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 «Разработка структуры веб-сайта для интернет-магазина продажи курсов»

**Исполнитель**

студент(ка) 1 курса 3 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. Д. Козловский

подпись, дата

**Руководитель**

ассистент \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Комарова

должность, ученая степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Комарова

подпись дата инициалы и фамилия

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра информационных систем и технологий

УтверЖдаю

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_ В.В. Смелов

подпись инициалы и фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

**ЗАДАНИЕ**

**на курсовое проектирование**

по дисциплине «Компьютерные языки разметки»

Курс \_\_\_ Группа: \_\_\_\_ Специальность: 1-98 01 03 «Программное обеспечение информационной безопасности мобильных систем» **или** 1-40 05 01 «Информационные системы и технологии» **или** 1-40 01 01 «Программное обеспечение информационных технологий»

Студент: Абрамович Артём Александрович

**1. Тема:** Web-сайт интернет магазина продажи курсов

**2. Срок выполнения курсового проекта** : с 14 февраля 2022 г. по 10 мая 2022 г.

**3. Технические требования :**

3.1 Прототип веб-сайта должен быть разработан с использованием графических редакторов Figma/Adobe XD/Sketch.

3.2 Для хранения данных должен быть использован XML-формат.

3.3 Разметка содержания сайта должна быть выполнена с применением HTML5 и XML.

3.4 Для описания внешнего вида веб-страниц использовать Sass/CSS3.

3.5 Веб-сайт должен содержать:

– семантические теги HTML5;

– графические элементы в форме SVG;

– несколько веб-страниц;

3.6 Верстка сайта должна быть адаптивной и кроссбраузерной;

3.7 Для тестирования использовать Linter.

3.8 Проект и пояснения к проекту должны быть размещены на GitHub.

**4. Содержание пояснительной записки**

1. Титульный лист;

2. Задание на курсовое проектирование;

3. Введение;

4. Постановка задачи;

5. Проектирование веб-сайта

6. Реализация структуры веб-сайта

7. Тестирование веб-сайта

8. Заключение

9. Список использованных литературных источников

10. Приложения (полный исходный текст программы разработанного приложения с подробными комментариями)

**5. Форма представления на GitHub** **выполненного курсового проекта:**

– Теоретическая часть пояснительной записки курсового проекта должна быть представлена в формате MS Word.

– Оформление записки должно быть согласно правилам.

– Листинги представляются в приложении.

#### **Календарный план**

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование этапов курсового проекта | Срок выполнения этапов проекта |
| 1 | Задание на курсовое проектирование. Титульный лист | 14.02.2022 |
| 2 | Введение. Постановка задачи | 22.02.2022 |
| 3 | Проектирование веб-сайта | 28.02.2022 |
| 4 | Реализация структуры веб-сайта на HTML5 и внешнего оформления на Sass/СSS3 | 09.03.2022 |
| 5 | Тестирование веб-сайта | 30.03.2022 |
| 6 | Заключение | 22.04.2022 |
| 7 | Список использованных литературных источников. Приложения | 29.04.2022 |
| 6 | Подготовка и оформление пояснительной записки курсового проекта | 3.05.2022 |
| 7 | Сдача и защита курсового проекта | 10.05.2022 |

**5. Дата выдачи задания** «14» февраля 2022 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. И. Комарова

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата и подпись студента)

**Содержание**

[Введение 6](#_Toc103343756)

[1. Постановка задачи 7](#_Toc103343757)

[1.1. Обзор аналогичных решений 7](#_Toc103343758)

[1.2. Техническое задание 10](#_Toc103343759)

[1.3 Выбор средств реализации программного продукта 10](#_Toc103343760)

[1.4. Вывод 12](#_Toc103343761)

[2. Проектирование страниц веб-сайта 12](#_Toc103343762)

[2.1. Выбор способа вёрстки 12](#_Toc103343763)

[2.2. Выбор стилевого оформления 12](#_Toc103343764)

[2.3. Выбор шрифтового оформления 13](#_Toc103343765)

[2.4. Разработка логотипа 13](#_Toc103343766)

[2.5. Разработка пользовательских элементов 14](#_Toc103343767)

[2.6. Разработка спецэффектов 15](#_Toc103343768)

[2.7. Вывод 15](#_Toc103343769)

[3. Реализация структуры веб-сайта 16](#_Toc103343770)

[3.1. Структура HTML-документа 16](#_Toc103343771)

[3.2. Добавление таблиц стилей Sass и CSS 16](#_Toc103343772)

[3.3. Использование стандартов XML (SVG) 17](#_Toc103343773)

[3.4. Вывод 17](#_Toc103343774)

[4. Тестирование веб-сайта 18](#_Toc103343775)

[4.1. Адаптивный дизайн веб-сайта 18](#_Toc103343776)

[4.2. Кроссбраузерность веб-сайта 18](#_Toc103343777)

[4.3. Руководство пользователя 18](#_Toc103343778)

[4.4. Вывод 19](#_Toc103343779)

[5. Заключение 20](#_Toc103343780)

[6. Список использованных литературных источников. 21](#_Toc103343781)

[7. Приложение 22](#_Toc103343782)

[Приложения 1. Прототипы веб-страниц 22](#_Toc103343783)

[Приложение 2. Макет структуры веб-сайта 25](#_Toc103343784)

[Приложение 3. Листинг HTML-документа 28](#_Toc103343785)

[Приложение 4. Листинг Sass/Scss 30](#_Toc103343786)

[Приложение 5. Листинг Css 32](#_Toc103343787)

[Приложение 6. Листинг JavaScript 34](#_Toc103343789)

# **Введение**

Некоторым людям нужен адреналин и экстрим, чтобы почувствовать, что их жизнь идет вперед, а другие предпочитают пассивный отдых. Дело в том, что каждый из нас ощущает удовольствие, проводя отдых, где это желаемо. Это очень важно для людей, которые много работают. Есть много возможностей, чтобы сделать их жизнь более интересной и веселой. Наилучший вариант выбора возможностей для людей, которые хотят отлично отдохнуть поможет туристическое агентство.

Сегодня стало возможным зарезервировать место на морском курорте на другой стороне мира. Не выходя из дома, вы можете заказать билеты через Интернет или по телефону. Самолет доставит вас прямо туда, куда вы желаете, и через несколько часов после отбытия из своей страны, вы сможете оказаться на тропическом побережье, наслаждаясь чистейшим воздухом, плавая в кристально чистой, теплой воде тропического моря. Для этого и был разработан веб-сайт туристического агентства «Voyage».   
 Данный веб-сайт предоставляет услуги по заказу туристических туров по всему миру, каждый сможет найти что-то на свой вкус: от путешествий по крупным странам и городам до посещений достопримечательностей. Сайт выполнен с помощью языков разметки HTML, CSS, SCSS и XML.

Целевой аудиторией являются люди, заинтересованные в туристических услугах любого возраста.

И для заказа самых интересных и запоминающих туров вам сможет помочь мой проект «Веб-сайт туристического агентства “Voyage”».

Курсовой проект включает в себя 7 разделов: Постановка задачи, Проектирование страниц веб-сайта, Реализация структуры веб-сайта, Тестирование веб-сайта, Заключение, Список использованной литературы, Приложение.

**Цель курсовой работы**: разработать информационный веб-сайт по игре DOTA 2 с использованием HTML5 и XML, а также с применением CSS3.

**Задачи**:

1. Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
2. Создать макет веб-сайта при помощи онлайн-сервиса для разработки интерфейсов и прототипирования Figma
3. Разработать структуру веб-сайта.
4. Наполнить сайт информацией по теме.
5. Протестировать веб-сайт.

# **Постановка задачи**

## **1.1.** **Обзор аналогичных решений**

Курсовой проект представляет из себя информационный веб-сайт игры DOTA 2. Разработанный мною веб-сайт имеет множество аналогов, разного уровня проработки. В качестве примера можно привести несколько сайтов.

Аналог №1 — сайт dota2.ru.

