Муниципальное бюджетное общеобразовательное учреждение

«Красносельская средняя школа»

Красносельского муниципального района Костромской области

Проект по информатике на тему:

**«Веб-разработка. Разработка сайтов и веб-приложений»**

Выполнил: Сухобаевский Дмитрий Игоревич

учащийся 10 «Б» класса

Проверила: Незамаева Наталья Викторовна

учитель информатики

П. Красное – на – Волге

2024 год

**Оглавление**

Введение………………………………………………………………………………………………..3

Глава 1. Что такое интернет и как он работает……………………………………..4

Глава 2. Понятие веб-сайта и веб-страницы…………………………………………5

Глава 3. Структура сайта и как они работают…………………………………….6-7

Глава 4. Процесс создания сайтов……………………………………………………8-10

Глава 5. Методы и технологии для создания сайтов и веб-приложений……………………………………………………………………………………11-14

Подведение итогов………………………………………………………………………………15

Список источников……………………………………………………………………………….16

**Введение**

Я выбрал эту тему веб-разработки для своего проекта по нескольким причинам. Прежде всего, веб-разработка является для меня одним из самых увлекательных видов деятельности, с которым я собираюсь связать свою жизнь.

Помимо этого, веб-разработка является одним из наиболее актуальных и востребованных направлений в сфере информационных технологий.

Она предлагает огромную свободу для творчества благодаря мощным инструментам и технологиям, а также предлагает непрерывное обучение и развитие.

**Цель работы:** Познакомиться с профессией веб-разработчик и инструментами и методами для создания сайтов и веб-приложений.

**Задачи:**

1. Познакомиться с интернетом и тем, как он работает

2. Изучить сайты и их структуру

3. Узнать, как создаются сайты и какие технологии для этого используются

4. Подвести итоги

**Объект исследования:** Процесс создания веб-сайтов и его структура

**Предмет исследования:** Интернет-технологии и методы разработки веб-сайтов

**Используемые методы:** открытые документации, анализ существующих веб-сайтов, сравнительный анализ и тд.

**Глава 1. Что такое интернет и как он работает**

Веб-разработка напрямую связана с интернетом, поэтому перед тем как изучить тонкости профессии, мы должны больше узнать об интернете и его работе.

Интернет - это огромная сеть связанных компьютеров, которая позволяет нам общаться, обмениваться информацией и находить всевозможные ресурсы в онлайн-пространстве.

Основой функционирования интернета является передача данных по сети. Данные передаются с помощью протокола TCP/IP, который разбивает информацию на маленькие пакеты и отправляет их по сети. Эти пакеты доставляются посредством различных узлов и маршрутизаторов, которые направляют их на нужные адреса.

Ключевым элементом взаимодействия в интернете являются адреса IP. Каждому устройству, подключенному к сети, присваивается уникальный IP-адрес, который позволяет идентифицировать устройство в сети и обеспечивает точечную доставку данных.

Браузеры, такие как Google Chrome, Mozilla Firefox и Safari, позволяют нам взаимодействовать с интернетом. Они служат в качестве посредников между нами и веб-сайтами, позволяя открывать страницы, отправлять запросы и получать ответы от серверов.

Запросы и ответы проходят через множество промежуточных устройств, называемых маршрутизаторами, которые определяют оптимальный путь доставки данных. Веб-сайты хранятся на серверах, которые непрерывно подключены к интернету. Браузер получает HTML-код веб-страницы и отображает его пользователю. Интернет предоставляет большое количество информации, возможностей общения, развлечений и многого другого.

**Глава 2. Понятие веб-сайта и веб-страницы**

Веб-сайт является основным строительным блоком интернета и представляет собой коллекцию взаимосвязанных веб-страниц, объединенных общей тематикой или целью. Он служит средством передачи информации, обмена данными и коммуникации.

Веб-страница, в свою очередь, является основным элементом веб-сайта. Она представляет собой документ в формате HTML, содержащий текст, изображения, мультимедийные элементы и ссылки на другие веб-страницы. Веб-страницы отображаются веб-браузером, который интерпретирует HTML-код и отображает его в виде веб-страницы на экране пользователя.

