**Поволжский Государственный Университет Телекоммуникаций и Информатики**Кафедра «Программная инженерия»

|  |  |
| --- | --- |
| Сдана на проверку  «\_\_» \_\_\_\_2024 г. | Допустить к защите  «\_\_» \_\_\_\_\_ 2024 г.  Защищена с оценкой  «\_\_\_» \_\_\_\_\_\_\_\_ 2024 г. |

**КУРСОВАЯ РАБОТА**По дисциплине: «Прикладное программирование»  
На тему: «Микросервисная архитектура для e-commerce платформы»  
  
Пояснительная записка

|  |  |
| --- | --- |
| Студент группы ИВТ-22\_\_\_\_\_\_\_ Романов М.А  (роспись) (ФИО)  Руководитель \_\_\_\_\_\_\_\_ к.т.н., доц. Ахметшина Э. Г.  (роспись) (ФИО) | 0 220085 0  (№ зачетной книжки) |

Самара 2024

Рецензия

Введение

Современные технологии и стремительное развитие электронной коммерции (e-commerce) предъявляют высокие требования к архитектуре веб-приложений. Для обеспечения гибкости, масштабируемости и надёжности, разработчики всё чаще обращаются к микросервисной архитектуре, которая предоставляет уникальные возможности для решения задач электронной коммерции.

Микросервисная архитектура представляет собой подход к разработке программного обеспечения, при котором приложение разделяется на множество мелких сервисов, каждый из которых выполняет свою конкретную задачу и взаимодействует с другими сервисами через стандартизированные интерфейсы. Такой подход позволяет упростить управление системой, повысить её отказоустойчивость и сократить время на внедрение новых функций.

Применение микросервисной архитектуры в контексте e-commerce платформы открывает широкий спектр возможностей. Каждый микросервис может быть ответственен за определённую область функционала, будь то управление товарами, обработка заказов, учёт платежей или работа с клиентскими данными. Это позволяет улучшить производительность системы, обеспечивая независимость и параллельность выполнения операций.

Целью данной работы является исследование особенностей применения микросервисной архитектуры в e-commerce платформах, анализ её преимуществ и недостатков, а также разработка рекомендаций по внедрению данной архитектуры для повышения эффективности работы системы. В процессе работы будут рассмотрены основные принципы проектирования микросервисов, используемые технологии и инструменты, а также примеры успешных реализаций в реальных проектах.

Результаты исследования могут быть полезны разработчикам, архитекторам и компаниям, планирующим переход на микросервисную архитектуру или её использование для создания высоконагруженных и надёжных e-commerce систем.

Цель и задачи работы

Целью данной работы является обоснование преимуществ микросервисной архитектуры для платформ электронной коммерции (e-commerce). В рамках исследования будут рассмотрены ключевые аспекты, влияющие на эффективность e-commerce систем, такие как масштабируемость, надёжность, производительность и гибкость. Особое внимание будет уделено анализу влияния микросервисного подхода на возможности адаптации платформ к изменениям требований рынка и запросам пользователей.

Для достижения поставленной цели необходимо:

1. Изучить основные принципы микросервисной архитектуры и их применение в контексте e-commerce.
2. Выявить преимущества микросервисного подхода по сравнению с монолитной архитектурой для решения задач электронной коммерции.
3. Рассмотреть реальные примеры использования микросервисов в известных e-commerce платформах.
4. Сверстать сайт интернет магазина, для внедрения в него микросервисной архитектуры.
5. Разработать рекомендации по внедрению микросервисной архитектуры для повышения эффективности и конкурентоспособности e-commerce систем.

Описание методов и технологий

Для реализации платформы электронной коммерции на основе микросервисной архитектуры используются современные технологии и подходы, обеспечивающие высокую производительность, гибкость и удобство разработки. В рамках данной работы применяются следующие инструменты и технологии:

1. React: Популярная библиотека для разработки пользовательских интерфейсов. React обеспечивает модульный подход к созданию компонентов, что идеально сочетается с принципами микросервисной архитектуры.
2. Next.js: Фреймворк на базе React, предлагающий серверный рендеринг (SSR) и статическую генерацию (SSG). Эти возможности позволяют улучшить производительность и SEO, что критично для e-commerce платформ.
3. Tailwind CSS: Утилитарный CSS-фреймворк, обеспечивающий гибкость и высокую скорость стилизации. Tailwind позволяет легко создавать адаптивные интерфейсы, используя классы напрямую в разметке.
4. ShadCN: Современный инструмент для управления темами и стилями, позволяющий настраивать пользовательский интерфейс в соответствии с уникальными требованиями бренда. Он интегрируется с Tailwind для улучшения визуального опыта пользователей.
5. JWT-аутентификация: Технология JSON Web Token используется для авторизации пользователей. Этот подход обеспечивает безопасность и масштабируемость, позволяя эффективно управлять доступом к различным частям системы.
6. MySQL: Реляционная база данных, выбранная для хранения и управления данными. MySQL обеспечивает высокую производительность, отказоустойчивость и гибкость при работе с большими объемами данных.
7. REST API: Подход к взаимодействию между микросервисами и фронтендом. REST API позволяет стандартизировать обмен данными, упрощая интеграцию и масштабирование системы.
8. Docker: Для удобного разворачивания сайта на любом десктопном устройстве

Вышеуказанные технологии обеспечивают эффективное управление стилями и упрощают разработку адаптивных, интерактивных и высокопроизводительных интерфейсов. Их совместное использование позволяет создавать платформы, отвечающие высоким требованиям рынка электронной коммерции.

Реализация

Дизайн:

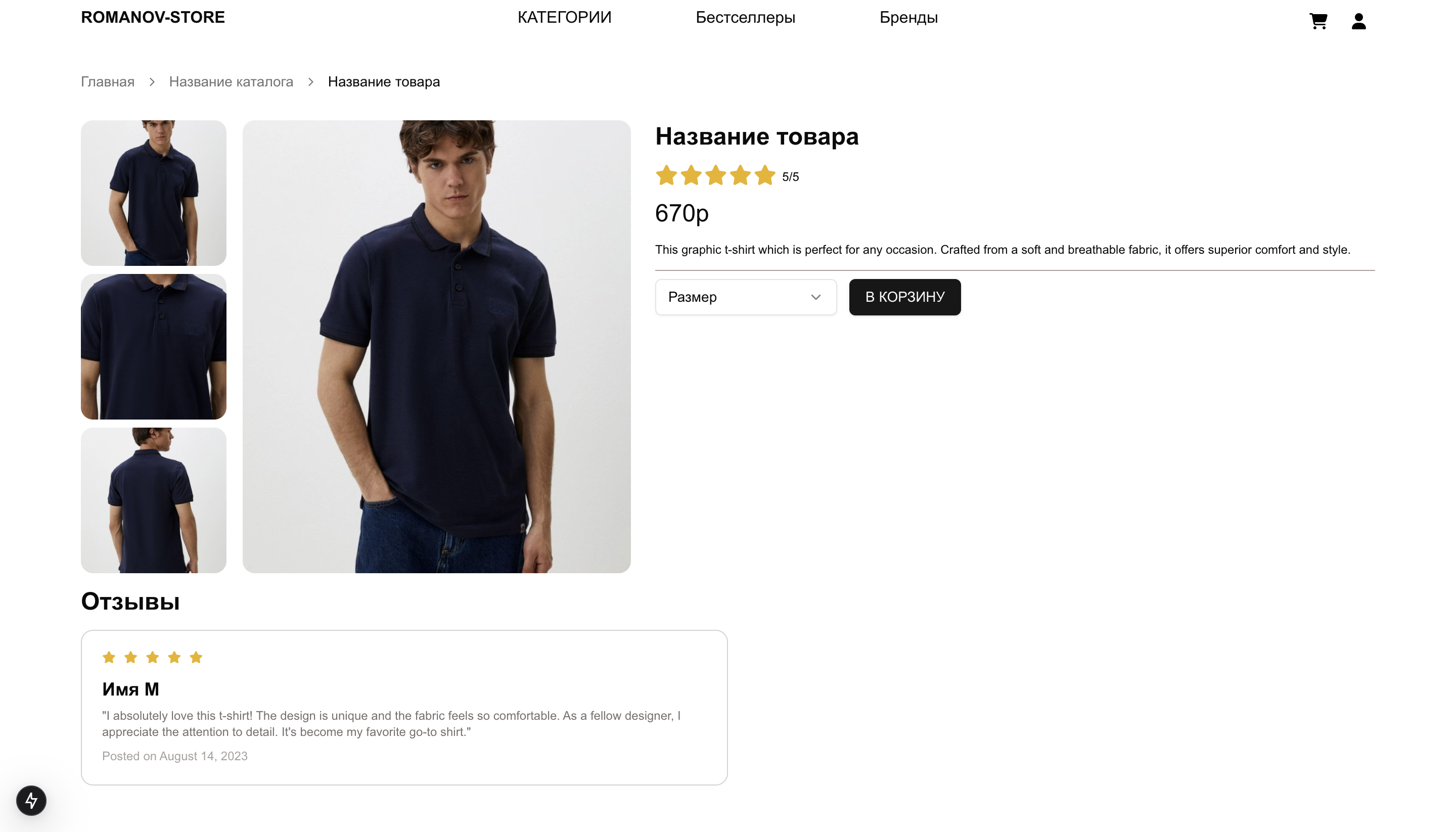


Рис 1 Страница товара

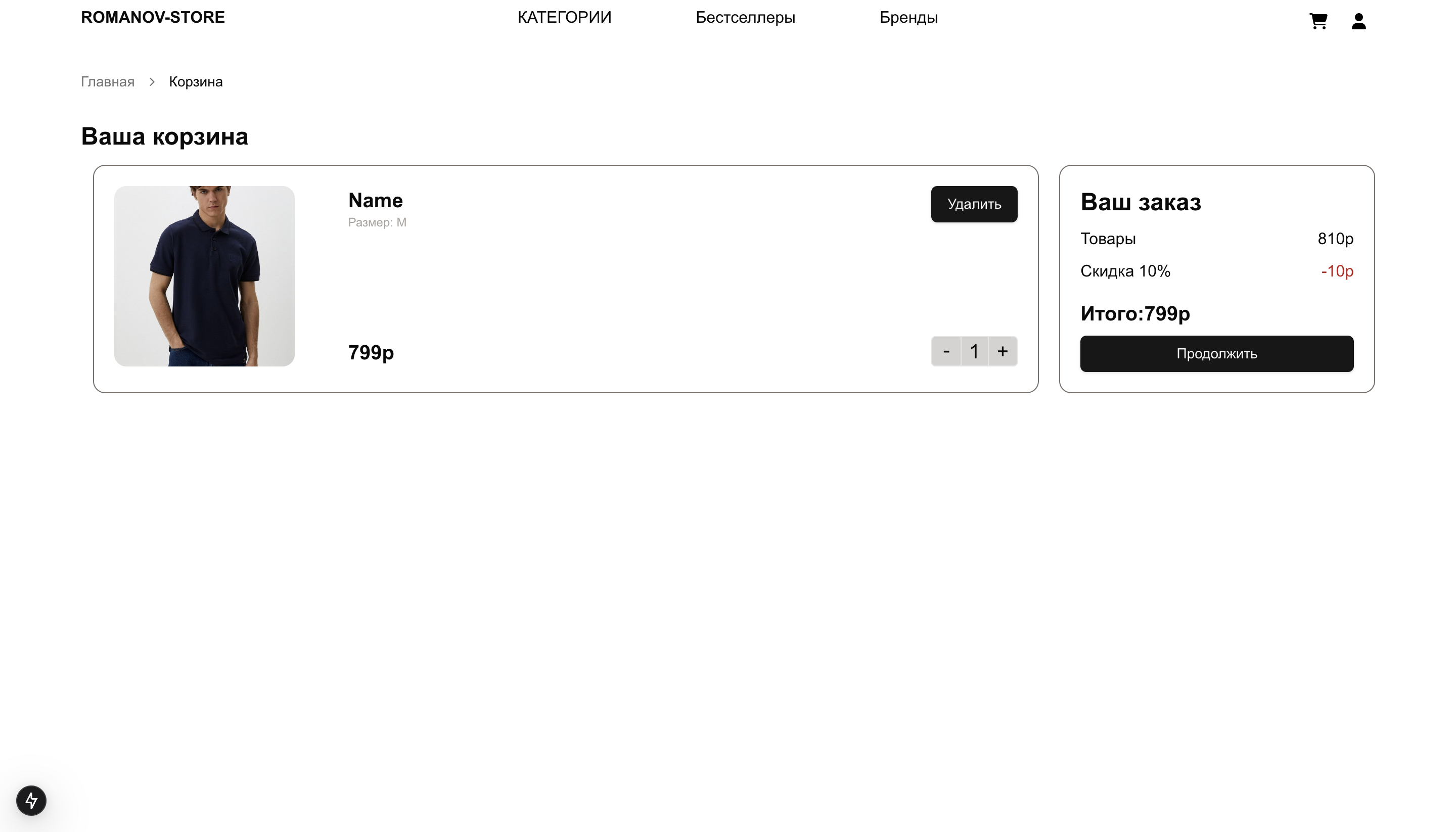


