Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Магнитогорский государственный технический университет   
им Г.И. Носова»

Многопрофильный колледж

Отделение №2 Информационные технологии и транспорт

**КУРСОВОЙ ПРОЕКТ**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

по ПМ.09. Проектирование, разработка и оптимизация веб-приложений

МДК.09.01. Проектирование и разработка веб-приложений

на тему: Разработка веб-приложения по анализу и цен в интернет-магазинах

Исполнитель: Хайсаров.В.И . студент 4 курса, группа ИспВ-20-1

Руководитель: Зорина И.Г., преподаватель МпК

Работа допущена к защите «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Магнитогорск, 2023

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Магнитогорский государственный технический университет   
им. Г. И. Носова»

Многопрофильный колледж

**КУРСОВОЙ ПРОЕКТ**

Тема Разработка веб-приложения по анализу цен в интернет-магазинах

Студент Хайсаров Владислав Ильнурович

Задание

Исходные данные:

по заданной предметной области:

1. выполнить моделирование информационной системы;
2. спроектировать и разработать базу данных;
3. разработать серверную и клиентскую части приложения;
4. разграничить права доступа;
5. выполнить тестирование информационной системы.

Состав и содержание проекта:

1. построение UML-диаграмм: вариантов использования, последовательности, состояния, деятельности, классов; потоков данных;
2. логическое проектирование, описание концептуальной модели данных, разработка базы данных в выбранной СУБД;
3. разработка интерфейса приложения, логики работы приложения: оформление заявок на услуги, вывод отчетной документации и т.д.;
4. администрирование приложения, разработка панели администратора;
5. разработка комплекта тестирования информационной системы.

Срок сдачи: « » 20 г.

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /И.Г. Зорина

« » 2023 г.

Задание получил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / В.И.Хайсаров

« » 2023г.

Магнитогорск, 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc152761985)

[1 ТЕОРЕТИКО-ИНФОРМАЦИОННЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc152761986)

[1.1 Анализ предметной области 7](#_Toc152761987)

[1.2 Анализ и оценка существующих программных продуктов 8](#_Toc152761988)

[1.2.1 Сайт Sravni.com 8](#_Toc152761989)

[1.2.2 Сайт shops-prices.ru 9](#_Toc152761990)

[1.2.3 Сайт shopoclick 10](#_Toc152761991)

[1.2.3 Сайт AllPricer 11](#_Toc152761992)

[1.2.3 Сайт FoodsPrice.ru 11](#_Toc152761993)

[1.3 Моделирование информационной системы 13](#_Toc152761994)

[1.4 Логическое и физическое проектирование баз данных 18](#_Toc152761995)

[1.5 Выбор и обоснование средств разработки 21](#_Toc152761996)

[2.1 Создание базы данных 23](#_Toc152761997)

[2.2 Логическая и физическая структура сайта 25](#_Toc152761998)

[2.3 Разработка интерфейса пользователя 28](#_Toc152761999)

[2.4 Разграничение прав доступа 33](#_Toc152762000)

[2.5 Руководство пользователя 35](#_Toc152762001)

[ЗАКЛЮЧЕНИЕ 40](#_Toc152762002)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 41](#_Toc152762003)

[ПРИЛОЖЕНИЕ А 42](#_Toc152762004)

# ВВЕДЕНИЕ

Сравнение цен на различные товары - важный аспект современной потребительской культуры. В условиях насыщенного рынка товарами и услугами выбор оптимальной цены становится сложной задачей.

Создание сайта для сравнения цен на различные товары является актуальной и полезной идеей в электронной коммерции.

В настоящее время все больше людей используют интернет для покупок, и они хотят быть уверены, что получают наилучшую цену за товары.

Актуальность данной темы обусловлена необходимостью экономии денег и выбором наиболее выгодного предложения. В ситуации, когда цены на товары разнятся в разных магазинах или онлайн-платформах, сравнение цен позволяет принимать обоснованные решения и выбирать наиболее выгодное предложение.

Это помогает сократить расходы на покупки и получить больше ценностей за средства. Для продавцов сравнение цен является инструментом конкурентного анализа. Могут отслеживать цены своих конкурентов и адаптировать стратегии ценообразования, чтобы привлечь больше клиентов. Сравнение цен на различные товары полезно не только для потребителей, но и для продавцов.

Потребители могут легко сопоставлять цены на товары из разных магазинов или онлайн-платформ, выбирая наиболее выгодное предложение. Это помогает сократить расходы на покупки и получить больше ценностей за средства. Для продавцов сравнение цен является инструментом конкурентного анализа. Могут отслеживать цены своих конкурентов и адаптировать стратегии ценообразования, чтобы привлечь больше клиентов.

Таким образом, сравнение цен на различные товары актуально и полезно как для потребителей, так и для продавцов. Позволяет принимать более обоснованные решения при покупках, снижает затраты на потребление и способствует развитию конкуренции на рынке.

Целью данной работы является сайт для сравнения цен на различные товары, помогая пользователям экономить время, деньги и сделать более обоснованный выбор при покупках.

Для достижения данной цели были поставлены следующие задачи:

* составить анализ предметной области;
* создать модель информационной системы;
* разработать базу данных;
* разработать интерфейс пользователя;
* разработать выходные документы;
* создать разграничение прав доступа;
* разработать руководство пользователя;

Практическая значимость темы поиска товаров заключается в том, что приложение помогает потребителям совершать осознанные и обоснованные покупки. При сравнении цен на различные товары, покупатели могут выбрать наиболее выгодное предложение и сэкономить деньги

Также, для продавцов и предпринимателей понимание динамики рынка и конкуренции необходимо для разработки эффективных стратегий ценообразования и повышения конкурентоспособности их товаров. Анализ цен помогает следить за динамикой изменения цен конкурентов, понимать предпочтения покупателей и адаптировать свою стратегию продаж. Таким образом, тема поиска товаров имеет практическую значимость как для потребителей, так и для продавцов. Сайт помогает принимать обоснованные решения при покупках, сокращать затраты и повышать конкурентоспособность на рынке.

# 1 ТЕОРЕТИКО-ИНФОРМАЦИОННЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Информационная система – это взаимосвязанная совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств.

Информационная система предназначена для удовлетворения конкретных информационных потребностей в рамках определенной предметной области, при этом результатом функционирования информационных систем является информационная продукция – документы, информационные массивы, базы данных и информационные услуги.

Структуру информационных систем составляет совокупность отдельных частей, называемых подсистемами.

Функциональные подсистемы реализуют и поддерживают модели, методы и алгоритмы получения управляющей информации. Состав функциональных подсистем весьма разнообразен и зависит от предметной области использования информационной системы, специфики хозяйственной деятельности объекта, управления.

Информационная система представляет собой среду, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технологические и программные средства. А информационная технология есть совокупность операций и действий над данными. Все процессы преобразования информации в информационной системе осуществляются с помощью информационных технологий.

# 1.1 Анализ предметной области

Предметная областью данного сайта по сравнению цен на товары является электронная коммерция и розничная торговля. На сайте предоставляется платформа, где пользователи могут сравнивать цены на различные товары, принимать информированные решения и выбирать наилучшие предложения. Анализ данной предметной области:

Товары и категории:

– Сайт включает в себя разнообразные товары, охватывая широкий спектр категорий, таких как электроника, одежда, обувь, бытовая техника и многое другое.

– Товары допускаются быть разделенными на подкатегории для удобства пользователей при поиске.

Цены и сравнение:

– Основной функционал сайта - предоставление возможности сравнивать цены на один и тот же товар у различных продавцов.

– Пользователи могут видеть актуальные цены, скидки и другие предложения от разных продавцов.

Пользовательский интерфейс:

– Сайт обладает интуитивно понятным и удобным интерфейсом для пользователей.

– Поисковые и фильтрационные опции предоставляют быстрый доступ к нужной информации.

Пользователи:

– Существует две основные роли пользователей: покупатели и администраторы.

– Покупатели могут просматривать товары, сравнивать цены, добавлять товары в корзину и совершать покупки.

– Администраторы управляют контентом, обновляют цены и следят за правильностью представленной информации.

Регистрация и аутентификация:

–Пользователи могут создавать учетные записи для сохранения персональных настроек, отслеживания заказов и получения персонализированных рекомендаций.

Обратная Связь и Отзывы:

– Сайт может включать возможность оставлять отзывы и рейтинги для товаров и продавцов, что помогает другим пользователям принимать более информированные решения.

Безопасность и конфиденциальность:

– Важными аспектами являются безопасность транзакций и конфиденциальность личной информации пользователей.

Услуги доставки:

– Пользователям может быть предложен выбор услуг доставки с указанием времени и стоимости.

Аналитика и управление сайтом:

– Администраторы имеют доступ к инструментам аналитики, которые позволяют им следить за активностью пользователей, эффективностью продаж и общим состоянием сайта.

Обновление и улучшение:

– Сайт должен регулярно обновляться и улучшаться, чтобы следовать за изменениями в рынке и предоставлять лучший опыт пользователям.

– Обеспечив удобство использования, точность информации и безопасность, сайт для сравнения цен на товары может стать незаменимым инструментом для онлайн-шопинга и привлечения пользователей.

1.2 Анализ и оценка существующих программных продуктов

Существует множество веб-ресурсов, посвященных сравнению товаров. Многие из них принадлежат крупным компаниям, но даже в таком случае не всегда предоставляется удобный и простой в использовании интерфейс. Вот несколько из них.

