```

```
{  
}
```

```
/*  
*****  
*****/  
*/
```

```
/* Sidebar + Content */
```

```
/*  
*****  
*****/  
*/
```

```
.sidebar
```

```
{  
}
```

```
.content
```

```
{  
}
```

```
/*  
*****  
*****/  
*/
```

```
/* Wrappers */
```

```
/*  
*****  
*****/  
*/
```

```
#banner-wrapper
```



```
{  
    position: relative;  
    background: url('../images/banner.jpg') center center no-repeat;  
    background-size: cover;  
    z-index: 1;  
}
```

```
/*
```

Note: The following style adds the textured/pixelated look to the banner

image. I added this to help mask the distortion introduced when a browser

resizes an image, but if you don't want this to happen just remove or comment

out the entire style.

```
*/
```

```
#banner-wrapper:before
```

```
{
```

```
    content: "";
```

```
    position: absolute;
```

```
    left: 0;
```

```
    top: 0;
```

```
    width: 100%;
```

```
    height: 100%;
```

```
    background: url('images/overlay.png');
```

```
    opacity: 0.175;
```

```
}
```

```
#main-wrapper
{
    background: #fff;
    border-top: solid 5px #e3daa8;
    border-bottom: solid 5px #e1cac5;
}

/*****
*****/

/* Header */
/*****
*****/

#header
{
}

/*****
*****/

/* Nav */
/*****
*****/

#nav
{
}

/*****
*****/
```

```

/* Galereya */
/*****
*****/
span.reference{
    font-family:Arial;
    display:block;
    font-size:12px;
    text-align:center;
    margin-bottom:10px;
}
span.reference a{
    color:#000;
    text-transform:uppercase;
    text-decoration:none;
    margin:0px 20px;
}
span.reference a:hover{
    color:#ddd;
}
/* Booklet jQuery Plugin Style*/
.booklet{
    -moz-box-shadow:0px 0px 1px #fff;
    -webkit-box-shadow:0px 0px 1px #fff;
    box-shadow:0px 0px 1px #fff;
    -moz-border-radius:10px;
    -webkit-border-radius:10px;
    border-radius:10px;
}
.booklet .b-wrap-left {

```

```
background:#fff url(../images/left_bg.jpg) no-repeat top left;
    -webkit-border-top-left-radius: 10px;
    -webkit-border-bottom-left-radius: 10px;
    -moz-border-radius-topleft:10px;
    -moz-border-radius-bottomleft: 10px;
    border-top-left-radius: 10px;
    border-bottom-left-radius: 10px;
}

.booklet .b-wrap-right {
    background:#efefef url(../images/right_bg.jpg) no-repeat top left;
    -webkit-border-top-right-radius: 10px;
    -webkit-border-bottom-right-radius: 10px;
    -moz-border-radius-topright: 10px;
    -moz-border-radius-bottomright: 10px;
    border-top-right-radius: 10px;
    border-bottom-right-radius: 10px;
}

.booklet .b-counter {
    bottom:10px;
    position:absolute;
    display:block;
    width:90%;
    height:20px;
    border-top:1px solid #ddd;
    color:#222;
    text-align:center;
    font-size:12px;
    padding:5px 0 0;
    background:transparent;
```

```
-moz-box-shadow:0px -1px 1px #fff;
    -webkit-box-shadow:0px -1px 1px #fff;
    box-shadow:0px -1px 1px #fff;
    opacity:0.8;
}
.book_wrapper{
    margin:0 auto;
    padding-top:50px;
    width:905px;
    height:540px;
    position:relative;
    background:transparent url(../images/bg.png) no-repeat 9px 27px;
}
.book_wrapper h1{
    color:#13386a;
    margin:5px 5px 5px 15px;
    font-size:26px;
    background:transparent url(../images/h1.png) no-repeat bottom left;
    padding-bottom:7px;
}
.book_wrapper p{
    font-size:16px;
    margin:5px 5px 5px 15px;
}
.book_wrapper a.article,
.book_wrapper a.demo{
    background:transparent url(../images/circle.png) no-repeat 50% 0px;
    display:block;
    width:95px;
```

```
height:41px;
    text-decoration:none;
    outline:none;
    font-size:16px;
    color:#555;
    float:left;
    line-height:41px;
    padding-left:47px;
}
.book_wrapper a.demo{
    margin-left:50px;
}
.book_wrapper a.article:hover,
.book_wrapper a.demo:hover{
    background-position:50% -41px;
    color:#13386a;
}
.book_wrapper img{
    margin:10px 0px 5px 35px;
    width:300px;
    padding:4px;
    border:1px solid #ddd;
    -moz-box-shadow:1px 1px 1px #fff;
    -webkit-box-shadow:1px 1px 1px #fff;
    box-shadow:1px 1px 1px #fff;
}
.booklet .b-wrap-right img{
    border:1px solid #E6E3C2;
}
```

