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 по тематике «Фильмы»»

**Исполнитель**

студент(ка) 1 курса 2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Хващевский Г. 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

[Введение 4](#_Toc103760024)

[1.Постановка задачи 5](#_Toc103760025)

[1.1Обзор аналогичных решений 5](#_Toc103760026)

[1.2 Формирование требований к программному продукту 6](#_Toc103760027)

[1.3 Техническое задание 7](#_Toc103760028)

[1.4Выбор средств реализации программного средства 7](#_Toc103760029)

[1.5 Вывод 8](#_Toc103760030)

[2 Проектирование страниц веб-сайта 9](#_Toc103760031)

[2.1 Выбор способа верстки 9](#_Toc103760032)

[2.2 Выбор стилевого оформления 9](#_Toc103760033)

[2.3 Структура сайта 9](#_Toc103760034)

[2.4 Выбор шрифтового оформления 10](#_Toc103760035)

[2.5 Разработка логотипа 11](#_Toc103760036)

[2.6 Разработка пользовательских элементов 11](#_Toc103760037)

[2.6 Прототипы 14](#_Toc103760038)

[2.7 Макет 14](#_Toc103760039)

[2.8Вывод 14](#_Toc103760040)

[3. Реализация структуры сайта 16](#_Toc103760041)

[3.1 Структура HTML документа 16](#_Toc103760042)

[3.2 Использование JavaScript 16](#_Toc103760043)

[3.3 Добавление таблиц стилей Sass и CSS 19](#_Toc103760044)

[3.4 Использование стандартов XML (SVG) 20](#_Toc103760045)

[3.5Вывод 21](#_Toc103760046)

[4 Тестирование сайта 22](#_Toc103760047)

[4.1 Адаптивный дизайн веб-сайта 22](#_Toc103760048)

[4.2 Кроссбраузерность веб-сайта 23](#_Toc103760049)

[4.3 Руководство пользователя 25](#_Toc103760050)

[4.4 Вывод 26](#_Toc103760051)

[Заключение 27](#_Toc103760052)

[Приложение 28](#_Toc103760053)

[Приложение 2 – макет веб-страницы 29](#_Toc103760054)

# Введение

С каждый днем Интернет интегрируется в нашу жизнь все глубже и глубже. Практически в каждой сфере жизни присутствует Интернет. Особенно сильно он переплетается с нашей повседневной жизнью после вспышки коронавируса. Люди изолировались в домах. Они стали реже выходить в магазины и чаще стали пользоваться интернет-магазинами. Больше не нужно ехать в магазин и стоять в очередях. Все это можно получить у себя дома, всего лишь нажимая кнопки на телефоне либо же компьютере.

Для большей заинтересованности потенциального покупателя многие сайты начали войну веб-дизайна. Каждый сайт стремился стать наиболее адоптированным и приятным на вид, интуитивно понятным и производительным.

Тема моей курсовой работы «Интернет-магазин гаджетов». В этом веб-сайте присутствует выбор различных товаров и возможность оформления покупки. Сайт должен быть максимально простым в дизайне, чтобы каждый мог зайти и сразу понять куда нажимать для достижения своих целей.

В курсовом проекте основными задачами будет создание сайта с адаптивной версткой, сверстать его максимально приятным для пользователя.

Задачи:

-Сортировка товаров по категориям: телефоны, ноутбуки, электронные часы

-Фильтрация товаров по ценам

-Создание формы для оформления покупки

В течении разработки курсового проекта планируется развитие навыков верстки адаптивного сайта с использованием HTML и CSS.

# 1.Постановка задачи

# 1.1Обзор аналогичных решений

При поисках идей для верстки сайта стоит учитывать то, что привлекательность сайта напрямую зависит от его простоты и его приятной цветовой палитрой. В качестве аналога для старта очень хорошо подошел сайт “Хайтек” представленный в рисунке.1.1

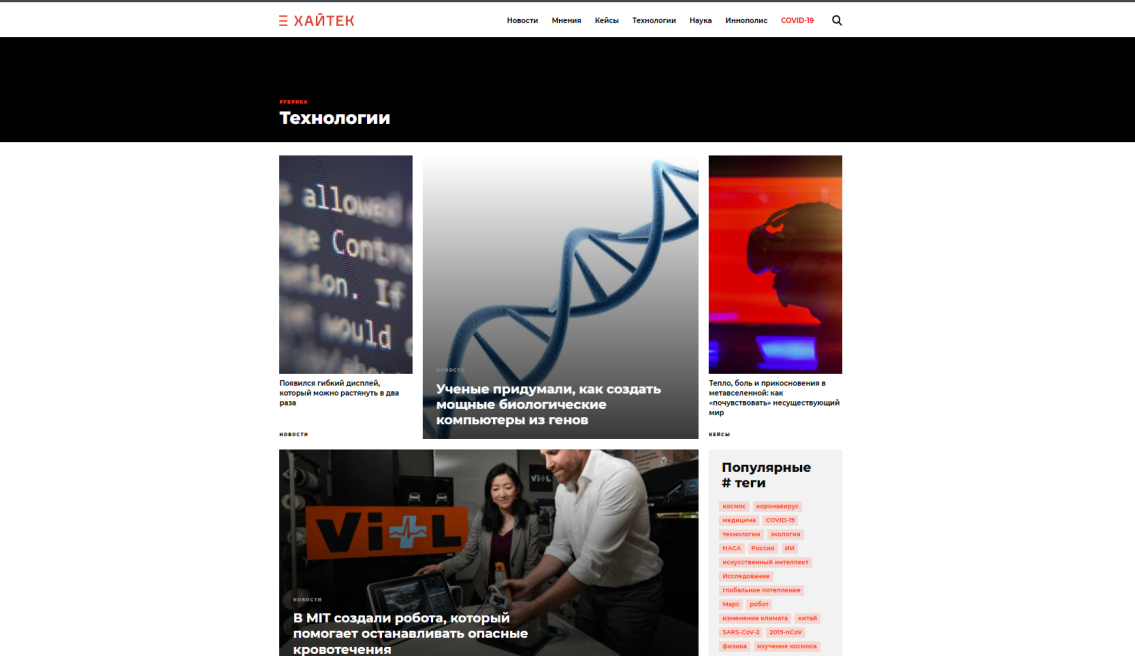


Рисунок 1.1 – аналог сайта

Основные идеи, взятые отсюда:

- акцентный оранжевый цвет, который делает сайт запоминающимся

- отсутствие лишней информации

-хедер, представленный на рисунке 1.2

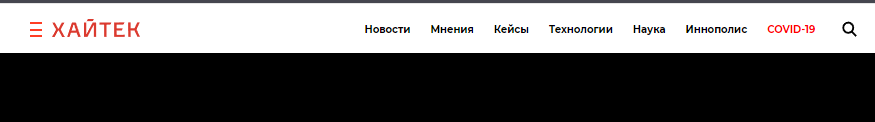


Рисунок 1.2 – хедер аналог

Основная страница со всеми товарами была оформлена как и у большинства интернет-магазинов. Основным примером для аналога был взят сайт “5 элемент”, представленный на рисунке 1.3. Тут присутствует вся важная информация для того, чтобы пользователь мог по первому взгляду выбирать нужный себе товар.

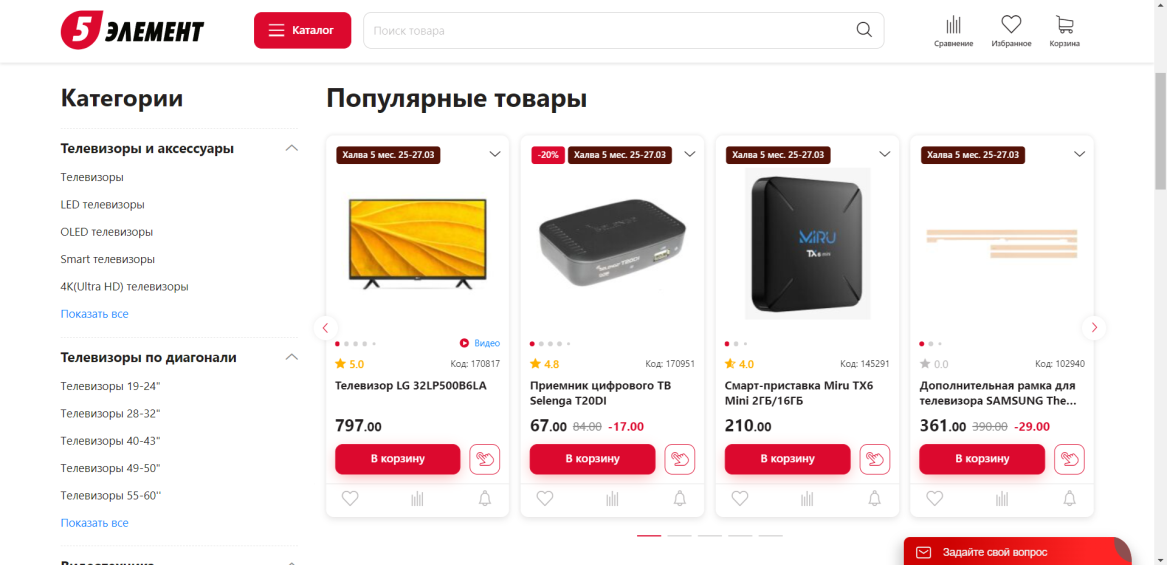


Рисунок 1.3 – аналог каталог

Страница с подробной информацией о товарах я так же создал используя идею аналога “5 элемент”, представленный на рисунке 1.4.

