GeekBrains

**Программа: Разработчик. Программист.**

**Специализация: Frontend программист**

**Разработка web-сайта для ломбарда**

IT-специалист:

Frontend-программист

Непомнящих Андрей Николаевич

Абакан 2024

**Дипломный проект «Разработка web-сайта для ломбарда»**

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ 3**](#_qhqykx8glii0)

[**ГЛАВА 1 Изучение потребностей и анализ информационного ресурса 5**](#_5vtwlatj9yo2)

[1.1Технико-экономическая характеристика предметной области и предприятия **5**](#_iduiecv5965r)

[1.2 Анализ структуры сайта 7](#_rgs8t8jgiwt4)

[1.3 Анализ потребностей пользователей 9](#_upb3l0p5bjun)

[**ГЛАВА 2 Основы разработки и проектирования веб-сайта 20**](#_jcodby882meg)

[2.1 Понятие веб-сайта 20](#_pjvh35qt5yg9)

[2.2 Проектирование веб-сайта. Этапы разработки 23](#_pbloq2y6waox)

[2.3 Средства разработки 28](#_6dd2k9xatg4v)

[2.4 Фреймворки 30](#_f3bdooao1yna)

[2.4.1 Фронтенд фреймворки 31](#_pmqu4zft38zn)

[2.4.2 Бэкенд-фреймворки 36](#_s4pw2cx6ntyb)

[2.4.3 Фулстек –фреймворки 37](#_xwg0sxn6jgob)

[2.5 Хостинг 40](#_svbzr8qbmhqq)

[2.5.1 Виды хостингов 43](#_ig6xrsl6jchi)

[**ГЛАВА 3 Разработка Web-сайта предприятия** 45](#_tir10umm8nr)

[3.1 Постановка комплекса задач разработки 48](#_zdcr2cxh7r36)

[3.2 Технологии, используемые для разработки сайта 48](#_x2tzm06tyecq)

[3.2.1 Разработка frontend части 50](#_iws2rk4cuupo)

[3.2.2 Хостинг 53](#_u1zb8d9ngg3f)

[**Заключение 58**](#_93ps1leg5wmf)

[**Список использованных интернет ресурсов 60**](#_w1ty3w8zck7v)

**Приложение**

# ВВЕДЕНИЕ

Для успешного развития компании нужны клиенты. А значит, возникает задача поиска клиентов и необходимость, в том, чтобы о компании узнали люди. В данном случае оптимальным решением данной проблемы может стать создание собственного сайта.

Главной задачей профессионально построенного сайта компании является превращение посетителя, зашедшего на сайт, в потенциального клиента.

Объект исследования дипломной работы – разработка web-сайта компании.

Предмет исследования дипломной работы – web-сайт компании, оказывающей услуги по скупке, продаже, оценке ценных предметов, электроники, бытовой техники, выдаче займов под залог, аренды бензо-электроинструмента.

Цель дипломной работы – разработать web-сайт компании, оказывающий услуги по скупке, продаже, оценке ценных предметов, электроники, бытовой техники, выдаче займов под залог, аренды бензо-электроинструмента.

Задачи дипломной работы:

• исследовать предметную область;

• исследовать существующие web-сайты компаний, оказывающих услуги по скупке, продаже, оценке ценных предметов, электроники, бытовой техники, выдаче займов под залог, аренды бензо-электроинструмента;

• исследовать средства для разработки web-сайта;

• разработать структуру web-сайта;

• реализовать разработанную модель web-сайта;

• заполнить web-сайт контентом.

# ГЛАВА 1 Изучение потребностей и анализ информационного ресурса

# 1.1 Технико-экономическая характеристика предметной области и предприятия

Полное юридическое наименование учреждения: ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ЛОМБАРД БРАТ".

ООО "ЛОМБАРД БРАТ" зарегистрирован 10 июля 2020 года.

При регистрации организации присвоен ОГРН 1201900001830, ИНН 1901144401 и КПП 190101001.

Юридический адрес: 655001, республика Хакасия, г. Абакан, ул. Крылова, д. 106, корп. 2, кв. 31.

Виды деятельности ОКВЭД:

Основной:

64.92.6: Деятельность по предоставлению ломбардами краткосрочных займов под залог движимого имущества;

## 1.2 Анализ структуры сайта

Структура сайта <https://ломбард-брат.рф> является древовидной, что является универсальным вариантом. Каждому пункту сайта или услуге отводится отдельная ветка: раздел или подраздел. Привычнее всего пользователям общаться именно с такими ресурсами. При древовидной структуре смысловая нагрузка делится между главной страницей и отдельными разделами, так как с ними линкуются отдельные страницы. Для продвижения это наиболее оптимальный вариант, позволяющий рекламировать сразу несколько разделов сайта.

## 1.3 Анализ потребностей пользователей

Анализ потребностей пользователей основывается на потребностях клиента который состоит из:

1. Рационального - безопасность, комфорт, надежность;
2. Эмоционального - статусность, новизна;
3. Сопряженные - “цепочка” нужной информации;
4. Не сопряженные - разовый поиск информации;
5. Скрытые - клиент не чувствует нужды;
6. Явные - клиент осознает нужду;

Все вышеописанные потребности относятся к “Иерархии потребностей по пирамиде Маслоу”.

# ГЛАВА 2 Основы разработки и проектирования веб-сайта

# 2.1 Понятие веб-сайта.

Веб-сайт (англ. Website, от web- паутина и site- «место») - объединённая под одним адресом и отраженная при помощи языков программирования совокупность документов, файлов и информации в сети Интернет. Веб-сайт может принадлежать частному лицу или организации. Вся совокупность сайтов определяют Всемирную Паутину. Для доступа пользователей к сайтам был разработан специальный протокол прикладного уровня передачи данных, иначе говоря, протокол HTTP  (англ. HyperTextTransferProtocol — «протокол передачи гипертекста»)

Страницы сайтов — это набор текстовых файлов, размеченных на языке HTML. Эти файлы, будучи загруженными посетителем на его компьютер, понимаются и обрабатываются браузером и выводятся на средство отображения пользователя (монитор, экран КПК, принтер или синтезатор речи).

Язык HTML позволяет форматировать текст, различать в нём функциональные элементы, создавать гипертекстовые ссылки (гиперссылки) и вставлять в отображаемую страницу изображения, звукозаписи и другие мультимедийные элементы.

Отображение страницы можно изменить добавлением стилей на языке CSS, что позволяет централизовать в определённом файле все элементы форматирования (размер и цвет заглавных букв 2-го уровня, размер и вид блока вставки и другое) или сценариев на языке JavaScript, с помощью которого имеется возможность просматривать страницы с событиями или действиями.

Страницы сайтов могут быть простым статичным набором файлов или создаваться специальной компьютерной программой на сервере. Они могут быть либо сделаны на заказ для отдельного сайта, либо быть готовым продуктом, рассчитанным на некоторый класс сайтов. Некоторые из них могут обеспечить владельцу сайта возможность гибкой настройки структурирования и вывода информации на веб-сайте. Такие управляющие программы называются системами управления содержимым (CMS).

Первый в мире сайт info.cern.ch появился 6 августа 1991 года. Его создатель, Тим Бернерс-Ли, опубликовал на нём описание новой технологии WorldWideWeb, основанной на протоколе передачи данных HTTP, системе адресации URI и языке гипертекстовой разметки HTML. Также на сайте были описаны принципы установки и работы серверов и браузеров. Сайт стал и первым в мире интернет-каталогом, так как позже Тим Бернерс-Ли разместил на нём список ссылок на другие сайты. Все инструменты, необходимые для работы первого сайта, Бернерс-Ли подготовил ещё раньше — в конце 1990 года появились первый гипертекстовый браузер WorldWideWeb с функционалом веб-редактора, первый сервер на базе NeXTcube и первые веб-страницы.

