

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес – информатика
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему « Автоматизация процесса продажи товаров через интернет-магазин»

Обучающийся

И.О. Полежаев
(И.О. Фамилия)



(личная подпись)

Руководитель

К. э.н., доцент, Т.А. Раченко
(ученая степень, звание, И.О. Фамилия)



Тольятти 2023



Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Аннотация

Ключевые слова: интернет-магазин, ООО «УМК-Сталь», автоматизация, технологии, php, mysql, javascript.

Общие сведения о работе: 66 страниц, 33 рисунка, 15 таблиц, 25 литературных источников.

Объектом исследования являются бизнес-процессы продажи товаров отдела сбыта ООО «УМК-Сталь» и автоматизация данных процессов посредством внедрения разработки интернет-магазина.

Цель работы – «Автоматизация процесса продажи товаров через интернет-магазин»

Разработка и последующее внедрение проекта интернет-магазина для организации ООО «УМК-Сталь» позволит автоматизировать существующие бизнес-процессы продаж, снизить зависимость от человеческого фактора и в дальнейшем масштабировать и развивать каналы продаж.

Описана структура организации, модель ведения бизнес-процессов, выявлены недостатки. Построена модель процессов в нотации IDEF0 «как есть» и «как должно быть». Поставлены задачи на проектирование. Произведён выбор технологий и инструментов для реализации интернет-магазина, произведено обоснование выбранных технологий и инструментов. Выбран вариант реализации СУБД и методология внедрения в рамках работы предприятия. Выбран дизайн сайта (интернет-магазина), контентное наполнение сайта и общая структура построения страниц.

Содержание

Введение.....	4
1 Теоретическая часть.....	6
1.1. Изучение организационной структуры и анализ бизнес-процессов ООО «УМК-Сталь».....	6
1.2. Изучение аппаратных и программных средств отдела сбыта.....	9
1.3. Постановка задачи.....	11
1.3.1 Анализ потребностей конечного пользователя	11
1.3.2 Реализуемый функционал	12
1.3.2 Пользовательский интерфейс	13
1.3.3 Масштабируемость	14
1.4. Выбор технологий для разработки интернет-магазина	15
2 Проектирование интернет-магазина	18
2.1 Планирование работ.....	18
2.2 Карта сайта.....	19
2.2 Диалог пользователя с системой	22
2.2.1 Транзитивная сеть	22
2.2.2 Диаграмма вариантов использования	25
2.3 Описание бизнес-логики интернет-магазина	27
2.4 Логическое проектирование базы данных	31
2.5 Проектирование базы данных на физическом уровне	35
3 Разработка интернет-магазина.....	41
3.1 Функции PHP	41
3.2 Функции JavaScript	47
Демонстрация работы интернет-магазина.....	50
Заключение	63
Список используемой литературы	64

Введение

Современные информационные технологии задают новые требования и тенденции практически в каждой области человеческих взаимоотношений, в том числе и рыночных. На сегодняшний день, невозможно эффективно реализовать свои задачи в области бизнеса, не имея эффективной платформы для реализации поставленных задач. Одной из самых эффективных площадок для реализации товаров и услуг становятся веб сайты и пользуются огромным спросом.

Интернет-торговля становится неотъемлемой частью продаж как таковых. Такой способ ведения бизнеса имеет неоспоримые преимущества перед традиционным способом продавать товары.

Частичная или полная автоматизация процесса ведёт к сокращению издержек, что является целью для всех организаций или предприятий, занимающихся торговлей. Таким способом замещается ручной труд человека и освобождаются финансовые ресурсы.

Объектом исследования являются бизнес-процессы продажи товаров отдела сбыта ООО «УМК-Сталь».

Предметом данной работы является автоматизация процесса продажи товаров через интернет-магазин» ООО «УМК-Сталь».

Актуальность работы заключается в растущих потребностях организации в масштабировании собственных бизнес-процессов. Разработка интернет-магазина позволит значительно сократить издержки и повысить уровень оптимизации бизнес-процессов, а также увеличит продажи.

Целью данной работы является разработка интернет-магазина для реализации продаж ООО «УМК-Сталь»

Для достижения поставленных целей необходимо исследовать бизнес-процессы отдела сбыта оценить степень автоматизации и обосновать

необходимость разработки. Также необходимо оценить необходимость разработки с помощью опросов персонала.

Разрабатываемый интернет-магазин необходимо снабдить базовыми функциями для возможности его полноценной работы. Нужно спроектировать понятный и удобный для покупателя интерфейс витрины товаров, возможность провести полный цикл от оформления до покупки.

Со стороны администратора магазина необходимо спроектировать интерфейс управления.

Задачами данной работы является следующий список:

- исследование структуры предприятия и отдела сбыта в частности;
- оценка степени автоматизации в процессах отдела сбыта;
- предложение вариантов улучшения в ведении бизнес-процессов отдела сбыта;
- составление задания на разработку;
- разработка интернет-магазина;
- тестирование разработанного интернет-магазина;
- заключение об успешности проектирования разработки.

Работа будет включать в себя три главы.

В первой главе описывается организационная структура предприятия в целом с детализацией деятельности отдела сбыта, выявляются слабые стороны существующих бизнес-процессов и ставится задача на разработку.

Во второй главе проектируется интернет-магазин, в задачи проектирования входят описание пользовательского интерфейса, применение технологий разработки и проектирование базы данных.

В третьей главе описывается процесс тестирования разработанного магазина с демонстрацией экранных форм.

В заключении подводятся итоги выполненной работы и делается вывод и применимости разработанного приложения интернет-магазина.

1. Теоретическая часть

1.1. Изучение организационной структуры и анализ бизнес-процессов ООО «УМК-Сталь»

Компания УМК-Сталь является управляющей организацией Надеждинского металлургического завода, завода МЗ Электросталь Тюмени и Богословского рудоуправления. Реализует стальной прокат строительного и конструкционного назначения: арматура, уголок, круги, шестигранник, квадрат, стальная заготовка.

Таким образом, УМК-Сталь является крупным предприятием с большим количеством сотрудников и сложной организационной структурой. Предприятие не только производит, но и реализует продукцию собственного производства.

Процесс реализации продукции выглядит следующим образом: покупатель (юридическое лицо) отправляет заявку на покупку продукции в управление отдела сбыта. Отдел сбыта, получив заявку на электронную почту, формирует запрос в соответствии с номенклатурой товаров, далее информация запрашивается в производственном управлении предприятия (о наличии, планах выпусках, сроках производства), после подтверждения информация об этом доходит до покупателя и заключается договор поставки.

Организационная структура предприятия представлена на рисунке 1.

Как видно из рисунка 1, - структура предприятия представлена в виде иерархической модели. Наивысший приоритет в управлении имеет директор предприятия, которому подчиняются отделы организации, а также коммерческий директор и технический директор. Структура управления внутри отделов также соответствует иерархической структуре, а именно, -

начальникам отделов подчиняются все сотрудники отделов, а сами начальники отделов подчиняются директору предприятия.

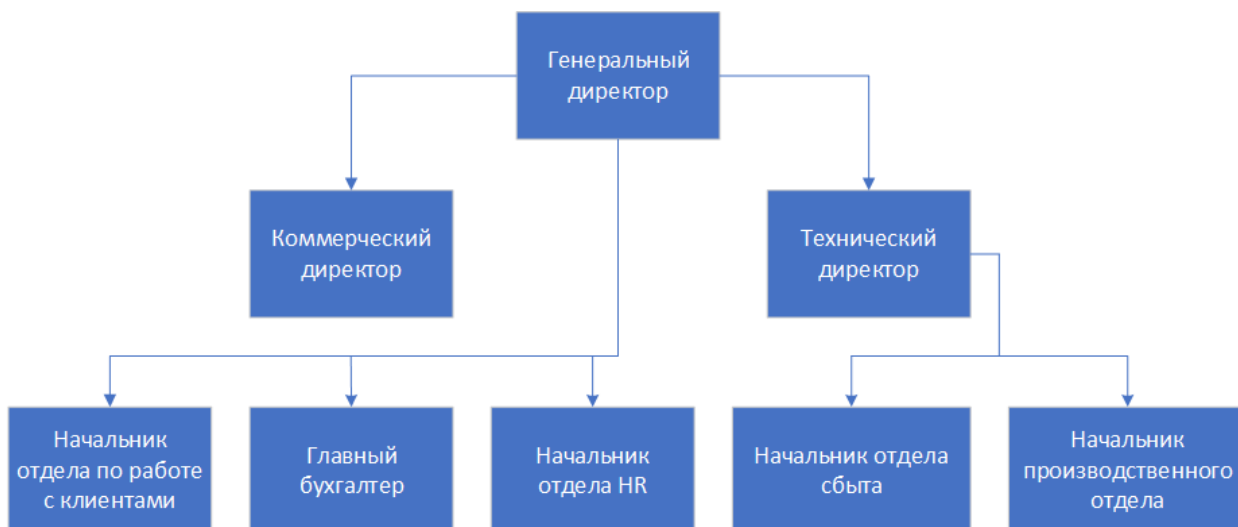


Рисунок 1 – Организационная структура предприятия

В данной работе рассматривается детализация работы и анализ деятельности непосредственно отдела сбыта, через который осуществляются все торговые операции [8].

Реализуемый торговый функционал можно представить в нотации IDEF0 с помощью контекстной диаграммы с последующей её декомпозицией.

Контекстная диаграмма представлена на рисунке 2.

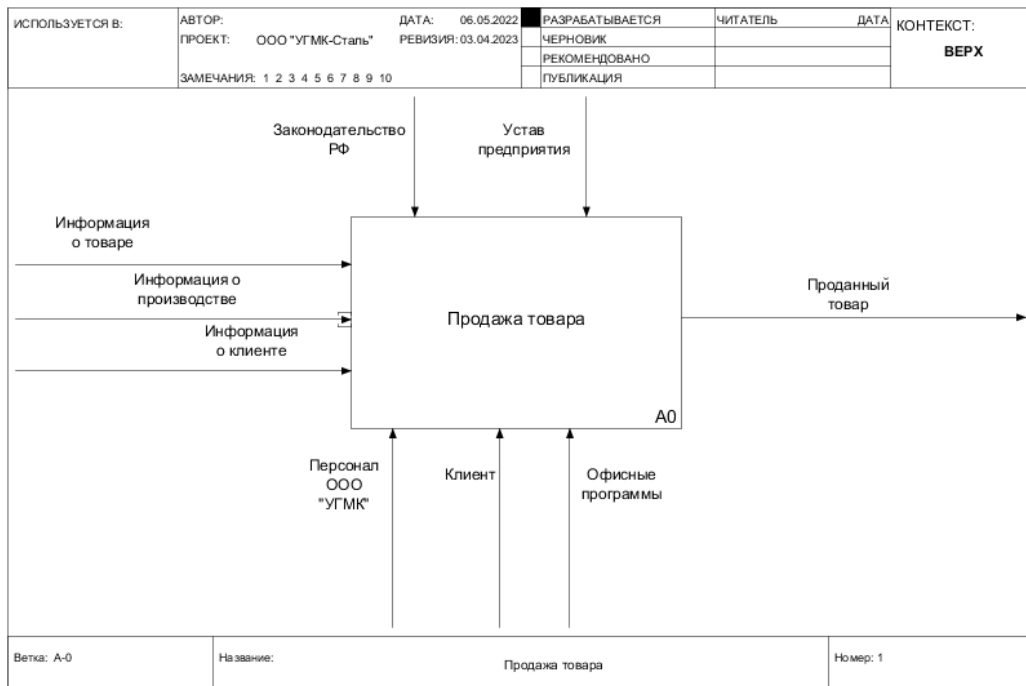


Рисунок 2 – Контекстная диаграмма

Декомпозиция контекстной диаграммы представлена на рисунке 3, данная диаграмма детализирует описание бизнес-процесса.

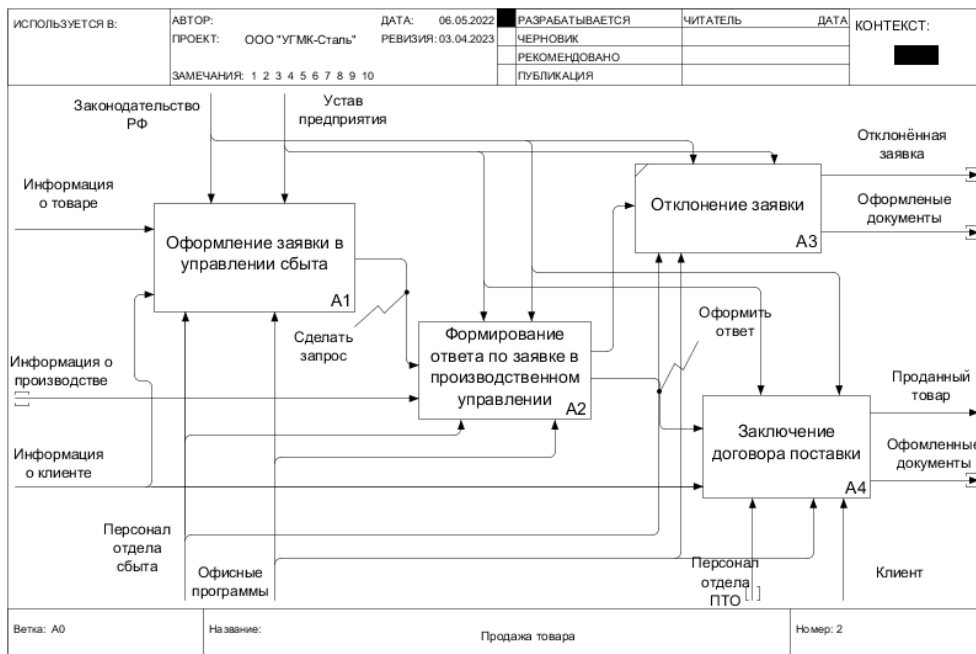


Рисунок 3 – Декомпозиция контекстной диаграммы

Как видно из рисунка 2 и 3 – процесс продажи продукции в УМК-Сталь организован в значительной степени с участием человеческого фактора и при отсутствии автоматизации, в той или иной степени. Процесс подачи заявок не автоматизирован. В каждой цепи так или иначе участвуют люди, проверяющие информацию вручную без использования централизованной базы данных. При таком подходе информация может быть неполной, неточной или даже ложной, что в конечном счёте приведёт к потере прибыли. Также существенным недостатком является сам принцип формирования заказа и ограничение заказов лишь юридическими лицами [1].

Исходя из этого для УМК-Сталь рациональным будет предложить разработку специальной системы по автоматизации продаж и учёта продукции. Подобный функционал может быть реализован посредством интернет-магазина, где каждый шаг по оформлению автоматизируется, а все данные, необходимые для ведения бизнес-процессов находятся централизованно в одном месте и надёжно защищены.

1.2. Изучение аппаратных и программных средств отдела сбыта

Современные организации с большим количеством персонала и частыми торговыми операциями в обязательном порядке используют программные и технические средства, упрощающие данные процессы, а также сокращающие время обработки информации. В частности, в отделе сбыта организована структура локальной сети из объединённых компьютеризированных рабочих мест, то есть, можно сказать, что работа сотрудников отдела организована посредством персональных компьютеров, объединённых в локальную сеть. Характеристики компьютеров сети, следующие:

- Операционная система Windows 10

- Последние установленные обновления для корректной работы
- Компьютер с процессором от 1.5 ГГц.
- Оперативная память не менее 1 Гб
- Монитор
- Принтер
- Установленный современный веб-браузер
- Доступ в интернет
- Локальный веб-сервер

В структуру локальной сети входит веб-сервер, центральный компьютер, который хранит результаты обработки информации, документы, отчёты. Документооборот осуществляется с помощью ручного труда, в качестве программных средств используются офисные пакеты программ семейства Microsoft Office. Следовательно, при обработке информации на этапе ввода, а также на этапе преобразования отсутствуют элементы автоматизации, играет существенную роль человеческий фактор [2].

