### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕУЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

### Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Курсовая работа**

Выполнили: ст. группы ВТ-222

Маркевич Алесандр Александрович

Максименко Александр

Михайлович

Проверил:

Черников Сергей Викторович

Белгород 2023 г.

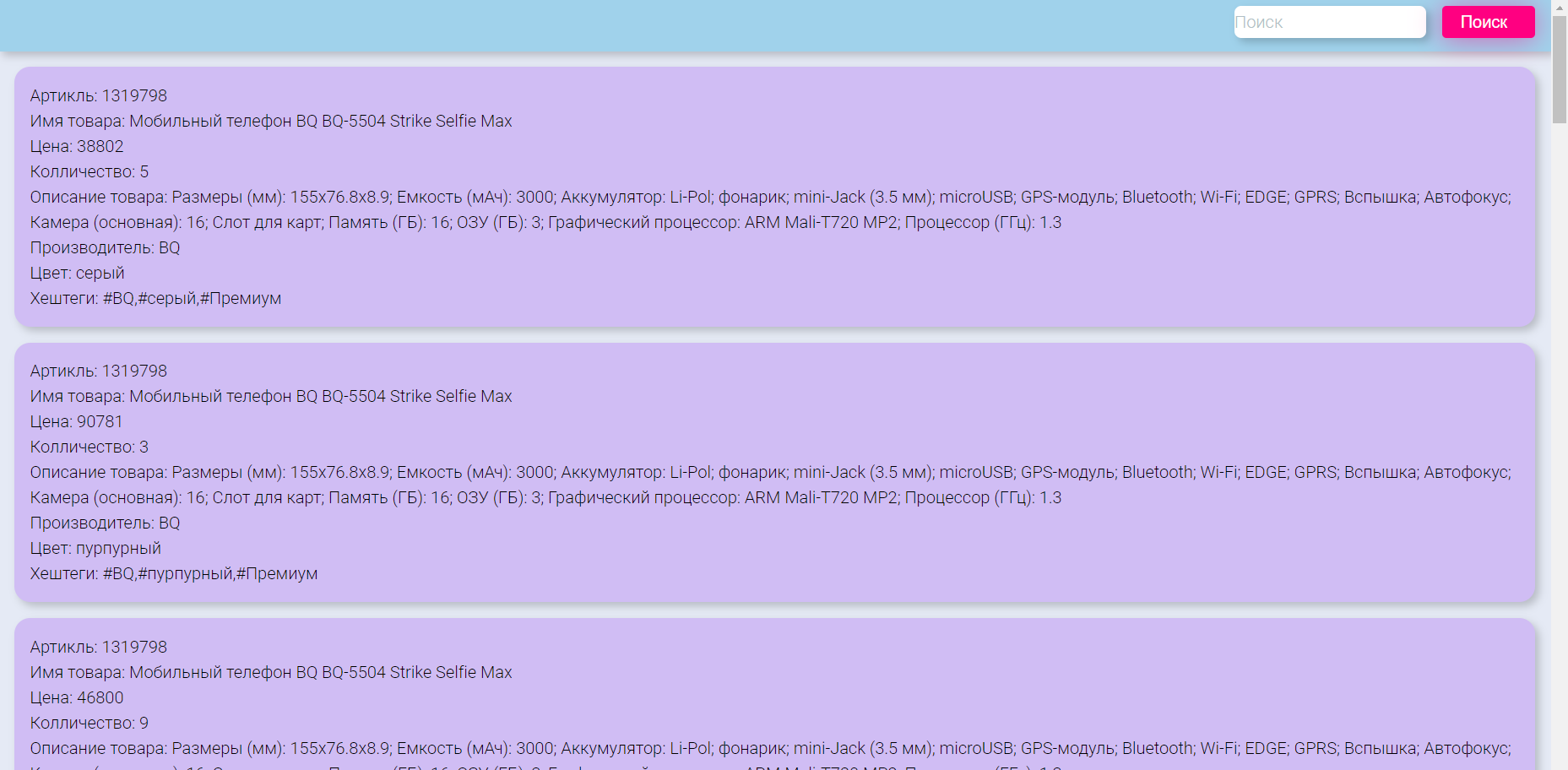
**Цель работы:** Программно реализовать алгоритмы сортировки и поиска, а также разработать прототип интернет-магазина.

**Стек технологий: C# -** версия 10.0**, .Net Core 7(Web api), TypeScript –** версия5.0, **Scss -** версия1.69.4, **HTML5**, **Angular –** версия 7, **Entity Framework Core –** версия 7, **Swagger –** версия 3.0.3, **PostgreSQL ~15**

**Часть frontend-приложения**

На стороне frontend были реализованы UI-компоненты, а также сортировки данных приходящих с backend части приложения.

Изначальный экран приложения пользователя выглядит таким образом:



После поиска:



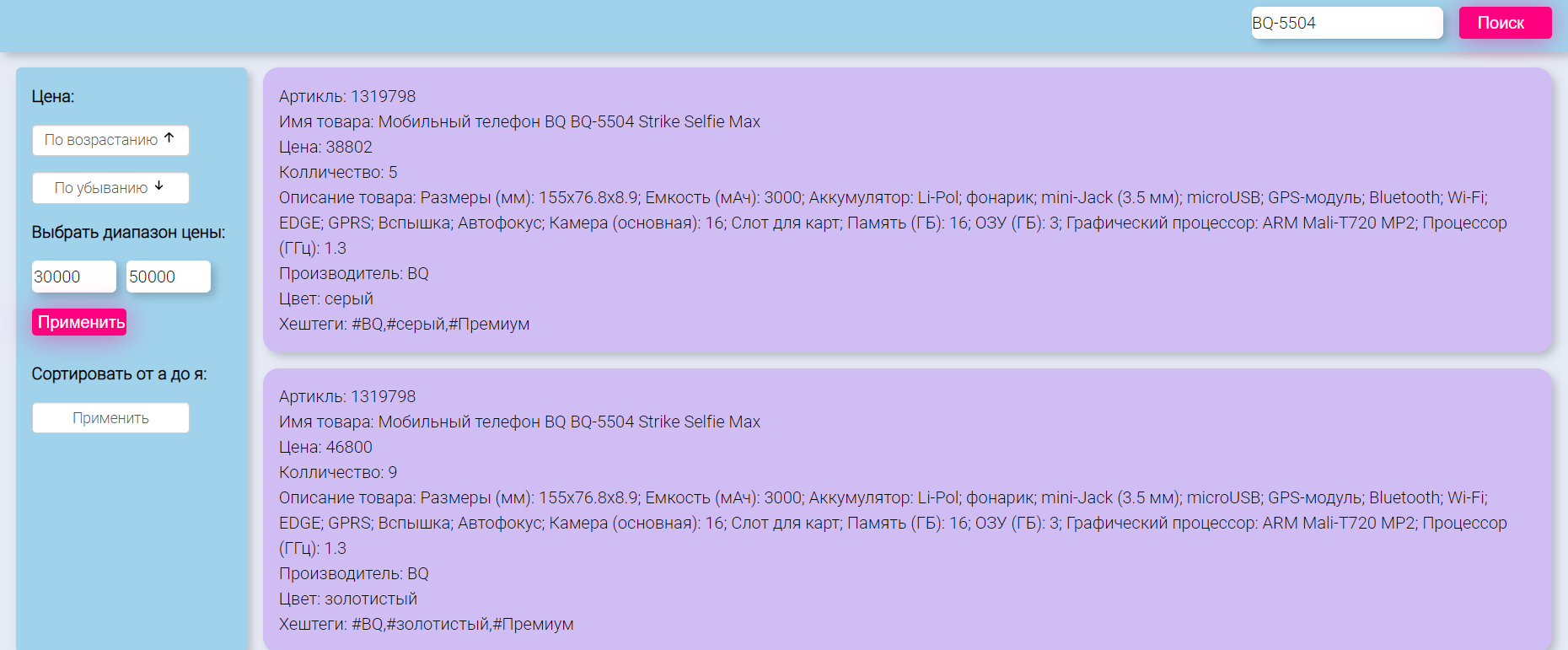
Использование сортировки по возрастанию цены:



Использование сортировки по убыванию цены:

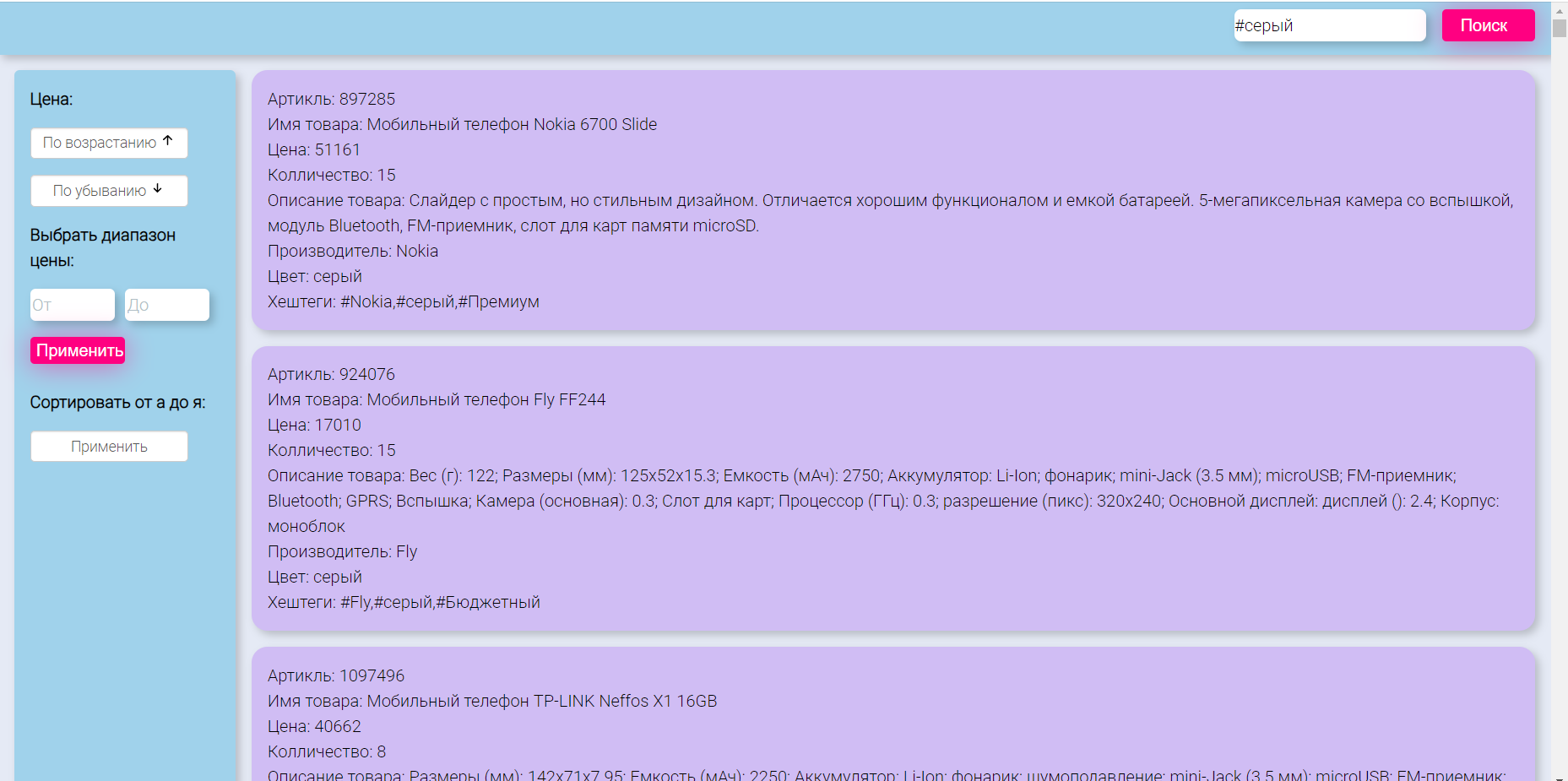


Использование сортировки диапазона цены:



Использование сортировки от a до я:

Было:



Стало:



Дополнительно. Пример ввода в поле более сложный запрос для сервера:

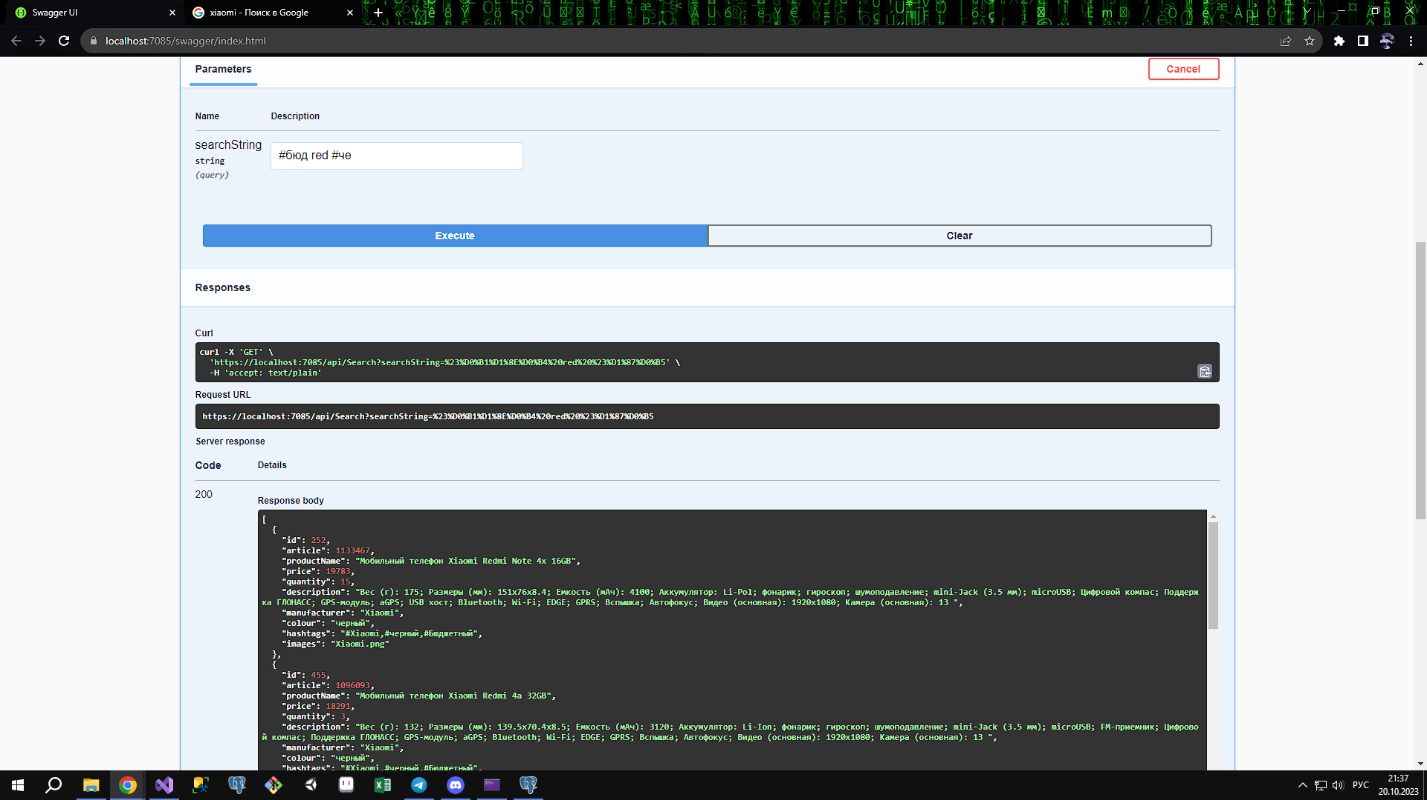


**Часть backend-приложения**

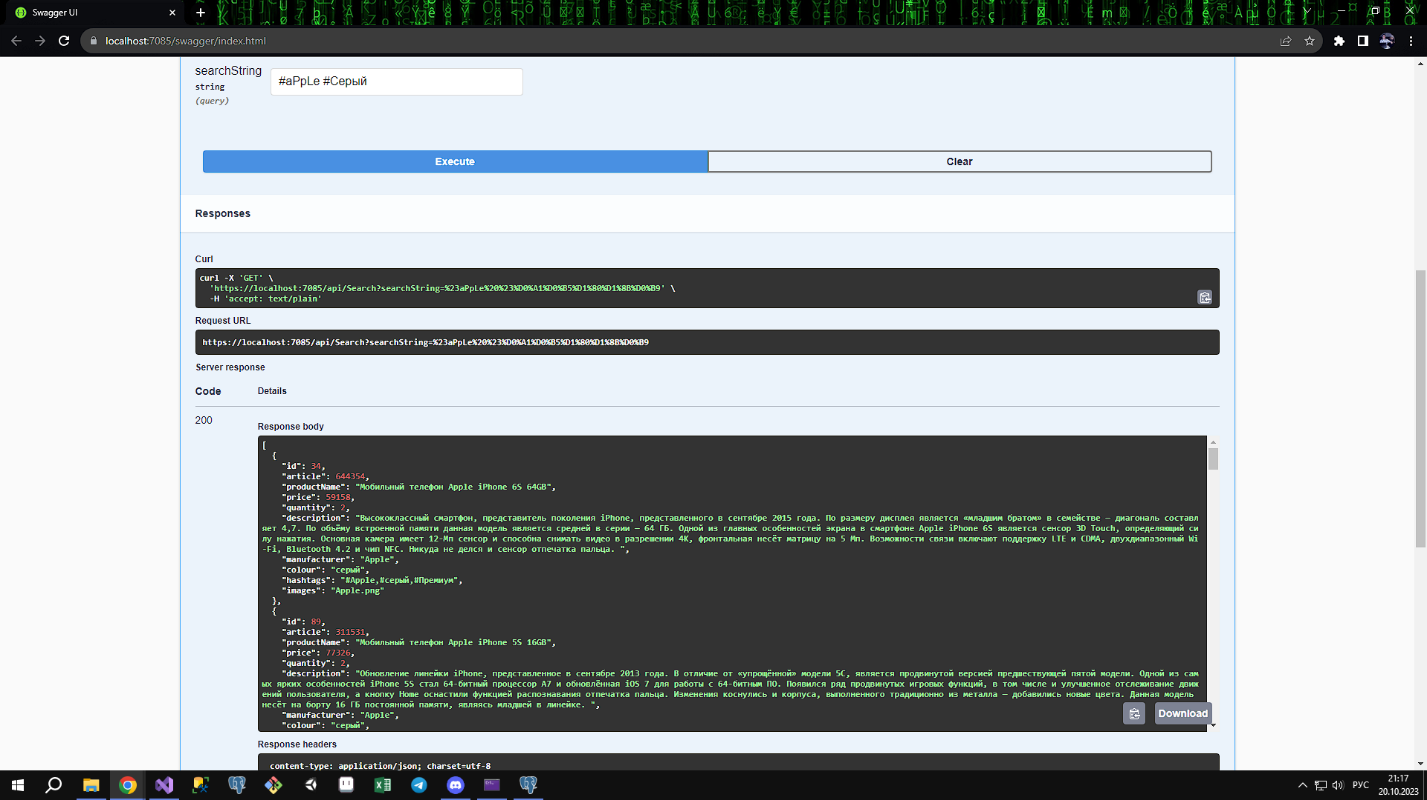
На стороне backend была реализована таблица в базе данных, модели под нее, GRUD-операции, а также алгоритм поиска.

