**МИНИСТЕРСТВО ОБРАЗОВАНИЯ НОВОСИБИРСКОЙ ОБЛАСТИ**

Государственное автономное профессиональное образовательное учреждение Новосибирской области

«Новосибирский колледж печати и информационных технологий»



ОТЧЁТ

по лабораторной работе №2

«Расчет количества калорий»

по дисциплине: «Разработка кода информационных систем»

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

Новосибирск 2024 г.

**Цель:** Создать веб-приложение, которое позволяет пользователям отслеживать потребление калорий в течение дня. Приложение должно включать функционал для добавления продуктов питания и их калорийности, расчета общей калорийности за день, а также сохранения данных в локальном хранилище браузера.

**Описание:**

**Разработка фронтенда:**

**Создать страницу, где пользователь может:**

* Добавлять продукт питания с указанием его названия и калорийности.
* Просматривать список добавленных продуктов с их калорийностью.
* Удалять продукты из списка.
* Видеть общую калорийность потребленных продуктов за день.

**Создание интерфейса:**

* HTML: Разметка страницы с полями для ввода названия продукта и калорийности, а также областью для отображения списка продуктов и общей калорийности.
* CSS: Стилизация элементов страницы, чтобы интерфейс был удобным и приятным для пользователя.
* JavaScript: Реализация функционала для добавления, удаления продуктов и вычисления общей калорийности.

**Сохранение данных:**

* Использовать localStorage для сохранения данных о продуктах на стороне клиента.
* При загрузке страницы загружать продукты из localStorage и отображать их на странице.

**Требования:**

**Минимальные требования:**

* Страница для ввода данных о продукте (название и калорийность).
* Возможность добавления продуктов в список.
* Отображение списка добавленных продуктов.
* Удаление продуктов из списка.
* Расчет и отображение общей калорийности.

**Дополнительные требования:**

* Возможность редактирования данных о продукте.
* Фильтрация продуктов по имени или калорийности.
* Сохранение данных между сеансами браузера с использованием localStorage.

**Выполнение работы**

* 1. **Главная страница index**

Подключения стилей

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Список продуктов</title>

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

</head>

Форма ввода продукта

<body>

    <div class="page">

        <h1>Список продуктов</h1>

        <form id="product-form">

            <input type="text" id="product-name" placeholder="Название продукта" required>

            <input type="number" id="calories" placeholder="Калорийность" required>

            <button type="submit">Добавить продукт</button>

        </form>

Сюда будут добавляться продукты и счет общей суммы

        <ul id="product-list"></ul>

        <p id="total-calories">Общая калорийность: <span id="total-calories-value">0</span> ккал</p>

Подключаем скрипт где есть функции для работы

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

    </div>

</body>

</html>

* 1. **Стили style**

Один шрифт на весь документ

body {

    font-family: Arial, sans-serif;

}

Заголовок страницы

h1 {

    text-align: center;

    color: rgb(28, 84, 3);

}

Форма ввода

form {

    display: flex;

    flex-wrap: wrap;

    margin-bottom: 20px;

    gap: 20px;

}

Поля ввода

input[type="text"],

input[type="number"] {

    flex: 1;

    margin-right: 10px;

    padding: 10px;

    border:1px solid rgb(28, 84, 3) ;

    border-radius: 10px;

    color: rgb(28, 84, 3);

}

Список с продуктами

ul {

    list-style-type: none;

    padding: 0;

}

li {

    display: flex;

    flex-wrap: wrap;

    justify-content: space-between;

    margin-bottom: 10px;

    padding: 10px;

    /\* border: 1px solid #ccc; \*/

    background-color: rgb(197, 255, 198);

    align-items: center;

    border-radius: 10px;

    color: rgb(28, 84, 3);

    gap: 20px;

    text-wrap: wrap;

    word-break: break-all;

    word-wrap: break-word;

}

Все кнопки

button {

    padding: 5px 10px;

    background-color: rgb(24, 143, 0);

    border-radius: 10px;

    cursor: pointer;

    border: none;

    color: white;

}

Вся страница

.page {

    max-width:1000px;

    margin: 0 auto;

    padding-inline: 20px;

}

Надпись «общая калорийность»

#total-calories {

    font-size: 20px;

    font-weight: bold;

    color:rgb(28, 84, 3);

