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

Факультет Информационных технологий

Кафедра Информационных систем и технологий

Специальность 6-05-0611-01 «Информационные системы и технологии»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОГО ПРОЕКТА**

по дисциплине «Скриптовые языки программирования»

Тема: Web-сайт «Кофейня: "SALTYSWEET"»

**Исполнитель**

студент 2 курса 3 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Я. В. Шидловская

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, 2024

**Содержание**

[Введение 3](#_1fob9te)

[1 Обзор технических методов и программных средств разработки сайта 4](#_3znysh7)

[1.1 Обоснование необходимости создания программного продукта 4](#_2et92p0)

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

[1.3 Выбор средств реализации программного продукта 7](#_3dy6vkm)

[1.4 Вывод 7](#_1t3h5sf)

[2 Последовательность разработки содержания и структуры веб-сайта с использованием компонентов 8](#_4d34og8)

[2.1 Техническое задание 8](#_2s8eyo1)

[2.2 Выбор стилевого оформления 8](#_17dp8vu)

[2.3 Выбор шрифтового оформления 9](#_3rdcrjn)

[2.4 Разработка логотипа 9](#_26in1rg)

[2.5 Разработка пользовательских элементов 10](#_lnxbz9)

[2.6 Вывод 10](#_35nkun2)

[3 Программная реализация проекта 11](#_1ksv4uv)

[3.1 Вёрстка программного продукта 11](#_44sinio)

3.2 Использование таблиц стилей CSS [13](#_2jxsxqh)

[3.3 Вывод 15](#_3j2qqm3)

[4 Описание хода работы программного средства 16](#_1y810tw)

[4.1 Руководство пользователя 16](#_4i7ojhp)

[4.2 Кроссбраузерное тестирование 18](#_2xcytpi)

[4.3 Вывод 20](#_3whwml4)

[Заключение 21](#_2bn6wsx)

[Список использованных литературных источников 22](#_qsh70q)

[5 Приложения 23](#_3as4poj)

**Введение**

Актуальность темы обусловлена тем, что интернет стал неотъемлемой частью современного общества, и присутствие бизнеса в онлайн-пространстве является необходимым условием для его успешного развития. Создание веб-сайтов, в том числе для малых предприятий, таких как кофейни, позволяет эффективно привлекать клиентов, обеспечивать удобный доступ к информации об услугах и продуктах, а также повышать лояльность аудитории.

На данный момент в области разработки веб-сайтов существует широкий спектр технологий и подходов, которые позволяют создавать функциональные, эстетичные и удобные в использовании платформы. Развитие адаптивного дизайна, систем управления контентом и инструментов для взаимодействия с клиентами (онлайн-меню, форма обратной связи, система бронирования) стало стандартом для современных веб-сайтов, однако внедрение таких решений требует анализа задач конкретного бизнеса и их эффективной интеграции.

Целью данного курсового проекта является разработка веб-сайта для кофейни, который будет представлять ее в цифровом пространстве, предоставлять клиентам необходимую информацию об услугах и продукции, а также способствовать увеличению продаж и формированию узнаваемости бренда.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Провести анализ требований к функционалу сайта кофейни и изучить успешные примеры конкурентов.

2. Разработать структуру и макет сайта с учетом удобства навигации для пользователей.

3. Реализовать основные функциональные элементы сайта, включая информацию о кофейне, онлайн-меню, контактные данные, карту местоположения и форму обратной связи.

4. Обеспечить адаптивность сайта для корректного отображения на различных устройствах.

5. Провести тестирование и устранение возможных ошибок для обеспечения стабильной работы сайта.

Курсовой проект состоит из следующих разделов:

* **Первый раздел** посвящен анализу требований к сайту и изучению современных технологий, применяемых при разработке веб-ресурсов.
* **Второй раздел** описывает процесс проектирования структуры и макета сайта, включая выбор визуального стиля и компоновки элементов.
* **Третий раздел** раскрывает этапы реализации веб-сайта, включая описание используемых технологий и кода.
* **Четвертый раздел** содержит результаты тестирования сайта и рекомендации по его дальнейшему развитию и сопровождению.

Таким образом, проект направлен на создание современного веб-сайта для кофейни, который будет удовлетворять потребности бизнеса и клиентов.