Тестирование поиска:

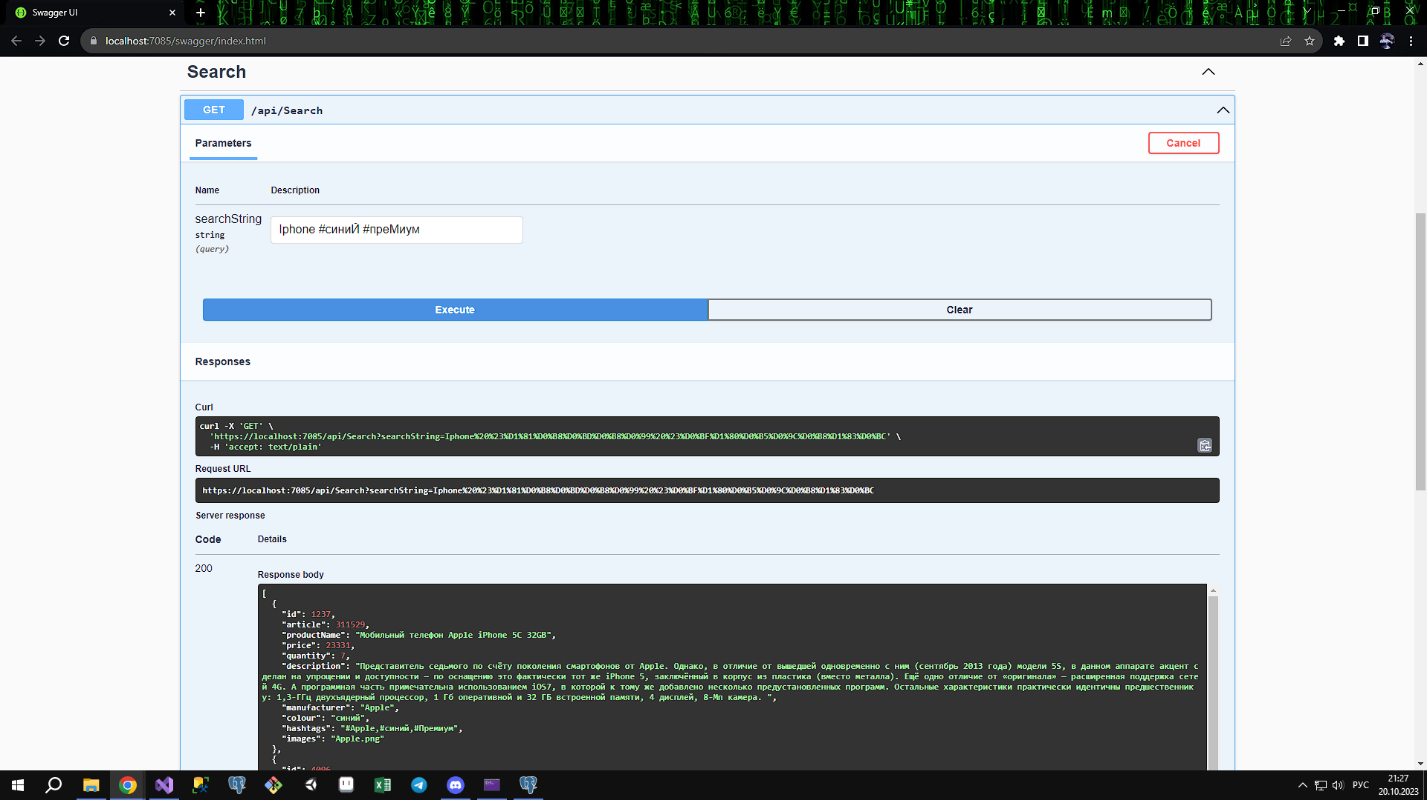
2 Хэштега и 1 имя (неполные)



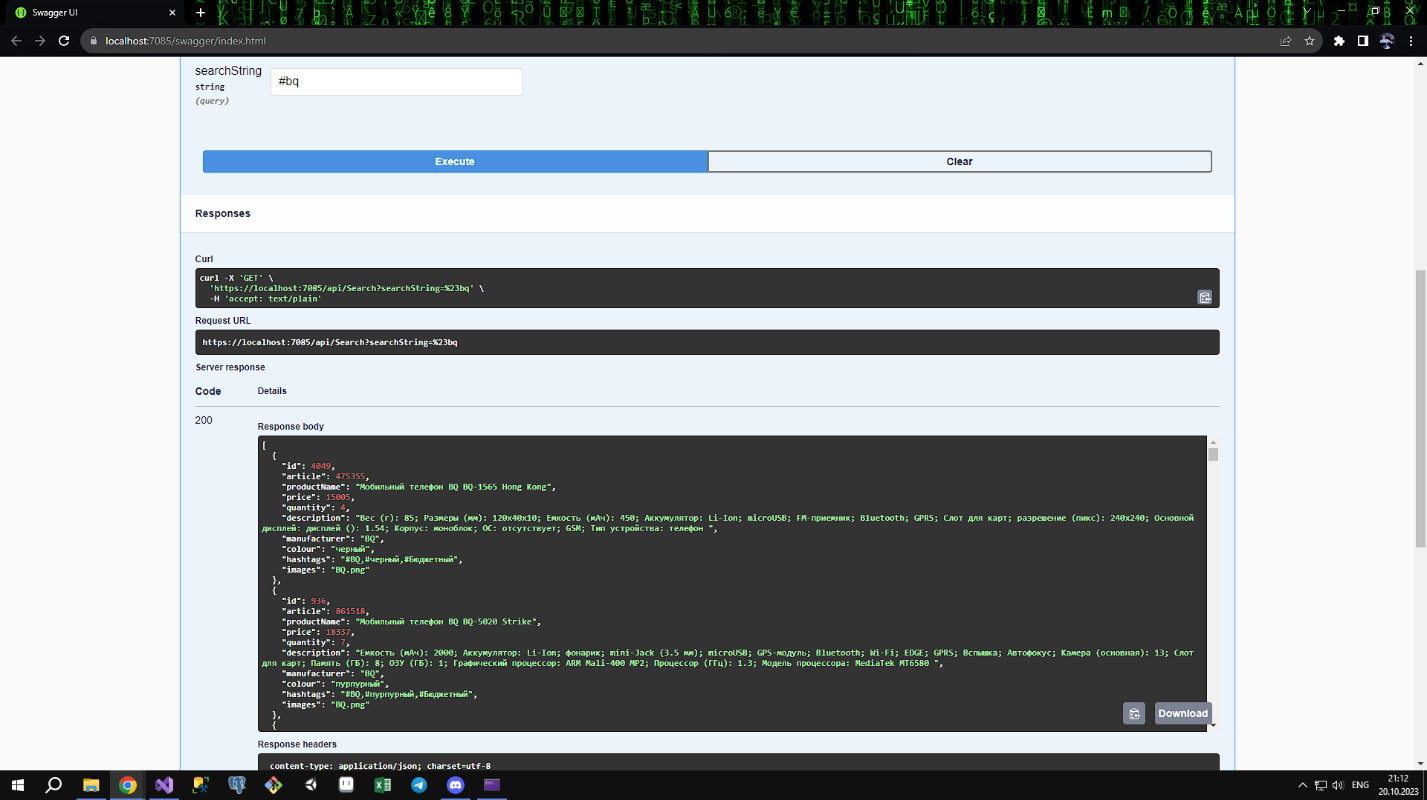
2 Хэштега (С большими буквами)