}

* 1. **Скрипт script**

Получение элементов из html

document.addEventListener('DOMContentLoaded', loadProducts);

const form = document.getElementById('product-form');

const productList = document.getElementById('product-list');

const totalCalories = document.getElementById('total-calories-value');

Добавление продукта по кнопке

form.addEventListener('submit', addProduct);

Добавление продукта (вызов функции)

function addProduct(event) {

    event.preventDefault();

    const productName = document.getElementById('product-name').value;

    const calories = parseInt(document.getElementById('calories').value);

    if (productName && calories) {

        const product = {

            id: Date.now(),

            name: productName,

            calories: calories

        };

        addProductToList(product);

        saveProduct(product);

        updateTotalCalories();

        document.getElementById('product-name').value = '';

        document.getElementById('calories').value = '';

    }

}

Добавление продукта в сипсок, создание элемента в html, вывод его на экран, добавление кнопки «редактировать» и «удалить»

function addProductToList(product) {

    const listItem = document.createElement('li');

    listItem.setAttribute('data-id', product.id); // Добавьте атрибут data-id

    listItem.textContent = product.name + ' (' + product.calories + ' ккал)';

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

    editButton.textContent = 'Редактировать';

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

        const updatedName = prompt('Введите новое название продукта:');

        const updatedCalories = parseInt(prompt('Введите новое количество калорий:'));

        if (updatedName && !isNaN(updatedCalories)) {

            editProduct(product.id, { name: updatedName, calories: updatedCalories });

        }

    });

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

    deleteButton.textContent = 'Удалить';

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

        deleteProduct(product.id);

        productList.removeChild(listItem);

        updateTotalCalories();

    });

    listItem.appendChild(editButton);

    listItem.appendChild(deleteButton);

    productList.appendChild(listItem);

}

Сохранение продукта в локальном хранилище

function saveProduct(product) {

    const products = getProducts();

    products.push(product);

    localStorage.setItem('products', JSON.stringify(products));

}

Удаление продукта

function deleteProduct(id) {

    const products = getProducts();

    const updatedProducts = products.filter(function(product) {

        return product.id !== id;

    });

    localStorage.setItem('products', JSON.stringify(updatedProducts));

    updateTotalCalories();

}

Обновление продуктов

function loadProducts() {

    const products = getProducts();

    products.forEach(function(product) {

        addProductToList(product);

    });

    updateTotalCalories();

}

Получение продуктов из хранилища

function getProducts() {

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

    return products ? JSON.parse(products) : [];

}

Обновление общей колорийности

function updateTotalCalories() {

    const products = getProducts();

    let totalCalories = 0;

    products.forEach(function(product) {

        totalCalories += product.calories;

    });

    const totalCaloriesElement = document.getElementById('total-calories-value');

    totalCaloriesElement.textContent = totalCalories;

}

Редактирование, так чтобы оно обновлялось и кнопки не исчезали

function editProduct(id, updatedProduct) {

    const products = getProducts();

    const updatedProducts = products.map(product => {

        if (product.id === id) {

            return { ...product, ...updatedProduct };

        }

        return product;

    });

    localStorage.setItem('products', JSON.stringify(updatedProducts));

    // Найдите элемент списка продукта и обновите его содержимое

    const listItem = document.querySelector(`li[data-id="${id}"]`);

    if (listItem) {

        listItem.textContent = updatedProduct.name + ' (' + updatedProduct.calories + ' ккал)';

        // Добавьте кнопки "Редактировать" и "Удалить" заново

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

        editButton.textContent = 'Редактировать';

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

            const updatedName = prompt('Введите новое название продукта:');

            const updatedCalories = parseInt(prompt('Введите новое количество калорий:'));

            if (updatedName && !isNaN(updatedCalories)) {

                editProduct(id, { name: updatedName, calories: updatedCalories });

            }

        });

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

        deleteButton.textContent = 'Удалить';

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

            deleteProduct(id);

            productList.removeChild(listItem);

            updateTotalCalories();

        });

        listItem.appendChild(editButton);

        listItem.appendChild(deleteButton);

