**Министерство науки и высшего образования Российской Федерации** **Федеральное государственное бюджетное образовательное учреждение высшего образования** **«Московский государственный технический университет** **имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Домашнее задание

по дисциплине «Разработка интернет приложений»

Выполнила:

студентка группы ИУ5-54Б

Харитонов А.А.

Проверил:

Канев А.И.

2022 г.

**СОДЕРЖАНИЕ**

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 3](#_Toc122938955)

[1. Цель 3](#_Toc122938956)

[2. Задачи 3](#_Toc122938957)

[3. Требования к функциональным характеристикам 3](#_Toc122938958)

[4. Требования к программному обеспечению 4](#_Toc122938959)

[ВВЕДЕНИЕ 6](#_Toc122938960)

[1. Описание предметной области 6](#_Toc122938961)

[2. Актуальность 6](#_Toc122938962)

[3. Бизнес-процесс 7](#_Toc122938963)

[1. Диаграмма прецедентов 7](#_Toc122938964)

[2. Диаграмма деятельности 8](#_Toc122938965)

[3. Диаграмма состояний 9](#_Toc122938966)

[4. Схема бизнес-процесса 10](#_Toc122938967)

[АРХИТЕКТУРА 11](#_Toc122938968)

[1. Диаграмма развертывания 11](#_Toc122938969)

[2. ER с назначением таблиц 12](#_Toc122938970)

[3. Диаграмма компонентов 12](#_Toc122938971)

[АЛГОРИТМЫ 13](#_Toc122938972)

[1. Диаграмма последовательности 13](#_Toc122938973)

[ОПИСАНИЕ ИНТЕРФЕЙСА 14](#_Toc122938974)

[1. Гость 14](#_Toc122938975)

[2. Пользователь 19](#_Toc122938976)

[3. Поставщик 21](#_Toc122938977)

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

# Цель

Цель состоит в изучении основ различных фреймворков и инструментов для создания полноценного интернет-приложения, а также в изучении построения диаграмм и документации, сопровождающих приложение.

# Задачи

1. Базовая шаблонизация в Django
2. Создание базового интерфейса для просмотра списка с ссылками и частью атрибутов
3. Создание базы данных, подключение к шаблонизатору
4. Создание страницы администратора
5. Создание веб-сервиса для получения и редактирования данных из БД
6. Работа с фронтендем
7. Подключение интерфейса фронтенда к веб-сервису
8. Возможность добавления товара в корзину (увеличение количества товара), а также удаление товара их корзины
9. Добавление фильтрации и поиска на странице товаров
10. Переключение между интерфейсом гостя и интерфейсом пользователя
11. Аутентификация и регистрация
12. Авторизация через хранение сессий
13. Создание нативного PWA приложения
14. Добавление роли пользователя-менеджера контента

# Требования к функциональным характеристикам

В разрабатываемом веб-приложении должны быть следующие роли пользователей:

1. Гость

Может просматривать товары и информацию о них. Может просматривать список поставщиков и их товары, выставленные для продажи. Имеет возможность фильтровать товары и поставщиков. Имеет возможности авторизации и регистрации.

1. Обычный пользователь

Имеет возможности гостя. Имеет страницу профиля с собственными заказами. Может добавлять и удалять товары к создаваемому заказу. Может создавать заказ на поставленную дату и поставленную локацию. Также пользователь может зарегистрироваться как поставщик. Может оплачивать заказ, если он был принят поставщиком.

1. Поставщик

Наследует возможности пользователя. Имеет личную страницу, отображаемую в общем списке поставщиков. Может добавлять и удалять товар. Может просматривать заказы, которые оформили у него другие пользователи. Может принимать или отклонять их.

1. Администратор

Работает с панелью администратора django. Способен редактировать, удалять, добавлять, просматривать поля любой таблицы системы.

# Требования к программному обеспечению

**Требования к ПО для клиента:**

Клиент может просматривать веб-приложение на таких браузерах, как Chrome, Opera и другие. Для этого характеристики ПО должны быть следующими:

1. **Windows**

Windows 7, Windows 8, Windows 8.1, Windows 10 или версии позднее.

Процессор Intel Pentium 4 или более поздней версии с поддержкой SSE3.

Примечание: также поддерживаются следующие серверные ОС: Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2 или Windows Server 2016.

1. **macOS**

macOS High Sierra 10.13 или более поздней версии

1. **Linux**

Ubuntu 18.04 (64-разрядная версия) или более поздней версии, Debian 10 или более поздней версии, openSUSE 15.2 или более поздней версии, Fedora Linux 32 или более поздней версии.

Процессор Intel Pentium 4 или более поздней версии с поддержкой SSE3.

1. **Требования к ПО для работы системы:**

Центральный процессор: с частотой от 1 ГГц

Оперативная память: 512 Мб и выше

Пространство на жестком диске: 20 Мб

Платформа: Microsoft Windows, Mac OS, Linux x32/x64

# ВВЕДЕНИЕ

# Описание предметной области

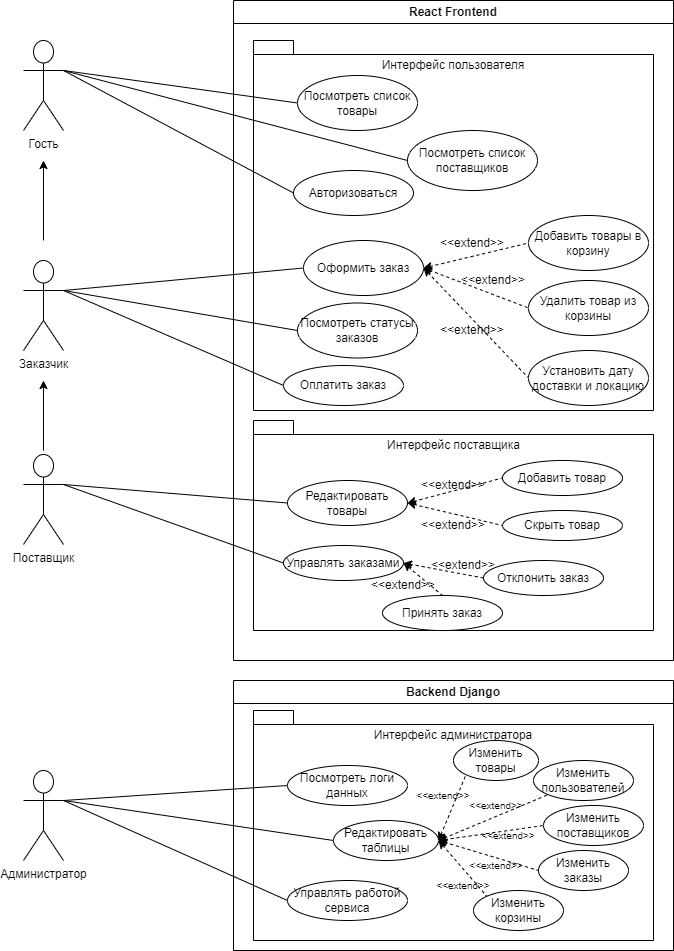
Предметная область домашнего задания: сайт организации оптовых поставок. Смысл веб-приложения состоит в оформлении пользователями оптовых заказов у поставщиков, выставляющих свои товары на продажу. Оптовая закупка схожа с обычной покупкой в интернете. Основные отличия – количества закупаемых товаров и логистика их доставки. Это накладывает некоторые ограничения на то, чтобы владелиц бизнеса закупал товары в обычном магазине. Данный сайт был построен таким образом, чтобы UX был максимально подобен опыту посещения обычного интернет-магазина. Самым ощутим отличием в разработанной системе является количество состояний жизненного цикла. Так после оформления заказа он отдаётся на утверждение поставщикам, которые рассматривают возможность и выгодность предлагаемого им заказа. Логистика доставки организуется поставщиками. Поэтому заказ считается легитимным только в случае, если каждый поставщик утвердит свои поставки. Только после этого пользователь имеет возможность внесения оплаты.

# Актуальность

Данное веб-приложение актуально для владельцев бизнеса имеющего задачу организовать поставку некоторого товара в установленное место в установленное время. Также данное приложение актуально для поставщика, желающего получать прибыль, совершая такие оптовые поставки.

# Бизнес-процесс

# Диаграмма прецедентов



На диаграмме описано возможные поведения пользователей приложения. Описаны следующие участники:

1. Гость

Имеет возможности авторизации, просмотра товаров и поставщиков.

1. Авторизированный пользователь

Имеет возможность формировать заказы.

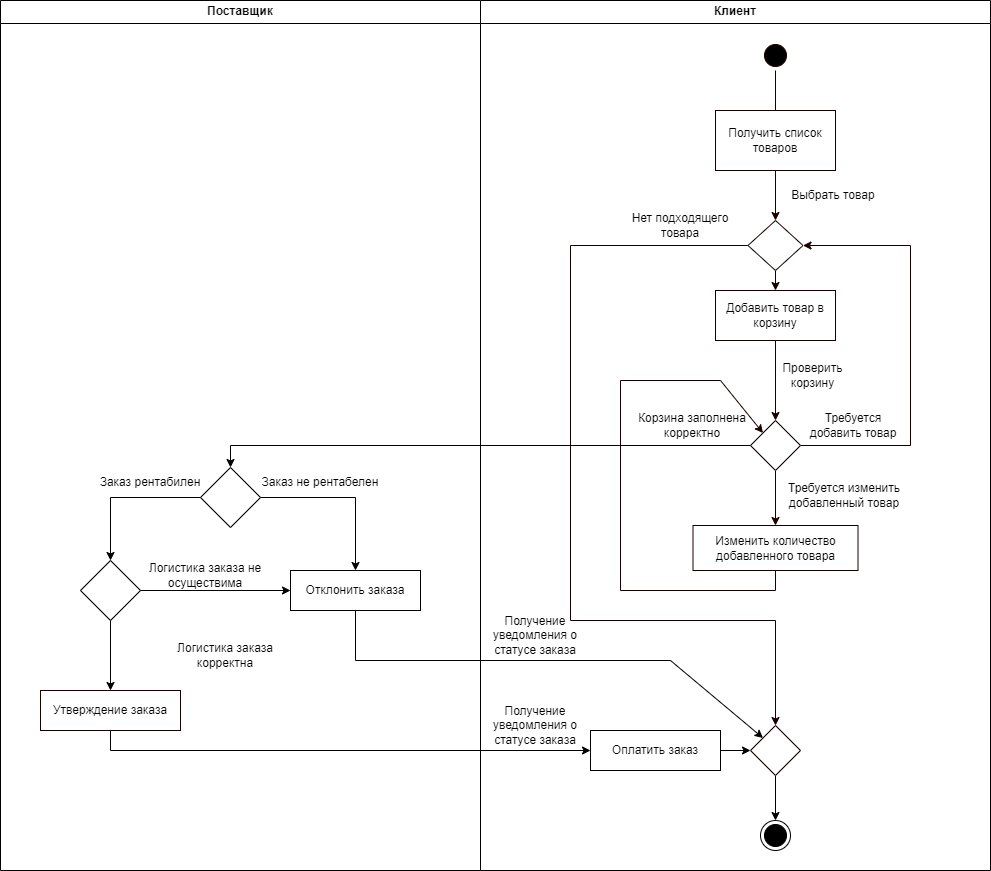
1. Поставщик

Имеет возможность добавлять или удалять товары, просматривать заказы и принимать или отклонять их.

1. Администратор

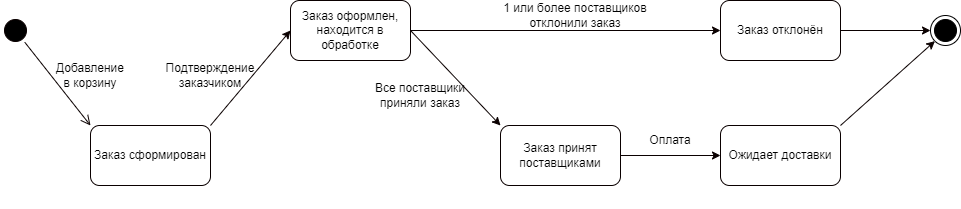
Имеет полный доступ к таблицам системы.

# Диаграмма деятельности



На данной диаграмме описан процесс добавления пользователем товаров в корзину, её оформление в заказ и его оплата в случае его легитимности. Пользователь может изменять количество отдельного товара в корзине, а также удалять или добавлять товары. После оформления заказа поставщики принимают решение о его легитимности, после чего пользователю приходит уведомление об изменившемся статусе заказа.