Предоставленный сайт выполнен в минималистическом стиле со множеством вкладок, информацией о предстоящих киберспортивных матчах и информацией о героях игры DOTA 2.

На главной странице сайта свежие новости игры DOTA 2, информация о форуме, также информация о предстоящих киберспортивных матчах.

Недостатками данного аналога являются слишком тёмные цвета дизайна, которые не очень приятны при поиске информации. Также на сайте присутствует множество рекламы, которая мешает поиску.

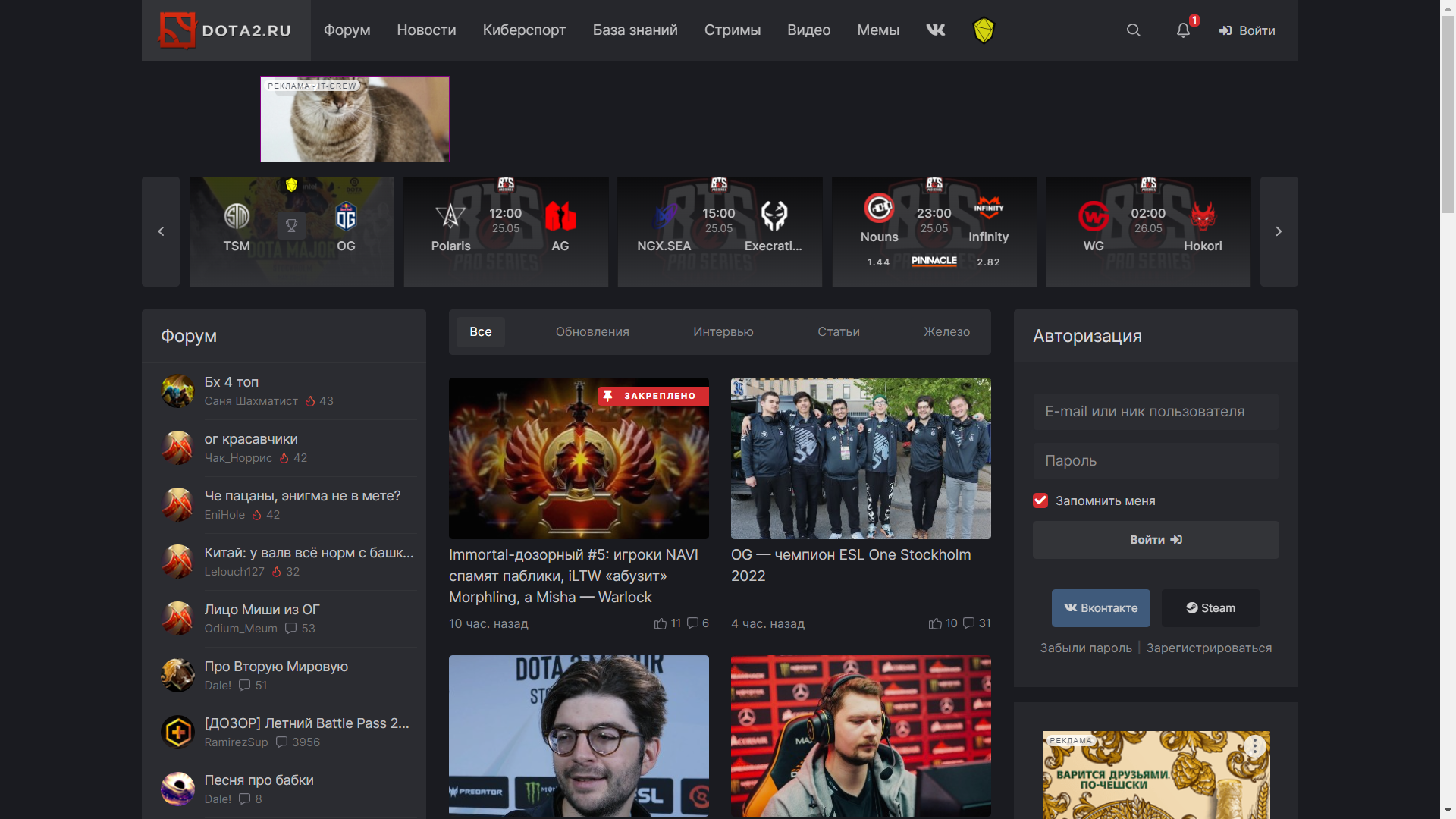


Рисунок 1.1

Хорошо проработана мобильная версия сайта. Оформление мобильной версии представлено на рисунке 1.2. Очень удобно находить нужную информацию и это не займет много времени, чтобы найти то, что вам понравится. Недостатком также является не удачно выбранный цвета оформления, в данном примере оттенки также слишком темные.

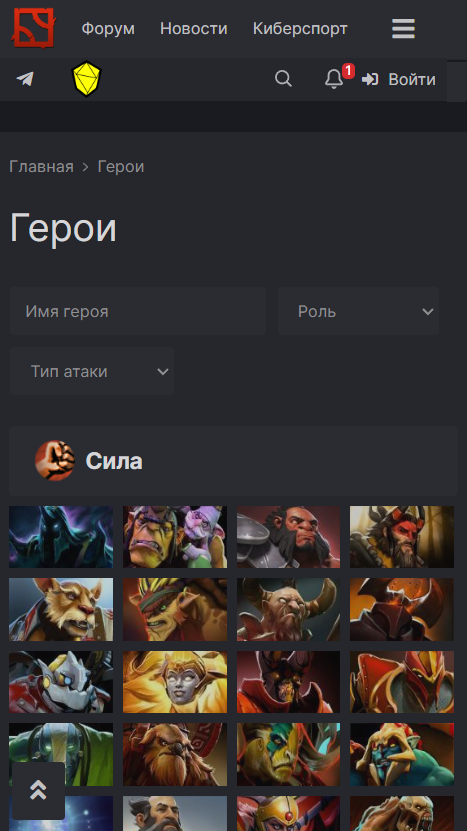


Рисунок 1.2

Аналог №2 — сайт dota2.fandom.com.

Dota 2 Вики - Это хороший сайт о туризме и обширная социальная сеть, где все желающие могут создать «Вики». Рассказы, мнения, отзывы, фотографии, видео – все это можно найти на сайте.

Недостатком является множество рекламы, которая мешает найти нужную информацию.

Дизайн сайта представлен на рисунке 1.3.