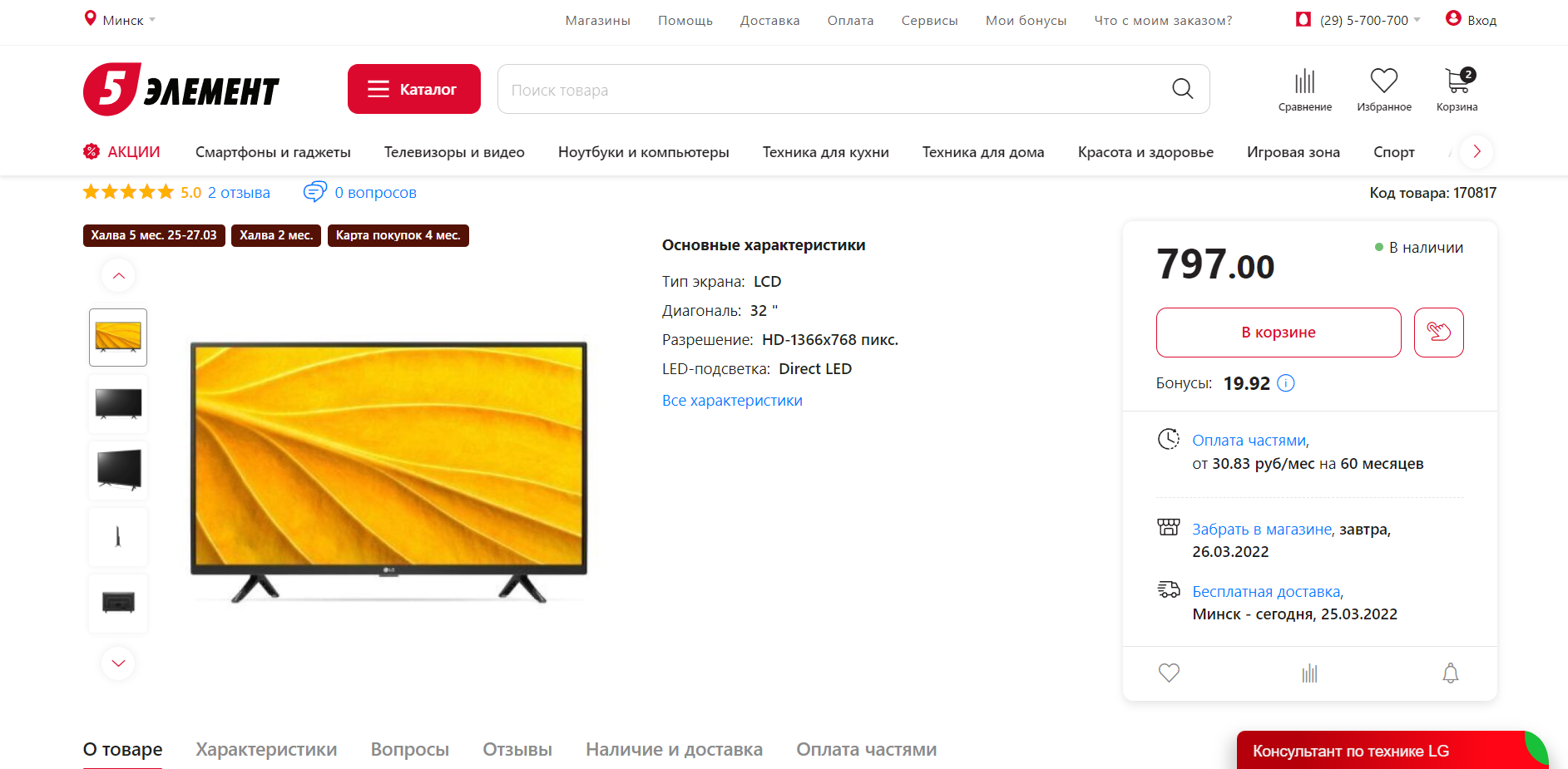


Рисунок 1.4 – аналог страницы с товаром

Тут указаны дополнительная информация, касаемо характеристик и цены товара. На этой странице минимум лишней информации и осталось лишь самое нужное.

# 1.2 Формирование требований к программному продукту

Основными требованиями к веб-сайту является минимальное количество сложных анимаций, которые ухудшают производительность сайта, адаптивность, понятность в интерфейсе, высококачественные картинки.

Сайт должен содержать в себе интуитивно понятный список товаров. Так же не менее важной функцией сайта должна быть форма, которая заполняется для оформления заказа. Оформление сайта должно быть понятно пользователю, чтобы он мог добраться до нужной себе информации.

# 1.3 Техническое задание

Основная задача сайта – дать пользователю возможность заказать товар, не выходя из дома. В сайте будет располагаться меню категорий, список товаров, кнопки покупки, форма для заполнения и оформления заказа.

Сайт будет направлен для людей, которые интересуются гаджетами и задумываются об их приобретении.

Основные задачи:

-создание сайта с множеством страниц с использованием программных средств

- создание интерфейса, который бы мог задержать пользователя на сайте как можно дольше.

- создание уникальных иконок и кнопок

-возможность ввода данных для оформления заказа

Основными требованиями к сайту являются динамическая верстка, легкие

анимации и функциональность в некоторых участках сайта.

# 1.4Выбор средств реализации программного средства

Для написания сайта была выбрана программа Visual Studio Code. Это удобный редактор кода, разработанный Microsoft.

Эта программа включает в себя отладчик, подсветку синтаксиса и подсказки при написании различных тегов или методов.

Основными языками для создания сайта были HTML5, CSS3, JavaScript.

HTML – язык структурной разметки для создания Web-страниц. Это не язык программирования, это язык разметки. Он отвечает за расположение элементов на странице с помощью тегов. В этом языке отсутствуют логические функции, поэтому вы не сможете в нем произвести арифметические подсчёты. Аббревиатура HTML – Hyper Text Markup Language.

CSS (Cascading Style Sheets – каскадные таблицы стилей) – язык описания внешнего вида документа. CSS предназначен для работы с шрифтами, таблицами, картинками и т.д. Благодаря этому языку мы можем сделать дизайн сайта таким, каким хотим. Все ограничивается лишь фантазией.

JavaScript – язык скриптов. Скрипт представляет собой набор инструкций, которые не требуют компиляции перед запуском. Код JS интерпретируется браузером во время загрузки веб-страницы. Благодаря этому языку мы можем наследовать элементы, управлять объектами на странице, например удалить первый элемент при нажатии на второй элемент. Этот язык позволяет создавать анимации, которые нельзя сделать в CSS.

XML – расширяемый язык разметки. XML разрабатывался как язык с простым формальным [синтаксисом](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BD%D1%82%D0%B0%D0%BA%D1%81%D0%B8%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), удобный для создания и обработки документов как программами, так и человеком, с акцентом на использование в Интернете. Язык является расширяемым, поскольку не влияет на разметку страницы.

Для создание макетов было выбрано приложение Figma, который представлен на рисунке 1.5

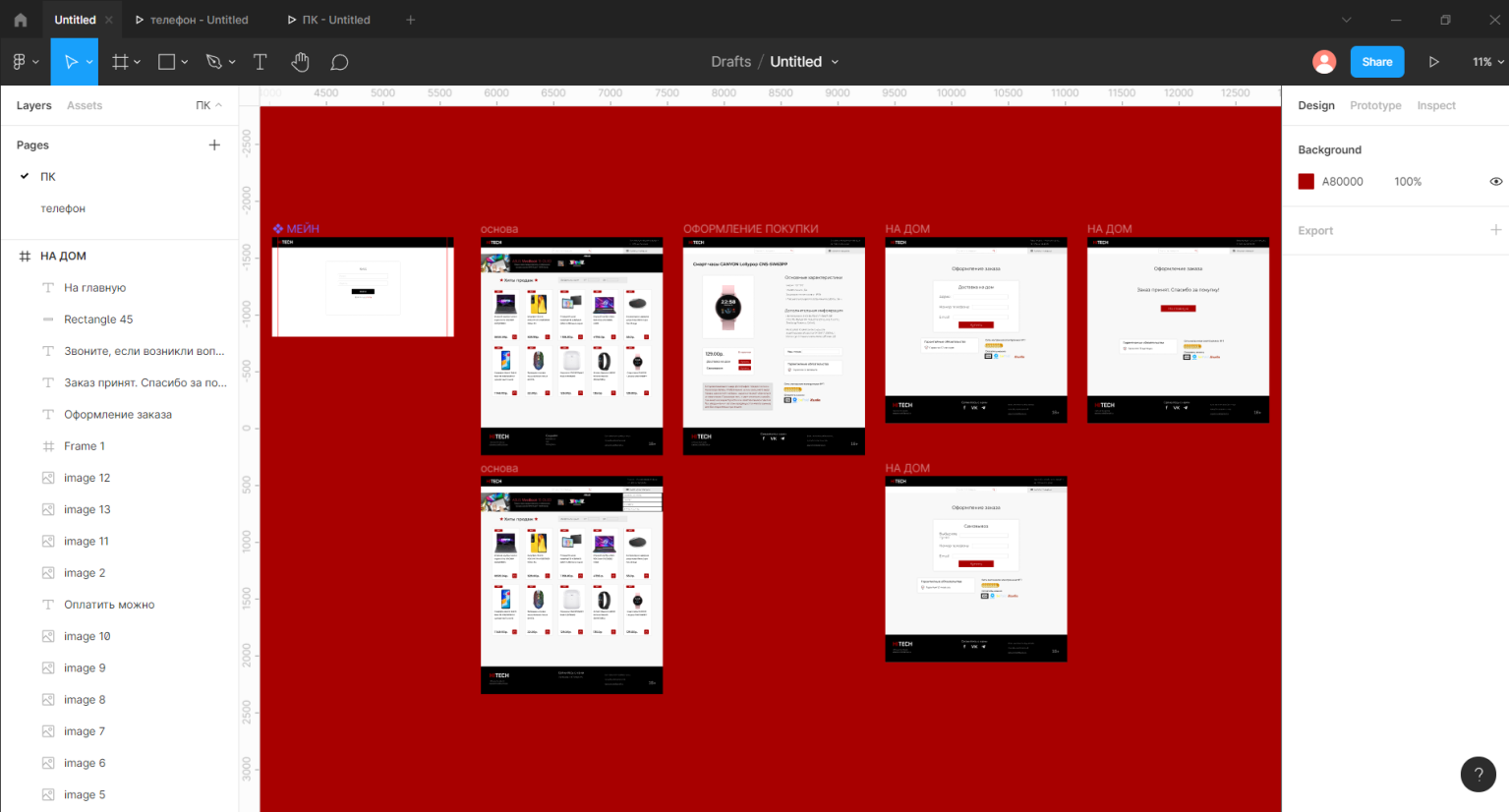


Рисунок 1.5 – Figma

Это приложение позволяет создавать предварительный вид сайта, которому мы будем следовать на протяжении всей курсовой работы. Основными функциями этого приложения являются предварительный показ сайта, работа с текстом и картинками, создание векторных изображений. В Figma очень удобное расположение кнопок, что позволяет наслаждаться процессом создания макета. Так же здесь можно создавать кнопки, которые буду переключать вас между фреймами в предварительном показе. Удобно еще то, что в Figma присутствует возможность просматривания CSS свойств элементов. Это сильно облегчает верстку.

