МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ   
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 1-40 05 01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Компьютерные языки разметки»

Тема: Web-сайт "Рецензии, обзоры и рекомендации фентези книг"

**Исполнитель**

студент(ка) 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Корело К.А.

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Харланович А. В.

должность, ученая степень, ученое звание подпись, дата

Допущен(а) к защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Харланович А. В.

подпись дата инициалы и фамилия

**Содержание**

[1 Обзор технических методов и программных средств разработки 5](#_Toc103954691)

[**1.1 Обзор аналогичных решений** 5](#_Toc103954692)

[**1.2 Техническое задание** 8](#_Toc103954693)

[**1.3 Постановка задач программного продукта** 9](#_Toc103954694)

[**1.4 Выбор средств реализации программного продукта** 9](#_Toc103954695)

[**1.5 Вывод** 10](#_Toc103954696)

[2. Макетирование страниц веб-сайта 11](#_Toc103954697)

[**2.1 Выбор способа вёрстки** 11](#_Toc103954698)

[**2.2 Разработка прототипа и макета страниц** 12](#_Toc103954699)

[**2.3 Выбор стилевого оформления** 12](#_Toc103954700)

[**2.4 Выбор шрифтового оформления** 12](#_Toc103954701)

[**2.5 Разработка логотипа** 12](#_Toc103954702)

[**2.6 Разработка пользовательских элементов** 13](#_Toc103954703)

[**2.7 Разработка спецэффектов** 14](#_Toc103954704)

[**2.8 Вывод** 15](#_Toc103954705)

[3. Реализация структуры веб-сайта 16](#_Toc103954706)

[**3.1 Структура HTML-документа** 16](#_Toc103954707)

[**3.2 Добавление таблиц стилей CSS** 18](#_Toc103954708)

[**3.3 Использование JavaScript** 19](#_Toc103954709)

[**3.4 Использование стандартов XML (SVG)** 20](#_Toc103954710)

[**3.5 Адаптивный дизайн веб-сайта** 20](#_Toc103954711)

[**3.6 Выводы** 22](#_Toc103954712)

[4. Тестирование веб-сайта 23](#_Toc103954713)

[**4.1 Кроссбраузерность веб-сайта** 23](#_Toc103954714)

[**4.2 Руководство пользователя** 24](#_Toc103954715)

[**4.3 Тестирование ссылок** 25](#_Toc103954716)

[Для начала тестирования мне кажется будет хорошей практикой проверить функционал всех ссылок. По моему мнению сайт довольно хорошо подготовлен к такому роду тестов, так как почти все элементы являются ссылками на страницы их прямого назначения. Единственная неприятная новость — это то что жанры хоть по своей структуре и являются ссылками, но не работают. 25](#_Toc103954717)

[**4.4 Тестирование форм** 25](#_Toc103954718)

[На всех страничка присутствует единственная форма и это форма поиска по сайту рисунок 4.1. После непродолжительных тестов и попыток, что-то найти оказалось, что форма не является рабочей и служит исключительно декоративной цели. 25](#_Toc103954719)

[ 25](#_Toc103954720)

[**4.5 Проверка HTML/CSS** 25](#_Toc103954721)

[**4.6 Выводы** 26](#_Toc103954722)

[Заключение 27](#_Toc103954723)

[**Список использованных литературных источников** 28](#_Toc103954724)

[**Приложение 2 Макет структуры веб-сайта** 30](#_Toc103954725)

[**Приложение 3 Создание бургер меню** 30](#_Toc103954726)

**Введение**

В наше время Интернет - это одно из наиболее важных средств массовой информации и коммуникации, которое находится по своей значимости в одном ряду с прессой, телефоном и телевидением. Любой человек, у которого есть компьютер, может загрузить с его помощью новости, иллюстрации, видео- или аудиоинформацию по любой интересующей его теме. Он может получать эту информацию тогда, когда он этого пожелает.

В наши дни вопрос «Зачем нужен сайт?» задается все реже. И это неудивительно, ведь ответ на него давно уже стал очевидным: сайт нужен, в основном, для того, чтобы помогать его владельцу донести информацию до интернет-пользователей, а также продать им свой товар или услугу.

Тема моей курсовой работы «Создание сайта рецензий, обзоров и рекомендаций фентези книг». Веб-сайт предоставляет информацию о фэнтези книгах, для поверхностного ознакомления с ними, чтобы, придя в магазин, покупатель не столкнулся с чувством разочарования, после её прочтения. Так же у пользователей будет возможность поделиться своими впечатлениями о прочитанном.

Цель курсовой работы: разработать веб-сайт для обзора фэнтези книг с использованием HTML5 и XML, а также с применением Sass/CSS3.

Задачи:

– Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.

– Разработать макет и прототип сайта.

–Разработать структуру веб-сайта.

–Наполнить сайт информацией по теме.

–Протестировать веб-сайт.

–Разработать руководство пользователя.

# 1 Обзор технических методов и программных средств разработки

## **1.1 Обзор аналогичных решений**

В данной сфере существует множество аналогов разного уровня проработки. В пример можно привести существующий сайт для покупки и небольшого ознакомления с книгами «ЛитРес». Данный веб-сайт предполагается как интерактивный веб-ресурс, выполняющий функции продажи книг в разные страны мира.

При обзоре сайта стоит отметить достаточно большие размеры картинок, что даёт покупателю более детально рассмотреть продукт. Что касается меню, оно является читабельным и достаточно удобно расположено на странице. Дизайн представлен на рисунке 1.1.

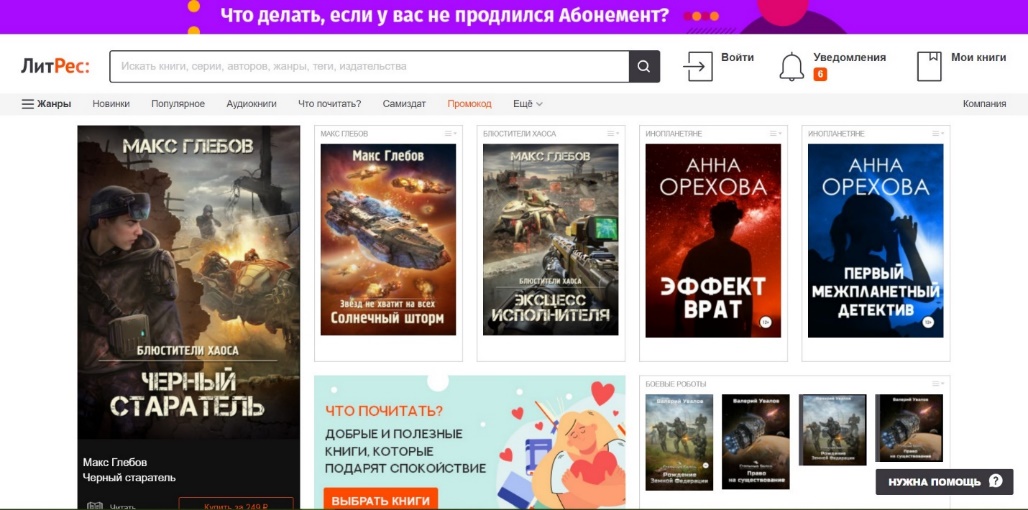


Рисунок 1.1 – Главная страница сайта ЛитРес

Из особенностей сайта, я бы выделила ярко-фиолетовый информационно-рекламный блок сверху страницы, который из-за своей контрастности мешает покупателю с ознакомлением с книгами.