# 1.2.1 Сайт Sravni.com

Sravni.com- это сервис, который помогает выгодно совершать покупки, затрачивая на онлайн-шоппинг минимум времени. Объединяя в единую базу актуальную информацию, получаемую от множества интернет-магазинов России, среди содержимого категорий, представленных на сайте, найдутся подходящие предложения для каждого пользователя. И это касается не только цены, но и технических характеристик товаров. Для тех, кто не определился с производителем и моделью товара, полезными окажутся функции «Подбор моделей» и «Хиты продаж». К каждой просматриваемой позиции каталог цен Sravni.com предлагает несколько аналогичных вариантов. Услуга «Прайс-контроль» делает сервис еще более комфортным: зайдя в карточку товара, укажите в соответствующем разделе свой электронный адрес и желаемую цену. Когда стоимость достаточно снизится, пользователь будет проинформированы сообщением (Рисунок 1).

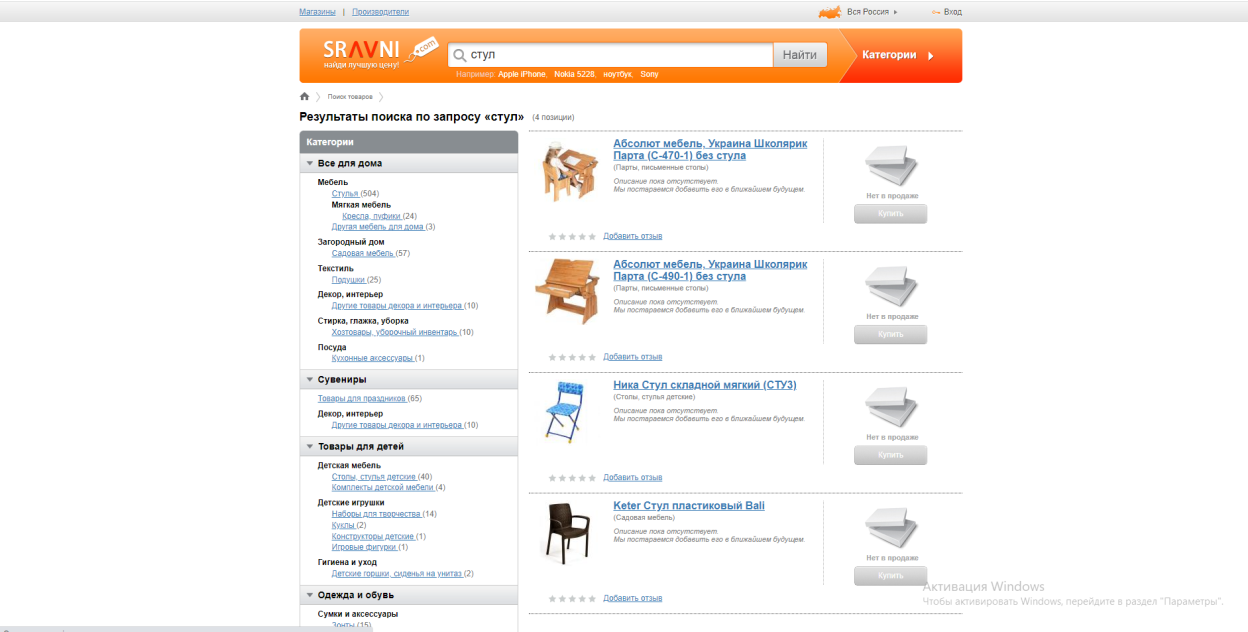


Рисунок 1 Sravni.ru

# 1.2.2 Сайт shops-prices.ru

shops-prices.ru многофункциональный сервис позволяет оперативно находить интересующий товар и сравнить цены на него в различных магазинах Москвы и всей России. В настоящее время ресурс охватывает следующие группы товаров: бытовая техника, компьютеры, ноутбуки, планшеты, электроника, товары для дома и сада, спортивные товары, авто товары. Так же сервис не является посредником между покупателем и продавцом. Покупка товара совершается непосредственно на сайте или в реальном магазине, выбранного продавца. Основная цель – это быстрое сравнение цен. Сервис дает возможность найти наиболее выгодное предложение и совершить обдуманную покупку. На странице каждого товара представлена полная информация, включающая технические характеристики, фото, описание и отзывы покупателей. Это позволяет принять правильное решение относительно выбора модели. Так же на странице появляется список магазинов, реализующих интересующий товар и цены в порядке возрастания.

Технологии сравнения цен постоянно совершенствуются, поэтому сайт регулярно обновляется. Благодаря постоянному мониторингу рынка расширяется каталог товаров и список предложений от магазинов, их реализующих. Сервис shops-prices.ru дает возможность составить полное и достоверное мнение о предложениях на рынке. Поисковая система оптимизирует выбор покупки. Справочник по ценам является выгодным, как для покупателей, так и для продавцов (Рисунок 2).

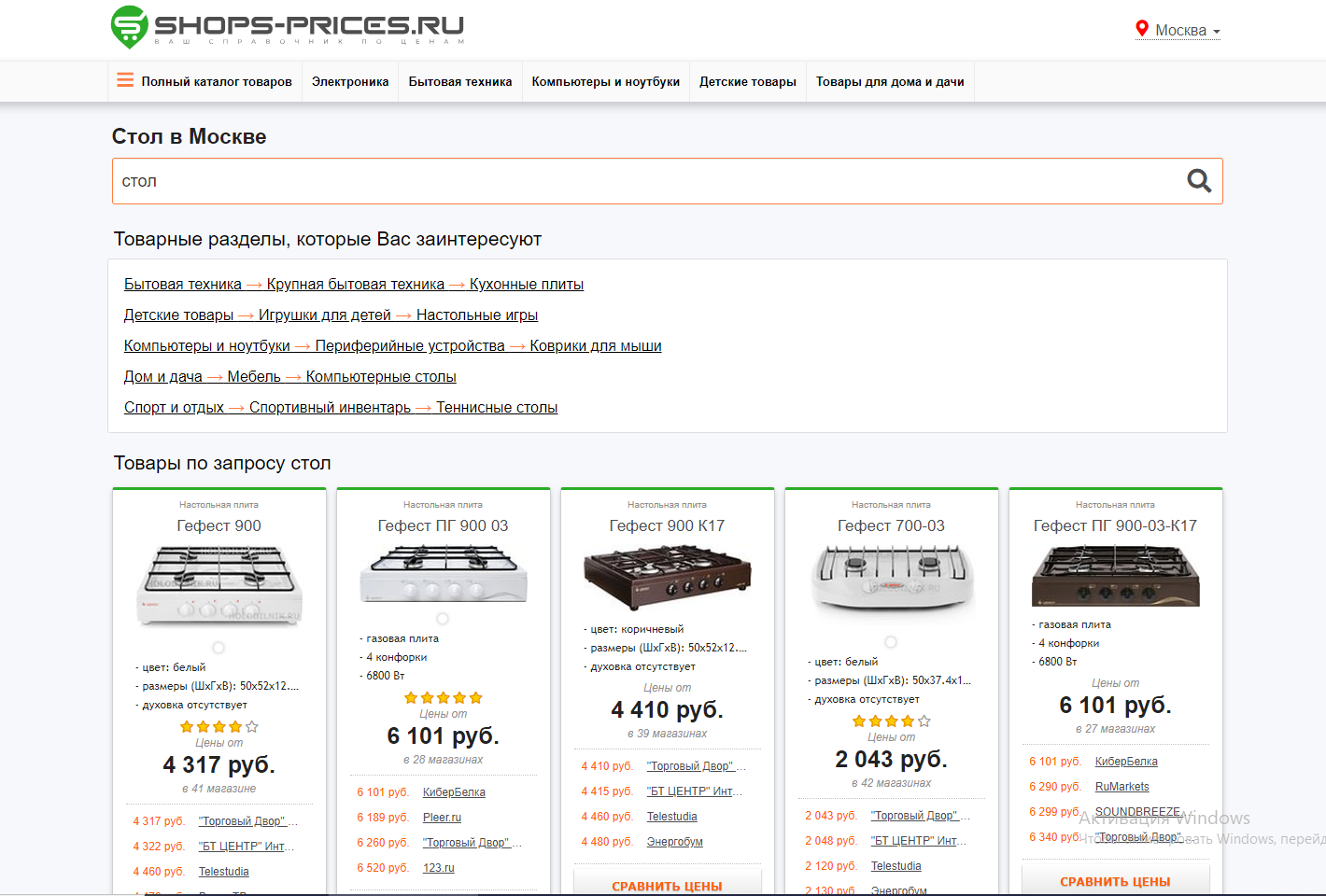


Рисунок 2 shops-prices

# 1.2.3 Сайт shopoclick

Shopoclick это сайт где пользователь может быстро сравнить цены на нужные товары в магазинах или онлайн-гипермаркетах, выбрать наилучшие и сохранить их. Реальные цены на товары, а не цены "по акциям", которые зачастую выше обычных в другом магазине.

Сайт в процессе наполнения, постоянно появляются новые города и разделы с товарами (Рисунок 3).



Рисунок 3 Шопоклик

# 1.2.3 Сайт AllPricer

AllPricer — адаптированный сервис по поиску товаров. На сайте представлены все доступные скидки и акции популярных интернет-магазинов России. Клиент может провести сравнение цен и выбрать наиболее подходящее предложение.

Многофункциональная платформа предлагает клиентам участие в аукционных и скидочных предложениях. Совершается массовая рассылка, оповещая клиентов о крупных распродажах. На сайте собраны все доступные акции и скидки, которые постоянно обновляются (Рисунок 4).

