

Министерство цифрового развития, связи и массовых коммуникаций РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Кафедра ПМиК
Допустить к защите
зав. кафедрой: д.т.н.

_____ Нечта И.В.

ВЫПУСКНАЯ
КВАЛИФИКАЦИОННАЯ РАБОТА
БАКАЛАВРА
РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА АВТОРСКИХ
ВЕЩЕЙ

Пояснительная записка

| | |
|--------------|-------------------------------------|
| Студент | _____ Пещеров В.А. _____ /...../ |
| Институт | _____ ИВТ _____ Группа _____ ИП-813 |
| Руководитель | _____ Перцев И.В. _____ // |

Новосибирск 2022г.

**Министерство цифрового развития, связи и массовых коммуникаций РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)**

Форма утверждена
научно-методическим советом
ФГОБУ ВПО «СибГУТИ»
протокол № 2 от 04.03. 2014 г.

КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА**

СТУДЕНТУ Пещерову Вячеславу
Александровичу

ГРУППЫ ИП-813

УТВЕРЖДАЮ

«_____» _____

зав. кафедрой ПМиК

_____/_____

Новосибирск 2022г.

1. Тема выпускной квалификационной работы бакалавра

Разработка интернет-магазина авторских вещей

утверждена приказом СибГУТИ от «21» января 2022 г. № 4/54о-22

2.Срок сдачи студентом законченной работы « 11 » июня 2022 г.

3.Исходные данные к работе

1. Дэвид Скляр. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов.: Пер. с англ. - СПб.: ООО "Альфа-книга", 2017. - 464 с. : ил. - Парал. тит. англ.

2. Руководство: Smarty [Электронный ресурс] // Smarty – URL: <https://www.smarty.net/docsv2/ru/>

3. HTML начало [Электронный ресурс] // wm-school – URL: https://wm-school.ru/html/html_start.html

4. Создание PHP + MySql сайта [Электронный ресурс] // Хабр – URL: <https://habr.com/ru/sandbox/14484/>

5. Этан Браун. Изучаем JavaScript: руководство по созданию современных веб-сайтов, 3-е изд.: Пер. с англ. - СПб.: ООО "Альфа-книга"; 2017. - 368 с. : ил. - Парал. тит.

6. Руководство: Русская документация по API jQuery [Электронный ресурс] // jquery-docs – URL: <https://jquery-docs.ru/>

7. Основы CSS [Электронный ресурс] // html5book.ru – URL: <https://html5book.ru/osnovy-css/>

| 4.Содержание пояснительной записки (перечень подлежащих разработке вопросов) | Сроки выполнения по разделам |
|---|---|
| Введение | 31.01.2022 – 14.02.2022 |
| Теоретическая часть | 15.02.2022 – 15.03.2022 |
| Руководство программиста | 16.03.2022 – 14.04.2022 |
| Руководство пользователя | 15.04.2022 – 19.05.2022 |
| Руководство администратора | 20.05.2022 – 30.05.2022 |
| Заключение | 31.05.2022 – 07.06.2022 |
| Список использованных источников | 08.06.2022 – 10.06.2022 |

Дата выдачи задания «21» _____ января _____ 2022 г.

Руководитель _____
подпись

Задание принял к исполнению «21» _____ января _____ 2022 г.

Студент _____

Министерство цифрового развития, связи и массовых коммуникаций РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

Форма утверждена научно-методическим
советом университета
протокол № 3 от 19.02.2015 г.

ОТЗЫВ

на выпускную квалификационную работу студента Пещерова В.А.
по теме «Разработка интернет – магазина авторских вещей»

Тема ВКР, посвященная разработке интернет - магазина в настоящее время является весьма актуальной.

В процессе создания приложения были использованы современное программное обеспечение и языки программирования. Разработаны интернет-магазин, пользовательская и администраторская часть.

Выполненная работа соответствует поставленной задаче и имеет практическую ценность.

Доля оригинальности текста ВКР по результатам проверки на сайте antiplagiat.ru составляет 86%. Считаю, что представленная ВКР заслуживает оценки “Хорошо”, а ее автор присуждения квалификации бакалавр.

Оценка уровней сформированности общекультурных и профессиональных компетенций обучающегося:

| Компетенции | | Уровень сформированности компетенции | | |
|------------------|-------|--------------------------------------|---------|--------|
| | | Высокий | Средний | Низкий |
| Общекультурные | ОК-7 | | v | |
| Профессиональные | ОПК-4 | | v | |
| | ОПК-5 | | v | |
| | ПК-1 | | v | |
| | ПК-2 | | v | |
| | ПК-3 | | v | |

Работа имеет практическую ценность
Работа внедрена
Рекомендую работу к внедрению
Рекомендую работу к опубликованию
Работа выполнена с применением ЭВМ

| |
|---|
| v |
| |
| |
| |
| v |

Тема предложена предприятием
Тема предложена студентом
Тема является фундаментальной
Рекомендую студента в магистратуру
Рекомендую студента в аспирантуру

| |
|---|
| |
| v |
| |
| |
| |

Руководитель бакалаврской работы _____ к.т.н., доцент Перцев Игорь Владимирович
09.06.2022

АННОТАЦИЯ

Выпускной квалификационной работы студента Пещерова Вячеслава Александровича по теме «Разработка интернет-магазина авторских вещей»

Объём работы - 43 страниц, на которых размещены 26 рисунков и 0 таблиц. При написании работы использовалось 14 источников.

Ключевые слова: Интернет-магазин, функции, JavaScript, PHP, AJAX, action, models, web-сайт, страницы.

Работа выполнена

на кафедре ПМиК

(название предприятия, подразделения)

Руководитель

доцент кафедры ПМиК Перцев Игорь Владимирович

(должность, уч.степень, звание, Фамилия Имя Отчество)

Основные результаты

- Разработан интернет-магазин
- Изучены новые средства разработки
- Получены новые знания в сфере web-разработки

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ..... | 3 |
| 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ..... | 5 |
| 1.1 Общие требования к сайту..... | 5 |
| 1.2 Выбор хостинга..... | 5 |
| 1.3 Среда разработки..... | 5 |
| 1.4 Обзор языков программирования и других инструментов, которые используют при разработке web-сайтов..... | 6 |
| 1.5 Информационные базы данных..... | 9 |
| 2. РУКОВОДСТВО ПРОГРАММИСТА..... | 10 |
| 2.1 Базы данных..... | 10 |
| 2.1 Функционал загрузки страниц..... | 12 |
| 2.2 Функционал работы с базой данных..... | 13 |
| 2.3 Сбор данных о категориях..... | 14 |
| 2.4 Работа с товарами..... | 15 |
| 2.5 Работа с корзиной..... | 16 |
| 2.6 Авторизация и регистрация пользователей..... | 19 |
| 2.7 Изменение данных пользователя..... | 24 |
| 2.8 Реализация поиска товаров..... | 27 |
| 2.9 Построение страниц с помощью шаблонизатора..... | 28 |
| 2.10 Оформление заказов..... | 29 |
| 2.11 Настройка административной панели..... | 31 |
| 2.11.1 Редактирование категорий..... | 32 |
| 2.11.2 Работа с товарами..... | 33 |
| 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ..... | 36 |
| 4. РУКОВОДСТВО АДМИНИСТРАТОРА..... | 40 |
| ЗАКЛЮЧЕНИЕ..... | 43 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 44 |
| ПРИЛОЖЕНИЕ А..... | 45 |
| ПРИЛОЖЕНИЕ Б..... | 46 |

ВВЕДЕНИЕ

В настоящее время сайты стали невероятно популярной вещью для распространения информации и заявления о себе в мире. Возможность пользователя узнать о компании или заказать какую-то услугу или воспользоваться быстрым сервисом для решения какой-либо своей проблемы, не выходя из дома, уже звучит очень притягательно и удобно, поэтому сайты с каждым днём набирают всё больше оборотов и направление веб-разработки вместе с ними.

Интернет-магазин является ярким примером такого удобства, так как он снижает ваши временные и энергетические затраты до минимума. Так как они предоставляют возможность пользователю купить что-либо или договориться о покупке, не выходя из дома. Не нужно больше ходить по куче магазинов в поисках какой-то определённой вещи и ведь не факт, что вы ещё найдёте её или она окажется у продавцов в наличии. Теперь все вещи у вас на расстоянии вытянутой руки, ведь к вашему выбору предоставляется миллиарды разных вещей, которые вы можете приобрести в несколько быстрых и простых кликов. Вы можете всегда выбирать: любые размеры, стили, формы, цвета, всё что угодно вы можете найти на страницах интернет-магазинов. И с их развитием теперь вам даже не надо никуда далеко ходить за вашей покупкой, ведь многие интернет-магазины предоставляют доставку практически до самой двери или до любого удобного вам почтового сервиса, вам лишь надо будет сходить и получить её, это невероятно удобно особенно для тех людей кто не обладает возможностью просто сходить в магазин и купить вещь напрямую, например, человек может жить в маленьком городе, деревне, посёлке и возможности купить что-либо в популярных, брендовых магазинах собственноручно у него попросту нету из-за отсутствия этих магазинов у себя, и интернет-магазины легко и просто решают данную проблему.

Также развитие интернет-магазинов даёт возможность многим людям развивать и свой собственный бизнес, так как с каждым годом интернет всё больше и больше становится частью нашей жизни, и вместе с этим всё больше пользователей может наткнуться на ваш магазин в интернете, и вы получите множество новых, разных клиентов и пользователей, которые как минимум узнают о вашем интернет-магазине и о товарах, которые вы продаёте, ведь несмотря где именно был обустроен ваш бизнес, вы сможете искать новых клиентов практически по всему миру, и это конечно же принесёт очень много дополнительной прибыли вашему бизнесу. Также развитие интернет-магазинов предоставляет возможность даже самому обычному человеку открыть свой бизнес, ведь для этого не нужно какого-то места для развёртывания так называемого прилавка, не нужно платить ежемесячную аренду, достаточно просто оплачивать свой домен, что в несколько раз дешевле и быстрее приведёт ваш интернет-магазин к

стабильному доходу, так как абсолютно всё находится внутри компьютера, вам лишь надо будет следить за заказами ваших клиентов и вовремя выполнять их.

Что же касается конкретно веб-разработки это одно из самых популярных и перспективных в наше время направлений, так как даже любое программное обеспечение или сервис рано или поздно просто переносится в интернет-пространство, примером этого могут являться, например, онлайн компиляторы, которые могут уже даже собирать проекты, или программное обеспечение, которое раньше находилось в виде программы установленной на компьютере, теперь многие из них перенеслись прямо в браузер, это экономит не только место на вашем компьютере, а также и мощности вашего компьютера, так как обрабатываются данные чаще всего именно на сервере и нагружается соответственно сервер, а не ваш компьютер. И именно веб-разработчик должен переносить программное обеспечение и другие сервисы в веб-пространство, именно он создаёт большинство сервисов и сайтов для бизнесов и постоянно развивает их и поддерживает. Именно поэтому это направление является максимально перспективным и нужным современном рынку.

Из вышесказанного можно сделать вывод, что интернет-магазины очень удобная и уже неотъемлемая часть нашего общества, интернет-магазины уже доставляют не только какие-то простые вещи по типу футболок, кружек, значков, а также уже доставляют продукты, услуги и многое другое, обеспечивая людей комфортом и меньшей энергозатратностью, а также более широким и интересным выбором, поэтому веб-разработка является очень перспективной в наше время, кучу бизнесов могут легко продвигаться с помощью интернета, кучу удобных сервисов реализуется каждый день, которые упрощают людям жизнь.

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Общие требования к сайту

Данный интернет-магазин должен обладать всеми стандартными функциями интернет-магазинов:

- Покупка
- Добавление товара в корзину и удаление товара из корзины
- Профиль клиента с возможностью просматривать статус заказа
- Взаимодействие с категориями, выбор нужной категории
- Регистрация и авторизация пользователя
- Оформления заказа
- Просмотр товара
- Возможность редактировать количество покупаемого товара
- Поиск
- Административная панель

1.2 Выбор хостинга

Первой задачей перед созданием моего интернет-магазина стало выбором хостинга, для данной работы я выбрал популярный хостинг хатрр, она отлично подходит для создания сервера на локальной платформе, а также отлично функционирует при отключении компьютера от интернета

- **Хостинг** - услуга по предоставлению ресурсов для размещения информации на сервере, постоянно имеющем доступ к сети. Обычно услуга хостинга входит в пакет по обслуживанию сайта и подразумевает, как минимум, размещение файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам.

- **Xampp** - кроссплатформенная сборка локального веб-сервера, содержащая Apache, MariaDB, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

1.3 Среда разработки

Второй задачей моей работы стал выбор ПО необходимого для разработки, так как для разработки мною использовался язык PHP, мне нужна была среда разработки, которая поддерживает её и дополнительные функции, который могут мне пригодиться, выбор пал на среду NetBeans IDE, в которой было удобно реализовывать проекты данного рода

- **NetBeans IDE** - свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других.

1.4 Обзор языков программирования и других инструментов, которые используют при разработке web-сайтов

В настоящее время существует куча языков программирования используемых для разработки web-сайтов, здесь мы рассмотрим все инструменты, которые чаще всего используют и backend разработчики и frontend разработчики, но в начале определимся что значит frontend и backend.

- **Backend** - (или веб-разработка на стороне сервера) охватывает все те вещи, которые работают в фоновом режиме, чтобы создавать веб-сайты или веб-приложения, например базы данных и скрипты.

- **Frontend** - (также известная как веб-разработка на стороне клиента) касается части веб-интерфейса, с которой пользователь взаимодействует - часть, с которой знакомы все, кто пользуется Интернетом.

Начнём с frontend, так как конечно же внешний вид нашего сайта, это первое что увидит пользователь и поэтому это очень важная часть разработки, которой обязательно нужно уделить должное количество внимания, для создания дизайна используют в основном следующие инструменты, первое это HTML:

- **HTML** - стандартизированный язык разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

Далее идёт CSS:

- **CSS** - формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Также может применяться к любым XML-документам, например, к SVG или XUL.

CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь

производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом.

Далее используют JavaScript:

- **JavaScript** - это язык скриптов, на котором держится весь frontend веб-разработки. Он позволяет перехватывать события и выполнять различные действия. Например, пользователь кликнул по какой-нибудь кнопке — сработало событие click. И, связав с ним, мы можем выполнить нужную нам функцию — открыть модальное окно или изменить цвет элемента.

JavaScript используется для того, чтобы делать страницы интерактивными, то есть дать пользователю возможность взаимодействовать с элементами. Когда страницы могут реагировать на какие-то действия, это делает их интереснее. Если, конечно, не намешано много безвкусицы.