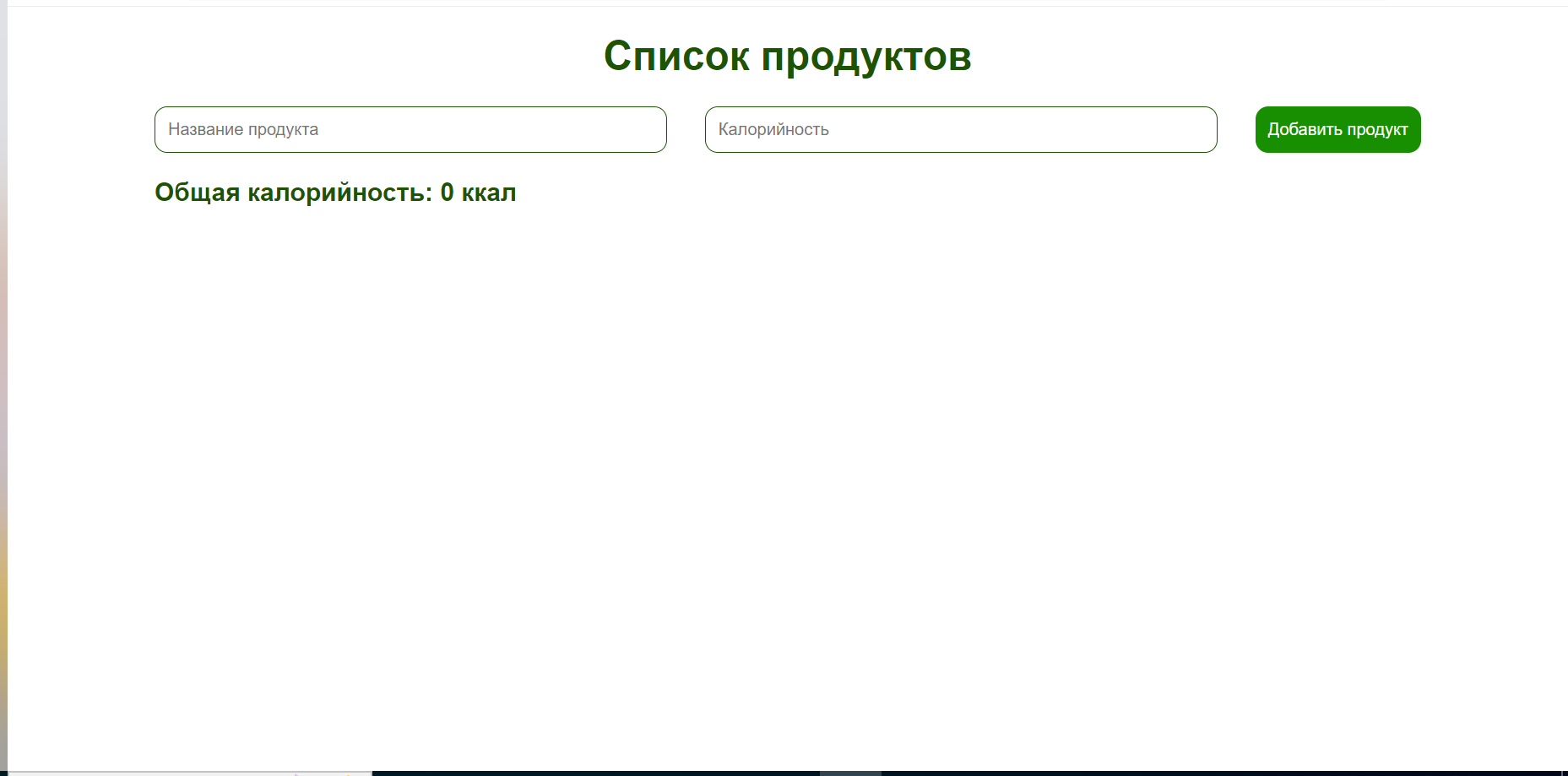
    }

    updateTotalCalories();