Основная цель веб-страницы - предоставить информацию пользователю или выполнить определенные функции. Она может быть статичной, то есть содержать постоянный набор данных, или динамической, с возможностью изменения содержимого в зависимости от действий пользователя или других факторов.

Веб-сайты могут иметь различные типы, включая информационные порталы, интернет-магазины, блоги, социальные сети, форумы и многое другое. Они могут быть разработаны для разных целей: предоставление информации, продажи товаров и услуг, обмен мнениями и коммуникация с другими людьми. Веб-сайты могут быть общедоступными или доступными только для определенных пользователей с помощью авторизации и аутентификации.

**Глава 3. Структура сайтов и как они работают**

Краткое описание структуры сайта:

1. Домашняя страница: это главная страница сайта, на которой обычно содержится общая информация о компании или о теме сайта.

2. Навигационная панель: это меню или список ссылок, которые помогают пользователям перемещаться по сайту и найти нужную информацию.

3. Разделы или страницы: сайт может содержать различные разделы или страницы, каждая из которых посвящена определенной тематике или информации. Например, "О нас", "Услуги", "Контакты" и т. д.

4. Мультимедийный контент: сайт может включать изображения, видео, аудио или другие мультимедийные элементы, которые помогают представить информацию более наглядно и привлекательно.

5. Футер или подвал: это раздел внизу страницы, который обычно содержит дополнительную информацию о сайте, ссылки на социальные сети, контактные данные и другую вспомогательную информацию.

6. Другие функциональные элементы: в зависимости от типа сайта, он может включать дополнительные функциональные элементы, такие как формы обратной связи, корзины покупок в интернет-магазине, комментарии пользователей и т. д.

##### Принцип работы сайтов:

##### 1. Пользователь вводит адрес сайта в браузер.

##### 2. Браузер отправляет запрос на сервер, указывая адрес сайта.

##### 3. Сервер получает запрос и ищет соответствующие файлы и данные, связанные с сайтом.

##### 4. Сервер отправляет браузеру запрошенные файлы и данные.

##### 5. Браузер получает файлы и данные и отображает их на экране пользователя.

##### 6. Пользователь взаимодействует с сайтом, кликая на ссылки, заполняя формы или выполняя другие действия.

##### 7. При необходимости браузер отправляет дополнительные запросы на сервер для получения дополнительной информации или выполнения определенных действий.

##### 8. Сервер обрабатывает запросы, взаимодействует с базами данных или другими компонентами и возвращает результаты обратно в браузер.

##### 9. Процесс взаимодействия между браузером и сервером продолжается, пока пользователь не покинет сайт или не выполнится необходимое действие.

##### 10 .Весь этот процесс происходит в реальном времени, что позволяет пользователям просматривать, взаимодействовать и получать информацию на сайте.

**Глава 4. Процесс создания сайтов**

1. Анализ и планирование

Перед тем как приступить к созданию веб-сайта, необходимо провести анализ и планирование. Важно определить цели и задачи сайта, а также целевую аудиторию. Проанализировать другие сайты на такую же тематику, на наличие недочетов, чтобы не допустить их в своем проекте или наоборот привлекательных тонкостей, которые можно будет добавить в свой сайт

2. Веб-дизайн и макетирование

Веб-дизайн играет ключевую роль в создании привлекательного и функционального веб-сайта. Начать нужно с разработки концепции дизайна, определить цветовую палитру, шрифты и общий стиль сайта, который соответствует его целям. Создать макеты, используя графические редакторы или специализированные инструменты для макетирования веб-страниц.

Нужно убедиться, что дизайн сайта отзывчивый, то есть адаптированный для работы на различных устройствах, таких как настольные компьютеры, планшеты и смартфоны. Это важно, поскольку потенциальные пользователи могут посещать ваш сайт с разных устройств.

