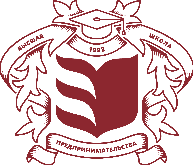
****

**Частное учреждение профессионального образования**

**«Высшая школа предпринимательства»**

**(ЧУПО «ВШП»)**

ДИПЛОМНЫЙ ПРОЕКТ

по основной образовательной программе  
среднего профессионального образования по специальности  
09.02.07 «Информационные системы и программирование» на тему:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| Выполнил студент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(фамилия, имя, отчество)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Руководитель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(ученая степень, фамилия, имя, отчество)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Рецензент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(ученая степень, фамилия, имя, отчество)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(должность)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(место работы)* |
| Председатель ГЭК | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(подпись)* | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(ученая степень, фамилия, имя, отчество)*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(прописью)* | Дата защиты: \_\_.\_\_.20\_\_г. |

Оглавление

[Введение 3](#_Toc199884570)

[Глава 1. Теоретическая часть 6](#_Toc199884571)

[**1.1 Изучение предметной области** 6](#_Toc199884572)

[**1.2 Анализ аналогов на рынке** 10](#_Toc199884573)

[**1.3 Выбор языка программирования и СУБД** 15](#_Toc199884574)

[Глава 2 Проектная часть 23](#_Toc199884575)

[**2.1. Разработка архитектуры базы данных** 23](#_Toc199884576)

[**2.2. Разработка интерфейса приложения** 30](#_Toc199884577)

[**2.3. Разработка функционала** 33](#_Toc199884578)

[Заключение 42](#_Toc199884579)

[Список источников 45](#_Toc199884580)

[Приложение 1 46](#_Toc199884581)

[Приложение 2 47](#_Toc199884582)

[Приложение 3 48](#_Toc199884583)

[Приложение 4 49](#_Toc199884584)

# **Введение**

**Актуальность**

В условиях стремительного развития цифровых технологий и увеличения популярности электронной коммерции, разработка каталога спортивной обуви приобретает особую актуальность. Современные потребители всё чаще предпочитают онлайн-покупки, что требует от производителей и ритейлеров создания удобных и функциональных цифровых платформ для представления своей продукции.​

Спортивная обувь сегодня является не только функциональной одеждой для занятий спортом, но и модным аксессуаром для повседневной носки. Ее дизайн и технологии постоянно развиваются, чтобы обеспечивать максимальный комфорт и защиту для стопы.[1] Согласно исследованиям, популярность спортивной обуви в базовых цветах остается высокой, что подчеркивает необходимость предоставления потребителям широкого ассортимента и подробной информации о товарах.

**Цель**

Целью данной дипломной работы является разработка веб-платформы для каталога спортивной обуви, предназначенной для оптимизации процессов управления ассортиментом, улучшения пользовательского опыта и повышения эффективности онлайн-продаж.

**Задачи**

​Для достижения поставленных задач по разработке веб-платформы каталога спортивной обуви мне надо выполнить следующие этапы:

1. Проектирование базы данных: разработать структуру базы данных, включая таблицы для хранения информации о товарах, категориях, размерах, ценах, пользователях и заказах.​
2. Разработка пользовательского интерфейса: создать адаптивный и интуитивно понятный интерфейс для пользователей, обеспечивающий удобную навигацию по каталогу товаров.​
3. Реализация функциональности корзины и оформления заказов: обеспечить возможность добавления товаров в корзину, изменения их количества и оформления заказов с вводом необходимых данных пользователя.​
4. Разработка административной панели: создать панель администратора для управления каталогом товаров, заказами и пользователями, включая возможность добавления, редактирования и удаления товаров, а также управления статусами заказов.​
5. Обеспечение безопасности и защиты данных: внедрить механизмы защиты от несанкционированного доступа, включая шифрование паролей и защиту от SQL-инъекций.​
6. Тестирование и оптимизация системы: провести функциональное и нагрузочное тестирование всех компонентов системы для выявления и устранения возможных ошибок, а также оптимизировать производительность системы для обеспечения быстрой и стабильной работы при различных нагрузках.​

**​** Объектом моей дипломной работы является веб-платформа для каталога спортивной обуви, предназначенная для онлайн-продажи и управления ассортиментом товаров. Предмет исследования

​ Также предметом исследования данной дипломной работы является процесс проектирования и разработки веб-платформы для каталога спортивной обуви, включая создание базы данных, пользовательского интерфейса и административной панели для управления ассортиментом и заказами

# **Глава 1. Теоретическая часть**

## **1.1 Изучение предметной области**

Перед началом всей разработки моей веб-платформы для каталога спортивной обуви нужно углубиться в теорию и понять, что такое вообще каталог, каким он бывает и как правильно он должен выглядеть.

Каталог товаров спортивной обуви должен представлять собой удобный список или витрина всех моделей обуви, которые доступны на сайте. Он нужен для того, чтобы покупатели могли легко просматривать ассортимент, узнавать подробную информацию о каждом товаре и быстро находить то, что им нужно.

В каталоге также должны быть модели в наличии, какие цвета и размеры доступны, сколько стоит каждая пара обуви, из какого материала она сделана, и как она выглядит на фото. То есть, каталог помогает человеку выбрать обувь, не выходя из дома.

Такой каталог играет важную роль на любом сайте, где что-то продаётся. Он как мостик между продавцом и покупателем: покупатель с его помощью может быстро найти нужный товар, сравнить его с другими и решить, что купить. А продавцу каталог помогает показать свой товар с лучшей стороны, рассказать о его преимуществах и привлечь больше клиентов.

Чем удобнее и понятнее сделан каталог, тем проще покупателю сориентироваться. Если в нём есть фильтры по размеру, цвету или цене — это экономит время. Если всё чётко подписано, есть фото и описание — это вызывает доверие. Поэтому создание хорошего каталога - важный шаг для любого интернет-магазина.

Для всего перечисленного нужно теперь разобраться каким методом лучше пользоваться и что они должны включать для реализации этого, а именно:

Ручной метод. Внесения данных в каталог товаров заключается в том, что я сам или сотрудники магазина вручную вносят все необходимые данные о товарах — названия, описания, цены, размеры, изображения и т. д. Преимущества этого метода заключаются в том, что специалист имеет полный контроль над каждым товаром и может детально указать все характеристики. Однако при таком подходе возникают сложности, особенно когда товарный ассортимент увеличивается. Визуализировать и обрабатывать большое количество данных становится сложнее. При ручном обновлении быстро оценить и просчитать все параметры, особенно в условиях изменений, не всегда возможно.

Автоматизированный метод. Он предполагает использование программного обеспечения для автоматической загрузки и обновления данных о товарах. В этом случае данные о товаре можно загружать и обновлять с помощью внешних источников или баз данных, что значительно ускоряет процесс. Автоматизация позволяет быстро составить предварительное расписание или обновить ассортимент с учетом всех изменений. Однако, несмотря на все преимущества, у этого метода есть свои ограничения. Объем данных, с которыми работает система, может быть слишком велик, чтобы их можно было удобно отобразить на экране компьютера для полноценной работы. Тем не менее, этот метод гораздо более эффективен для больших каталогов, поскольку позволяет быстро и точно обновлять информацию.