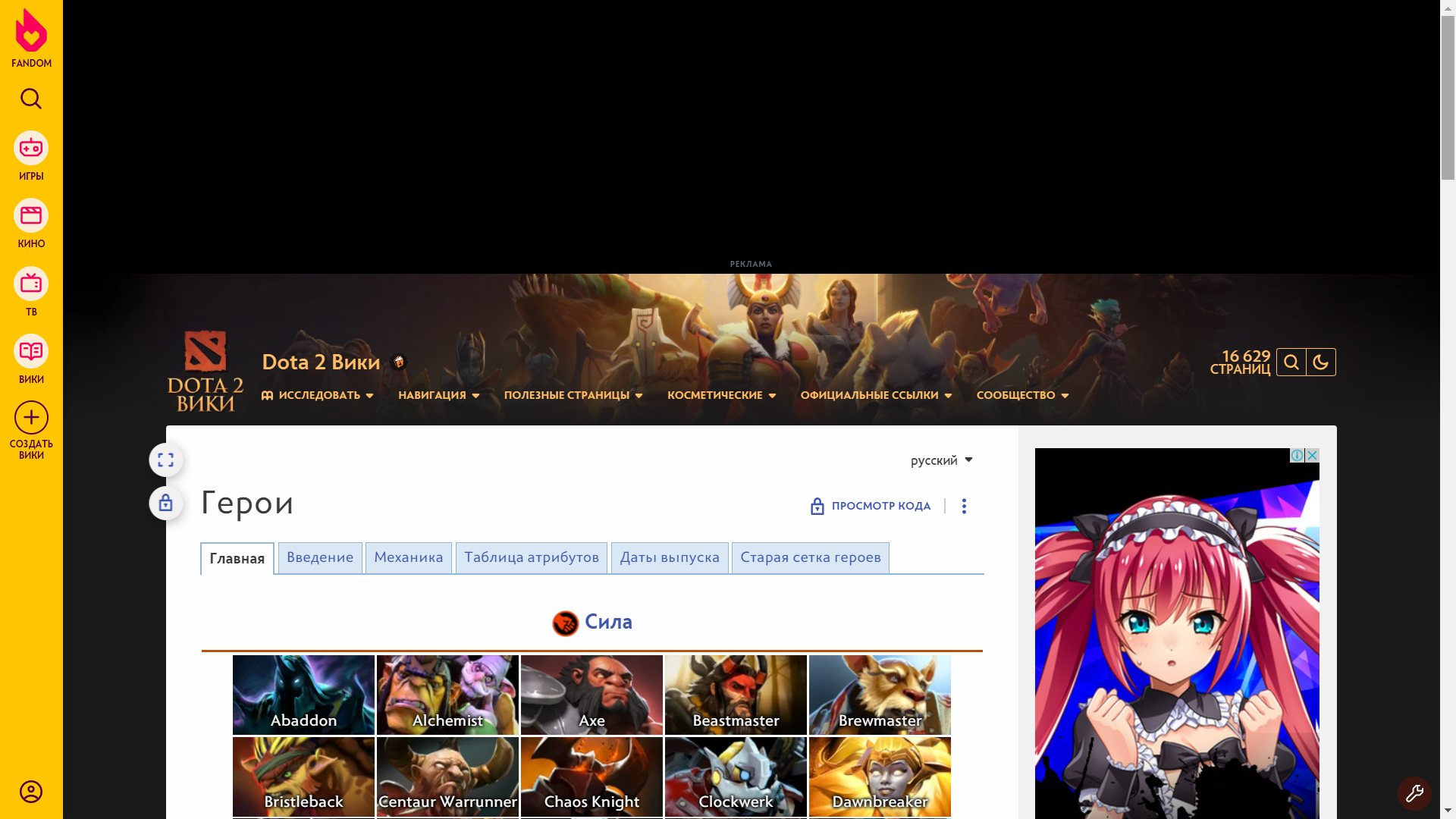


Рисунок 1.3

Также есть мобильная версия сайта. Она не очень удобно проработана. Каждый пользователь с легкостью найдет нужную ему информацию, а также сможет прочитать отзывы людей об игре DOTA 2. Оформление сайта представлен на рисунке 1.4.

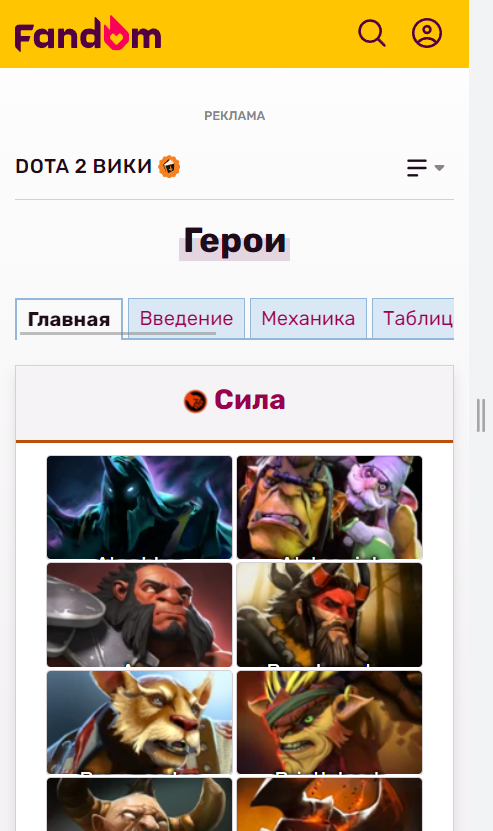


Рисунок 1.4

Посмотрев аналоги популярных сайтов можно сделать вывод о дизайне сайта и его главной задачи.

## **1.2. Техническое задание**

В данном проекте требуется создать многостраничный информационный сайт игры DOTA 2. Сайт должен содержать информацию о героях игры и об обновлениях, новостях игры.

Главная задача сайта — дать пользователю информацию о героях игры DOTA 2.

В мобильной версии(ширина до 560px) страница должна адаптироваться под ширину устройства. Начиная с ширины в 1000px должна открываться версия для персонального компьютера.

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа, адаптивность и кроссбраузерность сайта (корректное отображение в браузерах, таких как Google Chrome, Mozilla Firefox, Microsoft Edge, Opera.).

В проекте используется гипертекстовый язык разметки HTML, CSS/Sass, графические элементы в формате SVG; данные хранятся в XML-формате.

­­­Задачи курсового проекта:

- Размещение основной информации на сайте.

- Размещение информации об отдыхе.

## **1.3 Выбор средств реализации программного продукта**

Прежде всего следует создать макет, который соответствовал бы всем важнейшим требованиям, для этого можно использовать онлайн-сервис для разработки интерфейсов и прототипирования Figma. Проектирование макета одно из важнейших условий для создания читабельного сайта, в котором пользователь найдет полезную для себя информацию.

Сайт разрабатывался в редакторе Visual Studio Code 19. Страницы сайта могут просматриваться во всех популярных браузерах (Microsoft Edge, Safari, Opera, Mozilla Firefox, Google Chrome).

Visual Studio Code (рис. 1.6) — редактор исходного кода, разработанный Microsoft для таких платформ, как Windows, Linux и macOS. Позиционируется как редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга, в чём и заключается преимущества перед другими редакторами кода.

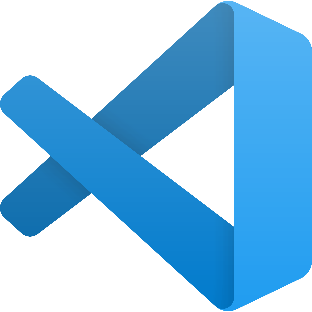


Рисунок 1.5

Для выполнения проекта использовались языки: HTML, CSS/Scss(Sass), XML, JavaScript. Структура сайта создана с помощью языка разметки HTML, дизайн сайта оформлен с помощью CSS/Scss. JS используется для создания XML-запроса, то есть для получения данных из XML-файла. Также JS используется для заполнения этими данными HTML-документа.

[HTML](https://ru.wikipedia.org/wiki/HTML) (Hyper Text Markup Language, расшифровывается как язык разметки гипертекстовых страниц) — основа всех современных сайтов. HTML является языком разметки, но не языком программирования. Под языком разметки понимается набор символов, созданный с целью передачи информации о строении текста. Это значит, что в отличии от языков программирования, язык разметки только управляет "внешним видом" текста для конечного пользователя. HTML-документ имеет расширение .html или .htm. В браузере он преобразуется в веб-страницу и состоит из набора тегов. Они как раз и помогают представлять текст на экране: благодаря им браузер понимает, что он читает не просто текст, а структурированную информацию, разбитую на блоки. Браузер интерпретирует HTML-документ, выстраивая его структуру (DOM) и отображая ее в соответствии с инструкциями, включенными в этот файл (таблицы стилей, скрипты).

CSS (Cascading Style Sheets) — язык таблиц стилей, который позволяет прикреплять стиль (например, шрифты и цвет) к структурированным документам (например, документам HTML и приложениям XML). Обычно CSS-стили используются для создания и изменения стиля элементов веб-страниц и пользовательских интерфейсов, написанных на языках HTML и XHTML, но также могут быть применены к любому виду XML-документа, в том числе XML, SVG и XUL. Каскадные таблицы стилей описывают правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств, остальные свойства не будут оказывать на него никакого влияния.

Scss – это дополнение (расширение) таблицы стилей CSS. Данный метыязык значительно упрощает работу, поддерживает некоторые элементы языков программирования (создание переменных, циклов, условных операторов).

