ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ МОСКОВСКОЙ ОБЛАСТИ  
«ПРОФЕССИОНАЛЬНЫЙ КОЛЛЕДЖ «МОСКОВИЯ»

09.02.07 Информационные системы и

программирование

К защите допущен(а)

Зам. директора по УР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Галицына А.А.

(подпись Ф.И.О.)

**ДИПЛОМНАЯ РАБОТА**

Тема: **Проектирование и разработка сайта с промоакциями средствами языка HTML.CSS и Java Script**

Выпускник \_Глушко С.С.\_ Группа № \_6ИС-19\_

(Ф.И.О.)

Работа выполнена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись выпускника)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Степанова А.В.\_\_\_ « \_» 20 23 г.

(подпись) (Ф.И.О.)

Работа защищена с оценкой

Протокол № от « » 20\_\_\_г.

**Государственное автономное профессиональное**

**образовательное учреждение московской области**

**«профессиональный КОЛЛЕДЖ «московия»**

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**для выполнения дипломной работы**

**Специальность (код и наименование):** 09.02.07 Информационные системы и программирование

**Группа:** 6ИС-19

**ФИО выпускника:** Глушко Сергей Сергеевич

**Тема Дипломной работы:** Проектирование и разработка сайта с промоакциями средствами языка HTML.CSS и Java Script.

**Дата выдачи задания:** «06» апреля 2023г.

**Работа должна быть сдана не позднее**: «01» июня 2023г.

**Перечень вопросов, подлежащих разработке в дипломной работе:**

**Введение:** при написании дипломной работы отобразить в введении актуальность выбранной темы, цели, задачи.

**Содержание пояснительной записи (перечень подлежащих разработке вопросов):**

1. Теоретические часть.

2. Практическая часть.

3. Экономическая часть.

4. Охрана труда.

**Заключение**: на основании календарного плана-задания обучающегося и индивидуального задания по выполнению дипломной работы, студент должен сделать заключение и внести свои предложения по данной работе.

**Приложения:** при наличии графических рисунков, диаграмм, таблиц большого объёма выносить в приложение.

Руководитель ДР: \_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Степанова А.В../

(подпись) (расшифровка подписи)

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Глушко С.С./

(подпись) (расшифровка подписи)

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 г.

**Государственное автономное профессиональное**

**образовательное учреждение московской области**

**«профессиональный КОЛЛЕДЖ «московия»**

**Календарный план-задание обучающегося**

**по выполнению дипломной работы**

**ИСПОЛНИТЕЛЬ РУКОВОДИТЕЛЬ ДР**

Фамилия Глушко Фамилия Степанова

Имя Сергей Имя Алевтина

Отчество Сергеевич Отчество Васильевна

Группа 6ИС-19 Должность Преподаватель

**Тема работы:** Проектирование и разработка сайта с промоакциями средствами языка HTML.CSS и Java Script.

**Содержание ДР (**перечень подлежащих разработке вопросов**) и сроки выполнения**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование вида работ** | **Сроки**  **исполнения** | **Отметка о выполнении, замечания руководителя** |
| 1. | Подбор, изучение и обработка литературы по проблематике дипломной работы. | «06» апреля 2023 г.  «09»апреля2023 г. | выполнено |
| 2. | Постановка задачи, составление плана и утверждение его руководителем дипломной работы. | «09» апреля 2023г.  «14»апреля 2023 г. | выполнено |
| 3. | Написание и представление руководителю первого раздела (обзор и постановка задачи) | «14» апреля 2023 г.  «22»апреля 2023 г. | выполнено |
| 4. | Разработка и представление руководителю второго раздела (результаты проведенного исследования) | «22» апреля 2023 г.  «30»апреля 2023 г. | выполнено |
| 5. | Анализ результатов, формирование выводов и разработка предложений. | «30» апреля 2023 г.  «10»мая 2023 г. | выполнено |
| 6. | Доработка дипломной работы в соответствии с замечаниями руководителя и представление ее на предзащиту. | «10» мая 2023 г.  «17» мая 2023 г. | выполнено |
| 7. | Подготовка тезисов доклада для защиты дипломной работы и обсуждение их с руководителем. | «17» мая 2023 г.  «24»мая 2023 г. | выполнено |
| 8. | Ознакомление выпускника с отзывом и рецензией на дипломную работу. | «24» мая 2023 г.  «27»мая 2023 г. | выполнено |
| 9. | Подготовка к защите с учетом замечаний, сделанных в отзыве и рецензии, подготовка презентации | «27» мая 2023 г.  «01»июня 2023 г. | выполнено |
| 10. | Защита дипломной работы | « 26» июня 2023 г. |  |

**Срок сдачи исполнителем законченной работы руководителю за 15 дней до защиты в соответствии с графиком ГИА**

|  |  |
| --- | --- |
| Подпись исполнителя\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Глушков С.С. / | |
| *(подпись) (Фамилия И.О.)* | |
| Заключение руководителя:  «Работа рекомендована к защите» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Степанова А.В./ |
| *(подпись) (Фамилия И.О.)* | |

Оглавление

[**ВВЕДЕНИЕ** 5](#_Toc138065597)

[**ГЛАВА I ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 7](#_Toc138065598)

[**1.1. Язык HTML** 7](#_Toc138065599)

[**1.2. Средства языка CSS и Java Script** 21](#_Toc138065600)

[**ГЛАВА II ПРАКТИЧЕСКАЯ ЧАСТЬ** 27](#_Toc138065601)

[2.1. План разработки веб-сайта с промоакциями средствами языка HTML, CSS и Java Script 27](#_Toc138065602)

[I Этап. Постановка цели и задачи сайта. 35](#_Toc138065603)

[II этап. Реализация сайта 36](#_Toc138065604)

[III этап. Поддержка сайта**.** 37](#_Toc138065605)

[IV этап. Безопасность сайта. 37](#_Toc138065606)

[2.2. Анализ сайтов схожей тематики 41](#_Toc138065607)

[**III Экономическая часть** 57](#_Toc138065608)

[**IV ТЕХНИКА БЕЗОПАСНОСТИ** 62](#_Toc138065609)

[Требования безопасности во время эксплуатации ЭВМ 62](#_Toc138065610)

[Требования безопасности во время работы 62](#_Toc138065611)

[Требования безопасности в аварийных ситуациях 64](#_Toc138065612)

[Требования безопасности после окончания работы 65](#_Toc138065613)

[**ЗАКЛЮЧЕНИЕ** 66](#_Toc138065614)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 67](#_Toc138065615)

[Приложение 69](#_Toc138065616)

# **ВВЕДЕНИЕ**

Интернет в современном мире стал полноценной социальной средой, в которой сотни людей встречаются, общаются, покупают и продаются товары, делятся новостями, мнениями, получают необходимую им информацию. В настоящее время невозможно представить современный бизнес без использования Интернет-технологий. Каждая уважающая себя фирма имеет представительство в Сети. Объясняется это тем, что сайт открывает огромный диапазон возможностей.

Web-сайт - это не только красиво оформленная информация о компании, но, прежде всего мощное и удобное средство работы с клиентами. Cайт - это лицо компании. В большинстве случаев потенциальные клиенты стремятся ознакомиться с сайтом, прежде чем иметь дело непосредственно с организацией.

Сегодня даже небольшие компании могут привлечь больше потенциальных клиентов, представив свой бизнес в Интернете. Поэтому создание динамичного и интерактивного веб-сайта имеет большое значение для онлайн-предприятий. Хорошо разработанный веб-сайт может не только значительно увеличить прибыль и число потенциальных клиентов, но и завоевать ценное доверие клиентов и партнеров.

В связи с актуальностью проблематики, связанной с разработкой и созданием web-сайтов, и была выбрана данная тема выпускной квалификационной работы.

Исследование было посвящено современному онлайн-присутствию компании как средству продвижения и поддержания положительного корпоративного имиджа.

**Целью выпускной квалификационной работы является** разработка проекта веб-сайта с промоакциями средствами языка HTML, CSS и Java Script.

**Для выполнения выпускной квалификационной работы были поставлены следующие задачи:**

1. изучить теоретические аспекты создания веб-сайта;

2. проанализировать средства для разработки веб-сайта;

3. разработать сайт с промоакциями средствами языка HTML, CSS и Java Script;

4. протестировать разработанный сайт.

**Объектом исследования** **выпускной квалификационной** работы является веб-сайт с промоакциями.

**Предмет исследования** **выпускной квалификационной работы** является процесс создания веб-сайта с промоакциями средствами языка HTML, CSS и Java Script.

**Практическая значимость** **выпускной квалификационной** работы заключается в разработке удобного для пользователей сайта с промоакциями средствами языка HTML, CSS и Java Script, позволяющей просматривать последние новости компании, а так же обращаться по различным вопросам.

**Выпускная квалификационная работа** состоит из следующих частей: введение, четырёх глав, заключения, списка литературы.

Во введении сформулирована актуальность темы, а также обозначены основные цели и задачи, а также содержится вводная информация по настоящему исследованию.

Основная часть посвящена анализу поставленной проблемы.

В заключении обобщены выводы по работе и исследования, будут сформулированы основные результаты исследования, a также кратко излагаются полученные автором выводы.

Список используемых источников содержит перечень источников, использованных при подготовке данной работы. Работа иллюстрирована таблицами, рисунками и диаграммами.

# **ГЛАВА 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

Графика

Графика

Графика

Аудио

Видео

Видео

Видео

Аудио

Аудио

Текст +HTML - код

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.

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:

Зная тип сайта и другую информацию, можно начинать накладывать на него структуру. Существуют две структурные стороны каждого Web-сайта: логическая и физическая структура. Логическая структура описывает документы, которые связаны с другими документами. Она определяет связи между ними. Однако логическое расположение документов в пределах сайта может не иметь отношения к реальному физическому их размещению. Физическая структура описывает, где документ находится в действительности, отражая, например, путь к каталогу документа на Web-сервере или его расположение в базе данных.

Существуют четыре основные логические организационные формы, используемые Web-сайтами: линейная, решетка, иерархия и паутина. Вариации на тему некоторых из схем также являются обычными, как и их сочетания, лежащие в основе более крупных сайтов. Выбор правильной логической организации сайта является важным шагом на пути создания удобного для использования сайта. К примеру, для сайта, ориентированного на online-продажи, более практичной будет линейная форма, при которой слайд 2 следует за слайдом 1. В некотором смысле пользователь почти вынужден просматривать содержимое сайта в том порядке, в каком этого хочет дизайнер. Если представление материала было организовано в другой манере, такой как дерево, то это может подстрекать пользователей осуществлять доступ к слайдам не по порядку и возможно уменьшит влияние, подталкивающее посетителей сайта к осуществлению покупки. В то же время последовательный доступ мало подходит для представления другой информации, такой, к примеру, как вопросы технической поддержки, поэтому принуждая пользователя пробираться через страницы с бесполезной информацией, можно вызвать его крайнее раздражение. Цель выбора наиболее подходящей для содержимого организационной формы состоит в том, чтобы сделать сложное содержимое более понятным.

Линейная форма - это самая простая структура сайта. Web-страницы идут одна за другой, и пользователь должен просматривать их как слайд-шоу. Она является наиболее популярной из всех структур сайтов по причине того, что традиционные печатные информационные средства следуют этому стилю организации. Представление информации в линейной форме часто бывает удобным при обсуждении пошаговых процессов, но существуют случаи, когда может требоваться дополнительная информация. Для обеспечения большей гибкости линейные формы могут модифицироваться, но при слишком большом расширении они превращаются в решетку, иерархию или паутину. Строго линейная организация способствует упорядоченному продвижению по основной части информации.

*Рисунок 1.1 - Строго линейная организация*

В Web такая форма может хорошо подходить для презентации, напоминающей "слайд-шоу" и дающей возможность новым посетителям ознакомиться с обзором предлагаемой компанией продукции и самой компанией. Используя контролируемую последовательную организацию, подобную линейной форме, дизайнер может быть уверен, что пользователь получает информацию в намеченном порядке.

Линейная организация с альтернативами имитирует интерактивность, предоставляя два или более вариантов перехода со страницы, которые, в конечном счете, оканчиваются указанием пользователю вернуться на другую страницу в последовательности.

*Рисунок 1.2 -Линейная форма с альтернативами*

Для этой формы организации существует множество применений. Например, сайт-вопросник, который на каждой странице задает пользователю вопрос, требующий ответа "Да" или "Нет" и продвигает пользователя на следующую страницу, основываясь на данном ответе. Хотя для пользователя это может выглядеть, как работа некоторого внутреннего механизма, в действительности существуют два заранее заданных пути, и у пользователя просто создается иллюзия интерактивности. Вне зависимости от наличия выбора линейная структура с альтернативами сохраняет общее линейное направление через набор документов. К сожалению, наличие нескольких путей делает предварительную загрузку страниц для этой формы сайта более сложной.

Линейная структура со свободой выбора хорошо работает, когда необходимо сохранить общее направление, но при этом нужно добавить легкие вариации, такие как пропуск определенных страниц. Такой тип гипертекстовой организации может быть полезен для online-обзоров, где некоторые пользователи имеют возможность пропустить определенные неактуальные вопросы. Принимая во внимание то, что такая структура в основном только позволяет перепрыгивать вперед, ее часто называют линейной структурой с переходами вперед.

*Рисунок 1.3 - Линейная форма со свободой выбора*

Линейная организация с боковыми ответвлениями позволяет контролировать отклонения от основного направления. При этом пользователь имеет возможность немного отклониться в сторону, однако структура все же принуждает его вернуться к основному пути, сохраняя первоначальное направление движения.

*Рисунок 1.4 - Линейная форма с боковыми ответвлениями*

Боковое ответвление может заканчиваться тупиком или возвратом обратно к основному направлению. Боковое ответвление в линейной последовательности подобно боковой врезке в журнальной статье. Оно позволяет не уводить пользователя далеко от основной темы, и при этом небольшой объем дополнительно представленной информации расширяет кругозор.