```
a#next_page_button,  
a#prev_page_button{  
    display:none;  
    position:absolute;  
    width:41px;  
    height:40px;  
    cursor:pointer;  
    margin-top:-20px;  
    top:50%;  
    background:transparent url(../images/buttons.png) no-repeat 0px -40px;  
}  
a#prev_page_button{  
    left:-30px;  
}  
a#next_page_button{  
    right:-30px;  
    background-position:-41px -40px;  
}  
a#next_page_button:hover{  
    background-position:-41px 0px;  
}  
a#prev_page_button:hover{  
    background-position:0px 0px;  
}  
.loading{  
    width:160px;  
    height:56px;  
    position: absolute;  
    top:50%;
```

```

margin-top:-28px;
    right:135px;
    line-height:56px;
    color:#fff;
    padding-left:60px;
    font-size:15px;
    background: #000 url(../images/ajax-loader.gif) no-repeat 10px 50%;
    opacity: 0.7;
    z-index:9999;
    -moz-border-radius:20px;
    -webkit-border-radius:20px;
    border-radius:20px;
    filter:progid:DXImageTransform.Microsoft.Alpha(opacity=70);
}

```

```

/* Banner */
/*****
*****/

```

```

#banner
{
    position: relative;
    background: rgba(21, 28, 23, 0.95);
    text-align: center;
    z-index: 2;
}

```



```
#banner:before
{
    content: "";
    position: absolute;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background: url('images/highlight.png') center center;
    background-size: 100% 200%;
}

#banner:after
{
    content: "";
    position: absolute;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background: url('images/overlay.png');
    opacity: 0.25;
}

#banner > *
{
    position: relative;
    z-index: 3;
}
```

```
#banner h2
{
    color: #fff;
}

#banner .byline
{
    color: #575d59;
}

#banner .button
{
    color: #b4b7b5;
    background: transparent !important;
    border: solid 4px rgba(87, 93, 89, 0.35);
    border-radius: 16px;
    -moz-transition: border-color .2s ease-in-out, color .2s ease-
in-out;
    -webkit-transition: border-color .2s ease-in-out, color .2s
ease-in-out;
    -o-transition: border-color .2s ease-in-out, color .2s ease-in-
out;
    -ms-transition: border-color .2s ease-in-out, color .2s ease-in-
out;
    transition: border-color .2s ease-in-out, color .2s ease-in-out;
}

#banner .button:hover
{
```

```
color: #e4e7e5;
        border-color: rgba(87, 93, 89, 0.75);
    }

    #banner .button:active
    {
        color: #fff;
        border-color: rgba(97, 103, 99, 1.0);
    }
```

```

    /*****
    *****/

    /* Main */
    /*****
    *****/

    #main
    {
    }

    /*****
    *****/

    /* Footer */
    /*****
    *****/

    #footer
    {
    }
```

```
#footer a
{
    color: inherit;
}

/*****
*****/

/* Copyright */
/*****
*****/

#copyright
{
    display: inline-block;
    background: #e7eae8;
    padding: 0.75em 2em 0.75em 2em;
    border-radius: 8px;
    font-size: 0.85em;
    color: #8b9790;
}

#copyright a
{
    color: inherit;
}
```

## ГЛАВНАЯ СТРАНИЦА САЙТА



БАЗА ДАННЫХ ПО АВТОСАЛОНАМ ГОРОДА НИЖНЕВАРТОВСКА

## БАЗА ДАННЫХ ПО АВТОСАЛОНАМ ГОРОДА НИЖНЕВАРТОВСКА

БЕСПЛАТНЫЙ МНОГОКАНАЛЬНЫЙ ТЕЛЕФОН: 8-800-010-03-00



### МОДЕЛЬНЫЙ РЯД АВТОМОБИЛЕЙ



**KIA**

Автосалон «Автомобили»

г. Нижний Новгород

т/л 477 100 руб/мес

т/л 5 000 руб/мес

т/л

т/л 741 000 руб/мес

т/л 0 000 руб/мес

т/л

т/л 001 000 руб/мес



**TOYOTA**

**TOYOTA**

Полный сервис

Салон

от 001 000 руб

Салон

от 1 000 000 руб

Салон



**HYUNDAI**

**HYUNDAI**

Салон

от 3 400 000 руб

Салон

от 2 500 000 руб

Салон

от 1 000 000 руб



**MITSUBISHI**

MITSUBISHI MOTORS

от 1 000 000 руб до 10 000 000 руб

от 10 000 руб

Бесплатная доставка