Рассмотрев все понятия этих методов для моей веб платформы лучше использовать автоматизированный метод, в рамках которого я реализовал несколько ключевых автоматизированных процессов, которые повышают эффективность работы с каталогом товаров и улучшают пользовательский опыт:

1. Автоматическое обновление данных товаров. В системе каталога должна быть реализована автоматическая синхронизация с базой данных, что позволяет обновлять товары без ручного ввода. Каждый товар имеет уникальный идентификатор, который помогает быстро изменять данные, такие как цена, описание и характеристики.
2. Интеграция с внешними источниками данных. Именно тут должна быть настроена автоматическая интеграция с внешними системами поставщиков, что позволяет получать актуальные данные о товарах. Эти данные, включая цену, наличие на складе и характеристики, автоматически синхронизируются с моей базой данных.
3. Автоматическая загрузка изображений товаров. Каждый товар в каталоге автоматически получает изображение через интеграцию с внешним источником или системой управления контентом. Когда добавляется новый товар, изображение подгружается автоматически, что упрощает процесс обновления и добавления новых товаров в каталог. Таким образом, должна обеспечиваться точность и своевременность визуальной информации, что важно для самих пользователей.
4. Автоматическое обновление информации о наличии товара. В этом разделе должна быть реализована система автоматического обновления информации о наличии товара в реальном времени. При изменении количества товара на складе или его продаже, система автоматически обновляет данные в каталоге. Это решение позволяет исключить продажи товаров, которых нет в наличии, и гарантирует актуальность информации для пользователей.
5. Автоматизированные уведомления для администраторов  
   В системе должны настроены уведомления для администраторов, которые информируют их о важных событиях, таких как новые заказы, изменения в наличии товаров или статусах заказов. Эти уведомления помогают администраторам оперативно реагировать на изменения и управлять процессами без задержек. Именно этот подход значительно ускорит принятие решений и улучшит качество обслуживания клиентов.
6. Автоматическое обновление статуса заказов. В рамках автоматизированного метода система автоматически обновляет статус заказов в зависимости от их текущего состояния (например, "в обработке", "отправлен", "доставлен"). Это позволяет клиентам и администраторам всегда быть в курсе текущего статуса каждого заказа.

Таким образом, я понял, что именно автоматизированный метод лучше подходит для моего каталога, а именно в повышении производительности и снижения затраты времени на администрирование в целом.

## **1.2 Анализ аналогов на рынке**

В современном мире создано огромное количество веб-платформ, где представлены каталоги спортивной обуви. Благодаря этому стало проще понять, что такое каталог и каким он должен быть. В интернете можно найти множество сайтов, которые, казалось бы, учли все возможные трудности, с которыми может столкнуться человек при создании подобного каталога.

Для анализа аналогов я рассмотрел их описание и внешний вид, но также испробовал весь функционал каталога и что-то взял на заметку для себя, чтобы реализовать в своей работе.

Веб-платформа *Street Beat* (street-beet.ru) ориентирована на молодежную аудиторию, предлагая стильный дизайн и актуальный ассортимент спортивной обуви, включая модели в стиле streetwear, с удобной навигацией и фильтрами по брендам, размерам, цветам и ценам. (Рис. 1)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 1 – Веб-платформа *Street Beat*

Street Beat (street-beet.ru) обладает удобным пользовательским интерфейсом и предоставляет широкий ассортимент спортивной обуви, включая модели в стиле streetwear. Также на этой платформе я подчеркнул удобную навигация по разделам: четкий выбор категории товаров (например, "Кроссовки", "Кеды", "Ботинки").

Единственный существенный минус на данной платформе - это отсутствие подробных технических описаний товаров, таких как информация о материалах и технологиях, что может затруднить выбор для покупателей, уделяющих внимание деталям и функциональности обуви.

Веб-платформа Nike (nike-sneakers.ru) Nike -  [американская](https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%B5%D0%B4%D0%B8%D0%BD%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%A8%D1%82%D0%B0%D1%82%D1%8B_%D0%90%D0%BC%D0%B5%D1%80%D0%B8%D0%BA%D0%B8) [транснациональная компания](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BD%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F), специализирующаяся на спортивной одежде и обуви[2]. В ней присутствует предлагающий стильный и функциональный онлайн-каталог. (Рис. 2) Данная платформа ориентирована на широкую аудиторию: от профессиональных спортсменов до любителей активного образа жизни и моды. Nike.com отличается современным дизайном, удобной навигацией и продуманным пользовательским интерфейсом, что делает процесс выбора и покупки максимально комфортным.

Изображение выглядит как текст, снимок экрана, Веб-сайт, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 2 – Веб-платформа *Nike*

Основные характеристики платформы Nike-sneakers.ru

1. Ассортимент товаров
2. Ценовая политика
3. Навигация и фильтры
4. Интерфейс и дизайн
5. Описание товаров
6. Гарантии и подлинность

В этой веб-платформе я выделил, минималистичный дизайн так как он помогает сосредоточить внимание на товарах и делает интерфейс понятным. Удобная навигация с четкими категориями и фильтрами позволит пользователям быстро находить нужные модели. Но и есть минус, что я как лично для своего каталога добавил больше описания, чем на (nike-sneakers.ru) для повышения доверия пользователей.

Веб-платформа *Lamoda*(lamoda.ru). Lamoda — это крупнейшая в России и СНГ онлайн-платформа для продажи товаров, связанных с модой и образом жизни.[3] На платформе Lamoda представлен широкий ассортимент спортивной обуви, которая удовлетворяет потребности разных категорий покупателей: от любителей повседневного комфорта до профессиональных спортсменов. (Риc. 3)

