**Введение**

Со временем технологии проникают в разные сферы деятельности человека. Еще полвека назад на заводах было множество людей, работающих в цехах, занимающиеся тем, что сейчас делает один или несколько компьютеров. Раньше различным предприятиям и фирмам необходимо было вкладывать огромные средства, покупая рекламные места, подготавливая различные баннеры, информационные стенды и другие вещи, направленные на распространение товаров или услуг.

Огромные средства расходовали на поиски покупателей, чаще всего в своем же регионе, и только после успеха на местном рынке уже задумывались о распространении известности компании дальше, за пределы области. Нанимался огромный штат сотрудников, занимающийся продвижением и распространением продукции компании в каждом городе, а это огромные расходы. Тратились средства на распечатку рекламных материалов, буклетов, журналов и других средств привлечения клиентов.

С приходом в нашу жизнь интернета появилось много новых способов получать и передавать информацию. Сейчас практически любой человек в любое время может получить доступ к совершенно любой информации, надо лишь сформировать запрос.

Конечно, этим и воспользовались маркетологи различных компаний в разных сферах. С помощью всемирной паутины, именуемой интернетом, можно распространять любую информацию на огромные расстояния, тратя в разы меньше денег и других ресурсов, но получая ощутимый результат намного быстрей!

Любая компания стремится как можно громче рассказать о себе, быть услышанной и уведенной как можно большей публикой. Обычно с помощью сети Интернет компании рассказывают о себе и своих товарах, оставляя свои контакты. Такие сайты именуются сайтами-визитками. Они достаточно просты, содержат лишь несколько страниц информации. Обычно это история фирмы, особенно если она существует не первый год, а пару десятилетий, как многие предприятия нашей необъятной страны, краткая информация о текущем положении, какая продукция выпускается, как популярна она, кто ей пользуется из известных брендов или личностей, и контакты, по которым можно связаться для дополнительных вопросов. Часто такие сайты заказывают и обычные люди, специалисты в той или иной области, желающие получить себе резюме в такой форме. Это очень удобно, при должном профессионализме разработчиков такой сайт клиент может дополнять новой информацией по мере необходимости. Какая же польза от этого сайта? Другие люди, которые каким-либо образом узнали, например, имя и фамилию человека, выходят на его сайт и могут узнать о нем подробней. Если это специалист в какой-либо области .на сайте обычно представлены примеры работ, сроки, цены, контрактная информация. Ознакомившись с этим, потенциальные клиенты могут стать реальными, а владелец сайта получит дополнительную прибыль.

Практически любой сайт, если он направлен на увеличение прибыли компании или человека, при должном умении, быстро окупается. В наше время очень важно любой компании иметь собственный сайт, на котором представлена вся необходимая информация, ведь намного проще клиентам искать через интернет, чем ходить по различным офисам в поисках выгодного предложения. Экономия времени огромная, как для отдельного человека, так и для компании, наравне с издержками.

Помимо простых информационных сайтов существуют и такие, на которых можно онлайн, т.е. прямо на сайте, купить товары или услуги компании. Такие сайты называются интернет-магазинами. Они сложней в разработке, но для компаний, занимающихся продажей товаров или услуг, они очень выгодны и актуальны. Типичный функционал таких сайтов обычно следующих – каталог товаров, где представлена вся продукция компании, корзина, куда помещаются выбранные пользователем товары, функция выбора адреса и способа доставки, оплаты товара. На данный момент не каждый человек готов так легко и просто оплатить какую-то сумму онлайн, не удостоверившись в надежности, поэтому такие сайты часто предоставляют возможность оплатить покупку при получении на почте или в пункте выдачи, но все зависит от самой компании, которая владеет сайтом. Постепенно появляются все новые и новые способы доказать, что сайт надежен, а покупки защищены. Часто это шифрование, дабы было невозможно перехватить траффик и узнать платежные данные, перенаправление на страницу банка, где и происходит оплата покупки.

На данный момент каждая компания стремится обзавестись собственным сайтом различной сложности и наполненности. Благодаря этому другие компании, занимающиеся разработкой и поддержкой сайтов, будут получать за это деньги и опыт.

