**Национальный исследовательский университет «МЭИ»**

**ИНЖЕНЕРНО-ЭКОНОМИЧЕСКИЙ ИНСТИТУТ**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра Безопасности и информационных технологий

**Отчет по дисциплине «Стандартизация, сертификация и управление качеством программного обеспечения».**

**Задание №27.**

Студент: Козин Станислав Вячеславович

Группа: ИЭоз-60-18

Подпись:

Руководитель:

Раскатова Марина Викторовна

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2022 г.

**Содержание**

[Глава 1. Постановка задачи 3](#_Toc104254170)

[1.1. Описание задачи 3](#_Toc104254171)

[1.2. Описание предметной области 3](#_Toc104254172)

[1.3. Функции программы 3](#_Toc104254173)

[1.4. Входные, выходные данные и ограничения 4](#_Toc104254174)

[1.5. Вид приложения, среда разработки 5](#_Toc104254175)

[1.6. Тестовые данные 5](#_Toc104254176)

[Глава 2. Разработка приложения 6](#_Toc104254177)

[2.1. Разработка структуры программы 6](#_Toc104254178)

[2.2. Разработка схемы алгоритма 7](#_Toc104254179)

[2.3. Разработка пользовательского интерфейса 8](#_Toc104254180)

[Глава 3. Реализация и тестирование приложения 12](#_Toc104254181)

[3.1. Описание разработанной программы 12](#_Toc104254182)

[3.2. Тестирование программы 13](#_Toc104254183)

[Вывод.... 20](#_Toc104254184)

[Список используемых источников 21](#_Toc104254185)

# Постановка задачи

## Описание задачи

Задача №27. Программа содержит список товаров на складе. Список имеет следующую структуру: наименование товара, количество товара в штуках, стоимость единицы товара. Программа должна заполнить список и вывести его на экран в виде таблицы, а затем при вводе наименования товара и требуемого количества товара вывести на экран сообщение о стоимости заказываемого товара и уменьшить общее количество данного товара на складе. Отсортируйте список по полю «наименование товара».

## Описание предметной области

На складе хранятся легковые автомобили. Вместимость склада — 256 штук. Описание товара на складе содержит наименование, цену и количество. Наименования уникальны, цены и количество не могут быть меньше нуля. Количество наименований в списке товаров может превышать 256.

В заказе содержится его номер, дата и список товаров (наименование, кол-во и цена). Номер заказа уникален и не может быть меньше 1. Для того, чтобы товар мог быть добавлен в заказ, его количество в нём должно быть меньше или равно количеству на складе. История заказов сохраняется. Заказ нельзя изменить, только удалить старый и создать новый. При удалении, количество товара на склад не возвращается.

## Функции программы

* 1. Ввод списка товаров и заказов в графическом интерфейсе, либо из файла;
  2. Вывод списка товаров в виде таблицы;
  3. Сортировка списка товаров по любому параметру;
  4. Заказ товара: пользователь выбирает наименование и кол-во, программа показывает стоимость и вычитает кол-во со склада.
  5. Сохранение списка товаров и заказов в файле.

## Входные, выходные данные и ограничения

Данные вводятся через графический интерфейс или из файла.

Таблица . — Входные данные

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Смысл** |
| name | string | Наименование товара |
| quantity | integer | Кол-во |
| price | float | Цены на товар |

Данные выводятся на экран и сохраняются в файл.

Таблица . — Выходные данные

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Смысл** |
| name | string | Наименование |
| quantity | integer | Кол-во |
| price | float | Цена |
| orderNumber | integer | Номер заказа |
| date | Date | Дата заказа |
| sum | float | Сумма заказа |

Предусмотрена обработка следующих ошибок.

Ошибки товаров:

1. Отрицательная цена;
2. Отрицательное количество;
3. Не указано наименование товара.

Ошибки заказов:

1. Количество товара в заказе меньше 1;
2. Номер заказа меньше 1;
3. Общее кол-во товаров в заказе больше 256;
4. Необходимого количества товаров нет на складе.

Ошибки склада:

1. Количество товаров больше 256;
2. Повторяющиеся наименования товаров;
3. Повторяющиеся номера заказов.

## Вид приложения, среда разработки

Веб-приложение.

Язык программирования: TypeScript.

Среда разработки: Visual Studio Code.