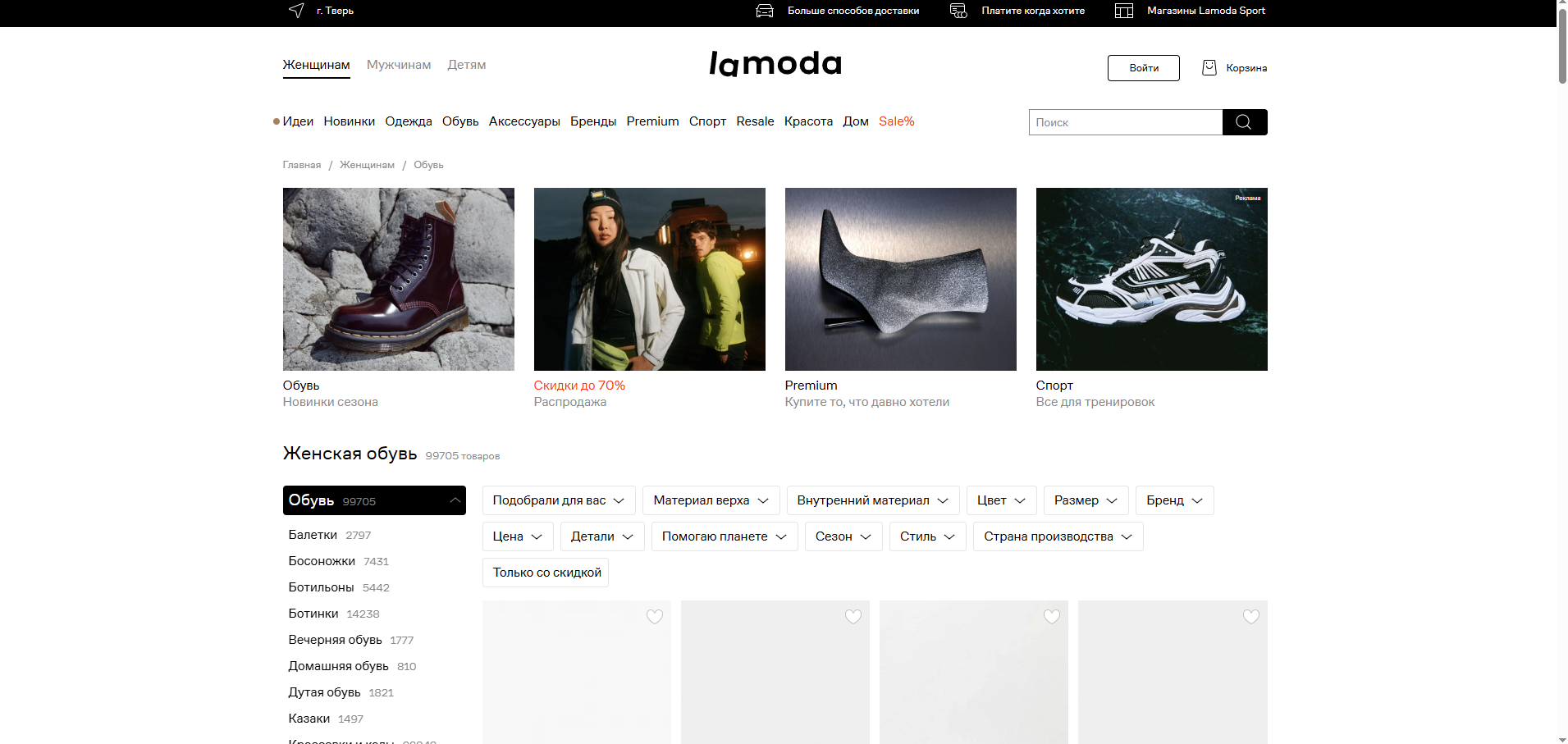
******

Рис. 3 – Веб-платформа *Lamoda*

Основные характеристики платформы lamoda.ru

1. Детальные описания товаров
2. Мобильная адаптация и приложение
3. Отзывы и рейтинги покупателей
4. Акции и скидки

В каталоге *Lamoda* я выделил систему скидок и акций как одно из ключевых преимуществ платформы которое мне понадобиться для реализации своего каталога. На сайте регулярно проводятся распродажи, включая сезонные скидки и распродажи. Но также надо заметить, что нужна необходимость четко контролировать баланс между акциями и базовыми ценами, чтобы не вызывать недоверие у клиентов.

Веб-платформа *Kari* (kari.com).Kari – международная сеть магазинов обуви и аксессуаров для взрослых и детей [4]. На платформе Kari представлен широкий ассортимент спортивной обуви, который включает модели для разных видов активности: бег, тренировки, футбол, повседневная носка и другие. (Рис. 4) Особенностью каталога является сочетание товаров от известных брендов (Nike, Adidas, Puma, Reebok) с более доступными вариантами собственных марок.

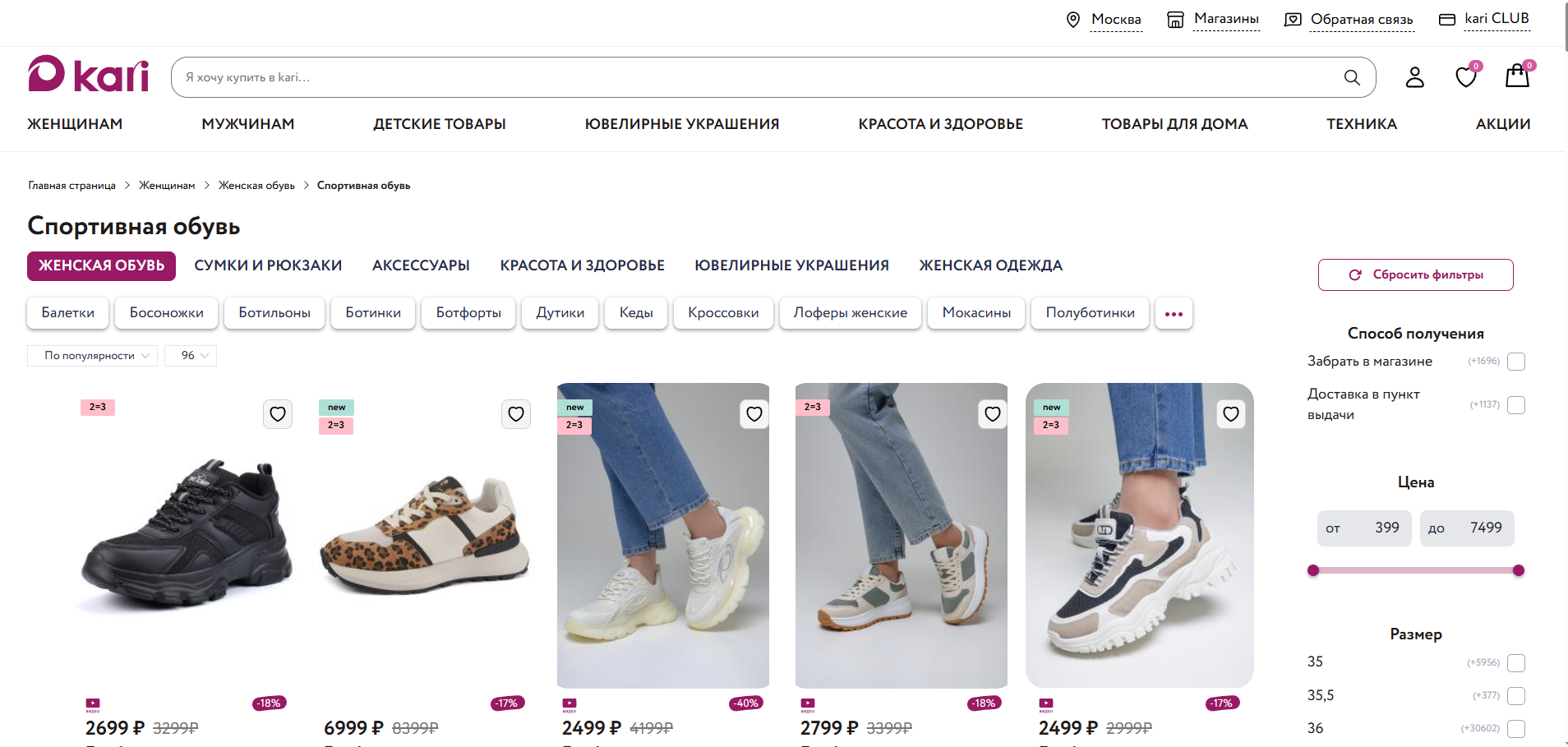
******

Рис. 4 – Веб-платформа *Kari*

Основные характеристики каталога Kari

1. Широкий ассортимент товаров
2. Удобная система фильтров
3. Эксклюзивные разделы и мобильная адаптация
4. Качественные фотографии и базовые описания

В каталоге *Kari* меня привлекла удобная система фильтров, которая позволяет быстро находить товары по размеру, цвету, бренду и назначению — это решение я планирую реализовать в своем каталоге для улучшения навигации. Также я обратил внимание на широкий ассортимент обуви, включая как премиальные бренды, так и доступные варианты, что помогает привлечь разнообразную аудиторию.