Далее в таблице 1 представлен анализ аппаратного и программного обеспечения в организации.

Таблица 1 - Анализ аппаратного и программного обеспечения в организации

№	Техническое/программное обеспечение	Количество
Техническое обеспечение		
1	Компьютер с процессором от 1.5 ГГц	25 шт.
2	Оперативная память не менее 1 Гб	25 шт.
3	Монитор	25шт.
4	Принтер	6 шт.
Программное обеспечение		
1	Операционная система Windows 10	25 шт.
2	Локальный веб-сервер	25 шт.

Продолжение таблицы 1

3	Веб-браузер Google Chrome	25 шт.
4	Пакет прикладных программ Microsoft Office 2019	25 шт.

1.3. Постановка задачи

Объектом исследования в данной работе является отдел сбыта, автоматизируемым процессом – продажи. В качестве методов исследования потребностей будет применен опрос сотрудников и клиентов.

В данной работе необходимо разработать Интернет-магазин, который будет работать по B2C-схеме. Данная схема позволит УМК-Сталь расширить свою аудиторию продаж, предлагать свой товар не только юридическим лицам, но и физическим лицам, а также индивидуальным предпринимателям, которые также являются физлицами.

Интернет-магазин – это в первую очередь сайт, представленный в сети интернет, который управляется с помощью определённых инструментов и технологий [1].

Необходимо определить набор требований к разрабатываемому сайту и выбрать стек технологий для реализации, но прежде всего необходимо учесть пожелания сотрудников отдела сбыта, а также персонала, который задействован в автоматизируемом бизнес-процессе.

1.3.1 Анализ потребностей конечного пользователя

Чтобы выяснить потребности сотрудников, которые так или иначе участвуют в бизнес-процессе необходимо провести беседу и записать результаты в таблицу. Разделим пользователей на категории и запишем общие пожелания.

Далее в таблице 2 представлены результаты анкетирования сотрудников отдела сбыта и клиентов.

Таблица 2 – Результаты анкетирования сотрудников отдела сбыта и клиентов

Категории работников	Общие пожелания
Начальник отдела сбыта	Необходимо современное решение учёта взаимоотношений с клиентами, чтобы исключить человеческий фактор и повысить производственные показатели.
Сотрудник отдела сбыта	Оформление заявок происходит при помощи офисных пакетов программ, это является рутинной работой, необходимо другое программное решение, где клиент взаимодействует с заказом с помощью сайта.
Клиент	Неудобно делать заказы, не видна номенклатура товара в актуальном состоянии, желательно разработать интернет-магазин.
Сотрудник отдела ПТО	Разработка интернет магазина позволила бы брать данные из заказов в административной части сайта для оформления договоров.

Конечными пользователями интернет-магазина будут в первую очередь клиенты, а также сотрудники отдела сбыта. Учёт пожеланий при проведении опроса позволил выявить потребности в разработке именно интернет-магазина.

1.3.2 Реализуемый функционал

Определим требования к функционалу разрабатываемого сайта:

- просмотр каталога товаров;
- просмотр подробной информации о товаре;
- просмотр информации о доставке и оплате;
- просмотр контактной информации;

- выбор товаров по категории;
- добавление товаров в корзину;
- изменение количества товаров в корзине;
- удаление товаров из корзины;
- оформление заказа;
- добавление новых товаров в каталог;
- добавление скидок товарам;
- добавление меток популярности товарами;
- сортировка товаров по заданным критериям;
- авторизация на сайте;
- регистрация на сайте.

В таблице 3 наглядно представлено распределение функций сайта в зависимости от типа пользователя.

Таблица 3 – Распределение функций сайта в зависимости от типа пользователя

Пользователь	Действие
Покупатель, Администратор	Просмотр в браузере
Покупатель	Действия с корзиной, Регистрация, Авторизация, Оформление заказа
Администратор	Вход в панель администрирования, Учёт всех заказов покупателей, Наполнение витрины товарами, удаление товаров

1.3.2 Пользовательский интерфейс

При дальнейшем проектировании интернет-магазина нужно создать описание пользовательского интерфейса, определить все необходимые для реализации функционала элементы.

«Так как интернет-магазин будет отображаться в браузере, то основным критерием внешнего вида будет корректное отображение страницы в современных браузерах» [5].

Взаимодействовать с браузером для авторизации и регистрации покупателя пользователь будет посредством веб-форм. Необходимо, чтобы формы содержали требуемый для полноценной работы магазина набор полей:

- Имя пользователя (логин)
- Электронная почта пользователя
- Пароль для входа в систему
- Телефон пользователя

1.3.3 Масштабируемость

«При проектировании интернет-магазина необходимо учесть возможность его дальнейшего развития» [7]. Например, может понадобиться дополнительный функционал как на уровне дизайна, так и на уровне исполнения бизнес-логики. «Чтобы учесть возможные будущие требования к развитию интернет-магазина необходимо использовать модульный подход к разработке и паттерны проектирования, в частности при разработке веб-интерфейса необходимо использовать заранее определённые наборы классов для оформления внешнего вида страниц, что позволит в дальнейшем добавлять элементы интерфейса и стилизовать их заранее подготовленными классами. Со стороны бизнес логики также необходимо организовать архитектуру приложения со стороны сервера, используя для этого паттерн MVC(model-view-controller), который будет описан далее и который позволит разделить логику и внешний вид, а также разбить весь код на модули. При появлении нового функционала и новых страниц можно будет добавлять новые модули, следуя правилам MVC, что позволит упростить внедрение

нового кода, а также существенно упростит поддержку существующего» [10].

1.4. Выбор технологий для разработки интернет-магазина

Для реализации функционала интернет-магазина и связи его с пользовательским интерфейсом в современной веб-разработке используется язык гипертекстовой разметки HTML. Он является по сути единственным, а значит и самым распространенным языком разметки, используемым для создания сайтов. Преимущества HTML заключаются в том, что он поддерживается всеми браузерами, относительно легок для освоения из-за простоты синтаксиса и работает с другими средствами для разработки сайтов.

Для реализации функционала интернет-магазина с помощью HTML нужно создать структуру страницы, на которой будет располагаться весь контент магазина. HTML-документ – это файл с расширением .html или .htm. на конце, доступный для просмотра через любой браузер, к примеру, через Google Chrome, Safari, Mozilla Firefox и т.п. Браузер визуализирует считанный HTML-файл в формате, удобном для просмотра пользователями.

Чтобы интернет-магазин имел единый стиль оформления, используется язык CSS. CSS (Cascading Style Sheets) – это язык описания внешнего вида документов HTML [23].

CSS задает стили элементам HTML, определяет цвет, шрифт, размеры, расположение и прочие элементы оформления. Принцип работы CSS заключается в том, что стили применяются к элементам HTML путем указания селектора — это название элемента, класса или идентификатора.

CSS позволяет оформлять не только отдельные элементы, но и весь документ целиком. Для этого можно использовать глобальные стили,

которые будут применяться ко всем элементам страницы. Для взаимодействия с HTML, CSS использует элементы HTML как "контейнеры" для стилей.

«CSS позволяет создавать несколько файлов стилей для одной страницы и использовать различные типы селекторов для задания стилей элементам. Также с помощью CSS можно создавать анимацию, определять псевдоэлементы и создавать интерактивные элементы веб-страниц» [11].

Данная часть структуры сайта называется фронтенд. Для того, чтобы фронтенд работал корректно, а также не перегружал мощности устройств пользователей, необходимо создать надежный бэкенд.

Бэкенд отвечает за стабильную работу сайта, хранение и структурирование данных и их безопасность, а также реализует ядро бизнес-логики сайта [19].

На данный момент существует множество средств для разработки бэкенда, каждый из которых обладает своими преимуществами и недостатками.

Node.js – это средство для разработки бэкенда, написанное на языке JavaScript. Оно позволяет разрабатывать масштабируемые и быстрые приложения с использованием неблокирующих операций ввода/вывода.

ASP.NET — представляет собой фреймворк, который используется для создания сайтов на таких языках, как Visual Basic (VB), C#, F# и другие. Язык использует архитектурный шаблон MVC (Model-View-Controller). В этой схеме контроллер принимает запросы пользователя и взаимодействует с моделью для обработки данных. Потом результат уже передается в представление, отображаясь в виде интерфейса веб-страницы [3].

C# — высокоуровневый язык программирования. C# популярен среди разработчиков благодаря некоторым преимуществам C++.

Также Python стал популярным языком программирования. «Он универсален и широко используется для веб-разработки. Из-за своей простоты хорошо подходит для начинающих программистов» [6].

В данной работе будет использоваться язык программирования PHP. Его преимущества состоят в том, что он хорошо интегрируется с другими средствами для разработки, такими как HTML и MySQL, а также подходит для написания бэкенда небольших сайтов. PHP крайне прост для освоения, но вместе с тем способен удовлетворить запросы профессиональных программистов.

В качестве базы данных магазина будет использоваться MySQL. MySQL – это система управления реляционными базами данных (СУБД) с открытым исходным кодом. Он используется для хранения и управления огромными объемами информации в Интернете и внутри предприятий. Он состоит из нескольких компонентов, включая сервер, который управляет базой данных, утилиты, позволяющие администраторам баз данных управлять ими, а также технологии репликации, которые позволяют распределять данные между несколькими серверами [10].

В таблице 4 представлен набор выбранных инструментов и технологий.

Таблица 4 - Набор выбранных инструментов и технологий для проектирования интернет-магазина

Функция	Название	Описание
Система управления базами данных	MySQL	Хранение, обработка и целостность данных.
Отрисовка сайта	HTML + CSS	Внешний вид в браузере
Взаимодействие с элементами веб - страниц	JavaScript	Реакция на действия пользователей со стороны интерфейса, посылка/обработка запросов на сервер.

Продолжение таблицы 4

Логика работы и взаимодействие с базой данных	PHP	Ядро программной части, расположенное на веб-сервере
Взаимодействие в DOM	Jquery	Библиотека JavaScript
Веб-сервер для локальной разработки	OpenServer	Программная оболочка серверной части сайта

2 Проектирование интернет-магазина

2.1 Планирование работ

Перед началом проектирования необходимо спланировать предстоящие работы. На рисунке 4 представлен алгоритм последовательности действий при разработке.

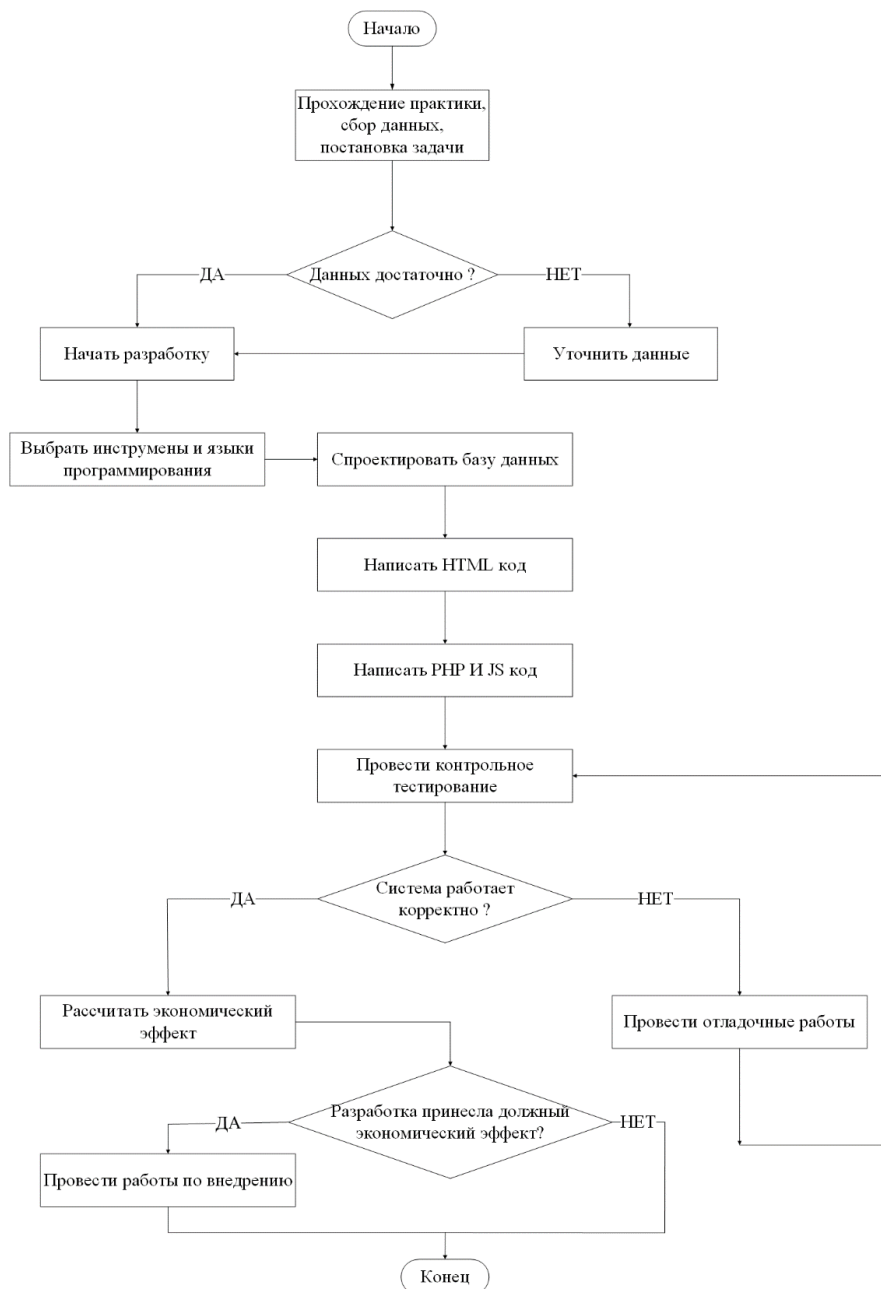


Рисунок 4 – Диаграмма последовательности действий.

Рисунок 4 демонстрирует последовательность выполнения предстоящих работ, а также последовательности работы при возможных вариантах ветвления условий.