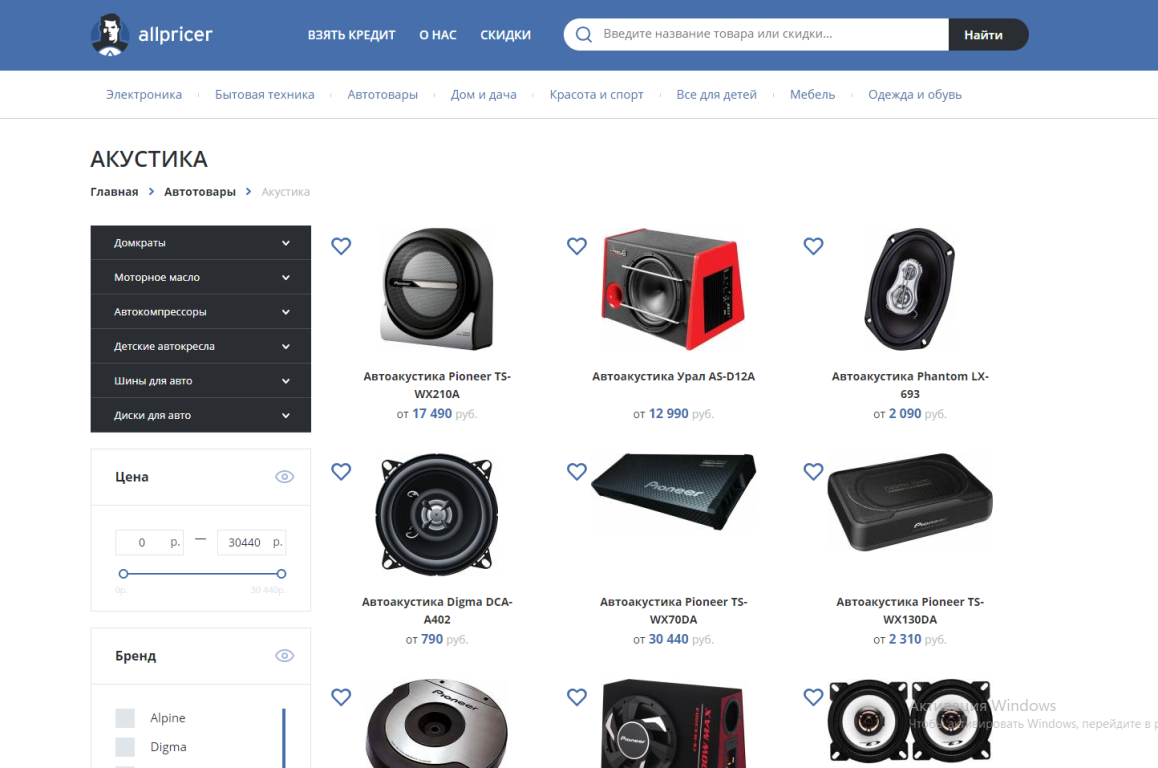


Рисунок 4 allpricer

# 1.2.3 Сайт FoodsPrice.ru

FoodsPrice.ru - информационно-аналитический проект, на котором любой покупатель может сформировать собственную продуктовую корзину и сравнить цены на продукты по большинству магазинов, выбирая наиболее оптимальные. Пятибалльная шкала оценки продуктов помогает выбрать наиболее рейтинговые в заданной категории, а мониторинг цен - наиболее оптимальные места для покупки. (Рисунок 5)

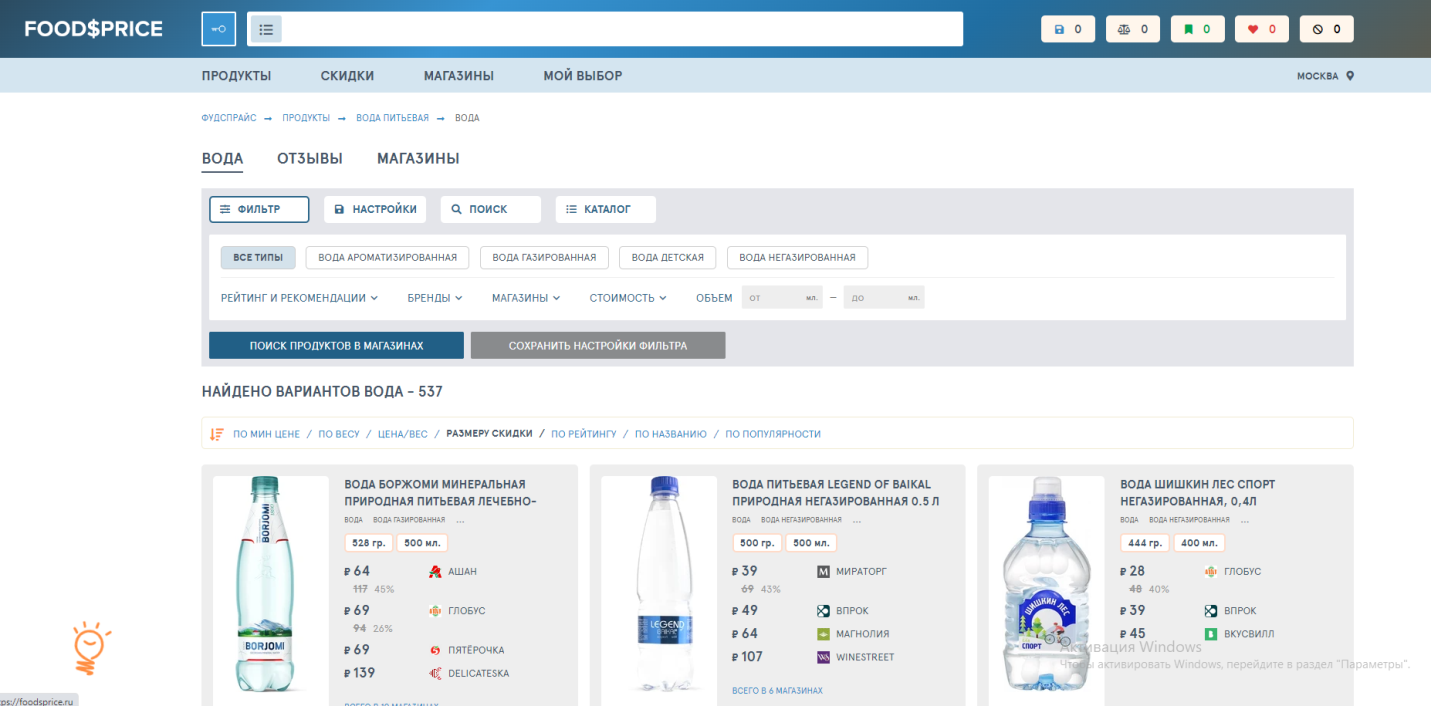


Рисунок 5 foodsprice

При анализе различных существующих сайтов для сравнения товаров, можно выделить определенные недостатки и отсутствующие элементы, которые вдохновили на решение создать собственный ресурс. Вот ключевые моменты, которые могли бы послужить мотивацией:

– Уникальные функции: Ни один из рассмотренных сайтов не предоставляет полный набор уникальных функций, которые могли бы значительно улучшить опыт пользователей. Создание собственного сайта дает возможность инноваций и внедрения уникальных возможностей.

– Комплексное сравнение: Некоторые сайты сосредотачиваются только на определенных категориях товаров или предоставляют ограниченный выбор функций для сравнения. Решение создать собственный ресурс позволяет охватить более широкий спектр категорий и предоставить полный инструментарий для сравнения.

– Новый уровень информативности: Существующие сайты могут не предоставлять достаточно подробной информации о товарах, что затрудняет процесс принятия решения. Новый сайт может сосредоточиться на увеличении информативности страниц товаров, включая дополнительные детали, сравнительные обзоры и прочие полезные сведения.

– Глобальное покрытие: Возможно, некоторые из существующих ресурсов ориентированы только на определенные регионы или страны. Создание сайта с глобальным охватом может привлечь большую аудиторию и обеспечить широкий выбор товаров.

– Инновационные методы монетизации: Возможно, видно пространство для внедрения новых моделей монетизации, которые будут выгодны как для владельцев сайта, так и для пользователей. Это может включать в себя новые партнерские программы, рекламные стратегии или другие инновационные подходы.

– Адаптация к изменениям рынка: Решение создать собственный сайт предоставляет гибкость для быстрой адаптации к изменениям в потребительском поведении, технологиях и требованиях рынка.

Таким образом, создание собственного сайта для сравнения товаров обосновано желанием предоставить уникальный и усовершенствованный опыт пользователя, а также эффективно конкурировать на рынке с учетом выявленных недостатков существующих ресурсов.

# 1.3 Моделирование информационной системы

Моделирование – это процесс создания абстрактной модели реальной системы или процесса, с целью изучения свойств и поведения. Моделирование информационной системы включает в себя создание модели, которая отображает информационные потоки, функциональные блоки и связи между ними.

Эта модель может быть использована для анализа и оптимизации работы информационной системы, а также для принятия решений по улучшению. В результате моделирования может быть разработана более эффективная и оптимальная информационная система, что позволяет снизить затраты на эксплуатацию и повысить качество предоставляемых услуг.

Диаграмма вариантов UML – это инструмент для моделирования системы, который отображает взаимодействие компонентов с актерами (пользователями системы) в различных сценариях использования.

Диаграмма вариантов помогает определить функциональные требования к системе и описать ситуации, в которых пользователь взаимодействует с сайтом. Это позволяет разработчикам лучше понимать, как система должна работать и как взаимодействовать с пользователем. Диаграмма вариантов может использоваться на стадии проектирования, а также как инструмент для общения между разработчиками и заказчиком проекта.

Диаграмма вариантов (рисунок 6).

