**Федеральное агентство связи**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Сибирский государственный университет телекоммуникаций и информатики»**

**(СибГУТИ)**

Новосибирск 2020 г.

Новосибирск 2020 г.

Кафедра ПМиК

Допустить к защите

зав. кафедрой: проф., д.т.н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фионов А.Н.

### **ВЫПУСКНАЯ**

### **КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

Разработка интернет сайта по выбору и продаже автомобилей

Пояснительная записка

Студент Соколов Кирилл Андреевич / /

Факультет ИВТ Группа ИП-613

Руководитель Перцев Игорь Владимирович / ……………../

**Содержание**

[**Введение 3**](#_Toc42783617)

[**1 Постановка задачи 5**](#_Toc42783618)

[1.1 Характеристика предметной области 5](#_Toc42783619)

[1.2 Формулировка задачи 7](#_Toc42783620)

[**2 Используемые средства разработки и ПО 8**](#_Toc42783621)

[2.1 HTML – язык гипертекстовой разметки 8](#_Toc42783622)

[2.3 CSS – язык каскадных таблиц стилей 13](#_Toc42783623)

[2.4 JavaScript (JQuery, Ajax) – язык программирования 14](#_Toc42783624)

[2.5 MySQL – система управления базами данными 16](#_Toc42783625)

[2.6 phpDesigner 8 – редактор кода 18](#_Toc42783626)

[2.7 Denwer (web-сервер Apache) 19](#_Toc42783627)

[**3 Анализ, проектирование и реализация сайта 20**](#_Toc42783628)

[3.1 Обзор и анализ методов проектирования сайта 20](#_Toc42783629)

[3.2 Проектирование сайта 22](#_Toc42783630)

[3.3 Принципы реализации элементов дизайна сайта 24](#_Toc42783631)

[3.4 Даталогическая модель системы 28](#_Toc42783632)

[3.5 Разработка структуры Web-сайта 31](#_Toc42783633)

[3.6 Проектирование и разработка Web-сайта 32](#_Toc42783634)

[3.7 Проектирование и разработка панели администратора Web-сайта 38](#_Toc42783635)

[**4 Описание разработанного сайта 42**](#_Toc42783636)

[4.1 Описание интерфейса пользователя 42](#_Toc42783637)

[4.2 Описание интерфейса администратора 49](#_Toc42783638)

[**Заключение 52**](#_Toc42783639)

[**Список использованных источников 54**](#_Toc42783640)

# 

# Введение

Сегодня тяжело представить компанию без сайта в сети интернет. Как это бывает, представительством компании в интернете становится интернет сайт. C его помощью компания получает новый канал привлечения клиентов к своим товарам или услугам.

В роли заказчика мы возьмем компанию “AutoSmart” (Вымышленное название). Данная компания была заинтересована в разработке сайта «разработке интернет сайта по выбору и продаже автомобилей».

Компании уже занимается много лет продажей автомобилей. Но со временем им стало поступать много заявок на различные автомобили (Легковые, Грузовые, Пассажирские) и жалобы клиентов в связи с отсутствием возможности заказа и покупки автомобиля через интернет сайт, и компания приняла решение разработать интернет сайт для удобства своих клиентов и привлечения новых, чтобы они могли выбирать автомобиль и оставлять заказ на его покупку, не выходя из дома.

Наличие своего собственного интернет сайта для продажи автомобилей онлайн позволит сократить временные расходы на обработку заказов, увеличить кол-во проданных автомобилей, посредством привлечения новых клиентов в сети интернет, а также получить новый канал связи с клиентом.

Главными пожеланиями заказчика стали простота использования интернет сайта как для администратора, так и пользователя, а также его запоминающийся дизайн.

Главной целью моего дипломного проекта является: ''создание интернет сайта по выбору и продаже автомобилей'' компании “AutoSmart”, а также облегчение связи с потенциальными клиентами за счет обратной связи, формирование положительного имиджа компании, как компании использующей современные технологии.

Концепцией дипломной работы является разработка интернет сайта, содержащего информацию о компании, ее автомобилях, видах, ценах, контактах.

Интернет-составляющая автомобильного бизнеса - не получает существенного дохода, а лишь помогает главному неэлектронному бизнесу расти и процветать. Компания получает очень большую экономию времени за счет находящейся информации на сайте. Сокращаются расходы на телефонные разговоры, диспетчерскую службу, уменьшается время на обработку заказов автомобилей и т.д.

Темой моей дипломной работы является «Разработка интернет сайта по выбору и продаже автомобилей».

Для реализации дипломного проекта нужно выполнить ряд шагов:

1. Изучить теоретические аспекты создания интернет-сайтов.

2. Провести анализ существующего ПО для разработки интернет-сайтов.

3. Рассмотреть языки программирования, используемые в разработке.

4. Провести анализ средств для создания базы данных для интернет-сайта.

5. Определить структуру и пользовательский интерфейс сайта.

6. Создать базу данных для интернет-сайта.

7. Провести анализ корректности работы созданного сайта.

Дипломная работа состоит из **четырех частей**.

**Первая часть** включает в себя характеристику предметной области и формулировку задачи.

**Вторая часть** включает в себя анализ и выбор программного обеспечения для реализации поставленной задачи по созданию интернет-сайта по выбору и продаже автомобилей, для имиджа компании “AutoSmart” в сети интернет.

**Третья часть** включает анализ, проектирование и разработку интернет-сайта.

**Четвертая часть** включает в себя описание разработанного сайта и его тестирование.

# Постановка задачи

# Характеристика предметной области

В конце прошлого столетия (90-x годах), когда Интернет появился в России, бизнесмены начали интересоваться о ведении бизнеса в сети интернет, так как данный вид бизнеса имел много преимуществ по сравнению с ведением бизнеса вне сети. Но в России на тот момент не было условий для развития онлайн торговли из-за низкого уровня жизни населения после развала Советского Союза и небольшого кол-ва пользователей интернет. Однако первые интернет-магазины все-таки сумели сформировать свой рынок сбыта, который динамично развивался и продолжает развиваться по сей день.

По мере роста Интернет аудитории популярность интернет-магазинов начинает стремительно расти в бизнес кругах, все чаще начинает возникать потребность в присутствии бизнеса в сети, и обеспечить это присутствие созданием интернет-магазина.

Первые покупатели интернет-магазинов получили возможность приобретать компьютерную технику и комплектующие. А еще через несколько лет в ассортимент торговли Интернет-магазинов значительно вырос и в него вошли аудио- и видео- бытовая техника, мобильные телефоны, товары повседневного спроса и даже продукты питания.

Хотя можно было наблюдать рост потребительского спроса в интернет-покупках, в 90-е – 2000-е годы на рынке стали появляется мелкие фирмы, которые своими часто недобросовестными методами ведения бизнеса, испортили имидж интернет-магазинов, чем вызвали большой отток покупателей. Имидж Интернет-торговли сильно пострадал в годы становления рынка.

В настоящее время доверие населения к покупке товаров в Интернете приходит в порядок и у экспертов есть мнение, что чем больше появляется успешных проектов и реализаций интернет-магазинов, тем быстрее закончится процесс восстановления утраченного доверия к покупкам и торговле в сети Интернет.

Интернет-магазины могут запросто конкурировать с магазинами розничной и даже оптовой торговли из-за отсутствия необходимости держать большое кол-во обслуживающего персонала и торгового зала. Не смотря на кризис перспективы развития интернет-рынка достаточно хороши. Кроме этого, многие пользователи и покупатели, говорят о том, что им удобней совершать покупки в Интернет-магазинах благодаря возможности делать покупки в удобное для них время и любом удобном месте, в том числе с экрана мобильного телефона или персонального компьютера.

Как утверждает статистика, за последние несколько лет электронная коммерция в нашей стране вышла на более высший уровень. Значительно возросло количество пользователей Интернета, открылись новые, очень разные по своей тематике интернет-магазины, а многие традиционные магазины поспешили продублировать свой бизнес в Сети, ибо это помогает привлечь новую аудиторию к своему продукту или товару. Однако мало кто объективно представляет себе, каким он должен быть, как обеспечить его наибольшую прибыльность и как вести учет доходов и расходов, понесенных или полученных при его создании и обслуживании.

Но хороший интернет-магазин – это не только отлично структурированный и достаточно полно описанный ассортимент, не только высокое качество обслуживающего персонала его клиентов, это еще и привлекательное внешнее оформление, дизайн, от удобной навигации и поисковой системы до интуитивно-понятной системы заказов, это множество мелочей и особенностей, которые помогут клиенту сделать покупку, это гарантия того, что клиент обязательно вернется этот интернет-магазин снова.

Совершение сделок/покупок через сеть Интернет в первую очередь требует правового регулирования передачи и обмена данными. В настоящее время в нашей стране активно продвигается, не без помощи государства, внедрение электронного документооборота с использованием электронной подписи.

На сегодня существует четыре основных типа интернет-магазина:

– Заказ товара по электронной почте. Товары отличаются крайне низкой ценой и высокой скоростью изготовления. Для начала создается каталог товаров в виде простой таблицы, которая рассылается клиентам, готовым сделать заказ. После чего клиент в ответе на сообщение высылает запрос на товар и его количество.

– Заказ товара происходит через корзину на сайте, однако сам заказ попадает менеджеру в почту. После этого он обрабатывается.

– Магазин с обработкой заказа через интернет-сайт. Вся информация хранится в базе данных на сервере, так что у продавца есть возможность отслеживать активность покупателя и управлять заказами с помощью панели администратора.

– Полнофункциональный интернет-магазин с возможностью оплаты с помощью электронных платежей. Встроенная система общения покупателя и продавца, системы скидок, бонусов, рассчитывающихся автоматически от количества или стоимости заказа.

Преимуществами интернет-магазина являются:

1. Уменьшение численности персонала за счет сокращения объема взаимодействия с клиентами;

2. Размещение интернет-магазина на хостинге во много раз дешевле и проще аренды торговых помещений и размещения товаров на полках, затраты на услуги маркетологов и мерчендайзеров;

3. Отсутствие нужды в кассовом обслуживании, все операции идут через сторонние ресурсы для оплаты товаров;

4. Возможность использования интернет-магазина как эффективного способа маркетингового исследования (учитывая стоимость этой услуги);

5. получение возможности сделать покупку, в любой точке планеты где есть доступ в сеть Интернет в любое время, а выбор и заказ товара займет у клиента несколько минут, если он точно знает, что хочет купить;

6. отсутствие ограничений на ассортимент;

7. снижение стоимости товаров за счет отсутствия арендных выплат.

# Формулировка задачи

Трудоемкость разработки интернет – сайтов на начальных этапах программирования оценивалась значительно ниже реально затрачиваемых усилий, что служило причиной дополнительных расходов и затягивания окончательных сроков готовности сайта, а также приводило к увеличению размеров программ и необходимости привлечения большего числа программистов, что потребовало дополнительных ресурсов для организации их согласованной работы.

Поэтому прежде чем решить эти проблемы и приступить к разработке системы необходимо иметь четкое описание разработки, адаптированной к конкретному проекту.

**Основания для разработки**

Основанием для разработки программного продукта служит задание по разработке интернет-сайта по выбору и продаже автомобилей, утвержденное на основании приказа.

**Назначение разработки**

Моя задача сделать специализированный интернет-сайт по выбору и продаже автомобилей, осуществляющий прямое подключение к БД (база данных имеет заданную структуру и разработана в среде MySql), а также реализующий интерфейс и некоторые функции, позволяющие пользователю провести выбор интересующего клиента автомобиля/ей, а также для привлечения новых клиентов в компанию.

Перечень требований пользователя к программному изделию

Сайт должен удовлетворять следующим требованиям пользователя:

1. подключение к базе данных;

2. добавление новых автомобилей в базу данных;

3. показывать заказанные автомобили;

4. возможность просмотра таблиц БД и их редактирование администратором;

5. каталог автомобилей;

6. настройка учетных записей пользователя;

7. поиск по базе данных.

Редактирование данных производиться в специально предназначенных для этого страницах веб-сайта, а именно в панели администратора.

**Информационная совместимость**

В качестве входных данных программа должна использовать файлы формата \*.sql, а также информацию, введенную вручную.

Выходными данными должны быть файлы формата \*.sql, а также информация на экране.

**Программные ограничения, совместимость**

База данных для модуля должна быть написана с помощью системы управления базами данных MySQL, а также работать под управлением операционной систем Windows 10/8/7/ XP/Vista.

# Используемые средства разработки и ПО

# HTML – язык гипертекстовой разметки

HTML (англ. Hype.rText Mar.kup Languag.e — язык гипертекстовой разметки) –основной язык, на котором создаются веб-страницы. Страница, которую пользователь видит в своем браузере, может состоять из множества различных файлов - изображений, видео роликов, сценариев Ja.vaScript, апплетов и т.д. – но основой страницы практически всегда является документ HTML. Другие языки разметки (прежде всего многочисленные приложения XML) пока еще слабо поддерживаются браузерами и поэтому не вытесняют HTML на компьютерах обычных пользователей. Взрывной рост Всемирной Паутины в начале 1990-х во многом обусловлен широким распространением этого языка и браузеров, поддерживающих написанные на нем страницы.

**HTML документ**

HTML следует правилам, которые содержатся в файле объявления типа документа (Do.cument Typ.e D.efinition, или DTD). DTD представляет собой XML-документ, определяющий, какие теги, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

DOCTYPE отвечает за корректное отображение веб-страницы браузером. DOCTYPE определяет не только версию HTML (например, h.tml), но и соответствующий DTD-файл в Интернете.

Элементы, находящиеся внутри тега <htm.l>, образуют дерево документа, так называемую объектную модель документа, DOM (do.cument obje.ct .m.odel). При этом элемент <ht.ml> является корневым элементом.

**Структура HTML элементов**

Чтобы разобраться во взаимодействии элементов веб-страницы, необходимо рассмотреть так называемые «родственные отношения» между элементами. Отношения между множественными вложенными элементами подразделяются на родительские, дочерние и сестринские.

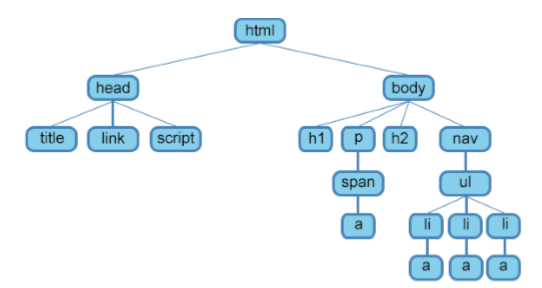


Рисунок 2.1.

**1. Предок** – элемент, который заключает в себе другие элементы. На рисунке 1 предком для всех элементов является <htm.l>. В то же время элемент <b.ody> является предком для всех содержащихся в нем тегов: <h1>, <p>, <s.pan>, <na.v> и т.д.

**2. Потомок** – элемент, расположенный внутри одного или более типов элементов. Например, <body> является потомком <html>, а элемент <p> является потомком одновременно для <body> и <ht.ml>.

**3. Родительский элемент** – элемент, связанный с другими элементами более низкого уровня, и находящийся на дереве выше их. На рисунке 1 <h.tml> является родительским только для <head> и <body>. Тег <p> является родительским только для <span>.

**4. Дочерний элемент** – элемент, непосредственно подчиненный другому элементу более высокого уровня. На рисунке 1 только элементы <h1>, <h2>, <p> и <nav> являются дочерними по отношению к <body>.

**5. Сестринский элемент** – элемент, имеющий общий родительский элемент с рассматриваемым, так называемые элементы одного уровня. На рисунке 1 <head> и <body> – элементы одного уровня, так же как и элементы <h1>, <h2> и <p> являются между собой сестринскими.

**Отображение документа**

В разных браузерах HTML-документ может отображаться по-разному. HTML тэги определяют только общий стиль отображения документа. Что означает, например, что HTML рассматривает заголовок первого уровня только как заголовок первого уровня, но не определяет, что заголовок первого уровня должен отображаться шрифтом Times размера 24 в центре верхней части страницы. Преимущество такого подхода заключается в том, что если пользователь решит изменить заголовок первого уровня на шрифт Helvetica размера 20 выравненный по левому краю, все, что ему надо сделать, это изменить определение заголовка первого уровня в его браузере. Таким образом, конкретное представление документа на экране монитора определяет пользователь конкретного клиента.

При разработке сайта важно помнить:

- Браузеры разных фирм иногда по-разному отображают встречающиеся в HTML-документе теги; есть возможности "за пределами стандарта", поддерживаемые только браузерами конкретной фирмы. При их использовании иной раз приходится для каждого браузера создавать свою версию сайта

- Старые версии браузеров не корректно отображают теги, соответствующих новым стандартам.

- В случае, когда в HTML-документе встречаются теги, неизвестные браузеру, он их просто игнорирует без всяких сообщений об ошибках.

Сайт приходится тестировать в браузерах разных производителей и разных версий, при разных разрешениях экрана, добиваясь должного результата.

* 1. **PHP – язык программирования**

PHP (англ. Hypertext Preprocessor - Препроцессор Гипертекста) – язык сценариев общего назначения с открытым исходным кодом.

PHP - язык программирования, специально разработанный для написания web-приложений (скриптов, сценариев), исполняющихся на Web-сервере. Синтаксис языка во многом основывается на синтаксисе C, Java и Perl. Он очень похож на С и на Perl, поэтому для профессионального программиста не составит труда его изучить. С другой стороны, язык PHP проще, чем C, и его может освоить веб-мастер, не знающий пока других языков программирования.