2.2 Карта сайта

Интернет-магазин – это веб сайт, а значит он должен состоять из множества страниц с различной информацией. Поэтому функционал магазина может быть реализован посредством отдельной страницы или страниц, а другие страницы на сайте могут быть информационными. Определим состав страниц в разрабатываемом сайте:

Сайт будет состоять из 11 страниц:

index.php – главная страница сайта

about.php – страница с информацией о УМК-Сталь

market.php – страница интернет-магазина

blog.php – страница новостей

video.php – страница с видеороликами по теме

contacts.php – страница с контактами предприятия

singleproduct.php – страница карточки товара

cart.php – страница корзины покупателей

login.php – страница регистрации

signup.php – страница авторизации

admin.php – страница администрирования

На основании данных о страницах составим карту сайта (Рисунок-5)

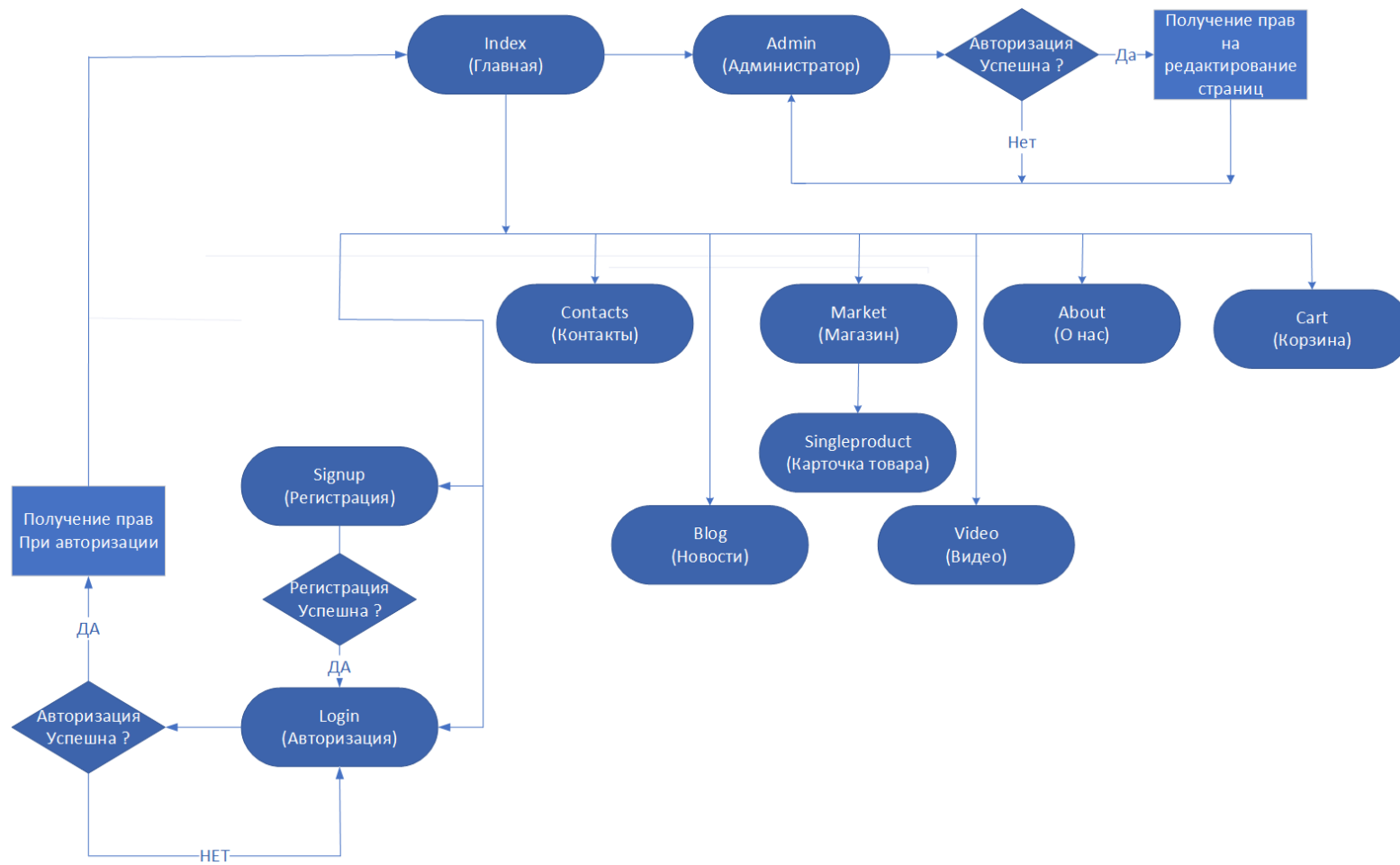


Рисунок 5 – Карта сайта

На рисунке 5 продемонстрированы в виде карты с переходами все страницы сайта интернет-магазина. Из данного рисунка виден весь список страниц сайта, а также возможные переходы от одной страницы к другой.

2.2 Диалог пользователя с системой

2.2.1 Транзитивная сеть

В процессе разработки диалога с пользователем, как правило, выделяют ряд состояний системы, в которых она остается неизменной, и пользователь может выполнять некоторые действия.

«Одним из наиболее простых способов отражения логики диалога является простая транзитивная сеть или сеть состояний и переходов. При таком подходе состояния системы отображаются в виде окружностей и соединяются дугами» [13].

Среди состояний выделяют начальное (S), при котором система находится в начале работы. Это состояние обозначается входной дугой, не имеющей источника. Начальное состояние в системе может быть одно, кроме того, в транзитивной сети выделяются финальные состояния (F), попадая в которые система завершает свою работу (таблица 4).

Сверху дуги именованы согласно входным сигналам или обратной связи, снизу – в зависимости от действия, которое система должна выполнить при переходе из одного состояния в другое по определенному входному сигналу (таблица 5).

Таблица 5 – Состояния системы

№	Название состояния
S	начальное состояние
S1	главная страница сайта
S2	страница «Видео»
S3	страница «О нас»
S4	страница «Контакты»
S5	страница «Карточка товара»
S6	страница «Корзина»
S7	окно «Оформить заказ»
S8	страница «Регистрация»
S9	страница «Вход»
S10	страница «Администратор»
S11	добавить товар
S12	удалить товар
S13	просмотр всех заказов
F	финальное состояние

Далее, в таблице 6, наглядно представлены переходы.

Таблица 6 – Переходы

№	Название перехода
1	Запуск браузера
2	Ввод адреса сайта
3	Переход на страницу «Видео»
5	Переход на страницу «О нас»
7	Переход на страницу «Контакты»
9	Переход на страницу «Карточка товара»
11	Переход на страницу «Корзина»
13	Оформление заказа
15	Переход на страницу «Регистрация»
17	Переход на страницу «Вход»
19	Ввод логина и пароля администратора
21	Добавление товара
23	Удаление товара
25	Просмотр заказов
4,6,8,10,12,14,16,18,20,22,24	Возврат назад
27	Закрытие сайта

Транзитивная сеть логики пользовательского интерфейса представлена на рисунке 6

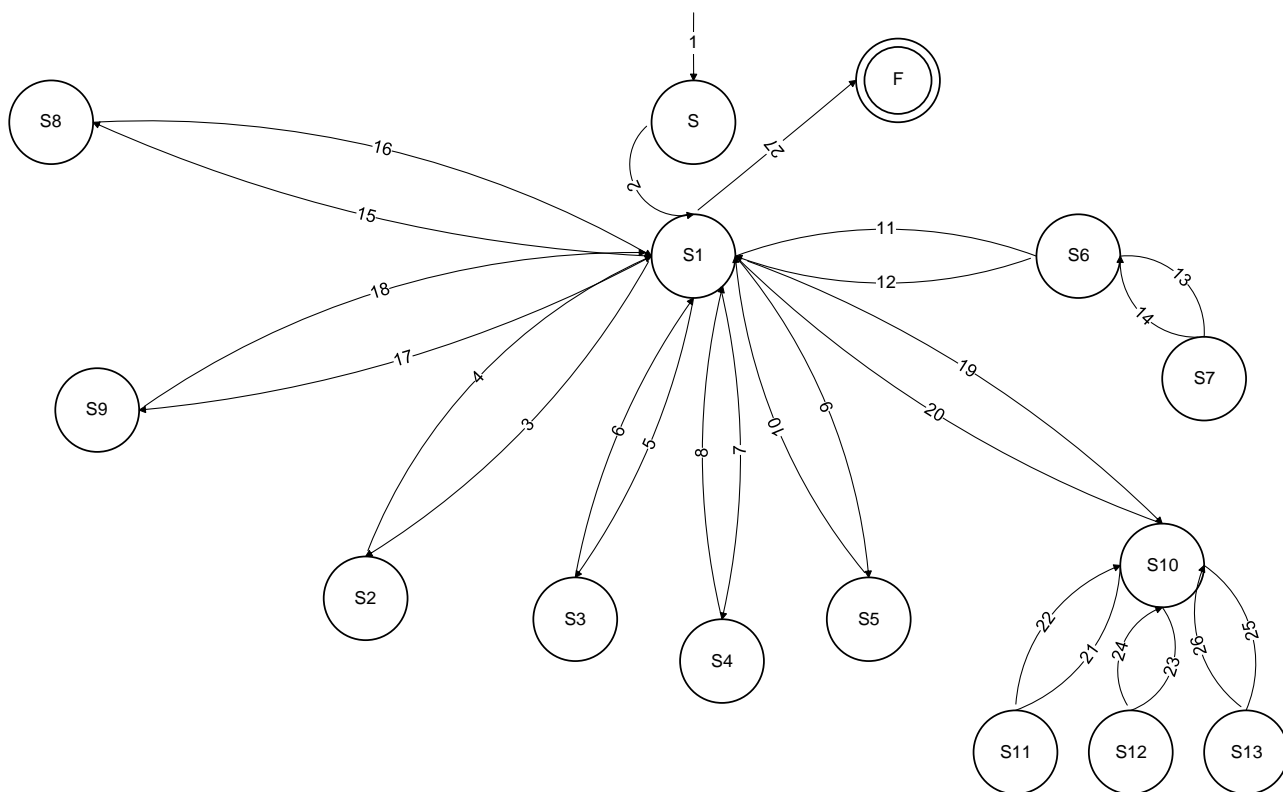


Рисунок 6 – Транзитивная сеть

Как видно из рисунка 6 – логика пользовательского интерфейса представлена в виде разветвлённой структуры.

2.2.2 Диаграмма вариантов использования

Основными функциональными требованиями, которые предъявляются к разрабатываемому интернет-магазину, являются:

- просмотр каталога товаров;
- просмотр категорий товаров;
- просмотр товаров в выбранной категории;
- добавление товара в корзину;

- изменение количества добавленного в корзину товара;
- удаление товара из корзины;
- очистка корзины;
- осуществление заказа после ввода персональных данных.

Наглядно приведенные функциональные требования изобразим в виде UML диаграммы вариантов использования (Рисунок 7).

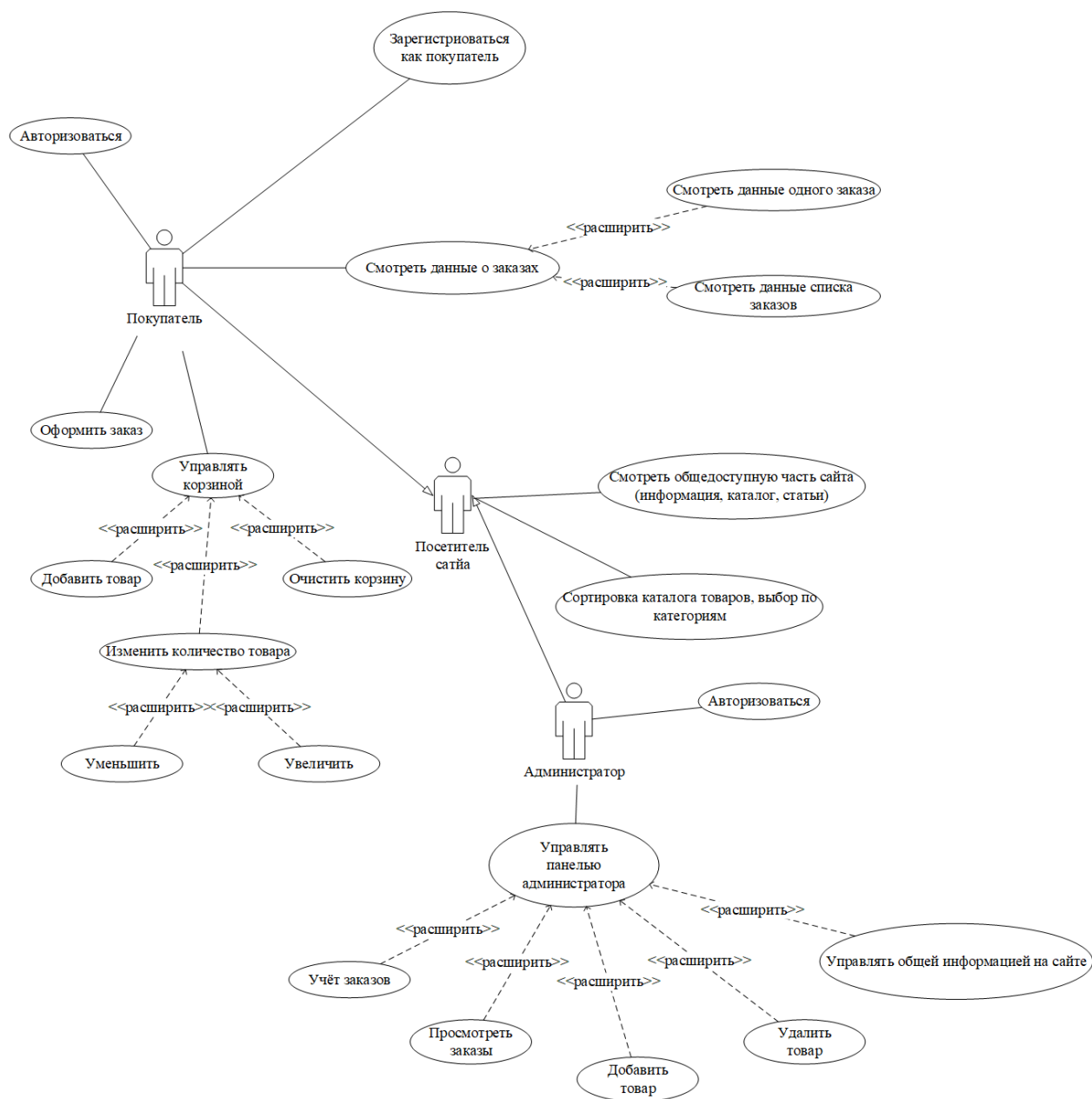


Рисунок 7 – Диаграмма вариантов использования

Диаграмма вариантов использования демонстрирует функциональные возможности проектируемого сайта с точки зрения распределения ролей и используется в качестве одной из основных диаграмм при проектировании систем [20].

2.3 Описание бизнес-логики интернет-магазина

Как описывалось ранее – применение определённых паттернов проектирования при разработке магазина является желательной практикой. «Стандартом проектирования приложений на PHP является паттерн MVC» [4], который будет применён в данной работе.

Схема взаимодействия MVC представлена на рисунке 8

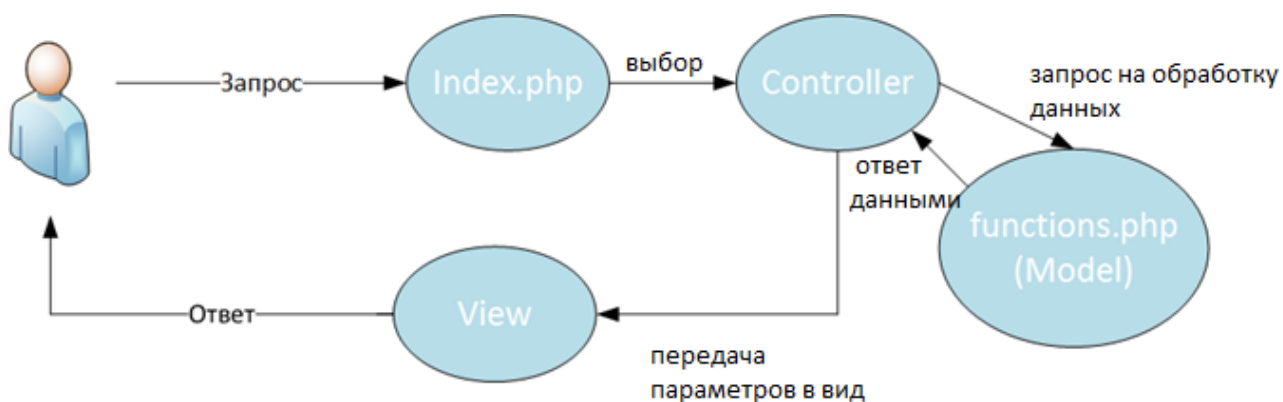


Рисунок 8 – Применение паттерна MVC

Принцип клиент-серверного взаимодействия интернет-магазина изображен на рисунке 9.