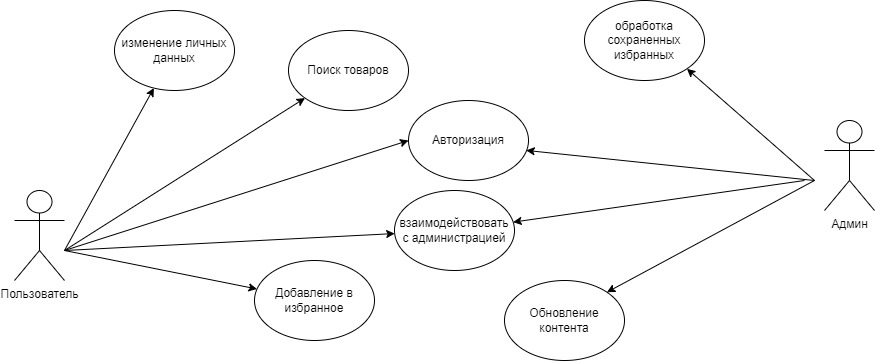
Рисунок 6 – Диаграмма вариантов

Диаграмма состояний UML – это инструмент моделирования системы, который отображает изменения состояний объектов в различных ситуациях.

Диаграмма состояний помогает описать поведение объекта в системе, где объект может находиться в различных состояниях и переходить между ними в зависимости от входных данных или событий. Диаграмма может помочь разработчикам лучше понять, как объекты будут взаимодействовать в системе, и как изменения состояний объектов будут влиять на работу системы.

Диаграмма состояний часто используется на стадии анализа и проектирования системы, а также в качестве инструмента общения между заказчиком и разработчиками. Может быть особенно полезна для систем, где объекты могут иметь сложное поведение и должны переходить между многими состояниями. Диаграмма состояний (рисунок 7).

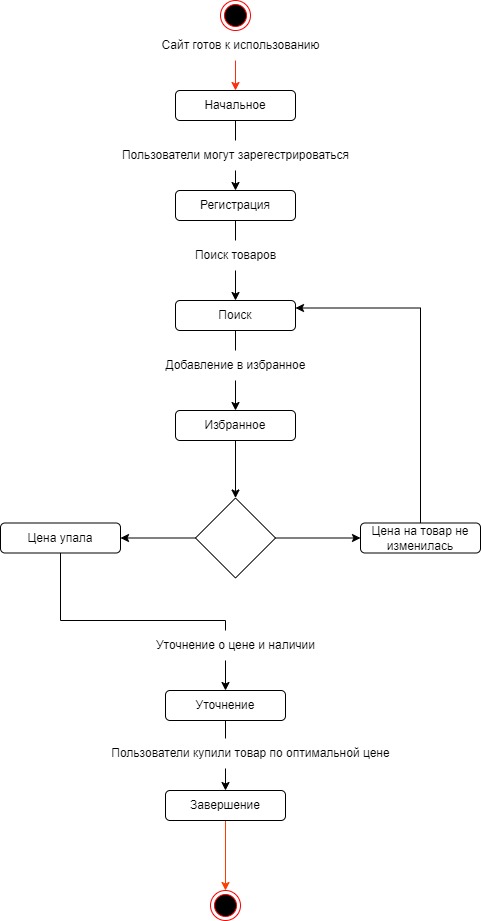


Рисунок 7 Диаграмма состояний

Рисунок 8 –Диаграмма состояний

Диаграмма деятельности UML – это инструмент моделирования системы, который отображает последовательность действий или операций, выполняемых в рамках процесса или деятельности в системе

Диаграмма деятельности помогает описать процессы и действия, которые будут выполняться в системе, и предоставляет визуальное представление последовательности выполнения действий. Может использоваться для моделирования многих видов процессов, включая бизнес-процессы, процессы разработки программного обеспечения и т. д.

Диаграмма деятельности может быть полезна на разных стадиях разработки системы, от определения требований до тестирования и сопровождения. Может помочь разработчикам и другим участникам проекта лучше понимать процессы и их взаимосвязь, а также использоваться для документирования процессов и общения между различными заинтересованными сторонами. Диаграмма деятельности (рисунок 8).

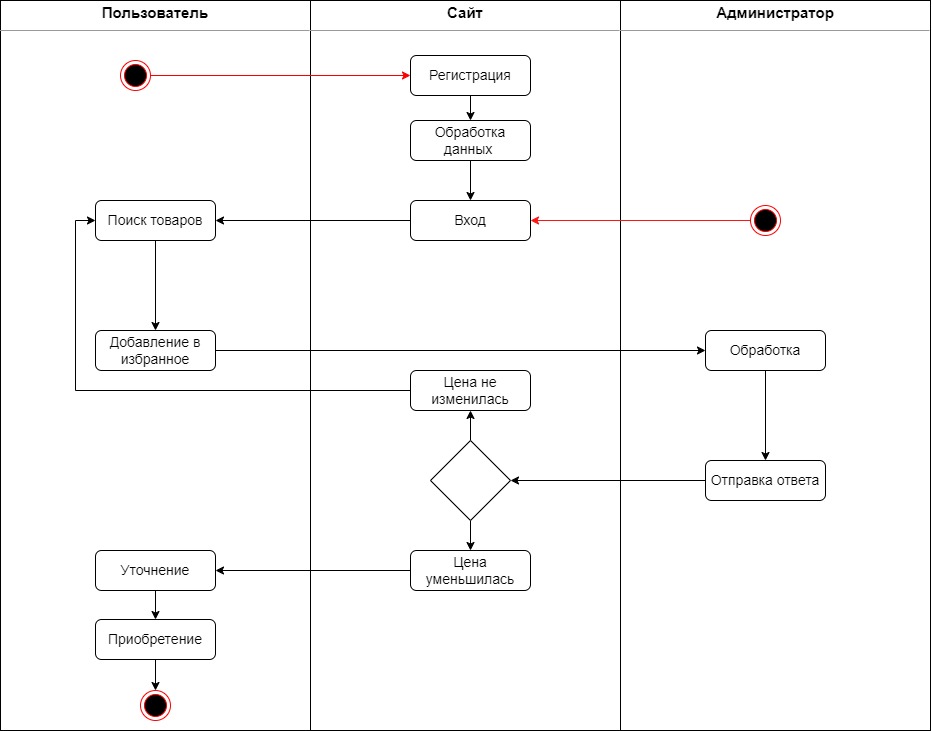


Рисунок 8 –Диаграмма деятельности

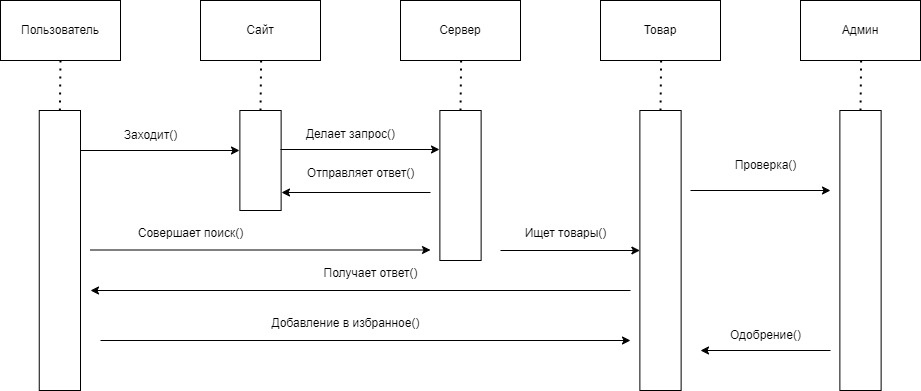
Диаграмма последовательности UML – это инструмент моделирования системы, который позволяет показать, как объекты взаимодействуют друг с другом в рамках определенной операции или процесса.

Диаграмма последовательности отображает последовательность шагов, которые объекты выполняют во время операции, и учитывает время, необходимое для выполнения каждого шага. Используется для описания сценариев использования системы, где объективное поведение играет ключевую роль, а также для моделирования взаимодействия между объектами в системе.

Диаграмма последовательности может быть полезна для разработчиков, чтобы лучше понять взаимодействие между объектами и понять, как действовать в конкретных ситуациях. Помогает в разработке приложений, где важно понимание взаимодействия между объектами, особенно если это сложное взаимодействие.

Как и другие диаграммы UML, диаграмма последовательности может использоваться на разных этапах процесса разработки, от определения требований до создания кода и тестирования.

Диаграмма последовательности (рисунок 9)

Рисунок 9 –Диаграмма последовательности

# 1.4 Логическое и физическое проектирование баз данных

Проектирование базы данных – это итерационный процесс, который имеет свое начало, но не имеет конца и состоит из бесконечного ряда уточнений. Это следует рассматривать, прежде всего, как процесс познания. Как только проектировщик приходит к пониманию работы предприятия и смысла обрабатываемых данных, а также выражает это понимание средствами выбранной модели данных, приобретенные знания могут показать, что требуется уточнение и в других частях проекта. Особо важную роль в общем процессе успешного создания системы играет концептуальное и логическое проектирование базы данных.

На этапе проектирования, прежде всего, формируются модели данных. Проектировщики в качестве исходной информации получают результаты анализа. Построение логической и физической моделей данных является основной частью проектирования базы данных. Полученная в процессе анализа информационная модель сначала преобразуется в логическую, а затем в физическую модель данных.

Параллельно с проектированием схемы базы данных выполняется проектирование процессов, чтобы получить спецификации (описания) всех модулей информационной системы. Оба эти процесса проектирования тесно связаны, поскольку часть бизнес–логики обычно, реализуется в базе данных (ограничения, триггеры, хранимые процедуры). Главная цель проектирования процессов заключается в отображении функций, полученных на этапе анализа, в модули информационной системы. При проектировании модулей определяют интерфейсы программ: разметку меню, вид окон, горячие клавиши и связанные с ними вызовы.

Физическое проектирование базы данных – процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах, на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты.

Физическое проектирование является вторым этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Во время предыдущего этапа проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области). Хотя эта структура не зависит от конкретной целевой системы управления базами данных, создается с учетом выбранной модели хранения данных, например, реляционной, сетевой или иерархической. Поэтому физическое проектирование неразрывно связано с конкретной системой управления базами данных. Между логическим и физическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе физического проектирования с целью повышения производительности системы, способны повлиять на структуру логической модели данных.