Посмотрев на рисунок 1.2 можно увидеть, что элементы расположены не структурированно, из-за чего пользователю сложно сфокусировать взгляд на определенном объекте. Еще можно заметить насколько маленьким и тусклым является текст в блоках, который не способствует восприятию пользователем и мешает чтению информации.

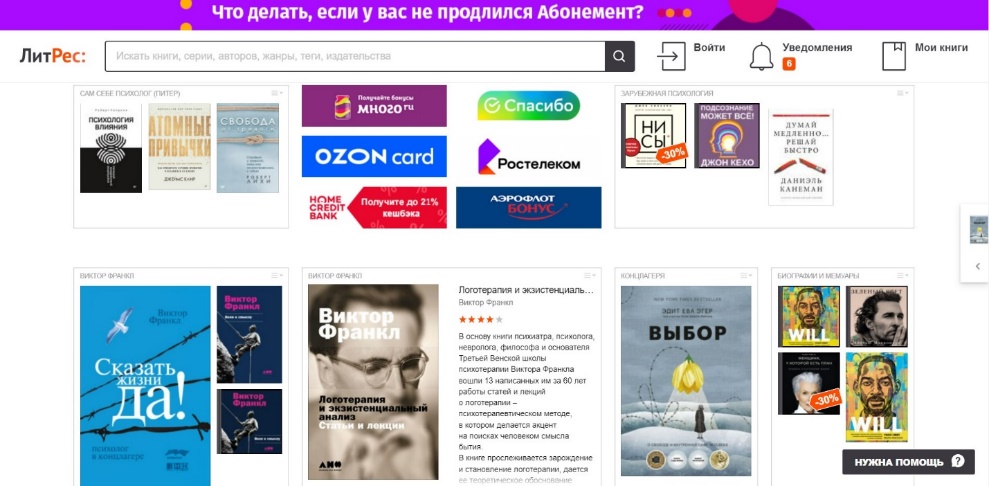


Рисунок 1.2 – Сайт для покупки книг «ЛитРес»

Что касается мобильной версии сайта, она сделана не лучше. На рисунке 1.3 можно увидеть горизонтальное меню, которое не становится выпадающим, а складывается в несколько строк, из-за чего выглядит не презентабельным.



Рисунок 1.3 – Мобильная версия сайта ЛитРес

Следующий сайт «Литнет». На рисунке 1.4 изображена главная страница сайта, Она не имеет большую цветовую палитру, но является очень функциональной и имеет большой шрифт. Попадая на страницу, сразу можно найти книгу по любому названию или выбрать жанр книги, которую хотим прочесть.

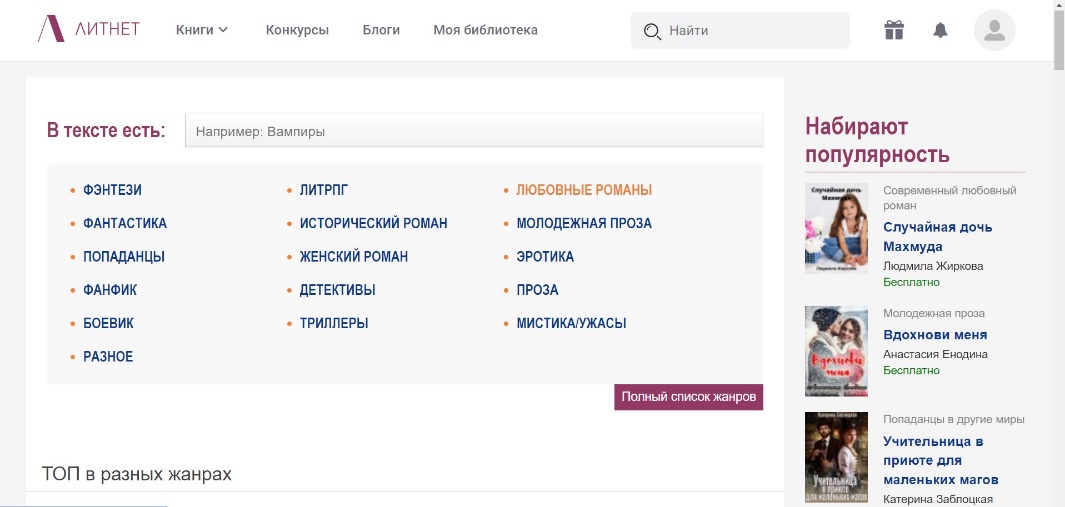


Рисунок 1.4 – Главная страница сайта Литнет

Пролистывая страницу вперёд, можно наткнутся на разные слайдеры, показанные на рисунке 1.5, а справа от них на новостные колонки, которые очень сильно могут помочь при выборе книги.

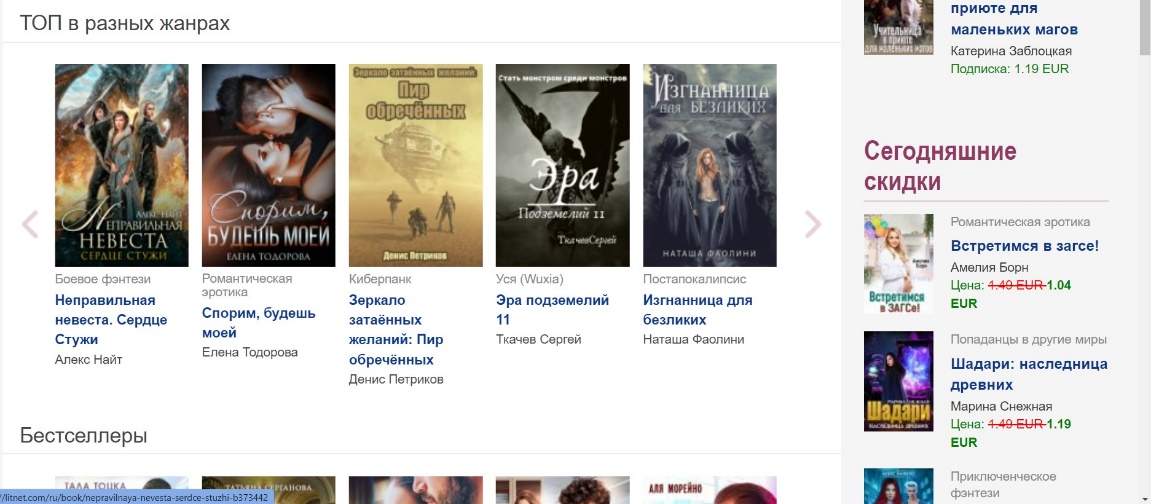


Рисунок 1.5 – Слайдеры сайта Литнет

Хорошо проработана мобильная версия сайта. Она имеет схожий дизайн с ПК версией. Дизайн можно увидеть на рисунке 1.6. Шрифт большой, слайдеры присутствуют, горизонтальное меню и блок с жанрами становится выпадающими.