Что касается России, то активное распространение Интернета началось в конце девяностых годов не смотря на то, что первый web-сайт на русском языке был создан уже в 1995 году. На сегодняшний день Рунет изобилует сайтами различной направленности.

«Отец» веба считал, что гипертекст может служить основой для сетей обмена данными, и ему удалось претворить свою идею в жизнь. Ещё в 1980 году Тим Бернерс-Ли создал гипертекстовое программное обеспечение Enquire, использующее для хранения данных случайные ассоциации. Затем, работая в Европейском центре ядерных исследований в Женеве (CERN), он предложил коллегам публиковать гипертекстовые документы, связанные между собой гиперссылками. Бернерс-Ли продемонстрировал возможность гипертекстового доступа к внутренним поисковику и документам, а также новостным ресурсам Интернета. В результате, в мае 1991 года в CERN был утверждён стандарт WWW. Тим Бернерс-Ли является «отцом» основополагающих технологий веба — HTTP, URI/URL и HTML, хотя их теоретические основы были заложены ещё раньше.

В 1940-х годах Вэнивар Буш выдвинул идеи расширения памяти человека с помощью технических устройств, а также индексации накопленной человечеством информации для её быстрого поиска. Тед Нельсон и ДагЭнгельбарт предложили технологию гипертекста — «ветвящегося» текста, предоставляющего читателю разные варианты чтения. Xanadu, так и не законченная гипертекстовая система Нельсона, была предназначена для хранения и поиска текста, в который введены взаимосвязи и «окна». Нельсон мечтал связать перекрёстными ссылками все тексты, созданные человечеством.

В настоящее время Тим Бернерс-Ли возглавляет основанный им Консорциум Всемирной паутины (WorldWideWebConsortium), который занимается разработкой и внедрением стандартов Интернета.

Сайты могут содержать подразделы, ориентированные целиком на ту или иную аудиторию. В этом случае такие разделы называют версиями сайта. Аудитория может различаться по виду используемого оборудования, по используемому языку аудитории. К примеру, известны так называемые мобильные версии сайта, предназначенные для работы с ними с использованием смартфона. Сайты могут иметь языковые версии (русскоязычная, англоязычная и другие).

По предоставляемой информации web-сайты можно разделить на:

• сайт визитка – самый простой вид сайта. Сайт такого типа можно сделать даже на простом HTML, без использования системы управления сайтом. Обычно сайт-визитка содержит от одной до пяти страниц. Сайты этого вида как правило содержат только общую информацию о владельце сайта и его контакты. Простота создания такого вида сайта делает стоимость его разработки сравнительно дешевой, что является очевидным преимуществом для заказчика;

• корпоративные сайты – это полнофункциональные представительства компаний в интернете. Этот тип сайта лучше всего подходит для серьёзных средних и крупных фирм. Корпоративные сайты содержат полную информацию о компании, а так же её деятельности. Такой тип сайта иногда называют виртуальным офисом, так как посещение такого сайта сравнимо с общением с менеджером по работе с клиентами. В первую очередь, корпоративные сайты нужны для формирования имиджа компании и предоставления посетителям и клиентам наиболее полной информации;

• интернет-витрина или интернет-каталог товаров – это вид сайтов, основная функция которых – продавать. На таких сайтах публикуется информация о товарах и контакты, обычно телефоны, по которым следует звонить желающим приобрести предлагаемый товар. На таких сайтах размещаются технические характеристики товаров, отзывы, рекомендации экспертов и т.д.;

• интернет-магазины – этот вид сайтов аналогичен интернетвитринам, но имеет дополнительный функционал в виде возможности заказать выбранный товар прямо через сайт;

• промо-сайты – сайты этого типа специализируются на раскрутке и продвижении какого-либо товара или бренда;

• тематические сайты – данный тип интернет сайтов характеризуется тем, что содержит информацию по какой-либо конкретной тематике. К ним можно отнести и интернет-энциклопедии;

• интернет-порталы – это тип сайтов, содержащих большое количество разнообразной информации. Как правило, порталы схожи по структуре с тематическими сайтами, но имеют более развитый функционал и большее количество сервисов и разделов. Также на порталах часто размещаются разделы для общения пользователей: чаты, блоги и форумы;

• блог – это тип сайтов, на которых владелец или редактор блога пишет посты со своими новостями, идеями или другой постоянно поступающей информацией. Отличительная особенность блогов заключается в актуальности публикуемой информации;

• поисковые системы – вид сайтов, предназначенных для поиска страниц, или определенной информации в интернете по определённым запросам;

• интернет-форумы – на сайтах этого вида пользователи могут создавать темы, а также комментировать их. Как правило, форумы ограничены одной специфической тематикой, хотя встречаются и форумы «обо всём»;

• социальные сети – тип сайтов, созданных для общения пользователей между собой. Часто такие сайты содержат рейтинги, страницы пользователей, группы и множество других сервисов. По доступности сервисов web-сайты делятся на:

• открытые – доступны для каждого пользователя;

• полузакрытые – для доступа к информации потребуется пройти процедуру регистрации;

• закрытые – недоступны для пользователей, не входящих в определенную группу, и доступ к такому web-сайту можно получить только через специальные приглашения.

По цели создания выделяют следующие типы сайтов:

• коммерческие – это самый распространенный вид интернетресурсов, ориентированный на привлечение потенциальных клиентов, партнеров и информирование их о товарах и услугах компании. Разработка сайта организации обычно подразумевает установку таких модулей, как новости организации, пресс-центр или статьи об организации, каталог товаров и услуг, портфолио, контакты, а так же система обратной связи. В отличие от одностраничных сайтов, коммерческий сайт часто имеет разветвленную систему меню с продуманной навигацией. Кроме того, в зависимости от видов деятельности организации-заказчика, на сайте могут размещаться такие дополнительные модули, как калькулятор стоимости услуг, система онлайнконсультаций, форум и т.д., вплоть до модулей сопряжения с корпоративной информационной системой;

• некоммерческие – web-сайты, перед которыми не ставится коммерческие задачи, решение которых так или иначе повлечет увеличение прибыли. Как правило, такие web-сайты принадлежат различным социальным учреждениям, органам власти, отдельным личностям, учебным заведениям. Некоммерческий web-сайт, чаще всего функционирует для обеспечения присутствия в сети Интернет и информационного обеспечения. Вторичными являются задачи организации обратной связи, а так же другие задачи.

По технологии создания web-сайты бывают:

• статические – не меняют своё содержимое или разметку при каждом запросе к серверу, а изменяются только тогда, когда администратор ресурса вручную вносит изменения в текстовом процессоре или в инструменте вебредактирования, например, таком как Adobe Dreamweaver. Большая часть web-сайтов базируется на основе статических страниц. Такая методика является весьма экономичной для публикации в сети информации, которая существенно не изменяется в течение месяцев или даже лет. Большинство CMS также используют статическую публикацию для предоставления вебсодержимого. В системах управления веб-содержимым страницы создаются и изменяются динамически под управлением базы данных через вебинтерфейс для редактирования, но затем опубликованные страницы считываются с web-сервера как обычные статические страницы. Статические страницы являются простыми, безопасными, менее склонными к технологическим ошибкам и сбоям, их хорошо видят поисковые системы;

• динамические – могут менять свое содержимое, а так же внешний вид в зависимости от действий пользователей или изменений данных, поддерживаемых приложением, или с течением времени, как на новостных ресурсах. Применяя сценарии, исполняемые на клиентской стороне (Flash AS, технология Ajax, XML), содержание может быть быстро изменено на компьютере пользователя без запросов к новой странице к серверу. Однако, большая часть динамического содержания в сети монтируется с помощью исполняемых на web-сервере языков программирования (Perl, PHP, ASP, JSP, Python). Оба подхода (исполняемый на стороне клиента и исполняемый на серверной части) используются в разносторонних web-сайтах с постоянно меняющимся содержанием и сложными интерактивными функциями. Динамические web-страницы предлагают большую гибкость, но процесс обработки уникально собранного содержания при каждом запросе к странице требует быстрого сервера, и даже очень мощный web-сервер может потерять производительность под грудой, которую создает большое количество запросов 10 к динамическим web-страницам за короткое время. Если системы обработки динамического web-содержимого плохо оптимизированы, зачастую они гораздо менее видимы поисковым системам, чем статические страницы.