Как правило, основной целью физического проектирования базы данных является описание способа физической реализации логического проекта базы данных. В случае реляционной модели данных под этим подразумевается следующее:

– создание набора реляционных таблиц и ограничений на основе информации, представленной в глобальной логической модели данных;

– определение конкретных структур хранения данных и методов доступа, обеспечивающих оптимальную производительность;

разработка средств защиты создаваемой системы;

– этапы концептуального и логического проектирования больших систем следует отделять от этапов физического проектирования;

–Связаны с совершенно разными аспектами системы, поскольку отвечают на вопрос, что делать, а не как делать;

– Выполняются в разное время, поскольку понять, что надо сделать, следует прежде, чем решить, как это сделать;

– Требуют совершенно разных навыков и опыта, поэтому требуют привлечения специалистов различного профиля.

Логическое проектирование базы данных – это процесс создания модели используемой на предприятии информации на основе выбранной модели организации данных, но без учета типа целевой системы управления базами данных и других физических аспектов реализации.

Второй этап проектирования базы данных называется логическим проектированием базы данных. Цель состоит в создании логической модели данных для исследуемой части предприятия. Концептуальная модель данных, созданная на предыдущем этапе, уточняется и преобразуется в логическую модель данных (например, реляционная модель).

Если концептуальная модель данных не зависит от любых физических аспектов реализации, то логическая модель данных создается на основе выбранной модели организации данных. Иначе говоря, на этом этапе уже должно быть известно, какая система управления базами данных будет использоваться в качестве целевой – реляционная, сетевая, иерархическая или объектно-ориентированная. Однако на этом этапе игнорируются все остальные характеристики, например, любые особенности физической организации ее структур хранения данных и построения индексов.

В процессе разработки логическая модель данных постоянно тестируется и проверяется на соответствие требованиям пользователей.

Сущность – любо различимый объект реального мира, информацию о котором необходимо хранить в базе данных. Сущность имеет поименованную характеристику - атрибут сущности. Наименование атрибута должно быть уникальным для конкретного типа сущности, но может быть одинаковым для сущности разных типов.

Связь – ассоциирование двух или более сущностей, если бы назначением базы данных было только хранение отдельных, не связанных между собой данных, то структура могла бы быть очень простой. Но одно из основных требований к организации баз данных - это обеспечение возможности отыскания одних сущностей по значениям других, для чего необходимо установить между ними определенные связи.

Метод «Сущность-связь» заключается в выявлении сущностей и установления связей между ними. Каждая сущность имеет идентификатор, который полностью определяет экземпляр сущности и характеристики. Связь также может иметь характеристики.

Существуют следующие типы связей:

– один к одному (связь, в которой существование одной сущности на одной стороне связи зависит от существования одной сущности на другой стороне связи);

– один ко многим (связь, в которой существование n – сущностей на одной стороне связи зависит от существования одной сущности на другой стороне связи);

– многие ко многим (связь, в которой существование n – сущностей на одной стороне связи зависит от существования n – сущностей на другой стороне связи).

При анализе предметной области были выделены следующие сущности:

Users;

Favorites;

На рисунке 10 представлена физическая модель базы данных.

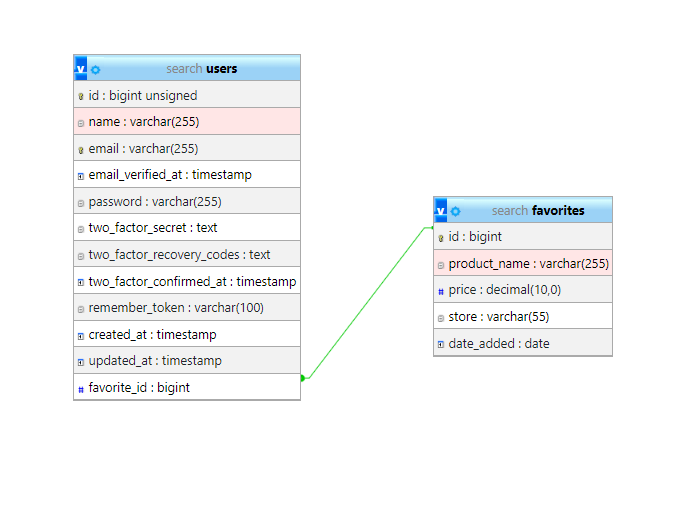


Рисунок 10 Схема базы данных

Таблица users (пользователь) содержит информацию о пользователе и является главной по отношению к таблице favorites (избранные).

Структура таблицы users:

* id – код категории (ключевое поле);
* name – имя пользователя;
* email – email пользователя;
* password –пароль;
* role - роль;

Таблица favorites (избранные) содержит информацию о избранных товарах пользователя

Структура таблицы favorites:

* id – код категории (ключевое поле);
* product\_name –имя продукта;
* price – цена;
* store – магазин;
* date\_added – дата;

# 1.5 Выбор и обоснование средств разработки

В качестве средства разработки для веб–сайта выбран фреймворк PHP – Laravel. Фреймворк гарантирует хорошую производительность, отдельно стоит упомянуть кэширование. Благодаря соответствующему драйверу файловая система сохраняет в себе большое количество различных элементов. Подобный подход способствует более быстрой разработке самых разных по сложности приложений. Система аутентификации в Laravel очень удобна, при использовании подобного можно даже контролировать доступ к имеющимся ресурсам. То есть неавторизованные пользователи, как говорится, не останутся незамеченными. Другие особенности фреймворка Laravel.

Безопасность. Laravel не только предотвращает вредоносную деятельность и попытки взлома веб–приложений, но и защищает код от манипуляций злоумышленниками.

Возможность использования шаблонов Blade. Это мощный инструмент, который позволяет веб–разработчикам использовать заранее созданные структурные шаблоны. Шаблонизация ускоряет создание веб–приложений, и при этом не ограничивает разработчиков в написании собственного кода. Использовать шаблоны Blade просто, понятно и эффективно. Движок Blade – легкий и не влияет на производительность создаваемого приложения.

Миграция баз данных. Эта процедура упрощает процесс создания и управления базами данных, и разработчик может отменить изменения, внесенные в таблицы. Программисты могут легко обмениваться базами данных с другими разработчиками, что делает совместную работу более согласованной, организованной и простой. Функция сопоставления обеспечивает простое обслуживание баз данных без необходимости их повторного создания.

Возможность использования объектно-ориентированных библиотек. Среди них, к примеру, есть библиотека аутентификации, в которую включено множество полезных функций. Все библиотеки просты в использовании. Фреймворк также предоставляет дополнительные модули, которые позволяют разработчикам делать приложения более отзывчивыми и структурированными. Некоторые модули по функциональности равнозначны готовым приложениям.

Скорость разработки. Laravel не только обеспечивает производительность и функциональность, но также делает процесс разработки намного быстрее и проще.

Помимо преимуществ PHP-фреймворка Laravel, интеграция JavaScript может значительно расширить функциональность веб-приложения и улучшить пользовательский опыт. JavaScript является мощным языком программирования, который выполняется в браузере пользователя и предоставляет возможности для создания интерактивных и динамичных элементов на веб-страницах.

Интерактивные элементы:

– Добавление JavaScript позволяет создавать интерактивные элементы, такие как выпадающие списки, модальные окна, карусели, и другие, что может значительно улучшить пользовательский интерфейс.

Валидация форм на стороне клиента:

– JavaScript может использоваться для валидации данных форм непосредственно на стороне клиента до отправки на сервер. Это улучшает пользовательский опыт, предотвращает лишние запросы и снижает нагрузку на сервер.

Обработка событий:

– JavaScript позволяет легко обрабатывать события, такие как клики, отправка форм, перемещение мыши и другие. Это открывает дверь для реализации динамического поведения веб-страниц.

Анимации:

– Использование JavaScript для создания анимаций может сделать веб-приложение более привлекательным и динамичным, что важно для улучшения визуального восприятия.

Итак, интеграция JavaScript в веб-приложение на основе Laravel может значительно усилить его функциональность, создавая более динамичный и привлекательный пользовательский опыт.

# 2.1 Создание базы данных

База данных - это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Базы данных активно используются для динамических сайтов со значительными объемами данных — часто это интернет-магазины, порталы, корпоративные сайты. В данном случае это база данных для детского сада.

Одной из целей данного курсового проекта явилось создание базы данных. База данных должна обеспечивать удобность и отвечать за скорость ориентирования на сайте, чтобы вся информация была структурирована и доступна.

Данная база данных представляет собой информацию о пользователях и их избранных

Создание таблицы «пользователь». Характеристики таблицы представлены в   
таблице 1.

 Schema::create('users', function (Blueprint $table) {

         $table->id();

         $table->string('name');

         $table->string('email')->unique();

         $table->string('password');

        });

Таблица 1- Характеристики полей таблицы «users»( пользователь)

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Свойство поля |
| id | BIGINT(11) | Ключевое поле |
| Обязательное поле |
| счетчик |
| role | VARCHAR(25) | Обязательное поле |
| email | VARCHAR(50) | Обязательное поле |
| password | VARCHAR(255) | Обязательное поле |

Создание таблицы «favorites» Характеристики таблицы представлены в таблице 2

CREATE TABLE `favorites` (

Schema::create('favorites', function (Blueprint $table) {

     $table->id();

     $table->string('product\_name');

     $table->decimal('price');

     $table->string('store');

     $table->timestamp('date\_added')->useCurrent();

$table->foreignId('user\_id')->constrained('users')->cascadeOnDelete();

     $table->int('product\_id');

     $table->string('comments');

        });

Таблица 2 Характеристики полей таблицы «favorites»( избранное)

|  |  |  |
| --- | --- | --- |
| Имя столбца | Тип данных | Свойство поля |
| id | BIGINT | Ключевое поле |
| Обязательное поле |
| счетчик |
| product\_name | VARCHAR(50) | Не обязательное поле |
| price | DECIMAL(10.2) | Не обязательное поле |
| store | VARCHAR(50) | Не обязательное поле |
| date\_added | DATE | Не обязательное поле |
| user\_id | BIGINT | Не обязательное поле |
| product\_id | INT | Не обязательное поле |
| comments | VARCHAR(255) | Не обязательное поле |

# Логическая и физическая структура сайта

Логическая структура сайта - это система ссылочного взаимодействия между страницами виртуального ресурса. Это должно продумываться на самых ранних этапах разработки сайта и являться основой функциональности дизайна.