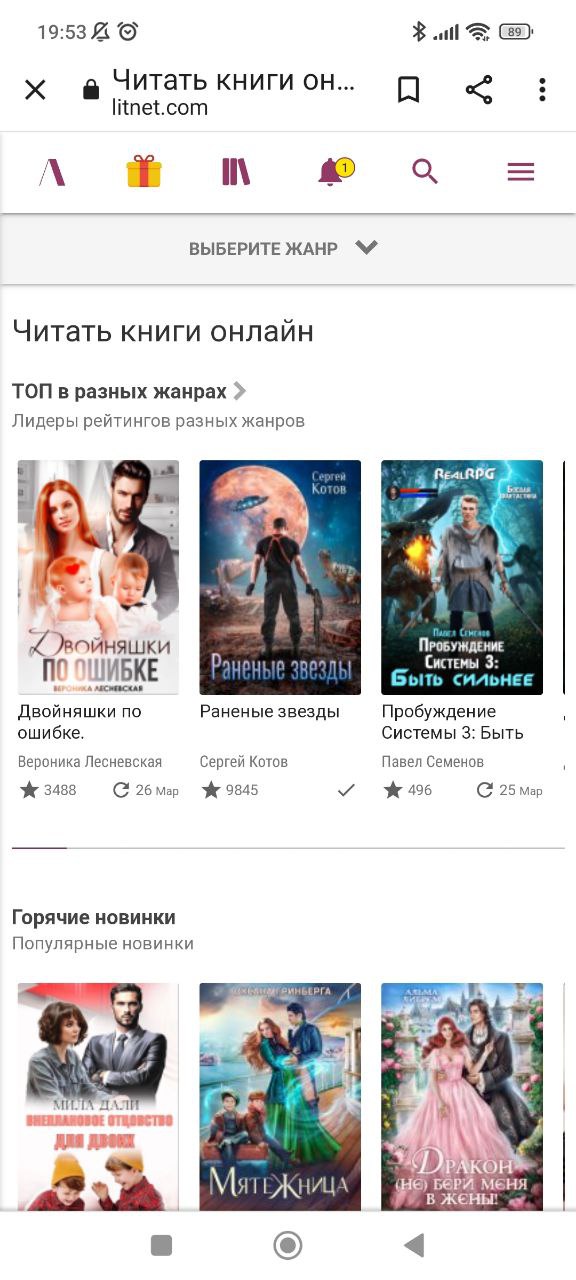


Рисунок 1.6 – Мобильная версия сайта Литнет

Следующий рассматриваемый сайт называется «Libex». Меню сделано очень узким и сливается с другими элементами страницы. Как можно увидеть на рисунке 1.7 шрифт сайта не имеет дизайна, он достаточно мелкий и практически не читабельный.

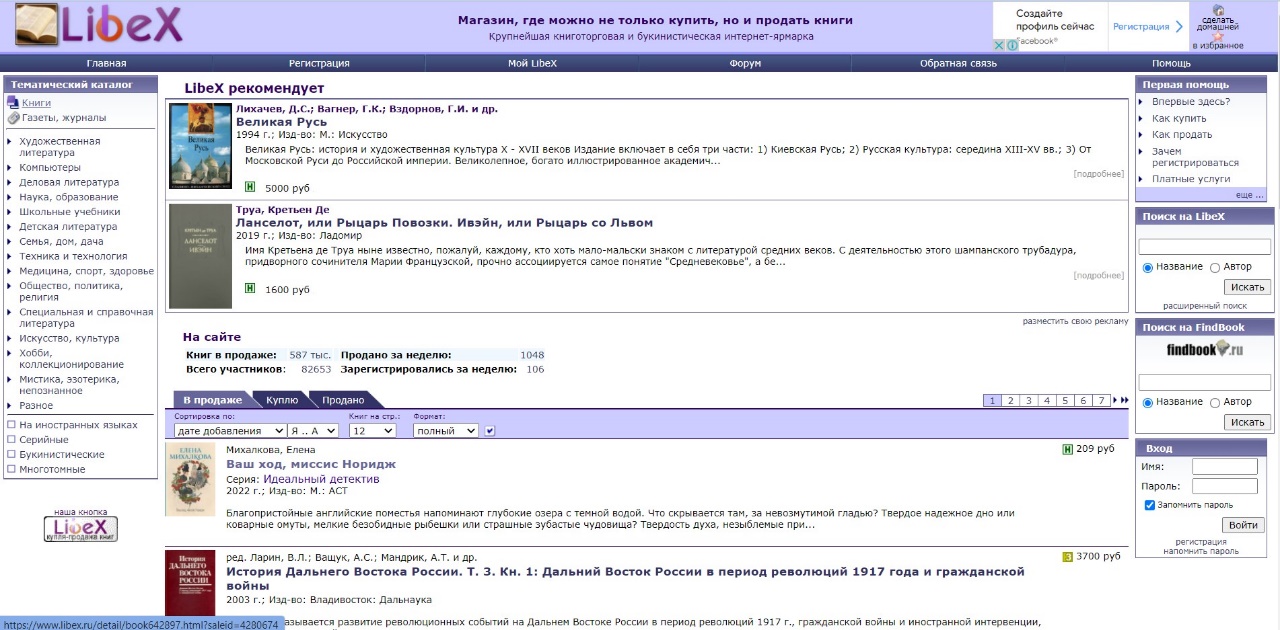


Рисунок 1.7 – Главная страница сайта Libex

В мобильной версии сайта, показанной на рисунке 1.8, полностью отсутствует адаптивность главной страницы.



Рисунок 1.8 – Мобильная версия сайта Libex

## **1.2 Техническое задание**

Основными требованиями к программному продукту являются адаптивность и легкость. Созданный сайт не должен иметь больших программных функций или очень сложных анимаций – все это замедлит скорость загрузки информации и ухудшит ее восприятие. Адаптивность была разработана на основе существующих современных браузеров и устройств.

В данной курсовой работе требуется создать многостраничный сайт для обзора книг в жанре фэнтези. Сайт должен содержать такую информацию как описание сайта, каталог книг, рецензии и отзывы на эти книги. Так же структура сайта должна предоставлять пользователю интуитивный короткий путь до необходимой страницы.

Главной задачей сайта является предоставление информации о фентези книгах для привлечения внимания и расширения кругозора клиентов. Сайт должен быть интерактивными содержать актуальную информацию. Для решения задачи всё элементы должны быть структурированы при представлении на странице.

## **1.3 Постановка задач программного продукта**

Основными задачами курсового проекта являются:

– разработка многостраничного сайта с использованием современных программных средств

– создание удобного интерфейса для возможности комфортного пребывания пользователя на сайте

– создание дизайна, делающего сайт более привлекательным для клиентов

– создание уникального логотип;

– разработка адаптивности для веб-страниц.

На главной странице меню должно располагаться сверху. Там будут предоставляется ссылки на основную информацию (жанры, новости, поиск книг, оставление отзывов).

В мобильной версии страница должна адаптироваться под ширину устройства (ширина до 560px).

Так же необходимо реализовать макеты страниц сайта для главной страницы, страницы для жанров, страницы для просмотра конкретной книги и страницы для оставления отзыва.

## **1.4 Выбор средств реализации программного продукта**

Сайт разрабатывался в текстовом редакторе Visual Studio Code. Страницы сайта могут просматриваться в таких браузерах как Internet Explorer, Microsoft Edge, Safari, Opera, Mozilla Firefox, Google Chrome.

Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

Для реализации проекта были выбраны языки: HTML, CSS/Sass, XML, JavaScript. Структура сайта создана с помощью языка разметки HTML, дизайн сайта оформлен с помощью CSS/Sass. В XML формате хранятся данные сайта. JavaScript позволит добавить динамичности сайта.

