**«Проектирование и разработка сайта с промоакциями средствами языка HTML, CSS и Java Script»**

СОДЕРЖАНИЕ  
ВВЕДЕНИЕ   
I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ  
1.1.Язык HTML

1.2. Средства языка CSS и Java Script

ВЫВОД  
II. ПРАКТИЧЕСКАЯ ЧАСТЬ   
2.1. План разработки веб-сайта с промоакциями средствами языка HTML, CSS и Java Script  
2.2. Анализ сайтов схожей тематики

ВЫВОД

III ТЕХНИКА БЕЗОПАСНОСТИ   
ВЫВОД  
ЗАКЛЮЧЕНИЕ   
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

**I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1.Язык HTML**

Информация, доступная пользователям Internet, располагается на компьютерах (web-серверах), на которых установлено специальное программное обеспечение. Значительная часть этой информации организована в виде web-сайтов.

Каждый из них имеет свое имя (адрес) в Internet. Web-сайт – это информация, представленная в определенном виде, которая располагается на web-сервере и имеет свое имя (адрес). Для просмотра web-сайтов на компьютере пользователя используются специальные программы, которые называются браузерами. Наиболее распространенными браузерами в настоящее время по оценкам компании Net Application являются Microsoft Internet Explorer и Firefox (рис. 1.1).



Рисунок 1.1 – Обзор состояния рынка браузеров

В зависимости от того, какое имя (адрес) сайта мы зададим в строке «Адрес», браузер будет загружать в свое окно соответствующую информацию.

Web-сайт состоит из связанных между собой web-страниц. Web-страница представляет собой текстовый файл с расширением \*.html (\*.htm), который содержит текстовую информацию и специальные команды – HTML-коды, определяющие в каком виде эта информация будет отображаться в окне браузера. Вся графическая, аудио- и видео-информация непосредственно в web-страницу не входит и представляет собой отдельные файлы с расширениями \*.gif, \*.jpg (графика), \*.mid, \*.mp3 (звук), \*.avi (видео). В HTML-коде страницы содержатся только указания на такие файлы (рис. 1.2).

page1.htm page2.htm page3.htm

Текст +HTML - код

Графика

Графика

Графика

Аудио

Видео

Видео

Видео

Аудио

Аудио

Текст +HTML - код

Текст +HTML - код

Рисунок 1.2 – Web-сайт

Каждая страница web-сайта также имеет свой Internet адрес, который состоит из адреса сайта и имени файла, соответствующего данной странице.

Таким образом, web-сайт – это информационный ресурс, состоящий из связанных между собой гипертекстовых документов (web-страниц), размещенный на web-сервере и имеющий индивидуальный адрес.

Что нужно для того, чтобы посмотреть какой-либо web-сайт? Посмотреть web-сайт может любой человек, имеющий компьютер, подключенный к Internet, web-браузер и адрес сайта.

TML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как <img> и <input>, напрямую вводят контент на страницу. Другие теги, такие как <p>, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

Язык [XHTML](https://ru.wikipedia.org/wiki/XHTML) является более строгим вариантом HTML, он следует синтаксису [XML](https://ru.wikipedia.org/wiki/XML) и является приложением языка XML в области разметки гипертекста.

В HTML можно встроить программный код на языке программирования [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript), для управления поведением и содержанием веб-страниц. Также включение [CSS](https://ru.wikipedia.org/wiki/CSS) в HTML описывает внешний вид и макет страницы.

Следует отметить, что HTML не является языком программирования. И дело здесь даже не в том, что у HTML нет компилятора, а только встроенный в браузер интерпретатор: Basic и JavaScript тоже обходятся одними интерпретаторами, но это не мешает им называться языками программирования. Дело в том, что в HTML отсутствует главный атрибут, присущий любому языку программирования, – команды. На HTML нельзя задать последовательность действий, а можно только описать, как браузер должен вводить на экран тот или иной документ.