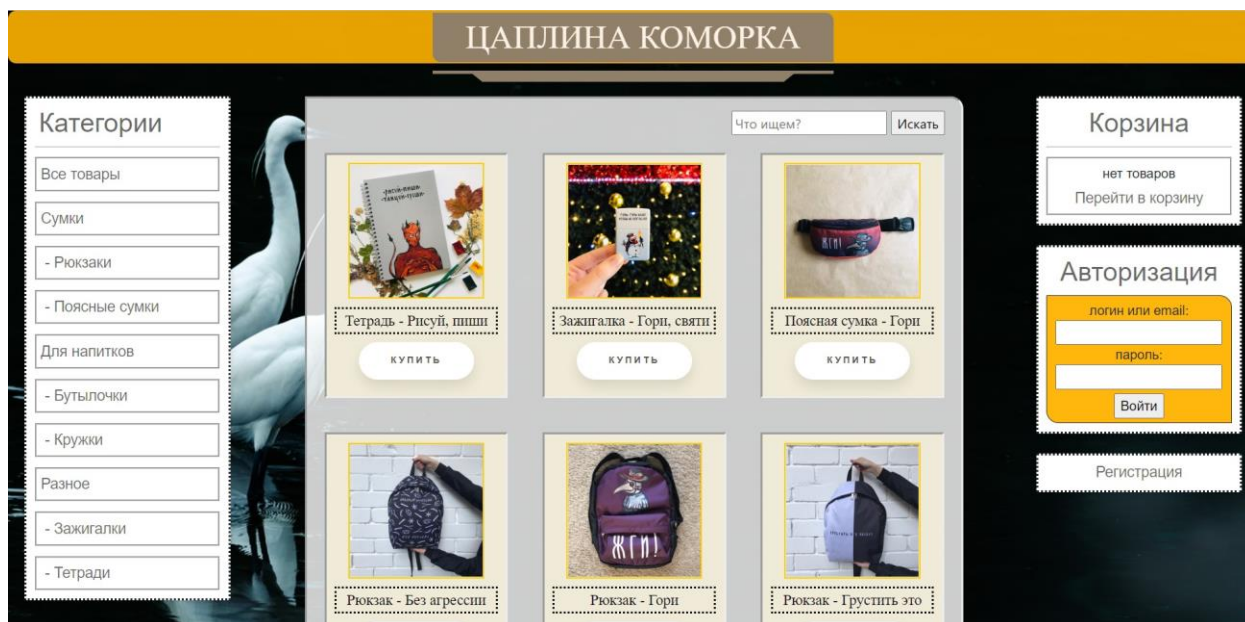


Рисунок 1.1. – Пример дизайна магазина.

С дизайном мы разобрались дальше переходим к двигателю сайта, который позволяет нашему сайту работать и выполнять нужные нам функции и задачи, для этого нам понадобится уже язык программирования, которые будет выполнять нужные нам команды, мною был выбран PHP:

- **PHP** - это скриптовый язык, используемый для быстрого создания динамических веб-страниц. Отличный выбор для frontend и backend разработчиков, чтобы добавить их в арсенал (но особенно для последних), он стоит за такими веб-гигантами, как WordPress и Facebook. PHP позволяет

быстро и легко расширять веб-приложения и запускать веб-сайты с повторяющимися серверными задачами (например, обновлять новостные ленты). Он имеет открытый исходный код и очень популярен в среде начинающих компаний, медиа-агентств и электронной коммерции - таких людей, которые часто нанимают новых веб-разработчиков.

При разработке различных web-сайтов в частности и моего, используют базы данных и для работы с ним большинство разработчиков используют язык SQL:

- **SQL** – это декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Является, прежде всего, информационно-логическим языком, предназначенным для описания, изменения и извлечения данных, хранимых в реляционных базах данных. В общем случае SQL (без ряда современных расширений) считается языком программирования не полным по Тьюрингу, но вместе с тем стандарт языка спецификацией SQL/PSM предусматривает возможность его процедурных расширений.

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

- создание в базе данных новой таблицы;
- добавление в таблицу новых записей;
- изменение записей;
- удаление записей;
- выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
- изменение структур таблиц.

Со временем SQL усложнился — обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) — и стал приобретать черты, свойственные языкам программирования.

Также существуют ещё множество инструментов, которые помогают упрощать процесс создания web-страниц, в частно лично для моего интернет-магазина мне понадобились дополнительные элементы, которые мне помогут облегчить задачу написание интернет-магазина, для этого я установил библиотеку JQuery, которая нужна для поддержки JavaScript, а также шаблонизатор Smarty, который в будущем поможет написать более удобную логику работы вёрстки

- **JQuery** - набор функций JavaScript, фокусирующийся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX. Разработка jQuery ведётся командой добровольцев на пожертвования.

- **Шаблонизатор** - программное обеспечение, позволяющее использовать шаблоны для генерации конечных документов с помощью декларативного языка разметки.

- **Smarty** - наиболее популярная и широко распространенная система шаблонов на PHP. Работа Smarty заключается в компилировании шаблонов. Это означает, что Smarty последовательно считывает разметку файла шаблона и создает на их основе готовый PHP сценарий, этот сценарий создается один раз, а далее просто выполняется. Smarty содержит в себе большое количество функций, которые позволяют создавать в шаблонах сложную логику (если она нужна для правильного отображения данных): подключение других шаблонов, циклический разбор массива и т.д. Конечно, Вы можете и не создавать сложную логику шаблона, а ограничиться лишь использованием чистого текста и переменных.

1.5 Информационные базы данных

В своей работе я также использовал базы данные, которые немного были упомянуты выше

- **База данных** — это место для хранения данных. Используется в том числе в клиент-серверной архитектуре. Это все интернет-магазины, сайты кинотеатров или авиабилетов. Вы делаете заказ, а система сохраняет ваши данные в базе.

Все используемые мной базы данные были созданы с помощью phpMyAdmin, которая позволяет очень удобно и быстро создавать нужные базы данных, в итоге мной создано была одна база данных, которая содержит в себе 5 таблиц, а именно для работы с категориями, работы с заказами, таблица продуктов, таблица покупок, и таблица пользователей.

- **phpMyAdmin** - веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных

2. РУКОВОДСТВО ПРОГРАММИСТА

2.1 Базы данных

Для разработки интернет-магазина была разработана база данных, состоящая из 5 таблиц, которые мне понадобились для разработки: tables_of_category, table_of_products, table_of_orders, table_of_purchase, info_users. Модель базы данных представлена на рисунке ниже.

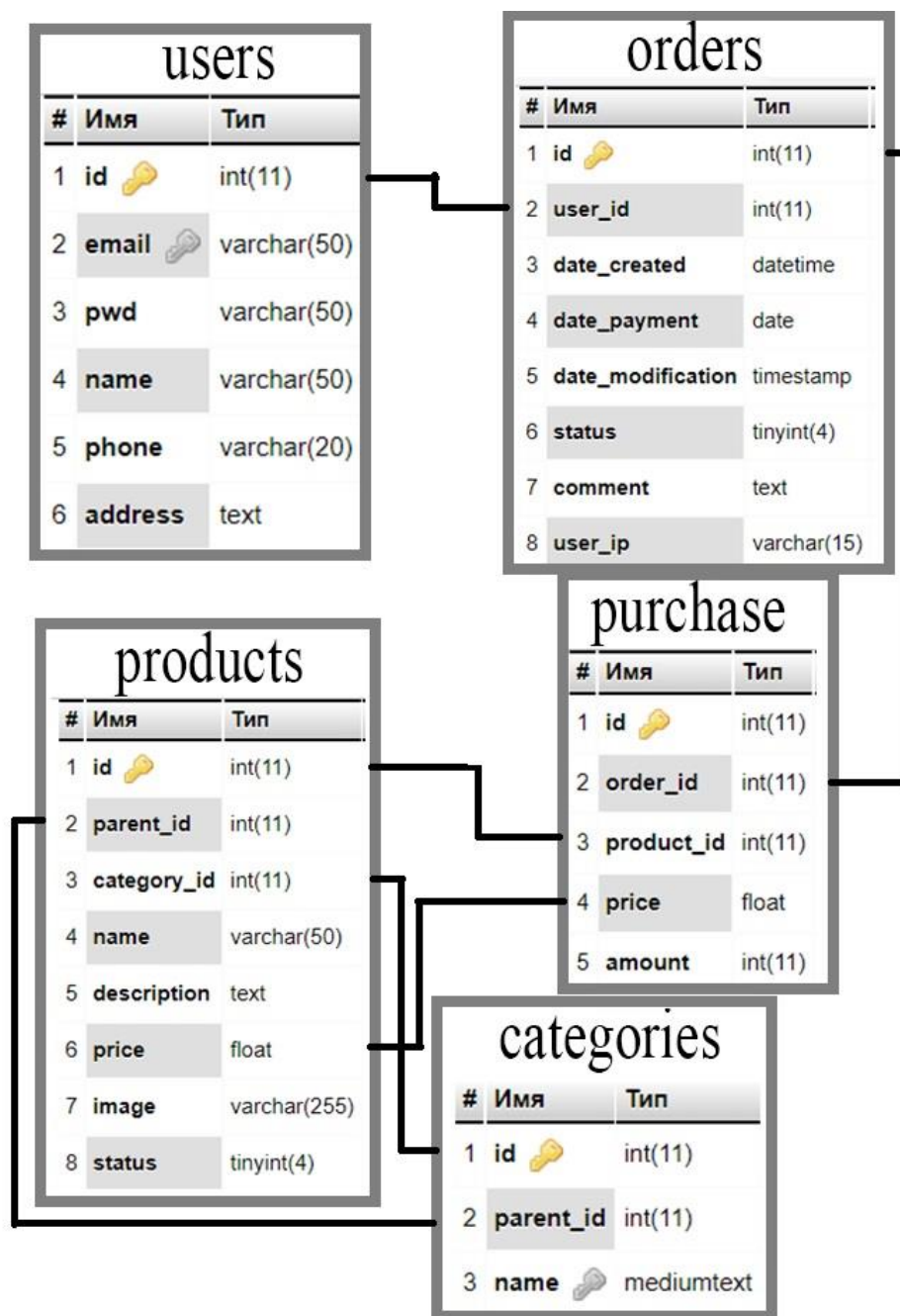


Рисунок 2.1. – Физическая модель базы данных.

Рассмотрим каждую из таблиц подробнее:

Таблица `tables_of_category` в ней указаны все категорий товаров и их родительские категории.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно |
|--------------------------|---|------------------|------------|-----------------|----------|------|--------------|-------------|----------------|
| <input type="checkbox"/> | 1 | id | int(11) | | | Нет | Нет | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | parent_id | int(11) | | | Нет | Нет | | |
| <input type="checkbox"/> | 3 | name | mediumtext | utf8_general_ci | | Нет | Нет | | |

Рисунок 2.2. – Таблица `tables_of_category`.

Таблица `table_of_orders` в ней указаны все заказы и данные заказа, сделанные пользователями.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно |
|--------------------------|---|--------------------------|-------------|-----------------|----------|------|---------------------|-------------|-------------------------------|
| <input type="checkbox"/> | 1 | id | int(11) | | | Нет | Нет | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | user_id | int(11) | | | Нет | Нет | | |
| <input type="checkbox"/> | 3 | date_created | datetime | | | Нет | Нет | | |
| <input type="checkbox"/> | 4 | date_payment | date | | | Да | NULL | | |
| <input type="checkbox"/> | 5 | date_modification | timestamp | | | Нет | current_timestamp() | | ON UPDATE CURRENT_TIMESTAMP() |
| <input type="checkbox"/> | 6 | status | tinyint(4) | | | Нет | 0 | | |
| <input type="checkbox"/> | 7 | comment | text | utf8_general_ci | | Нет | Нет | | |
| <input type="checkbox"/> | 8 | user_ip | varchar(15) | utf8_general_ci | | Нет | Нет | | |

Рисунок 2.3. – Таблица `table_of_orders`.

Таблица `table_of_products` в ней указаны все товары, которые есть у нас в магазине.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно |
|--------------------------|---|--------------------|--------------|-----------------|----------|------|--------------|-------------|----------------|
| <input type="checkbox"/> | 1 | id | int(11) | | | Нет | Нет | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | parent_id | int(11) | | | Нет | Нет | | |
| <input type="checkbox"/> | 3 | category_id | int(11) | | | Нет | Нет | | |
| <input type="checkbox"/> | 4 | name | varchar(50) | utf8_general_ci | | Нет | Нет | | |
| <input type="checkbox"/> | 5 | description | text | utf8_general_ci | | Нет | Нет | | |
| <input type="checkbox"/> | 6 | price | float | | | Нет | Нет | | |
| <input type="checkbox"/> | 7 | image | varchar(255) | utf8_general_ci | | Нет | Нет | | |
| <input type="checkbox"/> | 8 | status | tinyint(4) | | | Нет | 1 | | |

Рисунок 2.4. – Таблица `table_of_products`.

Таблица `table_of_purchase` в ней указаны товары купленные пользователем в каком-либо заказе.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно |
|--------------------------|---|-------------------|---------|-----------|----------|------|--------------|-------------|----------------|
| <input type="checkbox"/> | 1 | id | int(11) | | | Нет | <i>Нem</i> | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | order_id | int(11) | | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 3 | product_id | int(11) | | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 4 | price | float | | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 5 | amount | int(11) | | | Нет | <i>Нem</i> | | |

Рисунок 2.5. – Таблица `table_of_purchase`

Таблица `info_users` в ней указаны все пользователи зарегистрированные в нашем интернет-магазине, а также указаны их данные.

| | # | Имя | Тип | Сравнение | Атрибуты | Null | По умолчанию | Комментарии | Дополнительно |
|--------------------------|---|----------------|-------------|-----------------|----------|------|--------------|-------------|----------------|
| <input type="checkbox"/> | 1 | id | int(11) | | | Нет | <i>Нem</i> | | AUTO_INCREMENT |
| <input type="checkbox"/> | 2 | email | varchar(50) | utf8_general_ci | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 3 | pwd | varchar(50) | utf8_general_ci | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 4 | name | varchar(50) | utf8_general_ci | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 5 | phone | varchar(20) | utf8_general_ci | | Нет | <i>Нem</i> | | |
| <input type="checkbox"/> | 6 | address | text | utf8_general_ci | | Нет | <i>Нem</i> | | |

Рисунок 2.6. – Таблица `info_users`

2.1 Функционал загрузки страниц

За формирование страниц отвечают специальные файлы именуемые как “контроллеры” название файлов будет состоять из самого логического названия страницы и постфикс “controller”, а функция формирования страницы будет называться `indexAction`, которая будет располагаться в данном контроллере.

Все же запросы из браузера будет обрабатывать файл `index.php`, в нём мы будем определять какой именно “контроллер” в данный момент нам нужно будет подключить и какую функцию вызвать из данного “контроллера”, данную информацию мы будем получать из `url` браузера (Листинг 2.1) и с помощью функции `loading` (Листинг 2.2) мы собираем название функции загрузки страницы и вызываем её.