Логическая структура сайта - это не совокупность разрозненных страниц и графических ресурсов, а заранее структурированный контент с перекрестными ссылками между всеми страницами ресурса. Если структура реализована правильно, то эффективность веб-ресурса будет высокой.

Если структура сайта разработана правильно, то будет служить для посетителя проводником к информации, выложенной на ресурсе, и являющейся главным фактором, определяющим поведение пользователя. Разрабатывается, как правило, один раз, и в последующем остается без изменений, поэтому данному разделу сайт строения необходимо уделить особое внимание. Дело в том, что после индексации поисковиками вашего веб-сайта, адреса его страниц заносятся в базу данных поисковой системы. Если вы измените путь доступа хотя бы к некоторым разделам уже после индексации, то есть вероятность, что ресурс просто потеряется для ПС, и не будет выводиться на прежней позиции по запросу поисковика. Многие специалисты сайт строения рекомендуют использовать линейный пример структуры сайта, который очень прост для понимания и легко индексируется.

Физическая структура подразумевает алгоритм размещения физических файлов по поддиректориям папки, в которой опубликован ваш сайт. Физическая модель представлена на рисунках 11, 12, 13.

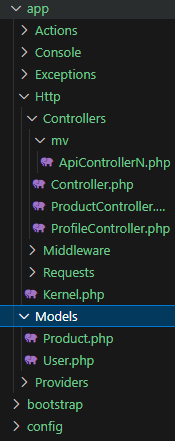


Рисунок 11 – Физическая структура сайта

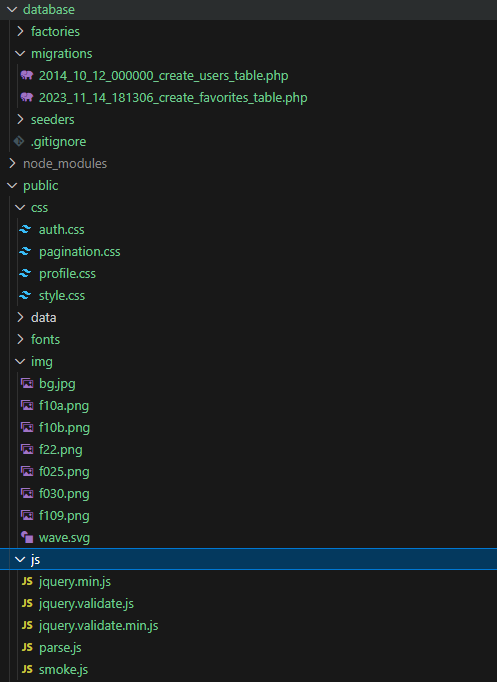


Рисунок 12 – Физическая структура сайта

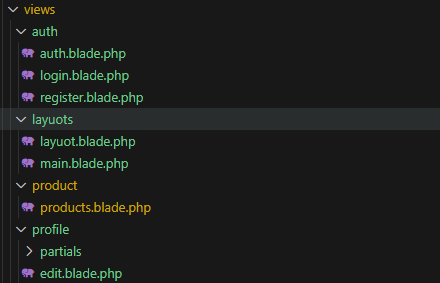


Рисунок 13 – Физическая структура сайта

# 2.3 Разработка интерфейса пользователя

Веб-интерфейс - веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера. Веб-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно — повсеместного распространения веб-браузеров.

Одним из основных требований к веб-интерфейсам является одинаковый внешний вид и одинаковая функциональность при работе в различных браузерах.

Классическим и наиболее популярным методом создания веб-интерфейсов является использование HTML с применением CSS и JavaScript. Однако различная реализация HTML, CSS, DOM и других спецификаций в браузерах вызывает проблемы при разработке веб-приложений и последующей поддержке. Кроме того, возможность пользователя настраивать многие параметры браузера (например, размер шрифта, цвета, отключение поддержки сценариев) может препятствовать корректной работе интерфейса.

UI-дизайн – User Interface, дословно переводится, как «Пользовательский интерфейс». Это определенный процесс визуализации, который позволяет реализовать прототип сайта, приложения или другого веб-ресурса. В первую очередь, UI включает активную работу над графической составляющей интерфейсом. Создаются анимации, иллюстрации, кнопки и другие элементы сайта, включая шрифты, цвета, формы.

Первоначально определяется пользовательский опыт и исследуется целевая аудитория. В зависимости от потребностей посетителей определяется цветовая палитра, формы, а также структура размещения объектов. Интерфейс пользователя должен быть понятным, и одна из главных задач UI-дизайнера – определить, удобен ли сайт визуально, легко ли попасть по кнопкам, хорошо ли читается текст.

Интерфейс - это набор, инструмент графической структуры приложения. Полностью состоит из кнопок, текстов, изображений, полей, и прочих элементов. От внешнего вида, удобства использования зависит конверсия сайта. Поэтому интернет магазины с плохим интерфейсом пользователя просто теряют своих клиентов.

UI-дизайнер делает так, чтобы взаимодействие пользователя с сайтом было гармоничным и эффективным. Проектирование пользовательского интерфейса строится на создании понятного визуала, чтобы посетитель сразу понял, как пользоваться продуктом (сайтом, интернет магазином).

Проектирование интерфейсов нужно не только для визуального оформления сайта. В первую очередь, это возможность удобно взаимодействовать с приложением и улучшать пользовательский опыт. UI нужен для выполнения следующих задач:

Привлечение посетителей. Пользователь оценивает внешний вид сайта и удобство использования в течение нескольких секунд. Если посетитель сразу находит то, зачем он пришел, устраивает визуализация и вообще, взаимодействие понятное и простое, пользователь останется. В противном случае сайт просто закрывается, и посетитель уходит к конкурентам.

Ускорение превращения посетителя в клиента. Если форма заказа простая, сайт стильно оформлен, вызывает доверие и положительные эмоции – повышается конверсия продаж. Клиенту приятнее находиться на страницах такого ресурса, нежели на сайте с перегруженным или неграмотным пользовательским интерфейсом.

Адаптация под различные устройства. Разработка интерфейса подразумевает создание адаптивного дизайна, который одинаково хорошо будет смотреться на различных устройствах. Владельцы некоторых интернет магазинов не задумываются, как ресурс будет отображаться на мобильных устройствах, хотя зачастую, преимущественная часть трафика идет именно со смартфонов.

Повышение уровня доверия. Создание фирменных цветов, шрифтов и других графических элементов делает интернет магазин узнаваемым и повышает уровень доверия пользователей. Постоянство – залог хорошего взаимодействия с клиентом.

UI нужен для того, чтобы наладить хорошие взаимоотношения с посетителем. Только так он становится клиентом. Зачастую владельцы интернет магазинов стараются сэкономить, отказываясь от услуг дизайнера и разработки пользовательского интерфейса, но такое решение приводит к большой ошибке. Сайт готов, реклама настроена, а продаж нет. А все потому, что пользователям просто неудобно или не нравится находиться на сайте. Но даже в таком случае можно все исправить.

Грань действительно тонкая, как правило, происходит взаимодействие UI и UX. Последний нужен для того, чтобы понять пользователя, а UI помогает применить полученные данные и сделать так, чтобы посетитель легко разобрался с продуктом.

Во время разработки пользовательского интерфейса необходимо детально изучить целевую аудиторию и удовлетворить все ее потребности. Поэтому разработка UI состоит из нескольких этапов:

Опираясь на исследования целевой аудитории, проанализировав потребности, поведение и желания клиента, дизайнер ставит задачи. Продумывается концепция пользовательского интерфейса, будущие элементы, цвета. Этот этап очень важен, и занимает значительную часть времени. Но именно во время разработки концепции решаются наиболее важные задачи.

Данный этап еще до разработки пользовательского интерфейса позволяет понять, как клиент будет видеть сайт. Тут же вносятся все изменения. Первоначально размечается расположение всех графических элементов, включая кнопки, формы, меню. Далее подбирается цветовая палитра с обязательным условием – все цвета должны быть гармоничны между собой. Подбираются изображения, их стилистика, и в итоге – создается макет.

Когда разрабатывается интерфейс приложения, дизайнер ориентируется на специальные правила оформления элементов, а также регламент, созданный компаниями Google и Apple. Поэтому труд немного облегчается.

Общий внешний вид

– Цветовая палитра**:** Черные цвета с серым

– Шрифты:Baron Neue-italic,Oswald-light.

Главная страница (рисунок 14)

– Простота использования**:** простая форма поиска, находящуюся в центре внимания. Подсказки для поисковых запросов, чтобы упростить ввод.

– Фильтры**:** Пользователю возможно легко применять фильтры для уточнения результатов, таких как категория товаров, ценовой диапазон и бренд.

Результаты поиска/Сравнения

– Отображение товаров: Список товаров информативен, содержит: название, цену и краткое описание.

– Цены и скидки: Выделена старая цена, и цена по скидке.

– Кнопки действия: Яркие кнопки для подробного просмотра товара, добавления в избранное и сравнения с другими товарами.