Большим достоинством языка программирования PHP является то, что PHP-скрипты выполняются на стороне сервера. PHP не зависит от скорости компьютера пользователя или его браузера, он полностью работает на сервере. Пользователь же получает HTML-файл как результат выполнения скрипта.

Сценарии PHP могут быть выделены в отдельные php файлы или интегрироваться в HTML документ.

PHP может генерировать и преобразовывать не только HTML документы, но и файлы изображений различных форматов. PHP может формировать данные в любом текстовом формате, включая XHTML и XML.

PHP – кроссплатформенный продукт. Дистрибутив PHP доступен для всех современных операционных систем. PHP поддерживается на большинстве вебсерверов, таких, как Apache, Microsoft Internet Information Server (IIS), Microsoft Personal Web Server и других.

PHP поставляется в 2-х вариантах - в качестве CGI препроцессора и в качестве отдельного модуля.

Язык и его интерпретатор создан группой энтузиастов в рамках проекта OpenSource. Лицензируется PHP своей, несовместимой с GNU GPL лицензией.

В сфере веб-программирования, PHP – один из самых популярных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей доступности, быстроте выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на базе лицензии PHP.

Популярность в сфере создания веб-сайтов определяется присутствием довольно большого комплекта интегрированных средств для создания вебприложений.

**Ключевые из них:**

– автоматическое извлечение POST и GET-параметров, а также переменных среды веб-сервера в предопределённые массивы;

– взаимодействие с множеством разных СУИБ (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO);

– автоматизированная отправка HTTP-заголовков;

– HTTP-авторизация; – работа с сессиями и cookies;

– работа с локальными и удалёнными файлами, сокетами;

– обработка файлов, загружаемых на сервер;

– работа с XForms.

На сегодня PHP используется десятками тысяч web-разработчиков. Согласно рейтингу корпорации TIOBE, который основывается на данных поисковых систем, в апреле 2016 года PHP занимает 6 место среди языков программирования. К самым крупным сайтам, использующим PHP, относятся Facebook, ВКонтакте, Wikipedia и прочие.

PHP можно применять и для разработки GUI-приложений, хотя он и не слишком распространён в этом качестве.

Для разработки кроссплатформенных приложений существуют пакеты PHP-GTK и PHP-Qt, представляющие собой обёртки для популярных библиотек виджетов.

Для разработки графических приложений для Windows существуют свободный пакет WinBinder (написан на Си, фактически – обёртка для WinAPI).

Также есть реализация PHP для .NET/Mono – Phalanger и для JVM – JPHP, результатом компиляции PHP-кода в Phalanger может быть любое .NETприложение, в то же время JPHP поддерживает расширение Swing, почти целиком портированное из среды Java.

История PHP началась в 1994 году, когда датский программист Расмус Лердорф создал набор скриптов на Perl/CGI для вывода и учёта просмотров его онлайн-резюме, обрабатывающий шаблоны HTML-документов. Лердорф дал название набору Personal Home Page (Личная Домашняя Страница). В скором времени функциональности и скорости Perl – интерпретатора скриптов – перестало хватать, и Лердорф с использованием языка C создал новый интерпретатор шаблонов PHP/FI (англ. Personal Home Page / Forms Interpreter – «персональная домашняя страница / интерпретатор форм»). 8 июня 1995 года, вышел первый публичный релиз – PHP/FI 2.0.

В 2014 году было проведено голосование, по результатам которого следующую версию назвали PHP 7, выход которой произошел в декабре 2015 года.

PHP7 основывается на экспериментальной ветке PHP, которая изначально имела название phpng (PHP Next Generation - Следующее поколение PHP), и создавалась с уклоном на большую производительности при сниженном потребления памяти. В новой версии присутствует возможность указывать тип данных, которые возвращаются из функции, а также добавлен контроль передаваемых типов для скалярных данных.

Для работы программы не надо описывать какие-либо переменные, применяемые модули и т. п. Любая программа может начинаться прямо с тега <php.

PHP – язык программирования с динамической типизацией, не требующим указания типа при объявлении переменных, как и объявления самих переменных. Преобразования между скалярными типами часто происходит неявно без дополнительных действий (впрочем, PHP предоставляет большие возможности и для явного преобразования типов).

PHP-скрипты обычно обрабатываются интерпретатором в порядке, обеспечивающем кроссплатформенность приложения:

1. лексический анализ исходного кода и генерация лексем,

2. синтаксический анализ полученных лексем,

3. генерация байт-кода,

4. выполнение байт-кода интерпретатором (без создания исполняемого файла).

Для повышения быстродействия приложений присутствует возможность использовать так называемые акселераторы - специальное программное обеспечение, призванные обеспечить ускорение работы скрипта. Принцип работы акселераторов заключается в кэшировании сгенерированного байт-кода в памяти и/или на диске, таким образом, из процесса работы приложения исключаются этапы 1–3, что ведёт к ускорению работы.

Важной особенностью является то, что разработчику не обязательно думать о распределении и освобождении памяти после работы его программы. Ядро PHP имеет средства для автоматического управления памятью – вся выделенная память возвращается системе после окончания работы скрипта.

# CSS – язык каскадных таблиц стилей

CS.S (англ. Cascad..ing St.y.le Shee. .ts – каскадные таблицы стилей) – язык задающий внешний вид документа, написанного с использованием языка гипертекстовой разметки, написанных с использованием языков разметки H.TML и XH.TML, также применим к любым XM.L-документам, например, к SV.G или X.UL.

C.SS применяется для определения цветов, шрифтов, размещения отдельных блоков и других аспектов представления внешнего вида веб-страниц. Главной целью разработки C.S.S считалось разделение описания логической структуры веб-страницы (которая выполняется с помощью HT.ML или других языков разметки) от описания внешнего образа этой веб-страницы (что выполняется с содействием формального языка CS.S). Такое деление может повысить доступность документа, дать большую гибкость и вероятность управления его представлением, вместе с тем понизить сложность и повторяемость в структурном содержимом. C.SS дает возможность демонстрировать один и тот же документ в разнообразных стилях или способах вывода, таких как экранное представление, печатное представление, чтение голосом, или при выводе устройствами, использующими шрифт Брайля.

**Приемы подключения CSS к документу**

Правила CSS пишутся на формальном языке CS.S и размещаются в таблицах стилей, это значит, что таблицы стилей охватывают в себе правила C.S.S. Эти таблицы стилей могут размящ.аться как в данном веб-документе, внешний вид которого они описывают, так и в отдельных файлах, которые имеют формат C.SS. Формат C.SS – это обыкновенный текстовый файл. В нем (файле .c.s.s) не находится ничего, кроме списка правил CS.S, а также комментариев к ним. Эти таблицы стилей могут быть подсоединены, внедрены в описываемый ими веб-документ четырьмя разными методами:

– когда таблица стилей располагаться в отдельном файле, она может быть подсоединена к веб-документу путем тега <li.nk>, размещающегося в данном документе среди тегов <hea.d> и </hea.d>;

– когда таблица стилей располагаться в отдельном файле, она может быть подключена к веб-документу путем директивы @impo.rt, занимающей место в этом документе среди тегов <st.yle> и </styl.e>. Все правила данной таблицы работают на протяжении всего документа;

– когда таблица стилей описана в самом документе, она может располагаться в нём среди тегов <styl.e> и </st.yle>;

– когда таблица стилей описана непосредственно в самом документе, она может располагаться в нём в теле какого-то отдельного тега этого документа. Все правила этой таблицы функционируют только на содержимое этого тега.

В первых двух случаях к документу применены внешние таблицы стилей, а в других двух случаях – внутренние таблицы стилей.

**Правила построения C**.**S**.**S**

В первых трёх случаях подключения таблицы CS.S к документу каждое правило C.SS из таблицы стилей имеет две основные части – селектор и блок объявлений. Селектор, находящийся в левой части правила, устанавливает, на какие части документа распространяется правило. Блок объявлений находится в правой части правила. Он входит в фигурные скобки, и состоит из одного или более объявлений, разделённых знаком «;». Каждое объявление представляет собой комбинацию свойства CS.S и значения, разделённых знаком ":". Селекторы могут группироваться в одной строке через запятую. Свойство применяется к каждому из них.

# JavaScript (JQuery, Ajax) – язык программирования

Язык программирования Jav.a.Script был разработан компанией Netscap.e Commu.nication Corp.oration в 1995 году. Изначально язык назывался Mo.cha, но затем был переименован в Live.Script. Еще позже язык был переименован в JavaScript под влиянием пополурности языка программирования Java. На синтаксис языка оказали сильное влиение такие языки как Java и Си. JavaScript создавался в основном для выполнения сценариев на стороне клиента.

Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа с помощью тега <script> и интерпретируется браузером по мере загрузки этого документа. С помощью JavaScript можно динамически изменять текст загружаемого HTML-документа и реагировать на события, связанные с действиями пользователя или изменениями состояния документа или окна.

JavaScript – объектно-ориентированный язык программирования. Программисту доступны многочисленные объекты, такие, как документы, гиперссылки, формы, фреймы и т.д. Объекты характеризуются описательной информацией (свойствами) и возможными действиями (методами).

Текст сценария оформляется как комментарий, чтобы не было проблем у посетителей, браузеры которых не понимают JavaScript. Кроме того, к символам, завершающим комментарий добавляется еще два символа "/", т.к. некоторые браузеры рассматривает строку, состоящую только из символов "->", как ошибочную.

**Переменные в Ja**.**vaScri**.**pt**

Имя переменной не должно совпадать с зарезервированными ключевыми словами JavaScript: abstract, boolean, break, byte, case, catch, char, class, cons.t, continue, default, do, double, else, extends, false, final, finally, float, for, function, goto, if, implements, import, in, insta.nseof, int, interface, long, native, new, null, package, private, protected, public, return, short, static, super, sinchronized, switch.

Переменные в JavaScript объявляются с помощью ключевого слова var. При объявлении тип переменной не указывается. Этот тип присваивается переменной только тогда, когда ей присваивается какое-либо значение.

Типы данных:

- Числа;

- Логические значения. Могут принимать значение tr.ue и false.

- Строки. Последовательность символов, заключенная в одинарные или в двойные кавычки. Строка, ограниченная одинарными кавычками может содержать двойные кавычки, а строка, ограниченная двойными кавычками,

может содержать одинарные кавычки.

**Объекты Jav**.**aScri**.**pt**

Язык JavaSc.ript является объектно-ориентированным. Объекты JavaSc.ript представляют собой наборы свойств и методов. Можно сказать, что свойства объектов – это данные, связанные с объектом, а методы – функции для обработки данных объекта.

В языке JavaS.cript имеется три вида объектов: встроенные объекты, объекты брa.узера и объекты, создаваемые программистом.

JavaScript поддерживает следующий набор встроенных объектов: Array, Boolean, Date, Global, Function, Math, Number, String.

**Безопасность**

Ja.vaScript дает возможность вероятным авторам вредоносного кода запускать его на любом ПК сети: для этого довольно открыть на нём веб-страницу. Это обуславливает присутствие двух принципиальных ограничений:

– JavaS.cript-программы исполняются в песочнице, в которой они могут исполнять лишь ограниченный круг операций, а не задачи программирования общего предназначения (скажем, создание файлов, работа с сокетами),

– для Java.Script-кода употребляется политика общего происхождения, в соответствии с которой скрипт, встроенный в страницу, не может получить доступ к ряду свойств объектов другой страницы при отличии в протоколе, хосте и номере порта этих страниц.

Кроме сего, создатели браузеров вносят дополнительные ограничения в ответ на имеющие место злоупотребления. Так появился, запрет на открытие окна, размер одной стороны которого меньше ста пикселей

**Наиболее популярные области применения JavaS**.**cript:**

* динамическое создание документа с помощью сценария;
* предварительная проверка введённых пользователем данных до передачи их на сервер;
* создание динамических HTML-страниц;
* взаимодействие с пользователем при решении "локальных" задач.

**jQuery**— [набор функ.ций JavaS.cript](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript), фокусирующийся на взаимодействии [**JavaSc**.**ript**](https://ru.wikipedia.org/wiki/JavaScript)и[**HTM**.**L**](https://ru.wikipedia.org/wiki/HTML).

Библиотека jQ.ue.ry помогает легко получать доступ к любому элементу [DO.M](https://ru.wikipedia.org/wiki/Document_Object_Model), обращаться к атрибутам и содержимому элементов [DO.M](https://ru.wikipedia.org/wiki/Document_Object_Model), манипулировать ими. Также библиотека jQu.ery предоставляет удобный [AP.I](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для работы с [AJAX](https://ru.wikipedia.org/wiki/AJAX). Разработка jQu.ery ведётся командой добровольцев на пожертвования.

**AJ**.**A**.**X (*Asy***.***nchr***.***onous*** .***Java***.***scrip***.***t and XML* — «асинхронный**[**JavaScript**](https://ru.wikipedia.org/wiki/JavaScript)**и**[**X**.**ML**](https://ru.wikipedia.org/wiki/XML)**»)** — подход к построению интерактивных [пользовательских интерфейсов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) [веб-приложений](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5), заключающийся в «фоновом» [обмене данными](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D0%B5%D0%BD_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%BC%D0%B8) [браузера](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B1%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) с [веб-сервером](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). В результате при обновлении данных [веб-страница](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. По-русски иногда произносится [тран.с.литом](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D0%B8%D1%82) как «а.я.кс». У аббревиатуры AJA.X нет устоявшегося аналога на кириллице.

**Сравнение стандартного подхода и AJAX**

**В классической модели веб-приложения:**

* Пользователь заходит на веб-страницу и нажимает на какой-нибудь её элемент.
* [Браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) формирует и отправляет [запрос](https://ru.wikipedia.org/wiki/HTTP) [серверу](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80).
* В ответ сервер генерирует совершенно новую веб-страницу и отправляет её браузеру и т. д., после чего браузер полностью перезагружает всю страницу.

**При использовании AJAX:**

* Пользователь заходит на веб-страницу и нажимает на какой-нибудь её элемент.
* [Скрипт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) (на языке [Jav.a.Scr.i.pt](https://ru.wikipedia.org/wiki/JavaScript)) определяет, какая информация необходима для обновления страницы.
* [Браузер](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) отправляет соответствующий запрос на [сервер](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)).
* Сервер возвращает только ту часть документа, на которую пришёл запрос.
* [Скрипт](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82) вносит изменения с учётом полученной информации (без полной перезагрузки страницы

# MySQL – система управления базами данными

**MySQ**.**L** - это самая популярная (а некоторые добавят, что еще и самая лучшая в мире) СУБД с открытым исходным кодом. На самом деле MyS.QL составляет все более значительную конкуренцию таким дорогостоящим гигантам, как O.racle и Mi.cro.soft SQL Se.rver. Продукт распространяется как под GNU Gener.al Pu.blic Lic.ense, так и под собственной про.п.риаритарной лицензией. Помимо этого, у лицензионных пользователей есть возможность запрашивать создание любой функциональности у разработчиков. Именно по такому заказу почти в самых ранних версиях появился механизм репликации.

My.S.QL является отличным реше.нием для малых и средних приложений. MyS.QL входит в состав серверов WA.MP, Ap.pServ, LAM.P и в с.бор.ки серверов Денвер, XAMPP, Vertri.goServ. Обычно MyS.QL используется в качестве сервера, к которому об.ращаются локальные или удалённые клиенты, однако в дистрибутив включена биб.ли.отека внутр. .еннего сервера, позволяющая включать My.SQ.L в автономные программы.

Гибкость СУБД MyS.QL обеспечивается поддержкой множества типов таблиц: пользователи имеют выбор между табли.цами типа MyI.SAM, поддерживающими полнотекстовый поиск, так и таблицами InnoDB, поддерживающие транзакции на уровне отдельных записей. Кроме того, СУ.БД MyS.QL ра.спр.остраняе.тся со специальным типо.м таблиц EXA.MPLE, демонстрирующим принципы создания новых типов таб.лиц. Благодаря открытой архитектуре и GPL-лицензии, в MySQL все время появляются новые типы таблиц. MYSQL является небольшой и быстрой реляционной СУБД, основанной на традициях Hu.ghes Tec.hnolo.gies M.ini SQL (mS.QL).

SQL – (Struc.tured Qu.ery Langu.age - структурированный язык запросов). SQL создан для работы с реляционными базами данных. Он позволяет пользователям взаимодействовать с базами данных (просматривать, иск.ать, добавлять и уп.рав.лять данными). M.yS.QL соответствует спецификации AN.SI 9.2 S.QL.

**Основными достоинствами M**.**yS**.**QL являются:**

- Мн.огопоточн.ость. Поддержка нескольких одновременных запросов;

- Оптимизация связей с присоединением многих данных за один проход;

- Записи фиксированной и переменной длины;

- ODB.C драйвер в комплекте с исходником;

- Гибкая система привилегий и паролей;

- Интерфейс с языками C и P.e.rl, PH.P;

- До 16 ключей в таблице. Каждый ключ может иметь до 15 полей;

- Поддержка ключевых полей и специальных полей в операторе CREATE;

- Поддержка чисел длинной от 1 до 4 байт (in.ts, fl.oat, dou.ble, fix.ed), строк переменной длины и меток времени;

- Интерфейс с языками C и Pe.rl;

- Основанная на потоках, быстрая система памяти;

- Утилита проверки и ремонта таблицы (is.am.ch.k);

- Все данные хранятся в формате IS.O8.859\_1;

- Все операции работы со строками не обращают внимания на регистр символов в обрабатываемых строках;

- Псевдонимы применимы как к таблицам, так и к отдельным колонкам в таблице;

- Все поля имеют значение по умолчанию. I.NSER.T можно использовать на любом подмножестве полей;

- Легкость управления таблицей, включая добавление и удаление ключей и полей.