HTML – это типичный язык разметки, т.е. с его помощью можно оформлять документы, создавать ссылки, но никак не писать программы. Даже специальные эффекты, которые можно увидеть на web-страницах, создаются не с помощью HTML, а с использованием дополнительных средств – например, встроенных в документ программ на языке JavaScript или PHP. Любой браузер позволяет просмотреть исходный текст HTML-документа.

В HTML-коде страницы содержатся указания на информацию, которую необходимо отобразить (текст и графика) или воспроизвести (звуковое сопровождение) в окне браузера, инструкции о том, как эта информация должна отображаться или воспроизводиться, а также ссылки на другие страницы.

HTML – это язык форматирования, в котором для задания параметров форматирования используется система тегов (дескрипторов). В обычном, некомпьютерном английском языке слово tag означает «признак», «метка», «маркер». Тег состоит из имени, заключенного в угловые скобки, и необязательного набора атрибутов. Как правило, теги бывают открывающийся и закрывающийся, между которыми размещается содержимое тега (текст и другие элементы). Конечный тег имеет то же имя, что и начальный, с той разницей, что перед именем конечного тега ставится косая черта (/).

Существуют теги, которые помещаются в начале и конце страницы (они определяют разделы страницы), теги, которые загружают на страницу рисунок, выполняют программы и влияют на отображение текста. HTML – текст представляет собой текст в формате ASCII. Гипертекстовая страница описывается на языке HTML и обрамляется парными маркерами: <HTML> и </HTML>. Комментарий записывается внутри маркера и не отображается браузером:

<!-- однострочный комментарий --> и <! многострочный комментарий >.

Страница состоит из заголовочной части <HEAD> и тела страницы <BODY>. Вот так выглядит разметка пустой web-страницы:

<HTML>

<HEAD>

<TITLE> Заглавие страницы </TITLE>

</HEAD>

<BODY>

<!-- Тело страницы -->

</BODY>

</HTML>

TML и JavaScript

Чтобы web-страница была интерактивной (могла взаимодействовать с пользователем) и динамичной, необходимо использовать так называемые скрипты или, иначе говоря, сценарии. Скрипт – программа, написанная на специальном языке программирования, которая встраивается в HTML-документ. Например, вы хотите, чтобы при щелчке кнопкой мыши на картинке последняя изменила слой вид или, скажем, произошел переход на другую страницу. Для этого необходим скрипт, который выполнит все, что нужно. То есть сценарии описывают поведение элементов HTML-документа и их взаимодействие с пользователем (например, реакцию на щелчок кнопки мыши, изменение содержания страницы в зависимости от действий пользователя). Вставка сценария в HTML-код осуществляется обычно по следующей схеме:

<HTML>

<HEAD>

...

<!—

<SCRIPT LANGUAGE='название\_языка' >

... (здесь располагается текст сценария)

</SCRIPT>

//-->

</HEAD>

...

<BODY>

...

</BODY>

</HTML>

Здесь использован типовой вариант, при котором сценарий вставляется внутрь тэга заголовка <HEAD>, хотя он может находиться в любом месте HTML-кода. Иногда даже требуется, чтобы скрипт был загружен, после загрузки некоторых элементов страницы,

Атрибут LANGUAGE тэга <SCRIPT> может принимать аргументы JavaScript, JScript, VBScript и VBS. Если атрибут не указан, то подразумевается JavaScript.

Символы <!-- и //--> образуют тэг комментария. Рекомендуется их использовать на тот случай, когда браузер пользователя не сможет интерпретировать сценарий. Обратите внимание, что тэг комментария в данном случае отличается от обычного тэга комментария < ! >, используемого вне сценария для того, чтобы вставить поясняющие тексты, не отображаемые на экране. Сценарий совсем необязательно размещать в том же HTML-документе, где он может понадобиться. Его можно разместить в отдельном текстовом файле, но с расширением js, а в HTML-документе написать такую инструкцию:

<SCRIPT SRC="адрес\_файла\_со\_скриптом"> </SCRIPT>.

