

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Вычислительной техники
(полное название кафедры)

Утверждаю

Зав. кафедрой Якименко А.А.

(подпись, инициалы, фамилия)

«__» _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Гунгер Антон Константинович

(фамилия, имя, отчество студента – автора работы)

Модернизация веб-сервиса по продаже оборудования

(тема работы)

Факультет автоматизации и вычислительной техники

(полное название факультета)

Направление подготовки 09.03.01 Информатика и вычислительная техника

(код и наименование направления подготовки бакалавра)

**Руководитель
от НГТУ**

Новицкая Юлия Вадимовна

(фамилия, имя, отчество)

Ст. преподаватель каф. ВТ, НГТУ

(ученая степень, ученое звание)

(подпись, дата)

**Автор выпускной
квалификационной работы**

Гунгер Антон Константинович

(фамилия, имя, отчество)

АВТФ, АВТ-709

(факультет, группа)

(подпись, дата)

Новосибирск 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра _____ **Вычислительной техники** _____
(полное название кафедры)

УТВЕРЖДАЮ

Зав. кафедрой _____ **Якименко А.А.** _____
(фамилия, имя, отчество)

(подпись, дата)

**ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ БАКАЛАВРА**

студенту _____ **Гунгеру Антону Константиновичу** _____
(фамилия, имя, отчество)

Направление подготовки **09.03.01 Информатика и вычислительная техника** _____
(код и наименование направления подготовки бакалавра)

Факультет автоматизации и вычислительной техники
(полное название факультета)

Тема _____ **Модернизация веб-сервиса по продаже оборудования** _____
(полное название темы выпускной квалификационной работы бакалавра)

Исходные данные (или цель работы) Исходные данные: язык гипертекстовой разметки HTML, язык программирования JavaScript, таблицы каскадных стилей CSS, язык программирования PHP, препроцессор SASS, программный комплекс Open server, современные браузеры, редактор исходного кода Visual Studio Code, система контроля версий Git.

Цель работы: модернизировать структуру сайта, разработать и протестировать сайт.

Структурные части работы Введение

1. Постановка задачи
2. Обзор предметной области
3. Принципы разработки веб-сервиса
4. Разработка веб-сервиса
5. Проектирование
6. Реализация

7. Тестирование

Заключение

Список использованных источников

Приложение

Задание согласовано и принято к исполнению.

**Руководитель
от НГТУ**

Новицкая Юлия Вадимовна

(фамилия, имя, отчество)

ст. препод. каф. ВТ, НГТУ

(ученая степень, ученое звание)

(подпись, дата)

Студент

Гунгер Антон Константинович

(фамилия, имя, отчество)

АВТФ, АВТ-709

(факультет, группа)

(подпись, дата)

Тема утверждена приказом по НГТУ № 737/2 от 25 февраля 2021г.

изменена приказом по НГТУ № _____ от « ____ » _____ 2021 г.

ВКР сдана в ГЭК № _____, тема сверена с данными приказа

(подпись секретаря государственной экзаменационной комиссии по защите ВКР, дата)

(фамилия, имя, отчество секретаря государственной
экзаменационной комиссии по защите ВКР)

РЕФЕРАТ

Выпускная квалификационная работа бакалавра выполнена студентом Гунгером А.К.

Руководитель проекта: старший преподаватель Новицкая Ю.В.

Тема: «Модернизация веб-сервиса по продаже оборудования».

Пояснительная записка содержит 62 страницы, 55 рисунков.

Цель выпускной квалификационной работы бакалавра – модернизация существующего сайта с применением современных веб–технологий.

Ключевые слова: МОДЕРНИЗАЦИЯ САЙТА, ВЕБ-СЕРВИС, PHP, JAVASCRIPT, JSON, БАЗА-ДАННЫХ, MYSQL, PHPMYADMIN, HTML, CSS, ВЕБ-СЕРВЕР.

В рамках выпускной работы рассмотрены аналогичные веб-сервисы, способы и технологии их реализации. Также были изучены основные современные технологии веб-программирования.

В процессе выполнения работы с помощью современных средств реализации веб–сайтов был модернизирован устаревший сайт.

Решены поставленные задачи:

- Изучен сайт, который требуется модернизировать;
- Сформированы требования по модернизации;
- Проанализированы и выбраны технологии для реализации;
- Разработаны клиентские и серверные части сайта и админ–панели сайта;
- Проведено тестирование модернизированного веб-сервиса.

В пояснительной записке дано полное описание поставленных задач и способы их выполнения. При модернизации веб-сервиса использовалось множество технологий и языков: HTML5, SASS, PHP, JS и другие.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1 ПОСТАНОВКА ЗАДАЧИ.....	8
2 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ	9
2.1 Обзор прежней версии сайта.....	9
2.2 Веб-сервисы	11
3 ПРИНЦИПЫ РАЗРАБОТКИ ВЕБ-СЕРВИСА	13
3.1 Стандарты front-end разработки.....	13
3.2 Инструменты front-end разработки.....	13
3.2.1 Язык разметки HTML	13
3.2.2 Стилизация разметки CSS	14
3.2.3 Язык программирования JavaScript	15
3.2.4 Адаптивный дизайн	15
3.3 Стандарты Back-end разработки	16
3.4 Инструменты и методы Back-end разработки	16
3.4.1 Язык программирования PHP	16
3.4.2 Взаимодействие Back-end и front-end	17
3.4.3 База данных и СУБД	18
3.4.4 СУБД MySQL	18
4 РАЗРАБОТКА ВЕБ-СЕРВИСА.....	20
4.1 Функциональные требования.....	20
4.2 Инструменты реализации	20
4.3 Системные требования	22
5 ПРОЕКТИРОВАНИЕ.....	23
5.1 Проектирование клиентской части сайта	23
5.2 Проектирование серверной части сайта	25
5.3 Проектирование админ-панели сайта	26
5.4 Структура базы данных	26
6 РЕАЛИЗАЦИЯ	29
6.1 Верстка сайта	29

6.2	Реализация клиентской части сайта	39
6.3	Реализация серверной части сайта.....	41
6.4	Реализация админ–панели сайта.....	43
7	ТЕСТИРОВАНИЕ И РАЗМЕЩЕНИЕ	50
7.1	Тестирование взаимодействия БД и сайта	50
7.2	Тестирование SMTP отправки почтовых сообщений.....	53
7.3	Тестирование сайта.....	54
7.4	Размещение веб-сервиса.....	57
7.4.1	Веб-сервер	57
7.4.2	Хостинг.....	57
	ЗАКЛЮЧЕНИЕ	59
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60
	ПРИЛОЖЕНИЕ	62

ВВЕДЕНИЕ

Современный мир окутан цифровыми технологиями. Сейчас большая часть населения Земли обладает доступным выходом в Интернет и посещает большое количество сайтов и веб-сервисов – социальные сети, онлайн-магазины, сайты компаний и многое другое.