MyS.QL имеет A.PI для языков Del.phi, C, C.++, Эйфель, J.a.v.a, Лисп, Pe.rl, P.H.P, P.ython, Ru.by, Smal.ltalk, Компонентный Паскаль и T.cl, библиотеки для языков платформы .N.E.T, а также обеспечивает поддержку для O.D.BC через OD.BC-драйвера My.O.DB.C.

MyODBC представляет собой драйвер O.DB.C (2.50) уровня 0 (с отдельными возможностями уровней 1 и 2) для подсоединения совместимого с ODBC приложения к M.yS.QL. My.ODB.C функционирует на всех системах Micro.soft Window.s и на основной массе платформ U.nix.

# ****phpDesigner 8 – редактор кода****

**ph**.**pDesigne**. **r 8** – это платный редактор кода (+ID.E) заточенный именно под **веб-разработку**, с его помощью вы сможете создавать Интернет-проекты на [**P**.**HP**](http://site-on.net/create/php)**,**[**CS.S**](http://site-on.net/create/html/3-vvedenie-v-css)**, JS** и[**H**.**TML**](http://site-on.net/create/html/1-osnovi-html), а та.кже по.высить свои навыки благодаря встроенной библиотеке функций. Имеется поддержка всех PHP Фреймворков (Ze.nd, CodeIgniter, Yii, Sym.fony and Prado), а также JavaS.c.ript Фреймворков (jQu. .ery, Ext JS, D.ojo, Mo.oTools, Prot.otype и других), Благодаря средствам для автоматической подсветки кода, **ph**.**pDe**.**si**.**gner** также подойдет WAMP/LAMP- и AJAX-разработчикам. Кроме этого программа содержит емкие библиотеки, содержащие более 3000 функций, доступ к которым легко осуществляется в процессе программирования.

**phpDesi**.**g**.**ner** использует самые современные наработки в области CSS и HTML, поэтому зачастую необходимо быть внимательным при поддержке старых браузеров, не поддерживающих некоторые функции.

Окно программы phpDesigner 8 (Рисунок 2.6):

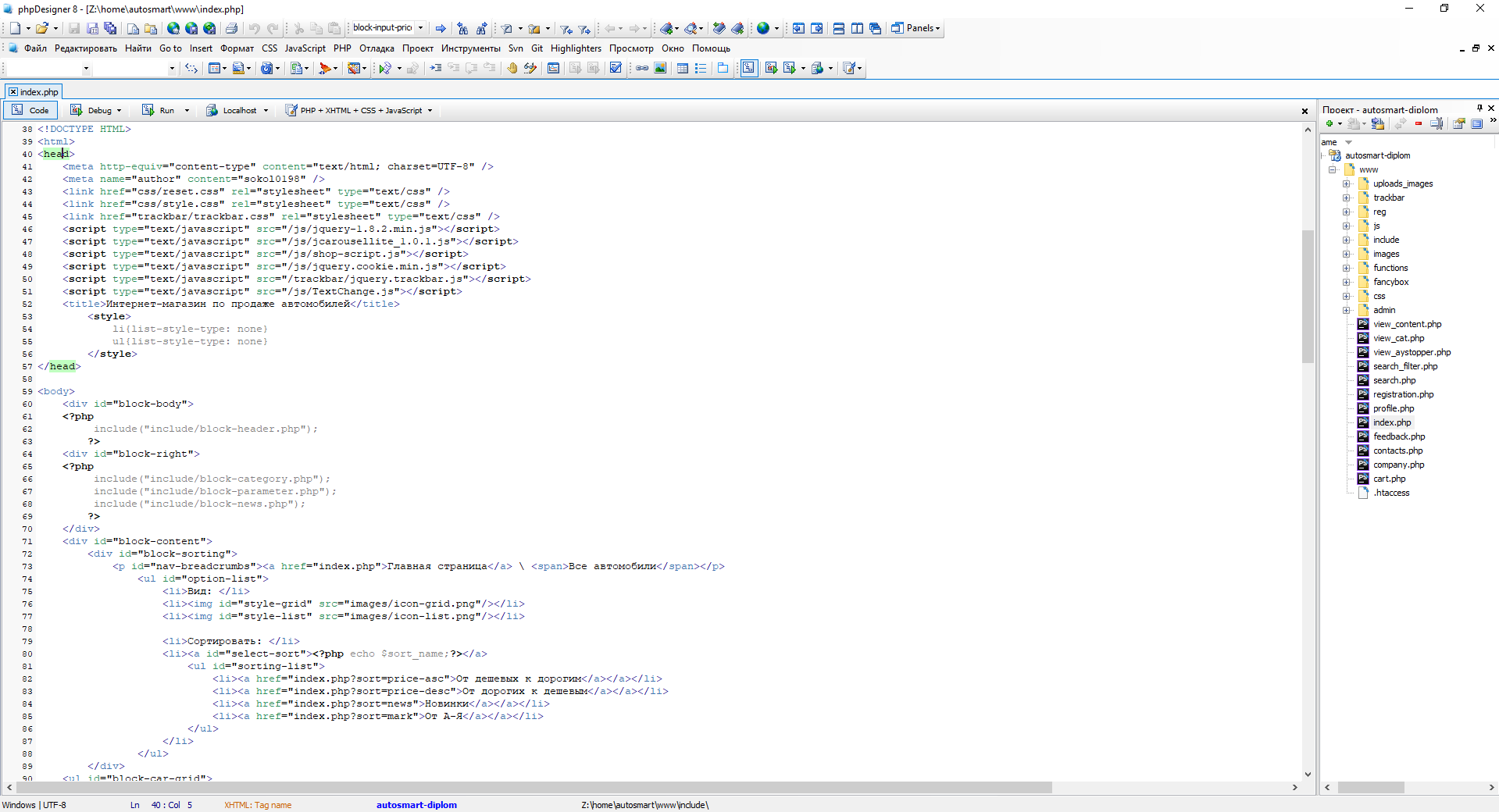


Рисунок 2.6 – Главное окно программы phpDesigner 8

**Основные возможности phpD**.**esigner 8**

· Подсветка синтаксиса для множества языков;

· Проверка синтаксиса для PH.P, .HTML и CS.S;

· Автоматическое завершение кода для PH.P, H.TML, CS.S и JavaScr.ipt;

· Поддержка таких библиотек Java.Script, как JQ.uery, Ex..tJS, YU.I, Do..jo, Mo.oTools и Prot.otype;

· Встроенная система подсказок для PHP и JavaS.cript;

· Инспектор кода для H.TM.L;

· Кодовый обозреватель для P.HP, CS.S и Jav.aScript;

· Переход к любому объявлению PH.P, CSS и Ja.vaScr.ipt;

· Отладка и профилирование PHP скриптов с помощью Xd.ebu.g;

· Поддержка FT.P, SF.TP и To.rt.ois.eS.VN;

· Интуитивная навигация по объектам вашего кода;

· Наличие встроенных помощников;

· Возможность смены скинов.

# Denwer (web-сервер Apache)

**Denw**..**er** (от сокр. *Д.н.w.р* или *ДНВР* — джентльменский набор [We.b](https://ru.wikipedia.org/wiki/Web)-разработчика) — набор дистрибутивов (локальный сервер [WAMP](https://ru.wikipedia.org/wiki/WAMP)) и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет-страниц) на локальном ПК под управлением ОС [Windows](https://ru.wikipedia.org/wiki/Windows).

**Apa**.**che** является кроссплатформенным программным обеспечением, поддерживая такими ОС как Linux, BSD, M.ac OS, Mic.rosoft Wind.ows.

Основными достоинствами Apac.he считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках.

Ядро Apa.che включает в себя основные функциональные возможности, такие как обработка конфигурационных файлов, протокол HTTP и система загрузки модулей. Ядро (в отличие от модулей) полностью разрабатывается Apac.he So.ftware Foun.dation, без участия сторонних программистов.

Ядро apa.che может функционировать в чистом виде, без использования модулей. Однако функциональность такого решения ограничена.

Ядро Apac.he полностью написано на языке программирования C.

Система конфигурации Apach.e основана на текстовых конфигурационных файлах. Имеет три условных уровня конфигурации:

1. Конфигурация сервера (htt.pd.con.f).
2. Конфигурация виртуального хоста
3. Конфигурация уровня директории (.h.taccess).

Имеет собственный язык конфигурационных файлов, основанный на блоках директив. Практически все параметры ядра могут быть изменены через конфигурационные файлы, вплоть до управления MPM.

Apache имеет встроенный механизм виртуальных хостов. Он позволяет полноценно обслуживать на одном IP-адресе множество сайтов (доменных имён), отображая для каждого из них собственное содержимое.

Существует множество модулей, добавляющих к Apache поддержку различных языков программирования и систем разработки.

К ним относятся:

1. PHP (mod\_p.hp).
2. Pyth.on (m.od\_pyth.on).
3. R.uby (ap.ache-rub.y).
4. Per.l (m.od\_perl).
5. ASP (ap.ache-a.sp).

Кроме того, Apache поддерживает механизмы CGI и FastCGI, что позволяет исполнять программы на практически всех языках программирования, в том числе C, C++, sh, Java.

# Анализ, проектирование и реализация сайта

# Обзор и анализ методов проектирования сайта

Заказчику необходимо вести контроль процесса создания сайта и наблюдать как проходят все этапы разработки сайта, но для этого не обязательно сильно вникать в процесс и достаточно изучать вопрос. Создание сайтов происходит при помощи разных методов разработки. Понимание нужно прежде всего для того, чтобы понять, чего заказчик на самом деле хочет получить в результате и правильно донести это команде разработчиков, которой будут доверены работы над сайтом. Кроме того, это поможет избежать чрезмерных затрат на реализацию проекта в случае его затягивания, так как все этапы разработки будут у заказчика перед глазами. Так же будет возможность вовремя понять, что в движке сайта не используется шаблон сайта и не выдается за собственную разработку командой разработчиков.

На сегодняшний день существует огромное количество различных методов создания сайтов, различающихся в зависимости от назначения и типа сайта, умений разработчика или финансовых возможностей заказчика. При этом классифицировать методы можно по разным признакам: используемому программному обеспечению, архитектуре сервера, технологическим признакам либо даже маркетинговым. Однако перечисленные особенности касаются, прежде всего, разработчика, провайдера либо регистратора.

Для будущего владельца сайта главную роль скорее сыграют такие факторы, как время выполнения и стоимость. Разнообразных же методов разработки, отличающихся друг от друга, десятки. Несмотря на множество технологических различий, принципиально отличающихся подходов к разработке можно выделить лишь несколько. Если группировать методы создания сайтов, исходя из того, что признак цена-время является самым востребованным, то можно выделить три основных метода:

1. **Разработка с нуля**

Данный метод наиболее длительный и трудный. При этом он требует знаний хотя бы одного языка веб-программирования (PHP, HTML, CSS), а также опыт работы с базами данных (MySQL). Можно также воспользоваться наиболее популярными программами для создания сайтов вручную, к которым относятся Dreamweaver, FrontPage, WebSite X5, а также Photoshop (для работы с изображениями).

Однако всю рутинную работу разработчику придется выполнить самому. Результаты при этом могут получиться более чем скромными, характерными для новичков и любителей в сфере SEO. Главным минусом данного метода является необходимость осуществлять все самостоятельно, затрачивая значительное количество времени и сил.

1. **Использование конструктора**

Самый простой метод разработки, поскольку не требует знаний профессионального веб-программиста. Однако на качественную работу также рассчитывать не стоит. Существует множество инструментов для разработки по тем либо иным шаблонам и макетам. К самым популярным и хорошо изученным конструкторам относятся «narod.ru», «setup.ru», «umi.ru», «ucoz.ru» и подобные им.

Зачастую, воспользовавшись конструктором на начальном этапе, разработчик может впоследствии пожалеть о потерянном на изучение работы конструктора времени, поскольку данный метод таит в себе пару серьезных недостатков. Один заключается в ограниченности возможностей такого рода конструктора, невозможности реализовать более, нежели предусмотрено его функциями, а другой – в наличии специальных фрагментов кода и сложности раскрутки созданного ресурса.

1. **Использование системы управления сайтом**

Самый быстрый, а также широко используемый метод. На сегодняшний день широко используются такие системы («движки»), как WordPress, Joomla, Django, Drupal и прочие, которые идеально подойдут для сайтов либо блогов. Для форумов более подходящими станут: phpbb, ExBB, PunBB, SMF.

Использование движков (CMS) предоставляет доступ к множеству современных шаблонов и схем построения, а также значительно ускоряет и упрощает работу разработчика.

Будущий владелец сайта сам выбирает, какой из методов использовать. Конечно же, он также может заказать сайт в веб-студии, ведь разработанный с применением новейших технологий и инструментов слаженным коллективом дизайнеров, программистов, оптимизаторов, верстальщиков, копирайтеров, аниматоров профессиональной веб-студии сайт всегда предпочтителен.

# 

# Проектирование сайта

При проектировании нашего сайта необходимо придерживаться следующих требований:

– не создавать больше 5-7 пунктов меню, та как это порог для оптимального восприятия содержащейся в нем информации;

– не использовать свыше 30-40 кбайт графики на одну страницу; – использовать только картинки в форматах JPG, JPEG, PNG и GIF;

– пользователь, который находится на сайте, должен иметь право попасть в любое место вашего сайта или же странички;

– разделять большие массивы информации;

– называть созданные странички только на английском языке;

– избегать горизонтальных прокруток экрана;

– цвет ссылки не должен быть похожим на фон страницы;

– создавать страницы, которые доступны всем пользователям вне зависимости от того, каким браузером или экраном они пользуются;

– графика на сайте должна иметь лишь вспомогательное значение;

– не использовать на сайте текст, набранный в верхнем регистре;

– размер каждой страницы сайта не должен быть выше 80-100 килобайт вместе с графикой (приемлемо – 40-50), за исключением страниц с большим объемом информации.

Со спецификой разрешения экрана связана проблема выбора между меняющимися страницами (меняется размер и осуществляется регулировка под различные размеры окон) и фиксированными по размеру (что дает возможность разработчику лучше управлять размерами страницы). Каждый из подходов имеет свои веские аргументы. Безусловно, можно найти доводы за и против любого из них.

**Различают следующие типы макетов сайта:**

* 1. **Фиксированный макет сайта**

Считается, что фиксированный дизайн –это самый простой макетсайта.Обычно он располагается по центру окна и имеет, как можно понять из названия, фиксированные параметры ширины. Его размеры не зависят от разрешения, он в любом случае сохраняет заданные значения. Преимущества такого варианта очевидны.

Во-первых, данный макет очень легко верстать. Все параметры: ширину будущего сайта, величину его текстовых модулей, графических изображений и так далее – разработчик задает уже на стадии подготовки, так что верстальщику остается только следовать этим указаниям. Кроме того, когда известна ширина всех колонок, размеры картинок и баннеров подбирать гораздо проще. Ну а браузеры гораздо лучше воспринимают именно фиксированный макет страницы сайта, так что на отладку работы уходит гораздо меньше времени.

Однако на фоне таких значимых плюсов нельзя забывать и о минусах, главный из которых – неэффективное использование свободного пространства. Так, если задать слишком большую ширину, то на компактных экранах нетбуков сайт будет отображаться крайне не слишком привлекательно, появится горизонтальная полоса прокрутки. А при большом расширении монитора по краям сайта просто останутся две белые полосы.

Тем не менее в современном сайтостроении такие html макеты сайтов сохраняют популярность. Особенно часто в таком ключе разрабатывается макет сайта визитки.

* 1. **Резиновый макет сайта**

Резиновый макет сайта имеет ширину, установленную в процентном соотношении. Это значит, что при изменении размера окна ширина и высота сайта также будет изменяться. В отличие от фиксированного макета, который на больших экранах отображается непривлекательно, «резиновый сайт» занимает всю площадь окна. При этом подобные веб-страницы удачно смотрятся не только на мониторах ПК или ноутбуков, но корректно отображаются и на экранах телефонов, а с их печатью никогда не возникает сложностей. Зато есть другая проблема: на больших мониторах текст сильно растягивается, читать его становится некомфортно. Конечно, в этом случае очень помогает контроль ширины при помощи свойства max-width, а некоторые владельцы больших мониторов просто уменьшают размер окна, и все же проблема остается. Кроме того, разработчикам гораздо труднее контролировать то, что видят пользователи различных устройств на своих экранах, а, значит, отладка происходит гораздо сложнее.

# 

* 1. **Эластичный макет сайта**

По своему внешнему виду такие psd макеты сайтов ничем не будут отличаться от резиновых либо фиксированных – ровно до того момента, пока вы не попробуете поменять в браузере размер шрифта. В этом случае вы заметите, что вместе с параметрами шрифта изменились и другие модули вебстраницы: шапка, футер, графические элементы. Размер всего макета в целом, равно как и его отдельных модулей, здесь задается не в пикселях или процентах, а в em – относительной единице длины.