Решетка - это двунаправленная линейная структура, в которой присутствуют как горизонтальные, так и вертикальные связи между элементами. Так как решетка имеет пространственную организацию, она хорошо подходит для совокупности родственных элементов, однако структура строгой решетки является редкой для Web. Правильно разработанная решетка имеет горизонтальные и вертикальные ориентиры, поэтому пользователь не чувствует себя заблудившимся внутри сайта. Например, предметы в каталоге одежды могут быть собраны в категории, такие как рубашки, брюки и куртки. Другой способ организации информации может состоять в разбиении по ценовым категориям. Структура в стиле решетки позволит пользователю легко просматривать как цены, так и определенную линию одежды.

*Рисунок 1.5 - Решетка*

Решетчатая структура очень регулярна, и пользователь может легко ориентироваться, однако существует не так много видов однородной информации, достаточно хорошо приспособленной для такого рода организации. Единственное заметное исключение - это каталог изделий.

**Иерархия**

Наиболее часто встречающейся в Web гипертекстовой структурой является форма дерева или иерархии. Хотя иерархия может не обеспечивать пространственной структуры, присущей решетке, или предсказуемости и управляемости линейной структуры, она необычайно важна, т. к. может быть модифицирована для того, чтобы скрыть или показать столько информации, сколько необходимо. Иерархии начинаются с корневой страницы, которая часто является домашней страницей сайта или раздела. Она часто сильно отличается по внешнему виду от других страниц сайта. Для перехода с домашней страницы представлены различные варианты выбора. По мере продвижения пользователя вглубь сайта варианты выбора становятся все более специфичными, пока не приведут к пункту назначения или конечной странице в дереве исследуемого сайта. По этой причине существует тенденция описывать деревья с позиций их глубины и ширины. Узкое дерево предлагает только несколько вариантов выбора, но может требовать большого количества щелчков кнопкой мыши, чтобы достигнуть конечного места назначения.

*Рисунок 1.6 - Узкая иерархия*

Широкое дерево или широкая иерархия основывается на большом количестве вариантов выбора. Ее основной недостаток заключается в том, что она может предлагать слишком много вариантов в виде страниц, имеющих большое количество выходящих с них ссылок.

*Рисунок 1.7 - Широкая иерархия*

Несмотря на то, что пользователь должен только один или два раза щелкнуть кнопкой мыши, чтобы добраться до необходимого ему содержимого, время, потраченное на изучение всех первичных вариантов выбора, может быть чрезмерным.

Структура стандартного дерева используется очень редко. В стандартном дереве нет перекрестных ссылок, и для достижения других частей дерева часто требуется вернуться назад. Хотя в Web для возврата можно задействовать кнопку "Назад" браузера, на страницы часто добавляются ссылки, чтобы те пользователи, кто попал на страницу не через основной путь, могли перемещаться по сайту.

Во многих случаях страницы связаны перекрестными ссылками при помощи панели навигации или явных обратных ссылок, помогающих пользователям быстро перемещаться по структуре сайта.

**А**

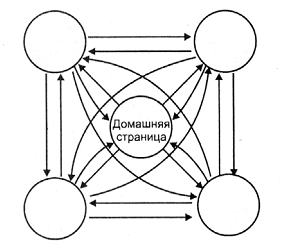
**В**

**С**

*Рисунок 1.8 Запутанные деревья*

Распространенным приемом является создание для сайта панели навигации, которая содержит основные разделы сайта, такие как "главная", "о компании", "продукция", "новости" и "контакт". С такой панелью навигации намного проще перепрыгнуть из раздела в раздел без необходимости осуществлять для этого переходы назад.

Web-сайт, каждая страница которого связана ссылкой с каждой другой страницей этого Web-сайта, может рассматриваться в качестве структуры, называемой полное связывание.



*Рисунок 1.9 - Полное связывание*

При полном связывании количество ссылок равно числу страниц, помноженному на число страниц минус один. Это означает, что для сайта из пяти страниц количество ссылок будет равно 20. Для сайта из 20 страниц количество ссылок составит 90. Для сайта из 100 страниц - число ссылок составит 9900 (100x99) и для 1000-страничного сайта их число будет в районе одного миллиона! В действительности с точки зрения практичности полное связывание не является хорошим выбором. Если в качестве максимального допустимого количества ссылок на странице принять число девять, то сайт в стиле полного связывания может состоять из десяти страниц. В действительности, большинство сайтов склонны использовать частичное связывание с перекрестными ссылками только на наиболее важные страницы.

В некоторых случаях необходимо расширить иерархию, чтобы позволить вариантам выбора указывать на вершину дерева. Такая структура называется смешанной формой или смешанной иерархией, и дерево является доминирующей формой структуры. Смешанная форма, вероятно, является наиболее часто встречающимся в Web видом организации сайта.

Внутри смешанной формы могут содержаться линейные участки, пропуски и даже решетки. Один часто встречающийся тип смешанной структуры состоит в использовании линейной структуры для входа на сайт и дерева с момента достижения настоящей домашней страницы. Для этого используются выставленные на показ страницы с информацией о сайте, которые ведут к центральной странице, откуда пользователь может начать просмотр сайта.

*Рисунок 1.10 - Смешанная иерархия*

**Стиль паутины**

Когда в структуру набора документов внесено слишком много перекрестных ссылок, переходов вперед и других дополнений, для пользователя она становится непонятной. Если совокупность документов выглядит так, будто не имеет различимой структуры, то она называется паутиной.

*Рисунок 1.11 - Паутина*

Паутина может быть сложна для использования, т. к. она лишена отчетливой пространственной ориентации. Несмотря на то, что в случае правильного выбора можно получить быстрый доступ к информации, на сайте с такой неочевидной структурой сложно ориентироваться. Если структура сайта непонятна или неочевидна для пользователя, он может прибегнуть к навигации, основанной на использовании домашней страницы, и при решении новой задачи всегда будет возвращаться на верхний уровень структуры.

Юзабилити сайта является всеобъемлющим термином, определяющим комплекс мер, результатом которого является создание удобного и понятного сайта. Термин "юзабилити" можно рассматривать как "конечную суммарную степень удобства, меру интеллектуального усилия, необходимого для получения полезных качеств вещи, и скорость достижения положительного результата при управлении ею".

Существуют типовые ошибки, которые обычно приводят к тому, что пользователи сайта не могут найти интересующую их информацию. К таким ошибкам относятся плохая визуализация основных разделов сайта, сложная система навигации, отсутствие единого стиля оформления сайта, несоответствие содержимого ожиданиям посетителей и т.д. Все эти ошибки юзабилити обусловлены непониманием потребностей пользователей сайта, которые являются важнейшим элементом при проектировании сайта.

Особенности восприятия информации в Интернет. Нужно иметь в виду, что, во-первых, пользователи не читают тексты на сайтах так же, как книгу или газету. Они бегло просматривают страницу, пытаясь "зацепиться взглядом" за интересные им слова или выражения. Во-вторых, пользователи перемещаются по сайту не после того, как оценят и взвесят все возможности и выберут лучшее, а выбирают наиболее подходящий вариант из первых попавшихся.

Для юзабилити сайта характерен принцип "Чем проще, тем лучше"; наличие стандартных приемов в интерфейсе сайта также является преимуществом. Ведь если пользователь когда-то уже видел что-то подобное (в навигации, элементах страницы или чем-нибудь другом), он будет знать, как этим пользоваться.

Особенности поведения пользователей. В принципе, поведение пользователей на веб-сайте не особо отличаются от их поведения в, скажем, магазине. Пользователи мельком окидывают взглядом каждую новую страницу, просматривают отрывки текста и кликают на первую зацепившую их внимание ссылку, или хотя бы отдаленно похожую на ту, которую они искали. На самом деле, на большую часть страницы они даже не смотрят. Многие пользователи ищут что-нибудь интересное (или полезное) и "кликабельное"; как только на глаза попадается подходящий "кандидат", пользователь быстро кликает и переходит по ссылке. Если результат не оправдал его ожиданий, пользователь кликает "назад" и продолжает поиск.

Пользователи высоко ценят качество и правдивость информации. Если страница содержит высококачественный материал, пользователи готовы идти на компромисс с наличием рекламы на странице и её дизайном. Это одна из причин, почему сайты с не совсем хорошим дизайном, но с качественным содержанием получают большой трафик. Содержание намного важнее, чем обрамляющий его дизайн. Анализируя веб-страницу, пользователи ищут фиксированные точки, "якоря", которые будут направлять их по содержимому на странице. Очень простой принцип: если веб-сайт не может оправдать ожиданий пользователей, то дизайн не выполняет свою функцию, а компания теряет деньги. Чем навигация менее интуитивна, тем сильнее желание пользователя покинуть веб-сайт и найти альтернативу.

Пользователи не ищут самый быстрый способ для поиска желаемой информации. Они также не сканируют веб-сайты последовательно, переходя от одного раздела к другому. На самом деле они выбирают первый наиболее подходящий, по их мнению, вариант. Как только они видят ссылку, которая может привести их к искомой цели, велика вероятность, что по ссылке моментально кликнут.

Пользователям важно контролировать браузер и полагаться на правильную передачу данных через сайт. Например, им не нужно неожиданно открывающихся всплывающих окон и они хотят иметь возможность вернутся на предыдущую страницу, нажав кнопку "назад". Отсюда, необходимо следовать одной из важнейших практик - никогда не открывать ссылку в новом окне браузера.

В данной главе работы была рассмотрена роль Интернет в современном обществе, особое внимание было уделено классификации web-сайтов, описаны преимущества создания сайта, роль web-сайта в развитии бизнеса.

А также детально рассмотрены виды структурной организации, проектирование навигационной системы сайтов. Рассматривались и обсуждались основные требования юзабилити.

# **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.

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.

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.

JavaScript **-** этомультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком корпорации Oracle в США.

В 1992 году компания Nombas (впоследствии приобретённая Openwave) начала разработку встраиваемого скриптового языка Cmm (Си-минус-минус), который, по замыслу разработчиков, должен был стать достаточно мощным, чтобы заменить макросы, сохраняя при этом схожесть с Си, чтобы разработчикам не составляло труда изучить его. Главным отличием от Си была работа с памятью. В новом языке всё управление памятью осуществлялось автоматически: не было необходимости создавать буфера, объявлять переменные, осуществлять преобразование типов. В остальном языки сильно походили друг на друга: в частности, Cmm поддерживал стандартные функции и операторы Си.

Cmm был переименован в ScriptEase, поскольку исходное название звучало слишком негативно, а упоминание в нём Си «отпугивало» людей. На основе этого языка был создан проприетарный продукт CEnvi. В конце ноября 1995 года Nombas разработала версию CEnvi, внедряемую в веб-страницы. Страницы, которые можно было изменять с помощью скриптового языка, получили название Espresso Pages — они демонстрировали использование скриптового языка для создания игры, проверки пользовательского ввода в формы и создания анимации. Espresso Pages позиционировались как демоверсия, призванная помочь представить, что случится, если в браузер будет внедрён язык Cmm. Работали они только в 16-битовом Netscape Navigator под управлением Windows.

Самая первая реализация JavaScript была создана Бренданом Эйхом (англ. Brendan Eich) в компании Netscape, и с тех пор обновляется, чтобы соответствовать ECMA-262 Edition 5 и более поздним версиям. Этот движок называется SpiderMonkey и реализован на языке C/C++. Движок Rhino создан Норрисом Бойдом (англ. Norris Boyd) и реализован на языке Java. Как и SpiderMonkey, Rhino соответствует ECMA-262 Edition 5.  
Структурно JavaScript можно представить в виде объединения трёх чётко различимых друг от друга частей:

ядро (ECMAScript),

объектная модель браузера (Browser Object Model или BOM),  
объектная модель документа (Document Object Model или DOM).  
Если рассматривать JavaScript в отличных от браузера окружениях, то объектная модель браузера и объектная модель документа могут не поддерживаться.

Объектную модель документа иногда рассматривают как отдельную от JavaScript сущность, что согласуется с определением DOM как независимого от языка интерфейса документа. В противоположность этому ряд авторов находит BOM и DOM тесно взаимосвязанными с  
Ядро.

Основная статья: ECMAScript

ECMAScript не является браузерным языком и в нём не определяются методы ввода и вывода информации. Это, скорее, основа для построения скриптовых языков. Спецификация ECMAScript описывает типы данных, инструкции, ключевые и зарезервированные слова, операторы, объекты, регулярные выражения, не ограничивая авторов производных языков в расширении их новыми составляющими.

Объектная модель браузера

Объектная модель браузера — браузер-специфичная часть языка, являющаяся прослойкой между ядром и объектной моделью документа.

Основное предназначение объектной модели браузера — управление окнами браузера и обеспечение их взаимодействия. Каждое из окон браузера представляется объектом window, центральным объектом DOM. Объектная модель браузера на данный момент не стандартизирована, однако спецификация находится в разработке WHATWG (Спецификация 4) и W3C.

Помимо управления окнами, в рамках объектной модели браузера браузерами обычно обеспечивается поддержка следующих сущностей:

* управление фреймами,
* поддержка задержки в исполнении кода и зацикливания с задержкой,
* системные диалоги,
* управление адресом открытой страницы,
* управление информацией о браузере,
* управление информацией о параметрах монитора,
* ограниченное управление историей просмотра страниц,
* поддержка работы с HTTP cookie.

Объектная модель документа

Основная статья: Document Object Model

Объектная модель документа — интерфейс программирования приложений для HTML и XML-документов. Согласно DOM, документ (например, веб-страница) может быть представлен в виде дерева объектов, обладающих рядом свойств, которые позволяют производить с ним различные манипуляции:

* генерация и добавление узлов,
* получение узлов,
* изменение узлов,
* изменение связей между узлами,
* удаление узлов.

