Министерство образования и науки РФ

ФГБОУ ВПО «Ульяновский государственный университет»

Факультет Математики, информационных и авиационных технологий

Кафедра Телекоммуникационные технологии и сети

КУРСОВАЯ РАБОТА

по дисциплине «Базы данных»

« Проектирование и разработка базы данных Интернет-магазина»  
(название темы)

Направление бакалавриата: Информационные системы и технологии 09.03.02  
(наименование и номер специальности)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Работу выполнил студент | ИС-З-21 |  | C:\Users\IT\Downloads\photo_5454089681571529144_y.jpg 19.12.2022 |  | Гусев В.Е. |
| группа | Подпись, дата | Ф.И.О |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Научный руководитель | старший преподаватель |  |  |  | Бочкарева Ю.Е. |
| должность | Подпись, дата | Ф.И.О |

|  |  |
| --- | --- |
| Оценка |  |

Ульяновск  
2022

Оглавление

[Введение 3](#_Toc122344107)

[Глава 1. Обзор предметной области 5](#_Toc122344108)

[1.1. Обзор аналогичных систем 6](#_Toc122344109)

[1.2. Обзор технологий проектирования 10](#_Toc122344110)

[1.3. Обзор технологий разработки программы для управления БД 11](#_Toc122344111)

[Глава 2. Проектирование и разработка базы данных 19](#_Toc122344112)

[2.1. Разработка логической модели базы данных 19](#_Toc122344113)

[2.2. Разработка физической модели базы данных 19](#_Toc122344114)

[2.3. Разработка SQL-запросов для создания таблиц базы данных 20](#_Toc122344115)

[2.4. Разработка структурной схемы программы для управления БД 24](#_Toc122344116)

[2.5. Разработка программы для управления БД 26](#_Toc122344117)

[Заключение 54](#_Toc122344118)

[Список литературы 56](#_Toc122344119)

# Введение

Базы данных являются составной частью информационной системы, осуществляющей хранение и обработку данных в широком смысле, включая ввод и корректировку данных, удаление данных, выполнение различных запросов, арифметические и логические преобразования данных, составление отчетов, выдачу информации в различных формах на экране и в бумажном виде.

Данная база данных создается в рамках курсовой работы. Проектирование ведётся с точки зрения пользователя.

**Целью курсовой работы** является: изучение основ проектирования баз данных в различных приложениях; разработка приложения для управления базой данных; усвоение теоретических основ организации баз данных, включая принципы построения на концептуальном, логическом и физическом уровнях; получение навыков постановки и решения практических задач проектирования и эксплуатации баз данных.

**Задачей курсовой работы** является проектирование и создание базы данных (БД) по теме «Проектирование и разработка базы данных Интернет-магазина», а также разработать программу для управления базой данных, которая может быть написана на любом языке программирования и должна уметь подключаться к БД и отображать содержимое таблиц этой базы данных.

**Актуальность**: по мере развития российской экономики все большую актуальность приобретают производственные процессы, ориентированные на удовлетворение разнообразных потребностей рынка. Организационное планирование и управление тесным образом связано с количеством, формой и точностью имеющейся информации. Рациональные информационные потоки позволяют не только учитывать, контролировать и организовывать процессы складирования материалов и готовой продукции, а также их транспортировки, но и создавать жизнеспособную систему логистики, обеспечивающую высокий уровень обслуживания потребителей необходимыми товарами и услугами в заданных количествах в нужном месте и в установленное время. В данной курсовой работе требуется произвести учёт информации о товарах, заказов клиентов, пользователей, разработать программу и организовать вывод необходимой информации на удобный, созданный интерфейс базы данных “Интернет-магазин”.

В структуре базы данных интернет-магазина сущности соответствует таблица. Она представляет собой набор полей, которые соответствуют атрибутам сущности.

В каждой таблице обязательно должно быть поле, хранящее идентификатор записи. Значение идентификатора, как правило, формируется автоматически при добавлении новой записи и никогда не повторяется. Это необходимо потому, что связь между таблицами осуществляется как раз по этому полю, и при дублировании значений сбоев в работе ресурса не избежать.

**Глава 1. Обзор предметной области**

В настоящее время одним из приоритетов развития дистрибьюторской компании в России должна становиться логистика. Основными принципиальными задачами логистики являются оптимальное управление товарными потоками для обеспечения доставки товара в нужное время, в нужном месте, точно в срок и с минимальными издержками. Логистика в современном понимании - это наука о планировании, контроле и управлении доставкой, складированием, перемещением материальных потоков в процессе доставки, переработки и доведения готовой продукции до потребителя. Информационная логистическая система - гибкая структура, состоящая из персонала, производственных объектов, средств вычислительной техники, необходимых справочников, компьютерных программ, различных интерфейсов и процедур (технологий), объединенных связанной информацией, используемой в управлении организацией для планирования, контроля, анализа и регулирования логистической системы. Часто используется тождественный термин «логистическая информационная система» (ЛИС), которые, как правило, представляют собой автоматизированные системы управления логистическими процессами.  
 Экономический фактор. В современных условиях на первый план выдвигается поиск возможностей сокращения производственных затрат и издержек обращения ради получения прибыли. Логистика позволяет связать экономические интересы производителя продукции и ее потребителя. Организационно-экономический фактор. В условиях рынка, по мере возникновения и развития новых организационных форм, реализующих процессы товародвижения, все большее значение приобретают интеграционные формы управления и координации, обеспечение логистических процессов взаимодействия предприятий-изготовителей, потребителей, посредников, складов и транспорта.

Информационный фактор. Рыночная экономика способствует развитию информационных связей, которые являются причиной и следствием рыночных отношений, взаимно обуславливают друг друга. Информатика наиболее тесно связывает рынок и логистику, поскольку ее предметом, средством и составляющей логистических процессов являются информационные потоки.

Технический фактор. Этот фактор проявляется в том, что логистика как система, ее субъекты и объекты управления развиваются на основе современных технических достижений в транспортно-складском хозяйстве и компьютеризации управления.

Государственная поддержка процессов товародвижения. В современных условиях возникает задача регулирования процессов товародвижения не только на уровне предприятий, но и в масштабах регионов, а также в национальном масштабе. Высокая актуальность внедрения логистики связана с интенсификацией и расширением в нашей стране товарно-денежных отношений, с увеличением хозяйственных связей между предприятиями, с развитием производственной инфраструктуры и расширением хозяйственной самостоятельности предприятий и организаций.

**1.1. Обзор аналогичных систем**

Электронная коммерция стала неотъемлемой частью нашего общества. По мере интенсивного роста числа пользователей сети Интернет она становится все более мощным инструментом бизнеса, средством реализации товаров и услуг. Интернет стал основным источником информации для огромного числа потенциальных покупателей. Электронная коммерция - сфера мировой экономики, включающая в себя финансовые и торговые транзакции, которые проводятся при помощи компьютерных сетей, а также все виды бизнес-процессов, связанные с проведением таких транзакций.

Электронная коммерция предоставляет огромные возможности для развития своего дела. Однако для того чтобы использовать ее действительно эффективно мало быть талантливым предпринимателем, точно так же, как и мало уметь ориентироваться в безграничном виртуальном пространстве. Необходимо еще обладать рядом высокотехнологичных инструментов для ведения электронного бизнеса. Одним из них и является интернет-магазин.

Что такое интернет магазин? - Интернет-магазин является одной из разновидностей электронной коммерции. Продажа собственных товаров и услуг через интернет – основной способ коммерческого использования сайтов для большинства производственных и торговых предприятий. Компания создает сайт и размещает на нем информацию о своих товарах и услугах, ценах и гарантиях для покупателей. Интернет-магазины становятся одним из необходимых и эффективных инструментов увеличения числа продаж предприятия, увеличения оборота и повышения имиджа, успешного развития. Существует несколько определений «что такое интернет-магазин».

Интернет-магазин - специализированный вид бизнеса, основанный на дистанционном обслуживании и дающий возможность продавцу и покупателю совершать сделки по покупке-продаже того или иного вида товаров посредством сети Интернет.

Интернет-магазин - это интерактивный веб-сайт, рекламирующий товар или услугу, принимающий заказы на покупку, предлагающий пользователю выбор варианта расчета, способа получения заказа и выписывающий счет на оплату.

Интернет-магазин - это совокупность программ, работающих на Web-сайте, которые позволяют покупателю дистанционно выбрать товар из каталога и оформить его заказ.

Интернет-магазин является посредником между покупателем и складом производителя, которому достаточно иметь ресурс в Интернете и телефон. Магазин виртуального типа не нуждается в представительном офисе, складских помещениях, не несет расходы на содержание помещений, владелец магазина может управлять продажами даже из своей квартиры. Именно потому цены в таких магазинах минимальны. Купленные вещи подвозят потребителю непосредственно со склада производителя.

Виды интернет-магазинов. В настоящее время существует множество интернет-магазинов. Их можно классифицировать по следующим видам:

По объемам продаж:

* розничная продажа (продажи на одного клиента идут в небольших объемах, зачастую по одной единице товара);
* оптовая продажа (продажи идут в больших объемах, зачастую клиент является постоянным, что говорит о систематичности покупок).

По видам продаж:

* B2B (би-ту-би или «бизнес для бизнеса») — продажа или оказание услуг иным коммерческим предприятиям;
* B2C (би-ту-си или «бизнес для клиента») — продажа или оказание услуг конечному потребителю.

По отношению с поставщиками:

* имеют собственный склад (наличие реальных товарных запасов);
* работают по договорам с поставщиками (отсутствие значительных собственных запасов).

Возьмем, к примеру, популярный интернет-магазин Wildberries – это международный маркетплейс одежды, обуви, электроники, детских товаров, товаров для дома и других товаров. По соображениям, у них есть заказчики/пользователи/клиенты/покупатели, следовательно, и есть таблица в базе данных, которая может содержать в себе столбцы:

* Фамилия;
* Имя;
* Номер телефона;
* Адрес пункта выдачи;
* Персональная скидка;
* Сумма выкупа;
* Процент выкупа;
* Данные банковской карты;
* Отложенные товары;
* Корзина;
* Покупки и др.

У магазина есть пункты выдачи, которые могут содержать в себе:

* Страна;
* Регион;
* Город;
* Адрес;
* Сотрудники и др.

Если есть сотрудники, то таблица может содержать в себе:

* Фамилия;
* Имя;
* Отчество;
* Паспортные данные и др.

Самое главное в интернет-магазине, это товар. Таблица с товаром может содержать в себе:

* Артикул;
* Поставщик;
* Фото;
* Наименование (Кисель быстрого приготовления, Робот-пылесос и др.);
* Производитель (Nestle, Danone, Adidas, Polaris, Apple и др.);
* Категория (Обувь, мебель, игрушки, продукты, бытовая техника и др.);
* Подкатегория (Женская/мужская обувь, мебель для кухни, спальни, офиса, игрушки настольные, спортивные, радиоуправляемые, продукты - напитки, овощи, фрукты и др.);
* Цена без скидки;
* Цена со скидкой;
* Оценка товара;
* Описание товара (О товаре, описание, характеристики и т.д.);
* Кол-во покупок.

Т.к. имеются отзывы к товару, то и имеются колонки в таблице:

* Id пользователя, оставившего комментарий;
* Оценка товара;
* Дата написания комментария;
* Кол-во пользователей, кому понравился комментарий;
* Кол-во пользователей, кому не понравился комментарий;
* Текст комментария/отзыва.

**1.2. Обзор технологий проектирования**

Проведем проектирование базы данных для интернет-магазина.

Интернет-магазин (ИМ) предлагает широкому кругу потребителей одежду по различным категория: куртки, футболки, обувь, штаны и др., которые, в свою очередь, делятся на подразделы. Предположим, у ИМ есть свой собственный сайт, где покупатель выбирает одежду и кладет в корзину заказов. Заполняет форму заказа, определяя вид и адрес доставки, метод оплаты и прочее. Заказ автоматически обрабатывается на сервере магазина, данные заказа должны поступить в базу данных информационной системы магазина.

Постоянными пользователями базы данных будут операторы, обрабатывающие заказы, и сотрудники отдела доставки, распределяющие заказы между курьерами.

Подчеркнем особую важность использования базы данных специалистами в области маркетинга. Для руководителя и сотрудников отдела маркетинга база данных является основой для анализа товара, рынка и покупателя. Именно из базы данных маркетологи должны получать информацию об уровне реализации отдельных наименований, сводку доходности продуктов по различным категориям, его спроса, особенности и требования постоянных клиентов и др.

Пользователями баз данных являются и другие сотрудники компании, такие как руководитель компании, сотрудники бухгалтерии, менеджер по кадрам. Эта категория пользователей также должна получать информацию в определенной форме и степени структуризации в зависимости от цели: учета, анализа, планирования, управления кадрами. База данных заказов может быть основой для системы принятия решений и системы управления взаимоотношений с клиентами (CRM).

Так же, сами покупатели являются пользователями базы данных. Они осуществляют поиск товаров, им отображается ранее введенная информация, хранящиеся в корзине товары, история заказов и др.

Точки зрения отдельных групп пользователей (различные пользовательские представления) должны быть учтены при проектировании, а их требования должны быть сведены в единую модель базы данных.

**1.3. Обзор технологий разработки программы для управления БД**

Для разработки программы управления базой данных был выбран следующий стек технологий (набор технологий, на основе которых разрабатывается сайт или приложение. Стек включает в себя языки программирования, фреймворки (программная среда для разработки), системы управления базами данных, компиляторы (переводят текст, написанный на языке программирования, в набор машинных кодов) и так далее.):

* HTML;
* CSS;
* SVG;
* JavaScript;
* JQuery;
* Ajax;
* API;
* PHP;
* PDO PHP;
* MariaDB;
* SQL.

HTML - Стандартизированный язык гипертекстовой 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. HTML – своего рода, «скелет» сайта, у которого есть:

* голова (head. несет в себе служебную информацию о странице. Обычно содержится заголовок, ключевые слова, описание страницы и другие служебные данные. Также внутри него подключаются внешние ресурсы, например, стили. Содержимое этого тега не отображается на странице напрямую);
* тело (body. Предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера. Информацию, которую следует выводить в документе, следует располагать именно внутри этого контейнера. К такой информации относится текст, изображения, теги, скрипты JavaScript и т.д.).

CSS (англ. Cascading Style Sheets - «каскадные таблицы стилей») — формальный язык описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). 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.

Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана) или при выводе устройствами, использующими шрифт Брайля.

SVG (Scalable Vector Graphics) — это язык разметки векторной графики, созданный W3C, подмножество языка XML. Предназначен для описания векторной и смешанной (векторно-растровой) двухмерной графики, поддерживает анимацию и интерактивность. Его плюсы:

* SVG легко модифицируется (причём, как в графическом редакторе, так и на самой странице средствами CSS). Поменять расположение, форму, размер, пропорции, цвет, заливку и все прочие свойства составляющих частей изображения проще, чем в случае с растровой графикой. При работе с растром придётся хранить «исходник» в формате со слоями, все изменения делать в нём, экспортировать. С SVG обычно необходимости в «исходнике» просто нет.
  + Для экранов с повышенной плотностью пикселей достаточно одного изображения. В случае с растровой графикой на данный момент нужно, минимум, три версии картинки: 100% макетного размера, 200% и 300%. В случае использования SVG, достаточно одной версии — как и для любого векторного формата, пиксели, нужные для показа изображения, «возникают» непосредственно перед показом, исходя из необходимого пиксельного размера.
  + SVG быстро загружается. Дизайнеры и разработчики должны думать и о скорости загрузки страницы тоже, ведь чем больше времени проходит от отправки запроса до показа страницы, тем ниже конверсия проекта. Во-первых, SVG-файлы, как правило, имеют меньший размер файла, чем их растровые версии (исключения составляют изображения малого пиксельного размера и сложные векторные изображения со множеством форм). Во-вторых, в один SVG-файл можно добавить несколько версий изображения и показывать их по определённым условиям (уменьшение количества запросов к серверу). В-третьих, в SVG можно использовать «клонирование» — единожды прописать форму (градиент, текстуру) и многократно использовать её, ссылаясь на оригинал. Есть, правда, и минус: отрисовка SVG в браузере чуть медленнее, чем показ растрового изображения, но чтобы заметить эту разницу, нужно сравнивать большие и сложные изображения.
  + SVG легко сделать адаптивным. В том случае, когда необходимо на узких вьюпортах показывать логотип сайта в упрощённом варианте, SVG позволяет добиться этого всего за один запрос к серверу.
  + SVG может быть интерактивным. Внутри изображения могут быть ссылки, скрипты, интерактивные части могут реагировать на наведение и другие действия пользователя, можно добавить анимацию.

JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

jQuery — библиотека, 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.

AJAX, Ajax ( «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее. По-русски иногда произносится транслитом как «аякс». У аббревиатуры AJAX нет устоявшегося аналога на кириллице.

API (Application Programming Interface) — описание способов взаимодействия одной компьютерной программы с другими. Обычно входит в описание какого-либо интернет-протокола, программного каркаса или стандарта вызовов функций операционной системы. Часто реализуется отдельной программной библиотекой или сервисом операционной системы. Используется программистами при написании всевозможных приложений.

Проще говоря, это набор компонентов, с помощью которых компьютерная программа (бот или же сайт) может использовать другую программу.

API (интерфейс прикладного программирования) упрощает процесс программирования при создании приложений, абстрагируя базовую реализацию и предоставляя только объекты или действия, необходимые разработчику. Если графический интерфейс для почтового клиента может предоставить пользователю кнопку, которая выполнит все шаги для выборки и выделения новых писем, то API для ввода/вывода файлов может дать разработчику функцию, которая копирует файл из одного места в другое, не требуя от разработчика понимания операций файловой системы, происходящих за кулисами.

Если программу (модуль, библиотеку) рассматривать как чёрный ящик, то API — это набор «ручек», которые доступны пользователю данного ящика и которые он может вертеть и дёргать.

PHP (препроцессор гипертекста (HTML)) - это распространённый язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML.

PHP отличается от JavaScript тем, что PHP-скрипты выполняются на сервере и генерируют HTML, который посылается клиенту. Если бы у вас на сервере был размещён скрипт, подобный вышеприведённому, клиент получил бы только результат его выполнения, но не смог бы выяснить, какой именно код его произвёл. Вы даже можете настроить свой сервер таким образом, чтобы обычные HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

Доступ к базе данных обычно осуществляется с помощью интерфейса, который реализован как приложение к базе данных. Приложение может быть написано на различных языках программирования высокого уровня и с использованием различных программных средств. В данной программе управления базой данных используется PDO PHP.

PDO (PHP Data Objects) — расширение PHP, которое реализует взаимодействие с базами данных при помощи объектов. Профит в том, что отсутствует привязка к конкретной системе управления базами данных.

Предоставляемый интерфейс поддерживает, среди прочих, такие популярные СУБД:

* + MySQL;
  + SQLite;
  + PostgreSQL;
  + Microsoft SQL Server.

MariaDB — ответвление от системы управления базами данных MySQL. MySQL — свободная реляционная система управления базами данных. Является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

SQL (от англ. Structured Query Language — «язык структурированных запросов») — 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 неполным 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.

При всех своих изменениях SQL остаётся самым распространённым лингвистическим средством для взаимодействия прикладного программного обеспечения с базами данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

# Глава 2. Проектирование и разработка базы данных

## **2.1. Разработка логической модели базы данных**

При проектировании базы данных интернет-магазина, после описания предметной области, необходимо выбрать метод построения инфологической модели (ER-модель) и СУБД, в которой будет реализован проект.

Для построения ER модели (Рисунок №2.1.1) был выбран сайт visual-paradigm, т.к. она позволяет визуально отображать сложные структуры данных. Простая в использовании графическая среда упрощает разработку баз данных.

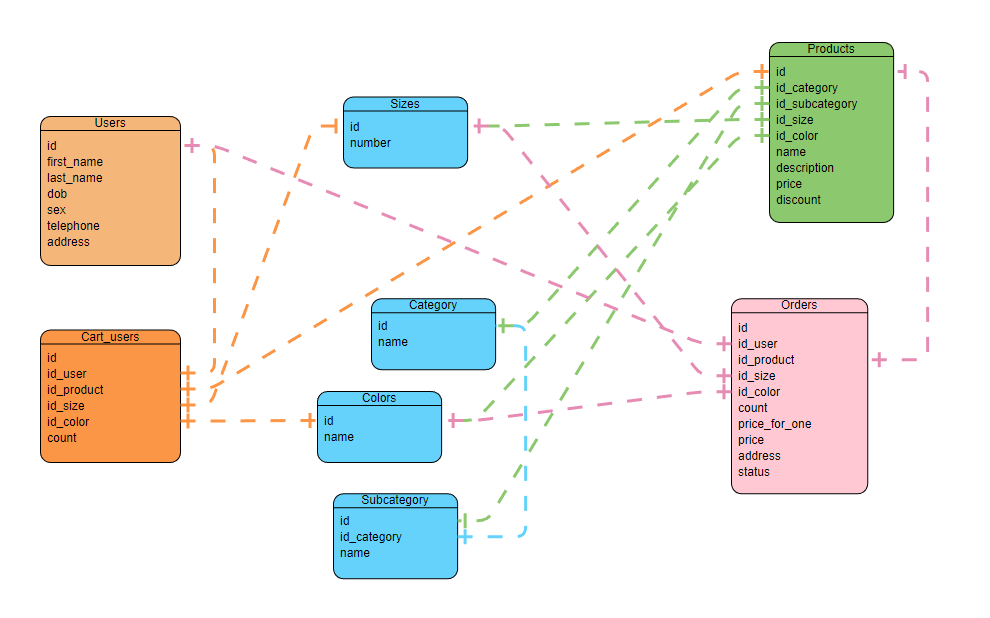


Рисунок №2.1.1. Построенная ER-модель базы данных интернет-магазина.

**2.2. Разработка физической модели базы данных**

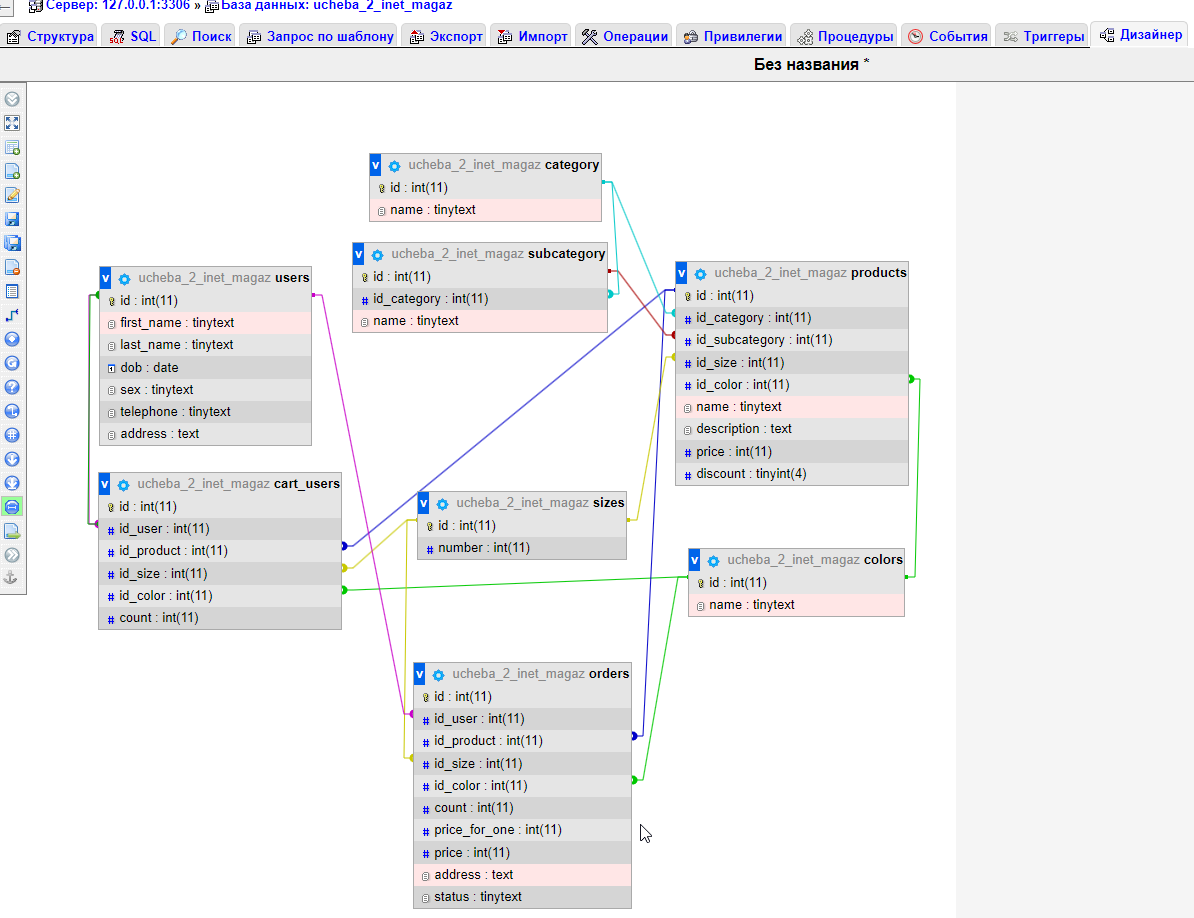
Физическая модель базы данных интернет-магазина (Рисунок №2.2.1) была создана на локальном хостинге phpMyAdmin под управлением MariaDB.

Рисунок №2.2.1. Физическая-модель базы данных интернет-магазина.

**2.3. Разработка SQL-запросов для создания таблиц базы данных**

Были сформированы SQL-запросы для создания таблиц «Пользователи», «Категории», «Подкатегории», «Размеры», «Цвета», «Товары», «Корзина пользователей», «Заказы» (Рисунок №2.3.1).

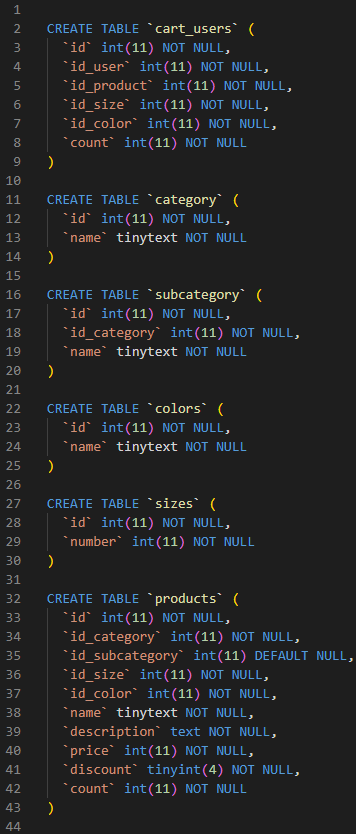
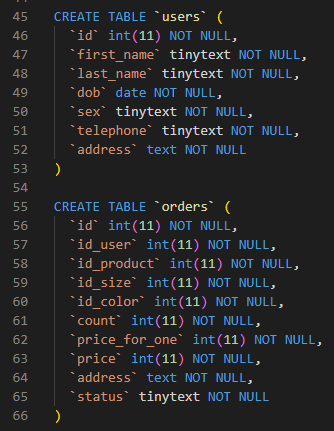
 

Рисунок №2.3.1. SQL-запросы для создания таблиц.

Далее, обозначили индексы (Рисунок №2.3.2).

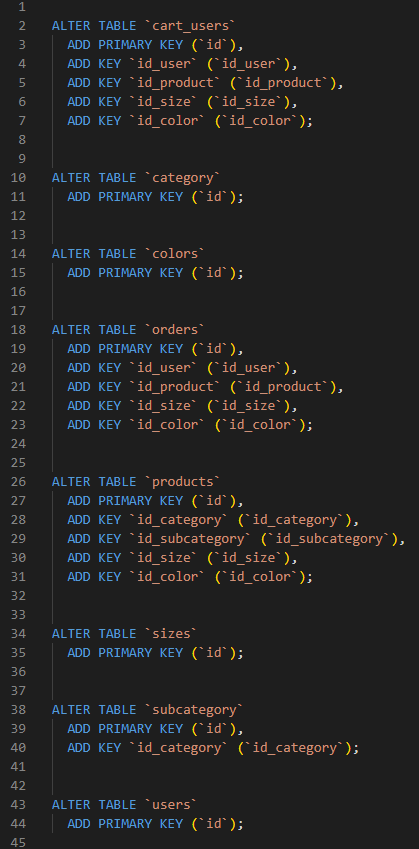


Рисунок №2.3.2. SQL-запрос на присваивания свойств к некоторым столбцам таблиц.

Установили AUTO\_INCREMENT (Рисунок № 2.3.3).

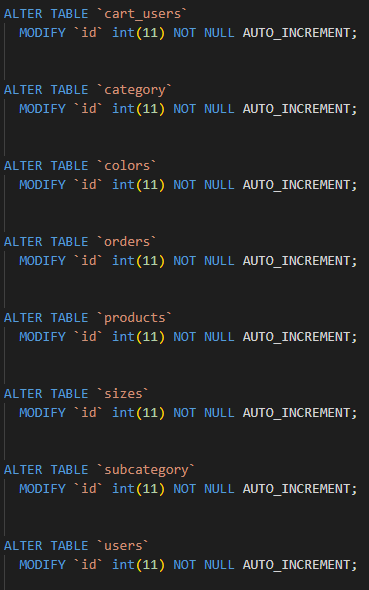


Рисунок №2.3.3. SQL-запрос на присваивания свойства AUTO\_INCREMENT к ключевым столбцам таблиц.

Добавили внешние ключи (Рисунок № 2.3.4).

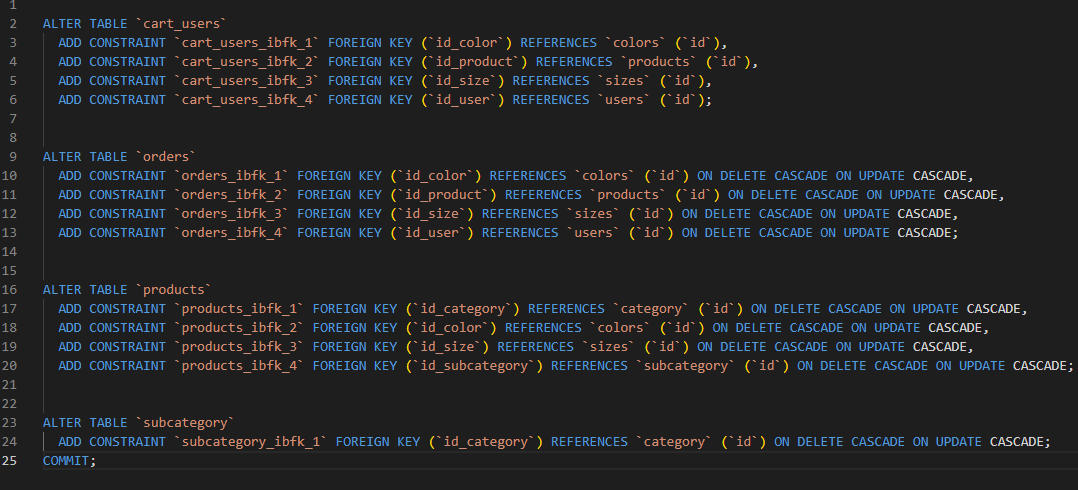


Рисунок №2.3.4. SQL-запрос на присваивание внешнего ключа к некоторым столбцам таблиц.

**2.4. Разработка структурной схемы программы для управления БД**

Макет программы (Рисунок №2.4.1) нарисован на платформе Figma. Слева на экране будет расположено «меню» с существующими таблицами интернет-магазина. При нажатии на которые, программа даст нам визуально понять, что мы выбрали. А также, после осуществления выбора, в правой части экрана, которое занимает большинство места, будет выведено содержимое таблицы вместе с кнопками действия напротив строк:

* Добавить;
* Удалить;
* Редактировать.