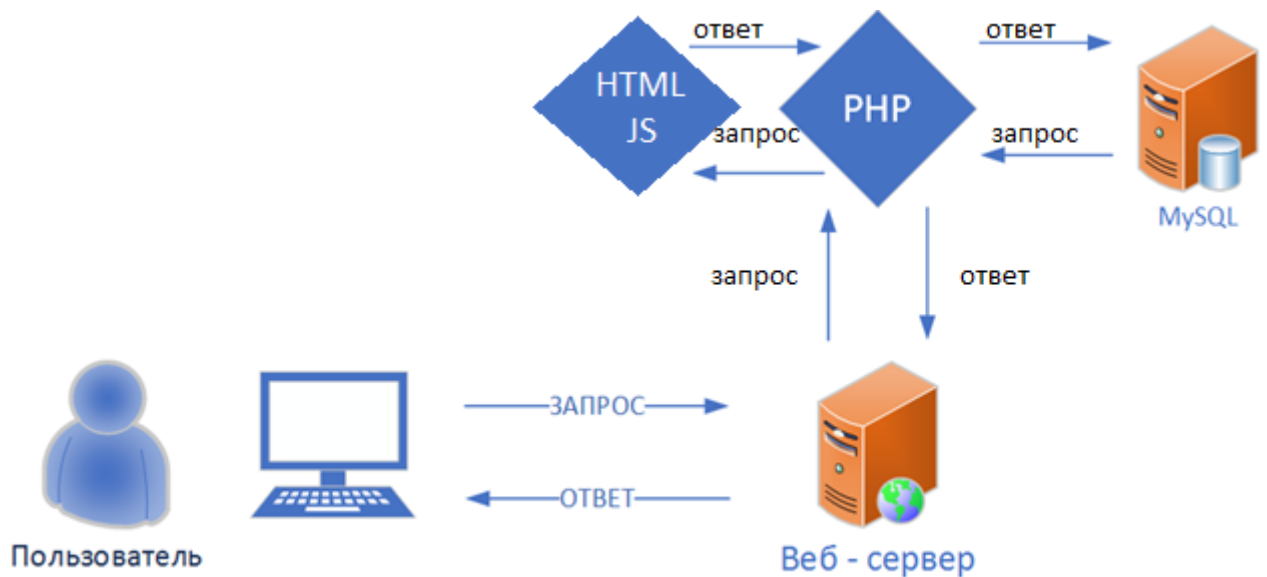


Рисунок 9 – Клиент-серверное взаимодействие

Файловая архитектура сайта представлена на рисунке 10.

.vscode	16.06.2022 16:23	Папка с файлами	
ajax	16.06.2022 16:23	Папка с файлами	
assets	16.06.2022 16:24	Папка с файлами	
templates	16.06.2022 16:23	Папка с файлами	
about	04.04.2022 19:00	Исходный файл ...	1 КБ
admin	20.03.2022 10:52	Исходный файл ...	2 КБ
blog	26.05.2022 17:04	Исходный файл ...	1 КБ
cart	16.03.2022 22:59	Исходный файл ...	2 КБ
contacts	16.03.2022 23:13	Исходный файл ...	1 КБ
functions	16.06.2022 17:48	Исходный файл ...	4 КБ
index	16.06.2022 17:36	Исходный файл ...	2 КБ
login	17.03.2022 9:34	Исходный файл ...	2 КБ
logout	02.02.2022 14:07	Исходный файл ...	1 КБ
market	04.04.2022 14:51	Исходный файл ...	2 КБ
order	20.03.2022 10:49	Исходный файл ...	2 КБ
signup	20.03.2022 10:44	Исходный файл ...	2 КБ
singleproduct	17.03.2022 9:50	Исходный файл ...	1 КБ
video	04.04.2022 18:46	Исходный файл ...	1 КБ

Рисунок 10 – Файловая архитектура сайта

Все виды (views) страниц сайта будут размещены в отдельной папке «templates», контроллеры будут иметь одноимённые названия со страницами сайта и выполнять все действия бизнес-логики, моделью станет файл

«functions.php». Набор функций модели будет содержать в себе функции обращения к базе данных, работы с шаблонами, работы с файлами cookies и прочие вспомогательные функции.

Таким образом – контроллер отдельной страницы будет подключать модель, затем формировать переменные и предавать их в вид, формировать этот вид и выводить полученный результат пользователю.

Код представлений страниц содержится в папке «templates» проекта. Каждый файл представлений страниц подключается соответствующим, как правило, одноимённым, контроллером. В общем виде – контроллер страницы собирает данные посредством модели, затем формирует отдельные фрагменты кода HTML из шаблонов «header.php», «main.php», «footer.php» и вставляет их в шаблон «layout.php» [24].

На рисунке 11 расположено изображение макета страниц сайта из папки templates, каждый элемент макета соответствует файлу php :

- Шаблон (layout.php)
- Верхняя часть (header.php)
- Нижняя часть (footer.php)
- Контент (main.php)

«Такой подход имеет ряд преимуществ и позволит использовать готовые компоненты кода в приложении. Достаточно будет прописать статическую часть кода в данных компонентах и подключать их на различных страницах. Это позволит избежать дублирование кода в разных местах» [1]. Каждой из страниц сайта будет применён макет на рисунке 12

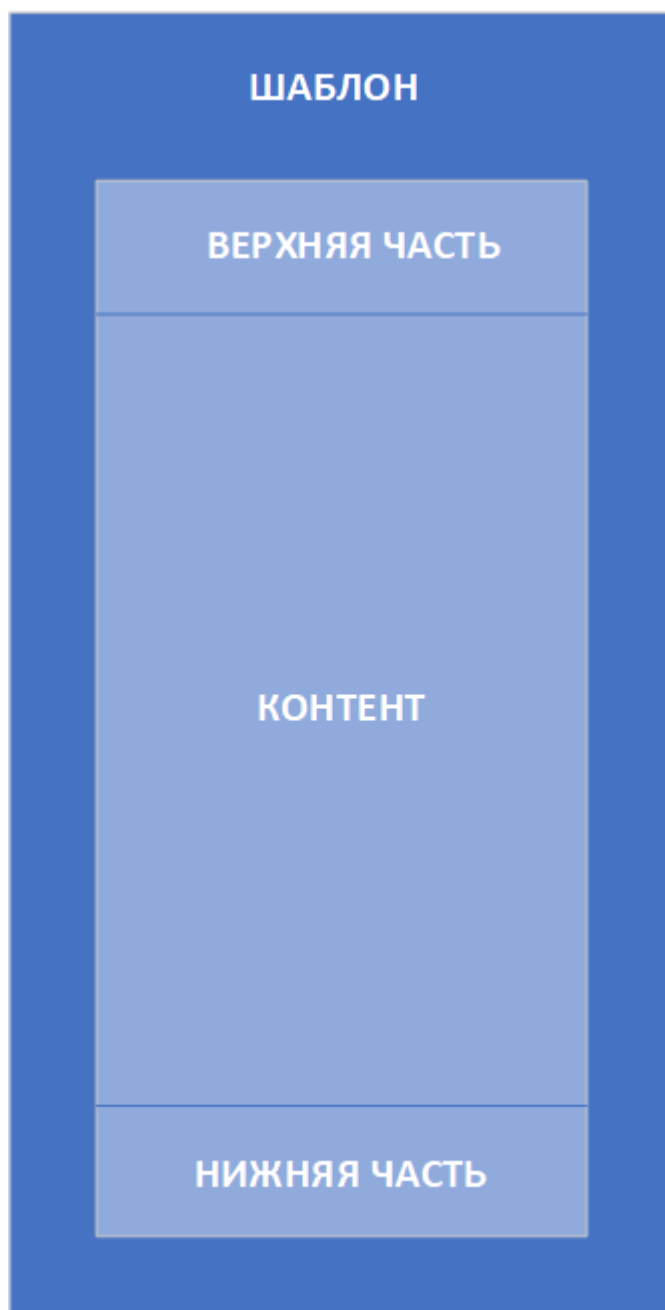


Рисунок 12 – Макет страницы

Как видно из рисунка 12 – при проектировании интерфейса использован компонентный подход.

2.4 Логическое проектирование базы данных

В базе данных будет храниться информация, которая отображается на сайте. К такой информации относится информация о товарах, заказах и учетных данных пользователей, видеороликах, новостях, администраторах. В базе данных будут содержаться таблицы с соответствующими данными. База данных будет создана с помощью выполнения SQL – кода в среде MySQL.

Необходимо выделить сущности и атрибуты сущностей, которые впоследствии будут преобразованы в таблицы. Исходя из требуемого функционала и методах его реализации можно выделить сущности. (Таблица 7)

Таблица 7 – Сущности базы данных

Сущности	Назначение
Products (товары)	Основные данные о товарах, такие как название, описание, категории, применение скидков, статус хита продаж, ссылка на изображение, назначение.
Categories (категории)	Наименование категорий товаров.
Cookies (куки)	Данные о составе заказа, который хранится в JSON формате.
Admins (администраторы)	Данные об администраторах, логин, пароль
Customers (Клиенты)	Данные о клиентах, телефон, email, пароль, флаг регистрации, текущий токен доступа к файлу куки.

Продолжение таблицы 7

Orders (Заказы)	Данные о заказах, покупатель, сумма, метод оплаты, дата создания, статус заказа, комментарий к заказу.
Orders-customers (Состав заказов)	Данные о составе заказа, покупатель, товар, количество.
Video (видеоролики)	Данные о видеороликах на сайте
Articles (новости)	Данные о новостях на сайте

Сущности будут объединены между собой связями, один-ко-многим и многие-ко-многим, рассмотрим их.

- Каждая категория товара может принадлежать многим товарам – один-ко-многим.
- Каждый клиент может осуществить множество заказов – один-ко-многим
- Каждый заказ может содержать множество товаров – один-ко-многим
- Каждому клиенту может быть присвоено множество файлов куки – один-ко-многим
- Очевидно, что понадобится дополнительная связь – многие-ко-многим для реализации возможности множеству заказов иметь множество товаров.

Построим инфологическую схему базы данных (рисунок 13)

На основе полученных связей необходимо составить ER – диаграмму. Отдельные сущности «администратор», «видео», «новости» не нуждаются в декомпозиции.