Страница продукта

**–** Информация о товаре: Подробные характеристики товара, включая размер, вес, материал и другие важные параметры.

**–** Добавление в избранное и сравнение: Пользователям возможно добавлять товар в избранное для последующего просмотра и в сравнение с другими товарами для выбора наилучшего варианта.

Корзина и Оформление заказа

– Прозрачность цен: Полная стоимость товара в избранных, включая все дополнительные расходы.

– Простота избранных: Удобные избранные, где пользователи могут удалять товары и видеть изменение цены.

Личный кабинет и Регистрация

– Личный кабинет: Личный кабинет с возможностью просмотра избранных, управления данными.

– Уведомления: Уведомления для информирования пользователей о скидках, акциях и статусе заказа.

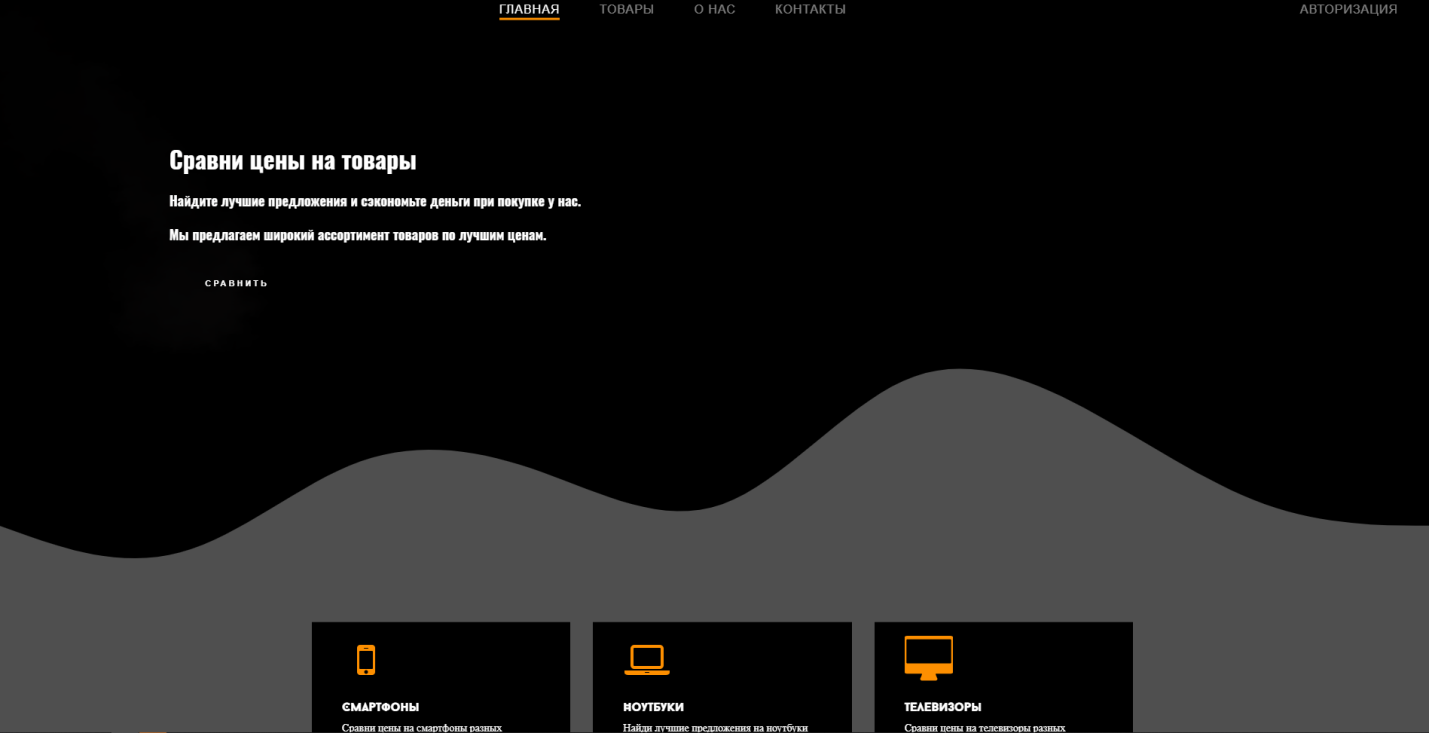


Рисунок 14 – Интерфейс на примере курсового сайта

**2.4 Разработка запросов**

В Laravel обычно используют Eloquent ORM вместо явных запросов MySQLi для более удобной и выразительной работы с базами данных. Вот пример, как вы можете переписать ваш текст, используя Eloquent в Laravel:

Модель User:

php artisan make:model User

Созданный файл User.php в папке app/Models:

app/Models/User.php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class User extends Model

{

protected $fillable = ['organization', 'number', 'email', 'password', 'role'];

}

Контроллер регистрации

app/Http/Controllers/AuthController.php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

class AuthController extends Controller

{

public function register(Request $request)

{

$request->validate([ 'organization' => 'required', 'number' => 'required', 'email' => 'required|email|unique:users,email', 'password' => 'required|min:6', ]);

$user = User::create([

'organization' => $request->input('organization'),

'number' => $request->input('number'),

'email' => $request->input('email'),

'password' => Hash::make($request->input('password')), 'role' => 'employer', ]);

return response()->json([ 'success' => true, 'role' => 'employer', 'userId' => $user->id, 'data' => 'Регистрация успешна',

]);

} }

Маршрут в routes/web.php:

use App\Http\Controllers\AuthController;

Route::post('/register',[AuthController::class,'register']);

Теперь есть пример, как можно реализовать регистрацию пользователя в Laravel, используя Eloquent и контроллер.

# 2.4 Разграничение прав доступа

Права доступа позволяют участникам иметь доступ к определенным функциям сайта курса или сайта проекта, в зависимости от ролей и того, какие функции назначил владелец или администратор сайта.

Роли - это наборы прав доступа. Некоторые роли дают участнику простой доступ к чтению материала, в то время как другие позволяют производить изменения в работе сайта и инструментов, например, добавление участников, изменение материала или прав доступа для других ролей. Все роли не эквивалентны.

При создании сайта, или при получении готового, пользователь получает роль с наибольшими правами доступа и самый широкий уровень допуска. После этого можно выбрать (в рамках дозволенного администратором), какие инструменты или функции (например, Форум, Календарь, Ресурсы) должны присутствовать на сайте. Для каждого из этих инструментов можно настроить права доступа, чтобы не допустить совершения участниками действий, которые выходят за рамки назначенных ролей.

Основные возможности

Роли могут давать различные возможности для работы с разными инструментами. Тем не менее, в большинстве случаев, создатель сайта проекта или преподаватель на сайте курса имеет все права доступа и может добавлять или удалять материал с сайта.

Пользователи, которые добавлены на уже существующий сайт, имеют более ограниченные в правах доступа роли. Права доступа этих ролей позволяют взаимодействие с сайтом, например, отправку сообщений в чате, но не позволяют совершать некоторых других действий, например, загружать файлы с помощью инструмента "Ресурсы" или заполнять Расписание.

Тем не менее, настройки для каждого инструмента различны. Администратор сайта определяет, какие роли установить и какие права доступа им предоставить. Роли и связанные с ними права доступа будут различны на разных сайтах.

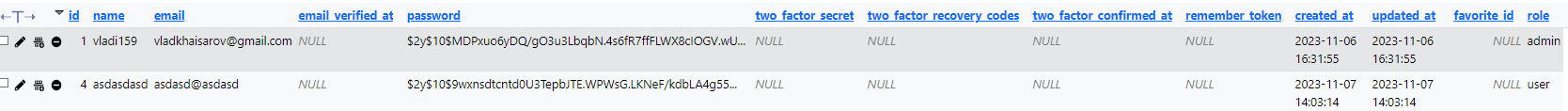
 Таблица пользователей и их ролей представлена выше (рисунок 15)

Рисунок 15 – Таблица пользователей

Форма авторизации представлена на рисунке 16

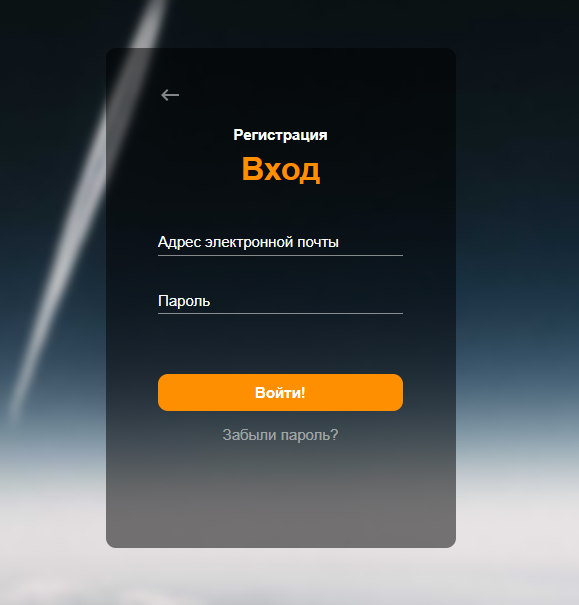


Рисунок 16 – Форма авторизации

Так же на сайте присутствует форма регистрации, которую можно наблюдать на рисунке 17.