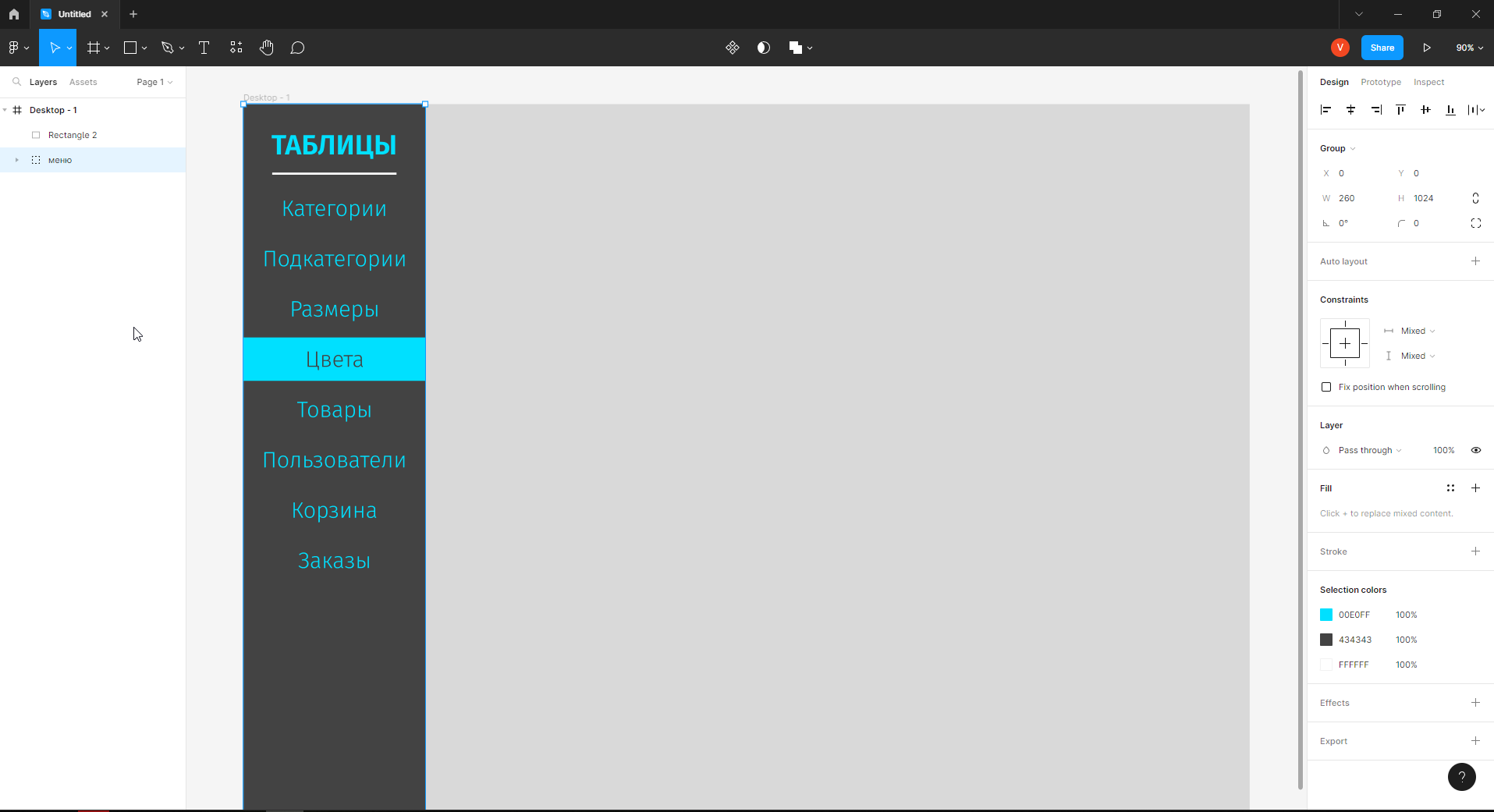


Рисунок №2.4.1. Макет программы управления БД.

Схема работоспособности программы выглядит следующим образом (Рисунок № 2.4.2).



Рисунок №2.4.2. Схема программы управления БД.

Пользовательская сторона (frontend) – то разработка пользовательских функций и интерфейса. К ним относится всё, что пользователи видят на сайте или в приложении, и с чем можно взаимодействовать: картинки, выпадающие списки, меню, анимация, карточки товаров, кнопки, чекбоксы, интерактивные элементы.

AJAX – это подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

Серверная сторона (backend) - это разработка бизнес-логики продукта (сайта или веб-приложения). backend отвечает за взаимодействие пользователя с внутренними данными, которые потом отображает frontend. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

**2.5. Разработка программы для управления БД**

Было решено, что программа управления базой данных (далее, ПУБД) будет web-версией. Преимущество WEB версии в том, что ее можно будет открыть абсолютно на любом устройстве, а именно на:

* Телефоне под управлением операционной системой Android, IOS и др.;
* Персональном компьютере (ПК) под управлением операционной системой Windows, Linux, MacOS и др.
* Планшетах и других устройствах, где можно открыть браузер и перейти на сайт.

ПУБД имеет следующую структуру (Рисунок № 2.5.1)

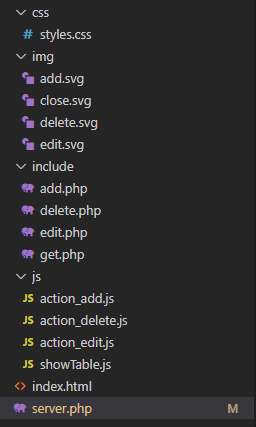


Рисунок № 2.5.1. Структура ПУБД

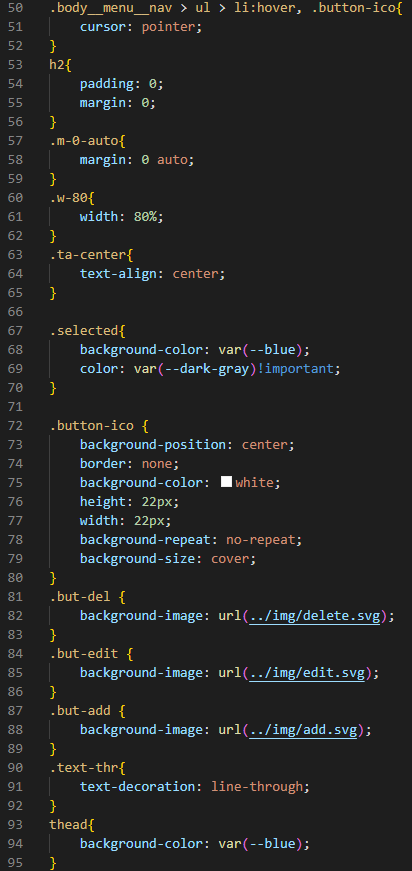
В корневой папке программы находятся 3 папки (img, include, js) и 2 файла (index.html, server.php).

index.html – это тот файл, который загружается и отображается пользователю при посещении сайта. Пользовательская сторона сайта написана на языке HTML (Рисунок 2.5.2).

Рисунок 2.5.2. Код страницы «index.html».

Так же, к frontend’у относится папка «css», которая содержит в себе файл «styles.css» - стили для ПУБД (Рисунок № 2.5.3).





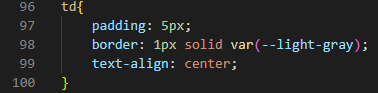


Рисунок № 2.5.3. Стили для ПУБД.

Некоторые стили кнопок ПУБД опираются на файлы в папке “img”, цель которой – хранение в себе картинок и иконок для пользовательского интерфейса программы. Она содержит файлы:

* add.svg – Иконка, символизирующая добавления какого-либо материала (Рисунок № 2.5.4);

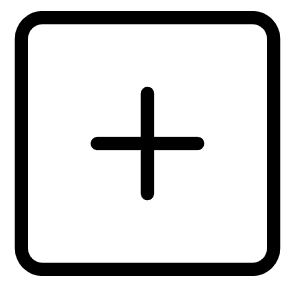
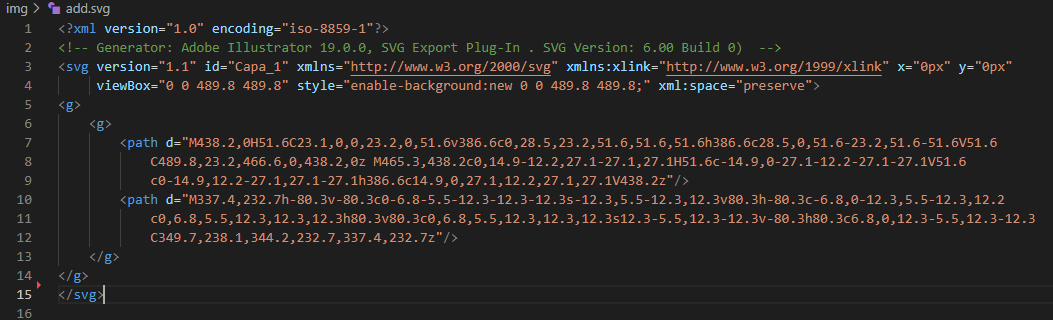


Рисунок № 2.5.4. Файл «add.svg».

* close.svg – Иконка, символизирующая закрытие чего-либо (Рисунок № 2.5.5);

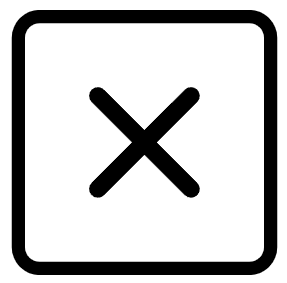
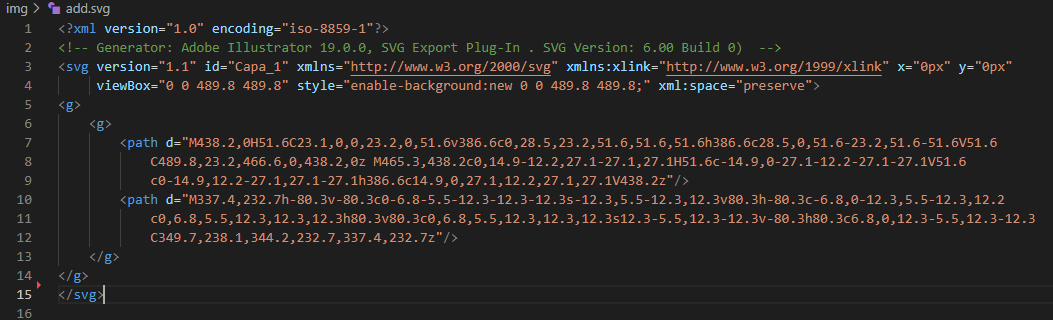


Рисунок № 2.5.5. Файл «close.svg».

* delete.svg – Иконка, символизирующая удаление какого-либо материала (Рисунок № 2.5.6);



Рисунок № 2.5.6. Код файла «delete.svg».

* edit.svg – Иконка, символизирующая редактирование какого-либо материала (Рисунок № 2.5.7);

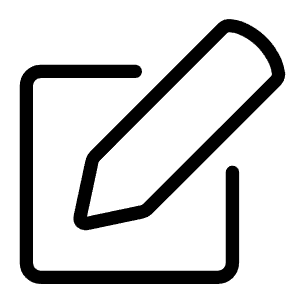
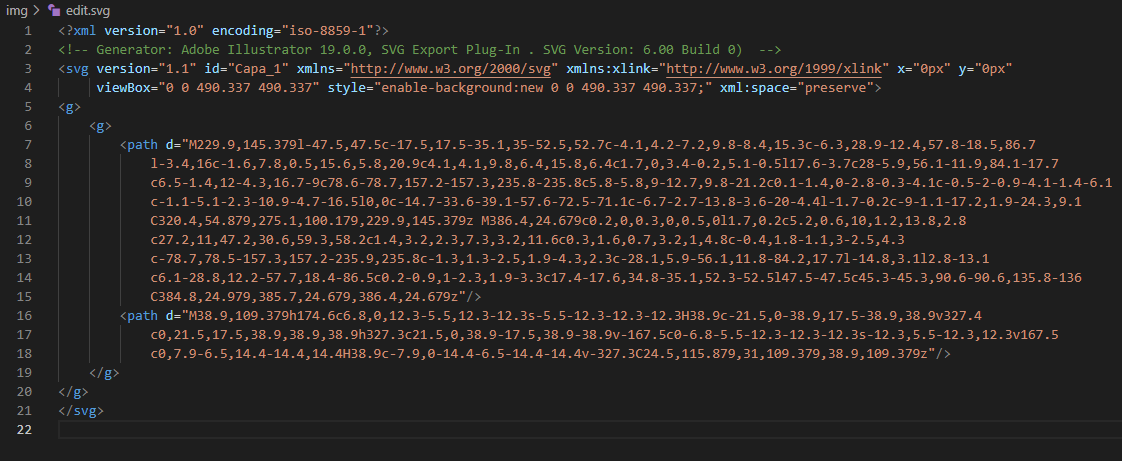


Рисунок № 2.5.7. Код файла «edit.svg».

В качестве иконок было выбрано расширение SVG.

Так же, index.html содержит в себе скрипты, которые находятся в папке «js»:

* showTable.js – функция отправки запроса на сервер с именем таблицы в качестве параметра и вывод результата пользователю (Рисунок № 2.5.8).

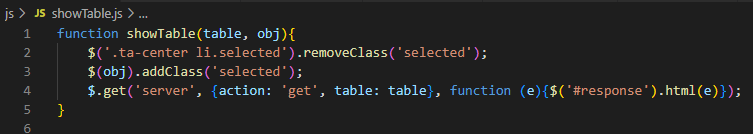
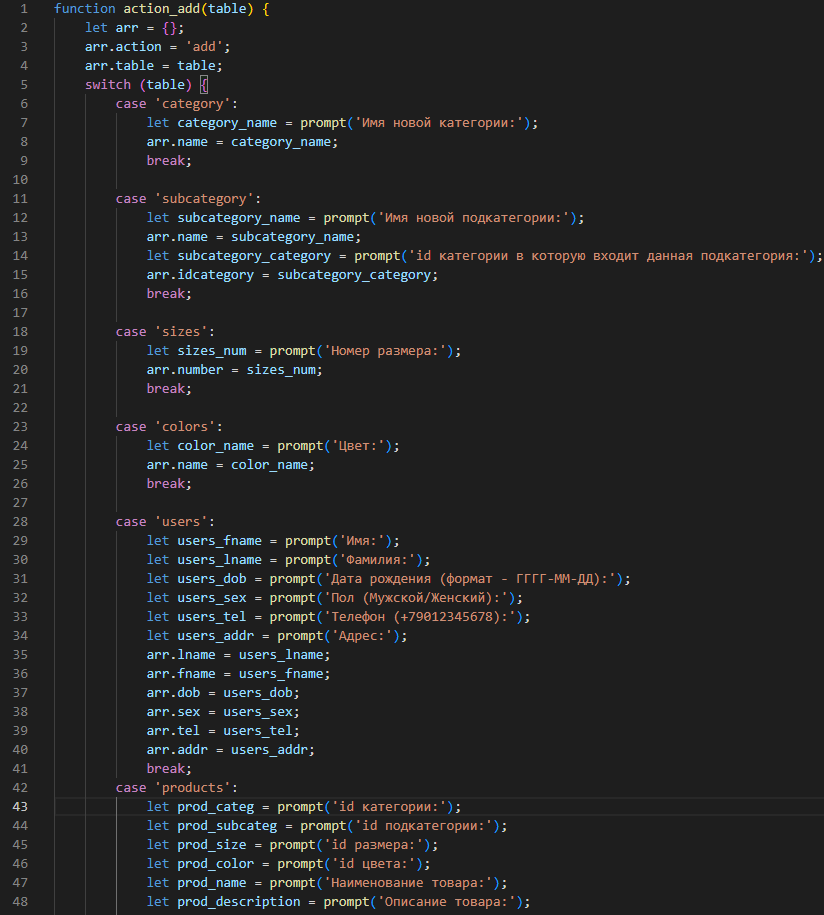


Рисунок № 2.5.8. Код файла «showTable.js»

* action\_add.js – функция запроса у пользователя данных для добавления новой записи в таблицу и их отправка на сервер (Рисунок № 2.5.9).



