

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е. Н. Скуратенко  
подпись  
«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.03 Прикладная информатика

Разработка интернет-магазина «Ole Lukoie magic studio»

Руководитель \_\_\_\_\_ доцент, к.ф.-м.н. А.Н. Таскин  
подпись, дата

Выпускник \_\_\_\_\_ А.Ю. Цыбуля  
подпись, дата

Консультанты  
по разделам:

Экономический \_\_\_\_\_ Е. Н. Скуратенко  
подпись, дата

Нормоконтролер \_\_\_\_\_ В. И. Кокова  
подпись, дата

Абакан 2021

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Е. Н. Скуратенко  
подпись  
«\_\_\_\_\_» \_\_\_\_\_ 2021 г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Цыбуля Анатолию Юрьевичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка интернет-магазина «Ole Lukoie magic studio»

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: А.Н. Таскин, доцент, к.ф.-м.н., ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ГАУ РХ «ЦИНТ Хакасии».

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки системы.
3. Расчёт затрат и оценка экономической эффективности реализации интернет-магазина «Ole Lukoie magic studio».

Перечень графического материала: нет

Руководитель ВКР

\_\_\_\_\_

подпись

А.Н. Таскин

Задание принял к исполнению

\_\_\_\_\_

подпись

А.Ю. Цыбуля

«08» апреля 2021 г.

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка интернет-магазина «Ole Lukoie magic studio»» содержит 100 страниц текстового документа, 79 рисунков, 10 таблиц, 15 использованных источников, 1 приложение.

СУБД MYSQL, ЗАПРОС, БАЗА ДАННЫХ, ТАБЛИЦЫ, НОРМАЛИЗАЦИЯ, ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ДАННЫЕ, ИНФОРМАЦИОННАЯ СИСТЕМА, ИНТЕРНЕТ-МАГАЗИН, ИНТЕРНЕТ-ВИТРИНА, ДИАГРАММА.

Цель данной выпускной квалификационной работы: разработать интернет-магазин «Ole Lukoie magic studio».

Чтобы достичь этой цели, необходимо выполнить следующие задачи:

- Проанализировать предметную область.
- Выбрать средства проектного решения.
- Разработать интернет-магазин «Ole Lukoie magic studio».
- Рассчитать экономическую эффективность реализации интернет-магазина «Ole Lukoie magic studio».

В результате работы была разработана информационная система «Интернет-магазин Ole Lukoie magic studio».

## SUMMARY

The theme of the graduation thesis is «Development of Online Store “Ole Lukoie Magic Studio” contains 100 pages of a text document, 79 figures, 10 tables, 15 reference items, 1 appendix.

DATA BASE MANAGEMENT SYSTEM (DBMS) of MYSQL, INQUIRY, DATABASE, TABLES, STANDARTIZATION, GRADUATION THESIS, SOFTWARE, DATA, IT SYSTEM, ONLINE STORE, ONLINE STOREFRONT, DIAGRAM.

The purpose of this graduation thesis: to develop online store “Ole Lukoie Magic Studio”.

Objectives to be determined:

- to analyze the subject area;
- to select means of the design solution;
- to develop online store “Ole Lukoie Magic Studio”;
- to calculate economic efficiency of the implementation of online store “Ole Lukoie Magic Studio”.

Results: the IT system of «Online Store “Ole Lukoie Magic Studio” » has been developed.

English language supervisor: \_\_\_\_\_

N.V. Chezybaeva

## СОДЕРЖАНИЕ

Введение.....	8
1 Аналитический раздел разработки интернет-магазина «Ole Lukoie magic studio» .....	9
1.1 Краткое описание деятельности предприятия «Ole Lukoie magic studio» .....	9
1.2 Предпосылки для разработки ПО. Определение потребностей и прикладных сфер, требующих автоматизации. Актуальность разработки.....	10
1.3 Постановка цели и задач .....	12
1.4 Бизнес-процессы предметной области .....	12
1.5. Принципы работы реализуемой информационной системы.....	15
1.6 Требования к системе .....	16
1.7 Анализ аналогичных программных продуктов .....	17
1.7.1 Планирование и определение функций .....	19
1.8 Обоснование и выбор средств проектных решений.....	22
1.8.1 Планирование инфраструктуры. Выбор оборудования.....	24
1.9 Выводы по аналитическому разделу разработки интернет-магазина «Ole Lukoie magic studio» .....	25
2 Проектный раздел разработки интернет-магазина «Ole Lukoie magic studio» .....	26
2.1 Разработка базы данных .....	26
2.2 Верстка страниц. ....	30
2.3 Написание серверной части системы.....	37
2.4 Написание скриптов .....	45

2.5 Создание админ-панели.....	48
2.6 Подключение системы оплаты .....	66
2.7 Наполнение магазина .....	76
2.8 Тестирование .....	77
2.9 Вывод по разделу «Проектный раздел разработки интернет-магазина «Ole Lukoie magic studio».....	77
3 Экономический раздел разработки интернет-магазина «Ole Lukoie magic studio» .....	78
3.1 Расчет капитальных затрат .....	78
3.2 Эксплуатационные затраты.....	83
3.3 Мероприятия по снижению вероятности рисков, имеющих наивысшую оценку .....	87
3.4 Определение доходов и экономической эффективности реализации проекта разработка онлайн-магазина «Ole Lukoie magic studio».....	89
3.5 Вывод по разделу «Экономический раздел разработки интернет-магазина «Ole Lukoie magic studio».....	94
Заключение .....	96
Список использованных источников .....	97
ПРИЛОЖЕНИЕ А.....	99

## ВВЕДЕНИЕ

Информационная система – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации для достижения цели управления. В современных условиях основным техническим средством обработки информации является персональный компьютер. Большинство современных информационных систем преобразуют не информацию, а данные. Поэтому часто их называют системами обработки данных.

Написание выпускной квалификационной работы несет следующую цель: разработать интернет-магазин «Ole Lukoie magic studio».

В первом разделе исследуется предметная область, дается краткое описание деятельности предприятия «Ole Lukoie magic studio», исследуются предпосылки для разработки ПО, анализируется существующий бизнес процесс, изучается предметная область проекта, выбирается методология разработки, схема жизненного цикла, планируются функции проектируемой системы, проектируется база данных, выбираются программные средства для разработки, планируется инфраструктура и строится модель взаимодействия пользователя с информационной системой

Во втором разделе происходит разработка базы данных, верстка страниц магазина, написание серверной части системы, написание скриптов, создание админ- панели, подключение системы оплаты, наполнение магазина и тестирование системы.

В третьем разделе определяется экономическая эффективность проекта. Определяется его стоимость, сравнивается со стоимостью разработки у компаний аналогов, и в конце дается оценка, является ли разработка системы экономически эффективной.



## **1 Аналитический раздел разработки интернет-магазина «Ole Lukoie magic studio»**

### **1.1 Краткое описание деятельности предприятия «Ole Lukoie magic studio»**

Государственное автономное учреждение Республики Хакасия "Центр информатизации и новых технологий Республики Хакасия" создано в 2014 г. Учредителем является Правительство Республики Хакасия, частично функции и полномочия учредителя осуществляют Управление по обеспечению деятельности Правительства Республики Хакасия, частично Министерство имущественных и земельных отношений Республики Хакасия .[15]

ГАУ РХ «ЦИНТ Хакасии» основано с целью создания современной, высокотехнологичной, работоспособной и надежной системы в Республике, которая обеспечит максимально эффективное информационное взаимодействие между органами государственной власти Республики, органами местного самоуправления, а также общественными институтами и гражданами.

На данный момент ГАУ РХ «ЦИНТ Хакасии» осуществляет техническую поддержку и сервисное обслуживание инфраструктуры региональных органов исполнительной власти, подведомственных учреждений, а также коммерческих организаций.

ГАУ РХ «ЦИНТ Хакасии» в том числе, занимается поддержкой частного бизнеса. И одним из таких бизнесов является студия по производству и продаже сувенирных изделий – Ole Lukoie magic studio. Дальше речь пойдет именно о ней.

Ole Lukoie magic studio – маленькая студия, которая занимается производством сувенирных изделий для украшения помещений, еды и иного использования, и продажей их за рубеж. Ориентированность студии –

англоязычный пользователь, поэтому все цены указаны в долларах, а интерфейс написан на английском языке. Так как студия маленькая, объемы производства небольшие, изделия продаются поштучно, или совсем небольшим оптом.

## 1.2 Предпосылки для разработки ПО. Определение потребностей и прикладных сфер, требующих автоматизации. Актуальность разработки

Основной предпосылкой разработки информационной системы является существование технически слабой системы, которая используется в данный момент. Эта система представляет из себя интернет-витрину, где отображается информация о товаре, но покупка товара может осуществляться только через администрацию сайта, что крайне неудобно, и занимает куда больше времени, чем у интернет-магазина.

Интернет-витрина студии представлена на рисунке 1.1.

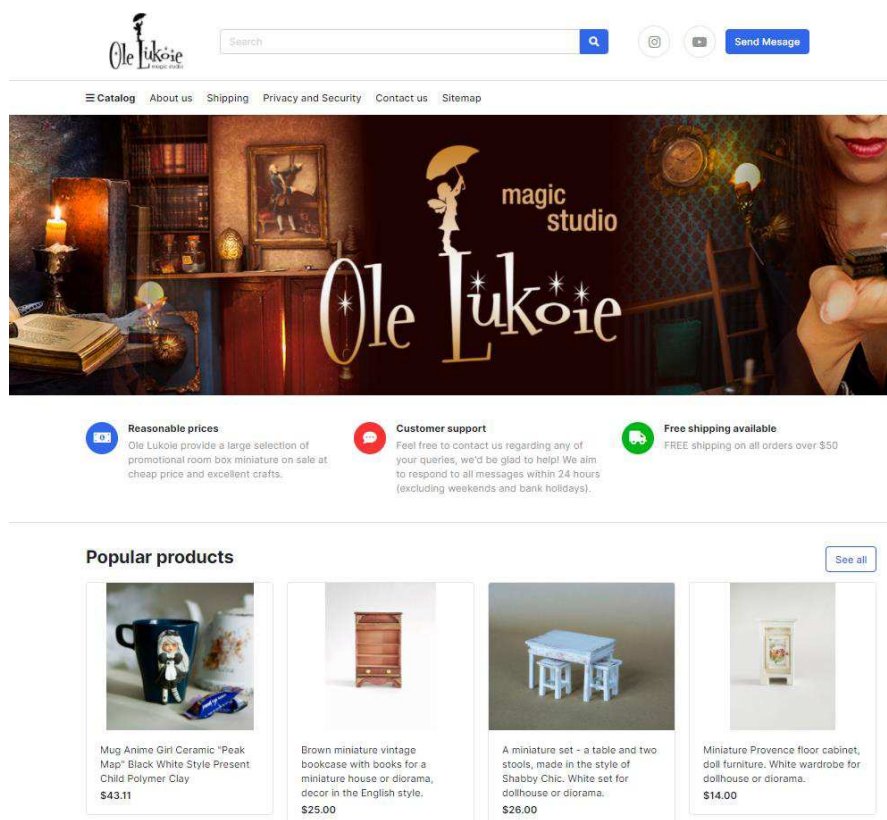


Рисунок 1.1 – Интернет-витрина студии

Карточка товара представлена на рисунке 1.2.

**A miniature set - a table and two stools, made in the style of Shabby Chic. White set for dollhouse or diorama.**



Price: \$ 26.00

Order

Category :

Miniature furniture

Shipping

Privacy and Security

Materials

wood, cardboard, varnish, acrylic

Description

The product is made of cardboard, which was glued and made more durable with a primer. Varnished.

This set fits perfectly into the English or classic interior of your miniature home.

This product is not for children to play.

Miniature for a doll house. This Art Deco set is designed to complement your little dollhouse with a miniature piece of furniture.

Dimensions: table (length 10 cm, height 6.5 cm, depth 7 cm), stool (length 3.5 cm, height 4.5 cm)

Рисунок 1.2 – Карточка товара

Главной целью проекта является упрощение работы заказа товара, а также более расширенный функционал сайта.

Сейчас пользователь, заходящий на сайт видит только текст, картинку и товары с ценником, и единственный вариант заказа товара – это прямая связь с администрацией сайта, что крайне неудобно ни для пользователя, ни для администрации.

Разрабатываемая система будет использоваться заказчиком, а именно студией по производству и продажи сувенирных изделий – Ole Lukoie magic studio.

Главная сфера, требующая автоматизации – заказ товара. Проект позволит самому заказчику осуществлять заказ, без участия администрации сайта. Также автоматизации подвергнется административная панель сайта – будет создан удобный инструмент добавления новых категорий и товаров, вследствие чего, администратор будет тратить меньше времени на добавление, изменение и удаление товаров и категорий.

Тему можно считать актуальной, так как любой современный магазин выиграет от существования интернет-версии. Она возьмет на себя задачи по маркетингу товаров, самого магазина, а также продажи товаров.

### **1.3 Постановка цели и задач**

Цель данной выпускной квалификационной работы: разработать интернет-магазин «Ole Lukoie magic studio».

Для создания интернет-магазина, необходимо выполнить следующие задачи:

- Проанализировать предметную область.
- Выбрать средства проектного решения.
- Разработать интернет-магазин «Ole Lukoie magic studio».
- Рассчитать экономическую эффективность реализации интернет-магазина «Ole Lukoie magic studio».

### **1.4 Бизнес-процессы предметной области**

Для того чтобы понять, как правильно автоматизировать процессы системы, необходимо проанализировать бизнес-процессы.

На рисунке 1.3 показана диаграмма IDEF0, на которой описан процесс заказа товара до создания интернет-магазина.

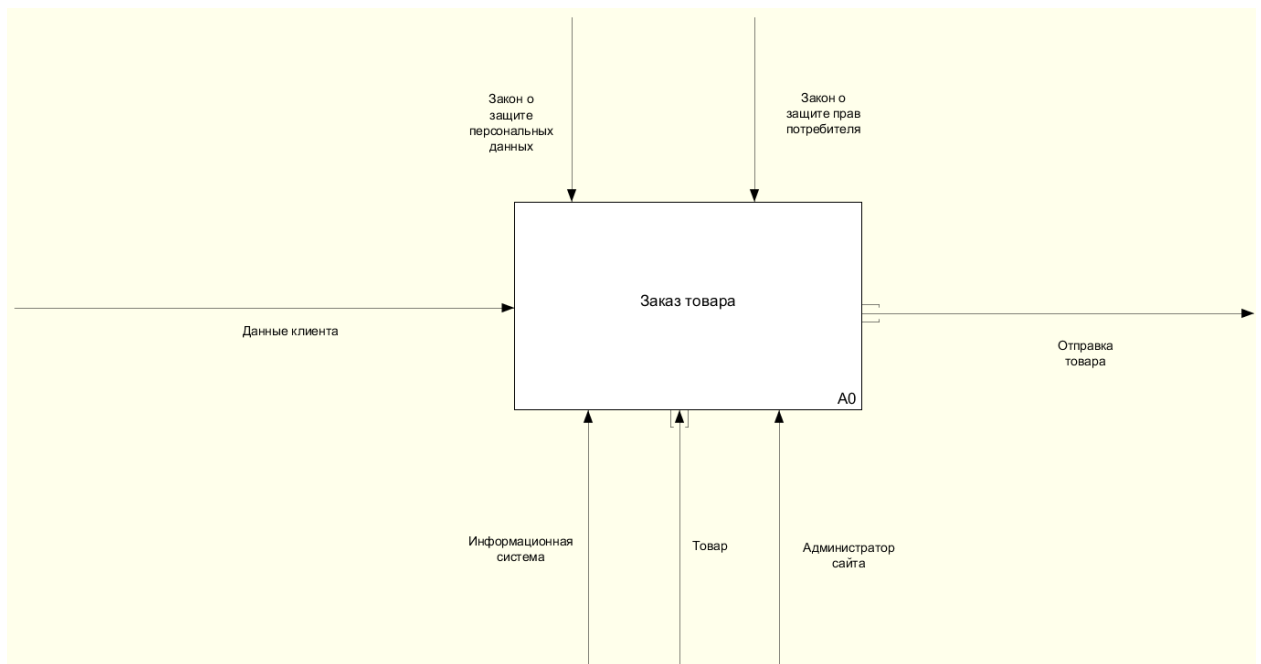


Рисунок 1.3 – Диаграмма IDEF0 процесса заказа товара

Входными данными являются данные клиента.

Механизмом процесса является информационная система, в которой происходит процесс, товар и администратор, который работает с этой системой.

Управляется процесс при помощи двух основных законов: «О защите персональных данных», «О защите прав потребителя».

В результате работы выходной информацией является отправка товара клиенту.

Декомпозиция диаграммы IDEF0 процесса заказа товара показана на рисунке 1.4.

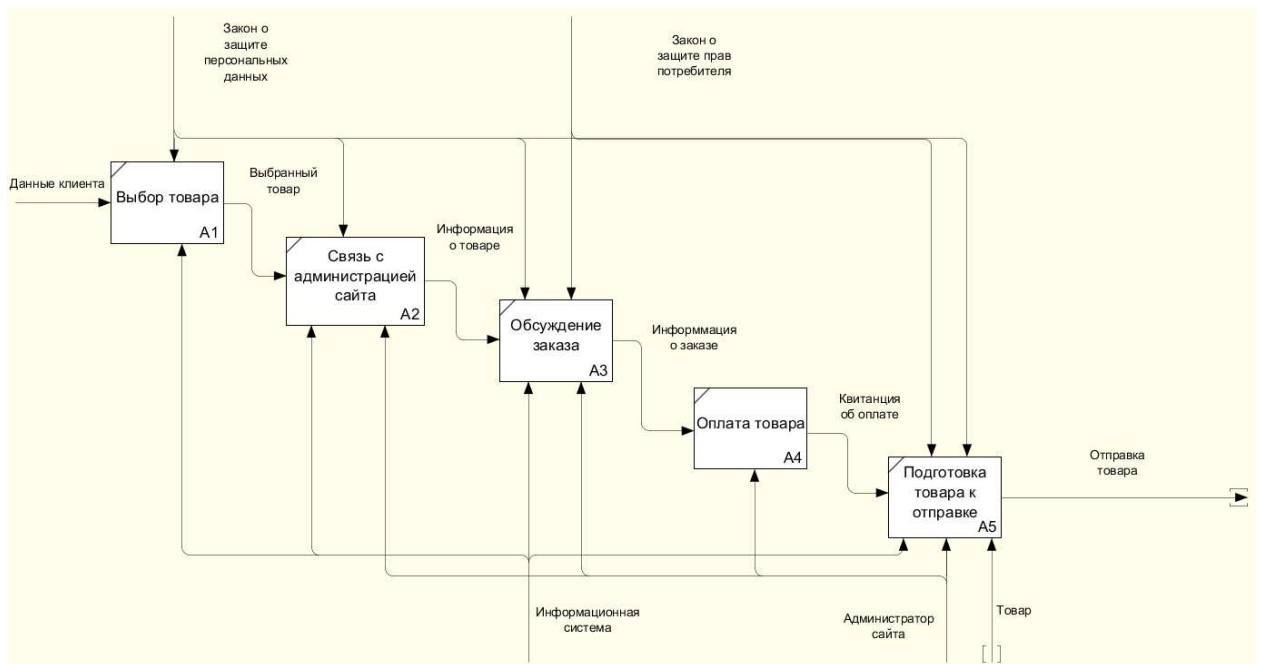


Рисунок 1.4 – Декомпозиция диаграммы IDEF0 процесса заказа товара

Процесс может быть декомпозирован на 5 основных блоков:

1. Выбор товара.
2. Связь с администрацией сайта.
3. Обсуждение заказа.
4. Оплата товара.
5. Подготовка товара к отправке.

Вывод: исходя из декомпозиции процесса заказа товара, можно сказать, что на связь, обсуждение заказа и оплату товара уходит очень много времени, и порой может растянуться на несколько дней, тогда как в разрабатываемой системе все эти процессы будут заменены несколькими полями на сайте, на заполнение которых уйдет пара минут. Единственный процесс, который автоматизировать на данном этапе не получится – это отправка товара, ее придется делать вручную.

## 1.5. Принципы работы реализуемой информационной системы

Интернет-магазин, в отличие от интернет-витрины обладает следующими функциями:

- предоставление онлайн-помощи покупателю;
- предоставление интерфейса к БД продаваемых товаров (в виде каталога, прайс-листа);
- работа с электронной корзиной покупателя;
- оформление заказов с выбором метода оплаты, доставки, страховки и выпиской счёта;
- проведение расчётов (при выборе электронных методов оплаты) или контроль факта оплаты (при использовании традиционных форм расчётов);
- предоставление покупателю средств отслеживания исполнения заказов;
- доставка товаров;
- сбор и анализ различной маркетинговой информации;
- обеспечение безопасности личной информации покупателей.

Разрабатываемая система позволит пользователю осуществлять покупки и заказ товара в несколько кликов, без связи с администрацией. В системе будет весь необходимый для современного интернет-магазина функционал, позволяющий быстро регистрироваться, осуществлять заказы, просматривать товары, связываться с администрацией для выяснения интересующих вопросов. В интернет-магазине при обслуживании клиента роль менеджера уже не является необходимой, т.к. заказы обрабатываются автоматически. Система будет работать на всех современных медиа-устройствах, поддерживающих интернет-браузер, а в частности компьютеры, телефоны, планшеты.

## 1.6 Требования к системе

Для понимания облика будущей системы первоначальным этапом является сбор и систематизации требований к программному продукту. Эти требования представлены в таблице 1.

Таблица 1 – Требования к информационной системе

№ п/п	Требование	Группа требований по ГОСТ	Приоритет требований
1	Десктопная версия	требования к системе в целом	Необходимо
2	Мобильная версия	требования к системе в целом	Желательно
3	Наличие многоязычного интерфейса	требования к функциям (задачам), выполняемым системой	Желательно
4	Возможность администрирования контента	требования к функциям (задачам), выполняемым системой	Необходимо
5	Система онлайн оплаты	требования к системе в целом	Необходимо
6	Наличие регистрации и авторизации пользователей	требования к системе в целом	Желательно
7	Предоставление различного набора прав доступа в зависимости от типа учетной записи	требования к системе в целом	Необходимо
8	Поддержка десктопной версии системы Windows 7,8,10	требования к видам обеспечения.	Необходимо
9	Наличие у программиста возможности внесения правок в рабочий код	требования к системе в целом	Необходимо



## Окончание таблицы 1

10	Наличие корзины для добавления товаров	требования к функциям (задачам), выполняемым системой	Необходимо
11	Наличие обратной связи с администрацией	требования к системе в целом	Необходимо
12	Наличие информации о сервисе и его работы	требования к функциям (задачам), выполняемым системой	Необходимо
13	Наличие ссылок на социальные сети	требования к системе в целом	Необходимо
14	Наличие поиска по сайту	требования к системе в целом	Необходимо
15	Наличие каталога товаров	требования к системе в целом	Необходимо
16	Наличие карточек товаров с информацией о них	требования к системе в целом	Необходимо

### 1.7 Анализ аналогичных программных продуктов

Областью анализа разрабатываемой информационной системы является интернет-магазин, специализирующийся на продаже сувенирных изделий.

Для точного анализа необходимо взять в пример конкурентов. Первый из них – интернет-магазин «Другие подарки» [4] представлен на рисунке 1.5.

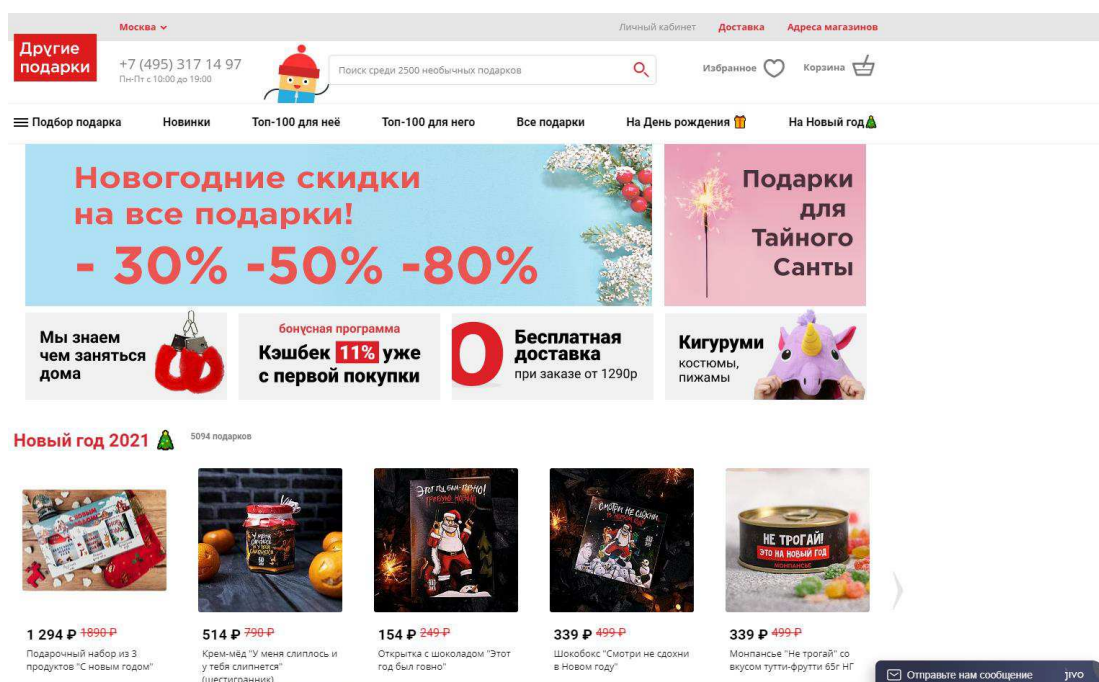


Рисунок 1.5 – Интернет-магазин «Другие подарки»

У рассматриваемого магазина стоит выделить наличие корзины, поиска по сайту, каталога товаров, а также различных вкладок с товарами разного назначения. Также следует отметить, что у данного магазина существуют физические магазины, и сайт позволяет узнать их адрес, чтобы посетить их в «живую».