Структура сайта - это связь документов, принадлежащих ему, между собой. Правильная структура позволит поисковым роботам быстро совершать обход ресурса, а посетителям легко перемещаться между его страницами. Все это напоминает схему здания. К примеру, для того чтобы попасть из комнаты А в комнату Б, необходимо пройти по лестнице на 2 этажа вверх, затем сделать 5 шагов и повернуть налево. Также из помещения А в помещение Б можно попасть, если подняться на лифте на 3 этажа, затем пойти по лестничному пролету вниз, сделать 2 поворота направо и закончить путь 10ю шагами. В то же время, для того чтобы покинуть здание из любой комнаты - можно выйти на общий балкон и спуститься в стеклянном лифте вниз. Структура сайта важна как для SEO, так и для посетителей ресурса. И поисковик, и человек не должны слишком долго путешествовать в поисках нужной информации. Потерявший терпение человек просто покинет ваш ресурс, а поисковик не сможет проиндексировать все важные страницы.

В большинстве случаев сайты размещаются по одному адресу (имеют одно доменное имя). Но случаются и исключения: несколько ресурсов могут иметь общий домен, или один ресурс можно разместить по нескольким адресам.

Место хранения любого сайта – веб-сервер (системный блок большого размера). Для доступа к сайтам используют браузер. Браузер, или веб-обозреватель (от англ. webbrowser, МФА:) — прикладное программное обеспечение для просмотра страниц, содержания веб-документов, компьютерных файлов и их каталогов; управления веб-приложениями, а также для решения других задач. В глобальной сети браузеры используют для запроса, обработки, манипулирования и отображения содержания веб-сайтов. Многие современные браузеры также могут использоваться для обмена файлами с серверами FTP, а также для непосредственного просмотра содержания файлов многих графических форматов (gif, jpeg, png, svg), аудио- и видеоформатов (mp3, mpeg), текстовых форматов (pdf, djvu) и других файлов.

Функциональные возможности браузеров постоянно расширяются и улучшаются благодаря конкуренции между их разработчиками и высоким темпам развития и внедрения информационных технологий. Несмотря на то, что браузеры разных изготовителей базируются на разных технологических решениях, большинство современных браузеров придерживается международных стандартов и рекомендаций W3C в области обработки и отображения данных. Стандартизация позволяет добиться предсказуемости в визуальном представлении информации конечному пользователю независимо от технологии, которая использована для её отображения в браузере.

Со времени начала применения браузеров во Всемирной паутине в начале 1990-х годов из простого средства просмотра текстовой информации браузер превратился в комплексное прикладное программное обеспечение для обработки данных и обеспечения интерфейса между информационными ресурсами и человеком. В последние годы многие разработчики браузеров сосредоточили свои усилия на повышении удобства пользовательского интерфейса браузеров для их использования в аппаратных устройствах, в которых применяются сенсорные экраны. Браузеры распространяются, как правило, бесплатно. Потребителям браузер может быть поставлен в форме самостоятельного (автономного) приложения или в составе комплектного программного обеспечения.

К примеру, браузеры InternetExplorer и MicrosoftEdge поставляются в составе операционной системы MicrosoftWindows;

MozillaFirefox — отдельно или в составе дистрибутивов Linux (например, Ubuntu);

Safari — в составе операционной системы Mac OS X;

GoogleChrome, Opera и другие браузеры — как самостоятельные приложения во множестве вариантов для различных операционных систем.

Рассмотрим также хостинг (англ. hosting) — услуга по предоставлению ресурсов для размещения информации на сервере, постоянно имеющем доступ к сети (обычно Интернет). Обычно услуга хостинга входит в пакет по обслуживанию сайта и подразумевает, как минимум, размещение файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам (веб-сервер). Как правило, в обслуживание уже входит предоставление места для почтовой корреспонденции, баз данных, DNS, файлового хранилища на специально выделенном файл-сервере и т. п., а также поддержка функционирования соответствующих сервисов. Хостинг базы данных, размещение файлов, хостинг электронной почты, услуги DNS могут предоставляться отдельно, как самостоятельные услуги либо входить в комплексную услугу.

**2.2   Проектирование веб-сайта. Этапы разработки.**

Этап разработки веб-сайта можно разделить на три этапа:

- Планирование;

- Дизайн;

- Разработка;

Планирование. На данном этапе необходимо определиться с темой сайта и приступать к поиску и сбору информации в соответствии с выбранной темой.

Следующим шагом будет разработка структуры проекта, которая подразумевает под собой разделы сайта, в соответствии  с которыми будет разрабатываться меню и дизайн, а материал располагаться по темам.

Далее нам нужно разработать макет проекта, который нам нужен для понимания того, как на нашем сайте будет располагаться блоки, меню, графика и информация. На начальном этапе разработки достаточно сделать это схематично.

Дизайн или верстка — создание структуры гипертекстового документа на основе HTML-разметки, при использовании таблиц стилей и сценариев, таким образом, чтобы элементы дизайна выглядели аналогично созданному ранее макету. Существует два вида верстки – блочная и табличная.

Табличная верстка – первый вид верстки, с которого началась эпоха сайтов. При табличной верстке страница поделена на соседствующие ячейки, что напоминает стандартную работу с таблицами в Excel. Минус такого подхода состоял в том, что приходилось создавать дополнительные таблицы, которые впоследствии могли остаться пустыми. Например, если требовалось разместить изображение и зафиксировать его положение, то необходимо было создать новую строку и разделить ее на несколько столбцов. Только один из них бы содержал изображение, а другие служили бы для него фиксаторами. Как таковая табличная верстка сейчас не используется, но без нее не обходятся при верстке электронных писем – там она, можно сказать, обязательна. Сама верстка разрабатывается с помощью тега , который задает основные параметры таблицы – длину, ширину и прочее. Внутри тега располагаются теги и, где первый необходим для создания строки, а второй – для столбца.

Самый актуальный вид верстки сайтов – блочный. Он основан на теге, с помощью которого создаются контейнеры, включающие в себя весь контент страницы или отдельного блока. Например, мы можем разделить сайт на несколько блоков: первый экран, о компании, контакты – для каждого блока будет отведен свой тег <div>. Внутри тега <div> уже находятся другие теги, отвечающие за те или иные элементы. При таком подходе язык разметки HTML всегда взаимодействует с CSS-стилями, которые преобразуют обычную страницу в стильное дизайнерское решение: добавляются цвета, устанавливаются отступы для элементов, задается базовая анимация и многое другое.

HTML и CSS обычно хранят в разных файлах – такой подход позволяет быстро вносить изменения и не путаться в больших проектах. Также стоит сказать, что блочная верстка позволяет легко создать адаптивный сайт, что в наше время является обязательным требованием для каждого проекта. Такая разработка позволяет не только создавать сайты для телефонов и планшетов, но и обеспечивает попадание сайта в топ выдачи поисковых систем.

Последним этапом является разработка или программирование. К программистам поступают готовые шаблоны страниц и указания дизайнеров по работе и организации элементов сайта. Программист создаёт программную основу сайта, делая её с нуля, используя фреймворк или CMS.

CMS — Система управления содержимым (англ. Contentmanagementsystem, CMS, система управления контентом) — информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления содержимым, иначе — контентом.