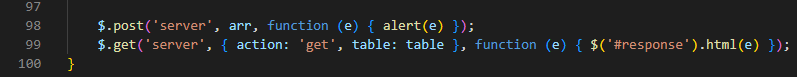


Рисунок № 2.5.9. Код файла «action\_add.js»

* action\_delete.js – Функция запроса на удаление записи из таблицы (Рисунок № 2.5.10).

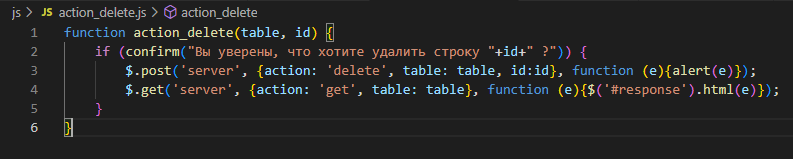
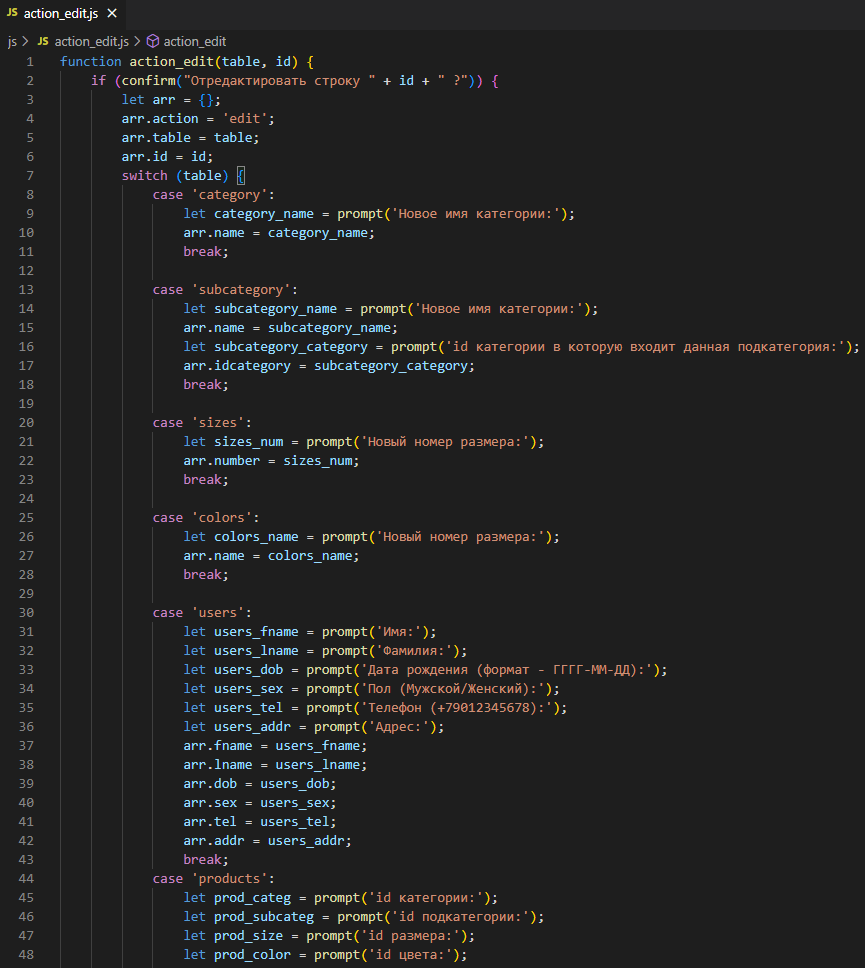


Рисунок № 2.5.10. Код файла «action\_delete.js»

* action\_edit.js – Функция редактирования и изменения данных строк в таблице (Рисунок № 2.5.10).



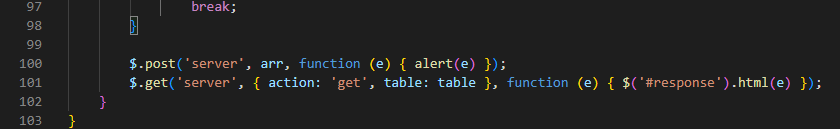
 

Рисунок № 2.5.11. Код файла «action\_edit.js»

Файл server.php – это серверный файл. Именно он принимает запросы на показ таблиц, удаление и редактирование строк. Код данного недоступен для обычного пользователя. Он написан на языке PHP (Рисунок 2.5.12).

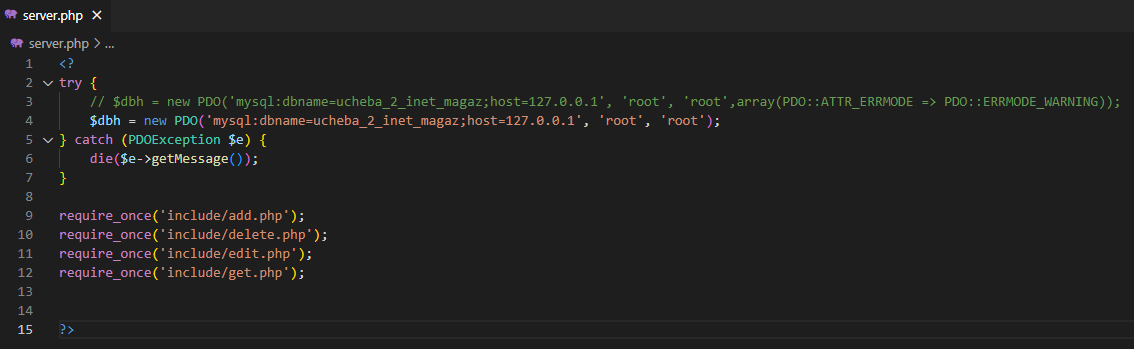


Рисунок № 2.5.12. Код файла «server.php»

В этом файле осуществляется соединение с базой данных и подключаются файлы:

* add.php – Функция которая принимает параметры, добавляет строку к определенной таблице через SQL запрос (рисунок № 2.5.13);

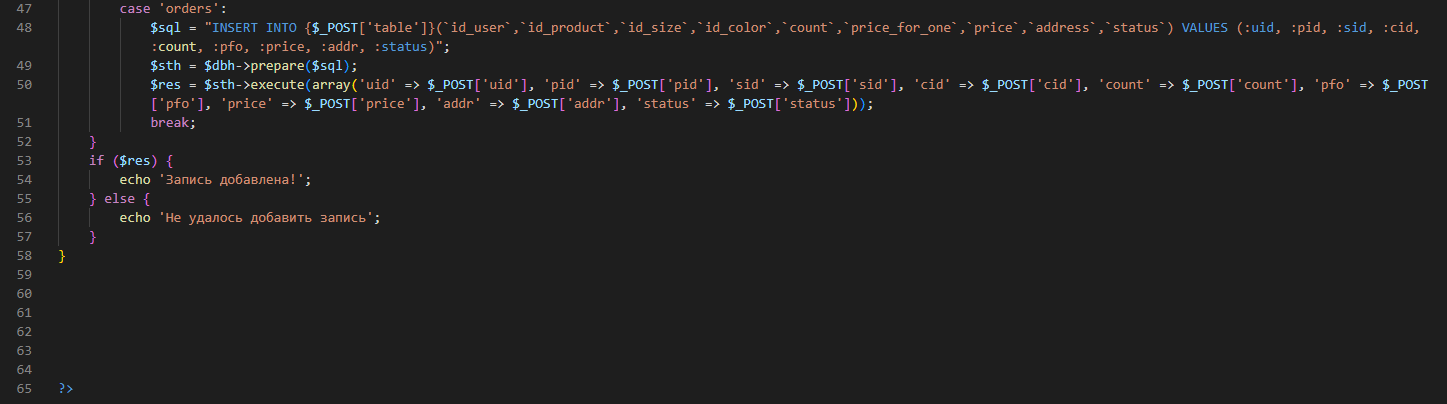
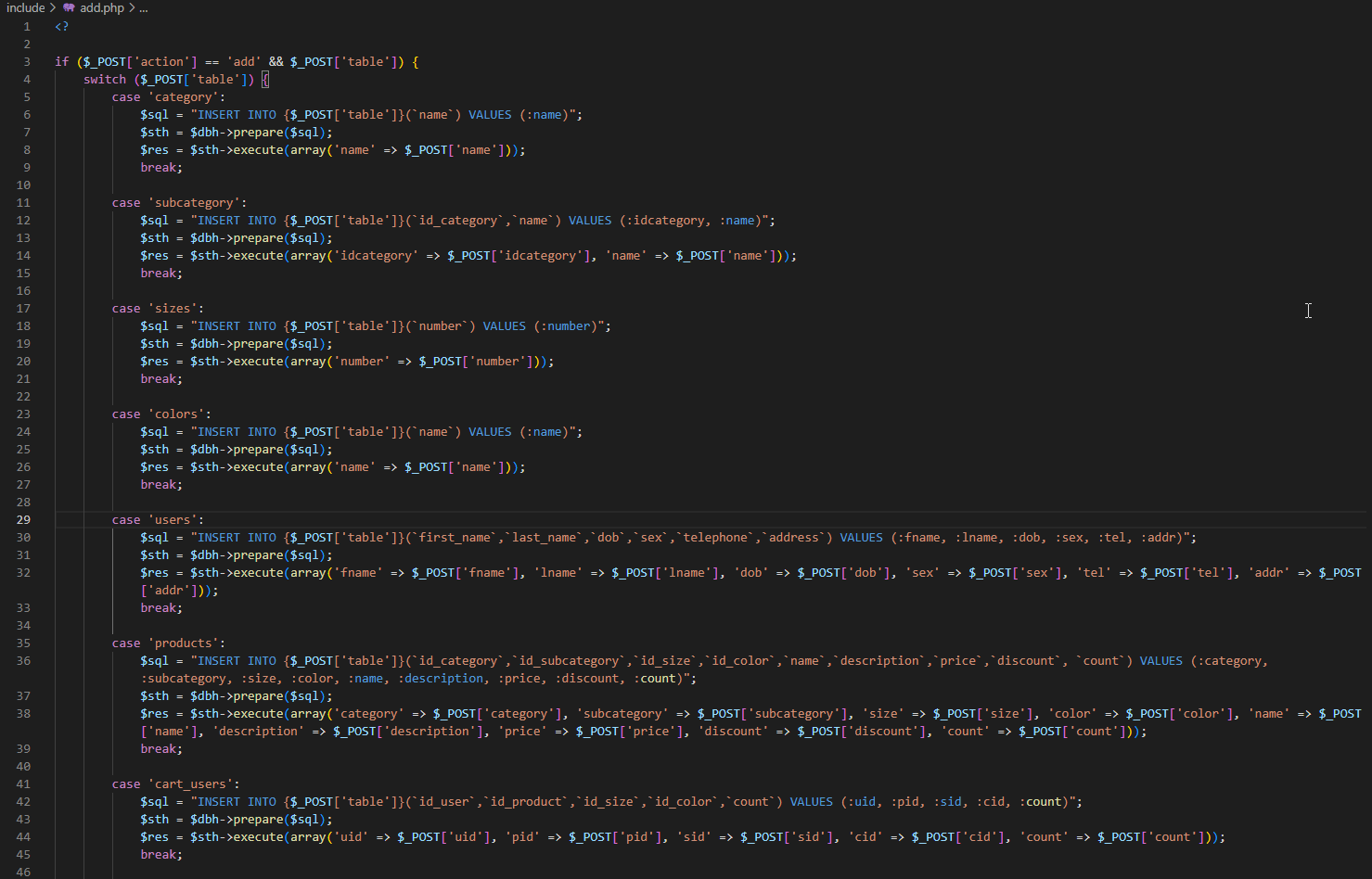


Рисунок № 2.5.13. Код файла «add.php»

* delete.php – Функция удаления строки из таблицы базы данных (Рисунок № 2.5.14);

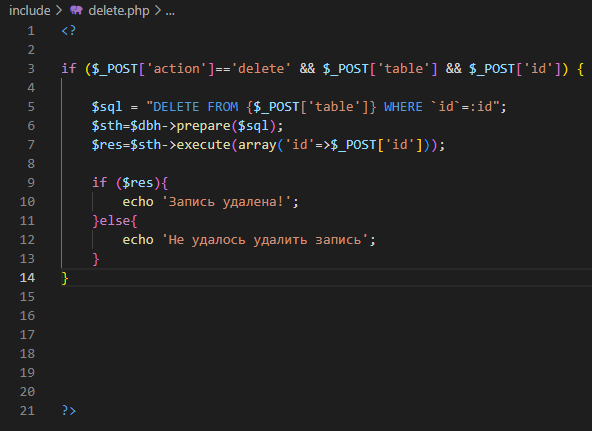
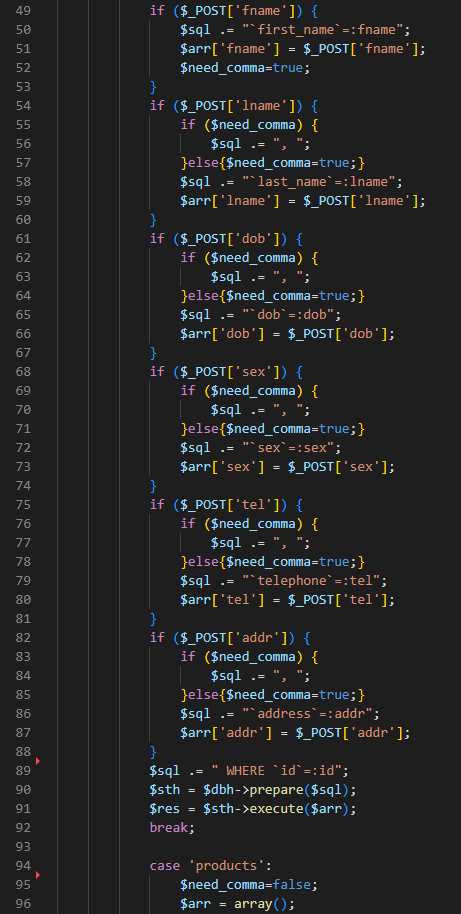
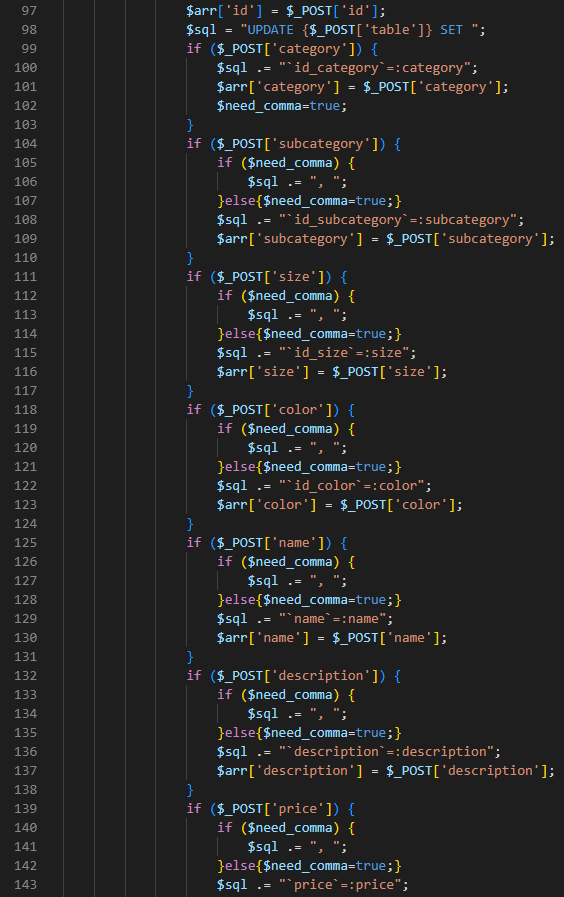