XML (eXtensible Markup Language) - это расширяемый язык разметки. XML изначально придумали для более удобного хранения и передачи данных. **XML — это не исполняемый код, а язык описания данных.** После описания данных с помощью XML, необходимо написать код (на одном из языков программирования), который сможет эти данные отправить/принять/обработать.

## **1.4. Вывод**

В данном разделе были рассмотрены аналогичные решения, были проанализированы их недостатки и преимущества, для создания в последующем качественного веб-сайта. С помощью этой информации будет создан сайт, подчеркивающий лучшие стороны аналогов.

Кроме того, были рассмотрены средства реализации программного продукта, такие как языки разметки, а также редактор кода VS Code. Были проанализированы основные дополнительные средства редактора. Все вышеперечисленные пункты понадобятся для достижения поставленной задачи и для создания в дальнейшем качественного продукта.

# **2. Проектирование страниц веб-сайта**

## **2.1. Выбор способа вёрстки**

В соответствие с требованиями к проекту, в частности адаптивность и кроссбраузерность сайта, было принято решение об использовании SASS.

Для создания отдельных блоков страницы, такие как горизонтальное меню, будет использована Flexbox-верстка. С помощью этой технологии можно очень просто и гибко расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров. CSS Flexbox поддерживается всеми используемые на сегодняшний момент современными браузерами (с использованием префиксов: IE10+, Edge12+, Firefox 2+, Chrome 4+, Safari 3.1+, Opera 12.1+, iOS Safari 3.2, Opera mini, Android 2.1+, Blackberry 7+).

Исходя из выбора способа верстки были расположены все основные элементы сайта и разработаны макеты страниц. Разработанные макеты сайта представлены в приложении 1.

## **2.2. Выбор стилевого оформления**

Для реализации стилевого оформления сайта было выбрано минималистическое направление с элементами классического веб-дизайна. Такой стиль позволяет привлечь внимание новых пользователей.

Сайт не загромождается лишними деталями (боковыми панелями и т.п.), большим количеством спецэффектов. Акцент на сайте ориентируется на основной контент.

Рисунок 2.1

Цветовые решения не слишком яркие. Основные цвета: синий, белый, серый, фиолетовый. Цветовая палитра представлена на рисунке 2.1

## **2.3. Выбор шрифтового оформления**

В данном проекте будут использованы два основных шрифта: Montserrat и Playfair Display SC. Шрифты подключаются с помощью правила @font-face.

Шрифт Montserrat – [геометрический шрифт без засечек](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Sans-serif#Geometric), относится к категориям гротески, декоративные. Разработан [аргентинским](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Argentina) графическим дизайнером Джульетой Улановской и выпущен в 2011 году. Благодаря большой [высоте](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/X-height), коротким [спускам](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Descender) и широким [отверстиям](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Counter_(typography)) этот шрифт обеспечивает высокую [разборчивость](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Legibility) даже в небольших размерах. Данный шрифт также является популярным выбором для [мобильных приложений](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Mobile_app) и [игр](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Mobile_game), включая такие как [Idle Supermarket, Tycoon](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/wiki/Hothead_Games" \o "Горячие игры) и [Snake vs Block](https://translated.turbopages.org/proxy_u/en-ru.ru.6f2a8994-62670321-613b80c5-74722d776562/https/en.wikipedia.org/w/index.php?title=Snake_vs_Block&action=edit&redlink=1" \o "Snake vs Block (страница не существует)). Гарнитура содержит 18 начертаний, поддерживает 73 языка.

Playfair Display SC – это шрифт с засечками. Он повторяет европейский стиль писания 18 века. Имеет очень большую высоту и короткие спуски. Шрифт может быть использован, например, в заголовках новостей или для стилистического эффекта в заголовках. Лучше всего использовать для: стилизованных, элегантных и сложных веб-сайтов: различных агентств, где есть портфолио, в блогах о путешествиях и моде.

## **2.4. Разработка логотипа**

Разработка логотипа осуществлялась в графическом редакторе Figma. Дизайн довольно минималистичен; было принято решение использовать официальный логотип игры в цветах подходящих для сайта: сине-белые цвета. Дизайн логотипа представлен на рисунке 2.2.

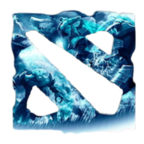
****

Рисунок 2.2

## **2.5. Разработка пользовательских элементов**

В проекте будет представлен элемент пользовательского интерфейса как веб форма.

Форма – это инструмент, с помощью которого HTML-документ может послать некоторую информацию в некоторую заранее определенную точку, где информация будет некоторым образом обработана (представлена на рисунке 2.3).

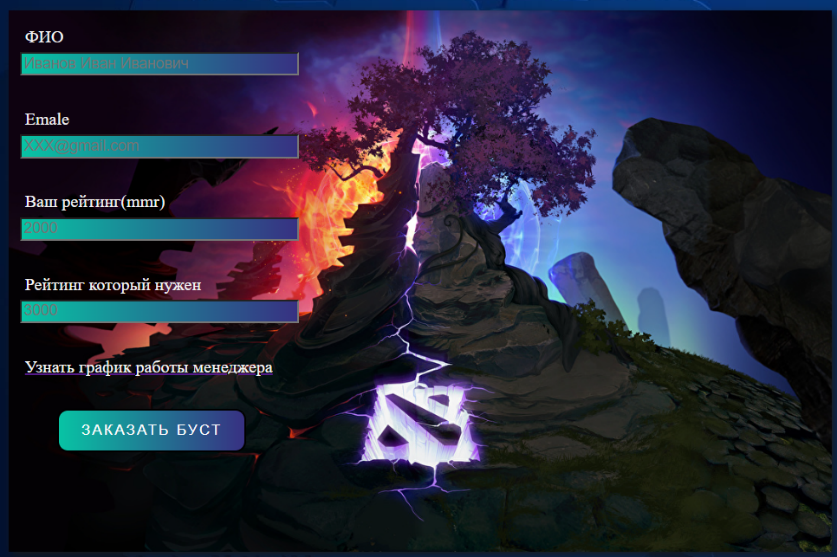


Рисунок 2.3 – пример веб формы

Также на главной странице в навигационной панели, которая содержит основные тематические страницы веб-сайта: «Новости», «Герои», «Буст». (рисунок 2.4)

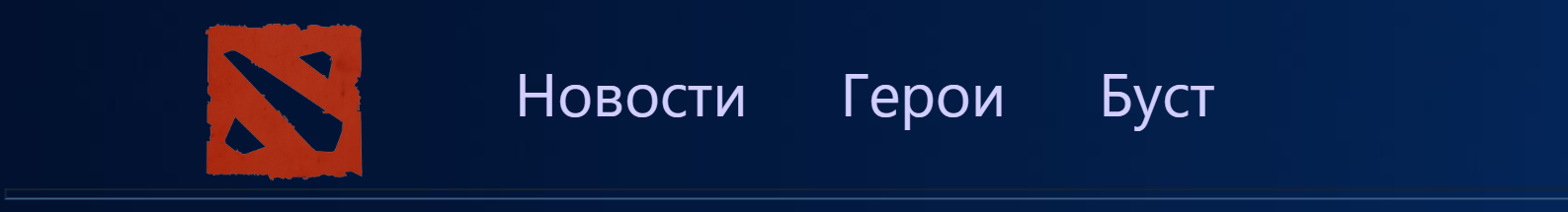


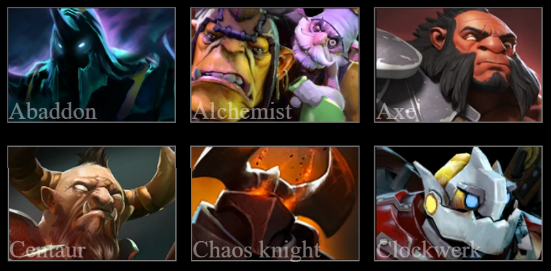
Рисунок 2.4 – пример панели

Панель с главными страницами использовалась на всех страницах для быстрого доступа к ним.

