Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

КУРСОВАЯ РАБОТА

По Разработка приложений для мобильных операционных систем

Тема Веб-приложение «Магазин компьютерной техники»

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Банденков В.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

В данной работе реализовано веб-приложение «Магазин компьютерной техники» с использованием Spring Boot и СУБД MySQL. Приложение обеспечивает автоматизацию учёта ассортимента, управления продажами и формирование аналитических отчётов. Реализован механизм авторизации и распределения ролей (администратор, пользователь), а также функционал по регистрации, покупке товаров, формированию файловых отчётов и отправке уведомлений по электронной почте.

This coursework project presents a “Computer Store” web application developed with Spring Boot and MySQL. The application automates inventory management, sales tracking, and generating analytical reports. It includes user authentication and role-based access (administrator and regular user), as well as features for registration, product purchase, file-based report generation, and email notifications.

Содержание

[Введение 5](#_Toc185902793)

[1. Анализ технического задания. 6](#_Toc185902794)

[1.1 Описание предметной области 6](#_Toc185902795)

[1.2 Формирование требований к системе 6](#_Toc185902796)

[1.3 Обоснование выбора средств разработки 7](#_Toc185902797)

[2. Разработка моделей данных 10](#_Toc185902798)

[2.1 Концептуальная модель базы данных 10](#_Toc185902799)

[2.2 Логическая модель базы данных 10](#_Toc185902800)

[2.3 Физическая модель базы данных 11](#_Toc185902801)

[3. Проектирование работы системы 11](#_Toc185902802)

[3.1 Архитектура системы 11](#_Toc185902803)

[3.2 Основные функциональные требования 15](#_Toc185902804)

[3.3 Сценарии использования 15](#_Toc185902805)

[4. Разработка и реализация системы 16](#_Toc185902806)

[5. Тестирование системы 19](#_Toc185902807)

[Заключение 22](#_Toc185902808)

[Список литературы: 22](#_Toc185902809)

[Приложение 1. 23](#_Toc185902810)

[Приложение 2 26](#_Toc185902811)

[Приложение 3 28](#_Toc185902812)

# Введение

В условиях динамичного развития рынка компьютерной техники автоматизация процессов розничной торговли играет ключевую роль. Увеличение числа производителей и моделей техники требует создания надежных систем, способных управлять данными о товарах, продажах и складе.

Целью данной работы является разработка веб-приложения «Магазин компьютерной техники», обеспечивающего учет товаров, управление продажами, формирование отчетов и анализ бизнес-процессов. Разработка ведется с использованием фреймворка Spring Boot, который обеспечивает высокую производительность и гибкость.

# Основные задачи:

# Проведение анализа технического задания.

# Проектирование архитектуры системы.

# Разработка базы данных и пользовательского интерфейса.

# Реализация и тестирование системы.

# Результатом станет приложение, которое автоматизирует деятельность магазина, улучшит учет и анализ данных, а также повысит удобство для пользователей.

# 1. Анализ технического задания.

# 1.1 Описание предметной области

Веб-приложение «Магазин компьютерной техники» предназначено для автоматизации процессов, связанных с продажей новой компьютерной техники. В данной системе пользователи взаимодействуют через различные функциональные модули, каждый из которых выполняет определенные задачи.

Основные объекты предметной области:

Клиенты:

Клиенты могут зарегистрироваться в системе, создавая учетные записи, связывая их с покупками.

Зарегистрированные пользователи могут просматривать доступные модели компьютерной техники, добавлять товары в корзину и оформлять заказы.

Клиенты могут просматривать историю своих покупок и получать уведомления.

Администраторы:

Администраторы могут управлять данными о товарах: добавлять, редактировать и удалять их.

Формировать отчеты по продажам, анализировать выручку за заданный период времени.

Управлять пользователями (изменять роли, блокировать или активировать учетные записи).

Отслеживать остатки на складе.

База данных (БД) будет основой информационной системы, обеспечивая хранение, обработку и доступ к информации о товарах, пользователях, заказах и отчетах.

# 1.2 Формирование требований к системе

Магазин занимается продажей компьютерной техники (компьютеры, ноутбуки, периферийные устройства, комплектующие и аксессуары).

База данных должна содержать следующие элементы: марка и модель техники, основные характеристики, цена, количество на складе, изображения товара.

Необходимо реализовать возможность регистрации и авторизации пользователей в системе с подтверждением через email.

Пользователи смогут оформлять заказы, отслеживать их статус и просматривать историю покупок.

Администраторы должны иметь возможность:

Управлять ассортиментом товаров (добавлять, редактировать и удалять).

Анализировать продажи с помощью отчетов.

Следить за остатками товаров.

Система должна поддерживать формирование отчетов в удобных форматах (например, PDF или Excel).

Должна быть реализована адаптивная клиентская часть и валидация данных как на клиентской, так и на серверной сторонах.

# 1.3 Обоснование выбора средств разработки

Для разработки курсового проекта была выбрана среда разработки Spring Boot с использованием языка программирования Java. Для управления базой данных был выбран MySQL Workbench.

К достоинствам Spring Boot относятся:

* Поддержка работы с несколькими языками программирования, включая Java и Kotlin.
* Удобный редактор кода.
* Возможность разработки веб-приложений с использованием мощных инструментов для написания, отладки и тестирования.
* Встроенные средства для управления зависимостями с помощью Maven или Gradle.
* Большая библиотека с готовыми шаблонами и компонентами для разработки ПО.
* Поддержка последних версий Java и Spring Framework.
* Предварительная проверка уже созданного приложения на предмет ошибок.
* Большой набор средств для тестирования каждого элемента приложения.
* Руководство по использованию Spring Boot, доступное на официальном сайте.

Spring Boot — это фреймворк, предназначенный для создания веб-приложений на платформе Java. Он предоставляет разработчикам мощные инструменты для написания, отладки и тестирования приложений, включая редакторы кода, визуальные редакторы интерфейса и встроенные средства для управления зависимостями.

Основные компоненты Spring Boot:

* Проектная структура: Spring Boot использует гибкую структуру проектов, где файлы и ресурсы организованы по каталогам. Основные каталоги включают:

src/main/java — содержит код приложения.

src/main/resources — включает ресурсы, такие как конфигурационные файлы и статические ресурсы.

src/main/resources/application.properties — файл, определяющий основные характеристики приложения, такие как его конфигурации и настройки.

* Модули: Каждый проект может содержать несколько модулей, включая основное приложение и библиотечные модули. Это позволяет разделять функциональность и переиспользовать код.
* Контроллеры и Сервисы:

Controller — это компонент, представляющий один экран пользовательского интерфейса. Он отвечает за взаимодействие с пользователем и управляет жизненным циклом интерфейса.

Service — это часть интерфейса, которая может быть встроена в Controller. Сервисы позволяют создавать многоразовые компоненты и адаптировать интерфейс под различные размеры экранов.

* View и Layout:

View — это базовый класс для всех компонентов пользовательского интерфейса, таких как кнопки, текстовые поля и изображения.

Layout — это контейнеры, которые определяют, как элементы пользовательского интерфейса размещаются на экране. Существуют различные типы макетов, такие как Thymeleaf, JSP и другие, каждый из которых имеет свои особенности.

* Библиотеки и зависимости: Spring Boot поддерживает управление зависимостями с помощью системы Maven или Gradle. Это позволяет легко интегрировать сторонние библиотеки и модули, расширяющие функциональность приложения.

MySQL Workbench — это интегрированная среда разработки для управления базами данных MySQL. Она предоставляет разработчикам мощные инструменты для проектирования, разработки, администрирования и мониторинга баз данных.

Основные компоненты MySQL Workbench:

* ER-диаграммы — для визуального проектирования схем баз данных.
* SQL-редактор — для написания и выполнения SQL-запросов.
* Администрирование — инструменты для управления пользователями, правами доступа и настройками сервера.
* Мониторинг — средства для отслеживания производительности и состояния базы данных.

Spring Boot и MySQL Workbench предоставляют разработчикам все необходимые инструменты для создания качественных веб-приложений, обеспечивая простоту разработки, отладки и тестирования. Понимание структуры и компонентов этих инструментов позволяет эффективно использовать их возможности для достижения поставленных целей.

2. Разработка моделей данных

# 2.1 Концептуальная модель базы данных

Первый этап создания автоматизированной информационной системы - разработка концептуальной модели. При создании концептуальной модели базы данных мы демонстрируем область предмета, для которой предназначена система. Однако концептуальная модель не является идеальной для разработки, поэтому необходимо нормализовать все таблицы для создания реляционной базы данных. Следующим этапом является построение логической модели, которая аналогична концептуальной.

Модель учитывает все аспекты исследуемой предметной области, что позволяет получить точный результат. Далее необходимо создать физическую модель данных, которая определит, как данные будут храниться и доступны на физическом уровне: названия таблиц и столбцов, типы данных, первичные и внешние ключи и т.д.

Первым шагом в разработке концептуальной модели данных является выделение сущностей, а затем создание самой модели. Эта модель нужна для дальнейшего разрабатывания базы данных и её изменения, например, в реляционную базу данных. Результат представлен на рисунке 1 в приложении 1.

# 2.2 Логическая модель базы данных

Логическая модель графически представляет структуру базы данных, включая сущности и их взаимосвязи. Она показывает связи между объектами предметной области и описывает атрибуты каждой сущности. В данной системе сущности и их связи, следующие:

Пользователи связаны с ролями пользователей, где каждая роль определяет уровень доступа;

Продукты связаны с производителями, чтобы указать бренд товара;

Продажи содержат данные о продукте, покупателе и его производителе, а также о дате и стоимости сделки;

Корзины связаны с элементами корзин, в которых хранятся товары, добавленные пользователем.

На рисунке 2 представлена логическая модель, отражающая структуру базы данных для системы управления продажами компьютерной техники. Пример: сущность "Продукты" содержит информацию о наименовании, описании, цене, количестве на складе и идентификаторе производителя.

# 2.3 Физическая модель базы данных

Физическая модель данных, представленная на рисунке 3 в приложении 1, составлена на основе логической модели, где уже отражены и учтены все основные аспекты исследуемой предметной области. После проектирования этой модели будет осуществляться непосредственно создание базы данных и ее таблиц.

