# **Введение**

Интернет — глобальная сеть компьютерных ресурсов с коллективным доступом на основе использования единой стандартной схемы адресации, предназначенная для хранения и передачи данных. Наиболее популярная система передачи данных Интернета — Всемирная паутина, World Wide Web. Она предоставляет доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету.

Впервые интернет появился в США как сеть, объединяющая компьютеры институтов, работающих на оборонную промышленность. Днем рождения интернета принято считать 29 октября 1969 года, когда между компьютерами Стэнфордского университета и Калифорнийского университета Лос-Анджелеса была установлена связь, во время которой были переданы всего три буквы — LOG.

Цель курсовой работы: разработать веб-сайт для магазина по продаже винных изделий с использованием HTML5 и XML, а также с применением CSS3.

Задачи:

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

2.Разработать макет сайта.

3.Разработать структуру веб-сайта.

4.Наполнить сайт информацией по теме.

5.Протестировать веб-сайт.

6.Разработать руководство пользователя.

# 

# **1. Обзор технических методов и программных средств разработки.**

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

Курсовой проект представляет из себя веб-сайт для магазина винных изделий. В бизнес-сфере, в частности в сфере продажи, уже существуют некоторые аналогичные решения. В этом разделе будут приведены веб-сайты существующих алкогольных магазинов.

Сайт vino&vino.by. Сайт имеет приятную для глаз цветовую палитру. Шрифт читабельный. Сайт динамический, интерактивный. Дизайн представлен на рисунке 1.1. и 1.2.

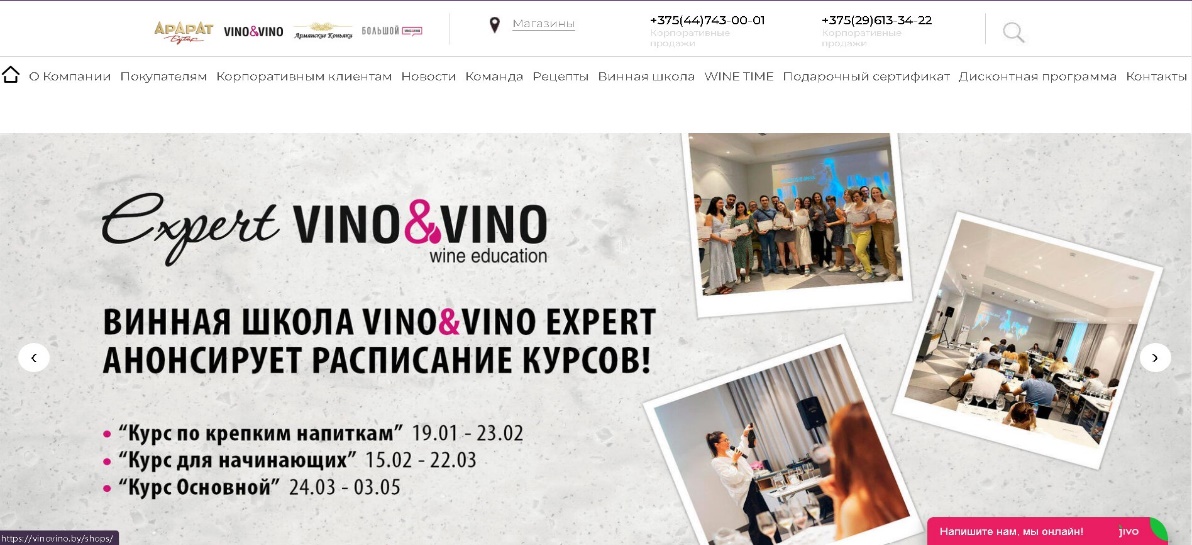


Рисунок1.1

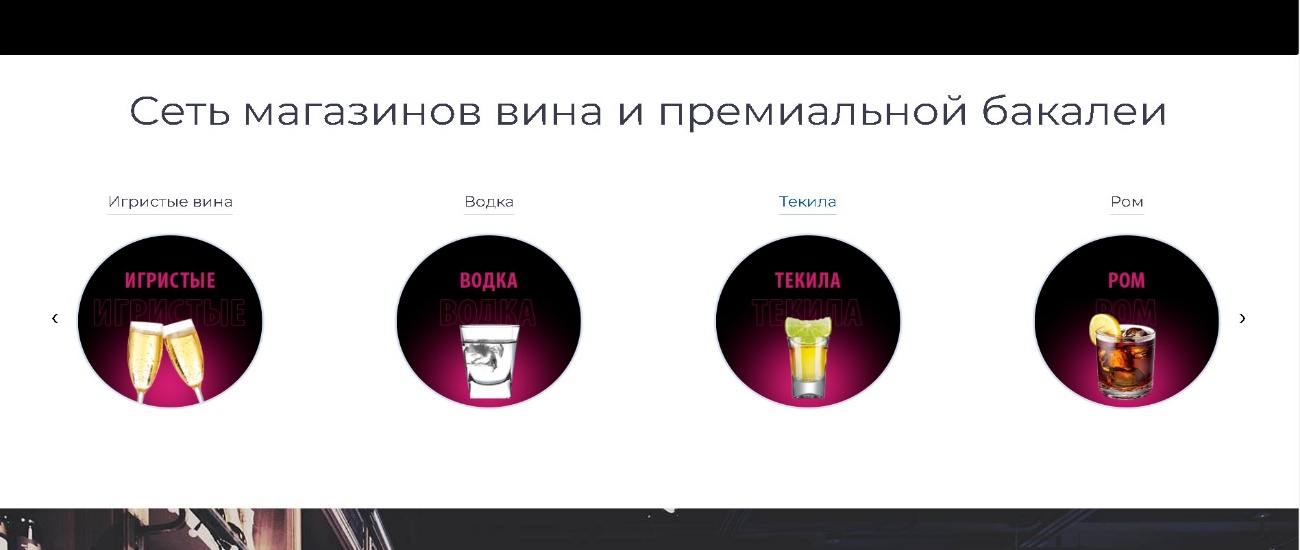


Рисунок 1.2

Данный сайт содержит все основные компоненты, имеет много различных динамических эффектов. Однако сайт содержит не очень проработанный подвал (footer). Он содержит довольно много мелкого текста, который плохо воспринимается. Также не очень удобно расположение меню перед самим подвалом. Дизайн представлен на рисунке 1.3.