# Диаграмма состояний

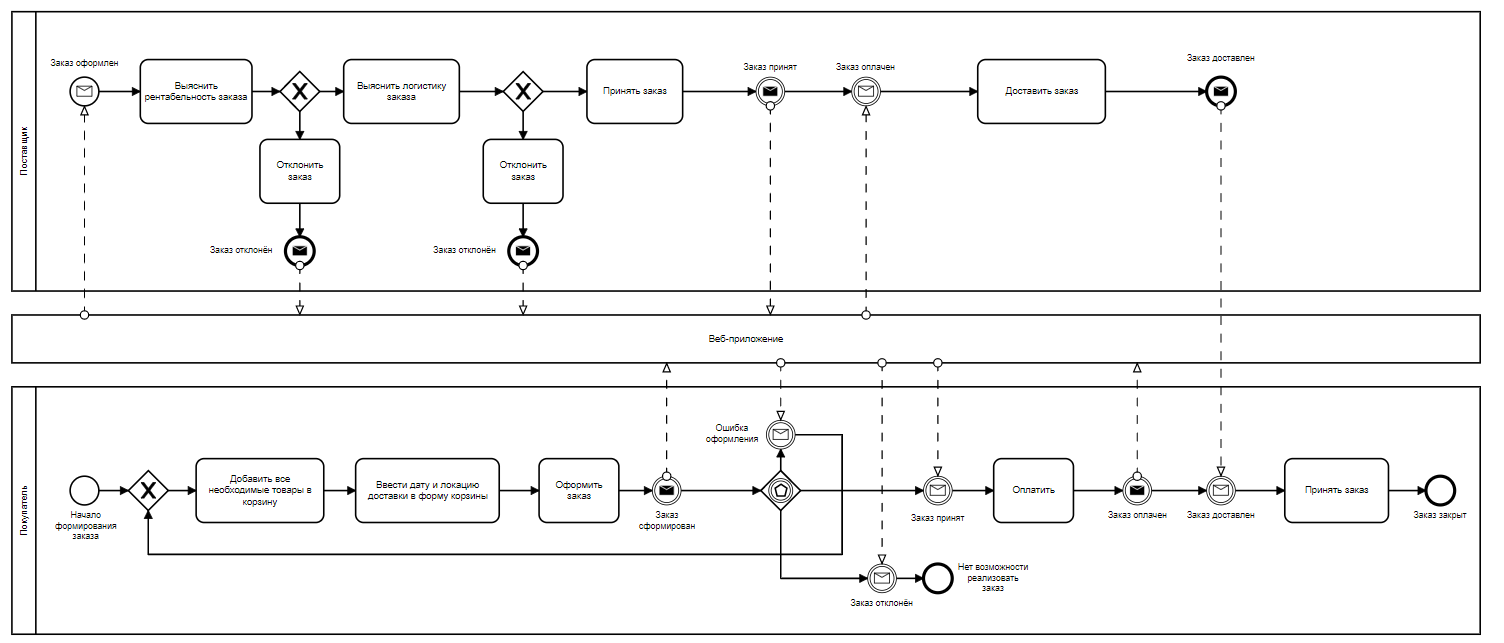


На диаграмме описан жизненный цикл заказа и его возможные состояния:

1. Заказ сформирован
2. Заказ оформлен
3. Заказ подтвержден
4. Заказ оплачен
5. Заказ отклонён

В этом жизненном цикле пользователь изменяет корзину, выбирая и удаляя товары, таким образом оформляя заказ. Далее поставщики принимают или отклоняют заказ. Если заказ принят всеми поставщиками, пользователь оплачивает заказ и ожидает доставки в указанную дату. Иначе ему будет необходимо переоформить заказ.

# Схема бизнес-процесса



На данной схеме описан бизнес-процесс оформления заказа пользователем и его обработка поставщиком.

# АРХИТЕКТУРА

# Диаграмма развертывания

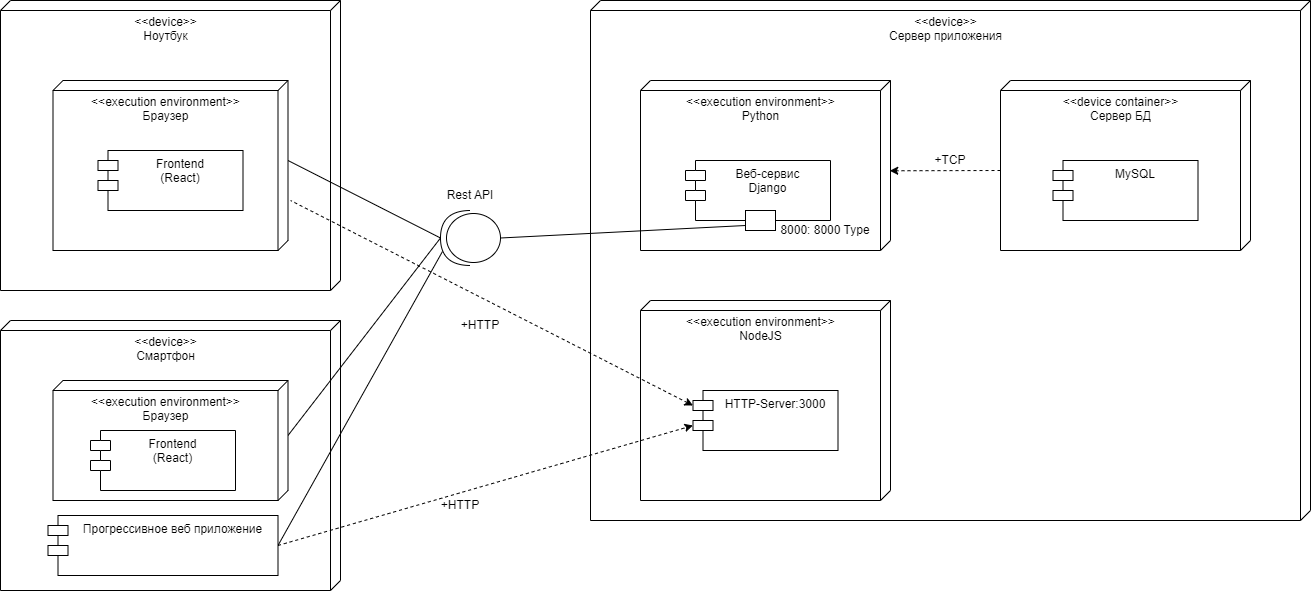
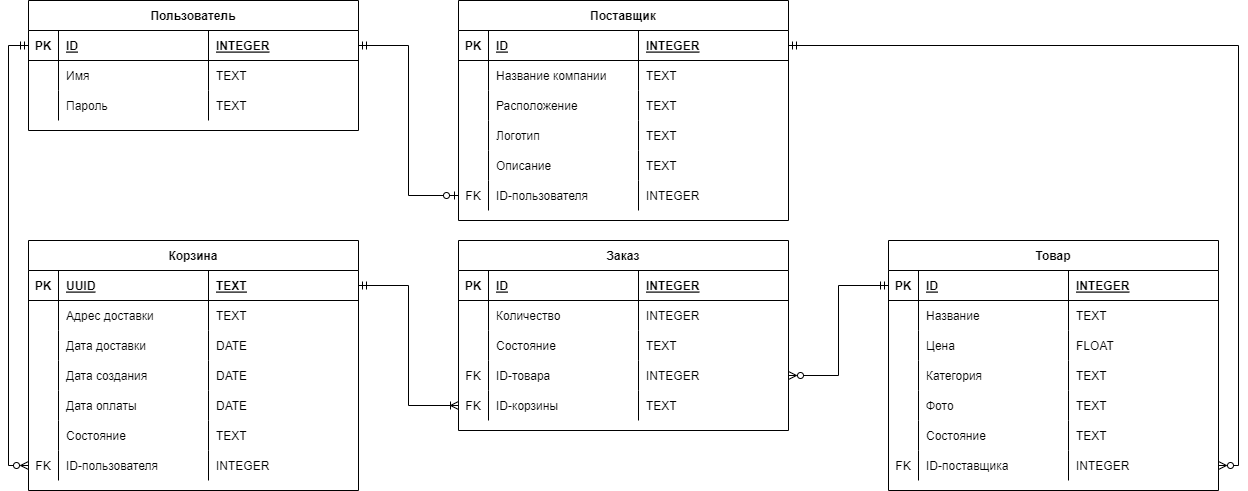


Диаграмма развертывания описывает общую структуру приложения, связь между сервером приложения и базой данных, устройствами.

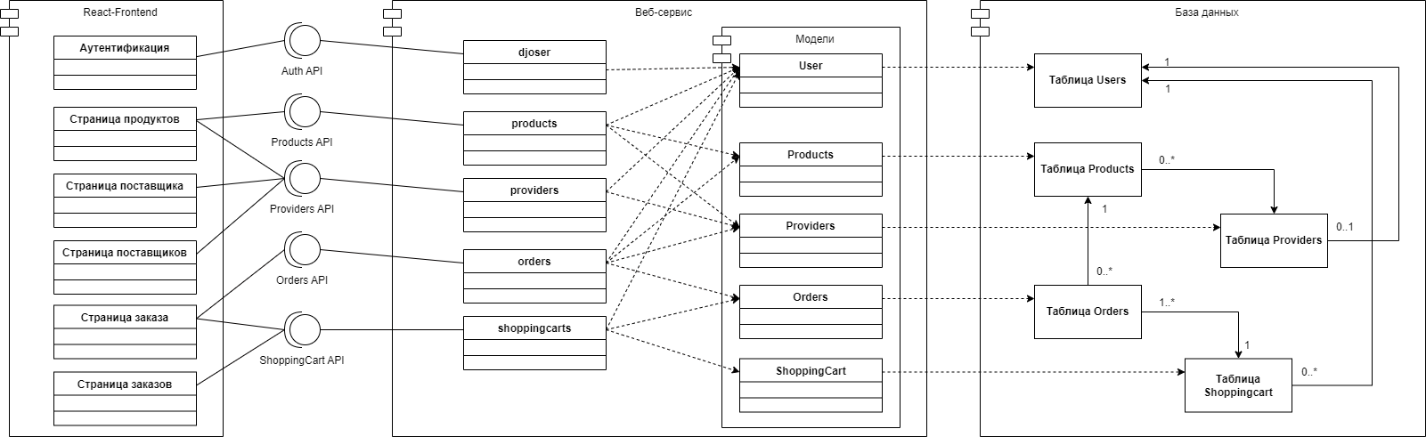
Для хранения базы данных используется MySQL, для реализации бэкенда (модели и представление данных базы в виде JSON-формата) используется Django (данные предоставляются на порте 8000). Для реализации фронтенда используется NodeJS (для браузерной версии приложения используется порт 3000). Для реализации приложения в браузере используются средства PWA. Для взаимодействия сайта (и мобильного приложения) с сервером используется Rest API. Сайт можно открыть в Chrome и других браузерах.

# ER с назначением таблиц



На данной схеме описаны сущности базы данных и связи между ними. Всего в схеме 5 сущностей: Пользователь, Поставщик, Корзина, Заказ, Товар.

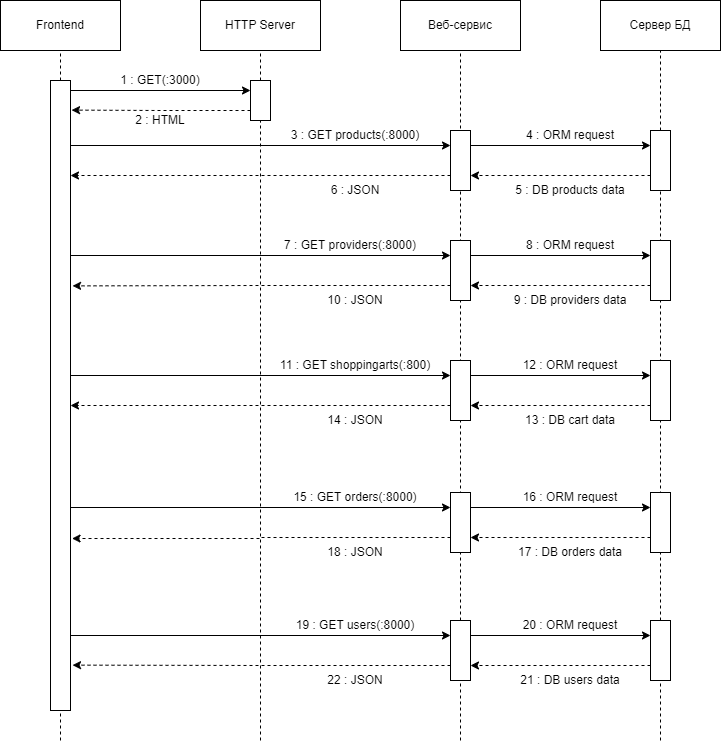
# Диаграмма компонентов



На диаграмме компонентов отражено взаимодействие между идентичными объектами веб-сервиса и фронтенда, а также связь объектов веб-сервиса с базой данных.