# 1.5 Вывод

При разработке макета использовался Figma, который очень хорошо подошел для этой роли. С помощью HTML создана структура сайта. CSS позволил нам создать красивый и адаптивный дизайн. Java Script дал нам возможность создания анимации, добавляя функциональность отдельным элементам. В итоге должен получиться сайт с адаптивной версткой. Дизайн должен быть максимально простым и в то же время запоминающимся.

# 2 Проектирование страниц веб-сайта

# 2.1 Выбор способа верстки

Сайт будет верстаться блочным способом. Этот способ наиболее актуальный на сегодняшний день. Каждый блок – элемент, в который можно вкладывать другие элементы. Блок и его содержание можно позиционировать, стилизовать при помощи CSS. Достоинством этого способа является простота кода, который легко читается. Так же благодаря неограниченности вложения элементов их можно группировать в другие блоки и работать над ними. К примеру вы хотите при нажатии на кнопку скрыть несколько элементов, но для каждого элемента писать скрипт долго. Вместо этого можно все нужные элементы вложить в другой элемент и создавать скрипт уже для общего элемента. Недостатков практически нет. Элементы div и span поддерживаются даже в старых браузерах. Блоки легко адаптировать, скрывать и изменять.

# 2.2 Выбор стилевого оформления

Сайт будет стилизован по идее минимализма – убрать все лишнее и выразить самое нужное. Для цветовой палитры я выбрал 3 цвета: черный(#000000), белый(#FFFFFF) и красный(#A80000). Эти цвета хорошо сочетаются. Малое количество цветов позволяет избавиться от информационного шума. Акцентный цвет на странице делает его запоминающимся и при вспоминании этого сайта сразу возникает ассоциация с акцентным цветом. Задний фон белый – это послужит лучшему выделению основной информации.

# 2.3 Структура сайта

Структура сайта будет представлена на рисунке 2.1

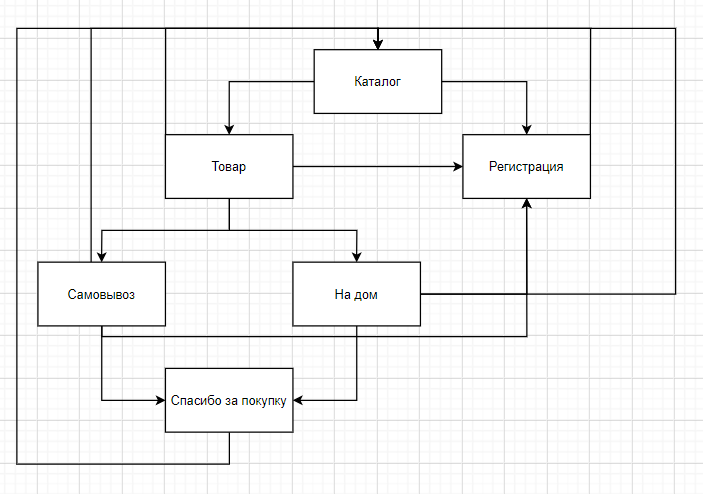


Рисунок 2.1 - структура сайта

В странице ,,Регистрация,, присутствует форма заполнения логина и пароля для входа.

В странице ,,Каталог,, представлен список товаров, меню категорий, а так же фильтрация по ценам. Здесь при нажатии на товар будет переход на страницу ,,Товар,,

В странице “Товар” присутствует информация о товаре, цена и две кнопки для оформления “Заказ на дом,, либо же “Самовывоз,,

В странице “Заказ на дом,, форма для заполнения информации касаемо покупателя. Здесь есть форма почты, номера телефона, адреса для доставки.

В странице ,,Самовывоз” форма для заполнения почты, номера телефона, выбора пункта самовывоза.

В двух последних страницах присутствует кнопка покупки и перехода на страницу ,,Благодарность за покупку,, В этой странице присутствует кнопка возврата на главную страницу.

“Footer” содержит в себе контактные данные для обращения, а так же логотип.

# 2.4 Выбор шрифтового оформления

Выбор упал на шрифт “Montserrat”. Этот шрифт один из самых популярных сегодня. Он не имеет лишних элементов на символах, что создает очень хорошую читабельность сайта. Его пример на рисунке 2.2

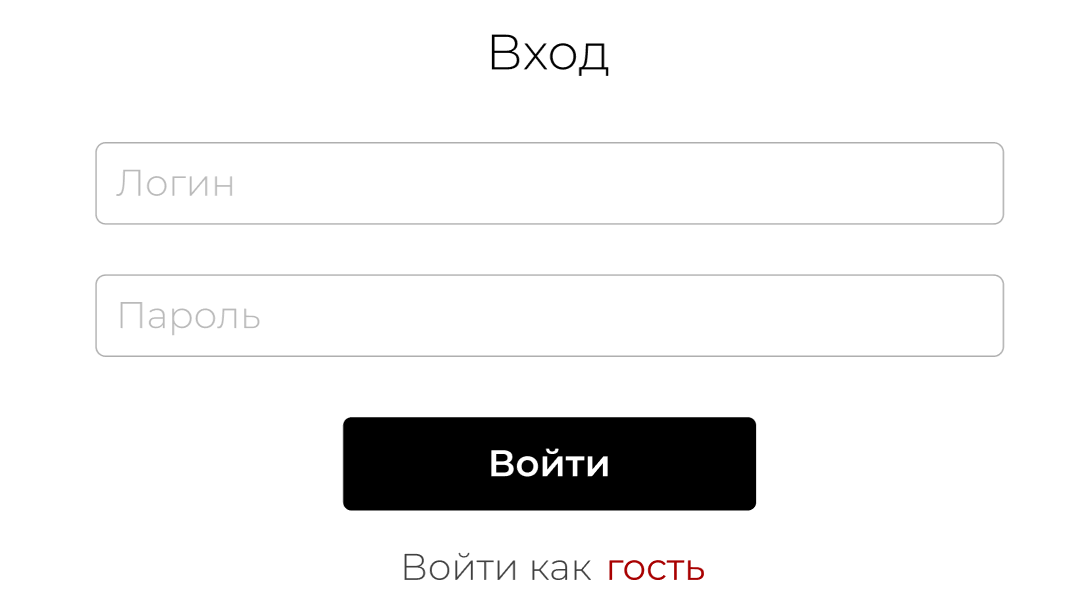


Рисунок 2.2 – поле ввода

Для логотипа был выбран шрифт “Quantico”. Этот шрифт при латинской раскладке придает символам строгую структуру, которая ассоциируется с технологиями, пример на рисунке 2.3



Рисунок 2.3 – логотип

Так же

# 2.5 Разработка логотипа

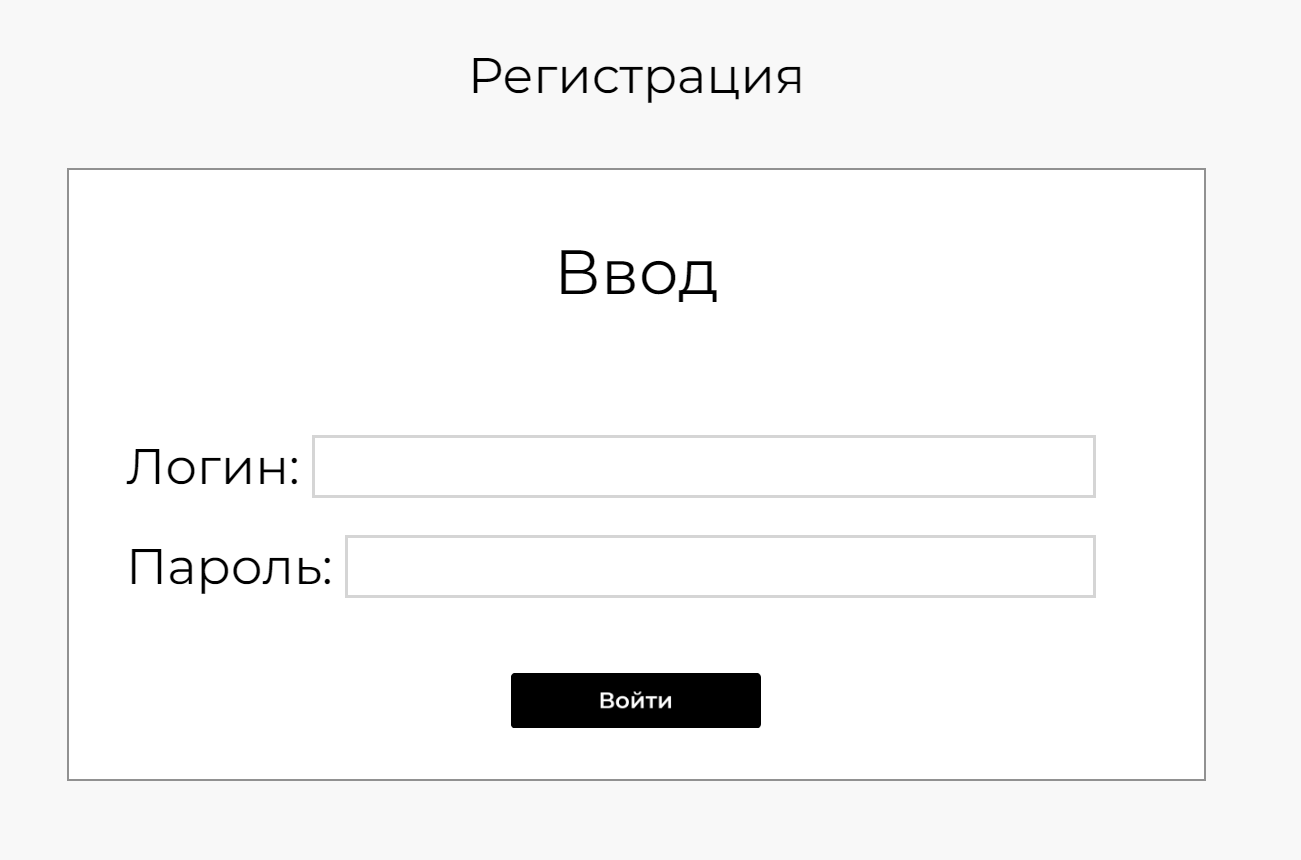
Логотип сайта напрямую связан со сферой интернет-магазина, а именно технологии. HITECH - (HIGHT TECHNOLOGY) . Его пример на рисунке 2.4