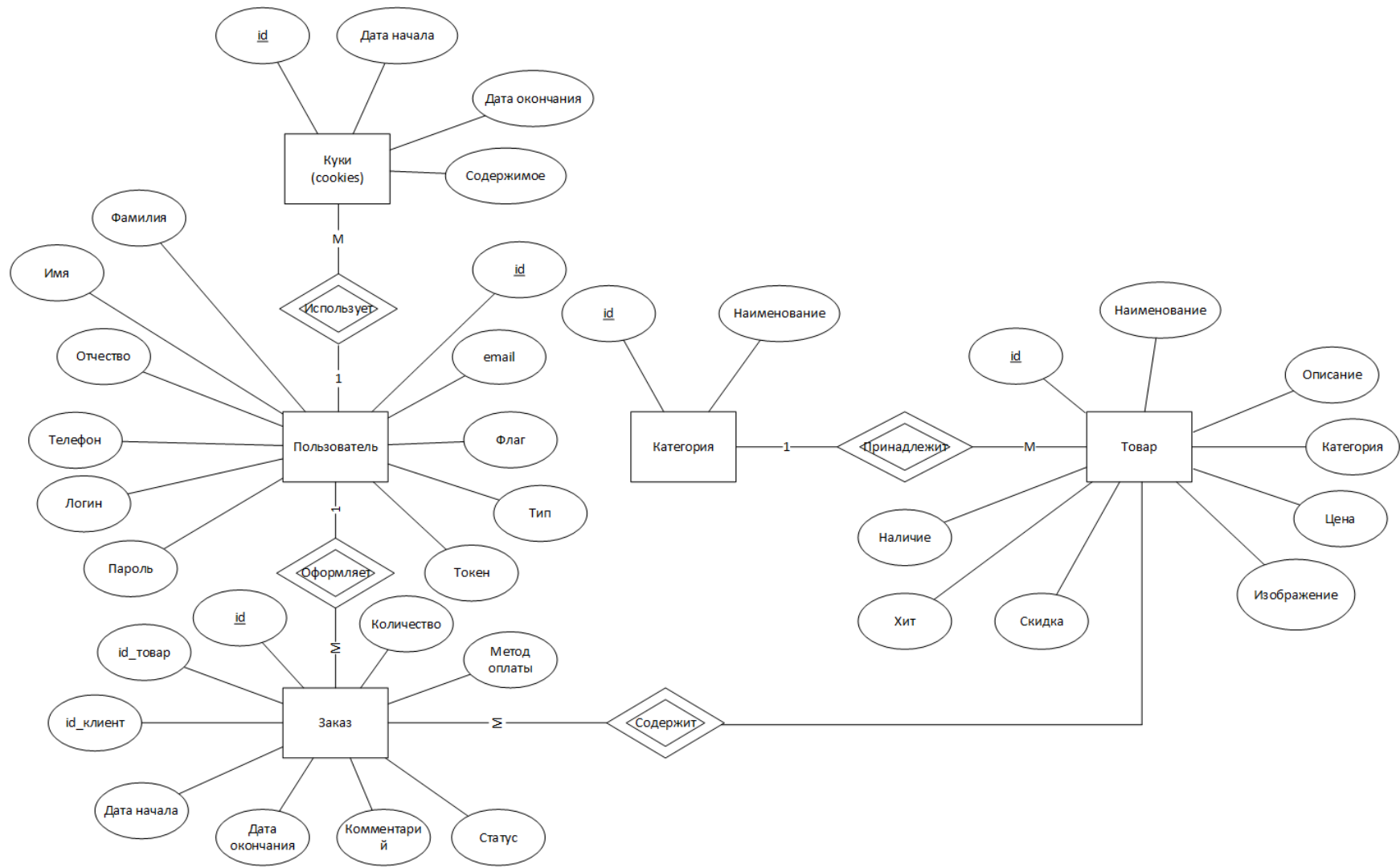


Рисунок 13 – Инфологическая схема базы данных

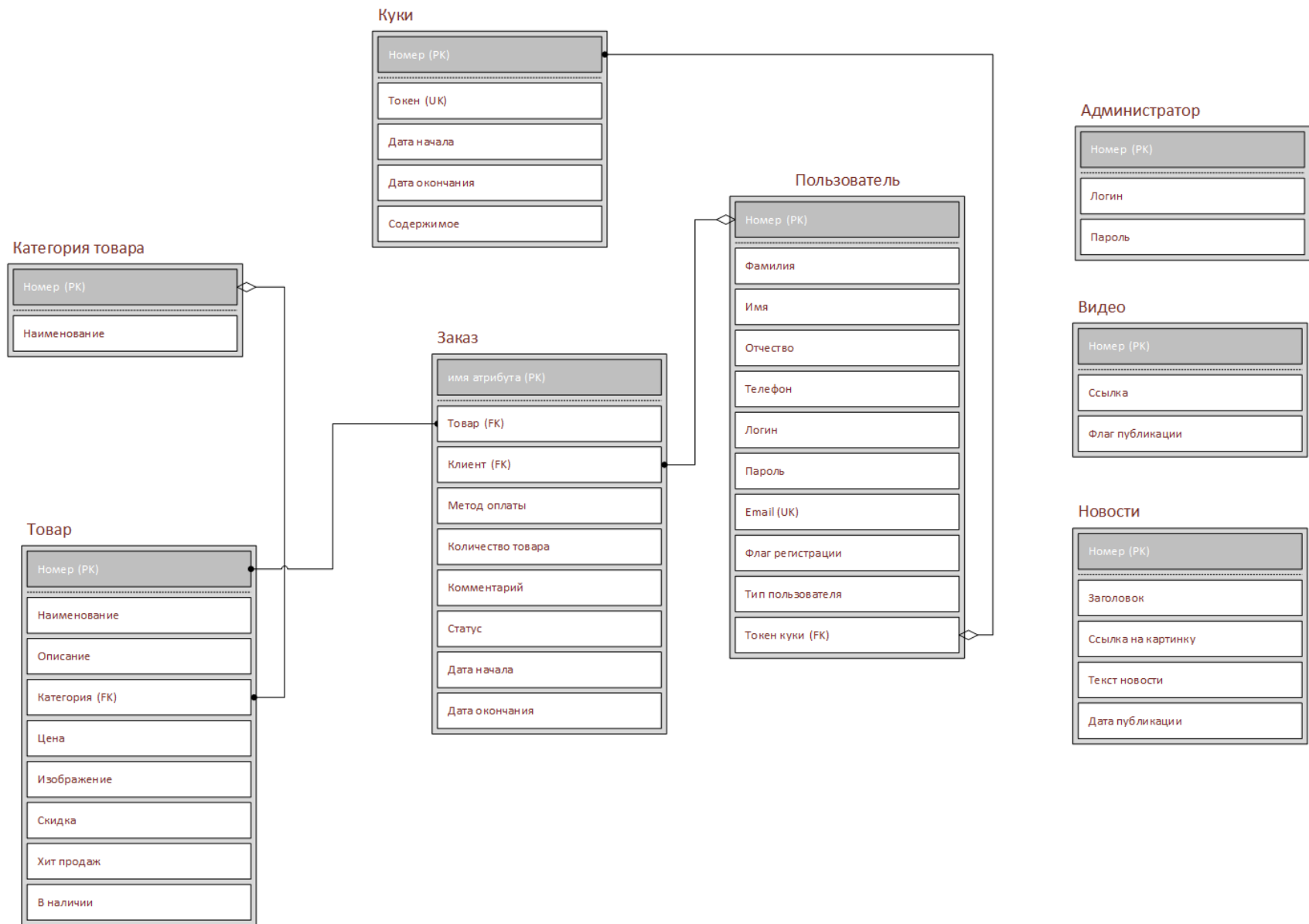


Рисунок 14 – ER – диаграмма базы данных

2.5 Проектирование базы данных на физическом уровне

«У каждой сущности должны быть атрибуты – физические элементы, характеризующие каждую сущность, каждая сущность должна быть преобразована в таблицу базы данных» [17].

Для данных, связанных с временем используем тип «Datetime», для числовых данных – тип «INT» и «TINYINT», для вариантов выбора – «ENUM», для текстовых данных – «VARCHAR». Исходя из этого необходимо составить набор отдельных сущностей базы данных в виде таблиц.

Рассмотрим вопрос безопасности базы данных. Безопасность данных – это состояние защищенности, при котором обеспечиваются конфиденциальность, доступность и целостность данных [15]

К параметрам безопасности можно отнести следующее:

- Конфиденциальность
- Целостность
- Доступность

В качестве обеспечения условий безопасности будут применены следующие меры:

- Хэширование данных пользователей(пароли)
- Защита от SQL – инъекций (подготовленные выражения PHP)
- Аутентификация пользователей
- Резервное копирование и восстановление данных

«В качестве ограничений целостности базы данных будут использованы следующие типы ключей – первичный, внешний, уникальный. Также каждый из атрибутов будет иметь характерный тип данных, ограниченных по длине» [7].

Далее в таблицах с 8 по 15 представлены: категории, куки, клиенты, заказы, товары, заказы-клиенты, видео и новости соответственно .

Таблица 8 – Категории

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
title	VARCHAR(100)	Символьный	-

Таблица 9 – Куки

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
token	VARCHAR(100)	Символьный	Уникальный
open	TIMESTAMP	Метка времени	-
expired	TIMESTAMP	Метка времени	-
content	JSON	JSON	-

Таблица 10 – Клиенты

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
name	VARCHAR(50)	Символьный	-
email	VARCHAR(100)	Символьный	Уникальный
phone	VARCHAR(15)	Символьный	-
password	VARCHAR(100)	Символьный	-

Продолжение таблицы 10

registration	TINYINT	Целочисленный	-
cookie	VARCHAR(100)	Символьный	Внешний

Таблица 11 – Заказы

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
customer	INT(10)	Целочисленный	Внешний
pay_method	ENUM	Перечисление	-
amount	DECIMAL(10,2)	Число с плавающей точкой	-
comment	VARCHAR(200)	Символьный	-
status	ENUM	Перечисление	-
date_in	DATETIME	Время	-

Таблица 12 – Товары

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
title	VARCHAR(100)	Символьный	-
description	VARCHAR(100)	Символьный	-
category	INT(10)	Целочисленный	Внешний

Продолжение таблицы 12

price	DECIMAL(10,2)	Число с плавающей точкой	-
img	VARCHAR(100)	Символьный	-
sale	ENUM	Перечисление	-
popular	ENUM	Перечисление	-
exist	ENUM	Перечисление	-

Таблица 13 – Заказы-клиенты

Наименование	Формат	Тип	Ключ
id_order	INT(10)	Целочисленный	Первичный, Внешний
id_customer	INT(10)	Целочисленный	Первичный, Внешний
id_product	INT(10)	Целочисленный	Первичный, Внешний
count	SMALLINT(5)	Целочисленный	-

Таблица 14 – Видео

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
link	VARCHAR(200)	Символьный	-
public	ENUM	Перечисление	-

Таблица 15 – Новости

Наименование	Формат	Тип	Ключ
id	INT(10)	Целочисленный	Первичный
text	TEXT	Текст	-
dt	TIMESTAMP	Целочисленный	-
img	VARCHAR(150)	Символьный	-
title	VARCHAR(150)	Символьный	-

Данные таблицы соответствуют требованиям нормализации реляционных БД. «Таблицы находятся в 1НФ, т.к. все их поля являются простыми (неделимыми), нет повторяющихся полей и каждый столбец таблиц хранит одно-единственное значение и не является ни списком, ни множеством значений. Таблицы находятся во 2НФ, т.к. они находятся в 1НФ, имеют простые первичные ключи и каждое не ключевое поле функционально зависит от первичного ключа. Таблицы находятся в 3НФ, т.к. они находятся во 2НФ и все не ключевые поля являются взаимно-независимыми» [12].

На основе собранных данных необходимо реализовать физическую базу данных.

Код создания базы данных представлен в приложении А.

В результате выполнения SQL – кода была получена база данных, схема которой показана на рисунке 15

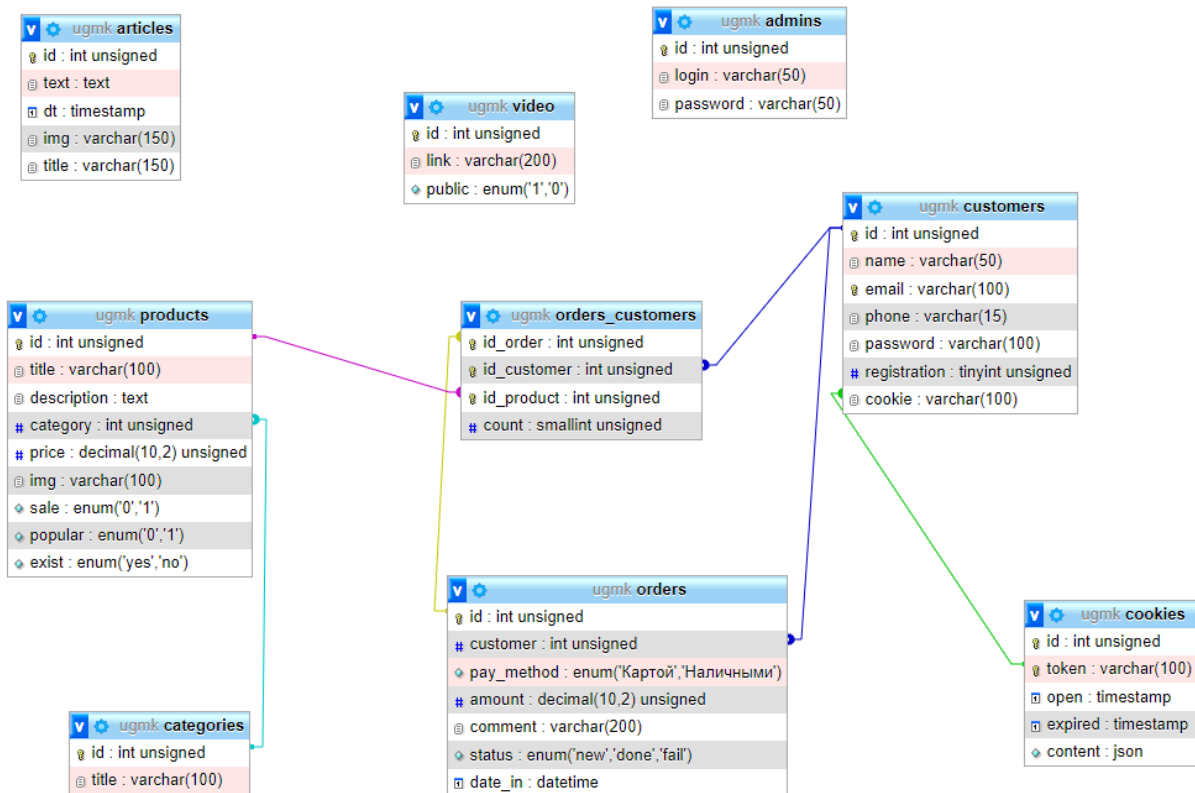


Рисунок 15 – Физическая схема базы данных

Рисунок 15 представляет собой физическую схему базы данных и демонстрирует состав таблиц и атрибутов.

3 Разработка интернет-магазина

3.1 Функции PHP

Рассмотрим все функции, которые будут использоваться в данном проекте. Функции расположены в файле `functions.php` Данные функции написаны самостоятельно исходя из бизнес-логики работы проектируемого сайта, описанной выше.

«Работа с базой данных основывается на встроенном в PHP классе PDO, который предоставляет все необходимые для этого методы и свойства. Подключение к базе данных инициализируется созданием экземпляра класса PDO с передачей в конструктор объекта параметров подключения, таких как наименование драйвера базы данных, URL адрес хоста подключения, имя базы данных, логин и пароль. На основе полученного экземпляра, выполняя различные методы класса PDO происходит работа с базой данных. Одним из основных методов класса PDO является метод `query`, аргументом которого может быть текст в виде любого SQL – запроса к базе данных, результатом исполнения данного метода является экземпляр класса `PDOStatement`, методы которого позволяют непосредственно извлекать полученные данные из базы» [9]. Примерами методов `PDOStatement` являются `fetch()` и `fetchAll()`, которые извлекают очередную строку из результирующего набора либо все строки сразу. Строки могут быть представлены в виде объекта или в виде массива, в зависимости от настроек PDO. Реализованные в данной работе функции `dbGo()`, `sql()`, `sqlInsert()` представляют собой обёртки, реализующие поведение вышеописанных классов и методов.

`dbGo()`- иницирует подключение к базе данных и возвращает экземпляр класса PDO в случае успешного подключения, а в случае сбоя при

подключении в поток вывода выдаётся информация об ошибке и скрипт завершается. (листинг 1)

```
function dbGo()
{
    try {
        $db = new PDO('mysql:host=' . DB_HOST . ';dbname=' . DB_NAME, DB_USER,
DB_PASS);

        $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

        $db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);
    } catch (PDOException $e) {

        print "Ошибка подключения: " . $e->getMessage() . "<br/>";

        die();

    }

    return $db;
}
```

Листинг 1 – Функция dbGo()

sql(\$q) - запускает внутри себя dbGo(), а затем исполняет переданный в неё в качестве аргумента SQL запрос (\$q), возвращая в виде результата экземпляр PDOStatement, к которому в зависимости от задачи можно применить методы fetch() или fetchAll(). (листинг 2).

```
function sql($q): PDOStatement
{
    try {

        $n = dbGo()->query($q);
```

```

} catch (PDOException $e) {

    print "Ошибка : " . $e->getMessage() . "<br/>";

    die();

}

return $n;

}

```

Листинг 2 – Функция sql()

getCategories() – извлекает данные таблицы категорий из базы и возвращает их в виде массива; (см. листинг 3)

```

function getCategories()

{

    $q = "SELECT * FROM categories";

    return sql($q) -> fetchAll();

}

```

Листинг 3– Функция getCategories()

getProducts(\$id) - извлекает данные таблицы товаров из базы и возвращает их в виде массива, если при вызове данной функции аргумент ей не передаётся, то извлекает данные о всех товарах, а если аргумент передан (номер категории), то извлекает данные о товарах соответствующей категории. Таким образом – аргумент \$id в данной функции является необязательным. (листинг 4)

```
function getProducts($id = 0)
{
    $q = "SELECT * FROM products";
    if($id) {
        $q .= " WHERE category='$id'";return sql($q) -> fetchAll()
    }
}
```

Листинг 4 – Функция getProducts()

delProduct(\$id) – удаляет запись об отдельном товаре из таблицы с товарами, в качестве аргумента принимает идентификатор товара. (листинг 5)

```
function delProduct($id){
    $q = "DELETE FROM products WHERE id='$id'";
    return sql($q);
}
```

Листинг 5 – Функция delProduct()

«Так как приложения на PHP после выполнения различных действий с данными пользователя, базы данных или внешних сервисов в конечном счёте возвращают пользователю HTML – страницу то встаёт вопрос о компоновке HTML – кода и работе с этим кодом самого PHP. PHP необходимо сгенерировать разметку, предварительно передав туда какие-то параметры, полученные в результате работы скрипта» [9].

