МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра анализа данных и искусственного интеллекта**

**ЛАБОРАТОРНАЯ РАБОТА №8**

**ДОКУМЕНТИРОВАНИЕ И РАЗВЕРТЫВАНИЕ ПС**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_И.Е. Дакукин

(подпись)

Направление подготовки   01.03.02 Прикладная математика и информатика курс 4

Направленность (профиль) Программирование и информационные технологии

Проверила  
канд. науч. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.Ю. Добровольская

Краснодар

2023

**Сценарное тестирование**

Сценарий 1 «Авторизация в системе»

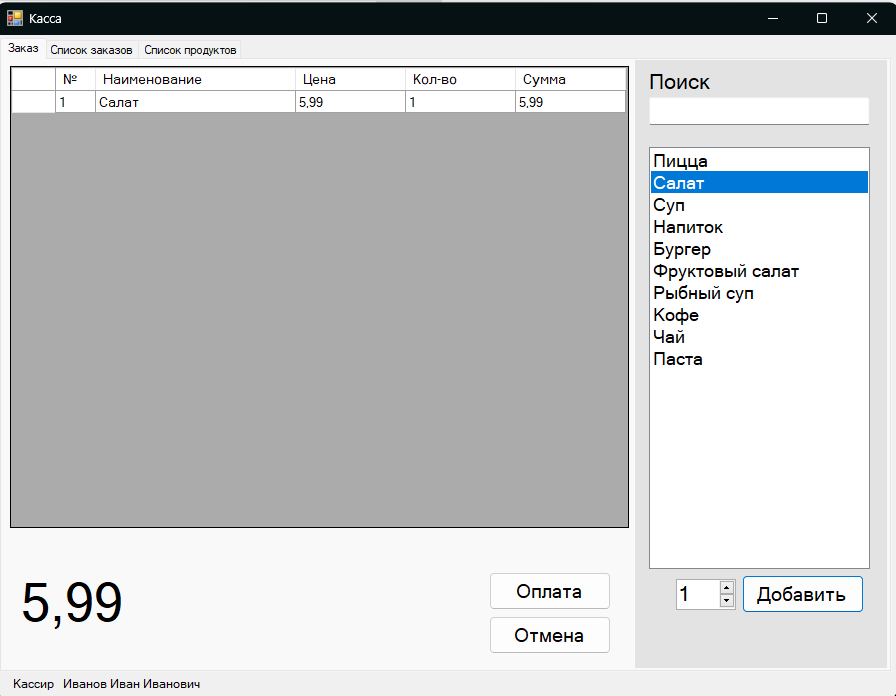
*Предусловие: система содержит данные для авторизации.*

1. Вводим логин и пароль в поля ввода
2. Нажимаем кнопку «войти»
3. Получаем доступ к системе
4. При неправильном вводе логина и/или пароля система должна вывести сообщение об ошибке
5. Если хотя бы одно из полей ввода пустое, система должна вывести сообщение об ошибке

Сценарий 2 «Добавить товар в заказ»

