

Введение

Современный мир не возможно представить без технологий, без нескончаемого информационного потока, информация окружает нас повсюду, в любой сфере нашей жизни. С появлением интернета этой информации стало еще больше, стало гораздо проще получать ее и доставлять. Предпринимателям стало гораздо проще находить клиентов, клиентам, нужные товары или услуги. В этом им помогают веб ресурсы, приложения. Веб-приложения - это программное обеспечение, доступное в сети интернет, которое позволяет пользователям получать доступ к различным функциям и сервисам. Они могут быть использованы в различных сферах, начиная от малого бизнеса и заканчивая крупными международными компаниями. Создание веб-приложений становится всё более и более востребованным в мире информационных технологий, поскольку это один из наиболее эффективных способов показать и рассказать потенциальным клиентам о себе. В отчете будет рассмотрено, какие факторы следует учитывать при разработке веб-приложений, и какие технологии, и языки программирования могут использоваться для их создания.

1 Языки верстки и программирования

Для создания веб-приложений необходимо владеть языками верстки и программирования. Существует множество языков программирования, но наиболее часто используемыми являются следующие:

HTML (HyperText Markup Language) - это язык разметки, используемый для создания структуры и содержания веб-страниц. Он используется вместе с CSS и JavaScript для создания пользовательских интерфейсов.

CSS (Cascading Style Sheets) - это язык, используемый для создания стилей и внешнего вида веб-страниц. Он позволяет разработчикам определить цвет, шрифт, размер и расположение элементов на странице.

JavaScript - это язык программирования, используемый для создания интерактивных элементов на веб-страницах. Он позволяет создавать динамические интерфейсы, выполнять анимации, валидацию форм и другие функции.

PHP (Hypertext Preprocessor) - это язык программирования, используемый для создания серверных приложений. Он позволяет создать динамические веб-сайты, обрабатывать формы, управлять базой данных и многое другое.

Python - это высокоуровневый язык программирования, который может использоваться для создания веб-приложений с помощью фреймворков, таких как Django или Flask. Он обладает широким спектром функций и библиотек, которые могут быть использованы для создания сложных веб-приложений.

Ruby - это язык программирования, который используется для создания веб-приложений, основанных на Ruby on Rails. Он обладает элегантным синтаксисом и прост в использовании.

Каждый из этих языков программирования имеет свои преимущества и недостатки, и оптимальный выбор зависит от конкретных требований проекта.

1.1 HTML

HTML — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS либо открывают с локального диска, затем интерпретируют код в интерфейс, который будет изображаться на экране монитора.

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML различные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть интегрированы в отображаемую страницу. HTML выдает средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с применением угловых скобок. Такие теги, как `` и `<input />`, напрямую вводят содержание на страницу. остальные теги, такие как `<p>`, обрамляют и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не показывают HTML-теги, но используют их для интерпретации содержимого страницы. Также есть возможность встроить код JavaScript в HTML, чтобы управлять поведением и содержанием веб-страницы. Кроме того, включение CSS в HTML позволяет описывать внешний вид и макет страницы.

1.2 CSS

CSS — это формальный язык описания внешнего вида документа (веб-страницы), написанный с применением языка разметки (чаще всего HTML либо XHTML). Его также можно применить к любому XML-документу, таким как SVG или XUL. CSS используется создателями веб-страниц для определения цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основная дизайнерская цель CSS — отгородить и отделить описание логической структуры веб-страницы (которое делается с помощью HTML или других языков разметки) от описания

внешнего вида этой веб-страницы (которое сейчас делается с помощью формальный язык CSS). Это разделение может увеличить доступность документа, обеспечить большую гибкость и контроль над его представлением, а также уменьшить сложность и повторение структурного содержимого.

1.3 Java Script

JavaScript — мультипарадигмальный язык программирования. Он поддерживает объектно-ориентированный, императивный и функциональный стили. Это реализация спецификации ECMAScript. JavaScript обычно используется в качестве встраиваемого языка для программного доступа к объектам приложения. Он широко используется в браузерах в качестве языка сценариев для создания интерактивных веб-страниц. Основные архитектурные особенности: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как первоклассные объекты.

1.4 Конструкторы

Конструкторы сайтов (или веб-конструкторы) – это программы или онлайн-сервисы, которые позволяют создавать сайты без необходимости знаний программирования. Они обычно работают в режиме "what you see is what you get" (WYSIWYG), что означает, что пользователь может работать с элементами сайта и редактировать его без необходимости написания кода.