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

**1.1 Обследование объекта и обоснование необходимости создания программного продукта**

Прежде чем приступить к разработке веб-сайта, необходимо проанализировать аналоги, чтобы понять, какие технологии используются для создания веб-сайтов, а также какие есть недостатки и преимущества веб-сайтов аналогов.

Есть множество разных сайтов кофеин. В качестве примера будут рассмотрены 2 сайта: «VARKA», «Cofix».

**«VARKA»**: при входе на сайт можно увидеть в шапке такие элементы как приложение для связи с кофейней, номер телефона. Логотип компании располагается в шапке профиля. Также в шапке сайта находится горизонтальное навигационное меню, где расположены все разделы сайта это всё представлен на рисунке 1.1

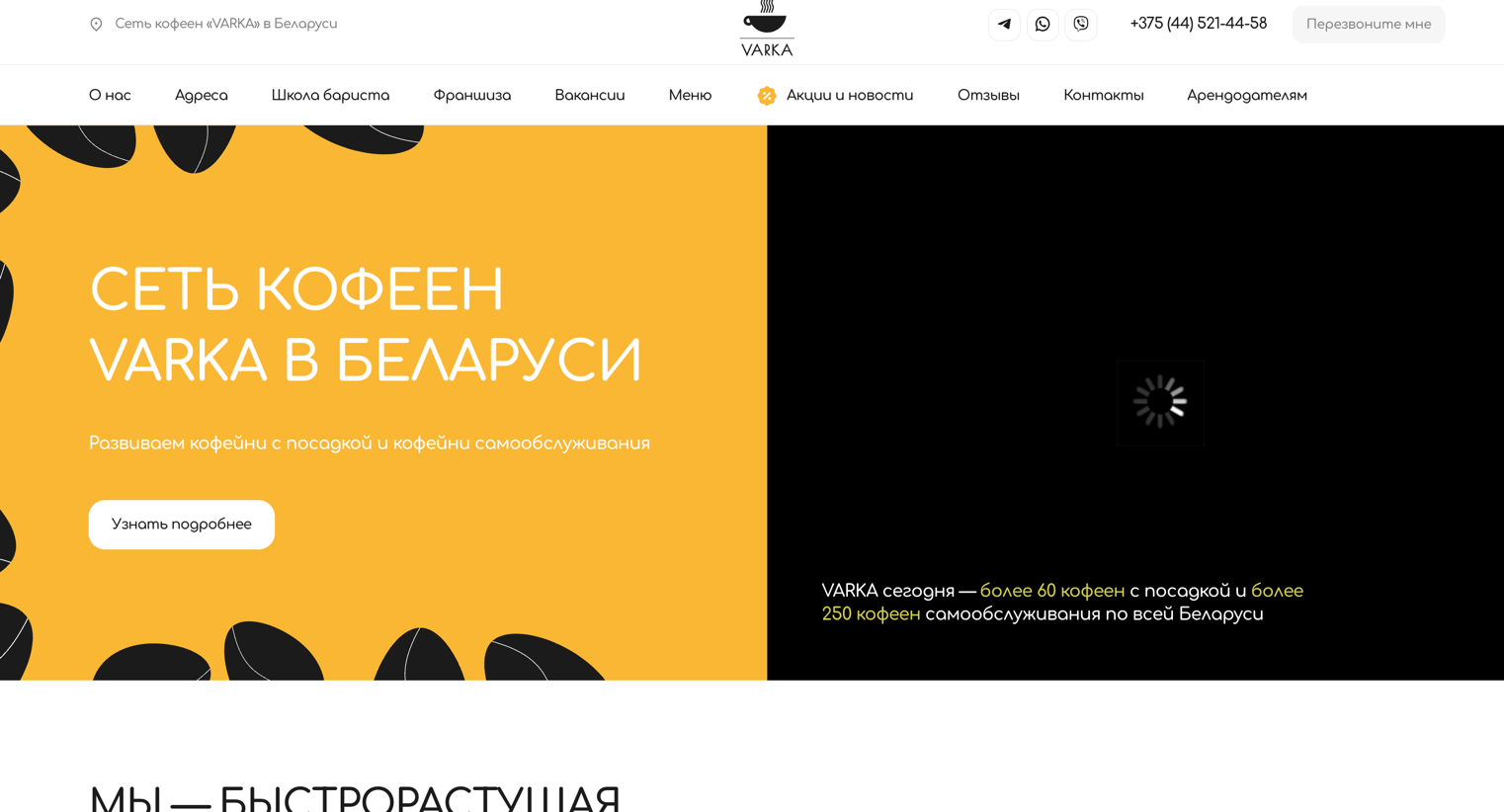


Рисунок 1.1 – Главная страница сайта «VARKA»

Сайт адаптивный, при уменьшении разрешения экрана в правом верхнем углу появляется выпадающее меню, в котором располагаются все разделы сайта. Дизайн представлен на рисунке 1.2.