Преимущества данного макета очевидны: во-первых, ваш сайт сохраняет стабильное соотношение элементов в любой ситуации и не «разваливается» при увеличении шрифта. «Эластичный» веб-ресурс будет хорошо смотреться в любой операционной системе, однако верстать такой макет не очень удобно, а тестировать сайт нужно довольно долго. Кроме того, сфера применения подобного дизайна ограничена. И все же многие разработчики отдают предпочтение именно такому макету, поскольку сайт в данном случае всегда сохраняет целостность элементов и первоначальный внешний вид, а это очень важно для раскрутки.

# 

* 1. **Адаптивный макет сайта**

Строго говоря, адаптивный макет сайта – это даже не один, а сразу несколько эскизов, показывающих, как будет отображаться страница в различных расширениях. То есть фактически дизайнер создает не один макет сайта в photoshop, а множество вариантов одной и той же страницы. В ходе

верстки заготавливается несколько правил под каждый спектр расширений, а также скрипты, которые определяют, каким правилом следует пользоваться в данном конкретном случае.

Основное преимущество данного дизайна – его бесспорное удобство для посетителя. Однако для разработчика это самый трудный тип дизайна, поскольку нужно фактически создать не один, а сразу несколько макетов с уникальной графикой и своим CSS. Вот почему его разработка и верстка отнимают так много времени, хотя тестировать его легче, чем эластичный или резиновый макет.

* 1. **Комбинированный макет сайта**

По большому счету, комбинированный тип не может считаться самостоятельным, так как лишь сочетает в себе элементы четырех предыдущих видов верстки. И в то же время это наиболее мобильный, самый гибкий вид. Комбинированным может быть макет сайта интернет магазина, визитки, корпоративного сайта – словом, любого веб-ресурса. Добавляя при разработке элементы резиновой, эластичной, фиксированной, адаптивной верстки, разработчик может создать идеальный макет продающего сайта, скрыть недостатки и подчеркнуть сильные стороны того или иного типа.

# Принципы реализации элементов дизайна сайта

Современный веб-дизайн основывается на принципах акцентирования, контраста, балансировки, выравнивания, повторения и удобстве восприятия. Чем больше внимания уделяется этим принципам, тем более удачным в итоге получается дизайн.

**Акцентирование**

Акцентирование – это подчеркивание особенной важности или значения какого-либо элемента. Во многом оно непосредственно связано, а то и совпадает с определением иерархии. Чтобы сохранить в процессе конструирования сайта принцип акцентирования, нужно провести анализ содержимого сайта и узнать, какая иерархия элементов присутствует в его содержимом. Выяснив это, возможно разработать сайт с грамотно осуществленной иерархией.

Превосходный способ выяснить, чему нужно уделить внимание – вообразить себе все элементы, которые необходимы на веб-странице. Потом нужно вообразить, какова будет расстановка значимости этих элементов. Вслед за этим, держа в уме свои прикидки, возможно начинать создавать дизайн, в котором зрительная иерархия веб-страницы будет отображать значимость элементов, установленную ранее.

Один из факторов, по которым это столь важно, заключается в том, что возможно избежать акцентирования на всех элементах сразу. Кроме того, это даст возможность избежать ловушки в виде так называемой случайной иерархии. Всегда лучше заблаговременно решить, какие конкретно элементы будут иметь преимущество в визуальном плане, а не забрасывать его на волю случая. Если пробовать сделать акцент на все элементы сразу, то в реальности внимание пользователя ни на чем не сосредоточиться.

**Контрастирование**

Контрастирование – это зрительное разделение двух и более элементов. Элементы с большим уровнем контрастирования смотрятся четкими и обособленными, а элементы с низким уровнем контрастирования выглядят сливающимися и имеют тенденцию сливаться в единую массу. Существует большое количество характеристик элементов дизайна, которыми возможно манипулировать с целью достижения нужного уровня контрастирования элементов, в том числе цвет, размер, местоположение, тип и толщину шрифта.

Если элементы дизайна будут выделяться на фоне друг друга, это придаст визуальное разнообразие сайту и даст возможность исключить заурядности в его внешнем виде. Кроме того, контрастирование дает возможность притягивать внимание, обеспечивая акцент на конкретных элементах.

Между главными принципами дизайна имеется взаимосвязь, которая выражается в том, что контрастирование затрагивает акцентирование, удобство восприятия и прочие принципы дизайна. Наибольшее воздействие контрастирование оказывает на визуальную иерархию веб-страницы, так как оно часто применяется для обеспечения желаемого акцента на определенных элементах.

Таким образом, оно позволяет быстро привлекать внимание к основным элементам, например, к содержимому, элементам, позволяющим содержать конкретные действия, или к тексту, указывающему, в чем состоит назначение сайта. Следует понимать цель, преследуемую сайтом, чтобы специально привлекать интерес пользователя к необходимым элементам путем продуманного контрастирования.

# 

**Балансировка**

Принцип балансировки вращается вокруг идеи о том, как разделены элементы в дизайне и как они сопоставляются с общим распределением визуальной нагрузки на веб-странице. От этого зависит то, в какой степени уравновешенным будет дизайн в визуальном плане. Группировка элементов в дизайне образовывает визуальную нагрузку.

По большей части эта нагрузка уравновешивается через применение равнозначной нагрузки, находящейся на другой чаше «весов», в следствии чего достигается баланс в дизайне. Если не сделать этого, то итогом будет чувство неуравновешенности дизайна, хотя это не значит, что дизайн непременно будет плохим. Однако качественно сбалансированный дизайн создает тонкое чувство уравновешенности и, как правило, в конечном счете становится более привлекательным.

Баланс бывает двух типов: симметричный и асимметричный. Симметричный баланс в дизайне достигается, когда левая и правая стороны дизайна относительно той или иной оси зеркально отображают друг друга и несут общую визуальную нагрузку. Как правило, это свойственно сайтам, где логотип и верхнее меню визуально размещаются по центру.

Асимметричный баланс достигается, когда визуальная нагрузка веб страницы равномерно распределяется по той или иной оси, но отдельные элементы двух составляющих дизайна не являются зеркально одинаковыми.

**Выравнивание**

Выравнивание – это расстановка элементов таким образом, чтобы они наиболее вплотную подступали к линиям или границам, которые они формируют. Этим достигается унификация элементов. Этот процесс часто называют работой с сеткой. Не выровненные элементы обладают свойством откалываться от общей группы и нуждаются в унификации, к которой программист и стремится. К примерам выровненных элементов можно причислить расставленные на одной линии заголовки двух столбцов, либо левые края находящихся друг над другом элементов, выровненных по отношению друг к другу.

Эти примеры достаточно очевидны, но на веб-страницах вероятны и гораздо более сложные выравнивания, которые дают возможность достигнуть унифицированного и визуально комфортного дизайна. Если придется столкнуться с задачей, которая состоит в том, чтобы взять имеющиеся на бумаге примеры дизайна и воплотить их в коде, то особо важным будет необходимость в соблюдении этого принципа.

При реализации этой задачи часто оказывается изнурительным и трудным воссоздать в дизайне выравнивание, которое проектировалось вначале. Однако трудность заключается не только в самой процедуре, но также и в том, что эти моменты очень просто выпустить из виду. Поэтому понимание того, что необходимо придерживаться принципа выравнивания, является существенным для веб-разработчика при воссоздании дизайна в коде.

**Повторение**

Повторение подразумевает многократное применение в дизайне одних и тех же элементов различными способами. Дизайн, который имеет повторения, стает унифицированным. Повторение может проявляться во множестве форм, в том числе в использовании того же цвета, очертаний, линий, шрифтов, изображений и подхода к стилизации.

Часто следование этому принципу является фактически неизбежным, так как если дизайн не содержит в себе повторяющихся элементов, то это как правило указывает на то, что он лишен логической связанности. Повторение несет в себе такую громадную выгоду, как предсказуемость. Пользователь ожидает увидеть уже знакомые ему вещи, если в основных элементах дизайна прослеживаются согласованность и постоянство.

Очень часто бывает так, что тот или иной сайт теряет свою визуальную связанность из-за того, что на каждой его странице дизайнер применяет радикально разные подходы к оформлению, вместо того, чтобы сформировать какую-то одну методику и следовать ей.

**Удобство восприятия**

Восприятие своего рода дорожка, которой пользователь придерживается при визуальном ознакомлении с элементами дизайна. Оно в основном зависит от того, на чем дизайнер акцентирует и в какой мере он обеспечил контрастирование элементов между собой. Фактически удобство восприятия целиком будет обусловливаться тем, как дизайнер скомпоновал все элементы страницы.

Пользователи склонны рассматривать те или иные вещи в предсказуемой манере. Обычно человек просматривает некий объект слева направо и сверху вниз. Именно поэтому сайты, которые навязывают пользователям просмотр контента справа налево, отталкивают тех, кто привык это делать наоборот.

В том, чтобы идти против естественного порядка вещей, нет ничего ошибочного; нужно лишь понимать все возможные результаты подобных действий. Каждый сайт предоставляет пользователю конкретное восприятие, которое может быть, как удобным, так и неудобным.

Сайт может иметь сложную структуру, но должен предоставлять естественный и комфортный подход к восприятию своего контента, чтобы не приходилось перепрыгивать взглядом по веб-странице. Сайт, который предоставляет удобное восприятие, способствует тому, чтобы у человека появилось желание еще раз пробежаться взглядом по тому, что он уже смотрел, то есть вызывать у него интерес.

# Даталогическая модель системы

База данных “db\_autosmart.sql” для интернета сайта по выбору и продаже автомобилей «AutoSmart» состоит из следующих таблиц:

Таблица 3.1 - Автомобили

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| cars\_id | Первичный ключ | Счетчик |
| title | Название | Текстовый |
| price | Цена автомобиля | Числовой |
| mark\_auto | Марка автомобиля | Текстовый |
| seo\_words | Ключевые слова | Текстовый |
| seo\_description | Мета описание | Текстовый |
| mini\_description | Краткое описание | Текстовый |
| image | Изображение автомобиля | Текстовый |
| description | Описание | Текстовый |
| mini\_featurescar | Краткое представление | Текстовый |
| featurescar | Представление | Текстовый |
| datetime | Дата и Время | Дата/Время |
| new | Новинки | Числовой |
| leader | Лучшая Цена | Числовой |
| sale | Распродажа | Числовой |
| visible | Видимость авто на сайте | Числовой |
| count\_views | Кол-во просмотров | Числовой |
| type\_car | Тип автомобиля | Текстовый |
| mark\_auto\_id | Идентификатор марки | Числовой |
| like\_cars | Кол-во отметок нравится | Числовой |

Таблица 3.2 - Категории автомобилей

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| id | Первичный ключ | Счетчик |
| type\_car | Тип автомобиля | Текстовый |
| mark\_auto | Марка автомобиля | Текстовый |

Таблица 3.3 - Регистрация пользователей

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| id | Первичный ключ | Счетчик |
| login | Логин | Текстовый |
| pass | Пароль | Текстовый |
| surname | Фамилия | Текстовый |
| name | Имя | Текстовый |
| patronymic | Отчество | Текстовый |
| email | Е-мэйл | Текстовый |
| phone | Телефон | Текстовый |
| address | Адрес | Текстовый |
| datetime | Дата и Время | Дата/Время |
| ip | IP адрес регистрации | Текстовый |

Таблица 3.4 - Регистрация администраторов

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| id | Первичный ключ | Счетчик |
| login | Логин | Текстовый |
| pass | Пароль | Текстовый |
| fio | ФИО | Текстовый |
| role | Роль | Текстовый |
| email | Е-мэйл | Текстовый |
| phone | Телефон | Текстовый |
| view\_orders | Просмотр заказов | Числовой |
| accept\_orders | Подтверждение заказов | Числовой |
| delete\_orders | Удаление заказов | Числовой |
| add\_car | Добавление автомобилей | Числовой |
| edit\_car | Редактирование автомобилей | Числовой |
| delete\_car | Удаление автомобилей | Числовой |
| view\_clients | Просмотр клиентов | Числовой |
| delete\_clients | Удаление клиентов | Числовой |
| add\_category | Добавление категорий | Числовой |
| view\_admin | Просмотр администраторов | Числовой |

Таблица 3.5 - Корзина

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| cart\_id | Первичный ключ | Счетчик |
| cart\_id\_cars | ID автомобиля в корзине | Числовой |
| cart\_price | Цена | Числовой |
| cart\_count | Кол-во | Числовой |
| cart\_datetime | Дата и Время | Дата/Время |
| cart\_ip | IP покупателя | Текстовый |

Таблица 3.6 - Купленные авто

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| buy\_id | Первичный ключ | Счетчик |
| buy\_id\_order | ID заказа | Числовой |
| buy\_id\_car | ID автомобиля | Числовой |

Таблица 3.7 - Заказы

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| order\_id | Первичный ключ | Счетчик |
| order\_datetime | Дата и время | Дата/Время |
| order\_confirmed | Статус заказа | Текстовый |
| order\_delivery | Способ доставки | Текстовый |
| order\_pay | Статус оплаты | Текстовый |
| order\_type\_pay | Способ оплаты | Текстовый |
| order\_fio | ФИО покупателя | Текстовый |
| order\_address | Адрес покупателя | Текстовый |
| order\_phone | Телефон покупателя | Текстовый |
| order\_note | Примечание | Текстовый |
| order\_email | Е-мэйл | Текстовый |

Таблица 3.8 - Загрузка изображений

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Назначение поля** | **Тип данных** |
| id | Первичный ключ | Счетчик |
| cars\_id | ID автомобиля | Числовой |
| image | Изображение | Текстовый |

# Разработка структуры Web-сайта

Мной было принято решение разработать струк.туру интернет сайта, сайта-визитки для ком.пании “AutoSmart”, которая занимается помощью в выборе и главное продажей автомобилей

Разрабатываемая структура интернет сайта выглядит следующим образом

(Рисунок 3.3):



Рисунок 3.3 – Структура интернет сайта «AutoSmart»

# Проектирование и разработка Web-сайта

Для основной части сайта мной был выбран следующий дизайн (Рисунок 3.4):

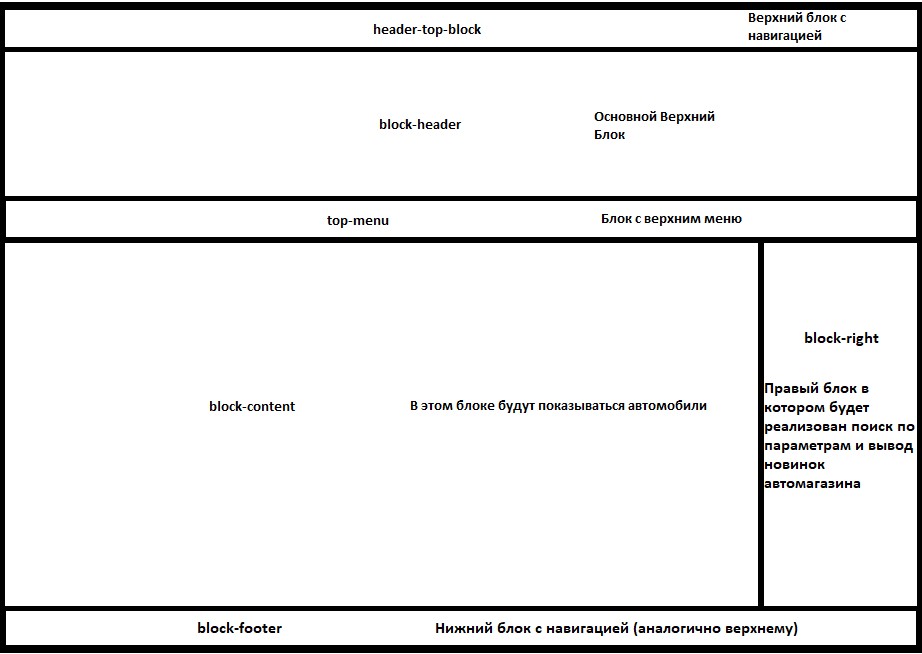


Рисунок 3.4 – Дизайн разрабатываемого интернет сайта «AutoSmart»

Как видно, я решил разбить мой сайт на множество блоков со своими определенными задачами, для каждого блока я создам отдельный php файл и просто его буду подключать, где он будет нужен.

На основе разработанной мной структуры была спроектирована главная страница Web-сайта. В ее составе находятся все основные структурные элементы, переход по которым происходит за счет гиперссылок.

В разработку сайта входит процесс создания макета Web-страниц, на который чуть позже будут накладываться все остальные элементы сайта. При этом необходимо будет формирование так называемых структурных блоков сайта - обособленных модулей, каждый из которых будет выполнять определенную роль и отвечает за определенный функционал ресурса.

**Состав Web-сайта включает в себя:**

1. **Шапка сайта.**

Самый верхний блок — это шапка сайта, заголовок сайта, хедер от англ. header. Место, в котором обычно находится хедер - верхняя часть страницы. Ориентация хедера – альбомная (рисунок 3.4).