Проанализировав другие каталоги обуви, я заметила, что почти все они ориентированы на обычных покупателей и предлагают стандартные функции: фильтры, отзывы и возможность примерки. Но часто не хватает подробной информации о товарах, например, о материалах или технологиях. Некоторые платформы могут подбирать обувь по заданным параметрам, но редко добавляют качественные фото или удобный просмотр. Также многие сайты плохо работают на телефонах или имеют запутанный интерфейс. Это показывает, что можно сделать лучше: создать удобный каталог с хорошими описаниями, качественными фото, мобильной версией и простой навигацией для всех категорий покупателей.

## **1.3 Выбор языка программирования и СУБД**

После анализа аналогов на рынке можно приступить к выбору языка программирования и системы управления базами данных (СУБД), которые обеспечат эффективную и удобную работу с каталогом спортивной обуви.

Чтобы разобраться с выбором языка программирования, начнем с пользовательской стороны, то есть с того, что видит и с чем взаимодействует пользователь. В первую очередь, рассмотрим языки программирования, подходящие для разработки веб-приложений, а именно Frontend.

Frontend — это всё, что видит и с чем взаимодействует пользователь при работе с веб-сайтом или приложением [5]. Фронтенд отвечает за внешний вид, удобство использования. Он включает в себя HTML, CSS и JavaScript, а также дополнительные инструменты (например, фреймворки), которые упрощают разработку и делают сайт более современным.

Я выбрал HTML и CSS в качестве основы для разработки пользовательской части сайта, так как эти технологии являются фундаментальными для создания веб-страниц. HTML отвечает за структуру и содержание страницы, а CSS — за её внешний вид и оформление.

HTML —  это язык разметки, который используется для создания структуры веб-страниц [6]. Он определяет, где будут расположены заголовки, тексты, изображения, кнопки и другие элементы. Без HTML невозможно построить каркас сайта, так как он задает основу для всех остальных технологий.

CSS используется для оформления и визуального стиля сайта, задавая цвета, шрифты, размеры элементов, отступы, анимации и адаптивность. С его помощью можно оформить карточки товаров (например, стиль кнопок "Купить", рамки, тени, цветовые акценты), обеспечить корректное отображение сайта на разных устройствах благодаря адаптивному дизайну, добавить анимации и переходы (например, плавное появление подменю или эффект при наведении на кнопку) и создать гармоничное сочетание цветов и шрифтов для привлекательного внешнего вида.

JavaScript подходит для создания динамической реализации быстрого поиска товаров по ключевым словам фильтрация по брендам размерам и цветам а также функционал сравнения товаров или добавления их в Избранное JavaScript обеспечивает мгновенную реакцию на действия пользователя что делает сайт более отзывчивым Node.js позволяет использовать один язык JavaScript как на стороне клиента так и на стороне сервера что упрощает разработку и делает процесс более эффективным.

Python подходит для разработки логики обработки заказов управления каталогом товаров и интеграции с платежными системами, а также для создания административной панели чтобы добавлять новые модели обуви редактировать описания и загружать фотографии Python имеет простой синтаксис что ускоряет разработку c фреймворками предоставляют готовые решения для работы с базами данных аутентификации пользователей и обработки запросов.

PHP подходит для создания структуры каталога с использованием CMS WordPress или разработки кастомного решения на Laravel а также для реализации функционала отзывов рейтингов и системы лояльности PHP широко используется для создания сайтов электронной коммерции благодаря своей совместимости с MySQL и множеству готовых решений, кроме того, он легко интегрируется с популярными платформами такими как WooCommerce для интернет-магазинов.

Именно язык PHP я и выбрал для бэкенда, так как он обеспечивает надежную обработку запросов пользователей и взаимодействие с базой данных. Это позволяет реализовать такие важные функции, как добавление товаров в каталог, фильтрацию по различным параметрам, работу с формами (регистрация, авторизация, оформление заказа) и интеграцию с платежными системами. Благодаря своей универсальности и широкому набору инструментов, PHP стал оптимальным выбором для разработки сайта каталога спортивной обуви.

При разработке каталога спортивной обуви можно использовать различные фреймворки, которые помогут ускорить процесс создания сайта, обеспечить его эффективность и современный внешний вид. Вот основные типы фреймворков и конкретные примеры, которые могут быть полезны для такого проекта:

Bootstrap — это популярный CSS-фреймворк, который предоставляет готовые компоненты для создания адаптивного дизайна. Он может использоваться для быстрого создания отзывчивого меню навигации, кнопок, карточек товаров и форм, обеспечивает готовую сетку для удобного расположения элементов каталога, а также позволяет легко стилизовать таблицы, модальные окна (попапы) и другие UI-компоненты, делая разработку сайта более эффективной и визуально привлекательной

React.js — это JavaScript-библиотека для создания динамических пользовательских интерфейсов. Она может использоваться для реализации живого поиска товаров и фильтрации без перезагрузки страницы, динамического обновления контента (например, изменение списка товаров при выборе бренда или размера), а также для создания сложных интерактивных элементов, таких как сравнение товаров или добавление в "Избранное".

Vue.js — это легковесный JavaScript-фреймворк для создания интерактивных интерфейсов. Он может использоваться для удобной работы с формами (например, оформление заказа или регистрация), а также для легкой интеграции с бэкендом и отображения данных о товарах.

Также есть и фреймворки для тестирования, например такой как PHPUnit

PHPUnit — это фреймворк для тестирования юнитов в языке программирования PHP, который позволяет проверять корректность работы бэкенд-логики[7]. Он может использоваться для тестирования таких функций, как обработка заказов, расчет скидок, работа с базами данных или проверка бизнес-логики приложения. PHPUnit помогает убедиться, что все компоненты серверной части работают правильно и без ошибок, обеспечивая надежность и стабильность проекта.

После выбора языка программирования и выбора фронтенда с фреймворками можно перейти к СУБД.

СУБД — связующее звено между базой данных, пользователем или приложением. С её помощью можно создавать, удалять и изменять базы данных, фильтровать и находить нужные элементы, менять их структуру и делать резервные копии.

База данных (БД) — это организованные структуры, предназначенные для хранения и управления информацией [8]. Базы данных бывают:

Текстовые базы данных — это простейший тип баз данных, где данные хранятся в виде текстовых файлов (например, .txt, .csv, .json). Такие базы данных не требуют специального программного обеспечения для работы, но подходят только для небольших объемов данных.

Реляционная база данных (БД) — это тип базы данных, в которой данные организованы в виде связанных таблиц, также называемых отношениями.

Сетевые базы данных — тип модели баз данных, которая представляет информацию в качестве структуры сетей, напоминающей граф.

Нереляционные базы данных (NoSQL) — это тип баз данных, для работы с которыми используется не только язык SQL, но и столбовые, грабовые, документно-ориентированные системы, а также системы модели «ключ — значение»

Для реализации моей веб-платформы каталога спортивной обуви я выбрал реляционную базу данных.

Примеры реляционных баз данных:

MySQL — одна из самых популярных систем управления базами данных (СУБД), которая активно используется в веб-разработке, анализе данных и других областях. Она отличается высокой производительностью, бесплатностью и широкой поддержкой. [9] Каждая таблица в этой базе данных состоит из строк и столбцов, а связи между таблицами обеспечивают целостность и структурированность данных. Для работы с MySQL используется язык запросов SQL (Structured Query Language).