Листинг 2.1 – Функционал загрузки страниц index.php

```
If(isset($_GET['controller'])){
    $name_of_controller = ucfirst($_GET['controller']);
}else{ $name_of_controller = 'Index';}
If(isset($_GET['action'])){
    $name_of_action = $_GET['action'];}else{ $name_of_action
='index';}
loading_full_page($smarty, $name_of_controller,
$name_of_action);
```

Листинг 2.2 – Функция отвечающая за сбор названия нужной функции и вызова её mainFunctions.php

```
function loading_full_page($smarty, $name_of_controller,
$name_of_action = 'index')
{include_once PathPrefix . $name_of_controller .
PathPostfix;
    $function = $name_of_action . 'Action';
    $function($smarty);}
```

2.2 Функционал работы с базой данных

За работу с таблицами из базы данных будут отвечать специальные файлы, именуемые “моделями”. Данных файлов будет 5 по количеству таблиц: CategoryTableModel.php, OrderTableModel.php, ProductTableModel.php, PurchTableModel.php, UserInfoTableModel.php. В данных файлах будут находиться все функции для работы с каждой из таблиц.

За подключение к самой базе данных будет отвечать отдельный файл database.php (Листинг 2.3).

Листинг 2.3 – Функционал отвечающий за подключение к базе данным database.php

```
function database() {
    $database_location = "127.0.0.1";
    $database_name = 'myshop';
    $database_user = 'root';
    $database_passwd = '';
    $database = new mysqli($dblocation, $dbuser, $dbpasswd,
$dbname);
    if (!$database) {
        echo "Ошибка доступа к базе данных MySQL";
        exit();}
    $database->set_charset( 'utf8');
    if ($database->connect_errno) {
        echo "Ошибка доступа к базе данных: {$dbname}";
```

```

        exit();
    }
    return $database;

```

2.3 Сбор данных о категориях

Для работы с категориями были разработаны следующие функции `get_absolutely_all_categories` (Листинг 2.4), которая возвращает массив собранных ею категорий, находится она в файле `CategoryTableModel.php`

Листинг 2.4 – Функция `get_absolutely_all_categories`, которая собирает все категории

```

function get_absolutely_all_categories() {
    $request_sql = 'SELECT * FROM tables_of_category WHERE
parent_id = \'0\';
    $database_result = database()->query($request_sql);
    $record_set_of_smarty = array();
    while($row = mysqli_fetch_assoc($database_result)) {
        $records_of_child_categories
get_child_category_based_on_parent($row['id']);
        if ($records_of_child_categories) {
            $row['children'] = $records_of_child_categories;
        }
        $record_set_of_smarty[] = $row;
    }

    return $record_set_of_smarty;
}

```

Далее для сбора конкретно дочерних категорий была создана функция `get_child_category_based_on_parent` (Листинг 2.5), которая находится в том же файле `CategoryTableModel.php`

Листинг 2.5 – Функция `get_child_category_based_on_parent`, отвечающая за сбор дочерних категорий на основе родительской

```

function get_child_category_based_on_parent($category_id) {
    $request_sql = "SELECT * FROM tables_of_category WHERE
parent_id = '{$category_id}'";
    $database_result = database()->query($request_sql);

    return
generate_array_of_smarty_records($database_result);
}

```

Для работы с определённой категорией была создана функция `get_category_data_by_id` (Листинг 2.6), расположена в файле `CategoryTableModel.php`.

Листинг 2.6 – Функция получения всей информации по категории с нужным id

```
function get_category_data_by_id($category_id) {

    $category_id = intval($category_id);

    $request_sql = "SELECT * FROM tables_of_category WHERE
id = '{$category_id}'";

    $database_result = database()->query($request_sql);
    return mysqli_fetch_assoc($database_result);
}
```

2.4 Работа с товарами

Для работы с товарами были разработаны следующие функции, находящиеся в файле ProductTableModel.php, а именно: `get_the_right_amount_products` (Листинг 2.7) функция, которая будет отвечать за сбор нужного количества товаров, `get_products_for_this_category` (Листинг 2.8), которая позволяет получить продукты относящиеся к конкретной категории, `get_all_products_for_general_category` (Листинг 2.9) функция отвечающая за вывод всех товаров из родительской или главной категории, `get_product_by_his_id` (Листинг 2.10) функция позволяет получить всю информацию о товаре по его id, `get_list_of_products_from_array` (Листинг 2.11) функция получает список продуктов (нужно для работы с корзиной).

Листинг 2.7 – Функция `get_the_right_amount_products`

```
function
get_the_right_amount_products($max_count_of_products =
null){
    $request_sql = "SELECT * FROM table_of_products ORDER BY id
DESC";
    if($max_count_of_products){
        $request_sql .= " LIMIT {$max_count_of_products}";
    }
    $database_result = database()->query($request_sql);
    return generate_array_of_smarty_records($database_result);
}
```

Листинг 2.8 – Функция `get_products_for_this_category`

```
function get_products_for_this_category($item_id){

    $request_sql = "SELECT * FROM table_of_products WHERE
category_id = {$item_id}";
    $database_result = database() ->query($request_sql);
    return generate_array_of_smarty_records($database_result);
}
```

Листинг 2.9 – Функция `get_all_products_for_general_category` – получает данные для общей категории

```
function get_all_products_for_general_category($item_id){
```

```

$request_sql = "SELECT * FROM table_of_products WHERE
parent_id = {$item_id}";
$database_result = database() ->query($request_sql);
return generate_array_of_smarty_records($database_result);
}

```

Листинг 2.10 – Функция get_list_of_products_from_array

```

function get_list_of_products_from_array($idS_items){
if($idS_items){
$string_id = implode(' ', $idS_items);
$request_sql = "SELECT * FROM table_of_products WHERE id IN
({$string_id})";
$conn = connectBD();
$database_result = database() ->query($request_sql);
return generate_array_of_smarty_records($database_result);
}
else{
return false;
}}

```

2.5 Работа с корзиной

Корзина в большинстве интернет-магазинов устроена так, чтобы её можно было видеть постоянно, и пользователь всегда имел доступ к ней, а также видел количество товаров в ней на данный момент.

Для этого мы контролируем сессию клиента в файле index.php, и постоянно держим количество его товаров.

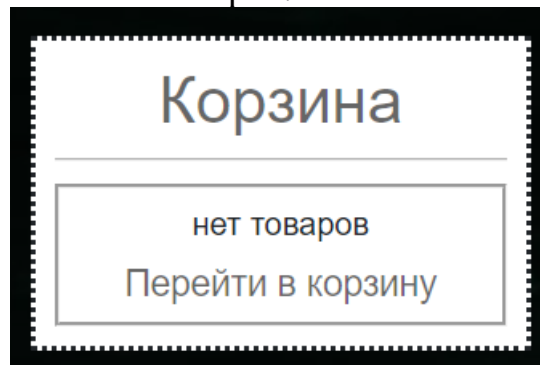


Рисунок 2.7 – Пример отображения пустой корзины

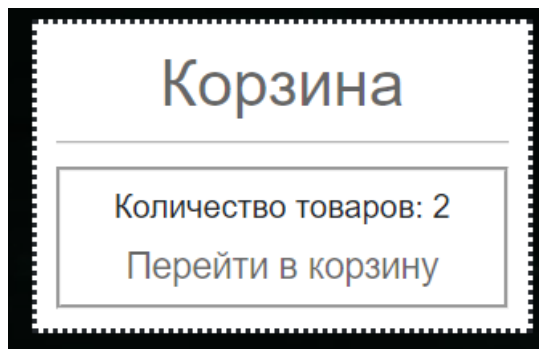


Рисунок 2.8 – Пример отображения корзины с товарами

Листинг 2.11 – Код, отвечающий за постоянное отображение количества элементов в корзине

```
session_start();
if (!isset($_SESSION['session_cart']))
    $_SESSION['session_cart'] = array();
$smarty->assign('number_of_products_in_the_cart',
count($_SESSION['session_cart']));
```

Для того, чтобы добавить товар в корзину или же удалить из неё были разработаны две функции в файле cartController.php, функция добавления в корзину add_to_cartAction(Листинг 2.12), функция удаление из корзины remove_from_cartAction(Листинг 2.13).

Листинг 2.12 – Функция add_to_cartAction

```
function add_to_cartAction() {

    If(isset($_GET['id'])){
        $item_id = intval($_GET['id']);
    }else{$item_id = null;}

    if ($item_id == null) return false;
    $result_data = array();
    if (isset($_SESSION['session_cart']) &&
array_search($item_id, $_SESSION['session_cart']) == false)
    {
        $_SESSION['session_cart'][] = $item_id;
        $result_data['count_of_items'] =
count($_SESSION['session_cart']);
        $result_data['success'] = 1;
    }else {
        $result_data['success'] = 0;
    }
    echo json_encode($result_data);
}
```

Листинг 2.13 – Функция удаления из корзины remove_from_cartAction

```

function remove_from_cartAction(){
    If(isset($_GET['id'])){
        $item_id = intval($_GET['id']);
    }else{$item_id = null;}

    if ($item_id == null) exit();

    $key_of_item = array_search($item_id,
$_SESSION['session_cart']);
    $result_data = array();
    if ($key_of_item) {
        unset($_SESSION['session_cart'][$key_of_item]);
        $result_data['success'] = 1;
        $result_data['count_of_items'] =
count($_SESSION['session_cart']);
    }else {
        $result_data['success'] = 0;
    }
    echo json_encode($result_data);
}

```

С помощью ajax-функции из библиотеки jquery мы будем отлавливать данные и изменять количество товаров в корзине по нажатию на гиперссылку, эти функции мы реализуем в файле main.js. Сама библиотека jquery была подключена в файле site_header.tpl.

Листинг 2.14 – Добавление в корзину add_item_to_cart

```

function add_item_to_cart(item_id) {
    $.ajax({
        type: 'POST',
        async: true,
        url: "/cart/add_to_cart/" + item_id + '/',
        dataType: 'json',
        success: function(data) {
            if(data['success']) {
                $('#number_of_products_in_the_cart').html(data['count_of_items']);
                $('#add_item_to_cart_' + item_id).hide();
                $('#remover_item_to_cart_' +
item_id).show();
            }
        }
    });
}

```

Листинг 2.15 – Удаление товара из корзины

```

function remover_item_from_cart(item_id) {

```

```

$.ajax({
    type: 'POST',
    async: true,
    url: "/cart/remove_from_cart/" + item_id + '/',
    dataType: 'json',
    success: function (data) {
        if (data['success']) {
            $('#number_of_products_in_the_cart').html(data['count_of_items']);

            $('#add_item_to_cart_' + item_id).show();
            $('#remover_item_to_cart_' +
item_id).hide();
        }
    }
});
}

```

Листинг 2.16 – Функция, которая изменяет цену в зависимости от количество товаров

```

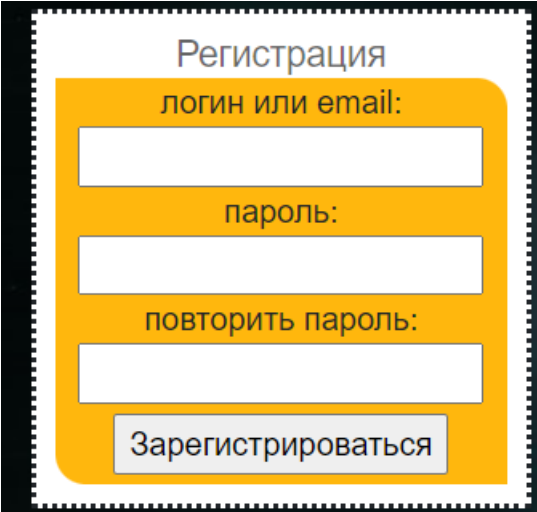
function check_full_price_of_item(item_id) {
    var count_new = $('#count_of_items_' + item_id).val();
    var price_of_item = $('#price_of_item_'
+ item_id).attr('value');
    var full_price_of_items = count_new * price_of_item;

    $('#full_price_of_items_' +
item_id).html(full_price_of_items);
}

```

2.6 Авторизация и регистрация пользователей

Для регистрации новых пользователей есть специальное окно “регистрация” куда пользователь вводит свои данные.



The image shows a registration form titled "Регистрация" (Registration). It is contained within a yellow rounded rectangle with a black border. The form has three input fields: "логин или email:" (login or email:), "пароль:" (password:), and "повторить пароль:" (repeat password:). Below these fields is a button labeled "Зарегистрироваться" (Register).

Рисунок 2.9 – Форма регистрации в интернет-магазине

Регистрация реализована с помощью ајах-запроса, функция `new_user_register` (Листинг 2.17) из файла `main.js`, обращается к `register_userAction` (Листинг 2.18) из файла `UserController.php`, которые в свою очередь считывает полученные данные проверяет и с помощью функции `verify_register_params` (Листинг 2.19), которая проверяет верны ли введены данные для регистрации, и с помощью функции `verify_user_email_in_bd` (Листинг 2.20), и когда все данные проходят проверку с помощью функции из файла `UserInfoTableModel.php` `new_user_register` (Листинг 2.21) пользователь заносится в базу данных.