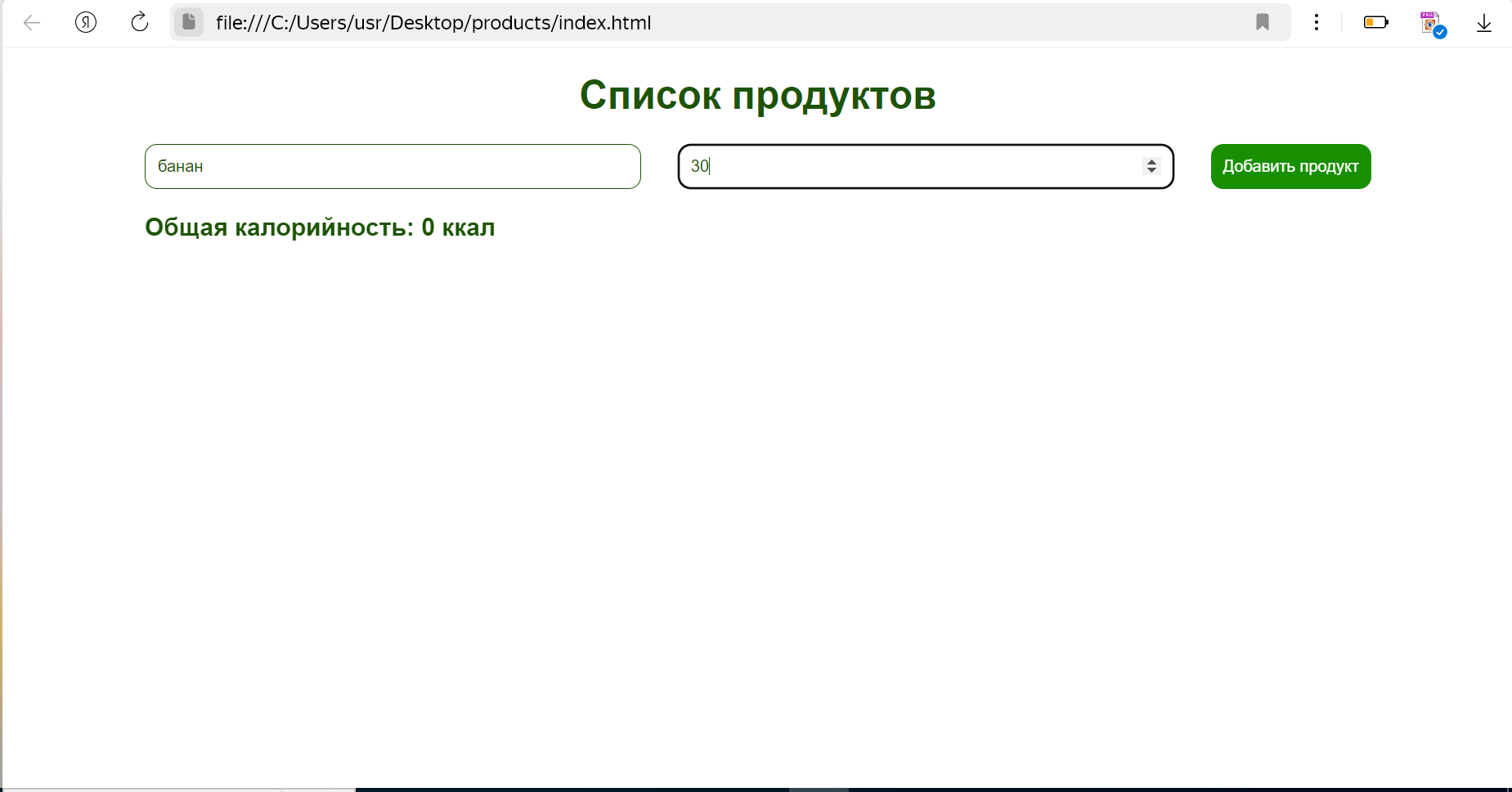
}

**Тестирование и показ приложения**

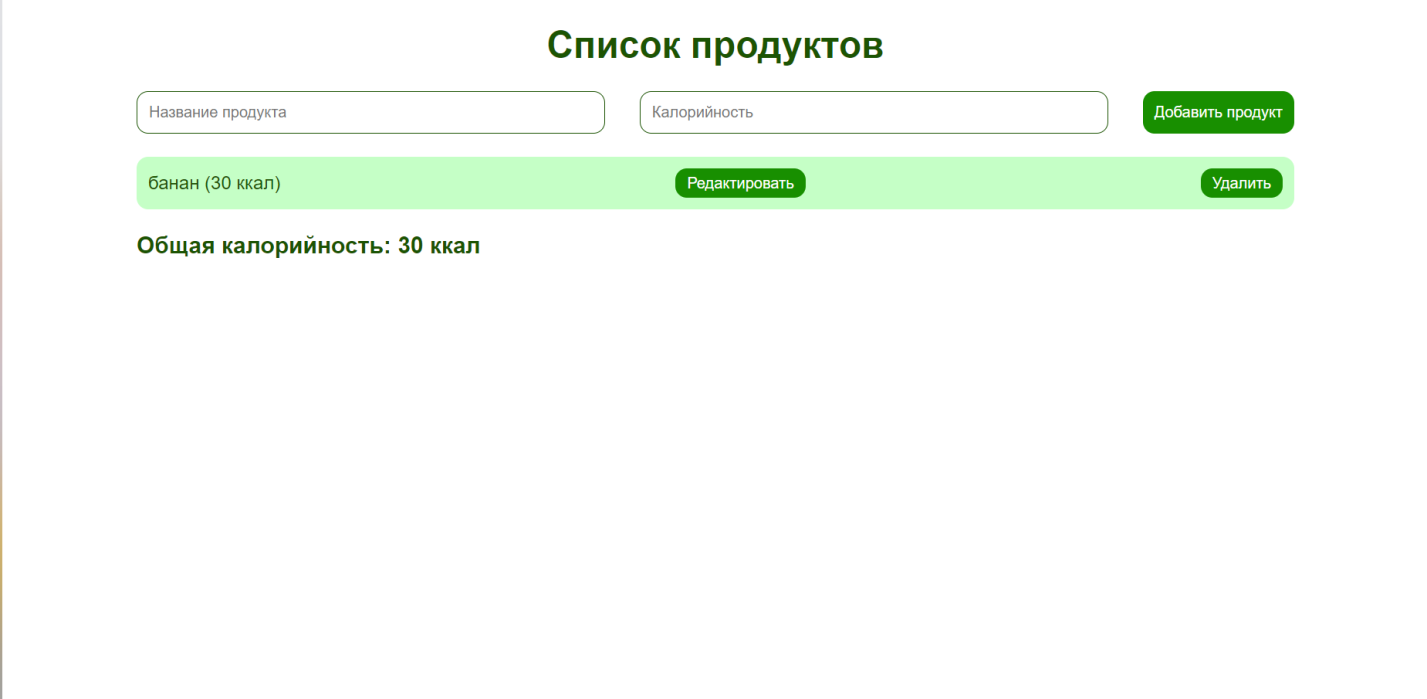