Количество активных пользователей в 2020 году составило 4,5 миллиарда человек, причем уникальных мобильных пользователей насчитывалось 4,18 миллиарда. Список активных пользователей социальных сетей в мире достиг 3,8 миллиарда человек, и почти каждый из них использует социальные сети на мобильных устройствах – 3,75 миллиарда.

Текущий показатель численности населения планеты составляет 7,76 миллиарда человек. Таким образом, 56% жителей Земли являются активными пользователями Сети.

Первый сайт был создан в 1991 году. Последняя статистика Интернета в 2020 году говорит, что в мире насчитывается около 1,8 миллиарда действующих сайтов.

Скорость расширения Всемирной паутины остается стабильной и не показывает никаких признаков замедления [1].

С каждым годом количество сайтов только растет, и эта тенденция будет сохраняться всегда, т.к. переход в цифровую эпоху неминуем для каждого и это только вопрос времени.

Также и технологии, связанные с Интернетом, не стоят на месте и каждый год обновляются и заменяются, вследствие чего сайты и веб-сервисы устаревают как внешне, так и технологически, именно поэтому важно раз в несколько лет обновлять «внутренности» своих сервисов.

На данный момент сайт компании «Маяк-авангард» устарел и нуждается в обновлении. Именно это является целью моей ВКР – модернизировать сайт компании по продаже оборудования.

1 ПОСТАНОВКА ЗАДАЧИ

Целью моей выпускной квалификационной работы является модернизация веб-сервиса по продаже оборудования.

Сайт компании «Маяк авангард» на сегодняшний день технически и морально устарел. Часть функционала больше не работает, а дизайнерское оформление оставляет желать лучшего. Поэтому данная версия сайта нуждается в модернизации.

Сайт должен содержать админ-панель, где будет предоставлена возможность получить доступ ко всем заказчикам продукции, просмотреть статистику заказов на сайте и среднемесячную посещаемость страниц, все запросы будут проходить через базу данных, откуда будут браться значения и уже на стороне сервера обрабатываться для вывода в админ-панель.

Для реализации задуманной системы потребуется решить следующие задачи:

- Изучить сайт, который требуется модернизировать;
- Сформировать требования;
- Проанализировать и выбрать технологию для реализации;
- Спроектировать клиентские и серверные части веб-сервиса;
- Реализовать поставленные цели;
- Провести тестирование.

2 ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

2.1 Обзор прежней версии сайта

Компания «Маяк-авангард» занимается индивидуальной разработкой и производством технологического оборудования для промышленной и пищевой отрасли. Основным источником клиентов компании является веб-сервис, который предоставляет услуги оформления заказов и просмотра доступной информации о продукции и производстве.

Сайт компании «Маяк-авангард» использует стандартные средства разработки, такие как HTML, CSS, JS. Страницы выполнены в старом стиле 2000 – 2010 и на сегодняшний день устарели.

Исходные файлы проекта находятся в ужасном состоянии и попросту не структурированы, а написанный код подлежит полному рефакторингу, так как он не упорядочен и просто разбросан по файлу.

Также на сайте имеется возможность выслать заявку на почту производителя и на данный момент эта функция выдает ошибку страницы. Описанный процесс изображен на рисунке 1.

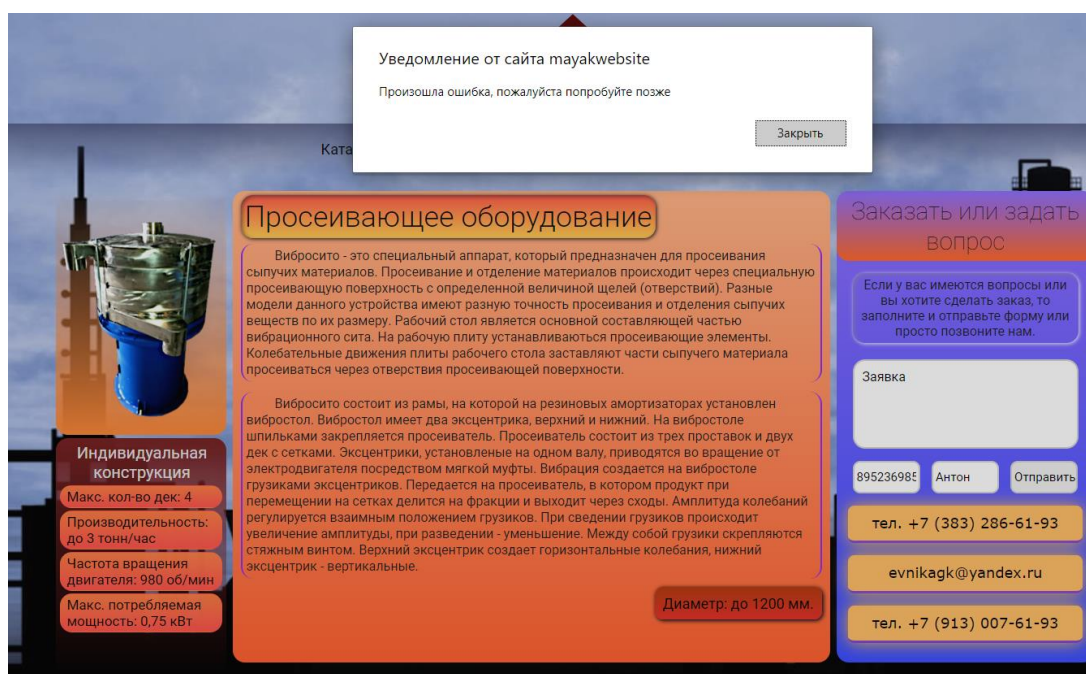


Рисунок 1 – Ошибка при отправке заявки

Структура сайта оставляет желать лучшего, некоторые ссылки перестали работать, а почтовые запросы не выполняются. Сайт содержит всего 8 страниц:

- Галерея;
- Ёмкостное оборудование;
- Сироповарки;
- Просеивающее оборудование;
- Диссоolverы;
- Бисерные матрицы;
- Реакторы.

Переход на любую вкладку страницы занимает большое количество времени вследствие плохой оптимизации медиа-контента и перегруженности скриптов.

Контактная информация утратила свой смысл, так как владельцы, адреса офисов и контактные номера давно поменялись, но так и не были обновлены на сайте. Главная страница сайта представлено на рисунке 2.



Рисунок 2 – Веб-сервис компании «Маяк авангард»

На главной странице располагается каталог позиций, по страницам которых может переходить пользователь. Данные разделы устарели, т.к.

товарный ряд поменялся за эти годы и все позиции, представленные на сайте, давно устарели.

Раздел галерея потерял свою актуальность и требуется полное обновление фотографий производства и оборудования. Страница «Галерея» представлена на рисунке 3.

