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 «Скриптовые языки программирования»

Тема: Web-сайт «Интернет-магазин по продаже винограда»

**Исполнитель**

Студент 2 курса 3 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Г. И. Шатерник

подпись, дата

**Руководитель**

доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Жиляк

должность, учен. степень, ученое звание подпись, дата

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

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

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

дата, подпись

Курсовой проект защищен с оценкой

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Жиляк

подпись дата инициалы и фамилия

Минск 2023

**Содержание**

[Введение 6](#_Toc153137891)

[**1.2. Технические задание** 11](#_Toc153137892)

[**1.3. Выбор средств реализации программного продукта** 12](#_Toc153137893)

[**1.4. Постановка задач программного продукта** 12](#_Toc153137894)

[**1.5. Вывод** 13](#_Toc153137895)

[2. Проектирование страниц веб-сайта 14](#_Toc153137896)

[**2.1. Выбор стилевого оформления** 14](#_Toc153137897)

[**2.2. Выбор шрифтового оформления** 15](#_Toc153137898)

[**2.3. Разработка логотипа** 15](#_Toc153137899)

[**2.4. Разработка пользовательских элементов** 15](#_Toc153137900)

[**2.5. Разработка спецэффектов** 16](#_Toc153137901)

[3. Реализация структуры веб-сайта 19](#_Toc153137902)

[**3.1. Структура приложения** 19](#_Toc153137903)

[**3.2. Использование Redux** 20](#_Toc153137904)

[**3.3. Выводы** 22](#_Toc153137905)

[4. Тестирование веб-сайта 23](#_Toc153137906)

[**4.1. Адаптивный дизайн веб-сайта** 23](#_Toc153137907)

[**4.2. Кроссбраузерность веб-сайта** 24](#_Toc153137908)

[**4.3. Руководство пользователя** 25](#_Toc153137909)

[**4.4. Выводы** 26](#_Toc153137910)

[Заключение 27](#_Toc153137911)

[Список использованных литературных источников 29](#_Toc153137912)

# Введение

Интернет-торговля является одной из наиболее динамичных и перспективных областей современной экономики. Она предоставляет уникальные возможности для развития бизнеса и привлечения широкой аудитории клиентов. В контексте данного курсового проекта рассматривается разработка интернет-магазина, специализирующегося на продаже винограда.

Виноград является одним из самых популярных и востребованных продуктов в сфере питания и здорового образа жизни. Его разнообразие сортов и вкусовых качеств делает его привлекательным для широкой аудитории потребителей. Однако, традиционные розничные магазины часто не могут предложить такое разнообразие выбора и удобство покупки, как интернет-магазины.

Поэтому, создание и поддержание эффективного веб-сайта является критически важным для современных магазинов, поскольку это помогает им выделяться среди конкурентов, укреплять свой имидж и улучшать качество обслуживания клиентов, что в конечном итоге способствует увеличению продаж и прибыли.

Цель курсового проекта: разработать веб-сайт автосалона Audi с использованием React.

Задачи:

1. Проанализировать существующие языки разметки, инструменты и библиотеки для создания веб-сайта.
2. Разработать макет и прототип сайта.
3. Разработать структуру веб-сайта.
4. Наполнить сайт информацией по теме.
5. Протестировать веб-сайт.
6. Разработать руководство пользователя.

**1. Постановка задачи**

**1.1 Обзор аналогичных решений.**

Перед тем как приступать к разработке своего веб-сайта можно изучить уже готовые решения и посмотреть, что по мнению других разработчиков является наиболее необходимым для пользователей, чтобы приложение действительно было полезным и удобным.

Аналог №1 – сайт dolinasad.by

В качестве первого аналога рассмотрим сайт «Долина растений». На рисунке 1.1 представлено оформление сайта.

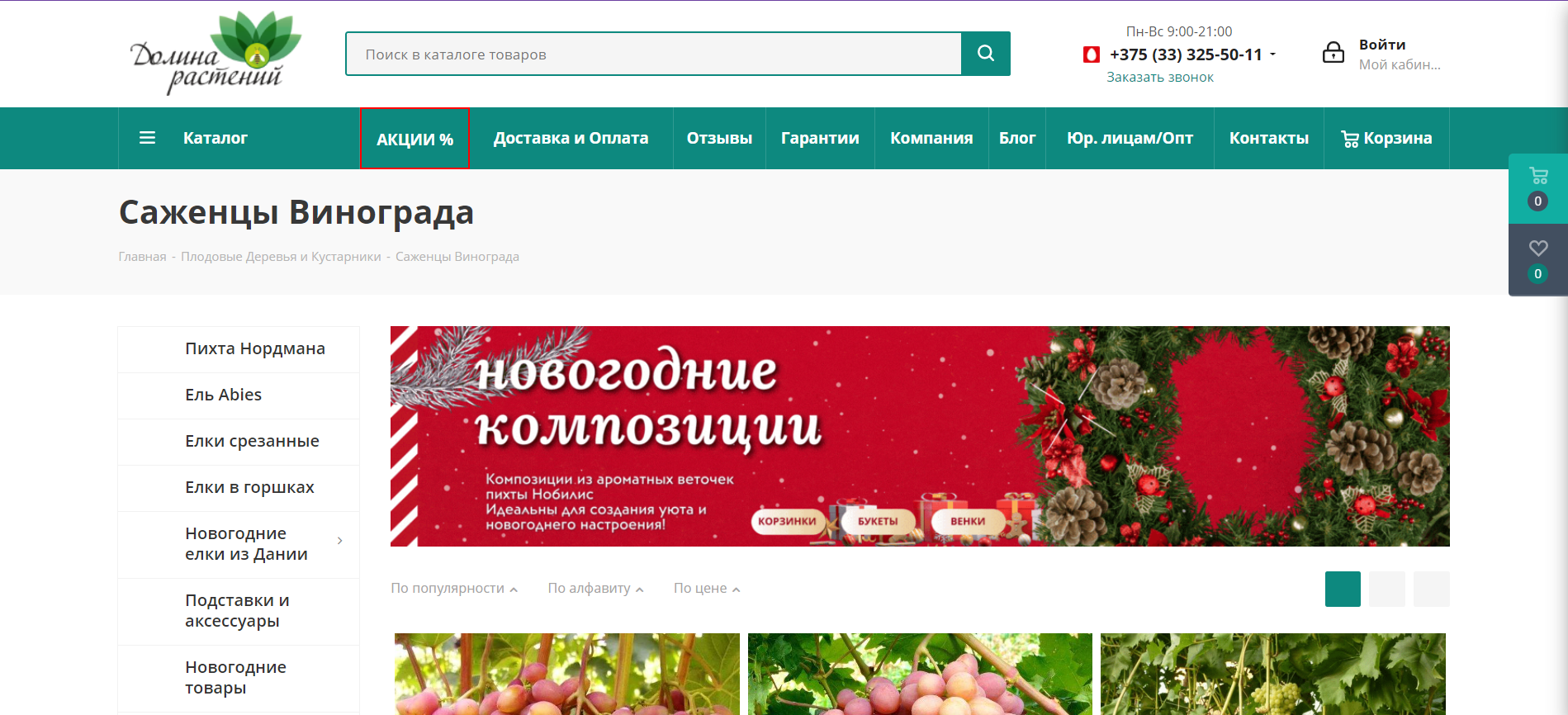


Рисунок 1.1 – Оформление сайта «Audi.by»