Фреймворк —  программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

**2.3   Средства разработки**

В процессе превращения Интернета из набора информационных ресурсов в инструмент ведения бизнеса, технологии создания сайтов существенно изменились. На сегодняшний день существует огромное количество различных методов создания сайтов, которые различаются в зависимости от назначения и типа сайта, умений и навыка разработчика. Каждый метод разработки сайта отличается друг от друга и имеет свои особенности. Можно выделить три основных метода:

1.  Разработка с нуля;

2.  Использование конструктора;

3.  Использование системы управления сайтом;

Разработка с нуля. Данный метод наиболее длительный и трудный. Вдобавок ко все, он требует от разработчика знаний языка Веб - программирования (PHP, HTML, CSS), а также опыт работы с базами данных (MySQL). Помощниками для разработчика станут популярные программы для создания сайтов вручную, к которым относятся Dreamweaver, FrontPage, WebSite X5, а также Photoshop (для работы с изображениями). Однако всю сложную и муторную работу разработчику придется выполнить самому. Самый главный недостатк данного метода заключается в том, что необходимо осуществить все самостоятельно, потратив на это огромное количество времени и собственных сил.

Использование конструктора. Конструкторы сайтов – это, онлайн- система, позволяющая из готового типового набора модулей и шаблонов разработать свой сайт. Так же есть возможность сразу разместить его в web. Это самый простой метод разработки, так как он не требует знаний профессионального Веб - программиста. Существует множество инструментов для разработки сайта, это всевозможные шаблоны и макеты. К самым  популярным  и  хорошо  изученным  конструкторам  относятся «narod.ru», «wix.ru», «ucoz.ru» и т.д. Зачастую, воспользовавшись конструктором на начальном этапе, разработчик может впоследствии пожалеть о потерянном времени на изучение работы конструктора, поскольку в данном методе тоже есть свои недостатки. Один из них заключается в ограниченности возможностей, то есть невозможность выйти за рамки функционала конструктора. Использование системы управления сайтом. Это самый быстрый и широко используемый метод, он довольно легок в изучении и использовании. На сегодняшний день широко используются такие системы или «движки», именно так их называют профессиональные программисты, как WordPress, Joomla, DУango, 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. Владелец сайта может сам выбрать, какой из методов использовать. Тут уже зависит от собственных возможностей, предпочтений и умений. На данный момент существует множество программ, которые способны предложить пользователю различные способы создания интернет проектов. Это стало возможным благодаря специализированным программам. Таким как ContentManagementSystem (CMS). В переводе обозначает «Система управления контентом».

Система управления сайтом – это программа, предназначенная для работы в Интернете. Хостинг – это удаленный Веб – сервер. Методы создания сайтов с использованием CMS является одним из самых популярных на сегодняшний день. CMS, представляет собой визуальную и программную оболочку, которую пользователь может заполнить необходимым контентом, а также по своему желанию изменить и настроить. Автоматизированные методы создания сайтов предусматривают разделение структуры сайта на «дизайн» и «контент». В этом случае легко можно изменять контент, не затрагивая дизайна сайта или его программного кода. При ручном создании сайта разделения структуры сайта на две отдельные «ветви» - дизайн и содержимое - не происходит. Первые CMS были написаны, на одном из классических языков программирования. Интерфейс пользователя создавался с использованием языков HTML и JavaScript, а административная часть представляла собой обычную программу, написанную на одном из языков программирования. Современные системы управления сайтом реализованы на языке PHP, который на сегодняшний день является безусловным лидером среди языков программирования, ориентированных для работы в Интернете. Современные системы управления сайтом реализованы на языке PHP. На сегодняшний день является лидером среди языков программирования. В качестве баз данных используется бесплатная база данных MySQL. Кроме MySQL, существуют еще и другие версии баз данных, но используются они значительно реже. База данных позволяет хранить огромные массивы различных данных, которыми вы будете наполнять сайт, а также данные самой CMS.

**2.4 Фреймворки**

Фреймворк - это набор готовых решений, библиотек, инструментов и шаблонов, предназначенных для разработки программного обеспечения. Фреймворк обеспечивает структуру и основу для создания приложений, упрощает процесс разработки и позволяет разработчикам сосредоточиться на бизнес-логике приложения, минимизируя необходимость писать повторяющийся код. Фреймворки обычно предлагают стандартизированный способ организации проектов и упрощают совместную работу разработчиков.

Фреймворки помогают:

- ускорить разработку, так как программисту не нужно писать много кода с нуля;

- предотвратить ошибки, так как работа фреймворков уже отлажена, разработчики регулярно поддерживают их, дополняют примерами использования и решения проблем;

- упростить адаптацию новых разработчиков в команде.

Фреймворки делятся по типам задач на:

- фронтенд-фреймворки, которые отвечают только за видимую часть сайта или приложения;

- бэкенд-фреймворки, которые отвечают за внутреннюю (серверную) часть сайтов и приложений;

- фулстек-фреймворки, которые отвечают одновременно за внешнюю и внутреннюю часть работы сайтов и приложений.

**2.4.1 Фронтенд фреймворки**

Фронтенд-фреймворк — это инструмент для разработки внешнего (пользовательского) интерфейса.

Чаще всего он основан на таких языках программирования, как JavaScript, HTML и CSS.

Среди решаемых задач — разработка дизайна UX/UI, SEO-оптимизация, управление взаимодействием с пользователем и многое другое.

Примеры фронтенд-фреймворков:

- React,

- Vue.js,

- Ember,

- Bootstrap,

- Angular.

**2.4.2 Бэкенд-фреймворки**

Бэкенд-фреймворк — это программное обеспечение, которое предоставляет инструменты и библиотеки для разработки серверной части веб-приложений.

Вот некоторые популярные бэкэнд-фреймворки:

- Ruby on Rails,

- Express.js,

- Next.js,

- Adonis.js,

- Drupal,

- Spring Boot,

- ASP.NET core.

Выбор бэкэнд-фреймворка зависит от конкретных потребностей приложения, навыков и предпочтений разработчика.

**2.4.3 Фулстек –фреймворки**

Эти фреймворки используют JavaScript и TypeScript для разработки веб-приложений. Они позволяют создавать пользовательские интерфейсы, логику на стороне сервера, интеграцию с базами данных и разработку API с использованием единого и согласованного подхода.

Вот некоторые фулл-стек фреймворки:

- Next.js,

- Nuxt,

- Meteor,

- Astro,

- Remix,

- CT3A,

- Redwood,

- SvelteKit,

- Blitz,

- Fresh,

- Amplication.

**2.5 Хостинг**

Хостинг — это услуга по хранению сайта. Она нужна, чтобы пользователи интернета могли посещать его круглосуточно. Компания, которая предоставляет эту услугу, называется хостером, или хостинг-провайдером. Понятие происходит от английского слова host, что значит «хозяин, принимающий гостей».

Любой сайт — это набор текстовых и видеофайлов, изображений, кода, баз данных и прочей информации. Чтобы люди могли увидеть их, сайт должен быть загружен на компьютер со специальным программным обеспечением и круглосуточным доступом в интернет. Такой компьютер называется сервером. У компании-хостера достаточно серверов для обслуживания огромного количества клиентов.

Физически серверы располагаются на специальных стойках в отдельном помещении. Оно называется дата-центром, центром обработки данных (ЦОД) или центром хранения и обработки данных (ЦХОД). В нём обеспечивают бесперебойную подачу электричества за счёт резервных генераторов, связь, обновление оборудования, безопасность. У хостера может быть свой ЦОД. Но чаще он арендует у дата-центра его серверы или место под размещение своих серверов.

Дата-центр может располагаться в любой точке мира. Это важно учитывать при выборе хостинга, если ваш сайт содержит информацию, которую по закону можно хранить только на территории РФ.