## **2.6. Разработка спецэффектов**

На сайте продумана простейшая анимация некоторых элементов. Для реализации отзывчивого интерфейса продумана реакция интерфейса по наведению курсора на ссылки. При наведении курсора на героя происходит анимация: иконка героя становится больше, имя героя более отчётливо видно, появляется задняя тень. На рисунке 2.6 представлен пример, где первой фазой иконки героя является состояние по умолчанию, а на рисунке 2.6 представлена вторая фаза.

Состояние 1:



Состояние 2:

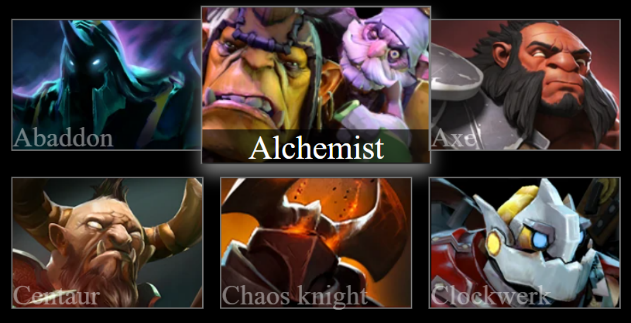


Рисунок 2.6

Примерно такая анимация использовалась на всех аналогах, а такжа в самой игре при выборе героя.

## **2.7. Вывод**

На данном этапе были созданы прототипы, а затем и макеты веб-страницы для облегчения последующей верстки. Было упорядочено расположение всех элементов на страницах (навигационного меню, логотипа, основного контента каждой страницы, расположение картинок). Было определено стилевое оформление веб-сайта, то есть основная цветовая палитра, шрифтовое оформление, элементы пользовательского интерфейса, а также эффекты и переходы, придающие страницам динамики.

# **3. Реализация структуры веб-сайта**

## **3.1. Структура HTML-документа**

Структура HTML-документа определяет базовый «скелет» для будущего веб-сайта. Код выполнен по стандарту HTML5.

Структура тела документа состоит из семантических тегов header, main и footer. Листинг представлен в (приложении 3).

В теге header находится горизонтальная навигационная панель. Она содержит ссылки на все основные разделы сайт, логотип и название компании. На других страницах веб-сайта элементы навигационного меню автоматически скрываются, а на их месте появляется так называемое “меню-бургер”. Навигационное меню было реализовано при помощи маркированного списка. Листинг представлен в (приложении 3).

Тег main выполняет роль главного контейнера всего сайт. Все теги заключены в него.

## **3.2. Добавление таблиц стилей Sass и CSS**

Добавление таблиц стилей Sass(Scss) и Css позволило упростить создание стилистического оформления страницы. Компиляция осуществлялась благодаря приложению Visual Studio Code. Преимуществами данного приложения являются компиляция файлов sass, scss.

Главными преимуществами Sass(Scss) перед Css это прежде всего автоматическая кроссбраузерность при компиляции Sass. Наличие вложенности в отличие от Css. Например, Sass облегчает применение свойств благодаря миксинам. Они позволяют один раз создать набор правил, чтобы потом использовать их многократно или смешивать с другими правилами. Переменные, управлять которыми намного проще в отличие от Css. А также импорт, Sass делает это без HTTP-запросов. Листинг представлен в (приложении 4).

В данной структуре используется внешнее подключение таблиц стилей через тег link, так как это самый мощный и удобный способ определения стилей и правил для веб-сайта. Стили хранятся в отдельном файле, который может быть использован для любых веб-страниц. В таблицах стилей используются селекторы по классу, по идентификатору, универсальный селектор, а также селекторы потомков, псевдоклассы и псевдоэлементы.

## **3.3. Использование стандартов XML (SVG)**

В проекте XML-формат был использован для хранения информации о расписании менеджеров. Листинг представлен в (приложении 5).

Для создания XML-запроса, то есть для получения данных из XML-файла, был использован JavaScript.

Для создания атрибута героев было принято использование SVG-формата. Этот формат позволяет картинкам сохранять свое качество, несмотря на размер экрана. Листинг предоставлен в (приложение 5).

JS-формат был использован для создания основной страницы героев с подключением всей информации о героях через json.

## **3.4. Вывод**

В этом разделе мы рассмотрели, как была реализована структура HTML-документа, как были созданы таблицы стилей CSS/Sass, как использовался SVG-формат, как был использован XML-формат для хранения данных и как и для чего был использован JavaScript.

# 

# **4. Тестирование веб-сайта**

## **4.1. Адаптивный дизайн веб-сайта**

Сегодня количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. И для того, чтобы людям Интернета было удобно использовать какой-либо сайт, он должен быть адаптивным.

Я выбрал несколько устройств, где сайт должен быть удобным для пользователей, такие как Ipad Mini, Iphone 8 Plus и ноутбуки от 1400px в ширину.

Для более удобной адаптации использовались медиа-запросы, а точнее правило import. Также было использовано такое свойство для блоков, как max-width, чтобы width был равен 100 процентов, чтобы сайт был адаптивен ни только для тех устройств, которые указаны выше, но и для других.

## **4.2. Кроссбраузерность веб-сайта**

Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

Для отображения свойств, которые в разных браузерах работают по-разному, было принято использования вендорных префиксов. Они автоматически прописаны после компиляции файла с расширением sass.

После написания основной структуры страницы на HTML и внешнего стилевого оформления сайт нужно протестировать в разных браузерах. В ходе тестирования было выяснено, что Internet Explorer не поддерживает семантические теги, из-за чего вся flex-box верстка поломалась, и некоторые свойства, например, как backdrop-filter.

## **4.3. Руководство пользователя**

Данный сайт создан для того, чтобы пользователь мог найти нужную информацию об интересующем герое игры DOTA 2, также новости об игре и повышении рейтинга.

Для удобного пользования веб-сайтом «DOTA 2 герои» реализовано:

- разработан удобный и приятный пользовательский интерфейс, чтобы даже человек, не работавший с компьютером достаточно, быстро мог разобраться в этом веб-сайте;

- веб-страница оснащена гиперссылками для быстрого перехода между страницами.

Вверху главной страницы, вы можете увидеть навигационную панель. Она содержит основные тематические страницы веб-сайта: «Новости», «Герои», «Буст».

На странице «Новости» пользователь может изучить свежие новости об игре и её обновлениях, следить за прохождением крупных турниров и прочем что связано с игрой.

На странице «Герои» присутствует вводная информация о героях игры DOTA 2 и сами герои.

На странице «Буст» предоставлена информация о всех предоставляемых вариантах повышения рейтинга в игре DOTA 2.

## **4.4. Вывод**

На данном этапе было проанализировано поведение веб-сайта при использовании с разных устройств, а также веб-ресурс был протестирован в большом количестве браузеров. 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 курсового проекта было разработка информационного веб-сайта по игре DOTA 2. Который предоставляет информацию о всех героях игры их способностях и истории. Помимо главной страницы была реализована страница: «Новости», на которой можно всю информацию связанную с игрой.

Для реализации макета веб-страницы были исследованы сайты схожей тематикой и исследованы недостатки веб-сайтов.

Была подготовлена среда разработки Visual Studio Code для комфортного и ускоренного написания кода для веб-страницы. А именно выбор необходимых плагинов и цветовой темы.

Также проходило испытания технологий для получения лучшего результата во время пользования веб-страницы. Проходил анализ полученной информации по итогу испытании технологий и применены те или иные выводы по использования нового функционала.

Реализация важных элементов страницы либо же скриптов были представлены в приложениях. Которые дают общую информацию о проекте.

Для достижения поставленной цели для курсового проекта функционально были реализованы следующие задачи:

1. Анализ аналогичных решений;
2. Выбор способа верстки;
3. Выбор стилевого оформления;
4. Разработка логотипа;
5. Разработка пользовательских элементов;
6. Разработка спецэффектов;
7. Добавление веб-сайту адаптивности;
8. Кроссбраузерность веб-сайта;
9. Создание руководства пользователя;

Список используемых литературных источников представлен в пункте 6. Также дополнительная информация находится в приложениях 1-6.