PostgreSQL — это мощная объектно-реляционная СУБД (система управления базами данных), которая предлагает расширенные возможности для работы с данными. Она поддерживает не только стандартные реляционные функции, но и более сложные механизмы, такие как работа с JSON, геоданными и транзакциями.

SQLite — это легковесная встраиваемая СУБД, которая хранит всю базу данных в одном файле. Она не требует отдельного сервера и работает непосредственно в приложении, что делает её простой и удобной для небольших проектов.

Microsoft Access — это реляционная система управления базами данных (СУБД), которая входит в состав пакета Microsoft Office. Она позволяет создавать, управлять и анализировать данные в виде таблиц, форм, запросов и отчетов. Access часто используется для небольших проектов или локальных приложений, где данные хранятся на одном компьютере или в локальной сети.

Microsoft SQL Server — это мощная реляционная система управления базами данных (СУБД), разработанная корпорацией Microsoft. Она предназначена для работы с большими объемами данных и поддерживает сложные запросы, аналитику и высокую производительность. SQL Server часто используется в корпоративных проектах, интернет-магазинах и приложениях, где важна надежность, безопасность и масштабируемость.

MariaDB — это реляционная система управления базами данных (СУБД), которая является форком (ответвлением) MySQL. Она была создана после того, как MySQL был приобретен компанией Oracle, чтобы сохранить открытую и свободную альтернативу с открытым исходным кодом. MariaDB полностью совместима с MySQL, но предлагает дополнительные функции, улучшенную производительность и больше возможностей для разработчиков.

Из всего списка доступных баз данных я выбрал реляционную СУБД MySQL для реализации каталога спортивной обуви. Этот выбор обусловлен не только тем, что у меня больше опыта работы именно с MySQL, включая выполнение различных проектов и заданий в колледже, но и тем, что для управления базой данных использовался phpMyAdmin.

PhpMyAdmin — это программа, которая позволяет управлять базами данных через удобный графический интерфейс. Ее устанавливают на сервер, где хранится сайт, а затем открывают в браузере для удаленного администрирования СУБД. Проще говоря, можно отдавать команды путем нажатия на кнопки в пользовательском веб-интерфейсе, а не вручную расписывать их в терминале.[10]

Основные возможности phpMyAdmin:

1. Управление базами данных:
   * Создание, редактирование, удаление баз данных, таблиц, полей и индексов.
   * Настройка связей между таблицами и управление структурой данных.
2. Импорт и экспорт данных:
   * Поддержка различных форматов (CSV, SQL, XML и др.) для импорта и экспорта данных.
   * Возможность создания резервных копий баз данных (дампов) для их сохранения или переноса.
3. Выполнение SQL-запросов:
   * Удобный редактор запросов для выполнения сложных SQL-команд.
   * Возможность тестирования и отладки запросов в реальном времени.
4. Поиск и фильтрация данных:
   * Поиск по всей базе данных или только по определенным таблицам.
   * Фильтрация данных для быстрого анализа нужной информации.
5. Настройка безопасности и доступа:
   * Управление правами пользователей и уровнями доступа к базам данных.
   * Возможность администрирования нескольких серверов MySQL одновременно.
6. Графические инструменты:
   * Генерация графиков макета базы данных для наглядного представления структуры.
   * Визуализация связей между таблицами.
7. Смена кодировки и оптимизация:
   * Возможность изменения кодировки баз данных и таблиц.
   * Оптимизация таблиц для повышения производительности.
8. Многопользовательский режим:
   * Поддержка работы с несколькими пользователями, каждый из которых может иметь свои права доступа.

Когда я разобрался с теоретической частью и определился с выбором СУБД для проекта, настало время приступать к практической реализации. Теперь, имея четкий план действий, можно начинать создавать базу данных, настраивать сервер и писать код для функционала сайта.

# **Глава 2 Проектная часть**

## **2.1. Разработка архитектуры базы данных**

Для разработки каталога спортивной обуви я создал базу данных с 7 основными таблицами products, sizes, product\_sizes, orders, order\_items, wishlist.

* products — это таблица товаров, в которой есть информация о каждом товаре: название, описание, цена, изображение
* sizes- эта таблица используется при создании формы выбора размера в карточке товара
* product\_sizes- в этой таблице обеспечивается многозначную связь между товарами и размерами
* orders- эта таблица заказов. В ней есть информация о созданном заказе: имя покупателя, адрес, телефон, сумма, статус, дата
* order\_items- эта таблица, в которой есть информация о заказе
* wishlist- это таблица с избранным, а именно какие товары пользователь добавил в "избранное"

Рассмотрим подробнее каждую таблицу:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 5. Таблица «products»

Сущность products состоит из информации о товарах, включая название, описание, цену и тип (обувь, шнурки, стельки). Она также содержит данные о поле и назначении товара для фильтрации в каталоге. Каждый товар связан с доступными размерами через таблицу product\_sizes. Эта сущность является основной для отображения товаров на сайте и формирования заказов.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6. Таблица «product\_sizes»

Таблица product\_sizes связывает товары из таблицы products с размерами из таблицы sizes. Она позволяет указать, какие размеры доступны у каждого товара. Это помогает гибко управлять ассортиментом и фильтровать товары по размерам. Без этой таблицы сложно было бы организовать выбор размеров при добавлении в корзину или отображении каталога.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 7. Таблица «sizes»

Сущность sizes хранит список всех доступных размеров обуви, таких как 36, 37, 40 и т. д. Она используется для централизованного управления размерами, чтобы избежать дублирования данных в других таблицах. Эта таблица связывается с таблицей products через таблицу product\_sizes, обеспечивая гибкость при выборе размеров в каталоге.  
Благодаря этой сущности пользователь может выбирать товары по конкретным размерам, а фильтрация в каталоге работает корректно.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8. Таблица «users»

Сущность users отвечает за хранение данных о зарегистрированных пользователях сайта. Каждый пользователь имеет уникальный идентификатор, имя, email и зашифрованный пароль. Эта таблица связана с заказами, избранным и корзиной через user\_id. Благодаря этой сущности реализуется авторизация, личный кабинет и история заказов. Также она позволяет персонализировать опыт использования сайта для каждого клиента.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9. Таблица «wishlist»

Сущность wishlist отвечает за хранение избранных товаров, которые пользователь добавил для последующего просмотра или покупки. Она представляет собой таблицу, в которой сохраняется связь между пользователем и товаром, который он поместил в избранное.

Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10. Таблица «orders»

Сущность orders отвечает за хранение информации о заказах, сделанных пользователями на сайте. В ней сохраняются данные о пользователе, контактная информация, адрес доставки, общая сумма и статус заказа. Каждый заказ получает уникальный идентификатор и дату создания, что позволяет отслеживать историю покупок.  
Статусы заказов (например, "ожидает оплаты", "доставлен") помогают в управлении и обработке заказов в админке. Эта таблица связывается с order\_items, чтобы хранить детали — какие именно товары и в каком количестве были куплены.