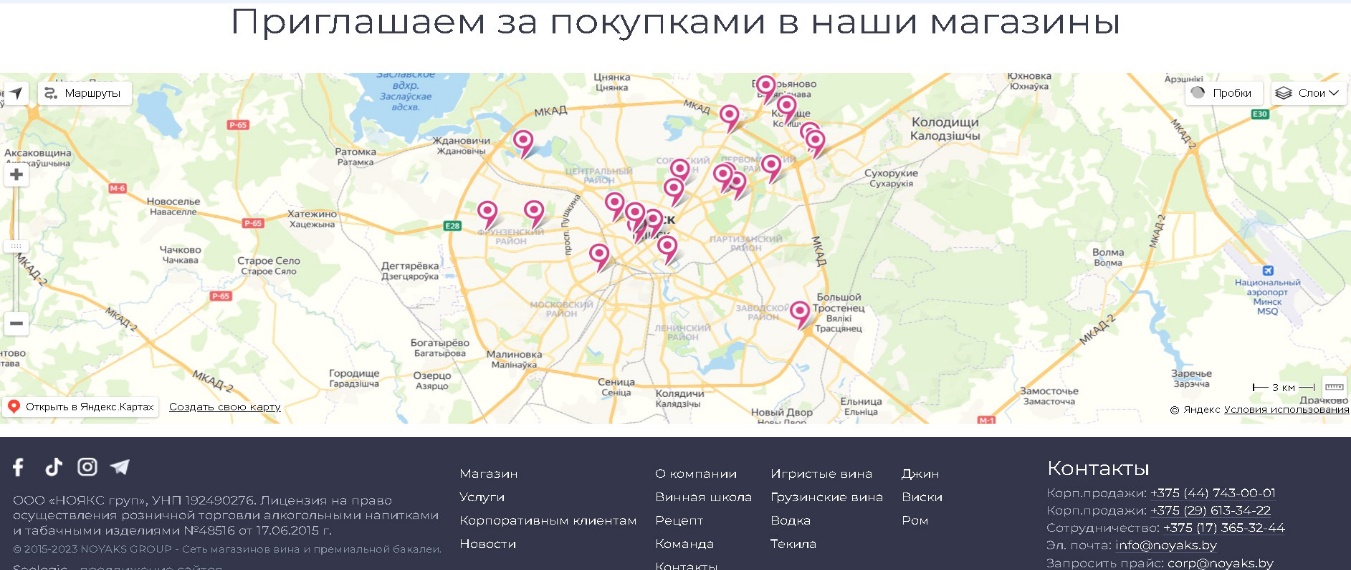


Рисунок 1.3

Есть мобильная версия сайта, имеется выпадающее меню. Однако оно также не очень удобно для пользователя. Текст может показаться слишком маленьким для экрана мобильного телефона. Кроме того, не очень приятна смесь различных текстовых стилей: различный цвет, размер текста. Дизайн представлен на рисунке 1.4.

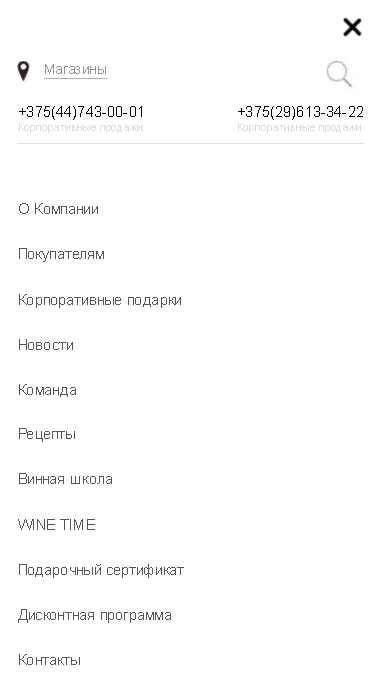


Рисунок 1.4

Следующий сайт goodwine.com.ua. В разделе «Оплата» шрифт не имеет дизайна, также он очень мелкий и практически не читабельный. Дизайн представлен на рисунке 1.5.

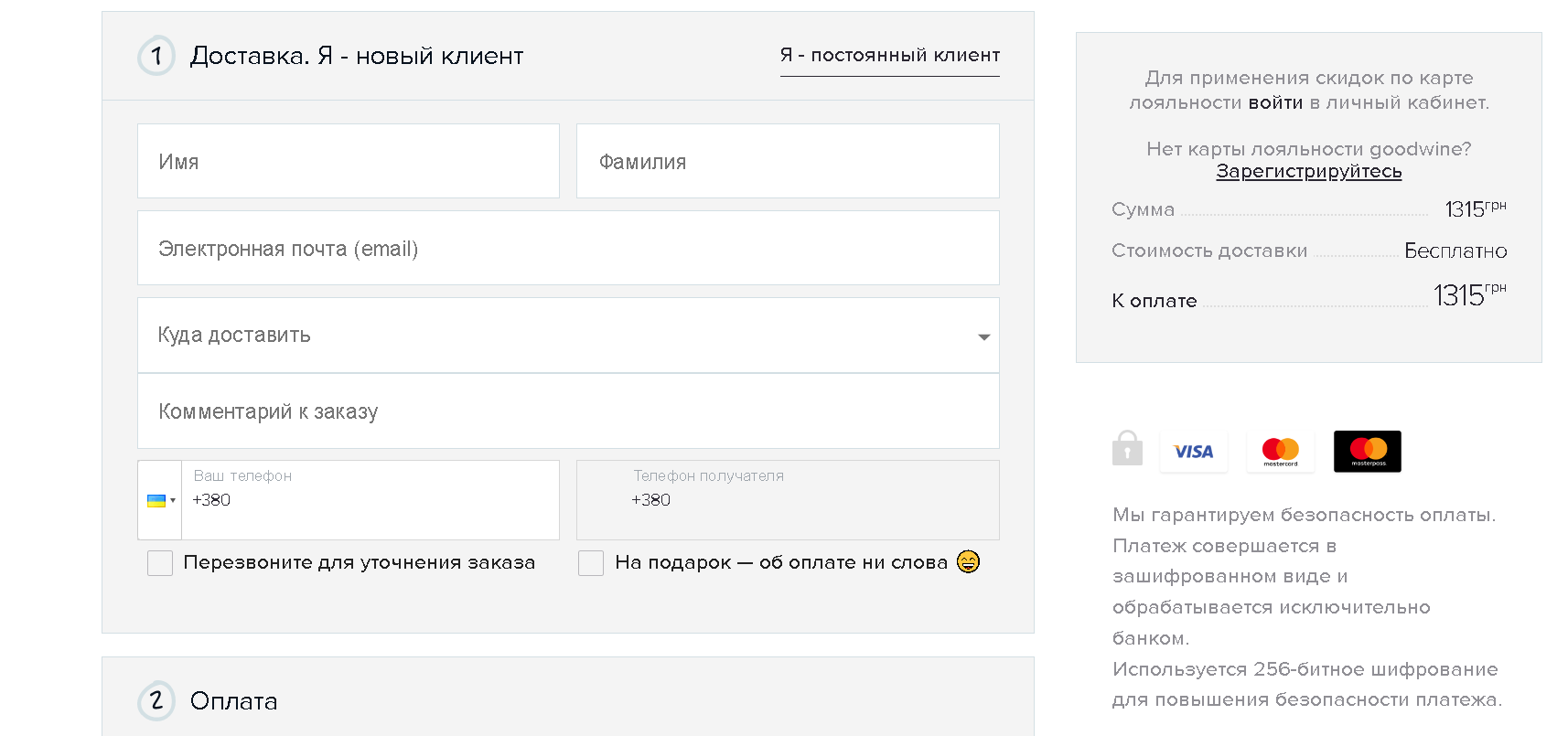


Рисунок 1.5

### **1.2 Техническое задание**

В данном проекте требуется создать многостраничный сайт магазина винных изделий. Сайт должен содержать общую информацию о магазине, такую как контактные данные, описание магазина, а также каталог товаров. Основным контентом являются текстовые данные, описания товаров, фотографии товаров.