Результатом курсового проекта стал готовый к работе веб-сайт, в котором размещена информация о игре DOTA 2. Данный ресурс стабилен в работе и не требует высоких знаний для работы с ним.

Таким образом, из выше сказанного, можно сделать вывод, что в курсовом проекте раскрывается все теоретические и практические аспекты темы и достигнуты поставленные цели и задачи роботы.

# **Список использованных литературных источников.**

1.Документация по JavaScrip/HTML [Электронный ресурс] / Справочный Режим доступа : <https://developer.mozilla.org> – Дата доступа 01.02.2022.

2.Справочник по HTML [Электронный ресурс] / Справочный

Режим доступа : <https://htmlbook.ru>

3.Документация по Figma [Электронный ресурс] / Справочный

Режим доступа : <https://help.figma.com>

4.Краткое руководство по XML [Электронный ресурс] / Справочный

Режим доступа : [https://xml.readthedocs.io](https://xml.readthedocs.io/)

5.Документация по Sass/Scss [Электронный ресурс] / Справочный

Режим доступа : <https://sass-scss.ru/guide/>

# **Приложение**

## **Приложения 1. Прототипы веб-страниц**

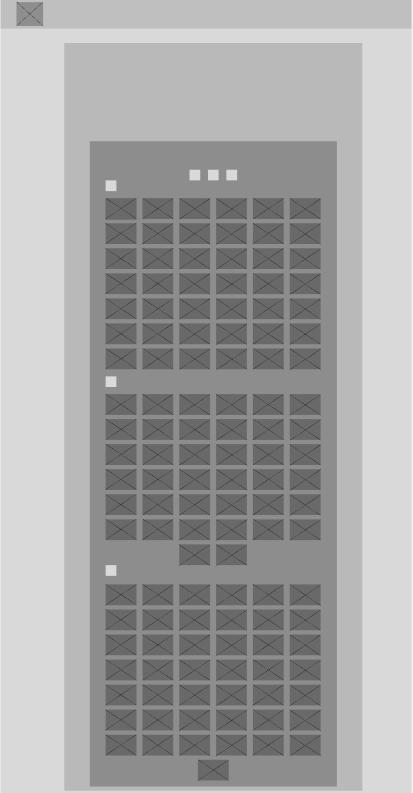


Рисунок 1- прототипы страницы «Герои»

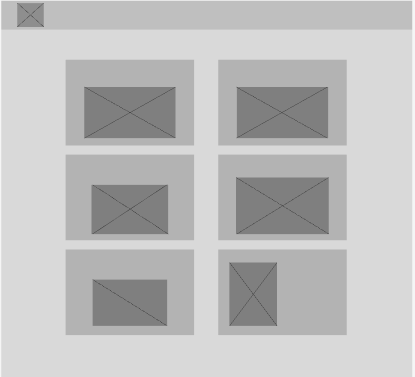


Рисунок 2- прототипы страницы «Новости»

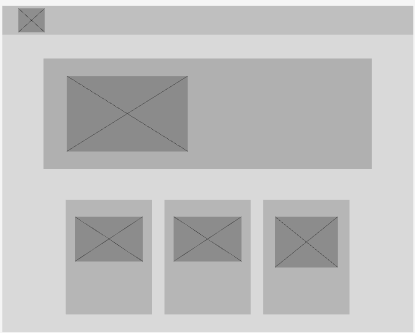


Рисунок 3- прототип страницы «Буст»

## **Приложение 2. Макет структуры веб-сайта**

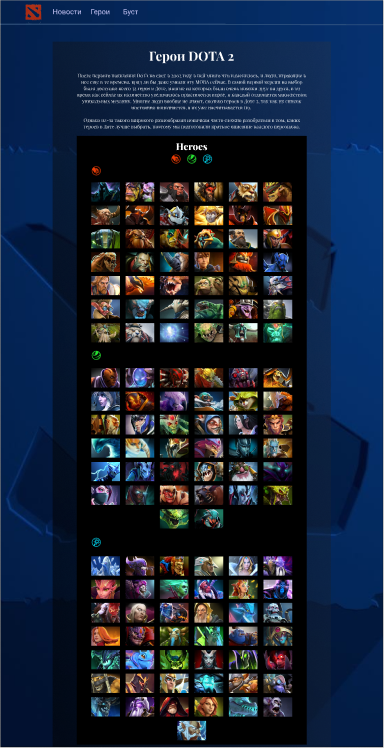


Рисунок 1- макеты страницы «Герои»



Рисунок 2- макеты страницы «Новости»

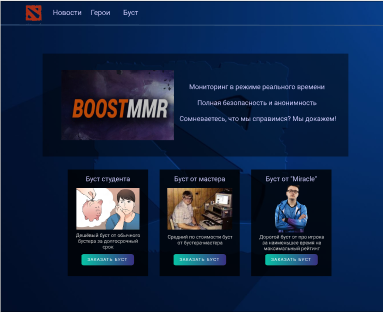


Рисунок 3- макет страницы «Буст»