Изображение выглядит как текст, снимок экрана, графический дизайн, книга

Автоматически созданное описание

Рисунок 1.2–Мобильная версия сайта «VARKA»

Из минусов данного веб-сайта можно выделить то, что на главной странице представлено множество разделов и информации, что может затруднить восприятие для новых пользователей. Из-за этого посетители могут чувствовать себя потерянными.

**«Cofix»**: На рисунке 1.3 можно увидеть дизайн сайта и главную страницу. Логотип компании располагается в шапке профиля. В шапке также можно увидеть горизонтальное меню со всеми разделами сайта.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 1.3 – Главная страница сайта «Cofix»

Мобильная версия сайта представлена на рисунке 1.4.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 1.4 – Мобильная версия сайта «Cofix»

Плюс этого веб-сайта в том, что активно использует акции и специальные предложения для привлечения клиентов.

## **1.2 Формирование требований к программному продукту**

Формирование требований к веб-сайту для кофейни — это ключевой этап разработки. Веб-сайт должен иметь удобный интерфейс с интуитивной навигацией и адаптивным дизайном для мобильных устройств. Важно предоставить полное меню с возможностью онлайн-заказа. Система лояльности должна поддерживать регистрацию пользователей, а также возможность оставлять и комментировать отзывы о напитках и сервисе.

Сайт должен содержать информацию о кофейне, включая ее контактные данные с интеграцией карт. В плане производительности важно обеспечить быструю загрузку страниц и защиту персональных данных пользователей с использованием безопасных методов оплаты. Кросс-браузерная совместимость с основными браузерами также имеет значение.

Контент сайта должен включать фотографии меню и описания блюд и напитков. Четкое формирование требований поможет создать качественный и востребованный продукт.

**1.3 Выбор средств реализации программного продукта**

Для реализации программного продукта применяется редактор кода WebStorm. При создании веб-приложения используются библиотека React и другие библиотеки, от которых зависят данные библиотеки. Веб-приложение разрабатывается преимущественно на языке JavaScript с использованием JSX — расширения этого языка, позволяющего использовать HTML5 в JavaScript-коде. В качестве стилевого оформления проекта применяются каскадные таблицы стилей CSS3. Для запуска и сборки приложения используются Node.js и его пакетный менеджер npm.

**1.4 Вывод**

В рамках данного раздела были проанализированы аналогичные решения, выявлена их сущность и специфика восприятия целевой аудиторией. Полученные знания были использованы при разработке последующего продукта, обладающего высоким качеством и отвечающего требованиям пользователей. Были определены цели и задачи для создания веб-приложения.

**2. Последовательность разработки содержания и структуры веб-сайта с использованием компонентов**

## **2.1 Техническое задание**

Основными требованиями к программному продукту являются адаптивность, лёгкость. На создаваемом сайте не может быть больших программных функций или очень сложных анимаций – всё это может сильно замедлить скорость загрузки информации и ухудшить её восприятие. Адаптивность разрабатывалась исходя из существующих актуальных браузеров и устройств.

Будет сделана разработка макета, создание прототипам с помощью HTML, CSS, ReactJS, Node.js.

Технические средства:

* HTML
* CSS
* ReactJS

Целевая аудитория:

Для лучшего понимания того, как должен выглядеть сайт кофейни, нужно понять, на какую целевую аудиторию будет рассчитан данный веб-ресурс.

Целевая аудитория для сайта кофейни включает любителей кофе, студентов, работающих людей, семьи с детьми, туристов, заботящихся о здоровье, гурманов и фрилансеров. Эти группы ищут качественный кофе, уютное место для учебы или работы, интересные гастрономические предложения и атмосферу для общения.

Также это могут быть и конкуренты, которым требуется оценить конкурентоспособность данного парфюмерного магазина.