Таким образом, можно сделать вывод, что JavaScript ("— это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах. Его разработал Brendan Eich, сооснователь проекта Mozilla, Mozilla Foundation и Mozilla Corporation.

# **ГЛАВА II ПРАКТИЧЕСКАЯ ЧАСТЬ**

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

Далее проанализируем этапы плана разработки веб-сайта .

Перед созданием сайта необходимо создать понятные и конкретные организации сайта. Грамотно разработанный дизайн сайта позволяет логически организовать работу по разработке и подготовке.

На рис. 8 ниже приведена общая схема разработки сайта.

Она включает в себя 4 ключевых этапа.



*Рисунок 8 – Блок-схема этапов проектирования веб-сайта*

Есть множество сервисов, с которыми создание кода из бесконечного поиска ошибок превратится в продуктивное и приятное занятие. Далее проанализируем сервисы, которые помогут облегчить разработку сайта.

Сервисы для работы с прототипами.

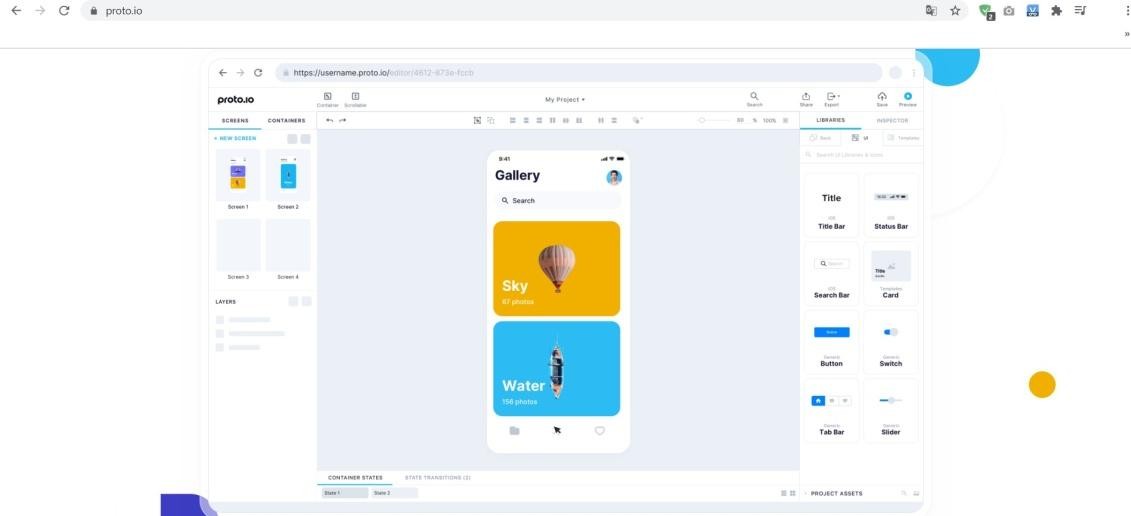
Proto.io. Создать прототип веб-ресурса, провести юзабилити-тестирование, проверить работу блоков и их взаимодействие, безопасность проекта и многое другое поможет онлайн сервис Proto.io (рис. 1). Просмотр данных, а также навигация и проверка функциональности прототипа происходят непосредственно в браузере. Тут же можно вносить необходимые правки.

Рисунок 1 – Сервис Proto.io

Figma. Инструмент Figma выполняет прототипирование сайта. С его помощью команда разработчиков может проектировать страницы, учитывая даже мелкие детали и получая доступную пользователю картину.

Инструменты для упрощения разработки.

Visual Studio Code (VSC). Редактировать, собирать и выполнять отладку исходного кода в VS Code (рис. 2) можно на macOS, Windows и Linux. Есть навигация, адаптеры узлов и отладок, собственные репозиториии и поддержка Microsoft Azure, Node, React, Python, Angular, Veu.

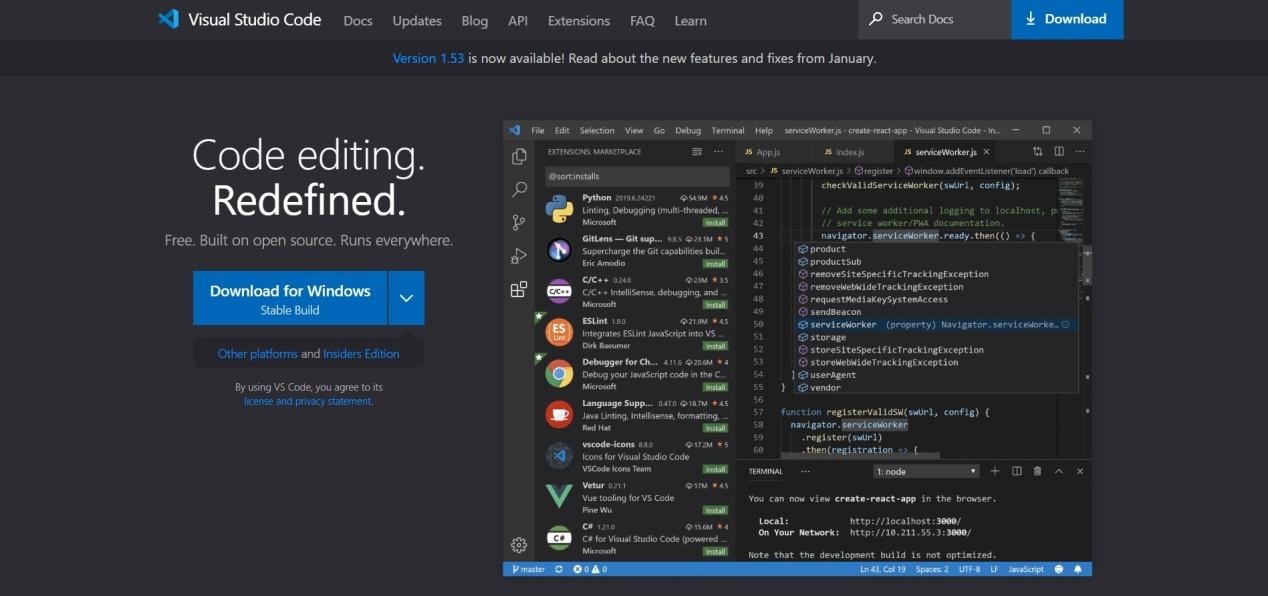


Рисунок 2 – Visual Studio Code

HUGO. Генератор статических сайтов Hugo (рис. 3) представляет собой программу для обработки структурированных исходных данных (медиафайлы, шаблоны, тексты) и генерирования файлов HTML, готовых к загрузке на сервер. Обработка данных осуществляет быстро, но взаимодействовать с программой нужно в режиме командной строки.



Рисунок 3 – Сервис Hugo

Sketch2Code. Преобразовать нарисованный от руки макет сайта в код HTML способен Artificial Intelligence (AI) Sketch2Code(рис. 4). Разработка Microsoft упрощает процесс верстки, а также предлагает образцы дизайна страниц проекта.

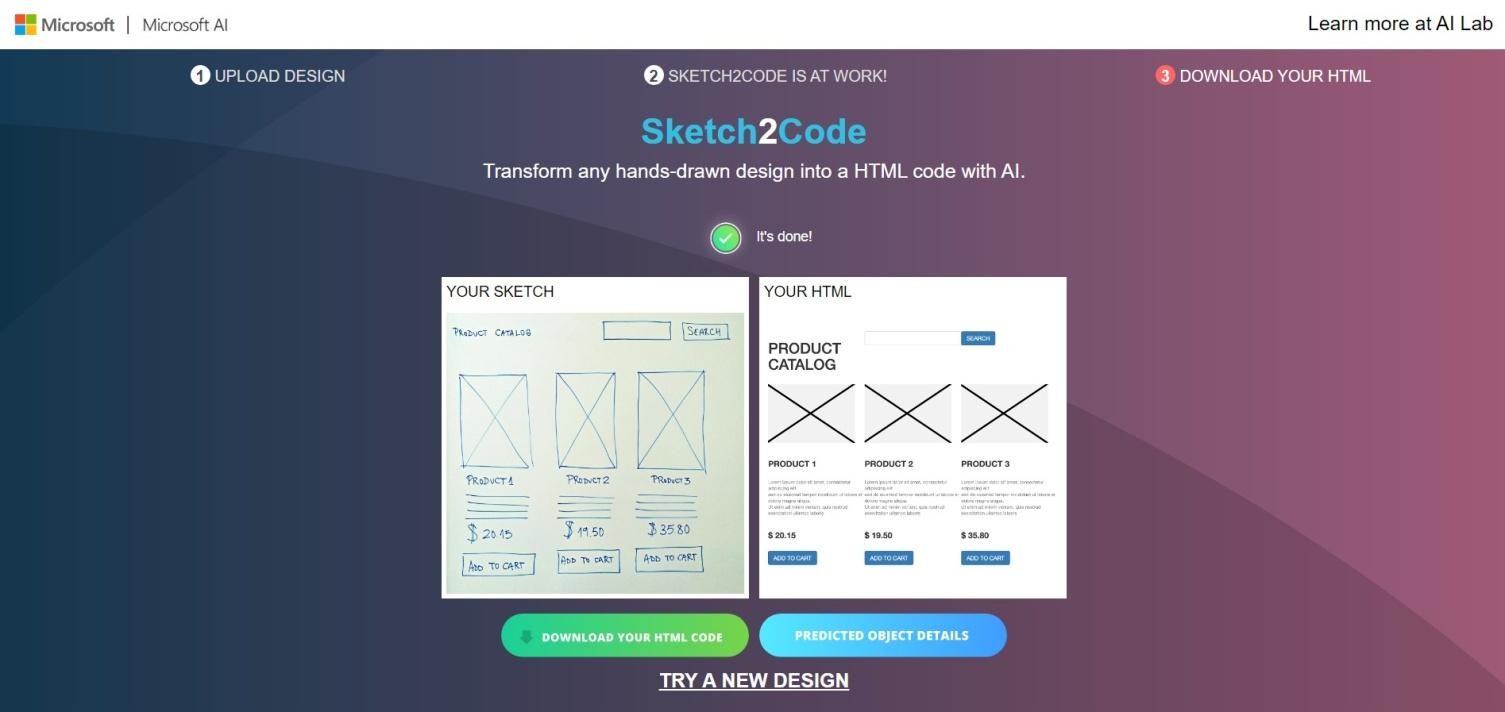


Рисунок 4 – Сервис Sketch2Code

Yarn. Сервис, созданный совместно специалистами Tilde, Facebook, Exponent и Google, упрощает сборку проектов. Yarn (рис. 5) является альтернативой npm, притом работает быстрее аналога на 20%.

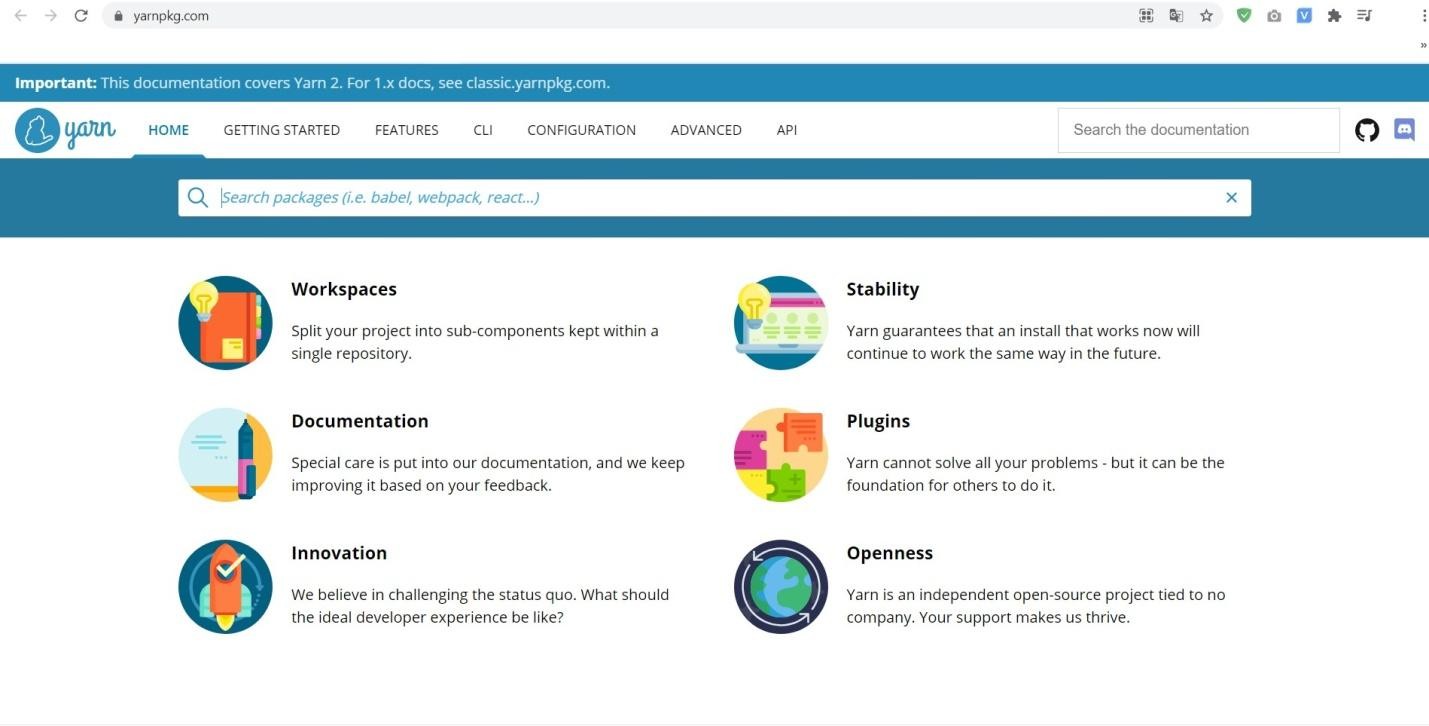


Рисунок 5 – Сервис Yarn

GitHub package registry. Сервис управления пакетами GitHub package registry поддерживает .Net (NuGet), JavaScript (npm), Ruby (RubyGams), Java (Maven) и Docker. Его функциональность постоянно расширяется, а на GitHub публикуют частные и публичные пакеты вместе с открытым кодом.