# 3. Проектирование работы системы

# 3.1 Архитектура системы

Конфигурационные классы:

MvcConfig: Настройка MVC для обработки статических ресурсов, таких как изображения и стили.

SecurityConfig: Настройка безопасности, включая аутентификацию и авторизацию пользователей.

DataInitializer: Инициализация начальных данных, включая создание администратора.

DateUtils: Утилиты для работы с датами.

Контроллеры:

AdminController: Управление административными функциями:

Управление пользователями, товарами и продажами.

Просмотр аналитики.

CartController: Управление корзиной:

Добавление, удаление товаров и очистка корзины.

Отображение содержимого корзины.

ErrorController: Обработка ошибок (например, 403 - доступ запрещен).

HomeController: Управление главной страницей:

Отображение популярного контента.

ManufacturerController: Управление производителями:

Добавление, редактирование и удаление производителей.

ProductController: Управление товарами:

Просмотр каталога товаров, деталей товара, добавление в корзину.

SaleController: Управление продажами:

Завершение продаж и просмотр истории продаж.

UserController: Управление пользователями:

Регистрация, авторизация, управление профилем и просмотр истории покупок.

Модели:

Product: Модель для хранения информации о товарах.

Sale: Модель для хранения информации о продажах.

User: Модель для хранения информации о пользователях.

Role: Перечисление для ролей пользователей (ROLE\_USER, ROLE\_ADMIN).

Cart: Модель для корзины пользователя.

CartItem: Модель для элемента корзины.

Manufacturer: Модель для хранения информации о производителях.

ProductDto: Объект передачи данных для модели Product.

Репозитории:

ProductRepository: Взаимодействие с таблицей товаров.

UserRepository: Управление данными пользователей.

SaleRepository: Работа с таблицей продаж.

CartRepository: Взаимодействие с корзинами пользователей.

ManufacturerRepository: Управление данными производителей.

Сервисы:

ProductService: Управление товарами:

Получение списка товаров, сохранение и удаление товаров, обновление остатков.

CartService: Управление корзинами:

Добавление, удаление товаров, очистка корзины.

SaleService: Управление продажами:

Завершение продаж и генерация отчетов.

UserService: Управление пользователями:

Регистрация, авторизация, обновление профиля.

EmailService: Отправка электронных писем (активация, уведомления).

CustomUserDetailsService:Реализация пользовательской аутентификации.

Утилиты:

DataInitializer: Инициализация начальных данных, таких как создание администратора.

DateUtils: Утилиты для работы с датами.

Таблицы базы данных:

products: Хранение информации о товарах.

users: Хранение информации о пользователях.

sales: Запись данных о продажах.

carts: Хранение корзин пользователей.

cart\_items: Элементы корзины.

manufacturers: Производители товаров.

Авторизация и регистрация:

Пользователь вводит данные на формах UserController:

Регистрация: Данные пользователя отправляются в UserService для валидации и добавления в базу данных. Пользователю отправляется письмо для активации аккаунта​

Авторизация: Пользователь вводит email и пароль, которые проверяются в UserService. При успешной проверке пользователь получает доступ к системе​

Просмотр товаров:

Пользователь просматривает товары на странице ProductController:

Данные о товарах загружаются из базы данных через ProductService.​

Добавление товара в корзину:

Пользователь выбирает товар на странице ProductController и нажимает "Добавить в корзину":

Товар добавляется в корзину пользователя через CartService.

Корзина пользователя хранится в таблице cart\_items​

Покупка товаров из корзины:

Пользователь переходит на страницу корзины через CartController:

Все товары, добавленные в корзину, загружаются из базы данных через CartService.

Пользователь подтверждает покупку всей корзины. CartService проверяет наличие товаров на складе, создает записи в таблице sales через SaleService и очищает корзину после покупки​

Профиль пользователя:

Пользователь может просматривать свои данные и историю покупок на странице UserController:

Данные о пользователе загружаются из базы данных через UserService.

История покупок извлекается из таблицы sales с использованием SaleService​

Панель администратора:

Администратор управляет пользователями, товарами, продажами и производителями через AdminController и ManufacturerController:

Управление пользователями: добавление, редактирование ролей, активация или деактивация учетных записей​

Управление товарами: добавление, редактирование и удаление товаров через AdminController и ProductService​

Управление производителями: добавление, редактирование и удаление производителей через ManufacturerController и ManufacturerService​

Управление продажами: просмотр и анализ данных о продажах через AdminController и SaleService​

# 3.2 Основные функциональные требования

Авторизация и регистрация:

Пользователь должен иметь возможность зарегистрироваться и авторизоваться.

Администратор управляет ролями пользователей.

Управление товарами:

Пользователь может просматривать товары и добавлять их в корзину.

Администратор добавляет, редактирует и удаляет товары.

Управление корзиной:

Пользователь добавляет товары в корзину, удаляет их и подтверждает покупку.

Управление производителями:

Администратор может добавлять, редактировать и удалять производителей.

Управление покупками:

Система должна записывать данные о продажах, обновлять количество товаров на складе и очищать корзину.

# 3.3 Сценарии использования

Регистрация нового пользователя:

Пользователь вводит данные на форме регистрации, которые обрабатываются через UserService. Пользователь получает письмо активации​

Авторизация пользователя:

Пользователь вводит email и пароль. Данные проверяются через UserService, и пользователь получает доступ к системе​

Добавление товара в корзину:

Пользователь добавляет товар через ProductController. CartService сохраняет товар в корзине пользователя​

Оформление покупки:

Пользователь завершает покупку всех товаров из корзины через CartController. Данные передаются в SaleService для создания записей о продажах​

Просмотр истории покупок:

История покупок загружается через UserController и отображается пользователю​

Управление товарами и производителями администратором:

Администратор добавляет, редактирует или удаляет товары через AdminController. Производители управляются через ManufacturerController​

# 4. Разработка и реализация системы

Разработка и внедрение веб-приложения для магазина компьютерной техники начинается с экранов авторизации (Приложение 3, рисунок 4) и регистрации пользователей (Приложение 3, рисунок 5).

Экран авторизации

Экран авторизации предоставляет возможность пользователям войти в систему, введя email и пароль. Интерфейс включает:

Поля для ввода email и пароля;

Кнопку "Авторизация";

Ссылку на форму регистрации для новых пользователей.

Форма авторизации реализована в контроллере UserController.

После ввода данных и нажатия на кнопку "Авторизация", система вызывает метод loadUserByUsername из CustomUserDetailsService, который проверяет наличие пользователя в базе данных.

В случае успешного входа пользователь перенаправляется на главную страницу со списком товаров (Приложение 3, рисунок 6). Если данные неверны, система отображает сообщение об ошибке.

Форма регистрации

Форма регистрации позволяет новым пользователям создать учетную запись. Интерфейс включает:

Поля для ввода имени, email, телефона и пароля;

Кнопку "Зарегистрироваться".

Особенности:

При попытке регистрации с уже существующим email выводится сообщение об ошибке.

Для активации учетной записи пользователю отправляется письмо с подтверждением (Приложение 3, рисунок 13). До активации пользователь не может войти в систему.

Главная страница и список товаров

После успешной авторизации пользователь попадает на главную страницу (Приложение 3, рисунок 6). Интерфейс включает:

Список доступных товаров с отображением названия, цены и кнопки "Подробнее";

Кнопку "Добавить в корзину" для каждого товара.

Функционал реализован в контроллере ProductController. При нажатии на "Подробнее" открывается детальная страница товара (Приложение 3, рисунок 7), где отображается описание, характеристики и производитель.

Корзина пользователя

Пользователь может добавлять товары в корзину. Корзина отображается на отдельной странице (Приложение 3, рисунок 11) и содержит:

Название товара;

Количество товаров;

Общую стоимость;

Кнопки для изменения количества или удаления товара.

Функционал корзины реализован в контроллере CartController. Пользователь может оформить заказ, нажав "Оформить заказ", что вызывает метод checkoutCart в сервисе CartService.

Профиль пользователя

Профиль пользователя (Приложение 3, рисунок 10) отображает:

Личные данные;

Историю покупок.

История покупок отображается в табличной форме с данными о товарах, дате покупки и цене.

Панель администратора (Приложение 3, рисунок 12)

Панель администратора предоставляет доступ к управлению системой. Она включает:

Управление пользователями: активация, удаление, назначение ролей (Приложение 3, рисунок 15);

Управление товарами: добавление, редактирование, удаление (Приложение 3, рисунок 16);

Управление производителями: добавление новых производителей и редактирование существующих (Приложение 3, рисунок 17);

Генерацию отчетов: возможность выгрузки данных о продажах в формате Excel (Приложение 3, рисунок 19);

Просмотр аналитики: графики и статистика по продажам (Приложение 3, рисунок 21).Данные функции реализованы в контроллерах AdminController, ManufacturerController и ProductController.

# 5. Тестирование системы

Основная цель проведения тестирования состоит в проверке соответствия реализации системы требуемой функциональности. Во время тестирования каждый тест регистрируется и его результаты сравниваются с ожидаемым. Если ожидаемый результат не совпадает с фактическим, это отмечается в протоколе тестирования. В таблице 1 представлена методика проведения тестирования разработанного программного продукта.

Таблица 1 – методика тестирования разработанной программы

|  |  |
| --- | --- |
| Выполненное действие | Полученный результат |
| Запуск исполняемого файла | Отображение формы авторизации. |
| Переход по ссылке | Отображение формы регистрации. |
| Заполнение данных | Отправка сообщения на указанный Email с целью подтверждения аккаунта |
| Заполнение существующими данными | Вывод сообщения "пользователь уже сущеcтвует!” |
| Авторизация под именем «admin» | Отображение интерфейса администратора. |
| Изменение товара | Отображение новых данных в форме |
| Удаление товара | Удаление данных из таблицы. |
| Добавление товара | Добавляется новый товар |
| Добавление производителя | Добавляется новый Производитель |
| Изменение производителя | Отображение новых данных в форме |
| Выбор нужных дат и нажатие на «Экспорт в Excel» | Скачивание файла с содержимым данных в рамках указанных админом дат |
| Авторизация пользователя | Отображение интерфейса пользователя. Возможность добавления товара в корзину |
| Нажатие кнопки «купить» | Отчистка корзины. В профиле пользователя добавляется история покупок. Отображение данных в т. Продаж у администратора |
| Нажатие на кнопку «продать» | Добавление товара в истории продаж пользователя. Добавление товара на главную страницу |

Продолжение таблицы 1

Результаты, полученные в ходе тестирования разработанного программного продукта, позволяют сделать заключение в том, что разработанная программа соответствует требованиям технического задания.