## **Приложение 3. Листинг HTML-документа**

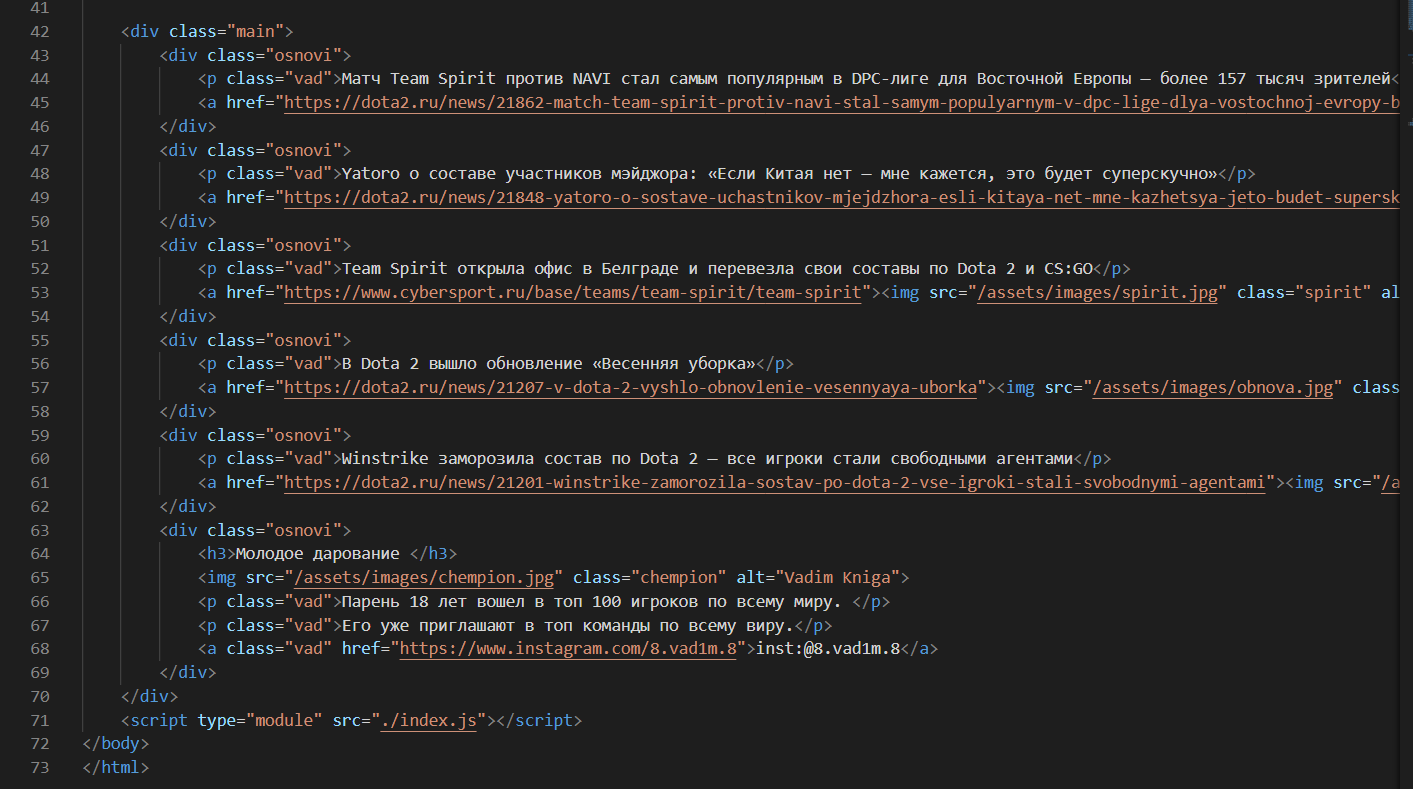


Рисунок 1-основная структура страницы «Новости»,

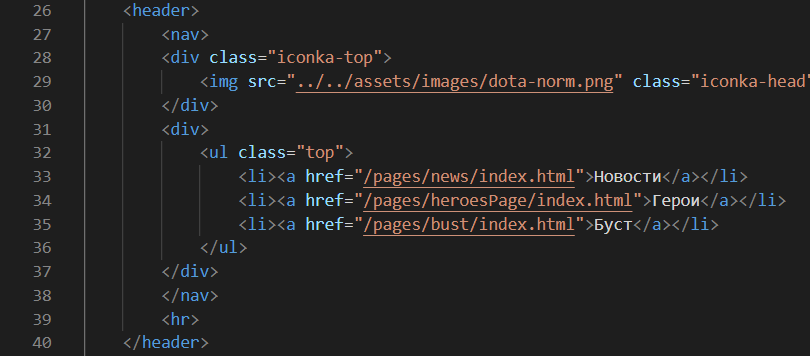


Рисунок 2-структура тега header

## **Приложение 4. Листинг Sass/Scss**

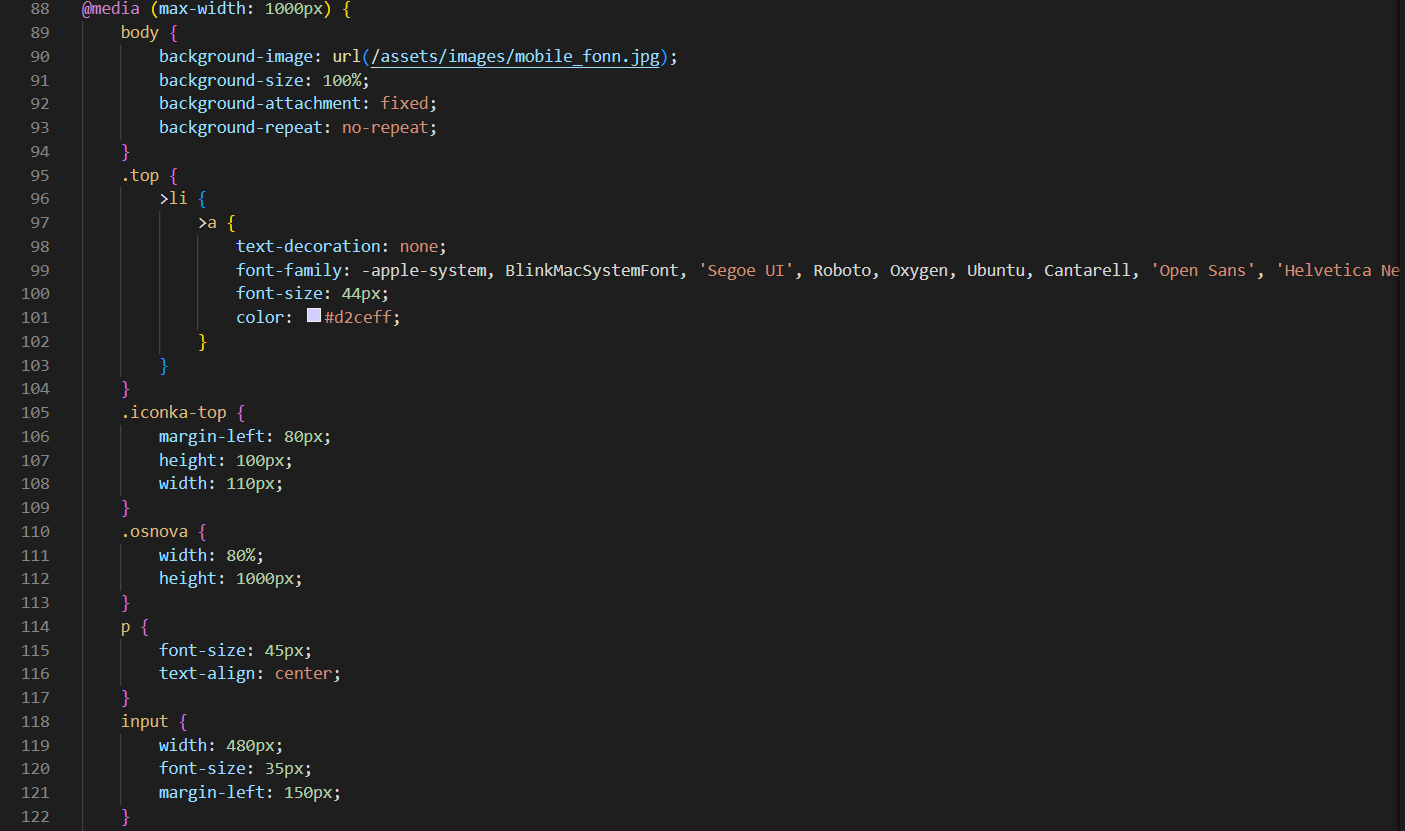


Рисунок 1 – Sass стили, применяемые ко всем страницам