Рекламодатели могут тоже заинтересоваться данным сайтом в случае, если он будет достаточно популярным, и могут предложить сотрудничество, что может положительно сказаться на жизнеспособности сайта.

**2.2 Выбор стилевого оформления**

При определении с дизайном и цветовой палитрой сайта были выбраны светлые тона, хорошо сочетающиеся между собой. Используется лаконичный шрифт, для улучшенной привлекательности сайта. На рисунке 2.2 изображена используемая цветовая палитра сайта.

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2.2 – Цветовая палитра сайта

Акцент выбора цветов делался на универсальную тематику: коричневые и зелёные тона хорошо акцентируют важные элементы, а универсальности добавляют оттенки черного, отлично подходящие в этом случае. Ключевой цвет для оформления текста - черный(#000000) используется в оформлении заголовка и в выделениях в текстах. # D6A04F используется для цветов кнопки по всем компонентам сайта. Зелёные цвета используются для выделения ценников. #FFFFFF используется для заднего фона страниц сайта, и в некоторых текстах сайта.

**2.3 Выбор шрифтового оформления**

В данном проекте используется один единственный шрифт Merriweather. Он присутствует во всех надписях сайта, и придает своими очертаниями сайту лаконичность, тем самым делая его привлекательным.

**2.4 Разработка логотипа**

Логотип –  графический знак, используемый территориальными образованиями, коммерческими предприятиями, организациями и частными лицами для повышения узнаваемости и распознаваемости в социуме. Логотип – это img изображение на рисунке 2.1

Изображение выглядит как логотип, дизайн

Автоматически созданное описание

Рисунок 2.1 – Логотип

**2.5 Разработка пользовательских элементов**

На странице будут представлены такие элементы пользовательского интерфейса как шапка сайта с горизонтальным меню, представлена на рисунке 2.2



Рисунок 2.2 – шапка сайта и горизонтальное меню

Для связи между клиентом и компанией на страницу будет добавлена контактная форма такая как: имя, фамилия, email, номер телефона и итоговая сумма покупки, которая представлена на рисунке 2.3.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 2.3 - Меню регистрации

**2.6 Вывод**

На данном этапе была определена структура компонентов, стилевое оформление, в том числе логотип, шрифты, цветовая палитра. Также были определены ключевые элементы пользовательского интерфейса.

3. Программная реализация проекта

## **3.1 Вёрстка программного продукта**

Этот код представляет собой компонент корзины покупок на React.

|  |
| --- |
| import React , { *Fragment*, useState, useContext } from 'react'; import { *ItemsContext* } from '../Context/CartContext'; import cartMug from '../../../src/assets/img/cart-mug.png'; import Spinner from '../Layout/Spinner'; import { *Link* } from 'react-router-dom';  const Cart = () => {   const [loading, setLoading] = useState(true);   const {items, removeItem} = useContext(*ItemsContext*);  setTimeout(() => {  setLoading(false);  }, 1500);   let acum = 0;   return (  <Fragment>  {loading ? (<Spinner/>) :  <>  <div className='itemContainer justify-content-start'>  <div><h1>Ваша корзина:</h1></div>  <div className='boxCart'>  {items.length !== 0 ?  <div className='cartList'>  {items.map( item => {  const pricePerProduct = item.price \* item.quantity;  acum = acum + pricePerProduct;  return(  <div key={item.id}>  <section className='boxCart--item'>  <div>  {item.title}  </div>  <div className='text-nowrap text-muted'>  Количество: {item.quantity}  </div>  <div className='text-nowrap' style={{color: "#0f7032"}}>  {pricePerProduct} руб  </div>  <div>  <button  type='button'  className='btn btn-danger btn-sm'  onClick={() => removeItem(item.id)}  >X  </button>  </div>  </section>  <div className='cartDivider'></div>  </div>    )  })}  </div> :  <div className='d-flex flex-column align-items-center my-auto gap-4 text-center'>  <img src={cartMug} alt="cart" style={{width: '15vw', maxWidth: '125px'}}/>  <h1 className='fs-1 sm: fs-5'>Корзина пуста :(</h1>  <Link to='/menu'><button className='btn btn-outline-secondary m- </div>}  </div>  {items.length !== 0 ? <div className='box d-flex w-100 justify-content-between align-items-center'>  <img src={cartMug} alt="cart" style={{width: '7vw', maxWidth: '120px'}}/>  <h1 className='m-0 fs-3'>Итого: {acum} руб</h1>  </div> : null}  </div>  {items.length === 0 ? null : <div className='cartButton'><Link to='/order'><button className='buttons' style={{backgroundColor: '#198754'}}>Далее</button></Link></div>}    </>}  </Fragment>  ); 3'>Вернуться к меню</button></Link>  ); };  export default Cart; |

Листинг 3.1 – Компонент корзины покупок

Этот компонент демонстрирует основные принципы работы с состоянием и контекстом в React, а также управление отображением в зависимости от состояния данных.