# Заключение

Данная работа посвящена разработке и реализации веб приложения для магазина компьютерной техники с использованием Spring Boot и MySQL. Реализация включала:

Анализ требований и проектирование базы данных;

Разработку интерфейсов и функциональных модулей;

Тестирование и отладку системы.

Результаты тестирования подтвердили соответствие системы заявленным требованиям. Реализованные функции обеспечивают:

Удобный интерфейс для пользователей;

Эффективное управление товарами, заказами и продажами;

Инструменты для администраторов, включая управление пользователями, товарами и производителями, а также генерацию отчетов.

Для дальнейшего улучшения системы рекомендуется:

Провести нагрузочное тестирование для оценки производительности при большом количестве данных.

Добавить интеграцию с внешними системами, такими как платежные сервисы

Система демонстрирует преимущества автоматизации бизнес-процессов с использованием современных технологий и предоставляет основу для дальнейшего расширения функциональности.

# Список литературы:

1. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов :под редакцией Р. Усманов. - 2-е изд. - Саратов : Профобразование, 2019. - 310

2. Свистунов, А. Н. Построение распределенных систем на Java : учебное пособие / А. Н. Свистунов. 3-е изд. Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. - 316 с.

3. Мухамедзянов, Р. Р. JAVA. Серверные приложения / Р. Р. Мухамедзянов. - Москва СОЛОН-Р, 2016. - 336 с

# Приложение 1.

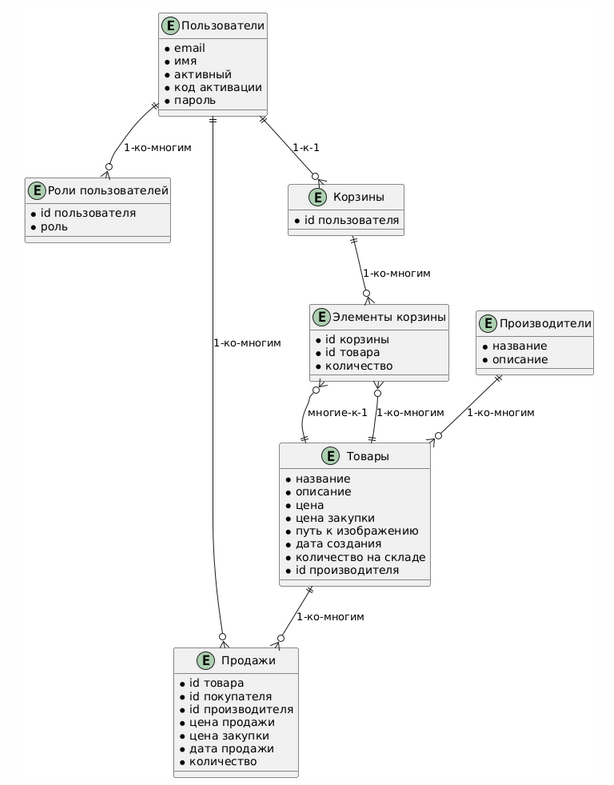


Рисунок 1 - Концептуальная модель данных

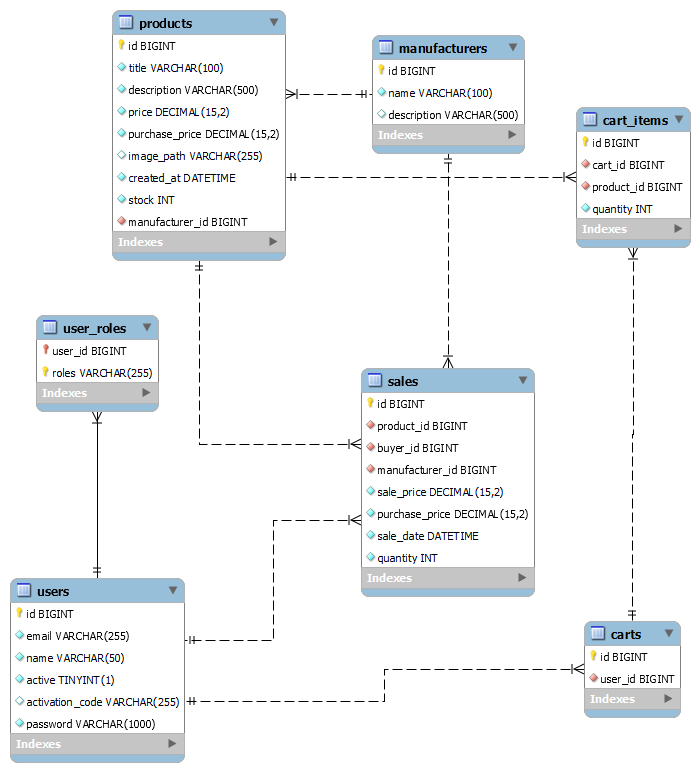


Рисунок 2 – Логическая модель данных

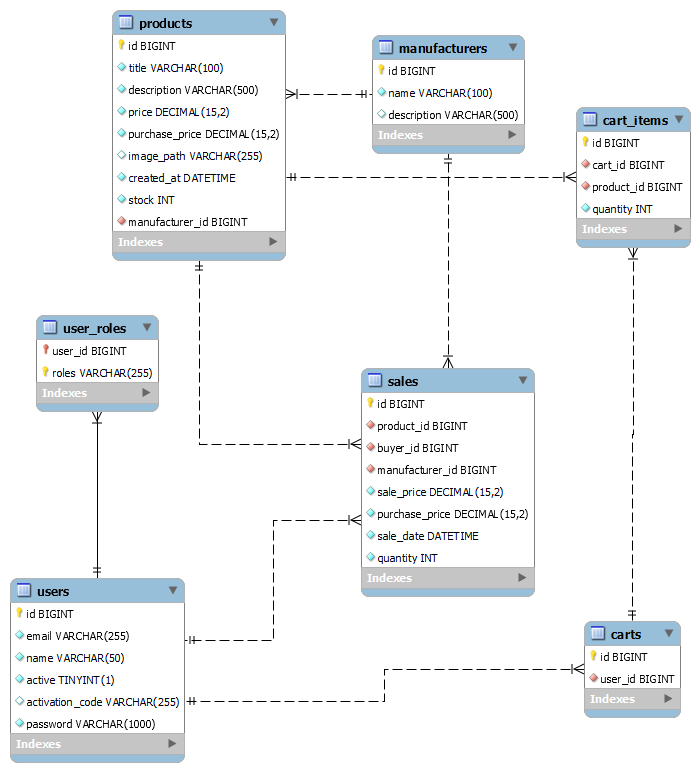


Рисунок 3 – Физическая модель данных

# Приложение 2

С подробным кодом программы вы можете ознакомиться по данной ссылке:

# https://github.com/Brbzdum/computer\_store.git

# Приложение 3

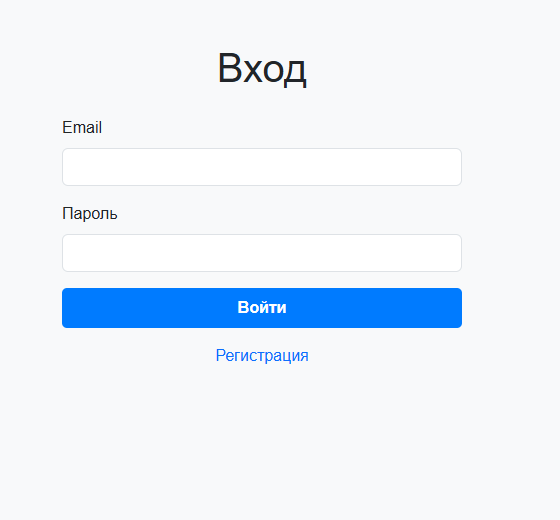


Рисунок 4 – Экран авторизации

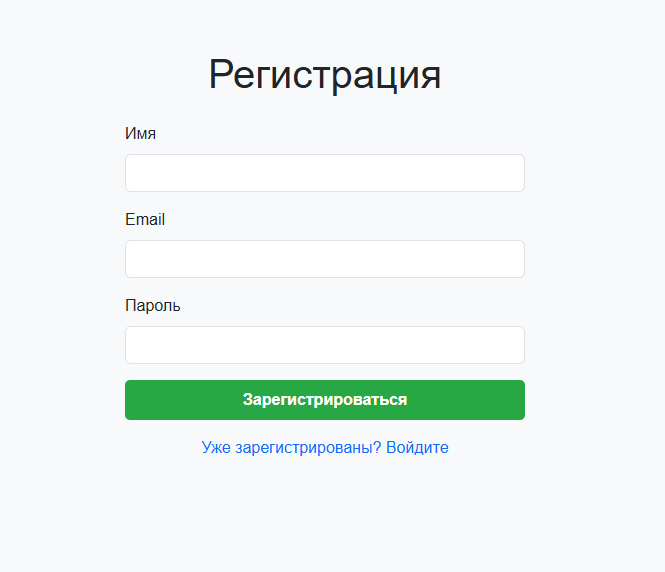


Рисунок 5 – Экран регистрации

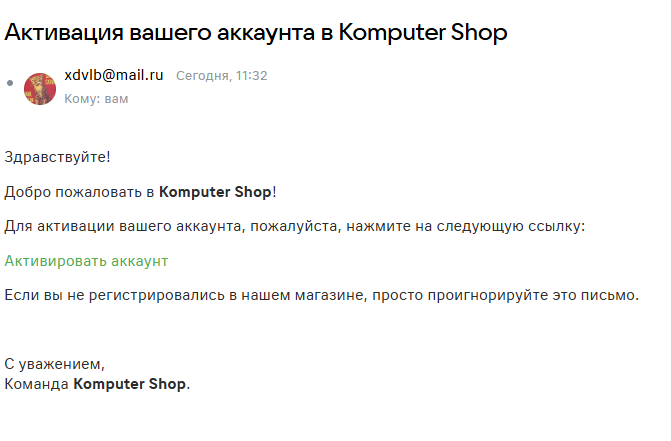


Рисунок 13 – Подтверждение регистрации

