Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ярославский государственный технический университет»

Кафедра «Информационные системы и технологии»

Отчет защищен

с оценкой \_\_\_\_\_\_

Преподаватель

\_\_\_\_\_\_\_\_\_\_ Д.В. Малышев

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ – «ВЕБ-ПРИЛОЖЕНИЯ ДИСТРИБЬЮТОРА СПОРТИВНЫХ ПРОДУКТОВ РАЗНЫХ БРЕНДОВ». FRONT-END**

Отчет о лабораторной работе №3

по курсу «Архитектура и паттерны Web-проектирования»

ЯГТУ 09.03.02 – 017 ЛР

Отчет выполнил

студент гр. ЦИС-37

\_\_\_\_\_\_\_\_\_\_ Ю.Н. Баканова

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025

2025

Содержание

[1 Введение 3](#_Toc198369126)

[2 Аналитическая часть 4](#_Toc198369127)

[2.1 Системный анализ предметной области 4](#_Toc198369128)

[2.2 UML проектирование 5](#_Toc198369129)

[2.2.1 Диаграмма вариантов использования 5](#_Toc198369130)

[3 Проектная часть 7](#_Toc198369131)

[3.1 Общая схема информационной системы 7](#_Toc198369132)

[3.3 Проектирование интерфейса клиентской части 8](#_Toc198369133)

[4 Технологическая часть 13](#_Toc198369134)

[4.1 Описание выбора инструментальных средств 13](#_Toc198369135)

[4.2 Краткое описание разрабатываемой системы 13](#_Toc198369136)

[4.3 Описание интерфейса 14](#_Toc198369137)

[4.4 Особенности разработки 15](#_Toc198369138)

[4.5 Особенности эксплуатации 16](#_Toc198369139)

[5 Заключение 18](#_Toc198369140)

[Приложение А. Ссылки 19](#_Toc198369141)

[А.1 Ссылка на репозиторий в GitHub 19](#_Toc198369142)

[А.2 Ссылка на проект в Figma 19](#_Toc198369143)

[Приложение Б. Листинги front-end 20](#_Toc198369144)

[Б.1 Код Login.cshtml 20](#_Toc198369145)

[Б.2 Код Catalog/Products.cshtml 21](#_Toc198369146)

[Б.3 Код Profile.cshtml 25](#_Toc198369147)

[Б.4 Код Cart.cshtml 30](#_Toc198369148)

[Б.5 Код Admin/Product/List.cshtml 35](#_Toc198369149)

[Б.6 Код Admin/Store/Edit.cshtml 36](#_Toc198369150)

# 1 Введение

В рамках данного проекта разработано веб-приложение дистрибьютора спортивных продуктов разных брендов, предназначенное для оптимизации процессов взаимодействия между потребителями и продавцами спортивного питания. Система обеспечивает удобную и быструю работу с каталогом товаров, позволяет пользователю находить нужную продукцию, сравнивать предложения разных производителей и бронировать товары в выбранных торговых точках.

Для реализации серверной части приложения использовался язык программирования C#, благодаря которому была обеспечена высокая производительность и надёжность логики взаимодействия с базой данных. Архитектура системы основана на паттерне MVC (Model–View–Controller), который разделяет представление, бизнес-логику и обработку данных, облегчая поддержку и развитие проекта. Интерфейс пользовательской части был предварительно разработан в среде Figma, а затем реализован с использованием HTML и CSS, что обеспечило современный, адаптивный и понятный интерфейс.

Целью создания системы является разработка единой онлайн-платформы, которая позволит пользователям быстро находить и бронировать спортивные продукты различных брендов, избавляя их от необходимости посещать множество сайтов и магазинов.

Основными задачами являются:

– Формирование единого цифрового пространства с продукцией разных брендов спортивного питания;

– Обеспечение быстрого поиска и фильтрации товаров по различным критериям (категория, бренд, цена и др.);

– Реализация функции онлайн-бронирования с возможностью выбора магазина для самовывоза;

– Создание личного кабинета пользователя для отслеживания заказов, редактирования данных и написания отзывов;

– Внедрение административной панели для управления товарами, пользователями, заказами и отзывами;

– Поддержка системы оценок и отзывов, позволяющей формировать прозрачную и надёжную среду покупок.

Результатом является комплексная система, объединяющая в себе функциональность интернет-магазина, информационного портала и сервиса предварительного заказа. Приложение направлено на улучшение пользовательского опыта, а также на расширение возможностей бизнеса в области дистрибуции спортивных товаров.

# Аналитическая часть

## Системный анализ предметной области

Разработка веб-приложения для дистрибуции спортивных продуктов различных брендов актуальна в условиях стремительно растущего интереса к здоровому образу жизни, фитнесу и спортивному питанию. Сегодня все больше людей обращают внимание на то, что они потребляют, выбирая качественные добавки, витамины и специализированное питание для достижения определённых спортивных целей. Однако рынок спортивных товаров в большинстве случаев представляет собой разрозненную систему, где каждый бренд работает через отдельные сайты или офлайн-точки продаж. Это создаёт множество неудобств для потребителей: от сложностей с поиском нужного продукта и сравнения цен — до отсутствия единых стандартов оформления заказа, оплаты и получения товара.

Проектируемое веб-приложение призвано устранить эти проблемы за счёт создания единой цифровой платформы, на которой будут собраны товары различных брендов спортивного питания. Назначение системы заключается в том, чтобы обеспечить пользователей доступом к структурированной и удобной для взаимодействия базе данных продукции с возможностью бронирования товаров в определённой торговой точке. Таким образом, клиент может выбрать интересующий его продукт, ознакомиться с характеристиками, проверить наличие и оформить бронь в ближайшем магазине — экономя при этом время и снижая риски, связанные с отсутствием нужного товара.

Цель создания данной системы заключается в упрощении и оптимизации процесса поиска, выбора и приобретения спортивного питания. Благодаря централизованной структуре, пользователи смогут взаимодействовать с большим ассортиментом продукции без необходимости переходить с одного ресурса на другой, получая при этом точную информацию о товаре и прозрачные условия покупки. Система выступает связующим звеном между покупателями и дистрибьюторами, улучшая как пользовательский опыт, так и бизнес-процессы в сфере продаж.

В рамках реализации поставленной цели проектируемая система решает ряд ключевых задач:

* Объединение ассортимента различных брендов на одной платформе. Система предоставляет единый доступ к товарам от разных производителей, что позволяет покупателям делать осознанный выбор без необходимости изучать отдельные сайты.
* Обеспечение удобного пользовательского интерфейса для поиска и фильтрации товаров. Пользователь может легко находить интересующий продукт по названию, бренду, категории или другим параметрам, что значительно сокращает время на выбор.
* Реализация функционала онлайн-бронирования товаров. Возможность зарезервировать товар в удобной торговой точке без немедленной покупки позволяет избежать ситуации, когда нужная позиция отсутствует в наличии.
* Предоставление личного кабинета для клиента. В личном профиле пользователь может управлять бронированиями, просматривать историю заказов, редактировать персональные данные и оставлять отзывы о продуктах.
* Администрирование ассортимента и заказов через панель администратора. Для администратора предусмотрен интерфейс управления пользователями, товарами и заказами. Он сможет добавлять новую продукцию, редактировать информацию, просматривать аналитику по продажам и управлять отзывами. Всё это делает систему эффективным инструментом для бизнеса, позволяющим охватить более широкую аудиторию и наладить удобное взаимодействие между брендами и конечными потребителями.
* Поддержка обратной связи и отзывов. Система позволяет пользователям оставлять оценки и комментарии, повышая доверие к товарам и улучшая прозрачность платформы.

Таким образом, проектируемая система не просто заменяет существующие инструменты для онлайн-продаж, а создаёт принципиально новую форму взаимодействия между клиентом и продавцом спортивного питания. Она сочетает в себе элементы информационного портала, системы бронирования и торговой площадки, что делает её внедрение целесообразным, эффективным и перспективным в условиях цифровизации торговли.

## UML проектирование

В этом разделе представлены UML-диаграммы, реализованные в соответствии с содержимым раздела 2.1.

### 2.2.1 Диаграмма вариантов использования

Диаграмма вариантов использования, представленная для веб-приложения дистрибуции спортивного питания, описывает ключевые сценарии взаимодействия пользователей с системой. Основные действующие лица — это гость, авторизованный пользователь и администратор. Каждый из них обладает определённым набором возможностей, соответствующим их роли в системе.

Гость — это пользователь, не прошедший авторизацию. Он имеет доступ к базовому функционалу приложения. В частности, гость может просматривать каталог товаров, осуществлять поиск по различным параметрам, переходить к карточке товара для изучения подробной информации, а также просматривать отзывы, оставленные другими пользователями. Кроме того, ему доступна возможность мониторинга наличия товара в магазинах, что позволяет понять, где именно продукт доступен для бронирования или покупки.

Авторизованный пользователь обладает расширенным функционалом по сравнению с гостем. Он может выполнять все действия, доступные неавторизованному пользователю, а также имеет право оставлять отзывы к товарам, оформлять заказы, управлять своей корзиной — заполнять корзину необходимыми позициями, удалять их. Также ему предоставляется возможность выбрать удобный магазин для получения товара и отслеживать статус своего заказа.

Администратор имеет наибольший уровень доступа и возможностей. В дополнение ко всем действиям пользователя он может управлять товарами, магазинами, заказами, пользователями и отзывами. Работа с объектами системы реализована через стандартные операции CRUD (создание, чтение, обновление, удаление). Администратор может просматривать данные зарегистрированных пользователей, управлять контентом каталога, изменять адреса магазинов, модерировать отзывы, а также вести мониторинг и корректировку статусов заказов. Таким образом, он играет роль управляющего и модератора всей системы.

Таким образом, диаграмма вариантов использования, представленная на рис. 1, наглядно демонстрирует архитектуру ролей и сценариев использования в системе. Она подчёркивает различие между базовыми действиями пользователей и расширенными административными возможностями, обеспечивая надёжное разделение прав доступа и удобное взаимодействие с системой на всех уровнях.

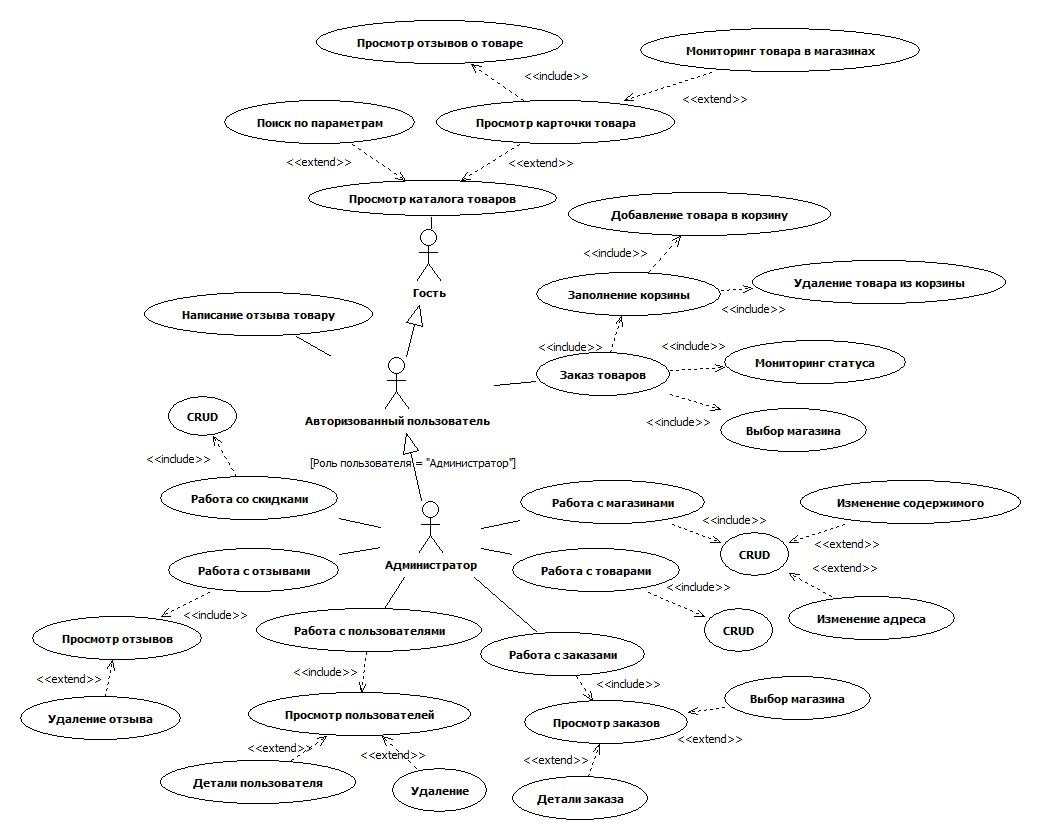


Рисунок 1 – Диаграмма вариантов использования

# Проектная часть

## Общая схема информационной системы

Общая схема информационной системы веб-приложения дистрибьютора спортивных продуктов разных брендов, представленная на рис. 2, отражает структуру интерфейса и взаимодействие пользователей с функциональными возможностями системы. Архитектура приложения разделена на два основных блока: пользовательскую часть и административную панель.