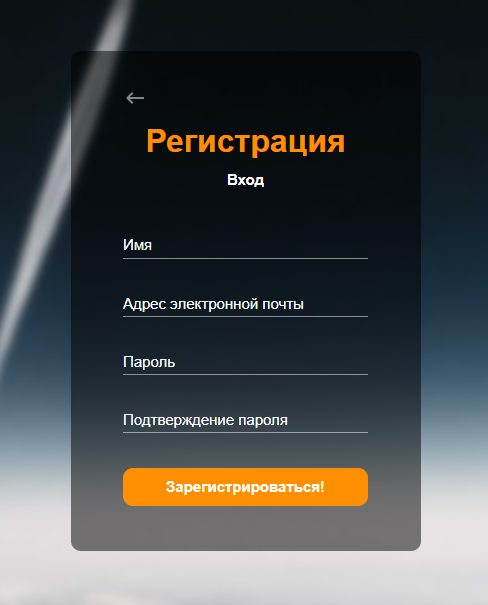


Рисунок 17 – Форма регистрации

# 2.5 Руководство пользователя

Инструкция по использованию программы (или просто «Инструкция пользователю», или «Руководство для пользователя») - это выдержка из полной документации, предназначенная для эксплуатации программы. Представляет собой независимый документ для пользователя программы, в котором описывается: что делает программа и как пользоваться.

«Инструкция пользователю» должна содержать всю необходимую для пользователя информацию и должна быть ему понятна без дополнительных материалов (без обращения к другим спецификациям). Следовательно, необходимая для этой инструкции информация переписывается полностью из соответствующих спецификаций.

Первая часть инструкции является описательной и должна содержать:

* наименование программы;
* краткое описание программы;
* перечень выполняемых программой функций;
* краткую характеристику метода (или методов) решения поставленной задачи, достоинства и недостатки;
* полную библиографическую ссылку на полное описание метода;
* описание входных и выходных данных.
* описание структуры базы данных (если она имеется), всех ее таблиц в словесной (вербальном) форме.

Вторая часть документа должна описывать порядок работы с программой. Должна содержать описание всех режимов работы программы, а также содержание всех печатей и диагностических сообщений, которые выдаются по ходу выполнения программы.

Следует помнить, что пользователь по своей квалификации не является программистом и поэтому работа с программой описывается на понятном ему языке и достаточно подробно, а именно:

* как запустить программу;
* как продолжить работу с программой (описывается подробный интерактивный режим работы пользователя с программой);
* подготовка и ввод исходных данных в программу;
* как реагировать на запросы программы;
* как вести работу в исключительных ситуациях;
* как реагировать на ошибки;
* как восстановить работу программы в случае аварийного его завершения;
* как получить требуемый результат;
* как правильно закончить работу с программой (запланированный программой выход);
* другие сведения, необходимые пользователю программы.

Руководство для обычного пользователя

При заходе на сайт у обычного пользователя имеются ряд ограничений. Например, незарегистрированный пользователь не может подать заявку на поступление в детский сад, а также оставлять отзывы на сайте. Он не может зайти в свой личный кабинет, так как его попросту нет. Для того чтобы зарегистрироваться или авторизоваться нужно пожать в меню навигации на соответствующую кнопку «Вход». Это можно увидеть на рисунке 18.

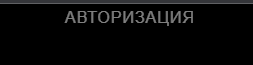


Рисунок 18 − Кнопка для входа в систему

Далее появляется возможность зайти в личный кабинет (рисунок 19) там можно посмотреть свои личные данные (рисунок 20) а также изменить при необходимости,

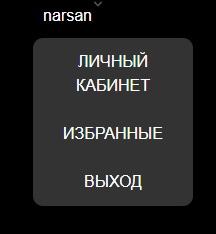


Рисунок 19 – Личный кабинет



Рисунок 20 – Личные данные

При переходе на интересующий товар выходит подробная информация об этом товаре (рисунок 21)



Рисунок 21 – Информация о товаре

А если пользователь зарегистрирован у него есть возможность добавить товар в избранное (рисунок 22).

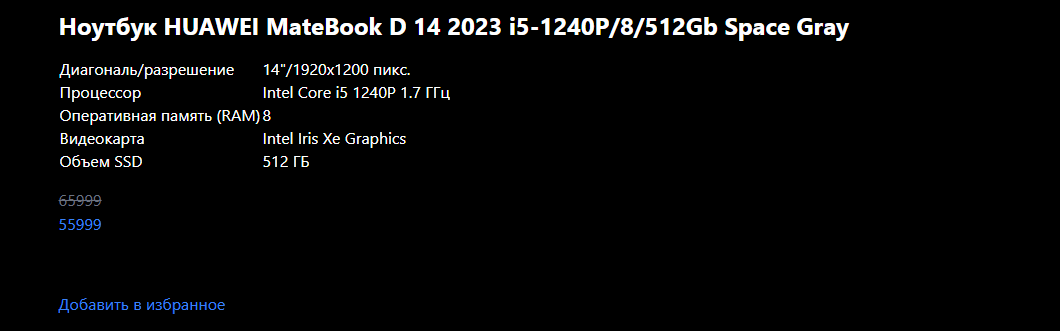


Рисунок 22 – Добавление товара в избранное

И далее свои избранные можно посмотреть в личном кабинете (рисунок 23)

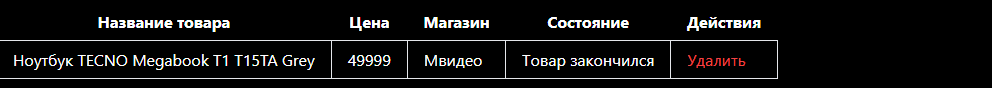


Рисунок 23 –Избранное

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта были получены и закреплены навыки программирования в среде PHP. В результате создано работоспособное приложение «Разработка автоматизированной информационной системы для сравнения цен на товары». Проведенная проверка работы программы не выявило существенных ошибок. Это, однако, не исключает возможности появления при проведении более глубокого и длительного тестирования. Веб-приложение создано для того чтобы люди могли просматривать и покупать выгодные товары.

В ходе работы над курсовым проектом были реализованные все задачи и цели проекта.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Лисьев, Г. А. Программное обеспечение компьютерных сетей и web-серверов [Электронный ресурс] : учебное пособие / Г. А. Лисьев, П. Ю. Романов, Ю. И. Аскерко. — Москва : ИНФРА-М, 2019. — 145 с. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/read?id=329984> .
2. Немцова, Т. И. Компьютерная графика и web-дизайн [Электронный ресурс] : учебное пособие  / Т. И. Немцова, Т. В. Казанкова, А. В. Шнякин ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 400 с. + Доп. материалы. -  Режим доступа: <https://znanium.com/read?id=329728>.
3. Цупин, В. А. Управление контентом. Практикум  [Электронный ресурс] : учебное пособие / В.А. Цупин, М.М. Ниматулаев. — Москва : ИНФРА-М, 2019. — 211 с. — (Высшее образование: Бакалавриат). — Режим доступа: <https://znanium.com/read?id=344856> .
4. Шаньгин, В. Ф. Комплексная защита информации в корпоративных системах  [Электронный ресурс] : учебное пособие / В.Ф. Шаньгин. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 592 с. — (Высшее образование: Бакалавриат). - Режим доступа: <https://znanium.com/read?id=330966>.
5. Винарский, Я. С. Web-аппликации в интернет-маркетинге: проектирование, создание и применение  [Электронный ресурс] : практическое пособие / Я. С. Винарский, Р. Д. Гутгарц. — Москва : ИНФРА-М, 2018. — 269 с. + Доп. материалы. — (Среднее профессиональное образование). - Режим доступа: <https://znanium.com/read?id=329175> .
6. Кроксен-Джон Д. Оптимизация интернет-магазина: Почему 95% посетителей вашего сайта ничего не покупают и как это исправить [Электронный ресурс] : Справочное пособие / Кроксен-Джон Д., Ван Тондер Й. - Москва :Альпина Паблишер, 2018. - 318 с.: ISBN 978-5-9614-7131-1 - Режим доступа: <https://znanium.com/read?id=333470> .
7. Мельник, Н. В. Искусство продвижения сайта. Полный курс SEO: от идеи до первых клиентов [Электронный ресурс] / Н. В. Мелькин. - Вологда:Инфра-Инженерия, 2017. - 268 с. – ISBN 978-5-9729-0139-5 - Режим доступа: <https://znanium.com/read?id=303061> .
8. Шамина, И. С. SEO-копирайтинг 2.0. Как писать тексты в эру семантического поиска [Электронный ресурс] / И. С. Шамина. - Вологда: Инфра-Инженерия, 2018. - 260 с.: ISBN 978-5-9729-0210-1 - Режим доступа: <https://znanium.com/read?id=326348> .
9. Официальная документация по Laravel – https://laravel.com/

ПРИЛОЖЕНИЕ А

Листинг 1 – Профиль controller

/\*Показывает форму редактирования профиля пользователя. \*/

class ProfileController extends Controller

{

    public function edit(Request $request): View

    {

        return view('profile.edit', [

            'user' => $request->user(),

        ]);

    }

/\*Обновляет профиль пользователя на основе входных данных из запроса\*/

    public function update(ProfileUpdateRequest $request): RedirectResponse

    {

        $request->user()->fill($request->validated());

        if ($request->user()->isDirty('email')) {

            $request->user()->email\_verified\_at = null;

        }

        $request->user()->save();

     return Redirect::route('profile.edit')->with('status', 'profile-updated');

    }

/\*Обновляет пароль пользователя на основе входных данных из запроса. \*/

    public function updatePassword(UpdatePasswordRequest $request): RedirectResponse

    {

        $user = $request->user();

        if (!Hash::check($request->current\_password, $user->password)) {

            return Redirect::back()->withErrors(['updatePassword' => 'The current password is incorrect.']);

        }

        $user->password = Hash::make($request->password);

        $user->save();

      return Redirect::route('profile.edit')->with('status', 'password-updated');

    }

/\* Удаляет профиль пользователя, выходит из системы и перенаправляет на главную страницу\*/

    public function destroy(Request $request): RedirectResponse

    {

        $request->validateWithBag('userDeletion', [

            'password' => ['required', 'current\_password'],

        ]);

        $user = $request->user();

        Auth::logout();

        $user->delete();

        $request->session()->invalidate();

        $request->session()->regenerateToken();

        return Redirect::to('/');

    }

}