CSS (Cascading Style Sheets, каскадные таблицы стилей) – это язык описания внешнего вида документа, написанного с использованием языка разметки. CSS позволяет представлять один и тот же документ в различных стилях. Основной целью разработки CSS являлось разделение 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.

## **1.5 Вывод**

В данном разделе рассмотрены аналогичные решения, проанализированы их недостатки и преимущества, предназначенные для последующего создания качественного сайта. Были поставлены конкретные задачи по созданию интерактивного информационного ресурса. Кроме того, были рассмотрены языки разметки, а также инструменты для реализации программного обеспечения, такие как редактор кода Visual Studio Code. Проанализированы особенности языка и основные дополнительные средства редактора. Все вышеперечисленные пункты будут необходимы для достижения поставленной цели и создания качественного продукта в будущем.

# 2. Макетирование страниц веб-сайта

## **2.1 Выбор способа вёрстки**

В соответствие с требованиями к проекту, при разработке веб-сайта будет использоваться язык разметки HTML, который структурирует и отображает контент сайта с помощью параграфов, блоков, списков, таблиц. Для придания стилей элементам языка гипертекстовой разметки HTML (HyperText Markup Language) будет использован язык стилей CSS. Для удобного написания разметки был использован редактор кода Visual Studio Code, а также плагины для него: Emmet (плагин, включающий в себя горячие клавиши для упрощения написания кода), Bracket Pair Colorizer (плагин для подсвечивания парных тегов), CSS Formater (плагин для форматирования CSS и SCSS кодов). Для придания динамичности сайту будет использован мультипарадигменный язык программирования JavaScript.

Для создания адаптивности и кроссбраузерности сайта, было принято решение об использовании блочной верстки сайта.

Основным элементом, применяемым в блочной верстке, является тег <div>. Участок кода, отделенный этим тегом, называется слоем. Все стилевые решения вынесены за границы кода html в каскадные таблицы стилей. Доступ к ним осуществляется через идентификаторы или классы css.

Блочная верстка – в данный момент является основным способом верстки. В отличие от табличной блочная верстка обладает рядом преимуществ:

– Отделение стиля элементов от кода html;

– Возможность наложения одного слоя на другой – такая возможность во многом облегчает позиционирование элементов.

– Лучшая индексация поисковиками;

– Высокая скорость загрузки страницы, состоящей от взаимно независимых элементов;

– Легкость создания визуальных эффектов (выпадающих меню, списков, всплывающих подсказок).

Единственный ощутимый минус блочной вёрстки — сделанные на ней сайты могут по-разному отображаться в обозревателях. Чтобы этого избежать, нужно делать вёрстку «кроссбраузерной», то есть одинаково отображаемой любым обозревателем.

Поэтому было решено использовать медиа-запросы для одинакового отображения сайта в различных обозревателях. Медиа-запросы – это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств. Иными словами, это конструкции, которые позволяют определять на основании некоторых условий какие стили необходимо использовать на веб-странице, а какие нет.

## **2.2 Разработка прототипа и макета страниц**

**Прототип**— это базовый макет сайта, который визуализирует расположение всех элементов и функций. Он позволяет наглядно проиллюстрировать все задумки, а также внести правки ценой минимальных усилий и расходов. Поэтому прототип был создан в первую очередь. Полный прототип сайта можно увидеть в приложении А.

Следующим шагом стало создание основного макета структуры сайта. Макеты сайта можно увидеть в приложении Б

## **2.3 Выбор стилевого оформления**

В настоящее время не существует единой классификации стилей web-дизайна. И это неудивительно: каждый ресурс является уникальным по многим параметрам, а стилевое оформление сайта не является точной наукой и не имеет четко закрепленных правил.

Для заинтересованности пользователя при работе с сайтом необходимо создать удобный и понятный интерфейс. Потому было решено использовать следующую цветовую палитру, показанную на рисунке 2.1. Такой стиль не будет слишком выделятся, но и не будет мрачным и серым, что позволит привлечь достаточное внимание к фотографиям книг на сайте.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Рисунок 2.1 – Основная цветовая палитра сайта

## **2.4 Выбор шрифтового оформления**

В данном проекте будет использоваться шрифт Cambria. Cambria была разработана для чтения на экране и хорошо смотрится при печати в небольших размерах. Он имеет очень ровное расстояние и пропорции. Диагональные и вертикальные линии и шерифы относительно сильны, в то время как горизонтальные ряды невелики и предназначены для того, чтобы подчеркнуть окончания штрихов, а не выделяться сами по себе. Этот принцип не заметен в курсиве, где строчные символы приглушены по стилю, чтобы быть в лучшем виде в качестве элементов словесных образов.

## **2.5 Разработка логотипа**

Разработка логотипа осуществлялась в графическом редакторе Adobe Photoshop.

Adobe Photoshop – многофункциональный редактор для работы с фото и видеофайлами (растровые изображения и несколько векторных инструментов). Программный продукт работает на ПК с операционными системами macOS, Windows и в мобильных версиях iOS, Windows Phone и Android.  Adobe Photoshop – необычный редактор, в котором можно не только корректировать и ретушировать изображения, но и преобразовывать формы, сохранять GIF-анимации, использовать слои и эффекты. Эти функции позволяют легко трансформировать, клонировать, улучшать изображения, применять различные фильтры для создания красочной, продающей картинки.

В дизайне хотелось что-нибудь необычного, но и простого. Поэтому за основу было взято слово «book», что переводится как «книга». К этому слову были подобраны несколько ассоциаций таких как: дом, парк, скамейка. А потом они были соединены вместе. Результат данного слияния можно увидеть на картинке 2.2.

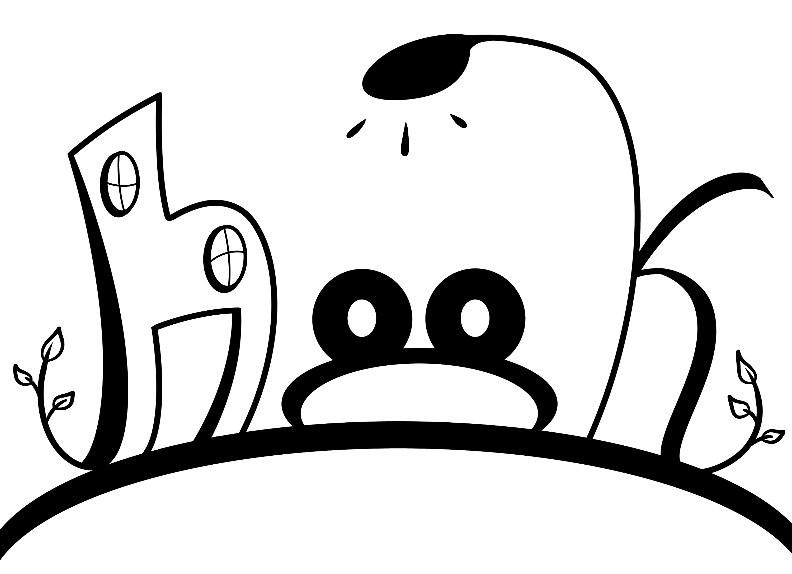


Рисунок 2.2 – Дизайн логотипа

Было принято решение не раскрашивать данный логотип в другие цвета для лучшего отображения на странице. Затем данная картинка была скачаны в pdf формате. А после этого, используя сайт image.online-convert.com, конвертирована в svg формат.

