**МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВОСИБИРСКОЙ ОБЛАСТИ**

Государственное автономное профессиональное образовательное учреждение Новосибирской области

«Новосибирский колледж печати и информационных технологий»



ОТЧЁТ

по лабораторной работе №1

Разработка одностраничного веб-приложения с использованием HTML, CSS и JavaScript»

по дисциплине: «Проектирования и разработка веб приложений»

|  |  |
| --- | --- |
| Выполнил:  Студент гр. 9-22, ИСиП  Андрущук Д. В. | Преподаватель:  Перевалов Г.Г. |

Новосибирск 2024 г

Оглавление

[1. ВВЕДЕНИЕ 3](#_Toc177160556)

[1.1. Цель и задачи 3](#_Toc177160557)

[1.2. Требования к выполнению 3](#_Toc177160558)

[2. РАЗРАБОТКА ПРОЕКТА 3](#_Toc177160559)

[2.1. Index.html– главная страница 4](#_Toc177160560)

[2.2. Style.css – подключение стилей для главной страницы 5](#_Toc177160561)

[2.3. Script.js – функционал сайта. 8](#_Toc177160562)

[3. ДЕМОНСТРАЦИЯ РАБОТЫ САЙТА 14](#_Toc177160563)

[4. ВЫВОД 15](#_Toc177160564)

# ВВЕДЕНИЕ

## Цель и задачи

Цель: Разработать одностраничное веб-приложение с использованием HTML, CSS и JavaScript

Задача:

* Создать одностраничное веб-приложение для онлайн-заказа товаров (например, магазина электроники).

## Требования к выполнению

Функциональность:

* Главная страница содержит список товаров (название, цена, описание, кнопка "Добавить в корзину").
* При клике на кнопку "Добавить в корзину" товар добавляется в виртуальную корзину (без перезагрузки страницы).
* Реализовать отображение корзины (количество товаров и общая сумма) в правом верхнем углу страницы.
* Возможность удалять товары из корзины.

Дополнительные требования:

* Использовать минимальный набор библиотек. Разрешено использование только чистого JavaScript, CSS (без фреймворков).
* Структурировать проект: отделить HTML, CSS и JavaScript- файлы.

# РАЗРАБОТКА ПРОЕКТА

Проект состоит из 3 файлов:

* Index.html – главная страница
* Style.css – подключение стилей для главной страницы
* Script.js – функционал сайта.

## Index.html– главная страница

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="style.css">

    <script src="script.js" defer></script>

</head>

Подключение CSS и JavaScript.

<body>

    <div class="header">

        <div class="cont1">

            <p>Новосибирск</p>

            <div class="nav">

                <p>Компьютеры</p>

                <p>Телевизоры</p>

                <p>Смартфоны</p>

                <p>Прикладные устройства</p>

            </div>

            <div class="phone">8 800 399 56 98</div>

        </div>

    </div>

Заголовок с навигацией.

<div class="page">

        <div class="cont">

            <div class="pr\_cont">

                <p class="title">Список продуктов</p>

                <div id="products"></div>

            </div>

Место куда вставляются продукты из списка объектов в JS.

            <div class="bask\_cont">

                <p class="title">Корзина</p>

                <div class="total">

                    <!-- <p>Количество</p> -->

                    <p id="quantity"></p>

                    <!-- <p>Общая стоимость</p> -->

                    <p id="total-price"></p>

                </div>

                <div id="basket"></div>

            </div>

        </div>

    </div>

</body>

</html>

Корзина, с местом куда будут добавляться товары и окном в котором написано сколько добавлено продуктов и общая стоимость заказа.

## Style.css – подключение стилей для главной страницы

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

}

body

{

    font-family: Arial, sans-serif;

}

Общие стили для всех элементов.

.header

{

    background-color: rgb(28, 84, 3);

    color: white;

    padding: 10px;

    text-align: center;

}

.cont1

{

    display: flex;

    gap: 30px;

    justify-content: space-between;

    align-items: center;

    /\* background-color: aqua; \*/

    padding: 20px;

    border-radius: 10px;

    margin-bottom: 20px;

    max-width:  1500px;

    margin: 0 auto;

}

.nav

{

    display: flex;

    gap: 30px;

    align-items: center;

}

.phone

{

    font-weight: bold;

    /\* font-size: 30px; \*/

    background-color: white;

    color: green;

    padding: 10px;

    border-radius: 10px;

}

Заголовок с навигацией.