Рис 2 Корзина

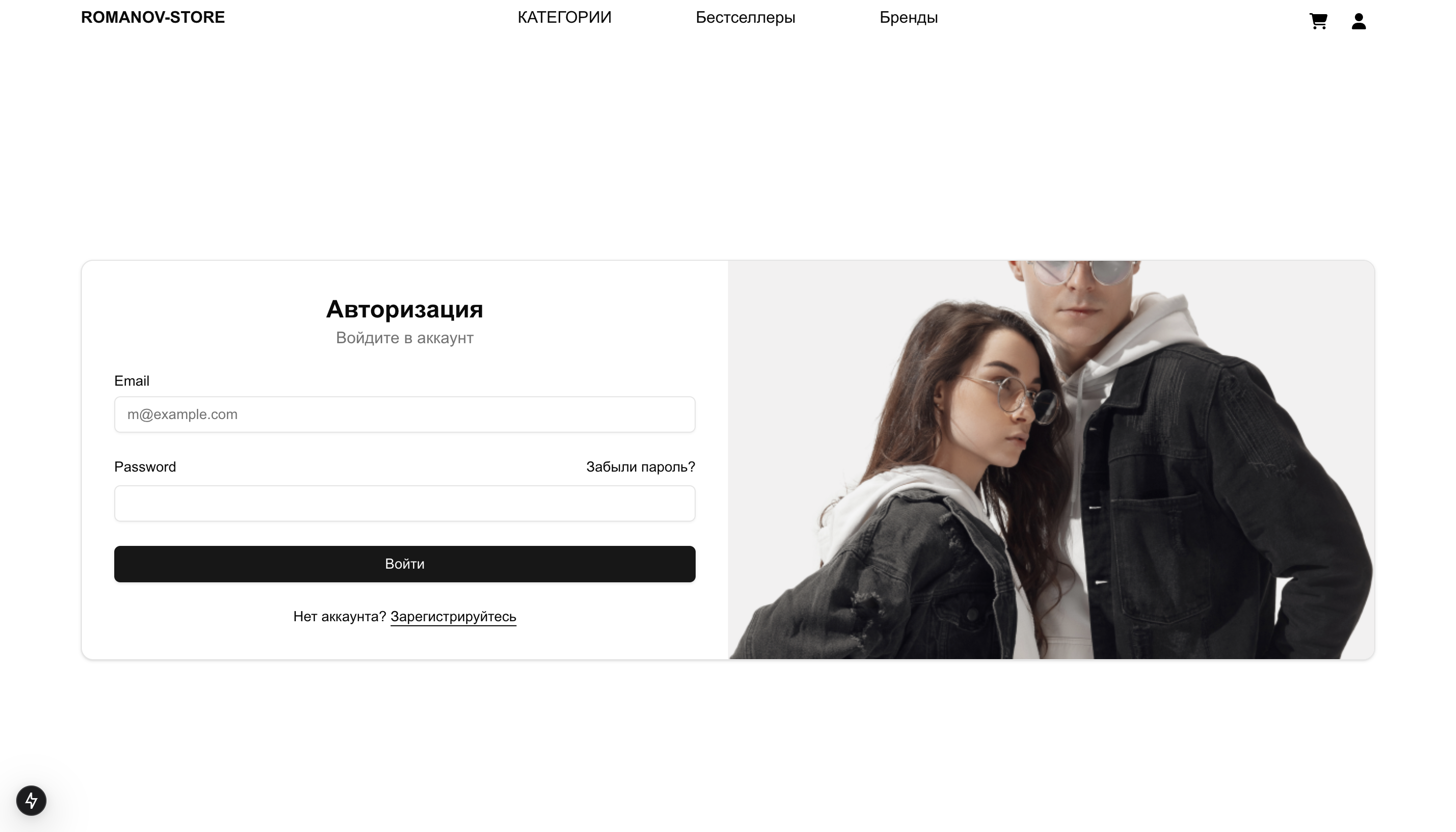