Сайт Долина растений призван привлечь целевую аудиторию, состоящую из потенциальных покупателей винограда, которые пока не определились с выбором сорта. Однако, после посещения сайта, эти люди, поймут какие сорта привлекательнее для них, благодаря его привлекательному внешнему виду и удобному интерфейсу. Для удобства посетителей, на сайте присутствует горизонтальное меню и ряд сортов представлен в виде сетки. Читабельный шрифт и приятные оттенки цветов делают пребывание на сайте комфортным. В целом, данный сайт является информативным, не вызывая впечатления захламленности.

Данный сайт имеет качественную мобильную версию. Выполненную в стиле версии для ПК. Дизайн представлен на рисунке 1.2.

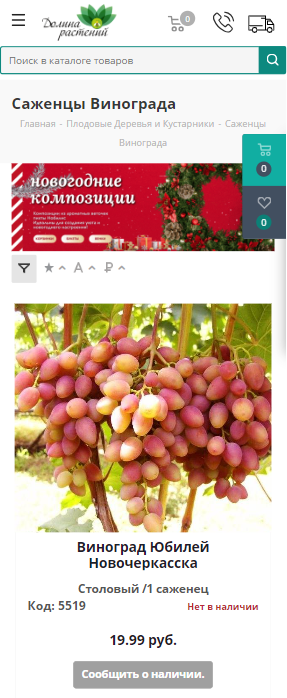


Рисунок 1.2 – Мобильная версия сайта «dolinasad.by»

Аналог №2

Следующий сайт – «bel-am.by». Оформление главной страницы сайта представлено на рисунке 1.3.

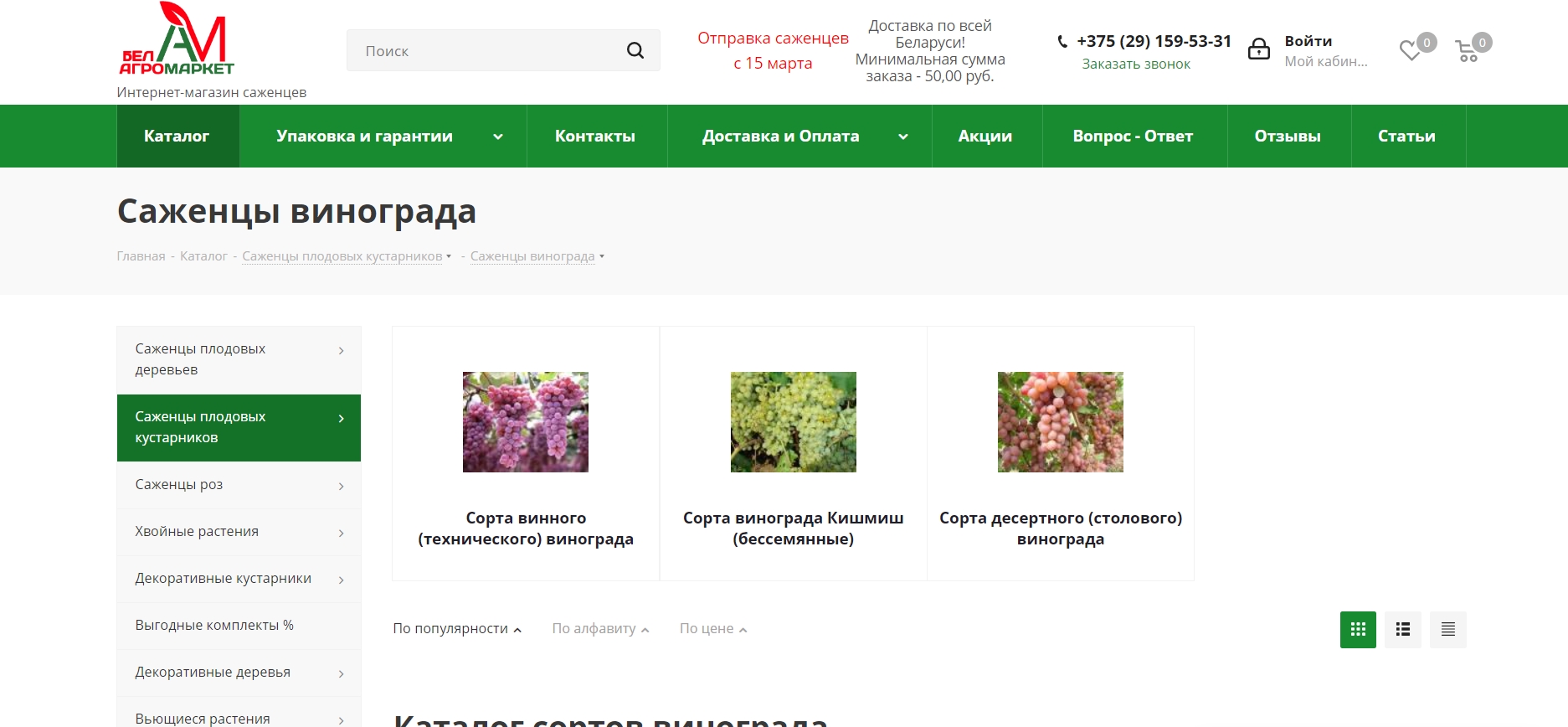


Рисунок 1.3 – «bel-am.by»

Сайт схож с сайтом dolinasad.by, так как является интернет-магазином по продаже винограда с широким выбором сортов. Данный сайт предоставляет пользователю более современный интерфейс, является более минималистичным. Является довольно удобным в использовании.

Шрифт читабельный, не вызывает дискомфорта при просмотре. На сайте присутствует много функций пользовательского интерфейса, чтобы привлечь внимание аудитории. Сайт использует приятные оттенки цветов, чтобы пользователь мог дольше находиться на странице без дискомфорта.

