



МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ С.Ю. ВИТТЕ

Факультет среднего профессионального образования (Колледж)
Специальность 09.02.05 Прикладная информатика (по отраслям)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА/ ДИПЛОМНАЯ РАБОТА

Студента Кривошеева Данила Дмитриевича
(фамилия, имя, отчество)

на тему: Разработка персонального сайта автомобильного салона с
применением PHP и MySQL
(тема работы)

по ПМ.2 Разработка, внедрение и адаптация программного обеспечения
отраслевой направленности

Руководитель работы Жукова Марина Александровна
(подпись) (ученая степень, звание, инициалы и фамилия)

Студент _____
(Подпись)
«17» декабря 2018 г.

Штрих-код

Москва, 2019 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	
1.1. Анализ предметной области.....	7
1.2. Анализ существующей информационной системы	8
ГЛАВА 2. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	
2.1. Анализ и выбор методологии проектирования и модели	11
жизненного цикла программного продукта	
2.3. Описание общей структуры проекта.....	20
ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	
3.1. Программно-аппаратная платформа.....	22
3.2. Выбор web-сервера.....	23
3.3. Среда разработки для модулей проекта	28
3.4. Обзор сред для разработки web-страниц	31
3.5. Реализация программного продукта	34
ГЛАВА 4. ОЦЕНКА РАЗРАБОТАННОГО ПРОГРАММНОГО ПРОДУКТА	
4.1. Критерии оценки качества.....	35
4.2. Тестирование разработанного программного обеспечения	36
ЗАКЛЮЧЕНИЕ.....	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39

ВВЕДЕНИЕ

В современном мире все уже практически не встретить дома, в котором нет персонального компьютера. Компьютер в настоящее время играет роль усилителя интеллектуальных возможностей человека и общества в целом, а коммуникационные средства, использующие компьютеры, служат для связи и передачи информации. Появление и развитие компьютеров и информационных технологий – стало для всего человечества практически нормой.

Многие люди уже не могут представить свою жизнь без персонального компьютера. Так как он большие возможности в общении или коллективном развлечении без отрыва от удобного домашнего кресла. Все большую популярность в этом мире приобретают информационные сайты, в которых можно найти свою информацию, которую вы можете повстречать на просторах онлайн территории.

Тенденция к информатизации наблюдается сейчас практически везде:

- в образовательных учреждениях;
- в средствах массовой информации;
- в правительственных организациях;
- в различные развлекательных заведениях;
- в промышленности;
- в оборонно-промышленном комплексе;
- в научно исследовательской деятельности;
- в средствах телекоммуникаций;
- в строительстве;
- в освоение водных, космических, подземных просторов, земли и

во многих других сферах деятельности человечества.

Не являются исключением и различные автосалоны, для которых информационное развитие на сегодняшний день не менее важно, чем для коммерческих организаций. С помощью современных технологий автосалоны могут заявить о себе, могут общаться с дилерами, делиться опытом в продаже продукции.

Мы живем в эпоху интернета, поэтому чтобы, какой либо автосалон собрал большую аудиторию надо разместить ее в интернете. В котором рассказывается о самой игре и ее возможностях.

Для увеличения аудитории автолюбителей, предполагается внедрить web-сайт, содержащий web-форум. На данный момент в Москве очень бурно развиваются web-технологии, для многих потребителей становится проще найти нужную информацию, в интернете, нежели искать ее в газете или ждать рекламы по телевидению. Так же для всех потенциальных покупателей будет доступна услуга web-форума, в котором пользователи смогут обмениваться своим опытом, советами и различными советами по автотематике.

Актуальность работы. В настоящее время в Москве Интернет как средство получения и обмена информации, завоевывает все большую популярность. Для многих пользователей Интернет является удобным, простым и быстрым средством для получения информации по автомиру.

Цель работы. Создание web-сайта по автосалону «Автосалон DANILA». Данный сайт должен обеспечивать поиск необходимой информации по сайту, получение ответа на свои вопросы от администратора сайта.

Для достижения поставленной цели были поставлены следующие задачи:

- изучить данные, которые предстоит вынести на web-сайт «Автосалон DANILA», создание web-форума по данной тематике;

- построить модель web-сайта;
- проанализировать методы решения поставленной цели;
- спроектировать web-интерфейс и web-форум и проанализировать взаимодействие между ними;
- выбрать средства и технологии разработки.

Объект и предмет исследования. Объектом исследования является структура работы автосалона. Предметом является создание программного комплекса, обеспечивающего работу web-сайта.

Методы исследования. Для решения поставленных задач были применены следующие общепринятые методы исследования: теоретические (анализ литературы по проблеме исследования, изучение технологий программирования и программных средств, необходимых для создания системы); специальные методы (анализ предметной области, проектирование и реализация системы).

Гипотеза исследования. Внедрение web-сайта «Автосалон DANILA» позволит расширить круг заинтересованных лиц продукции автосалона, упростить поиск интересующей информации для пользователя, повысить авторитет данного автосалона.

Практическая значимость. Создание и внедрение web-сайта позволит расширить круг заинтересованных лиц, автолюбителей, упростить пользователям поиск интересующей их информации, повысить рейтинг автосалона.

Модифицируемость. Подразумевает, что информационная система должна быть легко модифицируемой, что просто необходимо для информационной системы. Например, при появлении новой модели автомобиля, ее обязательно нужно добавить в информационную систему, а это невозможно без модифицируемости системы.

Новая информационная система будет создаваться на основе существующей информационной системы «Автосалон DANILA», с учетом выведенных требований и предполагаемых усовершенствований существующей информационной системы.

Усовершенствование существующего сайта будет направлено на внедрение технологий, с помощью которых будет производиться постоянное обеспечение пользователей новой информацией.