В данном блоке размещается:

* верхний блок с навигацией
* название сайта;
* главное меню;
* блок авторизации и регистрации;
* блок поиска;
* блок информации;
* корзина;

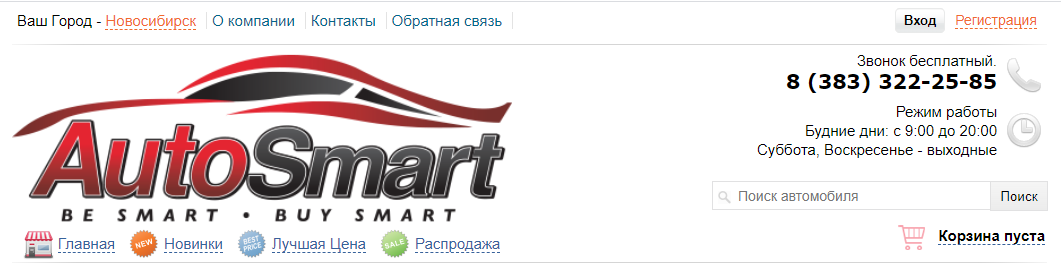


Рисунок 3.5 – Шапка интернет-сайта «AutoSmart»

1. **Основной блок (блок контента).**

Слово «Контент» происходит от английского «content» - содержание. Это самая большая по размерам и наиболее важная для посетителей - клиентов сайта часть страницы.

В данном блоке размещается:

* текстовый контент;
* графический контент.

Ширина области основного контента может измяться в зависимости от разрешения монитора – тут все зависит от типа используемого макета сайта (жесткий, резиновый) в нашем случае жесткий макет сайта. Резиновый макет позволяет менять ширину блока, с которого пользователь просматривает сайт.

Жесткий макет этого сделать не позволяет (рисунок 3.6):

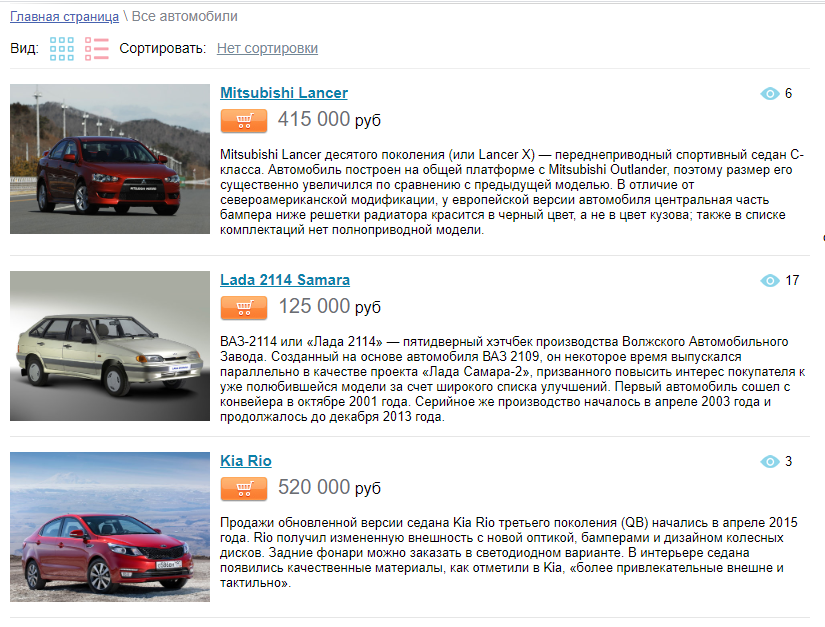


Рисунок 3.6 – Блок контента (линейно) интернет-сайта «AutoSmart»

Также на моем сайте можно сделать отображение сеткой, это зависти от предпочтений клиентов, кому как удобней просматривать контент (рисунок 3.7):

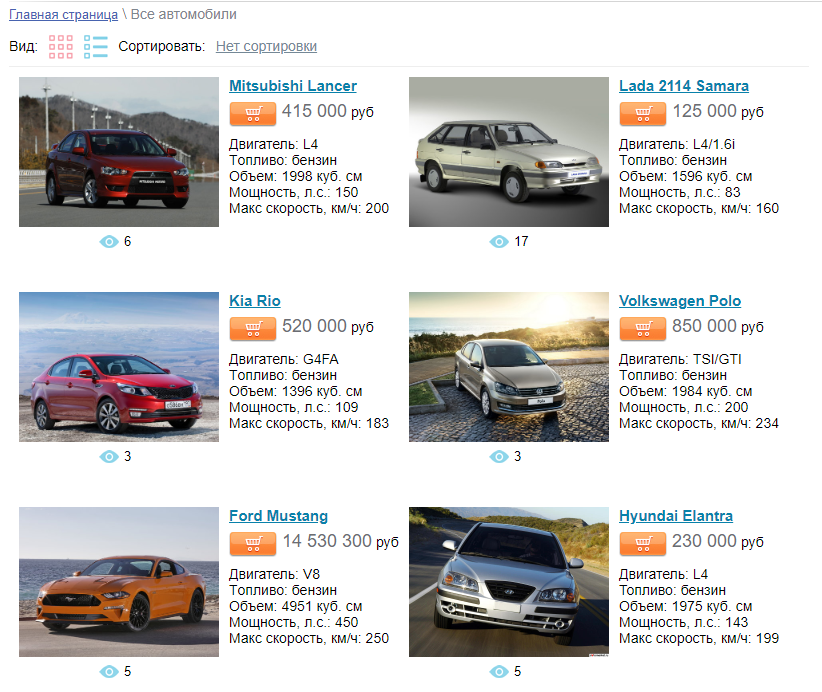


Рисунок 3.7 – Блок контента (сеткой) интернет-сайта «AutoSmart»

А еще можно сделать различный вывод контента, т.е. его порядок-сортировку данных (рисунок 3.8):

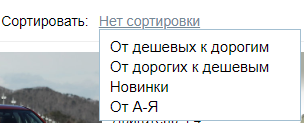


Рисунок 3.8 – Сортировка контента разрабатываемого интернет-сайта «AutoSmart»

1. **Правый SideBar (блок категории и блок новинок).**

Обычно SideBar называют правую или левую колонку интернет-сайта, которая размещается справа или слева от области основного контента. Иногда на интернет - сайте могут располагаться сразу два SideBara (Один – слева, другой - справа).

Как правило, наполнение SideBar не изменяется от одной страницы к другой, в отличии от содержимой области основного контента автомобилей (он может пополняться). Поэтому в SideBar обычно размешают блоки со ссылками на необходимую пользователю информацию, а также любую другую информацию. В SideBar разрабатываемого сайта входит:

* меню категорий автомобилей (block-category);
* Блок поиска автомобиля по параметрам (block-parameter)
* Блок новинок (block-news).

Ширина SideBar, четко фиксирована и не зависит от типа сайта (рисунок 3.9):

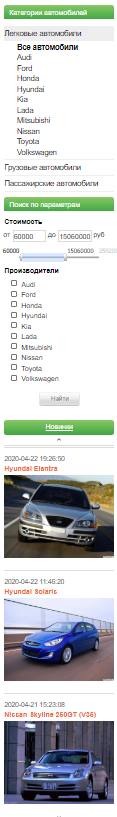


Рисунок 3.9 – Правый Блок моего сайта «AutoSmart»

1. **Нижний блок сайта (или footer).**

Слово «Footer» - подвал. Обычно футером называют часть Web-сайта, которая располагается в самом низу сайта, под всеми остальными блоками.

По аналогии с шапкой сайта, футер также имеет альбомную ориентацию.

Ширина футера, фиксированная и не меняется от изменения разрешения экрана пользователя (рисунок 3.10).

В блоке footer размещены:

* надпись © COPYRIGHT, 2020;
* ссылка на главную страницу авто сайта;
* ссылка на страницу “О Компании”;
* ссылка на форму “Обратная связь”

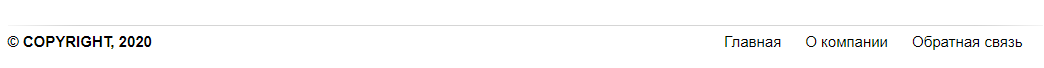
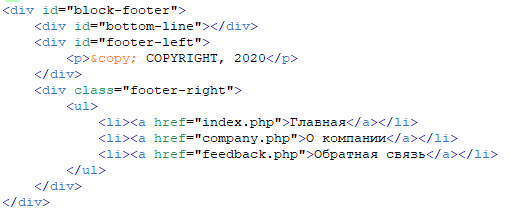


Рисунок 3.10 – Footer разрабатываемого интернет-сайта «AutoSmart»

Программная реализация (Рисунок 3.11):



# Проектирование и разработка панели администратора Web-сайта

Панель администратора для любого интернет сайта, является важной и неотъемлемой частью. Будь то интернет-магазин или просто любой информационный ресурс. Ведь благодаря панели администратора можно делать много разных вещей на своем сайте, например, добавлять товары, услуги или просто новости на сайт и удалять их же, а также редактировать привилегии администраторов.

В моем случае, на моем сайте есть панель администратора, которая наделена следующими функциями:

* просмотр заказов
* подтверждение заказов
* удаление заказов
* добавление автомобилей в базу
* редактирование информации об автомобиле
* удаление автомобилей из базы
* просмотр клиентов
* удаление клиентов
* добавление категорий автомобилей
* удаление категорий автомобилей
* просмотр администраторов

Для входа в панель администратора мной был выбран следующий дизайн (Рисунок 3.11):

**1 Страница (admin/login.php)**



Рисунок 3.11 – Блок для авторизации администратора

Для этой страницы был создан только 1 блок для авторизации администратора.

Выглядит он вот так (рисунок 3.12):

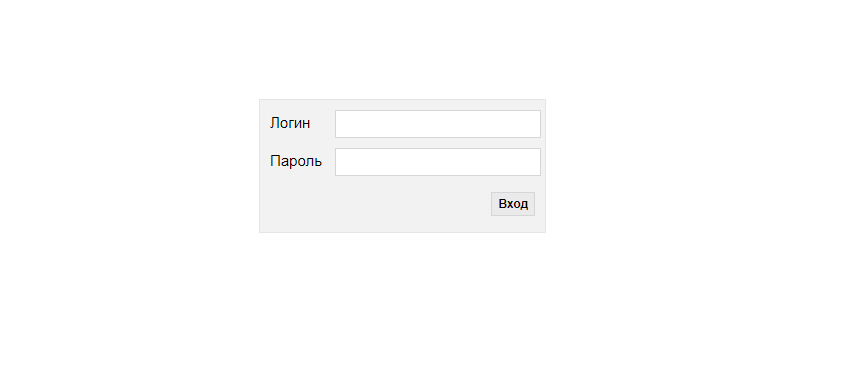


Рисунок 3.12 – Форма входа в Панель Администратора

1. **Страница (admin/index.php)**

Для панели администратора мной был выбран следующий дизайн (Рисунок 3.12):

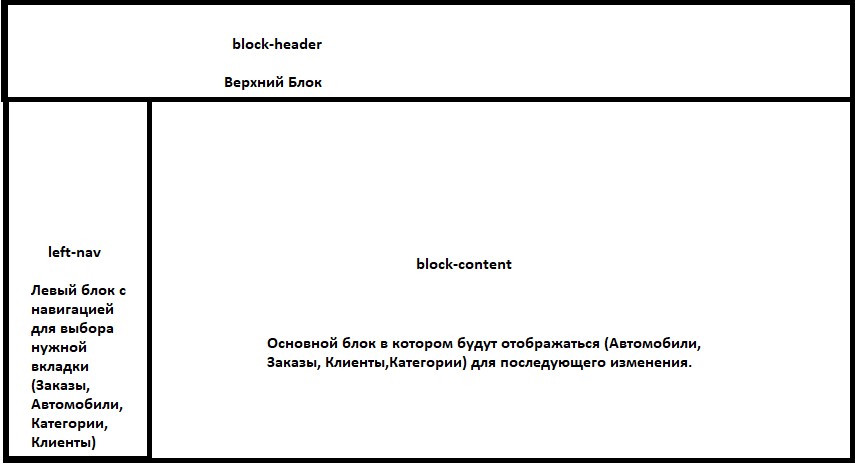
****

Рисунок 3.12 – Дизайн панели администратора

Для Панели Администратора мной было принято решение сделать (**3 блока**), каждый из которых будет выполнять свои определенные функции:

1. **Верхний Блок (block-header)**

Верхний блок — это шапка Панели Администратора, заголовок сайта, хедер от англ. header. Место, в котором обычно находится хедер - верхняя часть страницы. Ориентация хедера панели управления – альбомная (Рисунок 3.13).

В данном блоке размещается:

* Название;
* ссылка на главную страницу панели администратора;
* кнопка выход;
* отображение вашей роли на сайте.



Рисунок 3.13 – Header Панели Управления

1. **Левый Блок с навигацией (left-nav)**

В левом блоке у меня расположено меню, в котором находятся вкладки для отслеживания актуальной информации по заказам, автомобилям, категориям и клиентам, выглядит оно следующим образом (Рисунок 3.14):

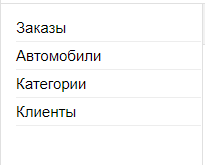


Рисунок 3.14 – Левое меню панели управления

1. **Блок с контентом (block-content)**

В блоке контента выводится вся необходимая информация администратору, будь то количество заказов или автомобилей, у меня при входе в панель администратора выводится общая статистика интернет-сайта, (Рисунок 3.15):



Рисунок 3.15 – Блок контента в панели администратора



Рисунок 3.16 - Вывод автомобилей интернет-сайта

# Описание разработанного сайта

# Описание интерфейса пользователя

Для начала работы с сайтом пользователю необходимо открыть браузер и в поисковой строке ввести название интернет-магазина (Рисунок 4.1):

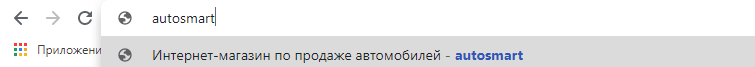


Рисунок 4.1 - Ввод названия интернет-магазина в браузере

После ввода адреса интернет-сайта пользователь попадает на главную страницу сайта (Рисунок 4.2):

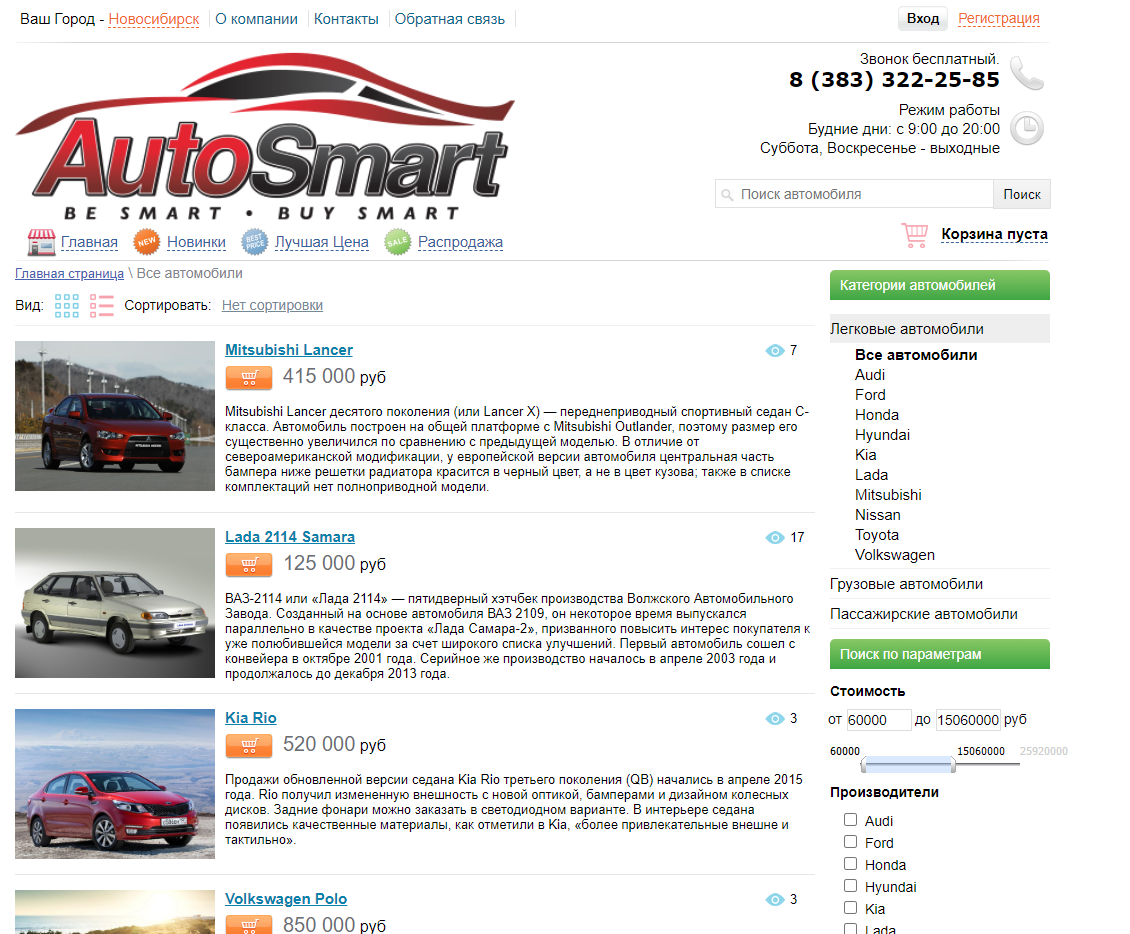


Рисунок 4.2 – Главная страница сайта

На главной странице сайта пользователь может просматривать доступные автомобили, которые есть в интернет-магазине “AutoSmart”. Для этого ему требуется выбрать в меню справа нужную категорию и при желании автомобиль (Рисунок 4.3). Так же есть возможность регистрации и входа (Рисунок 4.4) в свою учётную запись с возможностью редактирования информации о себе