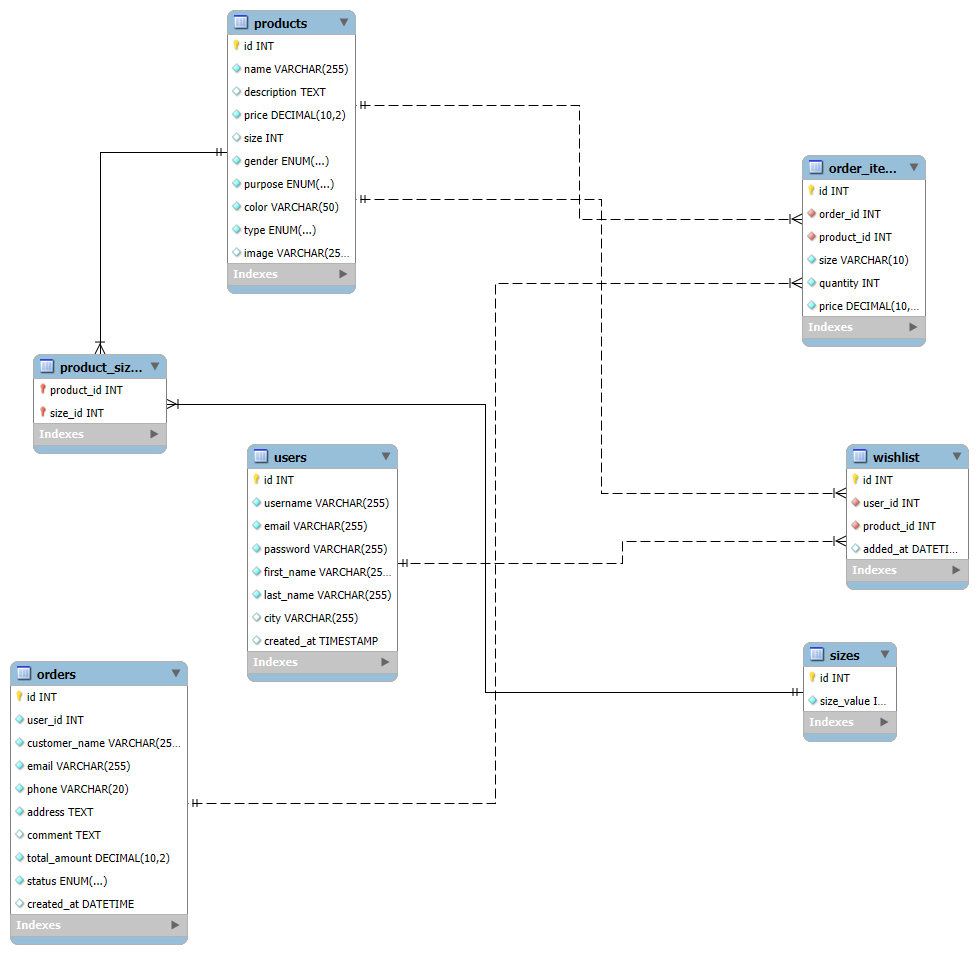
Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 11. Таблица «orders\_items»

Сущность order\_items хранит информацию о каждом товаре, который был в заказе — его ID, количество и цену на момент покупки. Она связана с таблицей orders через order\_id, чтобы понимать, к какому заказу относится каждый товар. Эта таблица позволяет формировать детализацию заказа - какие товары были куплены и в каком количестве. Цены сохраняются в этой таблице, чтобы сумма заказа не менялась даже при изменении цены товара в будущем. Благодаря ей пользователь и администратор могут увидеть полный состав заказа с точными данными о товарах.

ERD-диаграмма:



На изображении представлена ER-диаграмма, показывающая связи между таблицами базы данных интернет-магазина.

1. Таблица users связана с orders по полю user\_id, что означает - один пользователь может иметь несколько заказов.
2. Таблица orders связана с order\_items по полю order\_id, то есть в одном заказе может быть несколько товаров.
3. Таблица products связана с order\_items по поле product\_id, это значит - каждый товар может входить в состав разных заказов.
4. Также products связана с product\_sizes по полю id, что говорит о том, что у одного товара может быть несколько размеров.
5. Таблица sizes связана с product\_sizes по полю size\_id, то есть каждый размер может использоваться у нескольких товаров.
6. Таблица wishlist связана с users и products, что позволяет пользователю добавлять товары в список желаемого.
7. Все связи между таблицами реализованы через внешние ключи (foreign keys) для обеспечения целостности данных.
8. Такая структура позволяет гибко управлять товарами, заказами, пользователями и их взаимодействием в системе.
9. Диаграмма отражает логическую организацию данных, необходимую для работы интернет-магазина, включая корзину, историю заказов и управление каталогом.

После создания всех основных таблиц для каталога спортивной обуви я начал разработку пользовательского интерфейса приложения.

## **2.2. Разработка интерфейса приложения**

При создании интерфейса приложения для каталога спортивной обуви я ориентировался на популярные интернет-магазины и аналоги, чтобы сделать дизайн интуитивно понятным и удобным для пользователя. Изучив разные аналоги, я выбрала логическую структуру страниц: шапка с меню, фильтры, каталог, карточки товаров и навигация с пагинацией. Для улучшения пользовательского опыта были реализованы фильтрация по размерам, полу и назначению, а также система сортировки и поиска. Визуальное оформление выполнено на основе CSS-стилей из файла, с использованием цветовых акцентов, стилизованных кнопок и таблиц с цветными метками статусов. После всей разработки каталог выглядит так:

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 12. Вкладка «каталога»

На данной вкладке "Каталог спортивной обуви" отображается основное содержимое интернет-магазина, где пользователи могут просматривать и выбирать товары. В верхней части страницы находится шапка с названием раздела ("Каталог спортивной обуви") и кнопкой входа/выхода пользователя. Под шапкой расположена строка поиска для быстрого нахождения товаров по ключевым словам.

На странице реализован удобный интерфейс для выбора товаров, где слева расположены фильтры по размеру, полу, назначению, типу и цене, что позволяет пользователю быстро сузить круг поиска. Справа от фильтров находятся карточки товаров, организованные в виде сетки, каждая из которых содержит изображение, название, краткое описание, цену и кнопку "Посмотреть", ведущую на страницу с подробной информацией.

Дизайн сайта выполнен в строгом стиле: шапка - с черным фоном и белым текстом, фон всей страницы светло-серый, что обеспечивает хорошую читаемость. Карточки товаров имеют белый фон с тенью, чтобы выделяться на общем фоне и быть визуально разделенными. Кнопки "Посмотреть" выполнены в ярко-зеленом цвете, привлекают внимание и меняют цвет при наведении, подсказывая о возможности взаимодействия. Фильтры представлены выпадающими списками и ползунком для выбора диапазона цены, что делает их понятными и удобными в использовании.

Поиск товаров осуществляется через строку поиска, а фильтрация мгновенно обновляет список при изменении параметров благодаря JavaScript. При клике на кнопку "Посмотреть" пользователь переходит на детальную страницу товара, где может ознакомиться с характеристиками и добавить его в корзину.

Статусы заказов отмечаются цветными метками, которые помогают легко определить текущее состояние заказа: "Ожидает оплаты", "Оплачен", "В пути", "Доставлен" или "Отменён". Целевая аудитория — это покупатели спортивной обуви, шнуров и стелек, которым важно быстро найти подходящий товар по заданным параметрам. Основные цели страницы - упростить поиск товаров, обеспечить удобную навигацию и повысить скорость совершения покупок за счет четкого дизайна и продуманной функциональности.

Техническая реализация включает базу данных, в которой хранится информация о товарах, их характеристиках и наличии. Для оформления внешнего вида используются CSS-стили, создающие современный и приятный интерфейс. JavaScript применяется для динамической фильтрации и интерактивности элементов без перезагрузки страницы, что делает процесс выбора товаров более плавным и удобным. Шапка сайта содержит кнопку входа или выхода пользователя, что обеспечивает простой доступ к личному кабинету.