Пользовательская часть включает экраны, доступные как неавторизованным, так и авторизованным пользователям. Основной функционал реализован на страницах «Главная», «О нас», «Гид по питанию», «Каталог товаров» и «Карточка товара», где предоставляется информация о товарах, их наличие и отзывы. После авторизации становятся доступны дополнительные функции, такие как добавление товаров в корзину, оформление заказов, публикация отзывов и управление личными данными. Эти действия выделены красноватым цветом на схеме.

Административная панель включает разделы для управления товарами, заказами, магазинами, отзывами и пользователями. Администратор может добавлять и редактировать записи, отслеживать заказы и управлять контентом платформы.

Таким образом, структура схемы охватывает все переходы между экранами и позволяет визуально представить функциональные возможности системы на разных уровнях доступа.

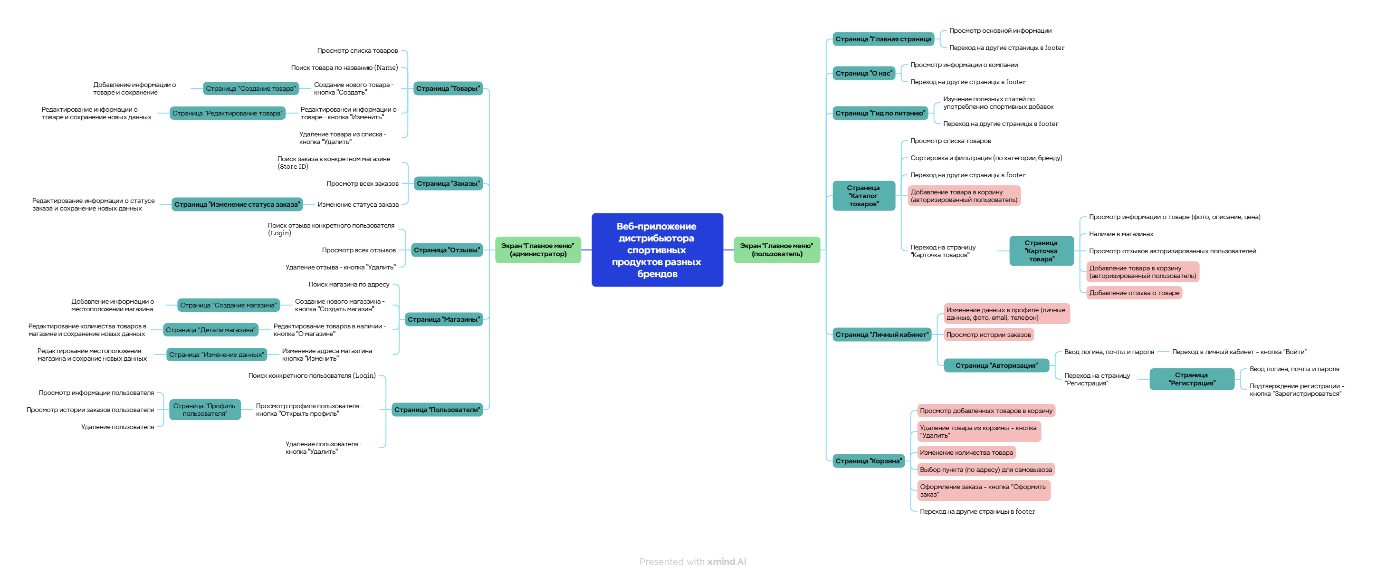


Рисунок 2 – Общая схема информационной системы веб-приложения

## 3.3 Проектирование интерфейса клиентской части

Проектирование интерфейса клиентской части веб-приложения осуществлялось в Figma с разрешением макетов 1440×900, что обеспечивает корректное отображение интерфейса на экранах настольных устройств. Основное внимание было уделено созданию интуитивно понятной навигации, визуальной целостности и соответствию тематике приложения.

Во всех страницах интерфейса верхняя часть содержит навигационные элементы, позволяющие пользователю быстро перейти к основным разделам платформы, таким как каталог товаров, личный кабинет, авторизация и справочные материалы. В нижней части каждой страницы расположен футер (подвал), содержащий контактную информацию для связи с технической поддержкой — телефон, адрес электронной почты — а также ссылки на вспомогательные страницы и повторяющиеся элементы навигации.

Главная страница служит входной точкой для пользователя: на ней размещены заголовок, краткое описание возможностей платформы и кнопки для перехода к регистрации или в каталог.

Страница «О нас» предоставляет информацию о компании, её целях, миссии и истории, сохраняя единый визуальный стиль.

Страница «Гид по питанию» ориентирована на пользователей, заинтересованных в здоровом образе жизни, и включает рекомендации по рациону и советы по составлению плана питания. Дизайн этой страницы ориентирован на тематику здорового образа жизни, с использованием соответствующей графики и цветовых решений.

Каждая страница разрабатывалась с учётом пользовательского удобства и функциональности, что обеспечивает комфортное взаимодействие с системой.

Результат проектирования статичных страниц («Главная», «О нас», «Гид по питанию») представлен на рис. 3.

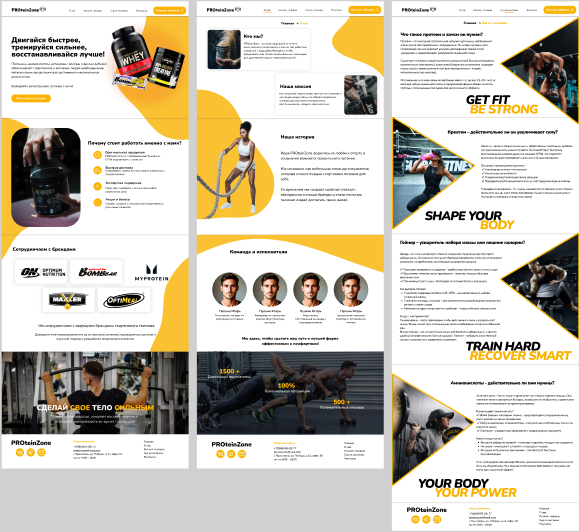


Рисунок 3 – Макеты статичных страниц

При переходе по кнопке «Личный кабинет» пользователь системы попадает на страницу для авторизации. Данный экран выполнен в фирменном стиле веб-приложения и содержит форму для ввода логина и пароля. Визуальное оформление страницы подчеркивает тематику здорового образа жизни и направлено на удобное и быстрое взаимодействие.

Если пользователь не зарегистрирован, он может перейти на страницу регистрации, ссылка на которую расположена под формой входа. На данной странице размещена форма для создания аккаунта, а также краткое описание преимуществ регистрации — возможность бронировать товары, оставлять отзывы и управлять личным профилем. Интерфейс страницы обеспечивает интуитивно понятную навигацию. Созданный макет страниц авторизации и регистрации представлен на рис. 4.

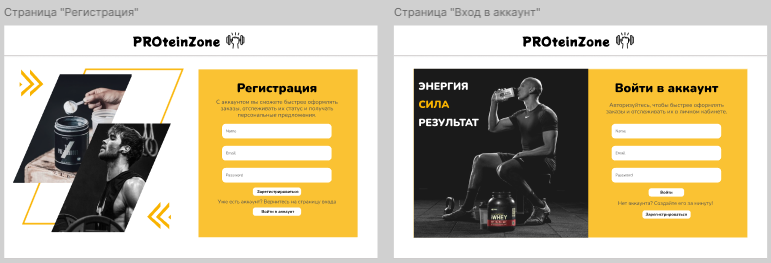


Рисунок 4 – Макеты страниц авторизации и регистрации

Рассмотрим следующие макеты для динамических страниц, которые будут формироваться на основе данных из БД. На рис. 5 представлены макеты для страниц, содержимое которых будет отображаться в соответствии с актуальной информацией, хранящейся в системе.

Каталог товаров представляет собой страницу, на которой пользователю доступен список всех представленных в системе спортивных продуктов. Каждый товар отображается в виде карточки с изображением, названием, ценой и кнопкой «Добавить в корзину». В левой части экрана размещены инструменты сортировки и фильтрации по категориям и брендам.

Карточка товара позволяет подробно ознакомиться с выбранным продуктом. Здесь отображаются название, цена, описание, наличие в магазинах и отзывы других пользователей. Авторизованный пользователь может добавить товар в корзину и оставить отзыв. Визуальное оформление направлено на удобство восприятия информации, с акцентом на ключевые характеристики продукта.

Страница личного кабинета предназначена для управления персональной информацией пользователя. В верхней части отображается фотография профиля и поля для редактирования личных данных, таких как имя, почта и номер телефона. Ниже размещён блок с историей заказов, где указана информация о товарах, стоимости и выбранных пунктах самовывоза.

Страница корзины служит для оформления покупки. Здесь отображаются добавленные товары с возможностью изменить их количество или удалить. Пользователь выбирает пункт самовывоза и может перейти к оформлению заказа через соответствующую кнопку. Итоговая стоимость автоматически рассчитывается.

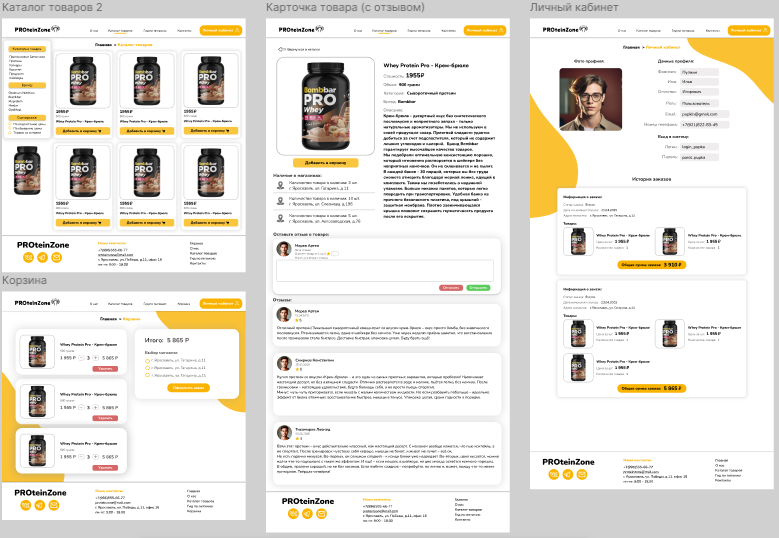


Рисунок 5 – Макеты динамических страниц пользователя

Макеты административной части сайта представляют собой функциональные интерфейсы, предназначенные для управления содержимым платформы.