Второй пример, это сервис «Россувенир» [5], представлен на рисунке 1.6.

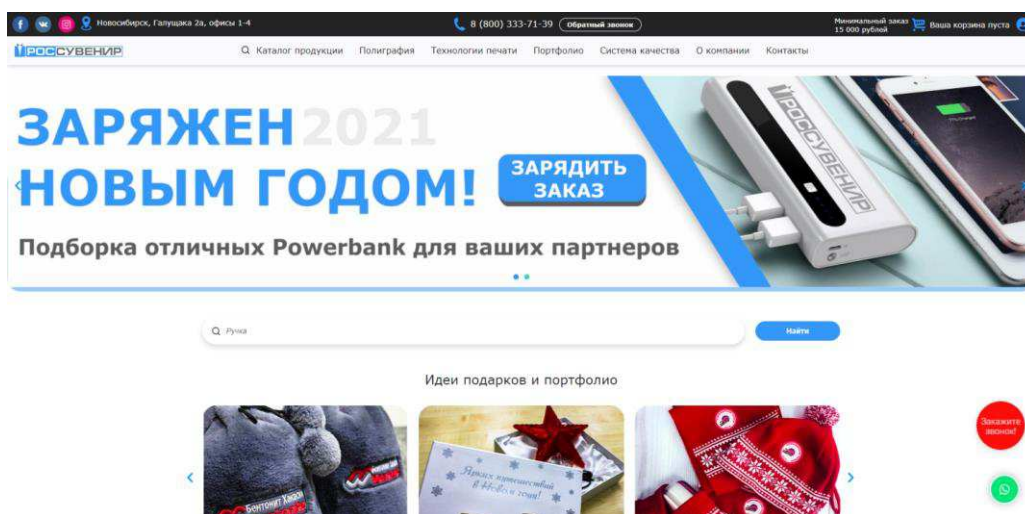


Рисунок 1.6 – Интернет-магазин «Россувенир»

У сервиса есть все необходимые для работы интернет-магазина инструменты, но также данный магазин отличается от предыдущего одной крайне интересной деталью. Как можно заметить в правом верхнем углу – стоимость минимального заказа фиксированная, ниже которой заказать товар не получится.

Подводя итог, можно выделить основные необходимые атрибуты для современного интернет-магазина. Это обязательно корзина, каталог товаров, и обратная связь. А также наличие симпатичного, но в то же время интуитивного и лаконичного дизайна.

### 1.7.1 Планирование и определение функций

Для дальнейшего планирования разработки информационной системы, необходимо построить основные модели работы программы.

Диаграмма use case – диаграмма, отражающая отношения между акторами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Диаграмма use case показана на рисунке 1.8.

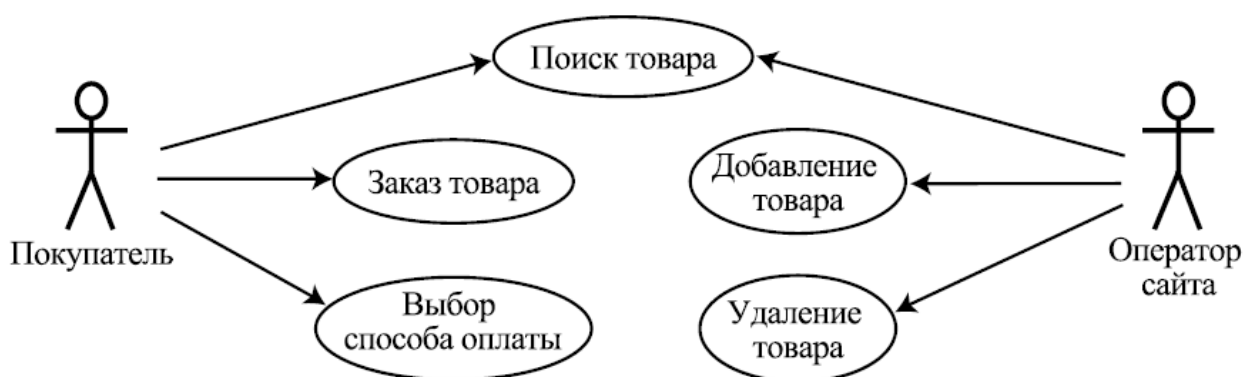


Рисунок 1.8 – Диаграмма use case

Далее следует декомпозиция диаграммы IDEF0 после автоматизации процесса. Диаграмма показана на рисунке 1.9.

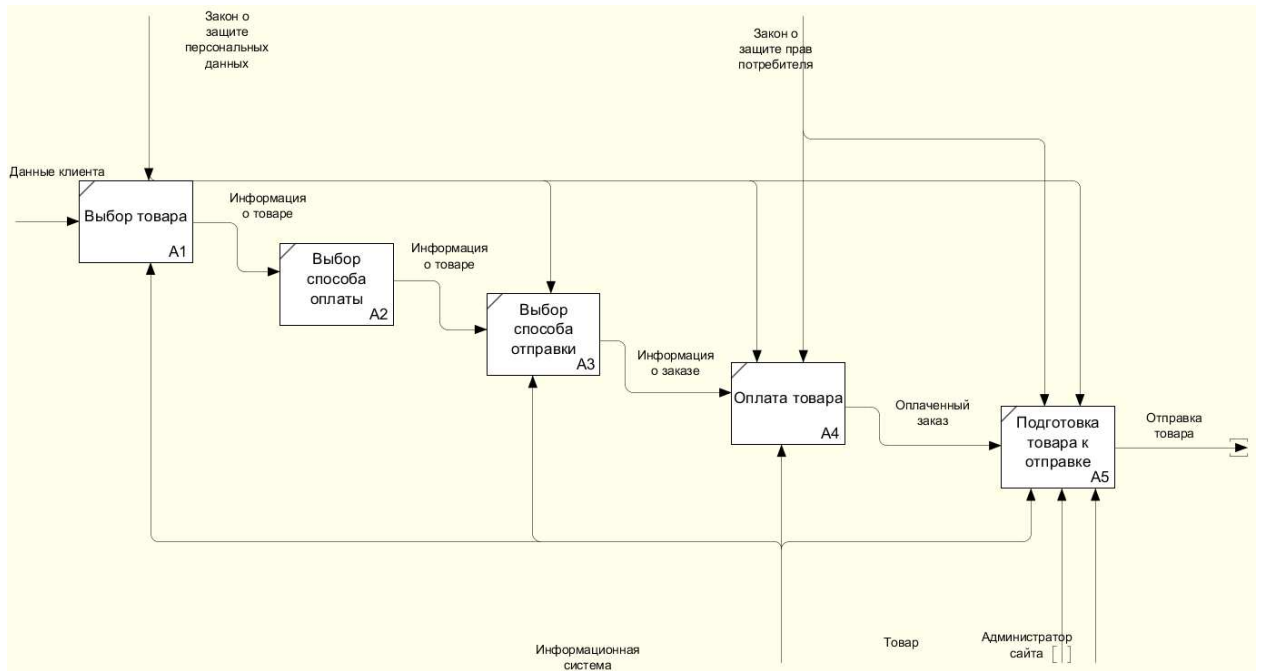


Рисунок 1.9 – Декомпозиция диаграммы IDEF0

Отличие процессов до и после автоматизации заключается в отсутствии необходимости связываться с администрацией и обсуждать заказ, вместо этого клиент сам выбирает способ оплаты и способ доставки, после чего оплачивает товар.

Входы и выходы остались неизменными.

Далее следует диаграмма DFD, показывающая работу системы изнутри, представлена на рисунке 1.10.

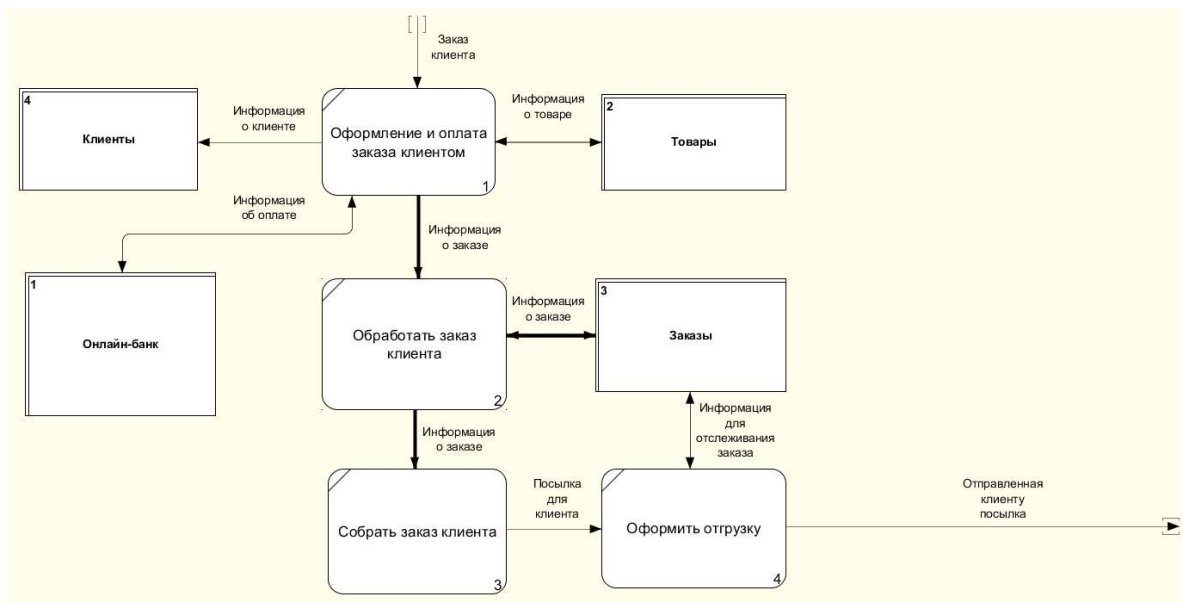


Рисунок 1.10 – Диаграмма DFD

На диаграмме можно видеть процесс заказа клиента. На диаграмме изображены 4 процесса:

1. Оформление и оплата заказа клиентом.
2. Обработать заказ клиента.
3. Собрать заказ клиента.
4. Оформить отгрузку.

1 внешняя сущность:

1. Онлайн банк.

И 3 хранилища данных:

1. Клиенты.
2. Заказы.
3. Товары

## 1.8 Обоснование и выбор средств проектных решений

Следующим этапом будет выбор технологий и программных средств разработки. При анализе уже существующих интернет-магазинов, изучение их структуры и поиске решений, складывается следующий инструментарий:

1. HTML – язык гипертекстовой разметки, служит для написания frontend интернет-магазина, без него обойтись нельзя.
2. CSS – по стандарту идет вместе с HTML, служит для визуального оформления страницы.
3. XAMPP - кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP. Программа, которая разворачивает веб-сервер на локальном компьютере. Существует пара альтернатив, например, USBWebserver, тут выбор зависит только от личного предпочтения, инструменты везде одинаковы.
4. Javascript – мультипарадигменный язык программирования. Служит для написания скриптов для frontend. Отвечает за визуальную составляющую сайта. Никакой альтернативы нет.
5. MySQL – свободная реляционная система управления базами данных. Стандартная система для работы с БД. Старая, простая, удобная. Альтернатив подобного масштаба не было выявлено.
6. Apache – свободный веб-сервер. Служит для работы с сайтом на локальном компьютере.
7. Любой интернет-браузер. Служит для работы со страницей интернет-магазина. Просмотр верстки, функционала и прямого взаимодействия с системой.
8. Любой текстовый редактор. Служит для написания кода.
9. PHP – язык программирования, служит для написания backend. Может быть заменен другими языками программирования, к

примеру, Python или Javascript, ниже приведено сравнение этих двух языков.

#### Особенности Python:

- Легко изучать, читать и поддерживать.
- Может работать на различных аппаратных платформах и использовать один и тот же интерфейс.
- Можно включить низкоуровневые модули в интерпретатор Python.
- Python предлагает идеальную структуру и поддержку для больших программ.
- Python предлагает поддержку автоматической сборки мусора.
- Поддерживает интерактивный режим тестирования и отладки.
- Предлагает высокоуровневые динамические типы данных, а также поддерживает динамическую проверку типов.
- Язык Python может быть интегрирован с программным кодом Java, C и C ++.

#### Особенности Javascript:

- Динамическая типизация – тип данных определяется в момент присваивания значения константе или переменной.
- Интерпретируемый язык – код приложения интерпретируется при обращении, не требуется предварительная компиляция.
- Функции как объекты первого класса, то есть функции в JavaScript можно возвращать из функций, передавать в качестве параметров в другие функции, присваивать переменным.
- Поддержка прототипного и объектно-ориентированного подхода.
- Универсальность – все популярные браузеры поддерживают JavaScript.

#### Особенности PHP:

- Интеграция баз данных. Поддерживает множество баз данных, таких как Oracle, MySQL и т. д.
- Простой в использовании.
- Он быстрее, чем другие языки скриптов, например, ASP и JSP.
- Открытый исходный код означает, что не нужно платить за использование PHP; можно свободно скачать и использовать его.
- PHP имеет некоторые predefined константы сообщений об ошибках, чтобы генерировать уведомления об ошибке или предупреждения.
- PHP предлагает регистрацию доступа, создавая сводку последних обращений пользователя.

По итогу можно сказать, что PHP проще и удобней в написании подобных проектов, и работа с БД у этого языка самая лучшая из всех языков для backend.

### **1.8.1 Планирование инфраструктуры. Выбор оборудования**

Далее следует выбор оборудования и планирование инфраструктуры.

В случае с интернет-магазином всё довольно просто, достаточно прибегнуть к помощи интернет-хостинга. Разворачивать интернет-магазин на какой-то локальной машине не имеет никакого смысла, так как современный интернет-хостинг очень дешевый, надежный и мощный.

Если разворачивать магазин на локальном компьютере, для этого нужен стабильный, быстрый интернет, компьютер или же сервер, и постоянный надзор за этой системой, так как интернет-магазин будет падать каждый раз, когда будет выключаться электроэнергия.

В то время как хостинг обеспечит постоянный доступ к сайту, в случае с неполадками с оборудованием перебросит на другой доступный сервер и



предложит огромный набор инструментов для работы со старыми версиями сайта, бэкапами и многим другим.

Современные хостинги со старта имеют все нужные для интернет-магазина инструменты: все нужные языки, программа для работы с базой данных, даже встроенный текстовый редактор.

Возможно на очень дальних дистанциях свой сервер себя окупит и принесет прибыль, но также не стоит забывать, что при росте интернет ресурса для него нужно все более мощное железо, и держать свой сервер где-то дома или в сети офиса не совсем правильно и даже небезопасно, с точки зрения информационной безопасности.

### **1.9 Выводы по аналитическому разделу разработки интернет-магазина «Ole Lukoie magic studio»**

В аналитическом разделе было описано предприятие «Ole Lukoie magic studio», для которого проектируется система и его деятельность. Далее были определены предпосылки для разработки системы, была изучена существующая и требующая замены система, была определена сфера автоматизации и актуальность проекта. Следующим этапом был проанализирован существующий бизнес-процесс, была построена его модель IDEF0, и сделан вывод о неэффективности существующего процесса и возможности его автоматизации. Далее были рассмотрены принципы реализуемой информационной системы, определены функции интернет-магазина, выгодно отличающие его от интернет-витрины. После этого был проведен анализ предметной области проектируемой информационной системы, где были определены основные атрибуты, необходимые для существования интернет-магазина. Далее были представлены требования к системе, поставлены цели и задачи.

Далее были выбраны методология разработки, и схема жизненного цикла, также при планировании и определении функций были составлены диаграммы данных, важные для понимания работы системы. Далее была спланирована база данных, и составлена ER-диаграмма. Следующим действием был выбор технологий и программных средств для разработки. Был изложен набор программных средств, с объяснением почему был выбран конкретный инструмент, а не аналогичный конкурент. Далее была спланирована инфраструктура и выбрано оборудование для работы системы, и в конце была построена диаграмма IDEF3, на которой изображена модель работы пользователя с информационной системой.

Подводя итог, можно сказать, что раздел получился информативным и полностью отвечающим на поставленные цели и задачи, и можно переходить к началу этапа проектирования.

## **2 Проектный раздел разработки интернет-магазина «Ole Lukoie magic studio»**

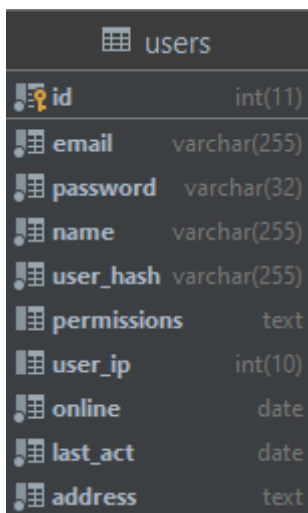
### **2.1 Разработка базы данных**

Для того чтобы понять, как пользователь взаимодействует с системой, необходимо построить модель IDEF3.

IDEF3 (англ. Integrated DEFINition for Process Description Capture Method) – методология моделирования и стандарт документирования процессов, происходящих в системе. Метод документирования технологических процессов представляет собой механизм документирования и сбора информации о процессах. IDEF3 показывает причинно-следственные связи между ситуациями и событиями в понятной эксперту форме, используя структурный метод выражения знаний о том, как функционирует система, процесс или предприятие.

Построенная модель показана в приложении А, на рисунке А.1.

Создание базы данных начинается с таблицы users. Она необходима для хранения данных о пользователях и администраторов сервиса. Структура таблицы представлена на рисунке 2.1.



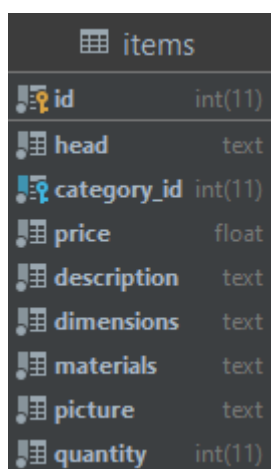
users	
id	int(11)
email	varchar(255)
password	varchar(32)
name	varchar(255)
user_hash	varchar(255)
permissions	text
user_ip	int(10)
online	date
last_act	date
address	text

Рисунок 2.1 – Структура таблицы users

Описание структуры таблицы users:

- id – автозаполняемое поле с id пользователя;
- email – email пользователя;
- password – пароль пользователя;
- name – имя пользователя;
- user\_hash – хэш пользователя, необходим для работы с сессиями;
- permissions – права пользователя, необходимые для работы админ-панели
- user-ip – ip-адрес пользователя;
- online – поле, показывающее авторизирован ли пользователь и жива ли его сессия;
- last\_act – дата последнего входа пользователя;
- address – адрес пользователя.

Далее идет таблица items. Таблица служит для хранения информации о товарах. Структура таблицы представлена на рисунке 2.2.



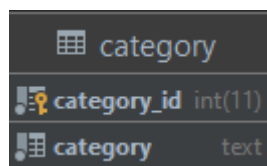
items	
id	int(11)
head	text
category_id	int(11)
price	float
description	text
dimensions	text
materials	text
picture	text
quantity	int(11)

Рисунок 2.2 – Структура таблицы items

Описание структуры таблицы items:

- id – автозаполняемое поле с id товара;
- head – заголовок товара;
- category\_id – id категории, поле нужно для связи с таблицей «Категории»;
- price – цена товара;
- description – описание товара;
- dimensions – размеры товара;
- materials – материалы товара;
- picture – путь до картинки, которая лежит в папке с сайтом;
- quantity – количество товара.

Далее идет таблица category. Это таблица – справочник, для обозначения категории товара. Структура таблицы представлена на рисунке 2.3.



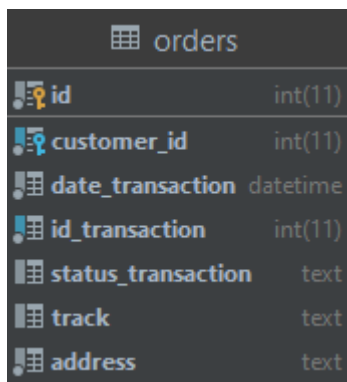
category	
category_id	int(11)
category	text

Рисунок 2.3 – Структура таблицы category

Описание структуры таблицы category:

- category\_id – поле, необходимое для связи с таблицей «items»;
- category – категория товара.

Последняя таблица orders. Эта таблица хранит данные о заказах. Структура таблицы показана на рисунке 2.4



orders	
id	int(11)
customer_id	int(11)
date_transaction	datetime
id_transaction	int(11)
status_transaction	text
track	text
address	text

Рисунок 2.4 – Структура таблицы orders

Описание структуры таблицы orders:

- id – автозаполняемое поле с id заказа;
- customer\_id – поле для связи с пользователем;
- date\_transaction – дата создания заказа;
- id\_transactions – id, для API QIWI, для получения деталей заказа;
- status\_transactions – статус заказа;
- track – track-номер заказа;
- address – адрес заказа.

Общая структура базы данных изображена на рисунке 2.5.

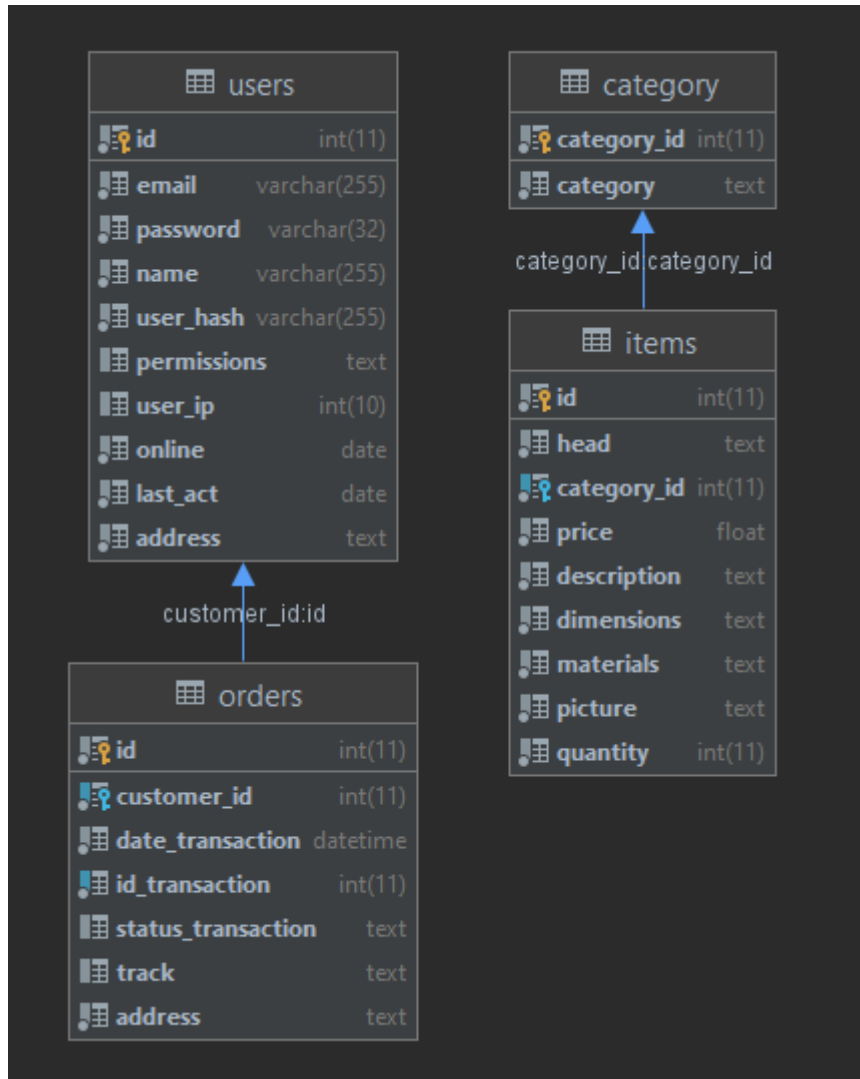


Рисунок 2.5 – Общая структура базы данных

На этом разработка БД завершена.

## 2.2 Верстка страниц

Главная страница изображена на рисунках 2.6 и 2.7.



- Reasonable prices**  
 Ole Lukoie provide a large selection of promotional room box miniature on sale at cheap price and excellent crafts.
- Customer support**  
 Feel free to contact us regarding any of your queries, we'd be glad to help! We aim to respond to all messages within 24 hours (excluding weekends and bank holidays).
- Free shipping available**  
 FREE shipping on all orders over \$50

### Popular products





[See all](#)



Рисунок 2.6 – Верхняя часть главной страницы

### Popular products

[See all](#)

 <p>Wardrobe Miniature Vintage Style Doll Furniture English Dollhouse Interior New</p> <p>Price: 49.36\$ <a href="#">Add to cart</a></p>	 <p>Magic Nightlight Roombox Steampunk Diorama Unique Decorate Miniature Interior</p> <p>Price: 522.49\$ <a href="#">Add to cart</a></p>	 <p>unique decorative night lights for home decor in the bedroom and dining room, Diorama</p> <p>Price: 220\$ <a href="#">Add to cart</a></p>	 <p>Decorative night light based on the fairy tale "The Chronicles of Narnia: The Lion, the Witch and the Wardrobe". Diorama, miniature lamp</p> <p>Price: 47\$ <a href="#">Add to cart</a></p>
---	---	---	--

	<b>Customers info</b> <a href="#">About us</a> <a href="#">Shipping</a> <a href="#">Privacy and Security</a> <a href="#">Contact us</a>	<b>Categories</b> <a href="#">Roombox</a> <a href="#">Miniature furniture</a> <a href="#">Decor</a>	<b>Social</b> <a href="#">vk.com</a> <a href="#">Instagram</a> <a href="#">Youtube</a>
---	---	--	---

Рисунок 2.7 – Нижняя часть главной страницы

Страница с информацией о компании представлена на рисунке 2.8.