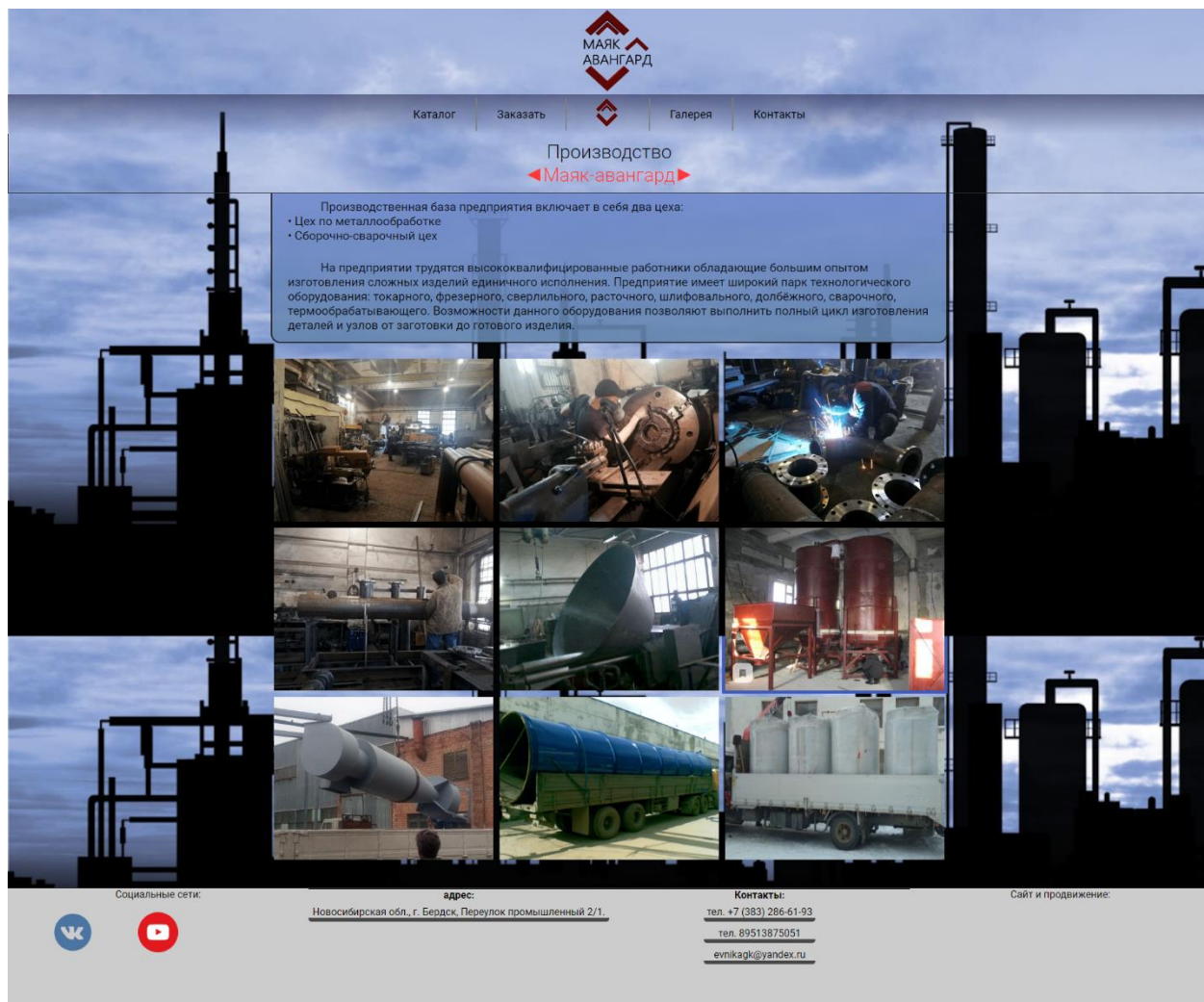


Рисунок 3 – Страница «Галерея»

2.2 Веб-сервисы

Веб-сервисы – это автономные модульные распределенные динамические приложения, которые можно описывать, публиковать, размещать или вызывать по сети для создания продуктов, процессов и цепочек поставок. Эти приложения могут быть локальными, распределенными или сетевыми. Веб-сервисы построены на основе открытых стандартов, таких как TCP/IP, HTTP, Java, HTML и XML [3].

Подобная архитектура предоставляет возможности создания современных сайтов, суть которых предполагает взаимодействие между различными модулями, написанными на стороне сервера, вызываемыми на стороне клиента и наоборот.

Также это блок управляемого кода, который можно вызывать удаленно с помощью HTTP [3].

HTTP (протокол передачи гипертекста) – это прикладной протокол передачи данных в Сети. На текущий момент используется для получения информации от веб-сайтов. Протокол HTTP основан на использовании технологии «клиент-сервер»: клиент, отправляющий запрос, является инициатором соединения; сервер, получающий запрос, выполняет его и отправляет клиенту результат.

HTTPS (безопасный протокол передачи гипертекста) – это расширение протокола HTTP, поддерживающее шифрование посредством криптографических протоколов SSL и TLS.

Чем отличаются HTTP от HTTPS:

- HTTPS не является отдельным протоколом передачи данных, а представляет собой расширение протокола HTTP с надстройкой шифрования;
- передаваемые по протоколу HTTP данные не защищены, HTTPS обеспечивает конфиденциальность информации путем ее шифрования;
- HTTP использует порт 80, HTTPS – порт 443 [2].

Блок управляемого кода можно активировать с помощью HTTP-запросов. Веб-сервисы позволяют раскрыть функциональность существующего кода через сеть. После того, как он выставлен в сети, другие приложения могут использовать функциональные возможности вашей программы [3].

3 ПРИНЦИПЫ РАЗРАБОТКИ ВЕБ-СЕРВИСА

3.1 Стандарты front-end разработки

Front-end разработка является важнейшей стадией проектирования веб-сервиса, основная задача – это обеспечить дизайн, который будет привлекать пользователя, а также создать удобное управление и взаимодействие с функциями, представленными на сайте.

Первым делом создается макет дизайна в сторонних программах, в процессе создания возникают правки и смены приоритетов, которые приходится отражать в макете, этим обычно занимаются дизайнеры. Главное взаимодействие макета и кода происходит через согласование с верстальщиками, которые помогают подобрать более явную и подходящую структуру сайта для проекта. И вот когда структура сайта определена – начинается верстка самого сайта, а это значит, что разработчики переносят нарисованный дизайн на реально рабочий прототип, который впоследствии наполняется контентом.

Стоит отметить, что общение между дизайнерами и верстальщиками не прекращается на протяжении всего жизненного цикла разработки веб-сервиса.