Сервисы для работы с медиафайлами.

FavIcon Generator. Создать фавикон сайта поможет онлайн-сервис FavIcon Generator. В несколько кликов он позволяет делать иконки размером 16х16, 32х32 и 48х48 пикселей.

Orion Icon. Библиотека Orion Icon предоставляет плоские, цветные, линейные и сплошные иконки на различные тематики. Доступны файлы SVG и векторная графика. Преимущество сервиса заключается в возможности создать свою коллекцию под конкретный проект.

Fontello. Генератор Fontello создает иконки в формате веб-шрифтов. На странице онлайн-сервиса представлены значки, которые можно редактировать и настраивать, формируя коллекцию для собственного проекта.

Фотохостинги.

Ресурсы с фотографиями и видео в высоком качестве:

* Flickr с 2004 года используют для хранения и использования цифровых фотографий и видеоматериалов.
* Pexels позволяет скачать изображение даже незарегистрированным пользователям.
* 500px на нем представлены работы профессиональных фотографов. Сервисы для тестирования.

Firebug. Плагин Firebug для браузера Firefox дает разработчикам возможность провести мониторинг, отладку и редактирование кода JavsScript, CSS и HTML.

Jest. Протестировать код на JavaScript позволяет фреймворком Jest. Сервис поддерживает проекты на Vue, Babel, React, TypeScript, Angular, Node и т.д.

PerfectPixel. Расширение для Chrome PerfectPixel помогает выполнить попиксельное сравнение макета сайта с имеющимся проектом. Полупрозрачное изображение отражается поверх страницы HTML упрощая проверку качества верстки.

Funkify. Увидеть сайт глазами людей с ограниченными возможностями можно при помощи Funkify (рис. 6) для браузера Chrome.

Расширение поддерживает несколько роботов-симуляторов:

* Blurry Bianca при помощи foggy filter предоставляет размытую картинку.
* Color Carl поддерживает фильтры для разных типов дальтонизма.
* Dyslexia Dani имитирует «танцующие» буквы в словах.
* Trembling Trevor показывает движение неконтролируемого курсора мыши.
* Tunnel Toby демонстрирует картинку глазами людей с тоннельным зрением (радиальным и прямоугольным).
* Peripheral Pierre позволяет настраивать режимы слепоты и размеры пятна (области, которую не воспринимает глаз).
* Elderly Ellen одновременно показывает размытую картинку с плохим контрастом и плохо контролируемым курсором.
* Sunshine Sue отображает картинку, как если бы на экран светило солнце (предусмотрено 3 режима настройки).
* Hyperactive Henny поддерживает визуализацию множества всплывающих вкладок, разных звуков и меняющихся настроек сайта.
* Keyboard Kim скрывает курсор – управлять перемещением по вкладке нужно при помощи клавиатуры.
* Robot Robin показывает, какие настройки есть на странице.

На протяжении четырех дней вся функциональность доступна бесплатно, а для дальнейшего использования нужно заплатить $4,99 за месяц. В платной версии есть настройки симуляторов, возможность создания собственных тренажеров и т.д.

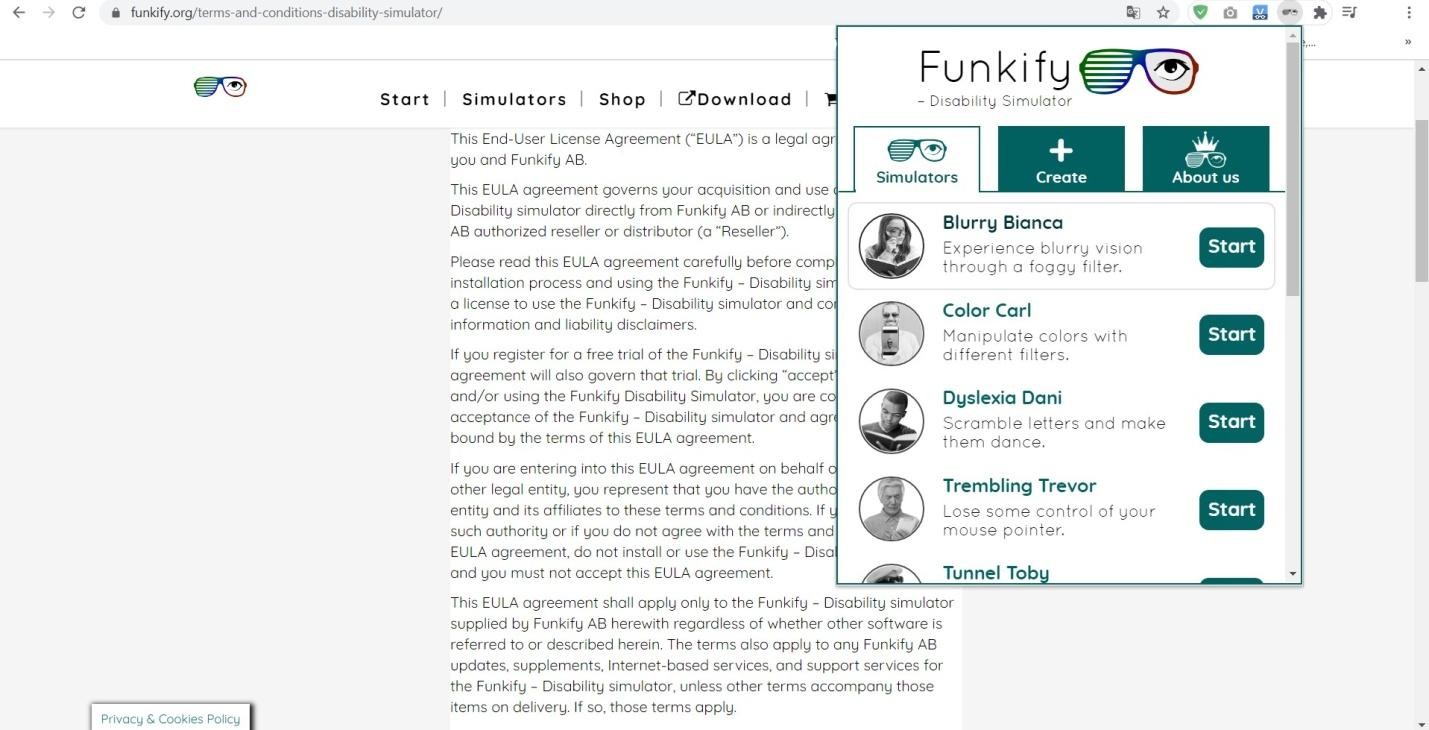


Рисунок 6 – Сервис Funkify

Endtest. Интеллектуальная платформа Endtest для автоматизированного тестирования работает с Safari, Chrome, Edge и Firefox. На официальном сайте есть инструкции по созданию тестов веб-приложений и мобильных приложений, информация

о расширенных настройках и способах размещения результатов на разных платформах (Slack, GitHub, Jenkins, Jira и других).

Load Impact. Облачный сервис генерирует тестовую нагрузку при большом объеме трафика для сайтов, веб-приложений, мобильных приложений и разнообразных API. При необходимости Load Impact эмулирует одновременное посещение ресурса более чем миллионом пользователей.

BugHerd. Выполнить дебаггин сайта на стадии разработки поможет BugHerd. Инструмент анализирует связи между пользователем и сервером. В случае ошибки передачи данных, разработчик получит развернутое описание неисправности и отчет по стабильности системы. BugHerd – облачный сервис, что упрощает синхронизацию работы географически распределенной команды. В настройках есть функции фильтрации и сортировки, делегирования задач, установки приоритетов и уровней доступа. Доступна возможность общения и обмена отзывами внутри команды. Оценить функциональность сервиса можно в бесплатной версии, а затем придется платить $29 в месяц.

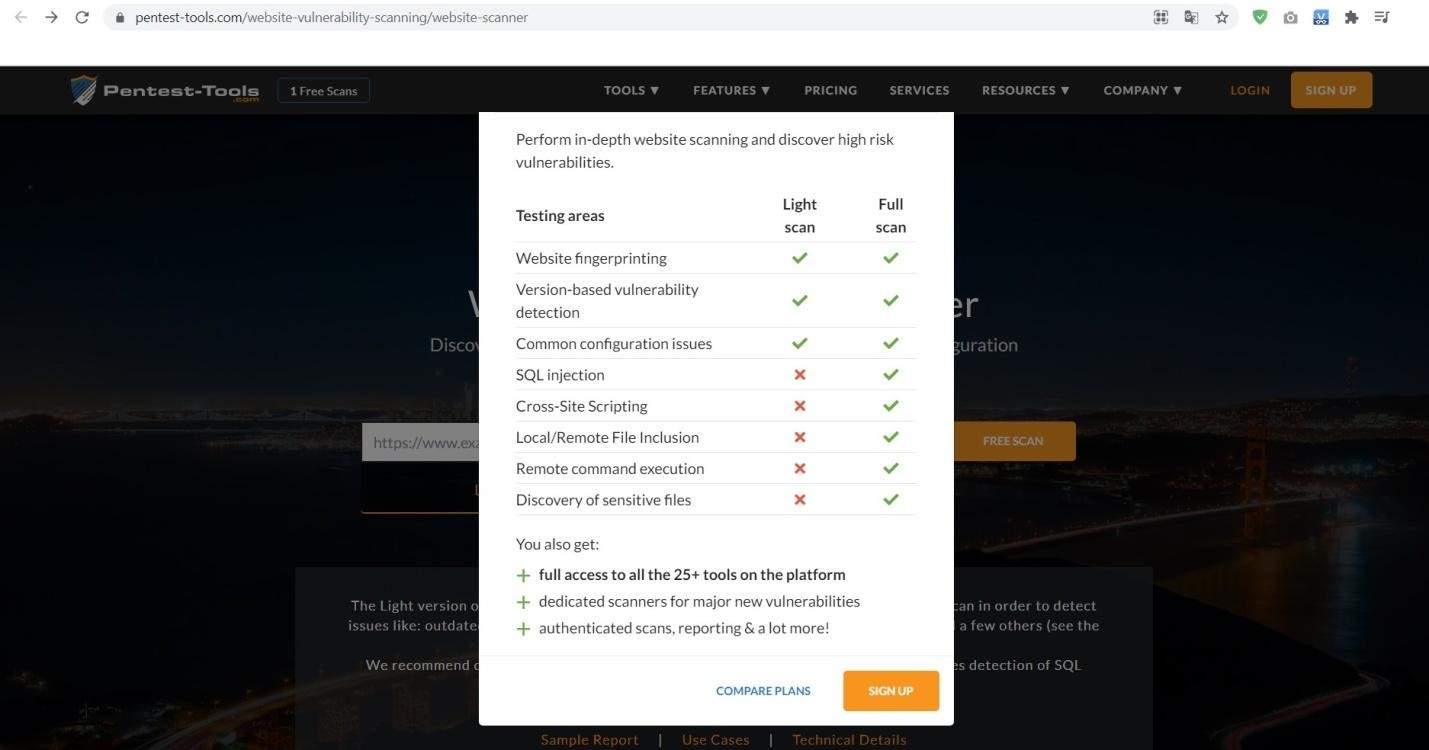
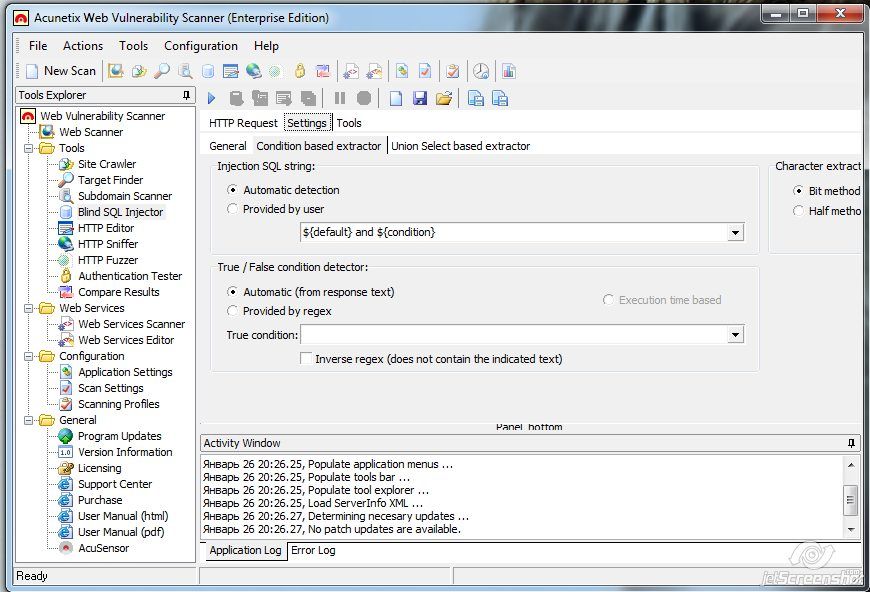
Website Vulnerability Scanner. Проверить уязвимости сайта позволяет разработанный в Pantest-Tools онлайн-инструмент Website Vulnerability Scanner (рис. 7). Он проверяет ресурс по ссылке и генерирует отчет.

Рисунок 7 – Сервис Website Vulnerability Scanner Инструменты



для проверки конверсии ресурса.