## **2.6 Разработка пользовательских элементов**

При оформлении необходимо учитывать размер графических файлов, грамотность пропорций и перспектив по объёму. Пользователь должен легко понимать структуру сайта по его меню, а инструменты навигации сайта должны быть понятны. Хорошая навигация учитывает не только разделы сайта, но и удобство работы с каждой его страницей.

В проекте будут представлены такие элементы пользовательского интерфейса, как навигационное меню, поле для поиска по сайту и слайдер.

Навигацию по сайту можно осуществлять с помощью шапки сайта. Там можно найти небольшой каталог жанров и кнопку для того, чтобы оставить отзыв о сайте. Эти элементы нужны для того, чтобы сайт стал удобным и понятным для пользователя. Элемент навигацию по сайту представлен на рисунке 2.3.



Рисунок 2.3 – Пример дизайна шапки сайта

Чтобы облегчить задачу пользователя по поиску нужной ему книги, на панель навигации был добавлен элемент поля поиска. Элемент поля поиска представлен на рисунке 2.4.

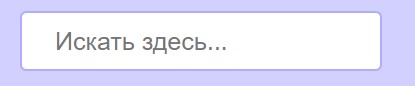


Рисунок 2.4 – Пример дизайна поисковой строки

Сейчас практически все предпочитают графический вид информации текстовой. Интернет-пользователи хотят получать максимум полезного контента за минимум времени.

Слайдер представленный на рисунке 2.5 позволит пользователю просмотреть важную информацию за короткое время и если она его заинтересует, то нажав на соответствующую картинку, пользователь сможет изучить данный контент более подробно.

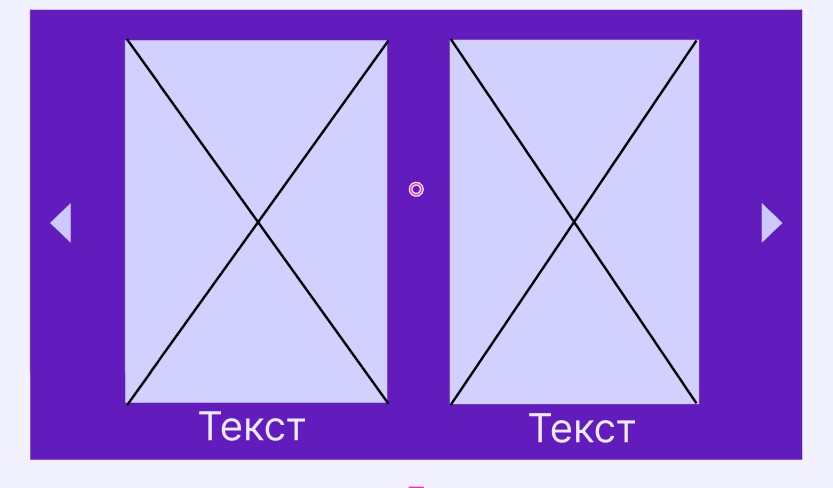


Рисунок 2.5 – Пример дизайна слайдера

## **2.7 Разработка спецэффектов**

Для придания сайту интерактивности и привлекательности на нем будут разработаны некоторые динамические эффекты и анимации.

На главной странице, показанной на рисунке 2.6, будет реализована анимация изменения цвета выбранного элемента меню при наведении курсора, а также изменение цвета выбранного элемента в каталоге жанров.

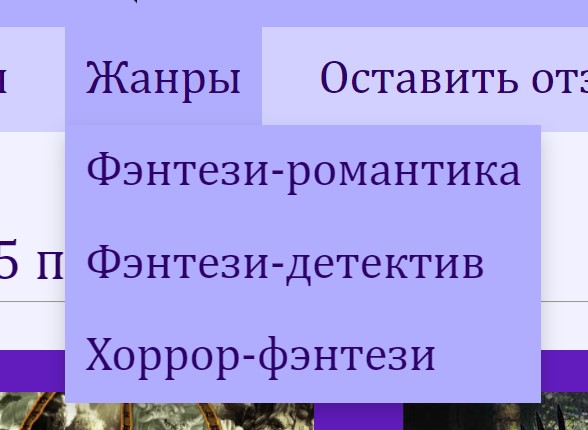


Рисунок 2.6 – изменение выбранного элемента меню при наведении курсора

Темой курсового проекта была разработка web-сайта для обзоров, рецензий и рекомендаций фэнтези книг, следовательно, был реализован элемент оценки книги в виде пяти звёзд и рядом рейтинг книги в виде числа, который можно увидеть на рисунке 2.7



Рисунок 2.7 – Элемент оценки книги в виде пяти звёзд

## **2.8 Вывод**

В данном разделе был создан прототип сайта, чтобы помочь визуализировать размещение элементов, расположение графики, показать его функциональность.

Далее были созданы макеты страниц сайта, чтобы показать, как будет выглядеть сайт с добавлением цветов и изображений. На этом этапе была подобрана цветовая палитра, шрифты, создан логотип и оформление отдельных элементов сайта.

Был определён способ верстки сайта, а также его стилевое оформление, элементы пользовательского интерфейса и эффекты с переходами, добавляющие динамики страницам.

# 3. Реализация структуры веб-сайта

Сайт разрабатывался в текстовом редакторе Visual Studio Code с помощью плагинов: Emmet (плагин, включающий в себя горячие клавиши для упрощения написания кода), Bracket Pair Colorizer (плагин для подсвечивания парных тегов), CSS Formater (плагин для форматирования CSS и SCSS кодов). Так же приложение имеет инструменты для работы с [Git](https://ru.wikipedia.org/wiki/Git" \o "Git) и возможность использовать несколько панелей, что делает его удобным для написания кода.

## **3.1 Структура HTML-документа**

При написании структуры документа был использован язык разметки HTML5. Он является стандартом для построения структуры сайтов и включает в себя теги, которые есть в любом HTML-файле. Эти теги и служебная информация нужны браузеру для корректного отображения информации.

Моя структура основного документа состоит из шести блоков <div>, два из которых повторяются.

Первый блок отвечает за отображение логотипа и приветственного текста.

Во втором блоке было реализовано навигационное меню, в котором присутствует выпадающая панель жанров, реализация которой представлена в листинге 3.1, созданная с помощью тега <button> и тегов для ссылок <a>.

|  |
| --- |
| <div class="dropdown">  <button class="dropbtn">Жанры</button>  <div class="dropdown-content">  <a href="Janre-1R.html">Фэнтези-романтика</a>  <a href="#">Фэнтези-детектив</a>  <a href="#">Хорор-фэнтези</a>  </div>  </div> |

Листинг 3.1 – Выпадающая панель жанров

Первых два блока присутствуют на всех страницах и являются абсолютно одинаковыми.

Третий блок отвечает за расположение текста для слайдера.