**3.2 Использование таблиц стилей CSS**

\* {  
 font-family: "Merriweather", serif;  
 font-weight: bold;  
}  
\*::selection {  
 background: rgb(201, 167, 117);  
}  
html {  
 overflow-x: hidden;  
}  
/\* Spinner \*/  
.cup {  
 position: absolute;  
 top: 40%;  
 left: 50%;  
 transform: translate(-50%, -50%);  
 width: 150px;  
 height: 180px;  
 border: 6px solid #e9e9e9;  
 border-top: 2px solid transparent;  
 border-radius: 5px 5px 15px 15px;  
 background: url(../img/liquid.png);  
 background-size: cover;  
 background-repeat: repeat-x;  
 animation: animate 15s linear infinite;  
 box-shadow: 0 0 0 6px rgb(75, 75, 75), 0 10px 35px #e9e9e9;  
}  
.cup::before {  
 content: '';  
 position: absolute;  
 width: 50px;  
 height: 80px;  
 border: 6px solid rgb(75, 75, 75);  
 right: -56px;  
 top: 30px;  
 border-radius: 2px 35px 35px 2px;  
}  
@keyframes animate {  
 0%  
 {  
 background-position: 0px 125px;  
 }  
 20%  
 {  
 background-position: 500px 75px;  
 }  
 40%  
 {  
 background-position: 1000px 25px;  
 }  
 80%  
 {  
 background-position: 1500px 0px;  
 }  
 100%  
 {  
 background-position: 2000px 0px;  
 }  
}  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
body {  
 background: rgb(233, 233, 233);  
}  
.home {  
 width: 100vw;  
 height: 100vh;  
 background-image: url(../img/homepage.png);  
 background-color: #f1ede7;  
 background-blend-mode: multiply;  
 background-size: cover;  
 display: flex;  
 flex-flow: row wrap;  
 align-items: center;  
 align-content: center;  
 justify-content: center;

}

Листинг 3.2 – фрагмент App.css

В приложении был использован стандартный CSS3 для написания стилей компонентам приложения. В листинге приведена реализация стиля для навигационного меню в шапке сайта, с указанием атрибутов, придающих ему плавность и красоту.

**3.2 Вывод**

На данном этапе была реализована структура веб-сайта. При вёрстке использовалась СSS Flexbox. Для анимации использовалась одна из библиотек React. Код был проверен на валидность

**4. Описание хода работы программного средства**

**4.1 Руководство пользователя**

При заходе на сайт кофейни посетителя встречает главная страница, которая содержит в себе картинку и посередине её есть кнопка, предлагающая покупателям перейти к меню сайта.

Изображение выглядит как кофейная чашка, кухонный прибор, устройство, иллюстрация

Автоматически созданное описание

Рисунок 4.1 – Главная страница интернет-магазина

На странице меню пользователь может просмотреть представленное меню, увидеть краткую информацию о продукции если что-то понравится можно перейти нажав на кнопку подробнее.

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 4.2 – Меню

На детальной странице продукции мы видим, что можем добавить в корзину или купить. После чего продукция добавляется в корзину.

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 4.3 – Детальная страница

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Автоматически созданное описание

Рисунок 4.4 – Страница корзины

На странице корзины, пользователь видит товара которые он выбрал на итоговую сумму и кнопку далее. После нажатия кнопки далее мы попадаем на страницу регистрации.

## **4.2 Кроссбраузерное тестирование**

Очень важно, чтобы наш сайт был кроссбраузерным. 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.

У каждого браузера есть свои встроенные, экспериментальные или нестандартные свойства и для того, чтобы они корректно работали было принято решение о внедрении вендорных префиксов. При отображении сайта на различных устройствах стилевое оформление немного изменяется. Пример работы в Google chrome на рисуноке 4.5.