  
Рисунок 2.4 - логотип

HI выделен красным, а TECH белым, чтобы было понятно, что логотип составлен из двух слов. Логотип окрашен акцентными цветами, которые будут преследовать пользователя на протяжении всего присутствия в сайте. Дизайн логотипа максимально строгий, что ассоциируется с конструкциями различных гаджетов, а именно угловатостью.

# 2.6 Разработка пользовательских элементов

На первой странице присутствует форма, показанная на рисунке 2.5

Рисунок 2.5–поле ввода

Две строки будут являться формами для ввода Логина и Пароля соответственно.

Кнопка Войти будет переводить на главную страницу сайта с товарами. Так же, если не хотите вводить данные, то ниже существует кнопка “гость”, которая так же переводит на главную страницу. На этой же странице есть меню с категориями, на рисунке 2.6

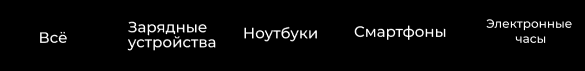


рисунок 2.6 – сортировка по категориям

Есть список товаров на выбор, на рисунке 2.7

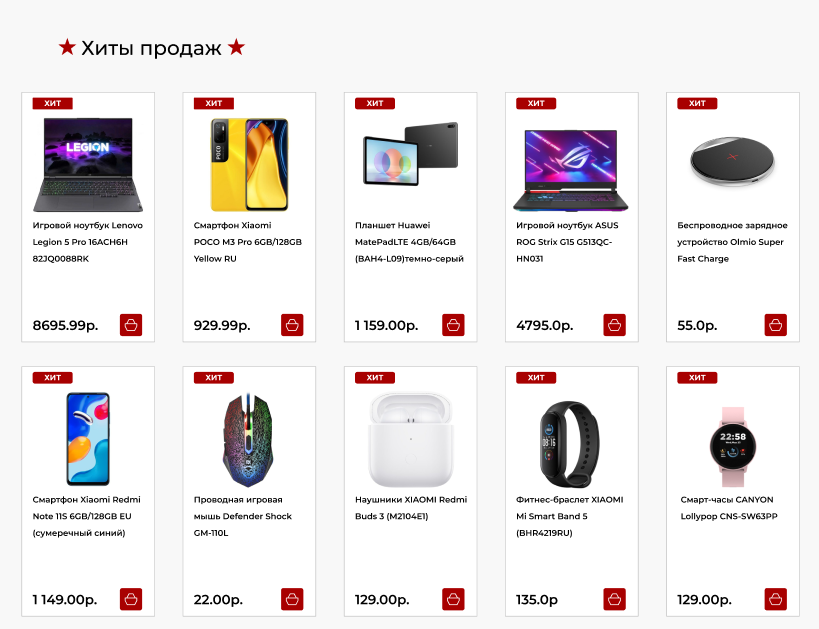


Рисунок 2.7 - каталог

На третьей странице с подробной информацией о товаре добавлена кнопка перехода к оформлению покупки, на рисунке 2.7.1

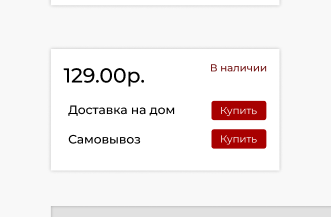


Рисунок 2.7.1 – выбор оформления покупки

На странице с оформлением заказа присутствует поле ввода информации для доставки товара на дом либо же самовывоза, показанный в рисунке 2.7.2

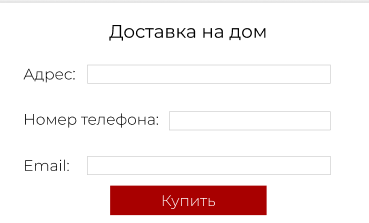


Рисунок. 2.7.2 – поле ввода

2.6 Прототипы

Прототип сайта необходим для понимания структуры, где какие картинки будут находиться и что они означают. Благодаря прототипам можно объяснить человеку, который будет верстать сайт, что и куда нужно вставлять. Пример прототипа представлен в приложении 1.

# 2.7 Макет

Макет сайта – это внешний вид сайта до начала его верстки. Обычно макет создают на основе прототипа и используют для того, чтобы брать с него информацию о блоках, картинках, тексте, которые верстальщик будет создавать при написании веб-страницы. Так же макет предназначен для ознакомления с веб-страницей перед ее использованием или же вёрсткой. Пример макета веб-страницы представлен в приложении 2.

# 2.8Вывод

Этот курсовой проект нацелен на продвижение такой идеи, как интернет-магазин. Эта сфера очень сильно помогла людям при коронавирусной инфекции тем, что позволил людям сохранять здоровье и не выходить из дома. Был разработан уникальный адаптивный дизайн. Идеей интернет-магазина я вдохновился потому, что сам недавно являлся её пользователем. Мне показалось интересным самому реализовать подобный проект.

Для начала был создан макет сайта на Figma, который помогает представить будущее сайта и облегчает верстку благодаря свойству программы показывать CSS. Основной задачей макета является создания плана сайта, как он выглядит, какая у него цветовая палитра, как расположены элементы и т.д. Это последний этап перед началом верстки сайта, с него мы начнем копировать элементы в HTML.

Основной задачей сайта является его адаптивность и удобность в использовании, а так же его производительность.

# 3. Реализация структуры сайта

# 3.1 Структура HTML документа

Для написания структуры документа был использован HTML5. Основная структура HTML кода представлена в приложении 3. Он является стандартом для построения структуры сайтов. Структура показана в листинге 3.

|  |
| --- |
| <body>  <header></header>  <div class=”relkama\_cats”></div>  <div class="center"></div>  <footer></footer>  </body> |

Листинг 3 – Структура HTML

Структура документа состоит из тегов header, reklama\_cats, center, footer.

В теге header находится логотип магазина, номер для служебной поддержки, введенный логин, а так же кнопка регистрации.

В теге reklama\_cats расположена реклама и меню сортировки товаров по категориям.

В теге center расположен каталог товаров

В теге footer расположен логотип компании, ссылки на социальные сети, возрастное ограничение.

На странице с товаром структура такая же, только в теге center вместо каталога стоит информация о выбранном товаре, а так же отсутствует блок с категориями.

Страницы с оформлением заказа и регистрацией отличаются тем, что в блоке “center” присутствует форма для заполнения, а блок “reklama\_cats” отсуствует.

# 3.2 Использование JavaScript

Через JS мы создали верификацию пароля при регистрации. Она будет выглядеть как на рисунке 4.2, а код будет в листинге 3.1

|  |
| --- |
| myInput.onkeyup = function() {  var lowerCaseLetters = /[a-z]/g;  if(myInput.value.match(lowerCaseLetters)) {  letter.classList.remove("invalid");  letter.classList.add("valid");  }else{  letter.classList.remove("valid");  letter.classList.add("invalid");}  var upperCaseLetters = /[A-Z]/g;  if(myInput.value.match(upperCaseLetters)) {  capital.classList.remove("invalid");  capital.classList.add("valid");  }else{capital.classList.remove("valid");  capital.classList.add("invalid");  }  var numbers = /[0-9]/g;  if(myInput.value.match(numbers)) {  number.classList.remove("invalid");  number.classList.add("valid");} else {  number.classList.remove("valid");  number.classList.add("invalid");}  if(nickname.value.length>0){  nick.classList.remove("invalid");  nick.classList.add("valid");  } else {nick.classList.remove("valid");  nick.classList.add("invalid");  }  if(myInput.value.length >= 8) {  length.classList.remove("invalid");  length.classList.add("valid");  } else {  length.classList.remove("valid");  length.classList.add("invalid");  }  if(myInput.value.length >= 8 && myInput.value.match(numbers) && myInput.value.match(upperCaseLetters) && myInput.value.match(lowerCaseLetters) && nickname.value.length>0 ){  document.querySelector(".input\_vhod").style.display = "block";}  else{document.querySelector(".input\_vhod").style.display = "none}} |

Листинг 3.1 – реализация верификации

Мы создали условия, что если в пароле присутствуют буквы a-z(строчные),A-Z(заглавные), 0-9(цифры), введено 8 символов, то появляется кнопка входа, иначе кнопки входа не будет.