3.2 Инструменты front-end разработки

3.2.1 Язык разметки HTML

HTML – это язык гипертекстовой разметки, который браузер использует для интерпретирования его в веб-элементы страниц. Этот язык является стандартом во всем мире. Его используют в каждом проекте, связанным с созданием веб-контента. С развитием технологий стандарты языка HTML менялись несколько раз, и на данный момент основным стандартом является HTML5 [12].

Структура HTML каждого документа проста и состоит из тегов-элементов, которые определяют положение друг друга в основном окне браузера. Каждый тег может иметь свои параметры и выполнять разные функции отображения.

В основном теги бывают двух видов:

- Одиночные: `<input>`;

- Парные: `<div>` `</div>`.

Каждый тег может иметь несколько атрибутов, которые отвечают за функционал и параметры отображения элемента, они прописываются в открывающем теге. Внутри тегов содержатся другие теги, либо текст, который будет отображаться в элементе.

3.2.2 Стилизация разметки CSS

CSS – язык таблиц стилей, используемый для формирования внешнего вида веб-страницы. Данный язык в основном используется в связке с HTML-кодом. Таблица стилей CSS предоставляют возможность задавать различные параметры для графического отображения элементов страницы.

Основные возможности CSS:

- Изменение шрифта текста;
- Изменение цвета текста и элементов;
- Изменение размеров элементов;
- Изменение оформления элементов;
- Изменение поведения элементов;
- Изменение расположения.

Само формирование стилей состоит из правил и их параметров. CSS можно прописывать напрямую в HTML-документе через обозначающие атрибуты или теги, или можно создавать отдельные файлы, которые можно подключить в HTML. CSS-файл состоит из селекторов, в которых расположены правила с параметрами. Впоследствии селекторы будут обозначены в тегах нужных элементов в HTML-файлах [13].

Применяя CSS, верстальщик может создавать неповторимые паттерны веб-сервисов. Задавая правила, они добиваются изменения формы, цветовой гаммы и расположения элементов, что в итоге даёт более воспринимаемый пользовательский интерфейс.

Технология CSS появилась позже HTML и была «свежим глотком воздуха» для веб-страниц тех времен. Сайты в то время были однотипными и

монотонными. Впоследствии CSS проходил несколько стадий обновления и на сегодняшний момент актуальной версией является CSS3.

3.2.3 Язык программирования JavaScript

JavaScript – мультипарадигменный язык программирования.

Поддерживает объектно–ориентированный, императивный и функциональный стили.

Современный JavaScript – это «безопасный» язык программирования. Он не предоставляет низкоуровневый доступ к памяти или процессору, потому что изначально был создан для браузеров, не требующих этого [14].

В браузере для JavaScript доступно всё, что связано с манипулированием веб–страницами, взаимодействием с пользователем и веб–сервером.

Например, в браузере JavaScript может:

- Добавлять новый HTML–код на страницу, изменять существующее содержимое, модифицировать стили;
- Реагировать на действия пользователя, щелчки мыши, перемещения указателя, нажатия клавиш;
- Отправлять сетевые запросы на удалённые сервера, скачивать и загружать файлы;
- Получать и устанавливать cookies, задавать вопросы посетителю, показывать сообщения;
- Запоминать данные на стороне клиента («local storage») [4].

3.2.4 Адаптивный дизайн

Адаптивный дизайн – способность корректного отображения сайта на устройствах с разным разрешением экрана. Веб–страницы с данным свойством используют заранее заложенные механизмы CSS, которые позволяют верстать сайт с учетом динамически изменяющегося расширения окна браузера, что помогает избежать заранее прописанных размеров и параметров элементов для каждого вида диагонали.

В современной практике каждый хороший веб-сайт должен быть адаптирован для персонального компьютера, планшета, смартфона и ноутбука. За счет добавления адаптивности работа верстальщика усложняется в несколько раз, так как ему приходится использовать дополнительные механизмы регулирования положения и размеров веб-элементов и проводить их тестирование на разных расширениях и платформах.

Также имеется еще ряд недостатков:

- Размер подгружаемого CSS и JS будет больше за счет дополнительного функционала, направленного на адаптивность;
- Соответственно время разработки будет значительно больше, чем для сайта без адаптивной верстки.

3.3 Стандарты Back-end разработки

Ранее были обговорены стандарты front-end разработки, которая должна предоставлять корректный и понятный интерфейс, но интерфейс должен работать с какими-то данными, именно здесь приходит на помощь back-end разработка.

Back-end разработка – это набор аппаратно-программных средств, при помощи которых реализуется логика работы сайта. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера [5]. Там происходит обработка данных, в том числе полученных из базы данных и тех, что приходят на сервер от клиента.

Сутью back-end является объединение сервера с пользователем и самым популярным инструментом для создания этой связи является язык программирования PHP.

3.4 Инструменты и методы Back-end разработки

3.4.1 Язык программирования PHP

Язык HTML дает достаточный функционал для создания веб-страниц, но все-таки это не язык программирования и обратиться к базе данных, или написать простое математическое выражение с его помощью невозможно. На

помощь HTML приходит PHP – Hypertext Preprocessor, данный инструмент является языком программирования, чаще всего его называют серверным.

Основной задачей PHP является внедрение обрабатываемых веб-элементов в html-код, чтобы HTML не представлял собой статический интерфейс. А при помощи PHP программист может выводить или записывать данные БД, обрабатывать условия и запросы JSON, а также именно PHP дает возможность взаимодействия с данными на странице [15].

Главным преимуществом серверного языка PHP является модульность и встраиваемость в HTML код, что в совокупности дает простое интегрирование PHP на сайт. Именно поэтому PHP находится на 8-й строчке в списке самых популярных языков программирования [10].

3.4.2 Взаимодействие Back-end и front-end

Взаимодействие front-end и back-end происходит по кругу: front-end отправляет пользовательскую информацию в back-end, там она обрабатывается и возвращается обратно, приняв понятную форму.

Как правило, этими видами работ занимаются разные специалисты, но каждому из них желательно понимать принципы, по которым работают коллеги. К примеру, дизайнеру интерфейсов важно хотя бы в общих чертах знать, что представляет собой back-end проекта, которым он занимается. Это поможет ему адекватно оценить, какие технические возможности могут быть у сайта или приложения.

Существует несколько вариантов взаимодействия front-end и back-end:

HTTP-запрос напрямую отправляется на сервер, сервер ищет информацию, встраивает ее в шаблон и возвращает в виде HTML-страницы;

Вариант с использованием инструментария AJAX (Asynchronous JavaScript and XML). В этом случае запрос отправляет JavaScript, загруженный в браузер, а ответ приходит в формате XML или JSON;

Как можно видеть, современная разработка – это сложный и многоуровневый процесс, который делится на две части: клиентскую и серверную. В арсенале front-end- и back-end-программистов есть масса

инструментов, выбор которых зависит от поставленных задач и сложности проекта. Оба вида разработки включают в себя технические и творческие компоненты [6].