# АЛГОРИТМЫ

# Диаграмма последовательности

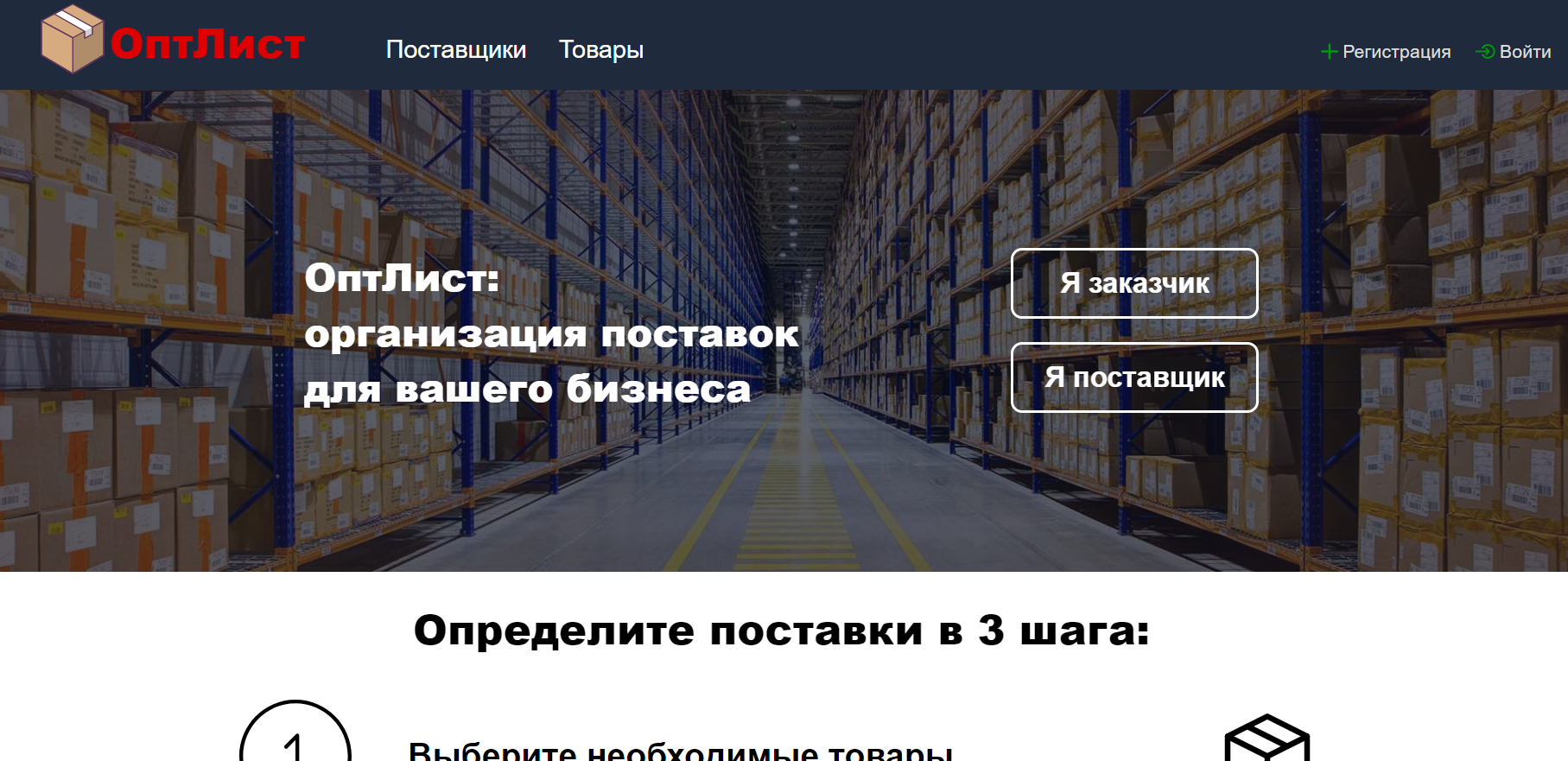


На диаграмме последовательности описано взаимодействие объектов приложения и их жизненный цикл на единой временной оси. Все данные на страницах получены с помощью обращения к API через AJAX-запросы, а API в свою очередь берёт данные из базы данных с помощью запросов ORM.

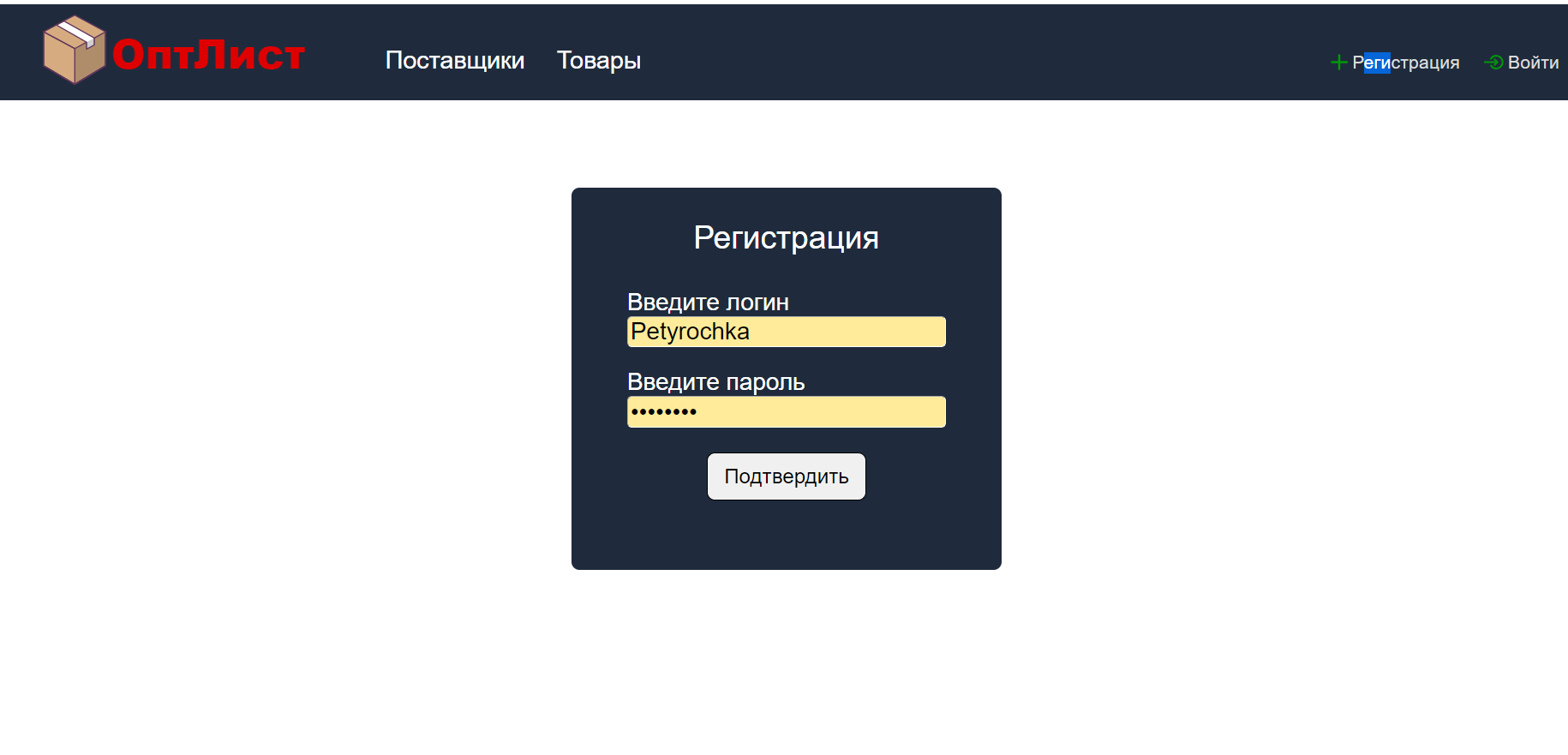
# ОПИСАНИЕ ИНТЕРФЕЙСА

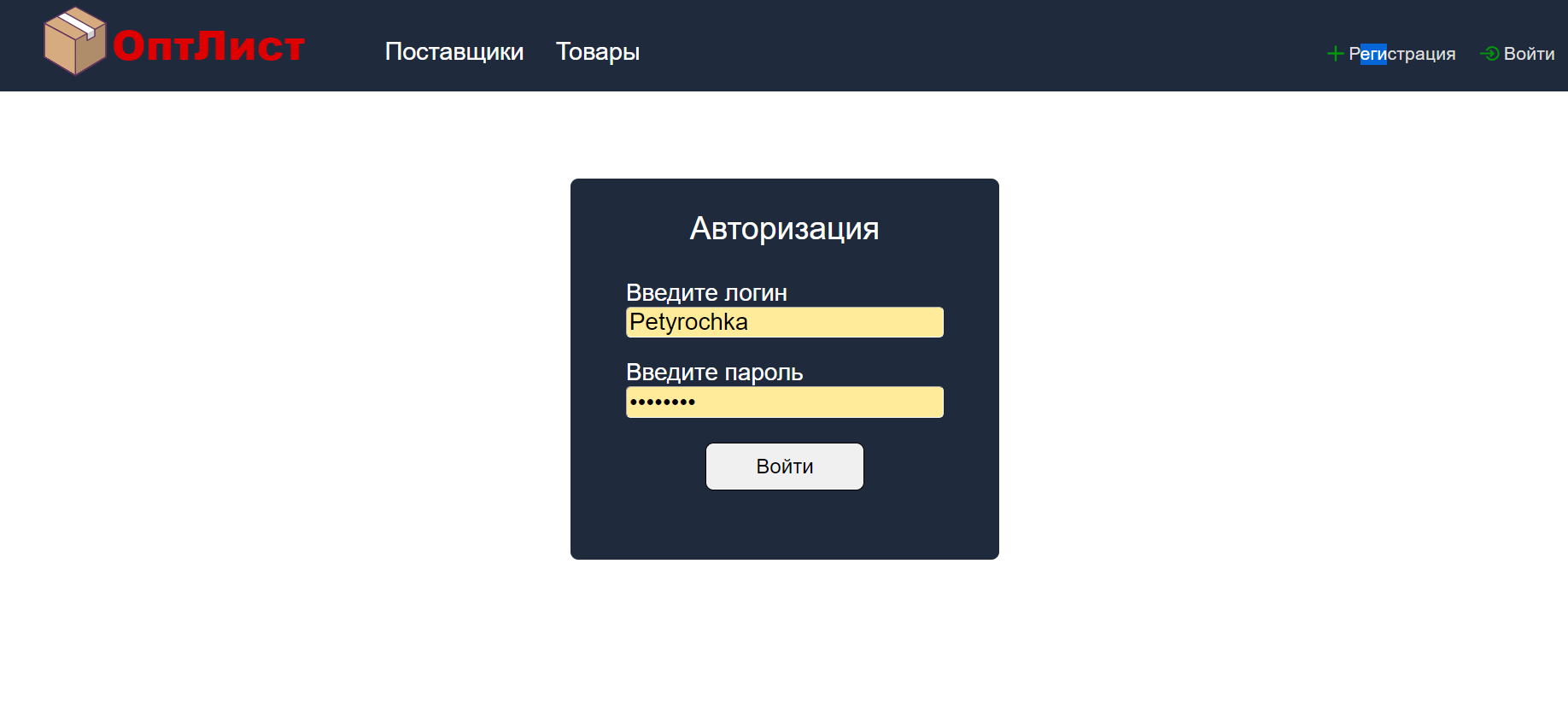
# Гость

При первичном посещении сайта пользователя встречает главная страница. На главной странице помимо некоторых декоративных навигационных панелей есть 2 тематические кнопки “Я заказчик” и “Я поставщик”. Первая перенаправляет пользователя в случае, если он не авторизован, на обычную регистрацию. Вторая также на регистрацию, однако по её прохождении, пользователь также будет сразу направлен на форму регистрации поставщика. Если пользователь авторизован, то он будет направлен в профиль пользователя или поставщика соответственно.