В четвёртом блоке был сделан сам слайдер так, как показано на листинге 3.2

|  |
| --- |
| <div class="scrollmenu">  <a href=""><img class="scrol\_photo" src="Новая папка/blast.jpg">  <figcaption>Властелин колец</figcaption>  </a>  <a href=""><img class="scrol\_photo" src="Новая папка/igraprest.jpg">  <figcaption>Игра престолов</figcaption>  </a>  <a href=""><img class="scrol\_photo" src="Новая папка/xobbit.jpg">  <figcaption>Хоббит</figcaption>  </a>  <a href=""><img class="scrol\_photo" src="Новая папка/xodzamok.jpg">  <figcaption>Ходячий замок</figcaption>  </a>  <a href=""><img class="scrol\_photo" src="Новая папка/labir.jpg">  <figcaption>Лабиринт эхо</figcaption>  </a>  </div> |

Листинг 3.2 – Реализация слайдера

Пятый и шестой блок — это повторение третьего и четвёртого блока, но уже с другой информацией.

Структура остальных страниц очень похожа на первую, отличие заключается в том, что в блоках <div> будет размещены другие элементы с информацией.

На странице жанров будет присутствовать одинаковые блоки, в которых будет находится информация о книгах в выбранном жанре.

Страница отзывов будет содержать в блоке форму такую, как в листинге 3.3, для написания комментария о сайте.

|  |
| --- |
| <form name="data">  <fieldset>  <legend><span>Отзыв</span></legend>  <div>  <span class="text">  <br>Имя:<br><input type='text' name='firstname' class="text"/><br><br>  Фамилия:<br><input type='text' name='lastname' class="text"/><br><br>  </div>  <span class="text">Ваш email:</span>  <div>  <input type='email' name='email' class="text"/><br><br>  </div>  <br>  <span class="text">Напечатайте отзыв:</span>  <div>  <textarea rows="4" class="text"></textarea>  <br><br>  </div>  <input class="text" type="button" value="отправить" onClick="fun1()">  </fieldset>  </form> |

Листинг 3.3 – Реализация формы для написания отзыва на сайте

А страница для книги будет иметь 3 блока, в одном прописана информация о книге, во втором аннотация на неё, и в третьем блоке можно будет оставить комментарий.

## **3.2 Добавление таблиц стилей CSS**

Для создания функционального и визуально приятного для пользователя сайта написание HTML-кода мало. Поэтому было решено использовать язык стилей CSS.

CSS (Cascading Style Sheets) — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML), чтобы задавать визуальное оформление web-страницы. При выполнении разметки элементов языком HTML документ получит основную структуру страницы. Это заголовки различного уровня, параграфы, маркированные и нумерованные списки, отступы, картинки и прочее.

Отделяя стиль представления документов от содержимого документов, CSS упрощает создание веб-страниц и обслуживание сайтов.

Стили CSS были созданы как отдельные файлы и подключены в описываемый ими веб-документ с помощью элемента <link>, включённого в элемент <head> как показано в листинге 3.4.

|  |
| --- |
| <head>  <link rel="stylesheet" href="style.css" />  <link rel="stylesheet" href="search.css" />  <link rel="stylesheet" href="scrollmenu.css" />  <link rel="stylesheet" href="janre.css" />  <link rel="stylesheet" href="booc-paje.css" />  </head> |

Листинг 3.4 – Ссылки на CSS файлы проекта

Применение стилей CSS позволило сделать навигационной меню визуально приятным и понятным. С его помощью была реализована анимация изменения цвета выбранного элемента меню при наведении курсора, а также изменение цвета выбранного элемента в каталоге жанров, который можно увидеть в листинге 3.5.

|  |
| --- |
| .topnav {  overflow: hidden;  background-color: #D2D0FF;  }  .dropdown .dropbtn {  font-size: 38px;  border: none;  outline: none;  color: #2d0168;  padding: 14px 16px;  background-color: inherit;  font-family: inherit;  margin: 0;  height: 79px;  }  .dropdown-content {  display: none;  position: absolute;  background-color: #B0ADFE;  min-width: 20%;  box-shadow: 0px 8px 16px 0px rgba(0, 0, 0, 0.2);  z-index: 1;  } |

Листинг 3.5 – Пример применения обнуляющего селектора

Добавление таблиц стилей CSS позволило упростить создание стилистического оформления страницы. Стили являются удобным, практичным и эффективным инструментом при вёрстке веб-страниц и оформления текста, ссылок, изображений и других элементов.

С помощью CSS-свойства font-family в листинге 3.6 был определён приоритетный список из одного или нескольких названий семейства шрифтов и/или общее имя шрифта для выбранного элемента. А с помощью атрибута src, который указывает адрес файла (URL), можно будет загрузить любой понравившийся вам шрифт во фрейм.

|  |
| --- |
| body {  font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;  } |

Листинг 3.6 – Определение шрифтов, которые будут использоваться на сайте

## **3.3 Использование JavaScript**

JavaScript— мультипарадигменный язык программирования. 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.

В данном курсовом проекте JavaScript был использован в виде функции для реализации адаптивного меню при уменьшении экрана. Код реализован в листинге 3.7

|  |
| --- |
| <script>  function myFunction() {  var x = document.getElementById("myTopnav");  if (x.className === "topnav") {  x.className += " responsive";  } else {  x.className = "topnav";  }  }  </script> |

Листинг 3.7 – 1-й пример использования JavaScript

В листинге 3.8 реализован вызов всплывающего окна при нажатии на кнопку «отправить» на странице написания отзыва о сайте.

|  |
| --- |
| <script>  function fun1() {  if (confirm("Отправить отзыв?"))  alert("Ваш отзыв отправлен");}  </script> |

Листинг 3.8 – 2-й пример использования JavaScript

## **3.4 Использование стандартов XML (SVG)**

SVG — XML язык разметки на основе векторной графики. По сути это текстовый файл, который является открытым веб-стандартом для описания двухмерных векторных изображений без потери качества при масштабировании. Также поддерживает интерактивность и анимацию. Разрабатывается с 1999 года и используется для того, чтобы описать с помощью языка программирования графические элементы сайта.

В моем сайте SVG использовалось для логотипа. Логотип был создан через графический онлайн-редактор Figma. Он был первоначально сохранен в формате PDF, а позже преобразован в формат SVG. Пример использования SVG продемонстрировано в листинге 3.9.

|  |  |
| --- | --- |
| <object type="image/svg+xml" data="Новая папка/log.svg" class="logo1"></object> |  |

Листинг 3.9 – Пример использования SVG

## **3.5 Адаптивный дизайн веб-сайта**

В наше время почти каждый человек на земле имеет свой смартфон-этакий маленький компьютер, который всегда можно взять с собой. Следовательно, большинство пользователей предпочитают использовать для поиска информации именно их, а не другие устройства. Из-за чего появилась необходимость делать сайты удобными не только для пользователей компьютеров, но и для пользователей мобильных устройств.

И на помощь к нам приходит адаптивный дизайн. Реализовать его можно разными способами, например, с помощью flexbox-вёрстки или медиа-запросов.

В данной работе я использовала медиа-запросы. Это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств. Иными словами, это конструкции, которые позволяют определять на основании некоторых условий какие стили необходимо использовать на веб-странице, а какие нет. Поэтому медиа-запросы являются самым лёгким и удобным способам сделать сайт подходящим под любые типы устройств.

При создании адаптивных веб-страниц также необходимо обратить внимание на метатег viewport. Данный тег обеспечивает корректное отображение адаптивных дизайнов сайтов на экранах устройств, имеющих высокую плотность пикселей. Иными словами, он устанавливает соответствие между CSS и физическим разрешением веб-страницы.