Главной задачей сайта является предоставление данных о товарах для привлечения внимания потенциальных клиентов. Сайт должен быть интерактивным, содержать актуальную информацию. Данный сайт предназначен для желающих приобрести винные изделия в данном магазине.

Меню на всех страницах будет располагаться сверху. В меню будут предоставляться ссылки на основную информацию (главная страница (home), о магазине (about), товары (shop), контакты (contact), и т.д.), а также ссылки на элементы каталога товаров.

В мобильной версии (ширина до 560px) страница должна адаптироваться под ширину устройства. Также должна быть разработана версия для планшетного устройства (ширина до 1024px). Начиная с ширины в 1024px должна открываться версия для персонального компьютера.

Основные требования к исполнителю: создание дизайна, проведение анализа существующих решений, создание прототипа. В проекте используется гипертекстовый язык разметки HTML, CSS/Sass, графические элементы в формате SVG; данные хранятся в XML-формате.

­­­Задачи курсового проекта:

- Упростить поиск информации о магазине для пользователей.

- размещение основной информации на сайте.

- размещение информации о товаре.

- представление имиджа компании.

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

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

Курсовая работа была выполнена в редакторе кода VS Code.

HTML (или HyperText Markup Language) — это стандартизированный язык разметки документов в интернете. Вопреки расхожему заблуждению, HTML – это не язык программирования. Он используется для того, чтобы все элементы на странице (тексты, рисунки, таблицы…) были расположены правильно; за его чтение отвечают специальные программы, которые всем знакомы – браузеры (веб-обозреватели). В 2014 году был создан HTML5. Разработчики ориентировались на устранение проблем предыдущих версий и реализацию современных функций:

* Высокая скорость работы сайта за счет удаления всего лишнего из кода и перераспределения функциональных элементов.
* Сайты на HTML5 используют совместно ресурсы браузеров и удаленных серверов, чтоб делает серфинг и просмотр сайтов более комфортным.
* Сайты стали более легкими, что очень важно для мобильного интернета, который работает медленнее стационарного.
* Медиа и интерактивные функции сайтов теперь решаются не путем установки на компьютер специальных приложений, а ресурсами самого кода сайта.

JavaScript — это полноценный динамический язык программирования, который применяется к HTML документу, и может обеспечить динамическую интерактивность на веб-сайтах.

CSS (или Cascading Style Sheets) — это формальный язык, который определяет внешний вид документа, написанного с использованием HTML.

Sass – это своего рода расширение, созданное для упрощения каскадных таблиц стилей (CSS).

XML (*eXtensible Markup Language)* — расширяемый язык разметки. Используется для хранения и передачи данных.

Курсовая работа выполнена в редакторе кода VS Code.

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. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга.

### **1.4 Вывод.**

В данном разделе были рассмотрены аналогичные решения, были проанализированы их недостатки и преимущества, для создания в последующем качественного веб-сайта. Были поставлены конкретные задачи для создания интерактивного информационного ресурса. Кроме того, были рассмотрены средства реализации программного продукта, такие как языки разметки, а также редактор кода VS Code. Были проанализированы особенности языков и основные дополнительные средства редактора. Все вышеперечисленные пункты понадобятся для достижения поставленной задачи и для создания в дальнейшем качественного продукта.

# 

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

### **2.1 Выбор способа верстки**

Исходя из требований к проекту, в частности адаптивность и кроссбраузерность сайта, было принято решение об использовании CSS-препроцессора Sass.

Для создания отдельных блоков страницы 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+).

Первым шагом было принято создать прототип сайта на основе рассмотренных аналогов, сделать прототип удобным и легким. По прототипу будет понятно, где будут располагаться основные блоки, картинки, текст. Прототип будет представлен в приложении А.

Вторым шагом было принято решение создать основной макет. Макет будет состоять из основных блоков, таких как header, nav, main, section и footer. Макет будет представлен в приложении Б.

### **2.2 Выбор стилевого оформления**

Для реализации стилевого оформления сайта было выбрано минималистическое направление с элементами классического веб-дизайна.

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

Цветовые решения яркие, хорошо сочетающиеся между собой. Основные цвета: #6C6C6C, #AC1313, #4B276F, #000000. Цветовая палитра представлена на рисунке 2.1.



Рисунок 2.1

### **2.3 Выбор шрифтового оформления**

В данном проекте будут использованы два основных шрифта:   
Poppins и Libre Franklin. Шрифты подключались с помощью библиотеки Google Fonts. Оба шрифта поддерживают кириллическое написание и не конфликтуют друг с другом по стилевым характеристикам.

Poppins является вариативным шрифтом и относится к категориям гротески, декоративные. Гарнитура содержит 20 файлов. Поддерживает 57 языков. Имеет лицензию SIL OFL 1.1. Можно использовать в коммерческой и не коммерческой деятельности. Разработка шрифта Poppins велась Jonny Pinhorn и Ninad Kale в студии типографики Indian Type Foundry. Он будет применен к заголовкам, а также к элементам меню.

Libre Franklin является вариативным шрифтом и относится к категориям гротески, декоративные, для заголовков, для текстов. Гарнитура содержит 2 файла. Поддерживает 62 языка. Имеет лицензию SIL OFL 1.1. Можно использовать в коммерческой и не коммерческой деятельности. Разработка шрифта Libre Franklin велась Pablo Impallari, Nhung Nguyen и Rodrigo Fuenzalida. Данный шрифт применим к основному тексту.

### **2.4 Разработка логотипа**

Разработка логотипа осуществлялась в графическом редакторе Adobe Illustrator. Дизайн довольно минималистичен: было принято решение не загромождать его множеством декоративных элементов. Логотип представляет из себя SVG-изображение бутылки вина. что отражает сфер, деятельности компании (магазин винных изделий). Дизайн логотипа представлен на рисунке 2.2.



Рисунок 2.2

### **2.5 Разработка пользовательских элементов**

В проекте будут представлены такие элементы пользовательского интерфейса, как кнопки (рисунок 2.3), поле ввода электронной почты пользователя (рисунок 2.4), чекбокс (рисунок 2.5) и слайдер (рисунок 2.6).



Рисунок 2.3



Рисунок 2.4



Рисунок 2.5



Рисунок 2.6

### **2.6 Разработка спецэффектов**

На сайте будут разработаны некоторые динамические эффекты и анимации, которые позволят придать сайту интерактивность и сделают его более привлекательным для пользователя. Например, при наведении на пункт меню указателем мыши, он будет изменять свой цвет на более красный и снизу будет плавно появляется красная линия (рисунок 2.6). Указатель мыши наведен на пункт «About».



Рисунок 2.6

Кнопки также будут иметь свои спецэффекты: при наведении на них указателя мыши, они будут плавно менять цвет с красного на чёрный (рисунок 2.7).



Рисунок 2.7

С помощью свойства transform со значением scale была сделана анимация товара в магазине: при наведении картинка товара будет увеличиваться; также будет подсвечиваться нижняя граница блока (рисунок 2.8).