В рамках моей дипломной работы будут рассмотрены различные способы создания сайтов, разнообразные фреймворки, такие как Laravel 5.2, Yii 2, Zend Framework, Angular, Backbone, шаблонизатор Blade, способы размещения сайта в сети Интернет, настройка сервера для удобного администрирования и обновления сайта по мере необходимости.

**Анализ методов решения проблемы**

Современная сфера создания сайтов очень обширна. В ней работают от простых школьников, научившихся делать простые сайты с помощью готовых решений, до профессионалов, создающих сайты любой сложности. Конечно, цены будут сильно зависеть как от сложности самого сайта, так и от исполнителя. Тот же школьник возьмет за простой сайт-визитку всего тысяч пять-шесть и сделает его за неделю, если не быстрей. Профессионал же может взяться за создание крупного сайта с кучей функционала, совмещающего в себе как интернет-магазин, так и какой-либо форум, но и возьмет он за это до нескольких сотен тысяч рублей. Поэтому найти подходящий вариант любой компании не составит труда, надо лишь приложить усилия к поиску.

Рассмотрим какие же варианты создания сайтов сейчас популярны. Первым рассмотрим достаточно популярный способ – тиражное решение. Такое название он получил из-за скорости и простоты разработки. Типичный цикл подобного сайта для разработчика выглядит примерно следующим образом: клиент обращается в компанию с желанием получить сайт. Обычно это сайт-визитка, содержащая несколько страниц. После утверждения содержания и дизайна, которые обычно создаются достаточно быстро, начинается разработка сайта, а точнее развертка уже подготовленной базы на каком-либо хостинге. Далее на такой каркас применяется дизайн, обычно на этапе разработки уже подготовленный к внедрению. Каркас еще называют движком, это распространенное название, которое очень часто встречается как в интернете, так и в разговорах среди разработчиков. Дизайн заранее разбит на блоки, части, которые внедряются в отдельные элементы каркаса, в дальнейшем из них строится каждая страница. После интеграции дизайна сайт наполняют контентом, информацией, которую предоставил клиент. На этом разработка завершена и сайт передается заказчику. Данный цикл разработки занимает, за исключением подготовки дизайна, до одной рабочей недели. Цена не превышает десяти тысяч рублей, чаще всего. Данный тип разработки очень популярен и пользуется огромны спросом.

Следующий способ можно назвать классическим. Цикл разработки выглядит так: обращаясь к компании, занимающейся разработкой, клиент имеет только лишь идею самого сайта. Для примера можно взять идею создания информационного сайта с возможностью просмотра трансляций каких-либо прямых включений, стримов игр. Такой функционал не получится сделать с помощью тиражных решений, если только не прибегать к разработке сторонних модулей для какого-либо движка, что теряет смысл и задумку тиражного решения. Но для упрощения процесса, чтобы одинаковый функционал не реализовывать каждый раз заново были созданы фреймворки, которые содержат в себе набор нужных инструментов и функций, которые часто требуются в различных проектах. Но есть и обратная сторона данного выбора – каждый фреймворк навязывает свое архитектурное решение. Конечно, популярные фреймворки поддерживают популярные же архитектуры, но все равно в некоторые моменты это или просто ограничивает разработчика, или заставляет его искать какое-то изощренное решение, что может непредсказуемо повлиять на проект. После выбора фреймворка начинается процесс реализации требуемого функционала на нем, согласно требованиям заказчика. Этот этап занимаем в среднем от пары рабочих недель до пары месяцев. Часто возникает ситуация, когда заказчик может поменять что-либо в проекте, из-за чего сроки зачастую сдвигаются. После завершения разработки начинается тестирование, благодаря которому находят и исправляют многие ошибки, которые могут возникнуть на сайте. Сайт наполняют контентом, согласовывают мелочи с заказчиком и передают ему готовый проект. В таком способе есть существенное отличие – после сдачи проекта сайт может остаться у компании-разработчика на поддержке. Она заключается в том, что любые найденные ошибки, баги или любое желание заказчика будет сделано теми, кто и разрабатывал сайт. Доработки делаются за дополнительную плату, а вот исправление ошибок не оплачивается, чаще всего. Данный способ разработки популярен и среди новичков, кто хочет научиться программировать, создавать сайты. Но тут возникает некоторая проблема – заказчику требуется сайт в кротчайшие сроки, а разработчик не всегда может уложиться в заданные сроки при малом опыте. Таким образом, данный способ так же является популярным, особенно среди тех компаний, которые имеют особенные требования к сайту и желают получить не просто сайт-визитку, коих в интернете множество.