## **Приложение 5. Листинг Css**

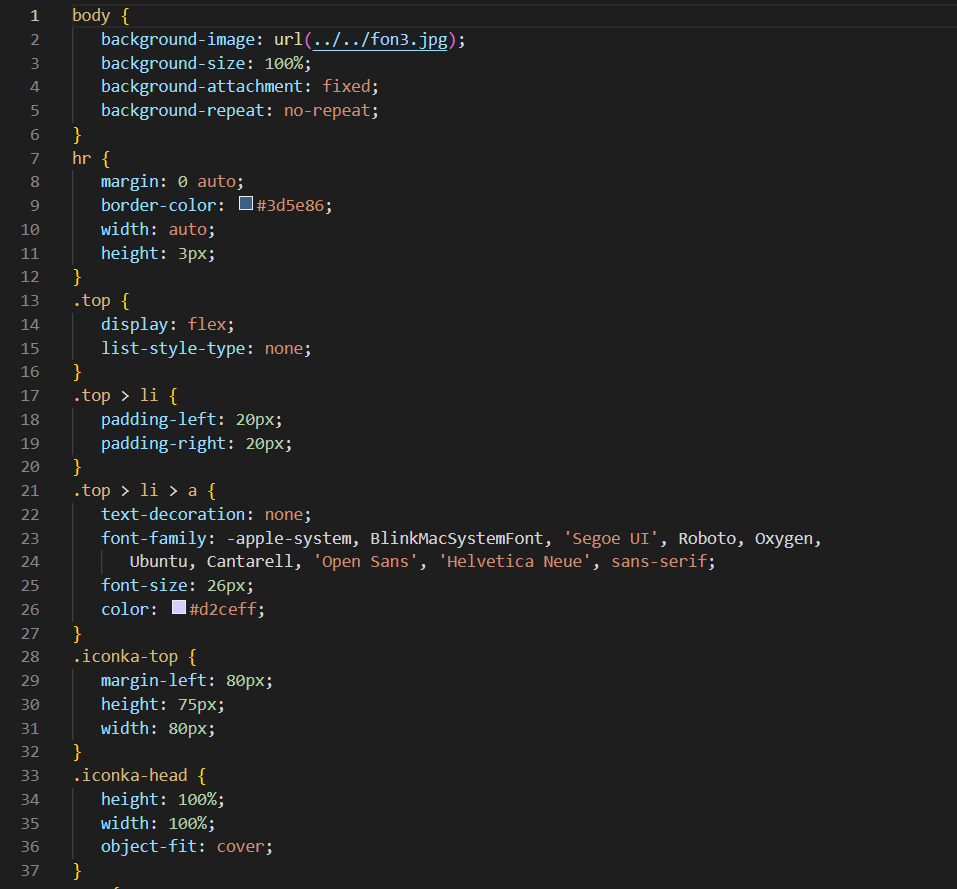


Рисунок 1 – структура Css применяемая ко всем страницам

## **Приложение 6. Листинг JavaScript**

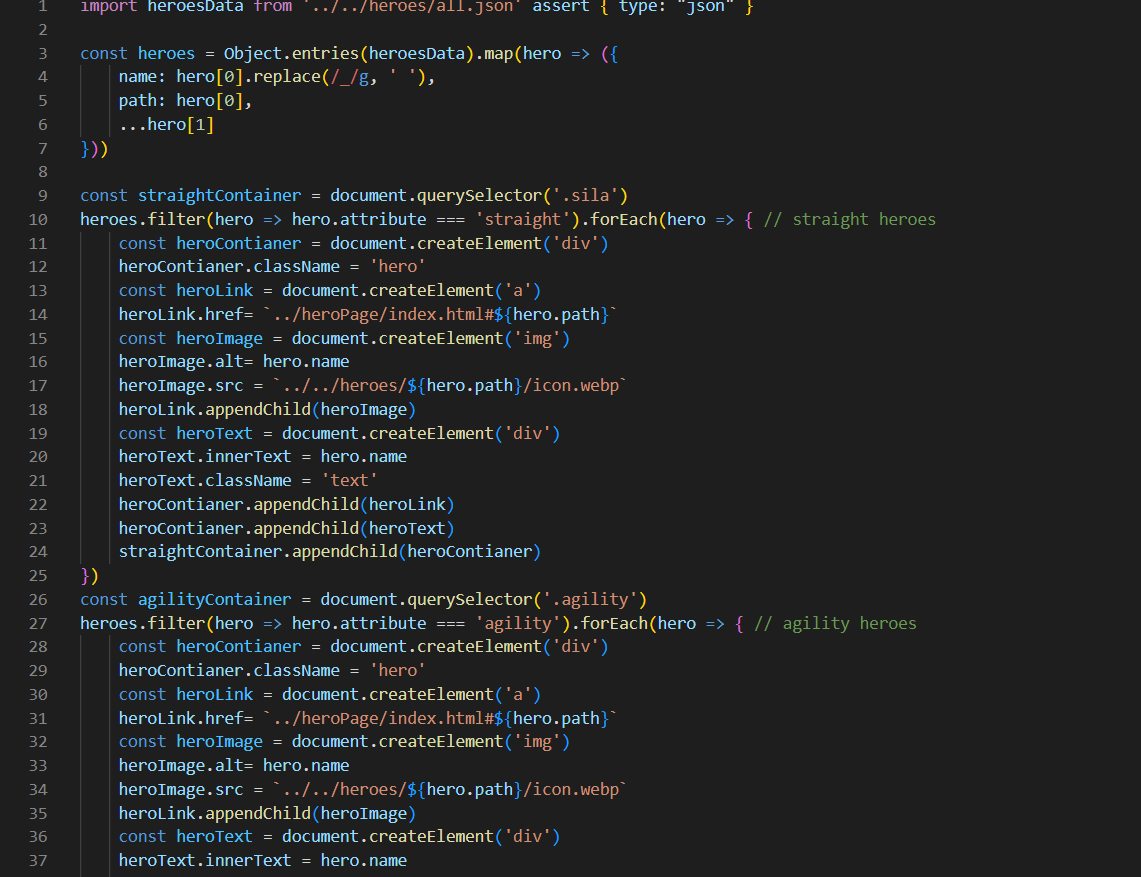


Рисунок 1 – Структура JS-скрипта, для использования XML-файла

**Приложение 7. Проверка Linter/ HTML Validator**

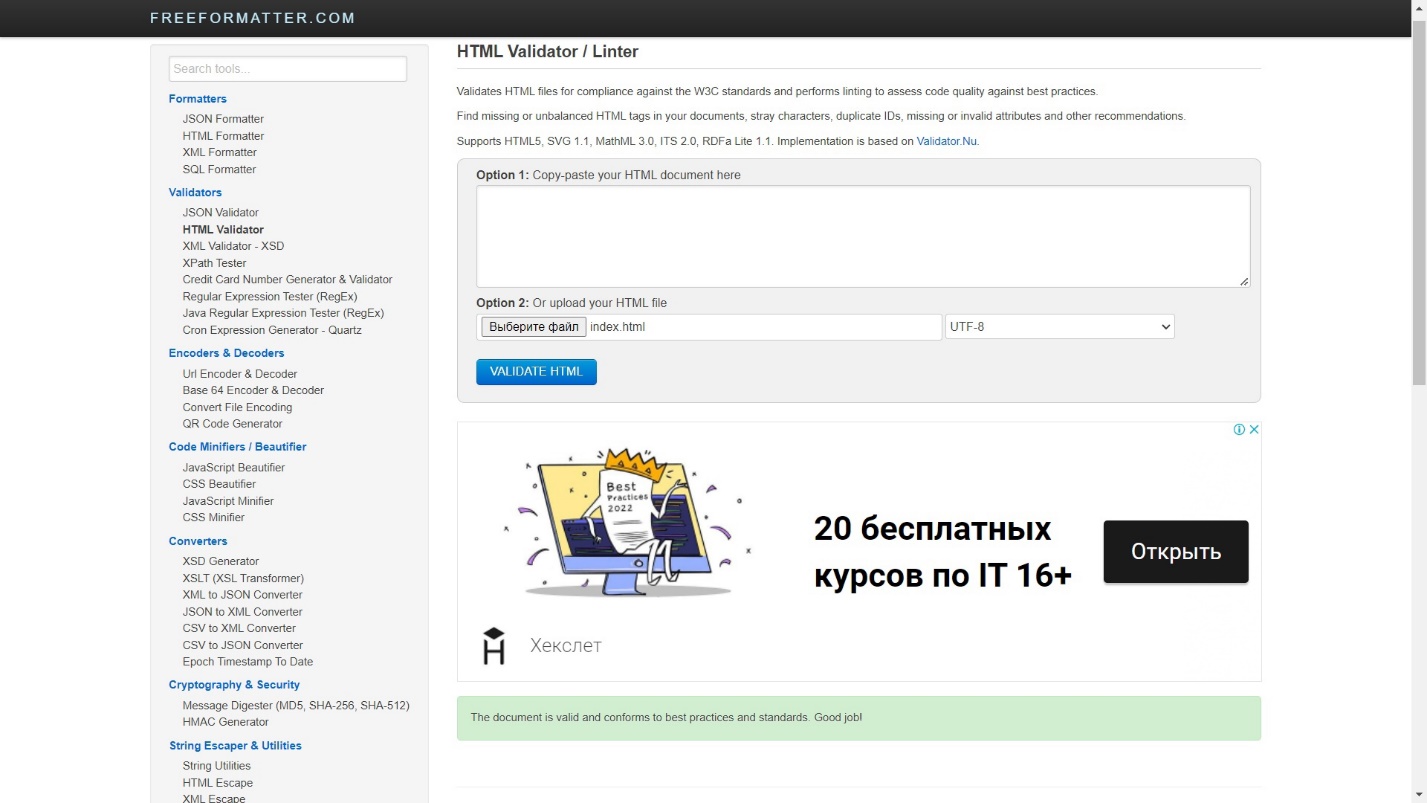


Рисунок 1 – проверка основного сайта