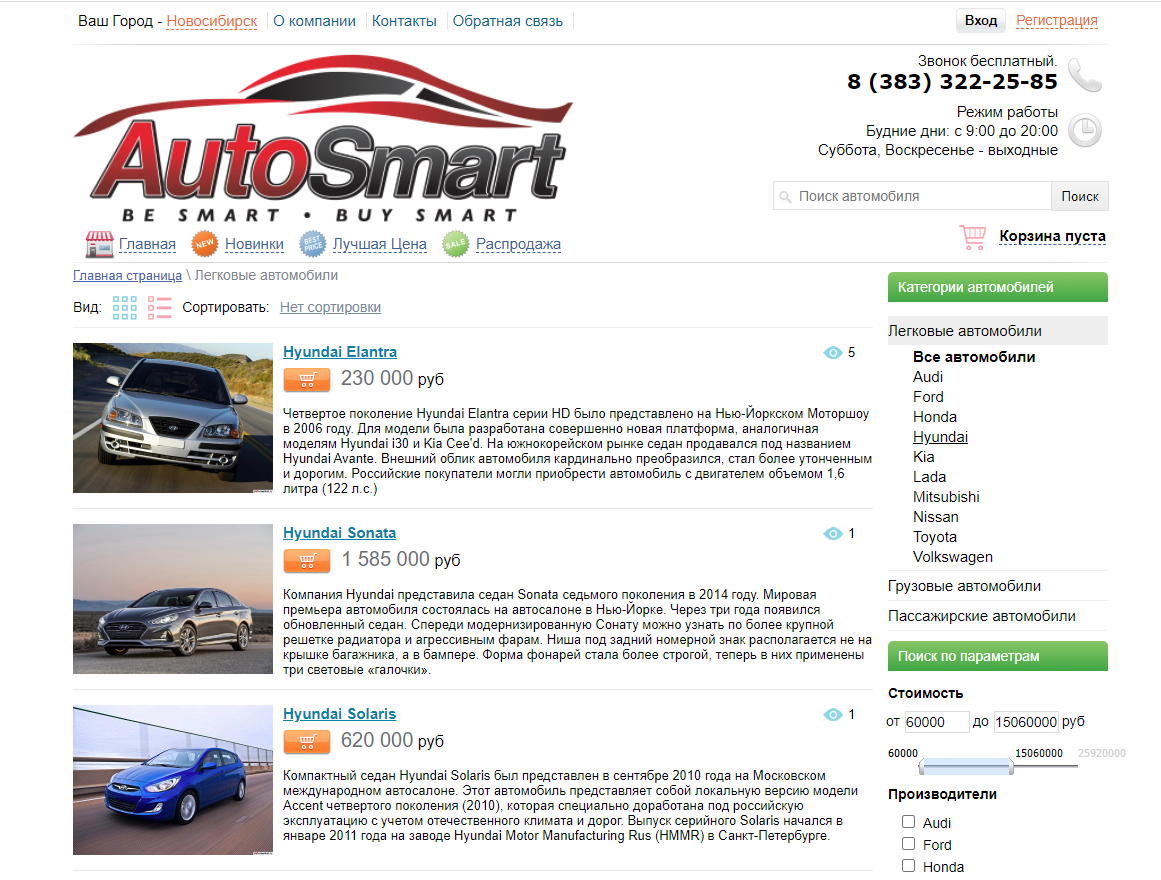


Рисунок 4.3 – Выбрал легковые автомобили марки Hyundai

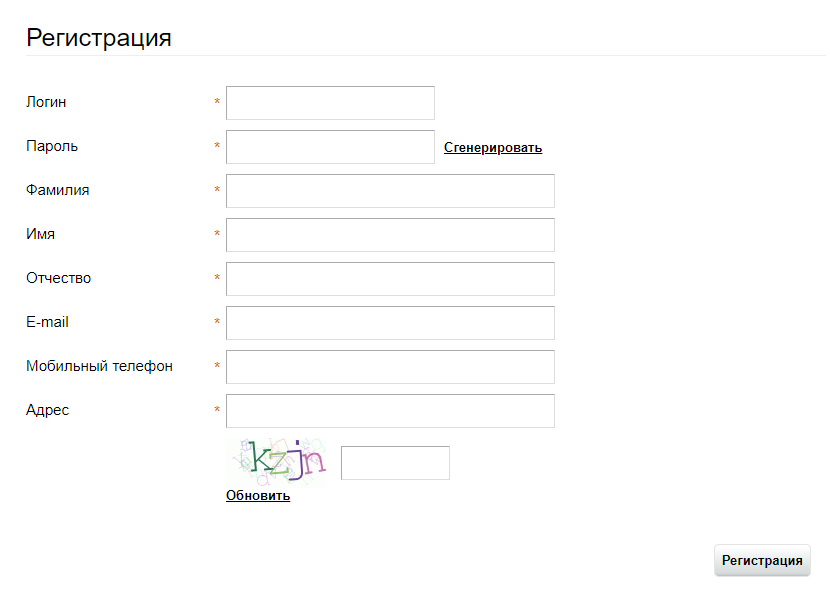
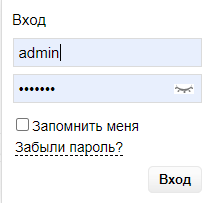
 

Рисунок 4.4 – Форма Регистрации и входа

Так же на главной странице пользователь может найти себе подходящий автомобиль, через поиск или поиск по параметрам, в моем случае можно задать диапазон цен и производителя автомобиля (Рисунок 4.5, 4.6):

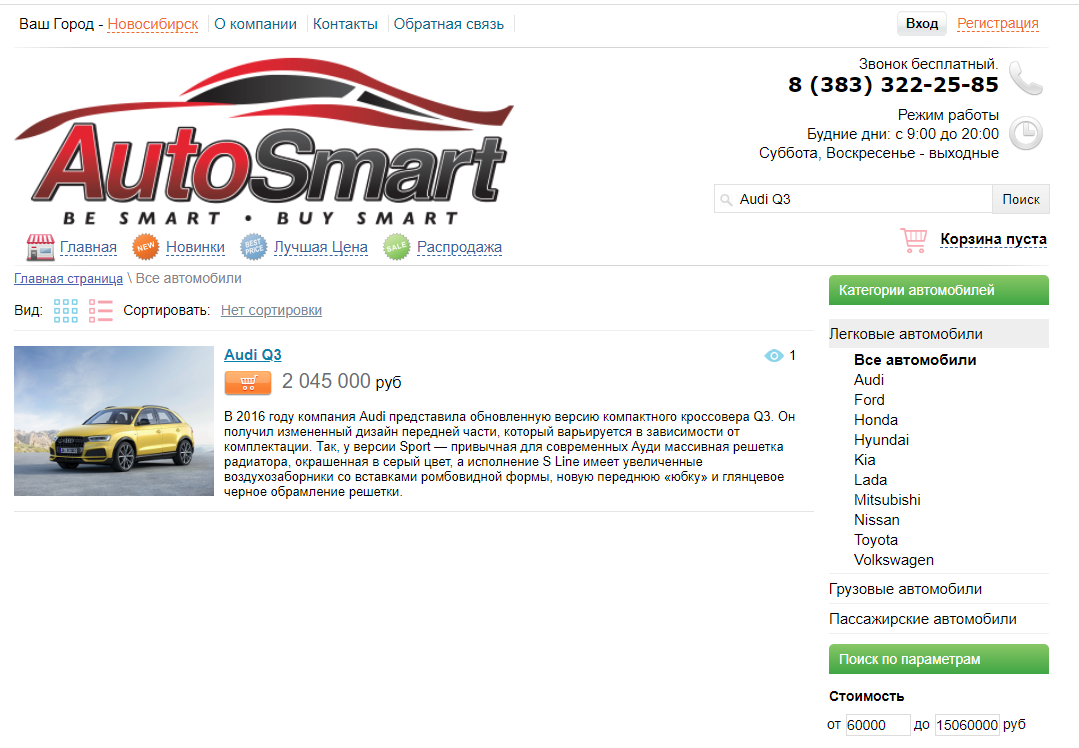


Рисунок 4.5 – Поиск Автомобиля через поиск

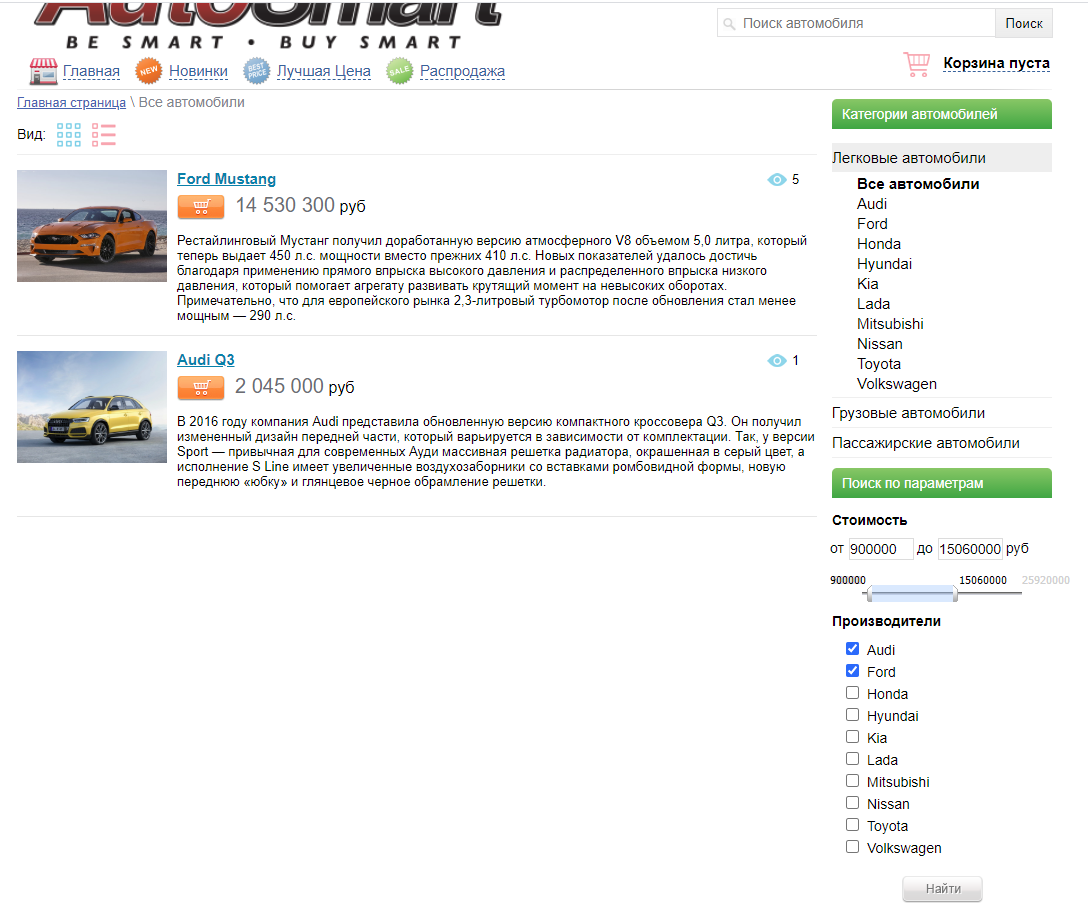


Рисунок 4.6 – Поиск Автомобиля по параметрам

При желании, пользователь может более подробно изучить понравившийся ему автомобиль нажав по его названию. После этого будет открыта страница просмотра автомобиля (view\_content.php) (Рисунок 4.7). Просмотр дополнительных фотографий автомобиля реализован с помощью готового плагина-галереи (jquery.fancybox.js) (Рисунок 4.8)

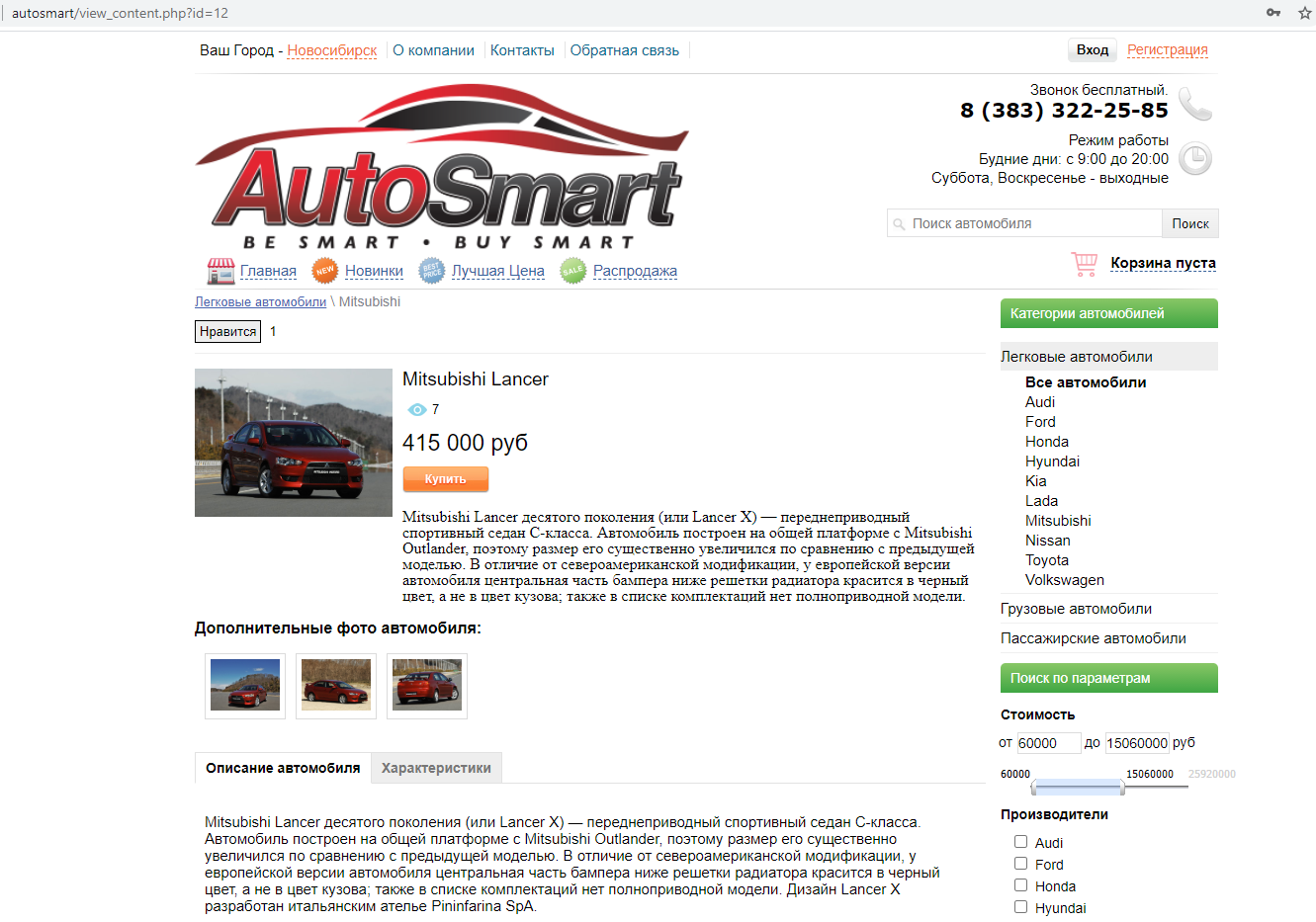


Рисунок 4.7 – Страница просмотра автомобиля



Рисунок 4.8 – Галерея автомобиля

Пользователь также может добавить понравившийся автомобиль в корзину путем нажатия соответствующей кнопки рядом с каждым автомобилем. После этого можно будет перейти в корзину и оплатить автомобиль (Рисунок 4.8). Также пользователь может попасть в личную покупательскую корзину путем нажатия соответствующей кнопки на навигационной панели (Рисунок 4.9).