Список популярных конструкторов сайтов включает технические и профессиональные инструменты, такие как:

1. Wix
2. Weebly
3. Squarespace

4. Shopify
5. WordPress.com
6. GoDaddy
7. Jimdo
8. Duda
9. Webflow
10. Tilda

Плюсы работы с онлайн конструктором сайтов:

1. Простота в использовании, не нужно знание HTML/CSS кода
2. Наполнение сайта контентом
3. Количество готовых шаблонов и макетов
4. Быстрота создания сайта
5. Низкая стоимость

Минусы:

1. Ограничение на дизайн, не всегда можно реализовать свои идеи.
2. Базовая оптимизация для SEO
3. Ограниченный функционал, возможны проблемы с интеграцией сторонних сервисов/платежных систем.
4. Зависимость от хостинг-провайдера.

В целом, конструкторы сайтов – это отличное решение для малого бизнеса и личных блогов. Они обеспечивают простоту в использовании и быстрый способ создания сайта, однако не всегда могут удовлетворить потребности более серьезных проектов.

Для создания нашего сайта был выбран сервис Tilda, так как для создания простого сайта он является лучшим вариантом.

2 Разработка веб-приложения

Разработка структуры сайта является одним из ключевых моментов его создания, во многом определяющим эффективность его работы в дальнейшем. С технической точки зрения структура сайта есть не что иное, как все его страницы, расположенные в порядке важности для пользователей, т.е. иерархия.

При разработке структуры будущего сайта очень важно учитывать, что сегодня в выигрыше остаются, в первую очередь, те интернет-ресурсы, которые могут предоставить своим пользователям быстрый доступ к любой информации, которая находится на их страницах.

Структура сайта фото-салона:

1. Главная

В которой будет располагаться шапка сайта с логотипом и навигацией по сайту. Также на главной странице будет расположена информация о салоне.

2. Галерея работ

Этот раздел является логическим продолжением предыдущего, в нем будут расположены три блока с работами мастера по пирсингу, готовыми татуировками, а также эскизы.

3. “Подвал” сайта, в котором будут расположены ссылки на социальные сети и номер для связи, а также ссылка на начало страницы.

3 Оптимизация веб-приложения

Оптимизация веб-приложений - это процесс улучшения быстродействия и производительности веб-сайта или приложения путем снижения времени загрузки страниц, уменьшения объема передаваемых данных, улучшения работы сервера и браузера.

Основные задачи при оптимизации веб-приложений:

Уменьшение времени загрузки страниц, что включает оптимизацию размера файлов, использование кэширования браузера, минификацию CSS, HTML и JavaScript.

Сокращение объема передаваемых данных. Достичь с помощью сжатия данных, удаления лишних элементов на странице, использования спрайтов и т.д.

Улучшение работы сервера. Может быть достигнуто путем настройки сервера, использованием кэша и сокращением времени обработки запросов.

Оптимизация работы браузера. Включает сокращение числа запросов, минимизацию использования плагинов и включение компрессии данных в браузере.

Процесс оптимизации веб-приложения начинается с анализа его состояния и проблем, с которыми сталкиваются пользователи. Далее создается план оптимизации, который включает определение задач, которые нужно решить, а также методов и инструментов, которые будут использованы для улучшения работы приложения.

Оптимизация веб-приложений может значительно ускорить работу приложения и улучшить пользовательский опыт. Это может быть особенно важно для онлайн-торговли, где быстрой загрузкой сайта может зависеть от продаж и лояльности клиентов.

Заключение

Во время прохождения практики и написания отчета я сделал выводы, что данная сфера является одной из самых актуальных и востребованных на современном рынке IT. В ходе выполнения работы было изучено множество современных технологий и инструментов, позволяющих упростить и ускорить процесс разработки, а также обнаружены некоторые особенности и проблемы, связанные с этим видом деятельности. Кроме того, были рассмотрены такие направления, как frontend-разработка, backend-разработка, базы данных и тестирование, что дало полное представление о разных аспектах создания веб-приложений.

В целом, выполнение данной практики позволило расширить знания и навыки в области разработки веб-приложений и получить практический опыт работы с современными технологиями и инструментами. Также было установлено, что успех любого проекта зависит от грамотного планирования, хорошо отлаженной командной работы и постоянного обучения.

Список использованных источников

1.