Визуально они выполнены в едином стиле с пользовательской частью, но ориентированы на задачи администрирования. Основной экран администратора содержит навигационные элементы, позволяющие быстро переходить между разделами: управление товарами, заказами, отзывами, магазинами и пользователями. Каждая страница отображает актуальные данные, загружаемые из базы данных, и позволяет выполнять ключевые операции — например, добавление, редактирование и удаление товаров, изменение статуса заказов, работа с отзывами, управление магазинами и учетными записями пользователей. Интерфейс разработан таким образом, чтобы обеспечить удобную и эффективную работу администратора для поддержания актуальности информации на платформе.

Результат создания макетов для административной панели сайта представлен на рис. 6.

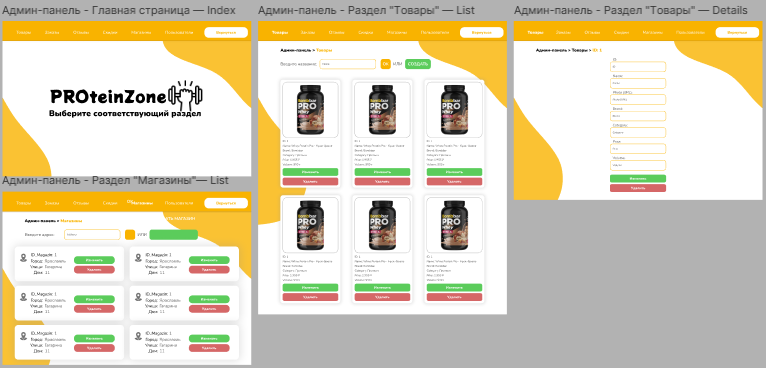


Рисунок 6 – Макеты для административной панели

В процессе создания макетов была сформирована визуальная основа веб-приложения, охватывающая как пользовательскую, так и административную части системы. Дизайны отражают основные пользовательские сценарии и обеспечивают понятную структуру взаимодействия с интерфейсом. Все элементы разработаны с учетом фирменного стиля и ориентированы на удобство и интуитивную навигацию.

# 4 Технологическая часть

## 4.1 Описание выбора инструментальных средств

Средства разработки программного обеспечения:

* Microsoft Visual Studio 2022;
* Visual Studio Code;
* .NET Framework SDK.

Системы управления базами данных (СУБД):

* PostgreSQL.

Дополнительно для разработки веб-приложения использовался фреймворк ASP .NET Core с реализацией паттерна MVC и следующими NuGet пакетами:

* Microsoft.EntityFrameworkCore.Tools.
* Npgsql.EntityFrameworkCore.PostgreSQL.

Дополнительно использовалась система контроля версий Git с помощью десктопного приложения GitHub Desktop.

## 4.2 Краткое описание разрабатываемой системы

Веб-приложение дистрибьютора спортивных продуктов разных брендов, предназначенное для оптимизации процессов взаимодействия между потребителями и продавцами спортивного питания.

Система предоставляет пользователю возможность просмотра каталога товаров, бронирования товаров в необходимой ему точке, оставления отзывов продуктам и просмотра корзины и предыдущих заказов. Товары также характеризуются брендом и категорией.

Система предоставляет администратору следующие возможности:

* Работа с товарами.
  + Просмотр товаров.
  + Создание нового товара.
  + Просмотр/изменение данных товара.
  + Удаление товара.
* Работа с заказами.
  + Просмотр заказов по бронированию.
  + Просмотр/изменение данных заказа.
* Работа с магазинами.
  + Просмотр магазинов.
  + Создание нового магазина.
  + Просмотр/изменение данных магазина (в том числе количества продукции на складе магазина).
  + Удаление магазина.
* Работа с отзывами.
  + Просмотр отзывов.
  + Удаление отзыва.
* Работа с пользователями.
  + Просмотр пользователей.
  + Добавление/изменение данных пользователя.
  + Удаление пользователя.

## 4.3 Описание интерфейса

Во всех страницах интерфейса верхняя часть содержит навигационные элементы, позволяющие пользователю быстро перейти к основным разделам платформы, таким как каталог товаров, личный кабинет, авторизация и справочные материалы. В нижней части каждой страницы расположен футер (подвал), содержащий контактную информацию для связи с технической поддержкой — телефон, адрес электронной почты — а также ссылки на вспомогательные страницы и повторяющиеся элементы навигации.

Главная страница служит входной точкой для пользователя: на ней размещены заголовок, краткое описание возможностей платформы и кнопки для перехода к регистрации или в каталог.

Страница «О нас» предоставляет информацию о компании, её целях, миссии и истории, сохраняя единый визуальный стиль.

Страница «Гид по питанию» ориентирована на пользователей, заинтересованных в здоровом образе жизни, и включает рекомендации по рациону и советы по составлению плана питания. Дизайн этой страницы ориентирован на тематику здорового образа жизни, с использованием соответствующей графики и цветовых решений.

При переходе по кнопке «Личный кабинет» пользователь системы попадает на страницу для авторизации. Данный экран выполнен в фирменном стиле веб-приложения и содержит форму для ввода логина и пароля. Визуальное оформление страницы подчеркивает тематику здорового образа жизни и направлено на удобное и быстрое взаимодействие.

Если пользователь не зарегистрирован, он может перейти на страницу регистрации, ссылка на которую расположена под формой входа. На данной странице размещена форма для создания аккаунта, а также краткое описание преимуществ регистрации — возможность бронировать товары, оставлять отзывы и управлять личным профилем. Интерфейс страницы обеспечивает интуитивно понятную навигацию.

Каталог товаров представляет собой страницу, на которой пользователю доступен список всех представленных в системе спортивных продуктов. Каждый товар отображается в виде карточки с изображением, названием, ценой и кнопкой «Добавить в корзину». В левой части экрана размещены инструменты сортировки и фильтрации по категориям и брендам.

Карточка товара позволяет подробно ознакомиться с выбранным продуктом. Здесь отображаются название, цена, описание, наличие в магазинах и отзывы других пользователей. Авторизованный пользователь может добавить товар в корзину и оставить отзыв. Визуальное оформление направлено на удобство восприятия информации, с акцентом на ключевые характеристики продукта.

Страница личного кабинета предназначена для управления персональной информацией пользователя. В верхней части отображается фотография профиля и поля для редактирования личных данных, таких как имя, почта и номер телефона. Ниже размещён блок с историей заказов, где указана информация о товарах, стоимости и выбранных пунктах самовывоза.

Страница корзины служит для оформления покупки. Здесь отображаются добавленные товары с возможностью изменить их количество или удалить. Пользователь выбирает пункт самовывоза и может перейти к оформлению заказа через соответствующую кнопку. Итоговая стоимость автоматически рассчитывается.

Основной экран администратора содержит навигационные элементы, позволяющие быстро переходить между разделами: управление товарами, заказами, отзывами, магазинами и пользователями. Каждая страница отображает актуальные данные, загружаемые из базы данных, и позволяет выполнять ключевые операции — например, добавление, редактирование и удаление товаров, изменение статуса заказов, работа с отзывами, управление магазинами и учетными записями пользователей. Интерфейс разработан таким образом, чтобы обеспечить удобную и эффективную работу администратора для поддержания актуальности информации на платформе.

## 4.4 Особенности разработки

Особенности разработки разрабатываемой системы:

* Архитектура системы. Для реализации системы использовался фреймворк разработки веб-приложений ASP .NET Core с применением языка программирования C# и с реализацией паттерна MVC.
* Технологический стек.
  + Frontend: HTML5, CSS3, JavaScript.
  + Backend: C# (ASP .NET Core MVC).
  + База данных: PostgreSQL.
  + API: RESTful.
* Модульность и расширяемость. Система разрабатывается с учетом возможности добавления новых функций и интеграции с внешними системами; модули разделены по функциональным направлениям.
* Интерфейс и пользовательский опыт. Особое внимание уделяется удобству интерфейса: простая навигация, отзывчивый дизайн, быстрый отклик системы. Для администраторов реализованы отдельные панели управления с интуитивным интерфейсом, обеспечивающим эффективность работы.
* Обработка данных и бизнес-логика реализована в соответствии с требованиями системы.
* Безопасность и защита данных. Используется контроль доступа по ролям (пользователь, администратор).

Коды важных программных модулей представлены в приложении В.

## 4.5 Особенности эксплуатации

Особенности эксплуатации разрабатываемой системы:

* Высокая доступность и надежность. Приложение должно обеспечивать работу 24/7 с минимальными простоями.
* Масштабируемость и производительность. Система должна быть способна обрабатывать растущее количество пользователей и данных, обеспечивая быстрый доступ при взаимодействии с большим объемом данных.
* Безопасность данных и транзакций. Система обеспечивает защиту личных данных пользователей от несанкционированного доступа. Система использует механику разделения функционала от роли пользователя.
* Удобство и интуитивность интерфейса. Пользователи должны легко ориентироваться в каталоге, бронировать товары и оставлять отзывы. Администраторы должны удобно управлять товарами, заказами, магазинами, отзывами и пользователями через понятную панель управления.
* Обеспечение актуальности информации. Автоматическая синхронизация наличия товаров на складах магазинов, обновление цен и характеристик товаров в реальном времени для предотвращения ошибок при бронировании и заказах.
* Обновление и сопровождение системы. Плановое обновление функциональности, исправление ошибок и мониторинг состояния системы для поддержания ее эффективности и безопасности.
* Обработка ошибок и пользовательский опыт. Обеспечение информативных сообщений об ошибках, быстрой реакции системы на сбой и минимизации потери данных при возникновении непредвиденных ситуаций.

# 5 Заключение

В рамках данного проекта была успешно разработана и реализована современная веб-система для дистрибьютора спортивных продуктов, объединяющая функции интернет-магазина, информационного портала и сервиса предварительного заказа. Использование технологий C# и ASP.NET Core обеспечило высокую производительность, надежность и масштабируемость системы, а архитектура на базе паттерна MVC позволила удобно разделить бизнес-логику, интерфейс и работу с данными, что значительно облегчает дальнейшее развитие и поддержку проекта.

Разработанный интерфейс, созданный с учетом современных требований к дизайну и удобству пользователя, гарантирует интуитивно понятное взаимодействие и адаптивность на различных устройствах. Реализованные функции поиска, фильтрации товаров, онлайн-бронирования, личного кабинета и административной панели создают полноценную платформу для эффективного взаимодействия между потребителями и продавцами спортивных товаров.

Результатом работы является комплексное решение, которое значительно упрощает процессы поиска, выбора и заказа спортивной продукции, повышает уровень прозрачности и доверия со стороны пользователей, а также расширяет возможности бизнеса по дистрибуции и продажам. Реализованная система отвечает поставленным задачам и может служить мощной базой для дальнейшего роста и развития компании в сфере спортивных товаров.