Листинг 2.17 – JS функция `new_user_register`

```
function new_user_register() {
    var data_for_post =
    get_data_from_html_forms('#register_form');
    $.ajax({
        type: 'POST',
        async: true,
        url: "/user/register_user/",
        data: data_for_post,
        dataType: 'json',
        success: function (data) {
            if(data['success']) {
                alert('Вы зарегистрировались на сайте');
                $('#register_form').hide();
                $('#user_connect').attr('href', '/user/');

                $('#user_connect').html(data['name_of_user']);
                $('#user_form_of_authorized').show();
                $('#form_login').hide();
                $('#button_save_order').show();

                $('#errorBlock').show()
            }else {
                alert(data['message']);
            }
        }
    });
}
```

Листинг 2.18 – Функция `register_userAction` отвечающая за регистрацию

```
function register_userAction() {
    $user_email_addres =
    isset($_REQUEST['user_email_addres']) ?
    $_REQUEST['user_email_addres'] : null;
    $user_email_addres = trim($user_email_addres);

    $first_password = isset($_REQUEST['first_password']) ?
    $_REQUEST['first_password'] : null;
```

```

    $second_password = isset($_REQUEST['second_password']) ?
$_REQUEST['second_password'] : null;

    $user_phone = isset($_REQUEST['user_phone']) ?
$_REQUEST['user_phone'] : null;
    $user_adress = isset($_REQUEST['user_adress']) ?
$_REQUEST['user_adress'] : null;
    $user_name = isset($_REQUEST['user_name']) ?
$_REQUEST['user_name'] : null;
    $user_name = trim($user_name);
    $result_data = null;
    $result_data =
verify_register_params($user_email_user_adress,
$first_password, $second_password);
    if(!$result_data &&
verify_user_email_in_bd($user_email_adress)) {
        $result_data['success'] = false;
        $result_data['message'] = "Такой email
('{$user_email_adress}') уже существует";
    }
    if(!$result_data) {
        $password_in_md5_form = md5($first_password);
        $data_of_user =
new_user_register($user_email_adress, $password_in_md5_form,
$user_name, $user_phone, $user_adress);

        if ($data_of_user['success']) {

            $result_data['message'] = 'Пользователь
зарегистрирован';
            $result_data['success'] = 1;$data_of_user =
$data_of_user[0];
            $result_data['name_of_user'] =
$data_of_user['user_name'] ? $data_of_user['user_name'] :
$data_of_user['user_email_adress'];
            $result_data['userEmail'] = $user_email_adress;

            $_SESSION['session_user'] = $data_of_user;
            $_SESSION['session_user']['name_of_display'] =
$data_of_user['user_name'] ? $data_of_user['user_name'] :
$data_of_user['user_email_adress'];
            //$_SESSION['session_user']['name_of_display'] =
$data_of_user['name_of_user'];

        }else{
            $result_data['success'] = 0;
            $result_data['message'] = 'Ошибка регистрации';
        }
    }

    echo json_encode($result_data);
}

```

Листинг 2.19 – Функция `verify_register_params`, которая отвечает за проверку введённых данных

```
function verify_register_params($user_email_addres,
    $first_password, $second_password)
{
    $result = null;
    if (!$user_email_addres) {
        $result['success'] = false; $result['message'] =
        'Email пуст';
    }

    else if (!$first_password) {
        $result['success'] = false; $result['message'] =
        'Пароль не введён';
    }

    else if (!$second_password) {
        $result['success'] = false; $result['message'] =
        'Повторите пароле';
    }

    else if ($first_password != $second_password) {
        $result['success'] = false; $result['message'] =
        'Пароли не совпадают';
    }
    return $result;
}
```

Листинг 2.20 – Функция `verify_user_email_in_bd`, отвечает за проверку существование логина нашего пользователя в базе данных

```
function verify_user_email_in_bd($user_email_addres) {

    $user_email_addres =
    mysqli_real_escape_string(database(), $user_email_addres);
    $request_sql = "SELECT `id` FROM `info_users` WHERE
    `user_email_addres` = '{$user_email_addres}'";

    $database_result = database()->query($request_sql);
    $database_result =
    generate_array_of_smarty_records($database_result);

    return $database_result;
}
```

Листинг 2.21 – Функция `new_user_register`, которая вносит данные нового пользователя в базу данных

```
function new_user_register($user_email_addres,
    $password_in_md5_form, $user_name, $user_phone,
    $user_adress) {
```

```

        $user_email_addres =
htmlspecialchars(mysqli_real_escape_string(database(),
$user_email_addres));
        $user_name =
htmlspecialchars(mysqli_real_escape_string(database(),
$user_name));
        $user_phone =
htmlspecialchars(mysqli_real_escape_string(database(),
$user_phone));
        $user_adress =
htmlspecialchars(mysqli_real_escape_string(database(),
$user_adress));

        $request_sql = "INSERT INTO info_users
(`user_email_addres`, `password`, `user_name`,
`user_phone`, `user_adress`)
            VALUES ('{$user_email_addres}',
'{$password_in_md5_form}', '{$user_name}', '{$user_phone}',
'{$user_adress}')";

        $database_result = database()->query($request_sql);

        if ($database_result) {
            $request_sql = "SELECT * FROM info_users WHERE
(`user_email_addres` = '{$user_email_addres}' and
`password` = '{$password_in_md5_form}')) LIMIT 1";
            $database_result = database()->query($request_sql);
            $database_result =
generate_array_of_smarty_records($database_result);

            if (isset($database_result [0])) {
                $database_result ['success'] = 1;
            }else {
                $database_result ['success'] = 0;
            }
        }else {
            $database_result ['success'] = 0;
        } return $database_result;}

```

Для выхода из аккаунта пользователя функция login_outAction(Листинг 2.22)

```

function login_outAction() {
    if(isset($_SESSION['session_user'])) {
        unset($_SESSION['session_user']);unset($_SESSION['session_ca
rt']);
        }main_page_ridirecting('/');}

```

Авторизация реализована аналогично регистрации, функция из main.js login собирает поля из формы авторизации, затем обращается к login_usersAction и в этой функции уже выполняется авторизация с помощью

функции login_users(Листинг 2.23) и после наш ајах получает результат выполнения.

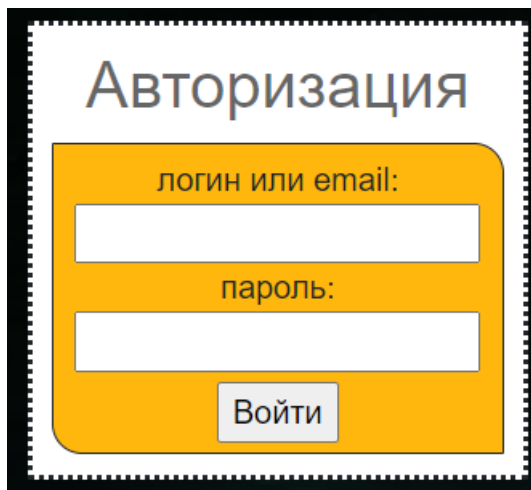
The image shows a web form for authentication. At the top, the title 'Авторизация' is displayed in a large, bold, blue font. Below the title, there is a yellow rectangular box with rounded corners. Inside this box, the text 'логин или email:' is followed by a white input field. Below this, the text 'пароль:' is followed by another white input field. At the bottom of the yellow box, there is a white button with the text 'Войти' in black. The entire form is enclosed in a black border with a dashed inner line.

Рисунок 2.10 – Форма для авторизации

Листинг 2.23 – Функция login_users, отвечающая за проверку наличия пользователя в базе данных

```
function login_users($user_email_addres, $password) {  
    $user_email_addres  
    htmlspecialchars(mysqli_real_escape_string(database(),  
    $user_email_addres));  
    $password = md5($password);  
    $request_sql = "SELECT * FROM `info_users` WHERE  
    (`user_email_addres` = '{ $user_email_addres}' AND  
    `password` = '{ $password}'))  
    LIMIT 1";  
    $database_result = database()->query($request_sql);  
    $database_result  
    generate_array_of_smarty_records($database_result);  
  
    if (isset($database_result [0])){  
        $database_result ['success'] = 1;  
    }else{  
        $database_result ['success'] = 0;  
    }  
    return $database_result;  
}
```

2.7 Изменение данных пользователя

Пользователь имеет свой профиль в интернет-магазине со своими данными, которые он может редактировать.

Профиль

Логин (email)
123

Имя

Телефон

Адрес

Новый пароль

Повторите пароль

Текущий пароль

Рисунок 2.11 – Профиль пользователя

Пользователь в любой момент может изменить данные о себе, обновлять данные мы будем с помощью функции `data_user_update` (Листинг 2.24) из `UserInfoTableModel.php`.

Листинг 2.24 – Функция внесения обновлённых данных в таблицу

```
function data_user_update($user_name, $user_phone,
    $user_address, $first_password, $second_password,
    $current_password) {
    $user_email_address =
    htmlspecialchars(mysqli_real_escape_string(database(),
    $_SESSION['session_user']['user_email_address']));
    $user_name =
    htmlspecialchars(mysqli_real_escape_string(database(),
    $user_name));
    $user_phone =
    htmlspecialchars(mysqli_real_escape_string(database(),
    $user_phone));
    $user_address =
    htmlspecialchars(mysqli_real_escape_string(database(), $user_
    address));
    $first_password = trim($first_password);
    $second_password =
    trim($second_password); $new_user_password = null;
    if ($first_password && ($first_password ==
    $second_password)) {$new_user_password =
    md5($first_password);}
    $request_sql = "UPDATE info_users SET ";
```

```

        if($new_user_password) {
            $request_sql .= "`password` = '{$new_user_password}',
";
        }
        $request_sql .= "`user_name` = '{$user_name}',
            `user_phone` = '{$user_phone}',
            `user_adress` = '{$user_adress}'
            WHERE `user_email_adress` =
 '{$user_email_adress}' AND `user_password` =
 '{$current_password}' LIMIT 1";
        $database_result = database()->query($request_sql);
        return $database_result;
    }
}

```

К данной функции мы аналогично регистрации и авторизации будем обращаться через событие onclick, которое вызывает свою функцию data_user_update из main.js, которая с помощью ajax-запроса будет обращаться к updateAction(Листинг 2.25) и из неё уже вызывается функция data_user_update(Листинг 2.24).

Листинг 2.25 – Функция обновления данных о пользователе

```

function updateAction() {
    if(!isset($_SESSION['session_user'])) {
        main_page_riderecting('/');
    }
    $result_data = array();
    $user_phone = isset($_REQUEST['user_phone']) ?
$_REQUEST['user_phone'] : null;
    $user_adress = isset($_REQUEST['user_adress']) ?
$_REQUEST['user_adress'] : null;
    $user_name = isset($_REQUEST['user_name']) ?
$_REQUEST['user_name'] : null;
    $first_password = isset($_REQUEST['first_password'])
? $_REQUEST['first_password'] : null;
    $second_password =
isset($_REQUEST['second_password']) ?
$_REQUEST['second_password'] : null;
    $current_password =
isset($_REQUEST['current_password']) ?
$_REQUEST['current_password'] : null;
    $current_password_in_md5_form = md5($current_password);
    if(!$current_password ||
($_SESSION['session_user']['user_password']) !=
$current_password_in_md5_form) {
        $result_data['success'] = 0;
        $result_data['message'] = 'Введён неверный текущий
пароль';
        echo json_encode($result_data);
        return false;
    }

    $result = data_user_update($user_name, $user_phone,
$user_adress, $first_password, $second_password,
$current_password_in_md5_form);
}

```

```

        if($result) {
            $result_data['success'] = 1;
            $result_data['message'] = 'Данные сохранены';
            $result_data['name_of_user'] = $user_name;

            $_SESSION['session_user']['user_name'] = $user_name;
            $_SESSION['session_user']['user_phone'] =
$user_phone;
            $_SESSION['session_user']['user_adress'] =
$user_adress;
            $new_user_password =
$_SESSION['session_user']['user_password'];
            if($first_password && ($first_password ==
$second_password)){
                $new_user_password = md5(trim($first_password));
            }
            $_SESSION['session_user']['user_password'] =
$new_user_password;
            If($user_name){
                $_SESSION['session_user']['name_of_display'] = $user_name ;
            }else{
                $_SESSION['session_user']['name_of_display'] =
                $_SESSION['session_user']['user_email_adress '];
            }}else {
                $result_data['success'] = 0;
                $result_data['message'] = 'Данные не сохранены
ошибка';
            }echo json_encode($result_data);}

```

2.8 Реализация поиска товаров

Поиск реализован по двум полям, он ищет по совпадению в названии и по совпадению в описании предмета, функция отвечающая за поиск search(Листинг 2.26). Также поиск может искать сразу несколько вещей перечисленных через пробел.

Листинг 2.26 – Функция поиска

```

function search(){
    if(isset($_POST['my_submit'])){
        $search = explode(" ", $_POST['search']);
        $count = count($search);
        $array = array();
        $i = 0;
        foreach($search as $key){
            $i++;
            if($i < $count){
                $array[] = "CONCAT (name, description) LIKE
'%" . $key . "%' OR ";
            }else

```



```

        $array[] = "CONCAT (name, description) LIKE
        '%" . $key . "%'";
    }

    $request_sql = "SELECT * FROM table_of_products WHERE
    ".implode($array, " ");
    $database_result = database()->query($request_sql);

    return
    generate_array_of_smarty_records($database_result);
    }
}

```

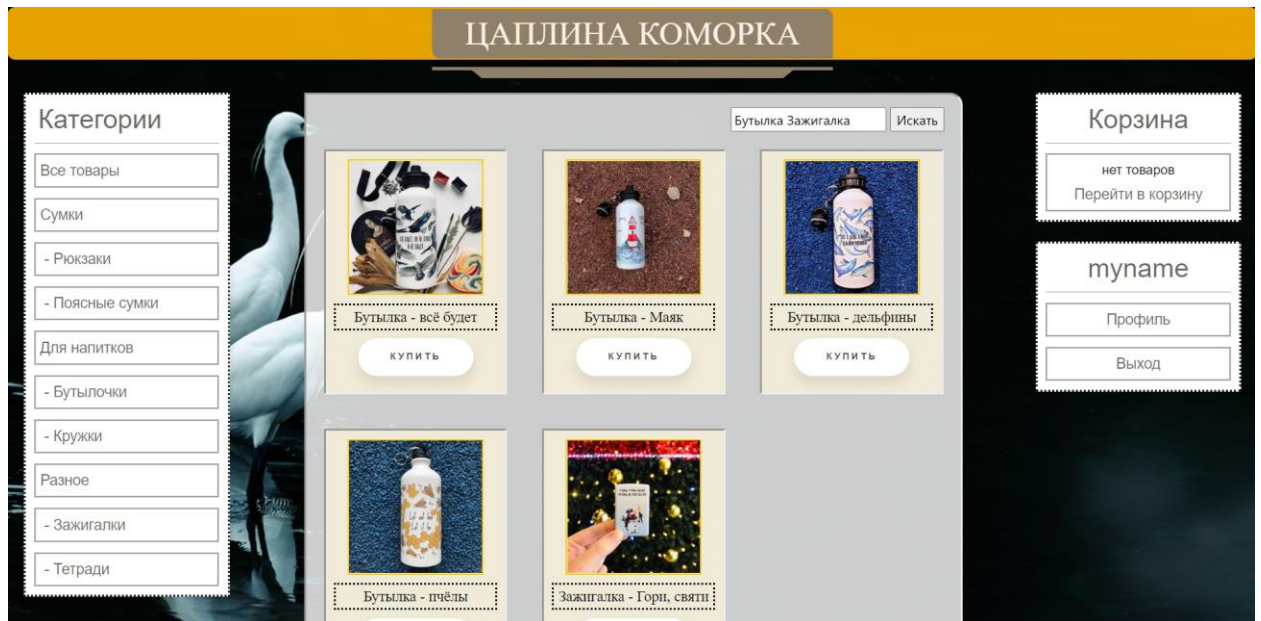


Рисунок 2.12 – Пример поиска по словам Бутылка и Зажигалка

2.9 Построение страниц с помощью шаблонизатора

Построение каждой страницы для сайта основано на коде html и шаблонизаторе smarty, для более упрощённой работы с ним была создана функция `loading_template` (Листинг 2.27).