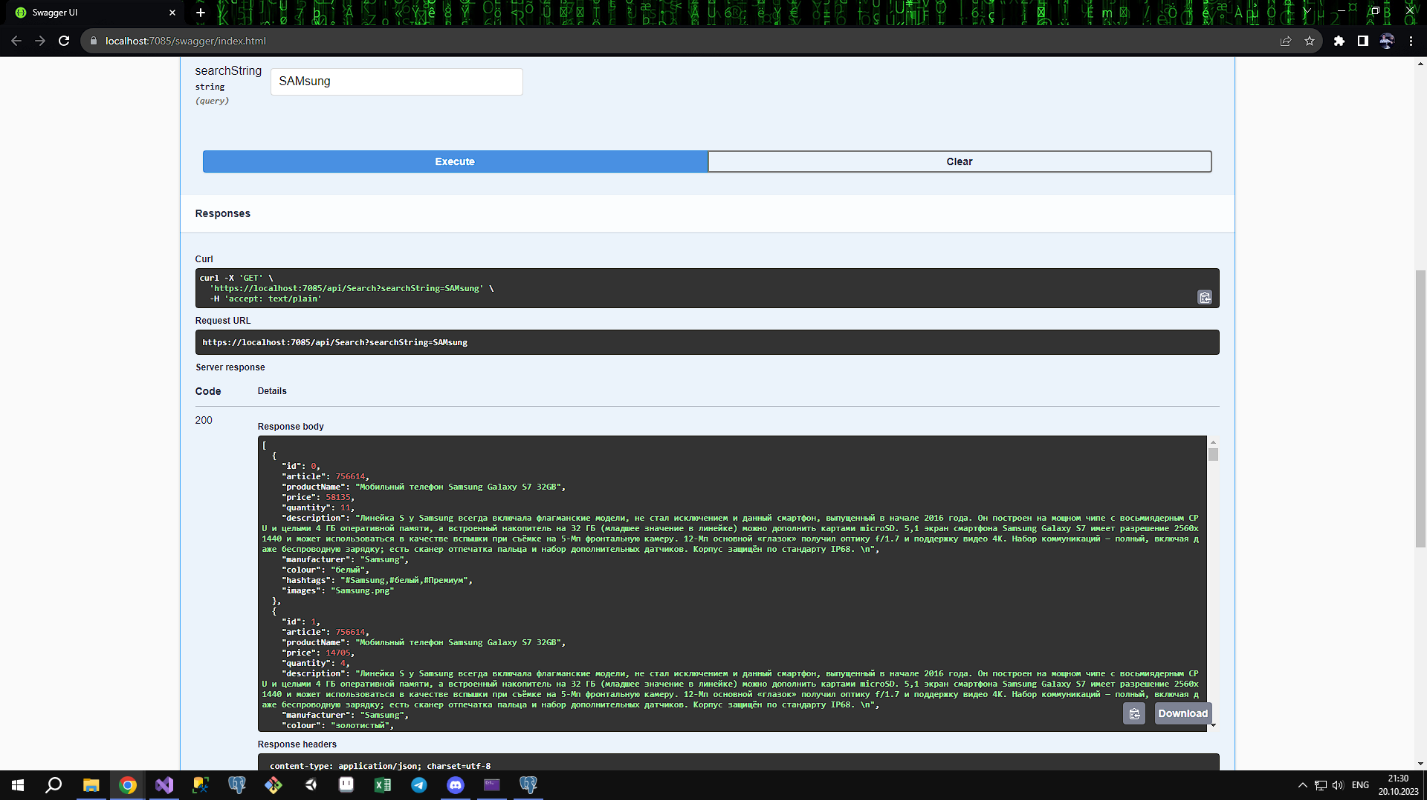
2 Хэштега и 1 имя (С большими буквами)