Причём сами параметры могут представлять собой уже ранее сгенерированные фрагменты кода, что даёт очень большую гибкость в управлении внешним видом получаемой страницы. Именно на этом принципе строится шаблонизация – PHP скрипт собирает мелкие куски

разметки и вставляет их в более крупные, затем эти более крупные вставляет в страницу и формирует окончательный вид, отправляя её в браузер. В данной работе также используются принципы шаблонизации, идеи которых реализует написанная функция `template($view, $arr, $path)` (см. листинг 6)

`template($view, $arr, $path)` принимает на вход три параметра: `$view` – название подключаемого вида, `$arr` – массив с данными, `$path` – путь к файлу директории. Применяя встроенную в PHP функцию `extract()` происходит распаковка массива `$arr`, в результате чего образуются переменные, далее включается буферизация вывода с помощью встроенной в PHP функции `ob_start()`, на основе параметров `$view` и `$path` формируется путь к подключаемому шаблону, в результате чего шаблон подключается с помощью конструкции `include()`, PHP интерпретатор считывает данный шаблон, накапливая в буфере текст и подставляя в него переменные. По окончании считывания шаблона происходит возврат из функции `template()`, результатом работы которой является сброс в переменную накопленного и обработанного HTML - текста.

```
function template(string $view, array $arr = [], string $path = './templates/'): string
{
    extract($arr);
    ob_start();
    include($path . $view . '.php');
    return ob_get_clean();
}
```

Листинг 6 – Функция `template()`

`getHTMLStyleSheets($paths)` – формирует участок HTML кода, отвечающий за подключение внешних файлов таблиц стилей, в качестве

аргумента передаётся массив с названиями подключаемых файлов без расширения “.css”(листинг 7)

```
function getHTMLStyleSheets(array $paths):string
{
    $HTML = "";
    foreach($paths as $path){
        $HTML .= "\t";
        $HTML .= "<link rel=\"stylesheet\" href=\"assets/css/$path.css\">";
        $HTML .= PHP_EOL;
    }
    return $HTML;
}
```

Листинг 7 – Функция getHTMLStyleSheets ()

getHTMLStyleScripts(\$paths) – формирует участок HTML кода, отвечающий за подключение внешних файлов скриптов, в качестве аргумента передаётся массив с названиями подключаемых файлов без расширения “.js”(листинг 8)

```
function getHTMLScripts(array $paths):string
{
    $HTML = "";
    foreach($paths as $path){
        $HTML .= "\t";
```

```

$HTML .= "<script src=\"assets/js/$path.js\"></script>";

$HTML .= PHP_EOL;

}

return $HTML;

}

```

Листинг 8 – Функция getHTMLStyleScripts ()

initCookie() – Производит установку файла cookie в браузер пользователя, затем производит запись с данными этого файла в таблицу “cookies” базы данных. В качестве данных генерируется псевдослучайный набор символов с помощью встроенной в PHP функции MD5()(листинг 9)

```

function initCookie(){

    $val = md5(time());

    setcookie('tok', $val);

    sqlInsert('cookies', ['token' => $val, 'content' => '{}']);

}

```

Листинг 9 – Функция initCookie ()

3.2 Функции JavaScript

При разработке сайта была использована библиотека JQuery. Это позволило решить ряд задач:

- формировать на лету и обновлять участки HTML кода;
- валидировать формы перед отправкой на сервер;

- обращаться асинхронно к серверу для получения данных;

В листингах 10 и 11 продемонстрировано получение HTML – кода на основе AJAX запросов.

```
$.get("/ajax/checkcookie.php", function (cart) {  
    $('#cart').html('(' + cart + ');'  
}  
)
```

Листинг 10 – Обновление HTML кода с данными корзины

```
$('.menu-list a').on('click', function(){  
    let id = $(this).attr('data-cat');  
    let url = "/ajax/products.php?id=" + id;  
    $('#products').load(url);  
})
```

Листинг 11 – Обновление HTML кода с данными каталога серверов

В листинге 12 приведён код, добавляющий данные в таблицу «cookies» базы данных при клике на кнопку на карточке товара.

```
$(document).on('click', '.product', function(e){  
    e.preventDefault();  
    let id = $(this).attr('data-id');  
    let url = "/ajax/addcookie.php?id=" + id;  
    $.get(url,
```



```

function(cart){
    $('#cart').html('(' + cart + ')')
}
);
})

```

Листинг 12 – Добавление записи в таблицу «cookies»

Следующий листинг кода демонстрирует обновление данных о расчёте суммарной стоимости всех заказов в корзине. (листинг 13)

```

$(document).on('click', '.ajax', function () {
    let id = $(this).attr('data-id');
    let url = "/ajax/delcookie.php?id=" + id;
    let price = $(this).parent().parent().find('.price').text();
    let count = $(this).parent().parent().find('.count').text();
    let amount = $('.amount').text();
    let newAmount = amount - price * count;
    $(this).parent().parent().fadeOut(500, function () {
        $.get(url,
            function (cart) {
                $('#cart').html('(' + cart + ')')
                $('.amount').html(newAmount);
            }
        );
        $(this).remove();
    })
}
);

```

}}

Листинг 13– Расчёт стоимости корзины покупателей

Демонстрация работы интернет-магазина

Главная страница сайта имеет следующий вид (рисунок 16).

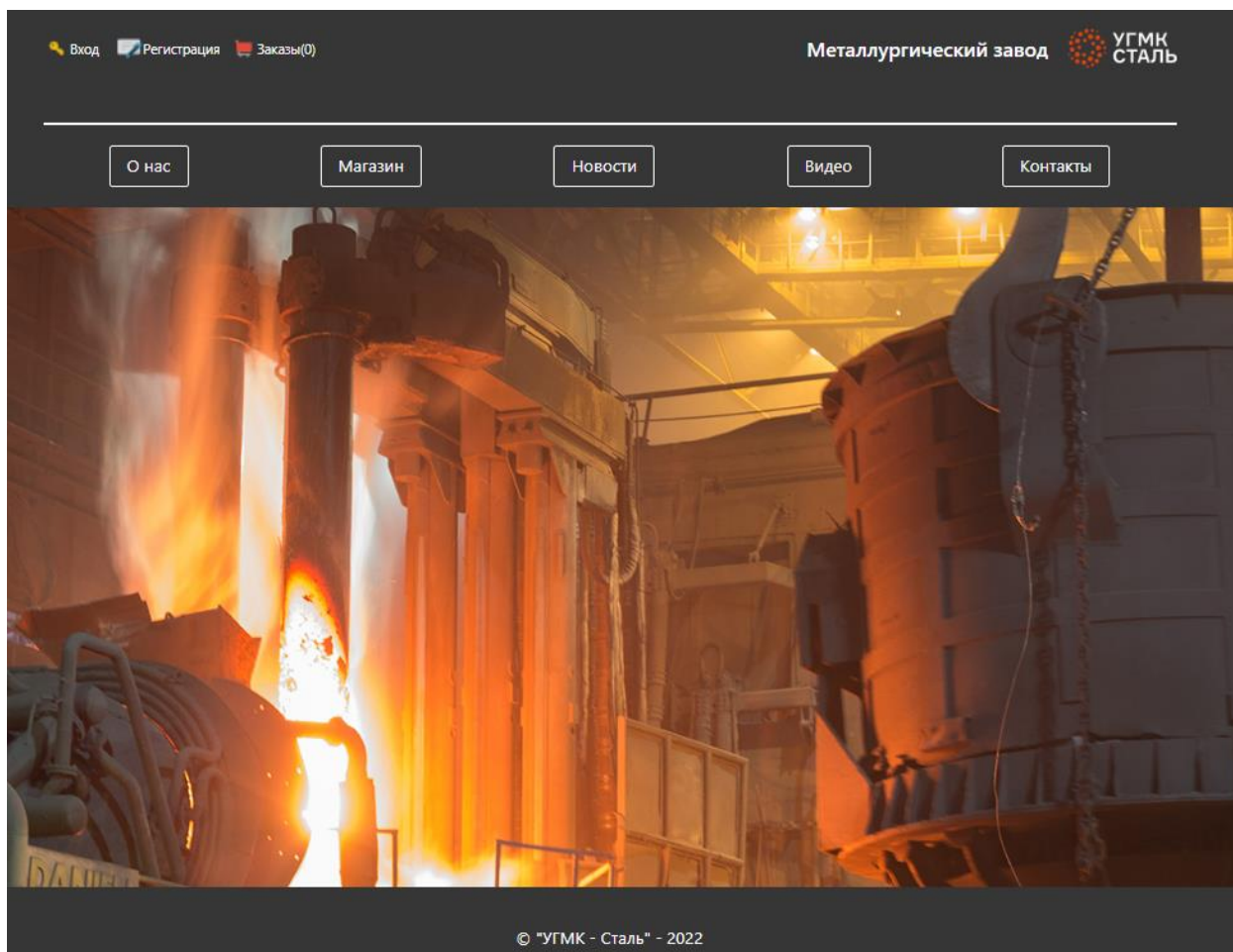


Рисунок 16 – Главная

Данная страница сайта является стартовой для сайта и первой, которую пользователь видит при открытии сайта. Функционально данная страница является витриной всего сайта, представляет навигацию по разделам и

ссылкам. Навигация выполнена в виде отдельных повторяющихся модулей «header» и «footer». В верхней части также используется логотип и ссылки для авторизации/регистрации [25].

Страница магазин имеет следующий вид (рисунок 17). Данные о товарах, такие как фото, цена, название запрашиваются из базы данных и выводятся на этой странице.

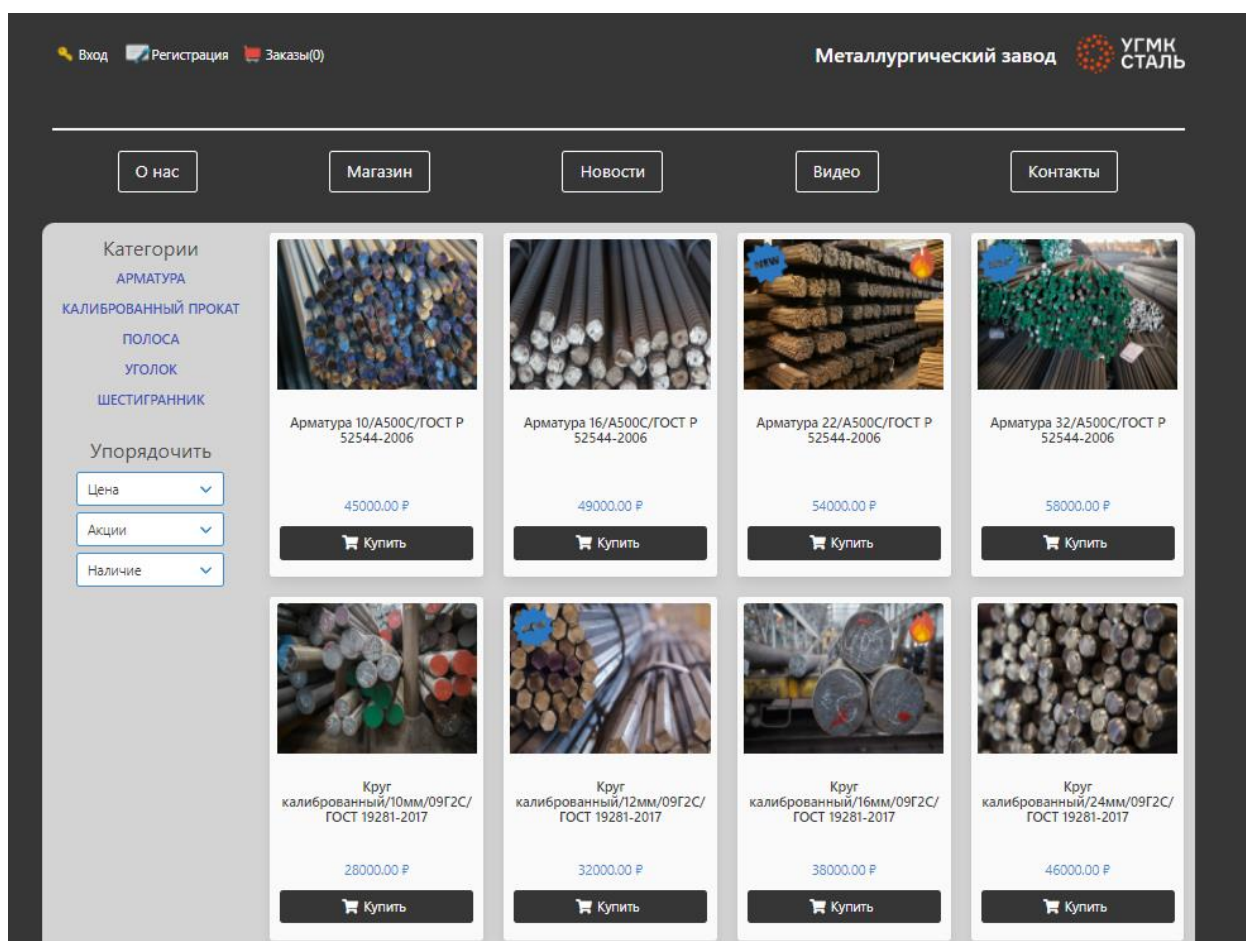


Рисунок 17 – Магазин

Страница имеет навигационную панель вверху экрана, панель ссылок ниже. Слева расположена навигация по категориям, нажав на одну из ссылок на панели слева будут подгружаться методом AJAX карточки товаров соответствующих категорий. Также слева расположена панель фильтрации по параметрам. Выбрав один или несколько параметров методом AJAX отправляется запрос на сортировку товаров с учётом переданных параметров.

Возвращается с сервера новое HTML представление каталога товаров. Каждый товар может иметь отметку о новизне или о популярности этого товара. Данные отметки хранятся в базе данных и могут быть заданы администратором [18].

Каждый из товаров имеет отдельную динамически формируемую страницу. При нажатии на любой из товаров откроется подробная информация про товар (рисунок 18). При запросе каждой подобной страницы по ссылке передаётся id товара в виде параметра GET, после чего по данному id происходит запрос к базе данных на получение информации об этом товаре.

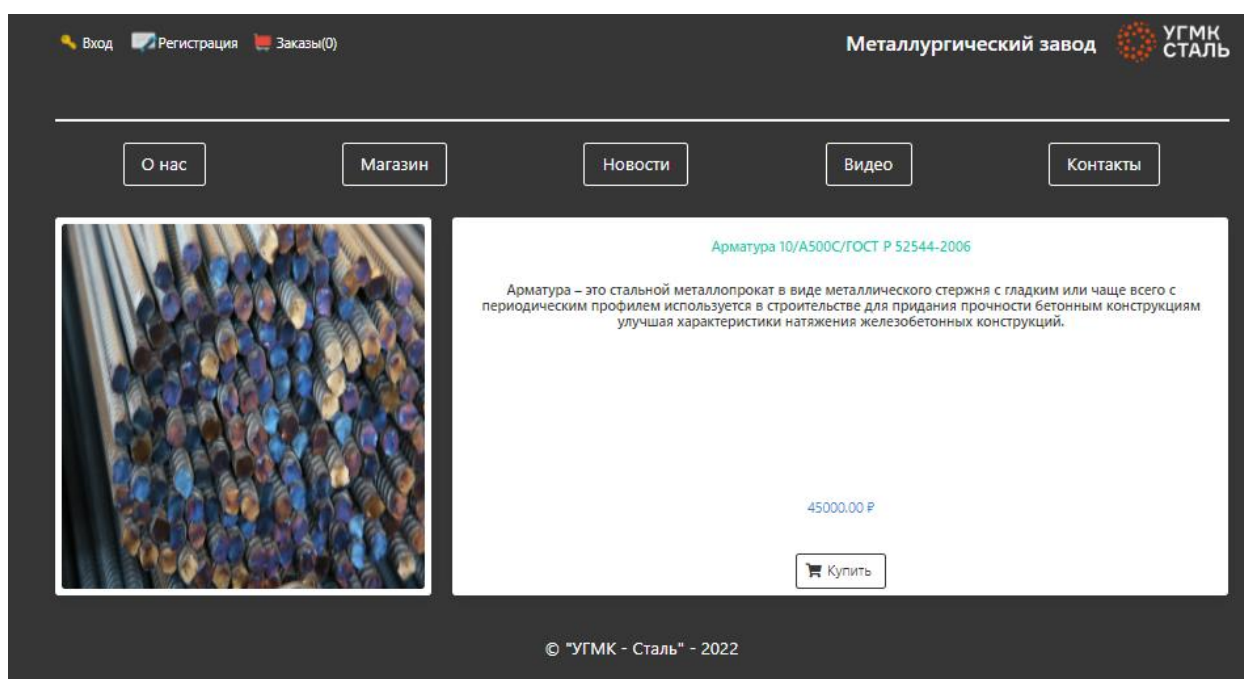


Рисунок 18 – Карточка товара

В карточке товара отображаются фотографии товара, его название, описание и стоимость.