# Приложение А. Ссылки

## А.1 Ссылка на репозиторий в GitHub

<https://github.com/Mordahaoff/Malyshev_Project>

## А.2 Ссылка на проект в Figma

<https://www.figma.com/design/2BPwor4lfCNe2YSz5auNsw/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82-%D0%9C%D0%B0%D0%BB%D1%8B%D1%88%D0%B5%D0%B2?node-id=0-1&t=ssbXK49G0GIfc6Wz-1>

# Приложение Б. Листинги front-end

## Б.1 Код Login.cshtml

@model UserAuthModel

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROteinZone - Вход</title>

<link rel="stylesheet" href="~/css/login.css">

</head>

<body>

<header>

<div class="header-container">

<h1 class="logo">PROteinZone</h1>

<img src="~/img/icon/icon\_logo.png" alt="Иконка" class="logo-icon">

</div>

<div class="header-line"></div>

</header>

<div class="container">

<div class="left">

<h2>ЭНЕРГИЯ <br> <span>СИЛА</span> <br> РЕЗУЛЬТАТ</h2>

<img src="~/img/open-sport.png" alt="Спортсмен" class="athlete-img">

</div>

<div class="right">

<h2>Войти в аккаунт</h2>

<p>Авторизуйтесь, чтобы быстрее оформлять заказы и отслеживать их в личном кабинете.</p>

<form asp-controller="Auth" asp-action="Login" method="POST">

<input type="text" class="input-field" placeholder="Login" asp-for="Login" required>

<input type="email" class="input-field" placeholder="Email" asp-for="Email" required>

<input type="password" class="input-field" placeholder="Password" minlength="5" asp-for="Password" required>

<button type="submit" class="btn login-btn">Войти</button>

</form>

<div class="register-actions">

<p class="register">Нет аккаунта? Создайте его за минуту!</p>

<button onclick="window.location.href='/Auth/Register/'" class="btn register-btn">Зарегистрироваться</button>

</div>

<p class="guest-link"><a asp-controller="Home" asp-action="Index">Продолжить, как гость</a></p>

</div>

</div>

</body>

</html>

## Б.2 Код Catalog/Products.cshtml

@model CatalogModel

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROteinZone - Каталог товаров</title>

<link rel="stylesheet" href="~/css/catalog.css">

</head>

<body>

<header>

<div class="header-container">

<h1 class="logo">PROteinZone</h1>

<img src="~/img/icon/icon\_logo.png" alt="Иконка" class="logo-icon">

</div>

<nav>

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li class="active"><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="NutritionGuide">Гид по питанию</a></li>

<li><a asp-controller="Order" asp-action="Cart">Корзина</a></li>

<li class="cabinet-btn">

<a asp-controller="User" asp-action="Profile">

Личный кабинет

<img src="~/img/icon/icon\_cabinet.png" alt="Личный кабинет">

</a>

</li>

</ul>

</nav>

</header>

<div class="header-line"></div>

<img src="~/img/vector/vector\_basket1.png" alt="Фон вектора" class="vector-basket1">

<div class="breadcrumbs-container">

<div class="breadcrumbs">

<a asp-controller="Home" asp-action="Index">Главная</a>

<span class="arrow"> > </span>

<span class="current">Каталог товаров</span>

</div>

</div>

<div class="catalog-container">

<section class="filters">

<div class="filter-section orange-header">

<h3>Категория товара:</h3>

<ul>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="1"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Протеиновые

батончики</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="2"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Протеин</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="3"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Гейнеры</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="4"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Креатин</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="5"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Предтрен</a></li>

<li> <a asp-controller="Catalog" asp-action="Products" asp-route-categoryId="6"

asp-route-brandId="@Model.BrandId" asp-route-sortId="@Model.SortId">Шейкеры</a></li>

</ul>

</div>

<div class="filter-section orange-header">

<h3>Бренд:</h3>

<ul>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="1"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="@Model.SortId">Optimum

Nutrition</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="2"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="@Model.SortId">Bombbar</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="3"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="@Model.SortId">Myprotein</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="4"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="@Model.SortId">Maxler</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="5"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="@Model.SortId">OptiMeal</a></li>

</ul>

</div>

<div class="filter-section orange-header">

<h3>Сортировка:</h3>

<ul class="radio-sort">

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="@Model.BrandId"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="1">По возрастанию цены</a></li>

<li><a asp-controller="Catalog" asp-action="Products" asp-route-brandId="@Model.BrandId"

asp-route-categoryId="@Model.CategoryId" asp-route-sortId="2">По убыванию цены</a></li>

</ul>

</div>

<div class="filter-section orange-header">

<h3><a asp-controller="Catalog" asp-action="Products">Сбросить</a></h3>

</div>

</section>

<div class="products-section">

<div class="no-category-message">Категория: "@Model.CategoryName"</div>

<div class="no-category-message">Бренд: "@Model.BrandName"</div>

<div class="no-category-message">Сортировка: "@Model.SortName"</div>

<section class="products">

@foreach (var product in Model.Products)

{

<div class="product-card">

<a asp-controller="Product" asp-action="Details" asp-route-id="@product.IdProduct">

<img src="@product.Photo" alt="@product.Name" class="product-image">

<div class="product-price">@product.Price ₽</div>

<div class="product-weight">@product.Volume @product.GetUnits()</div>

<div class="product-name">@product.Name</div>

</a>

<a asp-controller="Order" asp-action="AddProductToCart"

asp-route-id="@product.IdProduct"

class="btn add-to-cart-btn"

data-product-id="@product.IdProduct"> <!-- Добавьте data-атрибут -->

Добавить в корзину

</a>

</div>

}

</section>

</div>

</div>

<footer class="footer">

<div class="footer-container">

<div class="footer-content">

<div class="footer-brand">

<h2 class="footer-logo">PROteinZone</h2>

<div class="social-links">

<a href="#" class="social-circle"><img src="~/img/icon/telegram.png" alt="Telegram"></a>

<a href="#" class="social-circle"><img src="~/img/icon/vk.png" alt="VK"></a>

<a href="#" class="social-circle"><img src="~/img/icon/email.png" alt="Email"></a>

</div>

</div>

<div class="footer-contacts">

<h3>Наши контакты:</h3>

<p>+7(996)555-66-77</p>

<p>proteinzone@mail.com</p>

<p>г. Ярославль, ул. Победы, д.11, офис 16</p>

<p>пн-пт: 9:00 - 18:00</p>

</div>

<div class="footer-links">

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="Guide">Гид по питанию</a></li>

</ul>

</div>

</div>

<div class="footer-copyright">

<p>© 2025 PROteinZone. Все права защищены.</p>

</div>

</div>

</footer>

<script>

document.addEventListener('DOMContentLoaded', () => {

const buttons = document.querySelectorAll('.add-to-cart-btn');

const updateButtonState = (button, isAdded) => {

button.textContent = isAdded ? 'Товар добавлен в корзину' : 'Добавить в корзину';

button.classList.toggle('added', isAdded);

};

const checkCartStatus = async (productId) => {

try {

const response = await fetch(`/Order/IsProductInCart?productId=${productId}`);

if (!response.ok) throw new Error('Ошибка сети');

return await response.json();

} catch (error) {

console.error('Ошибка при проверке корзины:', error);

return false;

}

};

const initializeButtons = async () => {

if (!buttons.length) return;

// Проверяем статус для каждой кнопки

for (const button of buttons) {

const productId = button.dataset.productId;

const isInCart = await checkCartStatus(productId);

updateButtonState(button, isInCart);

}

};

// Добавляем обработчики для всех кнопок

buttons.forEach(button => {

button.addEventListener('click', async (e) => {

e.preventDefault();

const productId = button.dataset.productId;

const isCurrentlyInCart = button.textContent.includes('добавлен');

try {

const response = await fetch(

isCurrentlyInCart

? `/Order/RemoveProductFromCartAjax?productId=${productId}`

: `/Order/AddProductToCartAjax?productId=${productId}`,

{

method: 'POST',

headers: { 'Content-Type': 'application/json' }

}

);

if (response.ok) {

updateButtonState(button, !isCurrentlyInCart);

// Обновляем состояние всех кнопок этого товара на странице

document.querySelectorAll(`.add-to-cart-btn[data-product-id="${productId}"]`).forEach(btn => {

updateButtonState(btn, !isCurrentlyInCart);

});

} else {

console.error('Ошибка при обновлении корзины');

}

} catch (error) {

console.error('Ошибка:', error);

}

});

});

initializeButtons();

});

</script>

</body>

</html>

## Б.3 Код Profile.cshtml

@model User;

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROteinZone - Мой профиль</title>

<link rel="stylesheet" href="~/css/cabinet.css">

</head>

<body>

<header>

<div class="header-container">

<h1 class="logo">PROteinZone</h1>

<img src="~/img/icon/icon\_logo.png" alt="Иконка" class="logo-icon">

</div>

<nav>

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="NutritionGuide">Гид по питанию</a></li>

<li><a asp-controller="Order" asp-action="Cart">Корзина</a></li>

<li class="cabinet-btn">

<a asp-controller="User" asp-action="Profile">

Личный кабинет

<img src="~/img/icon/icon\_cabinet.png" alt="Личный кабинет">

</a>

</li>

</ul>

</nav>

</header>

<div class="header-line"></div>

<div class="breadcrumbs-container">

<div class="breadcrumbs">

<a asp-controller="Home" asp-action="Index">Главная</a>

<span class="arrow"> > </span>

<span class="current">Личный кабинет</span>

</div>

</div>

<main class="cabinet-container">

<form asp-controller="User" asp-action="Profile" method="post">

<input type="hidden" asp-for="IdUser" value="@Model.IdUser" />

<input type="hidden" asp-for="Login" value="@Model.Login" />

<input type="hidden" asp-for="Password" value="@Model.Password" />

<input type="hidden" asp-for="RoleId" value="@Model.RoleId" />

<input type="hidden" asp-for="Gender" value="@Model.Gender" />

<img src="~/img/vector/vector\_cabinet.png" alt="Фон вектора" class="vector-cabinet">

<div class="profile-section">

<div class="profile-photo">

<h3>Фото профиля:</h3>

<img src="@Model.Photo" alt="Фото профиля">

<div class="profile-buttons">

<button type="submit" class="btn save-btn">Сохранить изменения</button>

<a asp-controller="Auth" asp-action="Logout" class="btn logout-btn">Выйти из аккаунта</a>

@if (Model.RoleId == 2) // Проверяем, является ли пользователь администратором (ID роли = 2)

{

<a asp-controller="Menu" asp-action="Index" asp-area="Admin" class="btn admin-btn">Перейти на админ-панель</a>

}

</div>

</div>

<div class="profile-info">

<h3>Данные профиля:</h3>

<div class="input-group">

<label asp-for="Photo">Фото (URL):</label>

<input asp-for="Photo" type="text" value="@Model.Photo" />

</div>

<div class="input-group">

<label asp-for="Surname">Фамилия:</label>

<input asp-for="Surname" type="text" value="@Model.Surname" />

</div>

<div class="input-group">

<label asp-for="Name">Имя:</label>

<input asp-for="Name" type="text" value="@Model.Name">

</div>

<div class="input-group">

<label asp-for="Patronymic">Отчество:</label>

<input asp-for="Patronymic" type="text" value="@Model.Patronymic">

</div>

<div class="input-group">

<label>Роль:</label>

<input type="text" value="@Model.Role.Name" disabled>

</div>

<div class="input-group">

<label asp-for="Email">Email:</label>

<input asp-for="Email" type="email" value="@Model.Email">

</div>

<div class="input-group">

<label asp-for="Telephone">Номер телефона:</label>

<input asp-for="Telephone" type="tel" value="@Model.Telephone" placeholder="+7 XXX-XXX-XX-XX">

</div>

<h3>Вход в систему:</h3>

<div class="input-group">

<label>Логин:</label>

<input type="text" value="@Model.Login" disabled>

</div>

<div class="input-group">

<label>Пароль:</label>

<input type="text" value="@Model.Password" disabled>

</div>

</div>

</div>

</form>

</main>

<img src="~/img/vector/vector\_cabinet2.png" alt="Фон вектора" class="vector-cabinet2">

<section class="order-history">

<h2 class="order-title">История заказов</h2>

@if (Model.Orders.Any())

{

@foreach (var order in Model.Orders)

{

<div class="order-card">

<div class="order-info">

<h3>Информация о заказе:</h3>

<p><strong>Статус заказа:</strong> @order.StateOfOrder.Name</p>

<p><strong>Дата изменения статуса:</strong> @order.DateOfStatusChange</p>

<p><strong>Адрес магазина:</strong> г. @order.Store!.Address.City, ул. @order.Store!.Address.Street,

д.@order.Store!.Address.Building</p>

</div>

<h3>Товары:</h3>

<div class="order-products">

@foreach (var op in order.OrdersProducts)