Листинг 2.27 – Функционал загрузки шаблонов

```

function loading_templates($smarty, $name_of_template) {
    $smarty->display($name_of_template
    .Postfix_of_template);
}

```

Листинг 2.28 – Пример построения левого меню с помощью шаблонизатора, с помощью встроенной функции Smarty – `assign`, мы передаём `all_records_categories` на нужные страницы и уже шаблонизатор с помощью цикла прокручивает элементы и мы получаем меню

```

function indexAction($smarty){

    $all_records_categories = get_absolutely_all_categories ();

```

```

$smarty->assign('site_title', 'Цаплина коморка');
$smarty->assign('all_records_categories',
    $all_records_categories);
$smarty->assign('all_products_records',
    $all_products_records);

Loading_templates($smarty, 'site_header');
Loading_templates($smarty, 'index');

<div id="category_column">
<h3>Категории</h3>
<ul>
<li><a href="/" class="">Все товары</a></li>
{foreach $all_records_categories as $item}
<li><a
href="/?controller=category&id={$item['id']}">{$item['name']}
</a></li>

{if isset($item['children'])}
{foreach $item['children'] as $children_category}
<li><a
href="/?controller=category&id={$children_category['id']}">-
{$children_category['name']}</a></li>
{/foreach}{/if}
{/foreach}</div>

```

2.10 Оформление заказов

После перехода в корзину пользователь видит всё что он добавил в корзину, и он может редактировать количество товаров. Корзина формируется в файле cartController.php в функции orderAction.

Корзина

Данные заказа:

| № | Наименование | Количество | Цена за единицу | Цена | Действие |
|---|-----------------------|------------|-----------------|------|-------------------------|
| 1 | Бутылка - всё будет | 1 | 1000 | 1000 | Удалить |
| 2 | Тетрадь - Рисуй, пиши | 6 | 450 | 2700 | Удалить |

[Оформить заказ](#)

Рисунок 2.13 – Пример корзины

Далее происходит оформления заказа после нажатия на кнопку “оформить заказ” срабатывает onclick, который вызывает функцию order_save, где с помощью ajax-запроса мы будем обращаться к cartController.php и к save_orderAction внутри данной функции будет происходить формирование результата оформления заказа, с помощью функций create_new_order(Листинг 2.30) и purchase_details(Листинг 2.31) мы будем вносить данные о заказах в наши таблицы.

Листинг 2.30 – Функция create_new_order, которая добавляет информацию о заказе в таблицу

```
function create_new_order($user_name, $user_phone,
    $user_address) {$id_user = $_SESSION['session_user']['id'];


    $order_to_comment = "id пользователя: {$id_user}<br/>Имя
    пользователя: {$user_name}<br/>Тел {$user_phone}<br/>Адрес пользователя: {$user_address}";

    $ip_user = $_SERVER['REMOTE_ADDR'];
    $date_of_create = date('Y.m.d. H:i:s');

    $request_sql = "INSERT INTO table_of_orders (id_user,
    created, payment, status, order_to_comment, ip_user)
    VALUES ('{$id_user}', '{$date_of_create}', null,
    '0', '{$order_to_comment}', '{$ip_user}')";
    $database_result = database()->query($request_sql);
    if($database_result) {
        $request_sql = "SELECT id FROM table_of_orders ORDER
    BY id DESC LIMIT 1";
        $database_result = database()->query($request_sql);
        $database_result = generate_array_of_smarty_records($database_result);
        if(isset($database_result[0])) {
            return $database_result[0]['id'];
        }
    }
    return false;}
```

Листинг 2.31 – Функция purchase_details, которая сохраняет товары для созданного заказа в таблицу

```
function purchase_details($id_order, $session_cart) {
    $sql_request = "INSERT INTO table_of_purchase(order_id,
    product_id, price, amount)
    VALUES ";
    $products = array();
    foreach ($session_cart as $item) {
        $products[] = "({'{$id_order}', '{$item['id']}',
    '{$item['price']}', '{$item['count']}'})";
    }
    $sql_request .= implode($products, ',');
    $database_result = database()->query($sql_request);
    return $database_result;}
```



Данные заказа:

| № | Наименование | Количество | Цена за единицу | Стоимость |
|---|-------------------------|------------|---------------------|-------------|
| 1 | Кружка - Агент | 1 | 1000 | 1000 |
| 2 | Зажигалка - Гори, святи | 3 | 899 | 2697 |
| | | | Общая сумма: | 3697 |

Данные заказчика

Введите или обновите данные заказа ниже

ФИО полностью*:

Номер телефона*:

Мой адрес

Адрес доставки*:

Рисунок 2.14 – Пример страницы заказа

2.11 Настройка административной панели

Доступ в админку настроен через настройки файла .htaccess.

Далее с помощью утилиты htpasswd.exe мы задаём пароль, который сгенерирует файл pwdaccess с паролем. С помощью этой утилиты можно создать и других пользователей административной панелью. Для попадания в саму административную панель нужно к адресу сайта добавить /admin .

Войдите в систему, чтобы получить доступ к этому сайту

Требуется авторизация
Подключение к этому сайту не защищено.

Имя пользователя

Пароль

Вход **Отмена**

Рисунок 2.15 – Вход в административную панель

Меню:
[Открыть магазин](#)
[Информация](#)
[Категории](#)
[Товар](#)
[Заказы](#)

Редактирование сайта

Это административная панель магазина, здесь вы можете редактировать ваш магазин
В меню слева вы можете выбрать, что хотите изменить или посмотреть
Вы сейчас находитесь в разделе информация

Рисунок 2.16 – Административная панель, раздел информация

2.11.1 Редактирование категорий

Редактирование категорий, а именно переименование их или редактирование родительской категории происходит с помощью JavaScript функции `category_update`, которая вызывает аjax-запрос, который в свою очередь в файле `adminController.php` вызывает функцию `category_updateAction`, где внутри этой функции будет вызвана функция `category_data_update` (Листинг 2.33)

Листинг 2.33 – Функция category_data_update, которая меняет наименование категории или её родительскую категорию

```
function category_data_update($item_id, $id_of_parent = -1,
    $new_category_name = '') {
    $set = array();
    if($new_category_name)
        $set[] = "`name` = '{$new_category_name}'";

    if($id_of_parent > -1) {
        $set[] = "`parent_id` = '{$id_of_parent}'";
    }
    $string_value = implode($set, ', ');
    $request_sql = "UPDATE tables_of_category SET
{$string_value} WHERE id = '{$item_id}'";
    $database_result = database()->query($request_sql);
    return $database_result;
}
```

2.11.2 Работа с товарами

Для получения списка продуктов разработана функция get_all_products(Листинг 2.34). Для добавление нового товара была разработана функция insert_new_product(Листинг 2.35), которую вызывает функция из admin.js add_product.

Листинг 2.34 – Функция get_all_products

```
function get_all_products() {
    $sql_request = "SELECT * FROM table_of_products ORDER BY
category_id";
    //echo "test";
    $database_result = database()->query($sql_request);
    return
generate_array_of_smarty_records($database_result);
}
```

Листинг 2.35 – Функция insert_new_product, которая вносит наш новый продукт в базу данных

```
function insert_new_product($name_of_product,
    $price_of_item, $description_of_item, $category_of_item) {
    $request_sql = "INSERT INTO table_of_products SET
                                name          =
'{$name_of_product}',
                                price         =
'{$price_of_item}',
                                description    =
'{$description_of_item}',
                                category_id   =
'{$category_of_item}'";
    $database_result = database()->query($request_sql);
    return $database_result;
}
```

Обновление данных о продуктах происходит в функции `product_update_info`(Листинг 2.36).

Листинг 2.36 – Функция обновления данных о продуктах

```
function product_update_info($item_id, $name_of_product,
    $price_of_item, $itemStatus, $description_of_item,
    $category_of_item, $file_name = null) {
    $set = array();
    //echo "test";
    if($name_of_product) {
        $set[] = " name = '{$name_of_product}'";
    }
    if ($category_of_item) {
        $set[] = "category_id = '{$category_of_item}'";
    }
    if ($file_name) {
        $set[] = "image = '{$file_name}'";
    }

    if($price_of_item>0) {
        $set[] = "price = '{$price_of_item}'";
    }

    if ($itemStatus !== null) {
        $set[] = "status = '{$itemStatus}'";
    }

    $string_value = implode($set, ', ');
    $request_sql = "UPDATE table_of_products SET
{$string_value}
    WHERE `id` = '{$item_id}'";

    $database_result = database()->query($request_sql);

    return $database_result;
}
```

Загрузка же изображения происходит в отдельной функции `file_loading`(Листинг 2.37), которая вызывается из `uploadAction` из файла `adminController.php` и функции `image_product_update`(Листинг 2.38).

Листинг 2.37 – Функция загрузки файла

```
function file_loading($filename_loc, $path = '/upload/') {

    $max_size_of_picture = 2 * 1024 * 1024;
```

```

    $ext = pathinfo($_FILES['filename']['name'],
PATHINFO_EXTENSION);

    $info_path = pathinfo($filename_loc);

    if($ext != $info_path['extension']) return false;
    $file_name = $info_path['filename'] . '_' . time() . '.'
. $info_path['extension'];

    if($_FILES["filename"]["size"] > $max_size_of_picture) {
        return false;
    }
    $path = $_SERVER['DOCUMENT_ROOT'] . $path;
    if(!file_exists($path)){
        mkdir($path);
    }
    if(is_uploaded_file($_FILES['filename']['tmp_name'])){
        $result =
move_uploaded_file($_FILES['filename']['tmp_name'], $path .
$file_name);
        return ($result == true) ? $file_name : false;
    }else {
        return false;
    }
}

```

Листинг 2.38 – Функция обновления изображения товара в базе данных

```

function image_product_update($item_id, $file_name) {

    $database_result = product_update_info($item_id, null,
null, null, null, null, $file_name);

    return $database_result;
}

```

Функция удаление товара deleteProduct(Листинг 2.39).

Листинг 2.39 – Удаление товара из базы данных.

```

function deleteProduct($item_id) {
    $request_sql = "DELETE FROM table_of_products WHERE `id`
= '{$item_id}'";

    $database_result = database()->query($request_sql);
    return $database_result;
}

```


3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Для того чтобы попасть на сайт нужно перейти по локальному адресу `tsaplinacloset.local`, далее вы попадаете на главную страницу сайта

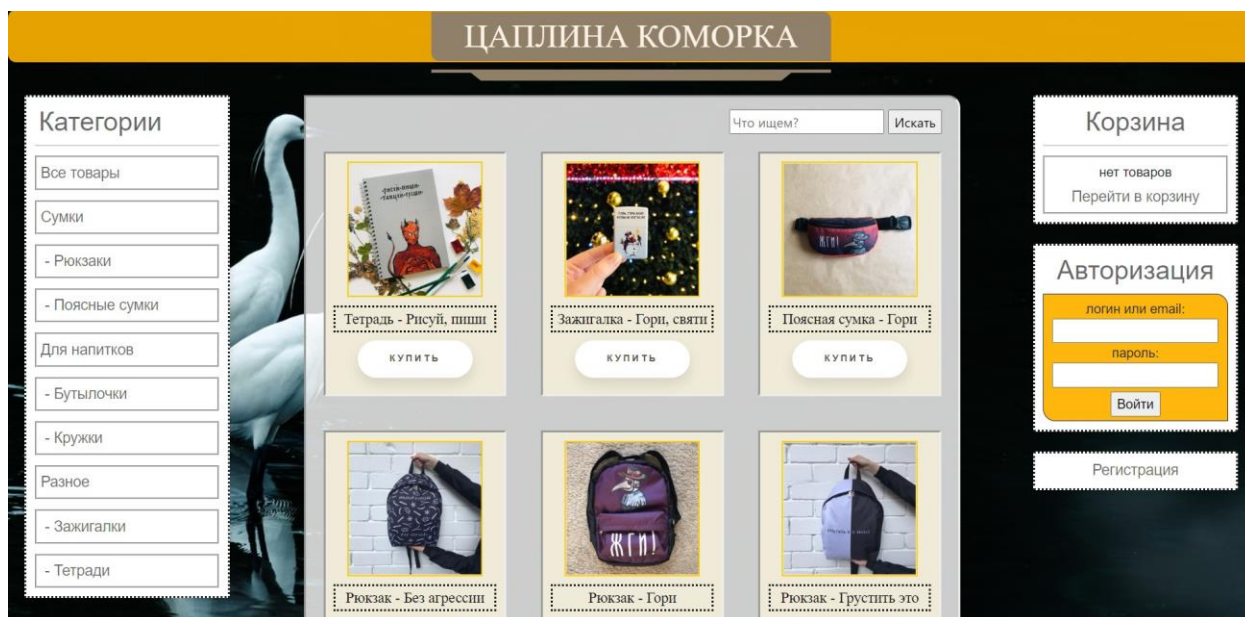


Рисунок 3.1 – Главная страница сайта

На главной странице пользователь может сделать 6 действий:

1) Пользователь может выбрать категорию товара, которую он хочет посмотреть для этого в левом меню ему нужно нажать на интересующую его категорию, на рисунке 3.2 список категорий выделен зелёной рамкой.

2) Пользователь может купить/посмотреть товар для этого ему нужно нажать на изображение нужного товара или на кнопку купить, на рисунке 3.2 товары выделены красной рамкой.

3) Пользователь может перейти в корзину для этого ему нужно нажать на кнопку перейти в корзину, на рисунке 3.2 корзина выделена синей рамкой.

4) Пользователь может авторизоваться для этого ему нужно уже иметь аккаунт в данном интернет-магазине и в поля “логин или email” и пароль ввести соответствующие данные, а затем нажать кнопку “войти”, на рисунке 3.2 поле авторизации выделено фиолетовым цветом

5) Пользователь может воспользоваться поиском по товарам для этого ему нужно в поле с подписью “Что ищем?” ввести какую-то информацию, а затем нажать на кнопку “Искать”, данный поиск ищет совпадения ведёной информации с названиям существующих товаров, а также описания существующих товаров, на рисунке 3.2 поле поиска выделено серым цветом

6) Пользователь может зарегистрироваться для этого ему нужно нажать на кнопку “Регистрация”, далее для пользователя выведутся поля, которые пользователь должен заполнить, а именно “логин или email”, “пароль”, “повторите пароль”, после заполнения всех полей нужно нажать на кнопку “Зарегистрироваться” и при корректности введённых данных пользователь

будет зарегистрирован в системе. На рисунке 3.2 форма регистрации отмечена оранжевой рамкой.