.cont {

    max-width: 1500px;

    margin: 0 auto;

    padding: 20px;

    display: grid;

    grid-template-columns: 1fr 1fr;

    gap: 20px;

    justify-content: space-between;

}

.pr\_cont

{

    padding-top: 20px;

}

.title

{

    margin-bottom: 20px;

    font-size: 30px;

    font-weight: bold;

}

.product {

    display: flex;

    gap: 30px;

    justify-content: space-between;

    background-color: rgb(146, 255, 195);

    padding: 20px;

    border-radius: 10px;

    margin-bottom: 20px;

    align-items: center;

}

.price

{

    font-weight: bold;

    font-size: 30px;

}

Список продуктов на странице.

.product\_basket

{

    display: flex;

    gap: 30px;

    justify-content: space-between;

    background-color: rgb(255, 255, 255);

    padding: 20px;

    border-radius: 10px;

    margin-bottom: 20px;

    align-items: center;

}

.total

{

    background-color: green;

    border-radius: 10px;

    padding: 20px;

    margin-bottom: 30px;

    color: white;

    font-size: 25px;

}

.bask\_cont

{

    padding: 20px;

    background-color: rgb(191, 255, 211);

    border-radius: 10px;

}

Корзина

button

{

    background-color: green;

    color: white;

    padding: 10px;

    border-radius: 10px;

    border: none;

    cursor: pointer;

    align-items: center;

    display: flex;

    height: fit-content;

}

Все кнопки на странице

## Script.js – функционал сайта.

Введение переменных для работы из HTML-элементов.

const total = document.getElementById("total-price");

const total1 = document.getElementById("quantity");

const list = document.getElementById('products');

const basket = document.getElementById("basket");

Список продуктов на странице

const list\_products = [

    {

        name: '43" (109 см) LED-телевизор DEXP A431 серый',

        description: 'Direct LED, 4K UltraHD, Wi-Fi, 60 Гц, Android TV, HDMI х 3, USB х 2 шт',

        price: 20599,

    },

    {

        name: '6.56" Смартфон Tecno SPARK GO 2024 64 ГБ золотистый',

        description: 'ядер - 8x(1.6 ГГц), 3 ГБ, 2 SIM, IPS, 1612x720, камера 13+0.8 Мп, 4G, GPS, FM, 5000 мА\*ч',

        price: 6999,

    },

    {

        name: '15.6" Ноутбук ASUS TUF Gaming F15 FX507ZC4-HN009 серый',

        description: '1920x1080, IPS, Intel Core i5-12500H, ядра: 4 + 8 х 2.5 ГГц + 1.8 ГГц, RAM 16 ГБ, SSD 512 ГБ, GeForce RTX 3050 для ноутбуков 4 ГБ, без ОС',

        price: 78999,

    },

    {

        name: 'ПК ARDOR GAMING NEO M143',

        description: 'Intel Core i5-12400F, 6 x 2.5 ГГц, 16 ГБ DDR4, GeForce RTX 3050, SSD 1000 ГБ, без ОС',

        price: 68799,

    },

    {

        name: '6.56" Смартфон Infinix SMART 8 64 ГБ зеленый',

        description: 'ядер - 8x(1.6 ГГц), 3 ГБ, 2 SIM, IPS, 1612x720, камера 13+0.2 Мп, 4G, GPS, FM, 5000 мА\*ч',

        price: 6999,

    },

    {

        name: 'Компактный фотоаппарат Sony ZV-1 черный',

        description: '20.1 Мп, фото: 5472x3648, видео Full HD, дисплей 3", Bluetooth, Wi-Fi',

        price: 69999,

    },

    {

        name: '11" Планшет Xiaomi Pad 6 Wi-Fi 256 ГБ черный',

        description: '2880x1800, IPS, 8x3.2 ГГц, 8 ГБ, 8840 мА\*ч, Android 13.x',

        price: 39999,

    },

    {

        name: 'Микрофон Fifine AM8 черный',

        description: 'проводной, настольный, подвесной, -50 дБ, от 50 Гц до 16000 Гц, jack 3.5 мм, USB Type-C, XLR',

        price: 6099,

    },

    {

        name: 'МФУ струйное Epson L3250',

        description: 'цветная печать, A4, 5760x1440 dpi, ч/б - 10 стр/мин (А4), USB, Wi-Fi, СНПЧ',

        price: 15000,

    },

    {

        name: 'Мышь проводная CBR CM-102 черный',

        description: '1200 dpi, USB Type-A, кнопки - 3',

        price: 99,

    },

    {

        name: '23.8" Монитор ARDOR GAMING PORTAL AF24H1 черный',

        description: '1920x1080@144 Гц, IPS, LED, 1000:1, 280 Кд/м², 178°/178°, DisplayPort 1.4 x2, HDMI 2.0 x2, AMD FreeSync Compatible, NVIDIA G-SYNC Compatible',

        price: 13999,

    },

];