{

<div class="product-item">

<a asp-controller="Product" asp-action="Details" asp-route-id="@op.ProductId">

<img src="@op.Product.Photo" alt="@op.Product.Name">

</a>

<div class="product-details">

<p class="product-name">@op.Product.Name</p>

<p>Цена за шт: <span class="price">@op.Product.Price ₽</span></p>

<p>Количество товара: <strong>@op.CountOfProduct</strong></p>

</div>

</div>

}

</div>

<div class="order-total-container">

<div class="order-total">

<span>Общая сумма заказа:</span>

<strong>@order.OrdersProducts.Sum(op => op.Product.Price \* op.CountOfProduct) ₽</strong>

</div>

</div>

</div>

}

}

else

{

<div class="empty-orders">

<p>У вас пока нет заказов</p>

</div>

}

</section>

<footer class="footer">

<div class="footer-container">

<div class="footer-content">

<div class="footer-brand">

<h2 class="footer-logo">PROteinZone</h2>

<div class="social-links">

<a href="#" class="social-circle"><img src="~/img/icon/telegram.png" alt="Telegram"></a>

<a href="#" class="social-circle"><img src="~/img/icon/vk.png" alt="VK"></a>

<a href="#" class="social-circle"><img src="~/img/icon/email.png" alt="Email"></a>

</div>

</div>

<div class="footer-contacts">

<h3>Наши контакты:</h3>

<p>+7(996)555-66-77</p>

<p>proteinzone@mail.com</p>

<p>г. Ярославль, ул. Победы, д.11, офис 16</p>

<p>пн-пт: 9:00 - 18:00</p>

</div>

<div class="footer-links">

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="Guide">Гид по питанию</a></li>

</ul>

</div>

</div>

<div class="footer-copyright">

<p>© 2025 PROteinZone. Все права защищены.</p>

</div>

</div>

</footer>

</body>

</html>

## Б.4 Код Cart.cshtml

@model CartModel;

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROteinZone - Корзина</title>

<link rel="stylesheet" href="~/css/basket.css">

</head>

<body>

<header>

<div class="header-container">

<h1 class="logo">PROteinZone</h1>

<img src="~/img/icon/icon\_logo.png" alt="Иконка" class="logo-icon">

</div>

<nav>

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="NutritionGuide">Гид по питанию</a></li>

<li class="active"><a asp-controller="Order" asp-action="Cart">Корзина</a></li>

<li class="cabinet-btn">

<a asp-controller="User" asp-action="Profile">

Личный кабинет

<img src="~/img/icon/icon\_cabinet.png" alt="Личный кабинет">

</a>

</li>

</ul>

</nav>

</header>

<div class="header-line"></div>

<div class="breadcrumbs-container">

<div class="breadcrumbs">

<a asp-controller="Home" asp-action="Index">Главная</a>

<span class="arrow"> > </span>

<span class="current">Корзина</span>

</div>

</div>

@if (Model.Order == null || !Model.Order.OrdersProducts.Any())

{

<div class="empty-cart">

<p>Ваша корзина пуста</p>

</div>

}

else

{

<form method="post" class="basket-container">

<input type="hidden" asp-for="Order!.IdOrder" value="@Model.Order.IdOrder" />

<input type="hidden" asp-for="Order!.UserId" value="@Model.Order.UserId" />

<input type="hidden" asp-for="Order!.StateOfOrderId" value="@Model.Order.StateOfOrderId" />

<div class="basket-products">

@for (int i = 0; i < Model.Products!.Count; i++)

{

var op = Model.Products[i];

<input type="hidden" asp-for="Products[i].IdOrdersProducts" value="@op.IdOrdersProducts" />

<input type="hidden" asp-for="Products[i].ProductId" value="@op.ProductId" />

<input type="hidden" asp-for="Products[i].OrderId" value="@op.OrderId" />

<div class="basket-item">

<img src="@op.Product.Photo" alt="@op.Product.Name" class="basket-img">

<div class="basket-info">

<h3>@op.Product.Name</h3>

<p>@op.Product.Volume @op.Product.GetUnits()</p>

<div class="basket-price-controls">

<span class="single-price" id="price@(i)">@op.Product.Price ₽</span>

<div class="counter">

<button type="button" class="counter-btn" onclick="decreaseCount('@(i)')">-</button>

<input type="number" asp-for="Products[i].CountOfProduct" id="countOfProduct@(i)"

value="@op.CountOfProduct" class="count" readonly />

<button type="button" class="counter-btn" onclick="increaseCount('@(i)')">+</button>

</div>

<span class="item-total" id="item-total@(i)">@(op.Product.Price\* op.CountOfProduct) ₽</span>

</div>

<a class="delete-btn" asp-controller="Order" asp-action="RemoveProductFromCart"

asp-route-id="@op.ProductId">Удалить</a>

</div>

</div>

}

</div>

<div class="basket-summary">

<h3>Итого: <span id="cart-total">@(Model.Order.OrdersProducts.Sum(op => op.Product.Price \*

op.CountOfProduct)) ₽</span></h3>

<div class="store-selection">

<p>Выбор магазина:</p>

@foreach (var store in Model.Stores)

{

<label>

<input asp-for="Order!.StoreId" value="@store.IdStore" type="radio" required> г.

@store.Address.City, ул. @store.Address.Street,

д.@store.Address.Building

</label>

if (Model.Stores.Last() != store)

{

<br>

}

}

</div>

<button type="submit" class="order-btn">Оформить заказ</button>

</div>

</form>

}

<footer class="footer">

<div class="footer-container">

<div class="footer-content">

<div class="footer-brand">

<h2 class="footer-logo">PROteinZone</h2>

<div class="social-links">

<a href="#" class="social-circle"><img src="~/img/icon/telegram.png" alt="Telegram"></a>

<a href="#" class="social-circle"><img src="~/img/icon/vk.png" alt="VK"></a>

<a href="#" class="social-circle"><img src="~/img/icon/email.png" alt="Email"></a>

</div>

</div>

<div class="footer-contacts">

<h3>Наши контакты:</h3>

<p>+7(996)555-66-77</p>

<p>proteinzone@mail.com</p>

<p>г. Ярославль, ул. Победы, д.11, офис 16</p>

<p>пн-пт: 9:00 - 18:00</p>

</div>

<div class="footer-links">

<ul>

<li><a asp-controller="Home" asp-action="Index">Главная</a></li>

<li><a asp-controller="Home" asp-action="About">О нас</a></li>

<li><a asp-controller="Catalog" asp-action="Products">Каталог товаров</a></li>

<li><a asp-controller="Home" asp-action="Guide">Гид по питанию</a></li>

</ul>

</div>

</div>

<div class="footer-copyright">

<p>© 2025 PROteinZone. Все права защищены.</p>

</div>

</div>

</footer>

<script>

function decreaseCount(elementId) {

const input = document.getElementById('countOfProduct' + elementId);

if (!input) return;

let currentValue = parseInt(input.value, 10);

if (isNaN(currentValue)) currentValue = 0;

if (currentValue > 1) {

currentValue -= 1;

input.value = currentValue;

// Получаем элементы для отображения суммы

const itemTotal = document.getElementById('item-total' + elementId);

const cartTotal = document.getElementById('cart-total');

const priceElement = document.getElementById('price' + elementId);

if (!itemTotal || !cartTotal || !priceElement) return;

// Предположим, что priceElement.value — строка вида "100 USD" или что-то подобное

const priceParts = priceElement.innerText.split(' ');

console.log(priceParts[0] + ' ' + priceParts[1]);

const pricePerItem = parseInt(priceParts[0], 10);

// Обновляем значения total

// Предположим, что itemTotal.value и cartTotal.value — строки вида "300 USD" или подобное

const itemTotalParts = itemTotal.innerText.split(' ');

console.log(itemTotalParts[0] + ' ' + itemTotalParts[1]);

const cartTotalParts = cartTotal.innerText.split(' ');

console.log(cartTotalParts[0] + ' ' + cartTotalParts[1]);

let itemTotalAmount = parseInt(itemTotalParts[0], 10);

let cartTotalAmount = parseInt(cartTotalParts[0], 10);

// Вычитаем цену за один товар

itemTotalAmount -= pricePerItem;

cartTotalAmount -= pricePerItem;

// Обновляем значения

itemTotal.innerText = itemTotalAmount + ' ' + itemTotalParts[1];

cartTotal.innerText = cartTotalAmount + ' ' + cartTotalParts[1];

console.log('pricePerItem = ' + pricePerItem + ' ' + priceParts[1]);

}

}

function increaseCount(elementId) {

const input = document.getElementById('countOfProduct' + elementId);

if (!input) return;

let currentValue = parseInt(input.value, 10);

if (isNaN(currentValue)) currentValue = 0;

currentValue += 1;

input.value = currentValue;

// Получаем элементы для отображения суммы

const itemTotal = document.getElementById('item-total' + elementId);

const cartTotal = document.getElementById('cart-total');

const priceElement = document.getElementById('price' + elementId);

if (!itemTotal || !cartTotal || !priceElement) return;

// Предположим, что priceElement.value — строка вида "100 USD" или что-то подобное

const priceParts = priceElement.innerText.split(' ');

console.log(priceParts[0] + ' ' + priceParts[1]);

const pricePerItem = parseInt(priceParts[0], 10);

// Обновляем значения total

// Предположим, что itemTotal.value и cartTotal.value — строки вида "300 USD" или подобное

const itemTotalParts = itemTotal.innerText.split(' ');

console.log(itemTotalParts[0] + ' ' + itemTotalParts[1]);

const cartTotalParts = cartTotal.innerText.split(' ');

console.log(cartTotalParts[0] + ' ' + cartTotalParts[1]);

let itemTotalAmount = parseInt(itemTotalParts[0], 10);

let cartTotalAmount = parseInt(cartTotalParts[0], 10);

// Прибавляем цену за один товар

itemTotalAmount += pricePerItem;

cartTotalAmount += pricePerItem;

// Обновляем значения

itemTotal.innerText = itemTotalAmount + ' ' + itemTotalParts[1];

cartTotal.innerText = cartTotalAmount + ' ' + cartTotalParts[1];

console.log('pricePerItem = ' + pricePerItem + ' ' + priceParts[1]);

}

</script>

</body>

</html>

## Б.5 Код Admin/Product/List.cshtml

@model IEnumerable<Malyshev\_Project.Models.Product>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>PROteinZone - Товары</title>

<link rel="stylesheet" href="~/css\_admin/product.css">

</head>

<body>

<header>

<nav>

<ul>

<li class="active"><a asp-controller="Product" asp-action="List" asp-area="Admin">Товары</a></li>

<li><a asp-controller="Order" asp-action="List" asp-area="Admin">Заказы</a></li>

<li><a asp-controller="Review" asp-action="List" asp-area="Admin">Отзывы</a></li>

<li><a asp-controller="Store" asp-action="List" asp-area="Admin">Магазины</a></li>

<li><a asp-controller="User" asp-action="List" asp-area="Admin">Пользователи</a></li>

<li class="back-btn">

<a asp-controller="Home" asp-action="Index" asp-area="">Вернуться</a>

</li>

</ul>

</nav>

</header>

<div class="header-line"></div>

<img src="~/img/vector/vector\_basket1.png" alt="Фон вектора" class="vector-basket1">

<img src="~/img/vector/vector\_basket2.png" alt="Фон вектора" class="vector-basket2">

<div class="breadcrumbs-search-container">

<div class="breadcrumbs-container">

<div class="breadcrumbs">

<a asp-controller="Menu" asp-action="Index">Админ-панель</a>

<span class="arrow"> > </span>

<span class="current">Товары</span>

</div>

</div>

<form method="get" class="search-container">

<div class="search-box">

<div class="search-row">

<label for="search-input">Введите название:</label>

<input type="text" id="search-input" name="productName" placeholder="Name"

value="@ViewData["Request"]" class="search-input">

<button type="submit" class="search-btn">OK</button>

<span class="search-or">ИЛИ</span>

<a asp-action="Create" class="create-btn">СОЗДАТЬ</a>

</div>

</div>

</form>

</div>

<div class="products">

@foreach (var item in Model)