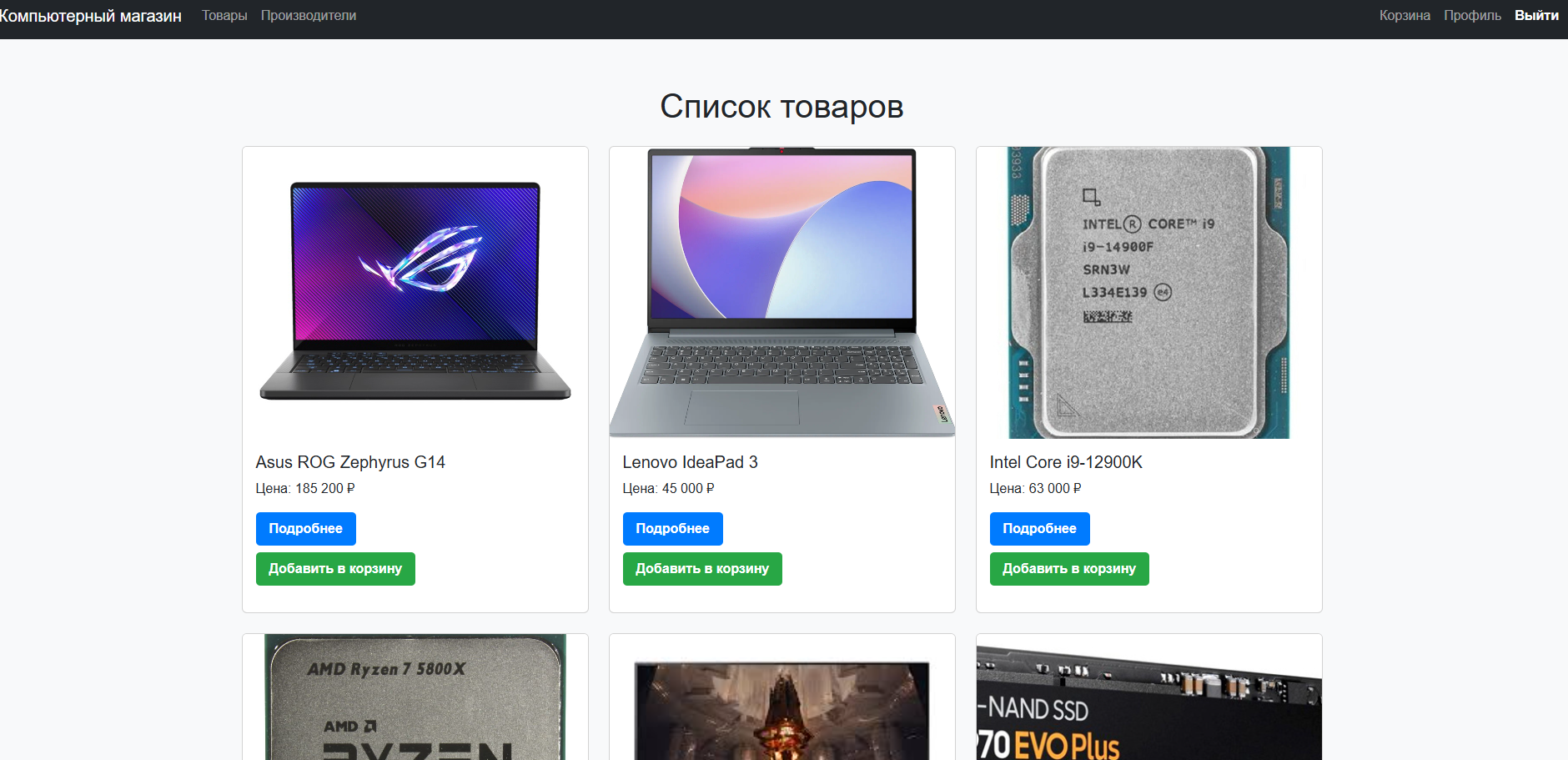


Рисунок 6 – Главный экран со списком товаров

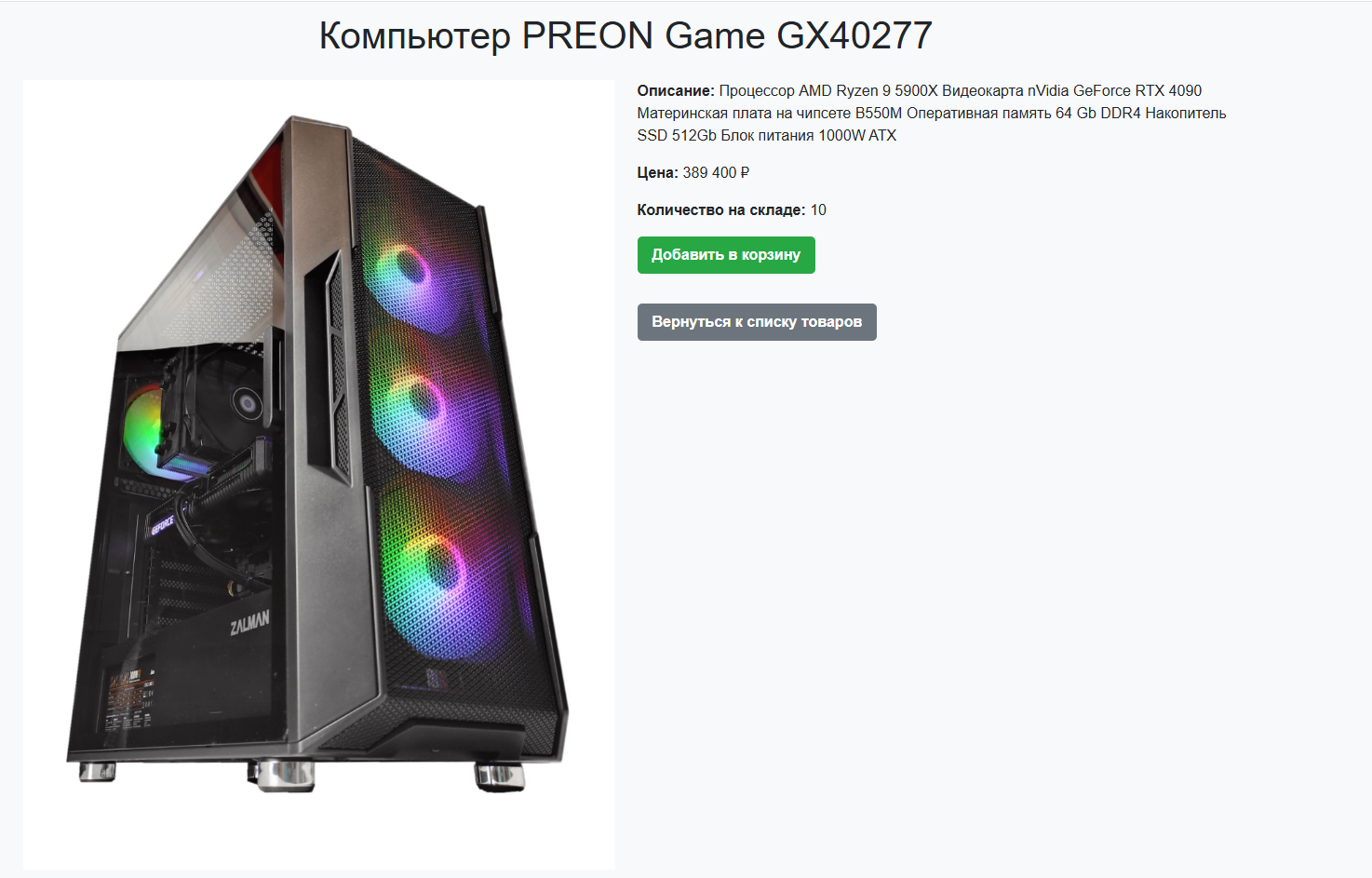


Рисунок 7 – Экран просмотра товара

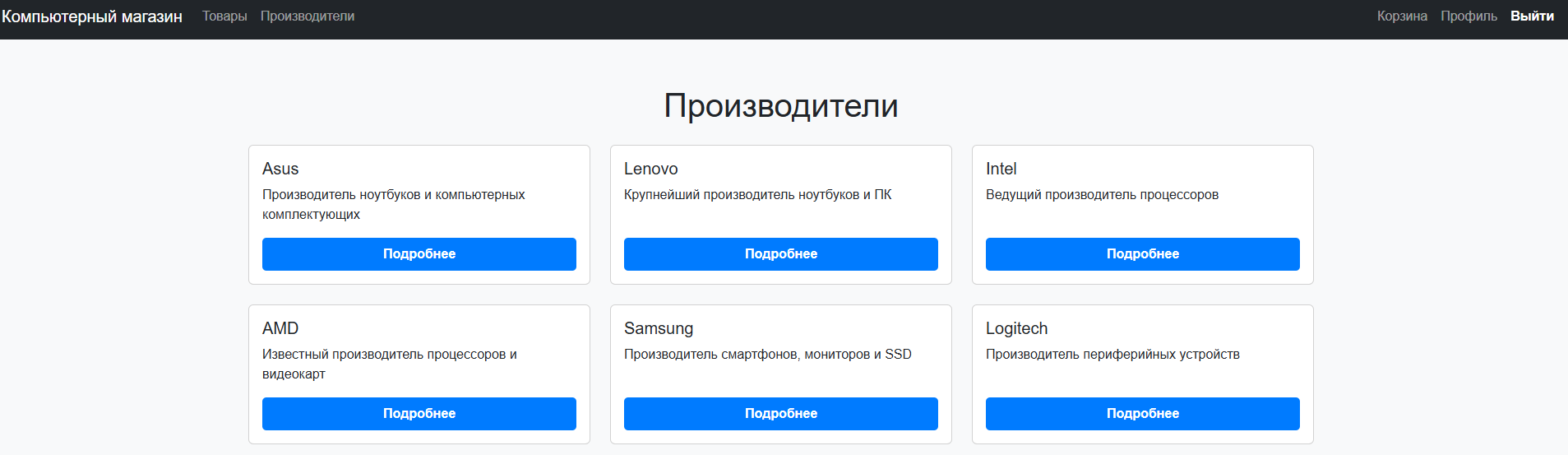


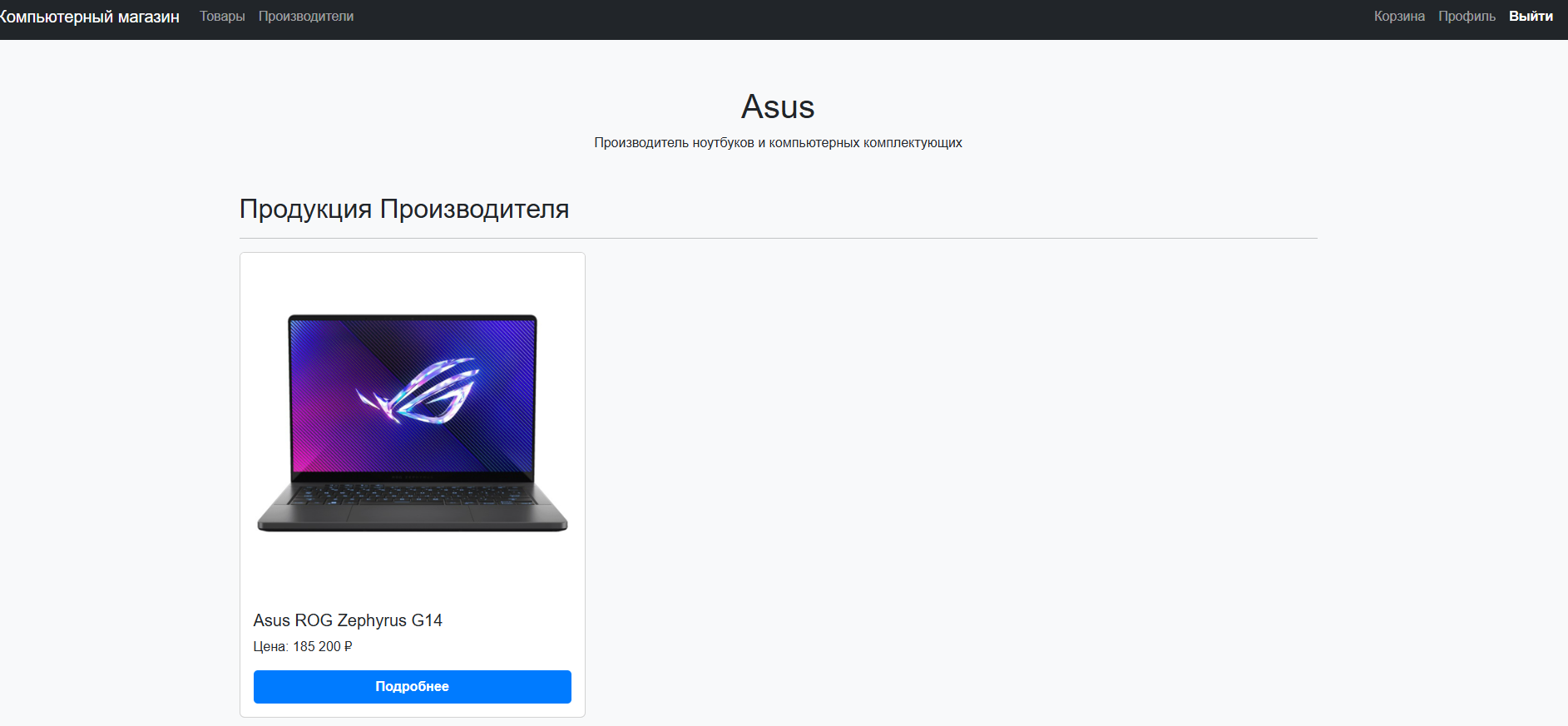
Рисунок 8– Экран со списком производителей   