Этот список продуктов представлен объектами чтобы его легче было удобно вывести на страницу. Он выводится на экран с помощью метода map().

list\_products.map(function(product) {

    const elem = document.createElement('div');

    elem.classList.add('product');

    const name = document.createElement('h3');

    name.textContent = product.name;

    elem.appendChild(name);

    const description = document.createElement('p');

    description.textContent = product.description;

    elem.appendChild(description);

    const price = document.createElement('p');

    price.textContent = product.price;

    price.classList.add('price');

    elem.appendChild(price);

    const button = document.createElement('button');

    button.innerHTML = 'Добавить <br>в <br>корзину';

    button.classList.add('button');

    button.addEventListener('click', function() {

        addToBasket(product);

    });

    elem.appendChild(button);

    list.appendChild(elem);

});

Здесь каждый продукт выводится в отдельном блоке и ему добавляется кнопка для добавления в корзину, нажатие на которое вызывает функцию, которая это делает.

function addToBasket(product)

{

    for (let i = 0; i < list\_basket.length; i++) {

        if (list\_basket[i].name === product.name) {

            list\_basket[i].count++;

            localStorage.setItem('products', JSON.stringify(list\_basket));

            location.reload();

            return;

        }

    }

    product.id = Date.now();

    product.count = 1;

    list\_basket.push(product);

    localStorage.setItem('products', JSON.stringify(list\_basket));

    console.log(list\_basket);

    location.reload();

    updateTotal();

    UpdateQuantity()

};

Эта функция добавляет информацию о продукте в список, если он добавлен первый раз, а далее просто увеличивается количество товара, повторно он не добавляется в корзину, список перезаписывается в localStorage, а затем обновляется количество и стоимость товаров в корзине. И каждому товару дается id по которому его можно удалить, нажав на нем соответствующую кнопку.

function deleteFromBasket(id) {

    for (let i = 0; i < list\_basket.length; i++) {

        if (list\_basket[i].id === id) {

            list\_basket.splice(i, 1);

            localStorage.setItem('products', JSON.stringify(list\_basket));

            location.reload();

            return;

        }

    }

    updateTotal();

    UpdateQuantity()

}

При удалении происходит поиск товара по id и удаление товара из списка на позиции его индекса. Список также перезаписывается в localStorage, обновляется страница, чтобы на экране действие отобразилось и обновляются данные заказа.

document.addEventListener('DOMContentLoaded', loadProducts);

Когда загружается страница, вызывается функция, которая загружает список продуктов в корзине.

function loadProducts()

{

    const storedProducts = localStorage.getItem('products');

    if (storedProducts) {

        list\_basket = JSON.parse(storedProducts);

        if (!Array.isArray(list\_basket)) {

            list\_basket = [];

        }

    } else {

        list\_basket = [];

    }

    update();

}

Эта функция получает список продуктов которые находятся в корзине и сохраняет в рабочий список, из которого данные выводятся на экран. Это делает функция update().

function update()

{

    list\_basket.map(function(product) {

        const basket\_el = document.createElement('div');

        const pr\_name = document.createElement('p');

        pr\_name.innerHTML = product.name;

        basket\_el.appendChild(pr\_name);

        const pr\_des = document.createElement('p');

        pr\_des.innerHTML = product.description;

        basket\_el.appendChild(pr\_des);

        const pr\_price = document.createElement('p');

        pr\_price.innerHTML = product.price;

        basket\_el.appendChild(pr\_price);

        const pr\_count = document.createElement('p');

        pr\_count.innerHTML = product.count;

        basket\_el.appendChild(pr\_count);

        const deleteButton = document.createElement('button');

        deleteButton.innerHTML = 'Удалить';

        deleteButton.addEventListener('click', function() {

            deleteFromBasket(product.id);