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 4.5 – Сайт в браузере Google Сhrome.

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 4.6 – Сайт в браузере Яндекс.

Изображение выглядит как текст, снимок экрана, еда

Автоматически созданное описание

Рисунок 4.7 – Сайт в браузере Safari.

Можно увидеть хорошую работу сайта в трёх браузерах.

## **4.3 Вывод**

На данном этапе мы проверили как будет вести себя веб-сайт на разных устройствах, а также протестировали его в большом количестве браузеров. По итогу выполнения данного этапа для веб-сайта был разработан адаптивный дизайн и кроссбраузерность. Также была проведена проверка в Linter, сайт является валидным.

**Заключение**

Целью данной курсовой работы было создание веб-сайт на тему «Кофейня: "SALTYSWEET"»

В рамках проекта был проведён анализ существующих решений, что позволило выявить ключевые тренды в дизайне и функционале современных веб-приложений. На этапе формирования технического задания были детально продуманы основные страницы сайта и их содержание.

Для разработки приложения был выбран современный набор инструментов, обеспечивающий эффективную реализацию поставленных задач. В ходе проектирования была разработана структура приложения, а также представлены примеры реализации некоторых функций. Особое внимание уделялось созданию пользовательского интерфейса и фирменного логотипа, что придало веб-сайту уникальный стиль и повысило его узнаваемость.

Пользователям было предоставлено руководство по использованию веб-сайта, чтобы сделать процесс работы с сайтом максимально удобным. На этапе тестирования было подтверждено, что приложение корректно работает во всех популярных браузерах, что гарантирует его кроссбраузерность.

В результате был создан кроссбраузерный веб-сайт кофейни, включающий уникальные элементы интерфейса и высокую функциональность. Приложение демонстрирует надёжность, и интегрирует современные технологии для обеспечения удобства пользователей.

Список использованных литературных источников

1. Документация React[Электронный ресурс]:[React – A JavaScript library for building user interfaces (reactjs.org)](https://reactjs.org/)
2. Building a REST API with Express, Node, and MongoDB [Электронный ресурс] / Справочник Режим доступа: <https://www.mongodb.com/languages/express> mongodb-rest-api-tutorial/.
3. Документация HTML5 [Электронный ресурс]. Режим доступа: [https://metanit.com/web/html5/](https://metanit.com/web/html5/%20) – Дата доступа: 6.10.2024
4. Документация CSS3[Электронный ресурс]. Режим доступа: [https://htmlbook.ru/css](https://htmlbook.ru/css%20) – Дата доступа: 10.10.2024
5. Сайт varka [Электронный ресурс]. Режим доступа: <https://varkacoffee.by/>– Дата доступа: 10.09.2024
6. Сайт cofix [Электронный ресурс]. Режим доступа: <https://cofix.global/ru-by/main/minsk/> – Дата доступа: 10.09.2024

**5. Приложения**

import React from 'react';  
import ReactDOM from 'react-dom/client';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
  
const root = ReactDOM.createRoot(*document*.getElementById('root'));  
root.render(  
 <React.StrictMode>  
 <App />  
 </React.StrictMode>  
);

Листинг index.js

import React from 'react';  
// Components  
import NavBar from './components/Layout/NavBar';  
import Home from './components/Home/Home';  
import ItemListContainer from './components/ItemListContainer/ItemListContainer';  
import ItemDetailContainer from './components/ItemDetailContainer/ItemDetailContainer';  
import Cart from './components/Cart/Cart'  
import OrderForm from './components/Cart/Order';  
import AboutUs from './components/AboutUs/About';  
// Routing  
import { BrowserRouter, Routes, Route } from 'react-router-dom';  
// Context  
import { ItemsProvider } from './components/Context/CartContext';  
  
function App() {  
   
 return (  
 <ItemsProvider>   
 <BrowserRouter>  
 <NavBar/>  
 <Routes>  
 <Route path='/' element={<Home/>}/>  
 <Route path='/menu' element={<ItemListContainer/>}/>  
 <Route path='/item/:id' element={<ItemDetailContainer/>}/>  
 <Route path='/categories/:categoryid' element={<ItemListContainer/>}/>  
 <Route path='/cart' element={<Cart/>}/>  
 <Route path='/order' element={<OrderForm/>}/>  
 <Route path='/about' element={<AboutUs/>}/>  
 </Routes>  
 </BrowserRouter>  
 </ItemsProvider>  
 );  
};  
  
export default App;