Структура и объем работы. Состоит из введения, четырех разделов, заключения.

ГЛАВА 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Анализ предметной области

Для успешного создания работоспособной, отказоустойчивой и удобной информационной системы необходимо тщательное изучение предметной области. Этап описания предметной области является важнейшим этапом при создании любой информационной системы. От полноты и качества проводимого исследования зависит судьба создаваемого программного продукта, так как именно на этом этапе выделяется круг автоматизируемых задач и закладываются требования к функциональности системы.

Для исследования предметной области Web-сайта, необходимо подробно рассмотреть структуру работы автосалона.

Функции сайта.

Информационная система - такая форма рекламы продукции, при которой просмотр и изучение товаров осуществляется через компьютерные сети. При этом в качестве потенциальных покупателей товаров (или услуг) могут выступать как частные лица, так и организации.

Глобальная сеть Internet сделала информационную систему доступной для фирм любого масштаба. Если раньше организация электронного обмена данными требовала заметных вложений в коммуникационную инфраструктуру и была доступна лишь крупным компаниям, то использование Internet позволяет сегодня вступить в ряды "электронных торговцев" и небольшим фирмам. Информационная система в WWW дает любой компании возможность привлекать клиентов со всего мира.

Под определение Информационная система подпадают системы, ориентированные на Интернет - «информационные сайты», речь о которых пойдет далее в курсовой работе. В то же время процедуры реклам, инициированных информацией из WWW, но использующих для обмена данными факс, телефон и пр., могут быть лишь частично отнесены к классу Информационных систем.

Отметим также, что, несмотря на то, что WWW является технологической базой Информационной системы, в ряде систем используются и другие коммуникационные возможности. Так, запросы к продавцу для уточнения параметров товара или для отзывов о нем могут быть посланы и через электронную почту.

1.2. Анализ существующей информационной системы

На данный момент в организации «Автосалон DANILA» существует информационная система (ИС), которая позволяет упростить движение информационных потоков внутри организации, и при взаимодействии с клиентами. Функциональность данной системы обеспечивается наличием: внутренней локальной компьютерной сети, выделенным каналом связи.

Информационная система позволяет:

- поддерживать контакт с другими организациями/клиентами по телефону;
- обеспечивать работу электронной почты организации и выход в Интернет.

Недостатки: при наработанной и автоматизированной информационной системе внутри организации, информативная работа с клиентами проводится только в самой организации.

Предполагаемые усовершенствования существующей информационной системы

В ближайшее время усовершенствование существующей ИС будет направлено на внедрение Интернет технологий, с помощью которых будет производиться поиск новых клиентов.

Первым шагом предполагается создание и внедрение web-сайта «Автосалон DANILA», на котором разместится кроме информации об организации и форум, в который заносятся отзывы пользователей, впоследствии обрабатываемые.

Функции сайта организации.

Данный сайт является интерактивной системой обеспечивающей следующие функции:

1. Информативная функция: сайт позволяет оперативно сравнивать необходимые товары и обеспечивает простоту поиска.
2. Практическая функция: web-сайт также ускоряет процесс написания сообщения и предъявление его на исполнение администратору.

Функции организации сайта.

Данный web-сайт является информационной системой обеспечивающей следующие функции:

1. Информативная функция: web-сайт обеспечивает доступное получение информации об автосалоне «Автосалон DANILA».
2. Практическая функция: web-сайт позволяет пользователю получить всю необходимую информацию о автосалоне и его продукции.

ВЫВОДЫ

Анализ автосалона позволил определить задачи для дополнения существующей информационной системы с помощью интернет-технологий.

Были выделены правила для автосалона и внедрения новых информационных технологий, таких как web-сайт.

Анализ существующей информационной системы позволил выявить ее положительные и отрицательные стороны для внесения положительных изменений и дополнений, не приводя к громоздкости или отказу работоспособности системы.

ГЛАВА 2. АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1. Анализ и выбор методологии проектирования и модели жизненного цикла программного продукта

При проектировании любого программного продукта важно учитывать, что от правильного проектирования программного обеспечения зависит качество его функционирования. При проектировании важно правильно выбрать методологию, по которой будет создаваться программный продукт, модель жизненного цикла, и тщательно разработать его общую архитектуру.

Были рассмотрены многие методологии, но описанию подлежит только методология RAD. Одним из возможных подходов к разработке программного обеспечения в рамках спиральной модели жизненного цикла является получившая в последнее время широкое распространение методология быстрой разработки приложений RAD (Rapid Application Development). Под этим термином обычно понимается процесс разработки программного обеспечения, содержащего 3 элемента:

- небольшую команду программистов (от 2 до 10 человек);
- короткий, но тщательно проработанный производственный график;
- повторяющийся цикл, при котором разработчики, по мере того, как приложение начинает обретать форму, запрашивают и реализуют в продукте требования, полученные через взаимодействие с заказчиком.

Команда разработчиков должна представлять из себя группу профессионалов, имеющих опыт в анализе, проектировании, генерации кода и тестировании программного обеспечения с использованием CASE-

средств. Члены коллектива должны также уметь трансформировать в рабочие прототипы предложения конечных пользователей.

Жизненный цикл программного обеспечения по методологии RAD состоит из трех фаз:

- фаза анализа и планирования требования;
- фаза проектирования;
- фаза построения.

На фазе анализа и планирования требований пользователи системы определяют функции, которые она должна выполнять, выделяют наиболее приоритетные из них, требующие проработки в первую очередь, описывают информационные потребности. Определение требований выполняется в основном силами пользователей под руководством специалистов-разработчиков. Ограничивается масштаб проекта, определяются временные рамки для каждой из последующих фаз. Кроме того, определяется сама возможность реализации данного проекта в установленных рамках финансирования, на данных аппаратных средствах и т.п. Результатом данной фазы должны быть список и приоритетность функций будущей информационной системы, предварительные функциональные и информационные модели информационной системы.