Например:

<SCRIPT SRC="my\_script . js"> </SCRIPT>.

Web-сайт состоит из связанных между собой гипертекстовых документов.

Гипертекст – это способ хранения и манипулирования информации, позволяющий устанавливать связи между любыми «информационными единицами». Связь между информационными единицами осуществляется по гиперссылкам. Гиперссылка – это выделенный фрагмент текста, с помощью которого осуществляется переход от одного документа к другому. Обычно гиперссылки выделяют синим цветом и подчеркиванием.

Навигационная схема web-сайта зависит от его структуры и определяет то, как пользователь будет по нему перемещаться и получать доступ к информации, которую представляет сайт.

Простота и удобство навигации является одним из важных факторов, определяющих посещаемость web-сайта. Пользователи должны быстро и легко перейти на любую страницу web-сайта, в том числе на начальную.

Существует несколько видов структурирования информационного материала на web-сайте:

Линейная структура

Материал весь располагается последовательно (рис. 1.3).

Рисунок 1.3 – Линейная структура

Иерархическая структура

Чаще всего структура web-сайта представляет собой иерархию. При этом сначала создают категории высшего уровня, а затем материал в логическом порядке размещают в категории, которые находятся ниже.

Иерархические структуры бывают двух видов: узкая глубокая и широкая неглубокая.

Узкая глубокая иерархия (рис. 1.4) характеризуется тем, что на верхнем уровне она имеет мало категорий.

Для получения нужной информации пользователь вынужден переходить на несколько уровней вниз.

Страница А

Страница N

Рисунок 1.4 – Узкая глубокая иерархия

Из приведенного примера видно, что для перехода к странице N пользователь должен произвести 4 щелчка мышью.

Широкая неглубокая иерархия (рис. 1.5)

Рисунок 1.5 – Широкая неглубокая иерархия

Ни первый, ни второй способ организации информации не является оптимальным. Лучше, когда иерархическая структура состоит из 3 – 4 уровней.

Нелинейная структура (рис. 1.6)

Смешанная структура

Существуют ситуации, когда представить информацию одним из описанных выше методов не представляется возможным. В этом случае применяют несколько схем одновременно. Однако такой подход имеет недостаток – он требует от пользователя концентрации внимания и дополнительных усилий. Чтобы помочь пользователю составляют карту web-сайта (site-map). На данной карте схематически представляют структуру размещения информации на web-сайте.

Замечено, что ориентироваться в дереве разветвлений с глубиной проникновения более чем на три уровня становится очень сложно, ведь пользователь заходя на Ваш сайт, ориентируется в его структуре как если бы, например, он действительно бродил по огромному зданию, с его этажами, коридорами и комнатами. Важно помнить, что навигация должна быть удобной и доступной из любой части каждой Вашей страницы, и должна быть налажена как между отдельными страницами сайта, так и между отдельными частями внутри одной страницы.

При организации web-сайта отдельные его страницы, связанные ссылками, могут физически находиться на совершенно удаленных друг от друга серверах, и все будет нормально функционировать незаметно для пользователя, хотя если один из серверов окажется, вдруг недоступен по какой-либо из причин, то соответственно часть web-сайта посетители не увидят.

**1.2. Средства языка CSS и Java Script**

CSS (/siːɛsɛs/ англ. Cascading Style Sheets «каскадные таблицы стилей») — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана) или при выводе устройствами, использующими шрифт Брайля.

Java – объектно-ориентированный язык программирования общего назначения, который был разработан Sun Microsystems. Разработка ведётся сообществом, организованным через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL. На текущий момент права на торговую марку принадлежат корпорации Oracle.

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор. Достоинством подобного способа выполнения программ является полная независимость байт-кода от ОС и оборудования, позволяющая выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другая важная особенность Java – гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят снижение производительности.