Google Lighthouse. Для проверки качества веб-проекта подойдет Google Lighthouse. Оно автоматически оценит качество поисковой оптимизации, уровень производительности и доступности сайта, а также сгенерирует детальные отчеты по результатам теста с рекомендациями по улучшению показателей.

LuckyOrange. Отследить поведение быстро покинувшего сайт пользователя позволяет Lucky Orange. Сервис с тепловыми картами воспроизводит действия человека внутри сессии и анализирует проблемы, которые мешают удерживать внимание аудитории. В аналитическом блоке можно посмотреть историю серфинга посетителя, количество просмотров, способ попадания на ресурс.

Meta Tags. Инструмент Meta Tags проверяет наличие метаданных сайта и сгенерирует их для Twitter, Linkedin, Facebook, Punterest, Slack, Google. Сервис необходим для SEO-настройки проекта, которая привлечет больше посетителей.

# I Этап. Постановка цели и задачи сайта.

На этом этапе необходимо определить, для чего нужен сайт, т.е. какие задач он должен решать: давать общее представление о организации или информировать пользователя, увеличивать продажи, организовать веб-торговлю и провести маркетинговые или рекламные кампании.

Цель и задача сайта практически всегда определяется заказчиком, в данном случае также необходимо определить содержимое на сайте контента. Нужно узнать, какие данные ждут посетителей, это помогает определить текстовую информацию на сайте.

Еще один не менее важный момент - составление ТЗ-задания.

Первая стадия является одним из самых важных этапов для реализации сайта.

# II этап. Реализация сайта

Это этап практический. На этом этапе осуществляется весь функционал сайта.

После подготовки задач и подготовки технической задачи наступает одна из стадий реализации сайта – разработка шаблона. В разработку дизайна входит работа над дизайном сайта и версткой.

Основываясь на поставленных задачах и техническом задании, разрабатываются макеты будущих сайтов. На данном этапе определяются какие графические элементы используются, как выглядит в целом дизайн, какой будет цветовой спектр, какая будет структура страниц. Макет доступен в формате PSD для загрузки дальнейших версий.

Далее происходит «нарезка» готового шаблона на части, а эти части прикрепляются к сайту. Для того, чтобы все элементы страницы были отображены в нужном положении, нужно сверстать страницу. Верстка – процесс написания HTML-кода и CSS. Этот код отвечает за местоположение элементов страницы.

Выбор системы контроля содержимого осуществляется по сравнительному анализу популярных CMS.

Затем выбирается хостинг провайдер и регистрируется домен. В интернете много хостингов, предлагающих площадки на размещение сайтов. Выбирать хостинг зависит от многих факторов: от стоимости и специальных предложений до стабильности хостинга. Регистрация доменного имени может осуществляться на выбранных хостингах или на любых других.

Работа и эффективность сайта тесно связаны с контентом, т.е. графическим и текстовым содержимым. Все тексты сайта должны содержать общие понятия сайта, соответствовать цели и задачи. Материалы наполнение сайта предоставляются заказчиком или заказчиком, которые могут обратиться в копирайтинг.

Интеграция в систему контроля контента происходит дальше. На нашем примере не нужно программировать сайт на нуле. Интеграция с системой управления контентом позволяет заменить стандартные шаблоны, предоставляемые CMS на собственные разработанные шаблоны. Оригинальное шаблонное оформление создано на основе его реализации.

# III этап. Поддержка сайта**.**

Следующий этап - поддержка сайта. На данном этапе проводятся доработки, обновления содержимого сайтов, тестирования и коррекции ошибок в работе сайта.

Актуализация данных на сайте, частота изменения содержимого – важная часть, поскольку это зависит от информированности посетителей сайта.

Тест может включать в себя различные тесты: просмотр сайта в разных версиях браузеров, просмотр сайта с увеличением или уменьшением шрифтов, отключение flash- плеера и JavaScript и многие другие. После выявления ошибок происходит корректировка и обновление сайта до устранения ошибок.

# IV этап. Безопасность сайта.

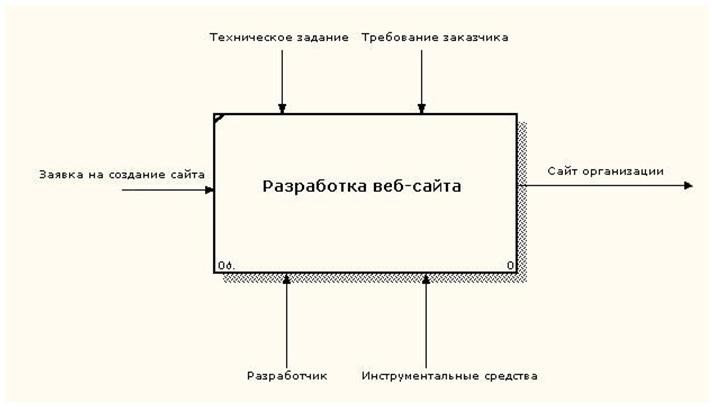
Безопасность сайта является достаточно важным нюансом работы системы. CMS WordPress — Open Source, злоумышленник, используя открытый исходный код, создает уязвимость, вирусы, нередко взломывает сайт. Таким образом, для стабильной работы сайта необходимо учитывать простое, но эффективное средство защиты сайта.

Каждая стадия проектирования связана с другой. Чтобы сохранить целостность и структуру сайта, необходимо спланировать этапы и определять ключевые элементы. Если не сделать этого и «перескочить» с одного и того же этапа, то большая вероятность, что в сущности и логике сайта и программных и дизайнерских моментов может быть утрачена или не соответствующая друг другу суть. Правильно разработанный сайт обеспечивает сохранение целостности сайта, реализацию всего функционала и повышение эффективности конверсии сайтов.

Модель функциональной разработки сайта.

Документирование процесса создания сайта часто требует представления логических связей между взаимодействиями функций в процессе разработки сайта. Среди многочисленных методологических подходов, позволяющих обрисовать процессы бизнеса, IDEF0 является англ. Функциональное моделирование - методика функционального моделирования выделена среди остальных. Модель IDEF0 позволяет рассматривать внутренние функциональные возможности взаимодействия с компонентами.

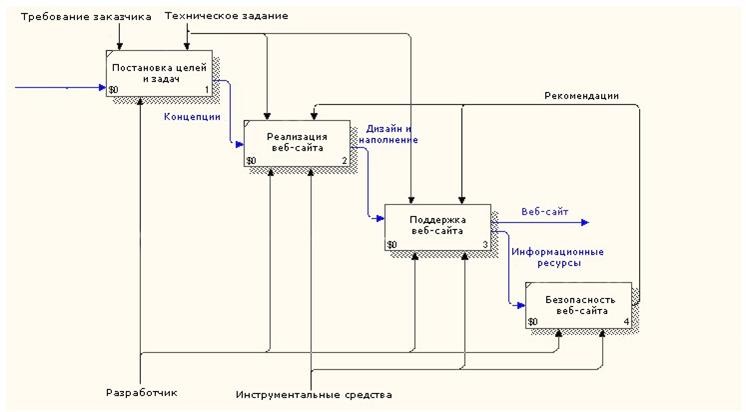
Мы рассмотрим подход к использованию IEF0 в процессе создания сайта. Согласно этому подходу, в первую очередь строятся контекстные диаграммы, отражающие связь процессов разработки сайта с внешним пространством. После выполнения модели процесса создания сайта нужно построить конфигурацию функций для декомпозиции.



На рис. представлена схема функционального процесса создания сайта.

*Рисунок 9 – Функциональная модель разработки веб- сайта*

При построении функциональной модели разработки веб-сайта можно выполнить декомпозицию работ, отраженных на рисунке 10.



*Рисунок 10 – Декомпозиция процесса разработки веб-сайта. Структура веб-сайта.*

На рисунке 11 ниже представлена внутренняя структура веб-сайта, которая учитывает всю специфику разработки.



*Рисунок 11 – Внутренняя структура веб-сайта*

Создание шаблона сайта является одним из основных этапов работы по созданию визуальной части будущего сайта. Процесс позволяет посмотреть на сайт до начала разработки, а также помогает передать идеи клиента дизайнеру. В данной статье мы подробно разберем процесс создания прототипа.

Прототип представляет собой упрощенную посадочную страницу, на которой схематично представлены все функции будущего лендинга. Это какой-то фундамент проектов, благодаря чему можно построить эффективную систему взаимодействия пользователей и интерфейса. Прототип представлен на рис. 12.



*Рисунок 12 – Прототип сайта*

Сайт онлайн-премьеры авторского короткометражного фильма было решено сделать в форме одностраничного лендинга.

# 2.2. Анализ сайтов схожей тематики

Все сайты и все элементы необходимо адаптировать под любые устройства.

Адаптивная верстка является таким способом создания сайтов, которые автоматически подходят к размерам и ориентации экрана устройства и их дизайн зависит от действия пользователя 10.

Google подтвердила, что в апреле 2021 года один из факторов рейтинга станет адаптация сайта к мобильным устройствам.

Первый шаг - базовая страница HTML. 13 лет.

HTML является стандартным языком разметки в создании сайтов, CSS является языком описывания стиля HTML-документов. Для создания базового веб-сайта мы будем использовать html и CSS.

*Рисунок 13 – Код базовой страницы*

<!DOCTYPE html> Декларация определяет этот документ как HTML5.

<html> элемент является корневым элементом HTML-страницы.

<head> элемент содержит мета-информацию о документе.

<title> элемент задает заголовок документа.

<meta> элемент должен определить кодировку UTF-8.

<meta> элемент с именем = «видовой экран» делает веб-сайт хорошо выглядеть на всех устройствах и разрешение экрана.

<style> элемент содержит стили для веб-сайта (макет/дизайн).

<body> элемент содержит видимое содержимое страницы.

<h1> элемент определяет большой заголовок.

<p> элемент определяет абзац.

Внутри <body> элемента нашего сайта, мы будем использовать наш «макет проекта» и создать:

* A header;
* A navigation bar;
* Main content;
* Side content;
* A footer.

HTML5 ввел несколько новых семантических элементов. Семантические элементы важны для использования, поскольку они определяют структуру веб-страниц и помогают читателям экрана и поисковым системам правильно читать страницу.

Вот некоторые из наиболее распространенных семантических HTML-элементов:

Элемент <Section> можно использовать для определения части веб-сайта с соответствующим содержимым.

Элемент < article > можно использовать для определения отдельного фрагмента содержимого.

Элемент <Header> можно использовать для определения заголовка (в документе, разделе или статье).

Элемент <Footer> можно использовать для определения нижнего колонтитула (в документе, разделе или статье).

Элемент <Nav> можно использовать для определения контейнера навигационных ссылок.

Заголовок обычно находится в верхней части веб-сайта (или прямо под верхним навигационным меню). Он часто содержит логотип или название сайта:

<div class=«header»>

<h1> Где ты, Евген? </h1>

<p> Сайт краткометражного фильма p>

</div>

Затем мы используем CSS для стиля заголовка:

.header {

Padding-top: 78px; text-align: align-center; Background-color: #1a119c;

color: white;

}

.header h1 {

font-size: 40px;

}

Далее создадим меню навигации для сайта.

Начнем с html. Прежде всего подключим библиотеку jquery:

<script src=«https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js»></script> Теперь разметка. Обычный ненумерованный список, и небольшой div. Код:

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec90835941» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska- menu=«3»>Фотогалерея</a>

</li>

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec90835943» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«4»>Логлайн</a>

</li>

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec90835945» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«5»>Синопсис</a>

</li>

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec244845426» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«6»>Интервью</a>

</li>

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec90835952» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«7»>Трейлер</a>

</li>

<li class=«menu spisok»>

<a class=«ssylka-menu» href=«#rec288667477» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«8»>Отзывы зрителей</a>

</li>

<li class=«menu spisok»>

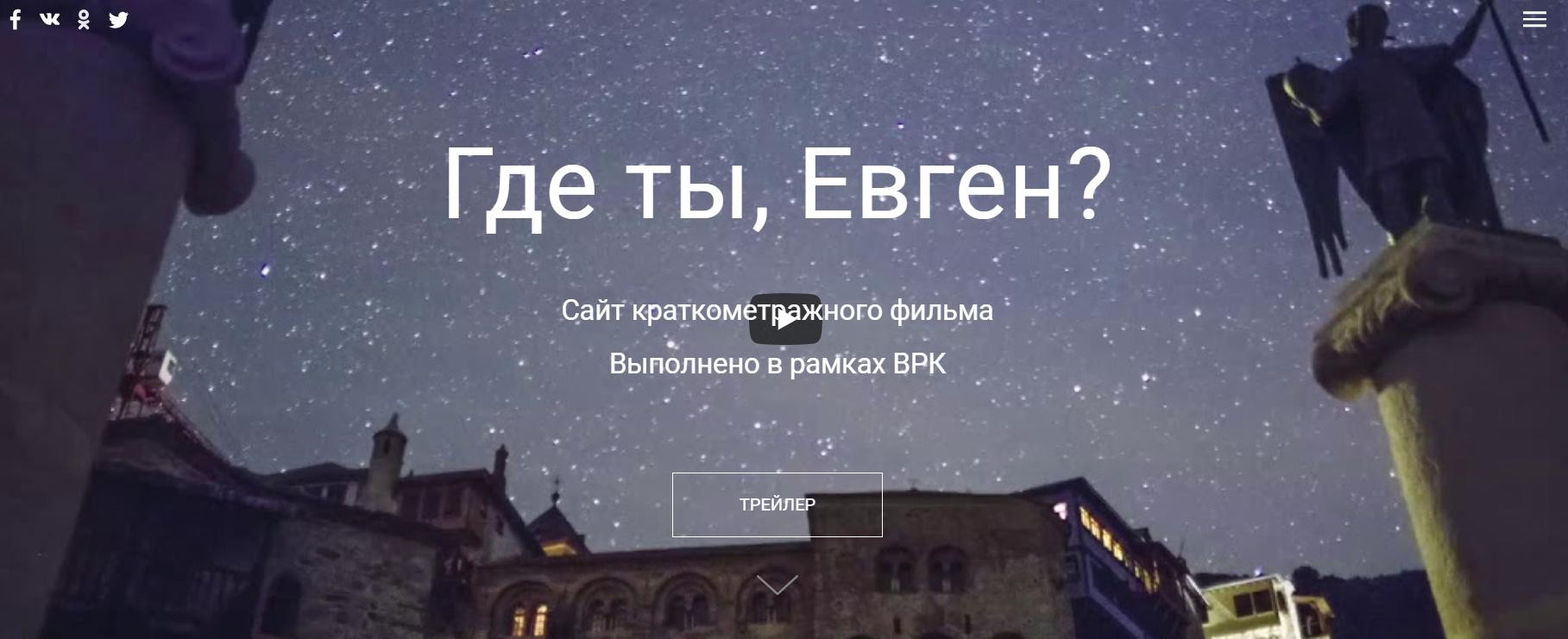
<a class=«ssylka-menu» href=«#rec90835966» site-men=«« style=«color:#ffffff;font-size:24px;font-weight:300;» nomer-spiska-menu=«9»>Контакты</a>