Сначала в списке нет продуктов



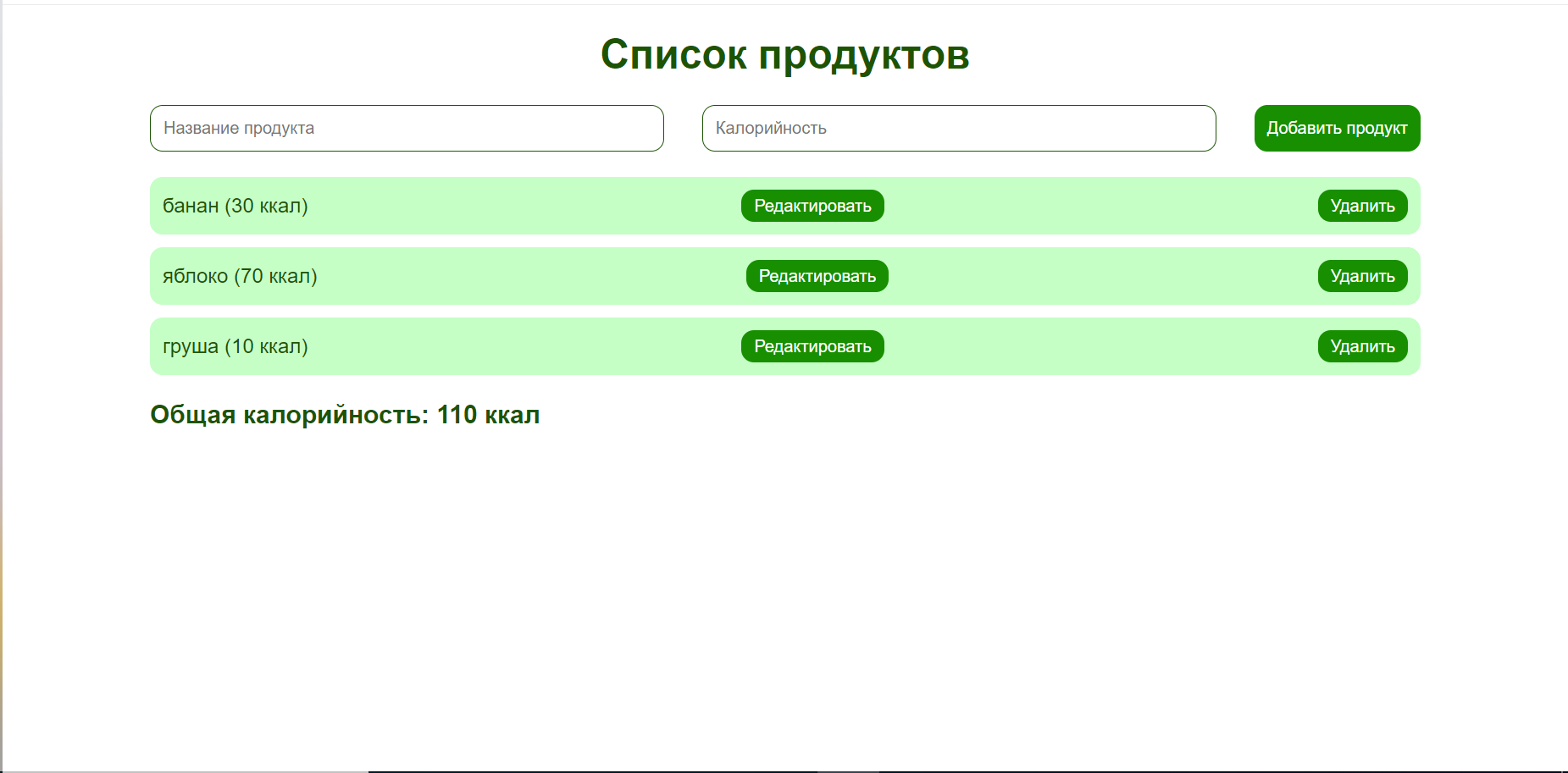
Вводим данные о продукте и нажимам «добавить продукт»