Мобильная версия представлена на рисунке 1.4.



Рисунок 1.4 – Мобильная версия сайта «bel-am.by»

Данный аналог выполнен хорошо, сайт является информативным и минималистичным, не возникает впечатления захламленности на сайте.

**1.2. Технические задание**

Поставлена задача реализовать полноценный многостраничный веб-сайт «Grape World». На сайте должна быть представлена информация о магазине, такая как сорта винограда в наличии, новостной блог, местонахождение магазина, контактная информация.

Для привлечения внимания аудитории главная задача сделать удобный интерактивный сайт с подробной информацией о продукте и актуальной информацией.

Основными задачами сайта являются:

* предоставление навигационного меню;
* предоставление информации о магазине;
* предоставление информации о сортах виногрдаа;

На главной странице меню будет расположено как сверху, так и снизу. В меню будут предоставляться ссылки на основную информацию.

Веб-страница должна быть кроссбраузерной и адаптивной. Начиная с ширины в 1284px должна открываться версия для персонального компьютера.

Технические средства:

Первое техническое средство – это прототип веб-сайта. Он должен быть разработан с использованием графических редакторов Figma.

Второе техническое средство – это разметка сайта. Разметка содержания сайта будет выполнена с применением HTML5 и React.

Третье техническое средство – таблица стилей. Для описания внешнего вида веб-страниц использован CSS3.

Шапка сайта, которая содержит меню навигации и логотип должна быть на всех страницах сайта одинаковая.

Структурированная информация. Оформление сайта должно быть таким, чтобы пользователь мог найти нужную ему информацию за маленький промежуток времени.

Приятный дизайн. Композиция в дизайне должна быть сбалансированной. На сайте следует правильного разместить и задать размер различных объектов.

Верстка сайта должна быть адаптивной и кроссбраузерной. Для описания внешнего вида веб-страниц был использован CSS3.Допускается редактирование изображений в графических редакторах.

* 1. Цветовая гамма: белый, желтый, серый, голубой, темно-синий цвета
  2. Формат используемых в проекте изображений: jpg.

**1.3. Выбор средств реализации программного продукта**

Курсовая работа выполняется в редакторе кода Visual Studio Code.

При создании веб-страниц были использованы языки: HTML, CSS, SCSS, React, Redux.

HTML (HyperText Markup Language) — стандартный язык разметки гипертекстовых страниц в Интернете. Страницы успешно интерпретируются браузерами, которые отображают их на экранах различных электронных устройств в удобном для человека виде.

CSS — это фактически язык стилей, который определяет отображение HTML-документов. CSS работает со шрифтами, с цветами символов и фона, с полями, со строками, с высотой и с шириной элементов отображения, с фоновыми изображениями, с позиционированием элементов и со многим другим.

Если HTML необходим для структурирования содержания страницы, то CSS необходим для того, чтобы форматировать это структурированное содержание.

SCSS (Sassy CSS) - язык препроцессора CSS, который позволяет использовать дополнительные функции и возможности, не предусмотренные в обычном CSS. SCSS добавляет новые функции, такие как переменные, миксины, вложенные правила, операторы и т.д., которые значительно упрощают и ускоряют процесс написания стилей для веб-сайтов.

Redux - библиотека управления состоянием приложения в экосистеме React и других JavaScript-приложений. Она предоставляет предсказуемый и однозначный способ управления состоянием, делая его централизованным и легко отслеживаемым.  
 React - JavaScript библиотека для разработки пользовательских интерфейсов. Созданная и поддерживаемая Facebook, React предназначен для построения эффективных и масштабируемых веб-приложений с использованием компонентного подхода.

**1.4. Постановка задач программного продукта**

Проект представляет собой веб-платформу, целью которой является обеспечение пользователей всей необходимой информацией об ассортименте сортов винограда, создавая интерактивный и информативный каталог. Сайт предоставит пользователям возможность легко ознакомиться с различными сортами, видами, странами-производителями, подробностями истории выращивания винограда.

**1.5. Вывод**

В результате проведенного анализа были выявлены как преимущества, так и недостатки аналогов, связанных с выбранной темой веб-сайта. Этот анализ позволил избежать повторения ошибок, совершенных другими разработчиками, и включить в создание сайта только наиболее эффективные и полезные решения.

Были сформулированы основные технические требования, которые будут руководить процессом разработки страницы. Эти требования определяют не только функциональность и дизайн сайта, но также и его производительность, безопасность и удобство использования. Учитывая эти требования на этапе разработки, можно сократить время и ресурсы, затрачиваемые на исправление ошибок и внесение изменений в будущем.

Для разработки программного продукта были выбраны оптимальные инструменты, включая языки программирования HTML, CSS и React. HTML используется для создания структуры и содержимого веб-страницы, CSS - для задания внешнего вида и стилей элементов, а React - для реализации интерактивности и динамического поведения сайта. Этот выбор позволит создать функциональный и эстетически привлекательный веб-сайт, соответствующий ожиданиям пользователей.

В целом, результаты анализа и принятые решения позволят создать высококачественный веб-сайт, сочетающий в себе преимущества аналогов и удовлетворяющий поставленным техническим требованиям.

**2. Проектирование страниц веб-сайта**

**2.1. Выбор стилевого оформления**

В выборе способа верстки для данного проекта, я учел не только требования к адаптивности и кроссбраузерности, но и стремился к максимальной эффективности и гибкости в создании макетов. В результате этого анализа я принял решение использовать адаптивную верстку из библиотеки Bootstrap, а также Flex-верстку для некоторых элементов, нуждающихся в дополнительном внимании.

Bootstrap - это один из наиболее популярных фреймворков для разработки веб-интерфейсов. Он предоставляет набор инструментов и стилей, основанных на HTML, CSS и JavaScript, которые позволяют разработчикам быстро и эффективно создавать современные и отзывчивые веб-сайты.

Одной из ключевых особенностей Bootstrap является его интеграция с flex-версткой. Flex-верстка - это новая и прогрессивная технология, которая позволяет создавать гибкие макеты и управлять элементами на веб-странице. Bootstrap предоставляет готовые классы и компоненты, основанные на flex-верстке, что делает процесс разработки более простым и удобным.

Преимущества использования Bootstrap и flex-верстки включают возможность легко изменять порядок элементов, управлять их размерами, выравниванием и расположением. Это особенно полезно при создании адаптивных веб-сайтов, которые должны хорошо выглядеть и функционировать на различных устройствах и экранах.