**2.5.1 Виды хостингов:**

**- Выделенный хостинг** (dedicated server). Вы арендуете у хостера сервер целиком — то есть целый компьютер оказывается у вас в безраздельном пользовании. Вы можете устанавливать любое ПО и выбирать комплектующие. Вся мощность этой машины ваша.

Собственный сервер можно сделать и на базе обычного компьютера. Это потребует специфических знаний и затрат на обеспечение его работоспособности. Такой сервер тоже будет выделенным.

Преимущества выделенного хостинга:

- высокая производительность — на сервере проект только одного клиента;

- максимальный контроль — можно выбирать ПО и комплектующие, необходимые именно для вашего проекта.

Недостатки:

- дорого — стоимость аренды в среднем начинается от 100 долларов в месяц и может превышать 1000 долларов. Чем мощнее сервер, тем выше цена;

- для поддержки хостинга нужна высокая квалификация и большие затраты.

**- Виртуальный хостинг** (shared hosting). Вид хостинга, когда один физический сервер сдают в одновременное пользование нескольким арендаторам. Арендаторов могут быть десятки и сотни. Специальное программное обеспечение следит за тем, чтобы мощности машины распределялись между клиентами более-менее равномерно, однако это происходит с некоторыми ограничениями.

Преимущества:

- лёгкий старт и простота управления;

- низкая стоимость — от 1,5 доллара в месяц.

Недостатки:

- Ограничены контроль и функциональность. Сервер предоставляют «как есть» и настраивают для всех клиентов одинаково. Нельзя, например, поставить дополнительное ПО или изменить файлы конфигурации.

- Ограничены объём и скорость передачи данных между пользователями и вашим сайтом.

- Могут быть проблемы с производительностью. Ресурсы сервера общие, клиенты используют их одновременно. Поэтому ваш сайт может начать виснуть из-за того, что в какой-то момент ему достанется мало оперативной памяти.

- Общий IP. На виртуальном хостинге под одним IP-адресом могут находиться сотни сайтов. И если одного из «соседей» заблокирует Роскомнадзор, то перестать работать могут и другие.

**- Виртуальный сервер**. На одном физическом сервере запускают изолированные друг от друга процессы, которые имитируют работу самостоятельного компьютера. Виртуальные серверы обозначают аббревиатурами VPS (virtual private server) и VDS (virtual dedicated server). Технические различия между ними есть, но это несущественно для большинства пользователей.

Клиент получает те же привилегии, что и при работе с выделенным сервером. Это и установка необходимой операционной системы, и своё программное обеспечение. Виртуальные серверы не влияют на работу друг друга.

Преимущества:

- защищённая среда;

- контроль и возможность тонкой настройки;

- стоимость ниже, чем у выделенного сервера;

- скорость работы выше, чем у виртуального хостинга.

- Недостатки:

- дороже виртуального хостинга;

- для администрирования необходимы знания.

**- Облачный хостинг.** Он использует ресурсы нескольких связанных между собой физических и виртуальных серверов, к которым добавлены системы хранения данных. При аренде клиент платит только за фактически используемые ресурсы и может увеличивать или уменьшать их в зависимости от потребностей. Оплата может быть почасовой — это даёт возможность настроить потребление мощностей в дневное и ночное время. Настройки регулируются в автоматическом режиме, что обеспечивает гибкость управления.

Преимущества:

- стабильность за счёт динамического распределения ресурсов;

- экономичность;

- масштабируемость;

- гибкость.

Недостатки:

- дороже, чем виртуальный хостинг;

- меньше прав администрирования.

Чтобы выбрать хостинг, нужно оценить свои потребности, подобрать несколько компаний и сравнить их.

Оценка потребностей нужна, чтобы понять, какой тип сервера вам подойдёт. Если вы новичок, у вас простой сайт или лендинг, то есть то, что не требует особых ресурсов для работы, — выбирайте виртуальный хостинг. Если проект предполагает массивные нагрузки и большой ресурс железа, а также нестандартный софт, то стоит обратить внимание на виртуальный сервер.

Облачный хостинг хорош для клиентов, которым критически важна постоянная работа сайта. Например, для интернет-магазинов. Выделенные серверы часто используют крупные компании, которым нужен максимум ресурса сервера. Также на них часто размещают крупные игровые проекты.

Подобрать хостинг можно в поисковой выдаче или на сайтах с рейтингами — например, на [hostings.info](https://ru.hostings.info/hostings/country/russia" \t "_blank). По данным этого ресурса, семь лучших хостеров России на январь 2023 года такие:

- [Fornex](https://fornex.com/ru/) — на рынке с 2007 года, 110 000 клиентов.

- [Ru-Center](https://www.nic.ru/) — на рынке с 2001 года, 900 000 клиентов.

- [AdminVPS](https://adminvps.ru/) — на рынке с 2012 года, 21 000 клиентов.

- [Timeweb](https://timeweb.com/ru/) — на рынке с 2006 года, 165 000 клиентов.

- «[Спрингхост](https://sprinthost.ru/" \t "_blank)» — на рынке с 2005 года, 358 669 клиентов.

- [Beget](https://beget.com/ru) — на рынке с 2009 года, 200 000 клиентов.

- [Reg.ru](https://www.reg.ru/) — на рынке с 2006 года, 370 0000 клиентов.

# ГЛАВА 3 Разработка Web-сайта предприятия

**3.1   Постановка комплекса задач разработки**

Ключевым моментом при создании web-сайта является постановка задачи. Были поставлены следующие задачи:

- Сайт должен улучшать имидж предприятия

- Сайт должен быть управляемым на уровне опытного пользователя

- Сайт должен быть управляемым на уровне обычного пользователя

- Сайт должен привлекать новых клиентов не из числа интернет-пользователей

- Сайт должен предлагать обслуживание клиентов на основании публичной оферты, а также возможность оформить заявку на услуги в режиме онлайн

- Сайт должен выступать средством маркетингового анализа потребностей потенциальных клиентов (например, какая из услуг наиболее востребована среди интернет - пользователей)

- Сайт должен функционировать с использованием дополнительных трудозатрат сотрудников компании

- Сайт должен функционировать без использования дополнительных трудозатрат сотрудников компании

- Сайт должен отражать достоверную и нужную информацию для его пользователей, иметь удобный графический интерфейс, а также иметь функцию обратной связи

**3.2 Технологии, используемые для разработки сайта**

## 3.2.1 Разработка frontend части

Оптимальным фреймворком был выбран Vue.js.

Vue (произносится /vjuː/, примерно как **view**) — JavaScript фреймворк для создания пользовательских интерфейсов. Он создан на стандартах HTML, CSS и JavaScript и предоставляет декларативную и компонентную модель программирования, которая помогает эффективно разрабатывать пользовательские интерфейсы, будь то простые или сложные.

Вот маленький пример:

*import { createApp } from 'vue'*

*createApp({*

*data() {*

*return {*

*count: 0*

*}*

*}*

*}).mount('#app')*

*<div id="app">*

*<button @click="count++">*

*Счётчик: {{ count }}*

*</button>*

*</div>*

Пример выше демонстрирует две главные фундаментальные

особенности Vue:

Декларативная отрисовка: Vue расширяет стандартный HTML синтаксисом шаблонов, который позволяет декларативно описывать финальный HTML на основе состояния JavaScript.

Реактивность: Vue автоматически отслеживает изменения состояния JavaScript и эффективно обновляет DOM, когда происходят изменения.