        });

        basket\_el.appendChild(deleteButton);

        basket\_el.classList.add('product\_basket');

        basket\_el.id = product.id;

        basket.appendChild(basket\_el);

        console.log(list\_basket);

    });

        updateTotal();

        UpdateQuantity()

}

Эта функция перебирает полученный список с продуктами и каждый продукт оформляет на экран, чтобы было удобно читать и добавляет кнопку для удаления всего количества товара. Затем обновляется информация о количестве товаров в корзине и общей стоимости заказа.

function updateTotal()

{

    let sum = 0;

    for (let i = 0; i < list\_basket.length; i++) {

        sum += list\_basket[i].price \* list\_basket[i].count;

    }

    total.innerHTML = "Общая стоимость: " + sum + " руб.";

}

Эта функция проходит по всему списку товаров, считает стоимость каждого и результаты суммирует, а потом результат выводит в специальном поле в верхнем правом углу.

function UpdateQuantity()

{

    let sum = 0;

    for (let i = 0; i < list\_basket.length; i++) {

        sum += list\_basket[i].count;

    }

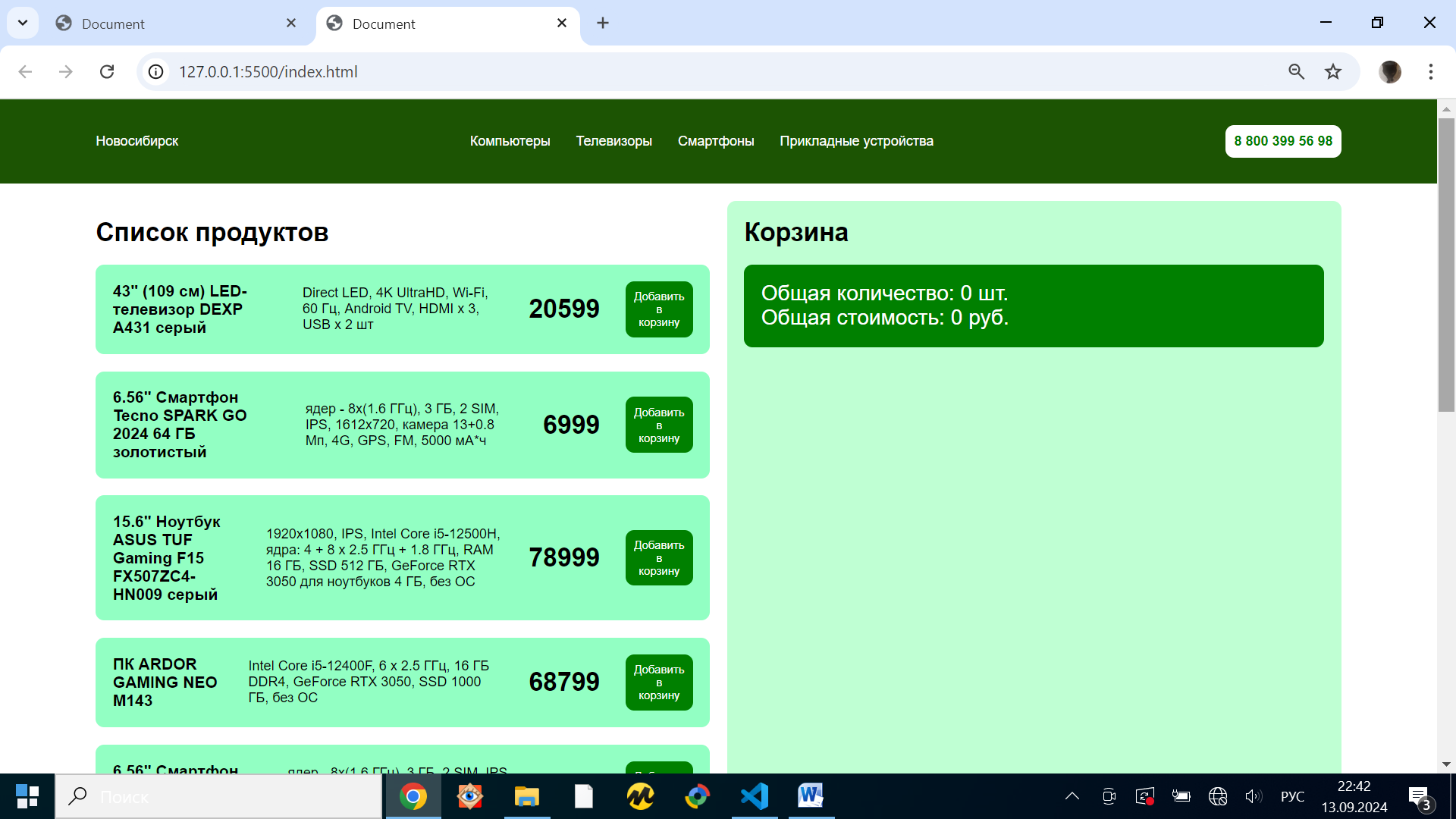
    total1.innerHTML = "Общая количество: " + sum + " шт.";