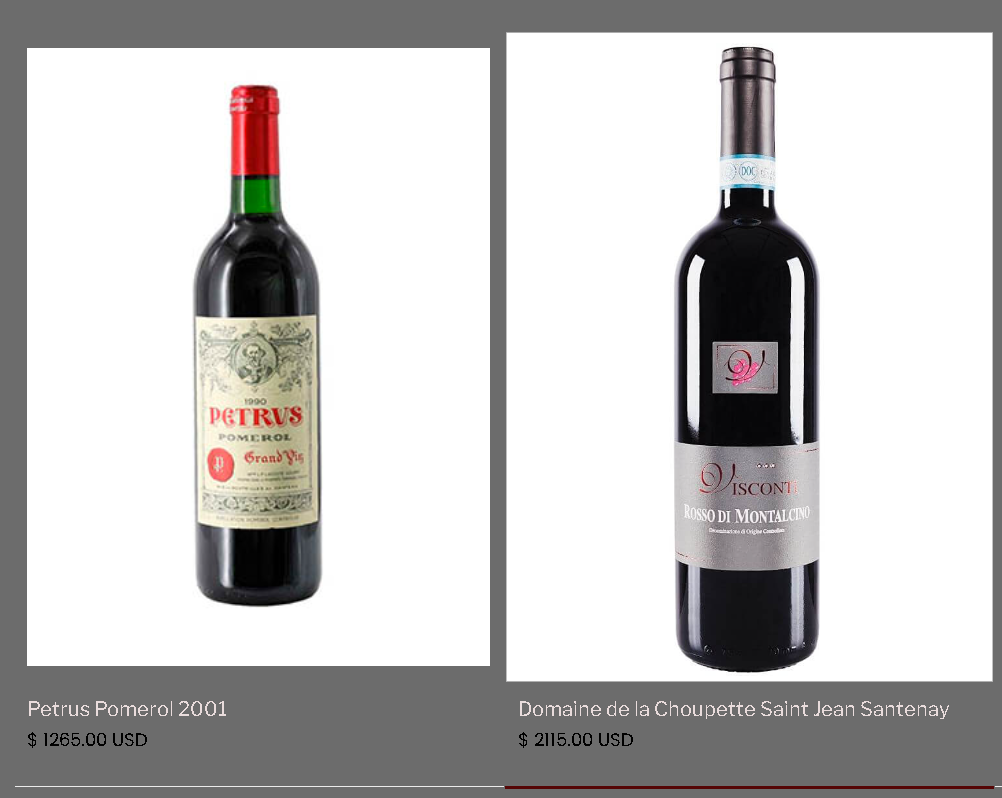


Рисунок 2.8

Также, на страницах будет присутствовать элемент scroll, который представлен на рисунке 2.9.

Scroll- это способ перемещения визуальной части окна вверх, вниз, вправо и влево для того чтобы увидеть дополнительную информацию в окне.

Кнопка scroll будет на всех устройствах и при нажатии на эту кнопку, скролл будет подниматься в вверх.



Рисунок 2.9

### **2.7 Вывод**

На данном этапе был создан прототип, а затем и макет сайта для облегчения последующей верстки. Было упорядочено расположение всех элементов на страницах (навигационного меню, логотипа, основного контента каждой страницы, расположение картинок, подвала сайта). Было определено стилевое оформление сайта, то есть основная цветовая гамма, шрифтовое оформление, а также динамические эффекты, применяемые к различным элементам.

# 

# **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.

Весь HTML, CSS, Sass, XML был протестирован с помощью онлайн - сервиса linter. Результат тестирования представлен на рисунке 3.1.

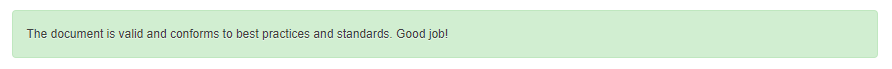


Рисунок 3.1

Структура тела документа состоит из семантических тегов header, nav, main, section и footer.

В теге header содержится главная информация о сайте. Код тега header показан в листинге 3.1.

|  |
| --- |
| <header class="header">  <a href="index.html">  <img class="logo" src="image/logo.svg" alt="logo"></a>  <div class="header\_\_links">  <nav class="header\_\_menu">  <a class="header\_\_link" href="#home">Home</a>  <a class="header\_\_link" href="#about">About</a>  <a class="header\_\_link" href="#shop">Shop</a>  <a class="header\_\_link" href="#contact">Contact</a>  </nav>  <a class="header\_\_cart" href="">  <p class="header\_\_cart-text">Cart</p>  </a>  <div class="containers" >  <div class="bar1"></div>  <div class="bar2"></div>  <div class="bar3"></div>  </div>  </div>  </header> |

Листинг 3.1– Структура тега header

В теге nav находится горизонтальная навигационная панель. Она содержит логотип компании и различные разделы сайта. Код тега nav находится в листинге 3.2.

|  |
| --- |
| <nav class="header\_\_menu">  <a class="header\_\_link" href="#home">Home</a>  <a class="header\_\_link" href="#about">About</a>  <a class="header\_\_link" href="#shop">Shop</a>  <a class="header\_\_link" href="#contact">Contact</a>  </nav> |

Листинг 3.2– Структура тега nav

В теге section находится различная второстепенная информация. Код одного из тега section продемонстрирован в листинге 3.3.

|  |
| --- |
| <section class="promo">  <div class="promo\_\_container">  <div class="promo\_\_text">  <p class="promo\_\_author">Best Wine</p>  <h1 class="promo\_\_title">Дегустация вина — путь к истине.</h1>  <p class="promo\_\_slogan">Секс, конечно, хорошо, но вы бывали в Виночерпии?</p>  <button class="promo\_\_button"> Shop Collection</button>  </section> |

Листинг 3.3 – Структура одного из тегов section

В теге footer содержится контактная информация. Код тега footer представлен в листинге 3.4.

|  |
| --- |
| <footer class="footer">  <div class="contacts-wrapper">  <img class="logo logo\_contacts" src="image/logo.svg" alt="logo">  <p class="contacts\_\_text">«Для того чтобы жить долго, приобрети для себя старого вина и старого друга». Пифагор (философ).</p>  </div>  <ul class="contacts-block">  <li class="contacts-block\_\_item contacts-block\_\_item\_email">  <h5 class="contacts-block\_\_title">email</h5>  <p class="contacts-block\_\_info"><a href="https://mail.google.com/mail/u/0/?fs=1&tf=cm&source=mailto&to=potteryshop@gmail.com" class="footer\_\_link" target="\_blank">wineandwine@gmail.com</a></p>  </li>  <li class="contacts-block\_\_item contacts-block\_\_item\_adress">  <h5 class="contacts-block\_\_title">find</h5>  <p class="contacts-block\_\_info"><a href="https://www.google.com/maps/place/VINO%26VINO/@53.904576,27.5592783,17z/data=!3m1!4b1!4m6!3m5!1s0x46dbcfc00447d2ff:0xc8fa6f589e858a27!8m2!3d53.904576!4d27.561467!16s%2Fg%2F11b7grbhj5?hl=ru" class="footer\_\_link" target="\_blank">st. Internationalnaya 36, ​​Minsk</a></a>  </li>  <li class="contacts-block\_\_item contacts-block\_\_item\_phone">  <h5 class="contacts-block\_\_title">call</h5>  <p class="contacts-block\_\_info">+375 33 3468667</p>  </li>  </ul>  </footer> |

Листинг 3.4 – Структура тега footer

### 

### **3.2 Добавление таблиц стиля CSS**

Добавление таблиц стиля CSS позволило упростить создание стилистического оформления страницы.