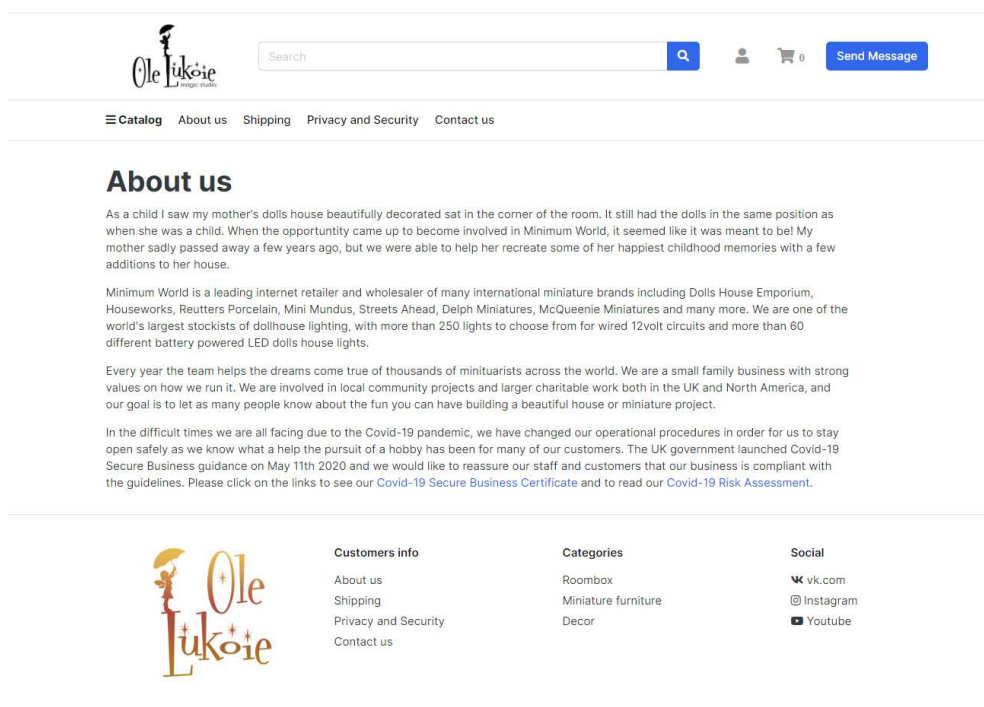


Рисунок 2.8 – Страница с информацией о компании

Страница с каталогами представлена на рисунке 2.9.

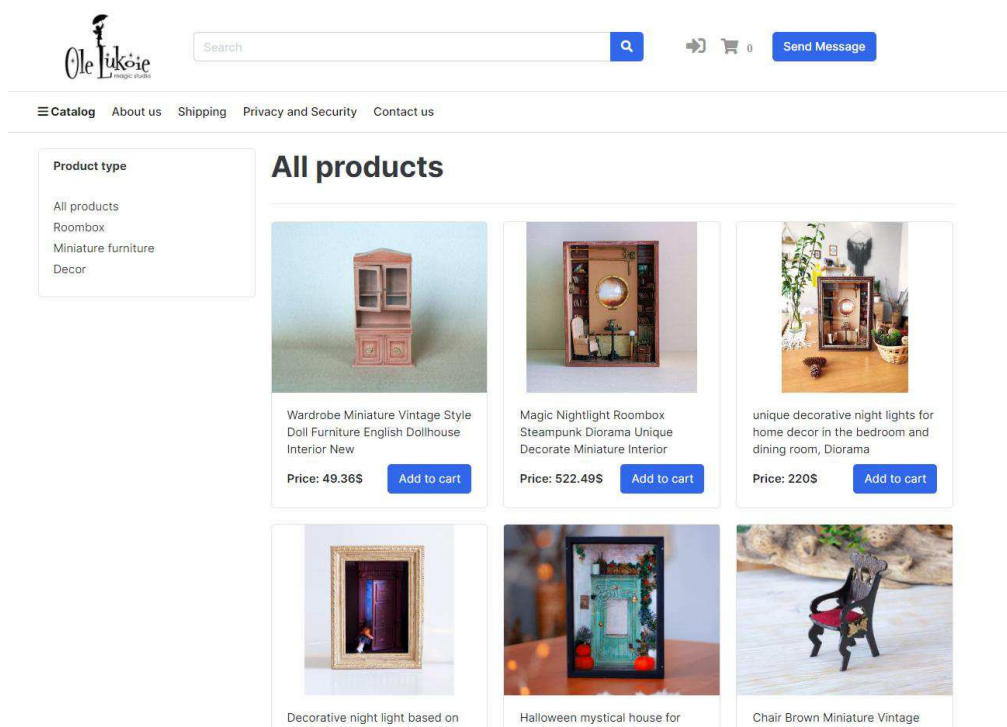


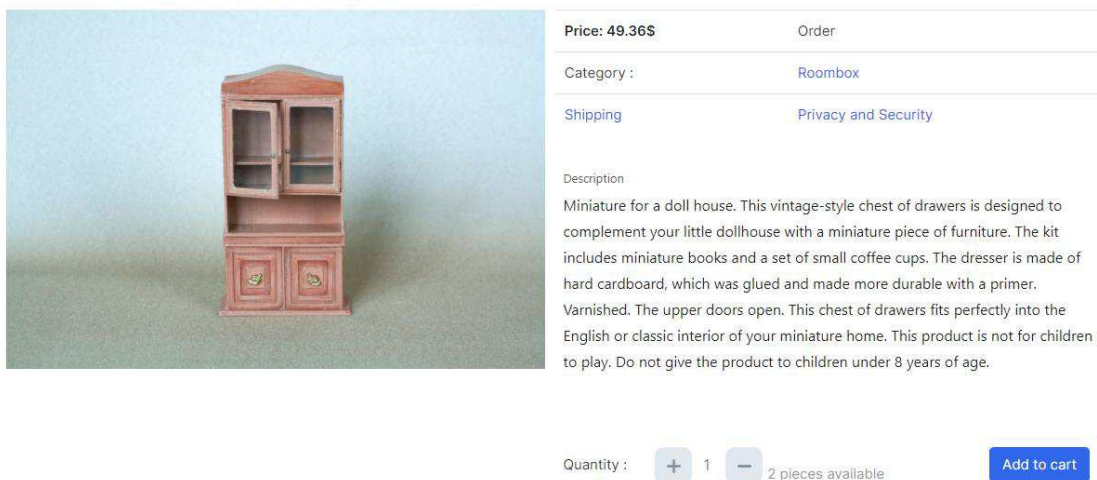
Рисунок 2.9 – Страница с каталогами



Точно также оформлены страницы «Shipping», «Privacy and Security» и «Contact us», которые содержат информацию о доставке, защите данных и безопасности, и контактах соответственно.

Карточка товара изображена на рисунке 2.10.

## Wardrobe Miniature Vintage Style Doll Furniture English Dollhouse Interior New



Price: 49.36\$ Order

Category : Roombox

Shipping Privacy and Security

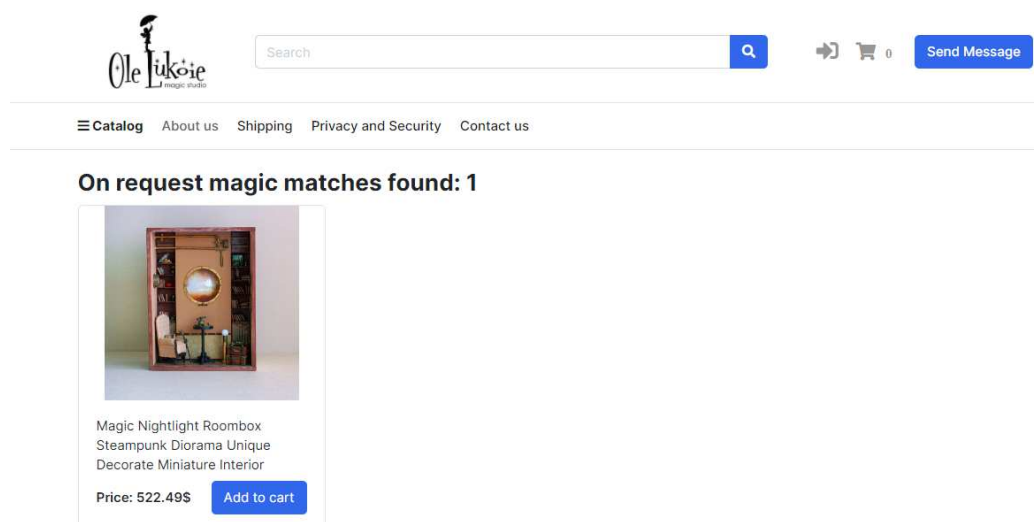
Description

Miniature for a doll house. This vintage-style chest of drawers is designed to complement your little dollhouse with a miniature piece of furniture. The kit includes miniature books and a set of small coffee cups. The dresser is made of hard cardboard, which was glued and made more durable with a primer. Varnished. The upper doors open. This chest of drawers fits perfectly into the English or classic interior of your miniature home. This product is not for children to play. Do not give the product to children under 8 years of age.

Quantity : + 1 - 2 pieces available Add to cart

Рисунок 2.10 – Карточка товара

Результат работы поискового поля показан на рисунке 2.11.



Ole Lukoie magic studio

Search

Send Message

≡ Catalog About us Shipping Privacy and Security Contact us

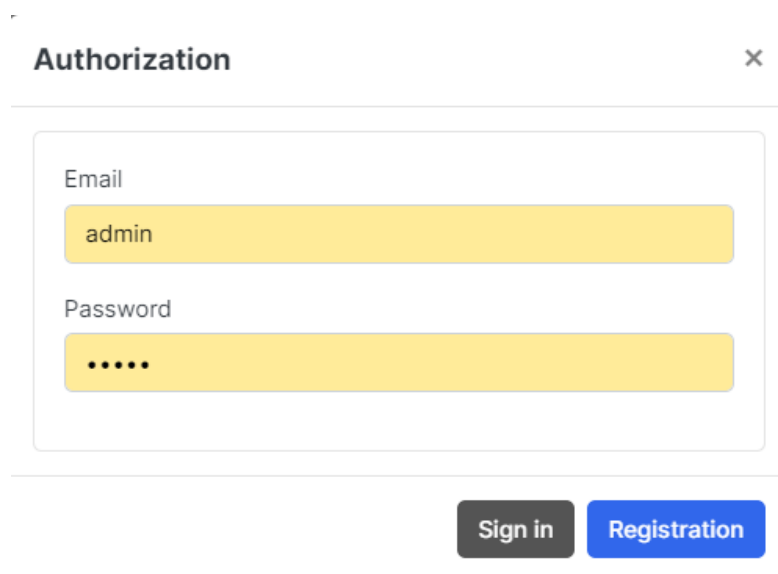
On request magic matches found: 1

Magic Nightlight Roombox  
Steampunk Diorama Unique  
Decorate Miniature Interior

Price: 522.49\$ Add to cart

Рисунок 2.11 – Результат работы поискового поля по запросу «magic»

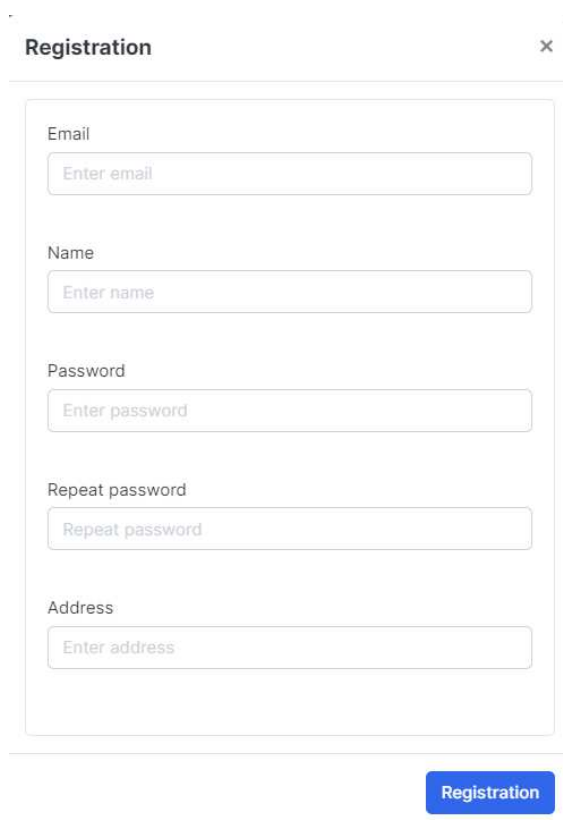
Поле авторизации показано на рисунке 2.12.



The image shows a web form titled "Authorization" with a close button (x) in the top right corner. The form contains two input fields: "Email" with the text "admin" and "Password" with five dots representing masked characters. Below the form are two buttons: "Sign in" (dark grey) and "Registration" (blue).

Рисунок 2.12 – Поле авторизации

Поле регистрации показано на рисунке 2.13.



The image shows a web form titled "Registration" with a close button (x) in the top right corner. The form contains five input fields: "Email" (placeholder: "Enter email"), "Name" (placeholder: "Enter name"), "Password" (placeholder: "Enter password"), "Repeat password" (placeholder: "Repeat password"), and "Address" (placeholder: "Enter address"). Below the form is a blue "Registration" button.

Рисунок 2.13 – Поле регистрации

Пустая корзина показана на рисунке 2.14.

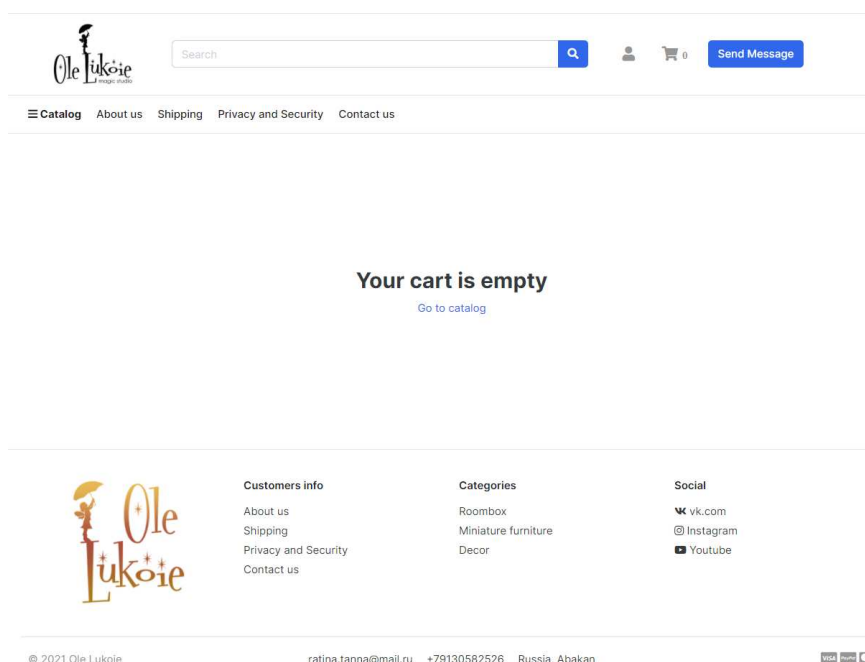


Рисунок 2.14 – Пустая корзина

Корзина с товарами показана на рисунке 2.15.

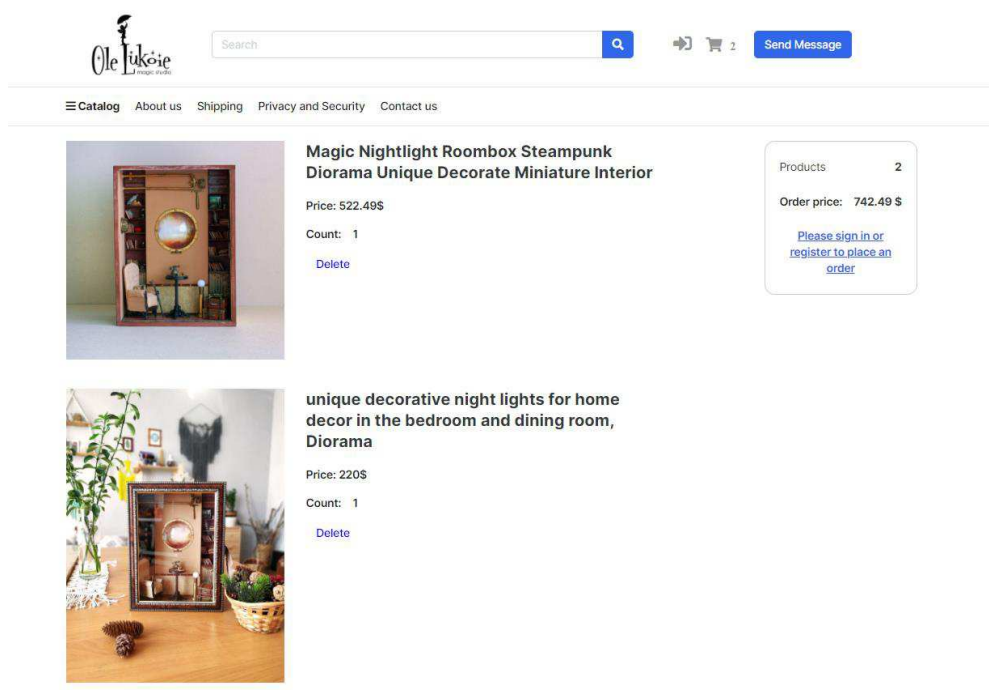


Рисунок 2.15 – Корзина с товарами

Отображение авторизованного пользователя показано на рисунке 2.16.



Рисунок 2.16 – Отображение авторизованного пользователя

Ошибка 404 показана на рисунке 2.17.

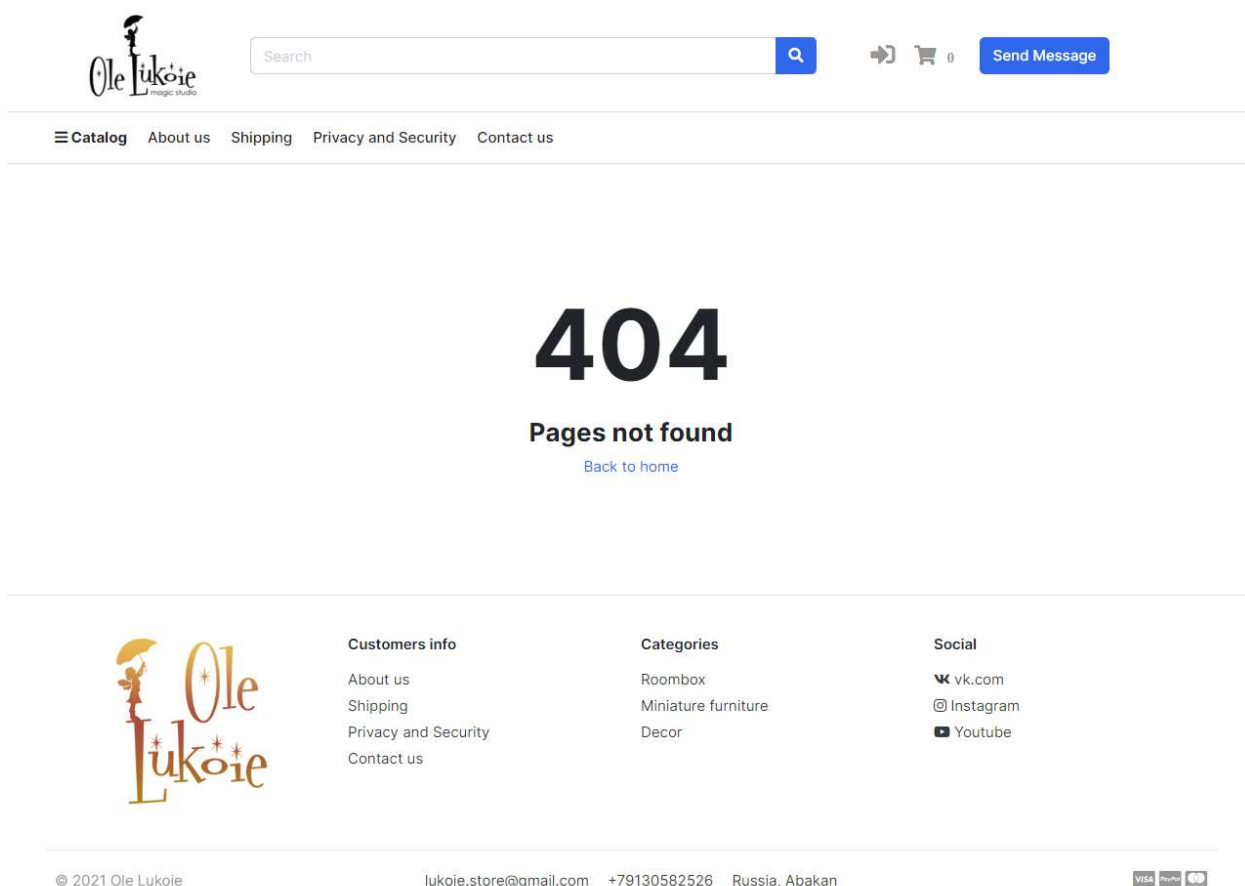


Рисунок 2.17 – Ошибка 404

Поле отправки сообщения показано на рисунке 2.18.

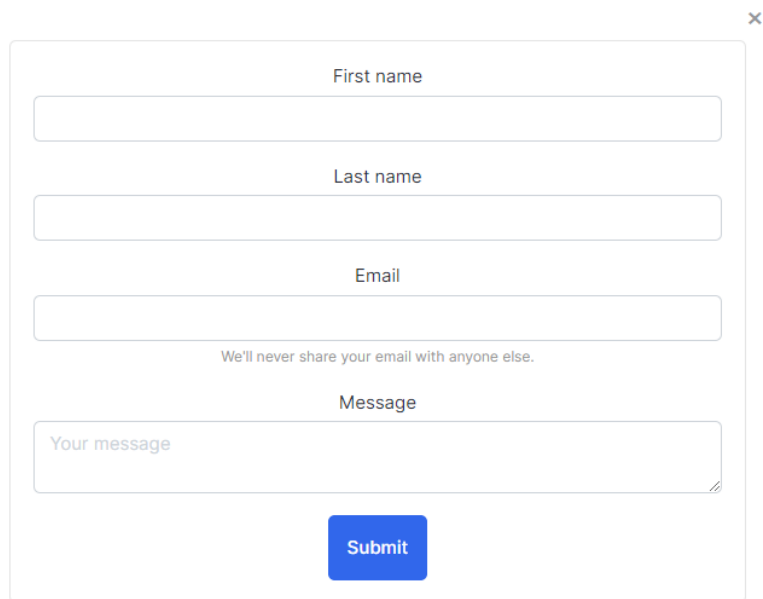
A screenshot of a contact form. It features four input fields: 'First name', 'Last name', 'Email', and 'Message'. Below the 'Email' field is a small text line: 'We'll never share your email with anyone else.' At the bottom of the form is a blue 'Submit' button. The form is enclosed in a light gray border with a small 'x' icon in the top right corner.

Рисунок 2.18 – Поле отправки сообщения администрации сайта

На этом верстка заканчивается.

### 2.3 Написание серверной части системы

Так как весь проект выполнен на PHP, и по сути, в каждом файле есть код, отвечающий за серверную часть, правильнее будет сконцентрироваться на основных файлах, отвечающих за работу базы данных, корзины, авторизации и регистрации.

Файл с данными для соединения с БД показан на рисунке 2.19.

```
1 <?php|
2 $host = 'localhost'; // адрес сервера
3 $database = 'store'; // имя базы данных
4 $user = 'root'; // имя пользователя
5 $password = ''; // пароль
6 ?>
```

Рисунок 2.19 – Файл с данными для связи с БД

Далее идет контроллер, который отвечает за авторизацию и деавторизацию пользователя в системе (Рисунок 2.20).

```

1  <?php
2  include ('connection.php'); //подключаемся к БД
3  include ('module_global.php'); //подключается файл с глобальными функциями
4  if(@$_GET['action'] == "out") out(); //если передана переменная action,
   «разавторизируем» пользователя
5  if (login()) //вызываем функцию login, которая определяет, авторизован
   пользователь или нет
6  {
7      $UID = $_SESSION['id']; //если пользователь авторизован, присваиваем
   переменной $UID его id
8  }
9  else //если пользователь не авторизован, проверяем, была ли нажата
   кнопка входа на сайт
10 {
11     if(isset($_POST['log_in'])) // Если была нажата кнопка входа,
   инициируем выполнение функции входа
12     {
13
14         $error = enter(); //функция входа на сайт
15         if (count($error) == 0) //если ошибки отсутствуют, авторизируем
   пользователя
16         {
17             $UID = $_SESSION['id']; // Записываем id пользователя в сессию
18             $response = [ // создаем массив данных для Ajax
19                 "status" => "1", // 1 означает, что ошибок нет
20             ];
21             echo json_encode($response); // функция для корректного вывода
   массива для js
22         }
23     else { // если есть ошибки, сообщаем это и выводим ошибку
24         $response = [
25             "status" => "0", // 0 означает что есть ошибки
26             "error" => $error[0], // возвращаем ошибку
27         ];
28         echo json_encode($response);
29     }
30 }
31 }
32 ?>

```

Рисунок 2.20 – Файл «controller.php»

Далее идет файл с функциями для работы с пользователем, такие как: авторизация, деавторизация, последняя дата входа на сайт, обновление данных о сессии.