Рисунок 9– Экран просмотра производителя и его товаров

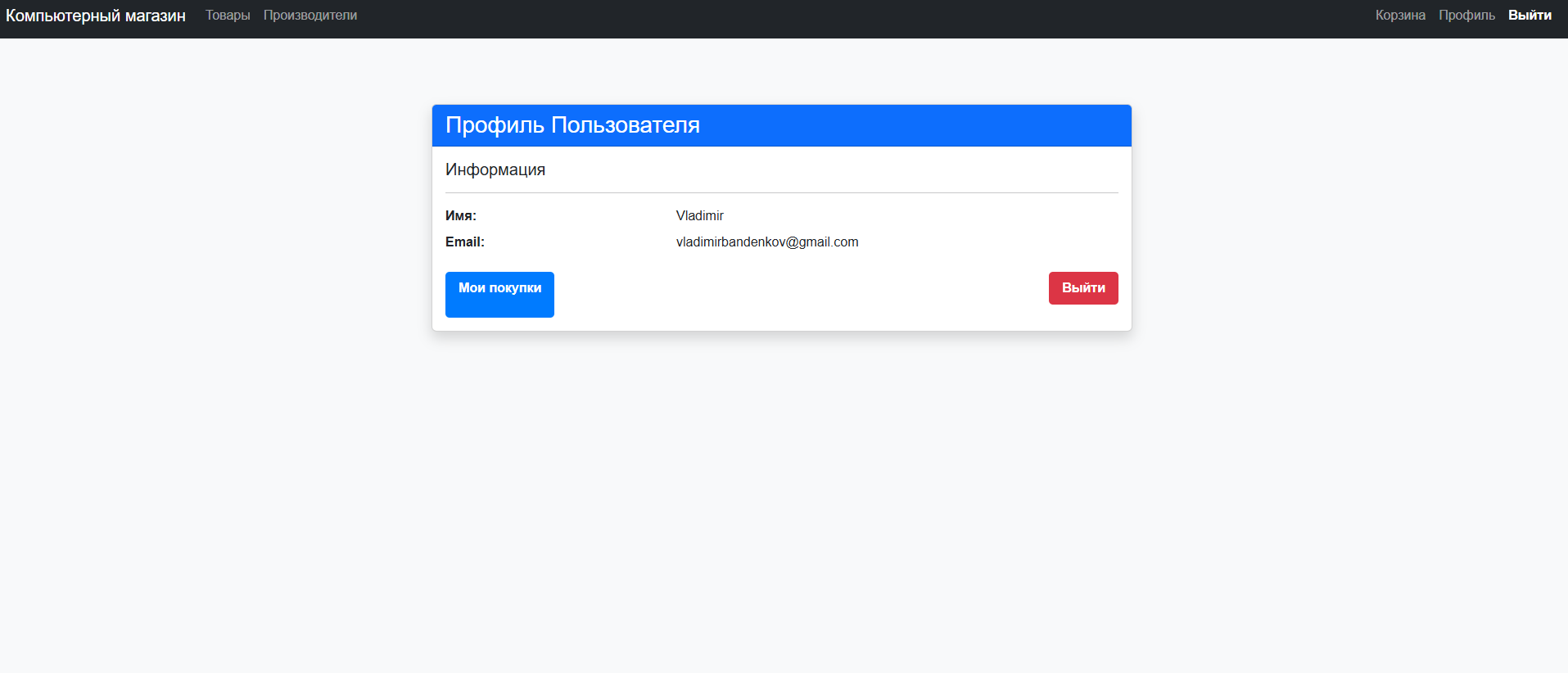


Рисунок 10– Профиль пользователя

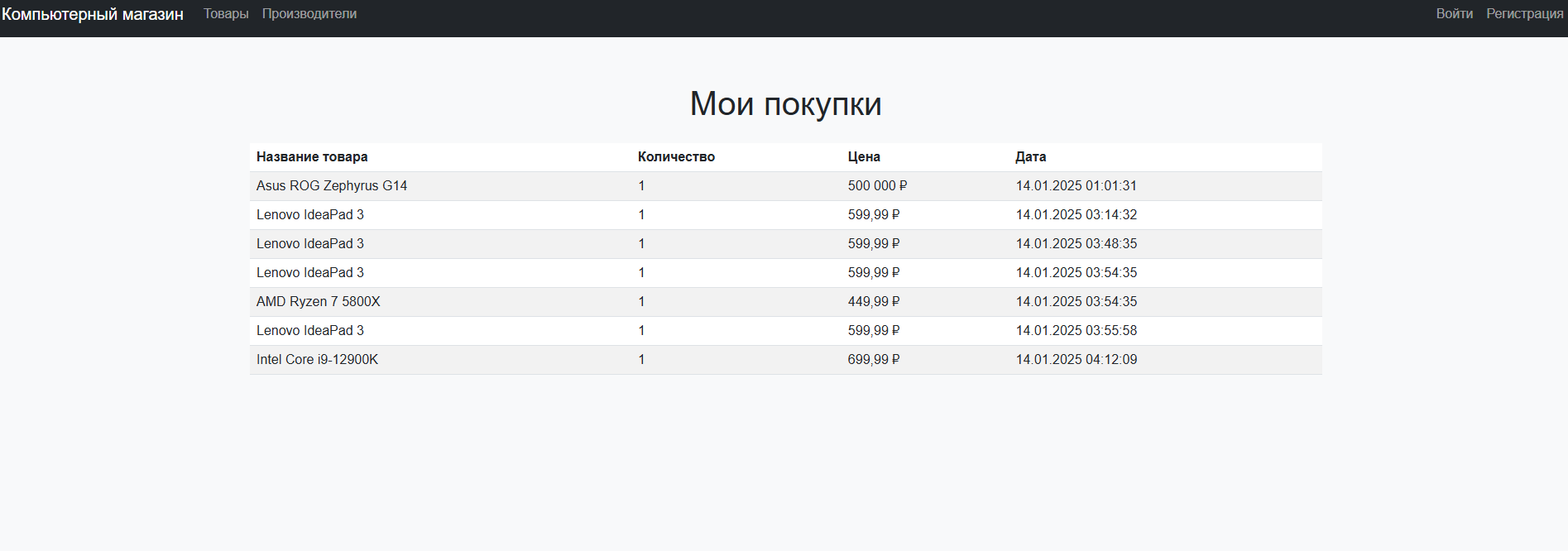


Рисунок 10 – История покупок

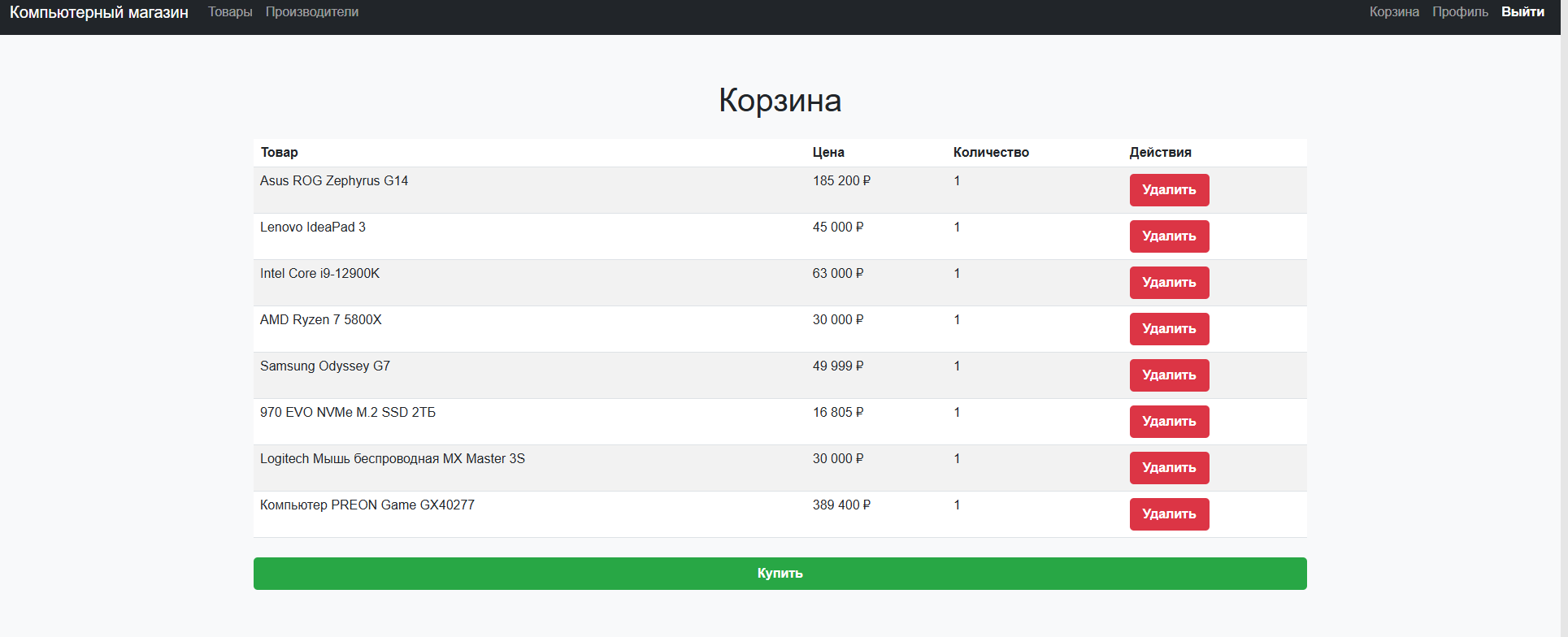


Рисунок 11 – Корзина пользователя

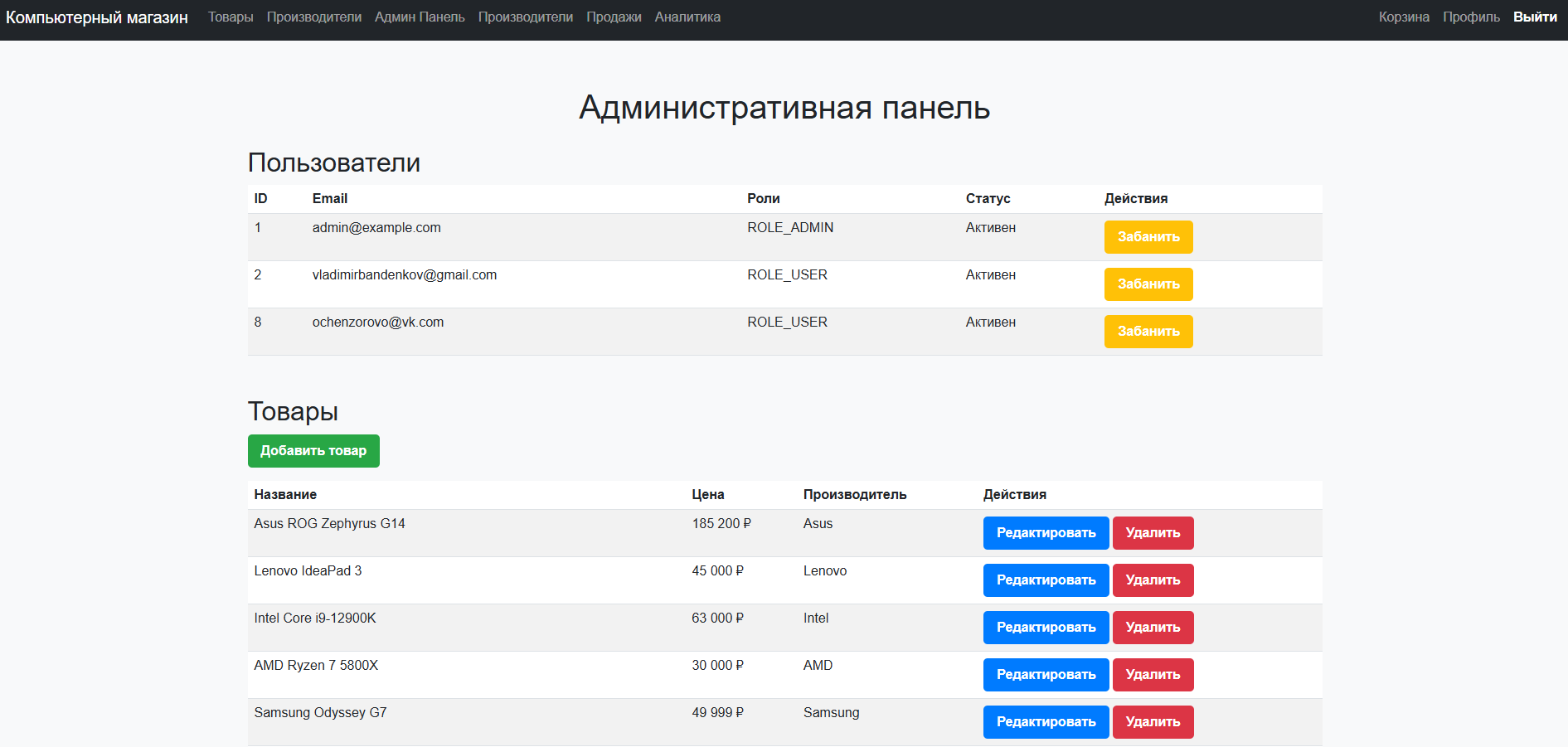


Рисунок 12 – Административная панель