</li>

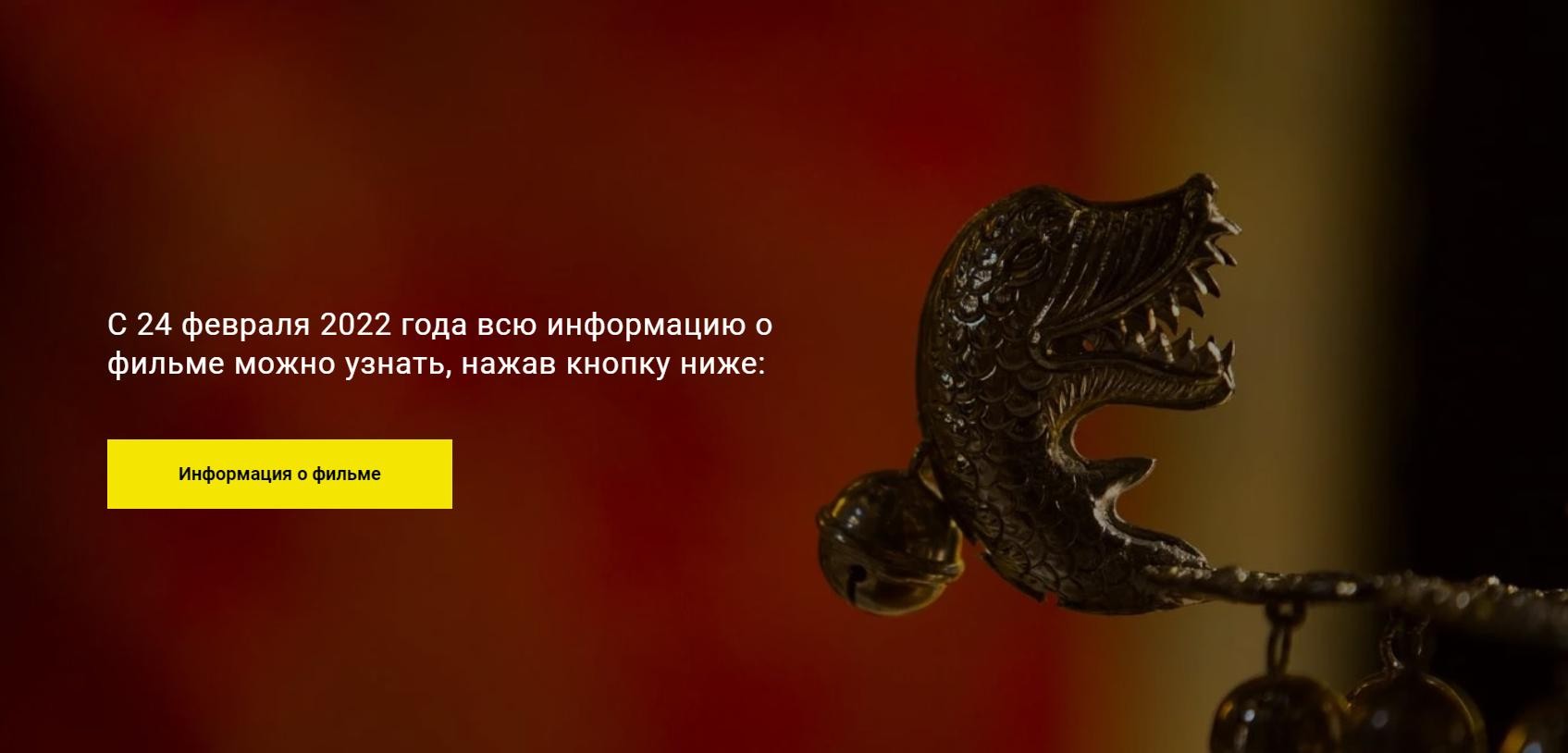
</ul>

</div>

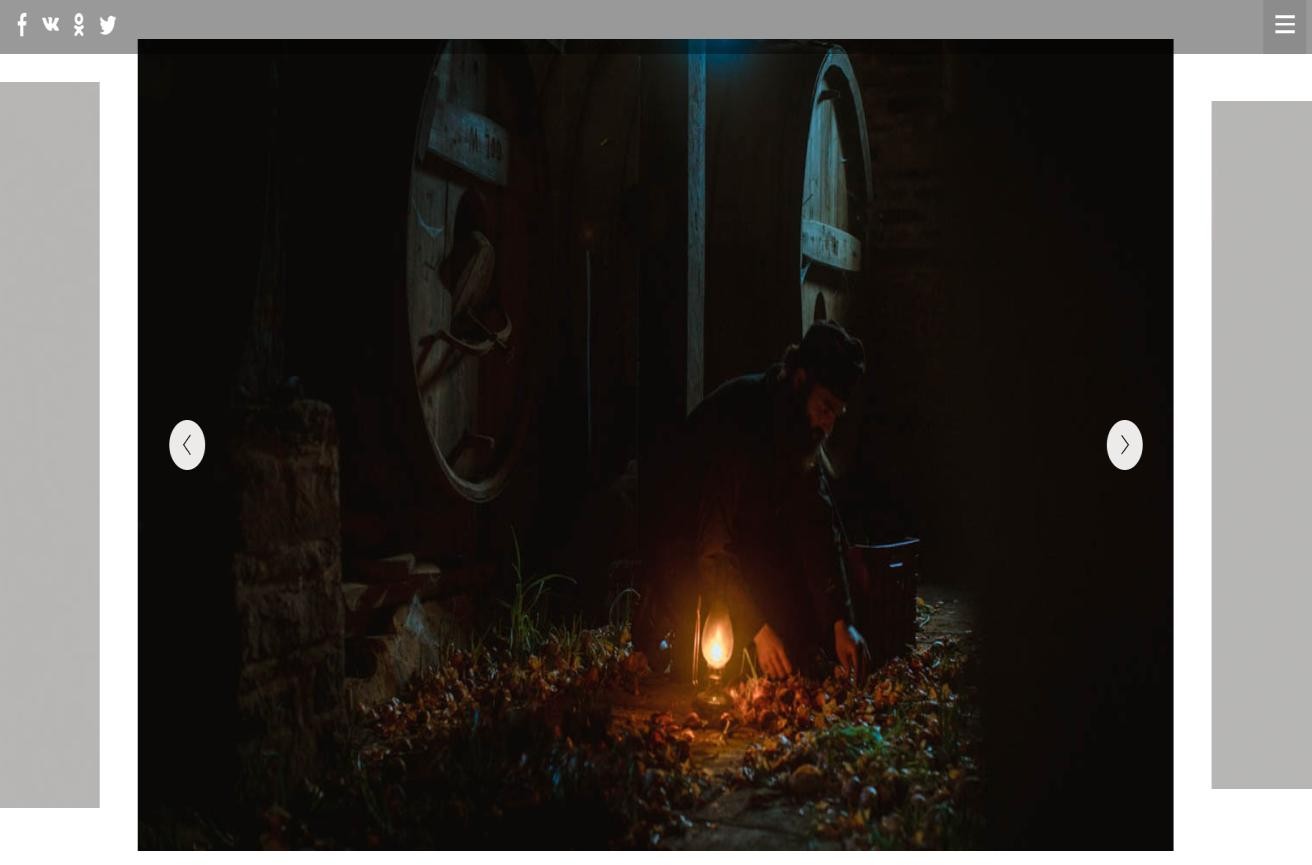
Первая страница показана на рисунке 1

*Рисунок 1 – Первая страница*

Далее идёт блок с информацией о дате создания фильма, с кнопкой узнать про фильм подробнее как показано на рисунке 2.



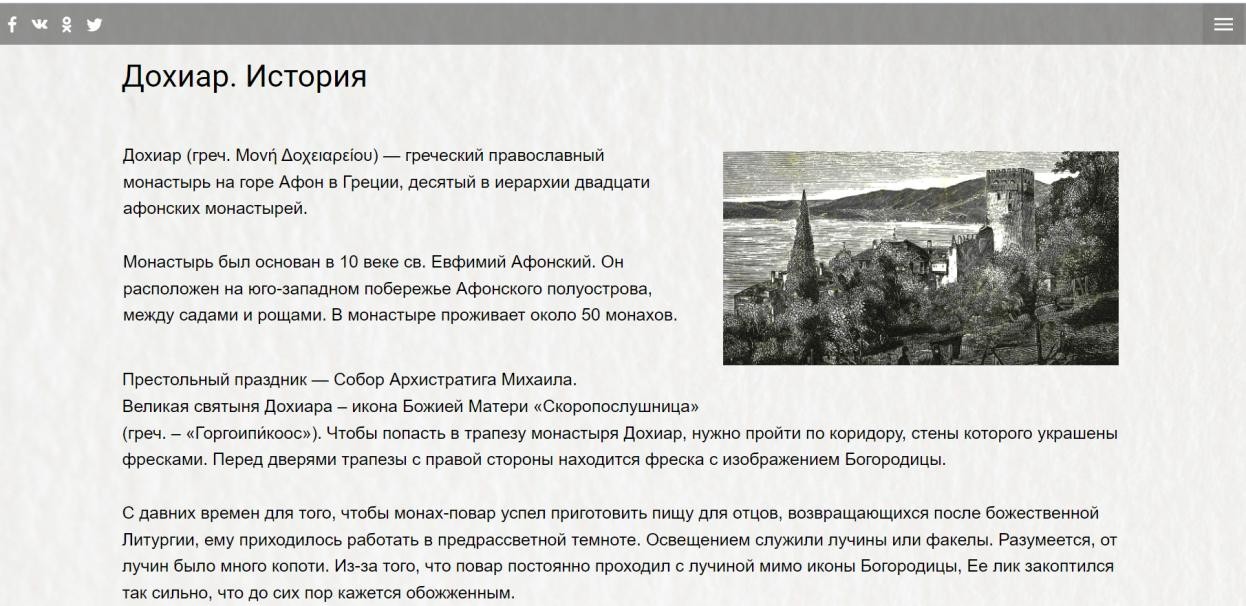
*Рисунок 2 – Информация о дате выхода*

Следующим блоком сайта идёт галерея фильма. Галерея сайта содержит кадры с фильма. Кадры можно пролистывать используя кнопку навигации на лево или на право как показано на рисунке 3.

*Рисунок 3 – Галерея кадров*

Далее идут блоки с информацией о фильме. Это историческая справка, сюжет, небольшое ознакомление с кадрами.

Блоки с информацией о фильме показаны на рисунке 4 и 5.



*Рисунок 4 – История о монастыре с фильма*



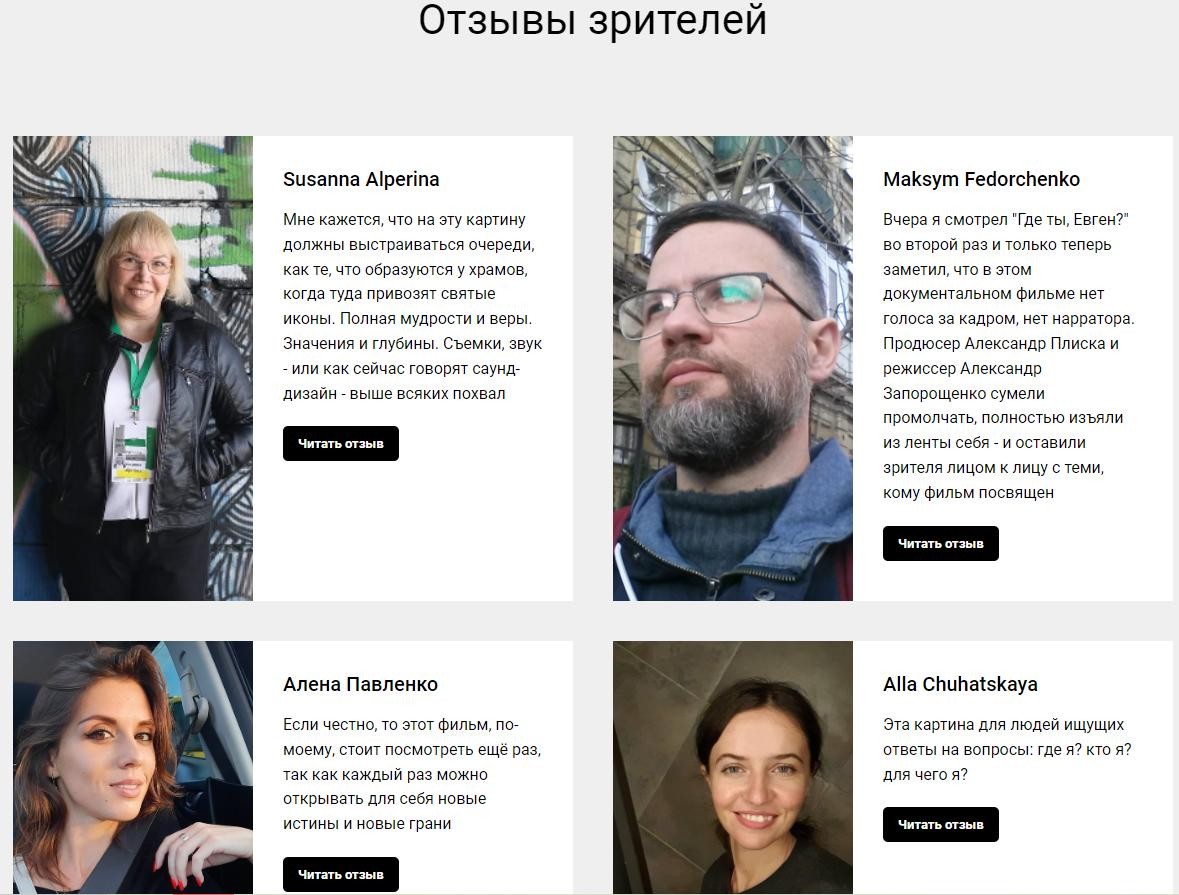
*Рисунок 5 – Страница сюжетной линии фильма*

Далее содержится блок трейлера фильма. При нажатии на проигрыватель воспроизведётся трейлер.