И последний способ, уже менее распространенный, но все равно пользующийся популярностью – чистая разработка, с нуля, как ее часто называют. Процесс в данном случае чем-то схож с предыдущим вариантом, но чаще к нему прибегают либо компании, желающие получить какой-то продукт или продвинуть свои услуги через интернет. Частая ситуация для данного способа следующая – какая-либо крупная компания решает разработать программный продукт, направленный на работу через интернет. Сайт для онлайн-заметок, например. Нанимается команда разработчиков, которая работает напрямую в компании, являясь ее отделом. Определяется желаемый результат, что хочет получить компания в итоге. Далее происходит декомпозиция проекта, любой функционал подробно делится на более маленькие составляющие, наиболее простые. Определив полную картину проекта, что и как требуется реализовать, процесс разработки делится на этапы. Каждый этап включает в себя набор функционала, который требуется реализовать и протестировать, чтобы быть уверенным в его работоспособности. После завершения всех этапов происходит полное тестирование, обычно для этого привлекаются пользователи, которым данный продукт может быть интересен. После этого продукт выпускают для широкой публики и он становится доступен каждому желающему. Конечно, бывает и так, что продукт изначально нацелен на небольшую аудиторию и даже после завершения разработки он доступен только далеко не всем, но это редкие случаи на фоне всех разработок. Далее компания-разработчик занимается поддержкой и развитием своего проекта. Так как сам проект пишется без использования какого-то крупного популярного каркаса, другим командам тяжелей вникнуть в архитектуру и процесс работы. Таким образом, данный способ разработки является самым затратным и сложными, но в то же время позволяет реализовать проект любой сложности.

**Описание среды разработки**

**Язык разметки HTML**

Любой сайт состоит из двух частей – серверной и клиентской. На серверной стороне обрабатывается информация, формируется ответ на запрос, хранится другая информация. В любом веб-приложении, как сейчас часто называют различные сайты, серверная сторона является очень важной составляющей и на ней происходят любые действия. Клиентская же сторона видна пользователям сайта, и они взаимодействуют именно с ней. На ней обычно оставляют логику отображения интерфейса, взаимодействия с пользователем и взаимодействия с сервером. Так же на клиенте может происходить подготовка введенных пользователем данных для последующей отправки на сервер. Такая схема построения общения клиентской и серверной стороны очень распространена и пользуется большой популярностью.

Конечно, любой сайт состоит не только из логики работы и передачи данных. Важнейшая составляющая для пользователя – интерфейс. Пользователи взаимодействуют через него с приложением и именно от него много зависит. Интерфейс создается на языке разметки HTML – HyperText Markup Language – язык гипертекстовой разметки. Он служит для создания структуры страниц, используя специальный синтаксис.

Каждая страница описывается специальным образом. Начинается страница с тега <html>, в котором уже вложены другие. Так же перед этим тегом может идти тег <!DOCTYPE ..>, в котором указывается спецификация, используемая на этой странице. Этот тег требуется браузерам, чтобы они могли правильно отобразить страницу, согласно логике, описанной в спецификации. Если данный тег не указан, браузер использует настройки по умолчанию, а это может привести к непланируемому поведению страницы.

В теге <html> указываются теги <head> и <body>. Тег <head> служит для описания различной информации на странице, которая не отображается на ней. В «голове» указываются ссылки на файлы стилей, скриптов, данные для СЕО обработки страницы, некоторая служебная информация для построения страницы, заголовок страницы, отображаемый в качестве заголовка окна или вкладки. Про стили страницы и скрипты мы поговорим позже. При получении страницы браузер сначала проверяет составляющую тега <head>, подключает требуемые файлы, если это необходимо. Часто возникают ситуации, когда из-за ошибок в содержимом этого тега страница может не загружаться совсем или работать не так, как ожидал разработчик или пользователь. Опытный программист не допустит такой ошибки и исправит на этапе разработки.