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.

Код, представленный в листинге 3.5 содержит стили CSS для верхней части веб-страницы - шапки или header.

Он определяет размер и отступы шапки, цвет фона, а также позиционирование элементов внутри нее.

Также, данный код содержит стили для кнопки-гамбургера, которая используется на мобильных устройствах для открытия/закрытия меню. Она состоит из трех горизонтальных баров, которые в зависимости от состояния кнопки (открыта/закрыта) меняют свою форму и положение.

|  |
| --- |
| .header {  max-width: 1440px;  padding: 0 150px 0 130px;  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-pack: justify;  -ms-flex-pack: justify;  justify-content: space-between;  margin: 0 auto;  background: #6c6c6c;  position: -webkit-sticky;  position: sticky;  top: 0;  left: 0;  z-index: 10;  }  .header .logo {  width: 119px;  height: 45px;  margin: 19px 0 0 0;  }  .header .header\_\_links {  -webkit-box-sizing: border-box;  box-sizing: border-box;  padding: 26px 0;  display: -webkit-box;  display: -ms-flexbox;  display: flex;  -webkit-box-pack: justify;  -ms-flex-pack: justify;  justify-content: space-between;  }  .header .header\_\_links .containers {  display: none;  cursor: pointer;  }  .header .header\_\_links .bar1,  .header .header\_\_links .bar2,  .header .header\_\_links .bar3 {  width: 35px;  height: 5px;  background-color: #333;  margin: 6px 0;  -webkit-transition: 0.4s;  transition: 0.4s;  }  .header .header\_\_links .change .bar1 {  -webkit-transform: rotate(-45deg) translate(-9px, 6px);  transform: rotate(-45deg) translate(-9px, 6px);  } |

Листинг 3.5 – Стили для верхней части веб-страницы

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

Было принято решение о работе с XML форматом. Главной причиной этому является простота использование данных, но легче в использовании был бы Json формат. К преимуществам формата Json можно отнести-удобочитаемость кода, простота создания объекта данных на стороне сервера и обработки данных на стороне клиента, возможность расширения, отладки и исправление ошибок, а также безопасность.

В проекте формат XML был использован для хранение данных о постах. Листинг 3.6.

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <root>  <block>  <img>image/product1.jpg</img>  <title> Petrus Pomerol 2001 </title>  <price>$ 1265.00 USD </price>  </block>  <block>  <img>image/product2.jpg</img>  <title> Domaine de la Choupette </title>  <price> $ 2115.00 USD </price>  </block>  <block>  <img>image/product3.jpg</img>  <title> ROMANEE-CONTI </title>  <price> $28.000 USD </price>  </block>  <block>  <img>image/product4.jpg</img>  <title> Pomerol Chateau </title>  <price> $ 2355.00 USD </price>  </block>  <block>  <img>image/product5.jpg</img>  <title> 50 50 Capannelle </title>  <price> $ 1115.00 USD </price>  </block>  <block>  <img>image/product6.jpg</img>  <title> Thibault Liger Belair </title>  <price> $ 3115.00 USD </price>  </block>  </root> |

Листинг 3.6 – Данные, представленные в XML-формате

XML документ использовался для описания всех предложенных курсов. В нем использовались следующие теги:

* <root> (тег-контейнер, содержащий все теги и весь XML файл);
* <block> (тег-контейнер, содержащий другие теги, представленные ниже, в которых буду содержатьcя в товаре) ‘
* <img> (тег, содержащий картинку товара);
* <title> (тег, содержащий название товара);
* <price> (тег, содержащий цену товара);

Для внедрения иконок было принято решения использовать расширение .svg, что позволило изменять размер экрана, при этом не изменить качество изображения. Листинг предоставлен в приложении А.

JavaScript был использован для создания анимации во время определенных событий: в навигационном меню присутствует меню-бургер; анимация при скролле и слайдер. Листинг представлен в приложении Б.

### **3.4 Выводы**

На данном этапе была реализована структура на HTML, а также созданы таблицы стилей CSS. Было продемонстрировано использование стандартов XML (SVG), а также языка программирования JS. 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 Адаптивный дизайн веб-сайта**

Сегодня количество пользователей, которые заходят в Интернет с мобильных устройств очень велико, и с каждым днем их становится все больше. А потому адаптация веб-сайта под маленькие экраны – актуальная задача и для курсового проекта.

Передо мной стояла задача: создать сайт, который будет сам подстраиваться под разные типы экранов. Главными условиями адаптивности является не фиксированный размер элементов страницы.

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

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

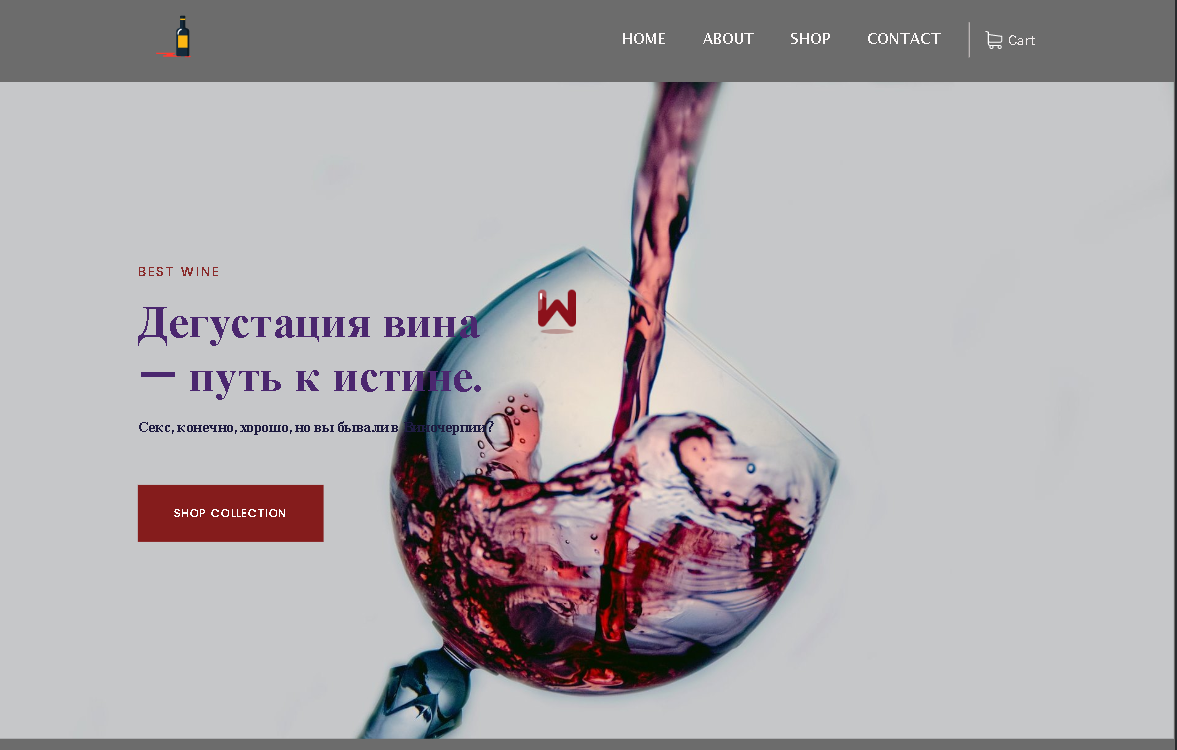
Кроссбраузерность – это способность веб-ресурса отображаться одинаково и работать во всех популярных браузерах, без перебоев в функционировании и ошибок в верстке, а также с одинаково корректной читабельностью контента.