}

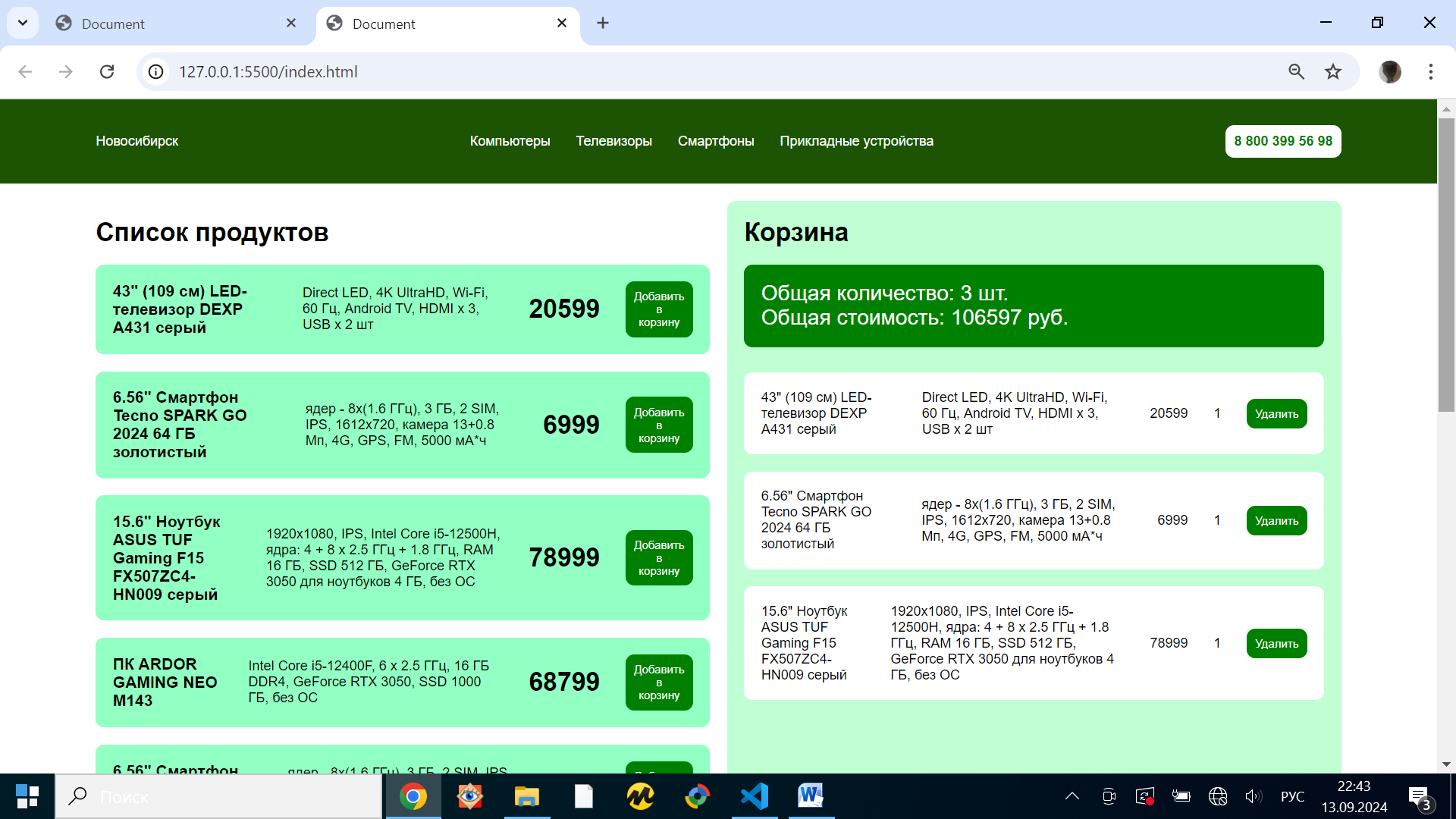
Эта функция аналогично считает количество товаров и выводит рядом со стоимостью.