Еще одним преимуществом Bootstrap является его возможность создавать анимации и интерактивность на веб-странице. С помощью предоставляемых классов и компонентов, разработчики могут легко добавлять плавные переходы, эффекты и анимации, что придает веб-сайту более привлекательный и современный вид.

Выбор Bootstrap и flex-верстки для разработки основных элементов веб-страницы является обоснованным решением, поскольку это позволяет создать адаптивный, гибкий и функциональный веб-сайт. Благодаря предоставляемым инструментам и стилям, разработчики могут сосредоточиться на создании уникального дизайна и функциональности, не тратя много времени на написание повторяющегося кода и решение проблем совместимости с различными браузерами и устройствами.

Для веб-сайта было выбрано стильное оформление в спокойных тонах, сочетающее элементы классического дизайна. Этот стиль привлекает внимание пользователей за счет минимализма и отсутствия лишних деталей.

Основные цвета: Цветовая палитра представлена на рисунке 2.1

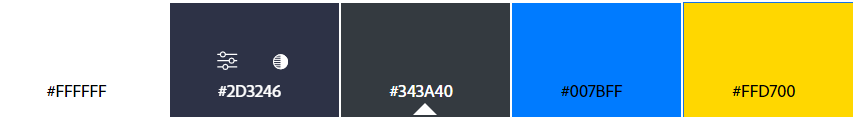


Рисунок 2.1 – Основные цветовые решения на странице

На сайте будет представлена только необходимая информация, с акцентом на контекст. Цветовая гамма выбрана сдержанной и не яркой.

**2.2. Выбор шрифтового оформления**

В качестве основного шрифта использован Open Sans. Выбор в сторону этого шрифта был сделан в качестве придания сайту более эстетичного вида. Шрифт

подключается с помощью правила @font-face.

Lato - популярный шрифт, разработанный для использования веб-дизайнерами и разработчиками веб-сайтов. Шрифт имеет чистый и современный вид, благодаря чему он подходит для широкого спектра веб-проектов, включая сайты, приложения, презентации и документы. Lato создан таким образом, чтобы обеспечивать высокую читаемость на экранах различных размеров и разрешений, что особенно важно для мобильных устройств. Он также имеет множество весов и начертаний, что позволяет выбирать наиболее подходящий для конкретного проекта стиль текста.

**2.3. Разработка логотипа**

Логотип был создан с помощью графического редактора Adobe Illustrator и его дизайн отличается минималистичностью. Было принято решение избежать излишней сложности и не использовать множество декоративных элементов. Результат логотипа представлен на рисунке 2.2.

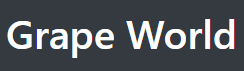


Рисунок 2.2 – Логотип сайта

**2.4. Разработка пользовательских элементов**

В проекте будут представлены такие элементы пользовательского интерфейса, как навигационное меню, кнопки, поле ввода информации пользователя, футер с ссылками на основную информацию и социальные сети.

Навигация – основной элемент для удобства сайта, это так называемая «карта», позволяющая перемещаться между различными страницами ресурса. Навигационное меню представлено на рисунке 2.3

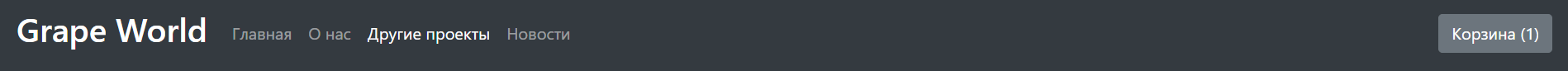


Рисунок 2.3 - Навигационное меню

Рассмотрим пользовательские элементы корзины, которую можно заполнять товарами, изменять количество товаров, просматривать итоговую стоимость. На рисунке 2.4 представлена кнопка, при нажатии на которую пользователь увидит перед собой вкладку корзины с пометкой количества товаров в ней, которая будет изменяться по мере заполнения корзины.



Рисунок 2.4 - Кнопка для заказа звонка

При переходе в корзину, в которой отсутствуют товары, пользователь увидит текст, уведомляющий его об этом. Рисунок 2.5.

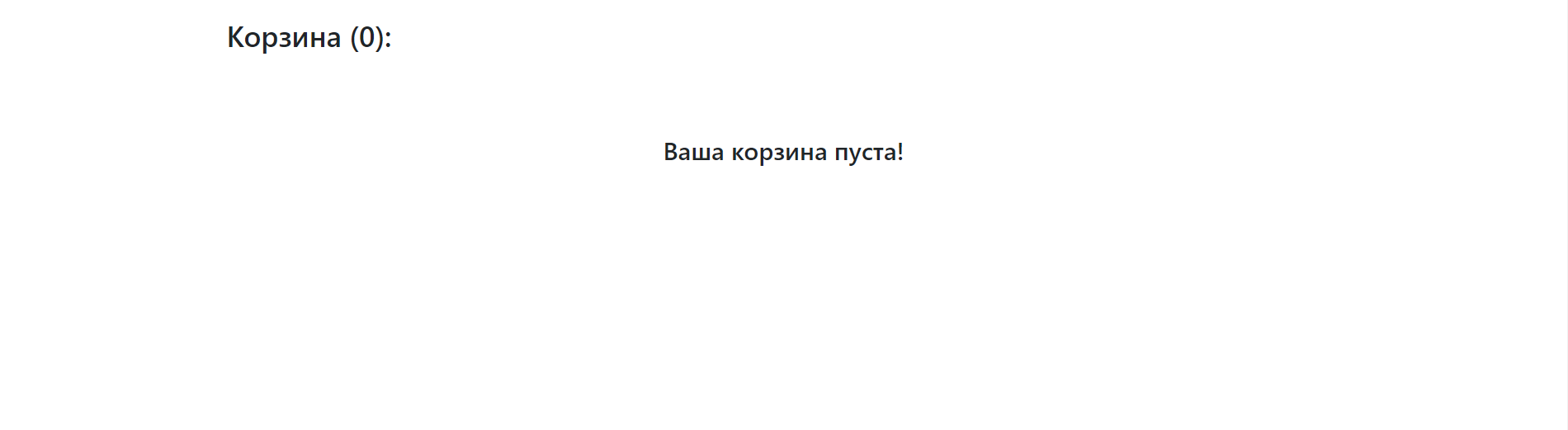


Рисунок 2.5 – Страница пустой корзины

В случае, когда пользователь выбрал товары и добавил их в корзину, перед ним появится страница со списком выбранных товаров (рисунок 2.6), с кнопкой для их удаления, при нажатии на которую пользователь увидит окно подтверждения действия (рисунок 2.7), а также поля и кнопки с возможностью изменения количества экземпляров товаров (рисунок 2.8)