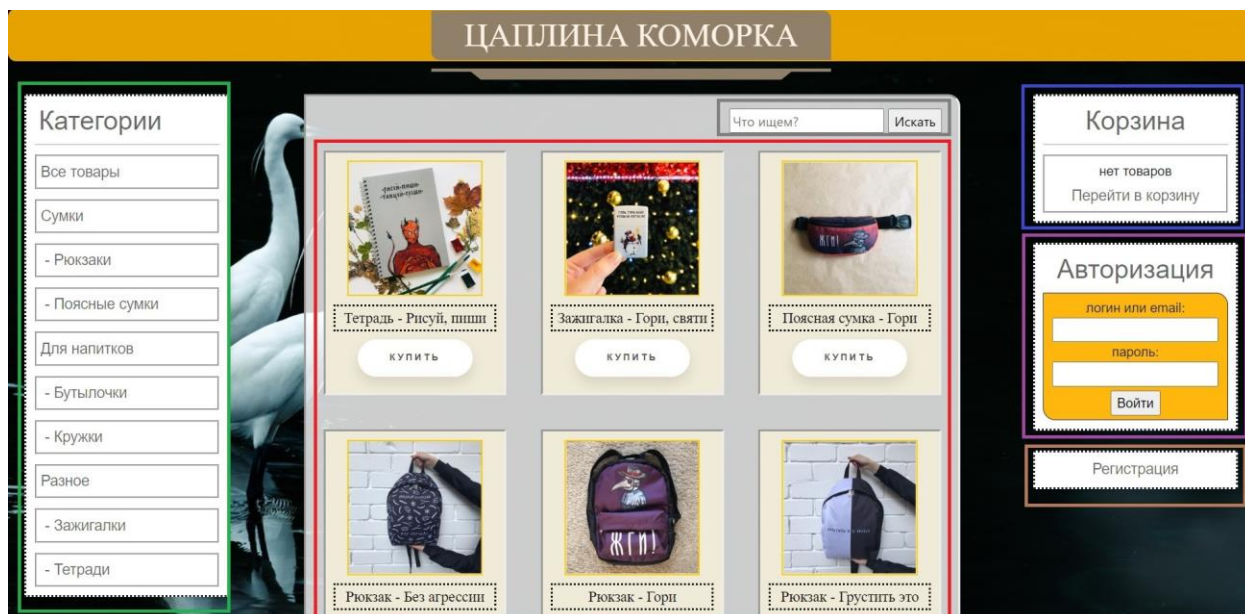


Рисунок 3.2 – Рисунок выбора действий, цветами обозначен, что может сделать пользователь

Для совершения покупки пользователю необходимо:

1) Нажать на изображение интересующего товара или нажать на кнопку “купить” рядом с интересующим товаром.

2) Далее на странице товара нужно нажать кнопку “добавить в корзину”, для отмены добавления нужно нажать кнопку “удалить из корзины”.

3) После добавления товара в корзину, нужно перейти в корзину нажав на кнопку “Перейти в корзину” на рисунке 3.2 корзина выделена синей рамкой.

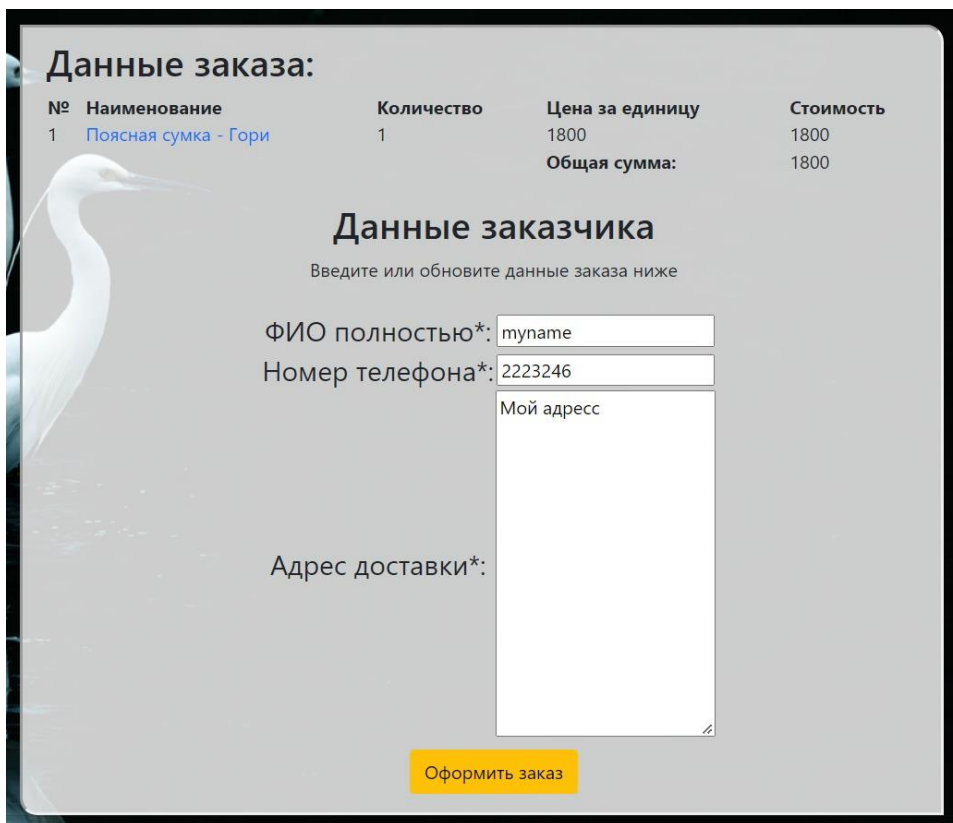
4) Затем в поле “количество” нужно ввести количество необходимого товара и после этого нажать на кнопку “оформить заказ”, также при желании отменить заказ на странице присутствует кнопку “удалить”, которая удалит данный товар.

5) После нажатия на кнопку “Оформить заказ” вы попадаете на страницу с данными вашего заказа, сайт предложит вам одно из двух действий:

5.1) Если вы были авторизованы или авторизовались на сайте, то сайт предложит вам при необходимости изменить свои данные (Рисунок 3.3), для того чтоб изменить данные введите в соответствующие поля новые данные и нажмите оформить заказ, ваши данные передадутся и заказ будет сохранён.

5.2) Если вы не были авторизованы до покупки (Рисунок 3.4), то сайт предложит вам либо авторизоваться в существующий аккаунт, либо провести регистрацию, если вы хотите авторизоваться введите в

соответствующие поля данные и затем нажмите кнопку “войти”, если вы хотите зарегистрироваться введите в соответствующие поля нужные данные и нажмите кнопку “зарегистрироваться” после вас перенаправит к пункту 5.1



Данные заказа:

| № | Наименование | Количество | Цена за единицу | Стоимость |
|---|----------------------|------------|-----------------|-----------|
| 1 | Поясная сумка - Гори | 1 | 1800 | 1800 |
| | | | Общая сумма: | 1800 |

Данные заказчика
Введите или обновите данные заказа ниже

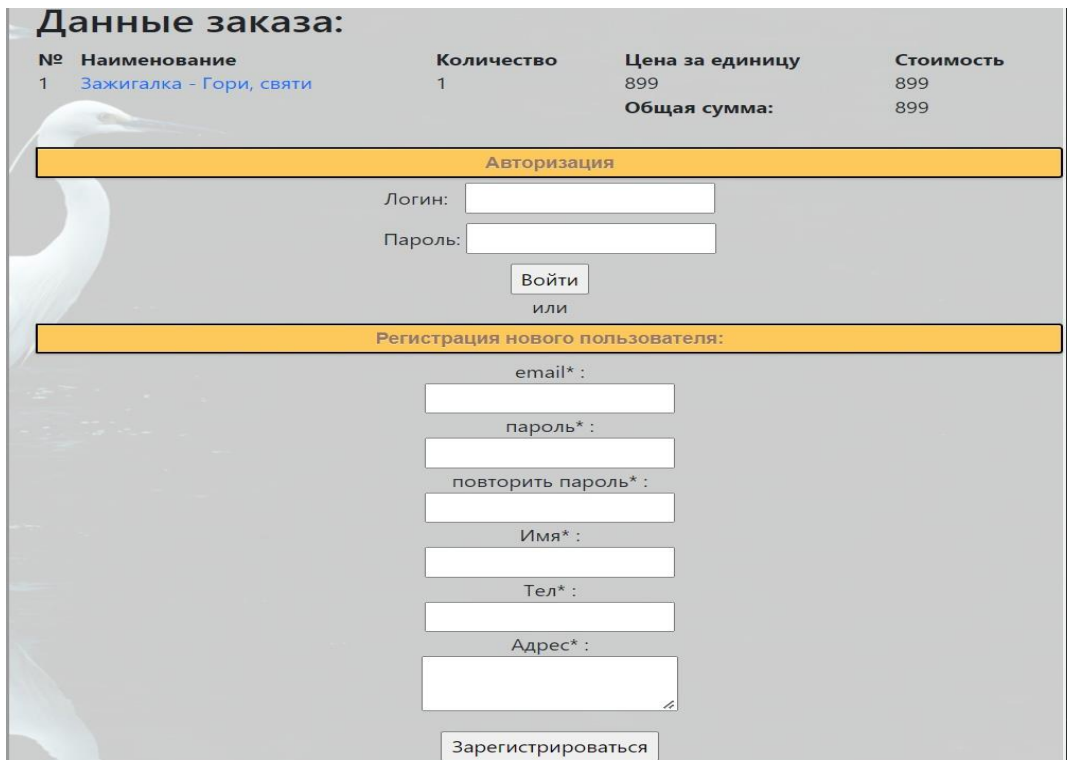
ФИО полностью*:

Номер телефона*:

Мой адрес:

Адрес доставки*:

Рисунок 3.3 – Форма оформления заказа



Данные заказа:

| № | Наименование | Количество | Цена за единицу | Стоимость |
|---|-------------------------|------------|-----------------|-----------|
| 1 | Зажигалка - Гори, святы | 1 | 899 | 899 |
| | | | Общая сумма: | 899 |

Авторизация

Логин:

Пароль:

или

Регистрация нового пользователя:

email* :

пароль* :

повторить пароль* :

Имя* :

Тел* :

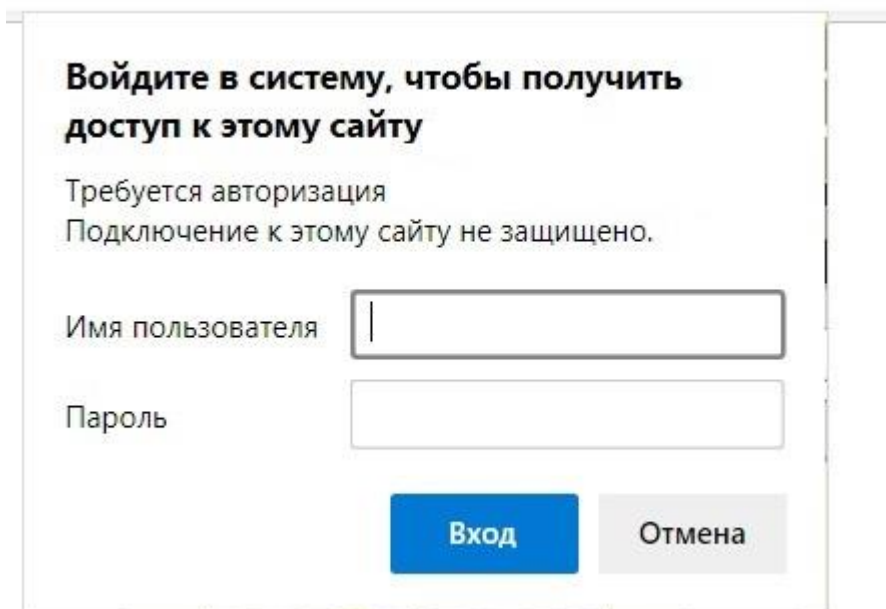
Адрес* :

Рисунок 3.4 – Предложение войти или зарегистрироваться для неавторизированных пользователей.

При желании в профиле можно изменить свои данные аккаунта для этого в авторизированном состоянии нажмите кнопку “профиль”, затем поменяйте данные, которые вам нужны и для сохранения этих данных заполните поле “текущий пароль” своим текущим паролем и нажмите кнопку “сохранить данные”.

4. РУКОВОДСТВО АДМИНИСТРАТОРА

Для перехода в административную панель нужно перейти по ссылке `tsaplinacloset.local/admin` . Далее нужно будет обязательно авторизоваться под пользователем `admin` и ввести пароль в соответствующие поля (Рисунок 4.1)



The screenshot shows a login window with the following elements:

- Header:** Войдите в систему, чтобы получить доступ к этому сайту
- Text:** Требуется авторизация, Подключение к этому сайту не защищено.
- Form Fields:**
 - Имя пользователя: A text input field with a vertical cursor.
 - Пароль: A password input field.
- Buttons:** Вход (blue) and Отмена (grey).

Рисунок 4.1 – Авторизация для административной панели

После авторизации вы попадаете в административную панель (Рисунок 4.2)



The screenshot shows the administrative panel interface:

- Section Header:** Редактирование сайта
- Left Menu:**
 - Меню:
 - Открыть магазин
 - Информация
 - Категории
 - Товар
 - Заказы
- Main Content Area:**
 - Это административная панель магазина, здесь вы можете редактировать ваш магазин
 - В меню слева вы можете выбрать, что хотите изменить или посмотреть
 - Вы сейчас находитесь в разделе информация

Рисунок 4.2 – Административная панель

Слева в административной панели находится меню, оно содержит 5 кнопок:

1) Кнопка “Открыть магазин” – кнопка возвращает вас на главную страницу магазина

2) Кнопка “Информация” – кнопка переносит вас на страницу с информацией, по умолчанию это главная страница административной панели

3) Кнопка “Категории” – кнопка, по переходу на которую вы попадаете на страницу, где можно редактировать категории (Рисунок 4.3). Редактировать можно два поля “Название” и “Родительская категория”, для редактирования “Названия” в поле с названием категории измените его на нужное, а затем нажмите на кнопку “сохранить”. Для редактирования “Родительская категория” нажмите на выпадающий список категорий, выберите нужную категорию, а затем нажмите кнопку “сохранить”.

4) Кнопка “Товар” – кнопка, по переходу на которую вы попадаете на страницу, где можно добавить или изменить товар. (Рисунок 4.4)

4.1) Для добавления товара в разделе “добавить продукт” заполните нужные поля, а затем нажмите на кнопку “добавить”

4.2) Для редактирования товара в разделе “редактировать” заполните или измените нужные поля и нажмите кнопку “сохранить”. Для загрузки изображения нажмите на кнопку “выбор файла”, а затем нажмите кнопку “загрузить”.