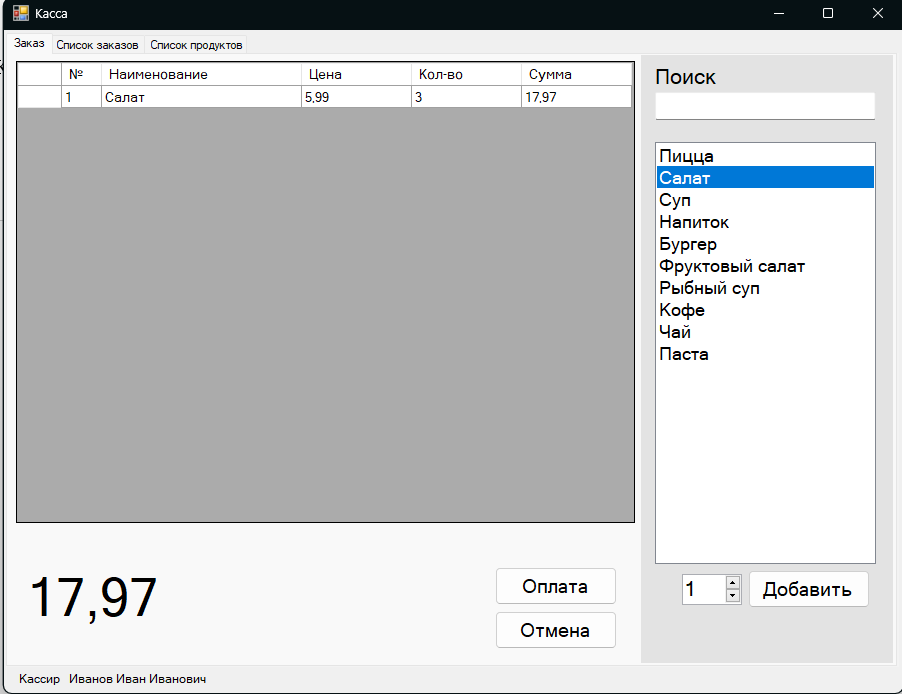
1. Выбираем товар из списка. При необходимости можно использовать строку поиска
2. Вводим количество товара.
3. Нажимаем кнопку «добавить»

Ожидаемый результат: строчка с информацией о товаре добавлена в основную таблицу, итоговая сумма была пересчитана.



Сценарий 3 «Изменить количество товаров»

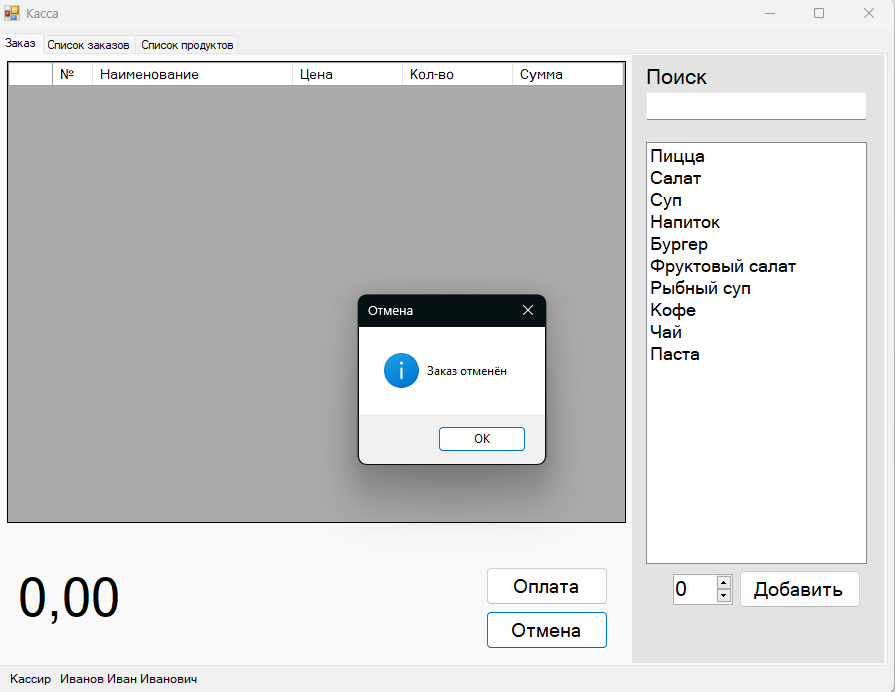
1. Добавляем один или несколько товаров в заказ.
2. Дважды нажимаем на ячейку, содержащую информацию, о количестве товара, которое мы хотим изменить.
3. Вводим новое значение
4. Нажимаем клавишу enter



Сценарий 4 «Отмена заказа»

1. Добавляем один или несколько товаров в заказ.
2. Нажимаем кнопку «отмена».