Листинг App.jsx

import React from 'react';  
import { *Link* } from 'react-router-dom';  
  
const Home = () => {  
 return (  
 <div className='home'>  
 <div className='home--text'>  
 <h1>Кофе и Вафли лучшая часть дня.</h1>  
 <p>Посмотри наше меню и выбери то, что понравится именно тебе!</p>  
 <Link to='/menu'><button className='buttons buttons--home'>Меню</button></Link>  
 </div>  
 <div className='home--image'>  
 </div>  
 </div>  
 );  
}  
  
export default Home;

Листинг Home.jsx

import React from 'react';  
import Slider from "react-slick";  
import './AboutUs.css';  
import "slick-carousel/slick/slick.css";  
import "slick-carousel/slick/slick-theme.css";  
  
import coffeeShopImg1 from '../../assets/img/salty-and-sweet-image-1.png';  
import coffeeShopImg2 from '../../assets/img/salty-and-sweet-image-2.png';  
import coffeeShopImg3 from '../../assets/img/salty-and-sweet-image-3.jpg';  
  
const AboutUs = () => {  
 const sliderSettings = {  
 dots: true,  
 infinite: true,  
 speed: 500,  
 slidesToShow: 1,  
 slidesToScroll: 1,  
 autoplay: true,  
 autoplaySpeed: 3000,  
 arrows: false,  
 };  
  
 return (  
 <div className='about-us'>  
 <div className='about-us--header'>  
 <h1 className='about-us--title'>О нас</h1>  
 <p className='about-us--subtitle'>  
 Добро пожаловать в <strong>Salty & Sweet</strong> — место, где вы сможете расслабиться,  
 насладиться лучшими кофейными напитками и попробовать наши знаменитые сладкие вафли.  
 Мы предлагаем не только вкусную еду, но и уютную атмосферу, которая делает нас особенными.  
 </p>  
 <p className='about-us--description'>  
 У нас вы найдёте широкий выбор десертов, сезонные напитки и качественный сервис.  
 Наши сотрудники всегда готовы помочь вам с выбором и создать для вас незабываемые впечатления.  
 Мы стремимся быть не просто кофейней, а местом для встреч, творчества и отдыха.  
 </p>  
 </div>  
  
 <div className='about-us--content'>  
 <div className='about-us--slider'>  
 <Slider {...sliderSettings}>  
 <div>  
 <img  
 src={coffeeShopImg1}  
 alt='Интерьер кофейни 1'  
 className='slider-photo'  
 />  
 </div>  
 <div>  
 <img  
 src={coffeeShopImg2}  
 alt='Интерьер кофейни 2'  
 className='slider-photo'  
 />  
 </div>  
 <div>  
 <img  
 src={coffeeShopImg3}  
 alt='Интерьер кофейни 3'  
 className='slider-photo'  
 />  
 </div>  
 </Slider>  
 </div>  
  
 {/\* Информация о кофейне \*/}  
 <div className='about-us--info'>  
 <h2 className='info-title'>Контакты</h2>  
 <p><strong>Адрес:</strong> ул. Притыцкого, д. 83, г. Минск</p>  
 <p><strong>Телефон:</strong> <a href="tel:+375293648541">+375 (29) 364-85-41</a></p>  
 <p>  
 <strong>Соцсети:</strong><br />  
 <a href="https://t.me/littleYanni" target="\_blank" rel="noopener noreferrer">  
 Наш Telegram  
 </a> |  
 <a href="https://www.instagram.com/saltysweet\_minsk/" target="\_blank" rel="noopener noreferrer">  
 Instagram  
 </a>  
 </p>  
 </div>  
 </div>  
  
 {/\* Карта \*/}  
 <div className='about-us--map'>  
 <iframe  
 title="Карта: как добраться до Salty & Sweet"  
 src="https://yandex.ru/map-widget/v1/-/CHAfyIYV"  
 width="560"  
 height="400"  
 style={{ border: 0, borderRadius: '8px', position: "relative" }}  
 frameborder="1"  
 allowfullscreen="true"  
 loading="lazy">  
 </iframe>  
 </div>  
  
 </div>  
 );  
};  
  
export default AboutUs;