3.4.3 База данных и СУБД

Для обработки информации на back-end используются данные и чаще всего нужно какое-то хранилище, где все необходимые массивы информации будут упорядоченно храниться. Подобными задачами занимается база данных, именно она предоставляет упорядоченную структуру хранения данных, которые имеют определенные правила для упрощения доступа к записанной информации.

Но БД отвечает только за хранение информации, для доступа к данным используются системы управления базами данных или СУБД.

СУБД организует доступ к внутренним параметрам и данным БД. Именно через этот управляющий орган осуществляется связь отображения данных и самих данных.

Через СУБД имеется возможность:

- Записывать данные;
- Читать данные;
- Изменять данные;
- Удалять данные.

Поэтому любой большой веб-сайт использует базы данных, так как через их трафик проходит огромное количество информации, которая впоследствии пригодится для дальнейшего использования ресурсом.

3.4.4 СУБД MySQL

Что касается общего определения, то MySQL – это система управления реляционными базами данных с открытым исходным кодом (СУРБД) с моделью клиент-сервер. СУРБД – это программное обеспечение или служба, используемая для создания и управления базами данных на основе реляционной модели [7].

По сути своей система базируется на сервере, то есть все данные записываются на сервер БД, и, чтобы получить к ним доступ, программист должен обратиться к серверу MySQL. Как раз для общения с MySQL используется серверный язык программирования – PHP.

Процессы, происходящие в среде MySQL:

- Система создает базу данных для хранения информации (ее сортировки, идентификации и т.п.);
- Клиенты (другие компьютеры в сети) подают запросы к базе с помощью специфичных SQL-команд;
- Серверное приложение обрабатывает запрос и выдает ответ клиенту (выдает запрошенные данные) [7].

4 РАЗРАБОТКА ВЕБ-СЕРВИСА

4.1 Функциональные требования

Для модернизации сайта требуется переписать практически все, так как старая версия сайта не удовлетворяет условиям современного UI. А это значит, что под полное редактирование попадает вся структура веб-элементов и все стили сайта.

На данный момент функционал сайта позволяет только просматривать информацию о товарах, контактах, и фотографии производства. Для пользователей и сотрудников этой функциональности будет мало, поэтому стоит описать новые функции, которые будут реализованы на сайте.

Таким образом, все элементы UI для взаимодействия с данными будут переписаны и реструктуризированы, или делегированы отдельным страницам. Также ранее оговаривалась, что функции обработки и отправки данных через эти элементы не работают, а значит, что все функции отправки тоже подвергаются переосмыслению и редактированию.

Также вследствие роста компании у веб-сервиса должна появиться админ-панель, с помощью которой менеджеры и маркетологи должны получать основную информацию о посещениях, заказах и популярности товаров.

Вследствие переосмысления сущности данных добавляются инструменты, которые будут структурировано хранить данные и давать возможность обращаться к ним, для этого добавим админ-панель, которая будет предоставлять доступ к хранящимся данным.

4.2 Инструменты реализации

Для модернизации сервиса понадобится набор инструментов, который подходит для быстрой и легкой разработки, а также придётся использовать часть старых скриптов, написанных для части сервиса, который не предполагается переносить на новые технологии.

Итак, для создания правильной верстки придётся использовать HTML 5, данная технология зарекомендовала себя как самое простое и быстрое решение, без лишних имплементаций и пристраивания сторонних технологий.

Для создания дизайн-составляющей будет использоваться CSS3 в качестве промежуточной технологии, так как данный язык не очень удобен при разработке большого объема стилей, будет использован препроцессор SASS, который расширяет возможности стандартного CSS и привносит множество новых функций, которые непременно пригодятся при разработке.

Соответственно в связке с HTML выгодно использовать JavaScript, так как он легко встраивается в код веб-элементов и дает возможность глубокого обхода DOM-дерева, что, в свою очередь, предполагает легкое взаимодействие с элементами UI. Также JS отлично подходит для обработки пользовательских данных и создания JSON-объектов.

Все вышесказанное будет использовано для реализации клиентской части сервиса.

Теперь стоит уделить внимание технологиям, которые организуют серверное пространство.

В первую очередь выбор падает на скриптовый язык программирования PHP. Он является самым предпочтительным для веб-разработчиков, так как прост в изучении и в использовании, а также отлично работает с HTML, который может непосредственно встраиваться прямо в код. С помощью PHP можно совершать SQL-запросы к базе данных и обрабатывать их на стороне сервера, после чего также легко организовывать их отправку на сторону клиента.

Для отображения статистических графиков используется API от Google. Речь идет о Google Charts, данная технология позволяет отображать различные диаграммы и графики. Серверу достаточно сформировать пакет параметров и значений, и просто отправить их сервису, а он, в свою очередь, вышлет веб-элемент, в котором будет составлен интерактивный график.

В качестве СУБД выбираем MySQL, который предоставляет легкую и понятную структуру хранения данных, а также она поддерживает транзакции на уровне отдельных записей и множество других функций.

Для общения клиента и сервера используется технология AJAX, так как позволяет быстро и легко передавать любые данные на сервер.

А для удобного управления MySQL на уровне проектирования БД будет использоваться веб-приложение PhpMyAdmin, которое позволит быстро просматривать таблицы, записи и другую информацию о БД, также там присутствует множество полезных функций:

- Просмотр логов;
- Редактирование БД через UI;
- Создание запросов;
- Экспорт/импорт БД;
- Создание процедур и триггеров.

4.3 Системные требования

Для самой разработки программный комплекс, который будет содержать нижеперечисленные технологии. Этого будет достаточно для всего цикла разработки.

В качестве веб-сервера будет взят Apache, а надстройкой для него будет являться программный комплекс OpenServer, так как он бесплатен и в нем уже произведены настройки для корректной работы сервера.

А для того чтобы можно было в полную меру использовать возможности сайта, его следует поместить на сервер, размещенный в сети Интернет, для этого понадобятся услуги хостинг-компаний, также хостинг должен предоставлять следующие инструменты:

- СУБД MySQL;
- Веб-сервер Apache;
- Веб-приложение PhpMyAdmin;
- Поддержка SMTP.

5 ПРОЕКТИРОВАНИЕ

5.1 Проектирование клиентской части сайта

Структура страниц сайта содержит всего две части: шаблонную информационно-навигационную панель или шапку, а также основной frame, где располагается весь полезный контент страницы.

Навигационный веб-элемент предполагает наличие кнопок, ссылающихся на следующие разделы сайта:

- Главная;
- Каталог;
- Контакты;
- Заказать;
- Производство.

Главная страница является начальной и содержит основную информацию о компании «Маяк-авангард». Также на этой странице располагаются разделы с категориями продукции и лучшими товарами, которые предлагает компания. Имеется контактная информация с подключенной картой, которая показывает расположение офиса и комплекса компании.