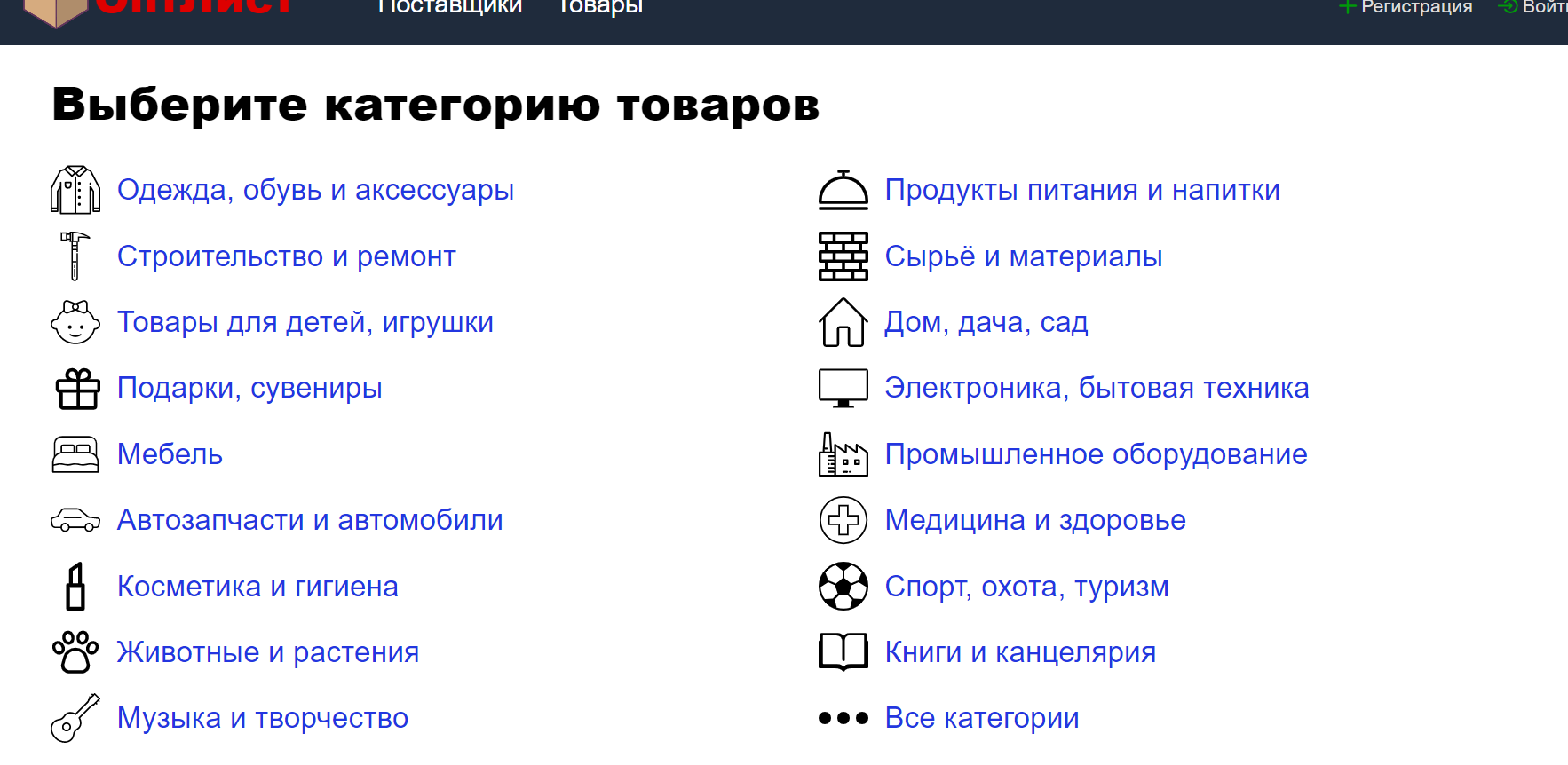


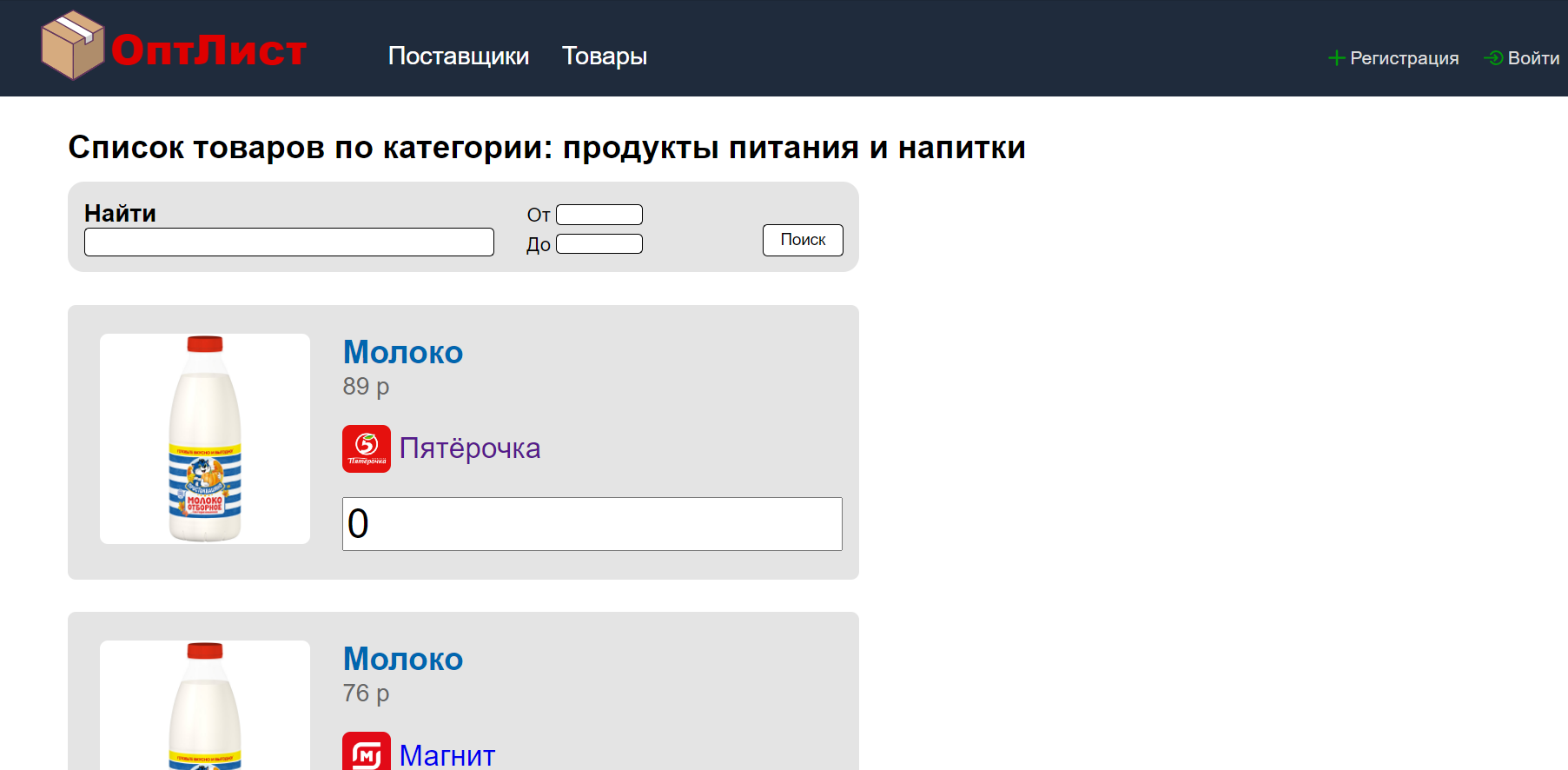
Гостю доступны авторизация и регистрация:



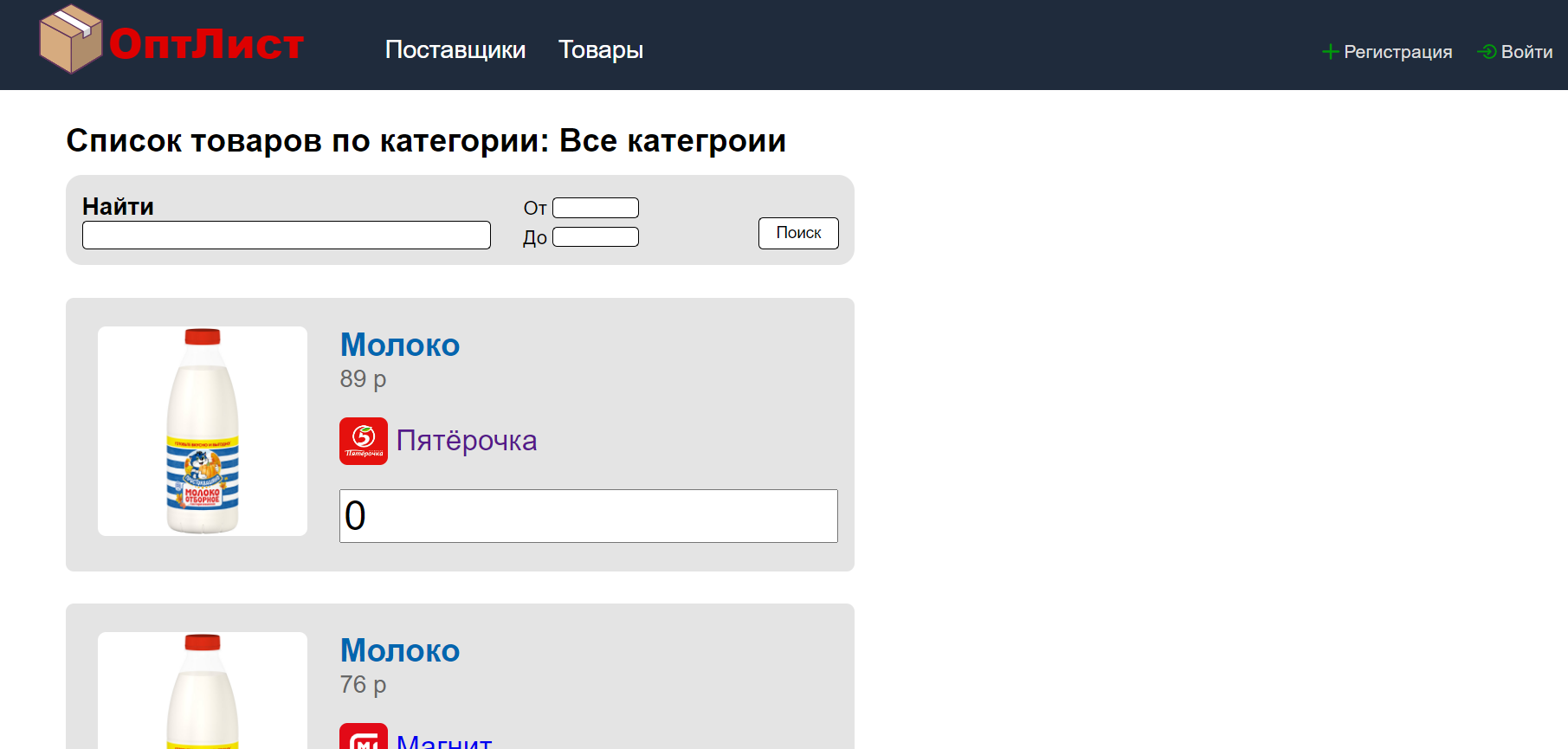


Гость способен просматривать товар. Прежде ему предлагается выбрать необходимую категорию товаров:

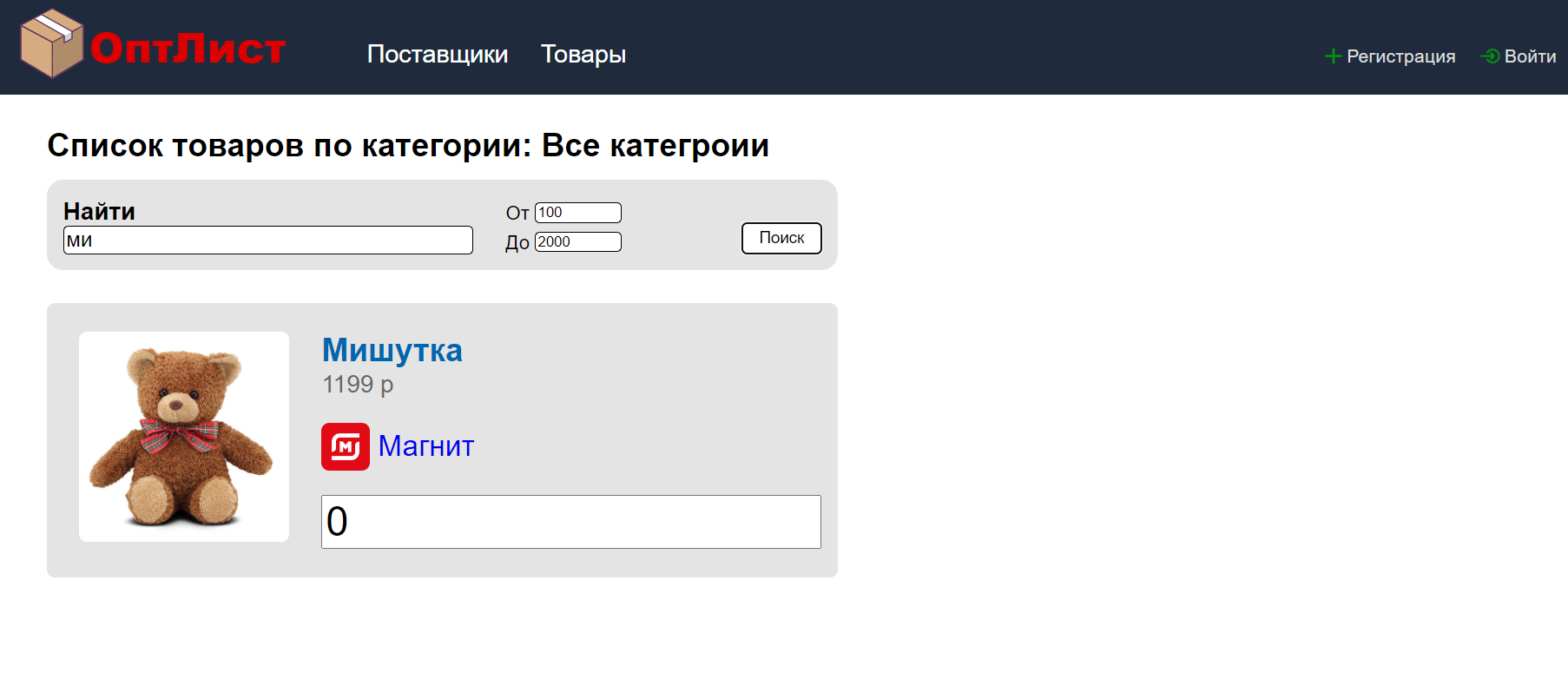


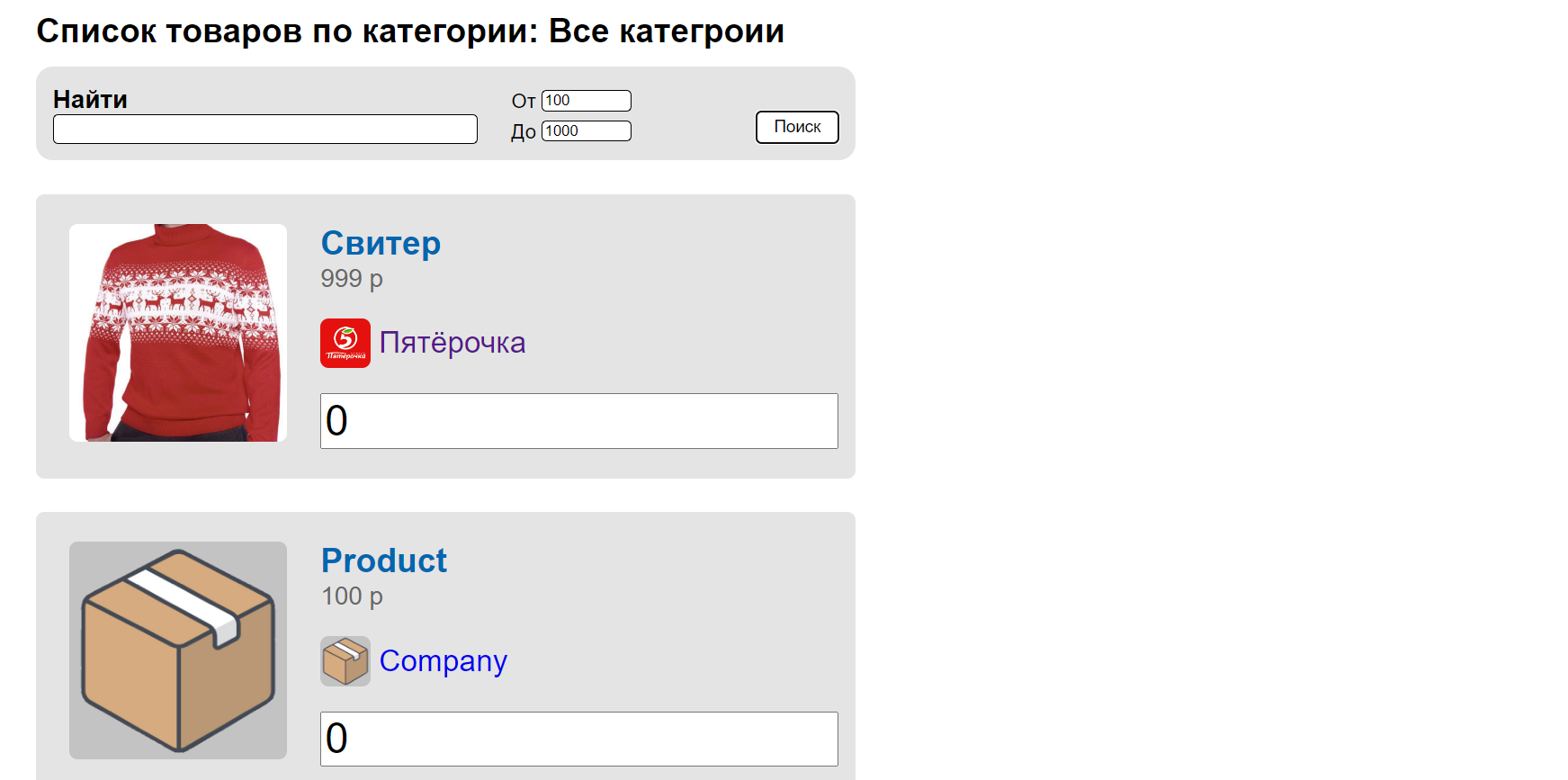


Пользователь может просматривать товары всех категорий:

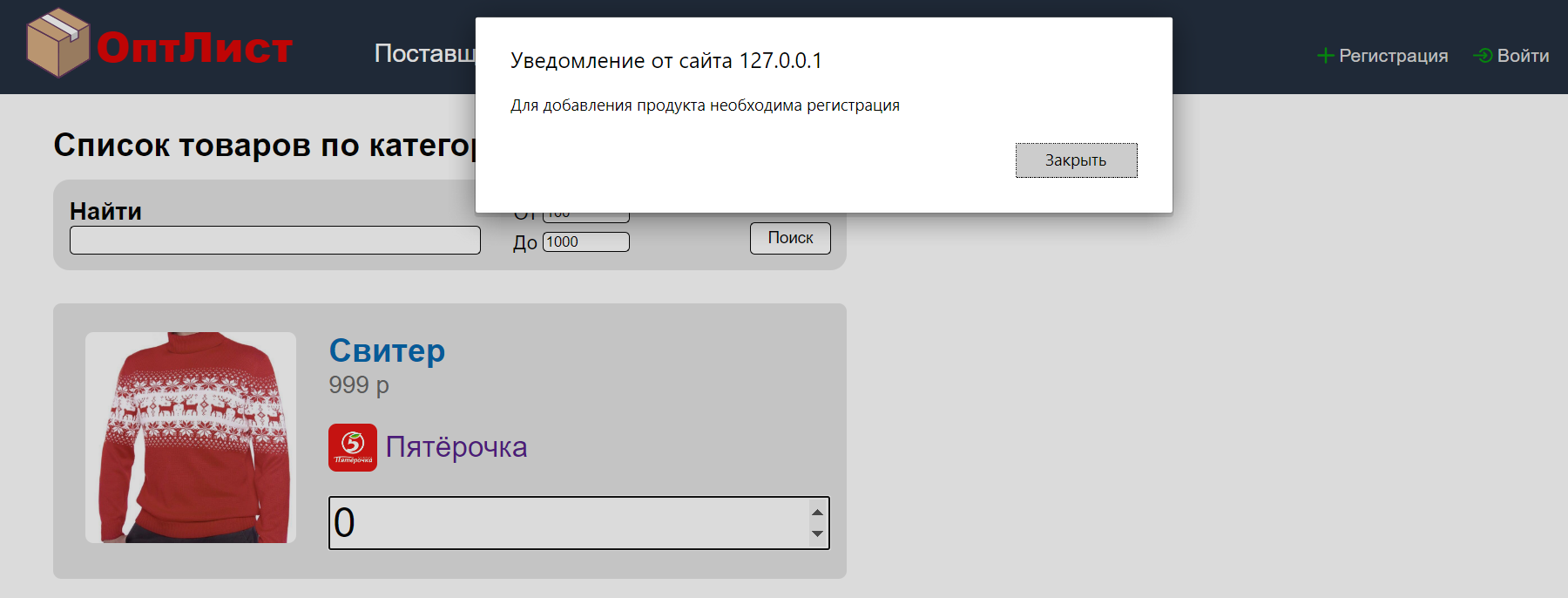


Пользователь способен фильтровать товары по стоимости и названию:

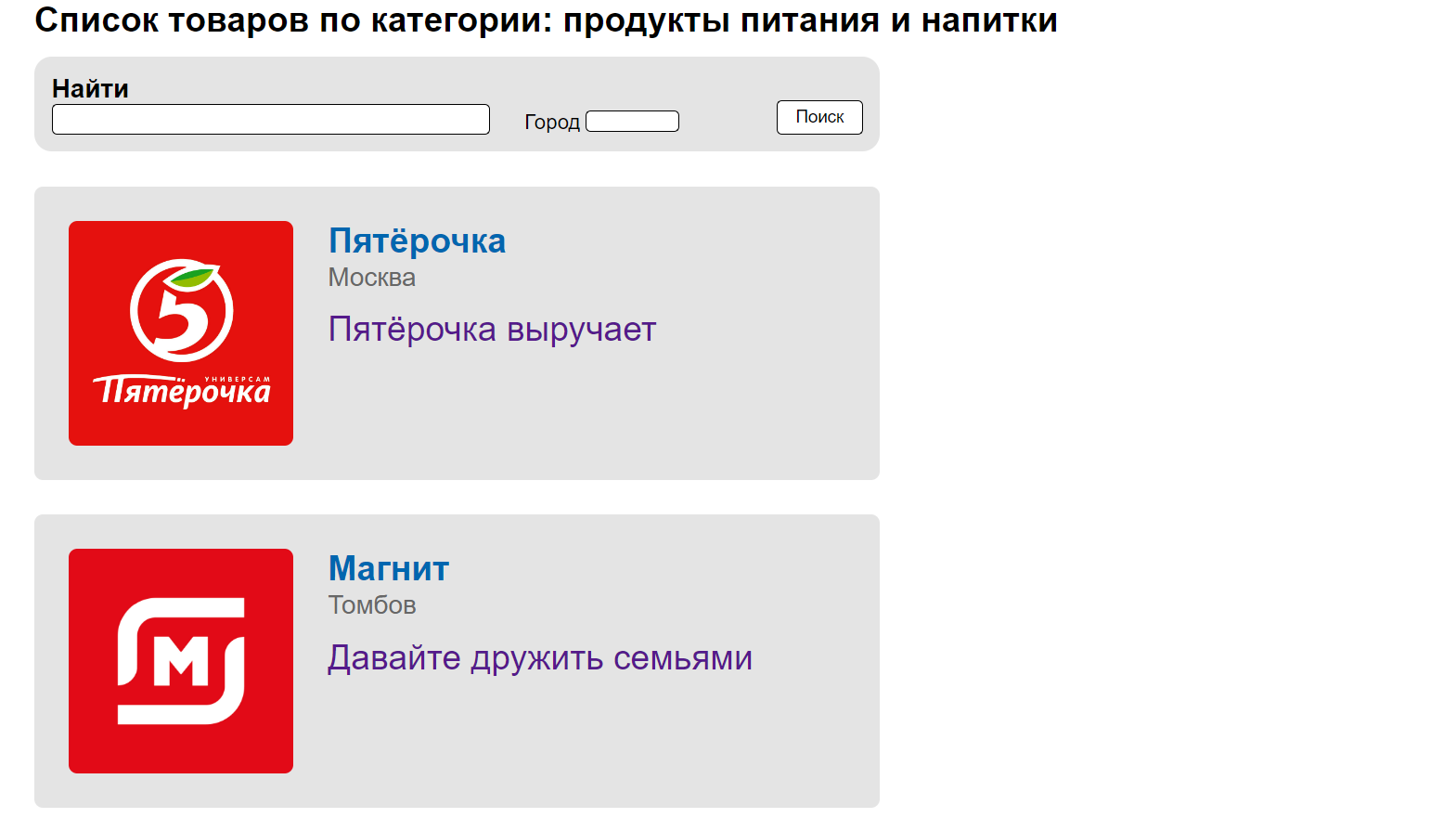




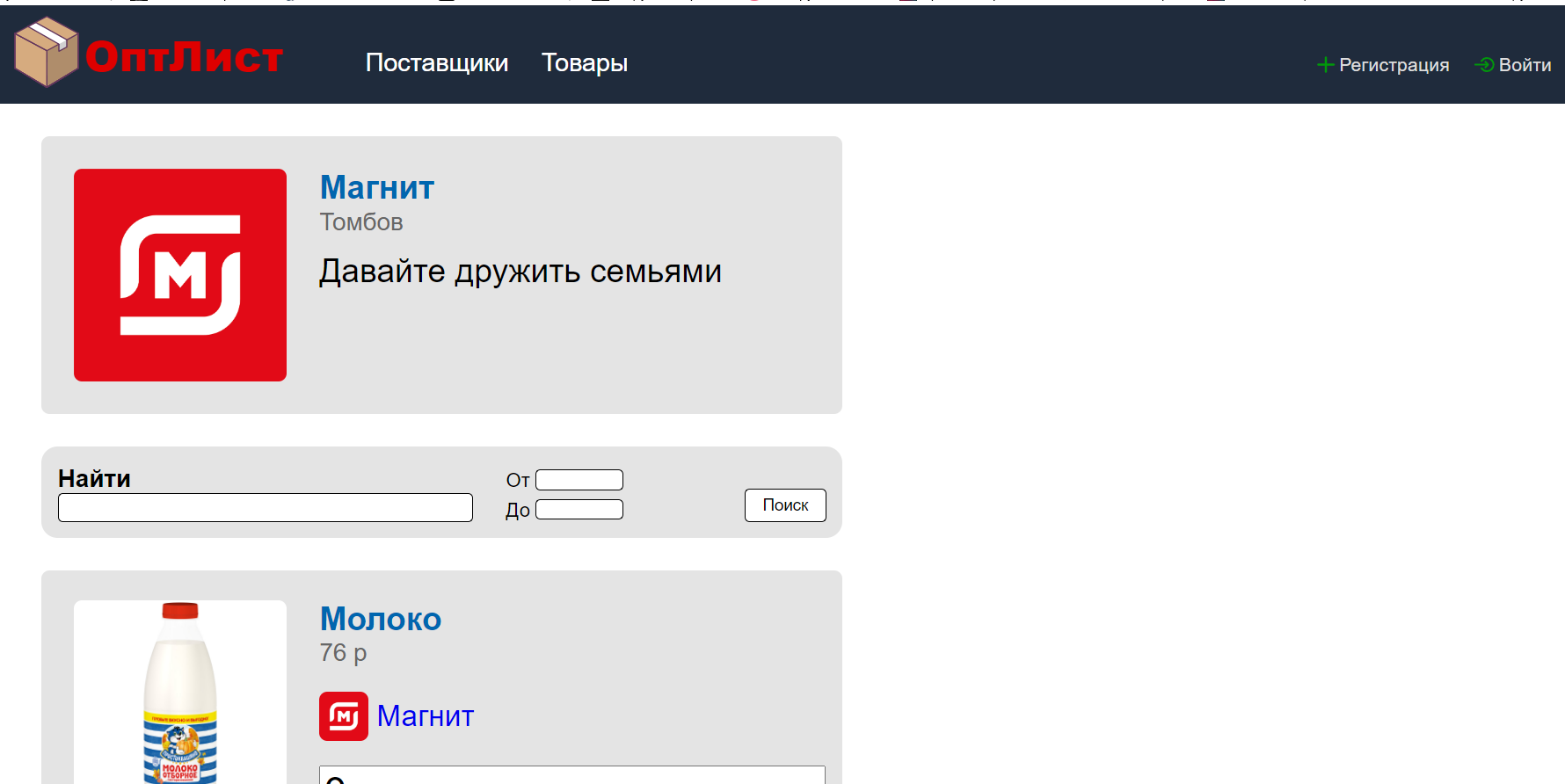
Если гость попытается добавить товар, то ему высветится уведомление о необходимость зарегистрироваться на сайте для разблокирования данного функционала:



По аналогии с товарами пользователь может просматривать поставщиков товаров:

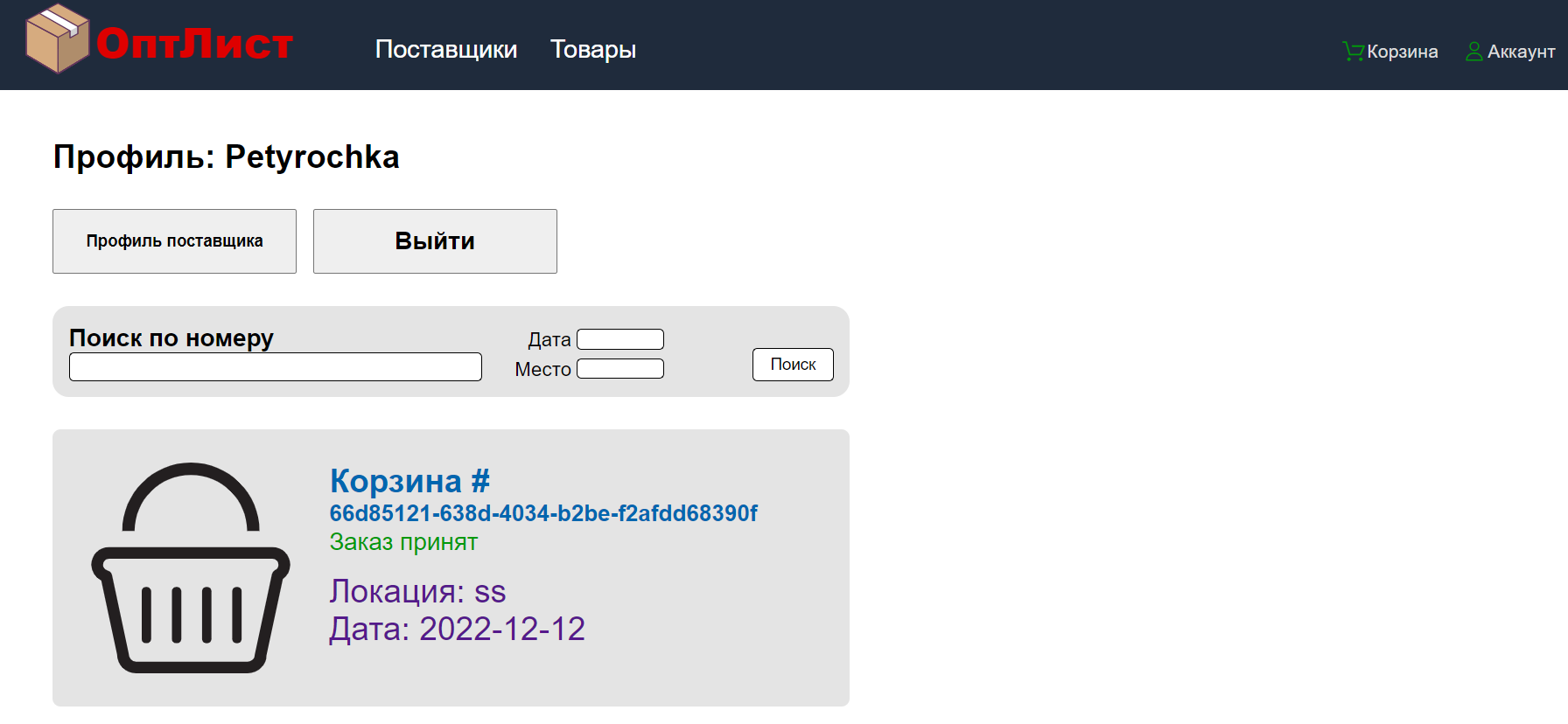


Нажав на карточку поставщика, гость перенаправляется на сайт данного страницу выбранного поставщика, где отображаются товары выбранной категории:

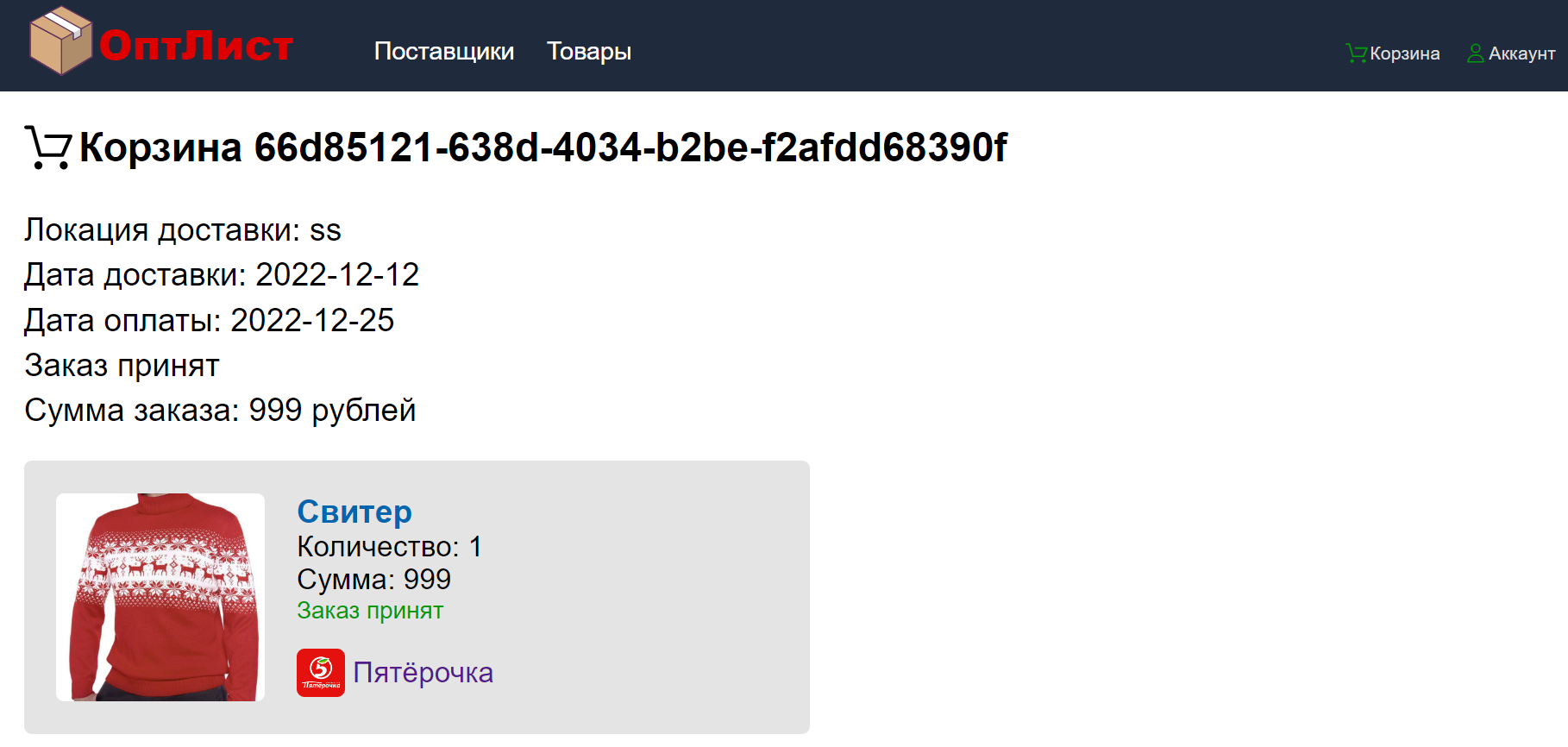


# Пользователь

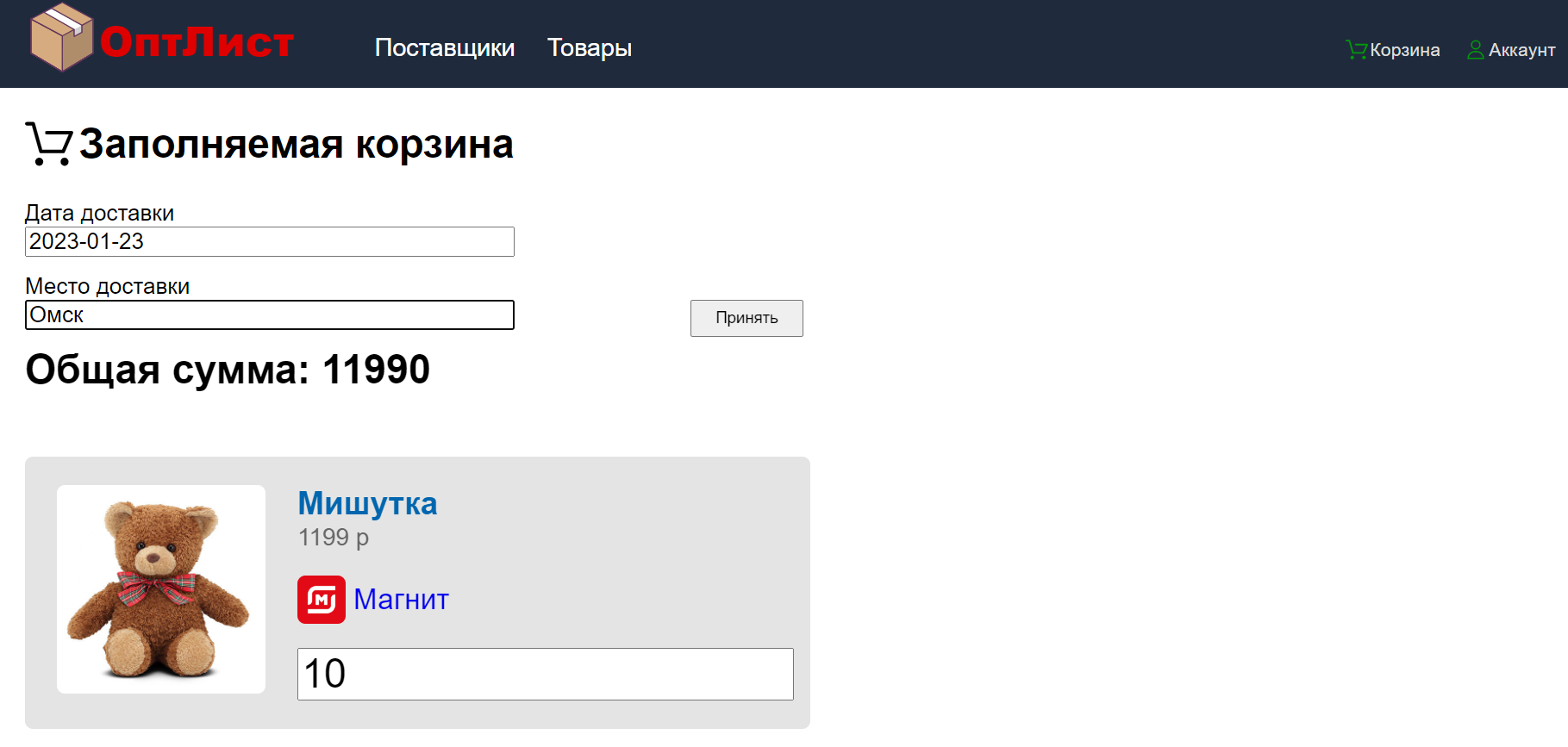
Страница авторизованного пользователя. Здесь отображаются оформленные заказы, которые также имеют возможность быть отфильтрованными.



