Содержание

[Введение 2](#_Toc122551536)

[1. Обзор технических методов и программных средств разработки веб-сайта. 3](#_Toc122551537)

[1.1 Выбор средств реализации программного продукта 3](#_Toc122551538)

[1.2. Техническое задание 4](#_Toc122551539)

[1.3. Постановка задач программного продукта 4](#_Toc122551540)

[1.4. Выводы 4](#_Toc122551541)

[2. Последовательность разработки содержания и структуры веб-сайта. 5](#_Toc122551542)

[2.1 Выбор стилевого оформления 5](#_Toc122551543)

[2.2. Разработка логотипа компании 5](#_Toc122551544)

[2.3. Выводы 6](#_Toc122551545)

[3. Описание хода работы программного средства 7](#_Toc122551546)

[3.1. Общие элементы 7](#_Toc122551547)

[3.2. Главная страница 8](#_Toc122551548)

[Заключение 9](#_Toc122551549)

[Список использованных источников 10](#_Toc122551550)

[Приложение 1 Листинг веб-страниц 11](#_Toc122551551)

**Введение**

За последние годы, технологии, связанные с всемирными компьютерными сетями и интернетом, сделали огромный скачок в развитии. Благодаря этому стало легче продвигать на рынок разные товары и услуги. К примеру, большое количество человек заинтересованы в услугах, предоставляемых интернет-магазинами. Последние исследования и опросы показали, что практически 5 млрд. человек активно пользуются интернетом, а в Беларуси этот показатель достигает 82,8% населения.

В наше время, каждая успешная организация имеет свой веб-сайт. Для успешного привлечения клиентов обязателен приятный дизайн и функционал веб-сайта, постоянные обновления и усовершенствования, позволяющие дать пользователю то, что ему необходимо. К примеру, оформить заказ товаров, находясь дома, в интернет-магазине.

Интернет-магазин спортивного питания - это отличная реализация компромисса между удобством осмотра и заказа интересующего товара, а также затраченного времени. Реализация данной темы на основе библиотеки компонентов React.js – отличное решение, позволяющее создать качественный веб-сайт, интегрировав все необходимые условия и задачи.

Тема курсовой работы: разработка веб-сайта интернет-магазина спортивного питания “Sports Nutritions”.

Целью курсового проекта является разработка веб-сайта интернет-магазина зоотоваров, используя JavaScript-библиотеку React.js.

Задачи, решаемые в рамках курсового проекта:

* рассмотреть основные шаги, связанные с разработкой современных веб-сайтов;
* последовательно разработать содержание и структуру веб-сайта c использованием компонентов;
* описать ход работы программного средства с копиями экрана пользовательского интерфейса основных элементов системы, а также ключевых функций разработанного программного кода.

Всё выше сказанное говорит об актуальности выбранной темы.

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

**1.1 Выбор средств реализации программного продукта**

Для достижения поставленной цели курсового проекта были использованы следующие средства: JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов – React.js, языки разметки – HTML5, каскадные таблицы стилей – CSS.

React — JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

HTML – стандартизированный язык разметки документов для просмотра веб-страниц в [браузере](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80). Веб-браузеры получают HTML документ от сервера по протоколам [HTTP](https://ru.wikipedia.org/wiki/HTTP)/[HTTPS](https://ru.wikipedia.org/wiki/HTTPS) или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

HTML5 – [язык для структурирования и представления](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) содержимого [всемирной паутины](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0). Пятая версия HTML.

CSS – каскадные таблицы стилей. Является формальным языком описания внешнего вида документа, написанного с использованием языка разметки.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript.

Из-за требования адаптивности сайта был выбран фреймворк Bootstrap. Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Фреймворк предлагает достаточное количество шаблонов и готовых решений; Кросс–браузерность и адаптивность. Сайты, созданные с помощью Bootstrap, идентично отображаются на разных устройствах и в современных браузерах; Простота использования.