Функция для входа на сайт представлена на рисунке 2.21.

```

1 <?php
2 function enter () // функция входа на сайт
3 {
4     include ('connection.php'); //подключаемся к БД
5     $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " .
        mysqli_error($link));
6     $error = array(); //массив для ошибок
7     if ($_POST['email'] != "" && $_POST['password'] != "") //если поля заполнены
8     {
9         $email = $_POST['email']; // ловим данные, отправленные из Ajax
10        $password = $_POST['password']; // ловим данные, отправленные из Ajax
11        $rez = mysqli_query($link, "SELECT * FROM users WHERE email='$email'"); //
        запрашивается строка из базы данных с логином, введённым пользователем
12        if (mysqli_num_rows($rez) == 1) //если нашлась одна строка, значит такой юзер
        существует в базе данных
13        {
14            $row = mysqli_fetch_assoc($rez);
15            if (md5(md5($password)) == $row['password']) //сравнивается хэшированный
            пароль из базы данных с хэшированным паролем, введённым
            пользователем
16            {
17                //пишутся логин и хэшированный пароль в cookie, также создаётся переменная
            сессии
18                setcookie ("email", $row['email'], time() + 50000); // создание cookie для
            нашего email
19                setcookie ("password", md5($row['email'].$row['password']), time() + 50000)
            ; // создание cookie для нашего пароля
20                $_SESSION['id'] = $row['id']; //записываем в сессию id
            пользователя
21                $id = $_SESSION['id'];
22                lastAct($id);
23                return $error;
24            }
25            else //если пароли не совпали
26            {
27                $error[] = "Invalid password";
28                return $error; // возвращаем массив с ошибкой
29            }
30        }
31        else //если такого пользователя не найдено в базе данных
32        {
33            $error[] = "Invalid username and password";
34            return $error; // возвращаем массив с ошибкой
35        }
36    }
37    else
38    {
39        $error[] = "The fields must not be empty!";
40        return $error; // возвращаем массив с ошибкой
41    }
42 }
43

```

Рисунок 2.21 – Функция входа на сайт

Функция, отвечающая за запись последней даты посещения сайта пользователем, показана на рисунке 2.22.

```

44 function lastAct($id)
45 {
46 include ('connection.php'); //подключаемся к БД
47 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка
    " . mysqli_error($link));
48 $tm = date("Y-n-j");
49 mysqli_query($link, "UPDATE users SET online='$tm', last_act='$tm' WHERE
    id='$id'"); }

```

Рисунок 2.22 – Функция, записывающая последнюю дату посещения

Далее идет функция, отвечающая за создание cookie, для корректного функционирования сессии (Рисунок 2.23).

```

51 function login () {
52 include ('connection.php'); //подключаемся к БД
53 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка
    " . mysqli_error($link));
54 @ini_set ("session.use_trans_sid", true); // включаем прозрачную
    поддержку SID для сессии
55 @session_start(); // запуск сессии
56 if (isset($_SESSION['id']))//если сессия есть
57 {
58 if(isset($_COOKIE['email']) && isset($_COOKIE['password'])) //если cookie
    есть, обновляется время их жизни и возвращается true
59 {
60 SetCookie("email", "", time() - 1, '/');
61 SetCookie("password","", time() - 1, '/');
62 setcookie ("email", $_COOKIE['email'], time() + 50000, '/');
63 setcookie ("password", $_COOKIE['password'], time() + 50000, '/');
64 $id = $_SESSION['id'];
65 lastAct($id);
66 return true;
67 }
68 else //иначе добавляются cookie с логином и паролем, чтобы после
    перезапуска браузера сессия не слетала
69 {
70 $rez = mysqli_query($link, "SELECT * FROM users WHERE id='{$_SESSION['id']
    }'"); //запрашивается строка с искомым id
71 if (mysqli_num_rows($rez) == 1) //если получена одна строка
72 {
73 $row = mysqli_fetch_assoc($rez); //она записывается в ассоциативный
    массив
74 setcookie ("email", $row['email'], time()+50000, '/');
75
76 setcookie ("password", md5($row['email'].$row['password']), time() + 50000
    , '/');
77 $id = $_SESSION['id'];
78 lastAct($id);
79 return true;
80 }

```

Рисунок 2.23 – Функция для установления «cookie», лист 1



```

81 else return false;
82 }
83 }
84 else //если сессии нет, проверяется существование cookie. Если они
      существуют, проверяется их валидность по базе данных
85 {
86 if(isset($_COOKIE['email']) && isset($_COOKIE['password'])) //если куки
      существуют
87 {
88 $rez = mysqli_query($link, "SELECT * FROM users WHERE email='{$_COOKIE['
      email']}'"); //запрашивается строка с искомым логином и
      паролем
89 @$row = mysqli_fetch_assoc($rez);
90 if(@mysqli_num_rows($rez) == 1 && md5($row['email'].$row['password']) == $_
      COOKIE['password']) //если логин и пароль нашлись в базе
      данных
91 {
92 $_SESSION['id'] = $row['id']; //записываем в сессию id
93 $id = $_SESSION['id'];
94 lastAct($id);
95 return true;
96 }
97 else //если данные из cookie не подошли, эти куки удаляются
98 {
99 setcookie("login", "");
100 setcookie("password", "");
101 // $_COOKIE['email'] = '';
102 // $_COOKIE['password'] = '';
103 return false;
104 }
105 }
106 else //если куки не существуют
107 {
108 return false;
109 }
110 }
111 }
112 ?>

```

Рисунок 2.23, лист 2

Далее идет файл для регистрации пользователя (Рисунок 2.24).

```

<?php
$data = $_POST;
$email_reg = $_POST['email_reg'];
$name_reg = $_POST['name_reg'];
$password_reg = $_POST['password_reg'];
$repassword_reg = $_POST['repassword_reg']; // Получаем все необходимые для регистрации данные из Ajax
require_once 'connection.php';
$link = mysqli_connect($host, $user, $password, $database)
    or die("Ошибка " . mysqli_error($link));
if(isset($_POST['do_signup'])) { // если была нажата кнопка регистрации, инициируется выполнение регистрации
    $errors = array();
    // Все последующие проверки, проверяют форму и выводят ошибку
    // Проверка на совпадение паролей
    if($repassword_reg != $password_reg) {
        $errors[] = "The repeated password was entered incorrectly!";
    }
    // Проверка есть ли вообще повторный пароль
    if($repassword_reg == '') {
        $errors[] = "Enter a second password";
    }
    if(trim($name_reg) == '') {
        $errors[] = "Enter name";
    }
    // Проверка есть ли пароль
    if($password_reg == '') {
        $errors[] = "Enter password";
    }
    // Проверка есть ли email
    if(trim($email_reg) == '') {
        $errors[] = "Enter Email";
    }
    if (mb_strlen($name_reg) < 1 || mb_strlen($name_reg) > 50){
        $errors[] = "Invalid name length";
    }
    if (mb_strlen($password_reg) < 2 || mb_strlen($password_reg) > 8){
        $errors[] = "Invalid password length (2 to 8 characters)";
    }
    if (!preg_match("/[0-9a-z_]+@[0-9a-z_\.]+\.[a-z]{2,3}/i", $email_reg)) {
        $errors[] = 'E-mail entered incorrectly';
    }
    $query1 = mysqli_query($link, "SELECT id FROM users WHERE email='".mysqli_real_escape_string($link, $email_reg)."'");
    if(mysqli_num_rows($query1) > 0)
    {
        $errors[] = "A user with this email already exists in the database";
    }
    // Если ошибок нет, то происходит регистрация
    if(empty($errors)) {
        $email = $email_reg;
        $name = $name_reg;
        // Пароль хешируется
        $pass = md5(md5(trim($password_reg)));
        // добавление пользователя
        mysqli_query($link, "INSERT INTO `users` (`email`, `name`, `password`) VALUES ('" . $email . "', '" . $name . "', '" . $pass . "')");
        $response = [
            "status" => "1",
        ];
        echo json_encode($response);
        // Подтверждение что всё хорошо
    } else {
        $response = [
            "status" => "0",
            "errors" => $errors[0],
        ];
        echo json_encode($response);
        // Если ошибка есть, то выводить её
    }
}
?>

```

Рисунок 2.24 – Файл регистрации пользователя

Последним файлом, для взаимодействия пользователя со своей учетной записью является функция выхода из учетной записи (Рисунок 2.25).

```

1 <?php
2 include ('connection.php'); //подключаемся к БД
3 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " .
    mysqli_error($link));
4 session_start();
5 $id = $_SESSION['id']; // записываем id пользователя из сессии
6 mysqli_query($link, "UPDATE users SET online=0 WHERE id='$id'"); //обнуляется
    поле online, говорящее, что пользователь вышел с сайта (пригодится в
    будущем)
7 unset($_SESSION['id']); //удалятся переменная сессии
8 setcookie("email", ""); //удаляются cookie с логином
9 setcookie("password", ""); //удаляются cookie с паролем
10 header("location: http://localhost/store"); //перенаправление на главную
    страницу сайта |
11 ?>

```

Рисунок 2.25 – Функция выхода из учетной записи

Далее идут функции взаимодействия с корзиной товаров.

Функция для изменения количества товаров в значке с изображением корзины в правом верхнем углу показана на рисунке 2.26.

```

1 <?php
2 $id = $_POST['id_tov']; //получаем id
3 $col = $_POST['col_tov']; //получаем количество
4 session_start(); //стартуем сессию
5 $temp = $_SESSION['cart']; //переносим сессию во временную переменную
6 $temp[$id]=$col; //изменяем количество
7 $_SESSION['cart'] = $temp; //сохраняем сессию
8 ?>

```

Рисунок 2.26 – Функция изменения количества товара в корзине

Функция для добавления товара в корзину показана на рисунке 2.27.

```

1 <?php
2 $id = $_POST['id_tov']; //получаем id товара
3 $q = $_POST['quantity']; // получаем количество товара
4 $price = $_POST['price']; // получаем цену товара
5 session_start(); // стартуем сессию
6 if (!isset($_SESSION['cart'])) { //если сессия корзины не существует
7     $temp[$id]['quantity'] = $q; //в массив заносим количество товара 1
8     $temp[$id]['price'] = $price; //в массив заносим количество товара 1
9 } else { //если в сессии корзины уже есть товары
10    $temp = $_SESSION['cart']; //заносим в массив старую сессию
11    if (!array_key_exists($id, $temp)) { //проверяем есть ли в корзине уже такой товар
12        $temp[$id]['quantity'] = $q; //в массив заносим количество товара 1
13        $temp[$id]['price'] = $price; //в массив заносим количество товара 1
14    }
15    else {
16        $temp[$id]['quantity'] = $temp[$id]['quantity'] + $q;
17        $temp[$id]['price'] = $price; //в массив заносим количество товара 1
18    }
19 }
20 $count = count($temp); //считаем товары в корзине
21 $_SESSION['cart'] = $temp; //записываем в сессию наш массив
22 // $_SESSION['cart'] = $price;
23 echo $count; //возвращаем количество товаров
24 ?>

```

Рисунок 2.27 – Функция для добавления товара в корзину

Функция для удаления товара из корзины представлена на рисунке 2.28.

```

1 <?php
2 $id = $_POST['id_tov']; //получаем id товара
3 $price_tov = $_POST['price_tov']; // получаем цену товара
4 $count_tov = $_POST['count_tov']; // получаем кол-во товара
5 session_start(); //стартуем сессию
6 $temp = $_SESSION['cart']; //переносим сессию во временную переменную
7 if ($temp){
8     unset ($temp[$id]); //удаляем запись
9 }
10 if (empty($_SESSION['cart'])){ // если корзина пуста
11 }
12 $_SESSION['cart'] = $temp; //сохраняем сессию
13 $response = [ // пишем ответ для Ajax
14     "count" => count($temp), //считаем товары в корзине
15     "price" => $price_tov, // возвращаем цену товаров
16     "count_tov" => $count_tov, //возвращаем количество товаров
17 ];
18 echo json_encode($response); // функция для правильного вывода массива в JS
19 ?>

```

Рисунок 2.28 – Функция для удаления товаров из корзины

На этом основная часть серверной обработки сервиса закончена.

## 2.4 Написание скриптов

Написание скриптов нужно для «оживления» кнопок сайта, и так как серверная часть описана в предыдущем разделе, необходимо описать основные скрипты на страницах сайта. Так как на сайте много однотипных скриптов, необходимо описать только самые главные.

Скрипт для регистрации пользователя показан на рисунке 2.29.

```
277 <script>
278     $(''.btn-registration').click(function () { //клик на кнопку
279         var email_reg = $(".email_reg").val(); // передаем email
280         var name_reg = $(".name_reg").val(); // передаем имя
281         var password_reg = $(".password_reg").val(); // передаем пароль
282         var reppassword_reg = $(".reppassword_reg").val(); // передаем повторный пароль
283         var do_signup = 1; // передаем нажатие на кнопку
284         $.ajax({ //передаем а́жак-запросом данные
285             type: "POST", //метод передачи данных
286             url: 'reg.php', //php-файл для обработки данных
287             data: {email_reg: email_reg, name_reg: name_reg, password_reg: password_reg,
288                 reppassword_reg: reppassword_reg, do_signup: do_signup}, //передаем наши данные
289             success: function(data) {
290                 var response = JSON.parse(data);
291                 if (response['status'] == 1){ // если регистрация прошла успешно, уведомляем об этом
292                     //пользователя
293                     $(".card-body-script").css({display: 'none'}); // скрываем поля для регистрации
294                     $(".modal-register").css({display: 'block'}); // показываем пользователю
295                     //сообщение об успешной регистрации
296                     $(".btn-register").css({display: 'block'}); // показываем пользователю кнопку
297                     //входа
298                 }
299                 else { // если пользователь неправильно ввел данные, уведомляем его об этом, и не
300                     //регистрируем его
301                     alert(response['errors']); // показываем ошибку
302                 }
303             });
304         });
305     });
306     $(''.login-script').click(function () { // после регистрации пользователя, он может нажать на
307         //кнопку входа, которые вывели ранее, прямо в поле для регистрации
308         $(".#exampleModal1").css({display: 'none'}); // после клика по кнопке входа скрываем окно
309         //регистрации
310     });
311 </script>
```

Рисунок 2.29 – Скрипт регистрации пользователя

Скрипт кнопки авторизации представлен на рисунке 2.30.

```

<script>
    $(' .btn-signin').click(function () { //клик на кнопку
        var email = $(".email").val(); // передаем email
        var password = $(".password").val(); // передаем пароль
        var log_in = 1; // передаем нажатие кнопки входа
        $.ajax({ //передаем а́жак-запросом данные
            type: "POST", //метод передачи данных
            url: 'controller.php', //php-файл для обработки данных
            data: {email: email, password: password, log_in: log_in}, //передаем наши данные
            success: function(data) {
                var response = JSON.parse(data); // функция для правильного чтения массива "data"
                if (response['status'] == 1){ // если ошибок нет, входим на сайт и перезагружаем
                    страницу
                    document.location.href = 'http://localhost/store'; // перенаправление на главную
                    страницу
                }
                else { // если ошибки есть, не авторизируем пользователя и выводим ошибку

                    alert(response['error']); // вывод ошибки
                }
            }
        });
    });
</script>

```

Рисунок 2.30 – Скрипт кнопки авторизации

Далее идут скрипты для взаимодействия пользователя с корзиной.

Скрипт для взаимодействия кнопок уменьшения и увеличения количества товара показан на рисунке 2.31.

```

<script type="text/javascript">
    $(' .minus-btn').on('click', function(e) { // клик по кнопке уменьшения количества товара
        e.preventDefault(); // функция для корректного выполнения скрипта
        var $this = $(this); // выделяем объект по которому сделан клик
        var $input = $this.closest('div').find('input'); // находим объект, где отображается
        количество товара
        var value = parseInt($input.val()); // берем из объекта с количеством товара текущее кол-во
        if (value > 1) { // действие выполняется, пока товара больше, чем 1
            value = value - 1; // меняем количество товара на один меньше
        } else { // если товара меньше, чем 1
            value = 0; // меняем значение кол-ва на 0
        }
        $input.val(value); // выводим кол-во товара в поле
    });
    $(' .plus-btn').on('click', function(e) { // клик по кнопке увеличения количества товара
        e.preventDefault(); // функция для корректного выполнения скрипта
        var $this = $(this); // выделяем объект по которому сделан клик
        var $input = $this.closest('div').find('input'); // находим объект, где отображается
        количество товара
        var value = parseInt($input.val()); // берем из объекта с количеством товара текущее кол-во
        if (value < 100) { // действие выполняется, пока товара меньше, чем 100
            value = value + 1; // меняем количество товара на один больше
        } else { // если товара больше, чем 100
            value = 100; // меняем значение кол-ва на 100
        }
        $input.val(value); // выводим кол-во товара в поле
    });
</script>

```

Рисунок 2.31 – Скрипт кнопок «Плюс» и «Минус»

Скрипт для добавления товара в корзину показан на рисунке 2.32.

```
<script>
    $(' .btn-buy').click(function () { //клик на кнопку
        var id = $(this).attr('id'); //получаем id этой кнопки
        var q = $(' .quantity_id').val(); // получаем кол-во товара
        var price = $(this).attr('price'); // получаем цену товара
        $.ajax({ //передаем а́жак-запросом данные
            type: "POST", //метод передачи данных
            url: 'addtocart.php', //php-файл для обработки данных
            data: {id_tov: id, quantity:q, price:price}, //передаем наши данные
            success: function(data) {
                $(' .basker_kol').html(data); //меняем количество товаров на кнопке корзины
            }
        });
        $(this).css({background: "#3AC267"}); // меняем фон на кнопке, чтобы пользователь понял,
        что она активировалась
        $(' .addToCart').html('In the cart: '+' +q); // пишем в кнопке сколько товара
        пользователь добавил в корзину
    });
</script>
```

Рисунок 2.32 – Скрипт кнопки «Добавление в корзину»

Далее идет кнопка удаления товара в самой корзине (Рисунок 2.33).

```

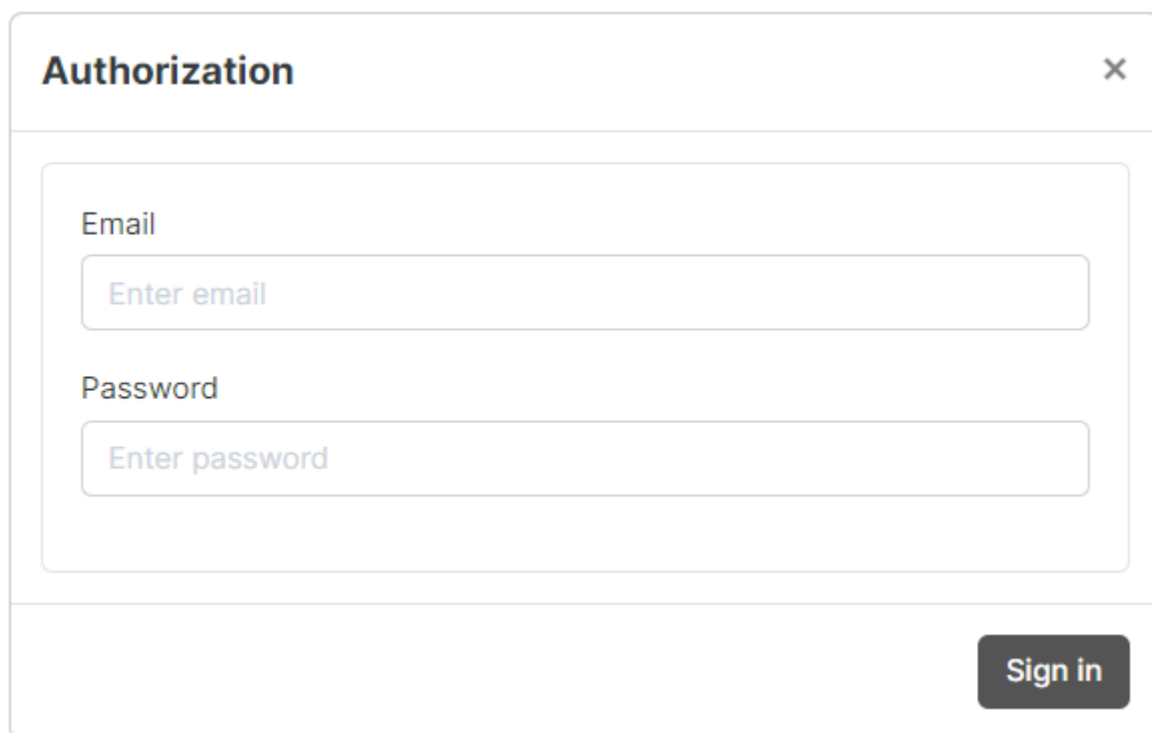
    $(' .btn-del').click(function () { //клик на кнопку
        var id = $(this).attr('id'); //получаем id товара
        var price_tov = $(this).attr('price_tov'); // получаем цену товара
        var count_tov = $(this).attr('quantity_tov'); // получаем кол-во товара
        var jabaproduct_count = $('#jabaproduct_id').text(); // получаем кол-во всех товаров в корзине
        $.ajax({ //а́жак-запрос
            type: "POST", //метод
            url: 'cartdel.php', //куда передаем
            data: {id_tov: id, price_tov: price_tov, count_tov: count_tov}, //данные
            success: function(data) { //получаем результат
                //тут можно пересчитать сумму
                var response = JSON.parse(data);
                $('#'+id).css('display', 'none'); //скрываем строку таблицы
                var col_price = parseFloat($('#jabaproduct_price').html()); // получаем цену
                удаляемого товара
                var answ = (col_price) - (parseFloat(response['price']))*response['count_tov'];
                // уменьшаем цену корзины на сумму удаляемого товара
                $(' .basker_kol').html(response['count']); // получаем кол-во товара на значке
                корзины
                $(' .jabaproduct').html(jabaproduct_count-response['count_tov']); // изменяем кол-во
                товаров в корзине на кол-во удаляемых товаров
                $(' .jabaproduct_price').html(answ.toFixed(2)); // ограничиваем цену покупки в
                корзине двумя знаками после точки
                var col = parseInt($('# .basker_kol').html()); // изменяем кол-во товара на
                значке корзины
                if (col == 0) { // если товаров в корзине нет, выводим сообщение об этом
                    $('# .error').html("<div style='padding-top: 185px; padding-bottom: 185px;
                    text-align: center;'><h2 >Your cart is empty</h2><a href='./category'>Go
                    to catalog</a></div>");
                }
            }
        });
    });
});
```

Рисунок 2.33 – Скрипт на удаление товара из корзины

На этом основные функции сайта закончены.

## 2.5 Создание админ-панели

Создание панели начинается с формы авторизации, которая доступна при вводе в URL следующего адреса: <https://lukoye.com/admin>. И выглядит как на рисунке 2.34.



The image shows a web form titled "Authorization" with a close button (X) in the top right corner. The form contains two input fields: "Email" with the placeholder text "Enter email" and "Password" with the placeholder text "Enter password". At the bottom right of the form is a dark button labeled "Sign in".

Рисунок 2.34 – Форма входа в админ-панель

Для удобства повествования серверная часть админ-панели также будет представлена в этом разделе, так как клиентская и администраторская часть системы разделены. Скрипт для входа в админ-панель представлен на рисунке 2.35.



```

1 <?php
2 session_start(); // стартуем сессию
3 $error = enter(); // вызываем функцию enter, которая записывает ошибки в переменную $error
4 if (count($error) == 0) { //если ошибки отсутствуют, авторизуем пользователя
5     $response = [ // генерируем ajax ответ
6         "status" => "1", // если авторизация успешна, присваиваем статус 1
7     ];
8     echo json_encode($response); // вывод статуса
9 } else {
10    $response = [
11        "status" => "0", // если авторизация не успешна, присваиваем статус 0
12        "error" => $error[0], // записываем ошибки
13    ];
14    echo json_encode($response); // вывод статуса и массива с ошибками
15 }

```

Рисунок 2.35 – Скрипт входа в админ-панель, лист 1

```

16 function enter () {
17     include ('../connection.php'); //подключаемся к БД
18     $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link));
19     $error = array(); //массив для ошибок
20     if ($_POST['email'] != "" && $_POST['password'] != "") { //если поля заполнены
21         $email = $_POST['email']; // получаем email
22         $password = $_POST['password']; // получаем пароль
23         $rez = mysqli_query($link, query: "SELECT * FROM users WHERE email='$email'"); //запрашивается строка из базы данных с логином, введённым пользователем
24         if (mysqli_num_rows($rez) == 1) { //если нашлась одна строка, значит такой юзер существует в базе данных
25             $row = mysqli_fetch_assoc($rez);
26             if (md5(md5($password)) == $row['password']) { //сравнивается хэшированный пароль из базы данных с хэшированным паролем, введённым пользователем
27                 if ($row['permissions'] == 'admin') { // если пользователь админ
28                     setcookie ("email_admin", $row['email'], time() + 50000); // записываем в куки email
29                     setcookie ("password_admin", md5( string: $row['email'].$row['password']), time() + 50000); // записываем в куки пароль
30                     $_SESSION['admin_id'] = $row['id']; //записываем в сессию id пользователя
31                     $id = $_SESSION['id']; // создаем переменную id для функции lastAct
32                     lastAct($id); // запускаем функцию, определяющую последний вход в учетную запись
33                     return $error;
34                 } else { // если пользователь не админ
35                     $error[] = "Вы не являетесь админом";
36                     return $error;
37                 }
38             } else { //если пароли не совпали
39                 $error[] = "Неправильный пароль";
40                 return $error;
41             }
42         } else { //если такого пользователя не найдено в базе данных
43             $error[] = "Неверный email и пароль";
44             return $error;
45         }
46     } else { // если одно или несколько полей пустые
47         $error[] = "Поля не должны быть пустыми!";
48         return $error;
49     }
50 }

```

Рисунок 2.35, лист 2

```

51 function lastAct($id) { // функция записи последнего входа в учетную запись
52     include ('../connection.php'); //подключаемся к БД
53     $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link));
54     $tm = date( format: "Y-n-j"); // генерируем актуальную дату
55     mysqli_query($link, query: "UPDATE users SET online='$tm', last_act='$tm' WHERE id=$id"); // записываем в базу данных дату последнего входа
56 }

```

Рисунок 2.35, лист 3

Как можно заметить, в скрипте используется Ажак-метод передачи данных, поэтому необходимо показать обратную часть Ажак-скрипта (Рисунок 2.36).

```
58 <script>
59     $(' .btn-signin').click(function () { //клик на кнопку
60         var email = $(' .email').val(); // получаем email
61         var password = $(' .password').val(); // получаем пароль
62         $.ajax({ //передаем ажак-запросом данные
63             type: "POST", //метод передачи данных
64             url: 'admin_module.php', //php-файл для обработки данных
65             data: {email: email, password: password}, //передаем наши данные
66             success: function(data) {
67                 var response = JSON.parse(data); // получаем ответ
68                 if (response['status'] == 1) { // если ответ положительный
69                     document.location.href = './catalog.php'; // осуществить вход
70                 } else { // если ответ отрицательный
71                     alert(response['error']); // вывести ошибку
72                 }
73             }
74         });
75     });
76 </script>
```

Рисунок 2.36 – Ажак скрипт для передачи данных

Если данные, которые ввел пользователь, соответствуют данным в базе данных, и у него есть права админа, произойдет вход в админ-панель, а именно на страницу с каталогом товаров (Рисунок 2.37).