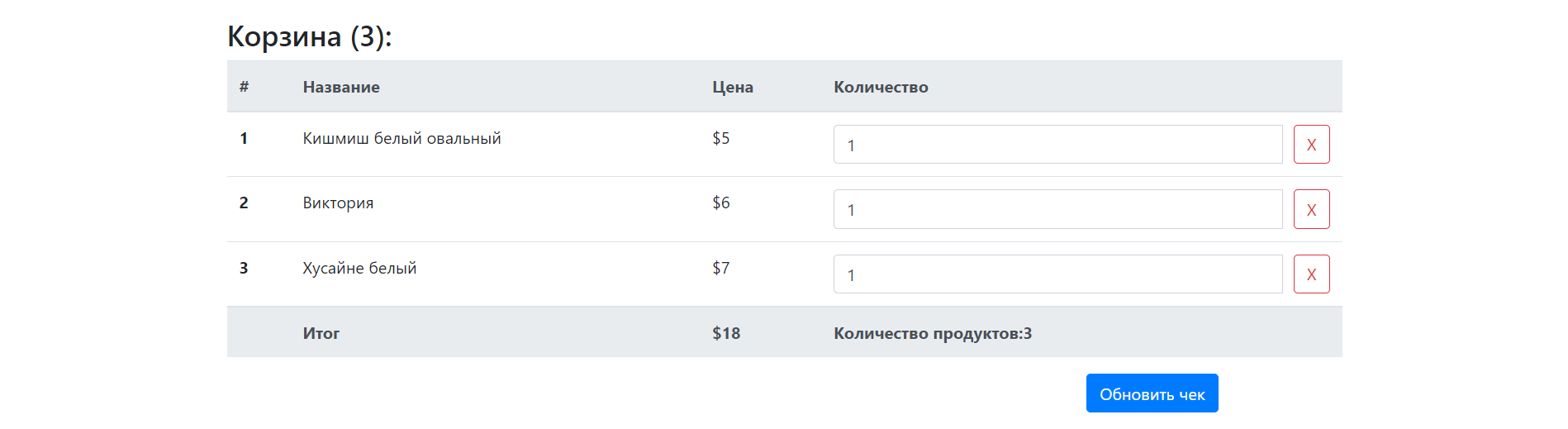


Рисунок 2.6 – Страница заполненной корзины

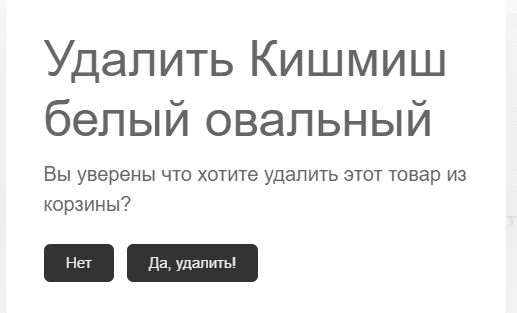


Рисунок 2.7 – Всплывающее окно с подтверждением действия при удалении

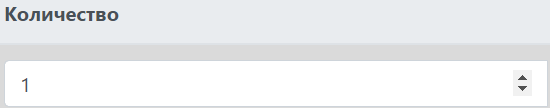


Рисунок 2.8 – Поле и кнопки изменения количества товаров

**2.5. Разработка спецэффектов**

На сайте реализованы динамические эффекты и анимации, которые добавляют интерактивности и делают сайт более привлекательным для пользователей. Например, при наведении на элементы, а также выбора элементов, навигационного меню появляется плавное изменение цвета.

Такое меню навигации представлено на рисунке 2.9.



Рисунок 2.9 - Навигационное меню

При наведении на одну из вспомогательных кнопок навигации, Пример представлен на рисунке 2.10.



Рисунок 2.10 – Изменение цвета при наведении на кнопки навигации

**2.6. Выводы**

В результате проведенной работы по макетированию веб-страниц, было достигнуто упорядочение и гармоничное расположение всех элементов на страницах. Был разработан дизайн, который включает в себя не только визуальное оформление, но и важные аспекты пользовательского интерфейса.

Важным этапом было подбор основной цветовой гаммы, которая соответствует общей концепции и атмосфере сайта. Цвета играют важную роль в создании эмоционального впечатления у пользователей, а правильно подобранная гамма помогает выделить ключевые элементы и обеспечить хорошую читабельность контента.

Также в процессе работы было уделено внимание шрифтовому оформлению. Подобранные шрифты должны быть удобочитаемыми и гармонично сочетаться с остальными элементами дизайна. Через шрифты можно подчеркнуть определенные акценты, а также передать настроение и стиль сайта.

Не менее важным аспектом является создание элементов пользовательского интерфейса, таких как кнопки, навигационные меню и другие интерактивные элементы. Их дизайн и расположение должны быть интуитивно понятными и удобными для пользователей, чтобы обеспечить легкую навигацию и взаимодействие с сайтом.

Для добавления динамики и эффектов на страницы были применены соответствующие техники, такие как анимации, плавные переходы и другие визуальные эффекты. Они придают страницам живость и интерактивность, что улучшает впечатление пользователей и делает сайт более привлекательным.

В целом, результатом проведенной работы стало создание красивого, функционального и привлекательного веб-сайта, который соответствует заданной концепции и требованиям пользователей. Эффективное макетирование позволило совместить эстетику и удобство использования, обеспечивая позитивное впечатление от посещения сайта.

**3. Реализация структуры веб-сайта**

**3.1. Структура** **приложения**

Данный курсовой проект был реализован при помощи React.

React — это библиотека JavaScript с открытым кодом для создания внешних пользовательских интерфейсов. В отличие от других библиотек JavaScript, предоставляющих полноценную платформу приложений, React ориентируется исключительно на создание представлений приложений через компонентами, которые сохраняют состояние и генерируют элементы пользовательского интерфейса.