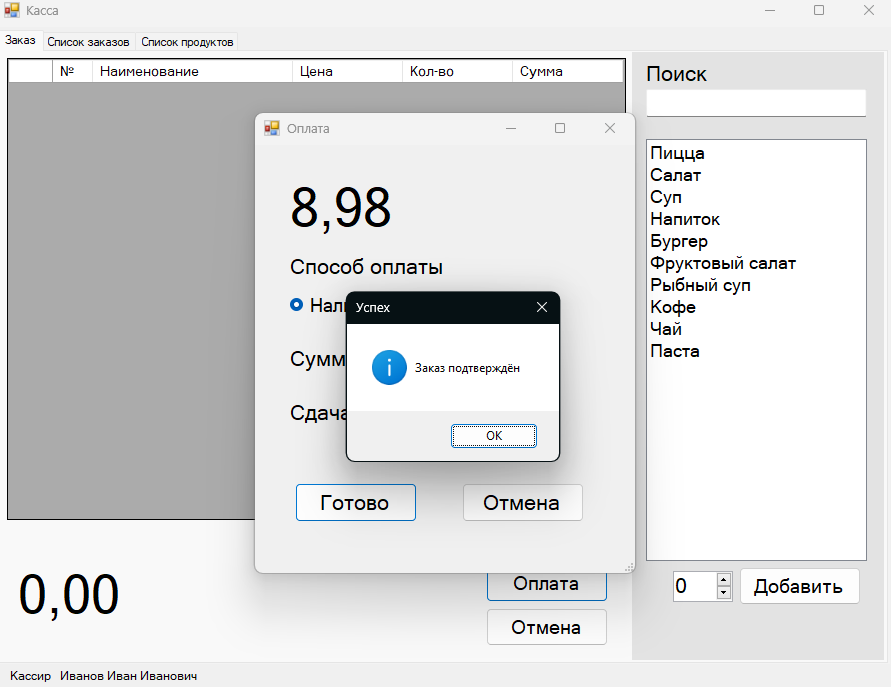
Ожидаемый результат: в базу данных добавляется запись о заказе со статусом «rejected». Выводится сообщение об отмене товара. Основное окно приложения возвращается к первоначальному виду (введённые данные очищаются).



Сценарий 4 «Оплата заказа»

1. Добавляем один или несколько товаров в заказ.
2. Нажимаем кнопку «оплата».
3. Выбираем способ оплаты.
4. При выборе способа оплаты «наличными» вводим внесённую сумму и получаем информацию о размере сдачи.
5. Нажимаем кнопку «Готово».

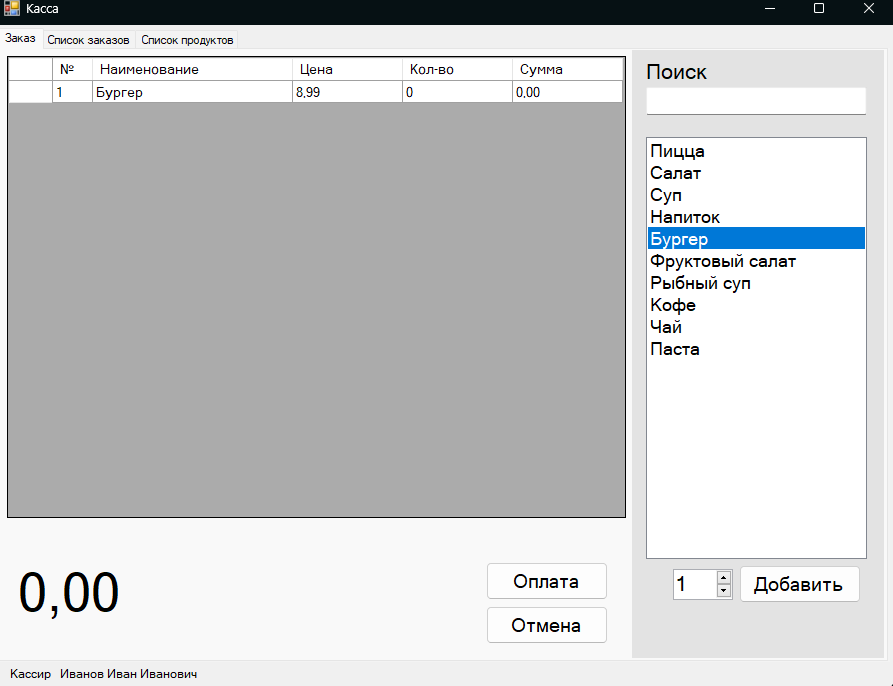
Ожидаемый результат: в базу данных добавляется запись о заказе со статусом «sucsessfull». Выводится сообщение об оплате товара. Основное окно приложения возвращается к первоначальному виду (введённые данные очищаются).