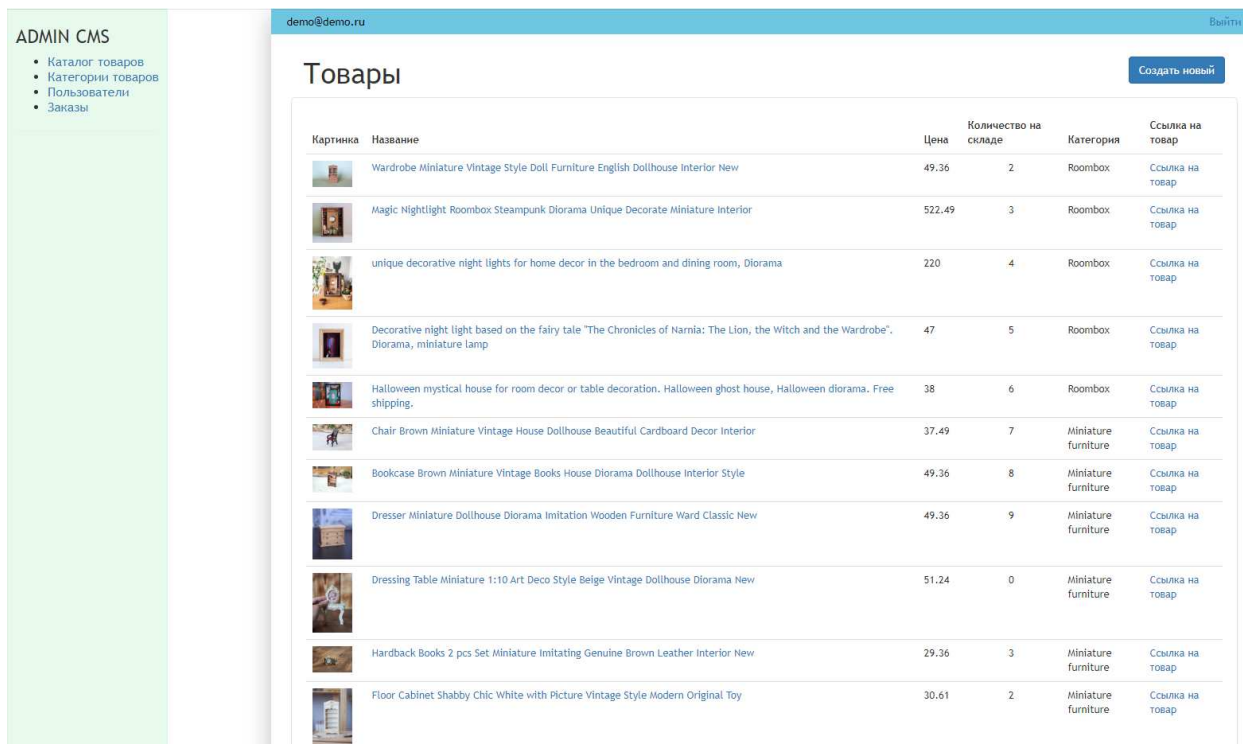


Рисунок 2.37 – Страница с каталогом товаров

Как можно заметить слева располагается список доступных для администратора ссылок. Это меню доступно из любой страницы админ-панели, и далее на скриншотах показываться не будет. Также можно обратить внимание на email адрес в верхней части страницы, это email администратора, который осуществил вход в панель, и кнопка выхода справа, она отвечает за выход из учетной записи администратора. Скрипт выхода из учетной записи в админ-панели показан на рисунке 2.38.

```

1 <?php
2 include ('../connection.php'); //подключаемся к БД
3 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link)); // связываемся с базой данных
4 session_start(); // стартуем сессию
5 echo $_COOKIE['email_admin']; // получаем куки с email администратора
6 echo $_COOKIE['password_admin']; // получаем куки с паролем администратора
7 $id = $_SESSION['id']; // записываем id пользователя из сессии
8 mysqli_query($link, query: "UPDATE users SET online=0 WHERE id='$id'"); //обнуляется поле online, говорящее, что пользователь вышел с сайта
9 unset($_SESSION['id']); //удаляются переменная сессии
10 setcookie('email_admin', "", time()-3600); // удаляются куки с email
11 setcookie('password_admin',"", time()-3600); // и паролем администратора
12 header( header: "location: ../admin"); //перенаправление на главную страницу сайта
13

```

Рисунок 2.38 – Скрипт выхода из учетной записи администратора

На этой же странице с каталогом товаров выведены все товары, из базы данных. На этой странице можно нажать кнопку создания нового товара, или изменить существующий, кликнув по его названию. В списке с товарами представлена необходимая информация о нем, а также ссылка на него в самом магазине.

На каждой странице присутствует защита от простого перебора URL адресов, чтобы пользователь, который не имеет доступа к админ-панели не смог зайти на страницы узнав их адрес (Рисунок 2.39).

```
23 <?php
24 session_start(); // старт сессии
25 include ('../connection.php'); //подключаемся к БД
26 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link)); // подключение к бд
27 @$email = $_COOKIE['email_admin']; // получаем email пользователя из куки
28 $rez = mysqli_query($link, query: "SELECT * FROM users WHERE email='$email'"); // делаем запрос в бд по полученному email
29 $row = mysqli_fetch_assoc($rez); // получаем данные запроса
30 if (@$row['permissions'] != 'admin') { // если пользователь не имеет прав администратора
31     header( header: "Location: ../admin"); // отправить на страницу входа
32 }
33 >
```

Рисунок 2.39 – Защита от несанкционированного вторжения

Представлять верстку страниц админ-панели не имеет смысла, она очень однотипная и не отличается друг от друга по написанию, поэтому далее по структуре идет страница создания нового товара, кнопка «Создать новый», которая отвечает за интерфейс создания нового товара (Рисунок 2.40).

## Товар:

The form consists of the following elements:

- Заголовок**: A text input field.
- Цена товара**: A text input field.
- Количество на складе**: A text input field.
- Категория**: A dropdown menu.
- Размер товара**: A rich text editor with a toolbar and a watermark 'ТУРИ'.
- Материалы товара**: A rich text editor with a toolbar and a watermark 'ТУРИ'.
- Описание**: A rich text editor with a toolbar and a watermark 'ТУРИ'.
- Изображение страницы**: A separate box containing a file selection button 'Выберите файл' and the text 'Файл не выбран'.
- Сохранить**: A blue button at the bottom left.

Рисунок 2.40 – Страница создания нового товара

Данная страница необходима для создания нового товара. Здесь можно указать все необходимые свойства товара, загрузить картинку и, если данные пройдут фильтрацию, товар будет успешно добавлен в базу данных, после чего пользователя перенаправит на страницу с этим товаром (Рисунок 2.41).

## Товар:

"Wardrobe Miniature Vintage Style Doll Furniture English Dollhouse Interior New"


<p><b>Заголовок</b></p> <input type="text" value="Wardrobe Miniature Vintage Style Doll Furniture English Dollhouse Interior New"/>	<p><b>Изображение страницы</b></p>  <p>Выберите файл   файл не выбран</p>
<p><b>Артикул</b></p> <input type="text" value="16"/>	
<p><b>Цена товара</b></p> <input type="text" value="49.36"/>	
<p><b>Количество на складе</b></p> <input type="text" value="2"/>	
<p><b>Категория</b></p> <input type="text" value="Roombox"/>	
<p><b>Размер товара</b></p> <div style="border: 1px solid #ccc; padding: 2px;"> <p>B / S</p> </div>	
<p><b>Материалы товара</b></p> <div style="border: 1px solid #ccc; padding: 2px;"> <p>UP</p> </div>	
<p><b>Описание</b></p> <div style="border: 1px solid #ccc; padding: 5px;"> <p>MARK IT UP!</p> <p>Miniature for a doll house. This vintage-style chest of drawers is designed to complement your little dollhouse with a miniature piece of furniture. The kit includes miniature books and a set of small coffee cups.</p> <p>The dresser is made of hard cardboard, which was glued and made more durable with a primer. Varnished. The upper doors open.</p> <p>This chest of drawers fits perfectly into the English or classic interior of your miniature home. This product is not for children to play.</p> <p>Do not give the product to children under 8 years of age.</p> </div>	
<p><input type="button" value="Сохранить"/> <input type="button" value="Удалить страницу"/></p>	

Рисунок 2.41 – Страница существующего товара

Это страница существующего товара, попасть на нее можно либо кликнув по названию товара в списке из рисунка 2.37, либо же, как уже было сказано выше, создав новый товар, система перенаправит пользователя на страницу этого товара. Необходима она для редактирования информации о существующем товаре, либо же удаления этого товара. Как можно заметить,

все поля, кроме поля «Артикул» поддаются редактированию. Поле «Артикул» не что иное, как поле «id» в базе данных. На этом работа с товарами закончена, далее идут категории. Попасть на эту страницу можно с помощью меню, изображенном на рисунке 2.37, кликнув по одноименной ссылке. На рисунке 2.42 показана страница с категориями.

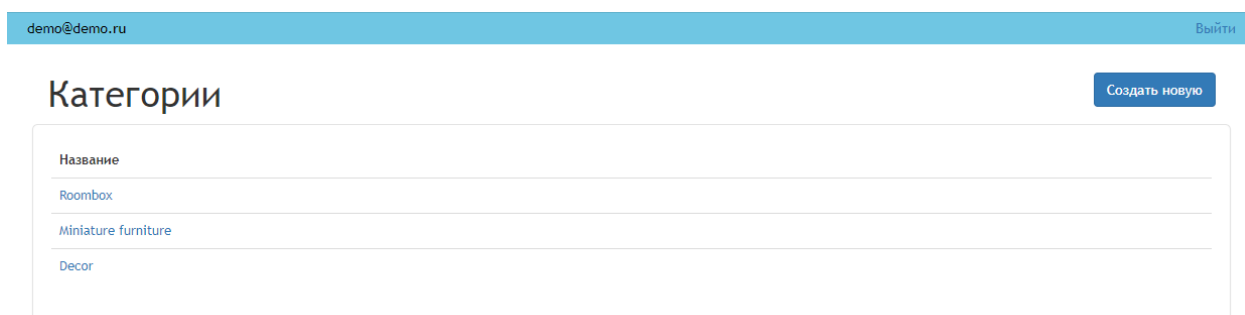


Рисунок 2.42 – Список категорий

На этой странице можно создать новую категорию, нажав соответствующую кнопку, а также изменить имеющуюся, нажав на название интересующей категории. Страница создания новой категории показана на рисунке 2.43.

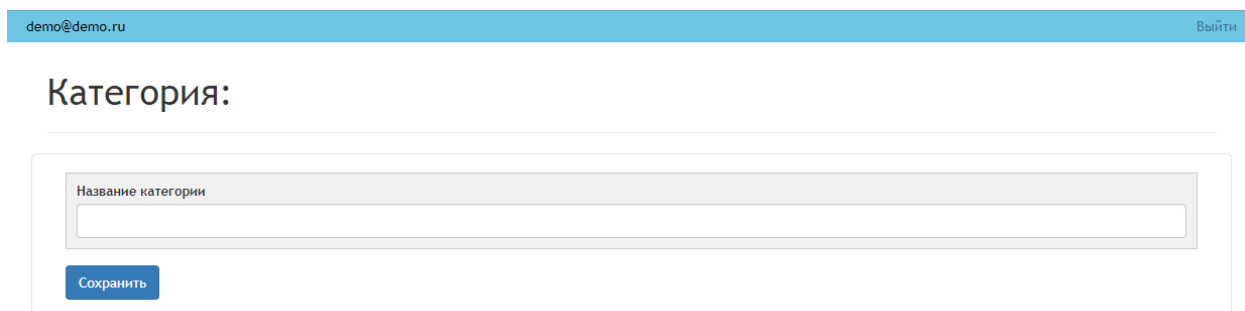


Рисунок 2.43 – Страница создания новой категории

На этой странице вводится название категории и, если такого названия нет, создается новая категория, и пользователя перенаправляет обратно на страницу с категориями, если же категория с таким же названием есть в базе

данных, новая категория не создается, а пользователь получает предупреждение (Рисунок 2.44).

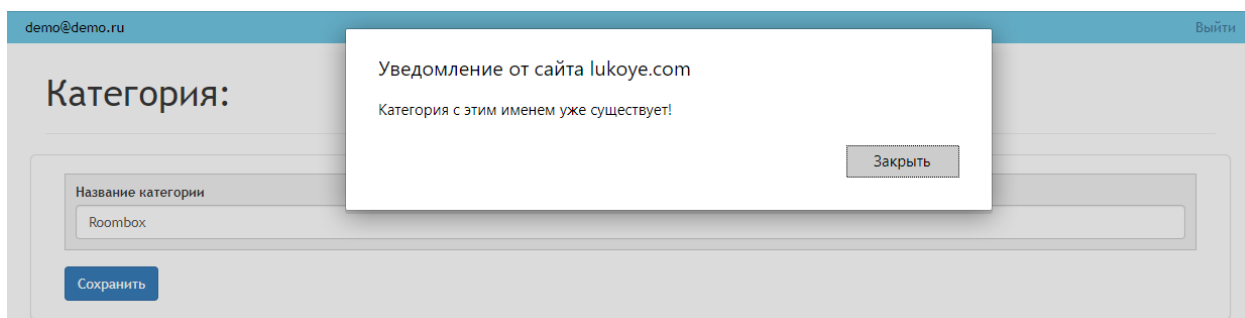


Рисунок 2.44 – Предупреждение о существовании категории с введенным именем

Далее следует страница существующей категории, попасть на нее можно нажав на название категории в списке с категориями на рисунке 2.42. Пример существующей категории показан на рисунке 2.45.

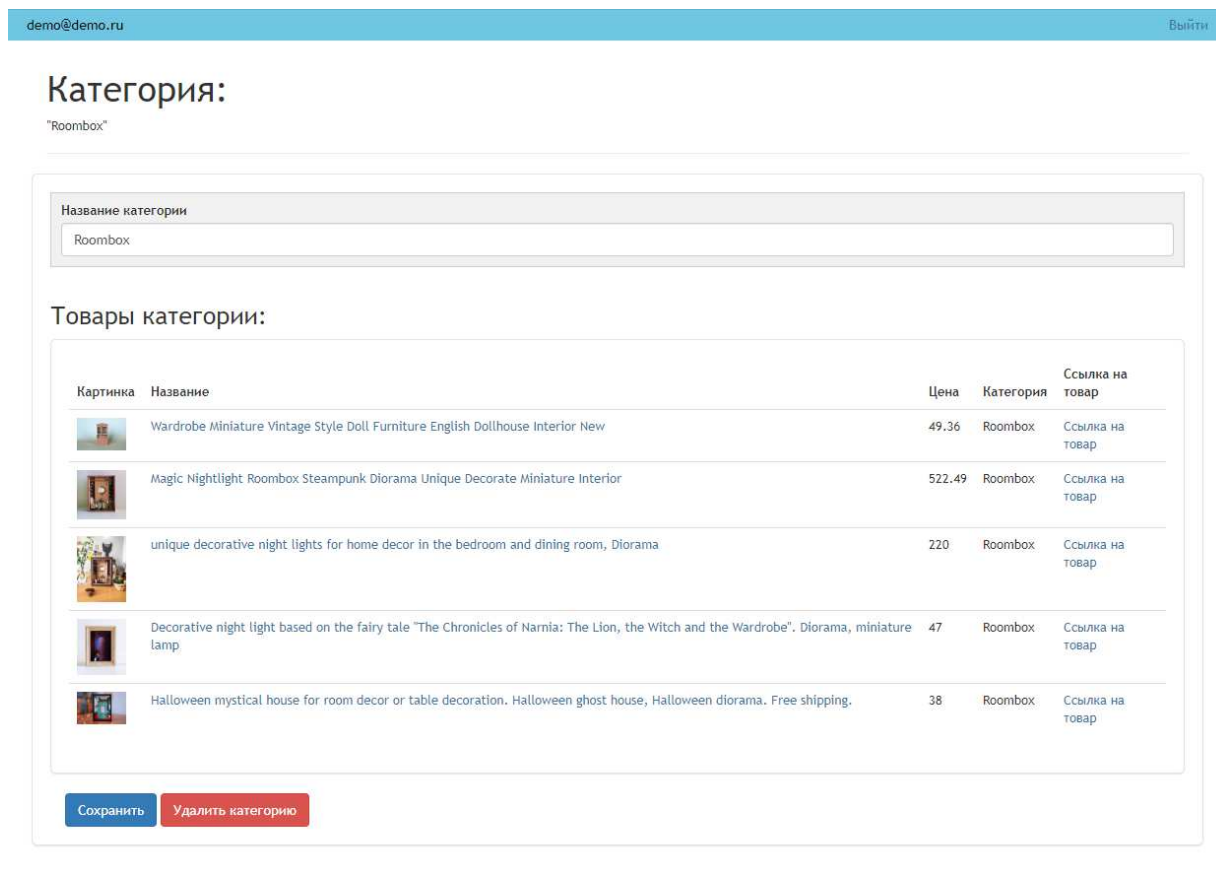


Рисунок 2.45 – Страница существующей категории



На этой странице есть возможность изменить название категории, а также увидеть список товаров этой категории, с доступными ссылками на изменение товара и этим товаром в магазине. Следует заметить, что система не даст администратору удалить категорию, если есть товары с этой категорией (Рисунок 2.46).

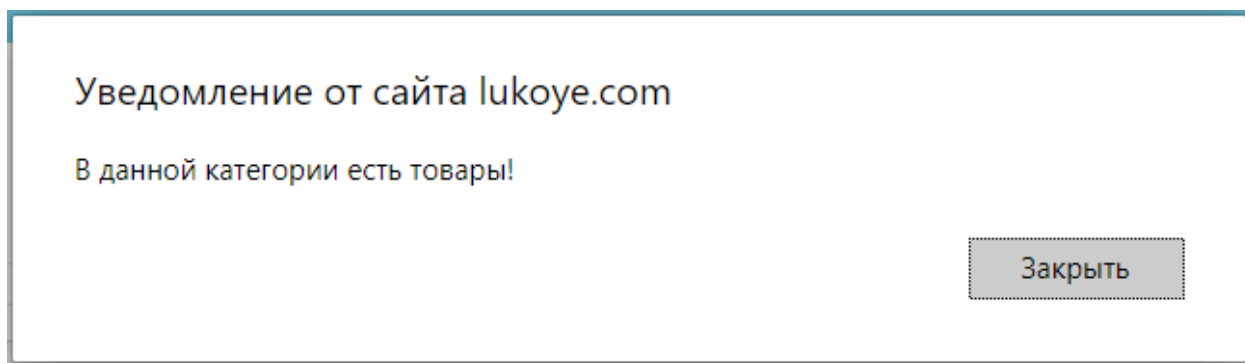


Рисунок 2.46 – Уведомление о невозможности удаления категории

Поэтому перед удалением необходимо убрать эту категорию у товаров, которые находятся в списке на странице категории.

На этом функционал взаимодействия с категориями заканчивается, далее идут пользователи. Попасть на страницу с пользователями можно через меню, нажав на соответствующую ссылку. Страница со списком пользователей показана на рисунке 2.47.

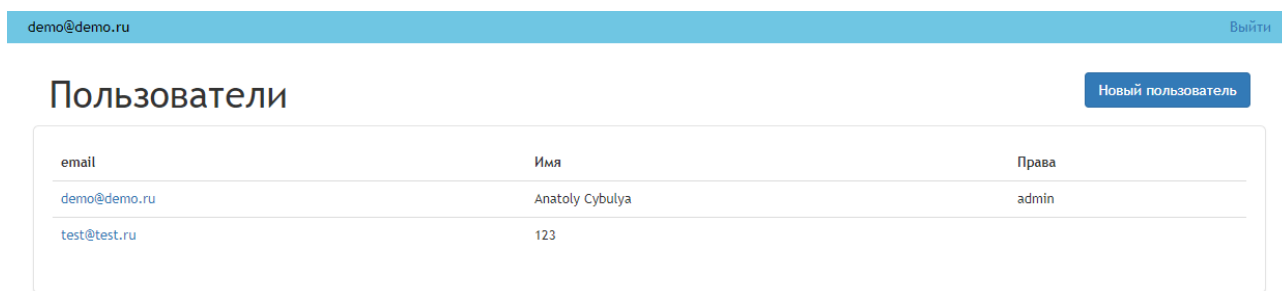


Рисунок 2.47 – Страница с пользователями

Аналогично предыдущим страницам, на странице с пользователями доступна кнопка создания нового пользователя, а также редактирование

существующего. Страница создания нового пользователя показана на рисунке 2.48.

The screenshot shows a web interface for creating a new user. At the top, there is a blue header bar with the email address 'demo@demo.ru' on the left and a 'Выйти' (Logout) link on the right. Below the header, the title 'Пользователь:' (User:) is displayed. The main content area contains a form with six input fields, each with a label above it: 'Имя' (Name), 'Email', 'Пароль' (Password), 'Повторите пароль' (Repeat password), 'Права' (Rights), and 'Адрес' (Address). Each field is a simple rectangular box. At the bottom left of the form, there is a blue button with the text 'Сохранить' (Save).

Рисунок 2.48 – Страница создания нового пользователя

Эта страница необходима для создания нового пользователя, для этого необходимо заполнить все поля, кроме поля «Права», в случае, если создаваемый пользователь не является администратором. Если все введенные данные прошли фильтрацию, создается новый пользователь и осуществляется перенаправление на страницу с этим пользователем, но, если данные не прошли фильтрацию (об этом чуть позже), система уведомит об этом и не создаст нового пользователя.

Далее идет страница существующего пользователя, на которую можно попасть, кликнув по имени пользователя в списке из рисунка 2.47, либо после успешного создания нового пользователя. Пример существующего пользователя показан на рисунке 2.49.

## Пользователь:

Anatoly Cybulya

Имя	Anatoly Cybulya
Email	demo@demo.ru
Id	39
Права	admin
Адрес	Russia, kraj Krasnoyarsky, gorod Minusinsk, ulitsa Tregubenko, dom 60A, kvartira 31

Заказы пользователя:

Id заказа	Id транзакции	Сумма заказа	Email покупателя	Статус транзакции	Статус заказа	Ссылка на заказ
41	1620916240	1.00	demo@demo.ru	WAITING	Shipping	<a href="#">Ссылка на заказ</a>
55	1620916065	1.00	demo@demo.ru	WAITING	Paid up	<a href="#">Ссылка на заказ</a>

[Сохранить](#) [Удалить пользователя](#)

Рисунок 2.49 – Страница существующего пользователя

Здесь можно заметить, что все данные кроме id можно редактировать, а также, на странице есть таблица с заказами пользователя, на которые можно быстро перейти по ссылке. И, разумеется, предусмотрена кнопка удаления.

Поле «Права» предусматривает два варианта заполнения – админ или пустая строка. Если в данном поле записана строка admin – значит этому пользователю разрешен вход в админ-панель.

И последняя страница в админ-панели, заказы. Страница с заказами показана на рисунке 2.50.