Корневой компонент веб-страницы является index. Это компонент имеет секцию Switch, в которой перечислены все возможные компоненты других страниц. Благодаря компоненту BasePage, раскрываются реактивность и основные плюсы фреймворка React- компоненты Footer и Header, находящиеся на каждой из страниц, прогружаются всего один раз при входе на страницу сайта, в то время как остальные компоненты, необходимые для каждой из страниц, рендерятся при необходимости

На рисунке 3.1 представлен код файла index.js.

import React from 'react';

import ReactDOM from 'react-dom';

import { Route, Switch } from 'react-router-dom';

import 'bootstrap/dist/css/bootstrap.css';

import 'bootstrap/dist/js/bootstrap';

import './index.css';

import RootStoreProvider from './RootStoreProvider';

import BasePage from './components/pages/BasePage';

ReactDOM.render(

<RootStoreProvider>

<Switch>

<Route exact path="/"

render={routeProps => (<BasePage {...routeProps} pageName="Home" />)} />

<Route path="/shopping-cart"

render={routeProps => (<BasePage {...routeProps} pageName="ShoppingCart" />)} />

<Route path="/about"

render={routeProps => (<BasePage {...routeProps} pageName="About" />)} />

<Route path="/other"

render={routeProps => (<BasePage {...routeProps} pageName="Other" />)} />

<Route path="/news"

render={routeProps => (<BasePage {...routeProps} pageName="News" />)} />

<Route path="/product-detail/:productId"

render={routeProps => (<BasePage {...routeProps} pageName="ProductDetailsPage" />)} />

<Route path="\*"

render={routeProps => (<BasePage {...routeProps} pageName="PageNotFound" />)} />

</Switch>

</RootStoreProvider>,

document.getElementById('root')

);

Рисунок 3.1- index.js

**3.2. Использование Redux**

Redux – это инструмент для управления состоянием данных и пользовательским интерфейсом в приложениях JavaScript с большим количеством сущностей. Представляет собой библиотеку JavaScript.

В Redux общее состояние приложения представлено одним объектом JavaScript — state (состояние). Неизменяемое дерево состояний доступно только для чтения, изменить ничего напрямую нельзя. Изменения возможны только при отправке action (действия).

Действие (action) — это JavaScript-объект, который лаконично описывает суть изменения. Единственное требование к объекту действия — это наличие свойства type, значением которого обычно является строка.

На рисунке 3.4 представлен листинг Redux action компонента.

import \* as types from './action-types';

import { fetchProducts, fetchProductDetails } from '../lib/fetchProducts';

export function addToCartAction(product) {

return {

type: types.ADD\_TO\_CART,

payload: { Id: product.Id, Title: product.Title, Price: product.Price }

};

}

export function removeFromCartAction(productId) {

return {

type: types.REMOVE\_FROM\_CART,

productId: productId

};

}

export function updateCartAction(payload) {

return {

type: types.UPDATE\_CART,

payload

};

}

export function getProducts(payload) {

return {

type: types.FETCH\_PRODUCTS,

payload: fetchProducts()

};

}

export function getProductDetails(productId) {

return {

type: types.FETCH\_PRODUCT\_DETAILS,

payload: fetchProductDetails(productId)

};

}

Рисунок 3.4 - Redux action

import { combineReducers } from 'redux';

import cartReducer from './cartReducer';

import productsReducer from './productsReducer';

// комбинированный редьюсер

export default combineReducers({

cart: cartReducer,

products: productsReducer

});

Рисунок 3.5- redux reducer

**3.3. Выводы**

На данном этапе была реализована структура приложения, а также был использован Redux.

Также были получены дополнительные знания о разработке React приложений.

**4. Тестирование веб-сайта**

**4.1. Адаптивный дизайн веб-сайта**

Адаптивность — одно из ключевых требований, предъявляемых к современным сайтам. Ресурс должен одинаково хорошо демонстрироваться на экране компьютера, планшете, смартфоне. Для этого создается адаптивный дизайн и используется адаптивная верстка. В итоге процесс веб-разработки усложняется, требует дополнительных ресурсов. Но результат стоит вложений: сайт отлично позиционируется на всех основных типах устройств, что гарантирует полноценный охват аудитории.

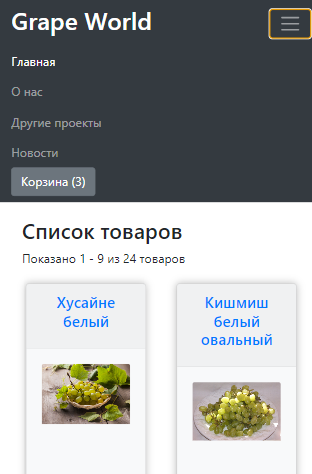
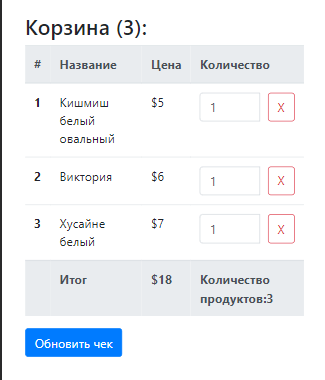
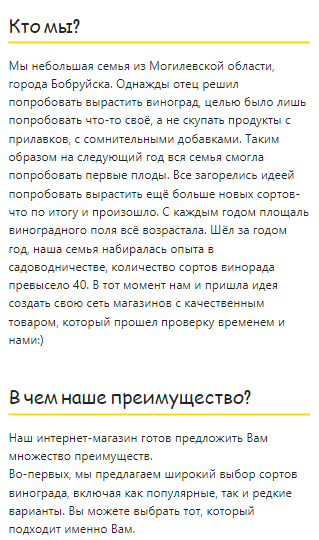
 

Рисунок 4.1-Адаптив страниц сайта

**4.2. Кроссбраузерность веб-сайта**

Кроссбраузерность - важное свойство, которое гарантирует правильное отображение веб-сайта в различных браузерах. Важно, чтобы сайт выглядел и работал одинаково хорошо как в Chrome, Firefox, Safari, Edge, так и в других популярных браузерах. Однако, для того чтобы добиться такой совместимости, необходимо учитывать особенности каждого браузера и применять соответствующие технологии и инструменты. Только тогда пользователи смогут увидеть сайт в полном объеме, независимо от того, какой браузер они используют.