**Документирование логических ошибок**

Логическая ошибка 1:

* Сущность: при изменении количества продуктов на значение 0, в таблице остаётся информация о выбранном продукте. Другими словами, таким образом невозможно удалить позицию из заказа полностью
* Тестовая последовательность манипуляций: в сценарии 3 на шаге 2 вводим значение 0.
* Форма проявления:



* Форма проявления в коде:

private void dataGridView1\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

if(dataGridView1.Rows.Count == 0 || e.ColumnIndex != 3)

return;

string name = dataGridView1[1, e.RowIndex].Value.ToString();

double quantity = Convert.ToDouble(dataGridView1[e.ColumnIndex, e.RowIndex].Value);

foreach (var product in products)

{

if (product.Name == name)

{

decimal subtotal = (decimal)quantity \* product.Price;

decimal prevSubtotal = Convert.ToDecimal(dataGridView1[4, e.RowIndex].Value);

dataGridView1[4, e.RowIndex].Value = subtotal;

label4.Text = (Decimal.Parse(label4.Text) - prevSubtotal + subtotal).ToString();

conn.changeOrder(quantity, subtotal, chequeId, product.ID);

dataGridView1.CurrentCell = null;

break;

}

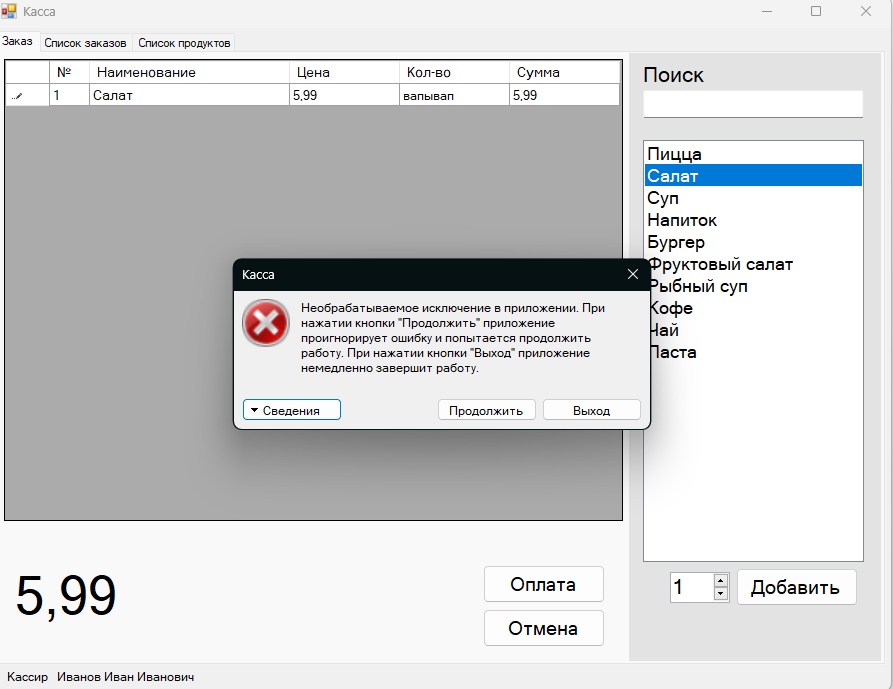
}

}

* Способ устранения: добавить проверку на ввод числа 0. При введение этого числа, удалять всю строку из таблицы.