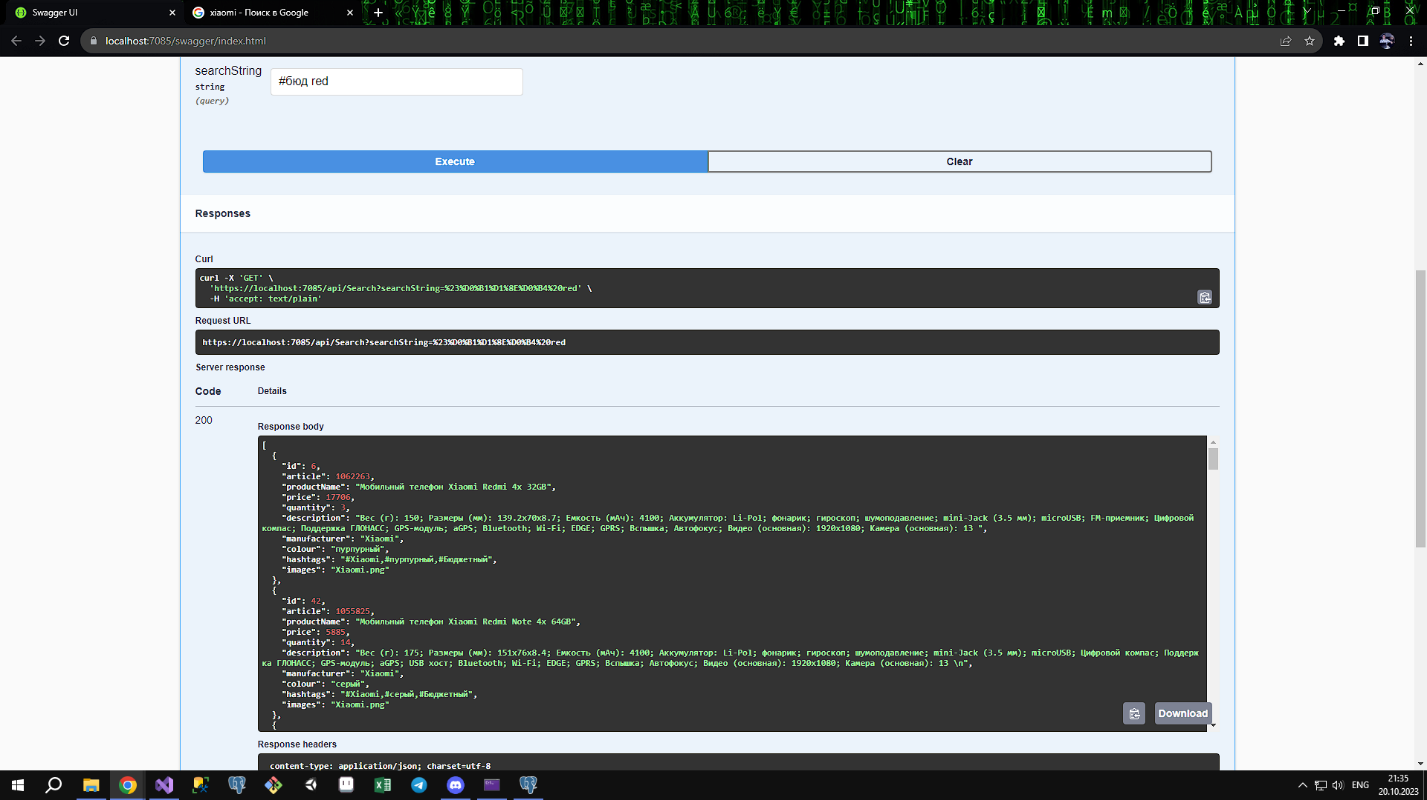
1 Хэштег



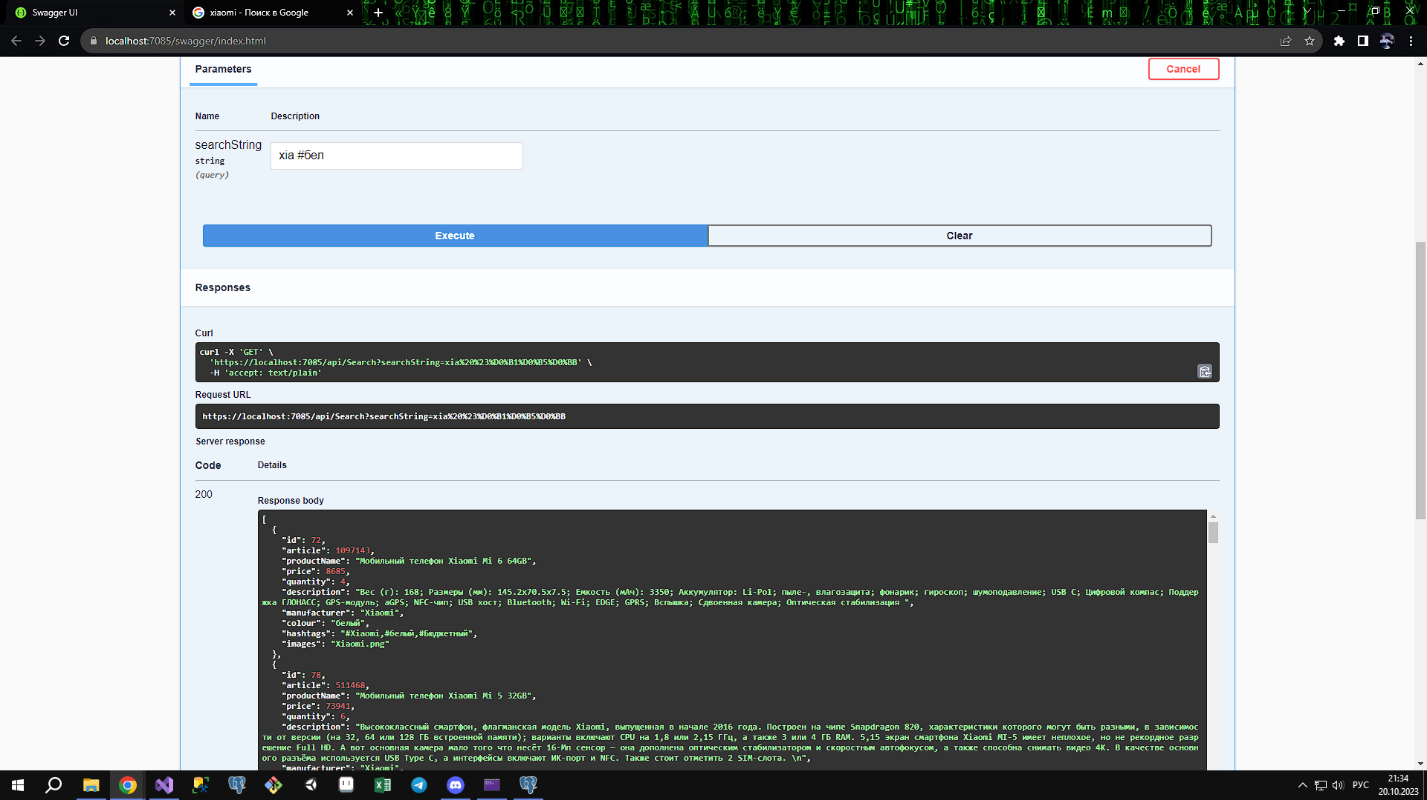
1 имя (С большими буквами)