3. Веб-разработка

После завершения дизайна наступает этап веб-разработки. Веб-разработка включает в себя создание веб-страниц с использованием языков разметки, таких как HTML и CSS. Для этого используются текстовые редакторы или интегрированные среды разработки для написания кода.

Для более сложных и интерактивных функций на сайте, таких как формы обратной связи или динамическое содержимое, может потребоваться использование JavaScript или других языков программирования, таких как PHP или Python. В этой фазе также может понадобиться работа с серверами, базами данных и CMS (системами управления контентом), чтобы обеспечить функциональность сайта, управление содержимым и его обновление.

4. Тестирование и оптимизация

После того как сайт полностью разработан, следует приступить к тестированию. Необходимо проверить работоспособность сайта на разных браузерах и устройствах, чтобы убедиться, что он отображается корректно и функционирует без ошибок. Также нужно проверить все ссылки, формы, анимации и другие интерактивные элементы, чтобы убедиться, что они работают правильно.

Важно оптимизировать сайт для улучшения его производительности и скорости загрузки. Оптимизировать изображения, использовать кэширование, сжатие файлов и другие методы для сокращения времени загрузки страницы.

5. Развертывание и публикация

После успешного завершения тестирования и оптимизации сайта готов к развертыванию. Разработчику нужно выбрать хостинг-провайдера и зарегистрировать доменное имя для сайта. Далее загрузить все файлы веб-сайта на сервер и убедиться, что они работают должным образом.

6. Обслуживание и обновление

Создание сайта - это непрерывный процесс. После его публикации важно следить за работоспособностью и поддержкой сайта. Регулярно обновлять содержимое и информацию на веб-сайте, исправлять ошибки и обеспечивать безопасность сайта с помощью обновлений и резервного копирования данных.

Также нужно придерживаться последних тенденций веб-разработки и следить за новыми технологиями, чтобы сайт оставался современным и конкурентоспособным.

В заключение, создание сайта требует систематического подхода и учета множества аспектов, от анализа и планирования до разработки и тестирования и публикации.

**Глава 5. Методы и технологии для создания сайтов и веб-приложений**

Веб-разработчики делятся на 3 типа: frontend-, backend- и fullstack-разработчики.

Если говорить простым языком, frontend-разработчики отвечают за то, что пользователь видит на сайте (интерфейс), backend-разработчики отвечают за то, что пользователь не видит на сайте (работа с сервером), а fullstack-разработчики занимаются сразу и тем, и тем, что делает их более востребованными на рынке, чем просто фронтендеры и бэкэндеры.

Ниже приведены технологии, которые используются и во frontend-, и в backend-разработке.

1. HTML и CSS

HTML (HyperText Markup Language) и CSS (Cascading Style Sheets) являются основными языками разметки веб-страниц. HTML определяет структуру контента на странице, а CSS отвечает за его визуальное представление и стиль. Эти языки позволяют создавать семантически правильные и стилизованные веб-страницы.

HTML5 внедряет новые возможности, такие как аудио и видео элементы, семантические теги и многое другое. CSS3 предлагает расширенные возможности для стилизации, анимации и адаптивного дизайна.

2. JavaScript

JavaScript является одним из самых распространенных языков программирования, который используется для создания интерактивности на веб-страницах. Он позволяет добавлять динамическое содержимое, обрабатывать события, создавать анимации, выполнять валидацию форм и многое другое.

Фреймворки и библиотеки JavaScript, такие как React, Angular и Vue.js, значительно упрощают разработку сложных веб-приложений и обеспечивают повторное использование кода.

3. Responsive Web Design

С увеличением числа пользователей, обращающихся к веб-сайтам с мобильных устройств, адаптивный дизайн (Responsive Web Design) становится все более важным. Это подход, который позволяет создавать сайты, которые автоматически подстраиваются под разные

экраны и устройства, обеспечивая оптимальный UX-дизайн

Веб-разработчики используют гибкие сетки, медиа-запросы и другие техники для создания адаптивных сайтов. Благодаря этому, сайт будет хорошо выглядеть и работать на настольных компьютерах, планшетах и смартфонах.