Логическая ошибка 2:

* Сущность: при изменении количества продуктов можно ввести любую строку в любом формате.
* Тестовая последовательность манипуляций: в сценарии 3 на шаге 2 вводим значение «вапывап».
* Форма проявления:



* Форма проявления в коде:

private void dataGridView1\_CellValueChanged(object sender, DataGridViewCellEventArgs e)

{

if(dataGridView1.Rows.Count == 0 || e.ColumnIndex != 3)

return;

string name = dataGridView1[1, e.RowIndex].Value.ToString();

double quantity = Convert.ToDouble(dataGridView1[e.ColumnIndex, e.RowIndex].Value);

foreach (var product in products)

{

if (product.Name == name)

{

decimal subtotal = (decimal)quantity \* product.Price;

decimal prevSubtotal = Convert.ToDecimal(dataGridView1[4, e.RowIndex].Value);

dataGridView1[4, e.RowIndex].Value = subtotal;

label4.Text = (Decimal.Parse(label4.Text) - prevSubtotal + subtotal).ToString();

conn.changeOrder(quantity, subtotal, chequeId, product.ID);

dataGridView1.CurrentCell = null;

break;

}

}

}

* Способ устранения: добавить проверку на формат вводимой строки. Если вводится не число, не менять значение ячейки.

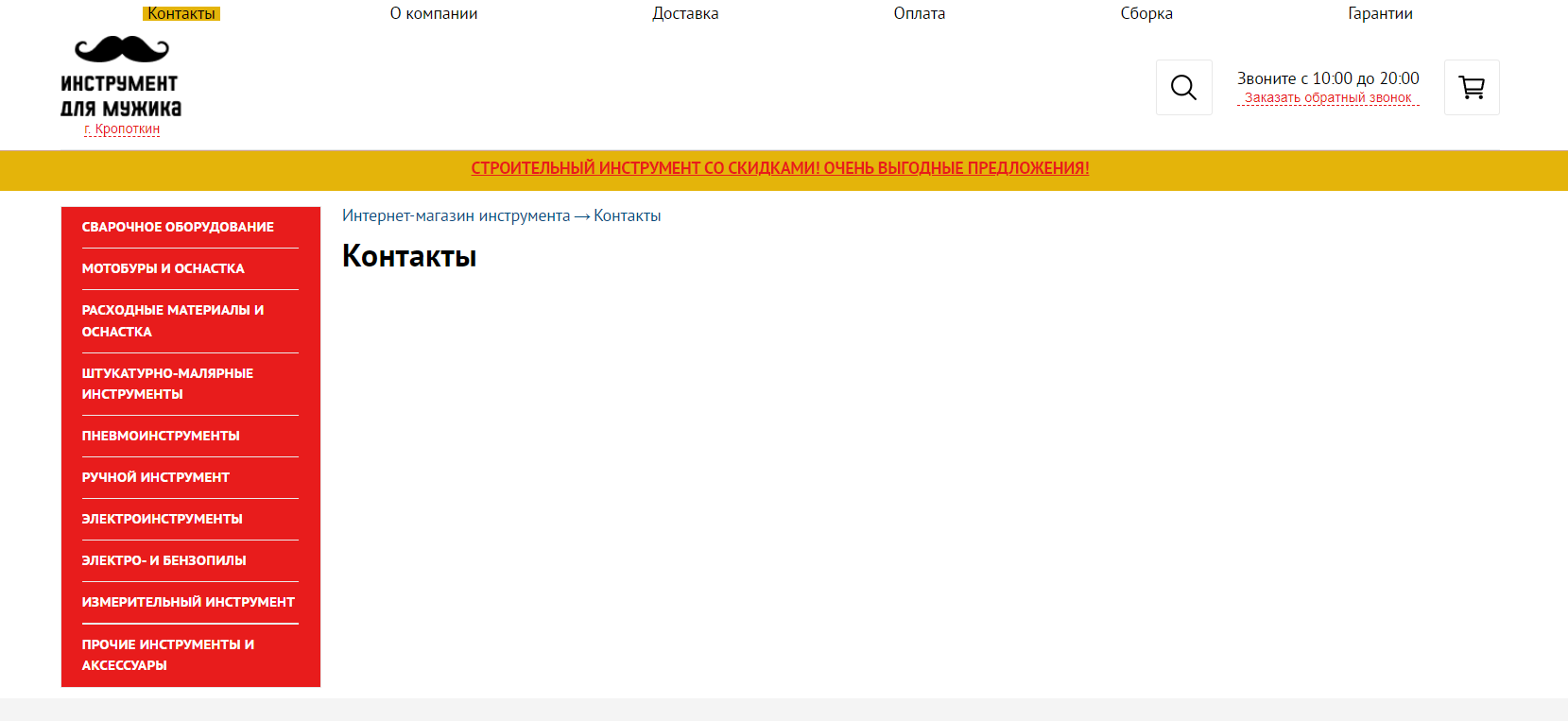
**Сценарное тестирование сайта kropotkin.sale-instrument.ru**