# ДЕМОНСТРАЦИЯ РАБОТЫ САЙТА

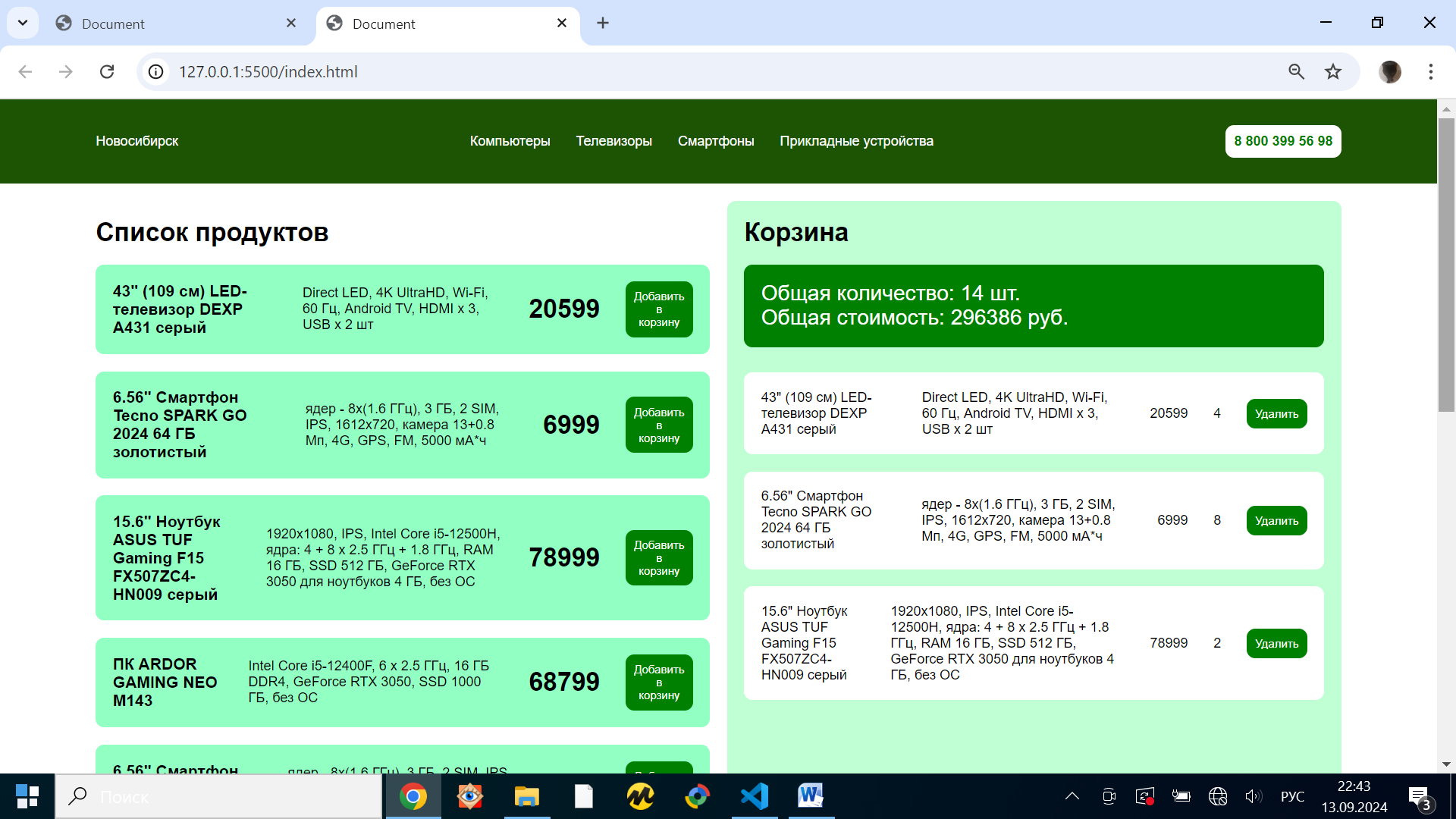
Вначале видно список продуктов и пустую корзину.