Основные возможности Java:

* Автоматическое управление памятью.
* Расширенные возможности обработки исключительных ситуаций.
* Богатый набор средств фильтрации ввода-вывода.
* Набор стандартных коллекций: массив, список, стек и т.п.
* Наличие простых средств создания сетевых приложений.
* Наличие классов, позволяющих выполнять HTTP-запросы и обрабатывать ответы.
* Встроенные в язык средства создания многопоточных приложений, которые потом были портированы на многие языки.
* Унифицированный доступ к базам данных.
* Поддержка обобщений.
* Поддержка лямбда-вычислений.
* Поддержка замыканий.
* Поддержка некоторых возможностей функционального программирования.

CSS используется создателями [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) для задания [цветов](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82), [шрифтов](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82), стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью [HTML](https://ru.wikipedia.org/wiki/HTML) или других [языков разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8)) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью [формального языка](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или [методах](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4) вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана) или при выводе устройствами, использующими [шрифт Брайля](https://ru.wikipedia.org/wiki/%D0%A8%D1%80%D0%B8%D1%84%D1%82_%D0%91%D1%80%D0%B0%D0%B9%D0%BB%D1%8F).

Способы подключения CSS к документу

Правила CSS могут располагаться как в самом [веб-документе](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82), внешний вид которого они описывают, так и во внешних [файлах](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB), имеющих [расширение](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D1%82_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0) .css. Формат CSS — это [текстовый файл](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%B0%D0%B9%D0%BB), в котором содержится перечень правил CSS и комментариев к ним.  
Стили CSS могут быть подключены или внедрены в описываемый ими [веб-документ](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82) четырьмя способами:

* когда описание стилей находится в отдельном файле, оно может быть подключено к документу посредством элемента <link>, включённого в элемент <head>:

<!DOCTYPE html>

<html>

<head>

.....

<link rel="stylesheet" type="text/css" href="style.css">

</head>

<body>

.....

</body>

</html>

* когда файл стилей размещается отдельно от родительского документа, он может быть подключён к документу инструкцией @import в элементе<style>:

<!DOCTYPE html>

<html>

<head>

.....

<style media="all">

@import url(style.css);

</style>

</head>

</html>

* когда стили описаны внутри документа, они могут быть включены в элемент <style>, который, включается в элемент <head>:

<!DOCTYPE html>

<html>

<head>

.....

<style>

body {

color: red;

}

</style>

</head>

<body>

.....

</body>

</html>

* когда стили описаны в теле документа, они могут располагаться в атрибутах отдельного элемента

<!DOCTYPE>

<html>

<head>

.....

</head>

<body>

<p style="font-size: 20px; color: green; font-family: arial, helvetica, sans-serif">

.....

</p>

</body>

</html>

В первых двух случаях к документу применены внешние стили, а во вторых — внутренние стили.

Для добавления CSS к [XML-документу](https://ru.wikipedia.org/wiki/XML) последний должен содержать специальную ссылку на файл стилей:

<?xml-stylesheet type="text/css" href="style.css"?>

Правила построения CSS

В первых трёх случаях подключения стилей CSS к документу (см. выше) каждое правило CSS из файла имеет две основные части — селектор и блок объявлений. Селектор, расположенный в левой части правила до знака «{», определяет, на какие части документа (возможно, специально обозначенные) распространяется правило. Блок объявлений располагается в правой части правила. Он помещается в фигурные скобки, и, в свою очередь, состоит из одного или более объявлений, разделённых знаком «;». Каждое объявление представляет собой сочетание свойства CSS и значения, разделённых знаком «:». Селекторы могут группироваться в одной строке через запятую. В таком случае свойство применяется к каждому из них.