Для создания прототипа(макета) сайта был выбран графический онлайн-редактор Figma. Был выбран именно этот редактор так как Figma может создавать базы компонентов и позволяет экономить массу времени и очень упрощает работу.

В качестве среды разработки выбрана Visual Studio Code – это популярная бесплатная программа для разработчиков, которая поддерживает все необходимые языки программирования для разработки проекта: JavaScript, HTML. CSS и т.п.

React Router DOM – это модуль узла, который предназначен для маршрутизации в веб-приложениях. Он позволяет инженерам создавать маршруты для одностраничного приложения React.

**1.2. Техническое задание**

Целью данного проекта является реализация разработки web-сайта с помощью библиотеки React.js, предоставляющего возможность к просмотру или покупке доступных товаров, возможность выбора необходимого товара по определённым критериям, возможность добавления товаров в корзину, а также возможность авторизоваться или зарегистрироваться.

Качественный веб-сайт должен быть готов выполнить следующие требования:

* у сайта должна быть адаптивность под все основные виды устройств;
* [сайт](http://fancytravel.by/) должен иметь простой и понятный для любого пользователя интерфейс;
* текст должен быть читабельным и понятным;
* у сайта должен быть свой уникальный дизайн;
* функционал сайта должен реализовывать все необходимые базовые действия, без ненужных функций.

На сайте выполнены следующие задачи:

* погрузка компонентов вместо полной перезагрузки веб-страницы;
* использование библиотеки Redux.js для работы с состояними;
* использование библиотеки RouterDOM;
* использование функций и классовых компонентов.

**1.3. Постановка задач программного продукта**

По итогу выполненного курсового проекта должны быть реализованы такие задачи как:

* веб-сайт, содержащий страницу с товарами;
* пользователи должны иметь возможность добавить товары в корзину;
* должны быть созданы формы регистрации и авторизации;
* у сайта должен быть свой уникальный дизайн;
* функционал сайта должен реализовывать все необходимые базовые действия, без ненужных функций.

Всё выше сказанное говорит об актуальности реализуемого проекта.

**1.4. Выводы**

В данном разделе были выбраны средства для реализации программного продукты. Было чётко описание техническое задание, прилагаемое к данному проекту. Были поставлены конкретные задачи для создания качественного, многофункционального проекта. Все описанные подпункты понадобятся для достижения поставленной цели.

1. **Последовательность разработки содержания и структуры веб-сайта.**

**2.1 Выбор стилевого оформления**

Сайт включает в себя:

– Главная

– О нас

– Контакты

На главной странице находится вся основная информация с возможностью обратиться к модальным окнам, предоставляющим информацию о разделах «О нас» и «Контакты» на другие страницы по кнопочным ссылкам.

При открытии модального «О нас» появляется краткое описание компании, история и цели.

Для получения справки о контактах данной компании необходимо обратиться к модальному окну «Контакты».

В качестве основных цветов веб-сайта были выбраны следующие цвета: белый и бирюзовый. Веб-сайт выполнен в минималистическом стиле (см. рис. 1)

Используемые шрифты: «Inter».