## Заказы:

Id заказа	Id транзакции	Сумма заказа	Email покупателя	Статус транзакции	Статус заказа	Ссылка на заказ
41	1620916240	1.00	demo@demo.ru	WAITING	Shipping	<a href="#">Ссылка на заказ</a>
49	1621179241	1.00	test@test.ru	WAITING	Paid	<a href="#">Ссылка на заказ</a>
50	1621186434	1.00	test@test.ru	WAITING	Paid	<a href="#">Ссылка на заказ</a>
55	1620916065	1.00	demo@demo.ru	WAITING	Paid up	<a href="#">Ссылка на заказ</a>

Рисунок 2.50 – Страница с заказами

На данной странице можно увидеть все заказы, которые хранятся в базе данных. И как можно заметить на странице присутствует лишь одна кнопка – ссылка на заказ. На рисунке 2.51 показана страница существующего заказа.

demo@demo.ru
Выйти

### Заказ № 41

**Id заказа**  
41

**Id транзакции**  
1620916240

**Сумма заказа**  
1.00

**Дата оплаты**  
2021-05-13 14:30:40

**Email покупателя**  
demo@demo.ru  
Ссылка на покупателя



**Адрес доставки**  
moscow

**Статус транзакции**  
WAITING

**Статус заказа**  
Shipping

**Трек номер**  
y972834982134982904309-49-2104234213421343214

**Товары:**

Картинка	Название	Цена	Категория	Количество товара	Ссылка на товар
	Magic HightLight Roombox Steampunk Diorama Unique Decorate Miniature Interior	522.49	Roombox	1	<a href="#">Ссылка на товар</a>
	unique decorative night lights for home decor in the bedroom and dining room, Diorama	220	Roombox	6	<a href="#">Ссылка на товар</a>

[Обновить](#)

Рисунок 2.51 – Страница существующего заказа

Как можно заметить, изменить можно только статус заказа на: shipping (доставляется), paid up (оплачен) и delivered (доставлен), а также track номер. Разумеется, эти изменения будет видеть и пользователь, кому принадлежит эта учетная запись. Также на странице виден список товаров, которые заказал пользователь и их количество. Функция удаления заказа пока не предусмотрена.

Все страницы описаны, теперь необходимо показать файл, который обрабатывает данные на этих страницах (Рисунок 2.52).

```
1 <?php
2 include ('../connection.php');
3 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link))
4 function can_upload($file) { // функция загрузки изображения
5     // если имя пустое, значит файл не выбран
6     if($file['name'] == '')
7         return 'Вы не выбрали файл.';
8     /* если размер файла 0, значит его не пропустили настройки
9     сервера из-за того, что он слишком большой */
10    if($file['size'] == 0)
11        return 'Файл слишком большой.';
12    // разбиваем имя файла по точке и получаем массив
13    $getMime = explode( 'separator', '.', $file['name']);
14    // нас интересует последний элемент массива - расширение
15    $mime = strtolower(end( $array: $getMime));
16    // объявим массив допустимых расширений
17    $types = array('jpg', 'png', 'gif', 'bmp', 'jpeg');
18    // если расширение не входит в список допустимых - return
19    if(!in_array($mime, $types))
20        return 'Недопустимый тип файла.';
21    return true;
22 }
23 function make_upload($file) {
24     // формируем уникальное имя картинки: случайное число и name
25     $name = mt_rand(0, 10000) . $file['name'];
26     copy($file['tmp_name'], to: '../picture/' . $name);
27     return $name;
28 }
29 if(isset($_FILES['file'])) {
30     // проверяем, можно ли загружать изображение
31     $check = can_upload($_FILES['file']);
32     if($check === true) {
33         // загружаем изображение на сервер
34         $picture = 'picture/'.make_upload($_FILES['file']);
35         echo "<strong>Файл успешно загружен!</strong>";
36     } else {
37         // выводим сообщение об ошибке
38         $picture = $_POST['picture'];
39         echo "<strong>$check</strong>";
40     }
}
```

Рисунок 2.52 – Файл с функциями для работы с админ-панелью, лист 1

```

42  if(isset($_POST['update_item'])) { // функция обновления товара
43      $head = $_POST['head'];
44      $id = $_POST['id'];
45      $price = $_POST['price'];
46      $dimensions = $_POST['dimensions'];
47      $materials = $_POST['materials'];
48      $category_id = $_POST['category_id'];
49      $description = $_POST['description'];
50      $quantity = $_POST['quantity'];
51      mysqli_multi_query($link, query: "UPDATE items SET head='$head', price='$price', dimensions='$dimensions',
52          materials='$materials', category_id=$category_id, description='$description', quantity='$quantity',
53          picture='$picture' WHERE id=$id ");
54      header( header: 'location:product.php?id='.$id.'');
55  }
56  if(isset($_POST['delete_item'])) { // функция удаления товара
57      $id = $_POST['id'];
58      mysqli_query($link, query: "DELETE FROM items WHERE id=$id");
59      header( header: 'location:catalog.php');
60  }
61  if(isset($_POST['add_item'])) { // функция добавления товара
62      $head = $_POST['head'];
63      $price = $_POST['price'];
64      $dimensions = $_POST['dimensions'];
65      $materials = $_POST['materials'];
66      $category_id = $_POST['category_id'];
67      $description = $_POST['description'];
68      $quantity = $_POST['quantity'];
69      mysqli_query($link, query: "INSERT INTO items (head, price, dimensions, materials, category_id,
70          description, quantity, picture) VALUES ('$head', '$price', '$dimensions', '$materials',
71          $category_id, '$description', '$quantity', '$picture' )");
72      $id = mysqli_insert_id($link);
73      header( header: 'location:product.php?id='.$id.'');
74  }
75  if(isset($_POST['update_category'])) { // функция обновления категории
76      $category_id = $_POST['category_id'];
77      $category = $_POST['category'];
78      mysqli_query($link, query: "UPDATE category SET category='$category' WHERE category_id=$category_id");
79      header( header: 'location:category_update.php?id='.$category_id.'');
80  }

```

Рисунок 2.52, лист 2

```

77  if(isset($_POST['category_ajax'])){ //функция удаления категории
78      $category_ajax = $_POST['category_ajax']; //получаем id товара
79      $query = "SELECT * FROM items WHERE category_id='{$category_ajax}'";
80      $result = mysqli_query($link, $query);
81      if (mysqli_num_rows($result) != 0) {
82          $response = [ // пишем ответ для Ajax
83              "status" => 0,
84          ];
85      } else {
86          mysqli_query($link, query: "DELETE FROM category WHERE category_id='{$category_ajax}'");
87          $response = [ // пишем ответ для Ajax
88              "status" => 1,
89          ];
90      }
91      echo json_encode($response);
92  }
93  if(isset($_POST['add_category'])) { //функция добавления категории
94      $category = $_POST['add_category'];
95      $query = "SELECT * FROM category WHERE category='{$category}'";
96      $result = mysqli_query($link, $query);
97      if (mysqli_num_rows($result) != 0) {
98          $response = [ // пишем ответ для Ajax
99              "status" => 0,
100         ];
101     } else {
102         mysqli_query($link, query: "INSERT INTO category (category) VALUES ('{$category}')");
103         $response = [ // пишем ответ для Ajax
104             "status" => 1,
105         ];
106     }
107     echo json_encode($response);
108 }

```

Рисунок 2.52, лист 3

```

109  if(isset($_POST['update_user'])) { //функция обновления пользователя
110      $name = $_POST['name'];
111      $email = $_POST['email'];
112      $permissions = $_POST['permissions'];
113      $id = $_POST['id'];
114      mysqli_query($link, query: "UPDATE users SET name='{$name}', email='{$email}', permissions='{$permissions}' WHERE id='{$id}'");
115      header( header: 'location:user.php?id='.$id.'');
116  }
117  if(isset($_POST['delete_user'])) { //функция удаления пользователя
118      $id = $_POST['id'];
119      mysqli_query($link, query: "DELETE FROM users WHERE id='{$id}'");
120      header( header: 'location:users.php');
121  }
122  if(isset($_POST['add_user'])) { // функция добавления пользователя
123      $name = $_POST['name'];
124      $email = $_POST['email'];
125      $permissions = $_POST['permissions'];
126      $password = $_POST['password'];
127      $repassword = $_POST['repassword'];
128      $address = $_POST['address'];
129      $query = "SELECT * FROM users WHERE email='{$email}'";
130      $result = mysqli_query($link, $query);
131      $errors = array();
132      // Все последующие проверки, проверяют форму и выводят ошибку
133      // Проверка на совпадение паролей
134      if(trim($name) == '') {
135          $errors[] = "Введите имя";
136      }
137      // Проверка есть ли email
138      if(trim($email) == '') {
139          $errors[] = "Введите email";
140      }
141      if (!preg_match( pattern: "/[0-9a-z_]+@[0-9a-z_\.]+\.[a-z]{2,3}/i", $email)) {
142          $errors[] = 'Некорректный email';
143      }

```

Рисунок 2.52, лист 4

```

144 // Проверка есть ли пароль
145 if($password == '') {
146     $errors[] = "Введите пароль";
147 }
148 // Проверка есть ли вообще повторный пароль
149 if($repassword == '') {
150     $errors[] = "Введите повторный пароль";
151 }
152 if($repassword != $password) {
153     $errors[] = "Введенные пароли не совпадают!";
154 }
155 if($address == '') {
156     $errors[] = "Введите адрес";
157 }
158 if (mb_strlen($name) < 1 || mb_strlen($name) > 50){
159     $errors[] = "Неверная длина имени (от 1 до 50 символов)";
160 }
161 if (mb_strlen($password) < 2 || mb_strlen($password) > 20){
162     $errors[] = "Неверная длина пароля (от 2 до 20 символов)";
163 }
164 if (mysqli_num_rows($result) != 0) {
165     $errors[] = 'Пользователь с таким email уже существует!';
166 }
167 if(empty($errors)) {
168     $password = md5(md5(trim($password)));
169     // Добавление пользователя
170     mysqli_query($link, query: "INSERT INTO users (name, email, permissions, password, address)
171     VALUES ('$name', '$email', '$permissions', '$password', '$address')");
172     $id = mysqli_insert_id($link);
173     $response = [
174         "status" => "1",
175         "id" => $id,
176     ];
177     // Подтверждение что всё хорошо
178 } else {
179     $response = [
180         "status" => "0",
181         "errors" => $errors[0],
182     ];

```

Рисунок 2.52, лист 5

```

183 // Если ошибка есть, то выводить её
184 }
185 echo json_encode($response);
186 }
187 if(isset($_POST['update_order'])) { // функция обновления заказа
188     $select_order = $_POST['select_order'];
189     $track = $_POST['track'];
190     $id = $_POST['id_order'];
191     mysqli_query($link, query: "UPDATE orders SET status_transaction=$select_order, track=$track WHERE id=$id");
192     header( header: 'location:order.php?id='.$id.'');
193 }

```

Рисунок 2.52, лист 6



Как можно понять, данный файл выполняет все действия с данными в админ-панели.

Стоит также обратить внимание на поля подобные тому, что показано на рисунке 2.53.

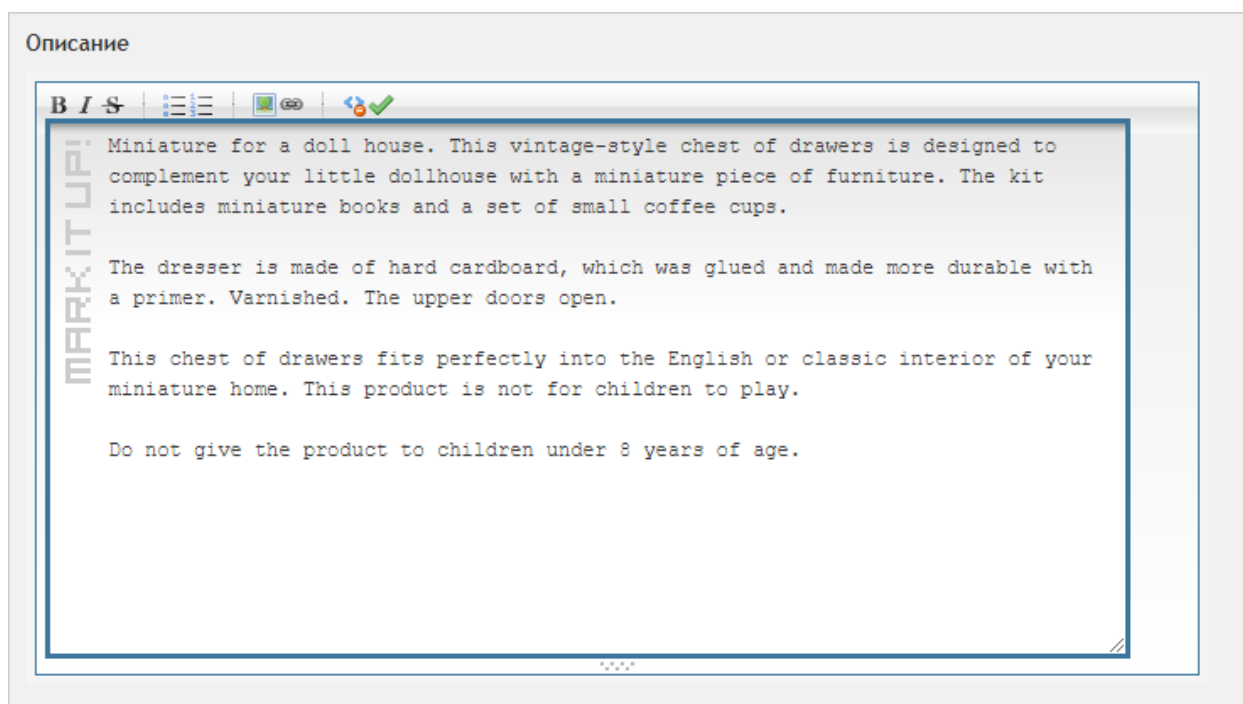


Рисунок 2.53 – Пример поля для работы с длинным текстом

Это сторонний плагин «MarkitUp!», позволяющий использовать удобный редактор для работы с bbcode. Bbcode – язык разметки, используемый для форматирования сообщений на многих электронных досках объявлений (BBS) и форумах. Для форматирования текста используются теги, подобные тегам HTML, но, в отличие от них, заключённые в квадратные скобки («[» и «]»). Перед отображением страницы движок форума производит разбор текста и преобразование его в HTML-код. Подключается он с помощью скрипта, показанного на рисунке 2.54.

```

172 <script type="text/javascript">
173 $(function() {
174     // Add markItUp! to your textarea in one line
175     // $('textarea').markItUp( { Settings }, { OptionalExtraSettings } );
176     $('#markItUp').markItUp(mySettings);
177     $('#markItUp1').markItUp(mySettings);
178     $('#markItUp2').markItUp(mySettings);
179     // You can add content from anywhere in your page
180     // $.markItUp( { Settings } );
181     $('.add').click(function() {
182         $('#markItUp').markItUp('insert',
183             {   openWith: '<opening tag>',
184                 closeWith: '<\/closing tag>',
185                 placeholder: "New content"
186             }
187         );
188         return false;
189     });
190     // And you can add/remove markItUp! whenever you want
191     // $(textarea).markItUpRemove();
192     $('.toggle').click(function() {
193         if ($("#markItUp.markItUpEditor").length === 1) {
194             $("#markItUp").markItUp('remove');
195             $("span", this).text("get markItUp! back");
196         } else {
197             $('#markItUp').markItUp(mySettings);
198             $("span", this).text("remove markItUp!");
199         }
200         return false;
201     });
202 });
203 </script>

```

Рисунок 2.54 – Скрипт для работы «MarkitUp!»

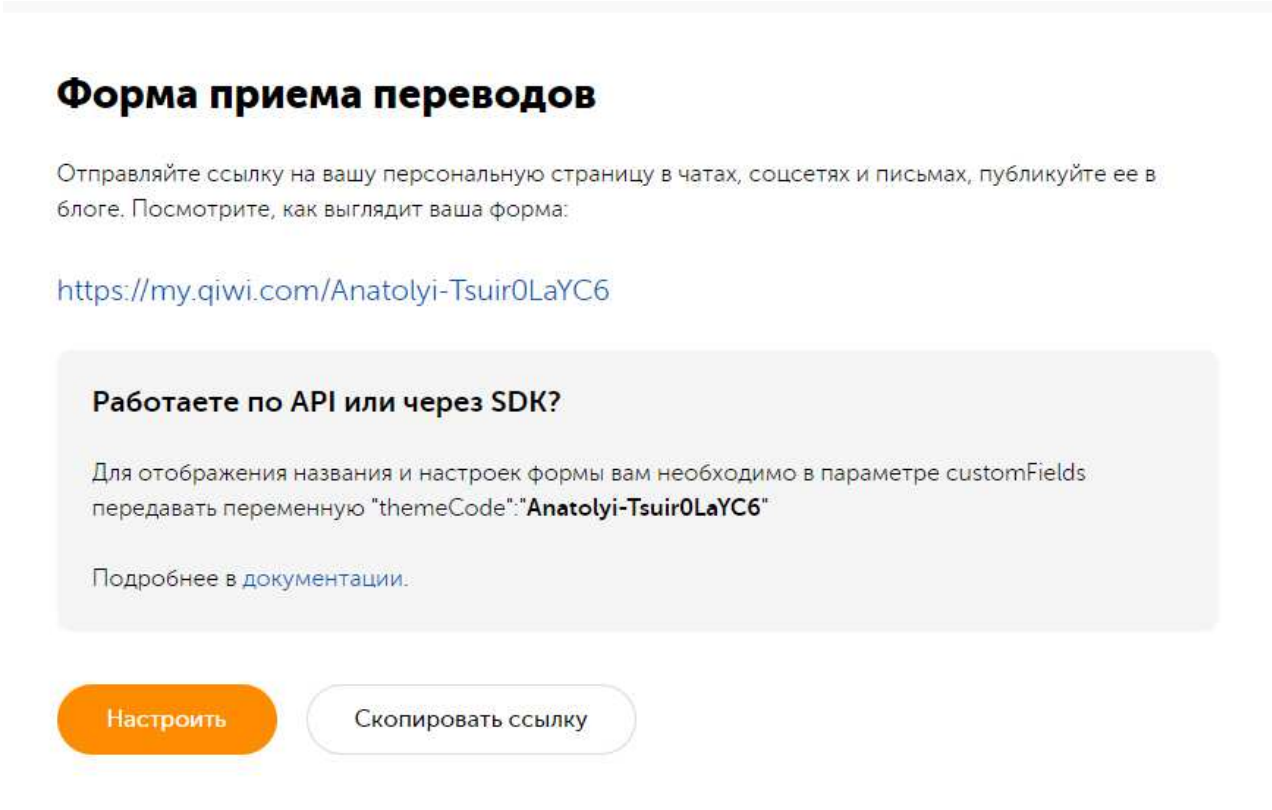
На этом работа с админ-панелью закончена.

## 2.6 Подключение системы оплаты

Оказалось, что подключение системы оплаты для интернет-магазина без истории это довольно большая проблема. И в тоже самое время, после некоторых событий в мире интернета, большое число площадок,

предоставляющих свои услуги в мире онлайн-оплаты либо закрылись, либо серьезно ужесточили правила для работы с ними, что не позволило подключить серьезную систему, такую как, ЮKassa. Как временный вариант был выбран QIWI P2P переводы. Важно заметить, что сайт не останется на данной системе, и позднее обзаведется нормальной системой оплаты.

Первым делом необходимо было зарегистрироваться на площадке QIWI, и настроить там личный кабинет, чтобы подготовиться к интеграции. После регистрации и настройки в личном кабинете будет доступна форма приема переводов (Рисунок 2.55).



## Форма приема переводов

Отправляйте ссылку на вашу персональную страницу в чатах, соцсетях и письмах, публикуйте ее в блоге. Посмотрите, как выглядит ваша форма:

<https://my.qiwi.com/Anatolyi-Tsuir0LaYC6>

**Работаете по API или через SDK?**

Для отображения названия и настроек формы вам необходимо в параметре customFields передавать переменную "themeCode": **"Anatolyi-Tsuir0LaYC6"**

Подробнее в [документации](#).

[Настроить](#) [Скопировать ссылку](#)

Рисунок 2.55 – Форма приема переводов

Также в личном кабинете доступна информация о выставленных счетах (Рисунок 2.56).

Выставление счетов   Отчет по счетам   Форма приема переводов   Виджеты   API

## Поиск счетов

Период  
05.05.2021-05.06.2021      Дата выставления ▾

Номер счета  
Номер полностью или несколько первых символов      Статус счета  
Все статусы ▾

Найти

05.05.2021-05.06.2021

статус	номер счета	дата выставления	дата оплаты	сумма счета	комментарий
выставлен	1622102091	27.05.2021 14:54:51		1 Р	Оплата заказа №162...
выставлен	1622102077	27.05.2021 14:54:38		267 Р	Оплата заказа №162...
выставлен	1622102048	27.05.2021 14:54:09		1 Р	Оплата заказа №162...
выставлен	1622102014	27.05.2021 14:53:35		1 Р	Оплата заказа №162...
выставлен	1622101999	27.05.2021 14:53:20		1 Р	Оплата заказа №162...
выставлен	1621923750	25.05.2021 13:22:32		742.49 Р	Оплата заказа №162...
выставлен	1621864956	24.05.2021 21:02:38		256.79 Р	Оплата заказа №162...

Рисунок 2.56 – Страница с выставленными счетами

На этой странице видны все выставленные системой счета, как оплаченные, так и не оплаченные, видна информация по ним, и самое главное – номер счета, по которому можно запросить информацию в интернет-магазине.

Также, в личном кабинете доступно много информации для интеграции и работы с сервисом, которые удобно расположены во вкладке «API» (Рисунок 2.57).

## API P2P-счетов

В API поддерживаются операции выставления и отмены счетов, а также проверка статуса выполнения операций. С помощью API вы можете автоматизировать работу протокола. Это значит, что вы можете встраивать форму переводов у себя на сайте или в приложении.

Подробное описание протокола и инструкции можно узнать в [документации](#). Посмотрите наши решения для популярных конструкторов сайтов (CMS):

— [wordpress woocommerce](#)

— [opencart](#)

— [1С-Битрикс](#)

Узнавайте о новых возможностях сервиса в нашем [Telegram-канале](#).

## Аутентификационные данные

Для работы вам потребуется публичный и секретный ключи. Секретный ключ нужен для обмена информацией между вашим сервисом и QIWI, а публичный позволяет выставить счет и открыть форму переводов.

Настройте серверные уведомления при создании пары ключей. И вы сможете узнавать, когда выставленные счета будут оплачены.

Подробнее — в [документации](#).

Помощь с интеграцией протокола можно получить в нашем чате [@qiwi\\_api\\_help\\_bot](#).

название пары	серверные уведомления	дата и время создания
apikey <a href="#">скопировать публичный ключ</a>		13.05.2021 18:28:31

[Создать пару ключей и настроить](#)

Рисунок 2.57 – Страница с информацией для интеграции по API

Для интеграции с сервисом необходимо установить библиотеку API PHP SDK, доступную на сайте Github (Рисунок 2.58).

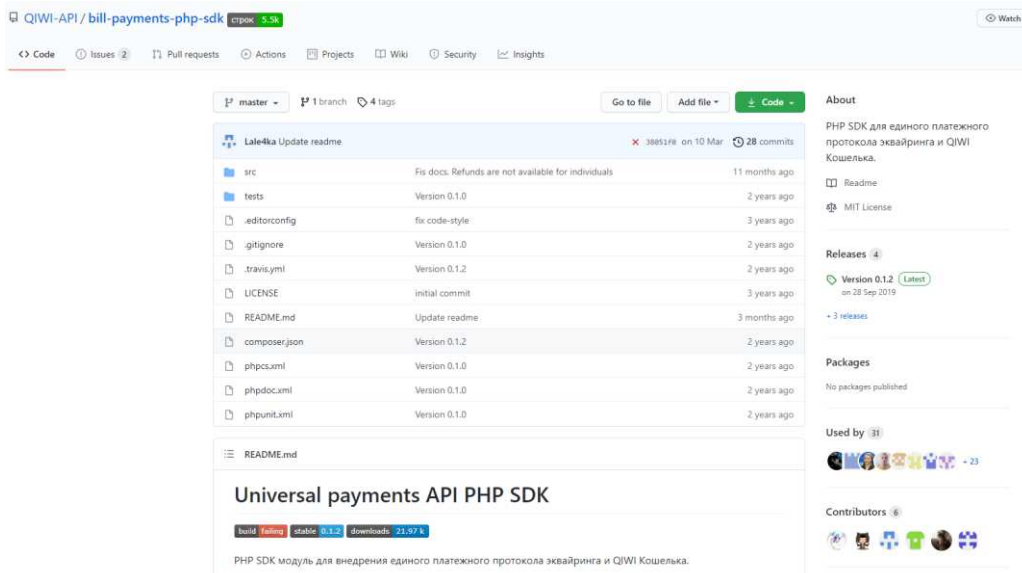


Рисунок 2.58 – Страница библиотеки на Github

Интеграцию можно совершать через composer либо просто скачать файлы и поместить их в каталог сайта, что и было сделано.