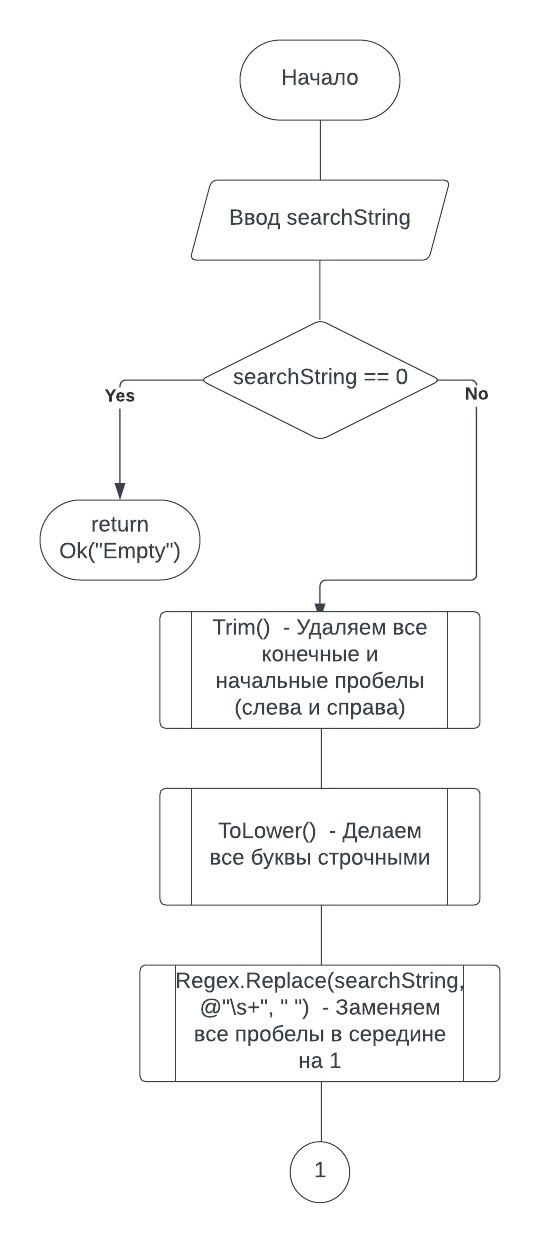
1 Хэштег и 1 имя (неполные)

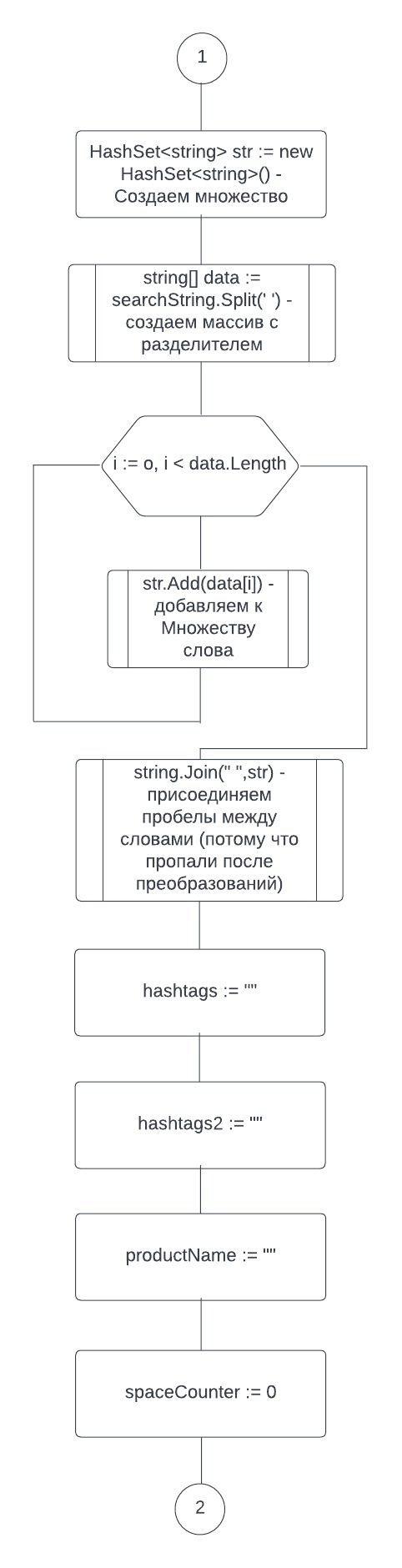


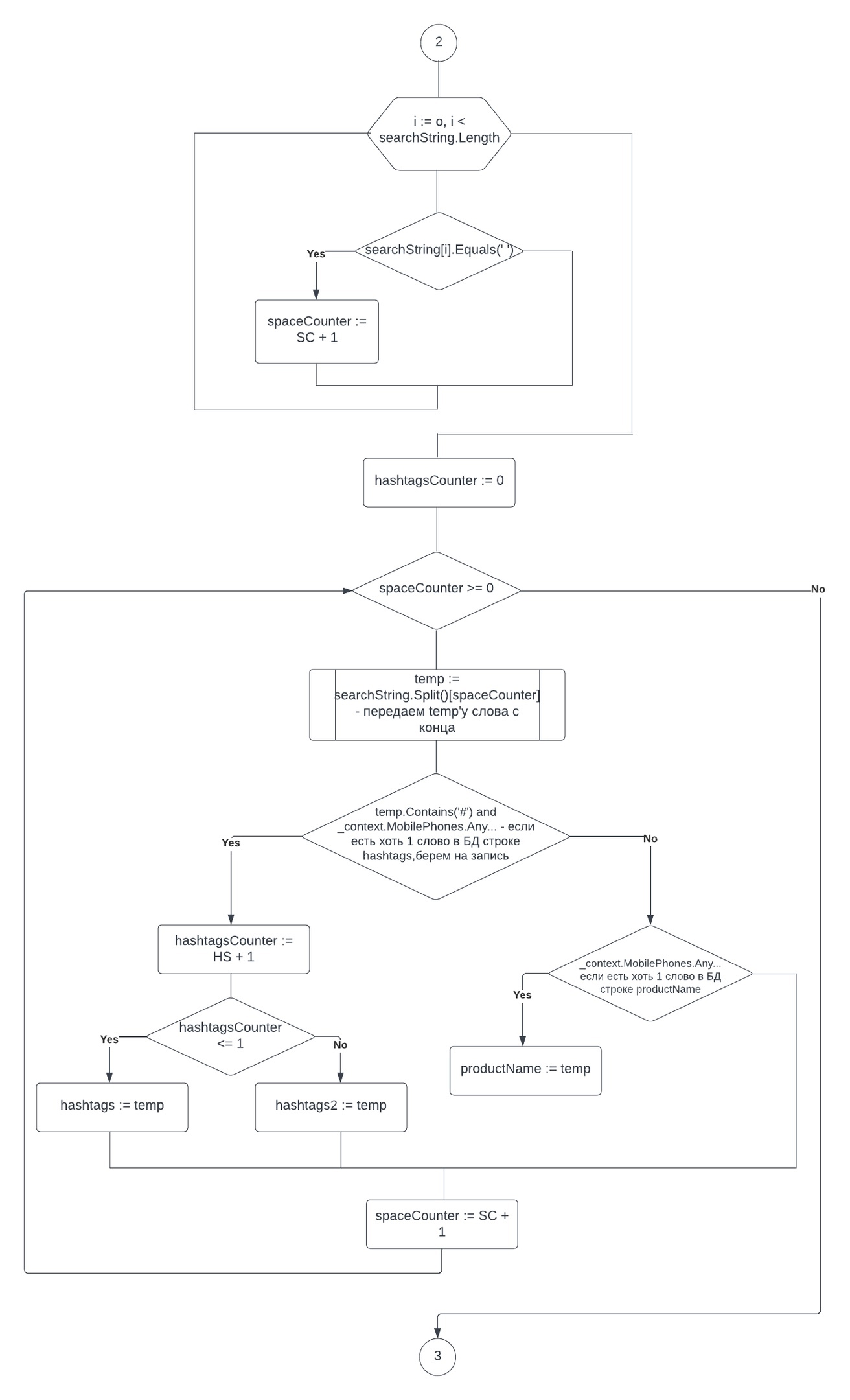
1 Имя и 1 хэштег (неполные)

