СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc103723356)

[1 ОБЩАЯ ЧАСТЬ 8](#_Toc103723357)

[1.1 Цель разработки 8](#_Toc103723358)

[1.2 Анализ использования разработки 9](#_Toc103723359)

[1.3 Анализ методов решения 9](#_Toc103723360)

[1.4 Обзор средств программирования 9](#_Toc103723361)

[1.5 Описание языка программирования 17](#_Toc103723362)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 21](#_Toc103723363)

[2.1 Постановка задачи 21](#_Toc103723364)

[2.2 Требования к входным и выходным данным 26](#_Toc103723365)

[2.3 Требования к составу и параметрам технических средств 28](#_Toc103723366)

[2.4 Описание алгоритма 29](#_Toc103723367)

[2.5 Структурная схема 30](#_Toc103723368)

[2.6 Текст программы с описанием 32](#_Toc103723369)

[2.7 Описание процесса отладки 37](#_Toc103723370)

[2.8 Описание результатов решения и их оценка 38](#_Toc103723371)

[3 ОРГАНИЗАЦИЯ ПРОИЗВОДСТВА 39](#_Toc103723372)

[3.1 Описание рабочего места на вычислительном центре 39](#_Toc103723373)

[3.2 Инструкция пользователю 44](#_Toc103723374)

[4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 47](#_Toc103723375)

[4.1 Определение технико-экономических показателей проекта 47](#_Toc103723376)

[4.2 Определение фонда оплаты труда на разработку и комплексные испытания программной системы 47](#_Toc103723377)

[4.3 Структура договорной цены на программное обеспечение 47](#_Toc103723378)

[4.4 Оценка конкурентоспособности ПС в сравнении с аналогом. 47](#_Toc103723379)

[5 ОХРАНА ТРУДА 48](#_Toc103723380)

[5.1 Мероприятия по устранению причин травматизма и профзаболеваний 49](#_Toc103723381)

[5.2 Мероприятия по обеспечению противопожарной защиты 51](#_Toc103723382)

[5.3 Мероприятия по обеспечению электробезопасности при эксплуатации технических средств 53](#_Toc103723383)

[5.4 Требования к помещению машинного зала 54](#_Toc103723384)

[5.5 Мероприятия по охране окружающей среды 58](#_Toc103723385)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 62](#_Toc103723386)

# ВВЕДЕНИЕ

Интернет стал для общества второй реальностью, мы предпочитаем его и для общения, и для творчества, и для покупок. Если раньше в интернете можно было заказать только некоторые виды продукции, то сейчас там есть всё от одежды до путевок на море. По последним данным аудитория в интернете стремительно растет, а продажи через интернет в крупных городах, достигают до 25%, при этом специалисты подчеркивают тенденцию к росту продаж именно через интернет. Интернет магазин является современным торговым каналом. С помощью интернет магазина, мы имеем возможность продавать свои товары или услуги огромной аудитории, использующей доступ в Интернет. Интернет магазин для покупателя это: экономия времени, денег и сил. Именно поэтому, по статистике, все больше и больше людей в России совершает свои покупки через интернет магазин.

Одним из интересных вопросов по этой теме является актуальность создания интернет магазина мебели. С одной стороны, мебель - это покупка ни на один год, поэтому важно не совершить ошибку при ее выборе, но с другой - часто выбор мебели в интернете гораздо больше, чем в магазинах, а цены намного меньше, это и делает заказы в интернете такими популярными.

Создать интернет магазин с привлекательным дизайном важно. Но намного важнее удобный интерфейс и комфортность работы посетителя с сайтом. Возможность сообщать о предстоящих акциях, скидках и других мероприятиях, проводить маркетинговые исследования, разрабатывать и выводить на электронный рынок продукции новые решения, уникальные торговые предложения и многое другое - неоспоримые качества интернет магазина. Кроме этого, создать интернет магазин - это еще и обозначить источник актуальной информации об ассортименте товара, его наличии. Зайти на сайт гораздо проще, чем звонить или, тем более ехать в физический магазин (если он конечно есть). Диалоги покупателей и представителей фирмы реализовались благодаря электронному маркетингу. Быстрота и легкость оформления заказов с использованием личного кабинета клиента, возможность создания целых информационных порталов расширяет возможности бизнеса как никогда раньше.

В прошлом к недостаткам покупок в интернете относили невозможность возврата и риск обмана, но со временем системы продаж совершенствуются, появляются всё новые возможности защиты, выбора и заказа. Жизнь людей меняется, мы все больше ценим время, а интернет - это прекрасная возможность его сэкономить. Интернет магазины мебели предлагают огромный выбор вариантов на многих сайтах. Мы можем в 3D формате посмотреть на диван, все размеры и цветовые возможности есть тут же, мы выбираем и сразу видим, каким будет тот или иной продукт. Важно, чтобы магазин был проверенным и работал на мебельном рынке достаточное время, ведь это показатель надежности, а значит - наша защита.

Актуальность заказа мебели в интернет магазине объясняется тем, что это удобно, практично и выгодно. Нам не нужно ходить по бесконечным магазинам и беспокоиться о доставке, достаточно выбрать понравившуюся модель и указать адрес. Мы не переплачиваем за аренду магазинами торговой площади, а значит, прилично экономим. Если мебель нам не понравилась при ближайшем рассмотрении, мы можем ее не покупать, сейчас большинство сервисов предоставляет такую возможность. Как видим, интернет магазины экономят наше время, а оно, как известно, бесценно.

## 1 ОБЩАЯ ЧАСТЬ

### 1.1 Цель разработки

Целью разработки является получение полноценного информационного ресурса, позволяющего организовать работу с информацией о мебели, заказах и отзывах о ней, с использованием гипертекстовых технологий.

Для достижения быстрого доступа к интересующей информации, а также её демонстрации конечному пользователю предусмотрен графический интерфейс с подсказками

### 1.2 Анализ использования разработки

Разработанный интернет-магазин мебели может быть использован небольшими компаниями для продажи своего товара; для хранения и обработки сведений о продажах, имеющемся товаре, а также для предоставления информации покупателю о мебели и возможности приобретения той, что ему понравилась, ее оценки и комментирования. Обработанные сведения о продажах могут быть использованы для оценки спроса на тот или иной товар, а история просмотра и заказов товара клиентом для персонализации предложения.

Пользователями сайта выступают администратор, главный бухгалтер, менеджер, обычный пользователь. Приобретение товаров от поставщиков осуществляется на основании договоров купли-продажи, в которых оговариваются условия поставки. Данные первичных документов по приходу товаров обобщаются в журнале поступления товаров, содержащем название приходного документа, его дату и номер, краткую характеристику документа, дату регистрации документа, сведения о поступивших товарах. Оформление и учет реализации товаров зависят от способа расчета за приобретаемые товары между покупателем и продавцом. Товары реализуются за наличный и безналичный расчет. Менеджер ведет журнал учета отпуска товаров, где указывается: номер по порядку, дата отпуска, наименование товаров, количество единиц и сумма отпуска, фамилия, инициалы и подпись лица, выдавшего товар. Данные первичных документов фиксируются в карточках учета, которые, исполняют роль регистров складского и бухгалтерского учета. Бухгалтер не реже одного раза в неделю осуществляет проверку записей в книгах или карточках. Основанием для отражения в бухгалтерском учете операций по поступлению и выбытию товаров являются товарные отчеты с приложенными к ним документами.

### 1.3 Анализ методов решения

Ручной метод – метод решения задачи, который предусматривает полностью самостоятельное написание программного кода без использования более удобных, вспомогательных средств и быстрых систем подсказок.

Механизированный метод – метод решения задачи, который предусматривает написания программного кода с использованием вспомогательных средств, систем подсказок и дописывание кода.

Частично автоматизированный метод – метод решения задачи, который предусматривает не полностью само написания кода, а использование уже готовых элементов и шаблонов, но некоторые вещи приходится делать самим.

Полностью автоматизированный метод – метод решения задачи, который предусматривает полностью использование готовых шаблонов и элементов, а также предоставление уже созданного администрирования программного продукта.

В последнее время появилось много возможностей создать качественный сайт с использованием так называемых CMS. Самые популярные из них - Wordpress, Drupal и так далее...

К плюсам CMS можно отнести:

* Лёгкость создания – для написания сайта на CMS не нужно быть программистом, это сможет сделать и обычный пользователь;
* Скорость создания – для создания сайта на CMS потребуется всего лишь несколько часов или дней, в зависимости от сложности сайта, не потребуется трата времени на составление архитектуры сайта и планирования базы данных;
* Большое количество расширений и плагинов – например, чтобы прикрутить к сайту какую-нибудь галерею фотографий, достаточно всего лишь найти данный плагин в интернете;
* Дешевизна – для создания сайта на CMS, не нужно платить большие деньги, так как система довольно проста, чтобы всё сделать самостоятельно;
* Популярность – при возникновении трудности, можно легко найти людей, которые уже встречались с данной проблемой и найдётся много разных способов решения проблемы.

Минусами CMS является:

* Сложность понятия кода CMS – при возникновении нетипичной задачи, требующая специализированной работы, можно зайти в тупик, так как очень сложно разбираться во внутреннем коде CMS;
* Очень низкая безопасность сайта – поскольку CMS очень популярны, их внутреннюю архитектуру знают все, это значит, что многие люди видят, как сделан сайт, так же есть много статей в интернете, как взломать те или иные CMS;
* Лёгкая порча сайта – поскольку у CMS постоянно выходят новые версии, то и версии плагинов должны соответствовать версии CMS, не редко ломаются сайты из-за не правильной версии плагина или расширения, и найти причину поломки может быть очень сложно;
* Сложность продвижение сайта в топы – поисковые системы видят, что сайт сделан с помощью CMS, и поэтому очень сложно подняться в топ, ведь серьёзная компания, которая заботится о благополучии своего бизнеса, не пожалеет денег на хорошего программиста.

Плюсы ручного метода:

* Уникальность сайта – о том, как устроен сайт знает только разработчик. Что бы взломать такой сайт, нужны хорошие специалисты;
* Легкость продвижения сайта в топы – продвижение сайта проходит гораздо быстрее, так как поисковик видит уникальность сайта;
* Легкость оптимизации под свои нужды – если нужно будет что-то добавить на сайт, то это не составит труда, так как не будет никаких конфликтов с плагинами, никаких споров с версиями и т.д.

Минусы ручного метода:

* Требование знаний языков программирования – создание сайтов с нуля подразумевает освоение языков программирования, таких как PHP, JavaScript, HTML, CSS, MySQL;
* Большие затраты времени на создание – создание хорошего, качественного сайта это работа не на один день, и обычно занимает большой промежуток времени, так как нужно построить план сайта, придумать дизайн и написать сам код с нуля:
* Дороговизна – если не самому создавать сайт, а заказывать его у специалиста, то потребуется отдать соответствующую сумму, так как хороший, качественный сайт, написанный с нуля, это не дешёвое удовольствие.

Для разработки интернет-магазина был выбран ручной метод, так как он намного безопаснее, чем сайт на CMS, а также легко оптимизировать и исправлять код, написанный собственноручно.

Для моделирования структуры базы данных применялись система ERWin и методы IDEF, позволяющие в автоматизированном режиме проектировать структуры базы данных.

### 1.4 Обзор средств программирования

**Node.js и PHP**

PHP — был создан в 1994 году Расмусом Лердфордом. Он создал программную оболочку (интерпретатор), которая устанавливается в качестве модуля для веб-сервера Apache или Nginx. Изначально разрабатывался как препроцессор гипертекстовых страниц, поэтому PHP может быть легко интегрирован в HTML код, однако, такой подход сейчас не является хорошей практикой, но все же для новичков такой подход был очевиден. Это способствовало популярности языка, поэтому 80% сайтов в интернете написаны на PHP, они же, в частности, работают под управлением WordPress CMS (20% сайтов в интернете).

Node.js был создан в 2009 году, Райаном Далем. Он создал программную платформу, основанную на JavaScript движке V8 от Google. Необычно то, что платформа имеет встроенные библиотеки для обработки запросов и ответов, тем самым вам не нужно использовать сторонний веб-сервер и какие-либо другие зависимости. Node.js набирает обороты и он используется такими компаниями, как: Microsoft, Yahoo, LinkedIn и PayPal.

**Сравнение порога вхождения PHP и Node.js**

Как быстро мы можем написать «Hello, world» страницу на том или ином языке программирования?

Скорость построения «Hello World» веб-страницы на PHP:

<?php

echo 'Hello World!';

?>

Этот код может быть написан где угодно, как правило, код пишут в файлах, оканчивающихся на расширение «.php». Если вы написали этот код в файле index.php и запустили на локальном сервере через порт 8000 (на своем компьютере, а не на реальном хостинге), то его отображение будет доступно по адресу: http://localhost:8000

Однако, используя готовый интерпретатор PHP со встроенным сервером довольно ненадежное занятие. Лучше использовать готовые решения, сборки Apache — XAMPP или виртуальные OS (Vagrant). Кстати, свой файл вы сможете загрузить на любой хостинг интернета.