Прогрессивный фреймворк[​](https://ru.vuejs.org/guide/introduction.html" \l "the-progressive-framework)

Vue — фреймворк и экосистема, охватывающая большинство функций, необходимых при разработке фронтенда. Однако веб чрезвычайно разнообразен — вещи, которые создаём в интернете, могут сильно отличаться и по форме и по масштабу. Учитывая это, Vue спроектирован так, чтобы быть гибким и постепенно адаптируемым. В зависимости от сценария использования Vue можно использовать по-разному:

- Улучшать статический HTML без добавления этапа сборки

- Встраиваться в качестве веб-компонентов на любой странице

- Одностраничные приложения (SPA)

- Fullstack / Отрисовка на стороне сервера (SSR)

- Jamstack / Генерация статического сайта (SSG)

- Нацеленность на настольные, мобильные устройства, WebGL и даже терминалы

Несмотря на гибкость, основные знания о том, как работает Vue будут общими для всех сценариев использования. Даже новичкам знания, полученные на этом пути, пригодятся по мере роста и достижения более амбициозных целей в будущем. Опытные разработчики смогут выбрать оптимальный способ использования Vue в зависимости от проблем, которые требуется решить, сохраняя при этом прежнюю производительность. Вот почему Vue называют «Прогрессивным фреймворком»: это фреймворк, который может расти вместе с вами и адаптироваться к вашим потребностям.

Однофайловые компоненты[​](https://ru.vuejs.org/guide/introduction.html#single-file-components)

В большинстве проектов Vue, где есть шаг сборки, компоненты Vue создаются с использованием файлов HTML-подобного формата, называемого **однофайловыми компонентами** (также известного как \*.vue файлы, сокращённо называемых **SFC**). Однофайловые компоненты Vue, как следует из названия, объединяют в себе логику компонента (JavaScript), шаблон (HTML) и стили (CSS) в одном файле. Вот предыдущий пример, написанный в формате однофайлового компонента:

*<script>*

*export default {*

*data() {*

*return {*

*count: 0*

*}*

*}*

*}*

*</script>*

*<template>*

*<button @click="count++">Счётчик: {{ count }}</button>*

*</template>*

*<style scoped>*

*button {*

*font-weight: bold;*

*}*

*</style>*

Однофайловые компоненты — отличительная особенность Vue и рекомендуемый способ создания компонентов Vue, **если** сценарий использования предполагает шаг сборки.

Стили API[​](https://ru.vuejs.org/guide/introduction.html#api-styles)

Компоненты Vue можно создавать с использованием двух различных стилей: **Options API** и **Composition API**.

Options API[​](https://ru.vuejs.org/guide/introduction.html#options-api)

При использовании Options API логика компонентов определяется с помощью объекта опций, таких как data, methods и mounted. Свойства, определённые в опциях, доступны через this внутри функций и указывают на экземпляр компонента:

*<script>*

*export default {*

*// Свойства возвращаемые из data() становятся реактивным*

*// состоянием компонента и к ним можно обращаться через `this`.*

*data() {*

*return {*

*count: 0*

*}*

*},*

*// Методы это функции, которые мутируют состояние и вызывают обновления.*

*// Они могут быть привязаны в качестве обработчиков событий в шаблонах.*

*methods: {*

*increment() {*

*this.count++*

*}*

*},*

*// Хуки жизненного цикла вызываются на разных этапах*

*// жизненного цикла компонента.*

*// Эта функция будет вызвана после монтирования компонента.*

*mounted() {*

*console.log(`Стартовое значение счётчика — ${this.count}.`)*

*}*

*}*

*</script>*

*<template>*

*<button @click="increment">Счётчик: {{ count }}</button>*

*</template>*

Composition API[​](https://ru.vuejs.org/guide/introduction.html#composition-api)

С помощью Composition API мы определяем логику компонента, используя импортированные функции API. В однофайловых компонентах, Composition API обычно используется с синтаксисом [<script setup>](https://ru.vuejs.org/api/sfc-script-setup.html). Атрибут setup — подсказка компилятору для выполнения специальных преобразований на этапе компиляции, что позволяет использовать Composition API с меньшим количеством кода. Например, импорты и переменные / функции, объявленные на верхнем уровне в <script setup> можно использовать напрямую в шаблоне.

Вот тот же компонент, с точно таким же шаблоном, но написанный с использованием Composition API и <script setup>:

*<script setup>*

*import { ref, onMounted } from 'vue'*

*// реактивное состояние*

*const count = ref(0)*

*// функции которые мутируют состояние и вызывают обновления*

*function increment() {*

*count.value++*

*}*

*// хуки жизненного цикла*

*onMounted(() => {*

*console.log(`Стартовое значение счётчика — ${count.value}.`)*

*})*

*</script>*

*<template>*

*<button @click="increment">Счётчик: {{ count }}</button>*

*</template>*

Оба стиля API способны полностью покрыть общие случаи использования. Это разные интерфейсы, работающие на основе одной и той же базовой системы. Фактически, Options API реализован на основе Composition API! Фундаментальные концепции и знания о Vue являются общими для двух стилей.

Options API сосредоточен вокруг концепции «экземпляра компонента» (this, как показано в примере), что обычно лучше согласуется с моделью мышления, основанной на классах, для пользователей с опытом работы с языками ООП. Он также более удобен для новичков, поскольку абстрагируется от деталей реактивности и обеспечивает организацию кода с помощью групп опций.

Composition API сосредоточен на объявлении переменных реактивного состояния непосредственно в области видимости функции, а также на композиции состояния из нескольких функций для решения сложных задач. Это более свободная форма, но для эффективного использования требующая понимания того, как работает реактивность во Vue. В свою очередь, его гибкость позволяет использовать более мощные шаблоны для организации и переиспользования логики.

Vue.js был выбран за его компонентный подход к созданию пользовательского интерфейса, использование директив и реактивных свойств для управления динамическим контентом, а также удобной интеграции с другими библиотеками и фреймворками.

Веб-приложение с использованием Vue.js состоит из компонентов, которые можно сравнить с независимыми блоками кода, отвечающими за отображение, логику и данные. Компоненты могут быть переиспользованы в разных частях приложения, что делает код более модульным и поддерживаемым.

Для создания интерфейса веб-приложения на Vue.js были использованы HTML-шаблоны, содержащие директивы Vue.js и JavaScript-код, отвечающий за логику и управление состоянием приложения.

Установка Vue CLI

*npm install -g @vue/cli*

Данный тип установки позволяет использовать компонентный подход в разработке веб приложения.

После установки создается папка в которой преднастроены все необходимые модули.

Запуск разработки производится командой:

*npm run serve*

**3.2.2 Хостинг**

Оптимальным хостингом был выбран Reg.ru

Reg.ru - российский [регистратор доменных имён](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%BE%D1%80_%D0%B4%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D1%85_%D0%B8%D0%BC%D1%91%D0%BD) и [хостинг-провайдер](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F" \o "Хостинговая компания), аккредитованный регистратор в доменах [.ru](https://ru.wikipedia.org/wiki/.ru) и [.рф](https://ru.wikipedia.org/wiki/.%D1%80%D1%84). На июль 2017 года компания обслуживала около 3 миллионов доменов и предоставляла хостинг для около 400 тысяч доменных имён.

История:

Компания Рег.ру появилась 22 мая 2006 года. Основателями выступили Алексей Королюк, до того возглавлявший студию веб-дизайна «ЦЭТИС», Филипп Гросс-Днепров, профессионально занимавшийся скупкой и перепродажей доменов, а также программист Валерий Студенников. Идея заняться регистрацией и обслуживанием доменных имён принадлежала Гросс-Днепрову, он же предоставил домен Рег.ру. Королюк планировал сконцентрироваться на услугах хостинга. В итоге компания сфокусировалась на этих двух направлениях.

В июле 2012 Рег.ру получила аккредитацию [ICANN](https://ru.wikipedia.org/wiki/ICANN).

В 2009—2010 годах Рег.ру выиграла в судах нескольких инстанций дело против компании «Хостинг-центр», незаконно использовавшей фирменный знак-«домик».