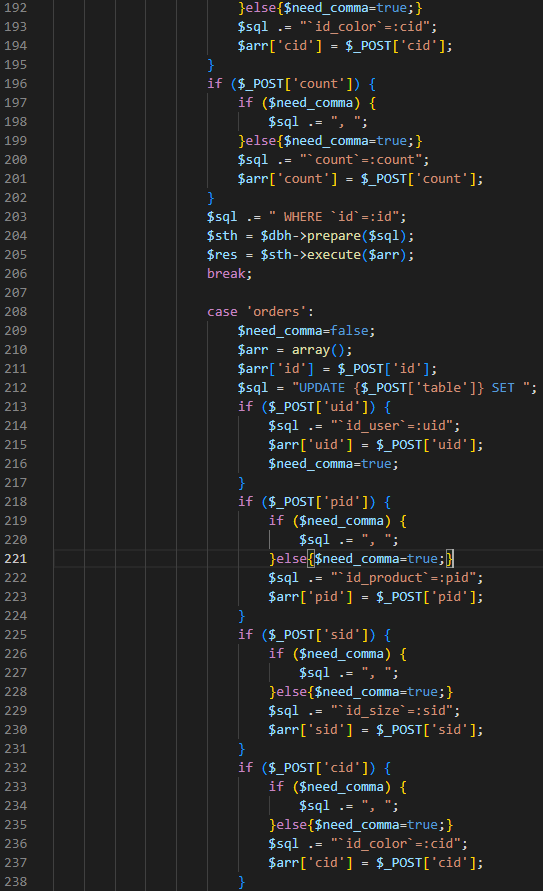
Рисунок № 2.5.14. Код файла «delete.php»

* edit.php – Функция редактирования значений ячеек строки таблицы в базе данных (рисунок № 2.5.15);









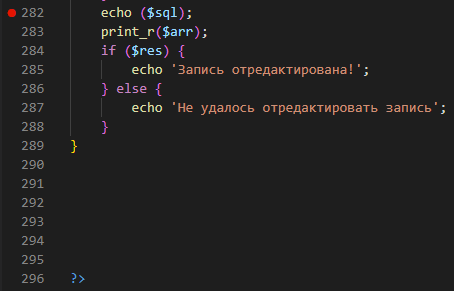
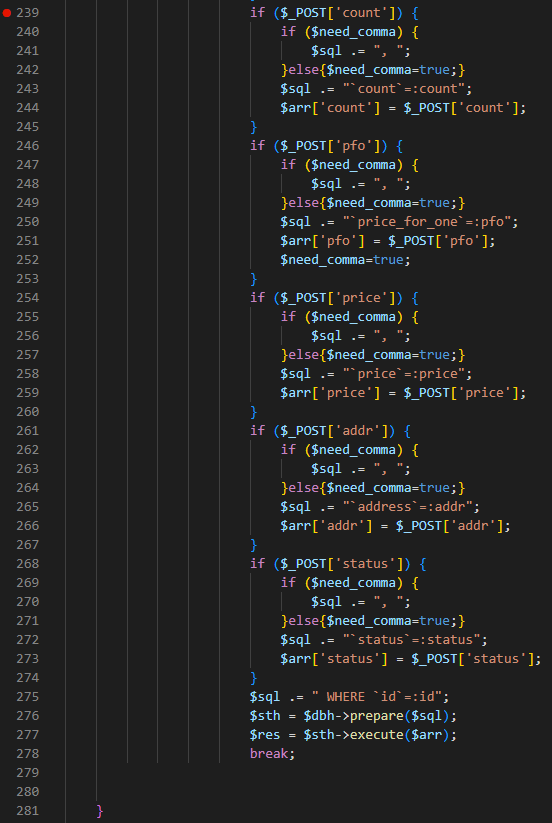


Рисунок № 2.5.15. Код файла «edit.php»

* get.php – функция запроса и вывода строк, выбранной пользователем, таблицы в базе данных (Рисунок № 2.5.16)

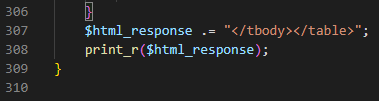
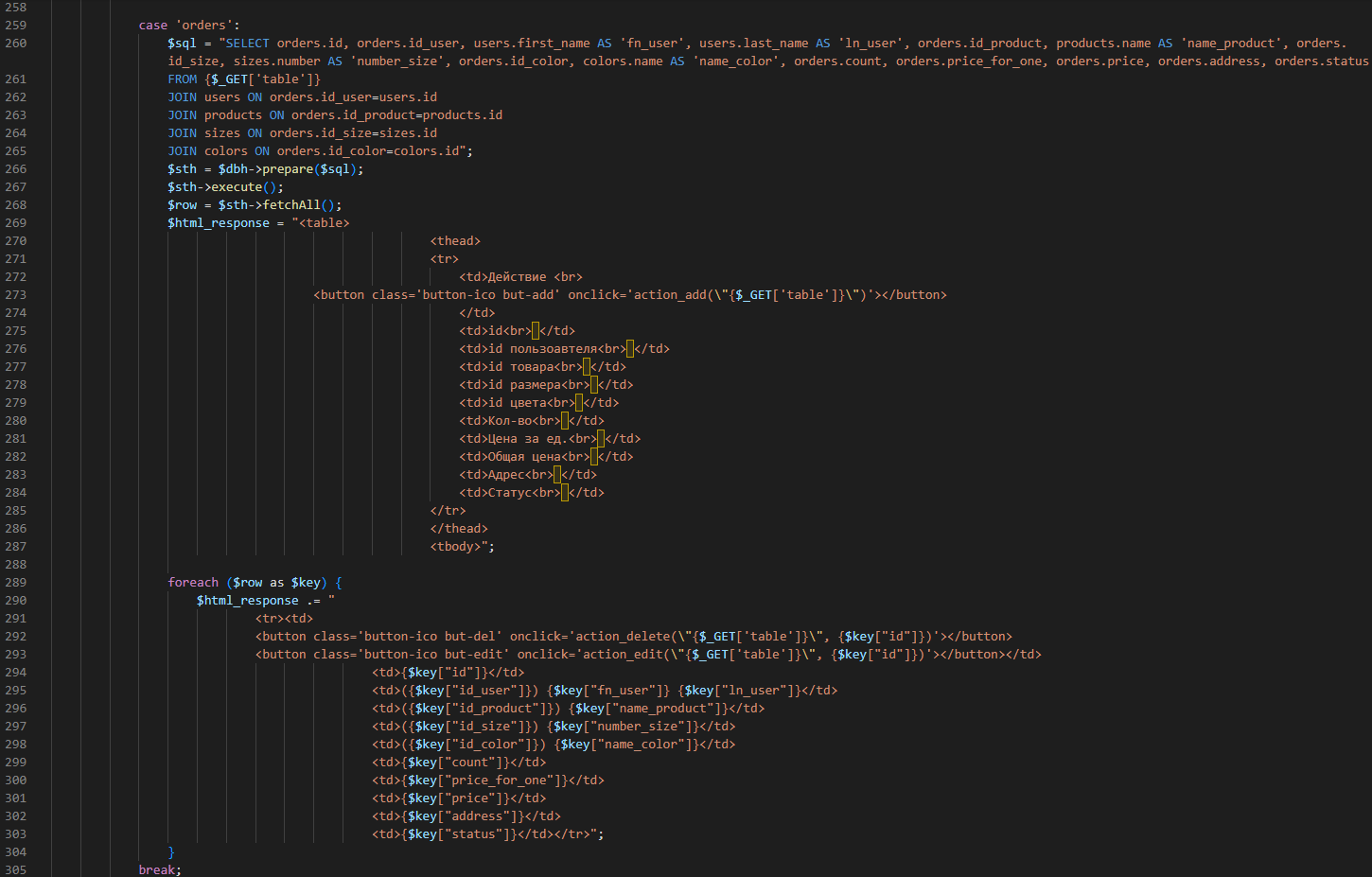
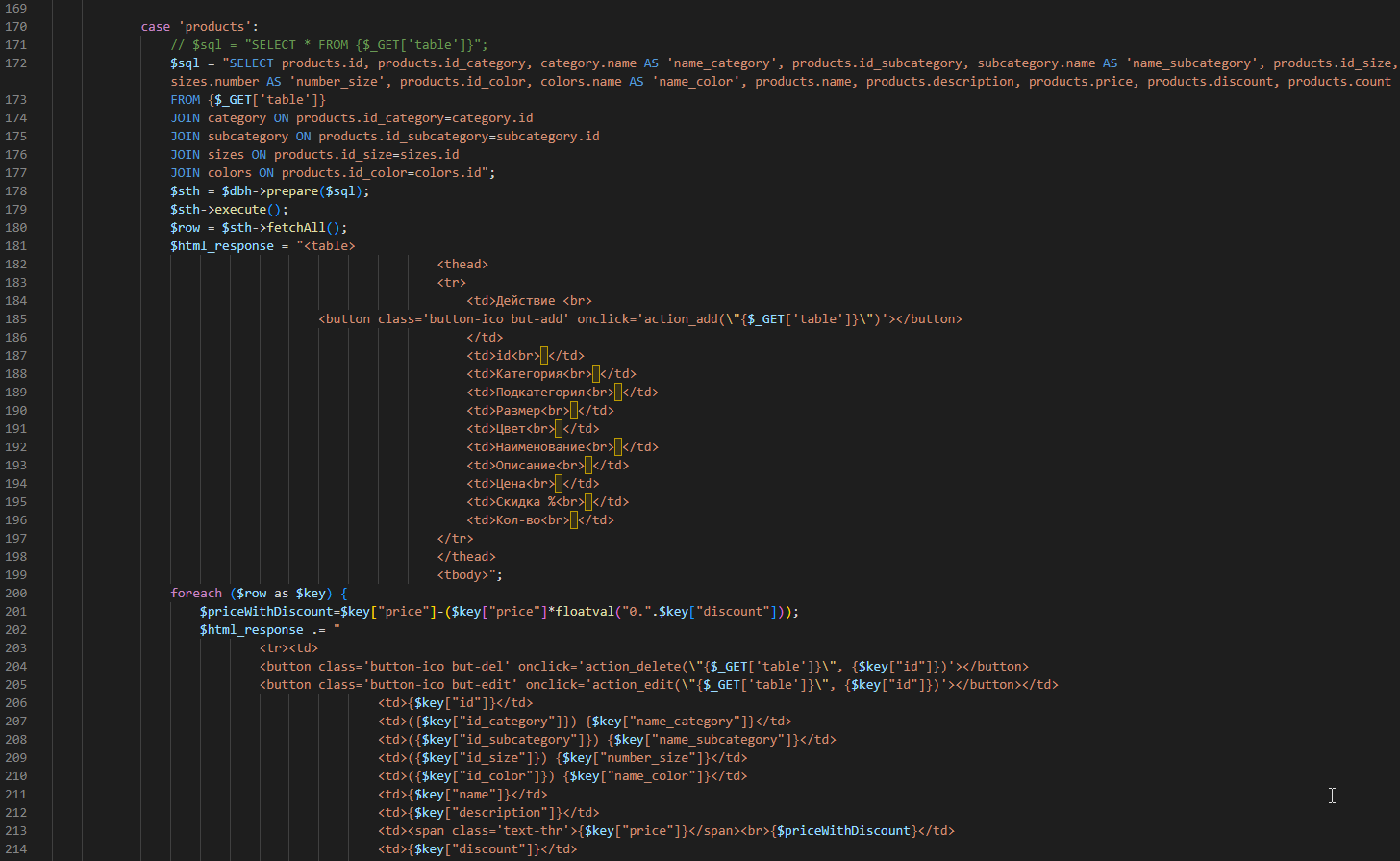
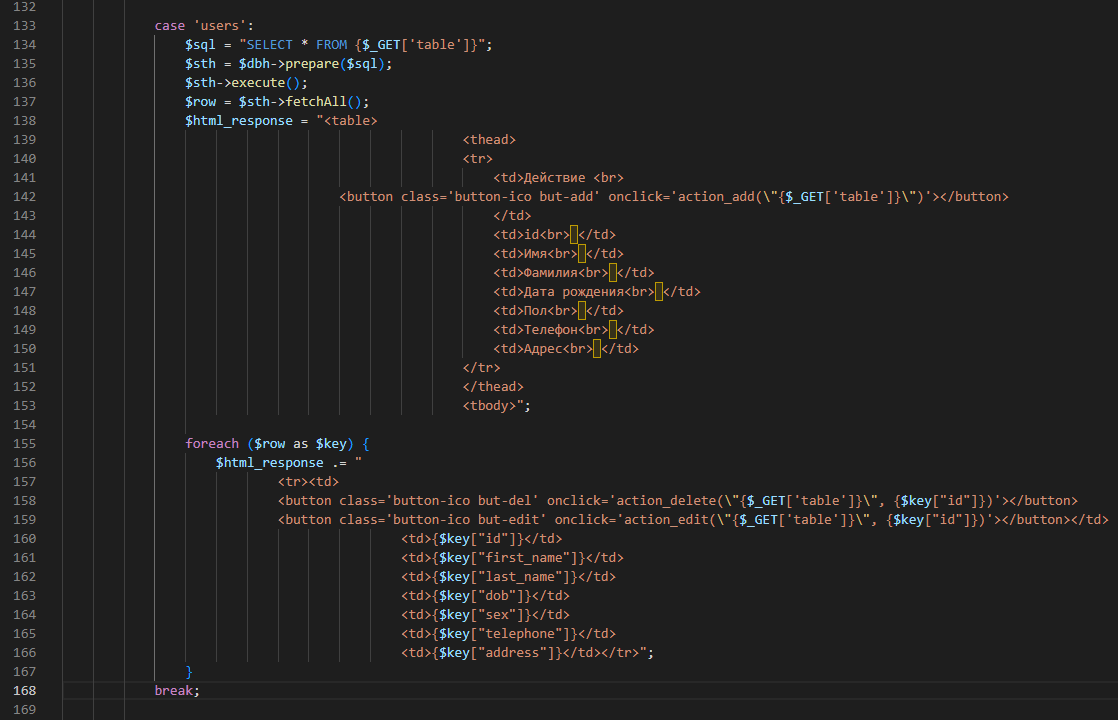
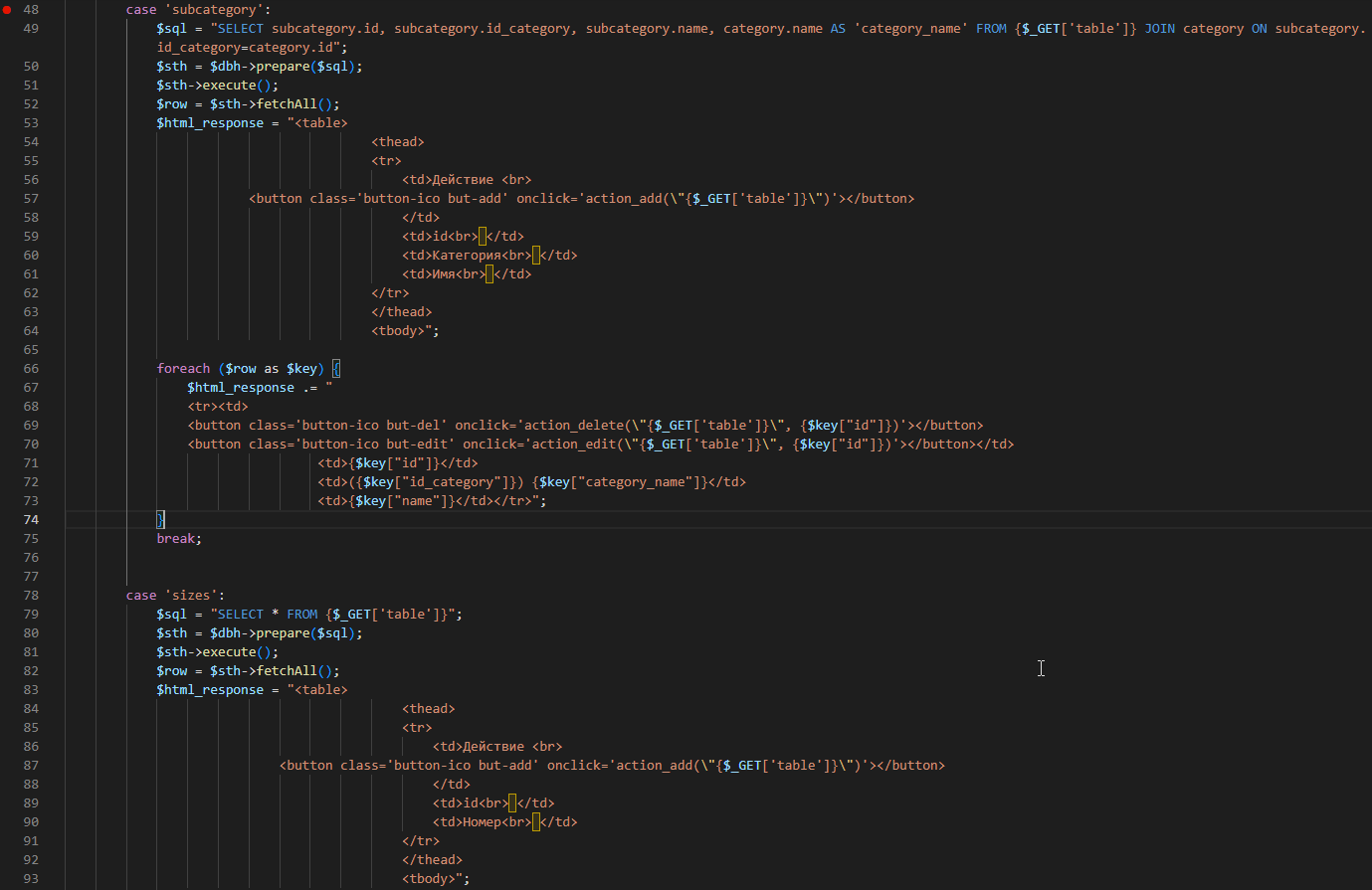
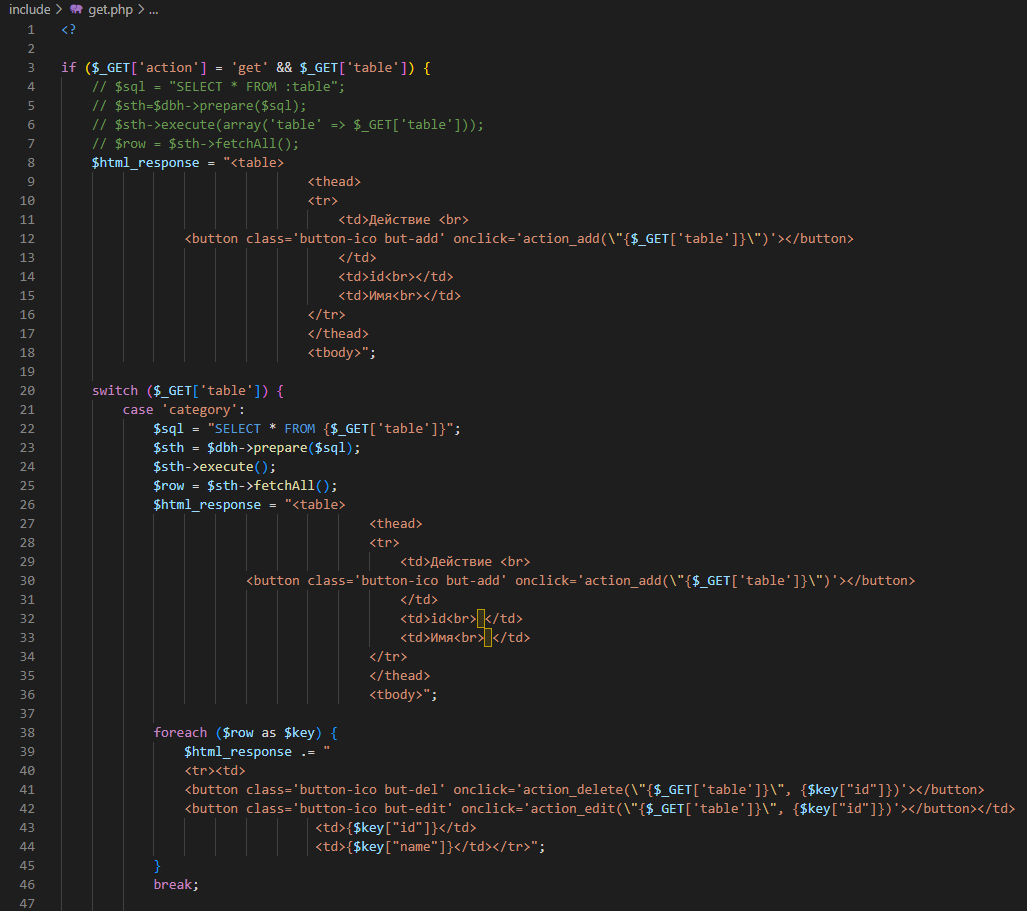