Рис 3 Авторизация Рис 4 Регистрация

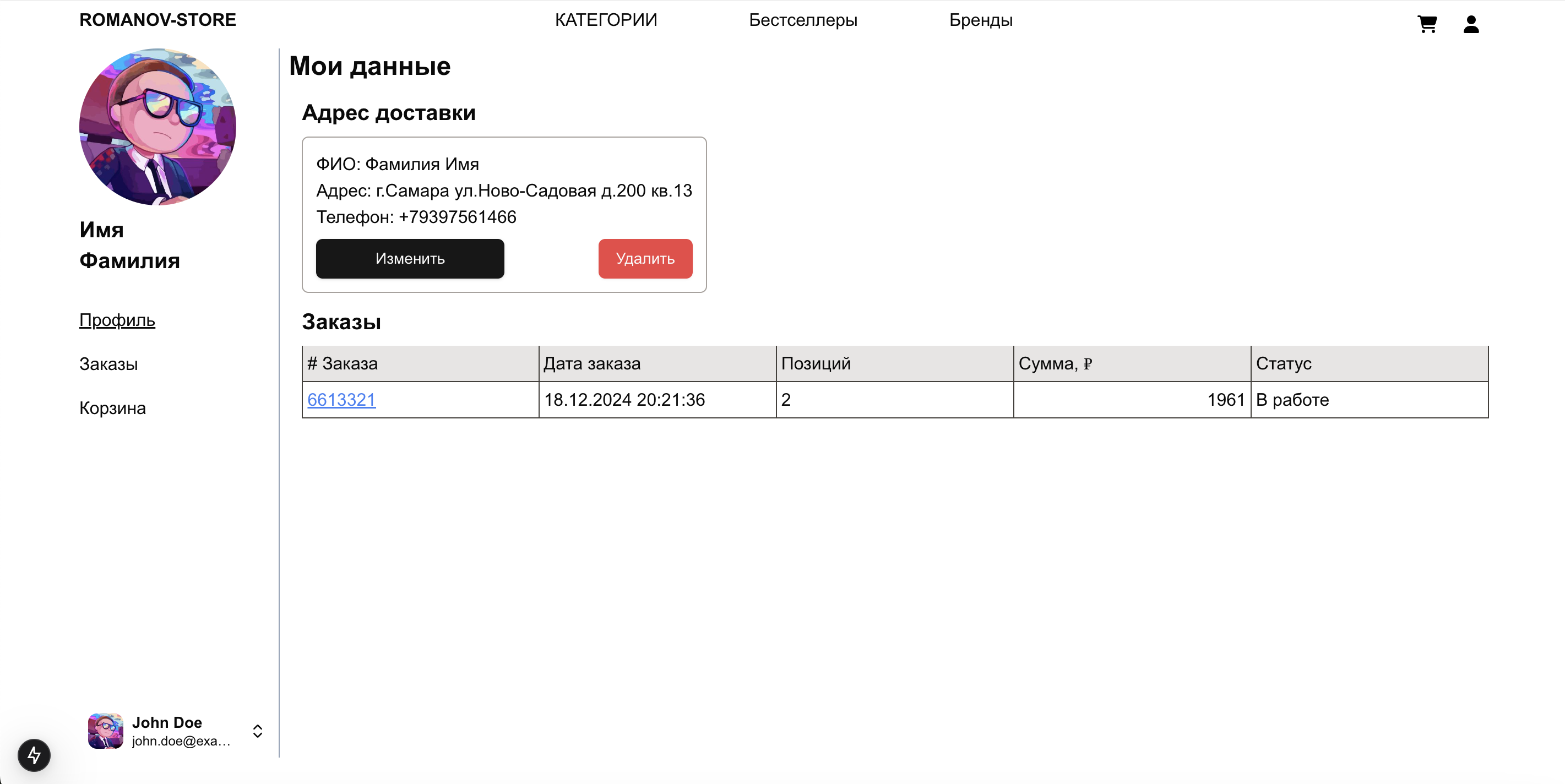


Рис 5 Личный кабинет