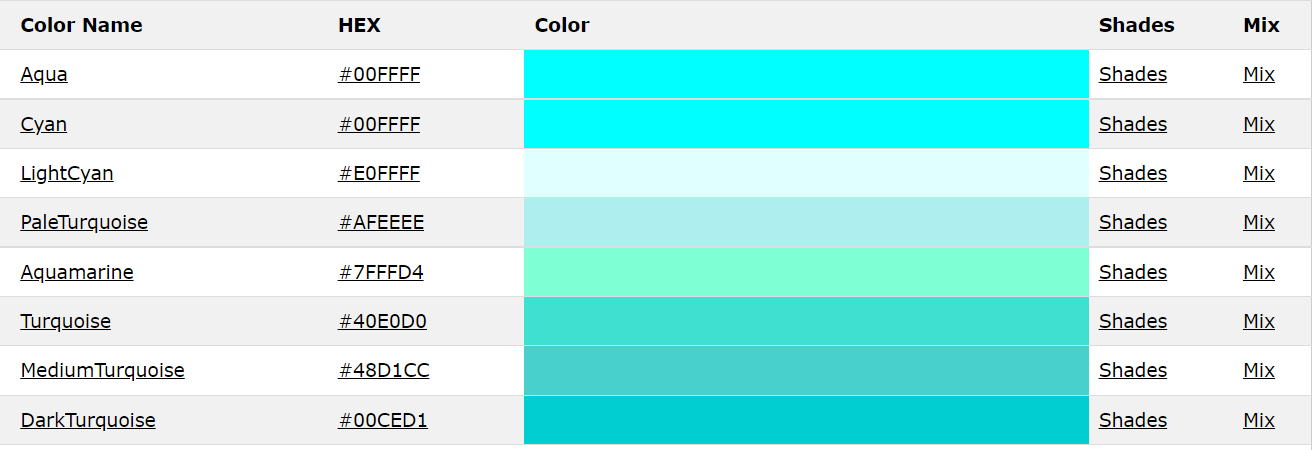


Рисунок 1 – Палитра используемых цветов

**2.2. Разработка логотипа компании**

Главный критерии логотипа – это уникальность и простота к восприятию. Логотип должен легко запоминаться, а также иметь общую ассоциацию у пользователей для быстрой запоминаемости. Поскольку разрабатываемый веб-сайт связан со спортивным питанием, то в основе названия компании и логотипа лежит слово «Спортивное питание.

**2.3. Выводы**

В данном разделе было определено и описано стилевое оформление, а также создан логотип веб-сайта. Цветовая гамма веб-сайта определена в зелёном и серых цветах. В виде основного шрифта используется шрифт «Inter».

**3. Описание хода работы программного средства**

Весь процесс начинается с создания необходимых папок, директорий и файлов. Подключаются библиотеки стилей, иконок, функциональные библиотеки, также библиотеки готовых компонентов.

Создаются необходимые файлы компонентов, реализуется структура и логика.

**3.1. Общие элементы**

В компоненте «Header.js» импортируются все необходимые компоненты, react, useState, а также стили для данного компонента. Реализуется навигационное меню со ссылками на модальные окна, а также корзина для заказанных товаров. Цвет фона белый, на котором расположено навигационное меню, логотип и корзина.

Главное меню расположено вверху веб-сайта и представляет собой блок со ссылками, ведущими на модальные окна и корзину (рис. 2).



Рисунок 2 – Меню на главной странице

При наведении на элементы навигационного меню, появляется курсор поинтер. Для закрытия модальных окон необходимо нажать на соответствующую иконку в открытом окне, либо воспользоваться одной из доступных кнопок.

При нажатии на иконку корзины появится блок, отображающий состав и состояние корзины. При условии добавленных в корзину товаров будет отображена краткая информация о товарах, такая как: название, изображение, стоимость и количество (рис. 3).

По итогу всех добавленных товаров и их количеству считается общая сумма справа внизу. Для закрытия корзины необходимо повторно нажать на её иконку. После закрытия, корзина сохраняет всю информацию о добавленных товарах.

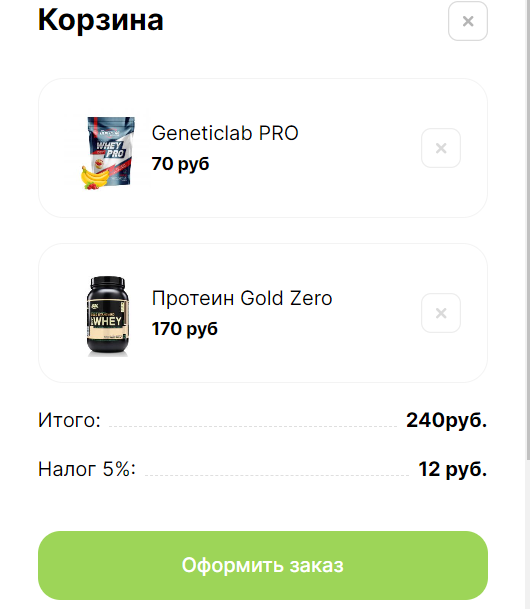


Рисунок 3 – Корзина с добавленным товаром

**3.2. Главная страница**

В самом начале главной страницы располагается заголовок и поиск, благодаря которому, мы можем быстро находить определенный товар.

После заголовка и поиска расположены все блоки товаров, представленные на веб-сайте. Блоки товаров реализованы с помощью компонентов “Card.js” Основная информация о выводимом товаре заполняется данными из массива «App.js» в компонент «Card.js», с помощью передаваемых свойств используется в компоненте «Card.js», реализуется вывод всех товаров на страницу веб-сайта (рис. 4).

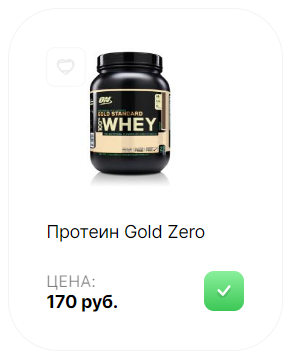
****

Рисунок 4 – Категории товаров

Каждый товар имеет своё изображение, название, тип товара, цену, кнопку добавления товара в корзину. При наведении на изображение товара изменяется тип курсора и размер изображения. При наведении на кнопку добавления элемента в корзину изменяется её размер и цвет.

На основной странице веб-сайта выводится только название, изображение, тип товара, цена и кнопка добавления товара в корзину. Блоки товаров расположены в виде сетки, что позволяет быстро и удобно просмотреть их.

**Заключение**

Во время процесса создания курсового проекта были закреплены знания и навыки по дисциплине «Компьютерные языки разметки». Был повышен уровень знаний в использовании HTML, SASS, JavaScript и JSX.

Была объяснена актуальность темы курсового проекта, разработан дизайн и реализована структура сайта, были отображены этапы разработки данного сайта вместе с его элементами и содержимым веб-страниц.

Таким образом, в итоге выполненной работы был создан веб-ресурс по теме курсового проекта.

**Список использованных источников**

1. Современный учебник JavaScript [Электронный ресурс] Режим доступа: <https://learn.javascript.ru/>
2. Справочник HTML [Электронный ресурс] Режим доступа: <http://htmlbook.ru/>
3. Жиляк Н.А. Информационные технологии. Разработка Web-сайта на основе HTML с использованием JavaScript : методические указания к выполнению курсовых / [сост. Н. А. Жиляк]. - Минск : БГТУ, 2012. - 59, [1] c.

**Приложение 1 Листинг веб-страниц**

**Index.js**

import React from 'react';

import { BrowserRouter as Router } from 'react-router-dom';

import ReactDOM from 'react-dom/client';

import './index.scss';

import App from './App';

import 'macro-css';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

  <React.StrictMode>

    <Router>

    <App />

    </Router>

  </React.StrictMode>

);

**App.js**

import React from 'react';

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

import Header from './components/Header';

import Drawer from './Drawer/Drawer';

import Home from './pages/Home';

import Favorites from './pages/Favorites';

import AppContext from './сontext';

//превращение массива объектов в реактовкские компоненты (автоматически, т.е не каждый раз вбивая наш компонент)

//то что мы передаем в компонент оно будет хранится в объекте props и из этого объекта мы сможем вытаскивать эти свойство(элементы)

const items = [

  {id: 1, title: 'Matriza', price: 120, imageUrl: "/img/nutritions/1.jpg" },

  {id: 2, title: 'Iso Whey Zero', price: 140, imageUrl: "/img/nutritions/2.jpg" },

  {id: 3, title: 'Протеин Gold Zero', price: 170, imageUrl: "/img/nutritions/3.jpg" },

  {id: 4, title: 'Geneticlab PRO', price: 70, imageUrl: "/img/nutritions/4.jpg" },

  {id: 5, title: 'Hydro Whey', price: 145, imageUrl: "/img/nutritions/5.jpg" },

  {id: 6, title: 'Soy Pro', price: 210, imageUrl: "/img/nutritions/6.jpg" },

  {id: 7, title: 'Melatonin', price: 130, imageUrl: "/img/nutritions/7.jpg" },

  {id: 8, title: 'Whey Protein Complex', price: 195, imageUrl: "/img/nutritions/8.jpg" },

  {id: 9, title: 'AmfNoXpress', price: 175, imageUrl: "/img/nutritions/9.jpg" },

  {id: 10, title: 'BCAA', price: 90, imageUrl: "/img/nutritions/10.jpg" },

  {id: 11, title: 'Chain Acids', price: 105, imageUrl: "/img/nutritions/11.jpg" },

  {id: 12, title: 'Mega Amino', price: 125, imageUrl: "/img/nutritions/12.jpg" },

];

function App() {

  const [cartOpened, setCartOpened] = React.useState(false);   //  для закрытия корзины

  const [cartItems, setCartItems] = React.useState([]); //добавление карточек

  const [favorites, setFavorites] = React.useState([]);

  const [searchValue, setSearchValue] = React.useState(''); // поиск

  const onAddToCart = (obj) => {

    if(cartItems.find((item) => item.title === obj.title)){

      setCartItems(cartItems => cartItems.filter(item => item.title !== obj.title));

    }

    else{

    setCartItems([...cartItems, obj]);

    }

  }

  const onChangeSearchInput = (event) =>{

    setSearchValue(event.target.value);

  }

  const onRemoveItem  = (title) =>{

    setCartItems((cartItems) => cartItems.filter(item => item.title !== title)); //берем массив возьми внем все что есть и пробежись по нему и отфильтруй тот элемент которого Id тот который мы передали функции

  }

  const onAddToFavorite = (obj) =>{

    if (favorites.find((obj) => obj.title == obj.title)){

      setFavorites(cartItems.filter(item => item.title !== obj.title));

    }

     else

    {

    setFavorites([...cartItems, obj]);

    }

  }

  return (

    <AppContext.Provider value={{cartItems, favorites, setCartOpened, setCartItems}}>

    <div className="wrapper clear">

      <Drawer items={cartItems} onRemove = {onRemoveItem} onClose={() => setCartOpened(false)} opened = {cartOpened} />

      {/\* {cartOpened ?  : null} при клике на крестик корзина закрывается \*/}

      <Header onClickCart={() => setCartOpened(true)} /> {/\* props для открытия корзины \*/}

      <Route path= "/" exact>

        <Home items = {items} searchValue = {searchValue} setSearchValue = {setSearchValue} onChangeSearchInput = {onChangeSearchInput} onAddToFavorite = {onAddToFavorite} onAddToCart = {onAddToCart}/>

      </Route>

      <Route path= "/favorites" exact>

        <Favorites items = {favorites} onAddToFavorite = {onAddToFavorite}/>

      </Route>

    </div>

    </AppContext.Provider>

  );

}

export default App;

**Header.js**

import React from 'react';

import {Link} from 'react-router-dom';

import AppContext from '../сontext';

function Header(props) {

  const {cartItems} = React.useContext(AppContext);

  const totalPrice = cartItems.reduce((sum, obj) => obj.price + sum, 0 );

  return (

    <header className="d-flex justify-between align-center p-40">

        <div className="d-flex  align-center">

          <img width={40} height={40} src='/img/logo.png' />

          <div>

            <h3 className="text-uppercase">

              Sports Nutrition

            </h3>

            <p className="opacity-5">

              Магазин лучшего спортивного питания

            </p>

          </div>

        </div>

      <ul className="d-flex">

        <li onClick={props.onClickCart} className="mr-30 cu-p">

          <svg width="20" height="20" viewBox="0 0 20 20" fill="none" xmlns="http://www.w3.org/2000/svg">