## Тестовые данные

Таблица . — Каталог товаров

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Стоимость, руб.** | **Кол-во** |
| Volkswagen Polo V | 850 000 | 6 |
| Kia Venga I | 920 000 | 5 |
| Volkswagen Caddy Maxi IV | 2 051 760 | 3 |
| Skoda Rapid I | 869 000 | 15 |
| Mitsubishi Outlander III | 1 195 000 | 9 |
| LADA (ВАЗ) Kalina I | 189 000 | 20 |

Таблица . — Заказ от 01.05.2022 9:32

|  |  |
| --- | --- |
| **Наименование** | **Кол-во** |
| Volkswagen Polo V | 1 |
| Kia Venga I | 1 |

Таблица . — Заказ от 01.05.2022 13:12

|  |  |
| --- | --- |
| **Наименование** | **Кол-во** |
| LADA (ВАЗ) Kalina I | 1 |

# Разработка приложения

## Разработка структуры программы

Для разработки используется декларативный подход. Программа будет разбита на компоненты, все данные будут находятся в Redux-хранилище [4].

На рисунке 2.1 изображена схема типов приложения:

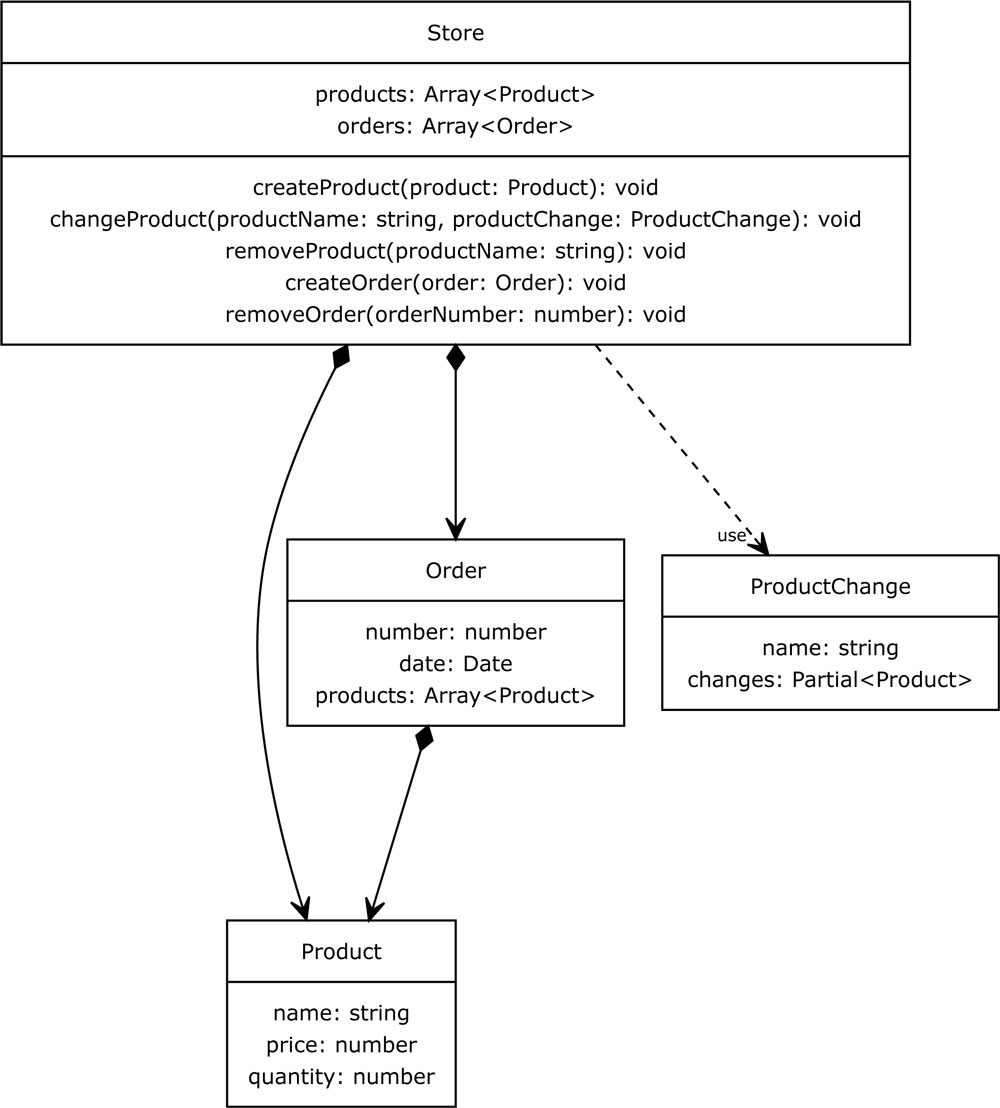


Рисунок . — схема типов приложения

Тип Product описывает товар, он содержит следующие поля:

1. name: string (наименование),
2. price: number (стоимость),
3. quantity: number (количество).