5) Кнопка “Заказы” – кнопка, по нажатию на которую попадаешь на страницу, где отображены все заказы пользователей. (Рисунок 4.5)

| Категории | | | | |
|-----------|----|---------------|------------------------|-----------|
| № | ID | Название | Родительская категория | Действие |
| 1 | 1 | Сумки | Главная категория | Сохранить |
| 2 | 2 | Для напитков | Главная категория | Сохранить |
| 3 | 3 | Разное | Главная категория | Сохранить |
| 4 | 4 | Рюкзаки | Сумки | Сохранить |
| 5 | 5 | Поясные сумки | Сумки | Сохранить |
| 6 | 6 | Бутылочки | Для напитков | Сохранить |
| 7 | 7 | Кружки | Для напитков | Сохранить |
| 8 | 8 | Зажигалки | Разное | Сохранить |
| 9 | 9 | Тетради | Разное | Сохранить |

Рисунок 4.3 – Раздел “Категории”

Товар

Добавить продукт

| Название | Цена | Категория | Описание | Сохранить |
|----------------------|----------------------|--------------------|----------------------|---|
| <input type="text"/> | <input type="text"/> | Выберите категорию | <input type="text"/> | <input type="button" value="Добавить"/> |

Редактировать



| № | ID | Название | Цена | Категория | Описание | Изображение | Сохранить | Удалить |
|---|----|----------------|-------|-----------|--------------------------------------|---|--|--|
| 1 | 10 | Рюкзак - Без а | 10057 | Рюкзаки | Описание временно изменено администр |  <div> <input type="button" value="Выбор файла"/> Не выбран ... один файл </div> <input type="button" value="Загрузить"/> | <input type="button" value="Сохранить"/> | <input type="button" value="Удалить"/> |
| 2 | 8 | Рюкзак - Груст | 1000 | Рюкзаки | <rp>Рюкзак : грустить это весело. |  | <input type="button" value="Сохранить"/> | <input type="button" value="Удалить"/> |

Рисунок 4.4 – Раздел редактирования/добавления товара

Заказы

| № | Действие | ID заказа | Статус | Дата создания | Статус оплаты | Дополнительная информация | Дата изменения заказа |
|---|---------------------------------------|-----------|--------------|---------------------|---------------|---|-----------------------|
| 1 | Показать товар заказа | 37 | Заказ открыт | 2022-06-08 00:00:00 | Неоплачено | id пользователя: 20 Имя: mupame Тел: 2223246 Адрес: Мой адресс | 2022-06-08 23:35:26 |
| 2 | Показать товар заказа | 36 | Заказ открыт | 2022-06-08 00:00:00 | Неоплачено | id пользователя: 20 Имя: mupame Тел: 2223246 Адрес: Мой адресс | 2022-06-08 21:16:17 |
| 3 | Показать товар заказа | 35 | Заказ открыт | 2022-04-27 00:00:00 | Неоплачено | id пользователя: 59 Имя: афырп Тел: 521 Адрес: а32 | 2022-04-30 15:23:02 |
| 4 | Показать товар заказа | 34 | Заказ открыт | 2022-04-24 00:00:00 | Неоплачено | id пользователя: 56 Имя: gqwfas Тел: 51261 Адрес: gasga | 2022-04-30 15:26:57 |
| 5 | Показать товар заказа | 33 | Заказ открыт | 2022-04-24 00:00:00 | Неоплачено | id пользователя: 56 Имя: gqwfas Тел: 51261 Адрес: gasga | 2022-04-30 15:06:37 |

Рисунок 4.5 – Раздел заказы

ЗАКЛЮЧЕНИЕ

При разработке “Интернет-магазина авторских вещей” были рассмотрены различные средства разработки, возможные варианты построения подобных интернет-магазинов, и соответствующие инструменты.

Поставленные задачи были выполнены в должном объёме, интернет-магазин функционирует, основные функции интернет-магазина были реализованы.

Я приобрёл хороший опыт разработки сайтов, который обязательно пригодится мне в будущем, а также улучшил свои знания по их реализации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Дэвид Скляр. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов.: Пер. с англ. - СПб.: ООО "Альфа-книга", 2017. - 464 с. : ил. - Парал. тит. англ.
- 2) Руководство: Smarty [Электронный ресурс] // Smarty – URL: <https://www.smarty.net/docsv2/ru/>
- 3) HTML начало [Электронный ресурс] // wm-school – URL: https://wm-school.ru/html/html_start.html
- 4) Создание PHP + MySql сайта [Электронный ресурс] // Хабр – URL: <https://habr.com/ru/sandbox/14484/>
- 5) Этан Браун. Изучаем JavaScript: руководство по созданию современных веб-сайтов, 3-е изд.: Пер. с англ. - СПб.: ООО "Альфа-книга"; 2017. - 368 с. : ил. - Парал. тит.
- 6) Руководство: Русская документация по API jQuery [Электронный ресурс] // jquery-docs – URL: <https://jquery-docs.ru/>
- 7) Основы CSS [Электронный ресурс] // html5book.ru – URL: <https://html5book.ru/osnovy-css/>
- 8) AJAX для новичков [Электронный ресурс] // Хабр – URL: <https://habr.com/ru/post/14246/>
- 9) PHP: объекты, шаблоны и методики программирования, 4-е изд. Пер. с англ. -М.: ООО "И.Д. Вильямс", 2015. - 576 с.: ил. - Парал. тит. англ.
- 10) Учебник по языку SQL (DDL, DML) на примере диалекта MS SQL Server. Часть первая [Электронный ресурс] // Хабр – URL: <https://habr.com/ru/post/255361/>
- 11) Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж. Г89 SQL: полное руководство, 3-е изд. : Пер. с англ. -М.: ООО "И.Д. Вильямс", 2015. -960 с. : ил. - Парал. тит. англ.
- 12) PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред.Е. В. Рогова, П. В. Лузанова. — СПб.: БХВ-Петербург, 2018. — 336 с.: ил.
- 13) Как вывести данные из MySQL – руководство не для шаманов [Электронный ресурс] // webformymself – URL: <https://webformymself.com/kak-vyvesti-dannye-iz-mysql-rukovodstvo-dlya-ne-shamanov/>
- 14) Документация MySQL [Электронный ресурс] // MySql – URL: <http://www.mysql.ru/docs/gruber/mg03.html>

ПРИЛОЖЕНИЕ А

Наиболее употребляемые текстовые сокращения

JS - JavaScript

СибГУТИ — Сибирский
государственный университет
телекоммуникаций и информатики

ПРИЛОЖЕНИЕ Б

Листинг программы

Листинг Б.1 – файл adminController.php

```
<?php
include_once '../model/IncludeModel.php';

$smarty->setTemplateDir(admin_prefix);
$smarty->assign('web_path', admin_web_path);

function indexAction($smarty) {

    $all_records_categories = get_all_categories();

    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('site_title', 'Редактирование сайта');
    loading_template($smarty, 'admin_header');
    loading_template($smarty, 'admin_main');
}

function categoryAction($smarty) {
    $records_of_all_category = get_full_list_category();
    $records_of_main_category = get_absolutely_main_categories();

    $smarty->assign('site_title', 'Редактирование сайта');
    $smarty->assign('all_records_categories', $records_of_all_category);
    $smarty->assign('records_of_main_category',
$records_of_main_category);

    loading_template($smarty, 'adminHeader');
    loading_template($smarty, 'adminCategory');
}

function category_updateAction() {
    $item_id = $_POST['item_id']; $id_of_parent = $_POST['id_of_parent'];
    $new_category_name = $_POST['newName'];
    $result = category_data_update($item_id, $id_of_parent,
$new_category_name);
    if($result) {
        $result_data['success'] = 1;
        $result_data['message'] = 'Категория была успешно изменена';
    }else{$result_data['success'] = 0;
        $result_data['message'] = 'Ошибка изменения';
    }echo json_encode($result_data);
    return;
}

function productsAction($smarty) {

    $all_records_categories = get_full_list_categories();
    $all_products_records = get_all_products();

    $smarty->assign('site_title', 'Редактирование сайта');
    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('all_products_records', $all_products_records);

    loading_template($smarty, 'adminHeader');
    loading_template($smarty, 'adminProducts');
```

```

}
function uploadAction() {
    $item_id = $_POST['item_id'];
    $upload_file_success = file_loading($_FILES['filename']['name'],
    '/images/products/');

    if (! $upload_file_success) {
        echo 'Ошибка загрузки!'; return;
    }
    $result = $_SERVER['DOCUMENT_ROOT'] . '/images/products/' .
    $upload_file_success;

    if ($result) {
        $result = product_update_infoImage($item_id,
    $upload_file_success);
        if ($result) {
            main_page_redirecting('/admin/products/');
        }
    }else{
        echo ('Ошибка загрузки файла');
    }
}

function file_loading($filename_loc, $path = '/upload/') {

    $max_size_of_picture = 2 * 1024 * 1024;
    $ext = pathinfo($_FILES['filename']['name'], PATHINFO_EXTENSION);

    $info_path = pathinfo($filename_loc);

    if($ext != $info_path['extension']) return false;
    $file_name = $info_path['filename'] . '_' . time() . '.' .
    $info_path['extension'];

    if($_FILES["filename"]["size"] > $max_size_of_picture) return false;

    $path = $_SERVER['DOCUMENT_ROOT'] . $path;

    if(!file_exists($path))mkdir($path);

    if(is_uploaded_file($_FILES['filename']['tmp_name'])) {
        $result = move_uploaded_file($_FILES['filename']['tmp_name'],
    $path . $file_name); return ($result == true) ? $file_name : false;
    }else {
        return false;
    }
}

function add_productAction() {
    $name_of_product = $_POST['itemName'];
    $price_of_item = $_POST['price_of_item'];
    $description_of_item = $_POST['description_of_item'];
    $category_of_item = $_POST['itemCatId'];
    $result = insert_new_product($itemName, $price_of_item,
    $description_of_item, $category_of_item);
    $table_of_products = get_all_products();
    if ($result) {

        $result_data['success'] = 1;
        $result_data['message'] = 'Продукт добавлен';

    }else{
        $result_data['success'] = 0;
        $result_data['message'] = 'Ошибка';
    }
}

```

```

    }

    echo json_encode($result_data);
    return;
}

function delete_productAction() {
    $item_id = isset($_GET['id']) ? intval($_GET['id']) : null;
    if (!$item_id) exit();
    $result = deleteProduct($item_id);
    if ($result) {
        $result_data['success'] = 1; $result_data['message'] = 'товар
удалён';
    } else {
        $result_data['success'] = 0; $result_data['message'] = 'ошибка
удаления';
    }
    echo json_encode($result_data);
}

function update_productAction() {
    $item_id = $_POST['item_id'];
    $name_of_product = $_POST['name_of_product'];
    $price_of_item = $_POST['price_of_item'];
    $itemStatus = $_POST['itemStatus'];
    $description_of_item = $_POST['description_of_item'];
    $category_of_item = $_POST['itemCatId'];

    $result = product_update_info($item_id, $name_of_product,
$price_of_item, $itemStatus, $description_of_item, $category_of_item);

    if ($result) {
        $result_data['success'] = 1;
        $result_data['message'] = 'Изменнено';
    } else {
        $result_data['success'] = 0;
        $result_data['message'] = 'Ошибка';
    } echo json_encode($result_data);
    return;
}

```

Листинг Б.2 – Файл cartController.php

```

<?php
include_once '../model/IncludeModel.php';

function add_to_cart_action() {

    If(isset($_GET['id'])) {
        $item_id = intval($_GET['id']);
    } else { $item_id = null; }

    if ($item_id == null) return false;
    $result_data = array();
    if (isset($_SESSION['session_cart']) && array_search($item_id,
$_SESSION['session_cart']) == false) {
        $_SESSION['session_cart'][] = $item_id;
        $result_data['count_of_items'] =
count($_SESSION['session_cart']);
        $result_data['success'] = 1;
    } else {
        $result_data['success'] = 0;
    }
    echo json_encode($result_data);
}

```

```

function remove_from_cart_action(){
If(isset($_GET['id'])){
    $item_id = intval($_GET['id']);
}else{$item_id = null;}

if ($item_id == null) exit();

$key_of_item = array_search($item_id, $_SESSION['session_cart']);
$result_data = array();
if ($key_of_item) {
unset($_SESSION['session_cart'][$key_of_item]);
$result_data['success'] = 1;
$result_data['count_of_items'] = count($_SESSION['session_cart']);
}else {
$result_data['success'] = 0;
}
echo json_encode($result_data);
}

function indexAction($smarty) {
    $idS_items = isset($_SESSION['session_cart']) ?
    $_SESSION['session_cart'] : array();

    $all_records_categories = get_absolutely_all_categories ();
    $all_products_records = get_list_of_products_from_array($idS_items);

    $smarty->assign('site_title', 'Корзина');
    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('all_products_records', $all_products_records);

    loading_template($smarty, 'site_header');
    loading_template($smarty, 'session_cart');}

function orderAction($smarty) {
    $idS_items = isset($_SESSION['session_cart']) ?
    $_SESSION['session_cart'] : null;

    if(!$idS_items){
        main_page_redirecting('/cart/');
        return;
    }
    $count_items = array();
    foreach ($idS_items as $item) {
        $postVar = 'count_of_items_' . $item;
        $count_items[$item] = isset($_POST[$postVar]) ? $_POST[$postVar]
: null;
    }
    $all_products_records = get_list_of_products_from_array($idS_items);
    $i =0;
    $totalPrice = 0;
    foreach ($all_products_records as &$item) {
        $item['count'] = isset($count_items[$item['id']]) ?
$count_items[$item['id']] : 0;

        if($item['count']) {
            $item['price_real'] = $item['count'] * $item['price'];
            $total_price += $item['price_real'];
        }else {
            unset($all_products_records[$i]);
        }
        $i++;
    }

    if(!$all_products_records) {

```

```

        echo " Корзина пуста";
        return;
    }

    $_SESSION['cart_sale'] = $all_products_records;

    $all_records_categories = get_absolutely_all_categories ();

    if(!isset($_SESSION['session_user']))
        $smarty->assign('hideLoginBox', 1);

    $smarty->assign('total_price', $total_price);
    $smarty->assign('site_title', 'Ваш заказ');
    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('all_products_records', $all_products_records);

    loading_template($smarty, 'site_header');
    loading_template($smarty, 'order');
}
function save_orderAction() {

    if ($session_cart == null) {
        $result_data['success'] = 0;
        $result_data['message'] = 'Товаров не найдено ';
        echo json_encode($result_data);
        return;
    }

    $user_name = $_POST['user_name']; $user_phone = $_POST['user_phone'];
    $user_adress = $_POST['user_adress'];
    $id_order = create_new_order($user_name, $user_phone, $user_adress);
    if(!$id_order) {
        $result_data['success'] = 0;
        $result_data['message'] = 'Ошибка создания заказа';
        echo json_encode($result_data);
        return;
    }
    $result = purchase_details($id_order, $session_cart);
    if ($result) {
        $result_data['success'] = 1;
        $result_data['message'] = 'Сохранено';
        unset($_SESSION['cart_sale']);
        unset($_SESSION['session_cart']);
    }else {
        $result_data['success'] = 0;
        $result_data['message'] = 'Ошибка внесения данных для
заказа'; }echo json_encode($result_data);}

```

Листинг Б.3 – Файл CategoryController.php

<?php

```

function indexAction($smarty) {
    If(isset($_GET['id'])){
        $category_id = $_GET['id']
    }else{
        $category_id = null;
    }
    if (!$category_id) exit();
    $all_products_records = null; $record_of_child_category = null;

    $record_of_all_category = get_category_data_by_id($category_id);