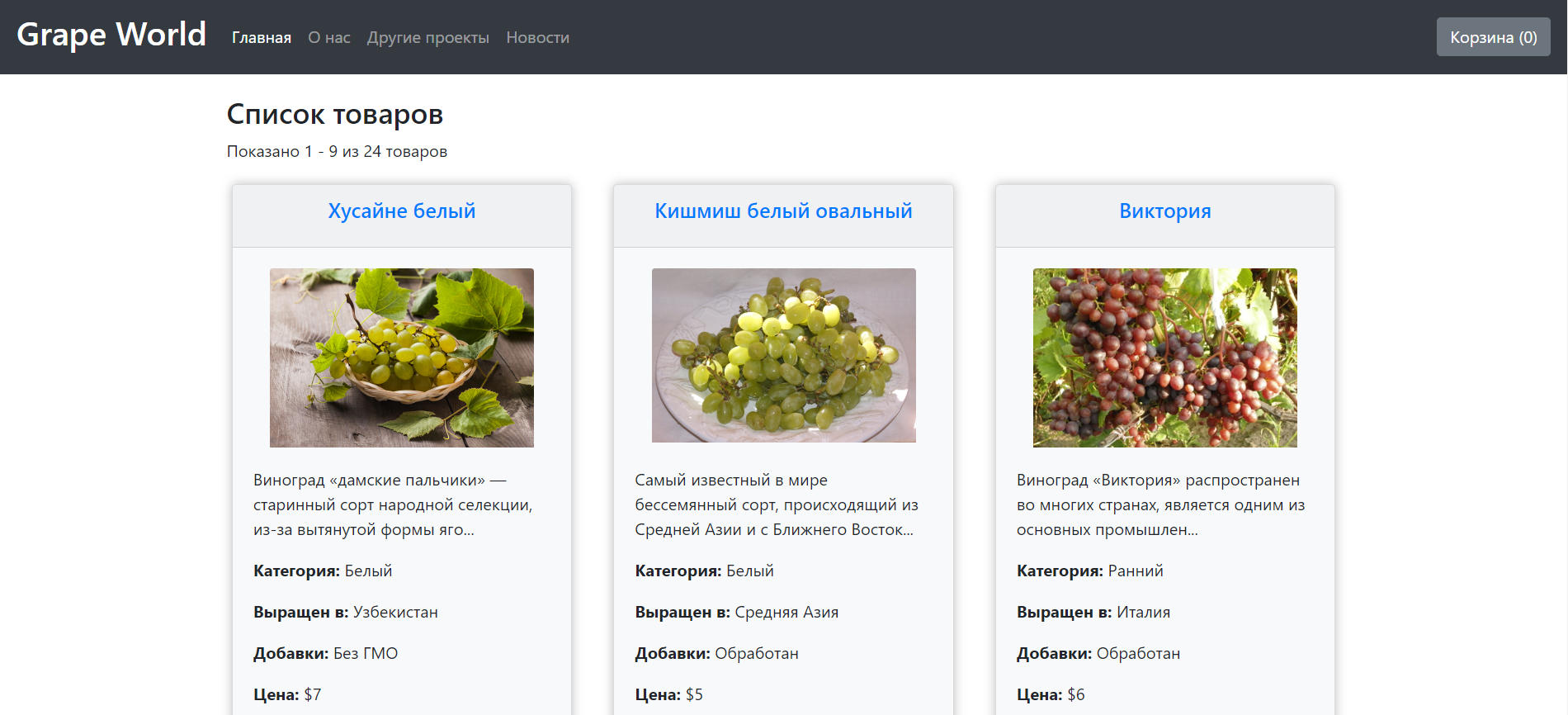


Рисунок 4.2 – Вид главной страницы в Microsoft Edge

Не требуется, чтобы сайт выглядел точно так же во всех браузерах. Важно учитывать предпочтения аудитории и создавать сайт, который отображается корректно в наиболее распространенных версиях популярных браузеров.

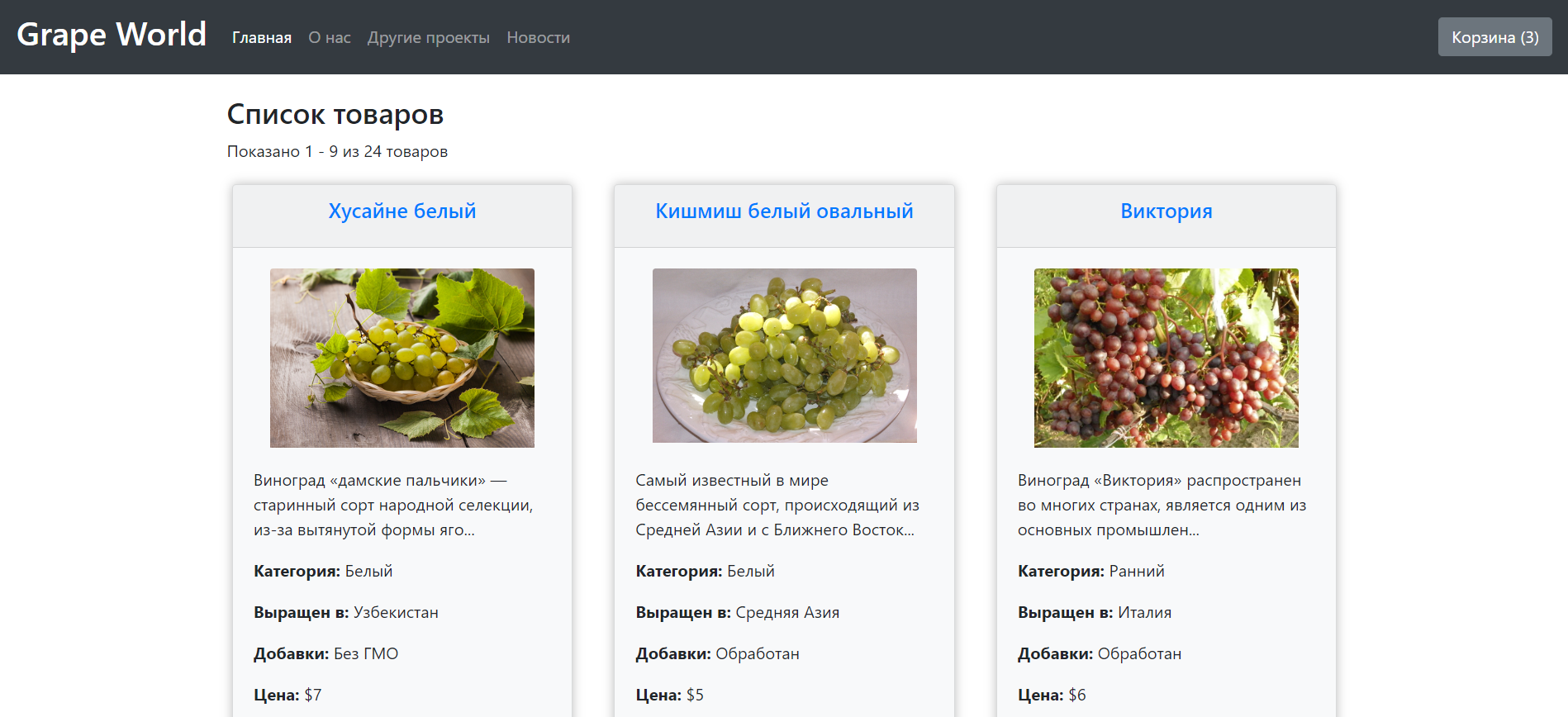


Рисунок 4.3 – Вид главной страницы в Google Chrome

Необходимо убедиться, что сайт отображается корректно и информация читабельна на разных браузерах. Для этого нет необходимости создавать идеальную визуализацию для каждого браузера, достаточно добиться схожей визуализации контента в популярных браузерах. Для этого можно использовать различные методы, например, вендорные префиксы, чтобы свойства работали корректно на разных браузерах. Главное, чтобы сайт отображался правильно, не имел недостатков в вёрстке и сохранял свою структуру во всех браузерах.