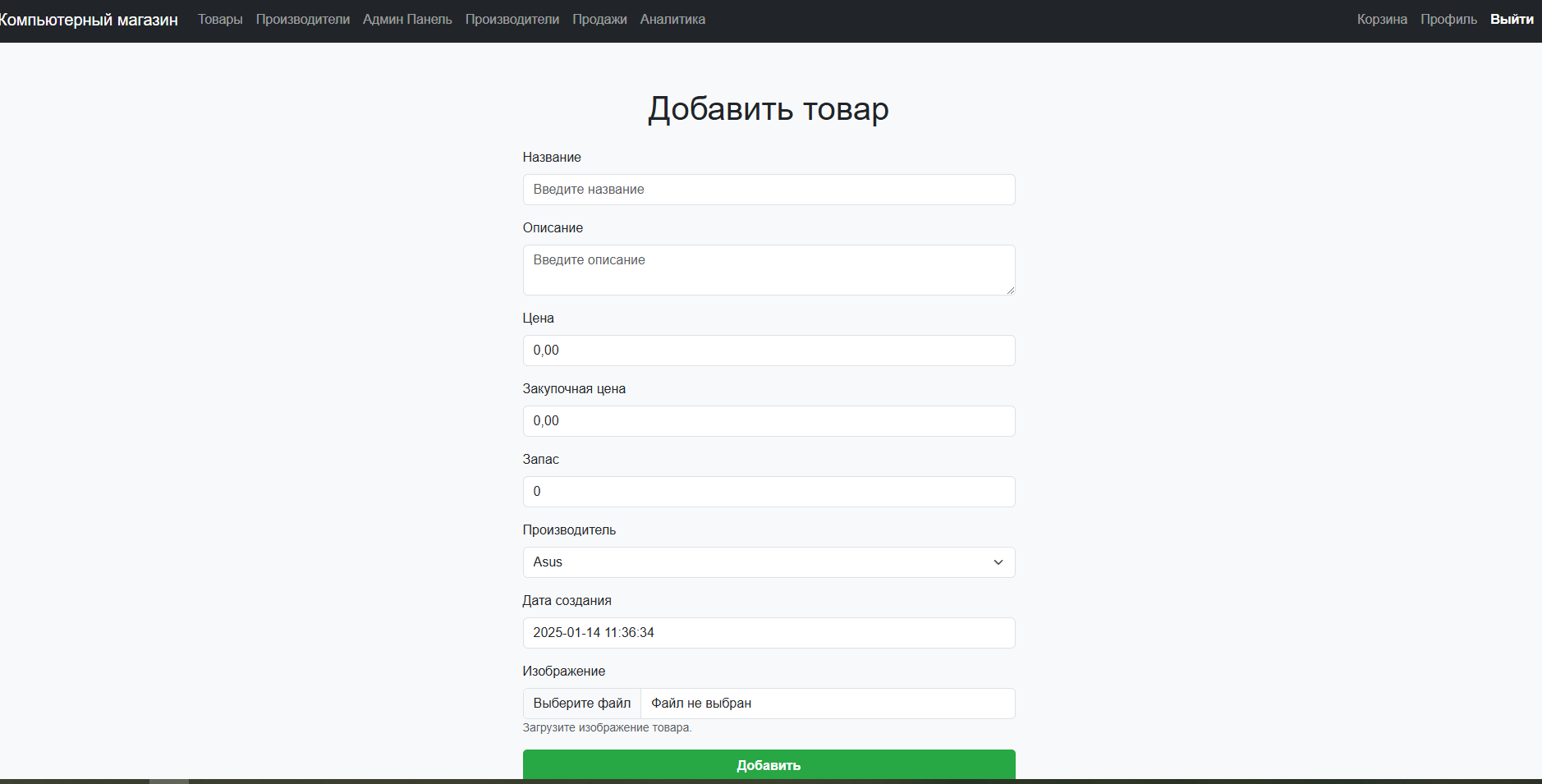


Рисунок 14 – Добавление товара

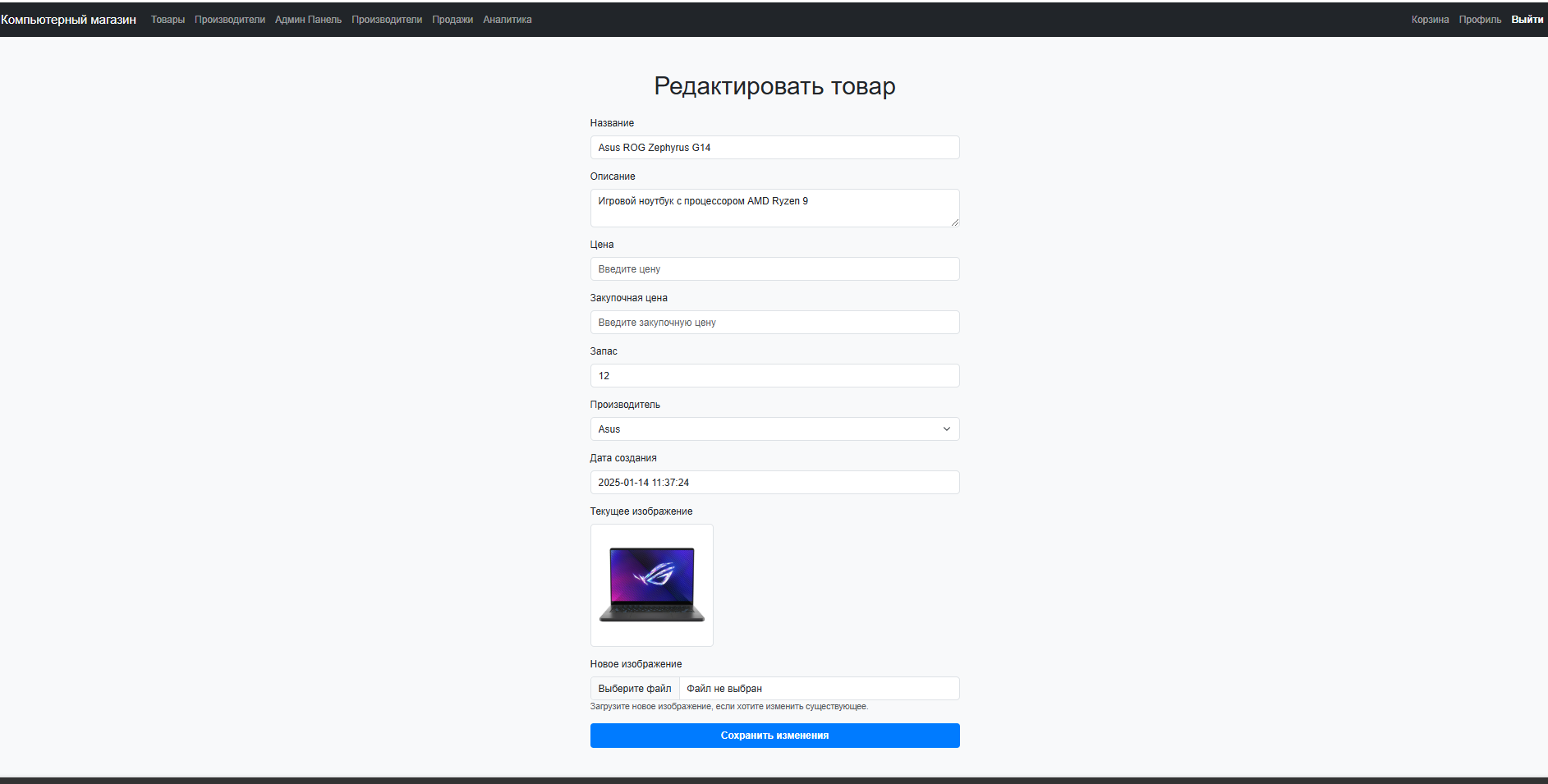


Рисунок 15 – Редактирование товара

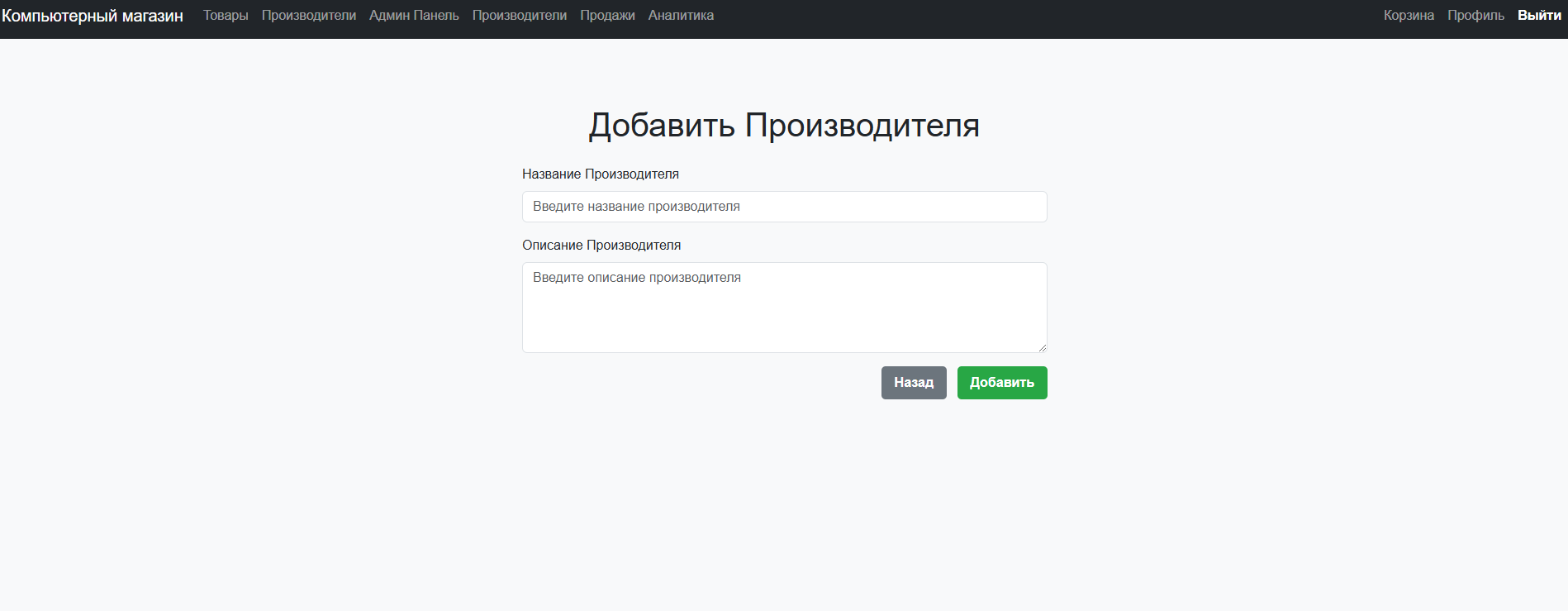


Рисунок 16 – Добавление производителя

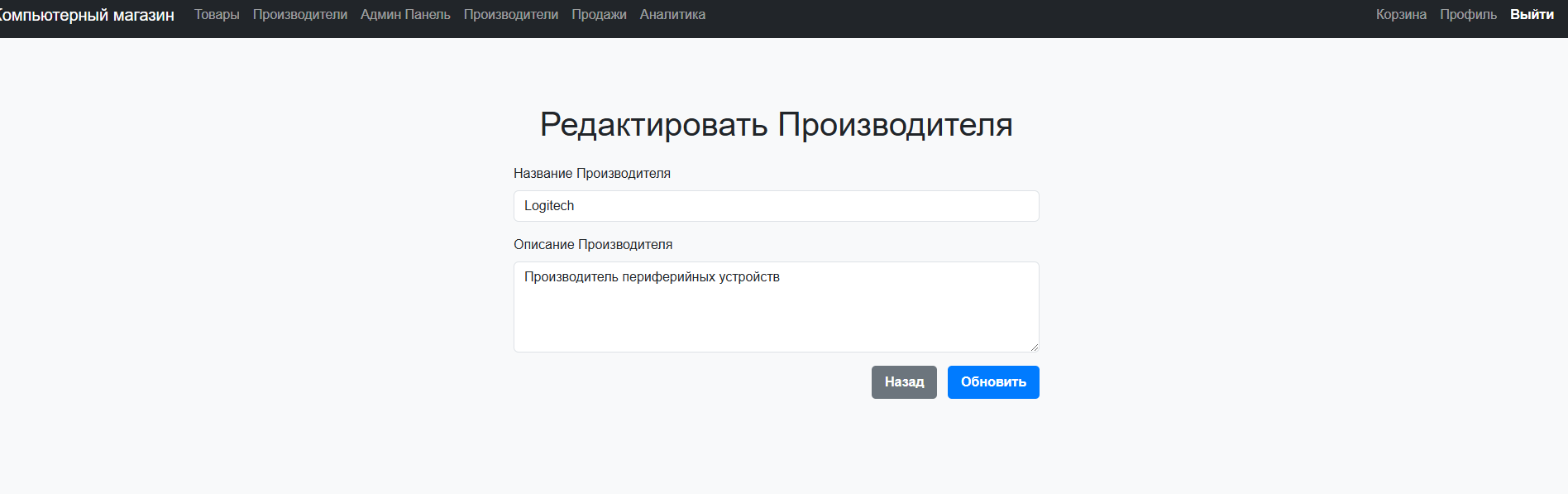


Рисунок 17 – Редактирование производителя

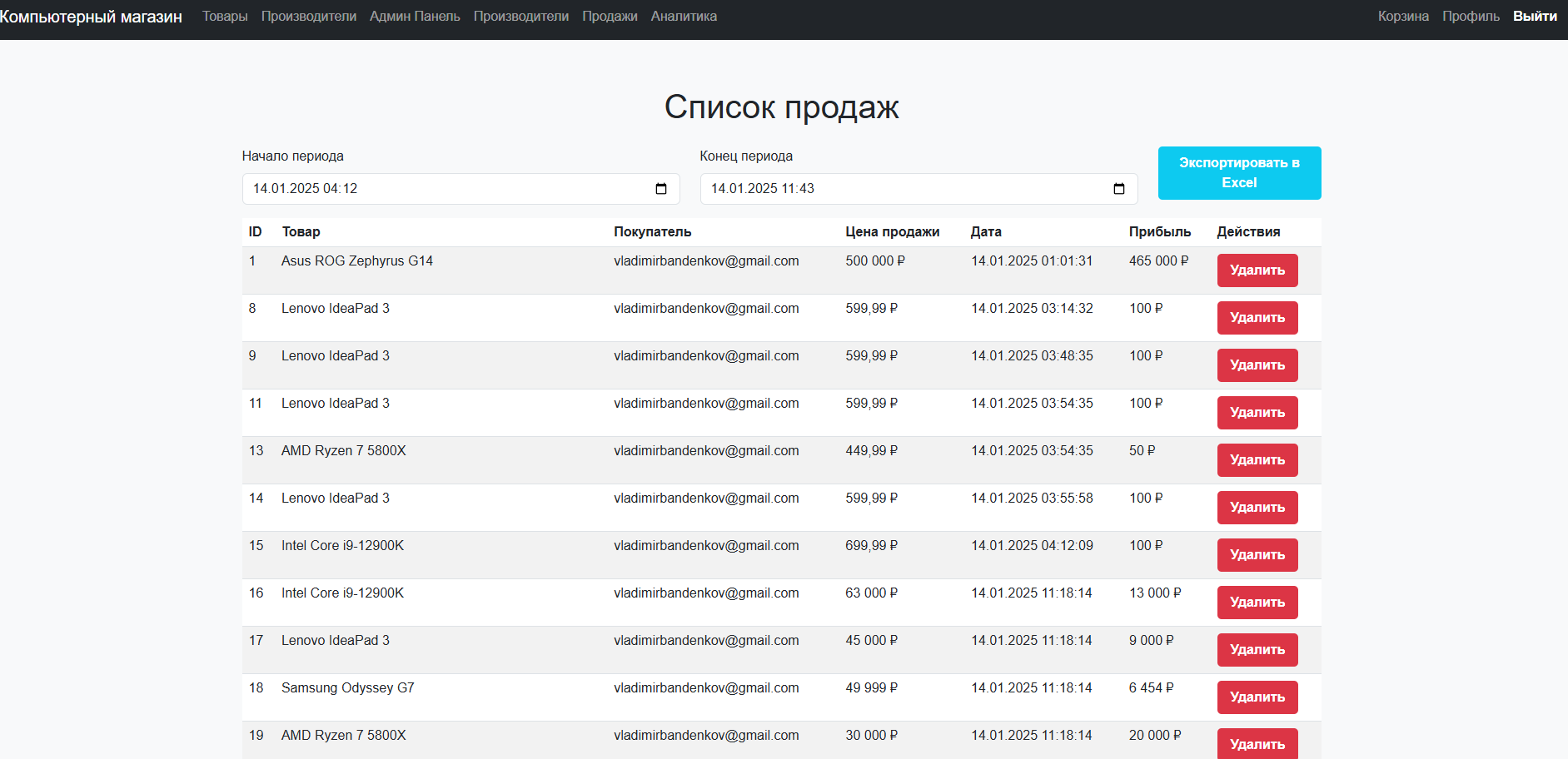