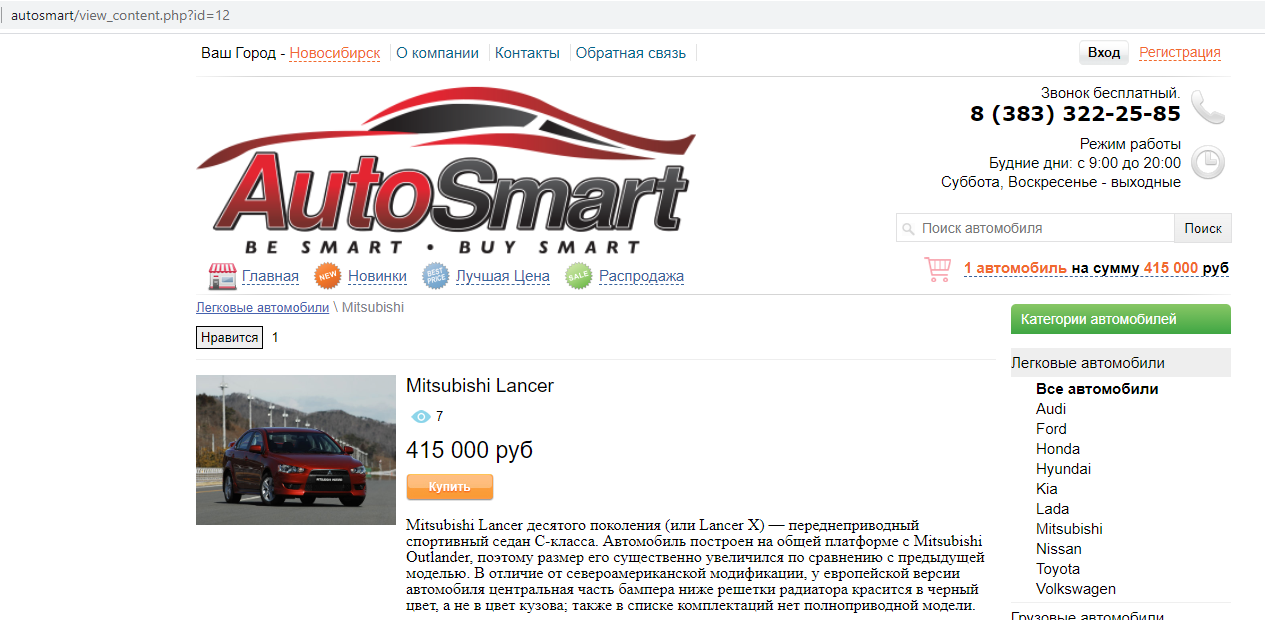


Рисунок 4.9 – Нажатие кнопки купить и отображение корзины

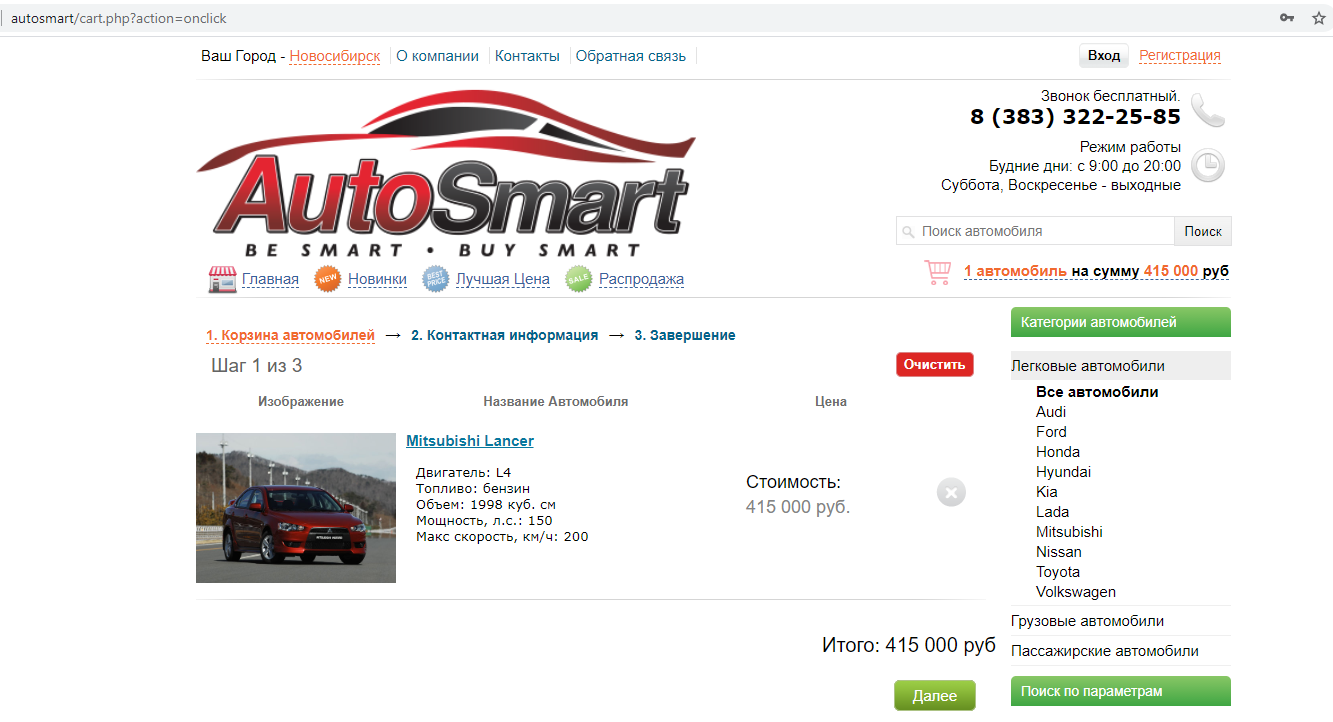


Рисунок 4.10 – Корзина

Также вверху и внизу сайта продублированы вкладки (Рисунок 4.11): О компании (Рисунок 4.12), Контакты (Рисунок 4.13) и Обратная связь (Рисунок 4.14)

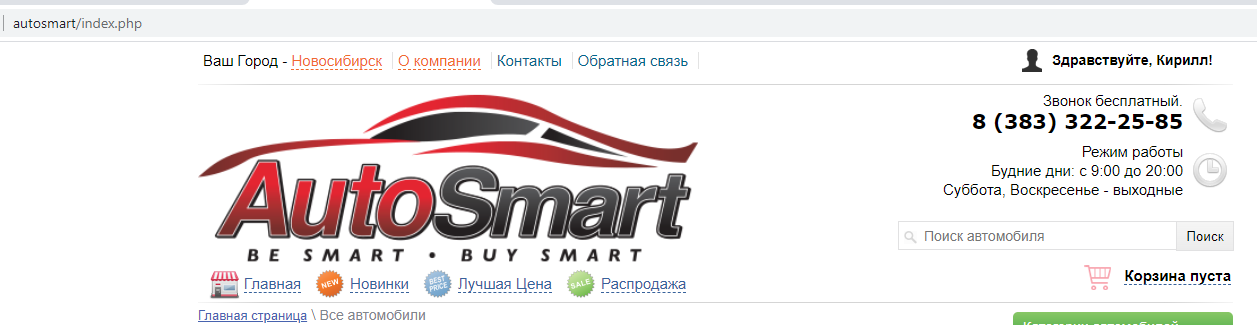




Рисунок 4.11 – Вкладки сверху и внизу

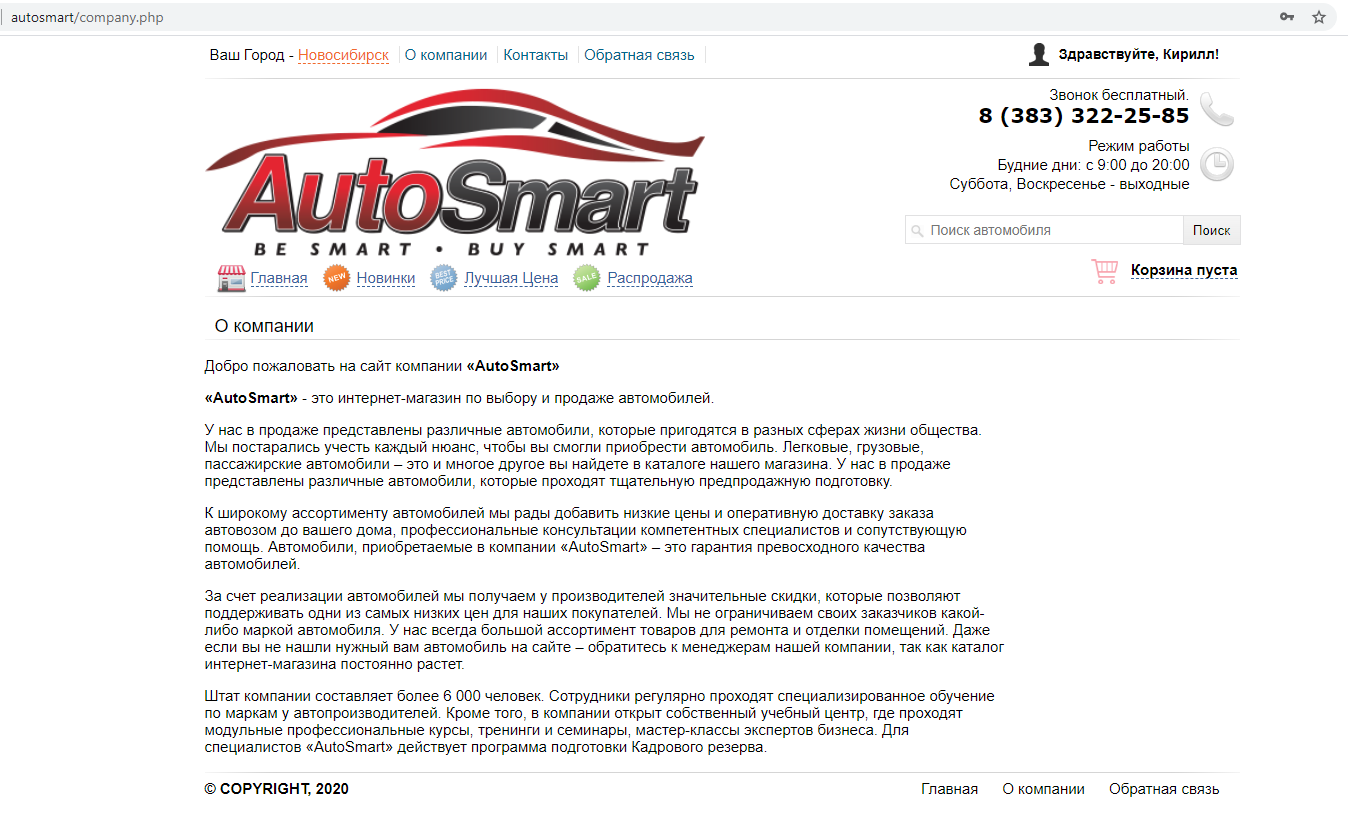


Рисунок 4.12 – Вкладка “О компании”

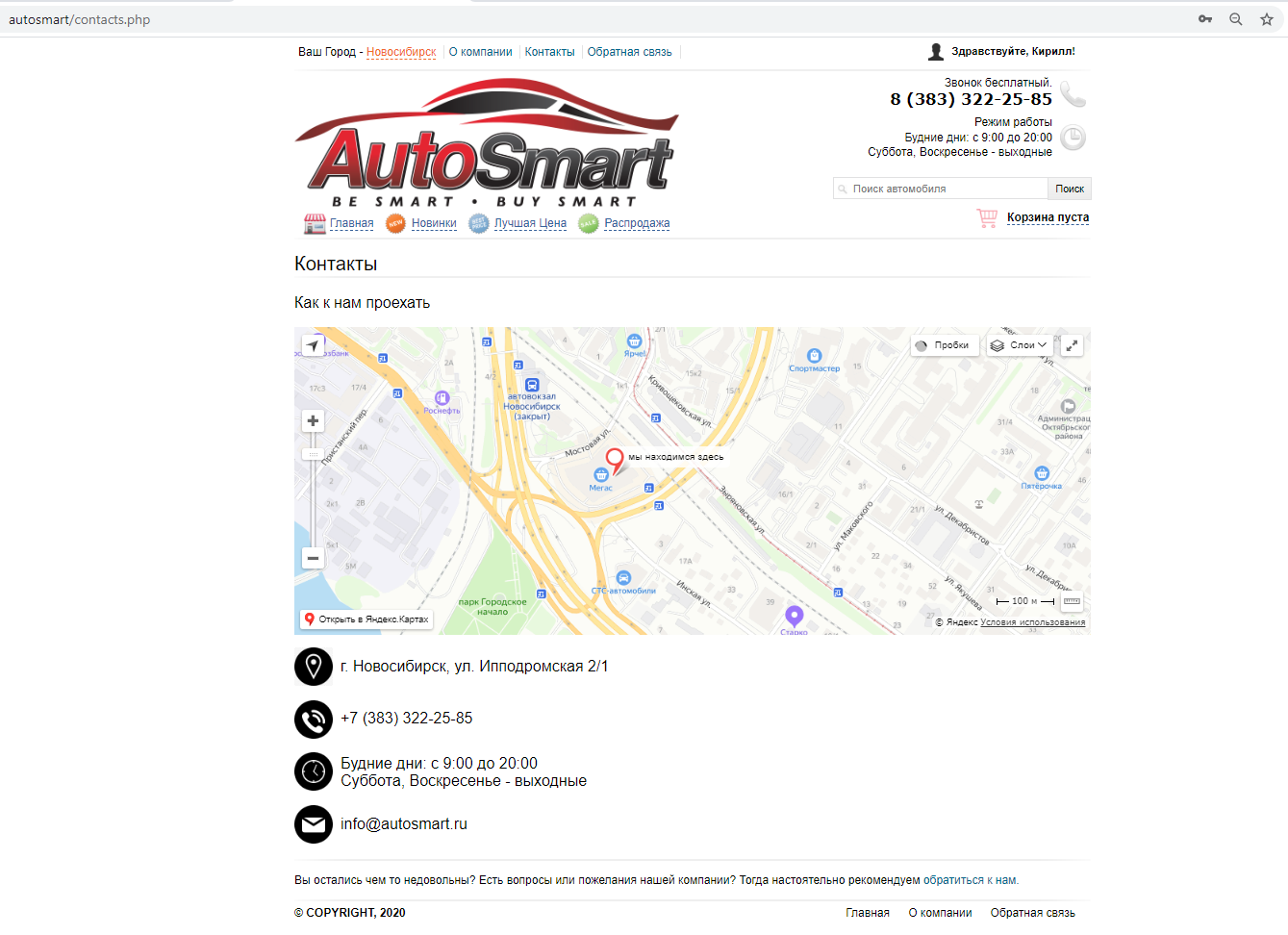


Рисунок 4.13 – Вкладка “Контакты”

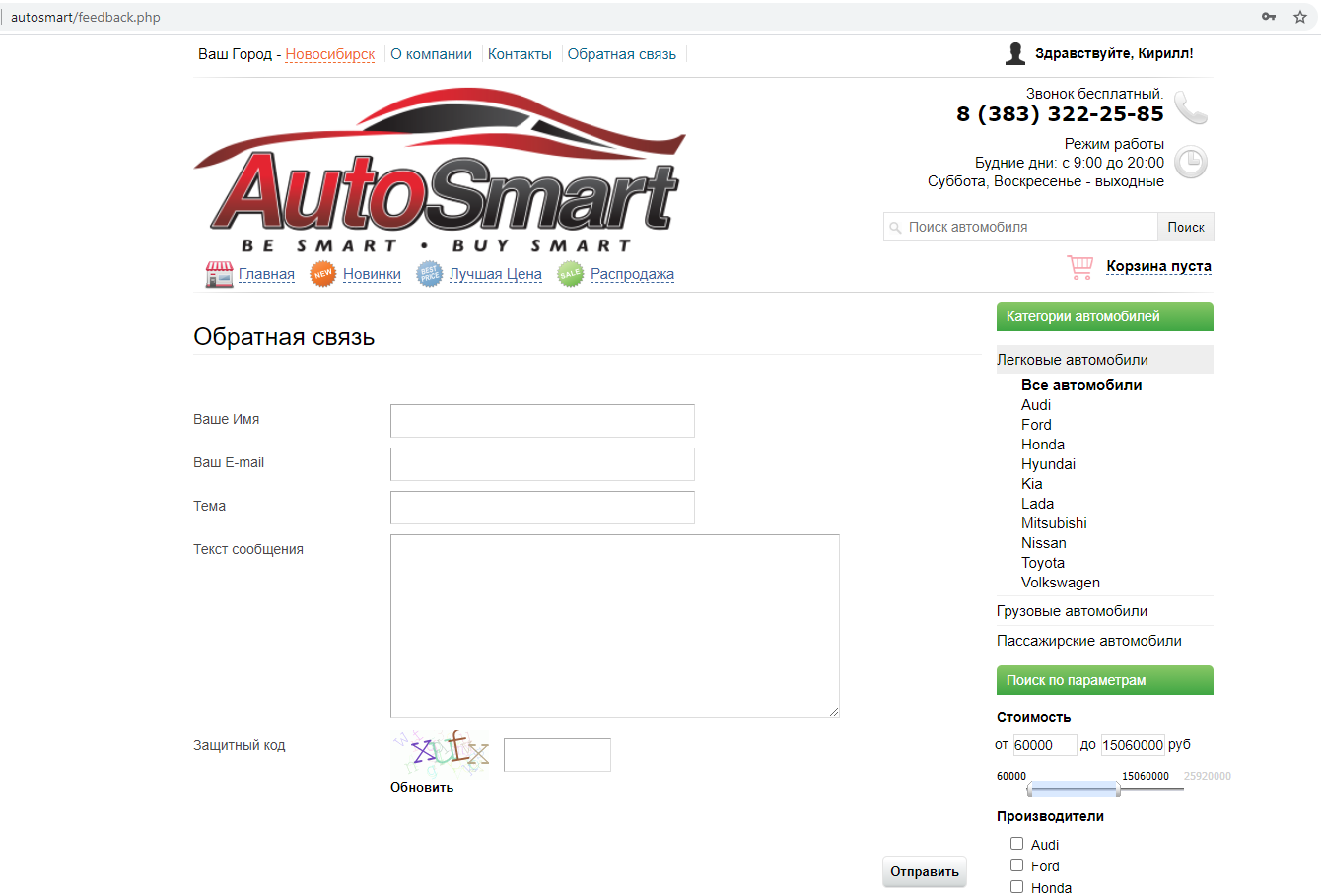


Рисунок 4.13 – Вкладка “Обратная связь”

# Описание интерфейса администратора

Для начала работы с сайтом администратору необходимо открыть браузер и в командной строке ввести название интернет-магазина (см. рис. 4.14):

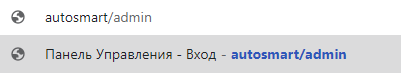


Рисунок 4.14. Ввод названия интернет-магазина в браузере

Перед началом работы администратор проходит авторизацию путем ввода логина и пароля для доступа к сайту (см. рис. 4.15):

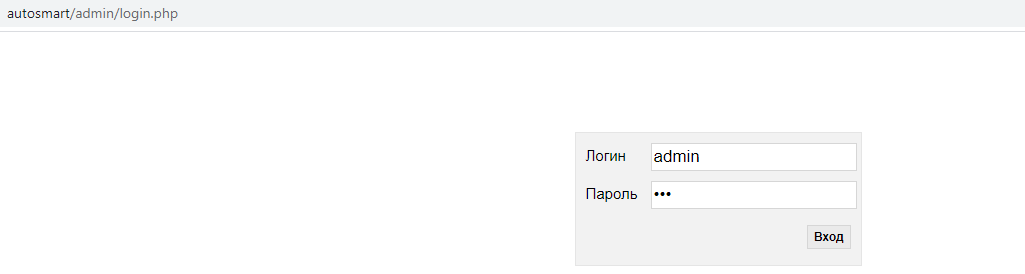


Рисунок 4.15. Форма для ввода логина и пароля

На данный момент на моем сайте зарегистрирован только один администратор, я сделал так чтобы Администратор с логином “admin” – был самым главным администратором сайта, он наделен особыми привилегиями такими как, удаление/добавление новых администраторов с различным уровнем доступа к данным сайта, а также остальными правами администратора, которые существуют (Рисунок 4.16.).

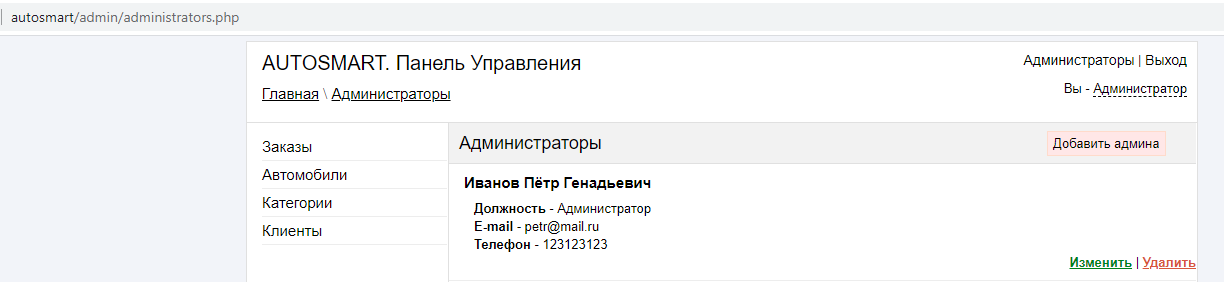


Рисунок 4.16. Страница удаления и добавления администраторов

На моем сайте администратор может выполнять множество функций, таких как:

* просмотр заказов
* подтверждение заказов
* удаление заказов
* добавление автомобилей в базу
* редактирование информации об автомобиле
* удаление автомобилей из базы
* просмотр клиентов
* удаление клиентов
* добавление категорий автомобилей
* удаление категорий автомобилей
* просмотр администраторов

# 

Вот некоторые примеры страниц, панели администратора (Рисунок 4.17), (Рисунок 4.18), (Рисунок 4.19):

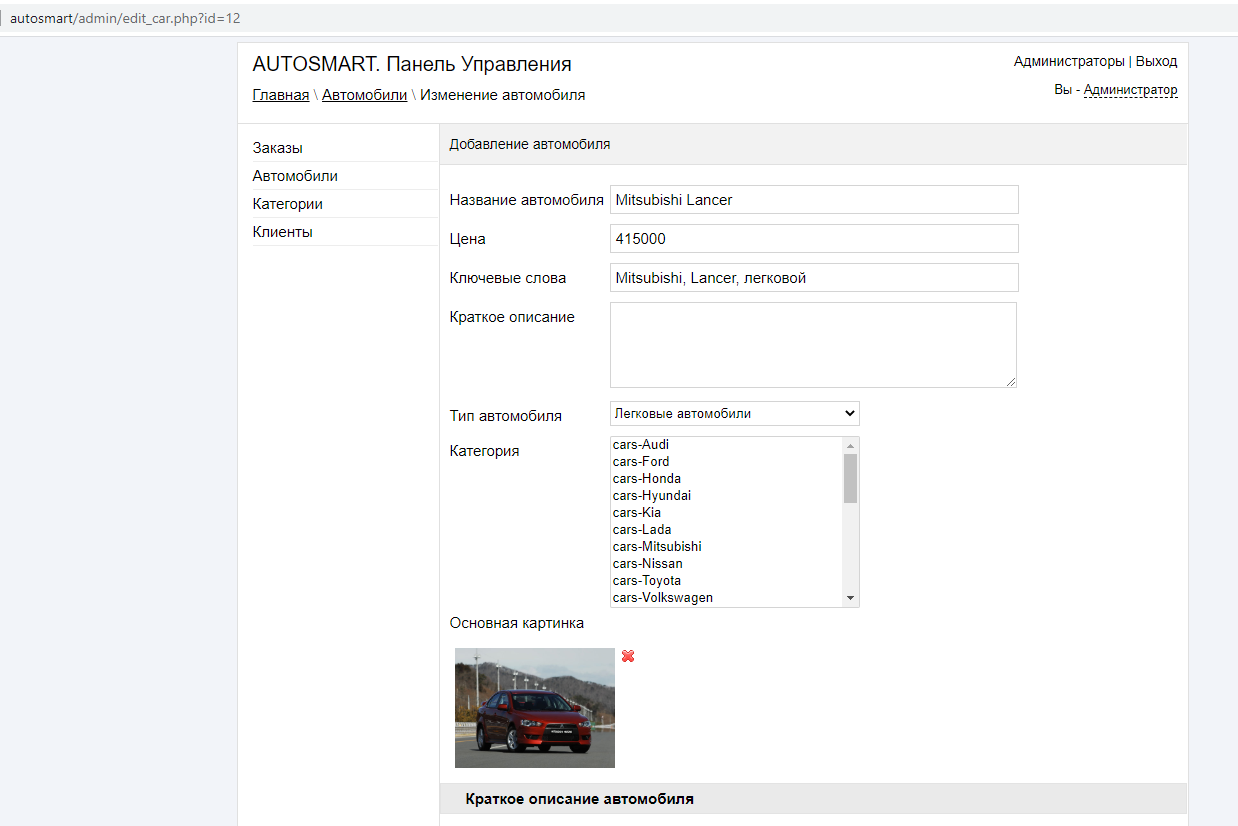


Рисунок 4.17 - Страница изменения информации об автомобиле

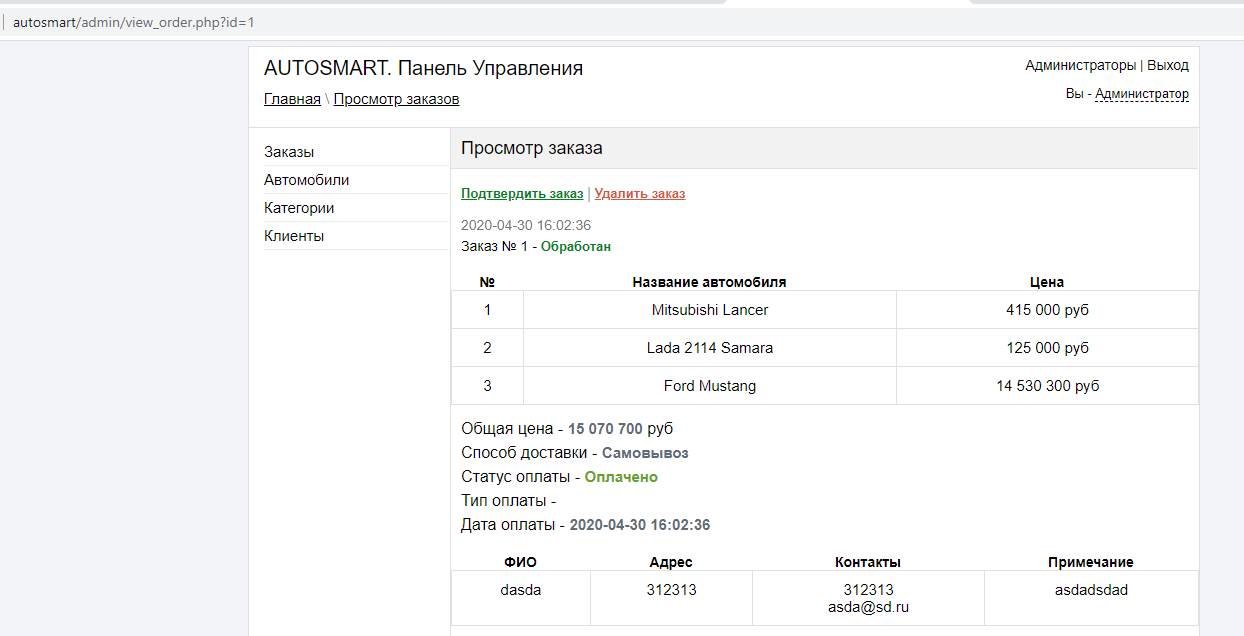


Рисунок 4.18 - Страница просмотра содержимого заказа

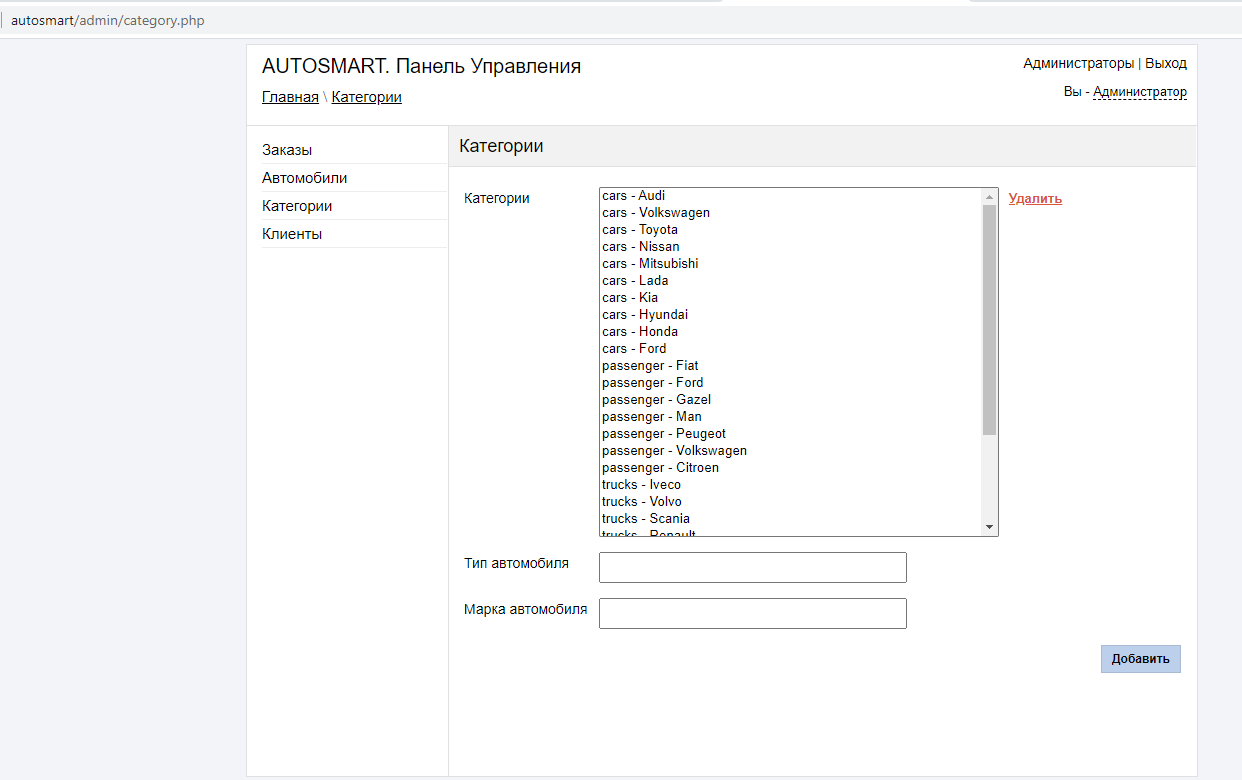


Рисунок 4.19 - Страница добавления/удаления категории

# Заключение

В настоящее время Интернет стал одним из ведущих инструментов ведения бизнеса. Это объясняется как популярностью Интернета, так и его преимуществами для ведения коммерческой деятельности. Присутствие торговой компании в Интернете необходимо для успешной конкурентной борьбы в современных условиях.

Моя бакалаврская работа показала актуальность развития электронного бизнеса в современных условиях, наметила пути открытия такого бизнеса или перехода к нему.

Представление каталога автомобилей в Интернете приведет к увеличению продаж через традиционный розничный салон автокомпании. Это связано с тем, что значительная часть посетителей выбирает товары в Интернет-магазинах, но делает покупки в традиционных магазинах. При этом чаще всего покупатель выбирает тот магазин, где нашел нужный товар.

Описаны используемые языки программирования, а также система управления базами данных, которая использовалась для построения базы данных интернет-магазина. База данных создавалась в среде MySQL. Разработана автоматизированная информационная система, обеспечивающая работу интернет-магазина, представленная в виде сайта. Разработанный интернет-сайт будет доступен всем пользователям сети Интернет после его размещения в Интернете, при условии прохождения процесса регистрации в системе. С его помощью пользователи смогут получать нужную информацию и покупать автомобили, онлайн не выходя из дома.

Система реализована с помощью среды PHP с использованием языка гипертекстовой разметки HTML и JavaScript. В качестве СУБД соответственно выбран MySQL.

Разработанный сайт был протестирован локально. Полученные результаты соответствовали прогнозируемым. Были произведены проверки вероятных ошибок и недочетов – система функционирует надежно, без перебоев.

Разработанный сайт удовлетворяет всем требованиям, установленным на этапе постановки задачи. При создании веб-сайта Интернет-магазина были разработаны модули аутентификации, заказа товара, покупки товара, обработки автомобилей (добавление, редактирование и удаление), а также поиск по сайту. Данные модули были успешно внедрены в структуру веб-сайта.

В ближайшем будущем нужно решить следующие задачи:

* получить отзывы клиентов, об измененном процессе сотрудничества с ними.
* удостоверится в получаемой эффективности от внедрения интернет-сайта по выбору и продаже автомобилей. Причем как от прогнозируемого уменьшения затрат на ведение переговоров с клиентами, так и о возросшей эффективности продажи автомобилей с использованием интернет технологий.
* на основании полученных отзывов произвести модернизацию интернет-сайта “AutoSmart”. И инвестировать средства в дальнейшее развитие данного проекта до полноценного корпоративного web-портала компании “AutoSmart”.
* Разработать приложение, позволяющее функционировать сайту на всех смартфонах с максимальной скоростью;
* Внедрить на сайт систему оплаты различными платежными системами, такими как VISA/ Master Card/ Мир/Яндекс деньги.

# Список использованных источников

1. Web-дизайн с нуля! (+ CD-ROM) / П.П. Константинов и др. - М.: Лучшие книги, **2015**. - 304 c.
2. Дронов, В. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов / В. Дронов. - М.: БХВ-Петербург, 2014. - **182** c.
3. Дунаев Вадим. (Х)HTML, скрипты и стили. Автор: Дунаев Вадим Год: 2009 Издание: БХВ-Петербург. Flash MX 2004. Самоучитель.
4. Лучанинов Д.В. Основы разработки web-сайтов образовательного назначения [Электронный ресурс]: учебное пособие/ Лучанинов Д.В.— Электрон. текстовые данные.— Саратов: Ай Пи Эр Медиа, 2018.— 105 c.
5. Дунаев В. В. Самоучитель JavaScript. - СПб. и др.: Питер, 2013
6. Фролов А.Б. Web-сайт. Разработка, создание, сопровождение [Электронный ресурс]: учебное пособие/ Фролов А.Б., Нагаева И.А., Кузнецов И.А.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2020.— 355 c.
7. Богун В.В. Web-программирование. Интерактивность статических Интернет-сайтов с применением форм [Электронный ресурс]: учебное пособие для СПО/Богун В.В.— Электрон. текстовые данные.— Саратов: Профобразование, Ай Пи Ар Медиа, 2020.— 65с.
8. Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites Третье издание Автор: Робин Никсон Издательство: Питер: 2015.
9. Балабанов И.Т. - «Торговля через виртуальный магазин» «Электронная коммерция» 2004г. С.195-197
10. Лоусон Б., Шарп Р. Л81 Изучаем HTML5. Библиотека специалиста. 2-е изд. — СПб.: Питер, 2012. — 304 с.: ил.
11. AJAX. Программирование для Интернета — Елена Бенкен, Геннадий Самков.<http://progbook.ru/ajax/1201-benken-ajax-programmirovanie-dlya-interneta.html>