Тип ProductChange описывает объект изменения товара, он содержит следующие поля:

1. name: string (наименование изменяемого товара),
2. changes: Partial<Product> (объект, содержащий поля из типа Product, которые нужно изменить).

Тип Order описывает заказ, он содержит следующие поля:

1. number: number (номер заказа),
2. date: Date (дата и время заказа),
3. products (массив товаров).

Сумма заказа будет вычисляться во время выполнения программы.

В Store хранятся массивы товаров и заказов, а также функции для работы с ними:

1. createProduct(product: Product) создаёт товар,
2. changeProduct(productName: string, changes: Partial<Product>) изменяет товар,
3. removeProduct(productName: string) удаляет товар,
4. createOrder(order: Order) создаёт заказ,
5. removeOrder(orderNumber: number) удаляет заказ.

## Разработка схемы алгоритма

Из-за того, что был выбран декларативный подход, представить работу всей программы в виде блок-схемы сложно. Поэтому, будет приведена схема работы основных функций.

На рисунке 2.2 изображена схема работы функций создания и изменения товара. Создание заказа происходит аналогично.

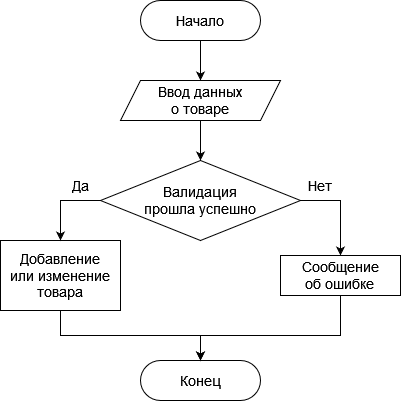


Рисунок 2.2 — алгоритм функций createProduct и changeProduct

## Разработка пользовательского интерфейса

Программа предназначена для администратора склада. На рисунке 2.3 изображена схема действий, доступных пользователю.



Рисунок . — диаграмма вариантов использования

В соответствии с функциями пользователя разработаны формы приложения.

На рисунке 2.4 изображена страница работы со складом:

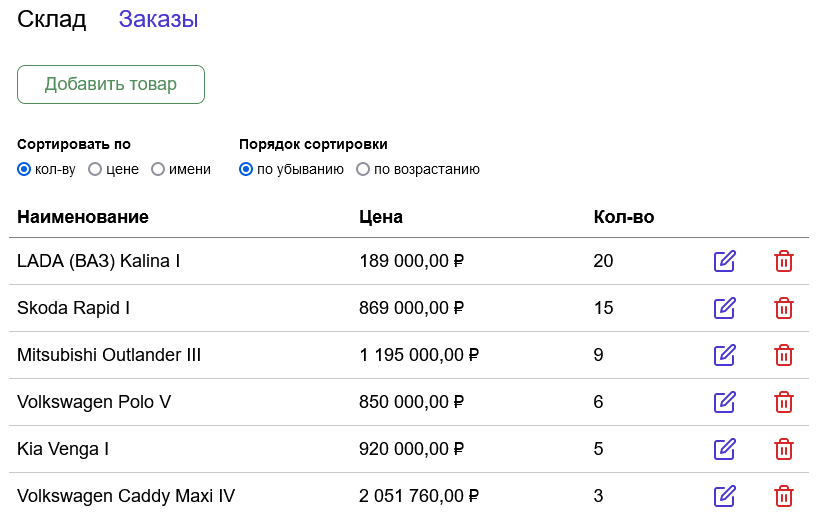


Рисунок . — страница работы со складом

На этой странице пользователь может добавлять товары на склад, а также изменять и удалять их. Также доступна сортировка товаров по свойствам, перечисленным на рисунке 2.4.