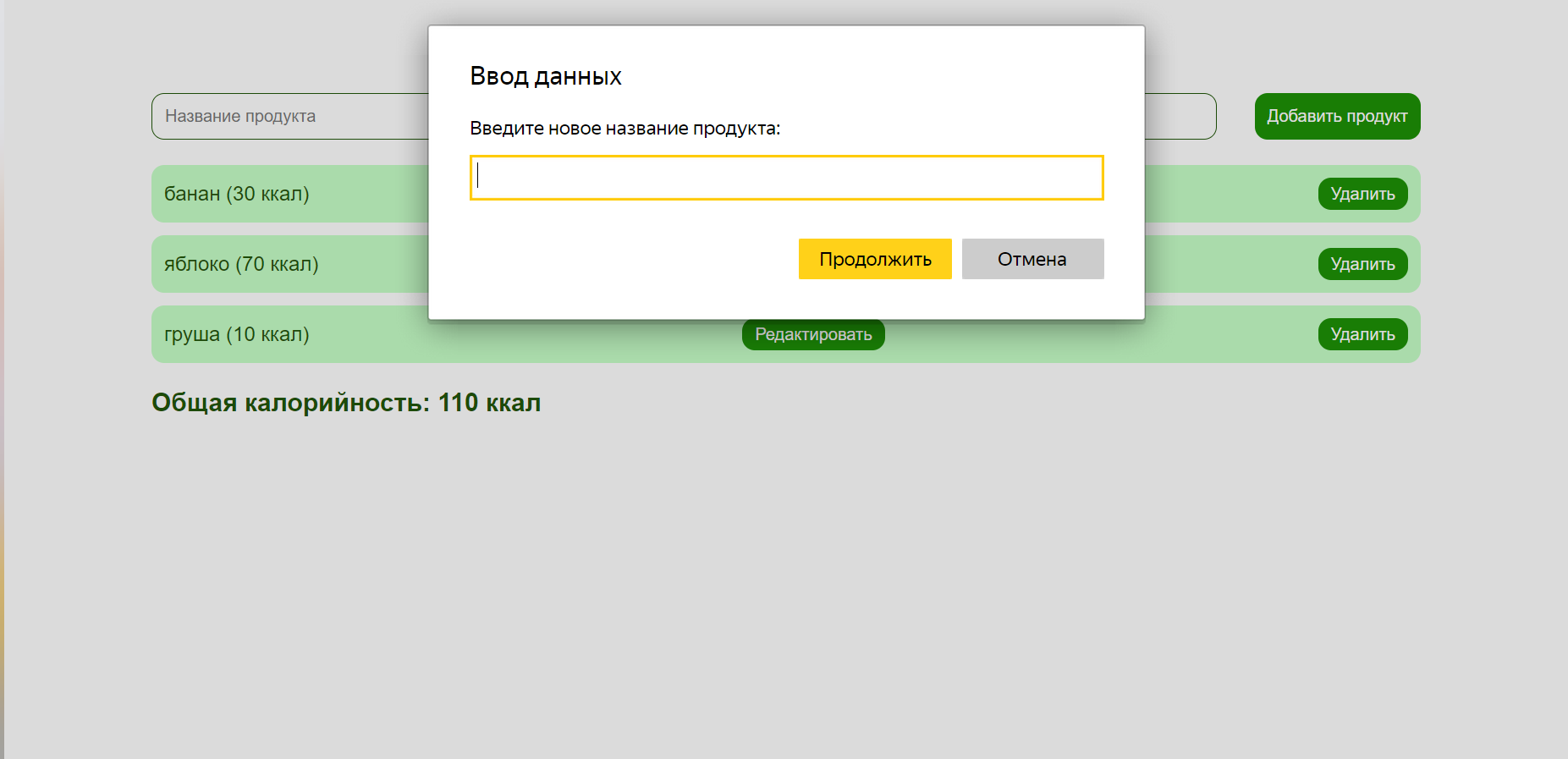
Добавление продукта, добавляем еще