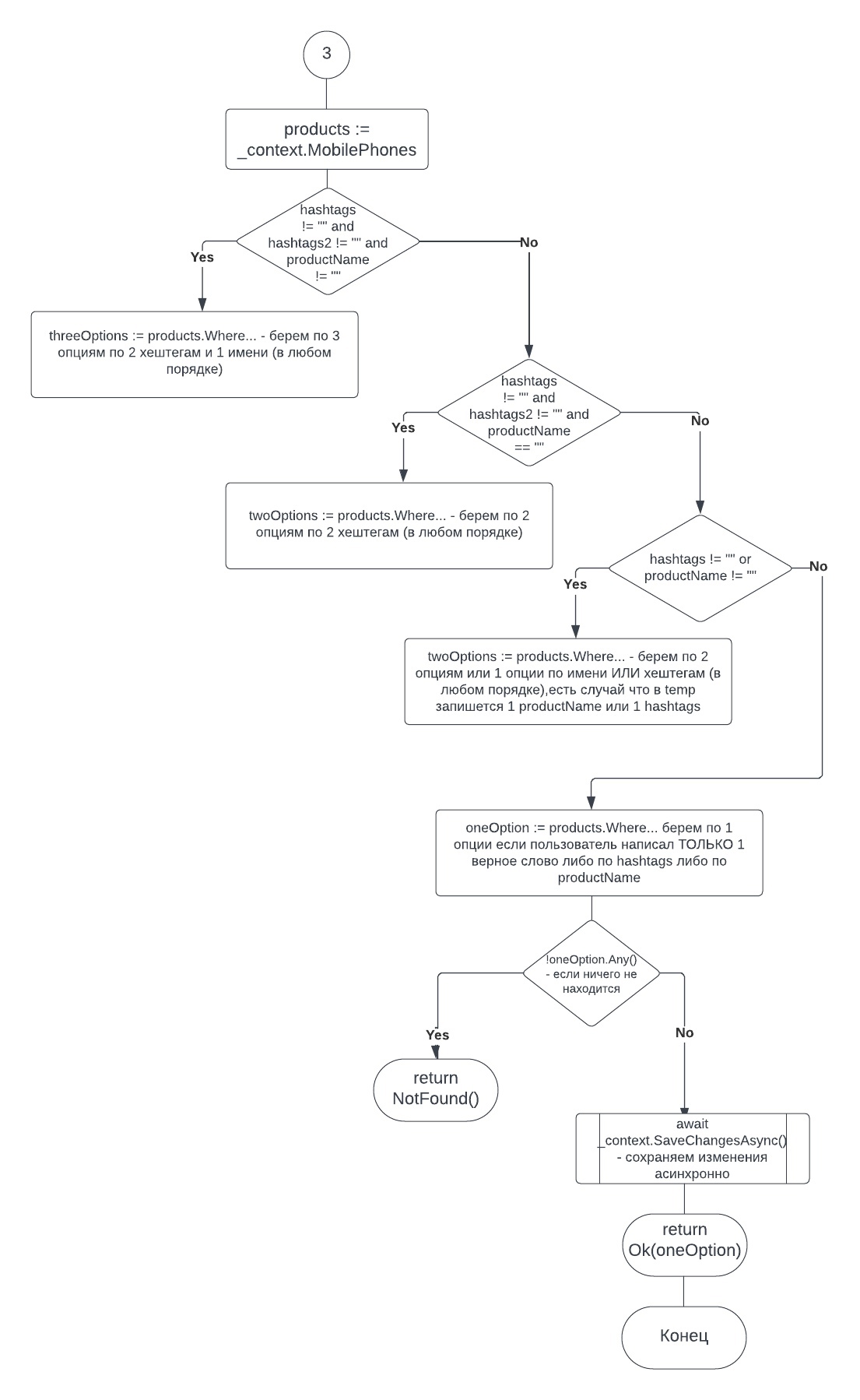


**Блок-схема алгоритма поиска:**









**Сложность данного алгоритма:**

**Худшая и Средняя – O(n) -** так как нет вложенных циклов**, Лучшая – O(1) –** так как присутствуют условия пустой строки или строки которой нет.

**Список литературы:**

1. <https://metanit.com/sharp/efcore/>
2. <https://learn.microsoft.com/ru-ru/ef/core/>
3. <https://angular.io/docs>

**Вывод.** Программно реализованы алгоритмы сортировки и поиска, а также разработан прототип интернет-магазина.

**Что можно поправить со стороны backend’a:**

В данном алгоритме идет уклон на запись и сохранение ограниченного числа переменных (слов) с конца строки до начала и проверка на пустую строку, также на нахождение слова в БД. Из-за этого при добавлении новых, допустим, хэштегов придется постоянно лезть в код и обновлять условия. Поэтому лучшим решением в будущем будет – автоматизация.

**Что не удалось реализовать в проекте**

Не удалось реализовать передачу изображения из backend в frontend.