Для выхода на международный рынок в октябре 2012 года у EuroDNS был выкуплен домен Reg.com, который, по данным компании, обошёлся в 360 тысяч евро. К этому времени Рег.ру обслуживала 1,3 миллиона доменов.

В начале 2014 года компания провела ребрендинг: изменила логотип и обновила сайт.

В октябре 2015 года Координационный центр национального домена сети интернет назначил Алексея Королюка председателем Комитета регистраторов зон .ru и .рф.

На начало 2017 года Рег.ру является крупнейшим российским аккредитованным регистратором доменных имён и хостинг-провайдером. Офисы компании расположены в более чем 30 городах.

Домены.

В 2007 году компания получила аккредитацию [Координационного центра национального домена сети Интернет](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BE%D1%80%D0%B4%D0%B8%D0%BD%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D1%86%D0%B5%D0%BD%D1%82%D1%80_%D0%BD%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%B3%D0%BE_%D0%B4%D0%BE%D0%BC%D0%B5%D0%BD%D0%B0_%D1%81%D0%B5%D1%82%D0%B8_%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82) в зонах [.ru](https://ru.wikipedia.org/wiki/.ru) и [.su](https://ru.wikipedia.org/wiki/.su).

2 ноября 2009 года Рег.ру первой получила аккредитацию для кириллической зоны [.РФ](https://ru.wikipedia.org/wiki/.%D0%A0%D0%A4). В 2010 году компания регистрировала домены в более чем 150 доменных зонах и обслуживала 500 тысяч доменных имён.

Компания росла и за счет поглощений: в декабре 2010 года был приобретён регистратор «Клевер Телеком», в следующем году Рег.ру получила доменные имена от регистратора «Регионтелеком» и 18 тысяч клиентов Reseller Club.

В марте 2011 года Рег.ру разработала логотип и фирменный стиль кириллической доменной зоны .рф, а также запустила серию телефонов SimaPhone.РФ на базе [iPhone 4](https://ru.wikipedia.org/wiki/IPhone_4" \o "IPhone 4) с дизайном [Дениса Симачёва](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%B0%D1%87%D1%91%D0%B2,_%D0%94%D0%B5%D0%BD%D0%B8%D1%81_%D0%9D%D0%B8%D0%BA%D0%BE%D0%BB%D0%B0%D0%B5%D0%B2%D0%B8%D1%87). В ноябре того же года компания была номинирована на «Премию Рунета» в номинации «За вклад в развитие домена .рф».

В 2011 году на сайте компании запустилась площадка по вторичной купле-продаже доменов «Магазин доменов» (на июль 2016 года было выставлено 105 тысяч лотов). По итогам 2011 года Рег.ру стала лидером по объёму регистраций в российских национальных доменных зонах, уступая лидерство по числу регистрируемых доменов только компании-регистратору [Ru-Center](https://ru.wikipedia.org/wiki/Ru-Center" \o "Ru-Center).

В 2012 году компания стала аккредитованным ICANN регистратором в зонах [.aero](https://ru.wikipedia.org/wiki/.aero), [.asia](https://ru.wikipedia.org/wiki/.asia), [.biz](https://ru.wikipedia.org/wiki/.biz), [.cat](https://ru.wikipedia.org/wiki/.cat), [.com](https://ru.wikipedia.org/wiki/.com), [.coop](https://ru.wikipedia.org/wiki/.coop), [.info](https://ru.wikipedia.org/wiki/.info), [.jobs](https://ru.wikipedia.org/wiki/.jobs), [.mobi](https://ru.wikipedia.org/wiki/.mobi), [.museum](https://ru.wikipedia.org/wiki/.museum), [.name](https://ru.wikipedia.org/wiki/.name), [.net](https://ru.wikipedia.org/wiki/.net), [.org](https://ru.wikipedia.org/wiki/.org), [.pro](https://ru.wikipedia.org/wiki/.pro), [.tel](https://ru.wikipedia.org/wiki/.tel), [.travel](https://ru.wikipedia.org/wiki/.travel), [.xxx](https://ru.wikipedia.org/wiki/.xxx).

В 2012 году компания была номинирована на «[Премию Рунета](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%BC%D0%B8%D1%8F_%D0%A0%D1%83%D0%BD%D0%B5%D1%82%D0%B0)» в номинации «.РФ и не только». В 2013 году Рег.ру заняла первое место по количеству зарегистрированных доменов в зоне .ru.

В 2014 году регистратор подписал соглашение об аккредитации с регистратурами доменных зон [.дети](https://ru.wikipedia.org/w/index.php?title=.%D0%B4%D0%B5%D1%82%D0%B8&action=edit&redlink=1), [.москва](https://ru.wikipedia.org/w/index.php?title=.%D0%BC%D0%BE%D1%81%D0%BA%D0%B2%D0%B0&action=edit&redlink=1) и [.moscow](https://ru.wikipedia.org/w/index.php?title=.moscow&action=edit&redlink=1), а также [.рус](https://ru.wikipedia.org/wiki/.%D1%80%D1%83%D1%81).

В 2015 году Рег.ру опередила Ru-Center по числу обслуживаемых доменов (31,8 % рынка). Весной 2016 года у регистратора Ru-Center изменились условия работы с клиентами. Благодаря этому Рег.ру только за март получил около 40 тысяч доменов на обслуживание, перенесённых от Ru-Center.

Общее число доменов к концу 2016 года превысило 2,7 миллионов, а к середине 2017 достигло 3 миллионов. На май 2017 Рег.ру занимал 16-е место в мировом рейтинге среди регистраторов доменов в новых зонах.

Хостинг.

В 2008 году Рег.ру получила лицензию на предоставление хостинг-услуг.

В феврале 2013 года стал предоставляться [Windows-хостинг](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3" \l "Windows-%D1%85%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3" \o "Виртуальный хостинг). В мае 2013 года компания запустила облачные сервисы на базе хостинг-платформы [Jelastic](https://ru.wikipedia.org/wiki/Jelastic" \o "Jelastic), позволяющие запускать и масштабировать веб-приложения на [Java](https://ru.wikipedia.org/wiki/Java" \o "Java) и [PHP](https://ru.wikipedia.org/wiki/PHP). В 2014 году совместно с [DDoS Guard](https://ru.wikipedia.org/wiki/DDoS_Guard" \o "DDoS Guard) компания предоставила клиентам услугу бесплатной защиты сайта от [DDoS-атак](https://ru.wikipedia.org/wiki/DoS-%D0%B0%D1%82%D0%B0%D0%BA%D0%B0).

В сентябре 2011 года Рег.ру и крупнейший в России хостинг-провайдер «[Мастерхост](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D1%82%D0%B5%D1%80%D1%85%D0%BE%D1%81%D1%82" \o "Мастерхост)» объявили о планах создать объединённую компанию. Сообщалось, что решение одобрено акционерами, а организационно-правовые вопросы будут решены позднее. Однако в ноябре основатель и генеральный директор «Мастерхоста» сообщил о продаже своей доли в компании и покинул пост генерального директора. Поскольку инициатором слияния со стороны «Мастерхоста» был именно Леонид Филатов, сделка не состоялась.

Рег.ру наращивала базу клиентов за счёт переноса клиентов от сторонних компаний: в декабре 2012 года хостер RedSoft (проект Joomla.ru) передал хостинг-инфраструктуру проекта Hostingjoomla.ru со всей клиентской базой, в июне 2013 года хостинг-провайдер 100MB.RU передал на сопровождение услуг хостинга более трех тысяч клиентов, в ноябре 2014 компания заключила соглашение о переносе клиентов от хостинг-провайдера Logol.