Рисунок № 2.5.16. Код файла «get.php»

По итогу, программа управления базой данных интернет-магазина имеет следующий вид (Рисунок № 2.5.17 – 2.5.30)

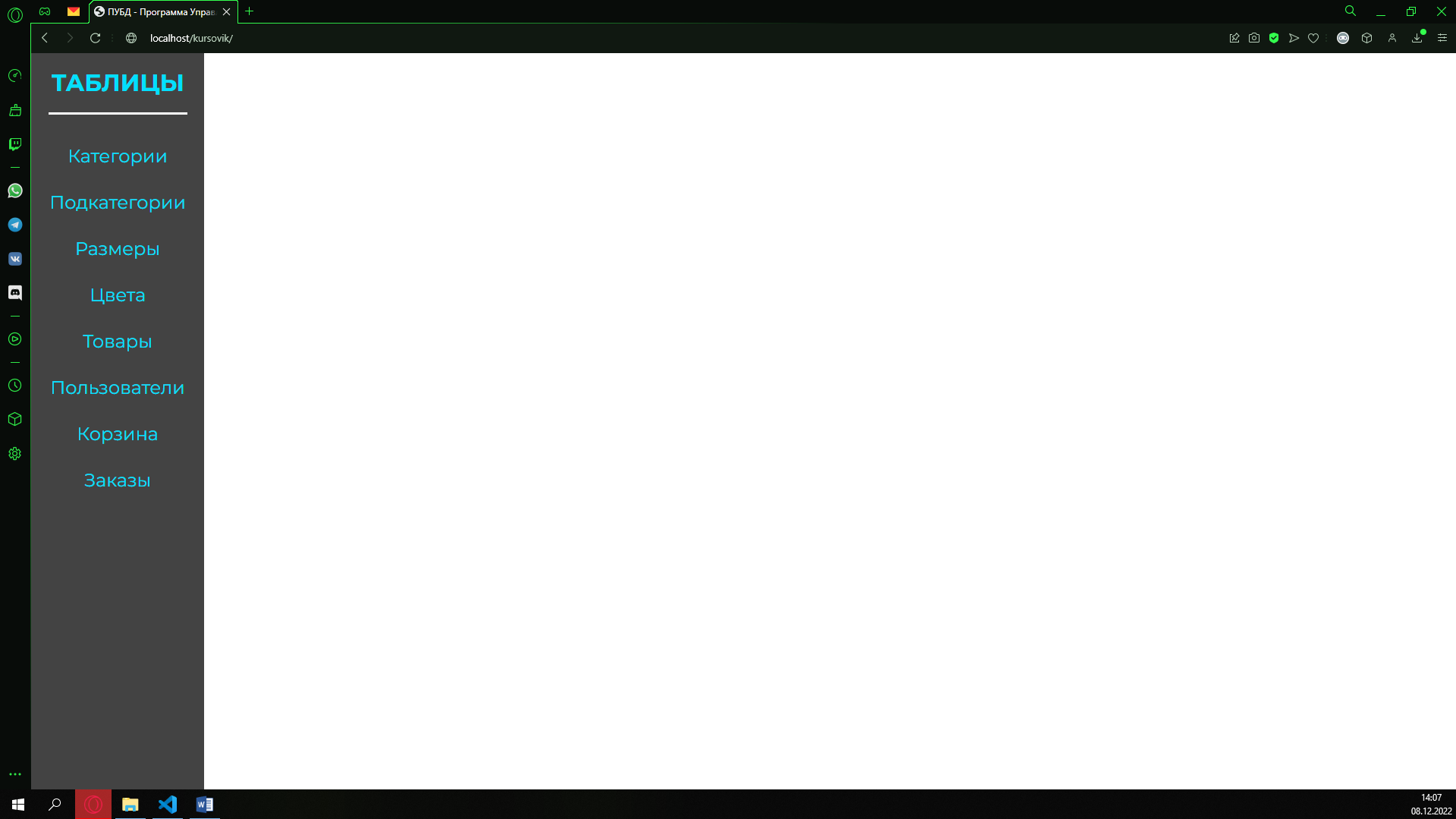


Рисунок № 2.5.17. Внешний вид ПУБД

Слева в программе находится «меню. При нажатии на интересующее название, отобразится содержимые таблицы – колонки и строки со значениями (рисунок № 2.5.18 – 2.5.25).

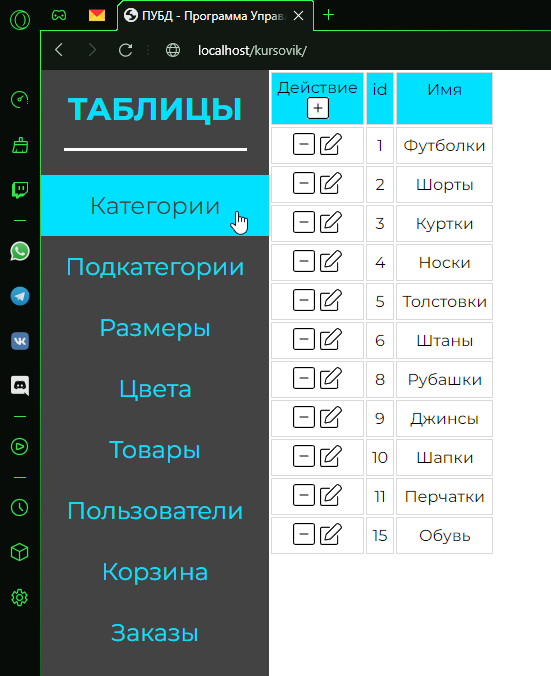


Рисунок № 2.5.18. Содержимое таблицы «Категории».

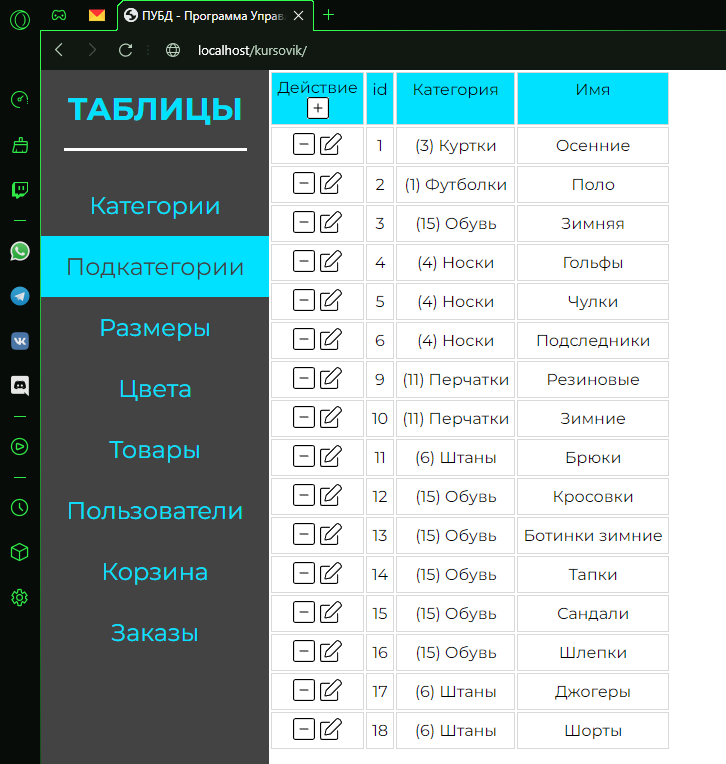


Рисунок № 2.5.19. Содержимое таблицы «Подкатегории».

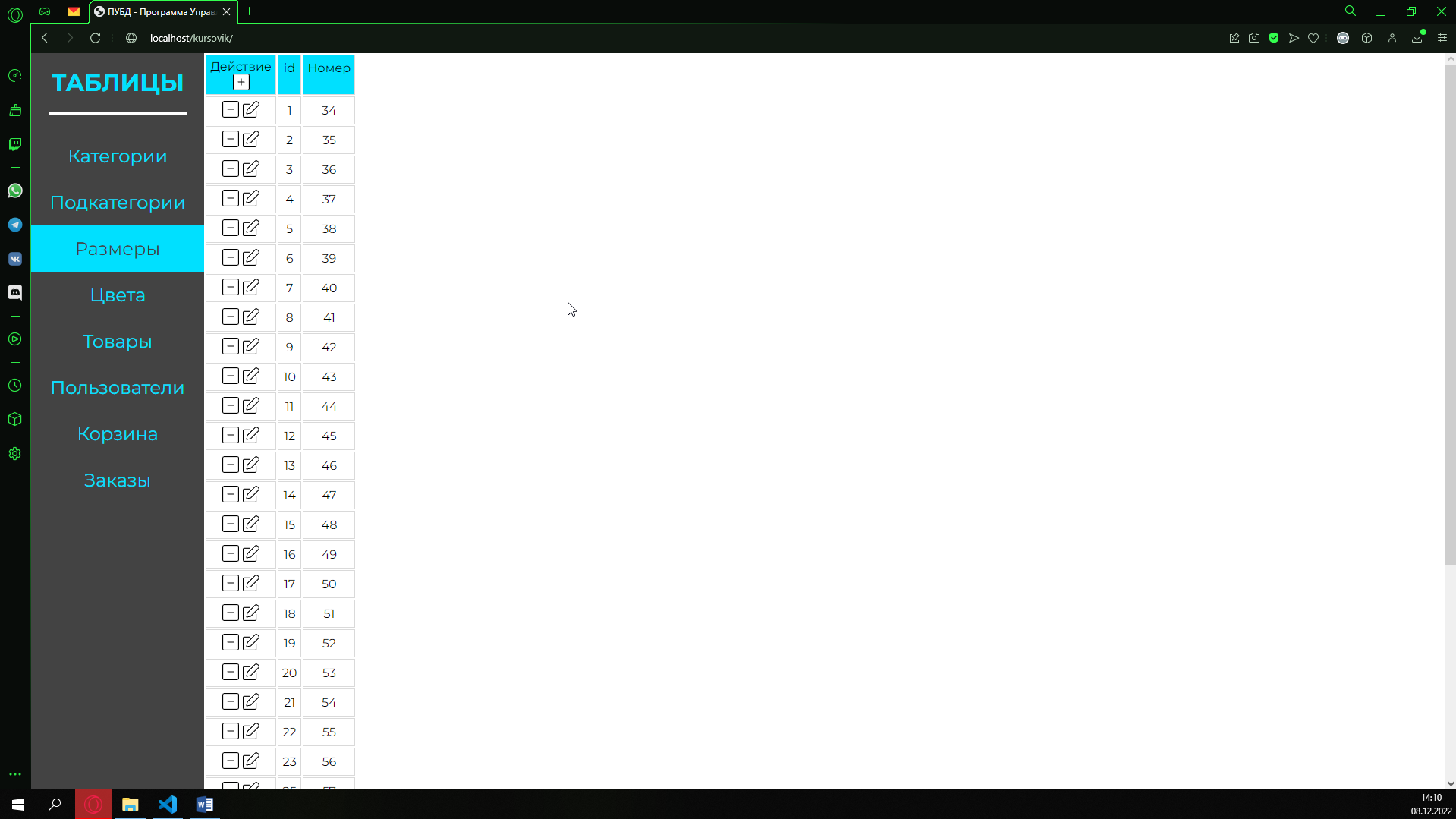


Рисунок № 2.5.20. Содержимое таблицы «Размеры».