На фазе проектирования часть пользователей принимает участие в техническом проектировании системы под руководством специалистов-разработчиков. CASE-средства используются для быстрого получения работающих прототипов приложений. Пользователи, непосредственно взаимодействуя с ними, уточняют и дополняют требования к системе, которые не были выявлены на предыдущей фазе. Более подробно рассматриваются процессы системы. Анализируется и, при необходимости, корректируется функциональная модель. Каждый процесс рассматривается детально. При необходимости для каждого элементарного процесса

создается частичный прототип: экран, диалог, отчет, устраняющий неясности или неоднозначности. Определяются требования разграничения доступа к данным. На этой же фазе происходит определение набора необходимой документации.

После детального определения состава процессов оценивается количество функциональных элементов разрабатываемой системы и принимается решение о разделении ИС на подсистемы, поддающиеся реализации одной командой разработчиков за приемлемое для RAD-проектов время - порядка 60-90 дней. С использованием CASE-средств проект распределяется между различными командами (делится функциональная модель).

Результатом данной фазы должны быть:

- общая информационная модель системы;
- функциональные модели системы в целом;
- точно определенные с помощью CASE-средства интерфейсы

между автономно разрабатываемыми подсистемами.

Все модели и прототипы должны быть получены с применением тех CASE-средств, которые будут использоваться в дальнейшем при построении системы. Данное требование вызвано тем, что в традиционном подходе при передаче информации о проекте с этапа на этап может произойти фактически неконтролируемое искажение данных. Применение единой среды хранения информации о проекте позволяет избежать этой опасности.

В отличие от традиционного подхода, при котором использовались специфические средства прототипирования, не предназначенные для построения реальных приложений, а прототипы выбрасывались после того, как выполняли задачу устранения неясностей в проекте, в подходе RAD

каждый прототип развивается в часть будущей системы. Таким образом, на следующую фазу передается более полная и полезная информация.

На фазе построения выполняется непосредственно сама быстрая разработка приложения. На данной фазе разработчики производят итеративное построение реальной системы на основе полученных в предыдущей фазе моделей, а также требований нефункционального характера. Программный код частично формируется при помощи автоматических генераторов, получающих информацию непосредственно из репозитория CASE-средств. Конечные пользователи на этой фазе оценивают получаемые результаты и вносят коррективы, если в процессе разработки система перестает удовлетворять определенным ранее требованиям. Тестирование системы осуществляется непосредственно в процессе разработки.

После окончания работ каждой отдельной команды разработчиков производится постепенная интеграция данной части системы с остальными, формируется полный программный код, выполняется тестирование совместной работы данной части приложения с остальными, а затем тестирование системы в целом. Завершается физическое проектирование системы:

- определяется необходимость распределения данных;
- производится анализ использования данных;
- определяются требования к аппаратным ресурсам
- определяются способы увеличения производительности
- завершается обработка документации проекта.

Результатом фазы является готовая система, удовлетворяющая всем согласованным требованиям.

На фазе внедрения производится обучение пользователей, организационные изменения и параллельно с внедрением новой системы

осуществляется работа с существующей системой (до полного внедрения новой). Так как фаза построения достаточно непродолжительна, планирование и подготовка к внедрению должны начинаться заранее, как правило, на этапе проектирования системы. Приведенная схема разработки информационной система не является абсолютной. Возможны различные варианты, зависящие, например, от начальных условий, в которых ведется разработка: разрабатывается совершенно новая система; уже было проведено обследование организации и существует модель его деятельности; на организации уже существует некоторая информационная система, которая может быть использована в качестве начального прототипа или должна быть интегрирована с разрабатываемой.

Следует, однако, отметить, что методология RAD, как и любая другая, не может претендовать на универсальность, она хороша в первую очередь для относительно небольших проектов, разрабатываемых для конкретного заказчика. Если же разрабатывается типовая система, которая не является законченным продуктом, а представляет собой комплекс типовых компонент, централизованно сопровождаемых, адаптируемых к программно-техническим платформам, СУБД, средствам телекоммуникации, организационно-экономическим особенностям объектов внедрения и интегрируемых с существующими разработками, на первый план выступают такие показатели проекта, как управляемость и качество, которые могут войти в противоречие с простотой и скоростью разработки. Для таких проектов необходимы высокий уровень планирования и жесткая дисциплина проектирования, строгое следование заранее разработанным протоколам и интерфейсам, что снижает скорость разработки.

Методология RAD неприменима для построения сложных расчетных программ, операционных систем или программ управления космическими

кораблями, т.е. программ, требующих написания большого объема (сотни тысяч строк) уникального кода.

Не подходят для разработки по методологии RAD приложения, в которых отсутствует ярко выраженная интерфейсная часть, наглядно определяющая логику работы системы (например, приложения реального времени) и приложения, от которых зависит безопасность людей (например, управление самолетом или атомной электростанцией), так как итеративный подход предполагает, что первые несколько версий наверняка не будут полностью работоспособны, что в данном случае исключается.

Оценка размера приложений производится на основе так называемых функциональных элементов (экраны, сообщения, отчеты, файлы и т.п.) Подобная метрика не зависит от языка программирования, на котором ведется разработка. Размер приложения, которое может быть выполнено по методологии RAD, для хорошо отлаженной среды разработки ИС с максимальным повторным использованием программных компонентов, определяется следующим образом:

< 1000 функциональных элементов - один человек,

1000-4000 функциональных элементов - одна команда разработчиков,

> 4000 функциональных элементов - 4000 функциональных элементов на одну команду разработчиков;

В качестве итога перечислим основные принципы методологии RAD:

- разработка приложений интеграциями;
- необходимое использование генераторов кода.