После формирования заголовочной информации страницы браузер начинает обрабатывать тег <body>. В нем содержится вся структура страницы, которая отобразится в окне. С помощью различных средств часть кода можно скрыть, заставив отображаться только в нужный момент, но такое поведение мы рассмотрим позже. В этом теге хранится структура страницы, что будет отображено и каким именно образом. Правильно построенная структура страницы – важнейшая часть при ее создании. Если Не предусмотреть каких-то особенностей страницы, сделав слишком большую вложенность тегов или наоборот, меньше необходимого, можно получить больше проблемы при разработке логики. На ранних этапах это не так критично, но чем больше проект, тем сложней будет изменить какой-то элемент на странице со сложной структурой. Как этого избежать будет описано позже. Самыми популярными тегами при создании разметки страницы являются <div>, <span>, <br>, но применение последнего обычно обусловлено острой необходимостью. Данный тег делает перенос строки в месте собственного нахождения. Это необходимо учитывать при разработке. Два других тега являются очень популярными из-за своей универсальности. Первый тег используется чаще всего, обычно с дополнительными параметрами, такими как class=”” или id=”” , или любые другие. Они служат для разделения одинаковых тегов, чтобы применить к ним какие-либо стили или обратиться из скрипта для получения любой информации или модификации. Тег <span> используется реже, но выполняет примерно такую же функцию. Разница с <div> заключается в том, что <span> не имеет блочного отображения по умолчанию и применяется преимущественно к текстам, а не блокам на странице.

Отображать различную информацию это хорошо и полезно, но надо и каким-то образом принимать информацию от пользователя. Для этого служит тег <form> и связанные с ним <input>, <button>. Первый используется для определения границ формы и некоторых ее атрибутов, так как ссылка, указывающая куда будут отправлены данные, например. Это очень удобно для форм входа, обратной связи, регистрации. Их легко использовать, создание страницы с формой займет не более получаса, если есть конечный адресат, куда будут отправлены данные. Так же есть способы получить информацию из полей ввода <input> и без обертки в виде формы, но это работает только через скрипты.

**стили CSS**

Каким же образом можно сделать пустую и неказистую страницу приятой для глаза и взаимодействия? Для этого была придумана каскадная таблица стилей. Она очень удобна тем, что разделяет описание структуры страницы и ее дизайна в разные файлы, два или больше. Стили удобны тем, что один и тот же стиль можно применить к нескольким элементам, как это делают с классами. Например, можно указать несколько элементов на странице с одним классом: <div class=”class1”><div class=”class1”></div></div><div class=”class1”></div>. После описания в файле стилей класса .class1 {}, он применится ко всем элементам. Это очень удобно, особенно с часто повторяющимся кодом дизайна, когда одинаковых элементов на странице достаточно много и они разбросаны по всей структуре.

Так же стили позволяют описывать не только конкретный класс или идентификатор, но и вложенные элементы. Например, если необходимо описать стиль всех элементов <li> в теге <ul> с классом .list, то в стилях это будет выглядеть следующим образом: ul.list>li {}. Данное описание очевидно, не требует дополнительного времени для разбора вложенности. Трудности с такими вещами могут возникать только у новичков, но им редко приходится сталкиваться с таким. Если правильно уметь использовать селекторы, а именно так это называется, то разработка укоряется и становится легче. Так же у этого способа есть и обратная сторона – можно запутаться, если сильно увлечься этой вложенностью, а не использовать описания новых классов, которые будут добавлены в структуре.

Совместное использование HTML и CSS, особенно их последних версий, может сильно упростить разработку, а так же поможет создать действительно красивый сайт, на котором будет приятно находиться. Правильное описание интерфейса сайта уменьшает проблемы как с отображением в различных браузерах и на разных устройствах, так и облегчает использование.

**шаблонизатор Blade**

**Язык JS**

**Фреймворки Backbone, Angular**

**Язык PHP**

**Фреймворки Yii2, Zend, Laravel**

**NodeJS как серверное решение**

**Редактор Sublime Text(2 и 3 версии)**

**Система контроля версий Git**