Для добавления товара пользователь нажимает кнопку «Добавить товар», после откроется форма, изображенная на рисунке 2.5:

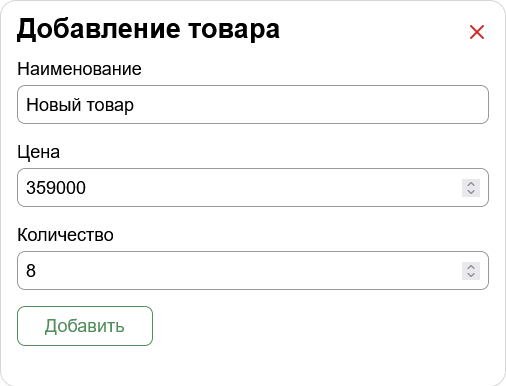


Рисунок . — форма добавления товара

Для редактирования товара пользователь нажимает на синюю кнопку в строке с товаром, после откроется форма, изображенная на рисунке 2.6:



Рисунок . — форма редактирования товара

Удаление производится нажатием на красную кнопку в строке с товаром.

На рисунке 2.7 изображена страница работы с заказами:

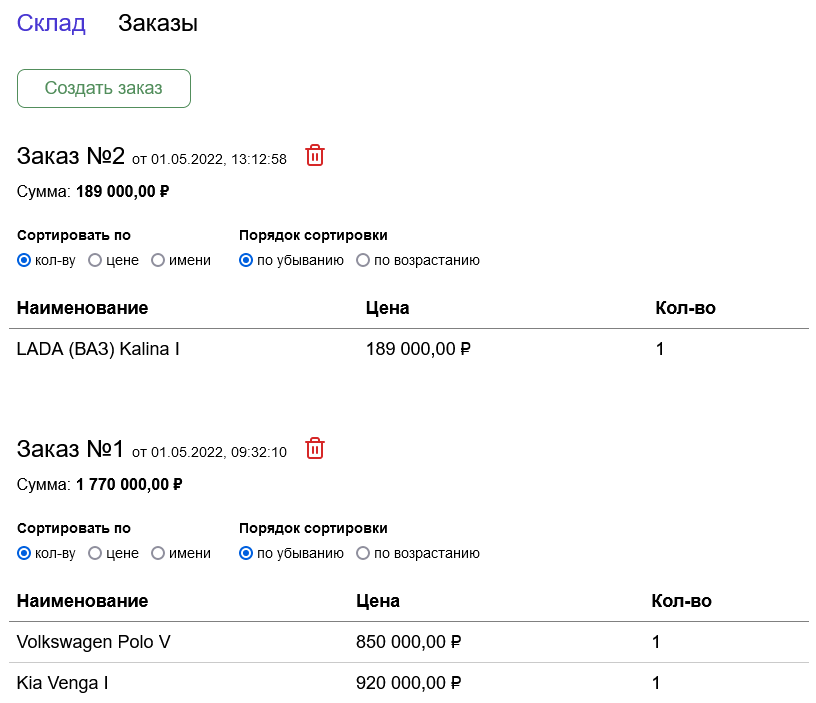


Рисунок . — страница работы с заказами

На этой странице пользователь может создавать и удалять заказы.

Удаление заказа производится нажатием на красную кнопку рядом с названием заказа.

Для добавления товара пользователь нажимает кнопку «Добавить товар», после откроется форма, изображенная на рисунке 2.8.

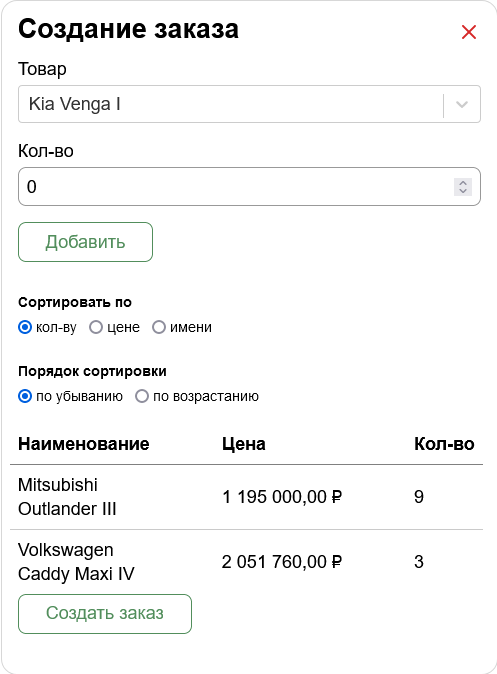


Рисунок . — форма создания заказа

# Реализация и тестирование приложения

## Описание разработанной программы

Программа разработана на языке TypeScript [1–2] с использованием библиотеки React [3]. Для управления состоянием приложения используется Redux с Redux Toolkit [4].