Рисунок 18 – Список продаж

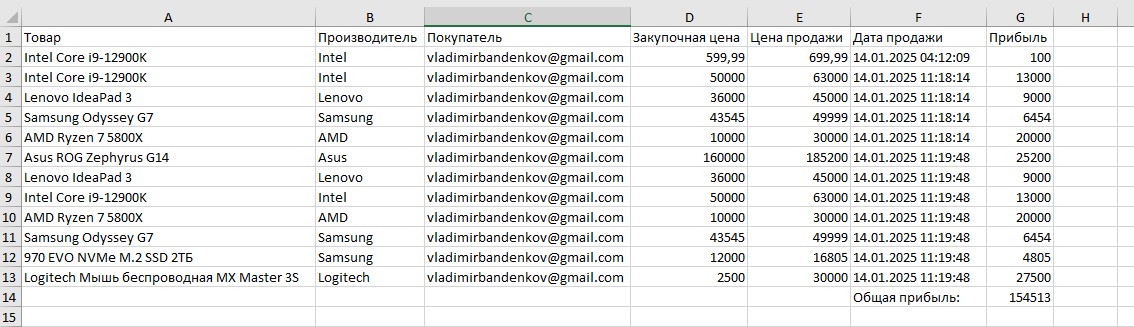


Рисунок 19 – Полученные данные в excel

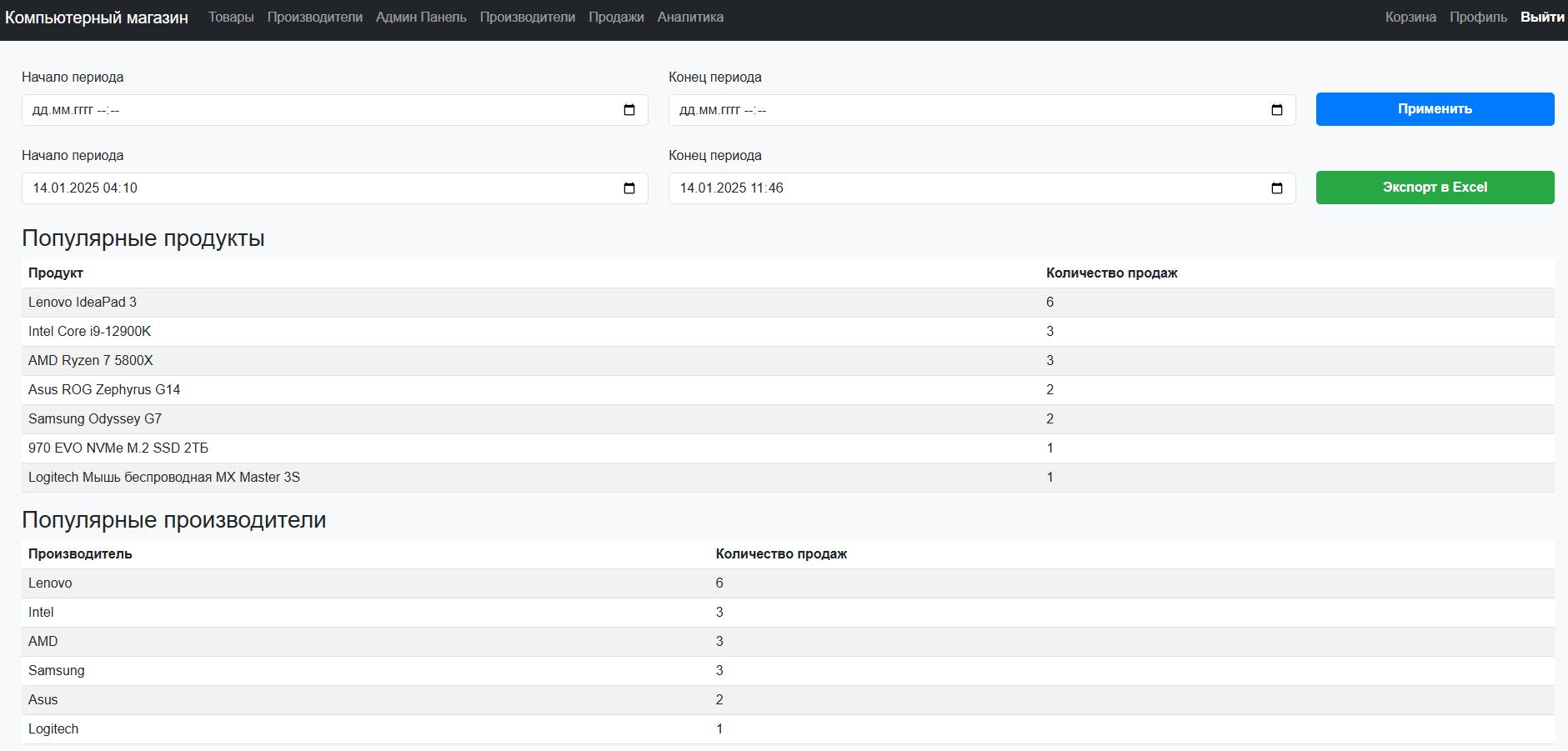
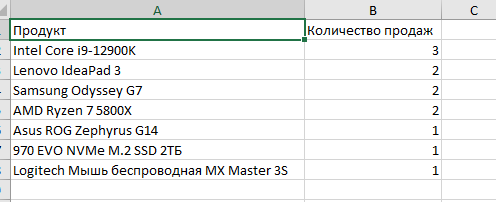


Рисунок 20 –Аналитика по популярным производителям и товарам



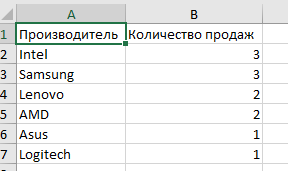


Рисунок 21,22 – Полученные данные в excel