При переходе в раздел «Каталог» нас перенаправляет ниже по странице к веб-элементу, содержащему категории производства и лучшие товары.

При переходе в раздел «Контакты» нас перенаправляет ниже по странице к веб-элементу, содержащему контактную информацию и местоположение объектов компании.

Для более удобного и заманчивого предложения баннеры с качествами компании были вынесены в единый веб-элемент и прописаны прокручивающие функции, что помогло привлечь внимание и сохранить место на веб-странице.

Также была полностью изменена цветовая палитра сайта, с темно-синего на бордовый, чтобы подчеркнуть цветовую гамму компании, изображенную на логотипе.

Раздел «Галерея» был переименован в «Производство» и полностью переработан с соблюдением всех стандартов современного UE/UX. Там располагаются фотографии цехов и оборудования, которое компания производит, а также страница содержит описание и информацию о производственном процессе.

После перехода в раздел «Каталог» можно увидеть представление двух категорий оборудования:

- Пищевое;
- Промышленное.

При нажатии на одну из категорий пользователь переходит на страницу товаров, где представлены различные разделы оборудования выбранной категории, для пищевого оборудования имеются следующие разделы:

- Сушка/рассев;
- Сироповарки;
- Ёмкости;
- Дозирование/смешивание;

А для промышленного оборудования имеются разделы:

- Вибросито;
- Деревообработка;
- Станции затаривания;
- Транспортирующее оборудование;
- Реакторы;
- Ёмкости;
- Сушка;
- Пивные котлы;
- Диссоolverы.

После перехода в раздел оборудования перед пользователем предстают характеристики, фото и описание каждой позиции. При этом имеется отдельный

элемент для просмотра информации и отображения всех позиций, при нажатии выводятся характеристики выбранной позиции.

Последним разделом является «Заказать», он был модернизирован. В прежней версии в данном разделе предлагалось заказать ёмкостное оборудование в зависимости от индивидуальных параметров, введенных пользователем. Теперь имеется возможность заказать не только ёмкостное оборудование, но и другие позиции, предлагаемые компанией.

5.2 Проектирование серверной части сайта

Для новой версии сайта понадобится организовать отправку сообщений с заявками на почту компании. Для этого придётся каким-то образом общаться с серверным языком программирования PHP. Тут поможет AJAX подход с JSON.

JSON – текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. Формат считается независимым от языка и может использоваться практически с любым языком программирования [8].

При этом будет использован AJAX, который является подходом к построению интерактивных пользовательских веб-приложений. Его основной принцип заключается в «фоновом» обмене данными браузера с веб-сервером.

При помощи AJAX и JSON формируется через JavaScript объект с данными о заявке, содержащий всю информацию, которую пользователь ввёл в интерактивных полях.

Также в проекте используется API FETCH. Он позволяет делать запросы, схожие с XMLHttpRequest (XHR). Основное отличие заключается в том, что Fetch API использует Promises (Обещания), которые позволяют использовать более простое и чистое API, избегая катастрофического количества callback'ов [9]. По своей сути является аналогом AJAX.

После чего клиент отправляет данный запрос, где запрос ожидает PHP-скрипт, обрабатывающий этот запрос, и, используя библиотеку и SMTP-протокол, отправляет сформированное сообщение на почту компании.

На рисунке 4 изображена схема взаимодействия сервера и клиента с использованием JSON.

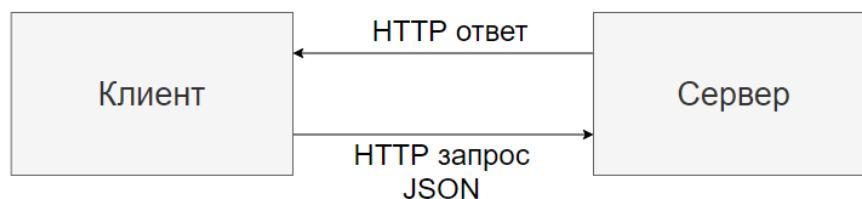


Рисунок 4 – Схема обмена данными клиента и сервера

5.3 Проектирование админ-панели сайта

Для создания админ-панели будем использовать PHP, в котором опишем структуру веб-элементов, стили и логику поведения. На этой странице нужно отображение и редактирование ранее сохраненных пользовательских данных, а также их обработка и вывод статистики по заказам. Для решения поставленной задачи отлично подойдет библиотека `mysqli`, которая позволит установить соединение с базой данных и даст возможность отправлять запросы и получать данные из таблиц.

Также для формирования статистических данных используется API – Google Charts.

Весь функционал будет представлен в одном файле, так как язык PHP может позволить использовать себя в качестве веб-страницы, а также чтобы не усложнять структуру админ-панели.

По сути будет отправляться запрос на выполнение функции и страница будет перезагружаться с уже внесенными в нее изменениями, что очень удобно, и не придется использовать сторонние библиотеки или фреймворки.

5.4 Структура базы данных

Для хранения данных об активности пользователей будет использоваться БД, которая содержит 4 таблицы:

- Users;
- comments;

- visitors;
- orders.

Общая структура БД представлена на рисунке 5.