Для работы с состоянием приложения используется Redux-хранилище.

В файле /src/store/productsSlice.ts находится описание хранилища товаров. В нём описаны необходимые типы, а также функции для добавления, изменения и удаления товаров.

В файле /src/store/ordersSlice.ts находится описание хранилища заказов. В нём описаны необходимые типы, а также функции для добавления и удаления заказов.

Интерфейс программы разбит на компоненты.

Компоненты ProductsPage и OrdersPage отрисовывают страницы для работы со складом и заказами. Переключение между страницами реализовано с помощью библиотеки React Router.

На страницах используются различные компоненты пользовательського интерфейса.

Компонент Header отвечает за переключение между страницами.

Компонент ProductsTable отвечает за вывод таблицы с товарами. В таблице предусмотрена сортировка по каждому свойству товара с выбором порядка сортировки. Сортировка реализована с помощью стандартного метода массивов sort. В таблице реализовано удаление и изменение товаров.

Также разработаны компоненты пользовательского ввода: кнопки и текстовое поле.

Компонент Input представляет собой поле для ввода текста или числа.

Компонент Radio реализует переключатель, который позволяет пользователю выбрать одну опцию из предопределённого набора.

Компоненты Button и IconButton представляют собой кнопки.

Листинг разработанной программы приведён в приложении.

## Тестирование программы

Тестирование программы проводилось по ГОСТ 19.301-79.

Объект испытаний — веб-приложение для работы со складом.

Цель испытаний — проверка работоспособности программы.

Средства испытаний — программа тестировалась на персональном компьютере, имеющем операционную систему Windows 10 в браузере Mozilla Firefox.

Порядок испытаний — сначала проводились испытания работы форм создания и изменения товаров, затем проверялась работа формы создания заказов, после чего проводились испытания всей программы целиком, используя готовый пользовательский интерфейс.

Методы испытаний — программа тестировалась по методу входа-выхода в нормальных, граничных и исключительных условиях для различных наборов тестовых данных. Исключительные ситуации описаны в разделе 1.4. «Входные, выходные данные и ограничения».

Результаты тестирования программы приведены на рисунках 3.1–3.13.