Разные пользователи используют разные браузеры, чтобы осложняет работу верстальщикам, так как не все свойства CSS используются одинаково во всех браузерах. Для этого и была придумана кроссбраузерность — способность веб-ресурса отображать одинаково и работать во всех популярных браузерах, без всяких проблем.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали было принято решения о внедрении вендорных префиксов. Они были автоматически прописаны после компиляции файла с расширением .scss.

Веб-сайт будет тестироваться в следующих браузерах: Google Chrome, Microsoft Edge и Opera.

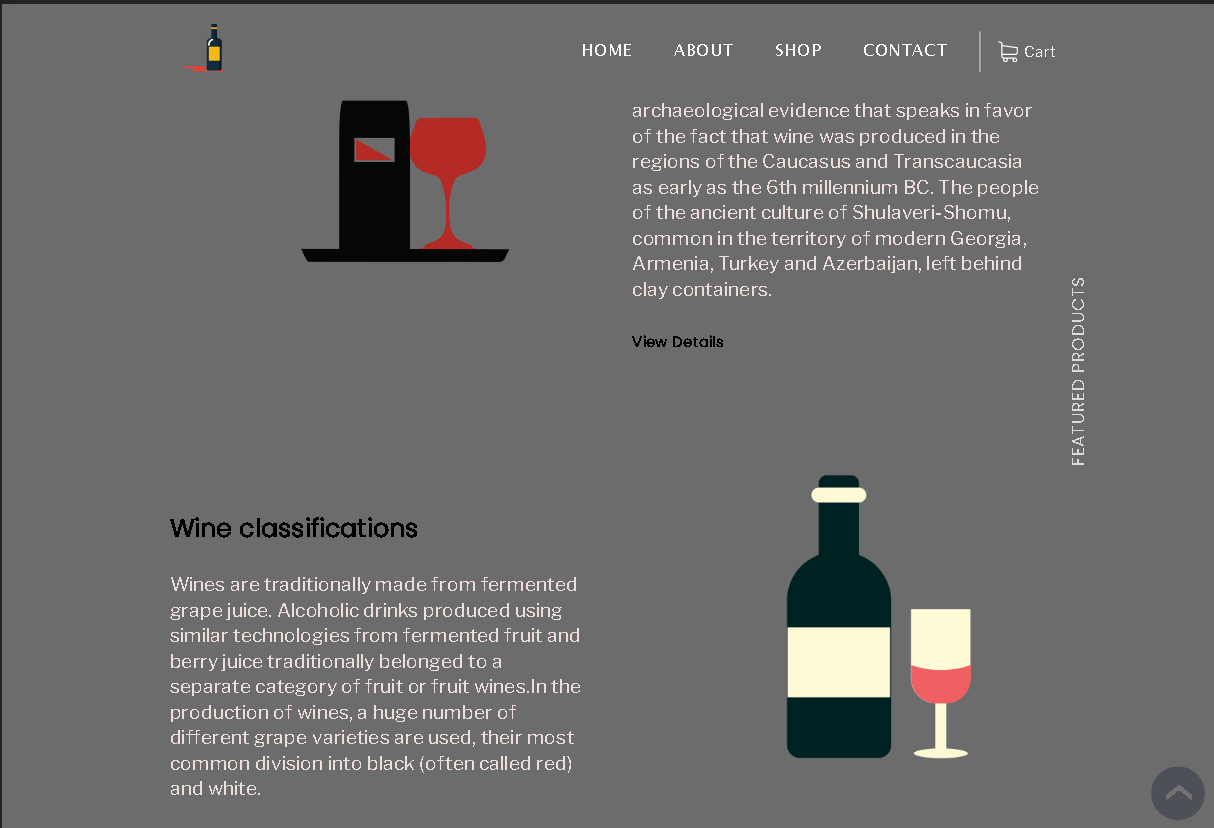
При проверке веб-сайта в Google Chrome никаких искажений текста, анимации, переходов не было выявлено. На рисунке 4.1 можно увидеть пример веб-страницы в Google Chrome.

а) б)

Рисунок 4.1 – Вид веб-страницы в Google Chrome а) мобильная версия; б) компьютерная версия

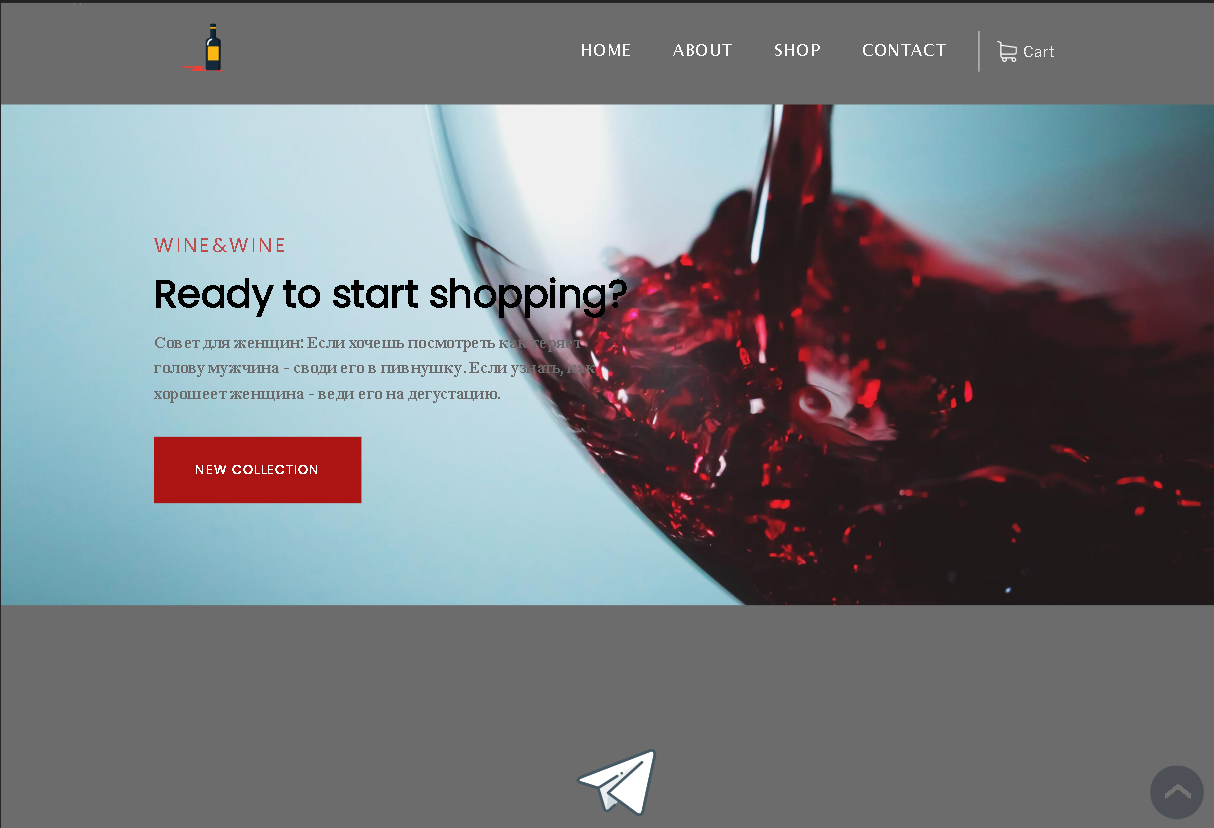
В Microsoft Edge всё так же работает без неисправностей, весь контент обладает правильным шрифтом, не наблюдается сдвиг текста или отсутствие картинок. На рисунке 4.2 можно увидеть пример веб-страницы в Microsoft Edge.