Сценарий 1 «Отображение вкладки контакты»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Перейти на вкладку «Контакты»

Ожидаемый результат: отображение контактных данных.

Фактический результат:



Сценарий 2 «Поиск по сайту»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Нажать на кнопку поиска

Ожидаемый результат: пользователю станет доступна строка поиска, не перекрывающая остальной интерфейс сайта

Фактический результат: Ввод в строку поиска недоступен

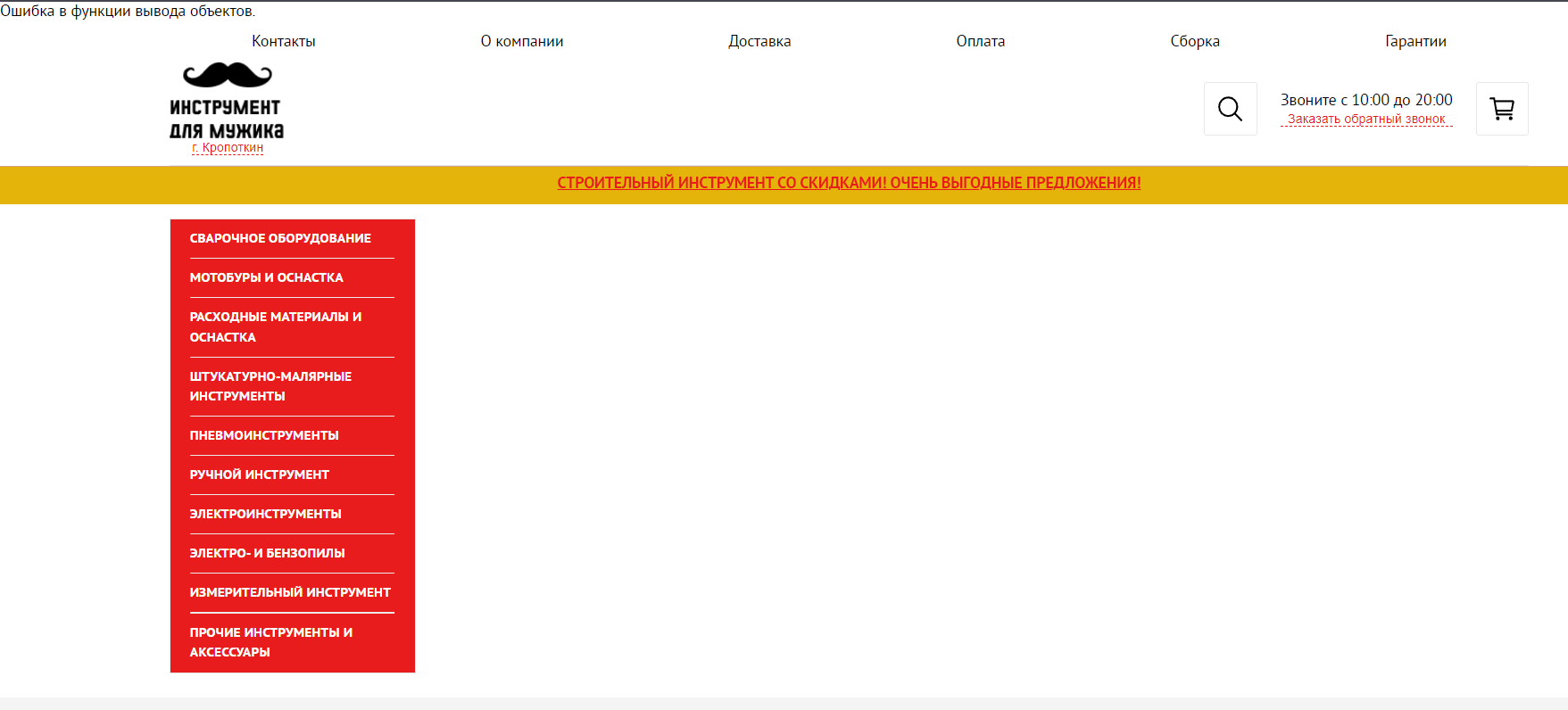


Сценарий 3 «Ввод сортируемой цены»:

1. Открыть сайт https://kropotkin.sale-instrument.ru/
2. Прокрутить страницу до блока «Каталог»
3. Выбрать любую из категорий
4. Выбирать любую из подкатегорий до тех пор, пока не отобразится страница со списком товаров
5. В поле для сортировки по цене ввести любой нечисловой символ
6. Нажать клавишу enter

Ожидаемый результат: сайт обработает ошибочный ввод и не изменит сортировку цены

Фактический результат:



**Нагрузочное тестирование ПС**

Тестирование БД.

В ходе тестирования было измерено время выполнения запросов INSERT, SELECT и TRUNCATE для следующих таблиц: cheque c 100, 500, 1500, 5000 и 10000 строками.

Запрос INSERT тестировался с помощью функций:

CREATE OR REPLACE FUNCTION insert\_random\_cheques(num\_rows INT)

RETURNS VOID AS $$

DECLARE

i INT := 1;

statuses TEXT[] := ARRAY['rejected', 'succesfull', 'created'];