Таким образом, после завершения этапа проектирования пользовательского интерфейса и определения ключевых элементов страницы, можно перейти к разработке функционала, который обеспечит полноценное взаимодействие пользователя с системой. На этом этапе необходимо реализовать логику работы фильтров, динамическое обновление списка товаров, интеграцию поисковой строки и работу с детальной информацией о товаре. Особое внимание следует уделить взаимодействию между компонентами интерфейса и серверной частью, чтобы данные отображались корректно и в реальном времени.

## **2.3. Разработка функционала**

После того как основная вкладка каталога была реализована, и пользователь имеет возможность просматривать список товаров с фильтрацией и поиском, внимание переключается на страницу конкретного товара - ведь именно здесь происходит ключевое взаимодействие с продуктом перед покупкой. Как только пользователь нажимает на кнопку "Посмотреть" в карточке товара, он переходит на детальную страницу, где ему становится доступна вся информация о выбранном продукте.

Изображение выглядит как текст, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 13. Вкладка «Страница товара»

Именно на этой странице пользователь может добавить товар в корзину, нажав на соответствующую кнопку на странице товара, что позволяет сформировать заказ и перейти к его оформлению.

Реализация функции "Добавление товара в корзину" выглядит так:

if (addToCartBtn) {

addToCartBtn.addEventListener('click', function (e) {

e.preventDefault();

const productId = this.getAttribute('data-product-id');

const quantity = parseInt(document.querySelector('.quantity-input').value);

const size = selectedSizeInput.value;

if (!size) {

alert('Пожалуйста, выберите размер');

return;

}

В этой функции я разработал логику добавления товара в корзину. При нажатии на кнопку «Добавить в корзину» происходит проверка наличия самой кнопки, чтобы избежать ошибок на других страницах. Далее, с помощью JavaScript, собираются данные о товаре: его идентификатор (через атрибут data-product-id), количество (из поля ввода) и размер (пользователь должен его выбрать). Перед добавлением проверяется, указан ли размер — если нет, выводится предупреждение. Таким образом, функция обеспечивает корректный сбор данных перед дальнейшей обработкой товара, например, его добавлением в корзину или отправкой на сервер.

Также на странице товара можно добавить товар в избранное просто нажав на кнопку “Добавить в избранное” и товар попадет в таблицу wishlist.

Реализация функции "Добавление товара в избранное" выглядит так:

addToCartBtn.addEventListener('click', function (e) {

const productId = this.getAttribute('data-product-id');

const size = selectedSizeInput.value;

fetch('add\_to\_cart.php', {

method: 'POST',

body: JSON.stringify({ product\_id: productId, quantity: 1, size: size })

});

});

В данной функции я реализовал клиентскую логику добавления товара в избранное с использованием JavaScript. При клике по кнопке «В избранное» извлекается уникальный идентификатор товара из пользовательского атрибута data-product-id. Далее осуществляется отправка асинхронного POST-запроса на серверный обработчик add\_to\_wishlist.php с помощью метода fetch, в теле которого передаётся ID товара в формате JSON. Такой подход позволяет добавлять товар в избранное без перезагрузки страницы, обеспечивая более отзывчивое и удобное взаимодействие пользователя с интерфейсом.

После реализации функционала добавления товара в избранное я приступил к разработке страницы «Корзина покупок» (см.Приложение1), которая является важным этапом пользовательского пути перед оформлением заказа. На данной странице пользователь может просматривать добавленные товары, управлять их количеством, применять промокоды и завершать покупку.

Интерфейс включает навигационное меню со ссылками на главную страницу, каталог товаров и корзину, а также кнопку «Аккаунт» для авторизованных пользователей. В верхней части страницы расположено поле ввода для применения промокода с примером "SUMMER20" и кнопкой активации. Ниже находятся основные действия: кнопка «Оформить заказ» — для завершения покупки и перехода к форме доставки, и «Продолжить покупки» -позволяющая вернуться к выбору товаров.

Главная функциональная задача страницы корзины -обеспечение пользователю возможности управлять выбранными товарами. Я реализовал две ключевые операции: изменение количества товара и его удаление из корзины:

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (isset($\_POST['remove\_item'])) {

$itemKey = $\_POST['item\_key'];

unset($\_SESSION['cart'][$itemKey]); // Удаление товара из корзины

header("Location: cart.php?msg=" . urlencode("Товар удалён из корзины."));

exit();

} elseif (isset($\_POST['update\_quantity'])) {

$itemKey = $\_POST['item\_key'];

$quantity = intval($\_POST['quantity']);

if ($quantity > 0) {

$\_SESSION['cart'][$itemKey] = $quantity; // Обновление количества

} else {

unset($\_SESSION['cart'][$itemKey]); // Удаление, если количество 0

}

header("Location: cart.php?msg=" . urlencode("Количество обновлено."));

exit();

В этом коде я реализовал возможность управлять содержимым корзины - а именно удалять товары и изменять количество. При отправке формы проверяется тип действия: если нужно удалить товар - он удаляется из сессии по ключу. Если же пользователь меняет количество - оно обновляется, а при значении 0 товар также удаляется из корзины. После каждого действия происходит переход обратно на страницу корзины с соответствующим уведомлением для пользователя.

После того как пользователь добавил нужные товары в корзину и при необходимости отредактировал их количество, он может перейти к оформлению заказа, нажав на кнопку «Оформить заказ» (см.Приложение 2). Это действие перенаправляет его на страницу checkout.php, предназначенную для завершения покупки.

Описывая страницу оформления заказа, я выделил два ключевых раздела:

1. Информация о заказе — здесь отображаются товары из корзины с указанием названия, размера, количества и стоимости каждой позиции, а также общей суммы заказа с учётом применённых скидок.
2. Форма контактной информации — пользователь заполняет обязательные поля: ФИО, Email, телефон и адрес доставки (обязательные поля помечены звездой). Также предусмотрено поле для комментария к заказу.

После заполнения формы и нажатия на кнопку «Подтвердить заказ» данные отправляются на сервер, где происходит сохранение информации о заказе в базу данных. После этого заказ становится доступен в административной панели для обработки менеджером, а также отображается в разделе "Мои заказы" в личном аккаунте пользователя.

Личный кабинет пользователя представляет собой персональное пространство, в котором пользователь может управлять своими данными и отслеживать историю покупок. Основным разделом личного кабинета является блок «Ваши заказы», содержащий информацию обо всех оформленных заказах.

Каждый заказ отображается с подробными сведениями:

1. дата оформления,
2. текущий статус (например, «Ожидает оплаты», «Отправлен»),
3. название товара,
4. цена за единицу и общее количество,
5. итоговая сумма заказа.
6. удаление заказа

Такая структура позволяет пользователю легко находить нужную информацию и контролировать статус своих покупок. Более подробно интерфейс личного кабинета представлен в (см. Приложении 3).

Для понимания как реализован статус заказа, я написал функцию:

function getStatusText($status) {

$statuses = [

'pending' => 'Ожидает оплаты',

'paid' => 'Оплачен',

'processing' => 'В обработке',

'shipped' => 'Отправлен',

'delivered' => 'Доставлен',

'cancelled' => 'Отменён'

];

return $statuses[$status] ?? 'Неизвестен';

}

В данной функции я реализовал отображение статусов заказов на русском языке, чтобы информация была понятна пользователю.