С помощью JS мы реализовали переход из главной страницы на страницу с товаром и загрузку в нее той информации, которая нам нужна для определенного выбранного товара. Пример кода в листинге 3.1

|  |
| --- |
| var ing = params.get("name");  onclick="location.href='./tovar.html?idenf=0&name='+ing"; |

Листинг 3.1 – реализация перехода на страницу с товаром

Url выглядит так: “./tovar.html?idenf=0&name=Jora”

Из url ссылки мы берем ключ с именем “name” и “idenf” которые равны “jora” и “0” соответственно. Эти ключи мы берем и вставляем в ссылку кнопки, которая будет выполнять функцию перехода на страницу с товаром с идентификатором “0” и введённым никнеймом “Jora”

На странице с каталогом присутствует очень много блоков с различными товарами, что оказалось неудобным для занесения информации через HTML документ, поэтому было решено создать структуру сайта, после чего с помощью языка JavaScript занести всю информацию в каждый из блоков. Пример массива Json на листинге 3.2

|  |
| --- |
| var tovar =[{  name:"Игровой ноутбук Lenovo Legion 5 Pro",  price:"8695.0р.",  screen:" 16.0' 2560x1600 px, IPS 165 Гц",  notification:"2021",  brend:"Lenovo",  series:"Lenovo Legion 5",  autonomy:"10ч",  dop\_info:'Дополнительная информация Изготовитель: Quarry Боровлянский с/с, 103/3-7, пом.7-57, р-н д.Дроздово',  garant\_data:"Гарантия 2 года",  photo:"./Photo/dell.jpg"} |

Листинг 3.2 – json массив

Для заноса информации по определенным блокам был использован скрипт, который представлен в листинге 3.3. Из JS файла мы берем информацию, обращаясь через имя элемента массива, в котором содержится нужная информация. Далее мы обращаемся к нужному блоку с помощью “getElementById” и пушим в него информацию, с помощью команды “innerHtml”

|  |
| --- |
| <script>  for (let i = 0; i < 10; i++) {  let figure = document.getElementById("figure\_" + [i]);  figure.innerHTML = tovar[i].name;  }  </script> |

Листинг.3.3 – занос информации по блокам

Так же нужно было реализовать загрузку информации на страницу с товаров в зависимости от товара. Для этого я использовал ключи в URL-ссылке. После основной ссылки на сайт нужно поставить “?” и после него написать “ключ=значение”, а на нужной странице мы достаем значение по ключу и используем его в качестве индекса массива, из которого мы будем загружать информацию товара на страницу. Пример использования в листинге 3.4

|  |
| --- |
| <script>  var params = new URLSearchParams(document.location.search);  var ing = params.get("name");  </script> |

Листинг.3.4 – скрипт по использованию ключа в URL-ссылке

# 3.3 Добавление таблиц стилей Sass и CSS

Для добавления приятного визуального восприятия страницы была использована каскадная таблица стилей – CSS. Это язык разметки для HTML, XML документов. Использование CSS представлено в приложении 4.

Без использования CSS структура документа будет непонятной пользователю, а текст – нечитаемым. Для того, чтобы воспользоваться CSS, нужно сначала построить структуру документа, или же определенного тега, а потом уже использование CSS. С помощью CSS можно менять шрифты, размеры блоков, их цвет, задавать им рамки, тени, анимации.

Адаптив на странице я реализовал с помощью Flex верстки. Пример блока и его код, содержащего flex свойство приведен в рисунке 3.5 и листинге 3.6 соответственно.

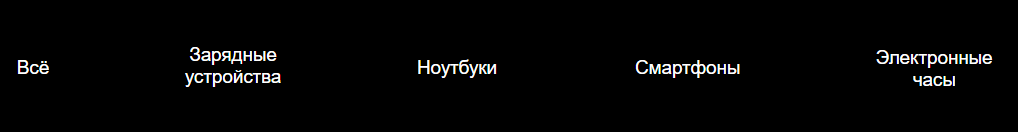


Рис.3.5 – элемент, содержащий flex верстку

|  |
| --- |
| .cats {  display:flex;  justify-content: space-between;  align-items: center;  font-family: 'Montserrat', sans-serif;  padding-left: 9%;  padding-right: 10%;  margin-top: 35px;  } |

Листинг 3.6 – код flex блока

Sass (Syntactically Awesome Style Sheets) - упрощает написание CSS-кода и позволяет динамически им манипулировать. Это отличный способ создания более функциональных CSS-кодов, который позволяет ускорить выполнение ежедневной работы веб-разработчиков и дизайнеров.

# 3.4 Использование стандартов XML (SVG)

При разработке сайта использовался XML для заноса информации в определенные блоки, а так же картинки для кнопки регистрации.

XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов как программами, так и человеком, с акцентом на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих, какие атрибуты и элементы могут входить в состав других элементов.

С помощью XML мы вставили картинку для кнопки регистрации и текст для предупреждения того, что товар на фото может не совпасть с реальным. Скрипт, реализовывающий это представлен в листинге 3.7

|  |
| --- |
| <script>  var xmlhttp = new XMLHttpRequest();  var path = "./xml/xml.xml";  xmlhttp.open("GET", path, true);  xmlhttp.send();  xmlhttp.onreadystatechange = function () {  if (this.status == 200) {  var xmlDoc = this.responseXML;  var warning\_xml=xmlDoc.getElementsByTagName("warning"); [0].childNodes[0].nodeValue;  var img\_xml=xmlDoc.getElementsByTagName("img") [0].childNodes[0].nodeValue;  let img=`<input type="image" id="registr" src="${img\_xml}"  onclick="location.href='./registration.html'" alt="" />`;  document.querySelector(".registration").innerHTML=img;  console.log(img\_xml);  let warning=`<div class="warning">${warning\_xml}</div>`;  document.querySelector(".musor").innerHTML=warning;  }}  </script> |

Листинг 3.7 – использование XML как БД

Объект new XMLHttpRequest()даёт возможность из JavaScript делать HTTP-запросы к серверу без перезагрузки страницы. C помощью функций open()и send() мы запрашиваем и отсылаем запрос соответственно. И если код ответа сервера 200, то загружаем информацию в HTML документ.

SVG – векторная графика, или же изображение, состоящее из векторов. Она активно используется в сайтах, где очень важно качество всех изображений, ведь вектора не меняются в качестве при уменьшении или увеличении. Был использован SVG на рисунке 3.8



Рисунок 3.8 – SVG изображения

Использование SVG изображение продемонстрировано в приложении 7.

# 3.5Вывод

При построении структуры сайта использовался язык разметки HTML. Для придачи дизайна сайту использовался CSS, а для загрузки информации на страницу XML и JavaScript/JSON. В качестве SVG были использованы изображения для связи в соцсетях.

# 4 Тестирование сайта

# 4.1 Адаптивный дизайн веб-сайта

В современном мире количество пользователей сети интернет растет каждый день. Соответственно среди всех сайтов нужно выделяться, чтобы привлечь к себе как можно больше пользователей. Одним из показателей привлекательности сайта является его адаптивность, ведь пользователь сразу же выйдет из сайта, если у него картинки и текст будут летать по всей странице.

Для адаптивности сайта я использовал @media запросы. Это работает следующим образом: я задаю максимальную ширину в пикселях, в пределах которых будет соблюдаться определенная верстка определенного элемента. Страница до @media запроса представлена на рисунке 4

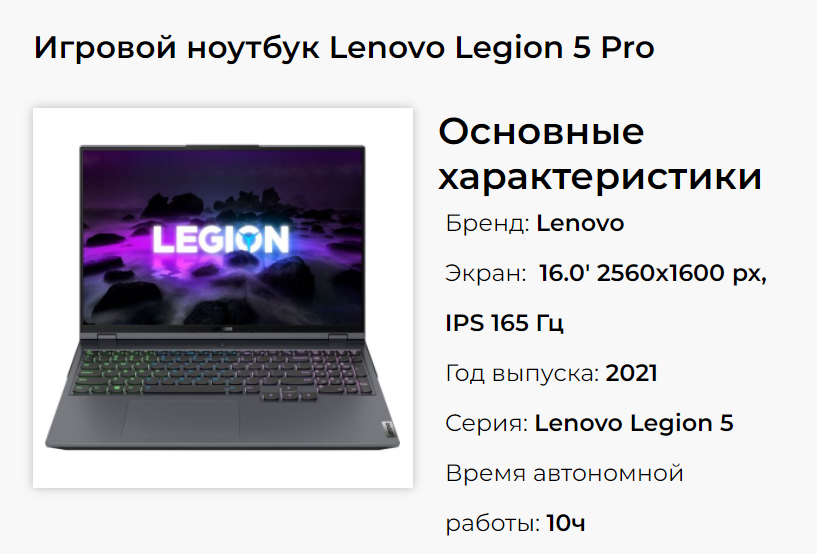


Рис.4 – компьютерная версия

Страница после @media запроса представлена на рисунке 4.1

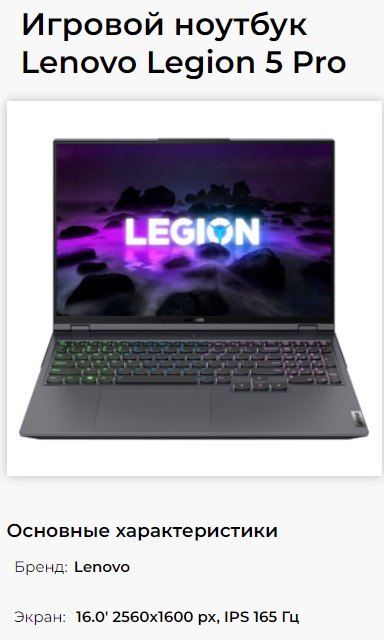


Рисунок.4.1 – мобильная версия

При условии, что ширина экрана будет меньше 630px, элементы будут расположены вертикально, а если больше, то горизонтально.