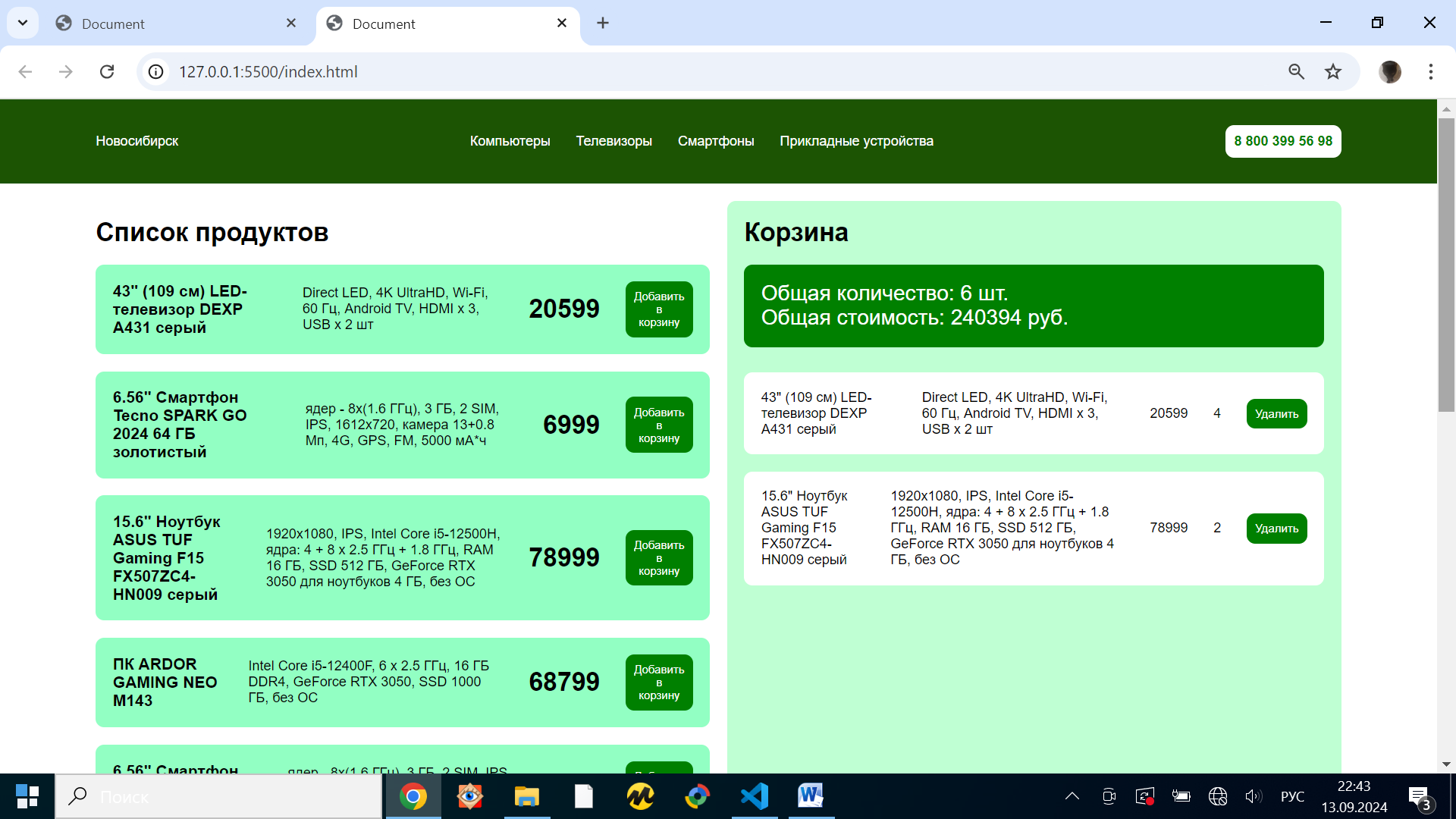
Далее были нажаты кнопки у 3 верхних товаров по 1 разу.



Потом были нажаты эти же кнопки повторно, видно что товар больше не добавился а только увеличилось его количество.



Затем была нажата кнопка удаления у второго товара в корзине. Видно что удалился нужный товар вместе с его количеством.



# ВЫВОД

В приложении выполняются все перечисленные в требованиях функции, оно удобно воспринимается, с ним удобно работать, данные все сохраняются, даже если закрыть вкладку. Функции работают без ошибок и задержек.