Листинг AboutUs.jsx

import React, {useState, createContext} from 'react';  
  
export const *ItemsContext* = createContext();  
  
export const ItemsProvider = ({children}) => {  
 const [items, setItems] = useState([]);  
  
 const addItem = (item, quantity) => {  
 if(isInCart(item.id)) {  
 const index = items.findIndex( e => e.id === item.id);  
 items[index].quantity = items[index].quantity + quantity;  
 setItems([...items]);  
 } else {  
 setItems([...items, { ...item, quantity }])  
 }  
 };  
  
 const removeItem = itemId => {  
 return setItems(items.filter( item => item.id !== itemId));  
 };  
  
 const clear = () => {  
 setItems([]);  
 };  
  
 const isInCart = id => {  
 return items.find( item => item.id === id );  
 };  
   
 return (  
 <ItemsContext.Provider value={{items, addItem, removeItem, clear, isInCart}}>  
 {children}  
 </ItemsContext.Provider>  
 );  
};

Листинг CartContext.jsx

import React, {*Fragment*, useState, useContext} from 'react';  
import { *ItemsContext* } from '../Context/CartContext';  
import { *Link* } from 'react-router-dom';  
import SuccessMsg from './SuccessMsg';  
import ErrorMsg from './ErrorMsg';  
import *actualDate* from '../../helpers/ActualDate';  
// Firebase - Firestore  
import { collection, addDoc } from 'firebase/firestore';  
import { *db* } from '../../firebase/firebaseConfig';  
  
  
const OrderForm = () => {  
  
 const {items, clear} = useContext(*ItemsContext*);  
  
 const [client, setClient] = useState({  
 name: '',  
 lastName: '',  
 email: '',  
 phone: '',  
 });  
  
 const pricePerProduct = items.map( item => {  
 return item.quantity \* item.price;  
 })  
  
 const totalPrice = pricePerProduct.reduce( (prev, curr) => {  
 return prev + curr;  
 }, 0)  
  
 const { name, lastName, email, phone } = client;  
  
 const initialState = {  
 buyer: {  
 name: name,  
 lastName: lastName,  
 email: email,  
 phone: phone,  
 },  
 items: [...items],  
 date: *actualDate*,  
 total: totalPrice  
 }  
  
 const validForm = (clientData) => {  
 const buyerData = *Object*.values(clientData);  
 return buyerData.some( item => item === '');  
 }  
  
 const [values, setValues] = useState(initialState);  
 const [orderID, setOrderID] = useState('');  
  
 const [error, setError] = useState(false);  
  
 const handleOnChange = (e) => {  
 const { value, name } = e.target;  
 setClient({  
 ...client,  
 [name]: value  
 });  
 };  
  
 const onSubmit = async (e) => {  
 e.preventDefault();  
  
 if(validForm(client)) {  
 setError(true);  
 } else {  
 const docRef = await addDoc(collection(*db*, "orders"), {initialState});  
   
 setOrderID(docRef.id);  
 setValues(initialState);  
 setError(false);  
 clear();  
 };  
 }  
 return (  
 <Fragment>  
 <div className='orderContainer'>  
 <div className='boxOrder'>  
 <h5>Введите информацию о себе:</h5>  
 <form className='orderForm' onSubmit={onSubmit}>  
 <input  
 placeholder='Имя'  
 name='name'  
 value={values.name}  
 onChange={handleOnChange}  
 />  
 <input  
 placeholder='Фамилия'  
 name='lastName'  
 value={values.lastName}  
 onChange={handleOnChange}  
 />  
 <input  
 placeholder='email'  
 name='email'  
 value={values.email}  
 onChange={handleOnChange}  
 />  
 <input  
 placeholder='Номер телефона'  
 name='phone'  
 value={values.phone}  
 onChange={handleOnChange}  
 />  
 <h4 className='mt-3'>Итого: {totalPrice} руб</h4>  
 <button  
 className='buttons mx-auto'  
 disabled={totalPrice === 0 ? true : false}  
 >Купить  
 </button>  
 </form>  
 {error === true && <ErrorMsg/>}  
 {orderID && <SuccessMsg orderID={orderID}/>}  
 </div>  
 {orderID && <Link to='/menu' className='mx-auto'><button className='btn btn-outline-secondary m-3'>Вернуться к меню</button></Link>}  
 </div>  
 </Fragment>  
 );  
};  
  
export default OrderForm;

Листинг Cart(Order.jsx)