Далее необходимо внедрить систему оплаты на сайт, полный файл с необходимыми настройками изображен на рисунке 2.59.

```

1 if (empty($_GET['id_json'])) { // если на страницу зашли способом, который не предположен системой, показать ошибку 404
2     header( header: "Location: 404.php");
3 }
4
5 require_once 'connection.php'; // файл с настройками для подключения к БД
6 require_once 'vendor/autoload.php'; // файл для связывания с библиотекой QIWI
7 const SECRET_KEY =
8 $billPayments = new QIWI\Api\BillPayments(SECRET_KEY); // ключи, необходимые для работы с библиотекой, и экземпляр объекта из этой библиотеки
9 $publicKey = '48e7QUxn9T7RyVEIMVZswXIFRSBE6iyCj2gCRwF3Dnh5XrasNTx3BGP1MsyXQFNKQhvkniQG8RTVhym3iPzN4cPNDQYrQVEIAnY5p6DUmJ86uXVnHGKhhC7tsETe
10 $hash = $_GET["hash"]; // получаем хэш суммы заказа, чтобы пользователь не смог обмануть систему, и купить товар по иной цене
11 $bill = time(); // функция получения времени заказа
12 $price = $_POST["price_json"]; // получаем сумму заказа
13 // $price = 1;
14 $array_products = $_POST["col_json"]; // получаем массив с данными обо всех заказанных товарах
15 $id_user = $_POST["id_json"]; // получаем id пользователя, совершившего заказ
16 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link)); // подключаемся к БД
17 $query = "SELECT * FROM orders WHERE id = $id_user"; // запрос к базе по заказчику
18 $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link)); // ответ от базы
19 $row = mysqli_fetch_assoc($result); // извлечение результирующего ряда в виде ассоциативного массива
20 $address = $row['address']; // получаем адрес, для записи в бд
21 $customFields = [ // создаем массив для транзакции
22     'themeCode' => 'Anatolyi-Tsuir@LaYCo',
23     'products' => $array_products,
24     'id_user' => $id_user,
25     'address' => $address,
26 ];
27 $params = [ // параметры транзакции
28     'publicKey' => $publicKey,
29     'amount' => $price,
30     'comment' => "Оплата заказа №". $bill,
31     'billId' => $bill,
32     'successUrl' => 'https://lukoye.com/success.php?id=' . $bill,
33     'customFields' => $customFields,
34 ];
35 $link = $billPayments->createPaymentForm($params); // создание ссылки для оплаты товара
36 if (sha1($price) != $hash) { // если переданный хэш не совпадает с хэшем переданной суммы, отправляем пользователя на страницу с ошибкой
37     header( header: "Location: https://lukoye.com/failed.php");
38 } else { // если переданный хэш совпадает с хэшем переданной суммы направляем пользователя оплачивать товар
39     header( header: "Location: " . $link);
40 }

```

Рисунок 2.59 – Файл с настройками для работы с системой оплаты

Секретный ключ, разумеется, не предоставляется в связи с защитой информации.

В процессе поиска системы оплаты существует огромное количество тонкостей, которые не позволяли прийти к требуемому результату, было перепробовано около 10 разных площадок, занимающихся онлайн-платежами, и все они по разным причинам не подходили. Поэтому как временный вариант пришлось воспользоваться QIWI. А временный он потому, что это банальный перевод с карты на карту, а не система приема платежей, и в данный момент отсутствует возможность принимать платежи из-за границы, что в корне не соответствует политике сайта. Так или иначе, скелет работы с системой оплаты уже готов, и переезд на серьезную площадку онлайн-платежей не станет проблемой и не займет много времени.

Далее необходимо на странице с корзиной настроить передачу всех данных в скрипт оплаты. Делается это простой формой (Рисунок 2.60).

```
<form action="merchant.php?hash='.sha1($price).'" method="post" class="form_script">
  <input type="text" hidden value='.json_encode($arr).' name="col_json" class="col_json">
  <input type="text" hidden value='. $price.' name="price_json" class="price_json">
  <input type="text" hidden value="'. $row_user['id'].'" name="id_json">';
  @$email = $_COOKIE['email'];
  if ($email!= '') { // если пользователь не авторизован, отправить в форму авторизации
    echo '<input class="addToCart btn-primary btn-buy" style="margin: 0 auto" type="submit" name="m_process" value="Checkout
  } else {
    echo '<div style="text-align: center"><a data-target="#exampleModal" data-toggle="modal" style=" cursor: pointer; text-d
  }
  echo '
</form>
```

Рисунок 2.60 – Передача данных в скрипт

После того, как пользователь нажал на кнопку оплаты заказа, и успешного прохождения скрипта оплаты, он направляется на страницу оплаты товара (Рисунок 2.61).

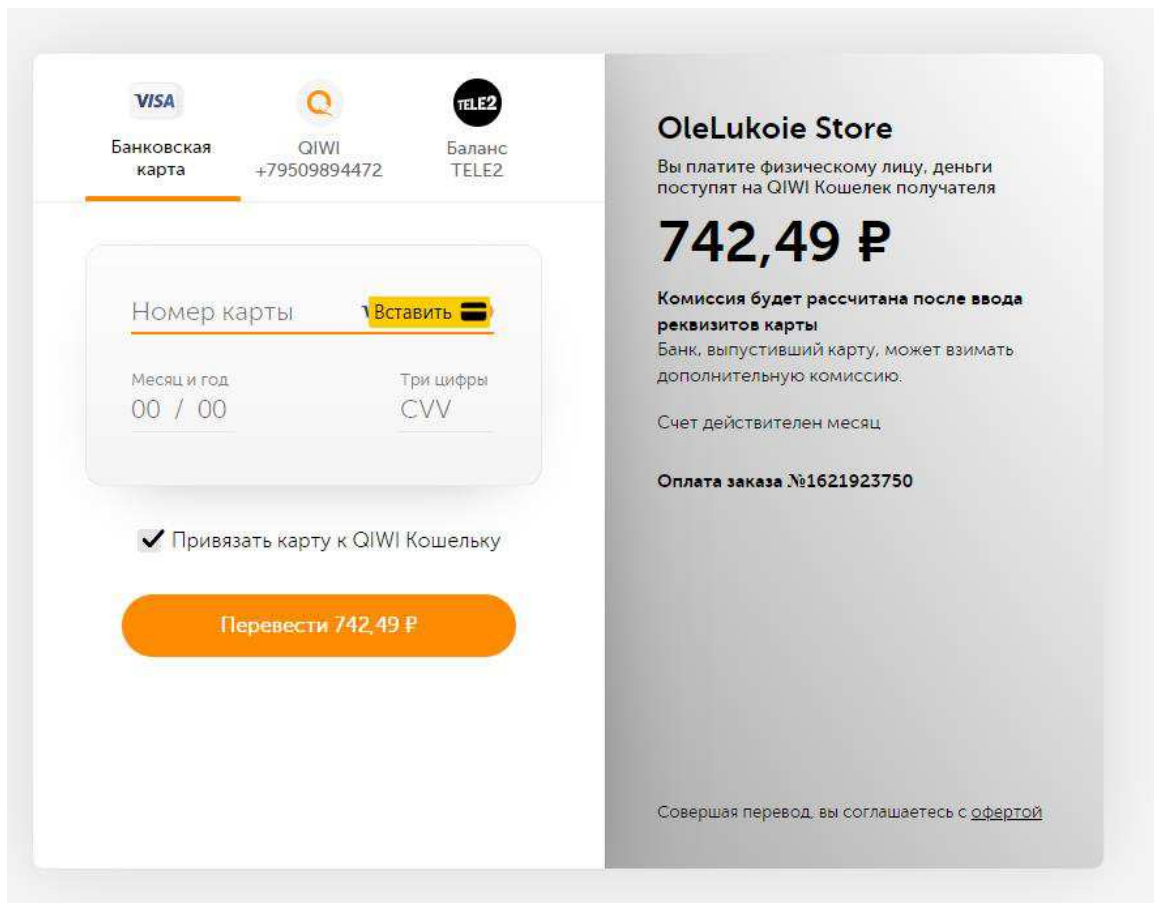


Рисунок 2.61 – Страница оплаты товара

В случае успешной оплаты пользователя перенаправляют в файл success.php, который заносит данные в базу (Рисунок 2.62).

```

1 <?php
2 if (empty($_GET['id'])){ // если на страницу зашли способом, который не предусмотрен системой, показать ошибку 404
3     header( header: "Location: 404.php");
4 }
5 require_once "./connection.php"; // файл с настройками для подключения к БД
6 require_once 'vendor/autoload.php'; // файл для связывания с библиотекой QIWI
7 const SECRET_KEY =
8 $billPayments = new QIWI\Api\BillPayments(SECRET_KEY); // ключи, необходимые для работы с библиотекой, и экземпляр объекта из этой библиотеки
9 $billId = $_GET["id"]; // получаем id транзакции
10 $link = mysqli_connect($host, $user, $password, $database) or die("Ошибка " . mysqli_error($link));
11 $query = "SELECT * FROM orders WHERE id_transaction = '$billId'";
12 $result = mysqli_query($link, $query) or die("Ошибка " . mysqli_error($link));
13 $row = mysqli_fetch_assoc($result);
14 $response = $billPayments->getBillInfo($billId); // информация о транзакции
15 $products = json_decode($response["customFields"]["products"], associative: true); // информация о заказанных товарах
16 $user_id = $response["customFields"]["id_user"]; // получение id заказчика
17 $address = $response["customFields"]["address"]; // адрес заказчика
18 $date_trans = $response["status"]["changedDateTime"]; // дата выполнения транзакции
19 $date_trans = date( format: 'Y-m-d H:i:s', strtotime($date_trans)); // форматирование даты для записи в БД
20 if($row == null) { // если в базе еще нет этого заказа
21     session_start(); // старт сессии
22     unset($_SESSION['cart']); // чистим корзину
23     mysqli_query($link, query: "INSERT INTO orders (customer_id, date_transaction, id_transaction, status_transaction, address)
24     VALUES ('$user_id', '$date_trans', '$billId', 'Paid up', '$address')"); // создание записи в базе данных
25     $id_order = mysqli_insert_id($link); // получение id последней добавленной в базу записи
26     header( header: "location: order.php?id=".$id_order.''); // перенаправление на страницу с заказами пользователя
27 } else {
28     header( header: 'location: failed.php'); // если такой заказ уже существует, отправить на страницу с ошибкой
29 }

```

Рисунок 2.62 – Файл с добавлением данных в БД



Итак, после успешной оплаты товара пользователь попадает в список своих заказов (Рисунок 2.63).

The screenshot shows the Ole Lukoie website interface. At the top, there is a navigation bar with the logo, a search bar, and links for 'Send Message', user profile, and shopping cart. Below the navigation bar, the user's name 'Anatoly Cybulya' and the heading 'List of ordered:' are displayed. A table lists two orders with columns for 'Id order', 'Order price', 'Order status', and 'Link'. The footer contains contact information, categories, social media links, and payment icons.

Id order	Order price	Order status	Link
41	1.00	WAITING	<a href="#">Link</a>
55	1.00	WAITING	<a href="#">Link</a>

Рисунок 2.63 – Список заказов пользователя

Здесь пользователь видит список всех своих заказов, даже те, которые уже доставлены.

Отсюда пользователь может зайти в нужный заказ, чтобы посмотреть его детали, для этого необходимо нажать кнопку «Link» в необходимой строчке таблицы (Рисунок 2.64).

The screenshot shows the 'Ole Lukoie' website interface. At the top, there is a search bar, a user profile icon, a shopping cart icon, and a 'Send Message' button. Below the navigation menu (Catalog, About us, Shipping, Privacy and Security, Contact us), the main heading is 'Order № 41'. The order details include: Status order: Shipping; Payment date: 2021-05-13 14:30:40; Order price: 1.00\$; Track number: CA123456789UA; Address order: moscow.

The 'Ordered goods:' section contains a table with the following data:


Image	Name	Price	Category	Count	Link
	Magic Nightlight Roombox Steampunk Diorama Unique Decorate Miniature Interior	522.49	Roombox	1	<a href="#">Link</a>
	unique decorative night lights for home decor in the bedroom and dining room, Diorama	220	Roombox	6	<a href="#">Link</a>



At the bottom of the page, there is a footer with the 'Ole Lukoie' logo, a 'Customers info' section (About us, Shipping, Privacy and Security, Contact us), a 'Categories' section (Roombox, Miniature furniture, Decor), and a 'Social' section with links to vk.com, Instagram, and Youtube.

Рисунок 2.64 – Пример заказа

Можно обратить внимание на то, что итоговая сумма в 1.00\$ не совпадает с ценами товаров, объясняется это тем, что в QIWI отсутствует возможность тестовой оплаты, и транзакции приходится делать реальные, тратя на это деньги, поэтому для этапа разработки системы оплаты, сумма заказа была выставлена статично в 1 доллар.

На этом система оплаты товара закончена, но так как был задет личный кабинет пользователя, можно также продемонстрировать страницу, на которой пользователь может редактировать свои данные (Рисунок 2.65).





0

Send Message

---

☰ Catalog
About us
Shipping
Privacy and Security
Contact us

---

## Anatoly Cybulya ✎

### Name change

Current name

Anatoly Cybulya

New name:

Enter name

Update

### Email change

Current email

demo@demo.ru

New email:

Enter email

Update

### Address change

Current address

Russia, kray Krasnoyarsky, gorod Minusinsk, ulitsa Tregubenko, dom 60A, kvartira 31

Address:

Enter address

Update

### Password change

New password:

Enter password

Repeat new password:

Repeat password

Update

### User data:

Name: Anatoly Cybulya  
 Email: demo@demo.ru  
 Address:  
 Russia, kray Krasnoyarsky, gorod Minusinsk, ulitsa Tregubenko, dom 60A, kvartira 31

Рисунок 2.65 – Редактирование информации в личном кабинете пользователя

Как видно, здесь можно изменить имя, email адрес, адрес доставки и пароль.

## 2.7 Наполнение магазина

Наполнение магазина было осуществлено на этапе верстки страницы, для правильной подгонки страниц сайта. Некоторые товары изображены на рисунке 2.66.

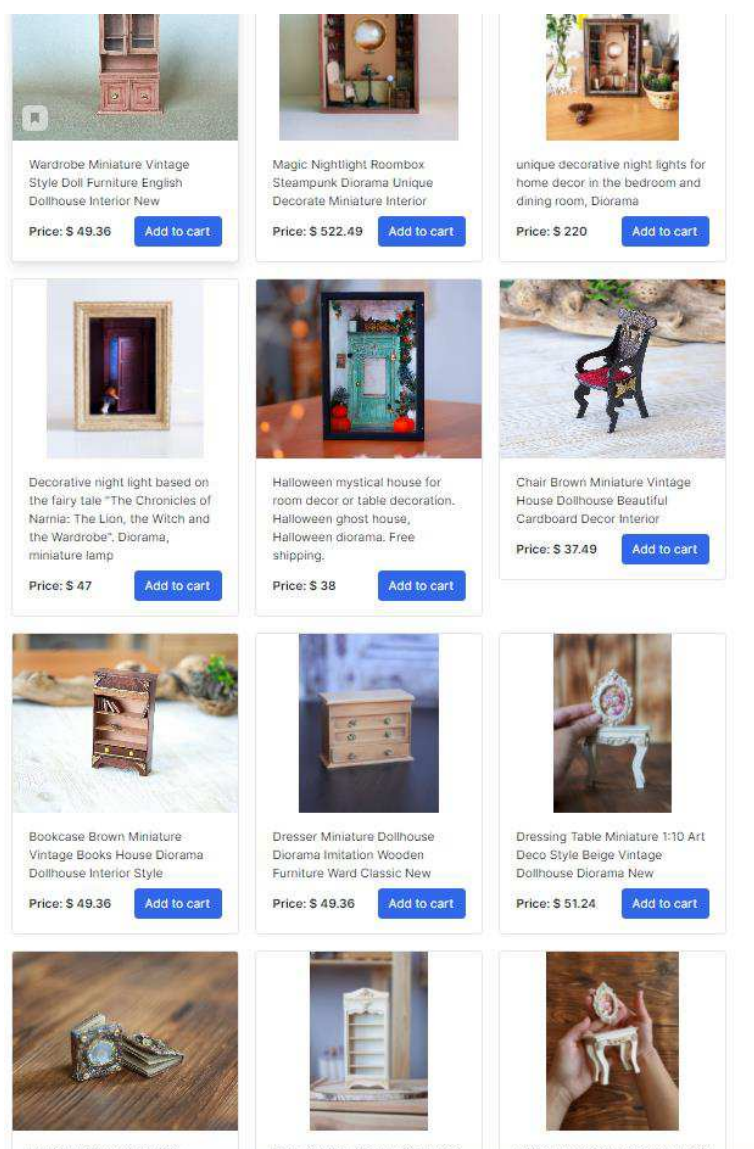


Рисунок 2.66 – Некоторые товары сервиса

## **2.8 Тестирование**

Тестирование системы было осуществлено при помощи модели «белого ящика». Модель белого ящика используется тогда, когда интерес исследователя направлен на структуру системы, на изучение внутреннего поведения системы, и когда при этом можно в силу определенных условий пренебречь взаимодействием системы с внешней средой. Иными словами, большинство багов и ошибок было отловлено на этапе создания системы, и проверки ее функциональности. Так как система не использовалась в реальных условиях и ей предстоит еще какое-то число доработок, но большая часть из них уже сделана.

## **2.9 Вывод по разделу «Проектный раздел разработки интернет-магазина «Ole Lukoie magic studio»**

На данном этапе создания системы, она готова на 99 процентов. Единственное, что остается переделать, это система оплаты, которая в данный момент не удовлетворяет требованиям системы, но так как возможность подключения к более серьезной системе оплаты отсутствовала, пришлось внедрять временный вариант, для того, что создать функционал заказа товара.

Дизайн системы полностью готов, и может только улучшатся в лучшую сторону, но и имеющийся интерфейс лаконичен, прост и легок в использовании.

База данных готова полностью, возможны изменения при интеграции другой системы оплаты, но на данном этапе в доработках не нуждается.

Серверная часть проекта аналогично с предыдущими пунктами не нуждается в доработке, до того момента, когда не будет внедрена иная система оплаты.

Хочется отметить, что работа выполнена по всем современным правилам безопасности и функционирования интернет-систем. Данные пользователя надежно защищены от попыток вторжения извне, все кнопки на сайте написаны при помощи технологии «Аjax», которая позволяет выполнять PHP скрипты, не перезагружая страницу, тем самым увеличивая позитивный отклик от взаимодействия с системой и скорость работы.

В итоге можно сказать, что система готова к работе с пользователями из России, так как оплата возможна только рублями, но в будущем, этот досадный промах будет исправлен, и сайт будет предоставлять возможность заказа из любой точки мира.

### **3 Экономический раздел разработки интернет-магазина «Ole Lukoie magic studio»**

#### **3.1 Расчет капитальных затрат**

**Расчет затрат на оборудование.** При правильной эксплуатации компьютеров, данная техника прослужит 4 года. Расчет затрат на оборудование представлен в таблице 3.1.

Таблица 3.1 – Расчет затрат на оборудование

№	Наименование	Количество	Стоимость, руб.	Срок службы, год.
1	Компьютер программиста			
1.1	Материнская плата ASRock A320M-DVS R4.0	1	3 799	4
1.2	Процессор AMD Ryzen 3 2200G BOX	1	9 499	4
1.3	Блок питания Aerocool VX PLUS 350W	1	1 499	4
1.4	240 ГБ SSD-накопитель SiliconPower Slim S55	1	2 599	4
1.5	18.5" Монитор HP 19ka	1	5 499	4

### Окончание таблицы 3.1

1.6	Клавиатура Defender Next	1	299	4
1.7	Компактная мышь проводная Oklick 115S	1	299	4
1.8	Операционная система Microsoft Windows 10 Home 32/64 bit SP2 Rus Only USB RS	1	9 480	4
	Итого:		32 973	

В затраты на оборудование будет входить компьютер программиста. Так как компьютер может использоваться 4 года, а проект разрабатывается 2 месяца, необходимо посчитать стоимость компьютера за 2 месяца.

$$32\,973 * 0,25 * 0,16 * 2 = 2\,637 \text{ рублей.}$$

$$K_{\text{свт}} = 2\,637 \text{ рублей.}$$

**Расчет заработной платы.** Для разработки данного приложения требуются 1 человек:

– Программист, со знаниями языков – Javascript, SQL, PHP, HTML, CSS, jQuery, Ajax.

Расчет зарплаты будет производиться с учетом северного коэффициента 1,3 и региональной надбавки 1,3, а также 1,302 обязательные отчисления в фонды.

Оклад программиста составит 10 000 в месяц, без учета коэффициентов и отчислений.

Система оплаты труда – сдельная. Расчет заработной платы представлен в таблице 3.2.

Таблица 3.2 – Расчет заработной платы

Наименование специалиста	Количество отработанных дней, днях	Заработная плата специалиста, руб.
Программист	60	$10\,000 * 1,3 * 1,3 * 1,302 * 2 = 44\,007$

Заработная плата специалиста составляет:

$K_{зп} = 44\,007$  рублей.

**Затраты на программное обеспечение.** Затраты на ПО представлены в таблице 3.3.

Таблица 3.3 – Затраты на ПО

Наименование	Назначение	Стоимость, руб.	Срок использования, год.
Microsoft Windows 10 Home	ПО для компьютера	9 480	4
Notepad++	Верстка	Бесплатно	Бессрочно
Xampp	Создание сайта	Бесплатно	Бессрочно
Google Chrome	Верстка	Бесплатно	Бессрочно

Стоимость операционной системы включена в стоимость компьютера, поэтому  $K_{кипс} = 0$ .

**Расчет затрат на прочие расходы.** На прочие затраты выделено 3% от затрат на проектирование. Это стандартный процент для прочих расходов, которого хватит на покрытие мелких проблем.

$K_{проч} = (44\,007 + 2\,637) * 0,03 = 1\,399$  рублей



**Расчет затрат на проектирование.** Сумма затрат на проектирование представлена в таблице 3.4.

Таблица 3.4 – Сумма затрат на проектирование

Затраты	Сумма затрат, руб.
Заработная плата специалистов	44 007
Затраты на программное обеспечение	0
Затраты на средства вычислительной техники	2 637
Затраты на прочие расходы	1 399
Итого:	48 043

$K_{пр} = 48\,043$  рублей

**Затраты на технические средства управления.** Так как сервис будет развернут на веб-хостинге, необходимо посчитать первый месяц использования хостинга, когда сайт будет переноситься.

Средний ценник за хостинг подобных сайтов = 165 рублей в месяц.

$K_{тс} = 165$  рублей

**Затраты на создание линий связи.** Для разработки приложения не потребуется создание и использование локальных сетей, поэтому  $K_{лс} = 0$  рублей.

**Затраты на программные средства.** Для работоспособности системы не требуются какие-либо платные ПО.

$K_{ло} = 0$  рублей.

**Затраты на формирование информационной базы.** Информационная база будет заполняться в период разработки и оплата создания этой базы включена в зарплату программиста.

$K_{\text{ио}} = 0$  рублей.

**Затраты на опытную эксплуатацию.** Опытная эксплуатация включена в работу программиста, и входит в его заработную плату.

$K_{\text{оэ}} = 0$  рублей.

**Расчет капитальных затрат.** Расчет всех капитальных затрат представлен в таблице 3.5.

Таблица 3.5 – Расчет капитальных затрат

Вид затрат	Затраты	Сумма затрат, рублей
Затраты на проектирование	Заработная плата специалистов	44 007
	Затраты на программное обеспечение	0
	Затраты на оборудование	2 637
	Затраты на прочие расходы	1 399
Капитальные затраты	Затраты на технические средства управления	165
	Затраты на создание линий связи	0
	Затраты на программные средства	0
	Затраты на формирование информационной базы	0

### Окончание таблицы 3.5

	Затраты на опытную эксплуатацию	0
Итого:		48 208

$$K = 48\,208 \text{ рублей}$$

### 3.2 Эксплуатационные затраты

Районный и северный коэффициенты в Республике Хакасия в сумме равны 60%.

Для поддержания работоспособности системы необходим один администратор.

Так как клиентов у магазина не много, и от администратора не требуется постоянного наблюдения за сайтом, его заработная плата составит 1 000 рублей в месяц с учетом всех коэффициентов и отчислений. Обуславливается эта сумма ожиданием небольшого числа клиентов на старте магазина, и соответственно малым объемом работы администратора.

Таким образом, чтобы вычислить заработную плату за год необходимо:  
 $1\,000 * 12 = 12\,000$  рублей.

$$C_{\text{зп}} = 12\,000 \text{ рублей.}$$

**Затраты на амортизационные отчисления.** Для работы системы администратору необходим компьютер. Компьютер администратора представлен в таблице 3.6.

Таблица 3.6 – Компьютер администратора

№	Наименование	Количество	Стоимость, руб.	Срок службы, год.
1	Компьютер администратора			
	15.6" Ноутбук Lenovo IdeaPad S145-15API	1	24 999	4
	Итого:		24 999	

Стоимость амортизации за год составит  $A_{\text{год}} = 24\,999/4 = 6\,249$  рублей.

**Затраты на использование глобальных сетей.** Для работы администратора необходим интернет. В среднем, месяц интернета стоит 249 рублей. Годовая стоимость интернета составит  $12 * 249 = 2\,988$  рублей.

Также потребуется посчитать сумму хостинга веб-сайта за год.

$165 * 12 = 1980$  рублей.

$C_{\text{ис}} = 2\,988 + 1980 = 4\,968$  рублей.

**Затраты на техническое обслуживание.** Затраты на обслуживание и ремонт техники составляют 1 000 рублей.

$C_{\text{то}} = 1\,000$  рублей.

**Затраты на носители информации.** Так как система будет развернута на веб-хостинге, который включает в себя носитель информации необходимого для работы системы объема, покупка дополнительных носителей не требуется.

$C_{\text{ни}} = 0$  рублей.

**Прочие затраты.** Прочие затраты равны 3% сумме всех затрат. Сумма всех затрат равна:

$$12\,000 + 6\,249 + 1\,000 + 4\,968 = 24\,217 \text{ рублей.}$$

3% от этой суммы равны: 726 рублей.

$$C_{\text{проч}} = 726 \text{ рублей.}$$

**Расчет эксплуатационных затрат.** Список эксплуатационных затрат показан в таблице 3.7.

Таблица 3.7 – Список эксплуатационных затрат

Состав затрат	Планируемая сумма расходов в год, руб.
Затраты на заработную плату персонала	12 000
Затраты на амортизационные отчисления	6 249
Затраты на использование глобальных сетей	4 968
Затраты на техническое обслуживание	1 000
Затраты на носители информации	0
Прочие затраты	726
Итого	24 943

**Расчет рисков реализации.** В начале проекта невозможно знать все, что случится в процессе его реализации, поэтому все инвестиции сопряжены с риском. Для расчета рисков необходимо проанализировать создание интернет-магазина по двум основным группам рисков:

1. Риски инвестирования в разработку проекта.