            <path d="M7.54548 18.1818C7.99735 18.1818 8.36366 17.8155 8.36366 17.3636C8.36366 16.9117 7.99735 16.5454 7.54548 16.5454C7.09361 16.5454 6.72729 16.9117 6.72729 17.3636C6.72729 17.8155 7.09361 18.1818 7.54548 18.1818Z" stroke="#9B9B9B" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" />

            <path d="M16.5455 18.1818C16.9973 18.1818 17.3637 17.8155 17.3637 17.3636C17.3637 16.9117 16.9973 16.5454 16.5455 16.5454C16.0936 16.5454 15.7273 16.9117 15.7273 17.3636C15.7273 17.8155 16.0936 18.1818 16.5455 18.1818Z" stroke="#9B9B9B" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" />

            <path d="M1 1H4.27273L6.46545 11.9555C6.54027 12.3321 6.7452 12.6705 7.04436 12.9113C7.34351 13.1522 7.71784 13.2801 8.10182 13.2727H16.0545C16.4385 13.2801 16.8129 13.1522 17.112 12.9113C17.4112 12.6705 17.6161 12.3321 17.6909 11.9555L19 5.09091H5.09091" stroke="#9B9B9B" stroke-width="2" stroke-linecap="round" stroke-linejoin="round" />

          </svg>

          <span>{totalPrice} руб.</span>

        </li>

      </ul>

    </header>

  );

}

export default Header;

**Drawer.js**

import React from "react";

import styles from './Drawer.module.scss';

import AppContext from '../сontext';

import Info from "../components/Info";

function Drawer({ onClose, onRemove, items = [], opened }) {

  const {cartItems} = React.useContext(AppContext);

  const totalPrice = cartItems.reduce((sum, obj) => obj.price + sum, 0 );

  const {setCartItems} = React.useContext(AppContext);

  const [isOrderComplete, setIsOrderComplete] = React.useState(false);

  const onClickOrder = () => {

    setIsOrderComplete(true);

    setCartItems([]);

  }

  return (

    <div className={`${styles.overlay} ${opened ? styles.overlayVisible : ' '}`}>

      <div className={styles.drawer} style={{ display: "flex", flexDirection: "column" }}>

        <h2 className="mb-30 d-flex justify-between ">Корзина <img onClick={onClose} className="cu-p" src="/img/btn-remove.svg" alt="Close" />

        </h2>

        {

          items.length > 0 ? <div>

             <div className="items" style={{ flex: 1, overflow: "auto" }}>

            {

              items.map((obj) => (<div className="cartItem d-flex align-center mb-20">

                <div style={{ backgroundImage: `url(${obj.imageUrl})` }} className="cartItemImg">

                </div>

                <div className="mr-20 flex">

                  <p className="mb-5">{obj.title}</p>

                  <b>{obj.price} руб</b>

                </div>

                <svg onClick={() => onRemove(obj.title)} className="removeBtn" width="32" height="32" viewBox="0 0 32 32" fill="none" xmlns="http://www.w3.org/2000/svg">

                  <rect x="0.5" y="0.5" width="31" height="31" rx="7.5" fill="white" stroke="#DBDBDB" />

                  <path d="M20.0799 18.6155L17.6311 16.1667L20.0798 13.718C21.0241 12.7738 19.5596 11.3093 18.6154 12.2536L16.1667 14.7023L13.7179 12.2535C12.7738 11.3095 11.3095 12.7738 12.2535 13.7179L14.7023 16.1667L12.2536 18.6154C11.3093 19.5596 12.7738 21.0241 13.718 20.0798L16.1667 17.6311L18.6155 20.0799C19.5597 21.0241 21.0241 19.5597 20.0799 18.6155Z" fill="#B5B5B5" />