Установить платформу Node.js проще простого, если вы работаете на unix-подобных системах, вы сможете это сделать и при помощи менеджеров пакетов. Давайте создадим страницу index.js:

const http = require('http');

const hostname = 'localhost';

const port = 8000;

http.createServer((req, res) => {

res.writeHead(200, { 'Content-Type': 'text/plain' });

res.end('Hello World\n');

}).listen(port, hostname, () => {

console.log(`Server running at http://${hostname}:${port}/`);

});

Мы повторили аналогичные действия, код на JavaScript, как правило, пишут в файлах, оканчивающихся на расширение «.js». Если вы написали этот код в файле index.js и запустили на локальном сервере через порт 8000 (на своем компьютере, а не на реальном хостинге), то его отображение также будет доступно по адресу (при условии, что php теперь работает на другом порту): http://localhost:8000

Теперь давайте оценим код, даже если вы хорошо знаете клиентский JavaScript, вам придется думать и разбираться с тем, что же здесь написано. Вы должны понимать замыкания и callback-функции, промисы, программирование на стороне Node.js требует некоторой сноровки.

PHP концептуально проще и выигрывает в этом плане. Хотя PHP имеет несколько программных зависимостей, однако, PHP менее сложный для изучения.

**Помощь и поддержка**

Сложно добиться большого успеха на своих знаниях без изучения каких-либо курсов, современных практик, чтобы развиваться, вам нужно задавать вопросы на форумах, таких как — StackOverflow. PHP выигрывает этот раунд легко, он имеет большое руководство на сайте php.net и двадцать лет часто встречаемых вопросов. Что бы вы ни делали, кто-то сталкивался с этой проблемой и пытался ее решить за вас.

У Node.js есть хорошая документация, но технология довольно молода, поэтому и ответов на форумах, будет значительно меньше.

**Инструменты разработчика**

Обе технологии имеют хороший выбор редакторов, интегрированных сред разработки, отладчиков, валидаторов и других инструментов. Тут можно дать ничью, но все же Node.js имеет прекрасный инструмент, npm — менеджер пакетов, c его помощью можно управлять модулями и зависимостями.

У PHP есть свой менеджер пакетов, разработанный под влиянием npm — Composer. Однако, ежели npm встроен по умолчанию, тот composer придется встраивать самостоятельно. Благодяря npm широкое распространение получили Gulp, Grunt — системы сборки front-end проектов.

**Среда разработки**

Где могут быть использованы эти технологии. Как их развернуть? Какими платформами поддерживаются? Веб-разработчикам часто нужно создавать приложения, которые только относятся к web, например, разработка онлайн-сервиса, сценарии преобразования данных и т.д.

На PHP вы можете разрабывать десктопные приложения или консольные утилиты, но в основном PHP нужен на стороне сервера и редко выходит за пределы этой границы.

Несколько лет назад, JavaScript использовался исключительно для браузера. С приходом Node.js вы можете писать десктопные и мобильные приложения, а также можно программировать и микроконтроллеры. Node.js расширил границы JavaScript.

**Производительность**

PHP не сутулится и есть реальные проекты, и варианты, которые позволяют PHP работать быстрее. Даже самый требовательный разработчик PHP редко беспокоится о скорости, но производительность Node.js, как правило, лучше. Конечно, производительность в значительной степени является следствием опыта и командной разработки, однако, Node.js имеет несколько преимуществ:

1. Меньше зависимостей

Все запросы к приложению PHP должны быть направлены на веб-сервер, который запускает интерпретатор PHP, который обрабатывает код и отдает его. Node.js не нужно так много зависимостей, и, хотя вы почти наверняка используете фреймворк на сервере, такой как express, он совсем легкий и управляет частью вашего приложения.

2. Быстрый интерпретатор

Node.js меньше и проворнее, чем РНР. Это связано с наследием Google, который сделал огромный вклад в производительность JavaScript движка — V8.

3. Приложения работают постоянно

РНР выполняет обычную клиент-серверную модель. Каждый запрос страницы инициирует приложение, загрузка параметров подключения к базе данных, извлечение информации и отображения HTML-кода. В Node.js приложение постоянно работает и его нужно инициализировать только один раз. Например, можно создать один объект подключения к базе данных, которая используется повторно при новом запросе. Правда, есть способы, реализации такого поведения и на PHP с использованием специальных систем, таких как memcached, но это не стандартная функция языка.

4. Событийный, неблокирующий поток ввода/вывода

PHP и большинство других серверных языков используют очевидную модель блокировки. Когда вы делаете запрос на извлечение информации из базы данных, запрос выполнит и завершит процесс, прежде чем перейдет к следующему оператору. В Node.js все иначе. В Node.js не нужно ждать. Вместо этого можно создать функцию обратного вызова, которая, прослушивая процесс, выполняется после того, как действие завершится.

Хоть Node.js приложения заметно быстрее, чем PHP тут есть и свои подводные камни.

Node.js / JavaScript работает в одном потоке, а большинство веб-серверов многопоточные и обрабатывают запросы параллельно. Написание асинхронного кода является сложным и несет свои собственные проблемы.

**Вывод**

Окончательный счет 2:3 в пользу Node.js. Node.js имеет крутую кривую обучения и не является идеальным для начинающих разработчиков, но он все же выигрывает. Просто. Если вы грамотный JavaScript-программист, который любит свой язык, Node.js не разочарует вас. Вы почувствуете себя свежее и получите освободительный опыт веб-разработки. Но не стоит сбрасывать PHP со счетов. PHP жив и есть мало оснований ставить подножку PHP, из-за того, что это выглядит модным. PHP проще изучить, вы сможете овладеть профессиональными приемами программирования, главное практика. PHP очень легко развернуть на сервере.

**Vue.js и React**

Vue.js и React – это JS-фреймворки с открытым исходным кодом для создания сложных пользовательских интерфейсов.

React обеспечивает повышенную гибкость благодаря использованию «компонентов» — коротких изолированных участков кода, которые помогают разработчикам создавать сложную логику и UI. React взаимодействует с HTML через virtual DOM — копию реального DOM-дерева элементов страницы. В копии все элементы представлены как объекты JavaScript. Эти элементы, вместе с декларативным стилем программирования React и односторонним связыванием данных, упрощают и ускоряют разработку.

Vue тоже использует virtual DOM, компоненты и, вдобавок, двустороннее связывание. Однако главная изюминка — его прогрессивная архитектура. Он разработан так, что позволяет разработчикам переносить существующие проекты во фреймворк, подключая фичи по одной, а не все сразу. Инфраструктура Vue.js позволяет создавать многофункциональные приложения. В зависимости от требований проекта вы можете использовать его как полноценный фреймворк или как легковесную библиотеку, а также как нечто среднее между ними.

**Сходство**

У Vue и React много общих черт и функций. Больше всего их роднит использование virtual DOM.

Кроме того, оба фреймворка:

* работают с любым существующим веб-приложением;
* помогают разработчикам легко и быстро приступить к работе;
* имеют основанную на компонентах архитектуру с управлением жизненными циклами компонентов;
* характеризуются повышенной гибкостью, скоростью и производительностью;
* имеют большие активные сообщества;
* предлагают широкий выбор библиотек и инструментов.

**Отличие Vue от React**

Между React.js и Vue.js есть несколько различий, которые определяют, для чего каждый из них лучше подходит. Основное заключается в методах, которые Vue и React используют для рендеринга контента в DOM. Vue использует HTML-шаблоны и JSX, а React — только JSX, который, по сути является расширением, позволяющим вставлять HTML непосредственно в код JS. JSX может упростить многие сложные задачи, но в то же время может усложнить простые.

Кроме того, ядро React’а предоставляет только компоненты, управление их состоянием и манипуляции с DOM. Остальное пишет и поддерживает сообщество. Опытные разработчики часто предпочитают такую свободу, но новичка может сбить с толку обилие сторонних библиотек и инструментов.

Хотя у Vue тоже немало решений, созданных сообществом, его часто используемые инструменты и такие сопутствующие библиотеки, как Vue-router, Vuex и Vue CLI, создает и поддерживает внутренняя команда разработчиков. Такая комбинация предварительно собранных и сторонних средств помогает удовлетворить потребности и желания, как начинающих, так и опытных разработчиков.

**Производительность**

Поскольку React и Vue имеют много одинаковых элементов, их производительность примерно равна. Оба фреймворка используют virtual DOM и ленивую загрузку для повышения производительности, и скорости загрузки страниц.

В некоторых ситуациях, правда, Vue явно превосходит React. Например, когда вы изменяете, состояние компонента React, остальные компоненты в его поддереве также будут повторно рендериться. Вы можете использовать неизменяемые структуры данных shouldComponentUpdate или PureComponent, чтобы предотвратить повторный рендеринг дочерних компонентов, однако это может повысить сложность и привести к несогласованности состояний DOM. Во Vue зависимости отслеживаются, и повторный рендеринг не нужен.

**Документация**

Здесь Vue — явный победитель. У него есть веб-сайт с подробным описанием на нескольких языках, а руководство и справку по методам API признают лучшими в отрасли. В документации — чёткие ответы на множество сложных вопросов. С другой стороны, так как сообщество Vue меньше сообщества React, может быть труднее получить ответы на вопросы, которые в документацию не вошли.

Документация React не дотягивает до уровня Vue, поэтому вам придется гораздо чаще обращаться к сообществу для решения проблем. Однако у React большое активное сообщество с огромным выбором учебных материалов.

**Безопасность**

И у Vue, и у React есть нюансы в вопросах безопасности, однако приложения Vue несколько проще защитить, чем React-приложения.

Хотя автоматически защитить приложения от XSS и других уязвимостей невозможно, разработчики Vue могут дезинфицировать HTML-код перед реализацией или использовать внешние библиотеки для защиты от атак. В тех случаях, когда вы знаете, что HTML безопасен, вы можете явно отображать содержимое HTML и защищать приложение как до, так и после рендеринга.

Безопасность React должен обеспечить разработчик, используя лучшие практики защиты от XSS-уязвимостей, атак на рендеринг на стороне сервера, SQL-инъекций и других угроз. Для защиты могут применяться eval, эксплуатация недостатков инъекций, безопасные приложения React Native.

Получается, что React прост в использовании, но для обеспечения безопасности React-приложений требуются глубокие знания и опыт.

**Вывод**

Если вам нужно сложное корпоративное приложение — выбирайте React. Он пользуется большей корпоративной поддержкой, большей популярностью среди разработчиков и имеет сообщество, способное ответить на любые вопросы. Также React-приложения легче масштабировать.

Если же вы ищете фреймворк для небольшого приложения или быстрой проверки идеи — посмотрите на Vue. Он пока не так широко поддерживается и используется, но его популярность растет, в первую очередь благодаря отличной документации, простоте использования и возможностям постепенного внедрения. Vue разрабатывается и поддерживается командой, имеет более широкий набор встроенных инструментов и решений. Что касается скорости разработки, с помощью Vue CLI 4 требуется всего несколько недель, чтобы настроить и выпустить продукт, готовый к продаже.

### 1.5 Описание языка программирования

**Среда программирования:**

**Visual Studio Code**

Visual Studio Code — это редактор исходного кода. Он имеет многоязычный интерфейс пользователя и поддерживает ряд языков программирования, подсветку синтаксиса, IntelliSense, рефакторинг, отладку, навигацию по коду, поддержку Git и другие возможности. Многие возможности Visual Studio Code недоступны через графический интерфейс, зачастую они используются через палитру команд или JSON-файлы (например, пользовательские настройки). Палитра команд представляет собой подобие командной строки, которая вызывается сочетанием клавиш.

VS Code также позволяет заменять кодовую страницу при сохранении документа, символы перевода строки и язык программирования текущего документа.

Также VS Code поддерживает редактирование и выполнение файлов типа «Блокнот Jupyter» (Jupyter Notebook (англ.)) напрямую «из коробки» без установки внешнего модуля в режиме визуального редактирования и в режиме редактирования исходного кода.

Посредством встроенного в продукт пользовательского интерфейса можно загрузить и установить несколько тысяч расширений только в категории «programming languages» (языки программирования).

Также расширения позволяют получить более удобный доступ к программам, таким как Docker, Git и другие. В расширениях можно найти линтеры кода, темы для редактора и поддержку синтаксиса отдельных языков.

**Языки программирования:**

**HTML**

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

Элементы HTML являются строительными блоками HTML страниц. С помощью HTML разные конструкции, изображения и другие объекты, такие как интерактивная веб-форма, могут быть встроены в отображаемую страницу. HTML предоставляет средства для создания заголовков, абзацев, списков, ссылок, цитат и других элементов. Элементы HTML выделяются тегами, записанными с использованием угловых скобок. Такие теги, как <img/> и <input/>, напрямую вводят контент на страницу. Другие теги, такие как <p>, окружают и оформляют текст внутри себя и могут включать другие теги в качестве подэлементов. Браузеры не отображают HTML-теги, но используют их для интерпретации содержимого страницы.

**CSS**

CSS — это каскадные таблицы стилей. Язык, который отвечает за описание внешнего вида HTML-документа. Подавляющее большинство современных веб-сайтов работают на основе связки HTML+CSS.

CSS позволяет отформатировать HTML контент на странице, сделать его более привлекательным для читателя. Изначально веб-разработчики использовали исключительно HTML — так было на заре развития интернет-технологий. С помощью разметки можно было выделить параграф, заголовок, изменить начертание текста.

Со временем этих возможностей стало мало — и появилась технология форматирования без изменения самого содержимого и структуры документа. CSS решил проблему избытка тегов форматирования, когда разные браузеры поддерживали разные теги. Их унификация и единая база упростили работу с веб-документами и облегчили жизнь веб-мастерам.

**JavaScript**

JavaScript – это мультипарадигменный язык программирования, который обычно применяется в качестве встраиваемого инструмента для программного доступа к различным объектам приложений. С точки зрения веб-разработки, без знаний этой технологии невозможно заниматься созданием современных интерактивных сайтов. Язык JS – это то, что «оживляет» разметку страниц (HTML) и пользовательский функционал (CMS) сайтов. С помощью этого языка реализуется возможность реакции страницы или отдельных ее элементов на действия посетителя. Сегодня JavaScript является базовым языком программирования для браузеров. Он полностью совместим с операционными системами Windows, Linux, Mac OS, а также всеми популярными мобильными платформами.

**Среда выполнения JavaScript**

**Node.js**

Node.js - это программная платформа, которая транслирует JavaScript в машинный код, исполняемый на стороне сервера. Таким образом, JavaScript можно использовать для создания серверной части.

Его можно использовать во многих местах, включая веб-приложения, многопоточные приложения, игровые серверы, браузеры и многое другое.

Node.js имеет многофункциональный инструмент, npm — менеджер пакетов, c его помощью можно управлять модулями и зависимостями.

В Node.js приложение постоянно работает, без разрыва связи, и его нужно инициализировать только один раз. Он также имеет различные библиотеки для доступа к SQL. Для подключения базы данных NoSQL, он использует JSON. Например, можно создать один объект подключения к базе данных, которая используется повторно при новом запросе.

**Программные платформы (Фреймворки)**

**Vue.js**

Vue — это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с современными инструментами и дополнительными библиотеками.

Vue virtual DOM, компоненты и, двустороннее связывание. Инфраструктура Vue.js позволяет создавать многофункциональные приложения. В зависимости от требований проекта вы можете использовать его как полноценный фреймворк или как легковесную библиотеку, а также как нечто среднее между ними.

Vue подходит для небольшого приложения или быстрой проверки идеи. Он пока не так широко поддерживается и используется, но его популярность растет, в первую очередь благодаря отличной документации, простоте использования и возможностям постепенного внедрения. Vue разрабатывается и поддерживается командой, имеет более широкий набор встроенных инструментов и решений.

**Express.js**

Express.js, фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом Vue.js, React или AngularJS для фронтенда.

**Библиотеки**

**Axios**

Axios — это JavaScript-библиотека позволяющая делать асинхронные HTTP-запросы. Она представляет собой HTTP-клиент, основанный на промисах и предназначенный для браузеров и для Node.js.

Использование Axios позволяет избежать написания больших объёмов шаблонного кода и сделать код чище и понятнее.

Axios умеет следить за ходом выгрузки данных. Это может стать решающим фактором выбора технологии для тех, кто разрабатывает приложение, позволяющее пользователям выгружать на сервер фотографии или видеофайлы.

**Sequelize**

Sequelize - это ORM (Object-Relational Mapping — объектно-реляционное отображение или преобразование) библиотека для приложений на Node.js, которая осуществляет сопоставление таблиц в базе данных и отношений между ними с классами. ORM хороши тем, что позволяют взаимодействовать с БД на языке приложения (JavaScript), т.е. без использования специально предназначенных для этого языков (SQL). Тем не менее, существуют ситуации, когда запрос к БД легче выполнить с помощью SQL (или можно выполнить только c помощью него). Sequelize может работать с рядом СУБД - MySQL, Postgres, MariaDB, SQLite, MS SQL Server.

**Vue-Router**

Vue-Router - это библиотека JavaScript, которая позволяет настроить маршрутизацию для одностраничных приложений (SPA), разработанных с помощью фреймворка Vue.js.

**CDN (Content Delivery Network)**

**Cloudinary**

Cloudinary - это географически распределённая сетевая инфраструктура, обеспечивающая быструю доставку изображений и видео пользователям веб-сервисов и сайтов.

Приложение поддерживает загрузку изображений, облачное хранилище, манипуляции с изображениями, оптимизацию изображений для Интернета и доставку. Cloudinary предлагает API-интерфейсы и гибкие возможности администратора для интеграции с новыми и существующими веб-приложениями и мобильными приложениями.

Cloudinary позволяет загружать и хранить неограниченное количество изображений конфиденциально и безопасно и включает автоматическое резервное копирование и исторические изменения. Предоставляет возможность управлять изображениями, используя широкий спектр опций, таких как применение эффектов, изменение размера, обрезка, обнаружение лиц, водяные знаки и многое другое.

Изображения передаются через всемирную сеть CDN Akamai для быстрой доставки, оптимизированной для любого устройства. Cloudinary выступает в качестве решения для управления цифровыми активами с медиа-библиотекой, которая использует RESTful-API и SDK для автоматизации управления изображениями. Также есть возможность получить доступ к расширенной аналитике и отчетам, чтобы оптимизировать и поднять производительность изображений.

## 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

### 2.1 Постановка задачи

**Требования к функциональным характеристикам:**

Сайт должен обеспечивать возможность выполнения следующих функций:

* Инициализация сайта (ввод информации о товарах)
* Редактирование данных
* Форма ввода данных
* Возможность регистрации / авторизации пользователя
* Возможность оценить товар и оставить комментарий
* Возможность заключения акта купли-продажи

**Требования к страницам пользователей сайта:**

**Требования к странице администратора:**

* Возможность добавлять / убирать работников магазина
* Возможность добавлять / убирать карточки товаров на сайт
* Возможность добавлять / убирать категории товаров
* Возможность удалять комментарии, нарушающие правила сайта

**Требования к странице главного бухгалтера:**

* Возможность подтверждать доставку товаров на склад от поставщика
* Возможность подтверждать получение оплаты за товар, изменяя его состояние в базе данных на «оплачено»
* Возможность подтверждать отправку товара со склада потребителю, изменяя состояние этого товара на «продано»

**Требования к странице менеджера:**

* Возможность просматривать статистику продаж и формировать отчет
* Возможность просматривать остаток товаров на складах и формировать заявки на заказ мебели от поставщиков
* Возможность оформлять покупку, непосредственно в магазине, для клиента, используя его логин учетной записи

**Требования к странице покупателя:**

* Возможность просматривать товары в корзине
* Возможность покупать мебель
* Возможность ставить оценку мебели и комментировать ее
* Возможность просматривать историю посещений карточек товаров

**Требования к базе данных**

**Функции базы данных:**

* Добавление / редактирование / удаление товара
* Добавление / редактирование / удаление категории товаров
* Регистрация / авторизация / удаление пользователей
* Главный администратор должен иметь возможность добавления / редактирования / удаления сотрудников
* Добавление / редактирование / удаление комментариев
* Пользователь должен иметь возможность добавления / редактирования / удаления товаров в корзине
* Возможность резервного копирования данных в БД

**Требования к таблицам базы данных**

Таблица **“Category” –** таблица, содержащая в себе информацию о каждой категории товара, имеющейся в магазине.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее категорию |
| name | string | нет | да | Название категории |
| ENname | string | нет | да | Название категории на английском |

Таблица **“Product” –** таблица, содержащая в себе информацию о товарах, имеющихся в магазине.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее товар |
| name | string | нет | нет | Название товара |
| description | text | да | нет | Описание товара |
| price | double | нет | нет | Цена товара |
| madeIn | string | нет | нет | Страна-производитель |
| materials | array(string) | нет | нет | Преобладающие материалы |
| dimensions | array(double)[3] | нет | нет | Габариты товара |
| weight | double | нет | нет | Вес товара |
| warranty | integer | да | нет | Гарантийный срок товара |
| count | integer | да | нет | Количество товара |
| categoryUUID | UUID | нет | нет | Уникальный идентификационный номер категории, к которой относится товар |

Таблица **“ProductPhoto” –** таблица, содержащая в себе информацию о каждой фотографии товара, имеющегося в магазине.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее фотографию |
| URL | string | нет | да | Ссылка на фотографию, расположенную на хостинге |
| publicID | string | нет | да | Уникальный идентификатор фотографии на хостинге |
| productUUID | UUID | нет | нет | Уникальное число, идентифицирующее товар, которому принадлежит фотография |

Таблица **“Order” –** таблица, содержащая в себе информацию о заказах на поставку мебели в магазин.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее заказ |
| createdAt | date | нет |  | Дата, когда заказ был сформирован |
| deliveryDate | date | нет | да | Дата, когда заказ был доставлен |
| count | integer | нет | нет | Количество доставленного товара |
| productUUID | UUID | нет | нет | Уникальное число, идентифицирующее товар, который был заказан |

Таблица **“User” –** таблица, содержащая в себе информацию о пользователях (покупателях), зарегистрировавшихся на сайте.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее пользователя |
| name | string | нет | нет | ФИО пользователя |
| login | string | нет | да | Логин пользователя |
| password | string | нет | нет | Пароль пользователя, хранится в виде хэша |
| roleUUID | UUID | нет | нет | Уникальное число, идентифицирующее роль пользователя (обычный человек, администратор, бухгалтер или менеджер) |
| email | string | нет | нет | E-mail пользователя |
| phone | string | да | нет | Телефон пользователя |
| photoURL | string | да | нет | Ссылка на фотографию профиля пользователя |

Таблица **“Token” –** таблица, содержащая в себе информацию о токенах – уникальных числах, принадлежащих конкретным пользователям и информирующих систему о том, что они вошли в личный кабинет на устройстве.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее токен |
| value | UUID | нет | да | Значение токена |
| userUUID | UUID | нет | нет | Уникальное число, идентифицирующее пользователя, которому принадлежит токен |

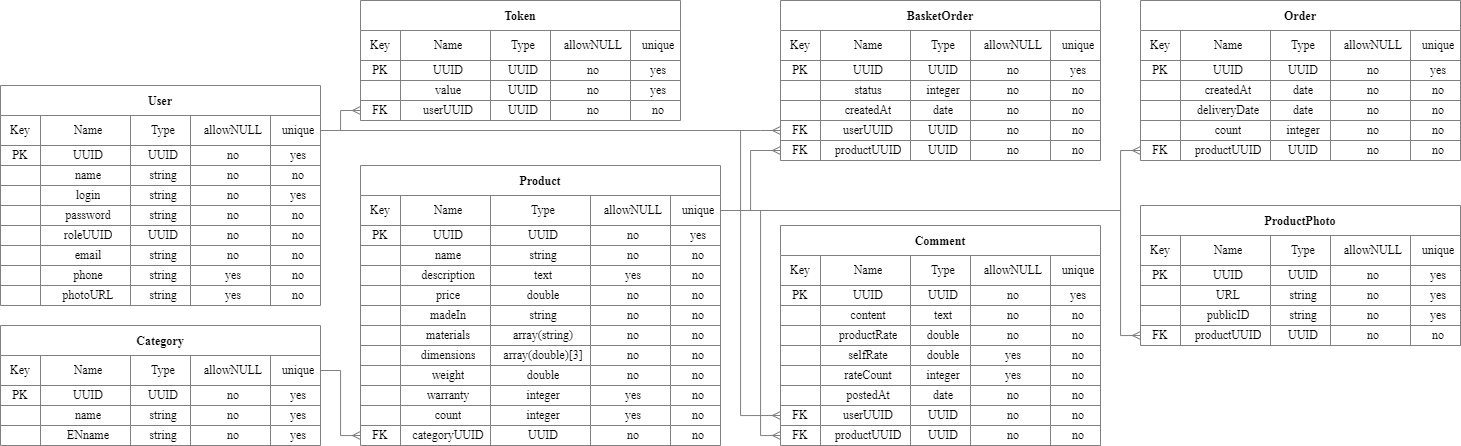
Таблица **“BasketOrder” –** таблица, содержащая в себе информацию о заказах пользователя, его избранном, а также информацию о когда-либо уже проданных товарах.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее запись в таблице |
| status | integer | нет | нет | Статус записи:  0 – товар в избранном  1 – товар в корзине  2 – товар ожидает отправки  3 – товар продан |
| createdAt | date | нет | нет | Дата, когда запись была создана |
| updatedAt | date | нет | нет | Дата, когда запись была обновлена |
| userUUID | UUID | нет | нет | Уникальное число, идентифицирующее пользователя, которому принадлежит запись |
| productUUID | UUID | нет | нет | Уникальное число, идентифицирующее товар, который был заказан |

Таблица **“Сomment” –** таблица, содержащая в себе информацию о комментариях, оставленных пользователями на странице товара.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название** | **Тип** | **Пустой** | **Уникальный** | **Описание** |
| UUID | UUID | нет | да | Уникальное число, идентифицирующее комментарий |
| content | text | нет | да | Сообщение комментария |
| productRate | double | нет | нет | Оценка товара |
| selfRate | double | да | нет | Оценка комментария |
| rateCount | integer | да | нет | Количество оценок комментария |
| postedAt | date | нет | нет | Дата публикации комментария |
| userUUID | UUID | нет | нет | Уникальное число, идентифицирующее пользователя, которому принадлежит комментарий |
| productUUID | UUID | нет | нет | Уникальное число, идентифицирующее товар, который был прокомментирован |

**Структура базы данных**

****

**Требования к надежности**

Требования к обеспечению надежного функционирования программы:

* предусмотреть разграничение доступа пользователей
* предусмотреть контроль вводимой информации при добавлении товара
* предусмотреть контроль вводимой информации при регистрации пользователя
* предусмотреть возможность восстановления пароля
* предусмотреть блокировку некорректных действий пользователя при работе с сайтом
* обеспечить целостность хранимой информации
* обеспечить безопасное использование личных данных пользователей (адресов, номеров карт, паролей)

### 2.2 Требования к входным и выходным данным

*Входные данные*

**Форма регистрации:**

* Логин
* Пароль
* Подтверждение пароля
* Имя пользователя
* Адрес электронной почты
* Телефон

**Форма входа в аккаунт**

* Логин
* Пароль

**Форма добавления категории товаров**

* Название категории

**Форма добавления товара**

* Название товара
* Описание товара
* Габариты товара
* Вес товара
* Цена товара
* Гарантия товара
* Количество товара на складе

*Выходные данные*

**Остаток товаров на складе**

**Проданные товары**

**Статистика продаж**

**Карточка товара**

* Название товара
* Описание товара
* Габариты товара
* Вес товара
* Цена товара
* Гарантия товара
* Оценка товара

**Комментарий пользователя**

* Имя пользователя
* Содержание комментария
* Оценка комментария

### 2.3 Требования к составу и параметрам технических средств

**Компьютер клиента**

**Минимальные**:

Операционная система:

* Windows: 8/10/11
* MacOS: 12.3.1
* Linux: Ubuntu 22.04/ Linux Mint 21/Debian 11.3 или др.

Браузер:

* Mozilla Firefox: версия 4.0 и выше
* Google Chrome: версия 9.0 и выше
* Safari: версия 8.0 и выше
* Opera: версия 12.0 и выше
* Internet Explorer: версия 11.0 и выше

Центральный процессор: любой, с частотой 1ГГц и выше

Графический процессор: любой с поддержкой DirectX9 и выше

Объем оперативной памяти: не менее 2ГБ

Свободное место на жёстком диске: 128Гб и более

Разрешение экрана: от 1280 x 800 и выше.

**Рекомендованные**:

Операционная система:

* Windows: 8/10/11
* MacOS: 12.3.1
* Linux: Ubuntu 22.04/ Linux Mint 21/Debian 11.3 или др.

Браузер:

* Mozilla Firefox: последняя актуальная версия
* Google Chrome: последняя актуальная версия
* Safari: последняя актуальная версия
* Opera: последняя актуальная версия

Центральный процессор: любой, с частотой 2ГГц и выше

Графический процессор: любой с поддержкой DirectX11 и выше

Объем оперативной памяти: не менее 4ГБ

Свободное место на жёстком диске: 256Гб и более

Рекомендуемое разрешение экрана: от 1920 x 1080 и выше.

**Сервер**

**Минимальные**:

* + Web сервер: Node.js версия 13.14.0 и выше
  + СУБД: PostgreSQL версия 12.0 и выше
  + Центральный процессор: любой, с частотой 2ГГц и выше
  + Объем оперативной памяти: 4ГБ
  + Свободное место на жёстком диске: 256Гб и более

**Рекомендованные**:

* Web сервер: Node.js версия 18.0.0 и выше
* СУБД: PostgreSQL версия 13.6 и выше
* Центральный процессор: любой, с частотой 4ГГц и выше
* Объем оперативной памяти: 8ГБ
* Свободное место на жёстком диске: 512Гб и более

### 2.4 Описание алгоритма

При переходе на сайт (<https://myroom-shop.herokuapp.com/>) происходит автоматическое перенаправление пользователя на страницу «О магазине» (<https://myroom-shop.herokuapp.com/about>), на которой находится основная информация о данном интернет ресурсе. Параллельно происходит запрос категорий товаров из базы данных и вывод их в левом боковом меню.

Затем пользователь может выбрать, на какую страницу он может перейти далее. При переходе по категории товара происходит запрос товаров данной категории. В основной части сайта появляется строка поиска конкретного товара, ниже появляется карусель популярных товаров текущей категории, под которой стопкой выводятся все товары необходимой категории, имеющиеся в магазине.

При клике по карточке товара пользователь переходит на страницу товара, на которой отображается вся информация о продукте. Параллельно подгружаются и выводятся комментарии и оценки других пользователей. Если пользователь хочет добавить товар в корзину или оценить/прокомментировать его, сначала происходит проверка, зарегистрирован ли он на сайте и вошел ли в свою учетную запись, если нет, ему предлагается это сделать. Если он соглашается, его перенаправляет на страницу аутентификации (<https://myroom-shop.herokuapp.com/login>).

При переходе на страницу аутентификации, отображается форма с полями ввода логина и пароля. При вводе всех необходимых данных происходит проверка на пустоту данных, после чего отсылается запрос на сервер, где проверяется, существует ли пользователь с таким логином, и, если да, проверяется соответствие введенного пароля. При возникновении ошибки, сервер отсылает соответствующий ответ и проблемное поле подсвечивается оранжевым цветом. При верно введённых данных сервер генерирует токен о отсылает его пользователю.

Со страницы аутентификации можно перейти на страницу регистрации (<https://myroom-shop.herokuapp.com/signup>). Регистрация проходит в два этапа: сначала пользователь вводит логин, пароль и подтверждение пароля. Проверяется, чтобы пароль и подтверждение совпадали, при нажатии кнопки «Далее» происходит запрос на сервер для проверки, не занят ли введенный пользователем логин, если нет, форма переходит на второй этап. Теперь пользователю нужно ввести свое имя, почту и телефон (пожеланию). При вводе почты и телефона происходит проверка корректности данных, и, если все хорошо, отсылается запрос на сервер для регистрации.

При входе в личный кабинет проверяется, является ли пользователь администратором, бухгалтером или менеджером, и, если да, происходит загрузка дополнительных страниц, недоступных обычному пользователю.

Администратор имеет возможность изменять категории, карточки товаров, рабочий персонал и комментарии, нарушающие правила сайта.

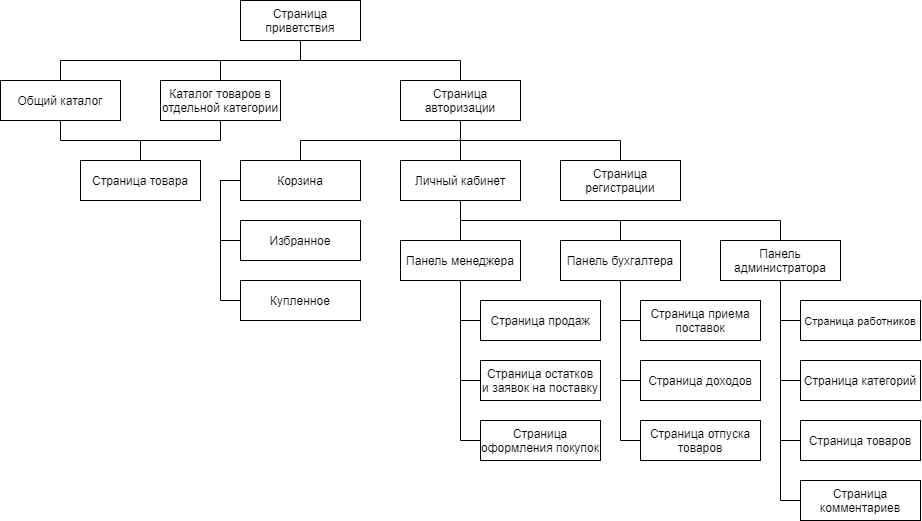
При переходе на страницу администрирования категорий происходит запрос всех существующих категорий в магазине и количество товаров, к ним принадлежащих. В верхней части страницы появляется строка создания новой категории, требующая ввести только название на русском языке. После нажатия на кнопку с изображением плюса, происходит запрос на сервер, где полученное название категории отправляется в TranslateAPI – сервис для перевода текста, и после получения ответа в виде английского названия категории, сервер обращается к базе данных, записывая в нее сведения о новой категории. После удачного создания категории она выводится в общем списке категорий, находящемся под строкой создания. Список представляет собой строки, содержащие русское название категории, которое можно изменить, количества товаров, принадлежащих данной категории и метки удаления. По нажатию на метки удаления, строки категорий помечаются, и в верхней строке появляется кнопка удаления выбранных категорий. При нажатии на нее, появляется диалоговое окно, с требованием подтвердить действие, и если администратор это делает, на сервер отправляется запрос на удаление категорий.

На странице администрирования товаров, перед тем, как что-то сделать, необходимо выбрать категорию, товары которой мы хотим изменить. При ее выборе происходит запрос товаров данной категории, и их вывод в виде строк, с возможностью развернуть и посмотреть подробности. Первая строка данного списка – пустая форма для добавления товара вручную. Каждая карточка товара внутри содержит также специальное меню с фотографиями продукта и пустой рамкой, по нажатию на которую появляется специальный виджет загрузки фотографий. В нем мы можем добавить товару его изображения, при их удачной загрузке на удаленный сервис сайт запоминает информацию об этих фотографиях и после закрытия виджета отправляет ее на сервер с целью добавления в базу данных. Карточки товаров имеют также метки удаления, функционал которых идентичен функционалу таких же меток на строках категорий. Для добавления нескольких товаров сразу, предусмотрена кнопка импорта из файла Excel.

При переходе на страницу администрирования комментариев происходит запрос всех комментариев, опубликованных на сайте, и их вывод в хронологическом порядке - от новых к старым. Администратор имеет возможность ознакомиться с содержанием комментария, и, если он нарушает правила сайта, удалить его.

При переходе на страницу администрирования персонала происходит запрос всех сотрудников магазина и вывод их краткой информации списком. Администратор может удалить страницу сотрудника.

### 2.5 Структурная схема



### 2.6 Текст программы с описанием

**Server.js**

Главный файл сервера, отвечает за начало его работы, основу конвейера обработки запросов.

const express = require('express') //подключаем фреймворк Express.js

const path = require('path') //подключаем встроенную библиотеку Path

const db = require("./dataBase/models") //подключаем объект базы данных

const categoriesData = require('./dataBase/originData/categories.json') //подключаем данные о категориях

const globalVars = require('./globalVars.json') //подключаем файл глобальных переменных

/\*подключаем приложение для управления обработки запросов RestAPI\*/

const mainAPI = require('./API/mainAPI')

const serverApp = express() //создаем главное приложение сервера

serverApp.locals.PORT = process.env.PORT || 3000 //определяем порт, на котором будет работать сервер

/\*указываем главному приложению сервера использовать парсер JSON для всех входящих запросов\*/

serverApp.use(express.json())

/\*указываем главному приложению сервера использовать парсер url для всех входящих POST запросов\*/

serverApp.use(express.urlencoded({ extended: true }))

/\*указываем главному приложению сервера обрабатывать входящие запросы RestAPI с помощью приложения обработки запросов RestAPI\*/

serverApp.use('/api', mainAPI)

/\*указываем главному приложению сервера искать и отправлять файлы по запросу типа

'/scripts/directory/files.type' из папки 'node\_modules', находящейся в папке проекта\*/

serverApp.use('/scripts', express.static(path.join(\_\_dirname, 'node\_modules')))

/\*указываем главному приложению сервера переписывать все остальные запросы не файлов

в запросы типа '/', что необходимо для реализации работы SPA\*/

serverApp.use((req, res, next) => {

if (req.url.indexOf('.') == -1) {

req.url = '/'

}

next()

})

/\*указываем главному приложению сервера искать и отправлять файлы по запросу типа

'/directory/files.type' из папки 'public', находящейся в папке проекта\*/

serverApp.use('/', express.static(path.join(\_\_dirname, 'public')))

/\*синхронизируем объект базы данных с реальной удаленной базой данных в интернете\*/

db.client.sync(/\*{force: true}\*/).then(() => {

console.log("Drop and re-sync db.") //сообщаем себе, что синхронизация прошла успешно

//db.Categories.bulkCreate(categoriesData)

globalVars.categories = categoriesData //кешируем информацию о категориях

/\*указываем главному приложению сервера начинать работать на определенном нами ранее порту\*/

serverApp.listen(serverApp.locals.PORT, () => {

console.log(`Server running on port ${serverApp.locals.PORT}`) //сообщаем себе, что сервер успешно запущен

})

})

**Models/index.js**

Файл, отвечающий за подключение к удаленной базе данных и предоставление инструментов для работы с ней.

const dbConfig = require("../dbConfig.js"); //подключаем реквизиты и конфигурацию удаленной базы данных

const { Sequelize, Op, DataTypes } = require("sequelize"); //подключаем библеотеку для работы с базой данных

/\*Создаем объект, через который будем взаимодествовать с БД

Передаем реквизиты БД и конфигурационные данные для подключения\*/

const client = new Sequelize(dbConfig.DB, dbConfig.USER, dbConfig.PASSWORD, {

host: dbConfig.HOST,

dialect: dbConfig.DIALECT,

dialectOptions: dbConfig.dialectOptions,

pool: dbConfig.pool,

});

const db = {}; //Создаем объект БД

/\*Сохраняем в объекте БД необходимы данные\*/

db.Sequelize = Sequelize;

db.Op = Op;

db.DataTypes = DataTypes;

db.client = client;

/\*Объявляем и сохраняем объекте БД наши таблицы\*/

db.Categories = require("./modelCategory.js")(client, Sequelize, DataTypes);

db.Products = require("./modelProduct.js")(client, Sequelize, DataTypes);

db.Users = require("./modelUser.js")(client, Sequelize, DataTypes);

/\*Устанавливаем связи между таблицами\*/

db.Categories.hasMany(db.Products, { as: "products" });

db.Products.belongsTo(db.Categories, {

foreignKey: "categoryId",

as: "category",

});

module.exports = db; //экспортируем объект базы данных

**index.html**

Главный файл, содержимое которого отображается у пользователя на экране. Благодаря фреймворку Vue.js позволяет отображать содержимое других страниц сайта и обновлять данные на них без перезагрузки основной страницы.

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--

Устанавливаем название для вкладки браузера

-->

<title>MyRoom Shop</title>

<!--

Устанавливаем иконку для вкладки браузера

-->

<link rel="shortcut icon" href="/icons/logo.svg" type="image/x-icon">

<!--

Подключаем стили для сайта

-->

<link rel="stylesheet" type="text/css" href="/css/main.css">

<link rel="stylesheet" type="text/css" href="/css/left\_menu.css">

<link rel="stylesheet" type="text/css" href="/css/right\_menu.css">

<link rel="stylesheet" type="text/css" href="/css/about.css" id="pageCSSLink">

<!--

Подключаем скрипты для работы сайта

-->

<script src='/scripts/vue/dist/vue.global.js'></script>

<script src='/scripts/vue-router/dist/vue-router.global.js'></script>

<script src='/scripts/axios/dist/axios.min.js'></script>

<script src='/scripts/validator/validator.min.js'></script>

<script src='/scripts/animejs/lib/anime.min.js'></script>

<script src='/scripts/vue-spinner/dist/vue-spinner.min.js'></script>

</head>

<body>

<!--

Создаем контейер для приложения Vue.js

-->

<div id="app">

<!--

Создаем левое меню

-->

<div id="left-menu">

<!--

Создаем логотип магазина, который одновременно является и ссылкой на страницу "О магазине"

-->

<router-link to="/about" id="logo">

<img src="/icons/logo.svg" alt="Логотип">

<span>MyRoom</span>

</router-link>

<!--

Выводим список категорий товаров

-->

<div id="category-list">

<span>Категории</span>

<ul>

<!--

Каждый пункт - это ссылка, ведущая в католог соответствующей категории

-->

<li v-for="item in categories" :key="item.uuid">

<router-link

:to="'/catalog/'+item.ENname"

id="category-button"

active-class="active">

{{item.name}}

</router-link>

</li>

</ul>

</div>

</div>

<!--

Главный блок, в нем будет динамически отображаться содержание текущей страницы, на которой находится пользователь

-->

<div id="main-content">

<router-view></router-view>

</div>

<!--

Создаем правое меню. Оно включает в себя шесть кнопок с минималистичными иконками, некоторые одновременно являются и ссылками на соответствующие им страницы. Кнопки правого меню: Профиль, Главный каталог, Корзина, Кнопка выхода из аккаунта, «О магазине», Кнопка "вверх"

-->

<div id="right-menu">

<div id="top">

<!--

Кнопка профиля

-->

<router-link

to="/profile"

id="profile\_icon"

class="icon"

active-class="active"

>

<img src="/icons/profile.svg" alt="Профиль">

</router-link>

<!--

Кнопка главного каталога

-->

<router-link

to="/catalog"

id="catalog\_icon"

class="icon"

active-class="active"

>

<img src="/icons/catalog.svg" alt="Каталог">

</router-link>

<!--

Кнопка корзины

-->

<router-link

to="/basket"

id="basket\_icon"

class="icon"

active-class="active"

>

<img src="/icons/basket.svg" alt="Корзина">

</router-link>

</div>

<div id="bottom">

<!--

Кнопка выхода из аккаунта

-->

<router-link

to="#"

id="logout\_icon"

class="icon"

@click="logoutReq"

>

<img src="/icons/logout.svg" alt="Выйти из аккаунта">

</router-link>

<!--

Кнопка "О магазине"

-->

<router-link

to="/about"

id="about\_icon"

class="icon"

active-class="active"

>

<img src="/icons/about.svg" alt="О магазине">

</router-link>

<!--

Кнопка "вверх"

-->

<router-link to="#" id="up\_icon" class="icon">

<img src="/icons/up.svg" alt="Вверх">

</router-link>

</div>

</div>

</div>

<!--

Подключаем главный файл скрипта страницы

-->

<script type="module" src='/js/main.js'></script>

</body>

</html>

### 2.7 Описание процесса отладки

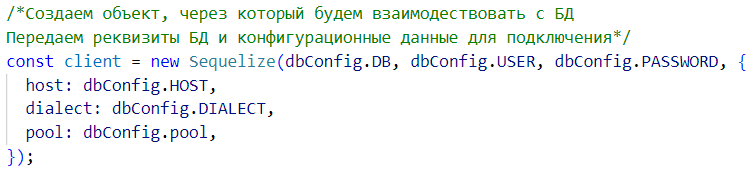


Рисунок 1 - Конфигурация подключения к базе данных

При данной конфигурации подключения к удаленной базе данных (Рисунок 1), происходила ошибка подключения – (Рисунок 2):

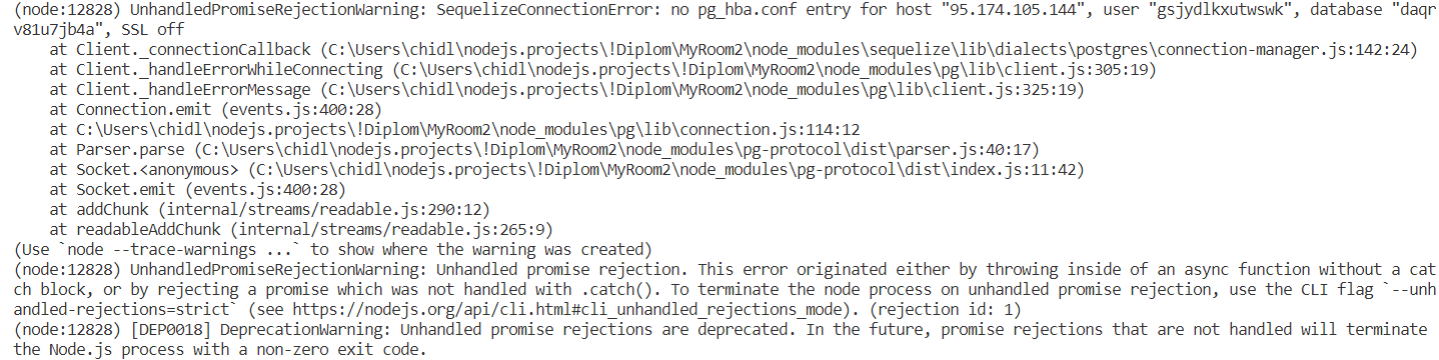
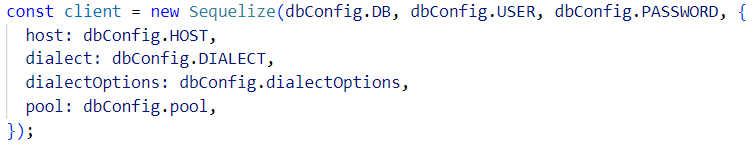
Решением послужило добавление в конфигурацию подключения дополнительного параметра (Рисунок 3) – dialectOptions (Рисунок 4)

Рисунок 2 - Ошибка подключения к базе данных



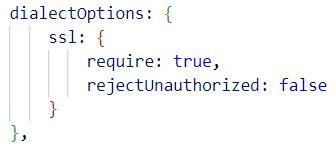


Рисунок – Корректная конфигурация подключения к базе данных

Рисунок – Значение параметра конфигурации



Рисунок – Ошибка при загрузке начальной страницы

При загрузке начальной страницы и попытке получить html код из объекта страниц сайта, появлялась ошибка (Рисунок 5). Причиной послужил способ получения этого объекта от сервера – асинхронный запрос с использованием библиотеки «Axios» (Рисунок 7). Для решения данной проблемы способ получения объекта страниц от сервера был изменён – теперь он запрашивался с использованием встроенной функции JavaScript XMLHttpRequest, которая делала синхронный GET-запрос на сервер (Рисунок 6).



Рисунок – Синхронный запрос, с использованием XMLHttpRequest

Рисунок – Асинхронный запрос, с использованием библиотеки «Axios»

### 2.8 Описание результатов решения и их оценка

Каждая новая часть начинается с нового листа, новый подраздел отделяется двумя пустыми строками.

## 3 ОРГАНИЗАЦИЯ ПРОИЗВОДСТВА

### 3.1 Описание рабочего места на вычислительном центре

**Вычислительный центр**

Широкие возможности ЭВМ, являющихся мощным средством разрешения научных проблем и экономического анализа, требуют соответствующего эффективного их использования. Наиболее эффективным способом максимального использования всех возможностей ЭВМ является создание ВЦ, в которых сосредоточены специалисты по целому ряду областей и которые оснащены электронными машинами различных классов. Необходимость такой организации объясняется тем, что, во-первых, эксплуатация машин требует постоянного совершенствования математических методов и приемов наиболее рационального решения и исследования различных физических проблем и инженерных задач. Во-вторых, эксплуатация машин, сосредоточенных в одном месте, является более эффективной как с точки зрения их наиболее рациональной загрузки и использования, так и с точки зрения их технического обслуживания. Кроме того, ВЦ быстрее и с наибольшим экономическим эффектом оправдывают большие материальные затраты на приобретение и эксплуатацию оборудования.

Одной из первых областей, в которых машины нашли широкое применение, была автоматизация сложных наук и инженерно-технических расчетов. Спецификой таких расчетов является большое количество операций, выполняемых машиной, и сравнительно малый объем вводимой информации по сравнению с объемом вычислений.

Компьютер содержит комплексы программных средств, называемых пакетами прикладных программ, призванные реализовать конкретные цели информационной технологии, а именно автоматизировать, т.е.:

облегчить труд оператора;

организовать хранение больших объемов информации;

обеспечить простоту, удобство доступа к информации, корректировки и т.д..

На организацию производства ВЦ существенно влияют конструкции и параметры основного и вспомогательного оборудования, которые должны отличать требованиям эргономики, оптимальному распределению функций в системе человек-машина; соответственно конструкция оборудования антропологическим и психофизиологическим данным организма работающего. Основы роста производительности является внедрение передового опыта, а также роста за счет сокращения времени набора исходных данных на клавиатуре, совмещение выполнения во времени нескольких элементов операции, рациональной подготовки и укладки документов и т.п.

**Рабочее место**

Планировкой рабочего места называют пространственное расположение основного и вспомогательного оборудования, оснастки и предметов труда, самого работающего, обеспечивающее рациональное выполнение трудовых движений и приемов, благоприятные и безопасные условия труда.

При организации рабочего места весьма важным фактором является рабочая поза работника, т.е. положение его корпуса, головы, рук и ног относительно орудий труда. Если работник работает сидя, ему необходимо обеспечить правильную и удобную посадку, что достигается устройством опоры для спины, рук, ног, правильной конструкцией сидения, способствующей равномерному распределению массы тела.

Все материальные элементы рабочего места разделяют на предметы постоянного и временного пользования и с учетом этого располагают в определенном порядке на местах постоянного хранения; это экономит трудовые движения и силы работающего. Инструмент, оснастка и предметы труда должны находиться на расстоянии 560-750 мм на уровне рук работника, тогда их использование не приводит к излишним движениям и наклонам. Важным элементом рациональной планировки рабочего места является учет индивидуальных антропометрических и психофизиологических данных работающего.

Рабочие места оборудуют соответствующей мебелью и инвентарем, отвечающим наиболее комфортабельным условиям работы и требованиям физиологии, психологии и эстетики.

Размещая производственные участки и оборудование, необходимо соблюдать следующие условия:

располагать оборудование и производственные участки в соответствии с последовательностью выполнения технологических операций;

выделять для размещения каждого структурного подразделения отдельную комнату;

производственные участки с большой численностью работающих располагать в светлых помещениях с естественным освещением;

создавать на работающих местах нормальные условия труда;

при расстановке оборудования соблюдать необходимые размеры промежутков между оборудованием, расстояний от стен, которые должны обеспечивать свободу передвижений людей, удобство выполнения работ и безопасность работающих; рабочие места операторов ПЭВМ, а также участка подготовки технических носителей информации следует располагать рядами; расположение мест может быть двухрядным, трехрядным, четырехрядным; расположение рядов может быть прямым и поперечным.

При реализации перечисленных условий необходимо экономно расходовать средства на приобретение техники и оборудование помещений.

Рассмотрим более подробно организацию рабочего места оператора с пользованием ПЭВМ.

Зрительные нагрузки связаны с воздействием на зрение дисплея. Чтобы условия труда оператора были благоприятными, снизилась нагрузка на зрение, видеотерминал должен соответствовать таким требованиям:

экран должен иметь антибликовое покрытие. Наилучшее сокращение отражений может быть достигнуто с помощью фильтров с просветленными поверхностями. Достаточные сокращения отражений достигаются также благодаря фильтрам из дымчатого стекла и матовым поверхностям экранов. Микроячеистые фильтры оправданы при ярком освещении тогда, когда при установке ВДТ невозможно учесть расположение осветительных приборов. Оптимальное подавление отражений может быть достигнуто в основном при строго вертикальном или слегка наклонном расположении дисплея. Самая верхняя используемая строка на экране не должна располагаться выше горизонтальной линии взгляда;

цвета знаков и фона должны быть согласованы между собой. При работе с текстовой информацией наиболее благоприятным для зрительной работы оператора является представление черных знаков на светлом фоне, так как при одинаковом контрасте разборчивость знаков на светлом фоне лучше, чем на темном;

для многоцветного отображения рекомендуется использовать одновременно максимум 6 цветов – пурпурный, голубой, синий, зеленый, желтый, красный, а также черный и белый, так как вероятность ошибки тем меньше, чем меньше цветов используется и чем больше разница между ними, а для одноцветного отображения – черный, белый, серый, желтый, оранжевый и зеленый. Красные и голубые цвета на границе видимого спектра применять нельзя;

необходимо регулярное тщательное обслуживание терминалов специалистами.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы.

Высота рабочей поверхности стола должна регулироваться в пределах 680-800 мм; при отсутствии такой возможности высота рабочей поверхности стола для ПЭВМ, на основании которых рассчитываются конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

Рабочий стол должен иметь пространство для постановки ног, которое составляет: высоту – не менее 600 мм, ширину – не менее 500 мм, глубину на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Конструкция рабочего стула должна поддерживать рациональную рабочую позу с ПЭВМ, позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения утомления.

Рабочий стул должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также расстоянию спинки от переднего края сиденья.

Поверхность сиденья, спинки и других элементов стула должна быть полумягкой с нескользящим, не электризующимся и воздухопроницаемым покрытием, обеспечивающим легкую очистку от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 º.

Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Рабочее место должно быть спланировано так, чтобы работать за компьютером было не только интересно, но и удобно. Если невозможно выделить для компьютера специальную комнату, то отдельный стол для него просто необходим. Кроме того, может понадобиться небольшой дополнительный столик или тумбочка для печатающего устройства. Нельзя устанавливать компьютер рядом с батареей центрального отопления.

Центр экрана монитора должен находится примерно на уровне глаз, расстояние между глазами и плоскостью экрана составлять не менее 40-50 см. Желательно, чтобы прямой солнечный свет не попадал на экран. По отношению к сидящему за столом окно, по возможности, должно быть слева или спереди. От яркого света следует защититься плотными шторами на окнах. Однако смотреть на экран монитора в полной темноте не рекомендуется, необходим дополнительный источник рассеянного света.

Описание рабочего места

* Процессор: Pentium(R) Dual - Core CPU E5200 2,50 ГГц
* Оперативная память: 2,00 ГБ
* Видеокарта: NVIDIA GeForсe 9500 GT
* Винчестер: 261,6 Гб
* Привод: OptiarcDVD-RWAD-7200A ATA Device

### 3.2 Инструкция пользователю

**Область применения**: интернет магазин «MyRoom» предназначен для продажи мебели и предоставления информации о ней покупателям.

**Возможности сайта**: регистрация, авторизация, просмотр и поиск товара, оплата товара в корзине (покупка), добавление отзыва, оценки товара.

**Уровень подготовки пользователя:** для работы с интернет магазином «MyRoom» пользователь должен обладать базовыми навыками работы на ПК, взаимодействия с графическим пользовательским интерфейсом ОС Windows (клавиатура, манипулятор типа «мышь», управление окнами, в т.ч. диалоговыми, и приложениями, ввод и редактирование информации) и взаимодействия с современными браузерами (Chrome, FireFox, Yandex и тд.).

**Назначение и условия применения**

**Виды деятельности и функции**

Функции, для автоматизации которых предназначен интернет магазин «MyRoom»:

* Формирование заказа пользователя в системе.
* Регистрация пользователя в системе.
* Сортировка товара.
* Обновление информации о наличии и ценах товаров.

**Условия применения**

Для корректной работы приложения рекомендуется:

* Подключение к интернету.
* Минимальное количество параллельно запущенных трудоёмких задач.
* Операционная система должна включать в себя русский язык.
* Операционная система должна быть от завода изготовителя устройства.

**Подготовка к работе**

Для начала работы с интернет магазином «MyRoom» необходимо:

* Запустить сайт <https://myroom-shop.herokuapp.com>
* Дать согласие на обработку cookie -файлов
* Пройти авторизацию (или регистрацию, если сайт посещается впервые)

**Описание операций**

Интернет магазин «MyRoom» выполняет следующие функции:

Не зависимо от того, в каком разделе сайта вы находитесь в данный момент, вы можете перейти на любую из основных страниц: «О нас» (Рисунок 8), общий каталог (Рисунок 10), каталог категории, профиль (Рисунок 9), корзина (после авторизации) (Рисунок 11). Для того чтобы сделать это, щелкните левой клавишей мыши по соответствующей кнопке.

********

Рисунок – кнопка профиля

Рисунок – кнопка главного каталога

Рисунок – кнопка корзины

Рисунок – кнопка «О нас»

**Страница «О нас».** На данной странице отображается основная информация о магазине, контактные данные, положение ближайшего магазина к покупателю.

**Страница общего каталога.** На данной странице представлены по несколько самых популярных товаров каждой категории, а также текущие скидки и акции. Чтобы перейти в каталог конкретной категории, нужно кликнуть на ее название. В верхней части страницы находится поисковая строка, позволяющая найти товар по названию, или его части. Чтобы это сделать, нужно кликнуть по текстовому полю, ввести часть названия товара и затем нажать на кнопку с изображением увеличительного стекла. Для каждой секции галереи карточек товара можно настроить сортировку (по умолчанию – от самого популярного до самого непопулярного). Чтобы перейти на страницу с подробным описанием товара, где можно тщательнее изучить сведения о нём, оценить его и оставить свой комментарий, нужно щелкнуть левой кнопкой мыши по названию этого товара на его карточке.

**Страница каталога категории.** На данной странице представлены все товары соответствующей категории. В верхней части страницы находится поисковая строка, позволяющая найти товар по названию, или его части. Чтобы это сделать, нужно кликнуть по текстовому полю, ввести часть названия товара и затем нажать на кнопку с изображением увеличительного стекла. Сортировка карточек товара по умолчанию – по алфавиту. Чтобы перейти на страницу с подробным описанием товара, где можно тщательнее изучить сведения о нём, оценить его и оставить свой комментарий, нужно щелкнуть левой кнопкой мыши по названию этого товара на его карточке.

**Страница профиля.** На данной странице до авторизации предоставляется выбор – пройти регистрацию на сайте, либо войти в свой личный кабинет. После авторизации на данной странице отображается основная информация о пользователе, доступная магазину, а также последние оставленные комментарии, оценки и просмотренные товары.

**Страница корзины.** На данной странице отображаются товары, которые пользователь собирается купить, их общая стоимость. В верхней части присутствует меню с категориями товаров, которые уже когда-то были куплены пользователем, находятся в процессе доставки или были возвращены. Данная страница доступна только после авторизации.

**Аварийные ситуации**

**Сбой подключения.** Для устранения данной проблемы следует проверить подключение к интернету. Если доступ в интернет имеется, то следует проверить разрешение на доступ к данному сайту. Для этого следует перейти в настройки брандмауэра операционной системы и проверить, нет ли запрета на подключение к данному сайту. Если доступ разрешен, то обратиться к системному администратору вашей организации, либо вашему провайдеру, если пользование сайтом происходит в домашних условиях.

**Ошибка регистрации.** Для устранения данной проблемы необходимо следовать указаниям в появившемся окошке, ниже формы регистрации. Проверить корректность написания своей электронной почты, корректность написания пароля и его подтверждения. В случае если все верно, обратиться в техподдержку.

**Ошибка отправки комментария.** Для устранения данной проблемы следует убедиться, что количество символов находится в пределах допустимого, и что ваш комментарий не нарушает правил сайта. В случае если все хорошо, обратиться в техподдержку.

## 4 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

### 4.1 Определение технико-экономических показателей проекта

**Расчет трудоемкости разработки программного продукта**

При создании программного продукта очень важно оценить его стоимость. Объем исходных текстов программы отражает трудоемкость и длительность разработки программного обеспечения. Объем программ в современных публикациях приводится в различных единицах, которые можно разделить на две группы:

* группа, характеризующая объем исходных текстов программ, которые разрабатываются и анализируются программистом (это символы в исходном тексте программы на любых языках программирования; лексемы, объединяющие группы символов, имеющих общее смысловое содержание в тексте программы; операторы языка программирования уровня ассемблера);
* группа, отражающая объем программы, размещаемой в ЭВМ (это байты, занятые текстом программы в памяти ЭВМ; слова памяти, обусловленные структурой данной реализующей ЭВМ, используемые для хранения исполняемой программы).

Базовым показателем для определения трудоемкости является либо выделение перечня работ по разработке программного продукта (способ №1), либо определение условного числа операторов в программе (способ №2).

В рамках дипломной работы был выбран способ №1 определения трудоемкости разработки программного продукта.

Интернет-магазин разработан программистом. Требования и критерии интернет-магазина сформулированы и согласованы с заказчиком.

Ниже приведен примерный перечень работ по разработке интернет-магазина:

* Определение целей и задач разработки.
* Разработка структуры интернет-магазина.
* Написание кода программы на языках JS, HTML, PHP.
* Наполнение контентом.
* Исправление ошибок.

Где – расчетная трудоемкость выполнения работы;

– минимальное время, необходимое для выполнения работы;

– максимальное время, необходимое для выполнения работы.

Минимальное и максимальное время работ рассчитывается разработчиком самостоятельно по уровню квалификации специалиста и его знаний в данной области, а также сложности выполняемой задачи.

Рассчитанные трудоемкости выполнения всех работ по созданию интернет-магазина представлены в табличном виде:

|  |  |  |  |
| --- | --- | --- | --- |
| **Таблица выполнения этапов работ по времени** | | | |
| **Этапы работ** | **Минимальное время (ч)** | **Максимальное время (ч)** | **Время ожидаемое (ч)** |
| Определение целей и задач разработки | 6 | 12 | 8,4 |
| Разработка структуры интернет-магазина | 4 | 7 | 5,2 |
| Написание кода программы на языках JS, HTML, PHP | 120 | 192 | 148,8 |
| Наполнение контентом | 7 | 18 | 11,4 |
| Исправление ошибок | 15 | 20 | 17 |
| Итого | 152 | 249 | 190,8 |

На разработку интернет-магазина требуется в среднем 191 рабочий час (примерно 8 дней), при минимальном времени в 152 часа и максимальном 249. При сопровождении интернет-магазина на период технического обслуживания и сопровождения, время выполнения работ может быть неограниченно, либо возрастать в значительном количестве, так как может существовать его дополнение, обновление, устранение неполадок в работе.

**Материальные затраты**

При разработке интернет-магазина использовались следующие материалы, представленные в таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| **Материальные затраты** | | | |
| **Материалы** | **Количество** | **Цена за 1ед. руб.** | **Сумма, руб.** |
| Node.js | 1,00 | 10 000,00 | 10 000,00 |
| Vue.js | 1,00 | 10 500,00 | 10 500,00 |
| Покупка домена | 1,00 | 9 000,00 | 9 000,00 |
| JS | 1,00 | 6 000,00 | 6 000,00 |
| Эл.энергия (кВт) | 150,00 | 5,70 | 855,00 |
| Бумага А4 | 3,00 | 750,00 | 2 250,00 |
| Хостинг сервера | 1,00 | 1 000,00 | 1 000,00 |
| Итого | – | – | 39 605,00 |

Для реализации проекта был куплен хостинг для размещения на нём интернет-магазина. Использована бумага формата А4 для описания и постановки задач заказчика. Затрата на язык JS, Vue.js и Node.js подразумевает курс обучения.

**Затраты на амортизацию**

Особенностью создания программы является износ оборудования и, как следствие, большие амортизационные отчисления. Амортизационное отчисление – это денежная оценка износа, как физического и морального, включается в себестоимость продукции. В условиях рынка величина амортизационного отчисления оказывает существенное влияние на экономику предприятия, то есть слишком высокая доля отчислений увеличивает издержки предприятия, а, следовательно, снижает конкурентоспособность и уменьшает размер прибыли.

Амортизационные отчисления рассчитывается по формуле.

где: Сп – балансовая стоимость компьютера (50000 руб.);

Н – норма амортизации 13%;

Fф – фактическое время работы оборудования;

Fоб – действительное время работы оборудования.

Действительное время работы оборудования находим по формуле:

где: q - число смен (1 смена);

К - коэффициент, учитывающий плановые простои (0,93);

F - годовой фонд рабочего времени (282 \* 8ч.).

Число рабочих смен при реализации полученного задания равнялось единице. Коэффициент простоя равен 0.93, так как учитываются характер работы и требования по нормализации труда.

ч.

руб.

Время работы оборудования 2 098,08 часа. Стоимость амортизационных отчислений равна 591 рубль 11 копеек.

### 4.2 Определение фонда оплаты труда на разработку и комплексные испытания программной системы

**Затраты на оплату труда**

Над проектом работает 1 человек, это программист-техник, он выполняет функции главного программиста и дизайнера.

Заработная плата работников рассчитывается по формуле:

где: Р – число работников определенной квалификации;

– расчетная трудоемкость выполнения работы;

Ом – месячный оклад, определяемый:

* в соответствии с минимальным размером оплаты труда в 2022 году в Ростовской области – 13890 руб., если программный продукт разрабатывается для потребностей бюджетного учреждения;
* в соответствии с примерной средней величиной оплаты труда (без учета конкретного направления деятельности программирования) в г. Таганроге в 2022 году для техника-программиста без опыта работы – 18 000 руб., если программный продукт разрабатывается для потребностей коммерческой организации;

ds – длительность смены (8 часов);

Др – среднее число рабочих дней (21).

Заработная плата работника начислена в размере 20 тысяч 442,86 рублей.

Полную заработную плату работника, участвующего в создании интернет-магазина определяем по формуле:

где: З – полная заработная плата работника участвующего в создании интернет-магазина;

ЗП – затраты на выплату заработной платы работника участвующего в создании интернет-магазина;

П – премия работника, участвующего в создании интернет-магазина (20% от уровня заработной платы).

Полная заработная плата работника за разработку программного продукта равна 24 тысячам 531 рублю 43 копейкам.

**Отчисления во внебюджетные фонды**

Отчисления на социальные нужды (во внебюджетные фонды) – это обязательные отчисления предприятий во внебюджетные фонды. Отчисления на социальные нужды включают прямые страховые выплаты:

* Пенсионный фонд РФ - 22%;
* Фонд социального страхования - 2,9%;
* Фонд обязательного медицинского страхования - 5,1%.

Общая ставка отчислений на социальные нужды составляет 30% от уровня заработной платы.

Расчет отчислений на социальные нужды производим по формуле:

где: ССН – общаяставка отчислений на социальные нужды;

З – полная заработная плата работника участвующего в создании интернет-магазина.

Таким образом, отчисления на социальные нужды составляют 7 тысяч 359 рублей 43 копейки.

**Прочие расходы**

В состав прочих расходов включены затраты на маркетинговые исследования, использования интернета, затраты, связанные с обслуживанием и организацией производства и прочие расходы. Прочие расходы принимаются в размере (15-30)% от уровня заработной платы.

Принимаем прочие расходы в размере 20% от уровня заработной платы до начисления премии, что равняется рублям 57 копейкам.

### 4.3 Структура договорной цены на программное обеспечение

**Составление сметы затрат на разработку программного продукта**

Смета затрат на создание интернет-магазина представлена в таблице:

|  |  |  |
| --- | --- | --- |
| **Смета затрат на создание интернет-магазина** | | |
| **Наименование** | **Сумма, руб.** | **Структура себестоимости, %** |
| 1. Материальные затраты (в том числе затраты на электроэнергию) | 39 605,00 | 52% |
| 2. Амортизационные отчисления | 591,11 | 1% |
| 3. Затраты на оплату труда | 24 531,43 | 32% |
| 4.Отчисления во внебюджетные фонды | 7 359,43 | 10% |
| 5. Прочие расходы | 4 088,57 | 5% |
| Итого | 76 175,54 | 100% |

|  |  |
| --- | --- |
| **Основные показатели при расчете цены программного продукта** | |
| **Наименование** | **Значение** |
| Уровень рентабельности | 0,25 |
| НДС | 20% |
| Планируемый размер прибыли | 19 043,89 руб. |
| Цена создания интернет-магазина | 95 219,43 руб. |
| Цена реализации проекта | 114 263,31 руб. |

Материальные расходы и затраты на оплату труда составляют 52% и 32% соответственно, что делает их основными частями затрат. Наименьший процент имеют амортизационные отчисления (1%). Суммируя оставшиеся расходы, итоговая сумма составит 76 175 рублей 54 копейки. Затраты средние, что говорит о необходимости повышенного вклада в программу со стороны заказчика. Добавив к ним процент НДС и планируемую прибыль, получим итоговую цену реализации проекта – 114 тысяч 263 рубля 31 копейка.

### 4.4 Оценка конкурентоспособности ПС в сравнении с аналогом.

Эксплуатационно-технический уровень (ЭТУ) разрабатываемого продукта–

## 5 ОХРАНА ТРУДА

Успех деятельности человека-оператора может быть обеспечен при оптимальной организации взаимодействия человека и машины, при соответствии технических характеристик оборудования физическими и интеллектуальными возможностями человека, при условии установления оптимальных темпов и режимов работы.

Комфортабельность труда и высокая производительность на рабочем месте оператора зависит от правильности выбора технического оборудования. Важнейшими характеристиками рабочего пространства являются зоны досягаемости. В машинном зале дисплеи и клавиатуры должны быть установлены в зонах, в которых возможны наиболее быстрые, точные, координированные и наименее утомительные движения.

При работе оператора в положении ”сидя” используются следующие параметры рабочего пространства:

ширина не менее - 700 мм;

глубина не менее - 400 мм;

высота рабочей поверхности стола над полом - 750 мм.

Под рабочей поверхностью стола предусматривается место для ног:

высота не менее - 600 мм;

ширина не менее - 500 мм;

глубина не менее - 400 мм.

При необходимости обзора рабочего места высота последнего должна превышать - 1200 мм. Кресло оператора должно обеспечивать надежную опору для тела. Высота сиденья примерно - 500 мм.

Проходы должны иметь ширину, позволяющую людям разминуться, примерно - 800 мм.

Однако большое значение в создании оптимальных условий труда имеют складывающиеся в коллективе взаимоотношения между работниками.

### 5.1 Мероприятия по устранению причин травматизма и профзаболеваний

Государственный стандарт (ГОСТ-74) определяет условия труда как совокупность факторов производственной среды, оказывающих влияние на здоровье и работоспособность человека в процессе труда.

Эти факторы можно разделить следующим образом:

* психофизиологические (нервно-психическая нагрузка, монотонность, ритм труда);
* санитарно-гигиенические условия;
* состояние воздуха;
* микроклимат;
* шум;
* освещение;
* эстетические;
* конструкторское и архитектурно-художественное оформление интерьеров;
* озеленение;
* социально-психологические.

Существуют научно разработанные методы, выявляющие влияние различных факторов среды на организм человека. Рациональная планировка и оборудование служебных помещений, внедрение физиологически обоснованных режимов труда и отдыха являются предпосылками высокоэффективной работы.

Рациональное освещение

Освещение считается нормальным, если можно длительное время работать без напряжения зрения. Служебные помещения ВЦ должны по возможности освещаться естественным светом, лучше, если естественный свет падает сверху слева. Рабочие столы не должны располагаться напротив окон. Яркость освещения для всего помещения должна быть одинаковой. Соотношение яркости света между рабочим местом и полом должно составлять не более 10/1. Между рабочим местом и потолком – не более 1/10. Между окнами и прилегающей площадью – не более 20/1.

Существует 5 видов освещения служебных помещений:

* прямой (свет падает прямо вниз);
* полупрямой (60-90% лучей падает вниз, а 10-40% направлены вверх);
* прямой косвенный (40-60% падает вниз);
* полукосвенный (10-40% света падает вниз, 60-90% направлено вверх);
* косвенный (90-100% света направлено вверх и отражается от потолка и стен).

Рекомендуется использовать полукосвенный или прямой косвенный тип освещения.

Микроклимат

В служебных помещениях ВЦ должен поддерживаться нормальный микроклимат. Воздух должен содержать не более 0,3% углекислоты. Постоянная температура – 18-20Сo. Относительная влажность воздуха в зимний период – 45-50%, а в летний – 50-55%. Должно проводиться кондиционирование воздуха. Оно повышает производительность труда на 15%.

Звукоизоляция и устранениешума

Шум вызывает напряжение, утомляемость, отрицательно влияет на ЦНС. Уровень шума измеряется в ДБ. В служебных помещениях ВЦ для проведения работы уровень шума не должен превышать 50Дб. Уровень шума для установления норм измеряется в следующих расчетных точках:

* на всех рабочих местах административных зданий на высоте 1,2 метра от уровня пола;
* в помещениях – 1,2 метра от пола в точках, расположенных на расстоянии 1,2 метра от стен.

Рационализация режимов труда и отдыха работников ВЦ

Хорошие условия труда благотворно влияют на работоспособность человека и производственные показатели, т.к. при меньшей утомляемости лучше используется рабочее время, повышается культура и производительность труда. Необходимость чередования периодов работы и отдыха обусловлено физиологическими закономерностями. Строго определенное чередование периодов работы и отдыха играют большую роль в поддержании трудового ритма. Существует несколько основных фаз работоспособности:

* врабатывание (нарастающая работоспособность, фаза длится от нескольких минут до 1,5 часа);
* высокая устойчивая работоспособность (эта фаза может удерживаться до 2,5 часов);
* падение работоспособности в результате развивающегося утомления.

Рационализация режимов труда и отдыха на ВЦ должна предусматривать регламентацию пауз на отдых и активизацию отдыха для более быстрого восстановления работоспособности с помощью производственной гимнастики, смены форм деятельности. Для работников ВТ и ЭВМ необходимо проводить гимнастику для глаз, стоя или сидя через каждые 20-25 минут.

### 5.2 Мероприятия по обеспечению противопожарной защиты

Пожар – неконтролируемое горение вне специального очага, наносящее материальный ущерб. Опасными факторами, воздействующими на людей при пожаре, являются: открытый огонь, повышенная температура воздуха, предметов и т.п.; токсичные продукты горения, дым, пониженная концентрация кислорода; обрушение и повреждение зданий, сооружений, взрывы. Наиболее вероятными причинами возникновения пожара в рассматриваемом помещении лаборатории являются причины электрического характера: короткое замыкание, перегрузки, искрение. В результате короткого замыкания, а также при плохом контакте на клеммах возникают искры, от которых могут загореться пластиковые корпуса ЭВМ и периферийных устройств.

Одной из наиболее важных задач пожарной профилактики является защита строительных конструкций от разрушения и обеспечения достаточной прочности, при условиях воздействия высоких температур при пожаре.

С обрушением конструкций процесс уничтожения материальных ценностей завершается, а тушение пожара в этом случае - бесполезно.

Для предотвращения распространения пожара с одной части здания в другую устанавливают противопожарные преграды в виде противопожарных смен, перегородок, перекрытий, дверей, клапанов и так далее.

В ВЦ противопожарные преграды в виде перегородок из несгораемых материалов устраивают между машинными залами, помещениями для размещения сервисной и периферийной аппаратуры, для хранения носителей информации, для персонала осуществляющего эксплуатацию ЭВМ.

Противопожарные стекла, окна, устраиваемые между машинным залом и помещением для размещения сервисной и периферийной аппаратуры, должны быть неоткрывающимися.

При пожаре, распространяющемся мгновенно, зачастую требуется производить эвакуацию людей из здания, охваченных огнем.

Процесс вынужденной эвакуации начинается одновременно из всех помещений здания и протекает в одном направлении в строку выходов. При этом создаются людские потоки с плотностью значительно большей, чем при нормальном движении, а в местах сужений может достигать предельных значений, опасных для человека.

Кратковременность процесса эвакуации достигается устройством эвакуационных путей и выходов, число, размеры и конструктивно-планированные решения, которые регламентированы строительными нормами.

В ВЦ выходы в машинном зале делают через тамбуры, оборудованные самозакрывающимися двупольными дверями со специальным уплотнением. Двери тамбура должны открываться в строку машинного зала.

Перегрузки и короткое замыкание в сети могут вызвать возгорание изоляции сетевых кабелей.

Для своевременного оповещения о пожаре в лаборатории установлены извещатели-датчики КИ-1, подключенные к системе автоматической пожарной сигнализации, реагирующие на повышенную температуру и дым. Для самостоятельной борьбы с пожаром в помещении лаборатории установлены два ручных углекислотных огнетушителя ОУ-5, расположенных в разных концах помещения.

Для предотвращения возникновения пожара, а также для уменьшения ущерба в случае возникновения пожара необходимо выполнять следующие мероприятия:

* проведение инструктажа по пожарной безопасности;
* обеспечение правильного режима эксплуатации ЭВМ и другого оборудования, установленного в лаборатории;
* проведение регулярного профилактического осмотра оборудования лаборатории;
* недопущение и предотвращение перегрузок в электрической сети;
* разработка плана эвакуации людей и материальных ценностей из помещения лабораторий.

Одной из наиболее важных задач пожарной профилактики является защита строительных конструкций от разрушения и обеспечения достаточной прочности, при условиях воздействия высоких температур при пожаре.

### 5.3 Мероприятия по обеспечению электробезопасности при эксплуатации технических средств

Электрические установки, к которым относится практически все оборудование электронных вычислительных машин, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведения профилактических работ человек может коснуться частей, находящихся под напряжением. Специфическая опасность электроустановок: токоведущие проводники, корпуса стоек ЭВМ и прочего оборудования, оказавшегося под напряжением в результате повреждения (пробоя) изоляции, не подают каких - либо сигналов, которые предупреждали бы человека об опасности. Реакция человека на электрический ток возникает лишь при протекании последнего через тело человека.

Проходя через тело человека, электрический ток оказывает на него сложное воздействие, вызывая при этом термическое, электролитическое, механическое и биологическое действие. Термическое действие тока проявляется в ожогах отдельных участков тела, нагреве тканей и биологических средств, что вызывает в них функциональные расстройства. Электролитическое действие тока выражается в разложении органической жидкости, крови и проявляется в изменении их физико-химического состава. Механическое действие тока приводит к разрыву мышечных тканей.

Биологическое действие тока заключается в способности тока раздражать и возбуждать живые ткани организма .

Исключительно важное значение для предотвращения электротравматизма имеет правильная организация обслуживания действующих электроустановок вычислительного центра, проведение ремонтных, монтажных и профилактических работ.

При этом под правильной организацией понимается строгое выполнение ряда организационных и технических мероприятий и средств:

* оформление работы нарядом или устным распоряжением;
* допуск к работе;
* надзор во время работы;
* оформление перерыва в работе;
* переводов на другое место;

### 5.4 Требования к помещению машинного зала

Помещения вычислительного центра, их размера (площадь, объем) должны в первую очередь соответствовать количеству работающих и размещаемому в них комплексу технических средств. В них предусматривают соответствующие параметры температуры, освещения, чистоты воздуха, обеспечивают изоляцию от производственных шумов и так далее.

Для обеспечения нормальных условий труда санитарные нормы СН 245-71 устанавливают на одного работающего, объем производственного помещения не менее 15 м3, площадь помещения выгороженного стенами или глухими перегородками не менее 4,5 м2.

Для эксплуатации электронных вычислительных машин следует предусматривать следующие помещения:

* основные помещения (машинный зал; помещения для размещения сервисной и периферийной аппаратуры);
* вспомогательные помещения (помещения для размещения приточно – вытяжных вентиляторов и устройств кондиционирования воздуха);
* производственные помещения (помещения для персонала, осуществляющего техническое обслуживание ЭВМ; помещение для персонала, осуществляющего эксплуатацию ЭВМ; помещение для приема/выдачи информации).

При этом все основные помещения вычислительного центра располагают в непосредственной близости друг от друга. Их оборудуют общеобменной вентиляцией и искусственным освещением.

К помещениям машинного зала и хранения магнитных носителей информации предъявляются особые требования. Площадь машинного зала должна соответствовать площади, необходимой по заводским техническим условиям данного типа ЭВМ. Высота зала должна быть 3 - 3,5 метра. Высоту подпольного пространства принимают равной 0,2 - 0,6 м. Конструкции, расположенные в этом пространстве, не должны иметь острых ребер и заусенцев для исключения различных повреждений кабелей и соединительных проводов. Габариты дверей машинного зала принимаются не менее 1,8 х 1,1 м из расчета возможности транспортирования оборудования.

Общие рабочие комнаты и кабинеты должны иметь естественное освещение. В остальных помещениях допускается искусственное освещение.

Рациональное цветовое оформление помещений направлено на улучшение санитарно-гигиенических условий труда, повышение его производительности и безопасности. Окраска производственных помещений вычислительного центра влияет на нервную систему человека, его настроение, производительность труда. Основные и производственные помещения целесообразно окрашивать в соответствии с цветом технических средств. Выбор цвета определяется радом факторов, в том числе конструкцией здания, характером выполняемой работы, освещенностью, количеством работающих.

Для снижения шума, создаваемого на рабочих местах внутренними источниками (техническими средствами, устройствами кондиционирования воздуха, компрессорами, насосами, преобразователями напряжения), а также шума, проникающего извне следует:

* ослабить шум самих источников, в частности, предусмотреть применение в их конструкциях акустических экранов, звукоизолирующих кожухов и так далее;
* снизить эффект суммарного воздействия на рабочие места отраженных звуковых волн за счет звукопоглощения энергии прямых звуковых волн поверхностями ограждающих конструкций
* применять рациональное расположение оборудования;
* использовать технические решения, направленные на изоляцию источников шума.

Одним из необходимых условий здорового и высокопроизводительного труда оператора является обеспечение чистого воздуха и нормальных метеорологических условий в рабочем помещении.

Метеорологические условия и микроклимат в производственных условиях определяются следующими параметрами:

* температура воздуха Т, С;
* относительная влажность Y, %;
* скорость движения воздуха на рабочем месте V, м/с;
* барометрическое давление Р, мм ртутного столба.

С целью обеспечения комфортных условий для обслуживающего персонала и надежности технологического процесса устанавливают дополнительные требования к воздушной среде помещений вычислительного центра. Так, в машинном зале температура воздуха должна быть 20 ± 2 оС. Относительная влажность воздуха в зале рекомендуется 55 ± 5%. В машинном зале, в хранилищах носителей информации рекомендуется поддерживать температуру и влажность воздуха постоянными, с относительно малыми колебаниями. Значительные колебания температуры приводят к изменению рабочих характеристик узлов и устройств ЭВМ.

В значительной мере производительность оператора зависит от освещенности помещения, а точнее сказать, от освещенности на рабочем месте. В зависимости от природы источника световой энергии различают естественное, искусственное и совмещенное освещение.

Естественное освещение подразделяют на боковое одностороннее или двустороннее, осуществляемое через окна; комбинированное, когда к верхнему освещению добавляется боковое.

В тех случаях, когда естественного освещения недостаточно в помещении устанавливают совмещенное освещение. При этом дополнительное искусственное освещение применяют не только в темное, но и в светлое время суток. Искусственное освещение может быть общим и комбинированным. При общем освещении рабочие места получают освещение от общей осветительной установки. Комбинированное освещение наряду с общим включает местное освещение, сосредотачивающее световой поток непосредственно на рабочих местах. Применение одного местного освещения недопустимо, так как возникает необходимость частой переадаптации зрения, создаются глубокие и резкие тени и другие неблагоприятные факторы.

Для эксплуатации ЭВМ следует предусматривать следующие помещения: машинный зал; помещение для хранения запасных деталей, инструментов; помещение для размещения устройств кондиционирования воздуха; помещение для персонала, осуществляющего эксплуатацию ЭВМ и другие. Одним из необходимых условий здорового и высокопроизводительного труда оператора является обеспечение чистоты воздуха и нормальных метеорологических условий.

Метеорологические условия и микроклимат в производственных условиях определяются следующими параметрами:

* температура воздуха Т;
* относительная влажность Y;
* скорость движения воздуха на рабочем месте V;
* барометрическое давление Р.

В дисплейном классе примерно следующие метеоусловия:

Т=22Сo, Y=60%, V=0.5м/с.

### 5.5 Мероприятия по охране окружающей среды

Для охраны окружающей среды от вредных воздействий производства широко используется экобиозащитная техника. К экобиозащитной технике относят средства и приспособления, защищающие человека и природную среду от воздействия опасных и вредных производственных факторов.

За основу определения эффективности защитных средств принимают классификацию по месту использования:

средства локализации источника, устанавливаемые непосредственно на выходе токсичных веществ и энергетических загрязнений из источника или частично встроенные в него;

защитные экраны – средства защиты на путях распространения опасных и вредных факторов;

индивидуальные средства, применяемые непосредственно для защиты человека.

Средства защиты атмосферы

Загрязнение воздуха производственных помещений может происходить за счет выбросов из технологического оборудования или при проведении технологических процессов без локализации выбросов. В этом случае возможно повышенное загрязнение воздуха рабочей зоны, а удаляемый из помещения вентиляционный воздух может стать причиной загрязнения атмосферного воздуха промышленных площадок и населенных мест.

На практике реализуются следующие варианты использования средств защиты атмосферы:

локализация токсических веществ в зоне их образования, очистка загрязненного воздуха в специальных аппаратах и его возврат в производственное или бытовое помещение, если после аппарата воздух соответствует нормативным требованиям к приточному воздуху;

локализация токсических веществ в зоне их образования, очистка загрязненного воздуха в специальных аппаратах, выброс и рассеивание в атмосфере (в ряде случаев перед выбросом отходящие газы разбавляют атмосферным воздухом);

очистка отработанных газов энергоустановок, например, двигатели внутреннего сгорания, в специальных агрегатах и выброс в атмосферу или производственную зону (рудники, карьеры и т. п.).

Средства защиты гидросферы

При выборе методов и технологического оборудования для очистки сточных вод необходимо учитывать, что требуемые эффективность и надежность очистного устройства обеспечиваются в определенном диапазоне концентраций примесей и расходов сточных вод. В соответствии с видами процессов, происходящих при очистке, принято все существующие методы делить на 3 группы: механические, физико-химические и биологические.

При механической очистке сточных вод от взвешенных веществ используют процеживание (в решетках и волокноуловителях), отстаивание (в песколовках, отстойниках и жироуловителях), обработку в поле действия центробежных сил (в открытых или напорных гидроциклонах и центрифугах) и фильтрование (зернистыми фильтрами или фильтрами-сепараторами).

Физико-химические методы используют для очистки сточных вод от растворенных примесей и от взвешенных веществ. В настоящее время, в связи с использованием оборотных систем водоснабжения, существенно увеличивается роль физико-химических методов очистки сточных вод, основными из которых являются:

* флотация;
* экстрация;
* нейтрализация;
* сорбция;
* ионообменная очистка;
* электрохимическая очистка.

Биологическая очистка сточных вод применяется для выделения из них растворенных органических веществ, и основана на способности микроорганизмов использовать для питания, содержащиеся в сточных водах органические вещества (кислоты, спирты, белки, углеводы и т. п.).

Биологическую очистку осуществляют в естественных условиях (на полях орошения, полях фильтрации и биологических прудах) и искусственных сооружениях (аэротэнки и биофильтры).

Средства защиты литосферы

Для защиты почв, лесных угодий, поверхностных и грунтовых вод от твердых и жидких отходов в настоящее время широко используют сбор и складирование промышленных и бытовых отходов на свалках и полигонах. Переработку промышленных отходов производят на специальных полигонах, предназначенных для централизованного сбора, обезвреживания и захоронения токсичных отходов промышленных предприятий, НИИ и учреждений. Переработка отходов на полигонах предусматривает использование физико-химических методов, термическое обезжиривание с утилизацией теплоты, прокаливание песка и формовочной земли, подрыв баллонов в специальной камере, затаривание отходов в герметичные контейнеры и их захоронение. Полигоны должны иметь санитарно-защитные зоны.

В 70-80 г. получила развитие термическая переработка отходов сжиганием их в печах на мусоросжигающих заводах. Термический способ переработки отходов экологичнее складирования на свалках и полигонах.

Более рациональным способом защиты литосферы от отходов производства и быта является освоение специальных технологий по сбору и переработке отходов. При сборе отходы необходимо сортировать. В ряде стран для сбора бытовых отходов на улицах городов установлены специальные контейнеры для бумаги, стекла, металла и т. д.

Наиболее эффективным методом решения проблемы защиты литосферы от промышленных отходов является применение безотходных и малоотходных технологий и производств.

Под безотходной технологией (безотходным производством, безотходной системой) понимают не просто технологию или производство того или иного продукта (или продуктов), а принцип организации и функционирования производства.

При этом рационально используются все компоненты сырья и энергия в замкнутом цикле:

первичные сырьевые ресурсы → производство →

→ потребление → вторичные сырьевые ресурсы,

то есть не нарушается сложившееся экологическое равновесие в биосфере.

Малоотходная технология является промежуточной ступенью при создании безотходного производства. При малоотходном производстве вредное воздействие на окружающую среду не превышает санитарных норм, но по техническим, экономическим, организационным или другим причинам часть сырья и материалов переходит в отходы и направляется на длительное хранение или захоронение.

Основой безотходных производств является комплексная переработка сырья с использованием всех его компонентов, поскольку отходы производства – это по тем или иным причинам неиспользованная или недоиспользованная часть сырья. Большое значение при этом приобретает разработка ресурсосберегающих технологий.

Малоотходная и безотходная технологии должны обеспечить: переработку отходов производства и потребления с получением товарной продукции или любое полезное их использование без нарушения экологического равновесия.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. METANIT.COM Сайт о программировании [сайт]   
   URL: <https://metanit.com/web/nodejs> (дата обращения 15.11.2021)
2. StackOverflow [сайт]   
   URL: <https://stackoverflow.com> (дата обращения 02.06.2022)
3. Habr [сайт]   
   URL: <https://habr.com/ru/all> (дата обращения 29.05.2022)
4. Vue.js Introduction [сайт]  
   URL: <https://vuejs.org/guide/introduction.html> (дата обращения 15.05.2022)
5. Vue Router The official router for Vue.js. [сайт]  
   URL: <https://router.vuejs.org/installation.html> (дата обращения 15.05.2022)
6. Sequelize [сайт]  
   URL: <https://sequelize.org/docs/v6/getting-started> (дата обращения 29.03.2022)
7. Cloudinary [сайт]  
   URL: [https://cloudinary.com/getting-started](https://cloudinary.com/console/c-63faae6763d11cb4ab04fff15d2a9b/getting-started) (дата обращения 15.04.2022)
8. Anime.js [сайт]  
   URL: <https://animejs.com/documentation> (дата обращения 09.12.2021)
9. Express [сайт]  
   URL: <https://expressjs.com/ru/guide/routing.html> (дата обращения 27.10.2021)
10. Node.js [сайт]  
    URL: <https://nodejsdev.ru/doc/express> (дата обращения 16.10.2021)
11. MDN Web Docs [сайт]  
    URL: <https://developer.mozilla.org/ru> (дата обращения 19.05.2022)
12. Google Fonts [сайт]  
    URL: <https://fonts.google.com> (дата обращения 08.11.2021)
13. ИТШЕФ [сайт]  
    URL: <https://itchief.ru> (дата обращения 24.03.2022)
14. Современный учебник JavaScript [сайт]  
    URL: <https://learn.javascript.ru> (дата обращения 21.10.2021)
15. HtmlBook [сайт]  
    URL: <http://htmlbook.ru> (дата обращения 04.12.2021)