# 4.2 Кроссбраузерность веб-сайта

Кроссбраузерность сайта – это совместимость сайта со всеми популярными браузерами. Условием кроссбраузерности будет то, что сайт будет отображаться правильно во всех браузерах. Каждый браузер имеет свои особенности в отображении страницы с CSS версткой, ведь у каких-то браузеров какие-то свойства работают, а в каких-то браузерах – нет. Пример работы веб-страницы в браузере Explorer на рисунке 4.2

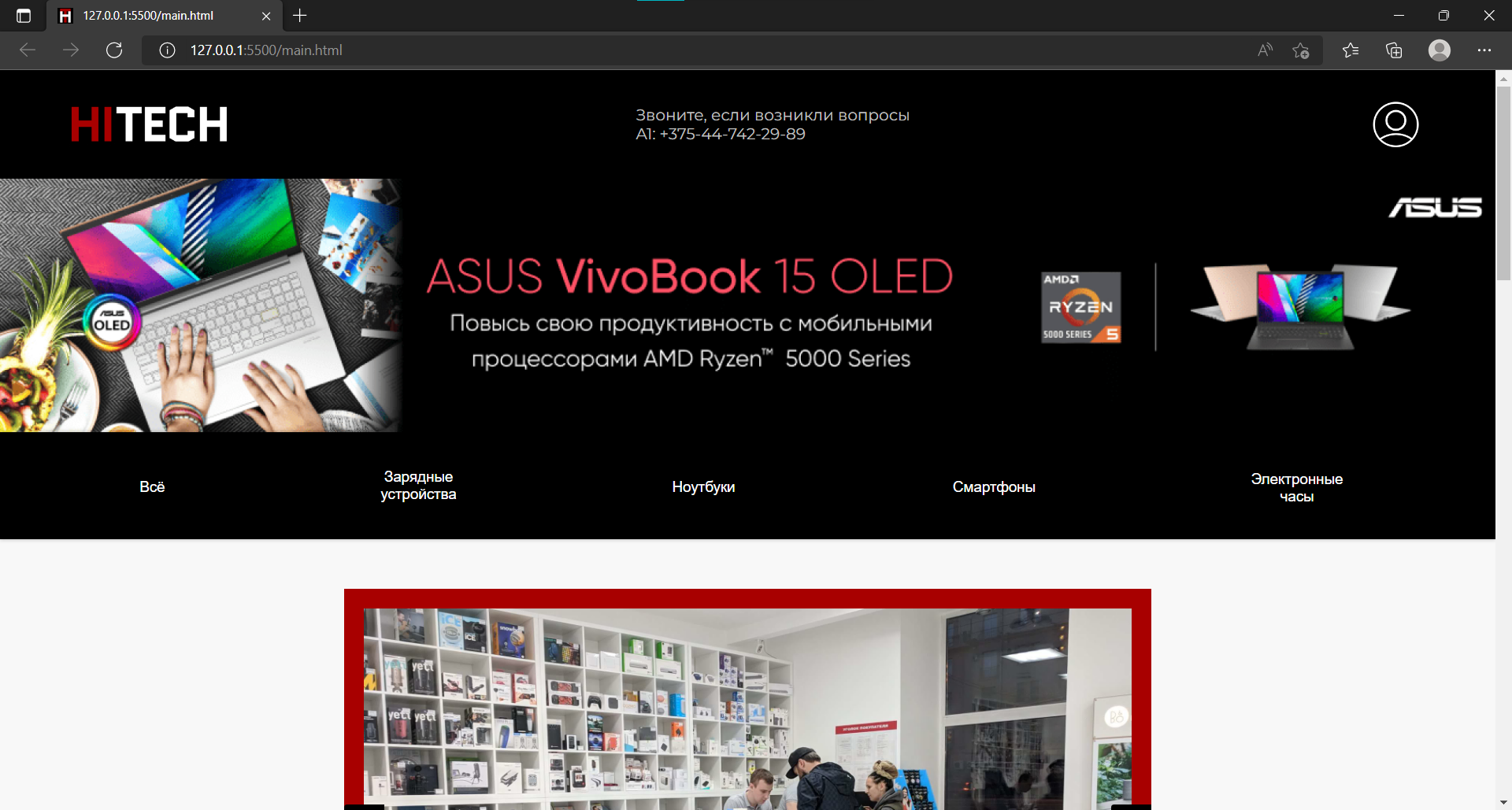


Рисунок 4.2 – веб-страница в Explorer

Пример работы веб-страницы в браузере Opera на рисунке 4.3

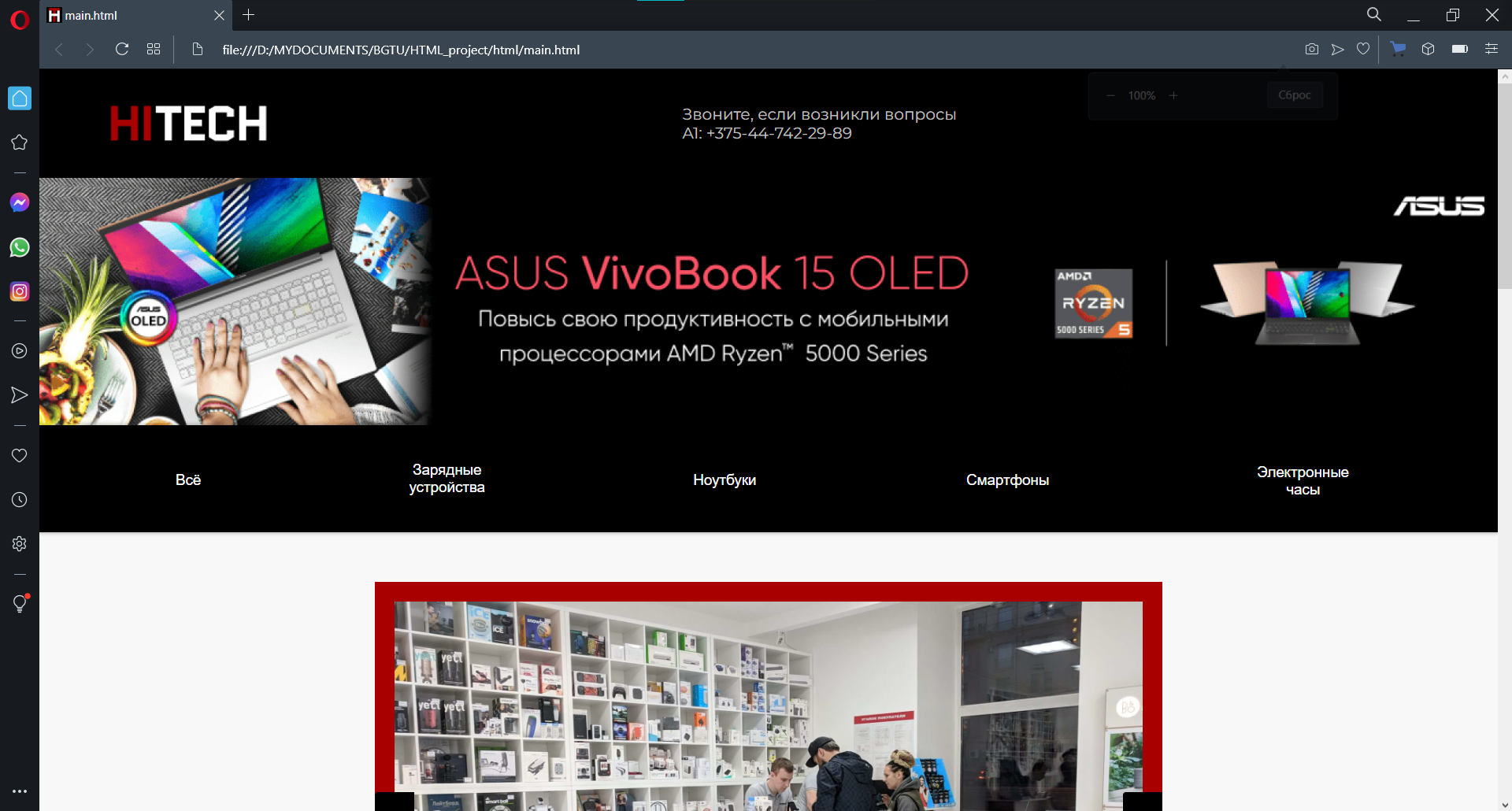


Рисунок 4.3 – веб-страница в Opera

Пример работы веб-сайта в браузере Yandex на рисунке 4.4

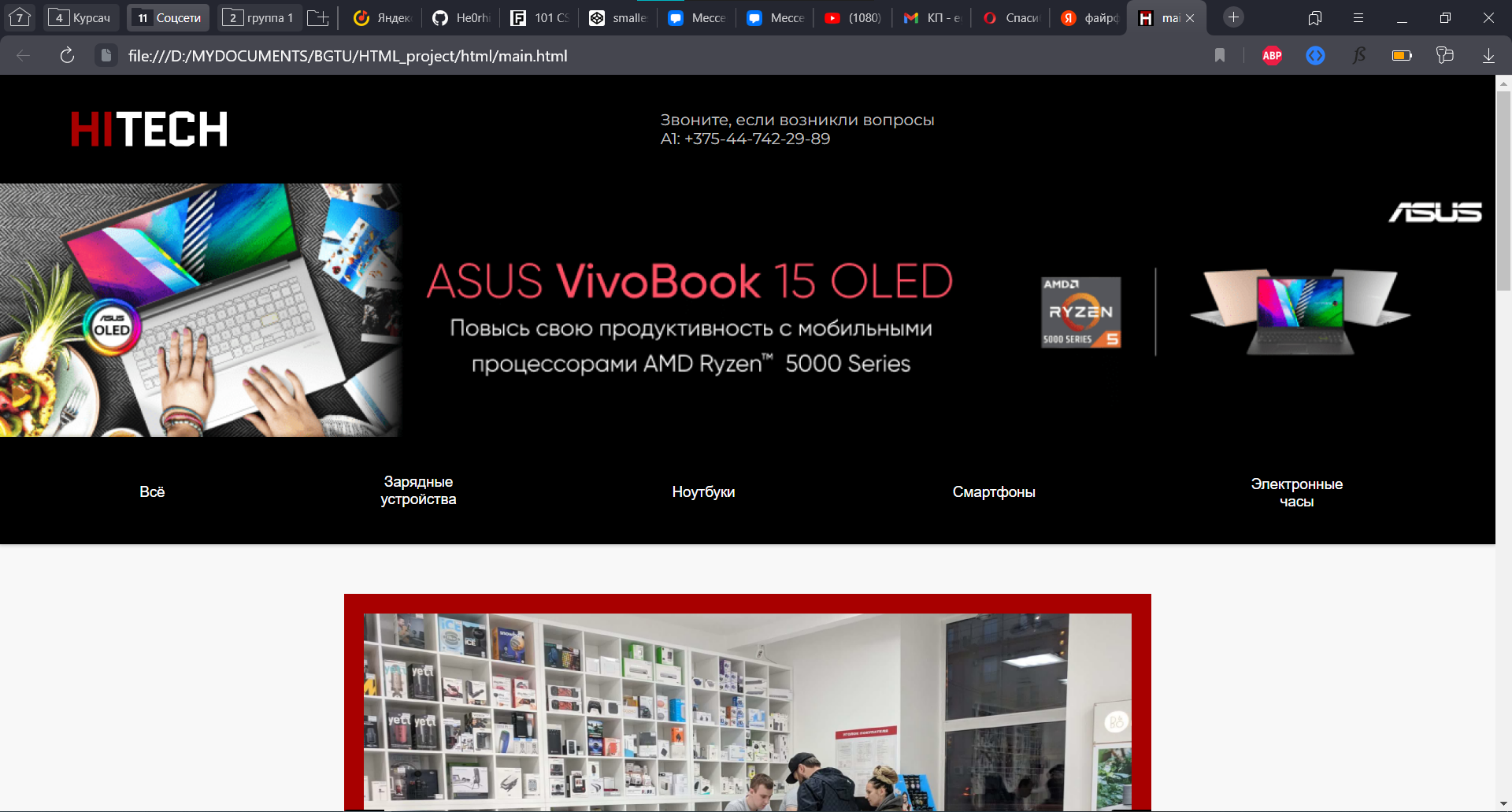


Рисунок 4.4 – веб-страница в Yandex

В этом разделе была показана кроссбраузерность веб-страницы на примере браузерах Yandex, Opera, Explorer.

# 4.3 Руководство пользователя

При входе на главную страницу пользователю будет дана возможность в хедере, а именно кнопка в правом верхнем углу, перейти на страницу с регистрацией, которая будет представлена на рисунке 4.2

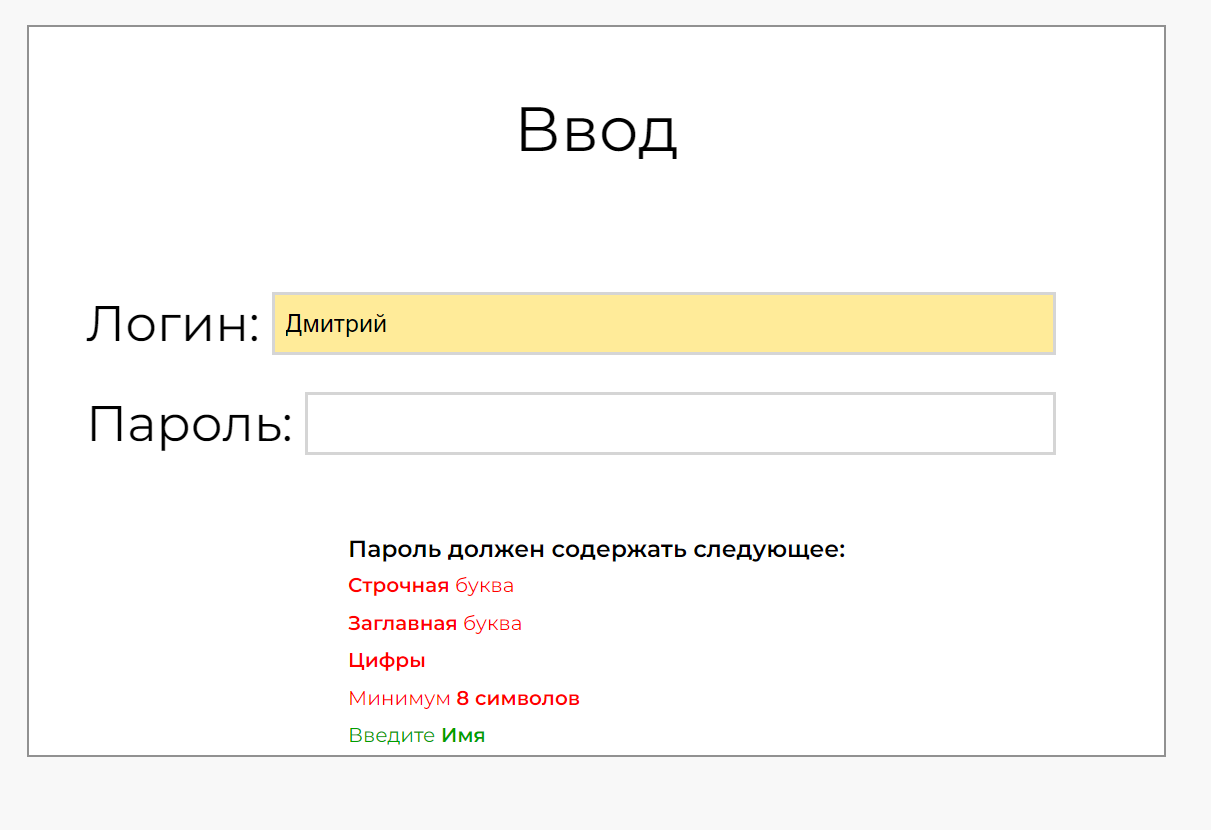