                </svg>

              </div>))

            }

          </div>

           <div className="cartTotalBlock">

           <ul>

             <li>

               <span>Итого: </span>

               <div></div>

              <b>{totalPrice}руб.</b>

             </li>

             <li>

               <span>Налог 5%: </span>

               <div></div>

               <b>{(totalPrice / 100)\*5} руб.</b>

             </li>

           </ul>

           <button onClick={onClickOrder} className="greenButton">Оформить заказ </button>

         </div>

          </div> : <Info title = {isOrderComplete ? "Заказ оформлен":"Корзина пустая"} description = {isOrderComplete ? "Ваш заказ №20 скоро будет передан курьерской доставке" : "Добавьте хотя бы одну спортивную добавку, чтобы сделать заказ"} imageUrl = {isOrderComplete ? "/img/order.png" : "/img/empty.jpg"} />

        }

      </div>

    </div>

  );

}

export default Drawer;

**Info.jsx**

import React from "react";

import AppContext from "../сontext";

const Info = ({ tilte, imageUrl, description}) => {

    const {setCartOpened} = React.useContext(AppContext);

    return(

        <div>

            <div class="cartEmpty d-flex align-center justify-center flex-column flex">

            <img class="mb-20" width={120} src={imageUrl} alt="Empty"></img>

            <h2>{tilte}</h2>

            <p class="opacity-6">{description}</p>

            <button onClick={() => setCartOpened(false)} className="greenButton">

              <img src="/img/arrow.svg" alt="Arrow" />

              Вернуться назад

            </button>

          </div>

        </div>

    )

}

export default Info;

**Home.jsx**

import Card from '../components/Card/Card';

function Home({items, searchValue, setSearchValue, onChangeSearchInput, onAddToFavorite, onAddToCart})

{

    return( <div className="content p-40">

    <div className="d-flex align-center justify-between mb-40">

      <h1>{searchValue ? `Поиск по запросу: "${searchValue}"` : 'Все спортивное питание'}</h1>

      <div className="search-block d-flex">

        <svg width="15" height="43" viewBox="0 0 16 16" fill="none" xmlns="http://www.w3.org/2000/svg">

          <path d="M15.25 15.25L11.8855 11.8795L15.25 15.25ZM13.75 7.375C13.75 9.06576 13.0784 10.6873 11.8828 11.8828C10.6873 13.0784 9.06576 13.75 7.375 13.75C5.68424 13.75 4.06274 13.0784 2.86719 11.8828C1.67165 10.6873 1 9.06576 1 7.375C1 5.68424 1.67165 4.06274 2.86719 2.86719C4.06274 1.67165 5.68424 1 7.375 1C9.06576 1 10.6873 1.67165 11.8828 2.86719C13.0784 4.06274 13.75 5.68424 13.75 7.375V7.375Z" stroke="#E4E4E4" stroke-width="2" stroke-linecap="round" />

        </svg>

        {searchValue && ( <img onClick={() => setSearchValue('')} className = "clear cu-p" src= "/img/btn-remove.svg" alt = "Clear" />)}

        <input onChange={onChangeSearchInput} value = {searchValue} placeholder="Поиск..." />

      </div>

    </div>

{/\* проходимся по массиву перед рендером будет искл все item в которых title нету того что написано в searchValue \*/}

    <div className="d-flex flex-wrap">

      {

        items.filter(item => item.title.toLowerCase().includes(searchValue.toLowerCase())).map((item, title) => (

          <Card

            key={title}

            id={item.id}

            title={item.title}

            price={item.price}

            imageUrl={item.imageUrl}

            onFavorite={(obj) => onAddToFavorite(obj)}

            onPlus={(obj) => onAddToCart(obj)} />

        ))

      }

    </div>

  </div>);

}

export default Home;