Подключение метатега viewport к странице приведено в листинге 3.10.

|  |
| --- |
| <meta name="viewport" content="width=device-width, initial-scale=1"> |

Листинг 3.10 – Подключение метатега viewport к странице

Создание медиа-запроса начинается с ключевого слова @media после которого указывается одно или несколько условий. В качестве условия можно указывать тип устройства или требования к определённой характеристике. Требование к определённой характеристике записывается в круглых скобках, как показано в листинге 3.11.

|  |
| --- |
| @media (max-width: 540px) {  . photo2 {  float: none;  margin-left: 5%;  }  } |

Листинг 3.11 – Пример написания медиа-запроса

Медиа-запрос состоит из необязательного типа носителя и/или ряда мультимедийных функций.

Типы устройств:

1) all (подходит для всех устройств)

2) print (Предназначен для материалов, загружаемых по страницам, и для документов, просматриваемых на экране в режиме предварительного просмотра)

3) screen (предназначен в первую очередь для цветных компьютерных экранов)

4) speech (Предназначен для синтезаторов речи)

Было использовано два разрешения после которых применялись новые свойства к элементам: 540 px и 1000 px. 1000 px и более — это разрешение для персональных компьютеров. 540 px — разрешение для мобильных устройств. В листинге 3.12 показано как применялись медиа запросы и тип устройства screen для создания выпадающего навигационного меню.

|  |
| --- |
| .topnav a {  float: left;  display: block;  }  .topnav .icon {  display: none;  }  .dropdown-content {  display: none;  }  .dropdown-content a {  display: block;  }  .dropdown:hover .dropdown-content {  display: block;  }  @media screen and (max-width: 540px) {  .topnav.responsive a {  float: none;  display: block;  text-align: left;  }  .topnav.responsive .dropdown .dropbtn {  display: block;  }  .topnav a:not(:first-child),  .dropdown .dropbtn {  display: none;  }  .topnav a.icon {  float: right;  display: block;  }  } |

Листинг 3.12 – Пример использования медиа-запросов для создания выпадающего навигационного меню

## **3.6 Выводы**

В этом разделе было рассказано про структуру HTML-документа, а также показано как она была реализована в данной курсовой работе. Какие теги были использованы и для чего они предназначены. Также было рассказано про каскадные таблицы стилей CSS, с помощью которых мы изменяли внешний вид нашего сайта. И не забудем про мультипарадигменный язык программирования JavaScript, благодаря которому мы добавили динамичности нашим страницам. Было рассказано про создание адаптивных веб-страниц с помощью медиа-запросов, с помощью чего и как они работают, и на что стоит обращать внимание при написании кода с помощью медиа-запросов.

Было определено стилевое оформление веб-сайта, созданы анимации, а также веб-сайт был подготовлен к следующему этапу – к тестированию.

# 4. Тестирование веб-сайта

## **4.1 Кроссбраузерность веб-сайта**

Существует множество различных браузеров с миллионами пользователей у каждого из них. Поэтому при создании веб-сайта или веб-приложения стоит проверить его совместимость со всеми имеющимися браузерами. При этом лучший и самый удобный способ сделать это — использовать кроссбраузерные инструменты проверки.

Кроссбраузерность – это способность веб-ресурса отображаться одинаково хорошо во всех популярных браузерах без перебоев в функционировании и ошибок в верстке, с одинаково корректной читабельностью контента. Это очень важный показатель как для поисковых систем, так и пользовательской аудитории.

Поэтому было решено проверить сайт на кроссбраузерность с помощью таких браузеров как Google Chrome, Microsoft Edge и Internet Explorer.

При проверке было выявлено две ошибки. Первая заключалась в медиа-запросах, которые были не до конца подключены. На рисунке 4.1 можно увидеть неправильное отображения сайта.

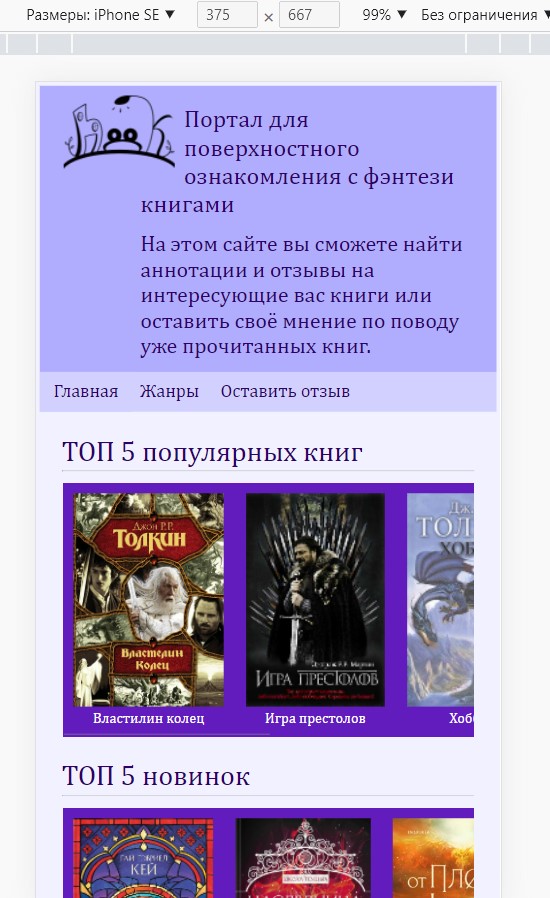


Рисунок 4.1 – Неправильное отображение сайта при подключении медиа-запросов

Ошибкой был следующий тег, который не был добавлен в код: <meta name="viewport" content="width=device-width, initial-scale=1">. Данный тег обеспечивает корректное отображение адаптивных дизайнов сайтов на экранах устройств, имеющих высокую плотность пикселей.

Второй ошибкой был ещё один тег, который не был добавлен на страницу при написании кода: <meta charset="utf-8">. Данный тег указывает кодировку документа, чтобы браузер мог правильно отображать текст на странице потому на картинке 4.2 можно увидеть неправильное отображение символов в браузере Internet Explorer.

## **4.2 Руководство пользователя**

Данный веб-сайт предоставляет возможность просматривать фильмы пользователю.

Для обеспечения надежности данного веб-сайта реализовано:

1. Разработан удобный пользовательский интерфейс, чтобы даже человек, не работавший с компьютером достаточно, быстро мог разобраться в этом веб-сайте.
2. Вверху главной страницы, вы можете увидеть навигационную панель. Она содержит основные тематические разделы веб-сайта: “Фильмы”, “Сериалы”.
3. Во вкладке “Фильмы” и” Сериалы” пользователь может изучить всё содержимое данного сайта.

## **4.3 Тестирование ссылок**

Для начала тестирования мне кажется будет хорошей практикой проверить функционал всех ссылок. По моему мнению сайт довольно хорошо подготовлен к такому роду тестов, так как почти все элементы являются ссылками на страницы их прямого назначения. Единственная неприятная новость — это то что жанры хоть по своей структуре и являются ссылками, но не работают.

## **4.4 Тестирование форм**

На всех страничка присутствует единственная форма и это форма поиска по сайту рисунок 4.1. После непродолжительных тестов и попыток, что-то найти оказалось, что форма не является рабочей и служит исключительно декоративной цели.



Рисунок 4.1 – Поисковая форма

Данный факт весьма снижает ценность ,качество ,а также желание пользователей посещать, а также пользоваться сайтом где не доработана функция поиска фильмов.