Рис. 4.2 – поле ввода

На странице регистрации пользователю обязательно нужно будет ввести логин и пароль, в котором обязательно должны быть заглавная буква, строчная, цифра и введено 8 символов. Только после верификации пароля появится кнопка “Вход”

На главной странице ниже пользователю будет представлена возможность сортировать элементы по категориям. В разделе “Хиты продаж” пользователь может посмотреть цену, название и фото предложенных товаров. При нажатии на название товара либо красной кнопки, представленной на рисунке 4.2, пользователю будет дана возможность перейти на страницу с информацией товара.

На странице с товаром пользователю будет предоставлена подробная информация по товару, а так же возможность оформления покупки методом “Самовывоз” или “Доставка на дом”.

На страницах по оформлению покупки пользователю нужно будет ввести Адрес/Пункт, номер телефона и почту.

# 4.4 Вывод

В процессе создания сайта мы научились придавать сайту адаптивность и кроссбраузерность, которая была проверена в браузерах Yandex, Opera, Explorer.

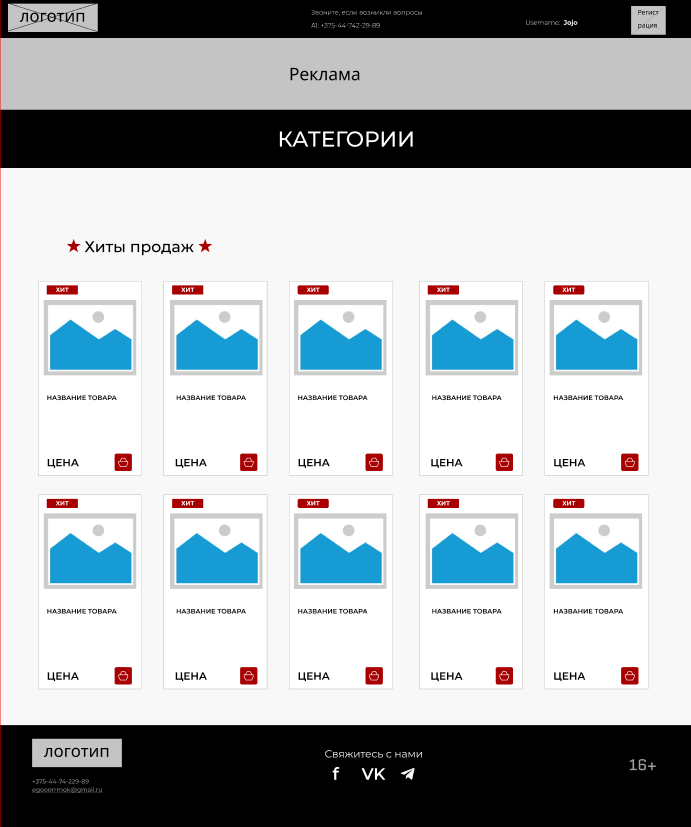
Адаптивность была реализована с помощью @media запросов.