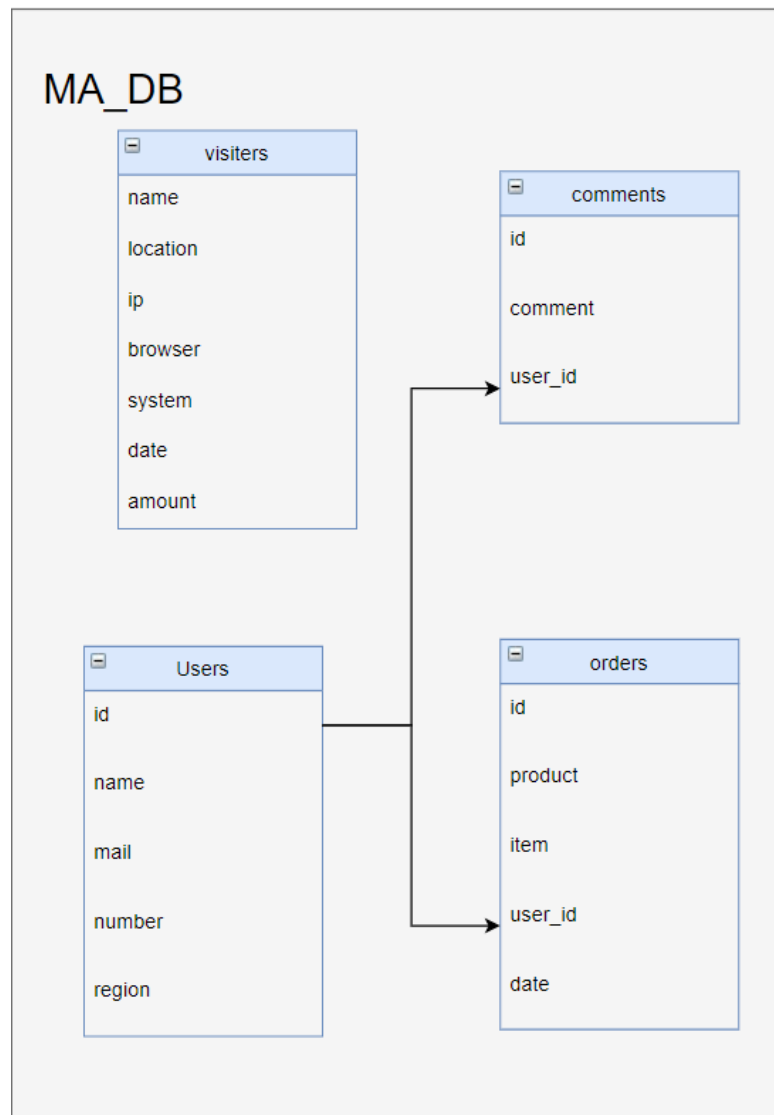


Рисунок 5 – Структура БД для сайта

Таблица Users имеет 4 поля:

- id – уникальный идентификатор записи;
- name – имя пользователя;
- mail – почтовый адрес пользователя;
- number – номер телефона пользователя.

Данная таблица используется для записи каждого пользователя, когда те совершают действия, связанные с отправкой заявки, или обращением на почту через сайт.

Таблица comments имеет 3 поля:

- id – уникальный идентификатор записи;
- comment – сообщение пользователя;
- user_id – связывающий идентификатор пользователя.

Эта таблица используется для записи каждого сообщения, когда пользователи отправляют сообщение через функционал сайта.

Таблица orders имеет 5 полей:

- id – уникальный идентификатор записи;
- product – позиция, которая была заказана;
- item – модель оборудования;
- date – дата отправки заявки;
- user_id – связывающий идентификатор пользователя.

Таблица orders используется для хранения данных при создании заявки на оборудование, они отправляются на почту, а также фиксируются в базе данных.

Таблица visitors имеет 7 полей:

- name – уникальное сгенерированное имя посетителя;
- location – геолокация посетителя;
- ip – ip-адрес посетителя;
- browser – наименование используемого браузера;
- system – операционная система пользователя;
- date – дата посещения;
- amount – количество посещений.

Таблица visitors используется для хранения данных при посещении незарегистрированных пользователей, в дальнейшем эти данные будут использоваться для написания статистических модулей.

6 РЕАЛИЗАЦИЯ

6.1 Верстка сайта

Верстка сайта происходила в html-документах, где структурированно размещались веб-элементы, которые впоследствии отображаются на странице. Туда же помещался контент сайта (текст, ссылки, изображения). Далее элементам прописывались стилистические классы, которые отсылали к файлам CSS, где были описаны основные стили для каждого веб-элемента страницы.

В результате верстки были созданы следующие веб-страницы:

- Главная;
- Производство;
- Заказать;
- Категория;
- Товары;
- Пищевое/технологическое оборудование;
- Админ-панель;
- Страница товаров.

Основой сайта является главная страница, которая служит лицом компании в интернет среде. Именно на неё в первую очередь попадает пользователь и она должна направить посетителя в нужном направлении.

Вместо серого цвета элементов и статического поведения – была выбрана палитра цветов, соответствующая логотипу компании, а также использовалось динамическое отображение для различных элементов, которая позволяет разнообразить поведение веб-элементов.

Главная страница сайта содержит в себе основные веб-элементы для навигации и информирования пользователя:

- Каталог
- Быстрые ссылки
- Информационная панель
- Контактная панель

Данная веб-страница изображена на рисунке 6.



Рисунок 6 – Главная страница сайта

На всех страницах имеется шаблонная шапка сайта, которая содержит контактную информацию и навигационные кнопки, данная панель изображена на рисунке 7.



Рисунок 7 – Навигационная панель

Для более отзывчивого интерфейса, при выборе какого-либо пункта навигационного меню этот пункт окрашивается в красный цвет.

HTML-код навигационного меню представлен на рисунке 8.

```
<div class="menu">
  <div class="element">
    
    <a href="/index.html"> главная </a>
  </div>

  <div class="element">
    

    <a href="/index.html#catalog">каталог</a>
  </div>

  <div class="element">
    
    <a href="/index.html#form">контакты</a>
  </div>

  <div class="element">
    
    <a href="/pages/order.html">заказать</a>
  </div>

  <div class="element">
    
    <a href="/pages/proizvodstvo.html">производство</a>
  </div>
</div>
```

Рисунок 8 – HTML-Код навигационного меню

Как видно из кода, каждый веб-элемент пункта содержит картинку для более простой ассоциации с названием раздела, и само название раздела, прописанного в теге «а», который исполняет функцию пересылки при нажатии на элемент.

Также на главной странице расположен блок каталога, который представлен на рисунке 9. Там располагаются основные категории товаров, и при нажатии пользователя перенаправляет на страницу с более подробным описанием каждой позиции категории.

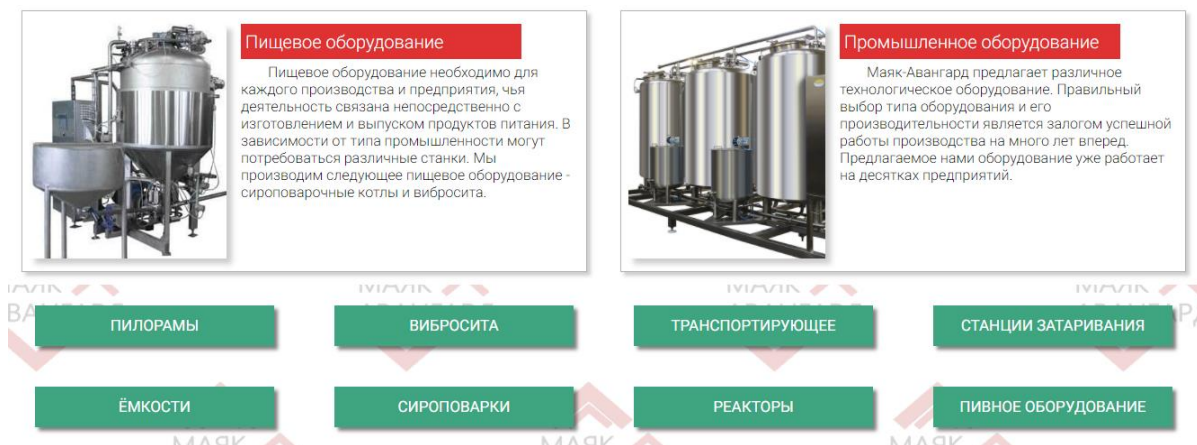


Рисунок 9 – Блок каталога

На рисунке 10 представлен код блока пищевой категории из каталога, который состоит из элемента названия, картинки оборудования в теге «img» и описания.

```
<div class="pos" id="pishevoe">
  <div
    class="photo"
    style="
      background-image: url(/assets/previews/pishevoe.jpg);
    "
  ></div>
  <div class="info">
    <div class="name">
      <p>Пищевое оборудование</p>
    </div>

    <div class="desc">
      Пищевое оборудование необходимо для каждого
      производства и предприятия, чья деятельность связана
      непосредственно с изготовлением и выпуском продуктов
      питания. В зависимости от типа промышленности могут
      потребоваться различные станки. Мы производим
      следующее пищевое оборудование - сироповарочные
      котлы и вибросита.
    </div>
  </div>
</div>
```

Рисунок 10 – HTML-код одной категории

На странице «Производство» расположена основная информация о компании и фотографии из цехов, где отображается профессиональная работа при создании и транспортировке различного оборудования. Данная страница представлена на рисунке 11.