Пользователь может формировать корзину покупок находясь как в авторизованном, так и в неавторизованном режиме. (см. рисунок 19).

Это значит, что оформить заказ можно, не проходя регистрацию

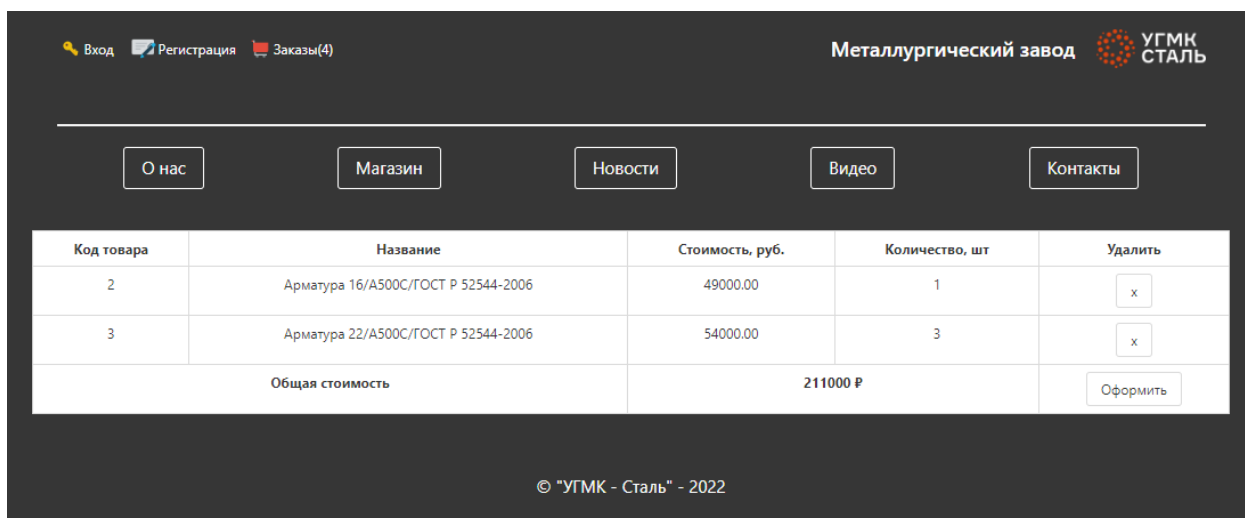


Рисунок 19 – Корзина заказа

В корзине можно увидеть все товары, которые были выбраны пользователем и кнопку оформления заказа. Состав корзины можно редактировать, удалив необходимые позиции заказа. Управление корзиной осуществляется методами JavaScript и AJAX запросами к базе данных.

При нажатии на кнопку «Оформить» происходит показ всплывающего окна с формой оформления заказа, при этом если пользователь авторизован, то данные о пользователе автоматически подставляются в соответствующие поля, а если не авторизован, то пользователю необходимо будет заполнить поля самостоятельно. (рисунок 20).

Оформление заказа

Ваше имя
Иван

Номер телефона
+79879888585

Email
ivan@mail.ru

Комментарий к заказу
нет

Форма оплаты
 Наличными Картой

Выбрано товаров: 4
На сумму: 211000 ₽

Оформить

Рисунок 20 – Форма оформления заказа

После нажатия на кнопку «Оформить» происходит отправка формы на сервер, после чего данные заказа записываются в базу данных, в результате чего создаётся новый оформленный заказ [21].

Также пользователь может пройти регистрацию, чтобы получить личный кабинет. (рисунок 21). При регистрации и авторизации данные заполненных форм передаются на сервер, в случае с регистрацией – происходит запись новой строки с данными пользователя в базу данных, а в случае авторизации – происходит сравнение переданных данных с записью в

базе и в случае их совпадения происходит запись данных о пользователе в сессию PHP.

The screenshot shows a dark-themed website header with navigation links: 'Вход', 'Регистрация', and 'Заказы(4)'. The main header contains the text 'Металлургический завод' and the logo 'УГМК СТАЛЬ'. Below the header is a horizontal menu with buttons for 'О нас', 'Магазин', 'Новости', 'Видео', and 'Контакты'. The central focus is a 'Форма регистрации' (Registration Form) with three input fields: 'Имя' (Name), 'Email', and 'Пароль' (Password). A yellow 'Регистрация' (Register) button is positioned below the fields. At the bottom of the page, the copyright notice '© "УГМК - Сталь" - 2022' is visible.

Рисунок 21 – Форма регистрации

После регистрации пользователь сможет попасть в личный кабинет (рисунок 22).

The screenshot shows the same website header and navigation menu as in Figure 21. The central focus is a 'Форма авторизации' (Authorization Form) with two input fields: 'Email' and 'Пароль' (Password). A yellow 'Вход' (Login) button is positioned below the fields. At the bottom of the page, the copyright notice '© "УГМК - Сталь" - 2022' is visible.

Рисунок 22 – Форма авторизации

Верхняя часть навигационной панели авторизованного пользователя содержит данные с его именем и email(рисунок 23). Таким образом, внешний вид верхней части меняется по сравнению со стандартным видом, это обусловлено наличием открытой сессии РНР. Так как сессия открыта – то происходит генерация HTML кода для авторизованного пользователя.

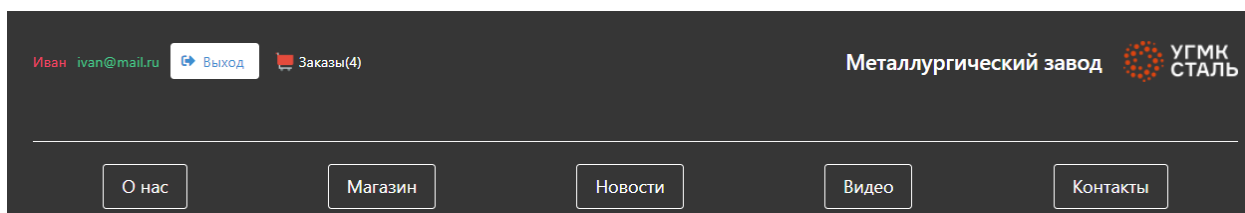


Рисунок 23 – Авторизованный пользователь

При оформлении заказа в личном кабинете пользователя будет выведено сообщение с номером его оформленного заказа. (рисунок 24).

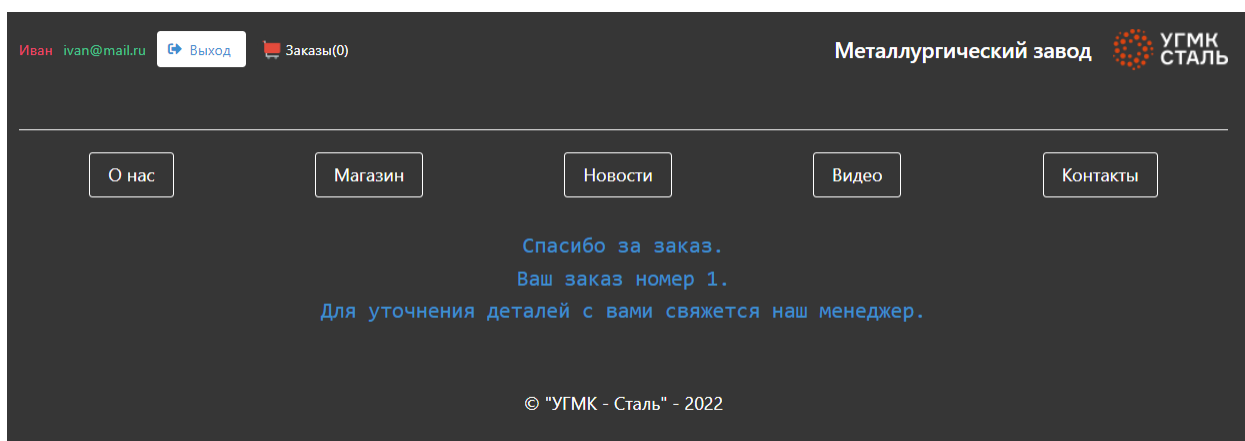


Рисунок 24 – Уведомление о заказе

С помощью административной панели предоставляется возможность управлять данными каталога товаров и следить за списком заказов. Чтобы попасть в административную панель необходимо в адресной строке браузера набрать адрес `ugmk/admin.php`, после чего откроется страница для авторизации. (рисунок 25).

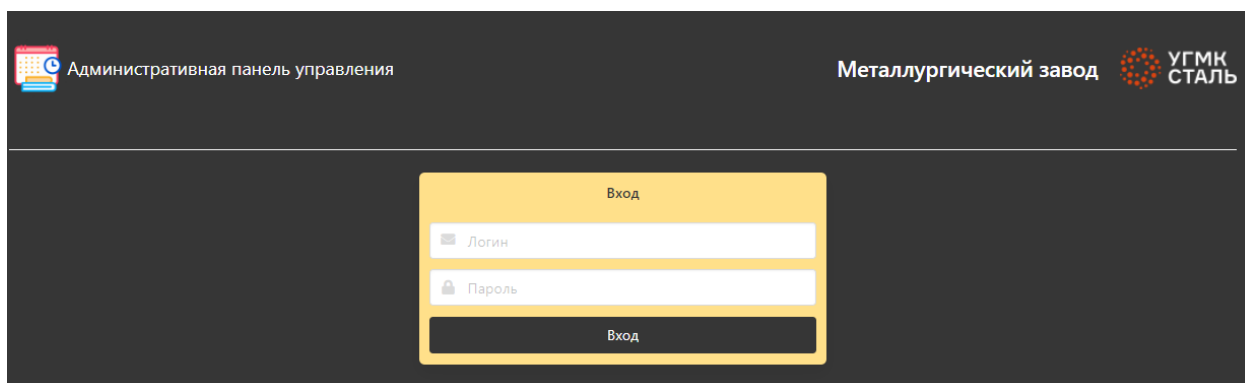


Рисунок 25 – Авторизация администратора

Процесс авторизации аналогичен, как и в случае с обычным пользователем – включается механизм сессий и сравнения переданных данных с записью в базе [22].

После прохождения авторизации в роли администратора сотруднику становится доступно управление каталогом. Страница панели администратора состоит из трёх частей:

- Таблица заказов (см. рисунок 26)
- Форма добавления товара (см. рисунок 27)
- Список товаров с возможностью удаления товара (см. рисунок 28)

Данные предварительно запрашиваются в базе, после чего выводятся на страницу.

Административная панель управления Металлургический завод УГМК СТАЛЬ

ЗАКАЗЫ

Номер заказа	Клиент	Телефон	Метод оплаты	Сумма	Комментарий	Состав заказа
1	Иван		Картой	211000.00	ок	<div style="background-color: #e0e0e0; padding: 2px; margin-bottom: 2px;">Арматура 16/A500C/ГОСТ Р 52544-2006-1шт.</div> <div style="background-color: #e0e0e0; padding: 2px;">Арматура 22/A500C/ГОСТ Р 52544-2006-3шт.</div>

Рисунок 26 – Список заказов

Добавление товара осуществляется посредством веб-формы с применением технологий AJAX – данные отправляются на сервер асинхронно без перезагрузки страницы, после чего скрипт PHP записывает новый товар в каталог и отправляет ответ на клиентский скрипт. Клиентский скрипт в случае получения положительного ответа отрисует список каталога внизу заново, добавив туда новый товар [16].

ДОБАВЛЕНИЕ НОВОГО ТОВАРА

Название

Описание

Арматура – это стальной металлопрокат в виде металлического стержня с гладким или чаще всего с периодическим профилем используется в строительстве для придания прочности бетонным конструкциям улучшая характеристики натяжения железобетонных конструкций.

Категория

Цена

Изображение

	arm.jpg	1021.8KB	
--	---------	----------	--

Дополнительно Популярный Новинка

Наличие Есть Нет

ДОБАВИТЬ

Рисунок 27 – Добавление товара

Фото	Наименование	Описание	Категория	Цена, руб.	Популярный	Новинка	Наличие	Удалить
	Арматура 10/A500C/ГОСТ Р 52544-2006	Арматура – это стальной металлопрокат в виде металлического стержня с гладким или чаще всего с периодическим профилем используется в строительстве для придания прочности бетонным конструкциям улучшая характеристики натяжения железобетонных конструкций.	1	45000.00	0	0	yes	
	Арматура 16/A500C/ГОСТ Р 52544-2006	Арматура – это стальной металлопрокат в виде металлического стержня с гладким или чаще всего с периодическим профилем используется в строительстве для придания прочности бетонным конструкциям улучшая характеристики натяжения железобетонных конструкций.	1	49000.00	0	0	yes	
	Арматура 22/A500C/ГОСТ Р 52544-2006	Арматура – это стальной металлопрокат в виде металлического стержня с гладким или чаще всего с периодическим профилем используется в строительстве для придания прочности бетонным конструкциям улучшая характеристики натяжения железобетонных конструкций. Арматура A500C — это самый распространённый	1	54000.00	1	1	yes	

Рисунок 28 – Каталог товаров

При добавлении товара происходит валидация всех полей формы с помощью JQuery.

ДОБАВЛЕНИЕ НОВОГО ТОВАРА

Название

Введите значение больше четырёх символов

Описание

Заполните поле не менее 10 символами

Категория

Цена

Изображение

Дополнительно Популярный Новинка

Наличие Есть Нет

ДОБАВИТЬ

Рисунок 29 – Валидация формы

Также на сайте разработаны несколько вспомогательных страниц, представленных на рисунках 30, 31, 32, 33. Данные страницы генерируются скриптами PHP, а контентная часть этих страниц запрашивается в базе данных в соответствующей таблице.

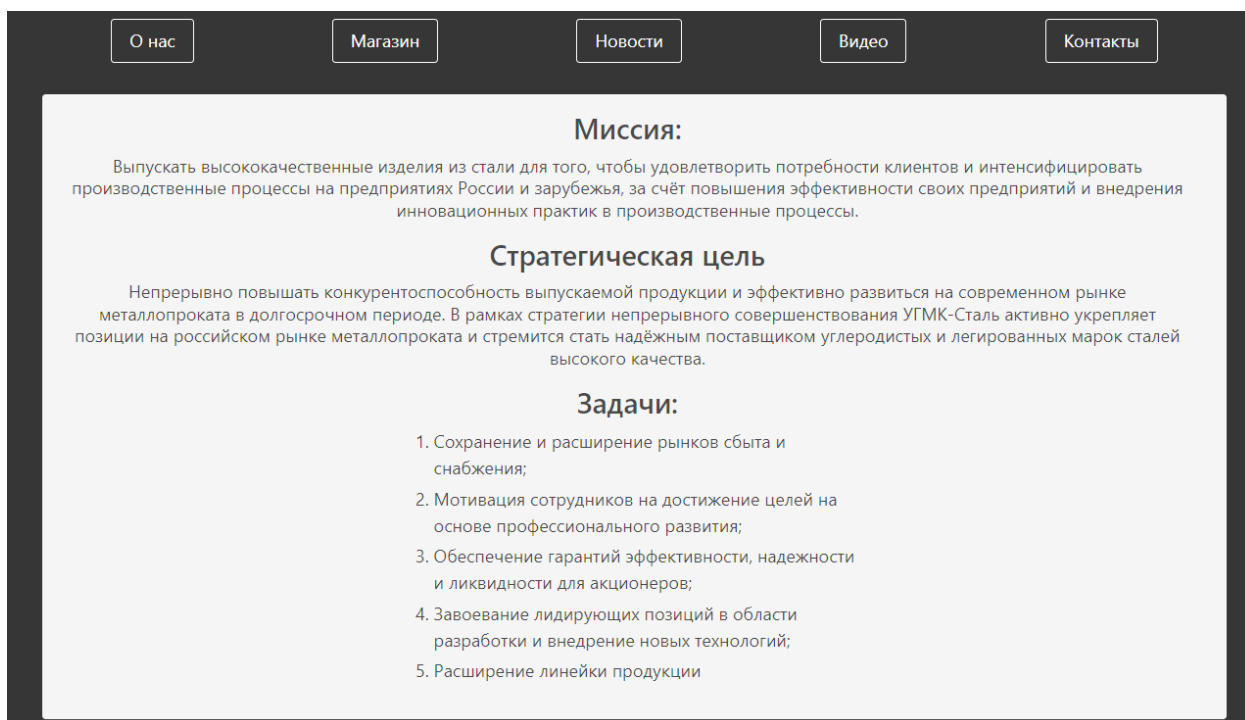


Рисунок 30 – Страница «О нас»

На рисунке 31 представлен скриншот страницы «Новости».