# Заключение

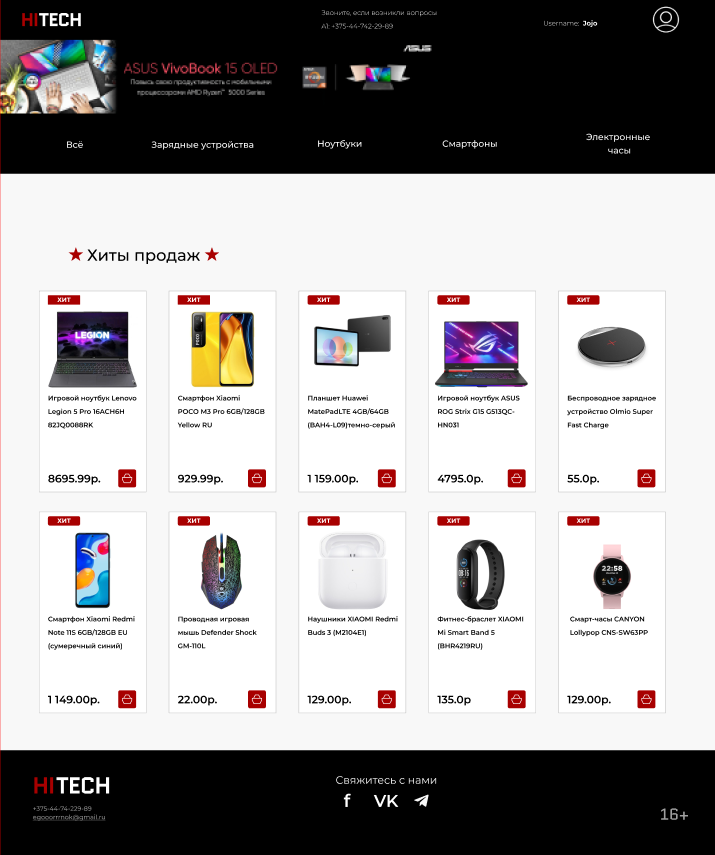
Главная цель курсового проекта – создание адаптивного, кроссбраузерного интернет магазина гаджетов “HITECH”. Сайт создан для оформления покупок и указания данных для доставки на дом либо же для самовывоза. Макет сайта был создан в приложении Figma. Адаптив был создан с помощью @media запросов, благодаря которым мы можем задавать стили элементам при определенной ширине экрана

# Приложение

**Приложение 1 – прототип веб-страницы**

****

Приложение 2 – макет веб-страницы



Приложение 3 – Листинг HTML

|  |
| --- |
| <body>  <div class="logo"><input type="image"  id="logotip" onclick="Main()"  src=".\Photo\hitech.png" alt=""></input></div>  <div class="zvonite">  <p>Звоните, если возникли вопросы</br>  A1: +375-44-742-29-89 </p>  </div>  <div class="login">Username:  <b class="login\_name"></b></div>  <div class="registration"></div>  </header>  <div class="reklama\_cats">  </div>  <div class="slider\_main">  </div>  <div class="center">  </div>  <footer>  <div class="logo\_footer">  </div>  <div class="social">  </div>  <div class="vozrast">  </div>  </footer>  </body> |

Приложение 4 – Листинг CSS

|  |
| --- |
| body{  background-color: #F8F8F8;  margin: 0%;  }  .adaptivn-slayder{  max-width: 90%;  position: relative;  margin: auto;  height: 100%;  margin-bottom: 15px;  padding: 20px;  background-color: #A80000;}    .adaptivn-slayder, .parag img {  object-fit: cover;  width: 100%;  height: 100%;}  .adaptivn-slayder .prev, .adaptivn-slayder .next {  cursor: pointer;  position: absolute;  top: 50%;  margin-top: -22px;  padding: 2%;  color: #ffffff;  transition: 0.8s ease;}  .slider\_main{  width: 60%;  margin-left: auto;  margin-right: auto;  margin-top: 50px;}  .adaptivn-slayder .next {  right: 0;  border-radius: 3px 0 0 3px;}  .prev {  left: 0;  border-radius: 3px 0 0 3px;}  .prev,.next{  background-color: black;}  .adaptivn-slayder .prev:hover,  .adaptivn-slayder .next:hover {  background-color: rgba(255, 255, 255);  color: black;}  a{width: 100%;}  #photo {width: 100%}  .osnova{height: auto;  width: 100%;}  header{  display: flex;  width: 100%;  top: 0;  height: 110px;  left: 0;  background-color:rgb(0, 0, 0);  padding: 0;  justify-content: space-between;  align-items: center;}  .zvonite{  color: rgb(193, 193, 193);  font-family: 'Montserrat', sans-serif;}  #logotip{  height: 100px;  margin-left: 35%;  width: 100%;}  .login{  color: rgb(223, 223, 223);  font-family: 'Montserrat', sans-serif;  padding: 10px;}  .registration{  height: 50px;  margin-right: 5%; }  #registr{  height: 100%;}  #text\_find{  width: 95%;  margin-left: 3%;}  .button{  font-size: 15px;}  .black\_reklama{  width: 100%;  height: 100%;  background-color: black;  display: flex;}  .reklama{  height: 100%;  width: 100%;}  .category{  width: 100%;  height: 109px;  text-align: center;  display: inline-block;  justify-content: space-between;  align-items: center;  box-shadow: 0px 0px 4px 1px rgba(0, 0, 0, 0.25);  background-color: black;}  .cats{  display: flex;  justify-content: space-between;  align-items: center;  font-family: 'Montserrat', sans-serif;  padding-left: 9%;  padding-right: 10%;  margin-top: 35px;}  .cat\_all{  font-weight: 300;  border: none;  background-color: black;  border-radius: 4px;  padding: 5px;  height: 40px;  color: white;}  .cat\_zarad{  border: none;  background-color: black;  border-radius: 4px;  padding: 2px 3px 40px 3px;  height: 40px;  color: white;}  .cat\_notebook{  border: none;  background-color: black;  border-radius: 4px;  padding: 5px;  height: 40px;  color: white;}  .cat\_smart{  border: none;  background-color: black;  border-radius: 4px;  padding: 5px;  height: 40px;  color: white;}  .cat\_clock{  border: none;  background-color: black;  padding: 3px 3px 35px 3px;  height: 40px;  color: white;}  .cat\_all:hover{  color: rgb(179, 179, 179);}  .cat\_zarad:hover{  color: rgb(179, 179, 179);  }  .cat\_clock:hover{  color: rgb(179, 179, 179);  }  .cat\_notebook:hover{  color: rgb(179, 179, 179);  }  .cat\_smart:hover{  color: rgb(179, 179, 179);  }  .hide{  display: none;}  .center{  font-family: 'Montserrat', sans-serif;  background-color: #F8F8F8;  margin-top: 3%;  height: auto;}  .sena{  background-color: #EAEAEA;  width: 600px;  height: 50px;  display: flex;  justify-content: space-between;  align-items: center;  max-width: 700px;}  .do{  margin-right: 3%;  display: inline-block;}  .shoose\_sena{  margin-left: 3%;}  .ot{  margin-left: 3%;  margin-right: 3%;  display: inline-block;}  #ot{  width: 60%;  padding-left: 3px;}  #do{  width: 60%;  padding-left: 3px;}  .hit{  margin-top: 5%;  margin-left: 8%;}  #hit{  width: 40%;}  .nadtovar{  display: flex;  margin-right: 5%;}  .tovar\_full{  display: flex;  flex-wrap: wrap;  column-gap: 5%;  margin-left: 10px;  margin-right: 10px;  font-weight: 600;  justify-content: center;}  #tih{width: 40%;}  .one{  border: 1px solid #BBBBBB;  background-color: white;  padding-top: 15px;  padding-left: 10px;  padding-right: 8px;  height: 350px;  width: 220px;  margin-top: 8%;}  a{  text-decoration: none;  color: black;}  a:hover{  color: #A80000;  text-shadow: 0px 3px 0px 0px #00000040 inset;}  #tovar\_photo{  width: 200px;  height: 200px;}  .musor{  margin-top: 5%;  width: 87%;  margin-left: 5%;  margin-right:5%;  display: flex;  justify-content: space-between;  align-items: center;}  .warning{  width: 100%;  background-color: #E1E1E1;  padding: 1%;  box-shadow: 0px 0px 7px 0px #00000040 inset;  margin-right: 30%;}  #tovar{  width: 100%;}  .buy{  display: flex;  justify-content: space-between;  align-items: center;  margin-top: 5px;  background: url(./Photo/buy.png);}  #buy{  width: 40px;  height: 40px;  border: none;}  #buy:hover{  box-shadow: 0px 0px 5px 2px #00000040 ;}  .anime{  transform:scale(0);  opacity: 0;}  .rubls{  font-size: 25px;}  footer{  height: auto;  width: 100%;  background-color: black;  color: white;  font-family: 'Montserrat', sans-serif;  margin-top: 10%;  display: flex;  justify-content: space-between;}  .logo\_footer{  padding-bottom: 7%;  width: 16%;  font-family: 'Montserrat', sans-serif;  margin-left: 5%;}  .logo\_f{  width: 100%;}  .nomer{  margin-left: 5%;  color: #a7a7a7;}  .gmail{  margin-left: 5%;  color: #a7a7a7;}  #logotip\_footer{  width: 100%;  margin-top: 2px;}  .social{  text-align: center;  height: 50%;  width: 30%;  margin-top: 5%;}  svg {fill:black; float:left}  .soc{  width: 50%;  display: flex;  justify-content: center;  margin-top: 1%;  margin-left: 25%;}  #facebook{  width: 23%;}  #telega{  width: 23%;  margin-left: 10%;}  #vk{  width: 23%;  margin-left: 10%;}  .vozrast{  width: 7%;  margin-right: 10%;  margin-top: 5%;}  #vozrast{  width: 100%;}  #svaz{  width: 70%;} |

Приложение 5 – Листинг XML файлов

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <?xml-stylesheet href="my-style.css" type="text/css"?>  <main>  <warning>  Сопровождающие товар фотографии предоставлены производителем, отображение на них внешнего вида товара зависит от настроек экрана и может отличаться от оригинала. Производитель может изменять дизайн, технические характеристики и комплектацию изделия без уведомления об этом продавца. Уточняйте важные для Вас параметры при заказе.  </warning>  <img>.\Photo\registr.png</img>  </main> |

Приложение 6 – Листинг SVG

|  |
| --- |
| <svg viewBox="0 0 489 499" fill="none" xmlns="http://www.w3.org/2000/svg">  <path d="M488.086 1.11499H0.0858154V498.115H488.086V1.11499Z" fill="black"/>  <path d="M213.736 375.115V177.715C213.736 158.115 219.452 142.482 230.886 130.815C242.552 118.915 259.119 112.965 280.586 112.965C288.286 112.965 295.519 113.782 302.286 115.415C309.286 117.048 315.119 119.615 319.786 123.115L307.886 156.015C304.619 153.448 300.886 151.582 296.686 150.415C292.486 149.015 288.169 148.315 283.736 148.315C274.636 148.315 267.752 150.882 263.086 156.015C258.419 160.915 256.086 168.382 256.086 178.415V199.415L257.486 219.015V375.115H213.736ZM182.936 224.615V189.615H307.536V224.615H182.936Z" fill="white"/>  </svg> |