*Рисунок 6 – Страница с трейлером фильма*

Блоком ниже показаны отзывы о фильме. Каждый отзыв содержит фото от кого отзыв и текст отзыва как показано на рисунке 7.



*Рисунок 7 – Блок с отзывами*

Выводы по разделу.

В этом разделе было произведено создание и подключение стилей для нашей целевой страницы, а именно:

* верстка верхнего раздела;
* верстка среднего раздела;
* верстка нижнего раздела.

По результатам нашей работы можно оценить дизайн-проект веб-сайта авторского короткометражного фильма. Главной работой была верстка целевой страницы со всеми разделами, которые важны для продвижения данного проекта в медиа-сфере.

Тестирование сайтов является последним и обязательным этапом технического развития сайта.

Он играет важную роль в создании актива, потому что он качественный. Дополнительные сроки эксплуатации актива зависят от качества тестирования. Увы, часто разработчики не обращают внимания на этот этап, используя свой опыт. В конце концов, существующие ошибки приносят огромные затраты времени и средств.

Ведь ресурсы, на которых ошибки, вызывают негатив посетителей, а затем их потери. В результате владельцу актива приходится платить за проработку и иногда за перепланировку. Сотканные фичи и взаимодействие с бывшими создателями дряного сайта оставляют лишь неприятные ощущения.

Тест нагрузки сайта.

Проверка загрузки сайта - это тест, который позволяет определить загрузку ресурса на скорость. Чем больше загруженных страниц тем сложнее продвижение и пользователям будет меньше комфорта. Таким образом, скорость скачивания сайтов должна быть регулярно контролирована. Тест по скорости скачивания сайта проводился в сервисе Google pageSpeed Inspects.

Система рейтингов выглядит так:

0-49 – плохо оптимизированный сайт;

50-89 - хороший оптимизированный сайт;

90-100 – идеально оптимизированный сайт.

После проверки сервис Google Page Insights получил 59 балла. На нашем примере это хороший оптимизированный сайт.

Подробнее об итогах:

Время первого загрузки контента – 1 - 2 секунды.

Первый скриншот содержимого - это показатель, который определяет временный интервал между загрузкой страницы, а также появлением первой картинки или текстового блока.

Индекс загрузки 3.9 сек.

Индекс загрузки определяет, как быстро становится доступно содержимое сайта для пользователя. Обзор.

Время загрузки для связи - 3,0 сек.

Время загрузки для общения - время, в котором страница абсолютно готова к взаимодействия с клиентом. Время скачивания достаточного контента – 1,5 секунды.

Этот параметр показывает время, в котором содержимое основной страницы будет опубликовано.

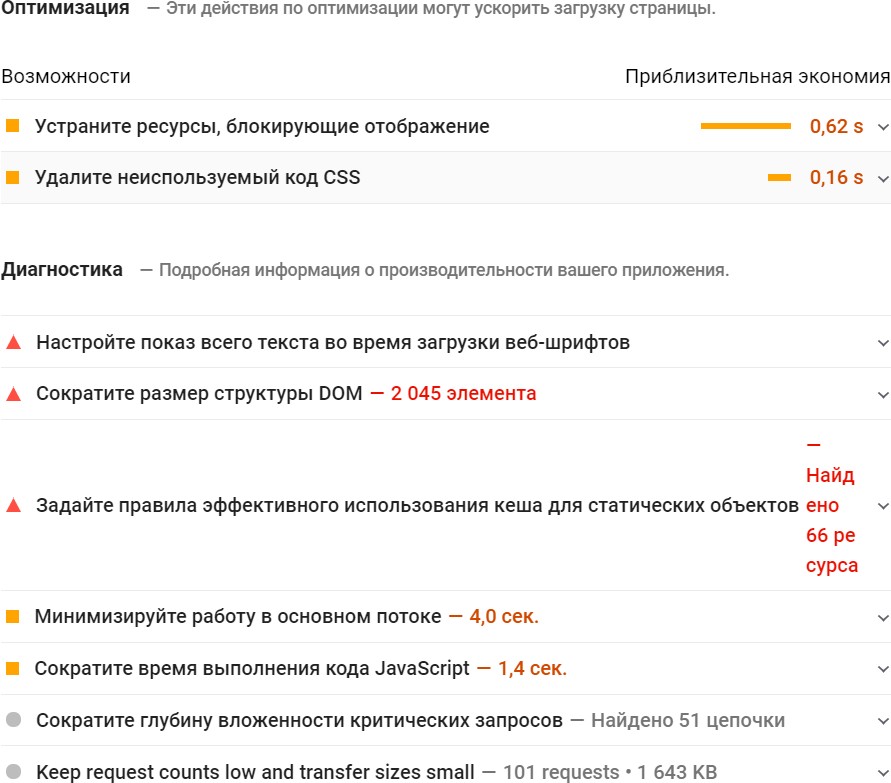
Время отключения ЦП — 2 - 9 секунд.

Этот параметр характеризует время того, как основной поток страницы был достаточно свободен для ручной обработки.

Максимальный потенциальный срок задержки FID - 680мс.

Постоянная задержка при первом входе FID указывает максимальное время для выполнения задач в миллисекундах.

Также после тестирования скорости скачивания сайта появились такие рекомендации для ускорения скачивания сайта.



*Рисунок 21 – Рекомендации для ускорения загрузки* сайта Проверка была выполнена в версии Opera 76.

Проверьте сайт на вирус.

За последнее время речь идет о вирусе на сайтах довольно часто. Посетив этот сайт, большая вероятность того, что вы будете подхватывать какую-то заразу, которая серьезно ухудшит ваши настроения.

Если вы владеете сайтом с вирусным кодом, то рискуете убить почти всех посетителей сайта. А не только из-за заражения компьютером, многие и не замечают, потому что не все используют антивирусы. Поисковые системы маркируют подобные сайты, а в результате поиска пользователи сообщают о том, что «Сайт может навредить вашему компьютеру» Яндекс.ru или «Сайт может навредить вашему компьютеру» Google.ru. Конечно, никто из здравомыслящих людей не перейдёт на сайт с подобной

«черной меткой».

Тест прошел через сервис 2ip.ru. В результате тестирования вирусов их не было выявлено ни (рис. 22).

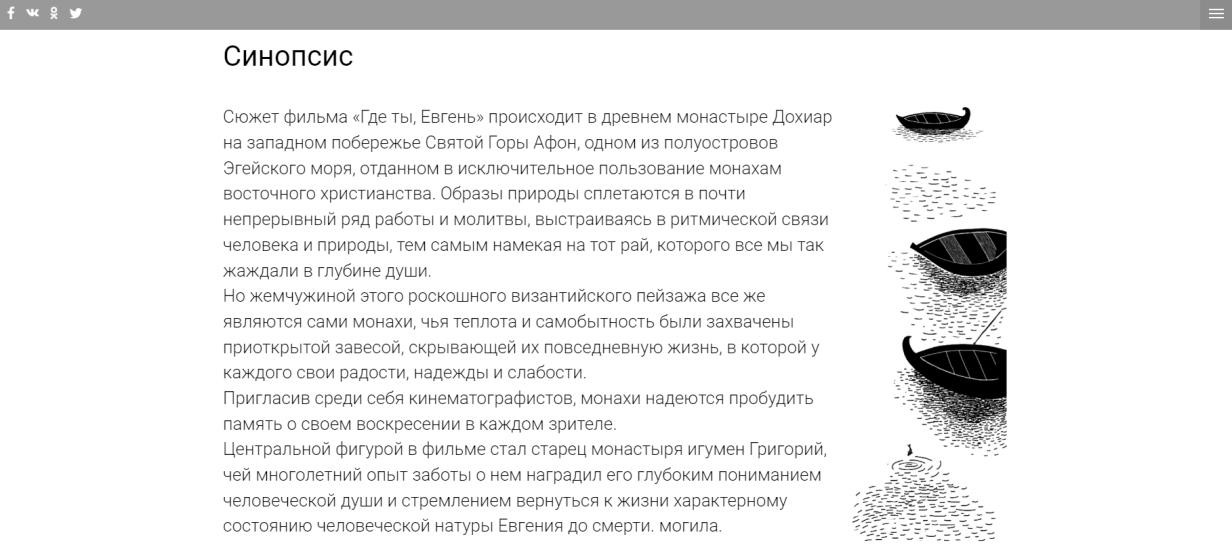
*Рисунок 22 – Результат проверки сайта на вирусы*

Одним из стандартных требований к разработке качественных сайтов является кроссбраузерная оптимизация и правильная отображение на многих современных устройствах. Таким образом, каждый веб-разработчик нередко сталкивается со спросом на тестирование сделанного веб-сайта по разным браузерам.

Кроссбраузер - это возможность отображать сайт и работать в любом браузере, идентична. Идентификация означает отсутствие шаблонов и возможность отображения материала с равной читабельностью. Очень часто понятие кроссбраузерности путают с соответствием попикселей, что в действительности - разные понятия. Для тестирования выбраны 4 наиболее популярные браузеры: Google, Chrome, Опера, Internet Explorer, Mozilla Firefox.

Результат всегда был одинаков.

Сайт правильно отображался на (рис. 23).



*Рисунок 23 – Пример отображения сайта в браузере Opera версия 76*

Тест на адаптивность.

Адаптивный Web-дизайн (responsive Web design, RWD) — это подход к разработке сайтов, согласно которому ресурс должен быть удобным для просмотра с любого устройства, независимо от размера экрана, будь то настольный компьютер, мобильный телефон или планшет.

Для максимально эффективного управления шириной колонок и отступами между ними наилучшим вариантом является использование 12 колоночной структуры. Тест на адаптивность проводился с помощью сервиса quirktools.com. Результаты были следующими:

Для теста на совместимость с мобильными устройствами был взят телефон Samsung galaxy A12 с разрешением 1600х980, результат показан на рисунке 24.

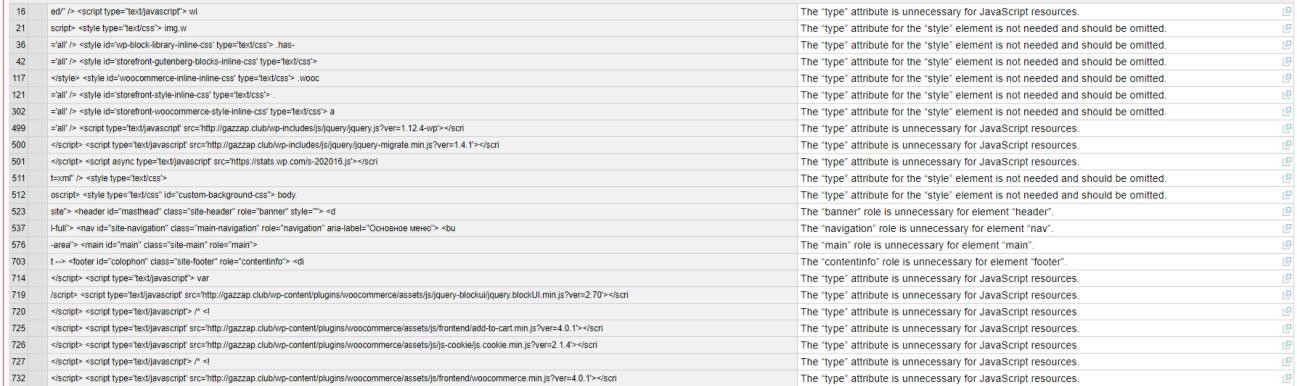
*Рисунок 24 – Отображение сайта на мобильном устройстве*



Проверка на валидность кода.

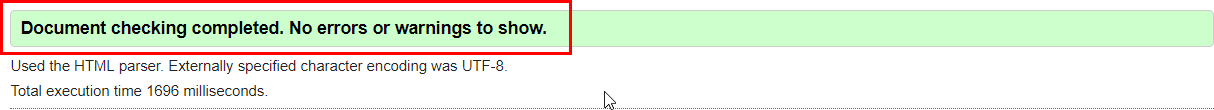
Валидация является одним из самых важных аспектов хорошего Web-дизайна. Проверка Web-страницы позволит определить, соответствует ли она стандартам,

разработанным Консорциумом Всемирной паутины (W3C). Проверка осуществлялась на сайте w3.org. Результаты проверки показаны на рисунке 25.



*Рисунок 25 – Проверка кода на валидность*

Было выявлено 4 ошибки. После исправления сайт проверили ещё раз. Результаты повторной проверки показаны на рисунке 26.