селектор, селектор {

свойство: значение;

свойство: значение;

свойство: значение;

}

В четвёртом случае подключения CSS к документу (см. список) правило CSS, являющееся значением атрибута style элемента, к которому он применяется, представляет собой перечень объявлений («свойство CSS : значение»), разделённых знаком «;».

Виды селекторов

Универсальный селектор

\* {

margin: 0;

padding: 0;

}

Селектор тегов

p {

font-family: arial, helvetica, sans-serif;

}

Селектор классов

.note {

color: red;

background-color: yellow;

font-weight: bold;

}

Селектор идентификаторов

#paragraph1 {

margin: 0;

}

Селектор атрибутов

a[href="http://www.somesite.com"] {

font-weight: bold;

}

Селектор потомков (контекстный селектор)

div#paragraph1 p.note {

color: blue;

}

Селектор дочерних элементов

p.note > b {

color:green;

}

.div {

border: 1px solid red;

padding-left: 20px;

}

.title {

font-size: 20px;

background-color: red;

}

Селектор элементов одного уровня

h1 + p {

font-size: 24px;

}

Селектор псевдоклассов

a:active {

color: blue;

}

Селектор псевдоэлементов

p::first-letter {

font-size: 32px;

}

Классы и идентификаторы элементов

Класс или идентификатор может быть присвоен какому-нибудь элементу HTML посредством атрибутов class или id этого элемента:

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>

Селекторы классов и идентификаторов

</title>

<style>

p.big {

font-family: arial, helvetica, sans-serif;

color: red;

}

div#first {

background-color: silver;

}

</style>

</head>

<body>

<div id="first">

<!-- Это div c серым фоном, стилизованный по id -->

</div>

<p class="big">

<!-- Это текст красного цвета -->

</p>

</body>

</html>

Основное различие между классами элементов и идентификаторами элементов в том, что идентификатор предназначен для одного элемента, тогда как класс обычно присваивают сразу нескольким. Тем не менее, современные браузеры, как правило, корректно отображают множественные элементы с одинаковым идентификатором. Также различие в том, что могут существовать множественные классы (когда класс элемента состоит из нескольких слов, разделённых пробелами). Для идентификаторов такое невозможно.

Важно отметить следующее отличие идентификатора от класса: идентификаторы широко используются в [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript) для нахождения уникального элемента в документе.

Имена классов и идентификаторов, в отличие от названий тегов и их атрибутов, чувствительны к регистру ввода букв.

Свойства классов и идентификаторов задаются с помощью соответствующих селекторов. Причём может быть задано как свойство класса в целом (в таком случае селектор начинается с „.“, например, „.big“) или свойство идентификатора самого по себе (в таком случае селектор начинается с „#“, например, „#first“), так и свойство какого-нибудь элемента этого класса или с этим идентификатором.

В CSS, помимо классов, задаваемых автором страницы, существует также ограниченный набор так называемых [псевдоклассов,](https://webref.ru/css" \l "pseudo-class) описывающих вид гиперссылок с определённым состоянием в документе, вид элемента, на котором находится фокус ввода, а также вид элементов, являющихся первыми дочерними элементами других элементов. Также в CSS существует четыре так называемых [псевдоэлемента](https://webref.ru/css" \l "pseudo-elements): первая буква, первая строка, применение специальных стилей до и после элемента.

**ВЫВОД**

В данной главе было изучено понятие сайта. Сайт – это одна или несколько логически связанных между собой веб-страниц; также место расположения контента сервера.

Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователями как единое целое. Классификация сайтов достаточно запутанная, потому что для нее применяется большое количество переменных: дизайн, тематика, способ создания, внутренняя структура, взаимодействие с аудиторией, цель, функциональные возможности и др. Разные виды веб-сайтов по-разному воспринимаются пользователями и по-разному продвигаются в поисковых системах. Веб-сайт может быть любым и классифицироваться, как угодно. Главная задача каждого веб-сайта – удовлетворять потребности пользователя, а все остальное второстепенно. Так же были рассмотрены инструменты и сервисы, которые помогают разработать и протестировать сайт.