а) б)

Рисунок 4.2 – Вид веб-страницы в Microsoft Edge а) мобильная версия; б) компьютерная версия

В Opera так же, как и в остальных браузерах, не наблюдается того, чтобы отличалось от требуемого результата. На рисунке 4.3 можно увидеть пример веб-страницы в Opera.

а) б)

Рисунок 4.3 – Вид веб-страницы в Opera а) мобильная версия; б) компьютерная версия

В ходе тестирования было принято решение о проведении проверки веб-сайта в браузере Internet Explorer. В ходе проверки было выявлено, что браузер не поддерживает некоторые свойства, такие как scroll-behavior (отвечает за поведение прокрутки для любого элемента на странице, например, плавность перехода по якорным ссылкам) и пользовательский скроллбар.

### **4.3 Руководство пользователя**

Данный сайт создан для того, чтобы пользователь мог заказать винные изделия, которые представлены в текстовом или графическом варианте.

Для удобно пользования сайтом реализовано:

- разработан удобный и приятный пользовательский интерфейс, который будет понятен для любого человека.

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

- для удобного выбора товара создана страница Shop, где пользователь может выбрать товар из определенной категории.

- для получения информации о продукте создана страница Product, где указана информация о том или ином продукте.

- на главной странице указаны самые популярные товары и все категории, которые доступны на сайте компании.

### **4.4 Выводы**

В данном разделе я рассмотрела, как сайт ведет в себя на разных устройствах и в разных браузерах. И ходе этого тестирования выявила некоторые ошибки и исправила их. Так же в этом разделе есть руководство пользователя, где кратко указана информация о функциональности сайта.

# **Заключение**

Целью курсового проекта была разработка веб-сайт для магазина по продаже винных изделий, который предоставляет возможность заказать разные продукты с любой точки мира. Также цель была в том, чтобы сделать удобный пользовательский интерфейс, чтобы сайт был адаптивен под разные устройства. Сайт содержит в себе 3 страницы, а именно главная страница, страница категории продуктов и страница информации о продукте.

Для реализации макета веб-страницы были исследованы другие сайты интернет-магазины винных изделий.

Для удобного написания кода веб-страницы была использована среда разработки VS Code, в которой были установлены плагины для облегченного пользования.

Но перед написанием кода, были изучены новые стандарты языков разметки и программирования. Были прочтены документации по использованию технологий, которые будут использоваться на этапе написания сайта.

После написания всего веб-сайта следовало его тестирование, в ходе которого были выяснены некоторые баги, которые были устранены в скором времени.

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

Для выполнения поставленной цели курсового проекта функционально были реализованы следующие задачи:

Анализ аналогичных решений;

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

И в результате разработки был подготовлен веб-сайт интернет-магазина винных изделий. Сайт работает стабильно и в дальнейшем он может принадлежать эксплуатации. Но сайт можно в дальнейшем дорабатывать, например, добавить расширенный поиск, который будет искать не только по странице, но и по всей базе данных продуктов сайта.

Если подытожить, то за этот курсовой были использованы как теоретические знания, так и практические знания разработки веб-сайтов, для выполнения всех поставленных задач.

# Список использованных источников

1. Документация по Figma [Электронный ресурс]. — Режим доступа: <https://help.figma.com> – Дата доступа 01.03.2022.
2. Гайд по Figma для начинающих [Электронный ресурс]. — Режим доступа: <https://tilda.education/articles-figma> – Дата доступа 01.03.2022.
3. Документация по Sass/SCSS [Электронный ресурс]. — Режим доступа: <https://sass-scss.ru/guide/> – Дата доступа 10.03.2022.
4. Документация по JavaScript [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org> – Дата доступа 10.03.2022.
5. Введение в кросс-браузерное тестирование [Электронный ресурс]. — Режим доступа:

<https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction>. — Дата обращения: 01.05.2022.