## 2. Риски внедрения проекта.

Риски проекта представлены в таблице 3.8.

Таблица 3.8 – Риски проекта

№	Группы рисков	Перечень рисков проекта	Уровень влияния риска на проект	Вероятность риска	Возможность предотвращения или снижения риска
1.	Риски инвестирования в разработку проекта	Риск соответствия	средний уровень	низкая	Составление четкого технического задания
		Увеличение стоимости разработки проекта	средний уровень	низкая	Заключение договора с фиксированными ценами
		Увеличение стоимости функционирования системы	низкий уровень	низкая	Поиск платформы для развертывания системы на этапе проектирования
		Риск связанный с используемыми ПО	низкий уровень	низкая	Выбор бесплатного ПО на стадии проектирования
2.	Риски внедрения проекта	Нехватка времени на внедрение системы	Высокий уровень	низкая	Выбор оптимального количества времени на внедрение на стадии планирования
		«Человеческий фактор»	Средний уровень	низкая	Подбор квалифицированного и опытного персонала

### Окончание таблицы 3.8

		Повышение цены на хостинг	Низкий уровень	низкая	Изучение рынка интернет-хостингов и подбор наиболее оптимального на стадии проектирования
--	--	---------------------------	----------------	--------	---

### 3.3 Мероприятия по снижению вероятности рисков, имеющих наивысшую оценку

При анализе рисков реализации проекта был найден один важный риск, требующий детального разьяснения.

Риск денежных потоков обуславливается более низкой прибылью на старте сервиса, чем ожидалось. Для того, чтобы уменьшить этот риск, нужно проанализировать сколько прибыли приносила интернет-витрина, и отталкиваясь от этих цифр делать прогноз. Стоит также учитывать, что на старте интернет-магазин может приносить как меньше средств, нежели интернет-витрина, так и больше, здесь все будет зависеть от маркетинга, либо его отсутствия.

**Расчет косвенных расходов.** Косвенные расходы – это затраты, связанные с производством и реализацией продукции (работ, услуг), которые нельзя напрямую отнести к изготовлению продукции (работ, услуг) и можно учесть в расходах в том периоде, когда они понесены.

Все затраты, которые не отнесены в учетной политике к прямым расходам и не являются внереализационными расходами, признаются косвенными расходами.

Косвенные расходы первой группы (IC 1) равны нулю, так как разработка типичная.

Косвенные расходы второй группы (IC 2) будут включать в себя затраты на маркетинг, и так как магазин занимается продажей не самых

популярных вещей, эти затраты не должны быть слишком большими, чтобы не уйти в минус. Так как реклама планируется в поисковом сервисе, в будущем можно будет отслеживать доходы от рекламы и принимать решения о дальнейшей покупке. На данном этапе принято решение ограничиться 3 000 рублей в месяц.

$$IC_2 = 3\,000 \text{ рублей.}$$

**Расчет по методике TCO.** Совокупная стоимость владения (TCO) – это современный метод оценки эффективности приобретения, владения и эксплуатации оборудования и комплектующих.

Для расчета затрат на реализацию проекта используется следующая формула TCO:

$$TCO = DE + IC_1 + IC_2, \tag{1}$$

где DE (directexpenses) – прямые расходы;

IC<sub>1</sub> (indirectcosts) – косвенные расходы первой группы;

IC<sub>2</sub> (indirectcosts) – косвенные расходы второй группы.

$$DE = DE_1 + DE_2 + DE_3 + DE_4 + DE_5 + DE_6 + DE_7 + DE_8, \tag{2}$$

где DE<sub>1</sub> – капитальные затраты;

DE<sub>2</sub> – расходы на управление информационными технологиями;

DE<sub>3</sub> – расходы на техническую поддержку автоматизированного обеспечения и программного обеспечения;

DE<sub>4</sub> – расходы на разработку прикладного программного обеспечения внутренними силами;

DE<sub>5</sub> – расходы на аутсорсинг;

DE<sub>6</sub> – командировочные расходы;



DE<sub>7</sub> – расходы на услуги связи;

DE<sub>8</sub> – другие группы расходов.

DE<sub>1</sub>=48 208 рублей.

DE<sub>2</sub>=12 000 рубль.

DE<sub>3</sub>=C<sub>то</sub>+C<sub>ао</sub>= 6 249 рублей + 1 000 рублей = 7 249 рублей.

DE<sub>4</sub>= 0 рублей.

DE<sub>5</sub>= 0 рублей, так как при разработке и внедрения приложения не использовалась помощь сторонних специалистов.

DE<sub>6</sub>= 0 рублей, так как для разработки приложения не требуется командировочных расходов.

DE<sub>7</sub>= 4 968 рублей.

DE<sub>8</sub>= C<sub>проч</sub>+ C<sub>ни</sub>= 726 рублей + 0 рублей = 726 рублей.

TCO = 48 043 + 12 000 + 7 249 + 4 968 + 726 + 3 000 \* 12 = 108 986 рублей.

### **3.4 Определение доходов и экономической эффективности реализации проекта разработка онлайн-магазина «Ole Lukoie magic studio»**

Для определения доходов и экономической эффективности проекта, необходимо провести анализ данного рынка.

Спрос. Спрос на интернет-магазины высок как никогда, но тематика сайта является не самой популярной среди массового пользователя, поэтому спрос на данные услуги можно оценить, как средний.

Новизна. На рынке присутствуют локальные магазины, продающие сувениры, но их не так много, и они не пользуются большой популярностью, поэтому фактор новизны можно считать высоким.

Наличие конкуренции. В настолько локальных магазинах гораздо важнее сам товар, а не качество и масштаб интернет-магазина, поэтому конкурентность магазина будет определяться качеством товара и его разнообразием.

Актуальность предложения. Актуальность не является сильной стороной проекта, так как предложение не новое и не прорывное, но магазин можно гибко продвигать рекламными методами, тем самым повышая интерес к нему среди целевой аудитории.

Реклама. Реклама интернет-магазина сегодня это очень гибкий и понятный инструмент, рекламу можно покупать у сайтов поисковиков, также можно закупать рекламу на других интернет-площадках, коллаборировать с другими сервисами, и еще куча разных возможностей. Так что продвижение такого интернет-магазина не является проблемой.

Потребительские свойства. У сувенирных изделий можно выделить следующие свойства:

- Эргономичные свойства: товары не занимают много места, являются эргономичными, создают ощущение удобства, комфортности, также удовлетворяют психологические потребности пользователя, даря ему чувство красоты окружающей среды.

- Душевный комфорт: товары дают состояние внутреннего спокойствия, отсутствие разлада с собой и окружающим миром.

- Функциональные свойства: изделия можно использовать как украшения дома, офиса, любого рабочего места, разнообразие товаров позволяет подобрать наиболее подходящий к интерьеру продукт. Сувениры могут вызвать интерес у коллекционеров, у людей, любящих подобные товары, и у простых людей, желающих устроить окружающий их быт. Помимо простых сувениров, есть и те, что имеют вполне понятный функционал, такие как кружки, ложки и т.д.

– Назначение: назначение сувениров каждый человек определяет для себя сам. Основное назначение – это элемент декора, либо в случае с кружкой, прямое использование.

– Безопасность: Изделия выполнены из безопасных для человека материалов, и не требуют использование химически сложных элементов. Качество товара предполагает использование только хорошего сырья для изготовления сувениров.

– Целостность композиции: Сувениры имеют внешне похожие свойства, такие как размер, цвета, материалы, что дает возможность создания целых композиций из различных сувениров.

Далее необходимо проанализировать рынок продуктов-аналогов, определить возможную стоимость создания интернет-магазина. Для этого нужно найти компании, занимающиеся созданием интернет-магазинов, так как для таких компаний не важно какими товарами торгует магазин, а важен лишь масштаб.

На самом деле точно сказать стоимость создания именно такого интернет-магазина довольно сложно, во-первых, мало студий в городе Абакан занимаются созданием таких магазинов, а во-вторых, сильно отличается цена.

Для примера одна из лучших студий Абакана берет от 500 тысяч до 4 миллионов (Рисунок 3.1).

M A G N E E X

Портфолио

Контакты

Подать заявку

## Отраслевые решения

Разработка интернет-магазина

от 500 тыс. руб до 4 млн. руб

Рисунок 3.1 – Стоимость разработки интернет-магазина в студии «Magneex»

Но тут стоит сказать, что магазин, разработанный этой студией, будет существенно более совершенный, чем тот, что был разработан, и настолько локальному магазину не нужен весь тот функционал, который предполагает большие розничные продажи, поэтому этот вариант не совсем подходит.

Следующая веб-студия Абакана берет меньше, и даже указывает сроки (Рисунок 3.2).

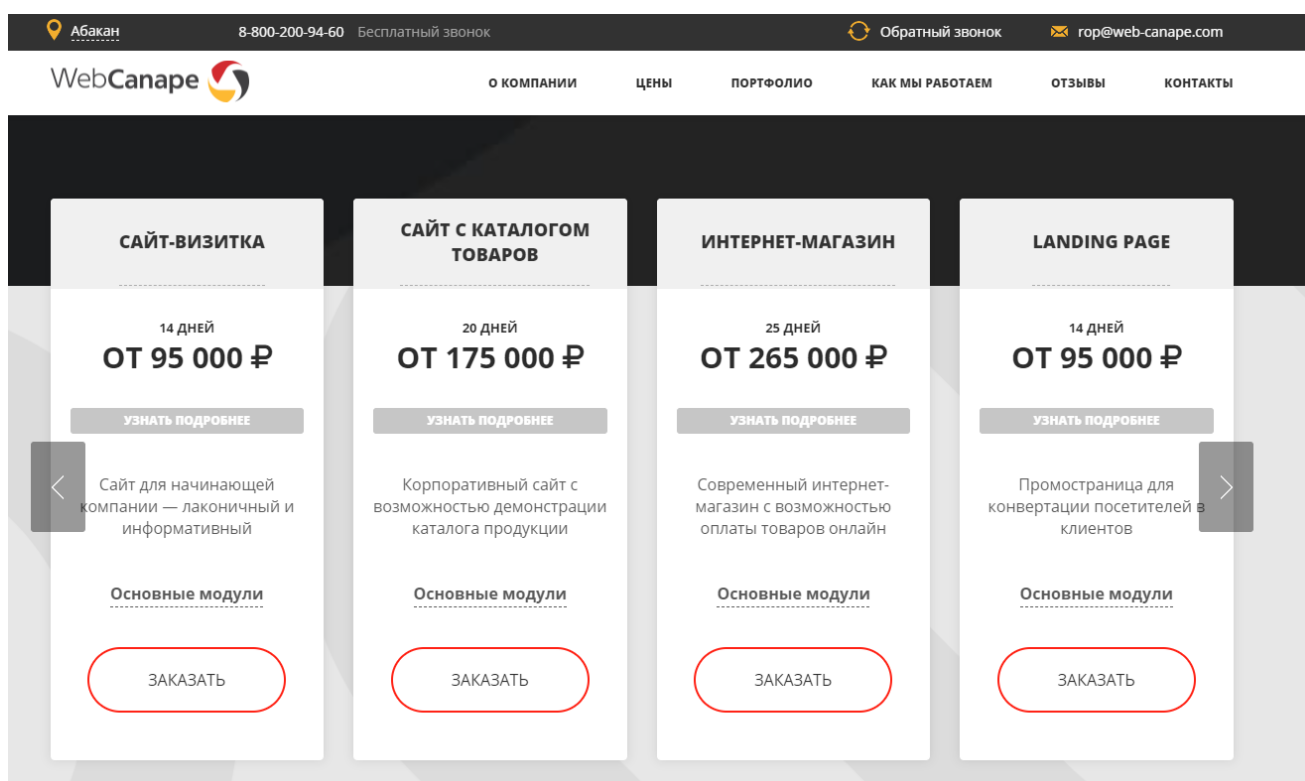


Рисунок 3.2 – Стоимость разработки сайта в студии «WebCanape»

Стоимость все равно высокая, чтобы проанализировать, возможно ли создание магазина за более скромную сумму, потребуется выйти за пределы города Абакан. Например, красноярская студия «Cyber Nevod». (Рисунок 3.3)

## Что входит в услугу:

Качественный, удобный и отзывчивый [дизайн сайта](#)

Удобная административная панель сайта

Удобный и функциональный каталог товаров

Набор стандартных пользовательских решений по функционалу (форма обратной связи, галерея изображений, фильтры, сортировка товаров и пр.)

Гарантия того, что сайт будет проиндексирован Яндексом и Google в течение 14 дней с момента завершения разработки

Использование стандартных платных или бесплатных движков сайта (CMS): Joomla, Bitrix, OpenCart

Персональный менеджер на всем этапе от технического задания до разработки

[Сопровождение сайта](#) в течение 3 месяцев с момента завершения разработки

Стоимость:

от **89 900 Р**

Срок разработки

**12-15 рабочих дней**

Закажите разработку интернет-магазина до 30.04.2021 и в подарок вы получите:

- домен в зоне ru или РФ на 1 год
- бесплатный SSL сертификат на 1 год
- настройку почты для домена (до 20 пользователей)
- подключение настройку счётчиков Google Analytics и Яндекс.Метрика для мониторинга и анализа трафика и автоматического расчета конверсии
- подключение и настройку онлайн-чата с пользователями (Bitrix, JivoSite или иной на ваш выбор)

ОТПРАВИТЬ ЗАЯВКУ

### Рисунок 3.3 – Стоимость создания интернет-магазина в «Cyber Nevod»

В глаза сразу бросается более приемлемая цена, и полный набор всех необходимых инструментов.

Исходя из увиденных цен, можно понять, что минимальная стоимость создания интернет-магазина под ключ – 90 тысяч рублей.

Теперь необходимо установить стоимость создания интернет-магазина «Ole Lukoie magic studio».

Исходя из увиденных примеров, можно с уверенностью ставить стоимость в 60 000 рублей, так как разрабатываемая система имеет слегка упрощенный функционал в сравнении с предложением студии «Cyber Nevod» за 90 000 рублей.

Далее необходимо установить стоимость товара по затратному методу.

Затратный метод – цена устанавливается исходя из рассчитанной стоимости капитальных затрат + минимальной нормой прибыли.

Минимальная норма прибыли в it-сфере равна 20 процентам от суммы разработки, то есть  $48\,208 * 0,2 = 9\,641$  рублей.

Таким образом получается:  $48\,208 + 9\,641 = 57\,849$  рублей.

Доходный метод – цена устанавливается на основе возможного дохода.

Сейчас компания продает 1-2 игрушки в месяц, после перехода на интернет-магазин планируется выйти на 5 игрушек в месяц как минимум, со средней ценой в 25 долларов, что равняется 9 547 рублей прибыли в месяц.

Таким образом:  $9\,547 * 12 = 114\,564$  рублей.

Далее нужно определить источники получения дохода. На старте проекта подразумевается лишь один источник – продажа товаров. Рекламные блоки на сайте при старте проекта негативно скажутся на пользовательском опыте, а также рекламные блоки слабо интегрируются в минималистичный дизайн магазина.

В итоге необходимо оценить экономическую эффективность реализации проекта. Для этого необходимо определить период окупаемости.

Так как возможный доход оценен в 9 547 рублей, это примерно 5 покупателей в месяц, период окупаемости составит: 6 месяцев.

Так как стоимость разработки магазина существенно дешевле самого дешевого предложения на рынке, и очень быстро окупится, проект определенно точно экономически эффективен.

### **3.5 Вывод по разделу «Экономический раздел разработки интернет-магазина «Ole Lukoie magic studio»**

В данном разделе были посчитаны капитальные и эксплуатационные затраты на создание проекта. Следующим этапом был произведен расчет рисков реализации проекта, а также описаны мероприятия по снижению вероятности рисков имеющих наивысшую оценку, после чего были посчитаны косвенные расходы и наконец подведена итоговая цена проекта. Она составила 108 986 рублей.

Следующим этапом было определение доходов и экономической эффективности реализации проекта, в ходе которого был проведен анализ рынка, выявлены сильные стороны проекта и проведен анализ конкурентов. По итогу этого этапа было установлено, что реализация проекта полностью

оправдана, так как стоимость создания подобных проектов у конкурентов стоит заметно дороже, и не всегда обладает тем набором характеристик, которые нужны проекту.

Подводя итог можно точно сказать, что создание проекта экономически эффективно.

## ЗАКЛЮЧЕНИЕ

В ходе работы были выполнены все цели и задачи. Была изучена теория проектирования информационных систем, были получены навыки анализа и автоматизации бизнес-процессов, проектирования, разработки и определения экономической эффективности проекта. Сначала была изучена предметная область проекта, были изучены бизнес-процессы, которые необходимо автоматизировать, выявлены необходимые для проекта атрибуты. Далее были поставлены цели создания системы, после чего было начато планирование разработки.

В планировании разработки максимально точно описывались инструменты создания и взаимодействия с системой, были созданы диаграммы взаимодействия различных данных.

Следующим этапом стала разработка системы, все модули были реализованы в полном объеме, и единственное место, нуждающееся в изменениях – это система оплаты, которая сейчас предоставляет возможность оплаты только рублями.

Завершающим этапом стало определение экономической эффективности проекта. Был проанализирован рынок компаний, которые предлагают услуги создания подобных систем, после чего был сделан вывод, что стоимость системы является приемлемой, и разработка системы экономически эффективна.

Результатом выпускной квалификационной работы стала полностью готовая к работе система. Однако это совсем не означает, что она не может быть улучшена, система на данном этапе требует к себе внимания и работе над выявлением новых ошибок и багов, а также нуждается в смене системы оплаты.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Академия SEO «seo-akademiya». Статья: Как оформить корзину на сайте интернет-магазина [Электронный ресурс]. – Режим доступа: <https://seo-akademiya.com/baza-znaniy/prodvizhenie-internet-magazina/kak-oformit-korzinu-na-sajte-internet-magazina>.
2. Блог про интернет-маркетинг «БЛОГ. CYBERMARKETING». Статья: Как правильно оформить корзину интернет-магазина [Электронный ресурс]. – Режим доступа: <https://blog.cybermarketing.ru/kak-pravilno-oformit-korzinu-internet-magazina>.
3. Википедия. Статья: Информационная система [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org>.
4. Интернет-магазин «Другие подарки» [Электронный ресурс]. – Режим доступа: <https://drugiepodarki.com>.
5. Интернет-магазин «Россувенир» [Электронный ресурс]. – Режим доступа: <https://rossouvenir.ru>.
6. Коллективный блог «Хабр». Статья: Надёжная авторизация для веб-сервиса за один вечер [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/323714>.
7. Онлайн-сервис для создания схем и визуализации данных «Lucidchart». Статья: Что такое ER-диаграмма и как ее создать? [Электронный ресурс]. – Режим доступа: <https://www.lucidchart.com/pages/ru/erd-диаграмма#top>.
8. Портал онлайн-образования «otus». Статья: Как сделать авторизацию на PHP? Пишем авторизацию пользователя [Электронный ресурс]. – Режим доступа: <https://otus.ru/nest/post/882>.
9. Руководство по PHP «PHP.net» [Электронный ресурс]. – Режим доступа: [www.php.net](http://www.php.net).

10. Справочник по bootstrap 4 «bootstrap-4» [Электронный ресурс]. – Режим доступа: <https://bootstrap-4.ru>.

11. Социальная сеть «golosi.io». Статья: PHP – Урок 14. Регистрация пользователей на сайте [Электронный ресурс]. – Режим доступа: <https://golosi.io/~gaxpx1tqntf/php-urok-14-registraciya-polzovatelei-na-saite>.

12. СТО 4.2-07-2014 Стандарт организации «Общие требования к построению, изложению и оформлению документов учебной деятельности» – Введ. 2014. – Красноярск: ИПК СФУ, 2014. – 61 с.

13. Учебные материалы для студентов «Studme.org». Статья: Модели жизненного цикла информационных систем [Электронный ресурс]. – Режим доступа: [https://studme.org/226156/informatika/modeli\\_zhiznennogo\\_tsikla\\_informatsionnyh\\_sistem](https://studme.org/226156/informatika/modeli_zhiznennogo_tsikla_informatsionnyh_sistem).

14. Хранилище документации «MDN Web Docs». Статья: Повторное введение в JavaScript (JS учебник) [Электронный ресурс]. – Режим доступа: [https://developer.mozilla.org/ru/docs/Web/JavaScript/A\\_re-introduction\\_to\\_JavaScript](https://developer.mozilla.org/ru/docs/Web/JavaScript/A_re-introduction_to_JavaScript).

15. Центр информационных технологий [Электронный ресурс]. – Режим доступа: <https://cintrh.ru>.

## ПРИЛОЖЕНИЕ А

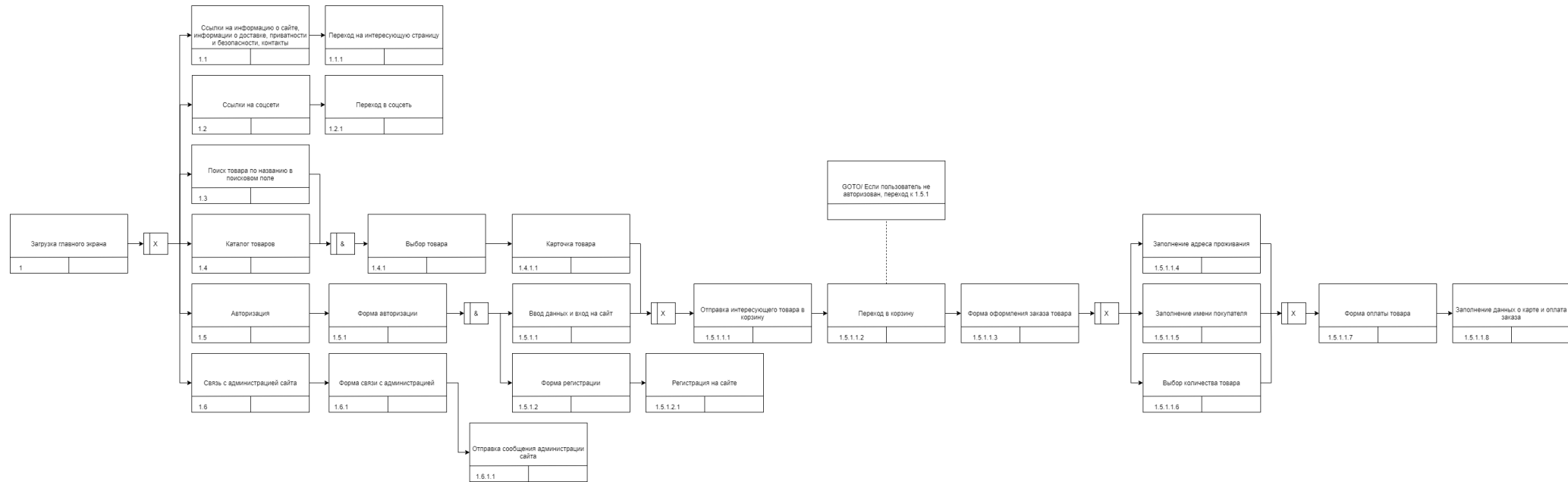


Рисунок А.1 – IDEF3 информационной системы

Выпускная квалификационная работа выполнена мной самостоятельно.  
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 15 наименований.

Один экземпляр сдан на кафедру.

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

\_\_\_\_\_ Цыбуля Анатоли Юрьевич  
подпись



Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО  
«Сибирский федеральный университет»

Кафедра прикладной информатики, математики и естественно-научных  
дисциплин

УТВЕРЖДАЮ

Заведующий кафедрой

  
Е.Н. Скуратенко

подпись

« 08 » апреля 2021 г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Студенту Цыбуля Анатолию Юрьевичу

Группа ХБ 17-03

Направление 09.03.03 Прикладная информатика

Тема выпускной квалификационной работы: Разработка интернет-магазина «Ole Lukoie magic studio»

Утверждена приказом по институту № 222 от 08.04.2021 г.

Руководитель ВКР: А.Н. Таскин, доцент, к.ф.-м.н., ХТИ – филиал СФУ

Исходные данные для ВКР: заказ ГАУ РХ «ЦИНТ Хакасии».

Перечень разделов ВКР:

1. Анализ предметной области. Выбор средств проектных решений.
2. Описание разработки системы.
3. Расчёт затрат и оценка экономической эффективности реализации интернет-магазина «Ole Lukoie magic studio».

Перечень графического материала: нет

Руководитель ВКР

  
подпись

А.Н. Таскин

Задание принял к исполнению

  
подпись

А.Ю. Цыбуля

«08» апреля 2021 г.

## SUMMARY

The theme of the graduation thesis is «Development of Online Store “Ole Lukoie Magic Studio” contains 100 pages of a text document, 79 figures, 10 tables, 15 reference items, 1 appendix.

DATA BASE MANAGEMENT SYSTEM (DBMS) of MYSQL, INQUIRY, DATABASE, TABLES, STANDARTIZATION, GRADUATION THESIS, SOFTWARE, DATA, IT SYSTEM, ONLINE STORE, ONLINE STOREFRONT, DIAGRAM.

The purpose of this graduation thesis: to develop online store “Ole Lukoie Magic Studio”.

Objectives to be determined:

- to analyze the subject area;
- to select means of the design solution;
- to develop online store “Ole Lukoie Magic Studio”;
- to calculate economic efficiency of the implementation of online store “Ole Lukoie Magic Studio”.

Results: the IT system of «Online Store “Ole Lukoie Magic Studio” » has been developed.

English language supervisor:



---

N.V. Chezybaeva



Выпускная квалификационная работа выполнена мной самостоятельно.  
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография 15 наименований.

Один экземпляр сдан на кафедру.

« 21 » 06 2021 г.

А.И. Цыбуля  
подпись

Цыбуля Анатолий Юрьевич