**II. ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1. План разработки веб-сайта с промоакциями средствами языка HTML, CSS и Java Script**

**2.2. Анализ сайтов схожей тематики**

**ВЫВОД**

**III ТЕХНИКА БЕЗОПАСНОСТИ**

**ВЫВОД**

**ЗАКЛЮЧЕНИЕ**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Башмачников А.И., Башмачников И.А. Разработка компьютерных учебников и обучающих систем. – М.: Филинъ, 2019. 616 с.  
2. Горнаков С.Г. Осваиваем популярные системы управления сайтом (CMS). СПб.: Питер, 2019. — 248 с.  
3. Горнаков, С. Г. Осваиваем популярные системы управления сайтом (CMS): учебник / С. Г. Горнаков. – Москва: ДМК Пресс, 2019. –336 с.  
4. Денис Колисниченко. Движок для вашего сайта. CMS Joomla, Slaed, PHP-Nuke. — Петербург: БХВ, 2020. — 352 с.  
5. ДУильямс Б., Дэмстра Д., Стэрн Х. WordPress для профессионалов. Разработка и дизайн сайтов. – М.: Филинъ, 2019. – 252 с.  
6. Каймин, В.А. Информатика: учебник / В.А. Каймин. – Москва:ИНФРА, 2019. – 232 с.  
7. Келли Г, Котлер Э, Веб-редизайн, 2-е издание. — СПб.:»СимволПлюс», 2020.— 416 с.  
8. Лукьянова, Г.Н. Инструменты онлайн бизнеса: книга / Г.Н.Лукьянова. −Москва: Литкон, 2019. − 245 с.  
9. Норт, Б. М. Joomla!: пактическое руководство / Б. М. Норт. - СПб: Символ, 2020. - 110 с.  
10. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript / Р. Никсон.- Санкт-Петербург: Питер, 2020. – 314 с.  
11. Олищук А, Разработка Web-приложений на PHP 5. Профессиональная работа.— М.: «Вильямс», 2020. — 352 с.  
12. Олифер, В.Г. Компьютерные сети. Принципы, технологии, протоколы: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – Изд. 4-е − Санкт−Петербург: Питер, 2019. – 944 с.  
13. Профессиональная разработка сайтов на Drupal 7 / Б. Мелансон [и др.; пер. с англ. И. Размайкина]. - Москва: Питер, 2020. – 420 с.  
14. Попов, В.К. Практикум по Интернет-технологиям/ В.К. Попов. – СПб: Питер, 2019. – 162 с.  
15. Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование / В.Ю. Пирогов. - СПб: БХВ-Петербург, 2019. – 87 с.  
16. Ромашов В. CMS Drupal. Система управления содержимым сайта. Санкт-Петербург: Питер, 2019. - 56 с.  
17. Стивен Шафер. HTML, XHTML и CSS. Библия пользователя, 5-е издание =HTML, XHTML, and CSS Bible, 5th Edition. – М.: «Диалектика», 2019. — 656 с.  
18. Стейнмец, У. PHP. 75 готовых решений для Веб-сайта / У. Стейнмец, Брайан Вард. - Санкт-Петербург: Наука и Техника, 2019. – 99 с.  
19. Фримен Эрик, Фримен Элизабет. Изучаем HTML, XHTML и CSS = Head First HTML with CSS & XHTML. –1-е изд. – М.: «Питер», 2019. – 656 с.  
20. Фрэйн, Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрэйн. - Санкт-Петербург: Питер, 2019. - 56 с.  
21. Халворсон, К. Контентная стратегия управления сайтом / К.Халворсон, М. Рэч. - Санкт-Петербург: Питер, 2020. – 326 с.  
22. Хортон У, Хортон К. Электронное обучение: инструменты и технологии. М.: ИД КУДИЦ-ОБРАЗ, 2020. 640 с.