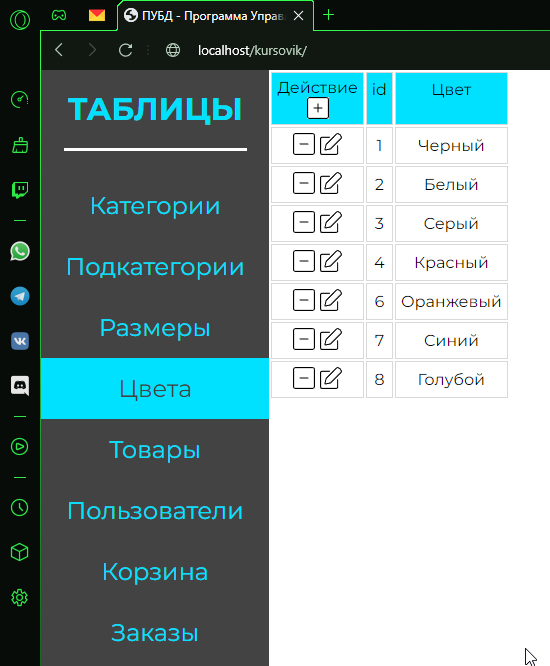


Рисунок № 2.5.21. Содержимое таблицы «Цвета».

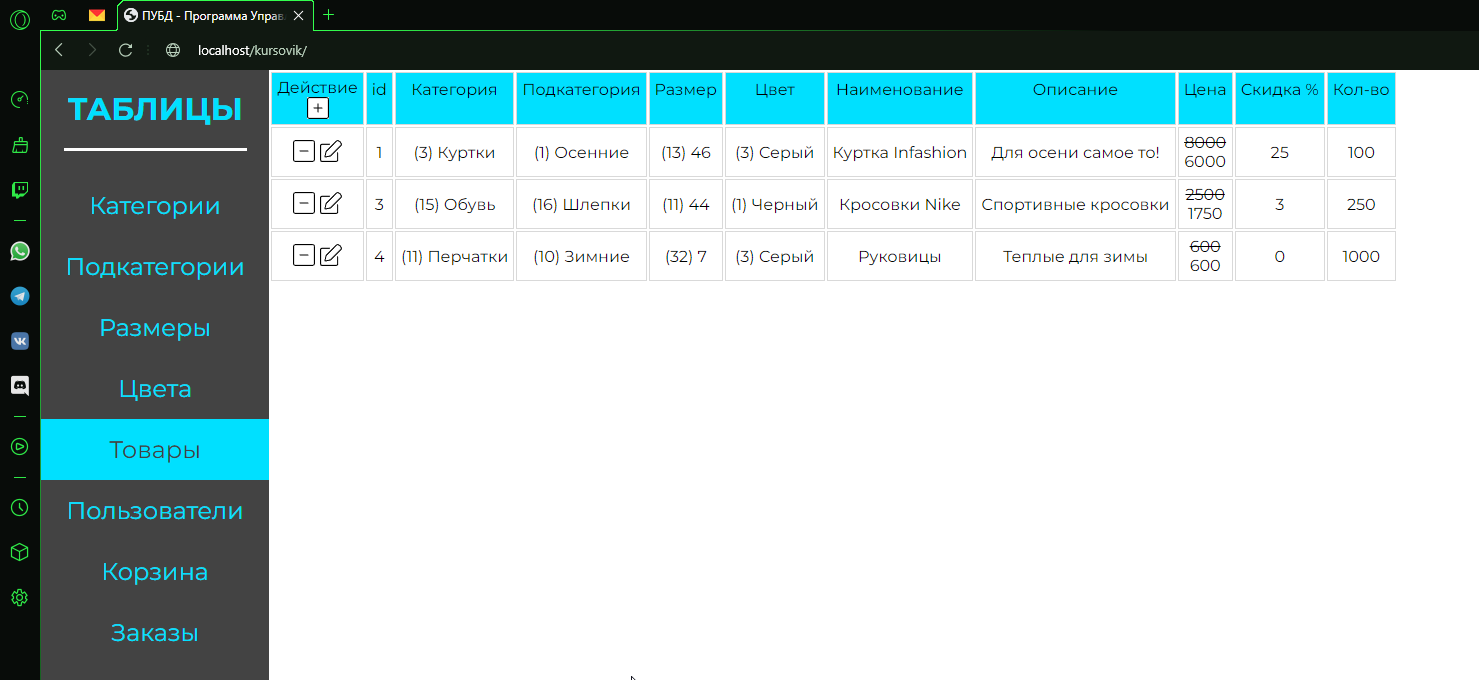


Рисунок № 2.5.22. Содержимое таблицы «Товары».

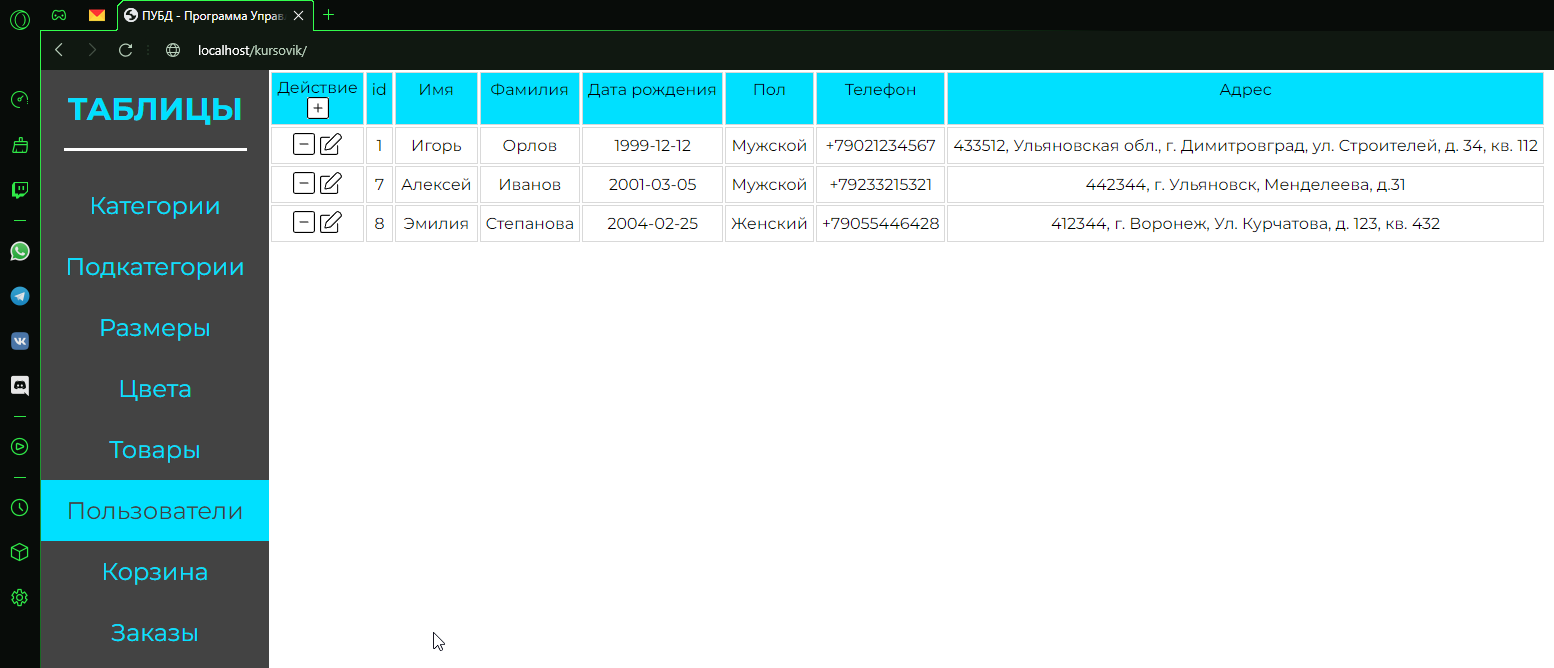


Рисунок № 2.5.23. Содержимое таблицы «Пользователи».

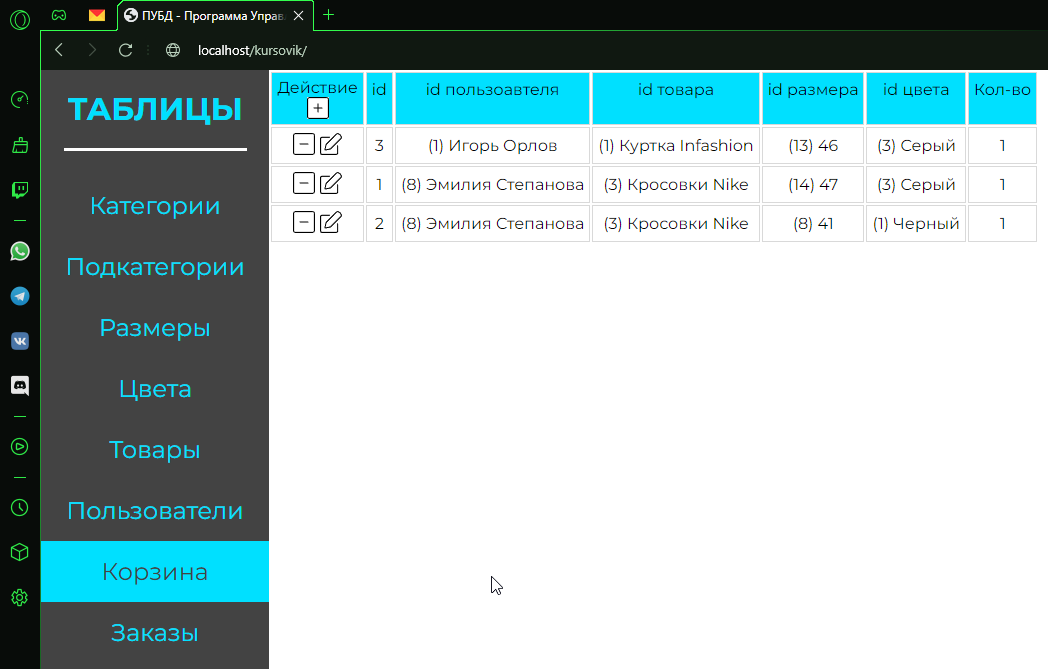


Рисунок № 2.5.24. Содержимое таблицы «Корзина».



Рисунок № 2.5.25. Содержимое таблицы «Заказы».

В программе имеется функция добавления строки в таблицу базы данных. Вызов функции происходит при нажатии на кнопку со знаком «+» (Рисунок 2.5.26).

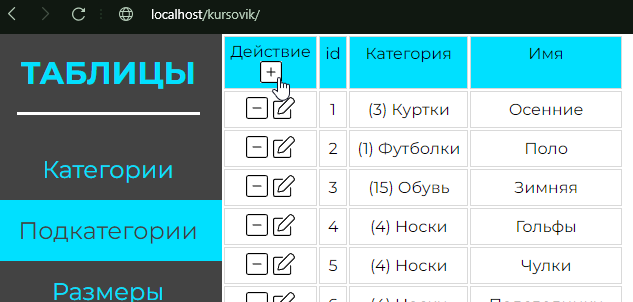


Рисунок № 2.5.26. Кнопка вызова функции добавления строки в таблицу БД.

После нажатии на кнопку добавления строки, функция начнет спрашивать значения у пользователя по каждой колонке (кроме идентификатора - id) таблицы (рисунок № 2.5.27 – 2.5.28).

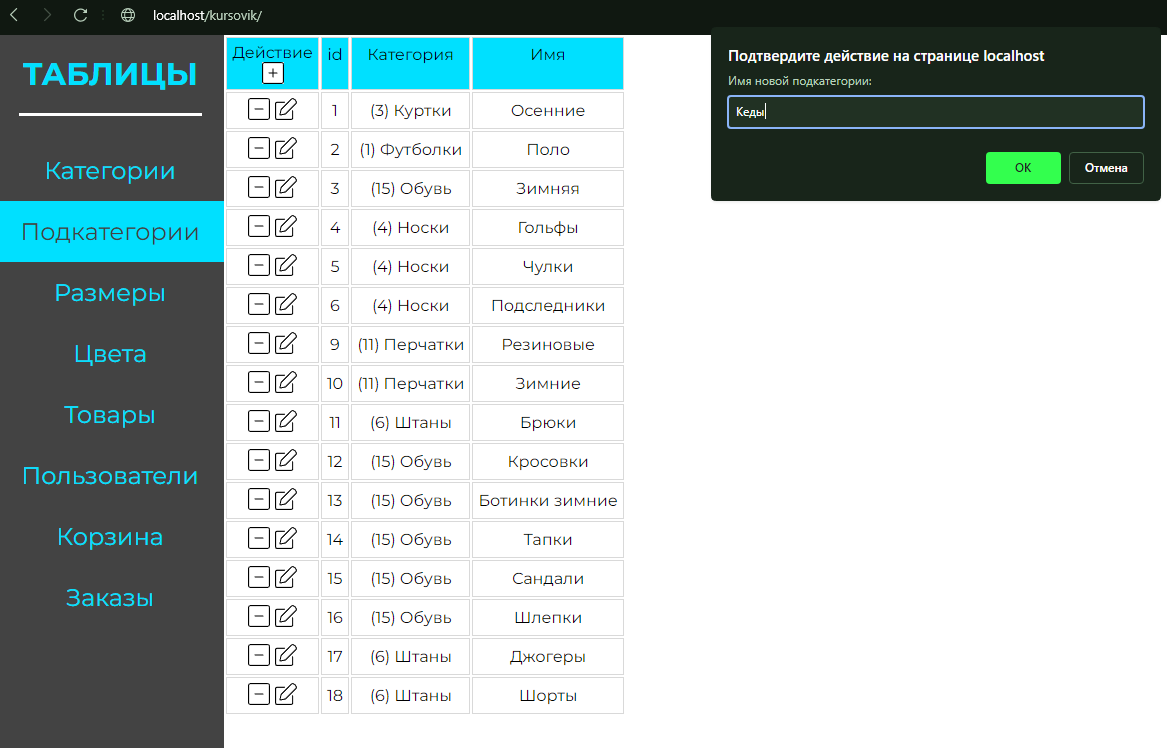


Рисунок № 2.5.27. Выполнение функции добавления строки в таблицу – Имя новой подкатегории.

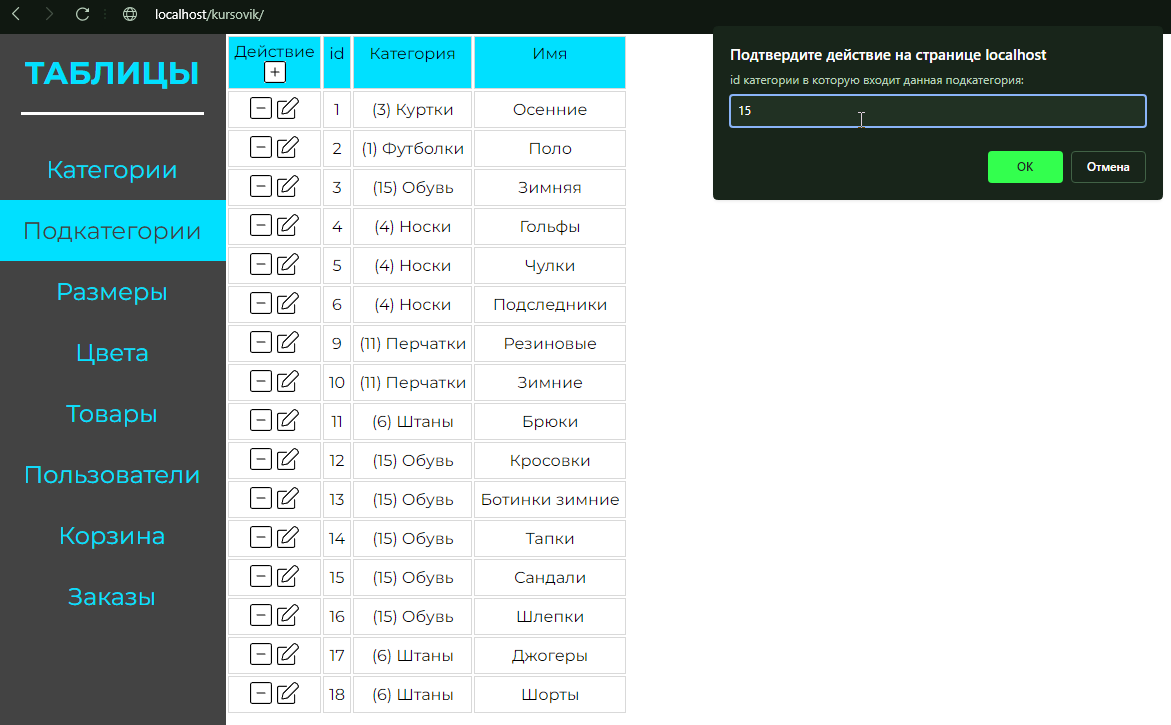


Рисунок № 2.5.28. Выполнение функции добавления строки в таблицу – Идентификатор родительской категории.