4. Content Management Systems (CMS)

CMS - это системы управления контентом, которые позволяют управлять содержимым сайта без необходимости написания кода. Популярные CMS, такие как WordPress, Joomla и Drupal, предлагают широкий набор функций и плагинов для создания и управления различными типами веб-сайтов.

CMS облегчают работу с контентом, позволяют быстро вносить изменения и добавлять новое содержимое. Они также предоставляют возможности по управлению пользователями, настройке SEO и другими аспектами веб-проекта.

5. Progressive Web Apps (PWA)

Прогрессивные веб-приложения (Progressive Web Apps) сочетают в себе преимущества веб-сайтов и приложений. Они предлагают более богатый UX-дизайн, позволяя пользователю работать офлайн, получать уведомления и иметь интерфейс, приближенный к настольным приложениям.

PWA стремится достичь высокой производительности при загрузке и работе, используя кэширование и другие техники оптимизации. Это может быть полезным, особенно для мобильных пользователей, которым необходимо иметь быстрый доступ к приложению, даже при отсутствии сети.

6. API и интеграции

API (Application Programming Interface) позволяют веб-приложениям обмениваться данными с другими системами и сервисами. Например, можно API социальных сетей для входа через аккаунт социальной сети на каком-нибудь сайте или интегрировать платежные системы для принятия онлайн-платежей.

API также позволяют получать данные из различных источников, таких как геолокация, погода, новости и другие, чтобы обогатить содержимое сайта.

В заключение, методы и технологии для создания сайтов постоянно развиваются, открывая новые возможности. Это лишь некоторые из ключевых методов и технологий, которые могут быть применены в веб-проекте. Важно выбрать подходящие инструменты и технологии, и следовать лучшим практикам, чтобы создать профессиональный и успешный веб-сайт.

**Глава 6. Подведение итогов**

Мы погрузились в увлекательный мир интернета и изучили его основы. Мы начали с понимания того, что Интернет является глобальной сетью, объединяющей миллионы компьютеров по всему миру.

Мы также изучили структуру веб-сайтов и поняли, что они состоят из различных веб-страниц, которые связаны друг с другом. Основными языками разметки, используемыми для создания веб-страниц являются HTML и CSS. HTML определяет структуру контента на странице, а CSS отвечает за ее визуальное оформление и стиль.

Мы узнали с чего стоит начинать создание сайта, что нужно составить и проанализировать перед написанием кода.

Мы также затронули некоторые основные технологии и инструменты веб-разработки, такими как JavaScript для добавления интерактивности на страницы, CMS (системы управления контентом) для удобного управления содержимым сайта, а также API для интеграции с другими сервисами и системами.

**Список источников**

[**habr.co**](https://habr.com/ru/companies/yandex_praktikum/articles/584504/)**m**

[**github.com**](https://github.com/mdn?ysclid=lruyyvvb3c975364966)

[**ru.hexlet.io**](https://ru.hexlet.io/blog/posts/chto-takoe-frontend-razrabotka?ysclid=lruz5cyyqv920364672)

**itproger.com**

[**htmlacademy.ru**](https://htmlacademy.ru/blog/job/programming-start?ysclid=lruz6bymfo995135703)

[**skillbox.ru**](https://skillbox.ru/media/code/chem_zanimaetsya_veb_razrabotchik_i_chto_nuzhno_umet_dlya_pervoy_raboty/?ysclid=lruz6xc6th197610699)

[**developer.mozilla.org**](https://developer.mozilla.org/ru/docs/Learn)

[**ru.stackoverflow.com**](https://ru.stackoverflow.com/)

**r**[**u.wikipedia.org**](https://ru.wikipedia.org/wiki/Stack_Overflow)

[**vc.ru**](https://vc.ru/dev/840923-frontend-i-backend-razrabotka-chto-eto-v-chem-raznica-i-chto-vybrat)