Процессы. Процесс преобразует значения данных. На DFD процесс изображается в виде эллипса, внутри которого помещается имя процесса; каждый процесс имеет фиксированное число входных и выходных данных, изображаемых стрелками.

Потоки данных. Поток данных соединяет выход объекта (или процесса) с входом другого объекта (или процесса). Он представляет промежуточные данные вычислений.

Первым шагом при построении иерархии является построение контекстных диаграмм. Обычно при проектировании относительно простых информационных систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Спиральная модель жизненного цикла. При использовании спиральной модели жизненного цикла для проектирования комплекса основной упор делается на начальные этапы: анализ и проектирование. На этих этапах реализуемость технических решений проверяется путем создания прототипов. Каждый виток спирали соответствует созданию фрагмента или версии комплекса, на нем уточняются цели и характеристики проекта, определяется его качество, и планируются работы следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта, и в результате выбирается обоснованный вариант, который доводится до реализации.

Результат появляется фактически на каждом витке спирали. Этот результат, который является промежуточным, анализируется, а затем выявленные недостатки продукта становятся поводом для инициирования следующего витка спирали. Таким образом, углубляются и последовательно конкретизируются детали проекта, и в итоге выбирается обоснованный вариант, который доводится до реализации. Спираль завершается тогда, когда клиент и разработчик приходят к согласию относительно результата.

Модель предполагает также свойства взаимодействия этапов:

- модель состоит из последовательно расположенных этапов в пределах одного витка спирали;
- внутри витка спирали этапы не имеют обратной связи; анализ результата осуществляется в конце витка и инициирует новый виток спирали;
- исправление ошибок происходит на этапе тестирования на каждом из витков спирали; фактически часть ошибок исправляется в пределах одного витка посредством связи этапов кодирования и тестирования; ошибки, которые не могут быть исправлены и требуют более глубоких структурных изменений, инициируют новый виток спирали;
- этапы могут перекрываться во времени в пределах одного витка спирали;
- результат появляется в конце каждого витка спирали и подвергается подробному анализу, анализируются новые требования заказчика и инициируется новый виток спирали;
- при переходе от витка к витку происходит накопление и повторное использование программных средств, моделей и прототипов;
- процесс ориентирован на развитие и модификацию системы в процессе ее проектирования, на анализ рисков и издержек в процессе проектирования.

Отметим, что основная особенность данной методологии состоит в концентрации сложности на начальных этапах жизненного цикла программного обеспечения (анализ, проектирование); при этом сложность и трудоемкость последующих этапов в пределах одного витка спирали относительно невысокие. По этой методологии предлагается способ снижения затрат в целом, при разработке программного обеспечения за

счет предотвращения потенциальных ошибок на этапах анализа и проектирования. Этап определения стратегии присутствует на первом витке спирали либо «склеен» с этапом анализа первого витка спирали.

Спиральная модель жизненного цикла позволяет устранить недостатки предыдущих моделей.

При спиральной схеме разработки неполное завершение работ на очередном этапе позволяет переходить на следующий этап. Незавершенная работа может выполняться на следующем витке спирали. Тем самым обеспечивается возможность предъявить пользователям системы ее некоторый работоспособный вариант для уточнения требований.

Архитектура программного продукта

Web-сайт имеет архитектуру многоуровневой информационной системы. Для описания данной архитектуры следует четко распределить обязанности всех ее компонентов и изучить методы их взаимодействия.

Архитектура данного web-сайта была разбита на 3 уровня (представление, логика, хранение).

- на уровне деловой логики размещены компоненты, обеспечивающие обработку данных в соответствии с логикой системы;
- уровень хранилища, обеспечивает сохранение информации web-форума;
- уровень представления предназначен для организации клиентского интерфейса.

Уровень пользовательского интерфейса. Подсистемы данного уровня обеспечивают взаимодействие пользователей с информационной системой Web-представительства. На данном уровне web-сайт предоставляет необходимую информацию.

При модернизации web-сайта предполагается внести дополнение в виде нескольких web-страниц, а так же PHP - файлов для корректной работы web-сайта и для предоставления удобного интерфейса.

Функции пользовательского интерфейса (web-сайта):

- предоставление пользователям необходимой информации;
- предоставление ссылок для скачивания;
- взаимодействие с администратором;

2.3. Описание общей структуры проекта

Данный web-сайт реализуется как набор программ, выполняющих функции проекта, изменение и дополнение данных, а также клиентский доступ. Главными требованиями, положенными в основу при разработке комплекса стали: легкое использование и расширяемость. Поэтому комплекс разбивается на 2 основных программных блоков (модулей):

- web-форум
- клиентские web-страницы.

Web-форум представляет собой программный продукт разработанный специально для того что бы пользователи могли свободно общаться между собой. Создание своего собственного сообщения или темы не представляет собой ничего сложного, так как интерфейс интуитивно понятен. Так в данном форуме существует, и страница администратора, на которой можно удалять, старые сообщения и ненужных пользователей. Весь форум реализован на PHP с поддержкой Java script.

ВЫВОДЫ

1. На основании исследования предметной области разработана схема архитектуры разрабатываемого программного продукта, что в дальнейшем поможет нам в программной реализации продукта.

2. Была произведена декомпозиция системы на компоненты в соответствии с решаемыми задачами и с учетом распределенной архитектуры программного комплекса, в результате были получены такие подсистемы, web - сайт, и web - форум.

3. Проанализированы методологии проектирования программных продуктов и выбрана наиболее подходящая методология для нашего варианта.

4. Построена логическая модель разрабатываемого программного продукта. Которая, в дальнейшем поможет при ведении работ по обновлению данных на сайте.