Функция getStatusText($status) принимает внутренний ключ статуса (например, 'pending', 'shipped' и т. д.) и возвращает соответствующее ему читаемое название на русском языке -например:

1. 'pending' → "Ожидает оплаты"
2. 'shipped' → "Отправлен"
3. 'delivered'→"Доставлен"

Это позволяет выводить статусы заказов в удобочитаемом виде на странице личного кабинета или в административной панели, вместо технических кодов.

Для управления товарами, статусами заказов и другими функциями, к которым обычный пользователь не имеет доступа, я реализовал административную панель. (см. Приложение 4)

Реализация функции добавления товара в каталог:

$stmt = $pdo->prepare("INSERT INTO products (name, description, price, size, gender, purpose, image)

VALUES (:name, :description, :price, :size, :gender, :purpose, :image)");

$stmt->execute([

':name' => $name,

':description' => $description,

':price' => $price,

':size' => $size,

':gender' => $gender,

':purpose' => $purpose,

':image' => $image]);

В данной функции я осуществил возможность добавления нового товара в каталог. Основной её фрагмент — это SQL-запрос с использованием INSERT INTO, который сохраняет данные о товаре (название, цена, размер и т. д.) в таблицу products. Для безопасного добавления используется подготовленный запрос с привязкой параметров через execute(), что защищает от SQL-инъекций.

Реализация редактирования товара в каталог:

$stmt = $pdo->prepare("UPDATE products SET

name = :name,

description = :description,

price = :price,

size = :size,

gender = :gender,

purpose = :purpose,

image = :image

WHERE id = :id");

$stmt->execute([

':name' => $name,

':description' => $description,

':price' => $price,

':size' => $size,

':gender' => $gender,

':purpose' => $purpose,

':image' => $image,

':id' => $id

])

Ну и, конечно, для возможности изменения данных о товаре я реализовал функцию редактирования. После выбора конкретного товара администратор попадает в форму, где может обновить такие поля, как название, описание, цена, размер, пол и назначение, а также заменить изображение.

Основной частью этой функции является SQL-запрос с использованием UPDATE, который обновляет запись в таблице products по уникальному идентификатору товара. Как и при добавлении, здесь применяются подготовленные запросы и привязка параметров через execute(), что обеспечивает безопасность и предотвращает SQL-инъекции.

Такая реализация позволяет администратору быстро и безопасно вносить изменения в карточки товаров, поддерживая актуальность каталога.

# **Заключение**

В ходе выполнения дипломного проекта была разработана веб-платформа для каталога спортивной обуви, предназначенная для оптимизации процессов управления ассортиментом, улучшения пользовательского опыта и повышения эффективности онлайн-продаж. Были достигнуты все поставленные задачи: спроектирована и реализована структура базы данных, создана адаптивная и удобная система пользовательского интерфейса, реализован функционал корзины и оформления заказов, а также разработана административная панель для управления товарами и заказами.

На этапе проектирования были изучены предметная область и аналоги на рынке, что позволило выявить ключевые особенности успешных интернет-магазинов. На основе проведенного анализа был сделан выбор в пользу автоматизированного метода управления данными, что обеспечивает гибкость, точность и скорость работы с каталогом товаров. Также были выбраны технологии реализации: язык программирования PHP для бэкенда, JavaScript для клиентской логики, HTML/CSS — для верстки, и реляционная СУБД MySQL для хранения данных.

Разработанная база данных состоит из 7 таблиц (products, sizes, product\_sizes, users, wishlist, orders, order\_items), связанных между собой через внешние ключи. Это позволило организовать данные таким образом, чтобы обеспечить целостность информации и простоту управления как товарами, так и заказами.

Интерфейс приложения был создан с учетом современных требований к юзабилити: реализованы фильтры по размерам, полу и назначению, система поиска и сортировки, карточки товаров с изображениями и ценами, а также детальная страница каждого товара. Визуальный стиль выполнен в строгом и чистом дизайне, обеспечивающем комфортную навигацию и высокую читаемость контента.

Функциональная часть включает:

* + 1. Добавление и удаление товаров из корзины.
    2. Обновление количества товаров.
    3. Применение промокодов.
    4. Оформление заказа с вводом контактной и адресной информации.
    5. Отслеживание заказов в личном кабинете пользователя.
    6. Изменение статуса заказа в административной панели.

Административная панель предоставляет возможность полного контроля над товарами и заказами: добавление, редактирование, удаление товаров, просмотр и изменение статусов заказов. Это делает систему удобной не только для конечных пользователей, но и для менеджеров магазина.

Проект полностью соответствует целям, заявленным во введении. Разработанная система позволяет:

* + 1. Увеличить продажи за счет удобного интерфейса и широкого охвата аудитории.
    2. Автоматизировать процессы управления товарами и заказами.
    3. Повысить уровень доверия пользователей благодаря четкой структуре и актуальной информации.

Таким образом, данный дипломный проект представляет собой полноценное решение для онлайн-торговли спортивной обувью, которое может быть использовано как основа для коммерческого интернет-магазина. В дальнейшем возможно расширение функционала за счет интеграции платежных систем, системы доставки, отзывов и рейтинга, мобильного приложения и других модулей.

# **Список источников**

1. Спортивная обувь: от функциональной одежды до модного аксессуара – [Электронный ресурс] / – Режим доступа:

<https://zonasporta.com/articles/sportivnyy-stily-odeghdy/sportivnaya-obuvy-ot-funkcionalynoy-odeghdy-do-modnogo-aksessuara>

1. Что означает Nike? – [Электронный ресурс] / – Режим доступа:

<https://yandex.ru/q/life/2135704065/>

1. Lamoda – [Электронный ресурс] / – Режим доступа:

<https://www.retail.ru/rbc/tradingnetworks/lamoda/>

1. Kari (компания) – [Электронный ресурс] / – Режим доступа:

<https://ru.ruwiki.ru/wiki/Kari_(компания)>

1. Фронтенд и Бэкенд: в чём разница и как они работают вместе?– [Электронный ресурс] / – Режим доступа:

<https://prog.academy/ru/blog/frontend-and-backend>

1. Что такое HTML?– [Электронный ресурс] / – Режим доступа:

<https://kotazzz.github.io/p/html01/>

1. Что такое PhpUnit в PHP?– [Электронный ресурс] / – Режим доступа:

<https://itproger.com/spravka/php/phpunit>

1. ОП.08 - 02 - Реляционные базы данных. Популярные СУБД. Понятие связи, типы связей– [Электронный ресурс] / – Режим доступа: [https://it.vshp.online/#/pages/op08/op08\_02](https://it.vshp.online/%23/pages/op08/op08_02)
2. ОП.08 - 03 - SQL. СУБД MySQL. MySQL Workbench. Создание и удаление БД– [Электронный ресурс] / – Режим доступа: <https://it.vshp.online/#/pages/op08/op08_03>
3. 10.Что такое phpMyAdmin– [Электронный ресурс] / – Режим доступа: <https://mchost.ru/articles/chto-takoe-phpmyadmin/>

# **Приложение 1**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.

# **Приложение 2**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

# **Приложение 3**

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

# **Приложение 4**

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, число

Контент, сгенерированный ИИ, может содержать ошибки.