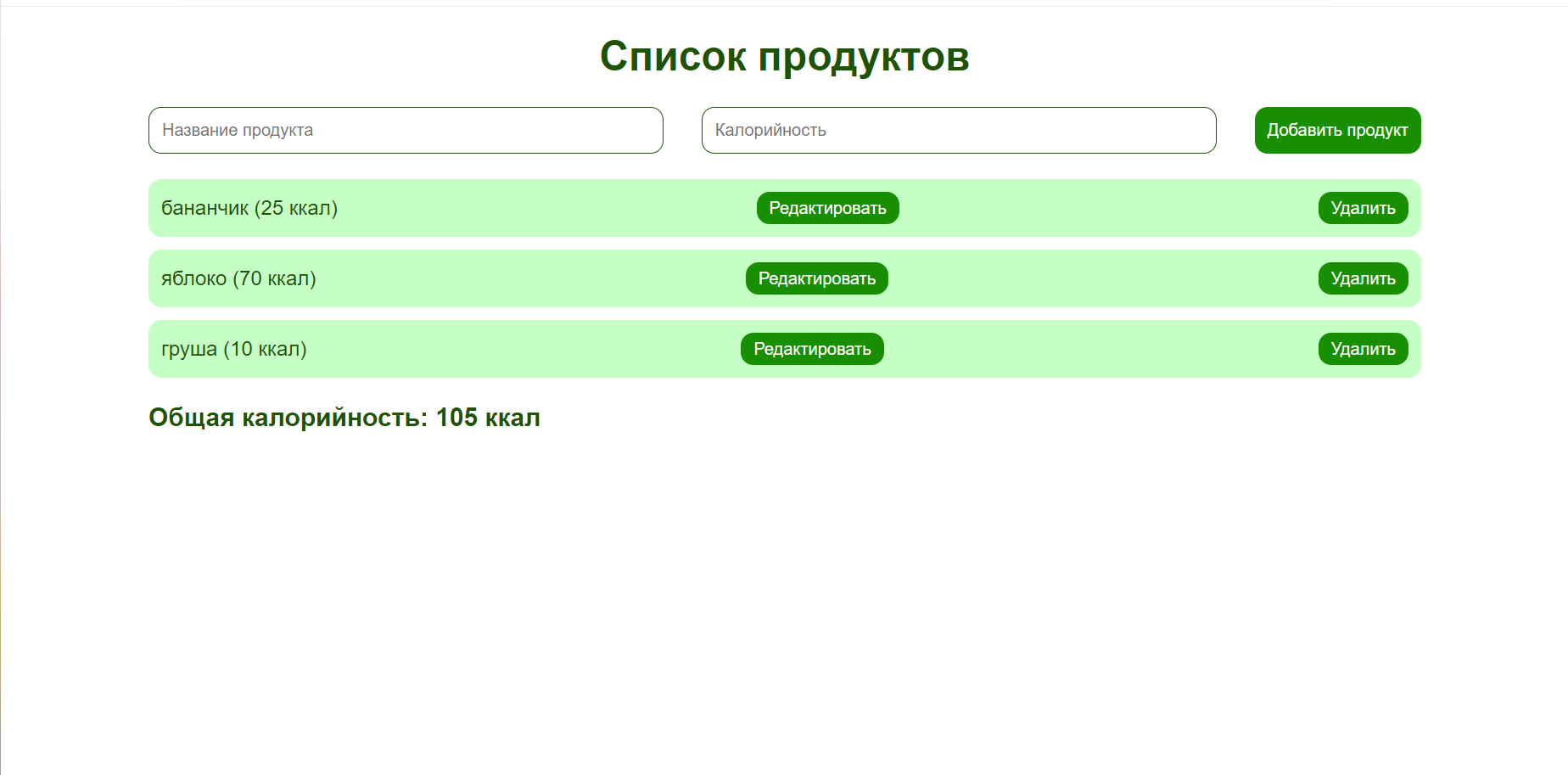
Добавлено 3 разных продукта, редактируем 1 продукт