{

<div class="product-card">

<a asp-controller="Product" asp-action="Details" asp-area="Admin" asp-route-id="@item.IdProduct"

class="product-link">

<img src="@item.Photo" alt="@item.Name" class="product-image" />

<p><strong>ID:</strong> @item.IdProduct</p>

<p><strong>Название:</strong> @item.Name</p>

<p><strong>Бренд:</strong> @item.Brand?.Name</p>

<p><strong>Категория:</strong> @item.Category?.Name</p>

<p><strong>Цена:</strong> @item.Price ₽</p>

<p><strong>Объем/кол-во:</strong> @item.Volume @item.GetUnits()</p>

</a>

<div class="card-buttons">

<a asp-action="Edit" asp-route-id="@item.IdProduct" class="btn edit-btn">Изменить</a>

<a asp-action="Delete" asp-route-id="@item.IdProduct" class="btn delete-btn">Удалить</a>

</div>

</div>

}

</div>

</body>

</html>

## Б.6 Код Admin/Store/Edit.cshtml

ВСТАВИТЬ КОД

# Приложение В. Скриншоты страниц

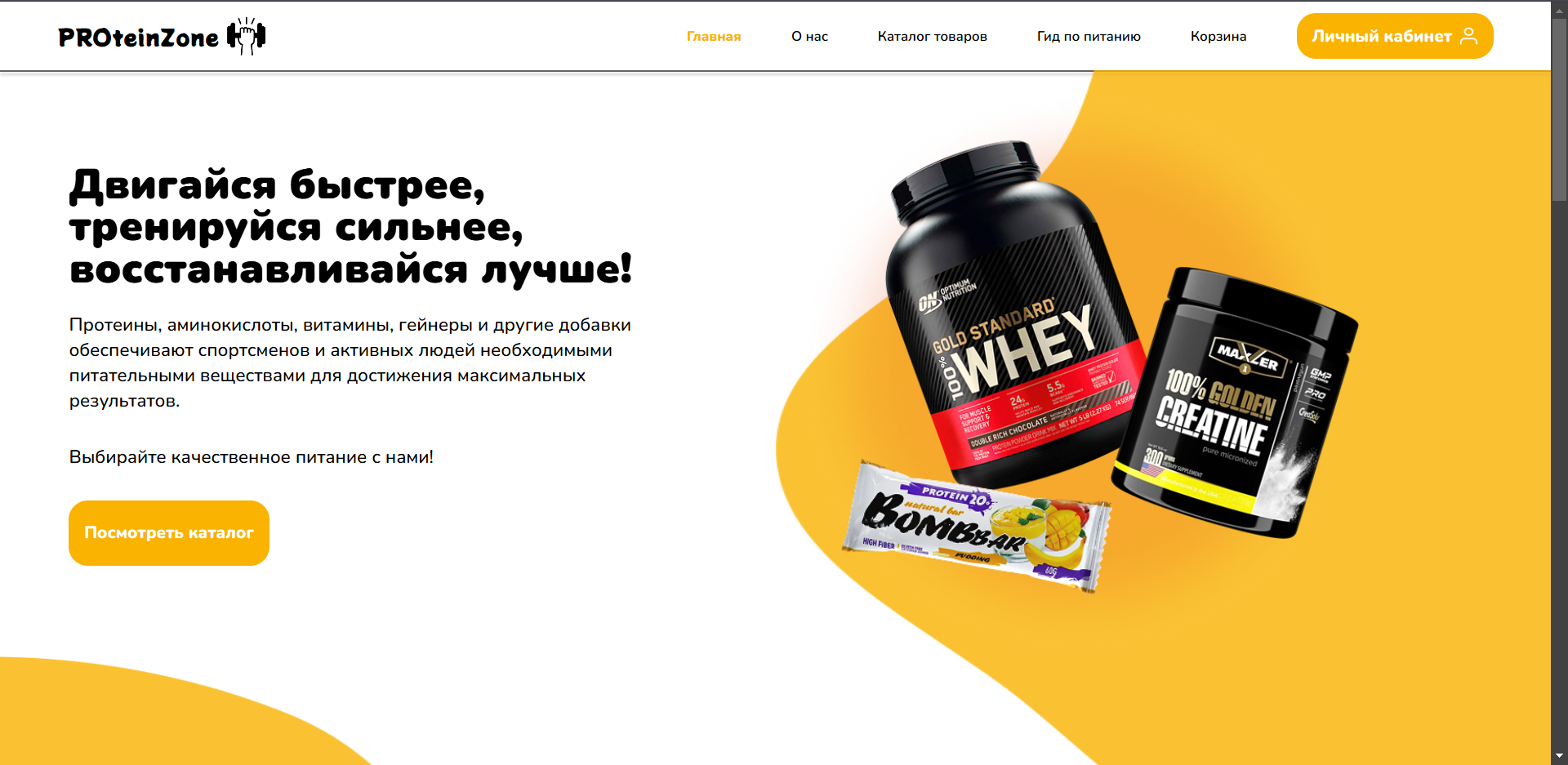


Рисунок В.1 – Главная страница

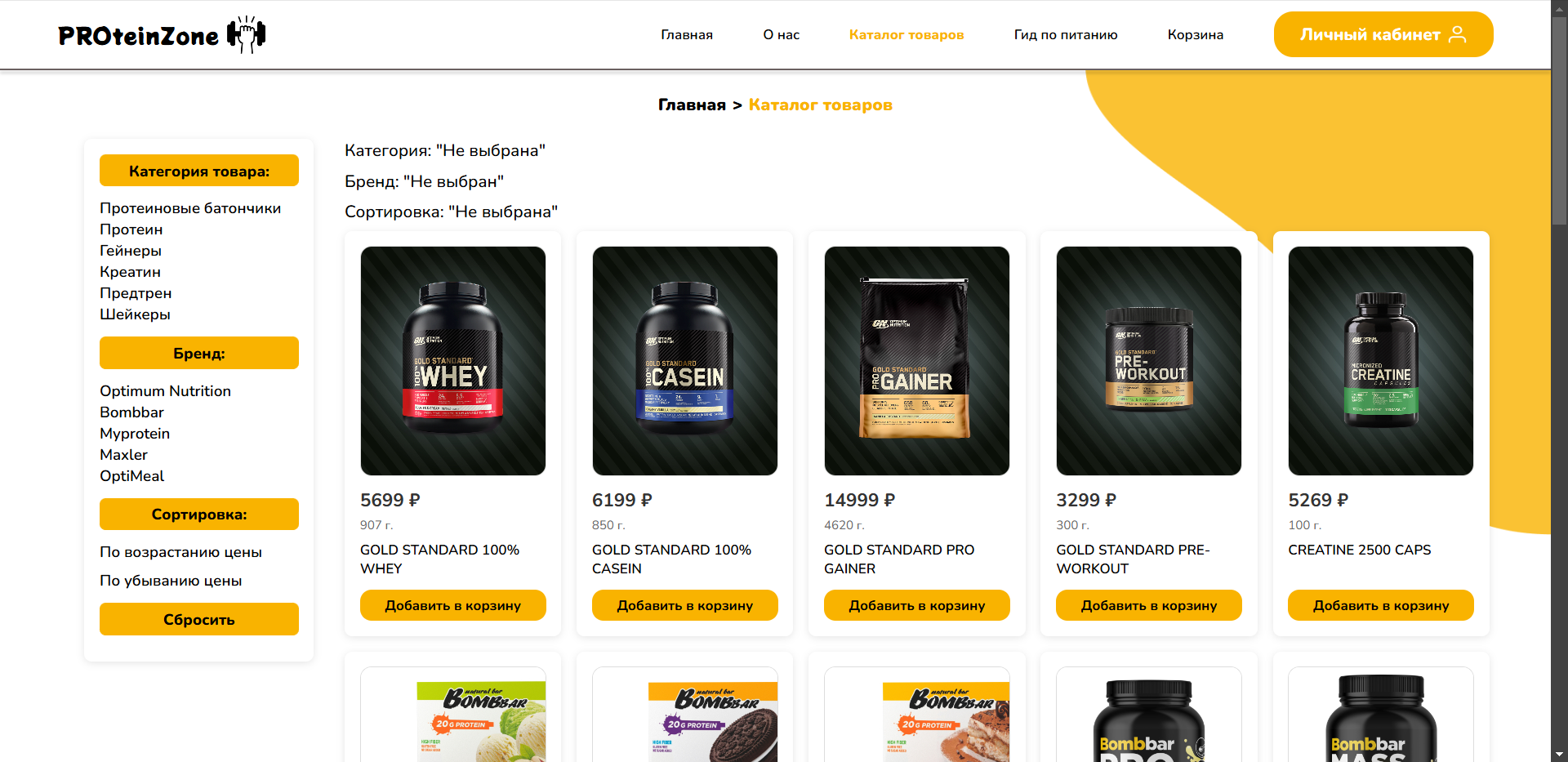


Рисунок В.2 – Каталог товаров

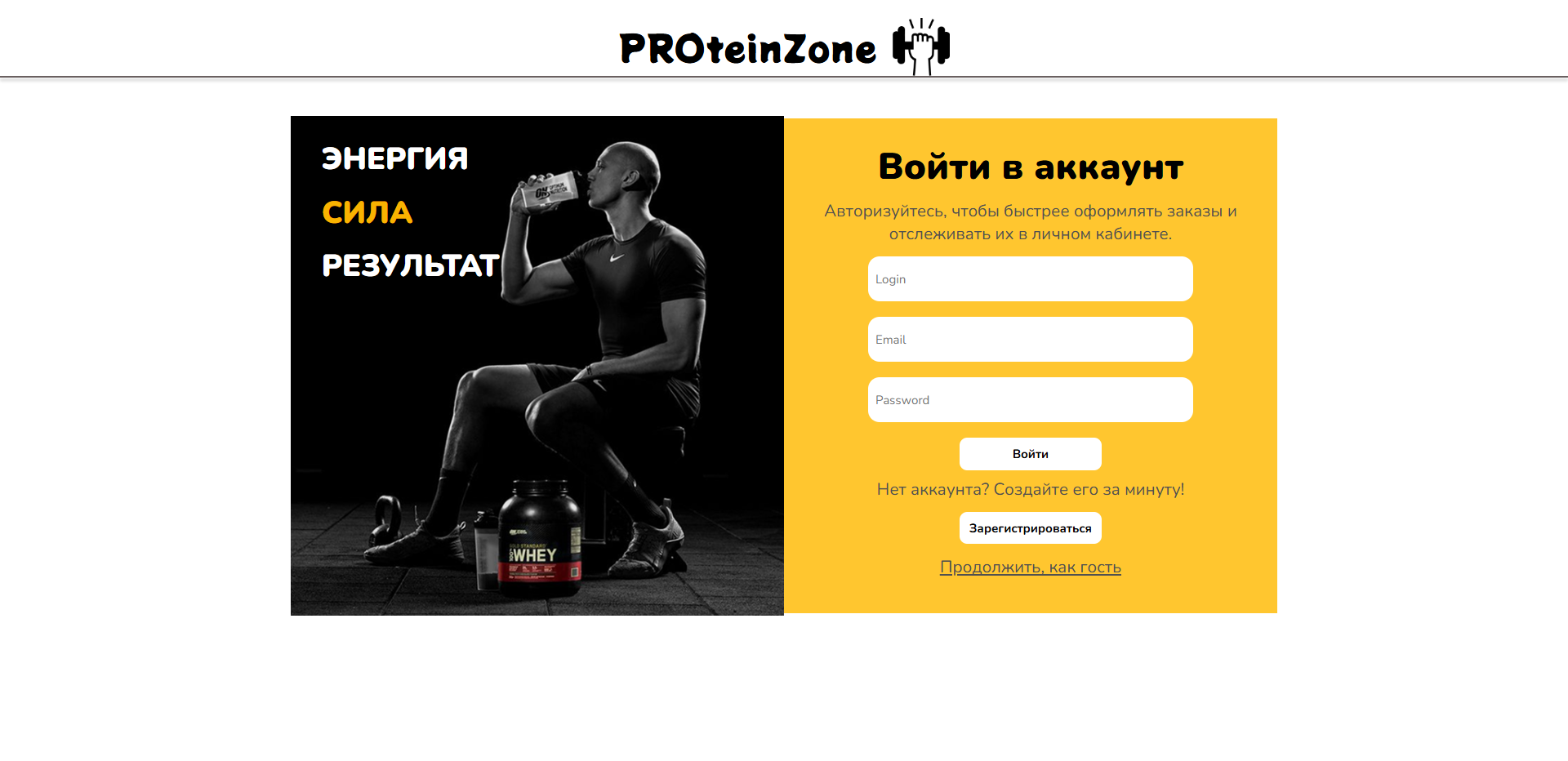