Страница уже оформленного заказа выглядит так:

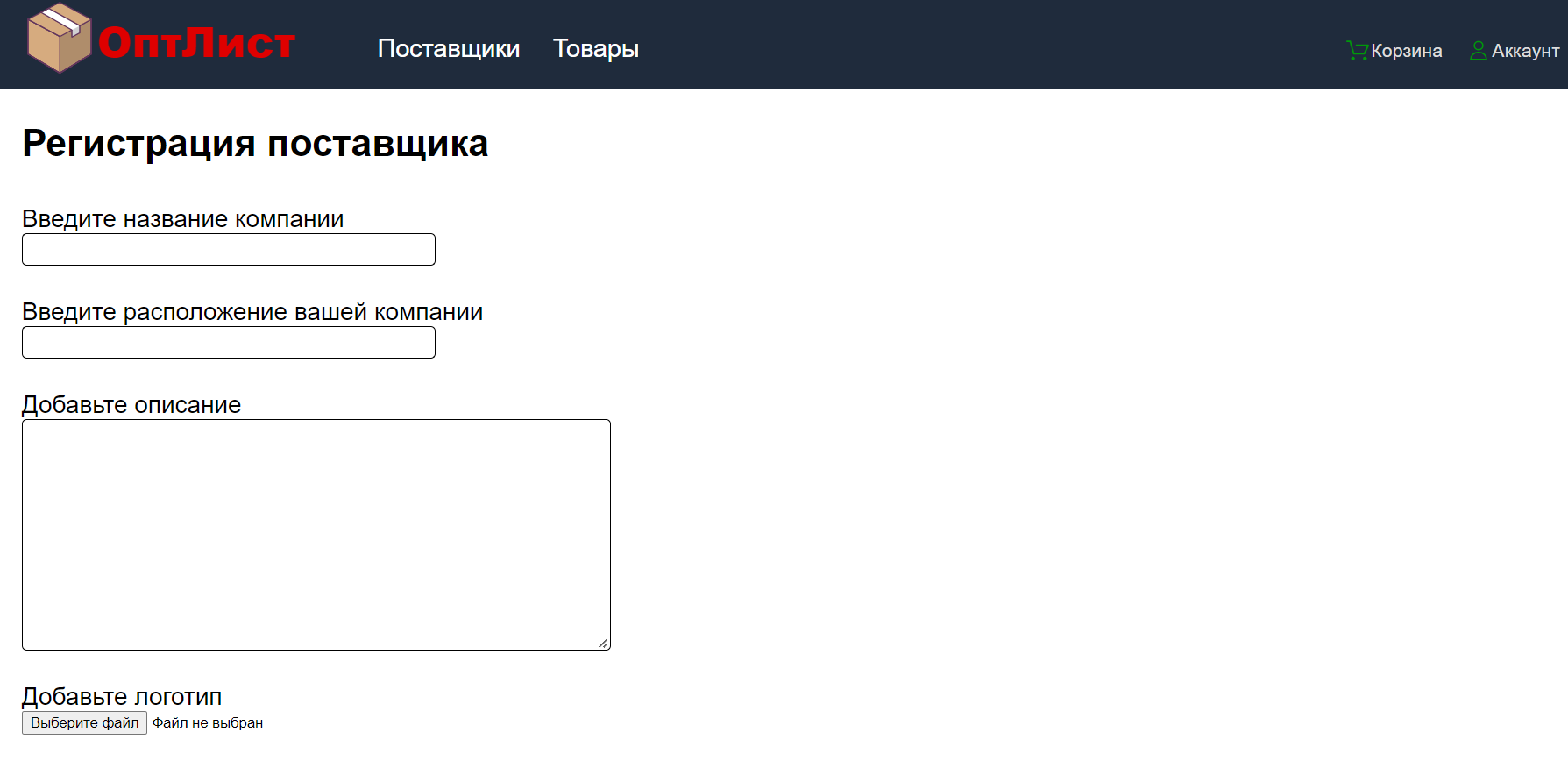


Также авторизованный пользователь способен формировать заказ в текущей корзине:

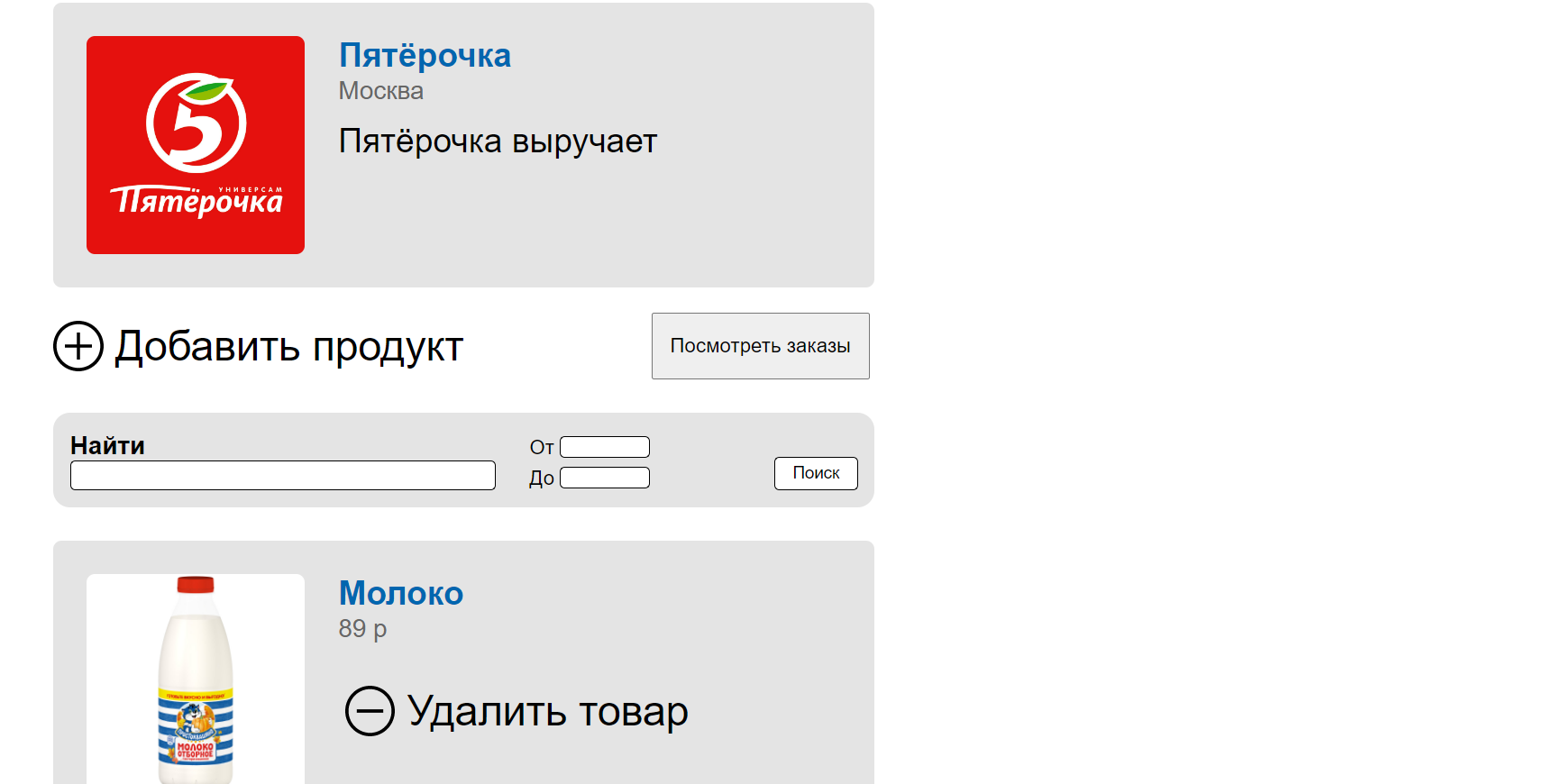


1. **Поставщик**

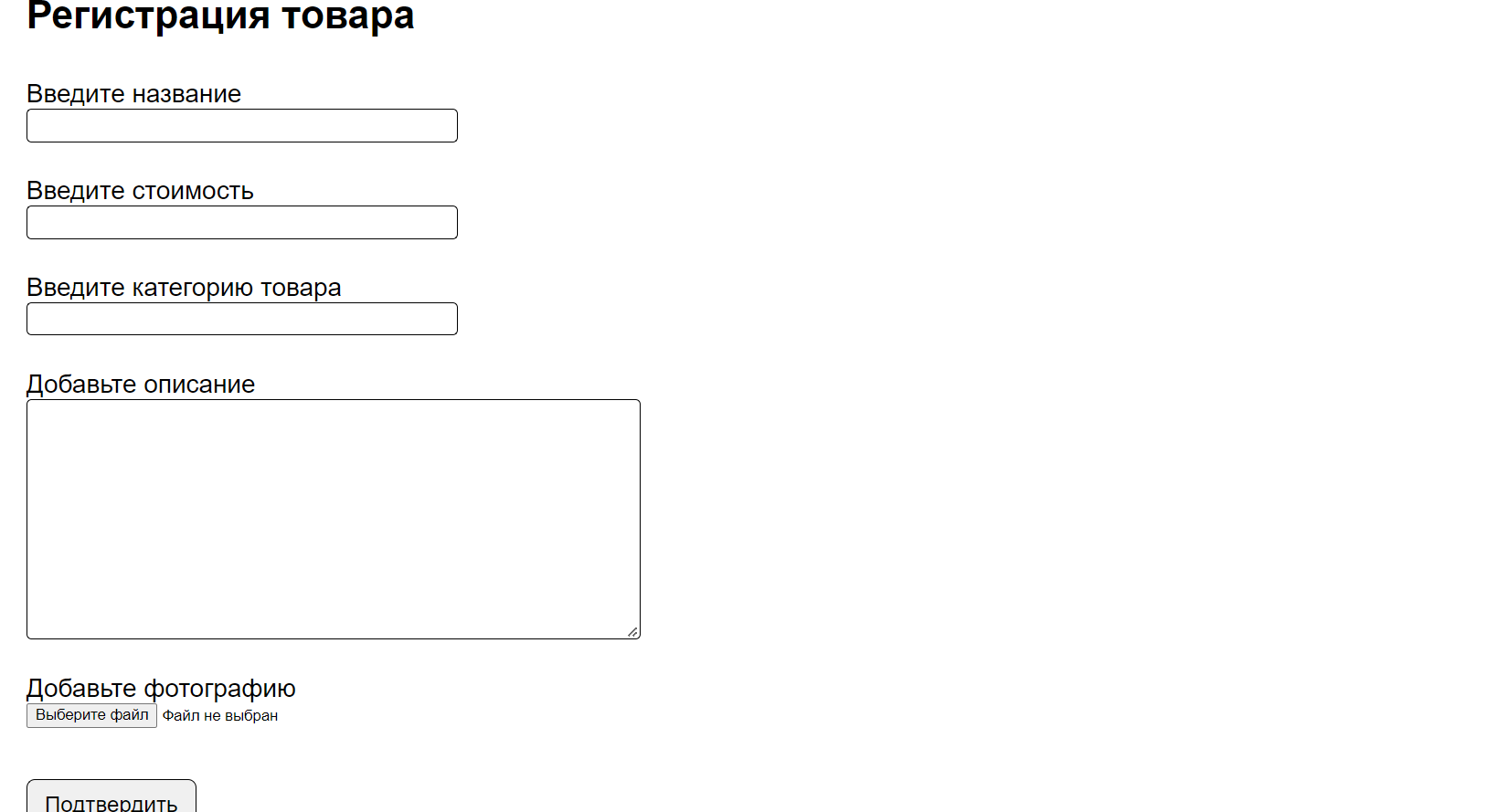
Авторизованный пользователь способен пройти авторизацию поставщика:



После прохождения авторизации пользователь будет перенаправлен на страницу поставщика, но с дополнительным функционалом, позволяющим редактировать выставленные товары:



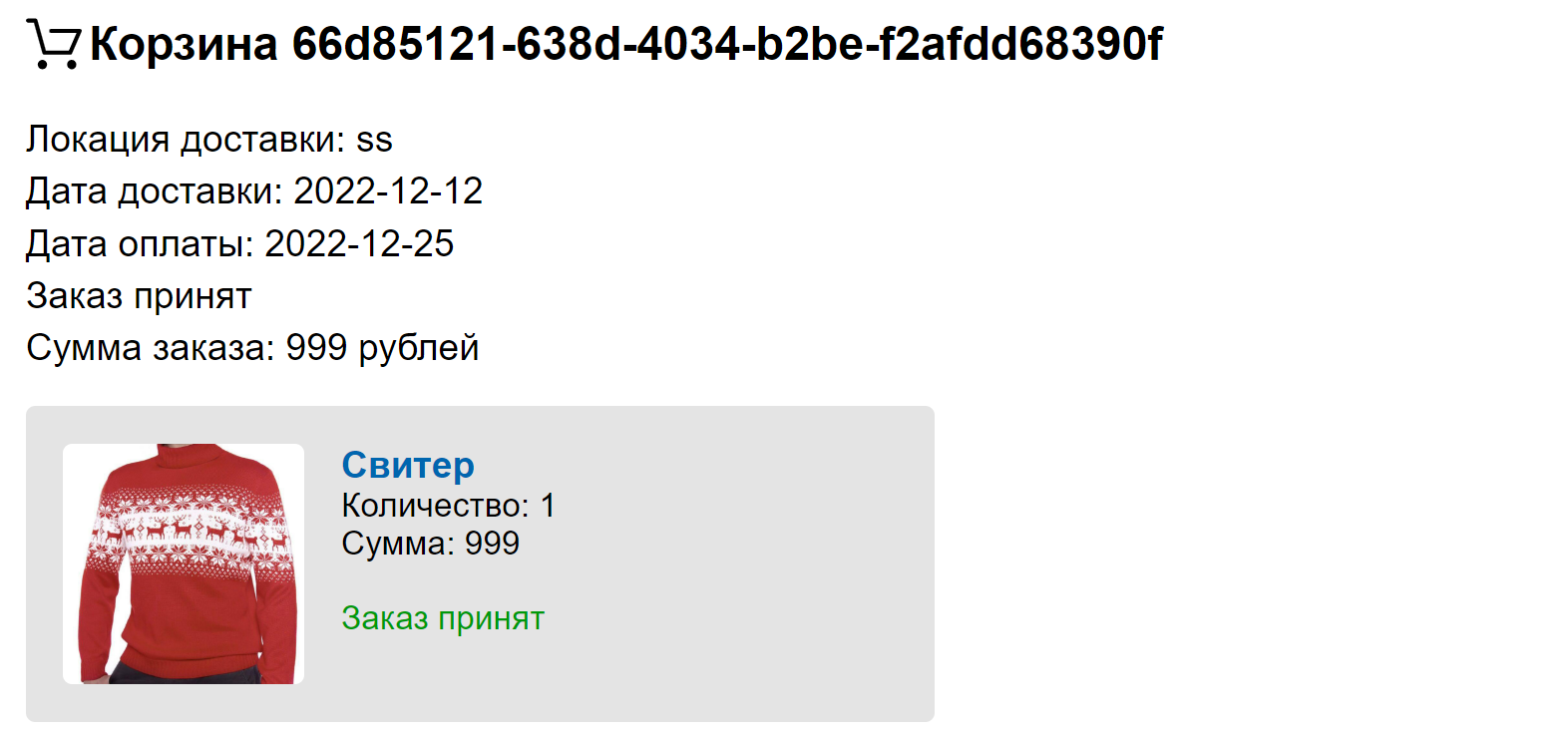
Здесь можно удалить и зарегистрировать продукт:

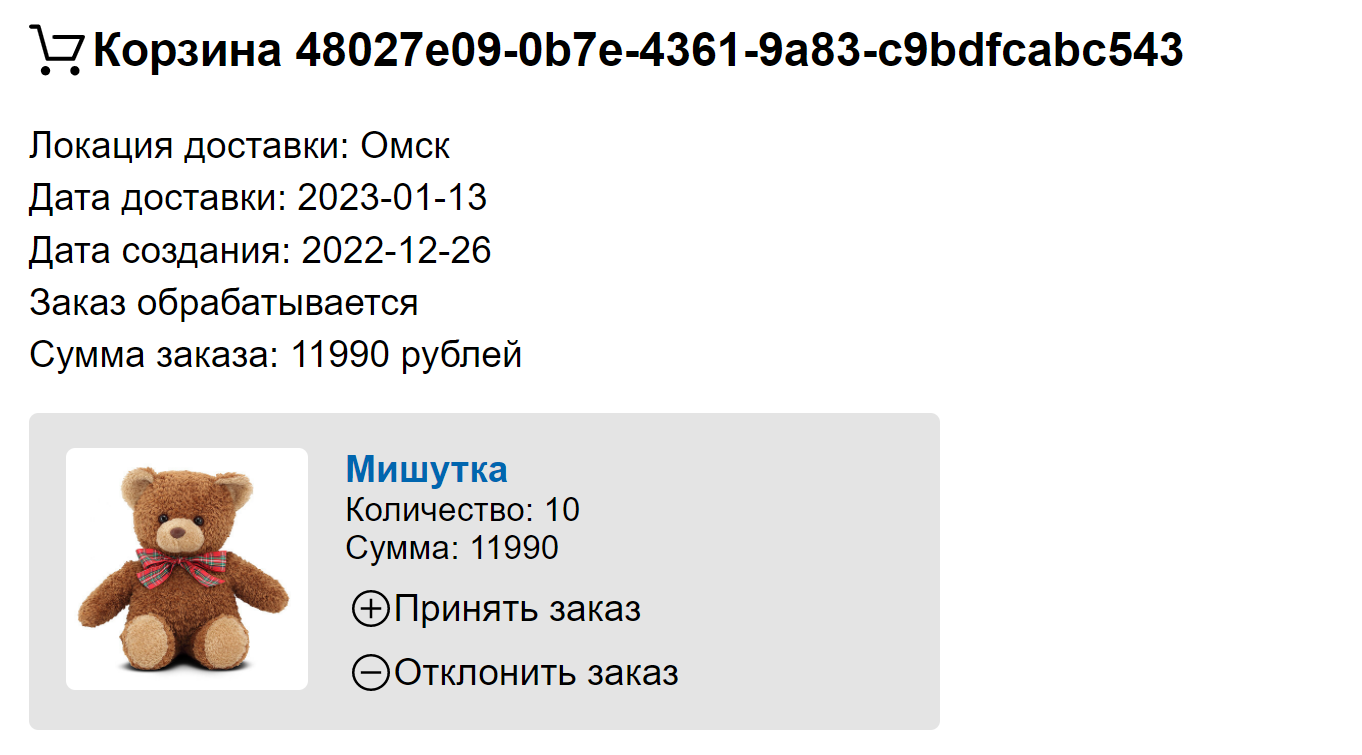


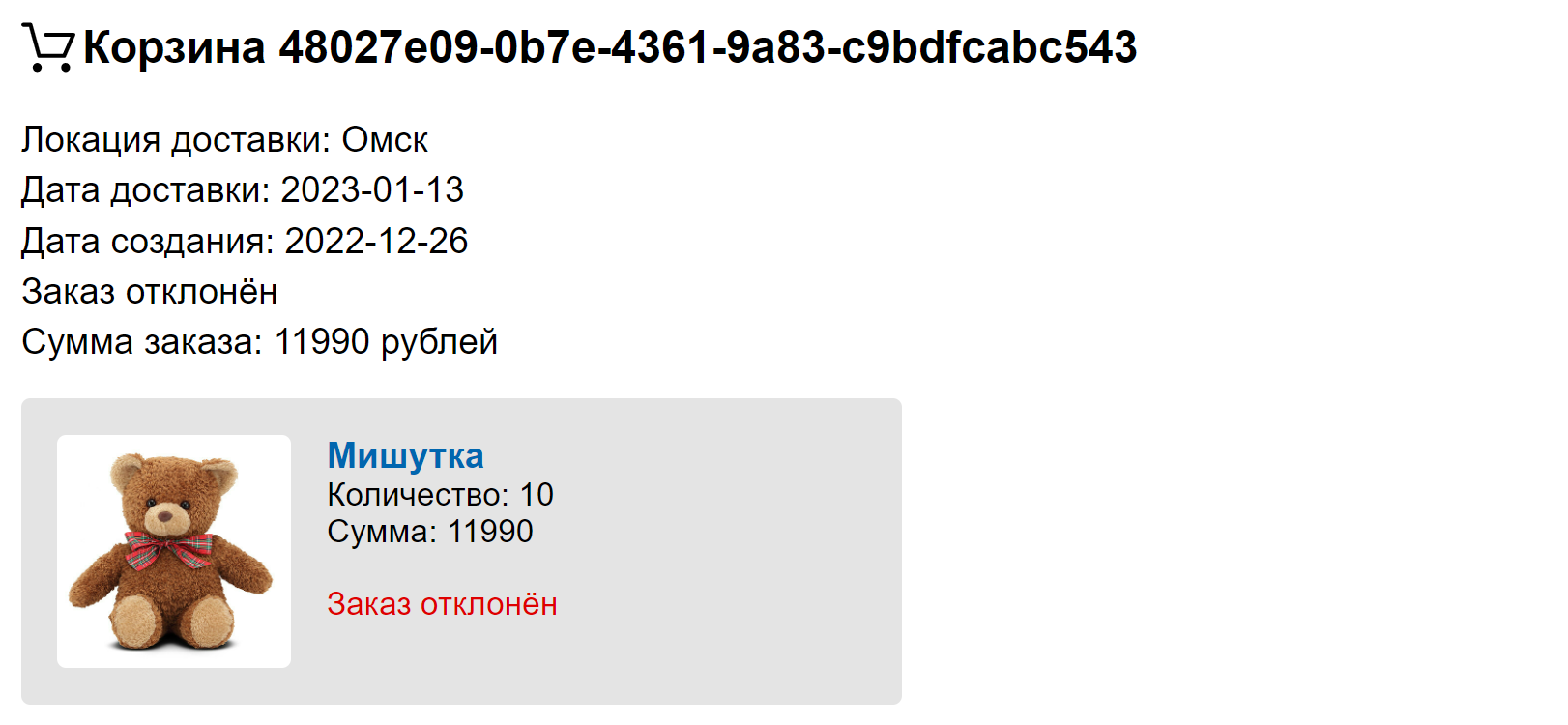
Также здесь можно посмотреть список корзин, содержащих заказанные товары:



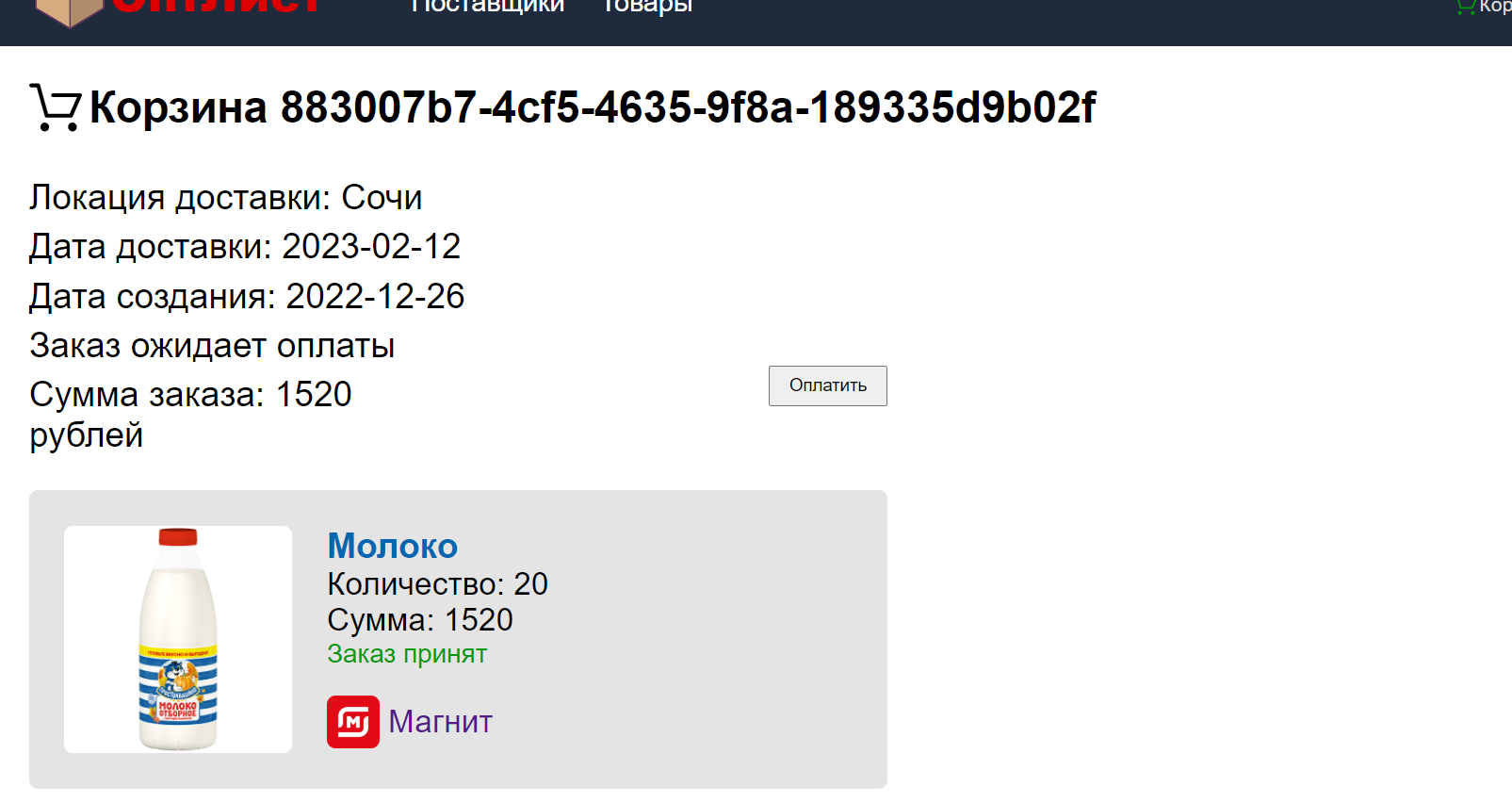
Можно просматривать данные корзины, а также изменять статус заказных товаров:







После утверждения легитимности заказа пользователь обязан его оплатить:



После оплаты заказ считается окончательно принятым и ожидает доставки в указанную дату:

