Государственное бюджетное профессиональное образовательное учреждение

Ростовской области

|  |
| --- |
| «Таганрогский колледж морского приборостроения» |

|  |
| --- |
|  |
| «Чистый дом» |
| Курсовой проект |
| Пояснительная записка |
| ТКМП.МДК.02.01.10.025 ПЗ |

|  |  |
| --- | --- |
| Руководитель  *Малыхина О.В.*  Студент П-320  *Журенко Д.М.* |  |
|  |

2023

**Содержание**

Титульный лист

Задание

Аннотация

Содержание

Введение

1 Общая часть

1.1Постановка задачи

1.2Анализ и исследование задачи, построение модели системы

1.3Разработка архитектуры решения

1.4 Обоснование и выбор средств разработки решения

1.5Формализация расчетов

1.5.1Проектирование базы данных

1.5.2Проектирование интерфейса

2 Описание программы

3 Тестирование программы

3.1 Разработка плана тестирования.

3.2Оценка результатов проведения тестирования

Заключение

Список использованных источников

**Задание**

Вашей задачей является разработка системы для компании, которая оказывает услуги

клиентам и продаёт определенные товары.

Данную систему можно разделить на несколько подсистем:

* подсистема по работе с клиентами,
* подсистема по работе с сотрудниками,
* подсистема по работе с товарами,
* подсистема по работе с услугами,
* подсистема для управления компанией.

Наша компания “Чистый дом” вдобавок к оказываемым услугам продает различную

бытовую химию. Клиентам очень удобно, что теперь нет необходимости заранее

заказывать в сторонних магазинах различную бытовую химию. Однако для

администраторов сервиса необходима система для учета товаров.

Вам предстоит разработать подсистему для работы с товарами нашей компании, которая

должна включать в себя следующий функционал:

* просмотр списка товаров,
* добавление/удаление/редактирование данных о товарах,
* предложения дополнительных товаров к основным,
* просмотр истории покупок товаров.

**Аннотация**

Данный курсовой проект описывает разработку веб-ресурса для автосервиса “Чистый дом” с целью упрощения работы с клиентами и увеличения эффективности продаж товаров. Система состоит из нескольких подсистем: по работе с клиентами, с сотрудниками, с товарами, с услугами и для управления компанией. Студентам предстоит разработать подсистему для работы с товарами компании, которая будет позволять просматривать список товаров, добавлять, удалять и редактировать данные, а также предлагать дополнительные товары к основным и просматривать историю покупок товаров компании. Основная задача заключается в создании системы, которая будет не только удобной для администраторов сервиса, но и для клиентов, которым будет предоставлена возможность приобретать нужные товары без необходимости посещения сторонних магазинов. Разработка данной системы будет способствовать повышению качества и эффективности работы веб-ресурса.

**Введение**

В современном мире у нас очень много обязательств на работе и в личной жизни, мы всегда куда-то спешим и у нас не хватает времени на домашние дела, в том числе на уборку дома. В этой ситуации все больше людей начинают использовать услуги компаний, которые предоставляют профессиональную уборку - и именно этой теме посвящен данный курсовой проект. В рамках данной работы рассмотрим веб-ресурс «Чистый дом», который является одним из лидеров на рынке профессиональной уборки в нашем регионе. Цель данного проекта заключается в разработке подсистемы для работы с товарами, которая включает в себя создание базы данных, учет и продажу товаров, а также управление всеми процессами, связанными с товарами. Данная подсистема - важная часть автоматизации работы компании и позволит сотрудникам более эффективно работать и обеспечивать клиентам лучшее обслуживание. Работа над проектом включает в себя разработку нескольких подсистем, таких как: работу с клиентами, работу с сотрудниками, работу с товарами и работу с услугами. В данном проекте мы сфокусируемся на создании подсистемы для работы с товарами. Цель данного исследования - разработать подсистему, которая будет включать в себя структуру базы данных и функционал для учета товаров. Кроме того, мы также планируем рассмотреть функционал системы, включающий в себя добавление, удаление и редактирование данных о товарах, предложение дополнительных товаров к основным и просмотр истории покупок товаров магазина. Для достижения этой цели мы будем использовать различные методы и технологии, которые позволят разработать эффективную и удобную систему для работы с товарами. В исследовании мы рассмотрим методы работы с базой данных, используя язык SQL, а также технологии программирования, такие как PHP. Ожидается, что разработка подсистемы для работы с товарами в автосервисе “Спасу колесо” значительно усовершенствует работу компании и повысит качество предоставляемых услуг.

**1. Общая часть**

**1.1. Постановка задачи**

Постановка задачи:

Целью данной курсовой работы является разработка подсистемы учета товаров для веб-ресурса "Чистый дом". Для достижения данной цели необходимо выполнить следующие задачи:

1. Изучение требований к подсистеме работы с товарами.

2. Разработка архитектуры подсистемы, включающей: - подсистему просмотра и редактирования списка товаров, - подсистему добавления/удаления/редактирования данных о товарах, - подсистему предложения дополнительных товаров к основным, - подсистему просмотра истории покупок товаров клиентов.

3. Разработка базы данных для хранения информации о товарах и их истории покупок.

4. Разработка интерфейса подсистемы работы с товарами.

5. Тестирование разработанной подсистемы на соответствие требованиям и корректировка необходимых функций.

6. Написание документации по работе с подсистемой работы с товарами. В результате выполнения курсовой работы должна быть создана работоспособная подсистема для работы с товарами нашей компании, обеспечивающая удобство для администрирования товаров и учета их движения

Для определения основных функциональных частей приложения, их назначение и взаимосвязи между ними, была разработана структурная схема. Структурная схема, представлена в соответствии с приложением В.

Для написания программы был выбран язык PHP, т.к. он является C-подобным скриптовым языком общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов.

**1.2. Анализ и исследование задачи, построение модели системы**

Анализ задачи:

Тема проекта - веб-ресурс «Чистый дом». Задача разработать систему для компании, которая оказывает услуги клиентам и продаёт товары. Система разделена на несколько подсистем: работа с клиентами, с сотрудниками, с товарами, с услугами и для управления компанией. Требуется разработать подсистему для работы с товарами, которая должна включать в себя просмотр списка товаров, добавление/удаление/редактирование данных о товарах, предложения дополнительных товаров к основным и просмотр истории покупок товаров клиентов.

Исследование задачи:

Для начала исследуем, какие данные необходимо предоставить в подсистеме для работы с товарами. Нам нужен список всех запчастей, которые продаются в нашей компании, цену на товар, описание, наличие и привязку к конкретному автомобилю. Для эффективного управления продажами и учета складских остатков, система должна позволять администратору добавлять, удалять и редактировать информацию о товарах. Также важно, чтобы система предлагала дополнительные товары к основным.. Наконец, система также должна предоставлять возможность просмотра истории покупок автосервиса. Это позволит администраторам управлять запасами и заказывать товары в соответствии с потребностями компании.

Построение модели системы:

Модель системы будет состоять из следующих элементов:

1. База данных товаров - здесь хранится информация о всех товарах, которые продаются компанией.

2. Функция добавления товара - позволяет администраторам добавлять информацию о новых товарах в базу данных.

3. Функция удаления товара - позволяет администраторам удалять информацию о товарах из базы данных.

4. Функция редактирования товара - позволяет администраторам изменять информацию о товарах в базе данных.

5. Функция просмотра списка товаров - позволяет пользователям просматривать все товары на сайте.

6. Функция предложения дополнительных товаров к основным - позволяет системе предлагать дополнительные товары клиентам при оформлении заказа.

7. Функция просмотра истории покупок товаров компании - позволяет администраторам просматривать историю продаж для управления запасами и заказов товаров.

Итоговая модель системы должна быть удобной для пользователей и администраторов, позволяя им легко обновлять и просматривать информацию о товарах, а также управлять запасами и заказами товаров для компании.

**1.3.Разработка архитектуры решения**

Для разработки подсистемы для работы с товарами веб-ресурса “Чистый дом” необходимо определить ее архитектуру. Архитектура системы должна обеспечивать исключительную производительность, надежность и безопасность при работе с данными клиентов и товаров сайта. В качестве архитектурного решения можно применить следующие подходы:

1. Микросервисная архитектура: Этот подход предполагает разбиение приложения на независимые микросервисы, каждый из которых отвечает за выполнение отдельной функциональности. Для работы с товарами можно выделить отдельный микросервис, который будет обеспечивать определенный функционал работы с товарами. Каждый микросервис может иметь свой собственный индивидуальный набор взаимодействующих с ним процессов и элементов, что снижает вероятность возникновения сбоев и повышает отказоустойчивость системы.
2. Клиент-серверная архитектура: Это классический подход в веб-разработке, который предполагает разделение приложения на клиентскую и серверную части. Основная логика и данные хранятся на сервере, а клиентская часть отвечает за отображение информации и взаимодействие с пользователем. В данной архитектуре в качестве серверной части может выступать отдельный сервер, на котором будет развернуто приложение для работы с товарами. Клиентская часть приложения будет предоставлять пользователю возможность просматривать, добавлять, удалять и редактировать данные о товарах.
3. Эволюционная архитектура: Она представляет собой подход к разработке ПО, основанный на постепенном приращении функционала системы. Компоненты и модули системы постепенно добавляются к уже имеющейся инфраструктуре. Для разработки подсистемы для работы с товарами можно начать с небольшого приложения, которое будет обеспечивать просмотр списка товаров и истории покупок. Дополнительный функционал, такой как добавление/удаление/редактирование данных о товарах и предложения необходимых запчастей к основному товару, может быть добавлен в последующих этапах разработки. При выборе одного из этих подходов нужно учитывать требования к производительности, ограничения базы данных, требования к защите данных и возможности разработчиков. Комбинация различных архитектурных подходов может обеспечить наилучший баланс между функциональностью, производительностью и безопасностью системы.

**1.4 Обоснование и выбор средств разработки решения**

Выбранное средство разработки решения для данного курсового проекта на тему "Чистый дом" - веб-ресурс для учета товаров нашего сайта "Чистый дом" - состоит из трех основных компонентов: HTML, CSS и JavaScript. Наиболее подходящим и распространенным инструментом, который обеспечивает все эти возможности является PHP. Выбор средства разработки веб-ресурса для данного проекта Ha сделан, исходя из следующих причин:

1. HTML - это стандартный язык разметки веб-страниц, который является основным языком для создания всей структуры и функциональности веб-сайта.

2. CSS - это язык таблиц стилей, который позволяет определить стиль веб-страницы, изменять ее внешний вид и улучшать пользовательский опыт. 3. JavaScript - это язык программирования, который позволяет создавать интерактивные элементы веб-страниц, улучшать навигацию по сайту и предоставлять пользователю средства для взаимодействия со страницей. 4. PHP - это скриптовый язык программирования, который является очень распространенным веб-средством для создания и управления базами данных.

5. База данных сделана через XAMPP. Это кроссплатформенный дистрибутив, который может использоваться на Unix, Linux, Mac и Windows. XAMPP включает в себя Apache, PHP, MySQL и Perl, что делает его идеальным выбором для разработки веб-приложений.

6. PHP является наиболее распространенным языком программирования для создания веб-приложений с открытым исходным кодом. Он поддерживает работу с большинством баз данных, включая MySQL, SQLite, PostgreSQL и другие. В результате, выбор использования CSS, PHP и JavaScript оправдан, так как они позволяют создавать качественный и функциональный веб-ресурс для работы с товарами на сайте. Кроме того, использование XAMPP позволит обеспечить надежность и стабильность базы данных и приложения в целом.

**1.5.Формализация расчетов**

Для формализации расчетов для разработки подсистемы учета товаров на веб-ресурсе "Чистый дом" необходимо определить следующие шаги:

1. Определить ожидаемые результаты работы подсистемы: - Возможность просмотра списка товаров - Возможность добавления/удаления/редактирования данных о товарах - Возможность предложения дополнительных товаров к основным - Возможность просмотра истории покупок товаров клиента.
2. Определить методы передачи данных: - Использование базы данных через xampp - Использование языков программирования: css, php, javascript
3. Определить логику работы подсистемы: - Клиенты могут просматривать список товаров и делать заказы - Администраторы автосервиса могут добавлять/удалять/редактировать данные о товарах и предложения к ним - Подсистема должна иметь возможность формирования отчетов о покупках товаров автосервиса
4. Определить использование инструментов программирования: - Для создания интерфейса можно использовать языки программирования HTML и CSS - Для обработки запросов и отображения данных необходимо использовать сценарии на языке программирования PHP - Для реализации динамической интерактивности можно использовать языки программирования AJAX и JavaScript
5. Определить требования к базе данных: - Необходимо определить таблицы хранения информации о товарах и предложениях к ним - Необходимо определить таблицы хранения информации о покупках автосервиса
6. Определить и описать процессы и взаимодействия между компонентами системы: - Клиенты должны иметь возможность через веб-интерфейс просматривать список товаров и оформлять покупку. После этого данные о заказе должны попасть в базу данных. - Администраторы должны иметь возможность через интерфейс управления добавлять/удалять/редактировать данные о товарах и описания к ним. - Подсистема должна иметь возможность формирования списка покупок автосервиса и вывода этой информации на экран. С учетом этих шагов необходимо разработать программный код и написать документацию по проекту.

**1.5.1.Проектирование базы данных**

Для проектирования базы данных для веб-ресурса «Оказание ремонтных автоуслуг» необходимо следовать определенной структуре. При этом можно использовать реляционную модель. Следующие таблицы могут быть использованы в данной базе данных:

1. Таблица "Клиенты"

- ID клиента (уникальный идентификатор)

- Имя клиента

- Адрес электронной почты

- Номер телефона

- Адрес

2. Таблица "Сотрудники"

- ID сотрудника (уникальный идентификатор)

- Имя сотрудника

- Должность

3. Таблица "Товары"

- ID товара (уникальный идентификатор)

- Название товара

- Категория товара

- Цена

- Описание товара

4. Таблица "Услуги"

- ID услуги (уникальный идентификатор)

- Название услуги

- Цена

- Описание услуги

5. Таблица "История покупок"

- ID покупки (уникальный идентификатор)

- ID товара

- ID клиента

- Количество товаров

- Дата покупки

Для связи этих таблиц можно использовать сводные ключи:

- ID клиента из таблицы "Клиенты" связывается с ID покупки из "Истории покупок"

- ID товара из таблицы "Товары" связывается с ID покупки из "Истории покупок"

С использованием PHP и JavaScript можно разработать интерфейс для взаимодействия с базой данных для выполнения вышеперечисленных функций. Просмотр списка товаров может происходить путем запроса на получение всех записей из таблицы "Товары" и отображения их на странице. Добавление, удаление и изменение данных о товарах может осуществляться при помощи SQL-запросов на добавление, удаление и обновление данных в таблице "Товары". Предложения дополнительных товаров могут быть сделаны путем добавления новых записей в таблицу "Товары" с признаком связи с другим товаром. Просмотр истории покупок может быть осуществлен путем запроса на получение всех записей из таблицы "История покупок" и отображения их на странице. Для удобства учета товаров администраторы сервиса могут использовать специальную админ-панель, где будут представлены все вышеперечисленные функции. Таким образом, разработка подсистемы для работы с товарами имеет несколько ключевых моментов: создание таблицы товаров, добавление, удаление, редактирование данных о товарах, предложения дополнительных товаров к основным и просмотр истории покупок товаров автосервиса. База данных может быть создана с помощью XAMPP, а интерфейс взаимодействия с базой данных может быть разработан с использованием PHP и JavaScript.

**1.5.2.Проектирование интерфейса**

Для успешного проектирования интерфейса подсистемы по работе с товарами необходимо учитывать следующие пункты:

1. Навигация по сайту должна быть простой и удобной. Для этого можно создать панель навигации, которая будет содержать ссылки на все подсистемы.

2. Главная страница должна содержать основную информацию о компании и ее услугах. Также на главной странице можно разместить список наиболее популярных товаров.

3. Страница со списком товаров должна содержать таблицу, которая будет включать следующие поля: название товара, стоимость, наличие, возможность заказа и редактирования.

4. Для добавления нового товара можно создать форму, которая будет содержать необходимые поля для заполнения: название, описание, стоимость, количество.

5. Для удаления товара из базы данных можно создать кнопку "Удалить". При нажатии на данную кнопку товар должен удалиться из базы данных. 6. Для редактирования данных о товаре можно создать форму, которая будет содержать все поля, заполненные текущими данными. Для изменения необходимых данных необходимо изменить соответствующие поля и нажать кнопку "Сохранить".

7. Предложения дополнительных товаров к основным могут быть оформлены в виде блока, который будет содержать подсказки с ссылками на дополнительные товары.

8. История покупок товаров автосервиса может быть оформлена в виде таблицы. Таблица должна содержать следующие поля: название товара, количество, дата покупки, общая сумма покупки.

9. Весь интерфейс должен быть адаптивным и подходить для использования на различных устройствах с разными разрешениями экранов.

10. Необходимо также придерживаться общих стандартов и требований к дизайну и интерфейсу веб-страниц, включая использование читабельных шрифтов, цветовую гамму, логотип, и т.д. Написание веб-ресурса на основе CSS, PHP, JavaScript и базы данных созданой в XAMPP, позволит создать профессиональный, удобный и функциональный интерфейс для подсистемы по работе с товарами нашей компании.

**2. Описание программы**

Программа, разработанная для автосервиса "Чистый дом", является веб-ресурсом, который оказывает услуги по продаже бытовой химии. В соответствии с этой задачей, была разработана система, которая включает в себя подсистемы по работе с клиентами, сотрудниками, товарами, услугами и для управления компанией.

Специальное внимание уделено разработке подсистемы для работы с товарами, которая состоит из следующего функционала:

- Просмотр списка товаров, где клиенты могут ознакомиться с наличием товаров и их ценами.

- Добавление/удаление/редактирование данных о товарах. Администраторы могут легко добавлять новые товары, редактировать информацию о существующих и удалять товары, которые больше не нужны.

- Предложения дополнительных товаров к основным. Пользователи могут ознакомиться с дополнительными предложениями, связанными с основными товарами, которые они уже выбрали. Это помогает клиентам сделать более информированный выбор и улучшить их опыт работы с компанией "Чистый дом".

- Просмотр истории покупок товаров клиента. Администраторы могут легко отслеживать историю покупок и узнавать, какие товары пользуются большой популярностью.

Для разработки веб-ресурса были использованы следующие технологии: CSS, PHP и JavaScript. База данных была создана с помощью XAMPP. Эти технологии помогли разработчикам создать простой в использовании и удобный веб-ресурс для компании, который позволяет клиентам и сотрудникам быстро получать доступ к нужным информационным и техническим подсистемам.

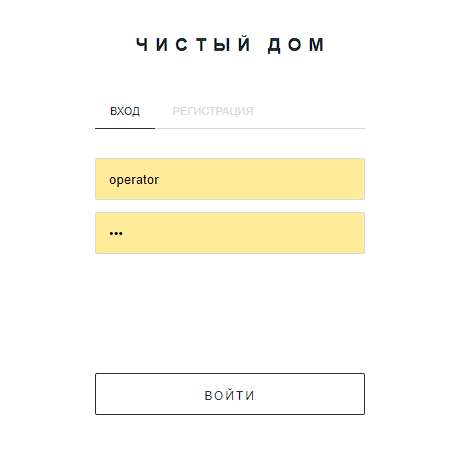
**3. Тестирование программы**

Способ проведения тестирования:

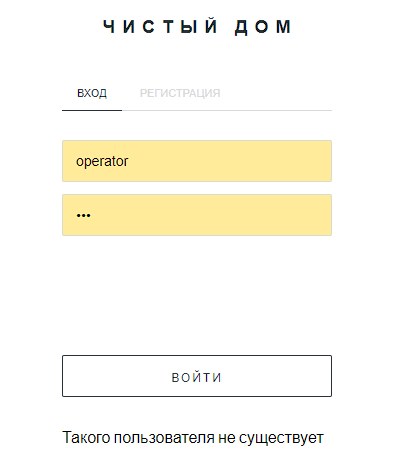
1. Проверить работоспособности авторизации
2. Проверить работоспособность функций товара
3. Проверить работоспособность функции выхода из аккаунта

Тестирование:

1. Для проверки авторизации мы введём логин и пароль в соответствующие поля и нажмём кнопку «Войти». Пример авторизации в соответствии с рисунком 5.1. Если ввести неправильный логин или пароль, то система выдаст ошибку в соответствии с рисунком 5.2.

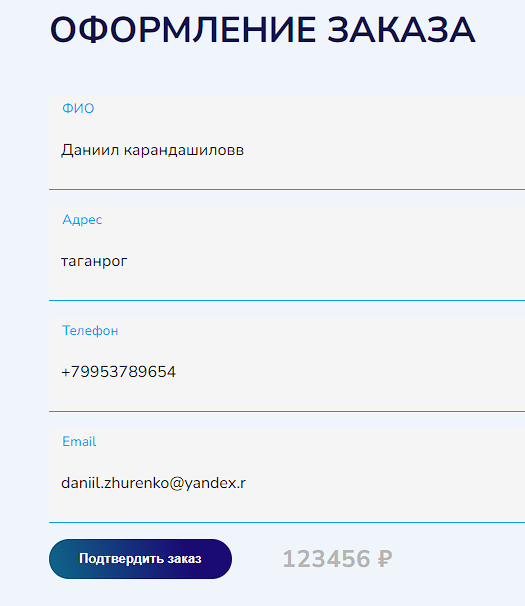
****

**Рисунок 1.1 Авторизация**

****

**Рисунок 1.2 Ошибка авторизации**

2)Для проверки работоспособности функций заказа товара мы сначала должны выбрать вариант который нам подходит



**Рисунок 1.3 Оформление заказа**

**4 Программирование**

Система автосервиса спроектирована на языке PHP и БД MySQL.

Описание запросов MySQL:

1. Создание таблицы «История».

CREATE TABLE `history` (

`id` int(11) NOT NULL,

`iduser` int(11) NOT NULL,

`id\_prod` int(11) NOT NULL,

`fio` varchar(100) NOT NULL,

`address` text NOT NULL,

`phone` text NOT NULL,

`email` text NOT NULL

1. Создание таблицы «Категории».

CREATE TABLE `category` (

`id` int(11) NOT NULL,

`title` varchar(100) NOT NULL,

`imgpath` varchar(500) NOT NULL

1. )Создание таблицы «Продукты».

CREATE TABLE `product` (

`id` int(11) NOT NULL,

`idcateg` int(11) NOT NULL,

`title` varchar(100) NOT NULL,

`descrip` text NOT NULL,

`price` int(11) NOT NULL,

`imgpath` varchar(500) NOT NULL

1. )Создание таблицы «Пользователи».

CREATE TABLE `users` (

`id` int(11) NOT NULL,

`usrgroup` int(11) NOT NULL,

`login` text NOT NULL,

`pass` text NOT NULL,

`email` text NOT NULL

)

Сценарии в коде программы:

1. log.php – Авторизация

Входные данные: логин и пароль  
Выходные данные: вход на сайт

1. delete\_таблица.php – Удаление строки из таблицы

Входные данные: ID строки  
Выходные данные: удалённая строка

1. update\_таблица.php – Изменение строки в таблице

Входные данные: ID строки и новые данные  
Выходные данные: изменённая строка

1. db.php – Подключение к базе данных

Входные данные: Название БД, пароль, имя учётной записи phpMyAdmin  
Выходные данные: Данные в таблицах

**5 Отладка, оптимизация и тестирование программы**

Способ проведения тестирования:

1. Проверить работоспособности авторизации
2. Проверить работоспособность функций записи, изменения и удаления нарядов
3. Проверить работоспособность функции записи, изменения и удаления сотрудников
4. Проверить работоспособность функции записи, изменения и удаления деталей
5. Проверить работоспособность функции выхода из аккаунта

Тестирование:

1. Для проверки авторизации мы введём логин и пароль в соответствующие поля и нажмём кнопку «Войти». Пример авторизации в соответствии с рисунком 5.1. Если ввести неправильный логин или пароль, то система выдаст ошибку в соответствии с рисунком 5.2.

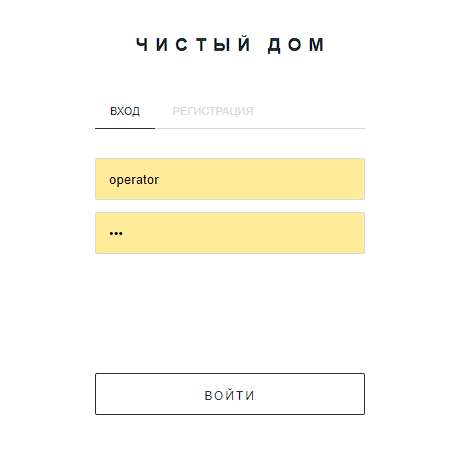
****

Рисунок 1.4 – Авторизация

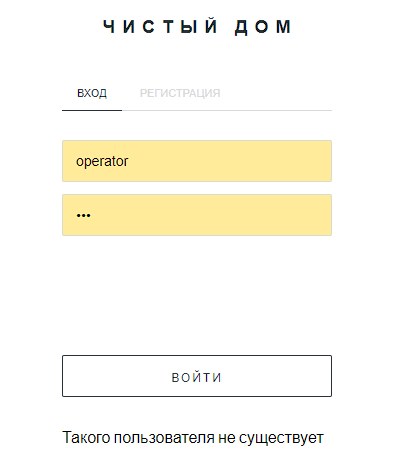
****

Рисунок 1.5 – Ошибка авторизации

1. Для проверки функций во вкладке «Купить» введём данные в поля для добавления заказа и нажмём кнопку «Подтвердить заказ» в соответствии с рисунком 1.6.

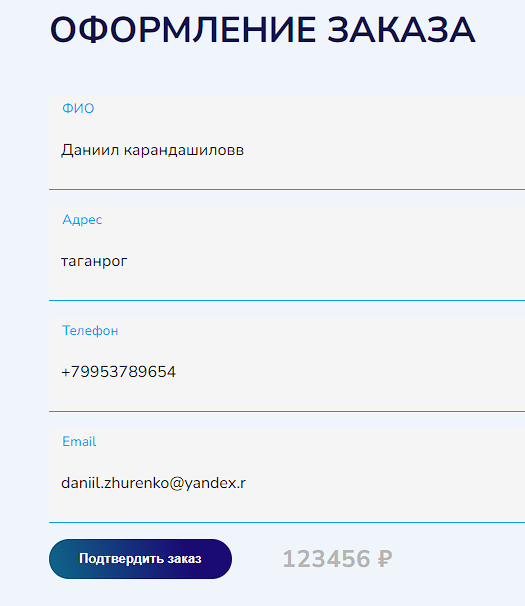


Рисунок 1.6 – Оформление заказа

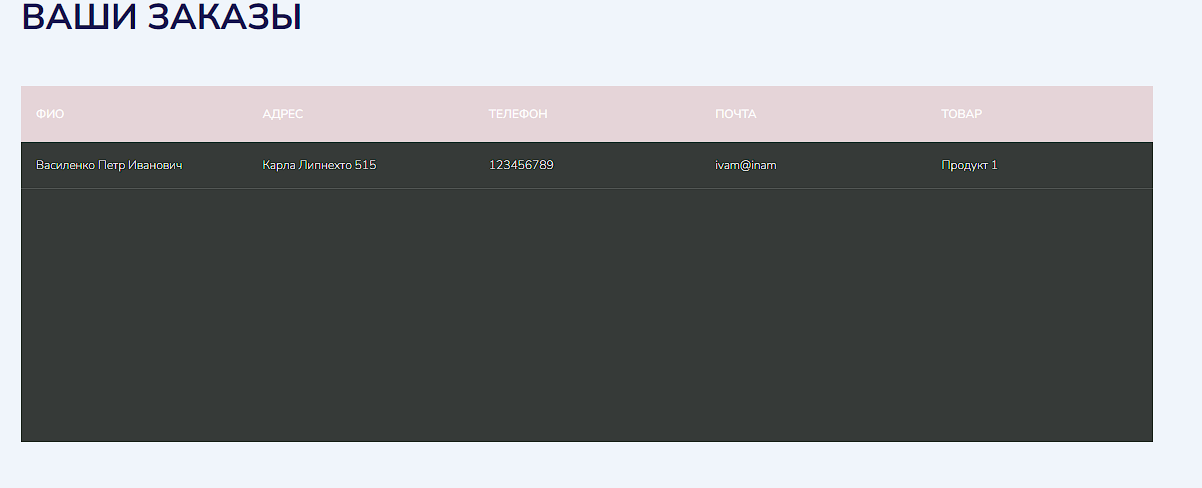


Рисунок 1.7 – Новый заказ

1. Для проверки возможности изменения товара во вкладке «Категория товаров» войдем под логином редактора и изменим товар нажав на значок редактирования

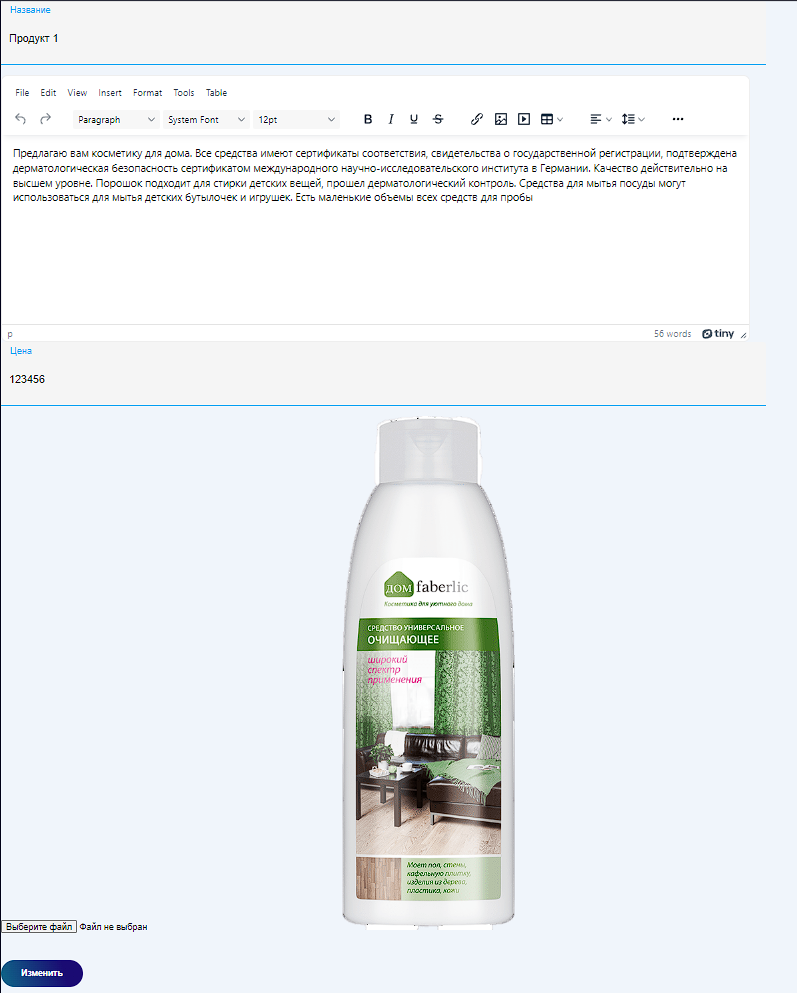


Рисунок 1.8 – Редактирование товара

1. Для проверки возможности добавления товара во вкладке «Категория товаров» войдем под логином редактора и добавим товар нажав на значок добавления товара

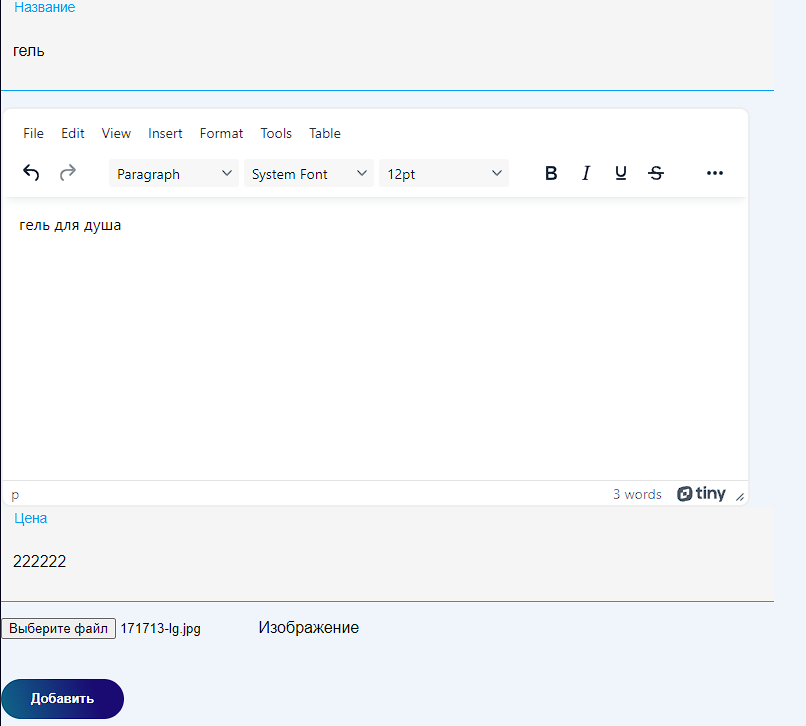


Рисунок 1.9 – Добавленный товар

После нажатия кнопки, данные на странице изменились. Продемонстрировано на рисунке 5.7.

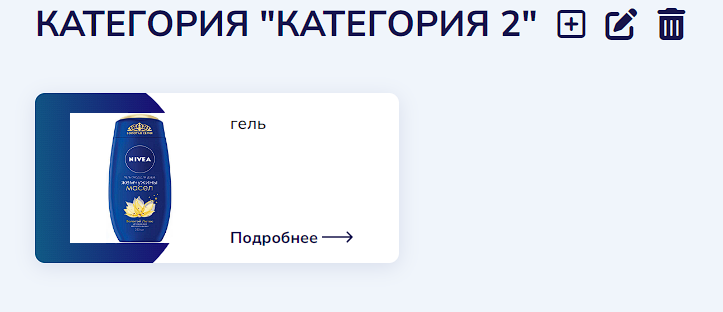
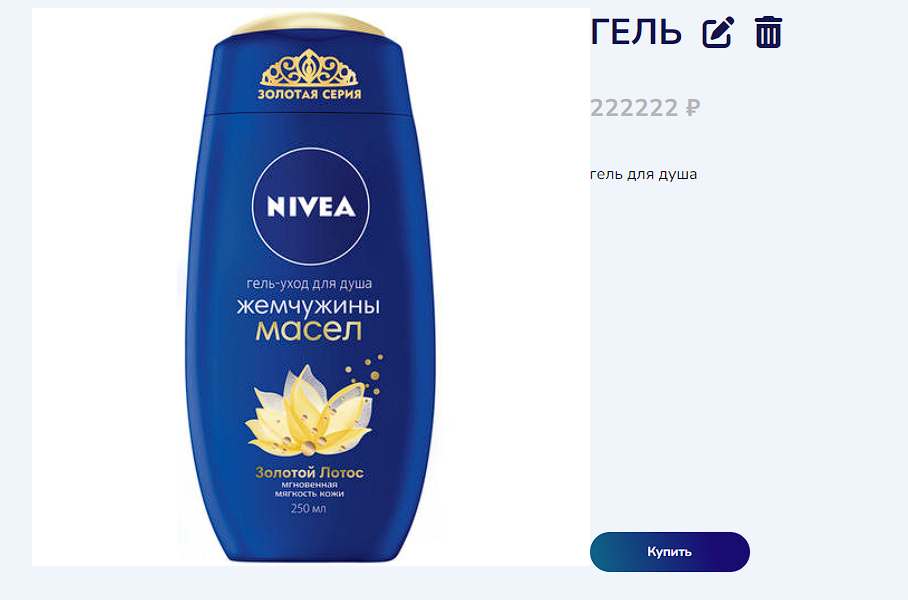


Рисунок 2.1 – Изменённая таблица нарядов

Для проверки удаления данных со страницы нажмём на кнопку «Удалить» в соответствии с рисунком 2.2.



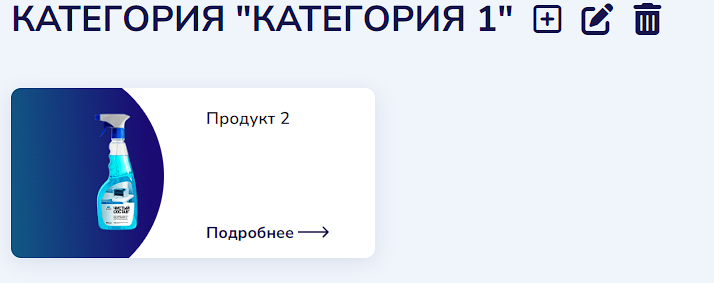


Рисунок 5.8 – страница с товарами после всех изменений

**Заключение**

В процессе разработки и работы над курсовым проектом была поставлена задача – реализовать программу для ведения веб-сайта “Чистый дом”. Данная задача была проанализирована, спроектирована, построена и выполнена. Были изучены и представлены примеры похожих программ, также был описан и спроектирован принцип работы программы и произведено тестирование исходной программы. Данный проект имеет недостатки, но также может быть доработан и улучшен.

**Список используемых источников**

1. <https://metanit.com/php/tutorial/>
2. <https://metanit.com/web/html5/>
3. <https://metanit.com/sql/>
4. <https://metanit.com/sql/mysql/>

**Приложение Б. Результаты выполнения программы**

При входе на сайт нас встречает окно авторизации, предназначенное для Покупателя сайта в соответствии с рисунком 2.3.

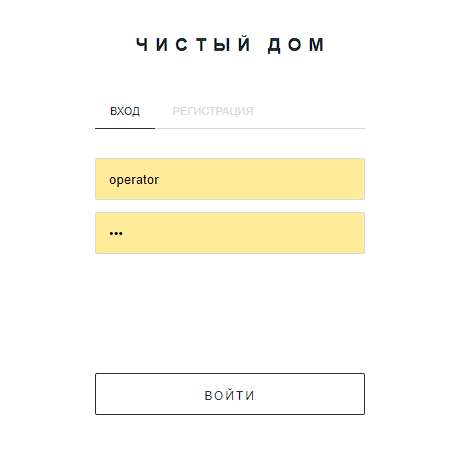
****

Рисунок 2.3 – Окно авторизации

После ввода логина и пароля нужно нажать кнопку «Войти», клиента переносит на страницу с товарами в соответствии с рисунком 2.4

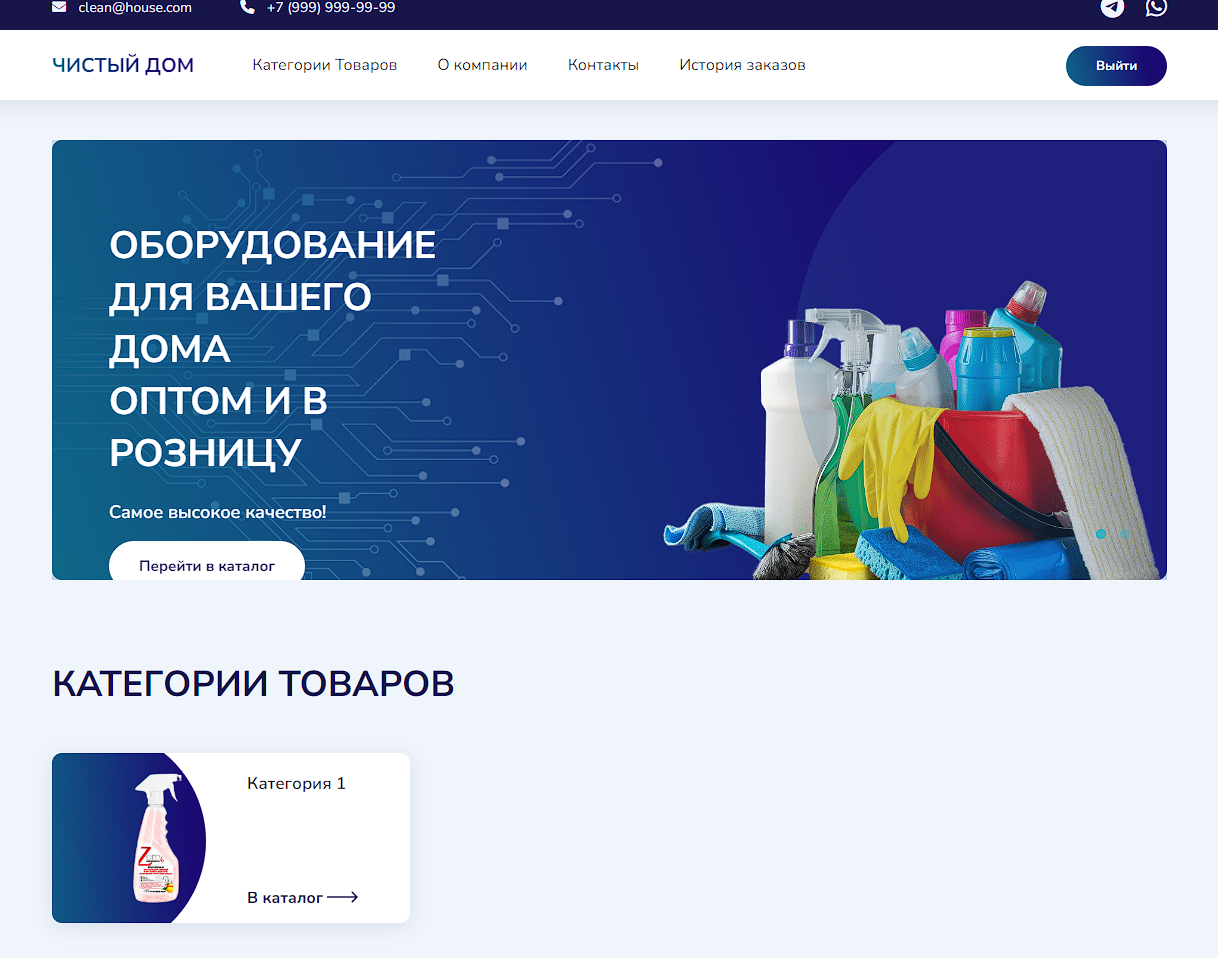


Рисунок 2.4- Страница с товарами

Далее есть возможность оформить заказ товара на вкладке «Купить» в соответствии с рисунком 2.5

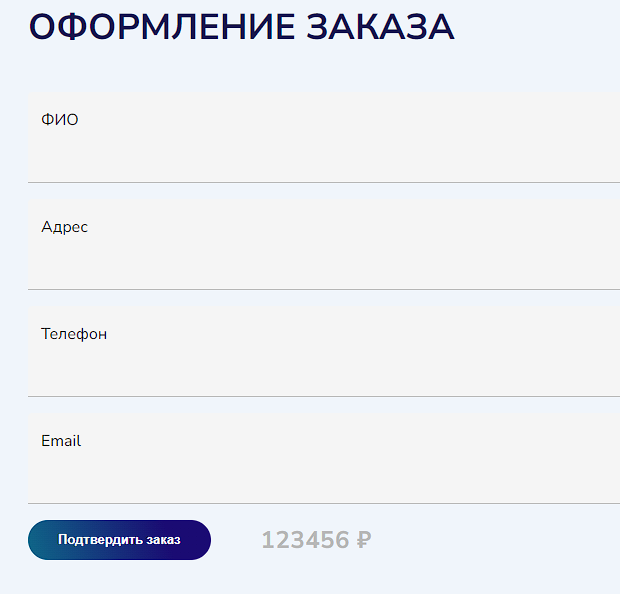


Рисунок 2.4- Страница оформления заказа