ГЛАВА 3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1. Программно-аппаратная платформа

После завершения этапов анализа и проектирования можно приступить к следующей фазе проекта - физической реализации непосредственно на выбранном языке программирования.

На протяжении этого этапа необходимо: произвести анализ программно-аппаратной базы пользователя и реализовать результаты проектирования на языке программирования. Для этих целей необходимо произвести анализ программно-аппаратной платформы и среды разработки для серверной части, а также минимальную конфигурацию программно-аппаратной платформы пользователя.

Сегодняшний рынок программного обеспечения предъявляет большие требования к создаваемым проектам. Так, для современных программных средств важными требованиями являются переносимость, мультиплатформенность и масштабируемость.

Под переносимостью подразумевается возможность использовать программное средство на разных программно-аппаратных платформах без существенной переработки кода.

Масштабируемость означает возможность добавления новых функций и свойств программного средства с минимальным изменением всего кода в целом. Идеальным является вариант, который позволяет наращивать мощность ПП без изменения основного кода, лишь добавляя новые модули.

Поэтому при разработке учитывались оба этих требования. Естественно, создать достаточно сложное ПО, которое работало бы на всех известных платформах, практически невозможно, но следует стремиться обеспечить его функциональность на самых распространенных платформах.

Исходя из того, что платформа IBM PC является наиболее распространенной в России, было принято решение разрабатывать веб-сайт именно под эту аппаратную платформу. Проанализировав системное программное обеспечение IBM PC-совместимой компьютерной техники, были получены следующие результаты: 73 % - OS семейства Windows, 16 % - Linux, 11 % - Free BSD, Open BSD, SCO, Mac OS X, Novell NetWare. Исходя из этих результатов, а так же из соображения, что программное обеспечение должно функционировать на как можно большем количестве платформ, было принято решение разрабатывать ПС с таким расчетом, чтобы обеспечить функционирование, как его отдельных компонентов, так и всего комплекса в целом на двух основных программных платформах: Windows и Linux. Причем это требования распространяется на ту часть разрабатываемого программного обеспечения, которое должно быть размещено на web-сервере.

3.2. Выбор web-сервера

В настоящее время рынок web-серверов поделён между Apache и Microsoft Information Server (IIS). На их долю приходится 85% рынка, ближайшие конкуренты Google и lighttpd имеют около 6%. Интересно распределение сайтов, функционирующих под управлением того или иного web-сервера по данным Netcraft [11] - самого авторитетного источника в области исследования рынка web-серверов.

Таблица 1.1 – Статистика использования web-серверов различных производителей

Разработчик	Кол-во серверов, октябрь 2010	Доля рынка, %	Кол-во серверов, ноябрь 2010	Доля рынка, %
Apache	68,155,320	47.73%	76,028,287	50.76%
Microsoft	53,017,735	37.13%	53,679,916	35.84%
Google	7,763,516	5.44%	7,910,879	5.28%
lighttpd	1,541,779	1.08%	1,505,122	1.00%
Sun	2,262,019	1.58%	619,262	0.41%

В таблице 1.1. представлена статистика использования web-серверов различных производителей, а в таблице 1.2 рейтинг конкретных продуктов. Как легко можно пронаблюдать лидерство принадлежит web-серверу Apache. Каковы же его преимущества. Рассмотрим их поподробнее и сравним его с ближайшим конкурентом - Microsoft IIS.

Степень защиты.

Легкое достижение высокой безопасности web-сервера Apache в сравнении с IIS является одним из основных аргументов в пользу выбора Apache.

Безопасность web-сервера довольно многогранное понятие, которое включает в себя следующие аспекты:

1. Защита от вывода сервера из строя
2. Защита от хищения информации
3. Защита базы данных в SQL Server

Защита от вывода сервера из строя. Сервер может быть выведен из строя в результате атаки хакеров или воздействия вируса. В 2001 году как и Apache, так и IIS оказались не готовы к новым видам хакерской атаки Denial of service (DoS). Затем вирусы класса Code Red поражали сервера Apache и IIS. Следует отметить, что IIS более уязвим для хакеров и вирусов, чем Apache.

Таблица 1.2 – Статистика использования web-серверов различных производителей

Поз.	Название	Кол-во серверов	Доля рынка, %
	Apache	66144734	52,65
	MS IIS	41254723	32,84
	GFE	5465538	4,35
	unknown	2989767	2,38
	Sun-One-Web-Server	2017698	1,61
	lighttpd	1471779	1,17
	Oversee	1162958	0,93
	Zeus	463449	0,37
	nginx	367291	0,29
	IdeaWebServer	337467	0,27
1	Apache-Coyote	334301	0,27

2	Netscape-Enterprise	225690	0,18
3	Rapidsite	208191	0,17
4	tigershark	203408	0,16
5	Jetty	190452	0,15
6	NOYB	177711	0,14
7	Resin	170046	0,14
8	Squeegit	138941	0,11
9	thttpd	136116	0,11
0	Apache-AdvancedExtranetServer	118043	0,09

Это связано с тем, что IIS не просто web-сервер, а готовое бизнес-решение, которое содержит в себе около 20 web-сервисов разного плана. Иногда вирусу и хакеру удается поразить одну из служб IIS (например, Index Search), что приводит к общей остановке сервера. В Apache для создания бизнес-решения приходится использовать продукты других фирм, поражения их системы безопасности не защищаются Apache как

прокол. Следует отметить, что проблемы безопасности не следует, не преувеличивать, не приуменьшать.

Стоимость владения Apache и IIS.

Сравнивая стоимость владения IIS и Apache можно сделать следующие выводы:

1. IIS и Apache бесплатны, если вы пользователь Windows 2000 Server

2. В IIS существенно быстрее и дешевле можно разработать или переделать сайт. Разработка сайта будет еще дешевле, если вы используете MS SQL 2000.