В декабре 2016 года стало известно о приобретении одного из старейших хостинг-провайдеров [Agava](https://ru.wikipedia.org/wiki/Agava" \o "Agava). Планировалось, что объединение компаний завершится до конца 2016 года, в 2017 будет принято решение о сохранении торговых марок Agava. В январе 2017 года было объявлено, что дата-центр Telehouse Caravan вливается в Рег.ру с сохранением бренда. В феврале к компании присоединился хостинг-дискаунтер Mne.ru.

В январе 2017 года Рег.ру арендовала изолированной машинный зал в московском дата-центре уровня Tier III компании DataPro.

По данным Рег.ру, на декабрь 2016 года компания обслуживала более 370 тысяч доменных имён на своём хостинге (11 % рынка).

14 декабря 2017 года компания запустила новое поколение облачных VPS.

|  |
| --- |
|  |
|  |

**Заключение**

Стремительное развитие общества привело к росту потребности в информации, увеличению количества новых пользователей сети, что в свою очередь способствовало глобальной информатизации. И обеспечение пользователей достоверной и полной информацией стало одной из главных задач специалистов в области информационных технологий, путем создания простого и удобного доступа к ней.

В ходе проделанной работы были исследованы возможные подходы к разработке веб-сервисов, подобраны подходящие для решаемой задачи инструменты. В эпоху бурного развития информационных технологий, электронное представительство в сети Интернет необходимо любой организации. Web-сайт позволяет решать целый ряд разнообразных задач, служит визитной карточкой учреждения, привлекает дополнительное внимание целевой аудитории.

Однако после разработки и размещения в Сети работа над сайтом вовсе не заканчивается. Анализ ситуации в Интернет показал, что сайты, которые вовремя не обновляются и не обеспечивают пользователей актуальной и своевременной информацией, быстро теряют свою аудиторию. Несоответствие содержания сайта требованиям пользователей – довольно распространенная проблема, которой посвящены многие исследования. Также довольно остро стоит проблема навигации по сайту: даже необходимая информация совершенно ни к чему на сайте, если ее невозможно найти. Главной задачей при создания сайта, в первую очередь следует обращать внимание на его структуру и информационную составляющую. Но есть и еще важнейший аспект, который нельзя обойти вниманием: эффективность поиска сайта в Интернет. При недостаточной раскрутке сайта его бывает невозможно найти в поисковых системах даже при прямом запросе. Для того, чтобы сайт выполнял имиджевую и рекламную функцию, необходимо с особой тщательностью подходить к регистрации его в каталогах и поисковых машинах, подбору ключевых слов и составлению описания. Web-сайт может достойно представлять организацию в сети Интернет, но только в том случае, если по нему постоянно ведется кропотливая работа.

Разработанный нами web-сайт, включает в себя полную информацию о компании, виде деятельности и оказываемых услугах. На web-сайте имеется информация с контактами и реквизитами компании, а так же отображение территориального расположения компании и схемы проезда, и иной контактной информацией.

Таким образом, в ходе выполнения дипломной работы была достигнута цель дипломного проекта, заключавшаяся в разработке и реализации web-сайта для ломбарда, чему способствовало выполнение следующих задач:

-   провести анализ предметной области;

-   выявить функциональные возможности веб-сайтов и их применение;

-   спроектировать web-интерфейс динамического сайта;

-   осуществить анализ и выбор инструментальных средств;

-   осуществить программную реализацию динамического сайта.

В ходе исследования была изучена  предметная область, а так же инструменты, необходимые для разработки web-сайта.

# Список использованных интернет ресурсов

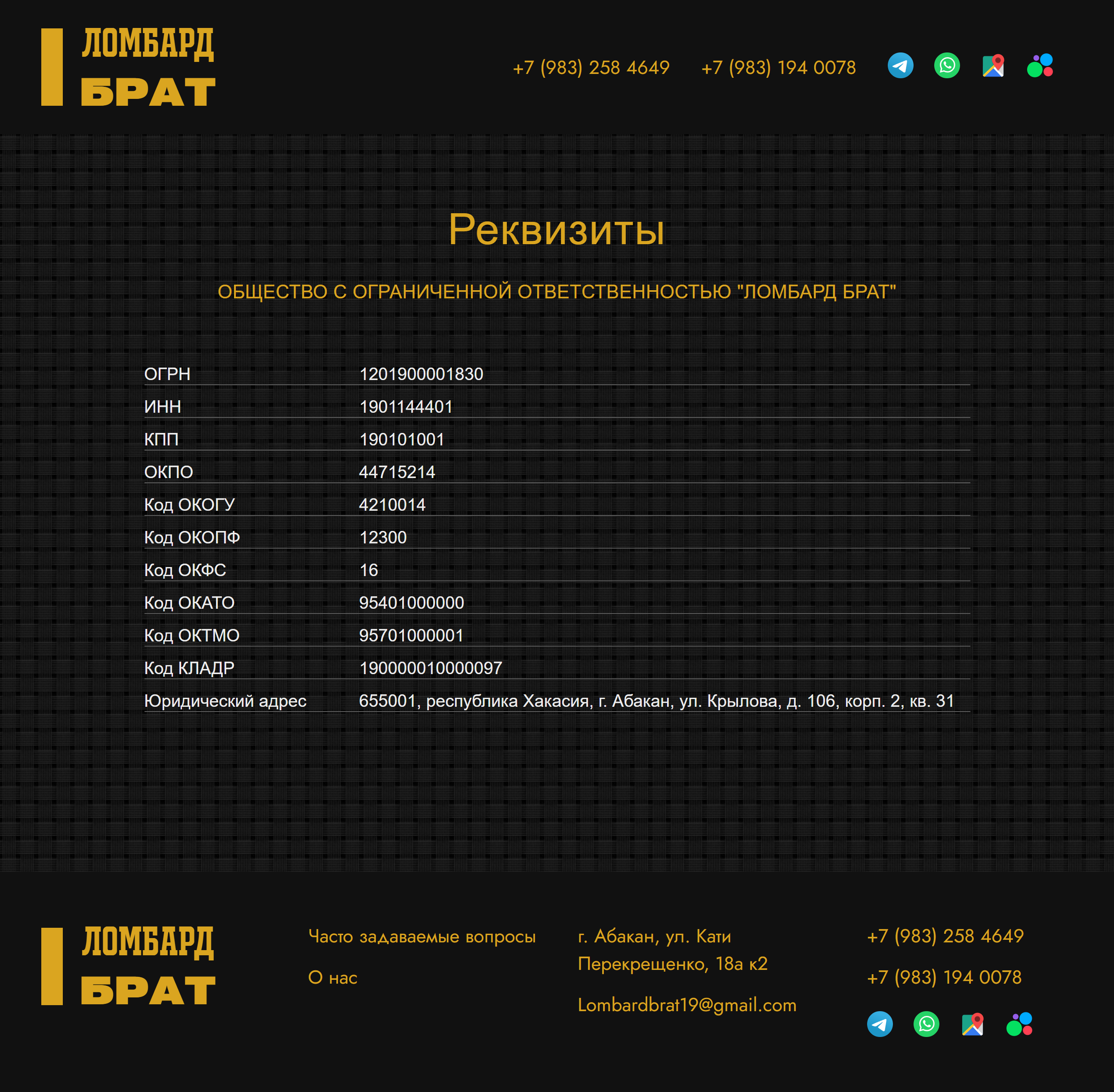
1. <https://gb.ru/>
2. <https://ru.vuejs.org/>
3. <https://www.reg.ru/>
4. <https://habr.com/ru/feed/>
5. <https://router.vuejs.org/>
6. <https://learn.javascript.ru/>
7. <https://dimox.name/>
8. <https://icon-icons.com/ru/>
9. <https://www.11zon.com/ru/>
10. <https://thecode.media/maps/>
11. <https://coolors.co/>
12. <https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>

**Приложение**

Вид главной страницы



Вид страницы «О нас»



Вид страницы «Часто задаваемые вопросы»