BEGIN

WHILE i <= num\_rows LOOP

INSERT INTO cheque (cashier\_id, status, total, handle\_date, in\_cash)

VALUES (

1,

statuses[1 + (i % 2)],

ROUND((random() \* 1000)::NUMERIC, 2)::MONEY,

current\_date - (random() \* 365)::INT,

random() < 0.5

);

i := i + 1;

END LOOP;

END;

$$ LANGUAGE plpgsql; $$ LANGUAGE plpgsql;

Таблица 1 – время выполнения запросов INSERT

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 4,626 |
| 500 | 17,309 |
| 1500 | 45,808 |
| 5000 | 173,438 |
| 10000 | 340,906 |

Запрос SELECT тестировался с помощью функции:

EXPLAIN ANALYZE SELECT \* FROM cheque

Таблица 2 – время выполнения запросов SELECT

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 0,034 |
| 500 | 0,064 |
| 1500 | 0,156 |
| 5000 | 0,411 |
| 10000 | 1,124 |

Запрос TRUNCATE тестировался с помощью функции:

EXPLAIN ANALYZE TRUNCATE cheque CASCADE

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 20,426 |
| 500 | 25,058 |
| 1500 | 26,843 |
| 5000 | 22,390 |
| 10000 | 23,533 |

Тестирование приложения

В ходе тестирования было измерено время полного запуска приложения в зависимости от количества строк в таблице tourist: 10, 50, 150, 500 и 1000.

Таблица 4 – время полного запуска приложения

|  |  |
| --- | --- |
| Кол-во строк | Время выполнения (в мс) |
| 100 | 1238 |
| 500 | 1241 |
| 1500 | 1352 |
| 5000 | 1840 |
| 10000 | 2943 |