3. У Apache дешевле стоит обеспечение безопасности. Пользователям IIS для снижения стоимости безопасности следует отключить ненужные сервисы, установить антивирусное ПО и использовать XML-средства MS SQL 2000.

4. Администрирование Apache дороже, чем IIS. Apache сложнее администрировать, но администратору IIS требуется чаще устанавливать антивирусные обновления. Однако установка обновлений для IIS не такая дорогая и сложная работа как написание конфигурационных файлов для Apache.

5. Персонал. Разработчики обойдутся дешевле, кроме того, вы можете использовать для разработки ваших специалистов в MS SQL 2000

В качестве основного web-сервера для ПП «Интернет-магазин «Покупка автомобилей»» был выбран сервер Apache. Немаловажными критериями для отбора стали мультиплатформенность и бесплатное распространение. Кроме того, Apache в настоящее время один из наиболее популярных web-серверов во всем мире, поэтому очень высока вероятность того, что он уже будет входить в серверную конфигурацию

системы, на которую будет производиться установка ПП «Интернет-магазин «Покупка автомобилей»».

Впрочем, при необходимости, в качестве web-сервера можно использовать любой сервер, поддерживающий взаимодействие с РНР.

3.3. Среда разработки для модулей проекта

Сервер должен обеспечивать доступ к web-форуму и быть достаточно производительным для обеспечения работы с несколькими пользователями. Клиент должен иметь удобный и привычный для него интерфейс. Причем как клиент, так и сервер должны обеспечивать между собой надежное и защищенное соединение. Самыми очевидными являются два подхода. Первый предполагает написание отдельных приложений сервера и клиента на каком-либо языке программирования. Но при этом подходе может возникнуть следующая проблема: при серьезном изменении структуры серверной части может потребоваться и серьезная переделка клиента, что не всегда бывает удобно. Поэтому при разработке серверного ядра было решено использовать трехуровневую технологию известную еще как технология «тонкого» клиента. Этот подход предполагает использовать в качестве клиентского приложения web-браузер, поддерживающий современные конструкции языка HTML, что позволяет организовать развитый пользовательский интерфейс. В качестве сервера выступает web-сервер с набором скриптов CGI, реализующий логику приложения и связь данных, хранящихся на сервере баз данных с пользовательским интерфейсом. Это дает следующие преимущества:

1. Не требуется устанавливать дополнительного клиентского программного обеспечения, так как web-браузер присутствует

практически в любой программной конфигурации современного персонального компьютера.

2. Среднему пользователю не требуется дополнительного обучения для работы с клиентской частью, так как обычно хватает начальных навыков по работе с web-браузером и навигации по www.

3. Для пользователя системы имеется возможность выбрать тот web-браузер, к которому он привык.

4. Современные версии web-браузеров поддерживают защищенные протоколы обмена данными. Это позволяет быстро и эффективно разрешить проблему безопасной передачи информации без дополнительного кодирования клиента.

5. Фактически, разработка сводится только к созданию сервера. При добавлении или изменении функций сервера не требуется модификация кода клиента, что существенно упрощает процесс модернизации и наиболее полно отвечает модульному принципу построения программы.

Существует несколько видов реализации серверной части приложения на основе web-сервера: серверный модуль, CGI-приложение, среды предварительной обработки страниц, встраиваемые приложения. Основное их назначение - организовать интерактивное взаимодействие с пользователем на основе динамического гипертекста (DHTML). Выбор конкретного метода обуславливается различными факторами, в частности, при разработке проекта нами были выдвинуты следующие требования к серверу:

1. Мультиплатформенность и переносимость программного кода (как и ко всему комплексу в целом) - серверное ядро должно без особых проблем и без изменения исходного кода функционировать как на различных системных платформах (Windows, Linux), так и web-серверах.

2. Высокая производительность.

3. Обеспечение защищенного соединения с клиентом.

После анализа средств разработки мы пришли к выводу, что вышеперечисленным требованиям сервер будет удовлетворять, если в качестве платформ для разработки выбрать PHP.

PHP (PHP: Hypertext Preprocessor) - один из самых мощных скриптовых языков, доступных на рынке. Он позволяет легко писать скрипты прямо в HTML файлах, и его код будет выполнен транслятором (PHP) на сервере до того, как эта страница будет послана в браузер. PHP аналогичен концепции Netscape's LiveWire Pro или Microsoft's ASP (Active Server Pages), но по заявлению разработчиков языка, PHP лучше обоих продуктов. Многие из его синтаксиса было позаимствовано из C, Java и Perl с добавлением некоторых уникальных особенностей. Главной целью языка является предоставление web-разработчикам быстрого создания динамически генерируемых страниц.

Главные возможности:

PHP безопасен. Использование PHP предоставляет web-серверу еще один уровень между пользователем и сервером, что уменьшает возможность пользователям вмешаться в работу сервера. PHP имеет множество возможностей, которые позволяют программисту строить безопасные приложения: работа с жестким кодом в директории во время компиляции PHP, возможность фильтровать параметры ввода, возможность игнорировать данные, посланные GET методом.

PHP имеет журнал доступа. При использовании контроля доступа в PHP, пользователи могут применять свои собственные счетчики посещений и входа. Эта возможность не использует центральный системный лог файл доступа, и позволяет получить мониторинг доступа в реальном времени. Скрипт для просмотра файла журнала событий предоставляет возможность быстро показывать, например, кто получал

доступ к набору страниц принадлежащих конкретному пользователю или разделу сервера. Пакет может быть настроен на генерацию заголовка каждой страницы с показом информации о доступе.