Рисунок В.3 – Страница авторизации

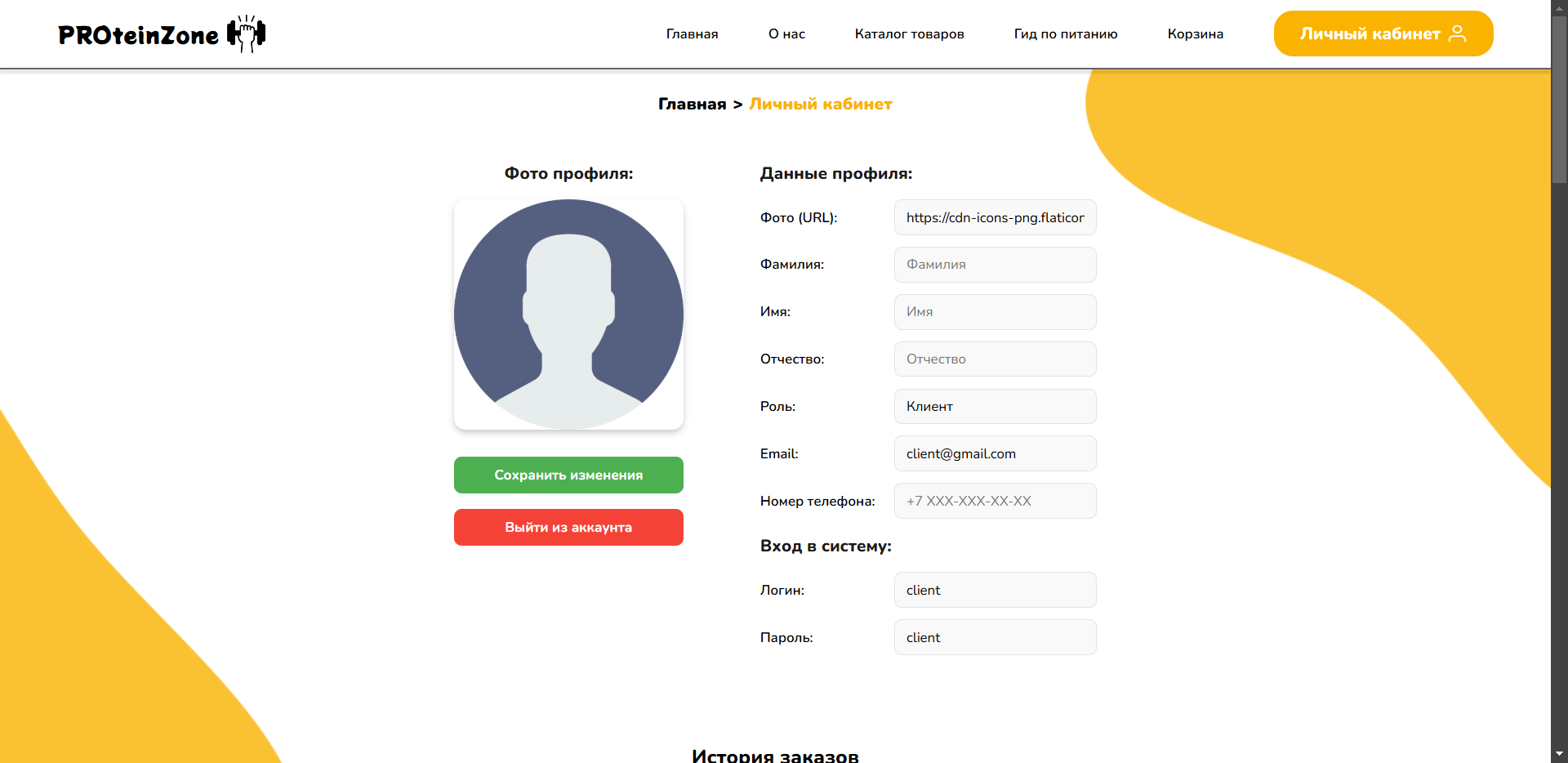


Рисунок В.4 – Профиль пользователя

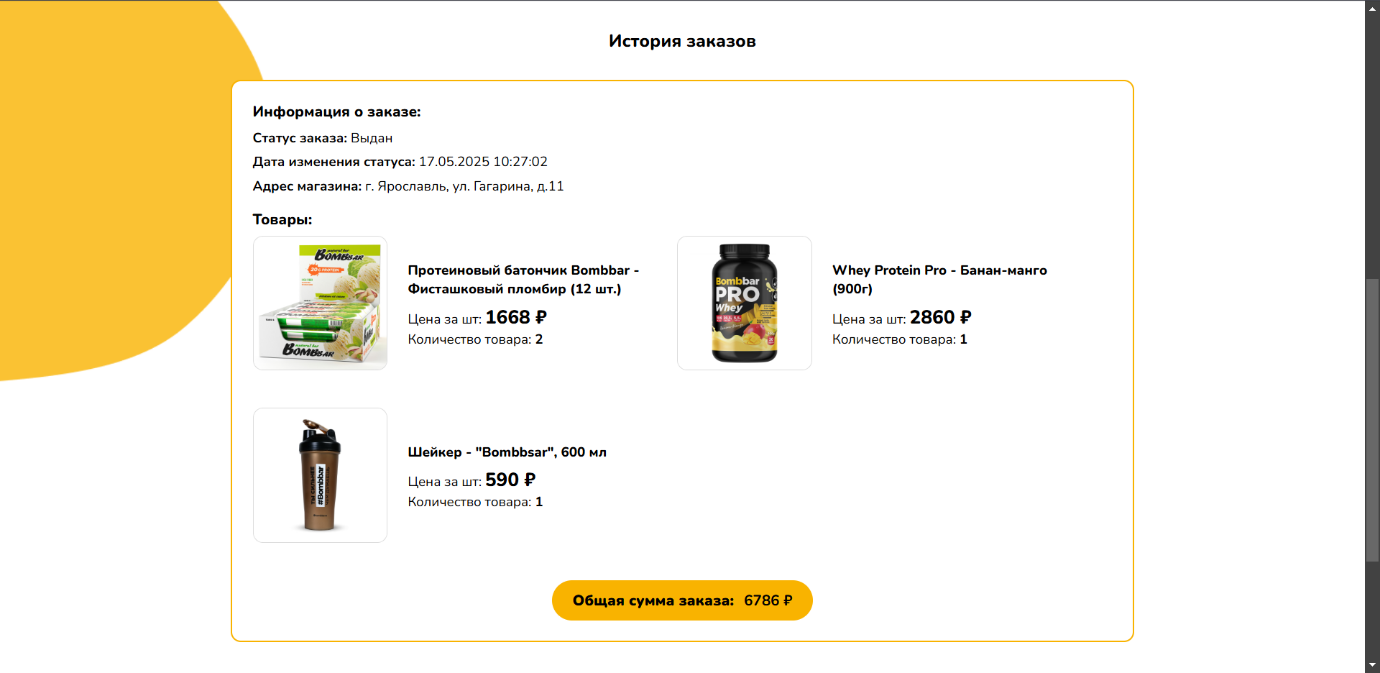


Рисунок В.5 – История заказов на профиле пользователя

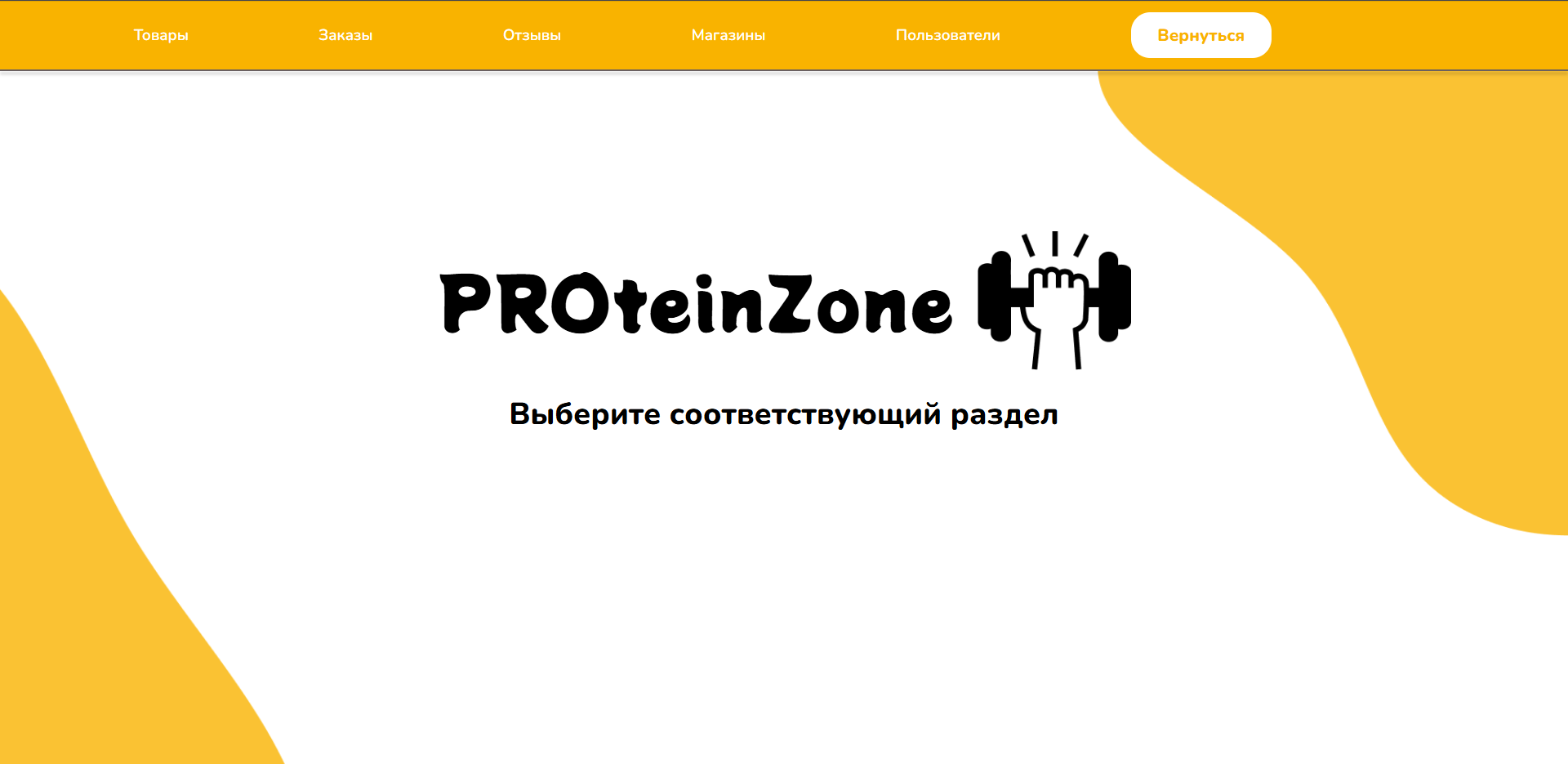


Рисунок В.6 – Админ-панель

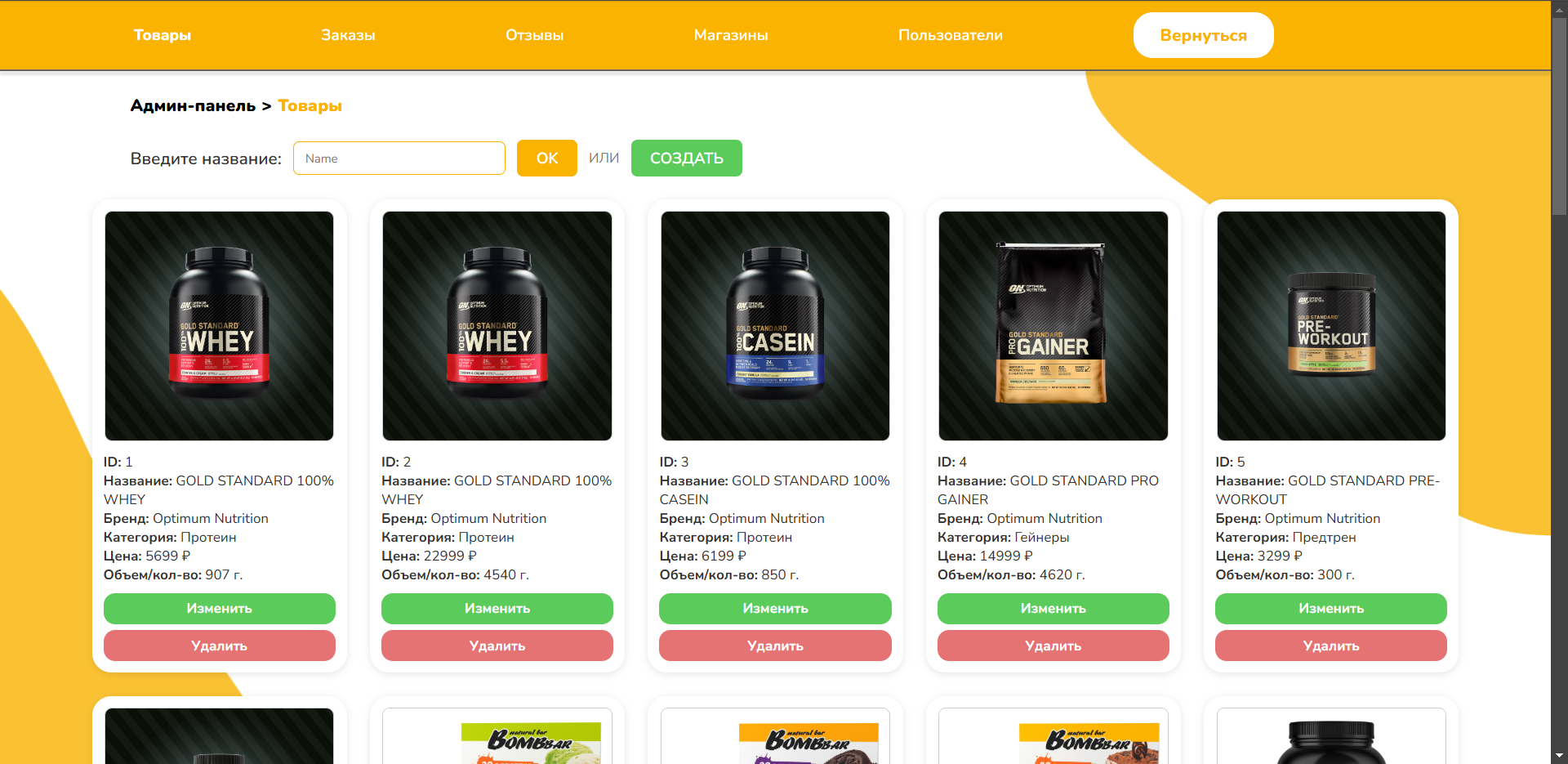


Рисунок В.7 – Раздел «Товары» в админ-панели

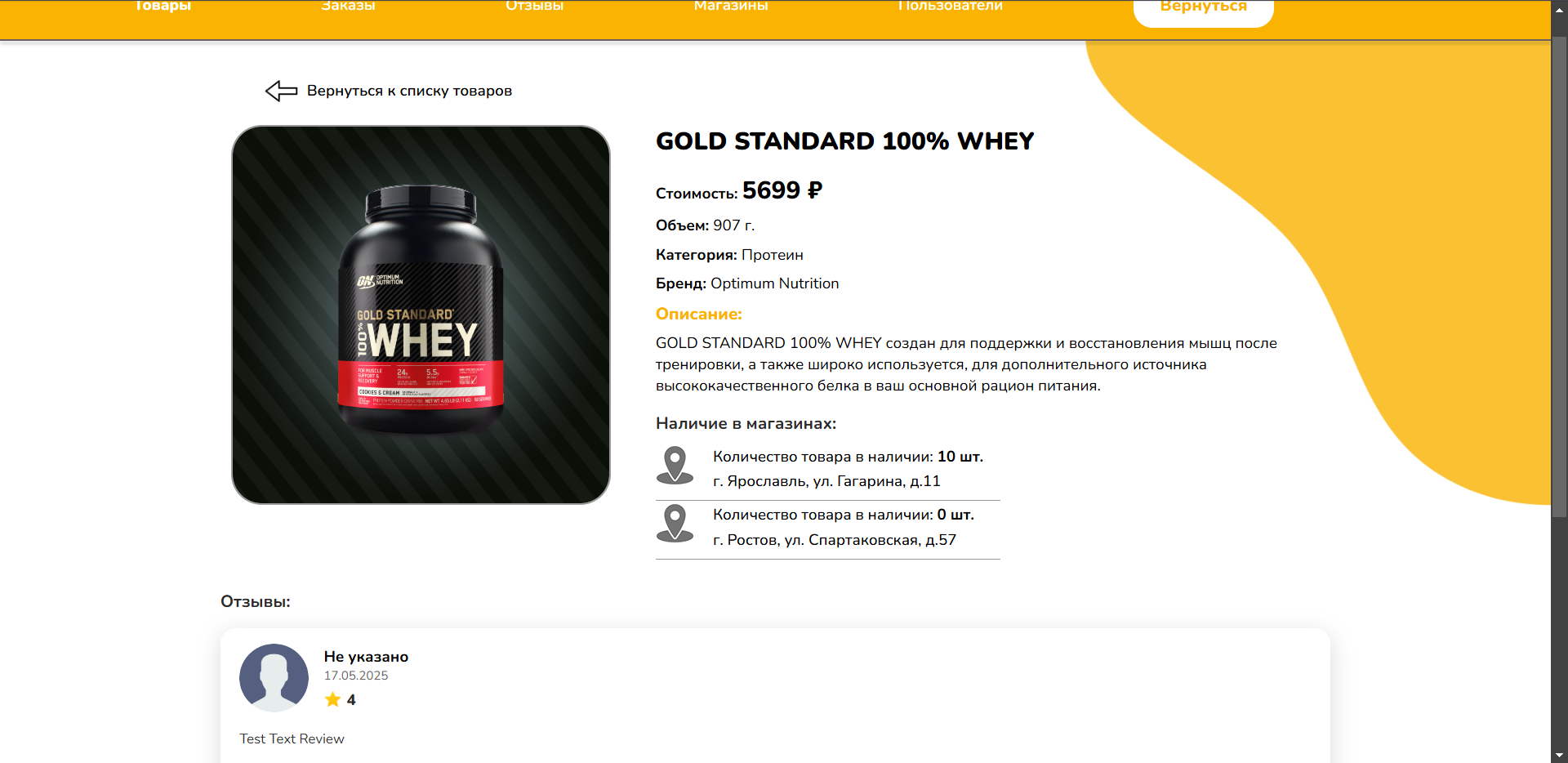


Рисунок В.8 – Просмотр информации о товаре

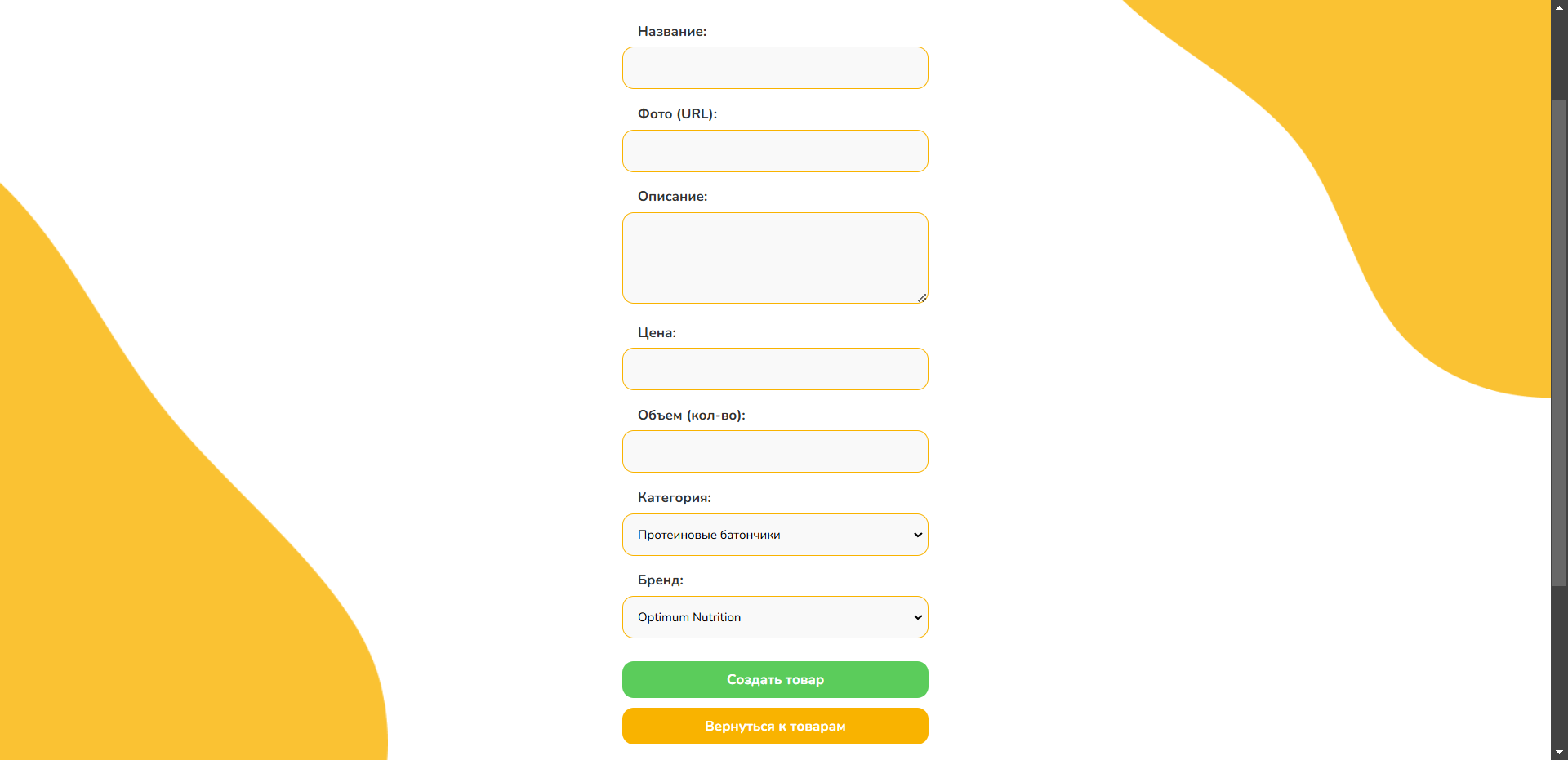


Рисунок В.9 – Страница для создания нового товара