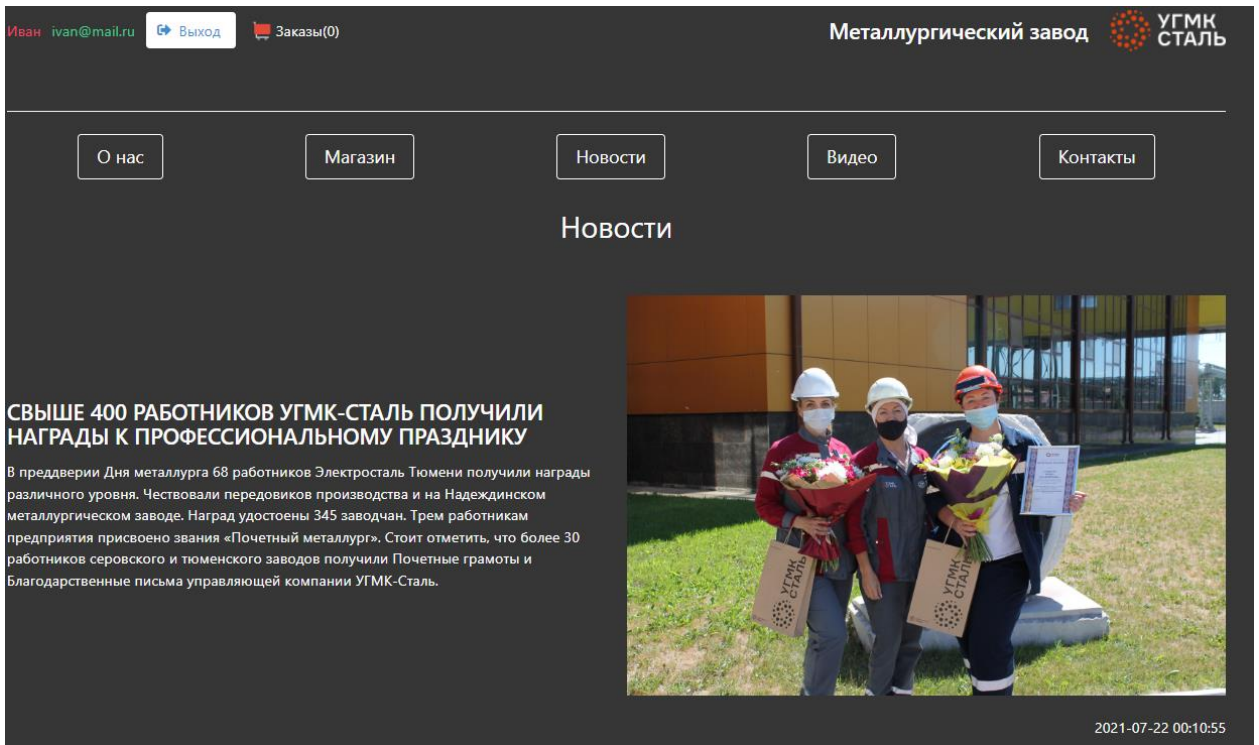


Рисунок 31 – Страница «Новости»

На рисунке 32 представлен скриншот страницы «Видео».

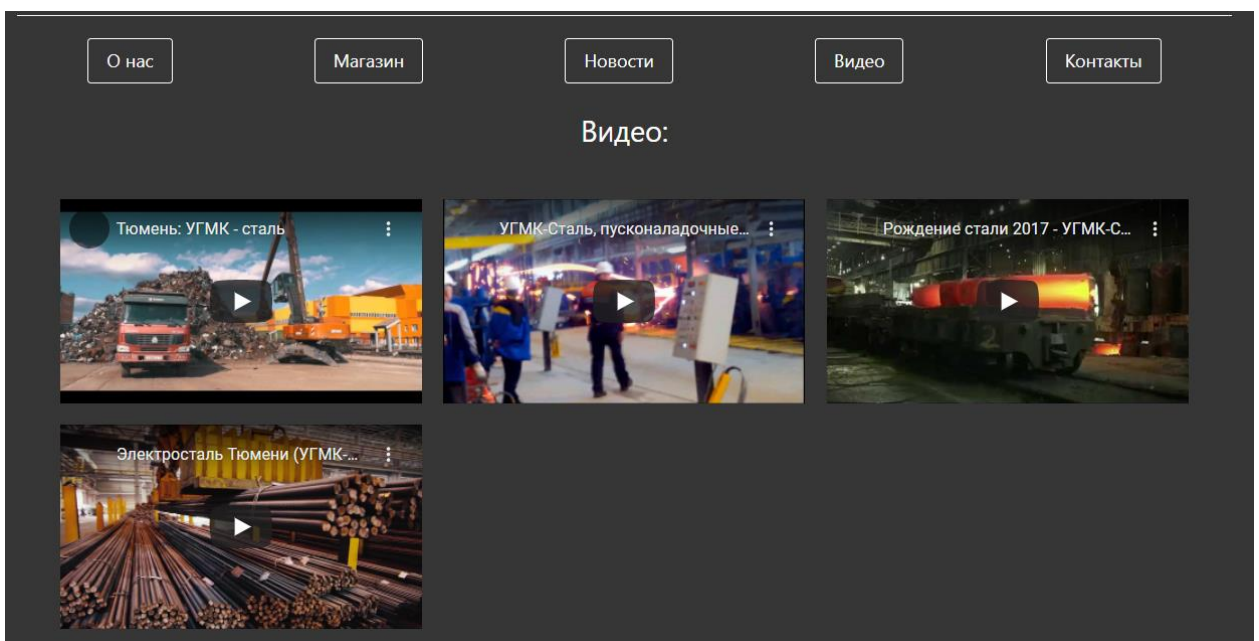


Рисунок 32 – Страница «Видео»

На рисунке 32 представлен скриншот страницы «Контакты».

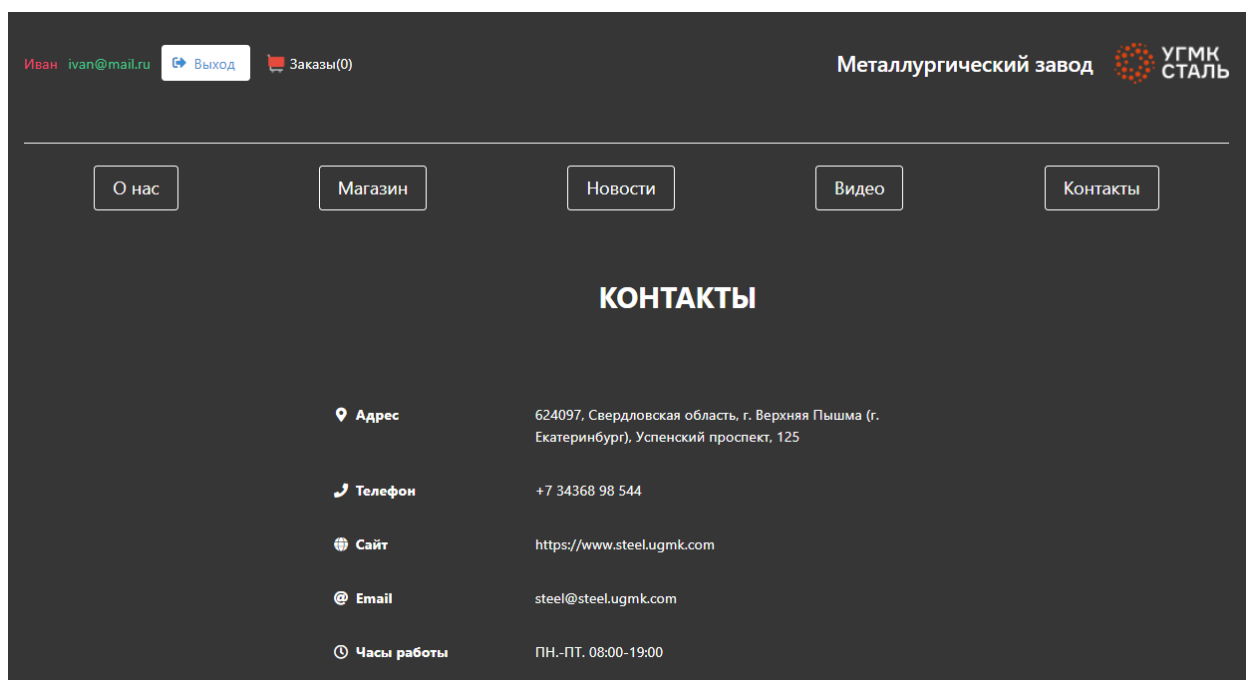


Рисунок 33 – Страница «Контакты»

Таким образом, в данной главе были продемонстрированы скриншоты функционального тестирования разработанного интернет-магазина, которые показали, что магазин работает исправно без ошибок и в соответствии с заданием. Интерфейс магазина соответствует заданным параметрам и заданию на разработку.

Заключение

В данной работе был спроектирован и реализован интернет-магазин для организации ООО «УМК-Сталь». Авторизация и регистрация на сайте выполнена отдельными модулями, реализующими полноценную валидацию и проверку данных, введённых покупателем при заполнении форм.

Панель администратора доступна по специальной ссылке, которую необходимо ввести в браузер. После заполнения формы авторизации администратора и отправки данных на сервер определяется возможность доступа для управления сайтом.

На сайте работает модуль фильтрации товаров, который позволяет гибко настроить сортировку по товарам.

При добавлении товара администратором могут быть выбраны дополнительные параметры товара. При наличии таких меток скрипт PHP сформирует графическое отображение таких товаров, добавив к ним соответствующие картинки.

Часть функционала магазина решается посредством асинхронных AJAX запросов из скриптов JavaScript к серверу: загрузка товаров по категории, формирование корзины, фильтрация, добавление товара в каталог и удаление из каталога.

Магазин реализован с помощью таких языков программирования как PHP и JavaScript, база данных выполнена на MySQL, что позволит легче интегрировать и развернуть сайт на реальном хостинге. Также при разработке были использованы CSS фреймворк Vulma и библиотека JQuery.

В дальнейшем при развитии магазина может быть предложен ряд улучшений функционала, например внедрение модуля оплаты и накопительной системы скидок для покупателей.

Список используемой литературы

1. Арисова Д. А., Чернова С. В. К вопросу о веб-разработках // Вестник науки и образования. 2018. №15-2 (51). URL: <https://cyberleninka.ru/article/n/k-voprosu-o-veb-razrabotkah> (дата обращения: 27.03.2023).
2. Астафьева, В. В. Принципы и правила проектирования пользовательского интерфейса // Молодой ученый. — 2020. — № 1 (291). — С. 10-14
3. Ахмеджанова З., Гафурова П. Применение html и css для создания интерактивных Веб сайтов // Евразийский Союз Ученых. 2019. №4-3 (61). URL: <https://cyberleninka.ru/article/n/primenenie-html-i-css-dlya-sozdaniya-interaktivnyh-veb-saytov> (дата обращения: 28.03.2023).
4. Боркова Е.А., Носкова П.О. Современное развитие рынка электронной торговли в макроэкономических условиях России // Российское предпринимательство. – 2019. – Том 20. – № 4. – С. 917-926. – doi: 10.18334/rp.20.4.40598
5. Васильев, А. Программирование на PHP в примерах и задачах/ изд. Эксмо, 2021. – 352 с.
6. В. Дронов. PHP и MySQL для начинающих /СПб.: БХВ-Петербург, 2021. — 432 с.: ил. — (Для начинающих)
7. В. Дронов. HTML, Н. Прохоренок. JavaScript, PHP и MySQL. Джентльменский набор Web-мастера /СПб.: БХВ-Петербург, 2019. —912 с.
8. Веру Леа: Секреты CSS. Идеальные решения ежедневных задач. Питер, 2019. – 336с.
9. Володченко В. С., Донцова Д. С., Миронова Т. А. Сапунова Е. В., Бышок К. А. Разработка web-сайта // Достижения науки и образования. 2019. №12 (53)
10. Головенчик Г. Сущность, классификация и особенности электронной коммерции // Наука и инновации. 2020. №4 (206)

11. Дакетт Джон. HTML и CSS. Разработка и дизайн вебсайтов./СПб.: БХВ-Петербург, 2020. - 330 с.
12. Д. В. Котеров. PHP 8. Наиболее полное руководство /СПб.: БХВ-Петербург, 2023. - 992 с.: ил. — (В подлиннике)
13. Кандиев С. С. Электронная торговля: особенности и тенденции развития // Молодой ученый. — 2023. — № 1 (448). — С. 76-81
14. Куликов С. Работа с MySQL, MS SQL Server и Oracle в примерах Практическое пособие для программистов и тестировщиков / Версия книги 1.0.3 от 12.04.2018. – 546 с.
15. М. Фленов. PHP глазами хакера/СПб.: БХВ-Петербург, 2020. —256 с.: ил. — (4 издание)
16. Р. Никсон. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. Питер, 2019. – 816с
17. Сайт <https://www.mysql.com> [Электронный ресурс]. URL: <https://www.mysql.com> (дата обращения: 10.03.2023).
18. Сайт <https://www.w3schools.com/sql/> - раздел “SQL tutorial” [Электронный ресурс]. URL: <https://www.w3schools.com/sql/> (дата обращения: 10.03.2023).
19. Свекис Л.Л. JavaScript с нуля до профи. / изд. Питер, 2023. - 480с.
20. Шабашов, В.Я. Организация доступа к данным из PHP приложений для различных СУБД – Москва ; Берлин : Директ-Медиа, 2019. – 121 с.
21. Schultz Bonnie R. Introductory Relational Database Design for Business /Wiley, 2018. – 456 p.
22. Joseph Edmonds, Lorna Jane Mitchell. The Art of Modern PHP 8. Packt Publishing, 2021. 690 p.
23. Kalob Taulien, PHP Crash Course: Learn PHP in 90 minutes. Packt Publishing, 2020. 383 p.
24. Milecia McGregor, JavaScript: The Hidden Parts. O'Reilly Media, Inc., 2023. 870 p.

25. Eric Meyer, Estelle Weyl. CSS: The Definitive Guide, 5th Edition, O'Reilly Media, Inc., 2023. 678 p.