Перечислим некоторые возможности и достоинства PHP:

поддержка большого количества платформ, включая Linux, Microsoft Windows, Mac OS X, FreeBSD, OpenBSD и т.д.;

интеграция с большинством современных web-серверов, например: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd и множество других;

работа либо в качестве CGI-приложения, либо как серверного модуля (для Apache, IIS, Netscape Enterprise Server);

поддержка как объектно-ориентированного, так и процедурного программирования;

отсутствие ограничений на размер генерируемого HTML файла;

взаимодействие с другими сервисами и приложениями с использованием LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM, CORBA;

работа с XML документами;

поддержка популярных архивных форматов (Zip, BZip2, GZip);

наличие средств обеспечения безопасного соединения

полностью бесплатное OpenSource-распространение.

Очень важно и то, что PHP обладает встроенной поддержкой защиты передаваемой по сети информации с помощью SSL.

3.4. Обзор сред для разработки web-страниц

Когда возникает необходимость создать web-сайт, неминуемо встает проблема выбора инструментального средства разработки. Всего можно

выделить три типа функциональных инструментов, с помощью которых сегодня разрабатывается большинство web-сайтов.

Визуальные редакторы, или WYSIWYG-редакторы («What You See Is What You Get» - «Что видишь, то и получаешь»), позволяют создавать электронные документы без знания языка HTML. Все, что потребуется, - это нажать на соответствующие кнопки, вызвать определенные меню и запустить конкретные команды. В том виде, как страница будет смотреться в рабочем окне программы, в конечном итоге его и будет отображать браузер. Тэговые HTML-редакторы представляют собой специальные программы, предназначенные для людей, разбирающихся в конструкциях языка HTML, и позволяющие нажатием конкретных кнопок вставлять в текущий документ определенные тэги или целые модули.

Такие редакторы пользуются популярностью у многих web-мастеров, так как автоматизируют некоторые задачи, умеют подсвечивать синтаксис языков программирования (помимо HTML еще JavaScript, Perl, PHP). Текстовые редакторы - это простые редакторы для работы с текстом. В таких программах иногда тоже присутствуют подсветка синтаксиса и автоматизация некоторых функций (по причине того, что с недавнего времени большинство разрабатываемых текстовых редакторов по умолчанию ориентированы на работу с электронными документами). Основное отличие от HTML-редакторов заключается в отсутствии возможности вставки тэгов и наличии средств, предназначенных исключительно для работы с простым текстом. Необходимо рассмотреть некоторое программное обеспечение в области HTML-редакторов.

HTMLPad 2015. Чрезвычайно функциональный и одновременно компактный редактор. Традиционная и удобная панель инструментов для

быстрого форматирования текста, создания форм, таблиц, вставки тэгов, причем не только HTML, но также JavaScript, ASP и SSI. Имеется обширный набор инструментов для подбора цветов, работы со шрифтами, структурирования текста и т. д. Из полезных дополнений можно отметить функцию проверки кода HTML («HTML Tidy»), простой, но удобный ftp-клиент и расширенный буфер обмена, что бывает особенно полезно для web-мастера. Редактор обладает функцией контекстной подсветки вводимого кода, благодаря чему найти нужное место даже в большом по объему файле довольно легко и удобно. В программе имеется "джентльменский набор" наиболее часто необходимых JavaScript и блоков HTML-кода (навигационная панель; ссылка для возврата к предыдущей странице; кнопка для закрытия текущего окна, что бывает полезно при работе с HTML-документами, отображаемыми в полноэкранный режим и т. д.). В составе справочной документации можно найти обширное и подробное руководство не только по HTML, но и по CSS, правда, на английском языке. Для просмотра редактируемого документа можно использовать встроенный браузер. Режим «Split screen», позволяющий в одной половине экрана работать с кодом, а в другой в реальном виде через встроенный браузер отслеживать вводимые изменения, превращает HTMLPad 2015 почти в визуальный редактор. Из недостатков можно отметить лишь то, что программа не поддерживает русский язык, но благодаря четко структурированному и интуитивно понятному интерфейсу в ней сможет разобраться даже начинающий пользователь.

Web Notepad. Программа Web Notepad предназначена для тех, кто по тем или иным причинам не приемлет автоматической вставки готовых HTML-конструкций, не привык после себя проверять корректность кода и не любит программным образом добавлять в документ новые команды.

То есть для тех, кто весь код от начала и до самого конца пишет вручную.

Простая и непритязательная утилита, обладающая из возможностей, отличающих ее от обычного Блокнота из состава Windows, только удобной подсветкой синтаксиса, многооконным интерфейсом и функцией загрузки редактируемого документа на сервер посредством ftp.

3.5. Реализация программного продукта

Страницы сайта были реализованы с помощью тегов HTML, динамическое отображение форм. Также при разработке страниц использовались таблицы, ссылки как внешние, так и внутренние.

Реализация web - форума:

Web-форум был реализован с помощью PHP, содержит в себе две страницы: первая используется пользователем, вторая используется администратором. Все данные, находящиеся в форуме доступны для просмотра и редактирования на странице администрирования.

Выводы;

1. Для данного проекта были проанализированы и выбраны программные средства разработки и поддержки проекта: web-сервер Apache, PHP. Описаны возможности данных средств разработки программного обеспечения, которые сыграли ключевую роль в пользу их выбора.

2. Описаны характеристики выбранного web-форума, а так же его структура

3. Наглядно представлена реализация проекта. Представлены основные модули программного продукта, такие как: главная страница, страница администрирования.

ГЛАВА 4. ОЦЕНКА РАЗРАБОТАННОГО ПРОГРАММНОГО ПРОДУКТА

4.1. Критерии оценки качества

Как во время разработки программного обеспечения, так и после ее окончания необходимо проводить анализ качества разработанного программного обеспечения, что в дальнейшем позволит определить качество и работоспособность конечного продукта. Анализ качества способствует выявлению и исправлению максимального количества ошибок при создании ПО, что ведет к максимальным показателям качества.