В результате выполнения функции, если данные введены корректно, то отобразится сообщении об успешном выполнении функции (рисунок № 2.5.29). Если же, данные введены некорректно (Неверный тип данных, пропущено значение, где не предусмотрено базой данных пустое значение в ячейке), то нам отобразится сообщение об ошибке (рисунок № 2.5.30)

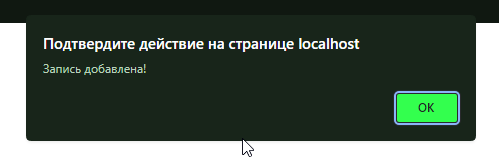


Рисунок № 2.5.29. Успешное выполнение функции добавления строки в таблицу.

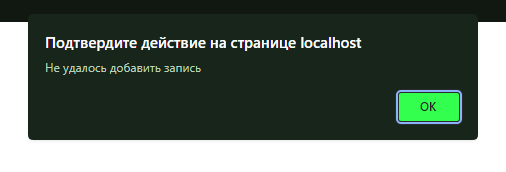


Рисунок № 2.5.30. Ошибка при выполнении функции добавления строки в таблицу.

После сообщения об успешном добавлении строки, в таблице можно наблюдать свежую запись (Рисунок № 2.5.31).



Рисунок № 2.5.31. Результат работы функции добавления строки в таблицу – новая запись.

Функция удаления строки. К примеру, в таблице имеется дубликат строки. Необходимо его удалить. Для этого, нужно нажать на кнопку со знаком «-», которая расположена в колонке «Действие» и на уровне строки, которую нужно удалить (рисунок № 2.5.32).

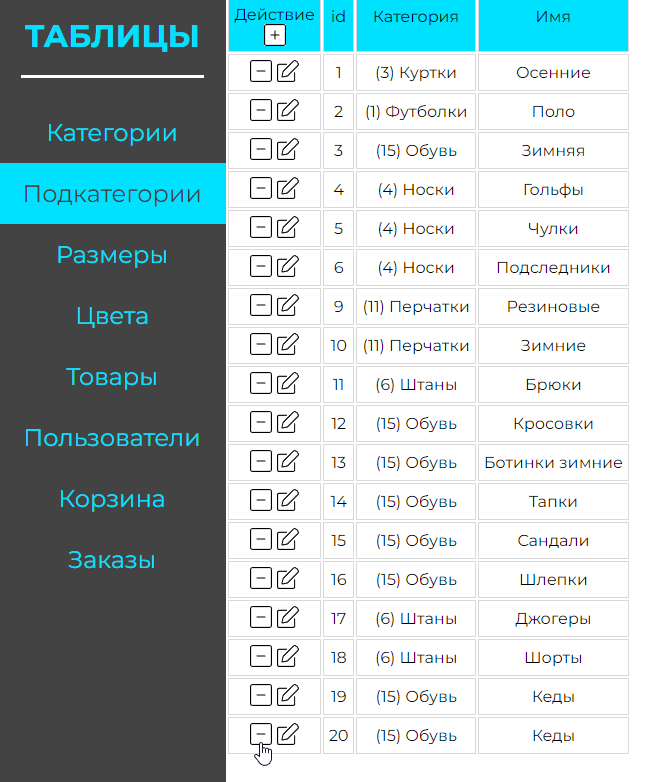


Рисунок № 2.5.32. Кнопка удаления строки.

После нажатия на кнопку удаления, функция просит подтвердить действия (рисунок № 2.5.33).

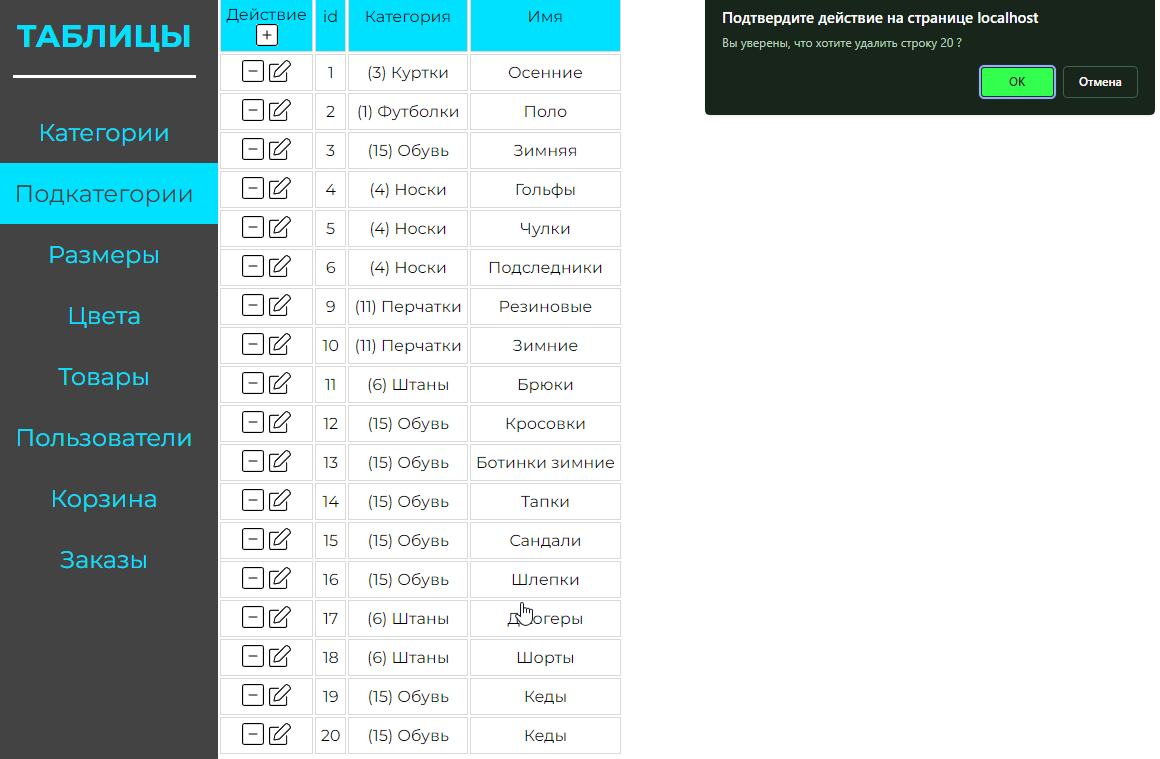


Рисунок № 2.5.33. Подтверждение удаления строки.

При положительном ответе на подтверждение своих действий удаления строки, пользователю выведет сообщение с результатом выполнения запроса (рисунок № 2.5.34).

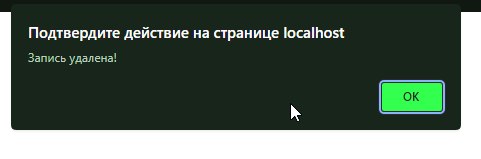


Рисунок № 2.5.34. Результат выполнения функции удаления.

По итогу выполнения функции, нужная строка из таблицы будет удалена (рисунок № 2.5.35).

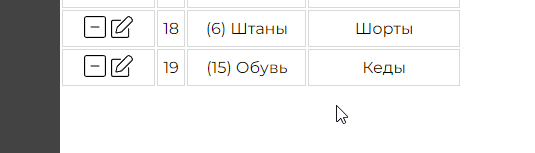


Рисунок № 2.5.35. Результат работы функции удаления - Строка №20 отсутствует в таблице.

Функция редактирования строки. К примеру, в строке таблицы имеется орфографическая ошибка, опечатка, данные несоответствующие действительности и др. Чтобы не удалять всю строку и не заполнять ее заново, была создана функция редактирования. Для этого, нужно нажать на кнопку со знаком «карандаша в квадрате», которая расположена в колонке «Действие» и на уровне строки, которую нужно отредактировать (рисунок № 2.5.36).

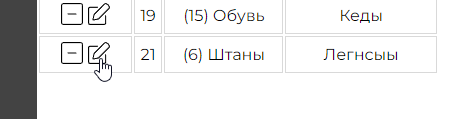


Рисунок № 2.5.36. Кнопка редактирования строки.

После нажатия на кнопку редактирования, функция просит подтвердить действия (рисунок № 2.5.37).

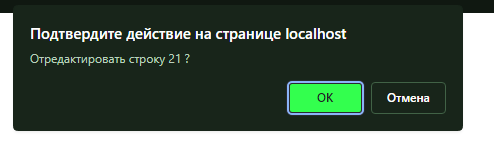


Рисунок № 2.5.37. Подтверждение редактирования строки.

Далее, функция начнет спрашивать новые значения у пользователя по каждой колонке. Если же интересно оставить старое значение, то стоит либо оставить поле ввода пустым либо ввести старое значение и нажать на кнопку «ОК», либо «Enter» (рисунок № 2.5.38 – 2.5.39).

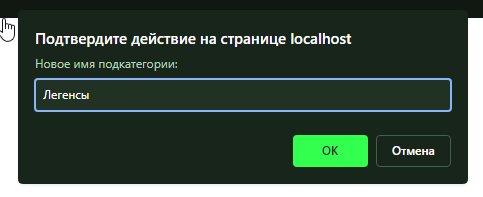


Рисунок № 2.5.38. Изменения значения ячейки – Ввод данных без ошибок.



Рисунок № 2.5.39. Пропуск этапа редактирования колонки и оставление старого значения.

По итогу, отобразится результат выполнения запроса на редактирование строки (Рисунок № 2.5.40).

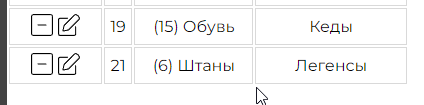
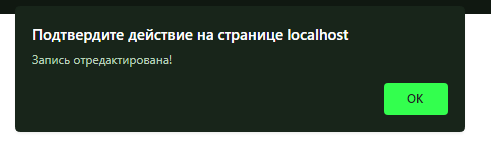


Рисунок № 2.5.40. Результат работы функции редактирования – В строке №21 наименование подкатегории («Легинсы») написано без ошибок.

**Заключение**

Базы данных - важнейшая составная часть информационных систем. Информационные системы предназначены для хранения и обработки больших объемов информации. Изначально такие системы существовали в письменном виде. Для этого использовались различные картотеки, папки, журналы, библиотечные каталоги и т.д. Любая информационная система должна выполнять три основные функции: ввод данных, запросы по данным, составление отчетов.

Преимущества, которые получает пользователь при использовании БД по сравнению с бумажными данными:

* + - Компактность - информация хранится в БД, нет необходимости хранить многотомные бумажные картотеки.
    - Скорость - скорость обработки информации (поиск, внесение изменений) компьютером намного выше ручной обработки.
    - Низкие трудозатраты - нет необходимости в утомительной ручной работе над данными.
    - Применимость - всегда доступна свежая информация.

Дополнительные преимущества появляются при использовании БД в многопользовательской среде, поскольку становится возможным осуществлять централизованное управление данными.

Современные системы управления базами данных обеспечивают как физическую (независимость от способа хранения и метода доступа), так и логическую независимость данных (возможность изменения одного приложения без изменения остальных приложений, работающих с этими же данными).

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями. СУБД обеспечивают правильность, полноту и непротиворечивость данных, а также удобный доступ к ним.

В ходе выполнения курсовой работы была создана базы данных интернет-магазина, которая содержит в себе таблицы: категории, подкатегории, размеры, цвета, товары, пользователи, корзина, заказы. Была также разработана программа. Оценивая ее преимущества, можно утверждать, что данная система обладает всеми необходимыми инструментами для отображения, создания, редактирования, удаления, хранения информации и ежедневного использования базы данных. Интерфейс программы прост и удобен. Созданная ПУБД позволяет получать данные о товарах, их категории, подкатегории, размерах, цвете, скидке, цене, количестве, о клиентах, совершающих покупки, о проданных им товарах, о сделанных клиентами заказах, об отложенных в корзину ими товаров.

# Список литературы

1. **Черноусова, А.М.** Создание и использование баз данных [Текст] / А.М. Черноусова. - Оренбург: ГОУ ОГУ, 2009. - 244 с.
2. **Дронов, В.А.** PHP и MySQL. 25 уроков для начинающих [Текст] / В.А. Дронов. - Санкт-Петербург: БХВ-Петербург, 2020. - 426 с.
3. **Бибо, Б.** Jquery в действии [Текст] / Б. Бибо, И. Кац. - Санкт-Петербург: Питер СПБ, 2017. - 528 с.
4. **Петин, В.** Сайт на AJAX под ключ Готовое решение для интернет-магазина [Текст] / В. Петин. - Санкт-Петербург: БХВ-Петербург, 2012. - 428 с.
5. **Кириллов, В.В.** Введение в реляционные базы данных [Текст] / В.В. Кириллов. - Санкт-Петербург: БХВ-Петербург, 2009. - 451 с.
6. **Янцев, В.В.** JavaScript и PHP. Content management system [Текст] / В.В. Янцев. - Санкт-Петербург: Лань, 2022. - 192 с.
7. **Васильев, А.** Программирование на PHP в примерах и задачах [Текст] / А. Васильев. - Москва: Эксмо, 2021. - 352 с.
8. **Дакетт, Д.** HTML и CSS. Разработка и дизайн веб-сайтов [Текст] / Д. Дакетт. - Москва: Эксмо, 2019. - 478 с.
9. **Дейт, К.Д.** Введение в системы баз данных [Текст] / К.Д. Дейт. - Санкт-Петербург: Издательский дом «Вильямс», 2001. - 270 с.
10. **Владимир, Д.** PHP и MySQL. 25 уроков для начинающих [Текст] / Д. Владимир. - Санкт-Петербург: БХВ-Петербург, 2021. - 426 с.
11. **Дубовик, Е.В.** Web на практике. CSS, HTML, JavaScript [Текст] / Е.В. Дубовик, А.П. Никольский, А.В. Кириченко. - Санкт-Петербург: Наука и Техника СПб, 2021. - 432 с.
12. **Кузнецов, М.В.** Самоучитель MySQL 5 [Текст] / М.В. Кузнецов, И.В. Симдянов. - Санкт-Петербург: БХВ-Петербург, 2010. - 560 с.
13. **Арно, Л.** Проектирование веб-API [Текст] / Л. Арно. - Москва: ДМК Пресс, 2020, - 440 с.
14. **МакГрат, М.** JavaScript для начинающих. 6-е издание [Текст] / М. МакГрат. - Москва: Эксмо, 2022. - 232 с.
15. **МакФарланд, Д.** JavaScript и jQuery [Текст] / Д. МакФарланд. - Москва: Эксмо, 2017. - 880 c.
16. **Васильев, Н.А.** JavaScript в примерах и задачах [Текст] / Н.А. Васильев. - Москва: Эксмо, 2019. - 721 c.
17. **Райордан, Р.** Основы реляционных баз данных [Текст] / Р. Райордан - Москва: Издательско-торговый дом «Русская Редакция», 2001. - 390 с.
18. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 4-е изд. [Текст] / Р.Никсон - Санкт-Петербург: Питер, 2016. - 768 с.
19. **Диго,** **С.М.** Базы данных. Проектирование и создание [Текст] / С.М. Диго - Москва: Издательский центр ЕАОИ, 2008. - 521 с.
20. **Кирупа, Ч.** JavaScript с нуля [Текст] / Ч.Кирупа - Санкт-Петербург: Питер СПБ, 2021. - 400 с.
21. **Пауэрс Ш.** Добавляем Ajax [Текст] / Ш. Пауэрс - Санкт-Петербург: БХВ-Петербург, 2009. - 448 с.
22. **Шмитт, К. Блессинг К.** Применение Web-стандартов CSS и Ajax для больших сайтов [Текст] / К. Шмитт, К. Блессинг - Москва: МК-Пресс, 2011. - 224 с.