```

```

        if ($record_of_all_category['parent_id'] == 0) {
            $record_of_child_category
        } else {
            $all_products_records
        }
        get_child_category_based_on_parent($category_id);
        get_products_for_this_category($category_id);

        $all_records_categories = get_absolutely_all_categories ();

        $smarty->assign('site_title', 'Товары категории'.
        $record_of_all_category['name']);

        $smarty->assign('record_of_all_category', $record_of_all_category);
        $smarty->assign('all_products_records', $all_products_records);
        $smarty->assign('record_of_child_category',
        $record_of_child_category);

        $smarty->assign('all_records_categories', $all_records_categories);

        loading_template($smarty, 'site_header');
        loading_template($smarty, 'site_category');}

```

Листинг Б.4 – Файл ProductController.php

```

<?php
function indexAction($smarty) {
    if(isset($_GET['id'])){
        $item_id = $_GET['id'];
    }else{
        $item_id = null;
    }
    if ($item_id == null) exit();
    $product_record = get_product_by_his_id($item_id);
    $all_records_categories = get_absolutely_all_categories();
    $smarty->assign('count_items_in_cart', 0);
    if (in_array($item_id, $_SESSION['session_cart'])){
        $smarty->assign('count_items_in_cart', 1);
    }
    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('site_title', '');
    $smarty->assign('product_record', $product_record);

    loading_template($smarty, 'site_header');
    loading_template($smarty, 'product');}

```

Листинг Б.5 – Файл RegisterController.php

```

<?php

include_once '../models/UserInfoTableModel.php';

function indexAction($smarty) {
    $smarty->assign('site_title', 'Регистрация');
    loading_template($smarty, 'register');
}

function register_userAction() {
    $user_email_addres = isset($_REQUEST['user_email_addres']) ?
$_REQUEST['user_email_addres ']: null;
    $user_email_addres = trim($user_email_addres);

    $first_password = isset($_REQUEST['first_password']) ?
$_REQUEST['first_password']: null;
    $second_password = isset($_REQUEST['second_password']) ?
$_REQUEST['second_password']: null;
}

```



```

        $user_phone = isset($_REQUEST['user_phone']) ?
$_REQUEST['user_phone'] : null;
        $user_adress = isset($_REQUEST['user_adress']) ?
$_REQUEST['user_adress'] : null;
        $user_name = isset($_REQUEST['user_name']) ? $_REQUEST['user_name']
: null;
        $user_name =trim($user_name);
        $result_data = null;
        $result_data = verify_register_params($user_email_adress,
$first_password, $second_password);
        if(!$result_data && verify_user_email_in_bd($user_email_adress)) {
            $result_data['success'] = false;
            $result_data['message'] = "Пользователь ('{$user_email_adress}')
уже зарегистрирован";}

        if(!$result_data) {
            $password_in_md5_form = md5($first_password);

            $data_of_user = new_user_register($user_email_adress,
$password_in_md5_form, $user_name, $user_phone, $user_adress);

            if ($data_of_user['success']) {

                $result_data['message'] = 'Пользователь успешно
зарегистрирован';
                $result_data['success'] = 1;

                $data_of_user = $data_of_user[0];

                $result_data['name_of_user'] = $data_of_user['user_name'] ?
$data_of_user['user_name'] : $data_of_user['email'];
                $result_data['userEmail'] = $user_email_adress;

                $_SESSION['session_user'] = $data_of_user;
                $_SESSION['session_user']['name_of_display'] =
$data_of_user['user_name'] ? $data_of_user['user_name'] :
$data_of_user['user_email_adress'];

            }else{
                $result_data['success'] = 0;
                $result_data['message'] = 'Ошибка регистрации';
            }
        }
        }echo json_encode($result_data);}

```

Листинг Б.6 – файл UserController.php

```

<?php

include_once '../models/IncludeModel.php';
function register_userAction() {
    if(isset($_REQUEST['user_email_adress'])) {
        $user_email_adress = $_REQUEST['user_email_adress'];
    }else{
        $user_email_adress = null;
    }

    $user_email_adress = trim($user_email_adress);
    if(isset($_REQUEST['first_password'])) {
        $first_password = $_REQUEST['first_password'];
    }else{
        $first_password = null;
    }

    if(isset($_REQUEST['second_password'])) {

```

```

        $second_password = $_REQUEST['second_password'];
    }else{
        $second_password = null;
    }

    if(isset($_REQUEST['user_phone'])){
        $user_phone = $_REQUEST['user_phone'];
    }else{
        $user_phone = null;
    }

    if(isset($_REQUEST['user_adress'])){
        $user_adress = $_REQUEST['user_adress'];
    }else{
        $user_adress = null;
    }

    if(isset($_REQUEST['user_name'])){
        $name = $_REQUEST['user_name'];
    }else{
        $name = null;
    }

    $user_name =trim($user_name);
    $result_data = null;
    $result_data = verify_register_params($user_email_addres,
    $first_password, $second_password);
    if(!$result_data && verify_user_email_in_bd($user_email_addres)) {
        $result_data['success'] = false;
        $result_data['message'] = "Пользователь ('{$user_email_addres}')
уже зарегистрирован";
    }

    if(!$result_data) {
        $password_in_md5_form = md5($first_password);

        $data_of_user = new_user_register($user_email_addres,
        $password_in_md5_form, $user_name, $user_phone, $user_adress);

        if ($data_of_user['success']) {

            $result_data['message'] = 'Пользователь зарегистрирован';
            $result_data['success'] = 1;
            $data_of_user = $data_of_user[0];

            $result_data['name_of_user'] = $data_of_user['user_name'] ?
            $data_of_user['user_name'] : $data_of_user['user_email_addres '];
            $result_data['userEmail'] = $user_email_addres;

            $_SESSION['session_user'] = $data_of_user;
            $_SESSION['session_user']['name_of_display'] =
            $data_of_user['user_name'] ? $data_of_user['user_name'] :
            $data_of_user['user_email_addres '];
        }else{
            $result_data['success'] = 0;
            $result_data['message'] = 'В регистрации возникла ошибка';
        }
    }echo json_encode($result_data);
}

function login_outAction() {
    if(isset($_SESSION['session_user'])) {

```

```

        unset($_SESSION['session_user']);
        unset($_SESSION['session_cart']);
    }main_page_ridirecting('/');
}
function login_usersAction() {
    $user_email_addres = isset($_REQUEST['user_email_addres ']) ?
$_REQUEST['user_email_addres '] : null;
    $user_email_addres = trim($user_email_addres);

    $user_password = isset($_REQUEST['user_password']) ?
$_REQUEST['user_password'] : null;
    $user_password = trim($user_password);

    $data_of_user = login_users($user_email_addres, $user_password);

    if($data_of_user['success']) {
        $data_of_user = $data_of_user[0];

        $_SESSION['session_user'] = $data_of_user;
        $_SESSION['session_user']['name_of_display'] =
$data_of_user['user_name'] ? $data_of_user['user_name'] :
$data_of_user['user_email_addres '];

        $result_data = $_SESSION['session_user'];
        $result_data['success'] = 1;
    }else{
        $result_data['success'] = 0;
        //$result_data['message'] = 'test';
        $result_data['message'] = 'Логин или пароль введны неверно';
    }

    echo json_encode($result_data);
}

function indexAction($smarty) {
    if (!isset($_SESSION['session_user'])) {
        main_page_ridirecting('/');
    }
    $all_records_categories = get_absolutely_all_categories ();
    $record_of_users_orders = get_current_user_orders();

    $smarty->assign('site_title', 'Страница пользователя');
    $smarty->assign('all_records_categories', $all_records_categories);
    $smarty->assign('record_of_users_orders', $record_of_users_orders);
    loading_template($smarty, 'site_header');
    loading_template($smarty, 'site_user');}

function updateAction() {
    if(!isset($_SESSION['session_user'])) {
        main_page_ridirecting('/');
    }
    $result_data = array();
    $user_phone = isset($_REQUEST['user_phone']) ?
$_REQUEST['user_phone'] : null;
    $user_adress = isset($_REQUEST['user_adress']) ?
$_REQUEST['user_adress'] : null;
    $user_name = isset($_REQUEST['user_name']) ?
$_REQUEST['user_name'] : null;
    $first_password = isset($_REQUEST['first_password']) ?
$_REQUEST['first_password'] : null;
    $second_password = isset($_REQUEST['second_password']) ?
$_REQUEST['second_password'] : null;

```

```

        $current_password = isset($_REQUEST['current_password']) ?
$_REQUEST['current_password'] : null;

        $current_password_in_md5_form = md5($current_password);
        if(!$current_password ||
($$_SESSION['session_user']['user_password']) !=
$current_password_in_md5_form) {
            $result_data['success'] = 0;
            $result_data['message'] = 'Текущий пароль не верный';

            echo json_encode($result_data);return false;
        }
        $result = data_user_update($user_name, $user_phone, $user_adress,
$first_password, $second_password, $current_password_in_md5_form);

        if($result) {
            $result_data['success'] = 1;
            $result_data['message'] = 'Данные изменены';
            $result_data['name_of_user'] = $user_name;

            $_SESSION['session_user']['user_name'] = $user_name;
            $_SESSION['session_user']['user_phone'] = $user_phone;
            $_SESSION['session_user']['user_adress'] = $user_adress;
            $new_user_password = $_SESSION['session_user']['user_password'];
            if($first_password && ($first_password == $second_password)){
                $new_user_password = md5(trim($first_password));
            }
            $_SESSION['session_user']['user_password'] = $new_user_password;
            $_SESSION['session_user']['name_of_display'] = $user_name ?
$user_name : $_SESSION['session_user']['user_email_adress '];
        }else {
            $result_data['success'] = 0;
            $result_data['message'] = 'Ошибка изменения';
        }echo json_encode($result_data);
    }
}

```

Листинг Б.7 – Файл mainFunctions.php

```

<?php
function loading($smarty, $name_of_controller, $name_of_action =
'index') {
    include_once PathPrefix . $name_of_controller . PathPostfix;
    $function = $name_of_action . 'Action';
    $function($smarty);}

function loading_template($smarty, $name_of_template) {
    $smarty->display($name_of_template .Postfix_of_template);
}

function generate_array_of_smarty_records($database_result) {

    if (!$database_result) return false;
    $record_set_of_smarty = array();
    while($row = mysqli_fetch_assoc($database_result))

        $record_set_of_smarty[] = $row;

    return $record_set_of_smarty;
}

function main_page_ridirecting($url) {

    if(!$url) $url = '/';
    header("Location: {$url}");
    exit;
}

```

Листинг Б.8 – Файл main.js

```
function add_item_to_cart(item_id) {
    console.log("add_item");
    $.ajax({
        type: 'POST',
        async: true,
        url: "/cart/add_to_cart/" + item_id + '/',
        dataType: 'json',
        success: function(data) {
            if(data['success']) {

$('#number_of_products_in_the_cart').html(data['count_of_items']);
                $('#add_item_to_cart_' + item_id).hide();
                $('#remover_item_to_cart_' + item_id).show();
            }
        }
    });
}

function remover_item_from_cart(item_id) {
    $.ajax({
        type: 'POST',
        async: true,
        url: "/cart/remove_from_cart/" + item_id + '/',
        dataType: 'json',
        success: function (data) {
            if (data['success']) {

$('#number_of_products_in_the_cart').html(data['count_of_items']);
            }
        }
    });
}

function check_full_price_of_item(item_id) {
    var count_new = $('#count_of_items_' + item_id).val();
    var price_of_item = $('#price_of_item_' + item_id).attr('value');
    var full_price_of_items = count_new * price_of_item;
    $('#full_price_of_items_' + item_id).html(full_price_of_items);
}

function get_data_from_html_forms(from_object_form) {
    var data = {};
    $('input, textarea, select', from_object_form).each(function() {
        if (this.name && this.name != '') {
            data[this.name] = this.value;
        }
    });
    return data;
}

function new_user_register() {
    var data_for_post = get_data_from_html_forms('#register_form');
    $.ajax({type: 'POST',
        async: false,
        url: "/user/register_user",
        data: data_for_post,
        dataType: 'json',
        success: function (data) {
            if(data['success']) {
                alert('Регистрация завершена');
                $('#register_form').hide();
                $('#user_connect').attr('href', '/user/');
                $('#user_connect').html(data['name_of_user']);
                $('#user_form_of_authorized').show();
                $('#form_login').hide();
                $('#button_save_order').show();

                $('#errorBlock').show()
            }
        }
    });
}
```

```

        alert(data['message']);
    }

    });
}

function login_out() {
    $.ajax({
        type: 'POST',
        url: '/user/login_out/',
        success: function() {
            $('#register_form').show();
            $('#user_form_of_authorized').hide();
        }
    });
}

function login() {
    var user_email_address = $('#loginEmail').val();
    var user_password = $('#loginPwd').val();

    var data_for_post = 'user_email_address=' + user_email_address
    + '&user_password=' + user_password;

    $.ajax({
        type: "POST",
        async: false,
        url: '/user/login/',
        data: data_for_post,
        dataType: 'json',
        success: function (data) {
            if(data['success']){

                $('#register_form').hide();
                $('#form_login').hide();

                $('#user_connect').attr('href', '/user/');
                $('#user_connect').html(data['name_of_display']);
                $('#user_form_of_authorized').show();
                $('#user_name').val(data['user_name']);
                $('#user_phone').val(data['user_phone']);
                $('#user_address').val(data['user_address']);

                $('#button_save_order').show();
            }else{
                alert(data['message']);
            }
        }
    });
}

function show_register_box() {

    if($('#hide_register_form').css('display') != 'block') {

        $('#hide_register_form').show();}else
        $('#hide_register_form').hide();}

function data_user_update() {

    var user_phone = $('#new_user_phone').val();
    var user_address = $('#new_user_address').val();
    var current_password = $('#current_password').val();
    var user_name = $('#new_user_name').val();

    var first_password = $('#new_first_password').val();

```

```

        var second_password      = $('#new_second_password').val();

        var data_for_post      = {user_phone:      user_phone,user_adress      :
user_adress,
                                first_password : first_password,
                                second_password : second_password,
                                current_password : current_password,
                                user_name : user_name};

        $.ajax({
            type:'POST',
            async: false,
            url: '/user/update/',
            data:data_for_post,
            dataType: 'json',
            success: function(data) {
                if(data['success']) {
                    $('#connect_user').html(data['name_of_user']);
                    alert(data['message']);
                }else{
                    alert(data['message']);
                }
            }
        });
    }

function order_save() {
    var data_for_post = get_data_from_html_forms('form');
    $.ajax({
        type: 'POST',
        async: false,
        url: '/cart/save_order/',
        data: data_for_post,
        dataType: 'json',
        success: function (data) {
            if(data['success']){
                alert(data['message']);
            }else {
                alert(data['message']);
            }
        }
    });
}

function products_show(id) {
    var name_of_object = "#id_for_orders_purch_" + id;
    if($(name_of_object).css('display') != 'table-row' )
        $(name_of_object).show();else
        $(name_of_object).hide();
}

```