## **4.5 Проверка HTML/CSS**

Последняя проверка, но не менее важной является проверка синтаксиса как HTML структуры, так и CSS формы на наличие устаревших или бесполезный тегов, или CSS параметров. По итогу исследования данных структур не было выявлено никаких устаревших тегов, ошибочных параметров, бесполезных свойств и т.д.

Также для проверки кода на какие-либо ошибки было принято решение провести внешнюю проверку, а конкретнее на специализированных на данной тематике сайтах по типу csslint.net данный ресурс можно также увидеть в списке литературы [6]. Результат такой проверки стал весьма приятным, его можно увидеть на рисунке 4.2.

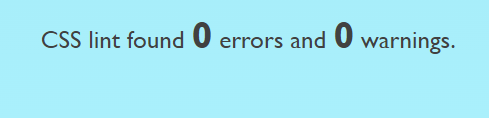


Рисунок 4.2 – Результат про верки на csslint.net

Таким образом можно сказать, что код является дееспособным.

## **4.6 Выводы**

На данном этапе было проанализировано поведение веб-сайта при использовании с разных устройств, а также веб-ресурс был протестирован в большом количестве браузеров. Также было добавлено руководство пользователя с прикрепленной план-схемой. По итогу выполнения данного этапа для веб-сайта был разработан адаптивный дизайн и кроссбраузерность. Работа, проделанная с сайтом, позволяет пользователям ориентироваться по сайту. Лишних элементов нет, что упрощает пользование сайтом. При пользовании сайтом при помощи мобильного устройства некоторые элементы меняют свое положение, для более удобного пользования.

# Заключение

Целью курсовой работы была разработать веб-сайт, который бы предоставлял информацию о сериалах. Сайт предоставляет возможность ознакомиться с интересующим тебя сериалом, а также почитать актуальные новости из мира кино.

Для создания макетов и прототипов сайта использовалась программа «Figma». Figma — онлайн-сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

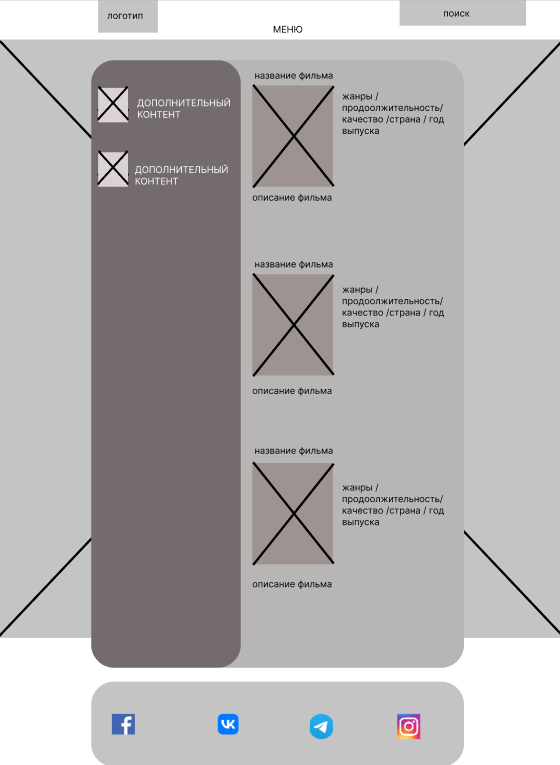
Результатом курсового проекта стал готовый к работе веб-сайт, в котором размещена информация о некоторых сериалах. Конечно, в дальнейшем возможно расширение списка сериалов, а также объём информации который пользователь сможет получить после посещение данного сайта.

**Список использованных литературных источников**

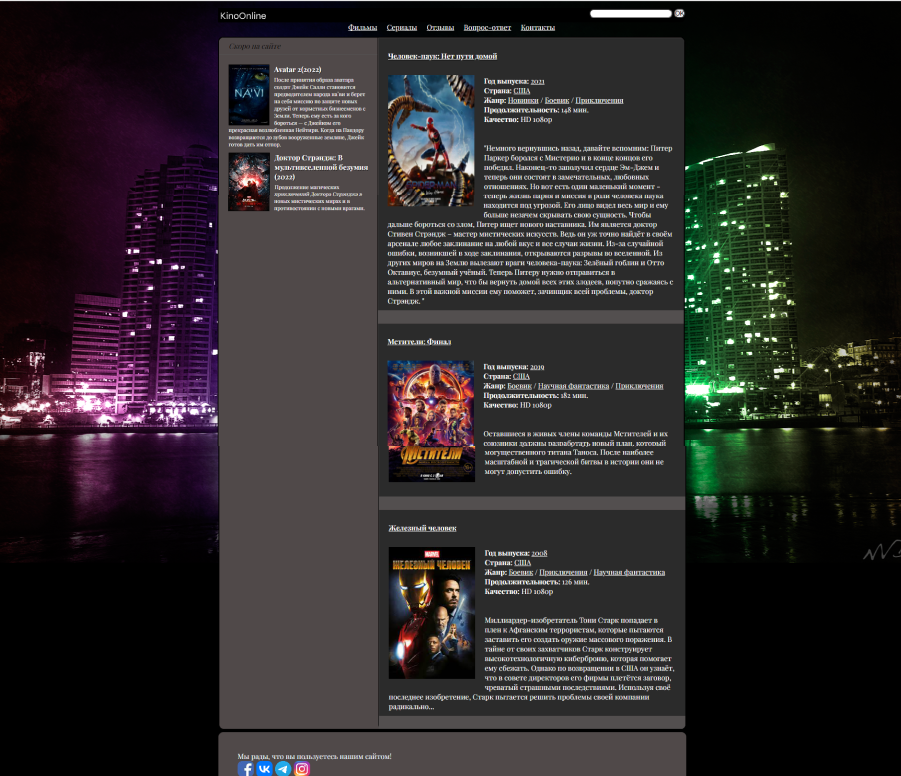
1. Документация по JavaScript/HTML [Электронный ресурс] / Справочный Режим доступа : <https://developer.mozilla.org> – Дата доступа 02.05.2022.
2. Документация по Figma [Электронный ресурс] / Справочный Режим доступа : <https://help.figma.com> – Дата доступа 02.05.2022.
3. Аналоговый ресурс использованный при разработке [Электронный ресурс] / Режим доступа: <https://kinogo.biz/> – Дата доступа 15.05.2022.
4. Аналоговый ресурс использованный при разработке [Электронный ресурс] / Режим доступа: <https://hd.kinopoisk.ru>– Дата доступа 15.05.2022.
5. Аналоговый ресурс использованный при разработке [Электронный ресурс] / Режим доступа: <https://megogo.net>– Дата доступа 15.05.2022.
6. Аналоговый ресурс использованный при проверке СSS кода [Электронный ресурс] / Режим доступа: <https://csslint.net>– Дата доступа 15.05.2022.

**Приложение**

**Приложение 1 Прототипы веб-страниц**

****

**Приложение 2 Макет структуры веб-сайта**



**Приложение 3 Создание бургер меню**

|  |  |
| --- | --- |
| До активации меню | После активации |

Элементы листинга:

|  |
| --- |
| .nav {      font-size: 30px;      position: relative;    }    [for=burger]{      display: block;      color: aliceblue;      position: relative;      text-align: center;      width: 30px;      font-size: 80px;    }    nav{      display: none;    }    #burger:checked + label + nav{      display: block;    } |