**4.3. Руководство пользователя**

Пользователь легко ориентируется на сайте благодаря интуитивно понятному интерфейсу и отсутствию ненужных элементов, что создает простоту и удобство использования. Шапка сайта позволяет переключаться между разделами сайта без необходимости возвращаться на главную страницу.

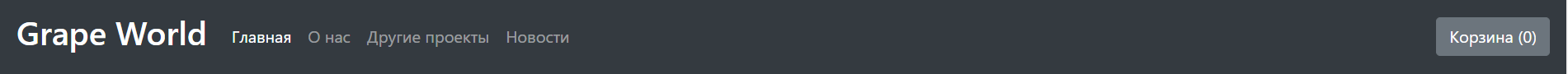


Рисунок 4.4 – Шапка сайта

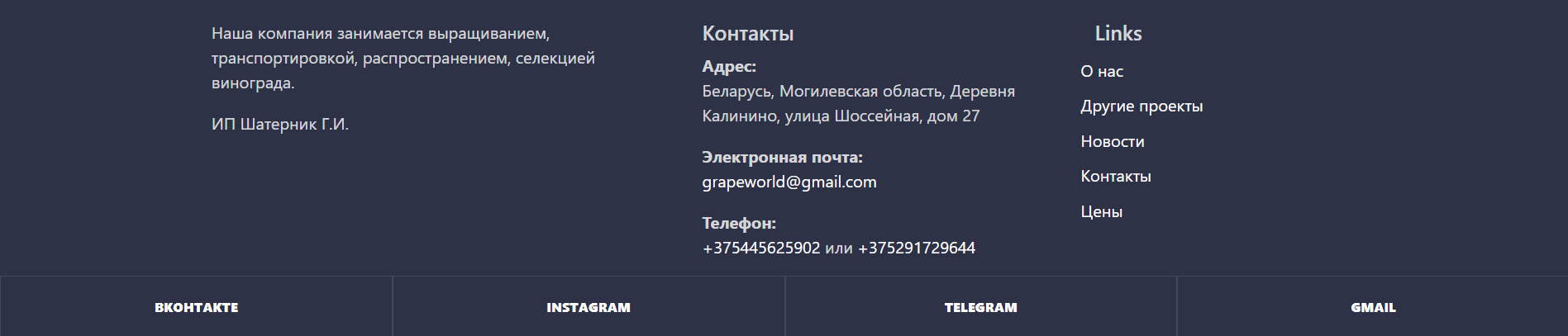


Рисунок 4.6 – «Подвал» сайта

**4.4. Выводы**

В этом разделе мы подробно рассмотрели процесс разработки нашего сайта, включая проверку его адаптивности на различных устройствах и исправление обнаруженных ошибок с помощью различных методов. Также мы уделили внимание кроссбраузерности и выявили проблемы с отображением сайта в некоторых браузерах, которые были успешно устранены. Важным этапом стала разработка подробного руководства пользователя, которое поможет нашим клиентам эффективно использовать функциональность сайта и легко ориентироваться в его блоках.

**Заключение**

Целью курсового проекта была реализация полноценного многостраничного веб-сайта интернет-магазина автомобилей. Сайт должен предоставлять доступ к информации. Необходимо было сделать удобный и практичный пользовательский интерфейс, сайт должен быть адаптивным и кроссбраузерным.

Сайт состоит из 3 страниц: главная страница, каталог автомобилей, страницы о компании Ауди.

Для каждой страницы использована адаптивная верстка, кроссбраузерность для поддержки большинством браузеров.

Для реализации макета страницы был проведен анализ других сайтов интернет-магазинов, выбрана самая нужная для пользователя информация, проанализирован интерфейс, элементы взаимодействия с пользователем, особое внимание уделялось адаптивности и кроссбраузерности некоторых аналогов.

При формировании технического задания были продумано содержание основных страниц веб-сайта.

Для написания кода веб-страницы использована среда разработки Visual Studio Code, в которой установлены плагины и расширения для удобной и облегченной разработки. Были разработаны пользовательские элементы, спецэффекты и логотип, для идентификации веб-сайта.

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.

При проектировании были разработаны прототипы, а далее созданы макеты дизайна страниц веб-сайта, при опоре на которые был реализован сам веб-сайт. Для разработки прототипов и макетов станиц было использована программа Figma, ранее не изученная в ходе учебного процесса.

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 были представлены в приложениях. Которые дают общую информацию о проекте.

Для выполнения поставленной цели курсового проекта были реализованы следующие задачи:

1. Анализ аналогов;
2. Анализ существующих языков разметки, инструментов библиотеки;
3. Разработка макета и прототипа;
4. Выбор способа верстки;
5. Выбор стилевого оформления;
6. Разработка логотипа;
7. Разработка основной структуры сайта;
8. Разработка пользовательских элементов;
9. Разработка спецэффектов;
10. Добавление веб-сайту адаптивности;
11. Кроссбраузерность веб-сайта;
12. Тестирование веб-сайта;

В результате проделанной работы был подготовлен веб-сайт интернет-магазина продажи винограда “Grape World”. Сайт работает стабильно, имеет некоторый функционал, и в дальнейшем он может подлежать эксплуатации.

Во время выполнения курсового проекта были использованы как теоретические, так и практические знания полученные в университете или выученные самостоятельно.

**Список использованных литературных источников**

1. Документация по Figma [Электронный ресурс]. — Режим доступа: <https://help.figma.com>
2. Документация по Sass/SCSS [Электронный ресурс]. — Режим доступа: <https://sass-scss.ru/guide/>
3. Документация по JavaScript/HTML [Электронный ресурс]. — Режим доступа: <https://developer.mozilla.org>
4. Введение в кроссбраузерное тестирование [Электронный ресурс]. — Режим доступа:

<https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Cross_browser_testing/Introduction>.

1. Документация по React.js [Электронный ресурс]. Режим доступа:

<https://react.dev/>

1. Документация по Redux.js [Электронный ресурс]. Режим доступа:

<https://redux.js.org/>

1. Документация по Bootstrap [Электронный ресурс]. Режим доступа:

<https://getbootstrap.com/>