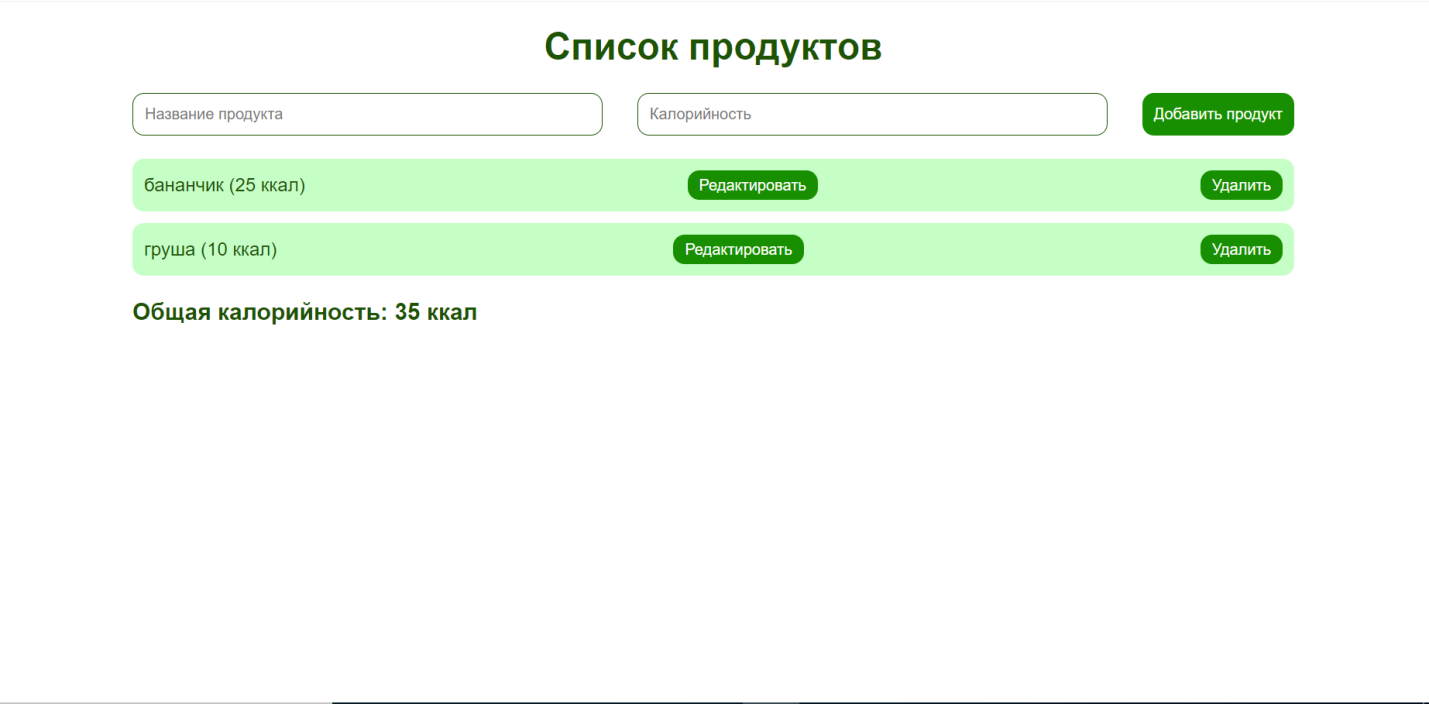
Редактирование продукта



Продукт отредактирован, удаляем продукт



Продукт удален



Вывод

Приложения выполняет требуемые функции и работает без ошибок