# **Приложение А Листинг SVG**

|  |
| --- |
| <?xml version="1.0" encoding="iso-8859-1"?>  <svg version="1.1" id="Capa\_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"  viewBox="0 0 216.294 216.294" style="enable-background:new 0 0 216.294 216.294;" xml:space="preserve">  <g>  <g id="group-158svg">  <path id="path-1\_65\_" style="fill:#FF3A2F;" d="M94.929,192.572h15.769c3.04,0,5.505,1.546,5.505,3.454  c0,1.909-2.465,3.457-5.505,3.457h-0.734c-3.04,0-5.505,1.546-5.505,3.455c0,1.907,2.465,3.455,5.505,3.455h33.899  c3.04,0,5.504,1.547,5.504,3.455c0,1.908-2.464,3.454-5.504,3.454h-40.738H58.352c-3.041,0-5.505-1.546-5.505-3.454  c0-1.908,2.464-3.455,5.505-3.455h9.408c3.04,0,5.504-1.548,5.504-3.455c0-1.909-2.464-3.455-5.504-3.455H22.985  c-3.04,0-5.505-1.548-5.505-3.457c0-1.908,2.465-3.454,5.505-3.454h10.582H94.929z"/>  <path id="path-2\_65\_" style="fill:#FF3A2F;" d="M192.529,216.294c3.471,0,6.286-1.768,6.286-3.946c0-2.18-2.815-3.947-6.286-3.947  h-19.262c-3.471,0-6.285,1.767-6.285,3.947c0,2.178,2.814,3.946,6.285,3.946H192.529z"/>  <path id="path-3\_65\_" style="fill:#142A3B;" d="M147.342,0.002c4.452,0,12.329,0.139,18.317,0.139  c3.764,0,4.456,3.383,4.456,3.383s0,39.681,0,51.868c0,6.906,2.061,7.853,3.597,8.941c2.031,1.427,7.545,2.964,10.092,4.058  c7.883,3.392,7.027,11.515,7.027,11.515s-0.056,105.041-0.056,124.543c0,4.605-2.178,6.55-5.944,6.55  c-17.114,0-37.096-0.139-53.316-0.139c-3.939,0-8.139-0.27-8.139-5.685c0-24.107,0-125.669,0-125.669s0.108-7.992,7.706-11.378  c6.675-2.987,6.001-2.439,8.899-4.203c3.172-1.919,3.255-3.112,3.255-8.397c0-30.89-0.17-48.748-0.17-52.004  C143.066,1.628,145.678,0.002,147.342,0.002z"/>  <path id="path-4\_65\_" style="opacity:0.35;fill:#020403;enable-background:new ;" d="M183.805,68.391  c-2.547-1.094-8.061-2.631-10.093-4.058c-1.535-1.088-3.596-2.035-3.596-8.941c0-12.189,0-51.868,0-51.868  s-0.692-3.383-4.456-3.383C159.671,0.141,151.796,0,147.342,0c-1.665,0-4.276,1.628-4.276,3.524c0,0.096,0,0.248,0,0.371  c3.628,0.037,7.597,0.083,10.976,0.083c3.764,0,4.455,3.383,4.455,3.383s0,39.679,0,51.866c0,6.908,2.061,7.855,3.597,8.941  c2.031,1.429,7.546,2.964,10.093,4.06c7.883,3.392,7.026,11.515,7.026,11.515s-0.056,105.039-0.056,124.541  c0,1.043-0.126,1.932-0.342,2.709c2.033,0.004,4.047,0.006,6.016,0.006c3.766,0,5.944-1.945,5.944-6.55  c0-19.502,0.056-124.543,0.056-124.543S191.688,71.783,183.805,68.391z"/>  <path id="path-5\_65\_" style="fill:#FBB713;" d="M169.781,22.23h-26.026v-7.21h26.026V22.23z"/>  <path id="path-6\_64\_" style="fill:#FBB713;" d="M182.154,163.907c0,1.47-1.161,2.657-2.571,2.657h-44.345  c-1.412,0-2.555-1.187-2.555-2.657v-58.395c0-1.465,1.143-2.654,2.555-2.654h44.345c1.41,0,2.571,1.189,2.571,2.654V163.907z"/>  <path id="path-7\_64\_" style="fill:#FFFFFF;" d="M132.684,98.675c-2.009,0-3.639-1.615-3.639-3.605  c0-18.926,12.495-26.186,19.103-27.384c1.972-0.343,3.873,0.942,4.233,2.899c0.36,1.95-0.938,3.821-2.902,4.189  c-0.601,0.119-13.156,2.85-13.156,20.296C136.323,97.06,134.693,98.675,132.684,98.675z"/>  </g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  <g>  </g>  </svg> |

Логотип сайта

# **Приложение Б Листинг JavaScript**

|  |
| --- |
| let containers = document.querySelector(".containers");  let header\_\_menu = document.querySelector(".header\_\_menu");  containers.addEventListener("click", () => {  containers.classList.toggle("change");  header\_\_menu.classList.toggle("active");  });  let header\_\_link = document.querySelectorAll(".header\_\_link");  header\_\_link.forEach((link) => {  link.addEventListener("click", () => {  containers.classList.remove("change");  header\_\_menu.classList.remove("active");  });  }); |

Меню-бургер

|  |
| --- |
| //Кнопка scroll  const scroll\_up = document.querySelector(".scroll");  scroll\_up.addEventListener("click", fun1);  window.onscroll = () => {    if (document.documentElement.scrollTop > 200)      scroll\_up.style.display = "block";    else {      scroll\_up.style.display = "none";    }  };  //Функция, чтобы scroll поднимал нас наверх  function fun1() {    window.scrollTo({      top: 0,      behavior: "smooth",    });  } |

Скрипт для кнопки scroll

|  |
| --- |
| //Кнопка слайдера  let promo\_\_button = document.querySelector(".promo\_\_button");  let main\_slider = document.querySelector(".main\_slider");  promo\_\_button.addEventListener("click", () => {  main\_slider.classList.add("active");  });  let close\_btn = document.querySelector(".close\_btn");  close\_btn.addEventListener("click", () => {  main\_slider.classList.remove("active");  }); |

Скрипт для кнопки, где размешен слайдер

|  |
| --- |
| const slideContainer = document.querySelector(".container"); // контейнер слайдов  const slide = document.querySelector(".slides"); // получаем контейнер слайдов  const nextBtn = document.getElementById("next-btn"); // кнопка вперед  const prevBtn = document.getElementById("prev-btn"); // кнопка назад  const interval = 3000; // интервал между слайдами  let slides = document.querySelectorAll(".slide"); // получаем все слайды  let index = 0; // индекс текущего слайда  let slideId;  const firstClone = slides[0].cloneNode(true); // клонируем первый слайд  и добавляем его в конец контейнера слайдов  const lastClone = slides[slides.length - 1].cloneNode(true); // клонируем последний слайд и добавляем его в начало контейнера слайдов  firstClone.id = "first-clone"; // добавляем id клонированному слайду  lastClone.id = "last-clone"; // добавляем клонированный слайд в конец контейнера слайдов  slide.append(firstClone); // добавляем клонированный слайд в конец контейнера слайдов  slide.prepend(lastClone); // добавляем клонированный слайд в начало контейнера слайдов  var slideWidth; // ширина слайда  setInterval(() => {    slideWidth = slides[0].clientWidth;  }, 0);  slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на нужную позицию  const startSlide = () => {    // запускаем слайдер    slideId = setInterval(() => {      moveToNextSlide(); //    }, interval);  };  const getSlides = () => document.querySelectorAll(".slide"); // получаем все слайды  slide.addEventListener("transitionend", () => {    // при завершении анимации    slides = getSlides(); //  получаем все слайды    if (slides[index].id === firstClone.id) {      // если переместились на первый слайд      slide.style.transition = "none"; // отключаем анимацию      index = 1; // возвращаем индекс на первый слайд      slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на первый слайд    }    if (slides[index].id === lastClone.id) {      // если переместились на последний слайд      slide.style.transition = "none"; // отключаем анимацию      index = slides.length - 2; // возвращаем индекс на последний слайд      slide.style.transform = `translateX(${-slideWidth \* index}px)`; // перемещаем слайд на последний слайд    }  });  const moveToNextSlide = () => {    slides = getSlides(); //  получаем все слайды    if (index >= slides.length - 1) return; // если переместились на последний слайд    index++; // увеличиваем индекс    slide.style.transition = ".7s ease-out";    slide.style.transform = `translateX(${-slideWidth \* index}px)`;  };  const moveToPreviousSlide = () => {    if (index <= 0) return; // если переместились на первый слайд    index--; // уменьшаем индекс    slide.style.transition = ".7s ease-out";    slide.style.transform = `translateX(${-slideWidth \* index}px)`;  };  slideContainer.addEventListener("mouseenter", () => {    // при наведении на контейнер слайдов    clearInterval(slideId); // останавливаем слайдер  });  slideContainer.addEventListener("mouseleave", startSlide); // при отведении курсора от контейнера слайдов  nextBtn.addEventListener("click", moveToNextSlide); // при нажатии на кнопку "следующий слайд"  prevBtn.addEventListener("click", moveToPreviousSlide); // при нажатии на кнопку "предыдущий слайд"  startSlide(); // запускаем слайдер  // Это бесконечный слайдер  с таймером и кнопками переключения слайдов |

Скрипт работы слайдера