*Рисунок 26 – Проверка на валидность после исправления ошибок*

Как видно на рисунке 26, после исправления ошибок и повторной проверки, валидатор не указывает на ошибки.

**Выводы:**

В данном разделе было протестирован сайт на следующие параметры:

* Тест на вирусы;
* Тест на адаптивность;
* Тест на скорость загрузки сайта.

Тест на вирусы показал, что на сайте не обнаружено вредоносного кода, который мог бы навредить компьютерам пользователей. Тест на адаптивность показал, что сайт адаптируется под все устройства. Тесть на скорость загрузки показал, что он загрузился за 1,2 секудны, что является хорошим результатом.

# **III Экономическая часть**

***При расчете экономической эффективности создания сайта определяются следующие показатели:***

1. Затраты:

-единовременные;

- ежемесячные (постоянные).

2. Доход (выручка).

3. Прибыль.

4. Срок окупаемости.

Расчет затрат выполняется по статьям калькуляции.

В таблице 1.1 представлены единовременные затраты на создание -сайта. капитальный

Таблица 1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | кол-во | кВт/час | кВт в сутки (примерно) | кВт в месяц |
| Компьютер | 1 | 0,17 | 1,53 | 45,9 |
| Энергообеспечение | 3 | 0,36 | 9,72 | 291,6 |
| Сплит | 1 | 0,7 | 6,3 | 189 |
| ИТОГО: |  | 1,23 |  | 526,5 |

Расчет электроэнергии для девятичасового рабочего дня.

Для предприятий 1 кВт / ч= 1,56 руб.

В месяц 1,56\*526,5= 821,34 рублей.

Заработная плата программисту составляет 15000 рублей.

Таблица 1.2

Расчет ежемесячных затрат на содержание Web-сайта.

|  |  |  |
| --- | --- | --- |
| Наименование | Сумма, руб. | ЕСН, руб. |
| Зарплата программиста | 15000 | 3947 |
| Зарплата курьера | 7000 | 1842 |
| Транспортные расходы курьера | 1700 |  |
| Электроэнергия | 1521,6 |  |
| Хостинг | 1200 |  |
| Интернет | 700 |  |
| Прочие расходы | 800 |  |
| Итого: | 23592,6 | 5129 |
| Всего расходов: | 28721,6 | |

RПост = 28721,6 - постоянные ежемесячные расходы.

Годовая сумма амортизационных отчислений рассчитывается по формуле:

А= Ф\*NА / 100% ,

где Ф - первоначальная стоимость основных фондов по видам, руб.;

NA - норма амортизации по видам основных фондов, в %.

Годовую сумму амортизационных отчислений отразим в таблице 1.3

Таблица 1.3

Расчет годовой суммы амортизационных отчислений

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Элементы основных фондов. | Кол-во | Стоимость, руб. | Сумма руб. | Норма амортизации, % | Амортизационные отчисления, руб. |
| Компьютер | 1 | 28000 | 28000 | 20% | 5600 |
| Сплит система | 1 | 19000 | 19000 | 20% | 3800 |
| Помещение | 13,6м2 | 5400 | 73440 | 3% | 2203,2 |
| ИТОГО: | | | | | 11603,2 |

Таким образом, годовая сумма амортизационных отчислений составляет 11603,2 рублей.

Исходя из того, что трудоёмкость создания информационной системы составляет 10 дней, рассчитываем амортизацию оборудования за этот период по формуле:

Афакт= Агод\*Тфакт /365,

Рассчитаем сумму амортизационных отчислений для перечисленной группы оборудования с учетом числа календарных дней на разработку программного обеспечения (Web-сайта) по формуле:

А = 11603,2\*10/365=317,9 руб.

Заработная плата программиста составляет 15000руб. Соответственно, затраты на заработную плату включаемые в себестоимость программы с учетом работы над программой в течение 10 дней составят:

ЗПпр=ЗПмес\*Тфакт / Д,

где ЗПпр - заработная плата в месяц программиста, руб.;

Тфакт - число календарных дней на разработку интернет - магазина;

Д - число дней в периоде (месяц).

ЗПпр = 15000\*10/22 = 6818 руб.

Отчисления на социальное страхование составят:

ЕСН = ЗПпр \* 26%

ЕСН = 1772,68 руб.

Таблица 1.4

Расчет ежемесячных материальных затрат.

|  |  |
| --- | --- |
| Наименование | Сумма, руб/мес. |
| Электроэнергия | 821,34 |
| Хостинг | 1200 |
| Интернет | 700 |
| Прочие расходы | 800 |
| Итого: | 3521,34 |

Зм=3521,34 рублей в месяц

Следовательно, затраты на период разработки программного продукта рассчитаем по формуле:

Зпр=Зм\*Тфакт / Д

где Зм - ежемесячные затраты, руб.;

Тфакт - число календарных дней на разработку интернет - магазина;

Д - число дней в периоде (месяц).

Зпр =3521,34\*10/22=1600,6 руб.

Рассчитаем себестоимость программного продукта по формуле:

Сст - себестоимость разработки программы

Сст = Зпр + ЗПпр + ЕСН + А

Сст = 6818+1772,68+1918,9+317 = 10826,58 рублей.

Данная себестоимость является приблизительной, так как в ней не учтены некоторые детали, которые существенно не повлияют на итог.

Сст ≈ 11000 рублей.

Исходя из нормального уровня рентабельности 20% мы можем определить цену разработанной нами программы:

Ц=Сст+Ссп\*R / 100%,

где Сст - себестоимость разработки программы;

R - планируемый уровень рентабельности.

Ц = 11000 + 11000\*20/100 = 13200 рублей.

Затраты на внедрение программного продукта составят 13200 рублей.

Предполагается, что самоокупаемость Web-сайта при ежемесячных затратах в 28721,6 рублей произойдет уже при увеличении товарооборота на 20% прибыли на 6%.

П = (Тобн + Тобн\* Тоб%) \*П%,

где П - предполагаемая прибыль;

Тобн - базовый вариант товарооборота;

Тоб% - предполагаемый процент прироста товарооборота;

П% - предполагаемый прирост прибыли.

П= (400000+400000\*20/100) \*6/100=28800руб

Так как продажи стройматериалы в компании составляют примерно 80% от общего товарооборота, соответственно самоокупаемость сайта произойдет при приросте товарооборота на 10% и приросте прибыли на 25%.

П1 = (100000+100000\*10/100) \*25/100 = 27500 руб.

Поэтому полная окупаемость Web-сайта с учетом затрат на внедрение произойдет за 1,5 месяца

Таким образом, произведенные расчеты подтверждают предположение о том, что получение прибыли Web-сайта ожидается со второго месяца после начала его деятельности.

Итак, можно сделать следующий вывод: финансовое состояние сайта - это способность сайта финансировать свою деятельность. Оно характеризуется обеспеченностью финансовыми ресурсами, необходимыми для размещения сайта на сервере, целесообразностью их размещения и эффективностью использования, финансовыми взаимоотношениями с другими похожими сайтами, платежеспособностью и финансовой устойчивостью. Оно также может быть устойчивым, неустойчивым.

В итоге получена приблизительная себестоимость продукции, равная ≈ 11000 рублей. В данном проекте были рассмотрены все показатели экономической эффективности. Приведены все формулы расчетов показателей экономической эффективности.

Размещено на Allbest.ru

# **IV ТЕХНИКА БЕЗОПАСНОСТИ**

Требования безопасности во время эксплуатации ЭВМ

Перед началом работы с ПК работник обязан:

– проветрить рабочее помещение;

проверить:

– устойчивость положения оборудования на рабочем столе;

– отсутствие видимых повреждений оборудования, дискет в дисководе системного блока;

– исправность и целостность питающих и соединительных кабелей, разъемов и штепсельных соединений, защитного заземления (зануления);

отрегулировать:

– освещенность на рабочем месте. При необходимости включить местное освещение;

– протереть поверхность экрана монитора, защитного фильтра (при его наличии) сухой мягкой тканевой салфеткой;

– убедиться в отсутствии отражений на экране монитора, встречного светового потока;

– включить оборудование ПК в электрическую сеть, соблюдая следующую последовательность: стабилизатор напряжения (если он используется), блок бесперебойного питания, периферийные устройства (принтер, монитор, сканер и другие устройства), системный блок.

Запрещается приступать к работе при:

– выраженном дрожании изображения на мониторе;

– обнаружении неисправности оборудования;

– наличии поврежденных кабелей или проводов, разъемов, штепсельных соединений;

– отсутствии или неисправности защитного заземления (зануления) оборудования.

Требования безопасности во время работы

Во время работы с ПК работник обязан:

– соблюдать требования охраны труда;

– содержать в порядке и чистоте свое рабочее место;

– держать открытыми вентиляционные отверстия оборудования;

– соблюдать оптимальное расстояние от экрана монитора до глаз.

Работу за экраном монитора следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья, или заменять другой работой с целью сокращения рабочей нагрузки у экрана.

Оператору и пользователю ПК во время работы запрещается:

– касаться одновременно экрана монитора и корпуса компьютера или клавиатуры;

– прикасаться к задней панели системного блока компьютера при включенном питании;

– переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

– самостоятельно вскрывать и производить ремонт компьютеров;

– класть посторонние предметы на клавиатуру, монитор, системный блок, мышь и другое компьютерное оборудование, а также на электрощиты, электропроводку и другое электрическое оборудование.

Время регламентированных перерывов в течение рабочего дня (смены) устанавливается в зависимости от его (ее) продолжительности, вида и категории трудовой деятельности.

Продолжительность непрерывной работы с ПК без регламентированного перерыва не должна превышать 2 часов.

Во время регламентированных перерывов для снижения нервно-эмоционального напряжения, утомления зрительного анализатора, улучшения функционального состояния нервной, сердечно-сосудистой, дыхательной систем, а также мышц плечевого пояса, рук, спины, шеи и ног целесообразно выполнять комплексы упражнений.

Работникам с высоким уровнем напряженности труда во время регламентированных перерывов и в конце рабочего дня показана психологическая разгрузка в специально оборудованных комнатах психологической разгрузки.

С целью уменьшения отрицательного влияния монотонности необходимо применять чередование операций.

При работе с текстовой информацией следует отдавать предпочтение физиологически наиболее оптимальному режиму представления черных символов на белом фоне.

Требования безопасности в аварийных ситуациях

Оператор и пользователь ПК обязан:

– во всех случаях обнаружения обрыва проводов электропитания, неисправности заземления и других повреждений оборудования немедленно отключить питание компьютера и сообщить руководителю;

– следить за исправностью компьютера и при появлении необычного звука, дыма, запаха горелой резины, огня, самопроизвольного отключения аппаратуры – прекратить работу на компьютере, немедленно отключить питание и сообщить руководителю;

– при любых случаях сбоя в работе компьютерного оборудования и программного обеспечения немедленно сообщить руководителю;

– в случае появления рези в глазах, резком ухудшении видимости, появлении боли в пальцах и кистях рук, усилении сердцебиения немедленно покинуть рабочее место, сообщить о происшедшем руководителю;

– при обнаружении человека, попавшего под напряжение, немедленно освободить его от действия тока путем отключения электропитания и до прибытия врача оказать потерпевшему первую медицинскую помощь. Сообщить руководителю; при возгорании оборудования отключить питание и принять меры к тушению очага пожара при помощи огнетушителя, вызвать пожарную команду и сообщить руководителю.

Требования безопасности после окончания работы

После окончания работ оператор и пользователь ПК обязан:

– корректно закрыть все активные задачи;

– при наличии дискеты в дисководе извлечь ее;

– выключить питание системного блока;

– выключить питание всех периферийных устройств;

– отключить блок бесперебойного питания;

– отключить питающий кабель от сети;

– осмотреть и привести в порядок рабочее место;

– о неисправностях оборудования и других замечаниях по работе с ПК сообщить непосредственному руководителю или лицам, осуществляющим техническое обслуживание оборудования; при необходимости вымыть с мылом руки.

В данном разделе были рассмотрены основные положения и нормы охраны труда, необходимые для работы оператору компьютерного набора.

# **ЗАКЛЮЧЕНИЕ**

В процессе разработки сайта в онлайн-премьере авторской короткометражной картины проанализировали теоретический материал о технологиях разработки сайта и рассмотрели их классификацию. Классификация сайта довольно сложная, поскольку для него используется большое число параметров: дизайн, тему, способ его создания, внутреннюю структуру, взаимодействие со стороны аудитории, целью, функциональными возможностями и т.д. Разные типы сайтов различаются в восприятии пользователей и разные виды продвижения в поисковой системе.

В разработке сайта проанализировали современные технологии веб-дизайна, которые позволяют создать интерактивную веб-страницу. Рассматривались основные правила и принципы работы с HTML-технологиями, CSS-технологиями, JavaScript- технологиями и PHP-технологиями и др.

После проведения теоретического анализа разработана структура сайта, выбраны шаблоны, соответствующие всем требованиям, а также шаблоны страниц.

Так же был протестирован сайт на следующие элементы:

* Тест на вирусы;
* Тест на адаптивность;
* Тест на скорость загрузки сайта.

Таким образом, проведена полная разработка дизайна сайта с добавлением в него авторский дизайнерских решений. Сайт является показательным и должен притягивать интерес пользователей.

Стиль и пользовательский интерфейс сайта не является нагруженным и выдерживает стиль проекта.

Разработка выполнена с учетом цели и поставленных для решения этой цели задач. Задачи, поставленные в начале работы, были выполнены. Цель достигнута.

# **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. 640с.

*Дипломная работа выполнена мною полностью*

*самостоятельно, текст работы выверен,*

*сноски и ссылки проверены*

\_\_\_\_\_\_\_\_\_\_\_\_\_/ Глушко С.С. «\_\_» июня 2023 г.

## Приложение