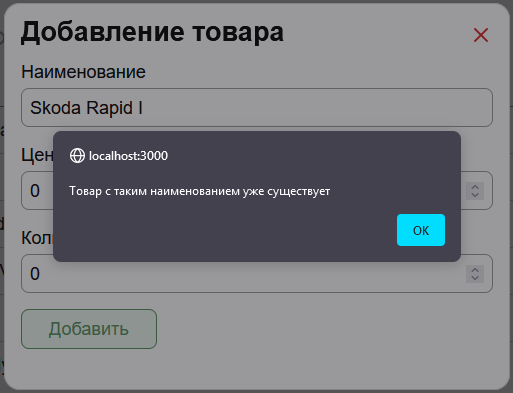


Рисунок . — Добавление уже существующего товара

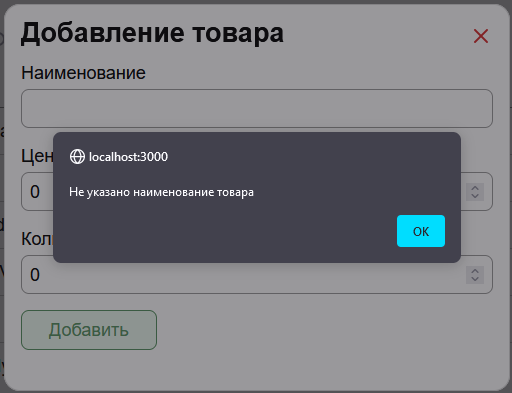


Рисунок . — Добавление товара с пустым наименованием

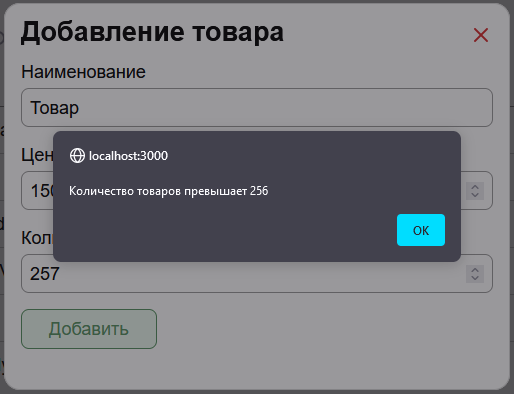


Рисунок . — Кол-во товаров превышает максимум

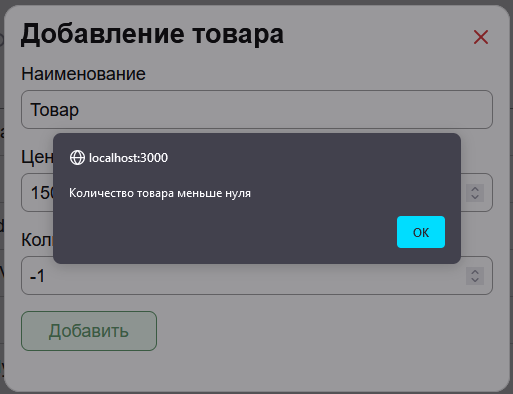


Рисунок . — Отрицательное кол-во товаров

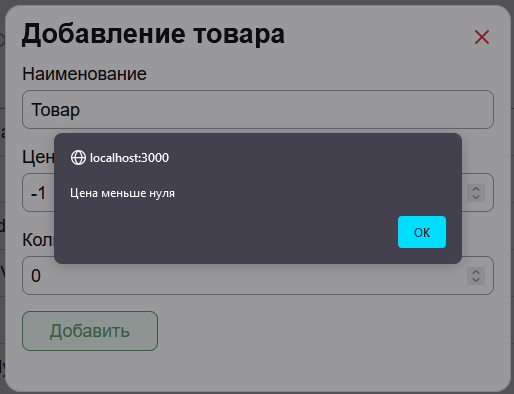


Рисунок . — Отрицательная цена

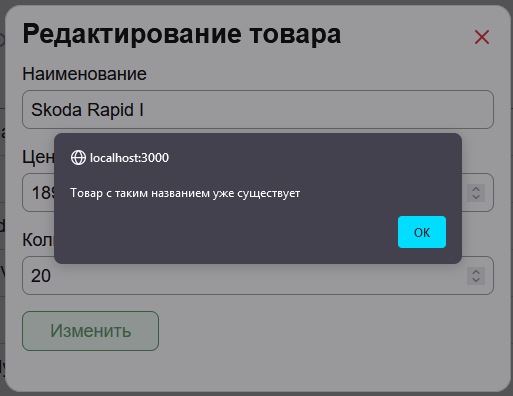


Рисунок . — Изменение наименования на уже существующее

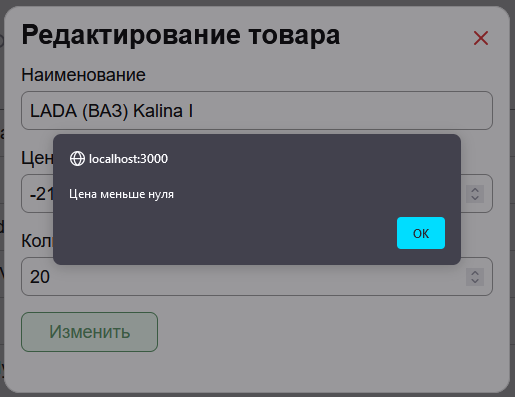


Рисунок . — Отрицательная цена

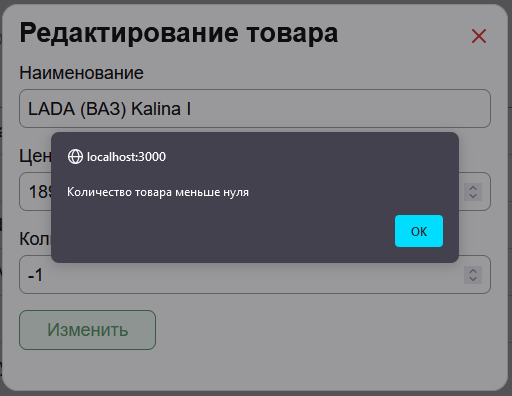


Рисунок . — Отрицательное кол-во товаров

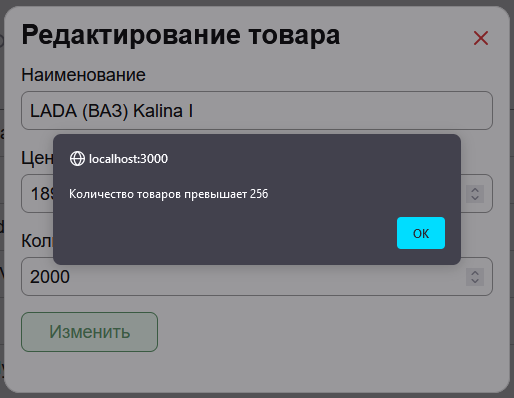


Рисунок . — Кол-во товаров превышает максимум

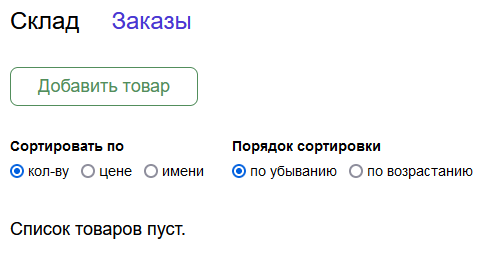


Рисунок . — Результат удаления всех товаров

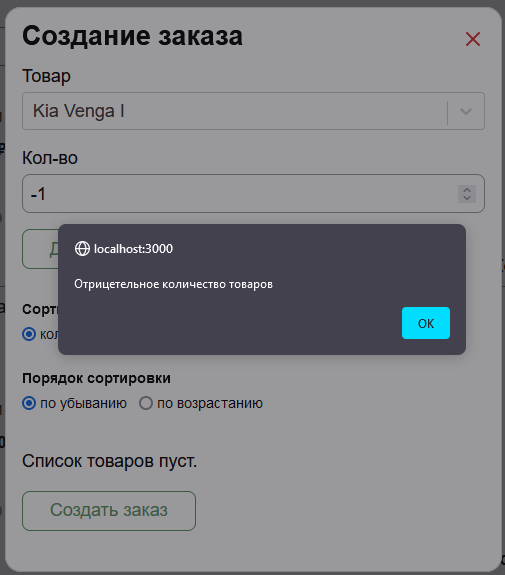


Рисунок . — Отрицательное кол-во товаров

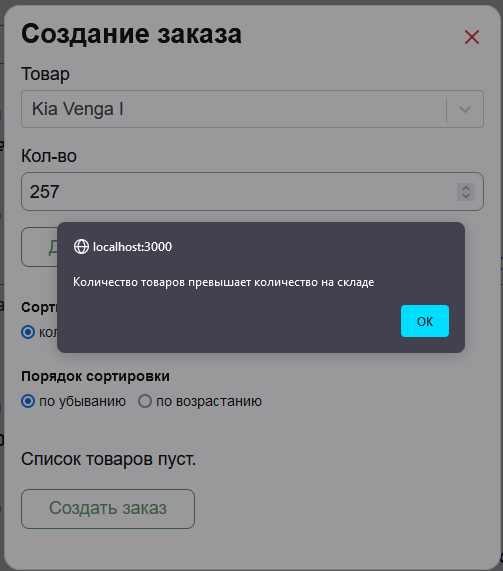


Рисунок . — Кол-во товаров превышает кол-во на складе

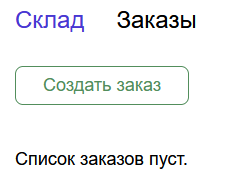


Рисунок . — Результат удаления всех заказов

# Вывод

В соответствии с указанным заданием, используя декларативный подход, было разработано веб-приложение. Интерфейс приложения прост и понятен, но имеет малое количество функций.

Для улучшения приложения следует реализовать следующее:

1. Загрузка данных из файла и сохранение их в файл;
2. Хранение данных на сервере, вместо пользовательского устройства;
3. Адаптировать приложение для мобильных устройств.

# Список используемых источников

* 1. Современный учебник JavaScript, [Электронный документ], (https://learn.javascript.ru/, дата обращения: 20.05.2022);
  2. Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений / Б. Черный. — Санкт-Петербург: Питер, 2021 г. – 352 с.;
  3. React — JavaScript-библиотека для создания пользовательских интерфейсов, [Электронный документ], (https://ru.reactjs.org/, дата обращения: 20.05.2022);
  4. Redux — a predictable state container for JavaScript apps, [Электронный документ], (https://redux.js.org/, дата обращения: 20.05.2022).