Качество программного обеспечения может быть оценено следующими характеристиками (Набор характеристик оценки качества ПО определен ГОСТом):

Функциональные возможности. Набор атрибутов, относящихся к сути набора функций и их конкретным свойствам. Функциями являются те, которые реализуют установленные или предполагаемые потребности:

Программный продукт обеспечивает свои функциональные возможности если он обладает набором функций для решения задач в соответствии с его назначением, при учете получения корректных результатов. Если программный продукт является открытой системой и обладает возможностью взаимодействия с другими программными средствами, при условии что он также обладает способностью предотвращать несанкционированный доступ как случайный, так и умышленный, к программам и данным.

Практичность. Набор атрибутов, относящихся к объему работ, требуемых для использования и индивидуальной оценки такого использования определенным или предполагаемым кругом пользователей.

При этом пользователи могут интерпретироваться как большинство непосредственных пользователей интерактивного программного обеспечения. Круг пользователей может включать операторов, конечных пользователей и косвенных пользователей, на которых влияет данное программное обеспечение или которые зависят от его использования. Практичность должна рассматриваться во всем разнообразии условий эксплуатации пользователем, которые могут влиять на программное обеспечение, включая подготовку к использованию и оценку результатов.

Эффективность. Набор атрибутов, относящихся к соотношению между уровнем качества функционирования программного обеспечения и объемом используемых ресурсов при установленных условиях.

Мобильность. Набор атрибутов, относящихся к способности программного обеспечения быть перенесенным из одного окружения в другое. Примечание - окружающая обстановка может включать организационное, техническое или программное окружение.

В процессе разработки web-сайта были выполнены все требования, предъявленные заказчиком.

4.2. Тестирование разработанного программного обеспечения

На стадии внедрения проводилось тестирование интерфейса готового программного обеспечения, и тестирование в режиме реального времени, которое проводилось пользователями совместно с разработчиками.

В процессе тестирования в реальном масштабе времени проверяются результаты обработки исходных данных с учетом времени их

поступления, длительности и приоритетности обработки, динамики использования памяти и взаимодействия с другими программами.

ВЫВОДЫ

Тестирование программного продукта проводилось совместно с конечными пользователями. При тестировании были выявлены определенные недостатки интерфейса, и программные ошибки.

Программные ошибки были обработаны и исправлены. Ошибки интерфейса были исправлены с учетом замечаний пользователей. Вследствие хорошей работы тестеров были выявлены почти все ошибки кода, необнаруженные ошибки в ближайшее время будут обнаружены и исправлены.

В виду сроков, в которые необходимо было разработать программный продукт и объема работы, можно поставить высокую оценку программного продукта. Это отметили как сам автор, так и команда.

Таким образом, программный продукт можно считать завершенным и внедрять его на хостинг.

ЗАКЛЮЧЕНИЕ

1. Проведенная работа позволила создать сайт для автосалона. Внедрение новых технологий позволяет проводить дальнейшие разработки с целью улучшения качества и расширения ее возможностей и в совокупности с модульной организацией легко реализовывать их программно.

2. С помощью данного программного продукта информационная система выходит на уровень web-технологий и позволяет создать представительство организации в Интернет. Web-сайт позволит обеспечить более тесное взаимодействие с пользователями через Интернет. С помощью данного программного продукта клиенты имеют возможность: получить нужную информацию не выходя из дома, а так же пообщаться с другими автолюбителями.

3. Разработанный отражает требования пользователей, обладает умеренными потребностями в аппаратных ресурсах, основано на платформенонезависимой технологии.

4. Web-сайт прошел ряд тестовых испытаний на предмет выявления ошибок в проектировании и реализации программы. Программный продукт соответствует всем требованиям ГОСТа и обладает функциональностью, надежностью, мобильностью и удобством использования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шкарина Л.Н. Методические рекомендации по созданию сайтов 2018г
2. Брябрин В.М. Программное обеспечение персональных ЭВМ. Изд. 2-е, стер. - М.: Наука, 2017г.
3. Грегори Йеркса, Ахмед Абуалсемид. Лучшие средства разработки; [Электронный ресурс] <http://zeiss.net.ru/docs/izone/izone132/pub/izone23.htm>
4. Ойхман Е.Г., Новоженев Ю.В. RAD технологии.; [Электронный ресурс] http://www.citforum.ru/seminars/cbd2000/cbd_day2_04.shtml
5. ЖЦ ПО [Электронный ресурс] http://www.codenet.ru/db/case/glava1_2.php
6. Модели жизненного цикла ПО. [Электронный ресурс] http://www.citforum.ru/database/case/glava1_2.shtml.
7. CASE - технологии и информационные системы. [Электронный ресурс] <http://blues.franko.lviv.ua/ami/books/Case/glava1~1.htm>
8. Основы методологии проектирования ИС. [Электронный ресурс] http://www.citforum.ru/database/case/glava1_1.shtml.
9. Модели жизненного цикла ПО. [Электронный ресурс] http://www.citforum.ru/php/catalog/database/case/glava1_3.shtml.
10. А.М. Вендров. Методология функционального моделирования SADT.[Электронныйресурс]<http://www.citforum.ru/database/php/case/glava2.shtml>.
11. Основы методологии проектирования ИС. [Электронный ресурс] http://www.citforum.ru/database/case/glava1_1.shtml.
12. ВЕБ-дизайн: книга Дмитрия Кирсанова. Издательство: Символ-Плюс, 2016г.

13. Джим Грей о сегодняшнем и завтрашнем дне баз данных. Системы
Управления Базами Данных [Электронный ресурс] <http://www.osp.ru>