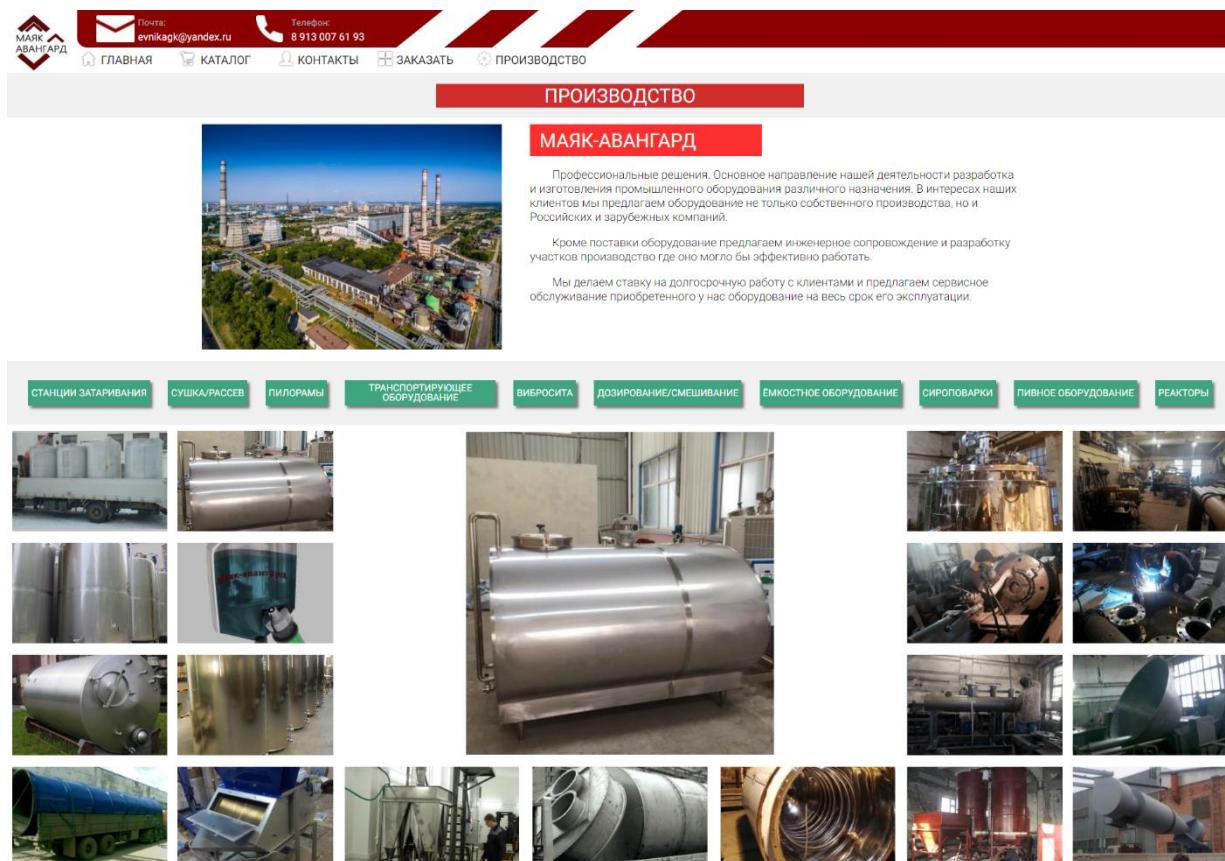


Рисунок 11 – Страница «Производство»

В данном разделе представлены медиа-материалы, которые помогают понять уровень компании и сформировать доверие клиента к компании.

В коде этой странице в основном используется тег «div» с параметром стиля «background-image» для отображения изображений на странице. Часть кода представлена на рисунке 12.

```
<div class="block_2x4">
  <div class="block">
    <div class="photo" style="background-image: url(/assets/gallery/new_gal9.jpeg);">
    </div>
    <div class="photo" style="background-image: url(/assets/gallery/Emk2.jpg);">
    </div>
  </div>
</div>
```

Рисунок 12 – HTML-код изображений на странице «Производство»

Страница «Заказать» представлена на рисунке 13. Она отвечает за составление заявки на оборудование по индивидуальным параметрам. На ней отображены элементы для выбора категории оборудования, а также различные поля для ввода информации.

Рисунок 13 – Страница «Заказать»

HTML-код со страницы «Заказать» изображен на рисунке 14, он отвечает за отображение вариантов категорий, которые доступны для индивидуального заказа.

```
<div class="order_line">
  <div class="infoBlock">

<div class="option" id="vibro">Вибросито

</div>
<div class="option" id="emkost">Ёмкость

</div>
<div class="option" id="dozat">Дозатор

</div>
<div class="option" id="konvey">Конвейер

</div>
<div class="option" id="diss_reakt">Диссольтвер/Реактор

</div>

<div class="text">
```

Рисунок 14 – HTML-код выбора категории

Ниже на рисунке 15 изображена страница «Технологическое оборудование», на ней отображается весь список оборудования для данной категории, и при нажатии на одну из них пользователь перенаправляется на страницу выбранного им оборудования.



Рисунок 15 – HTML-код выбора категории

HTML-код для отображения карточки категории изображен на рисунке 16.

```
<div class="card">
  <div
    class="photo"
    style="background-image: url(/assets/previews/zatar.png)"
  ></div>

  <div class="name">
    <p style="height: 46px;">станции затаривания</p>
  </div>

  <div class="desc">
    <div class="text">
      Станция затаривания предназначена для фасовки в мягкие контейнеры типа "Биг-Бэк" сыпучих
    </div>
  </div>
</div>
```

Рисунок 16 – HTML-код отображения одной категории

Основной страницей, которая отображает суть производства компании, является страница какой-либо категории, которая отображает основную информацию по каждой позиции товара, а также дает возможность сразу же оформить заявку на выбранное оборудование.

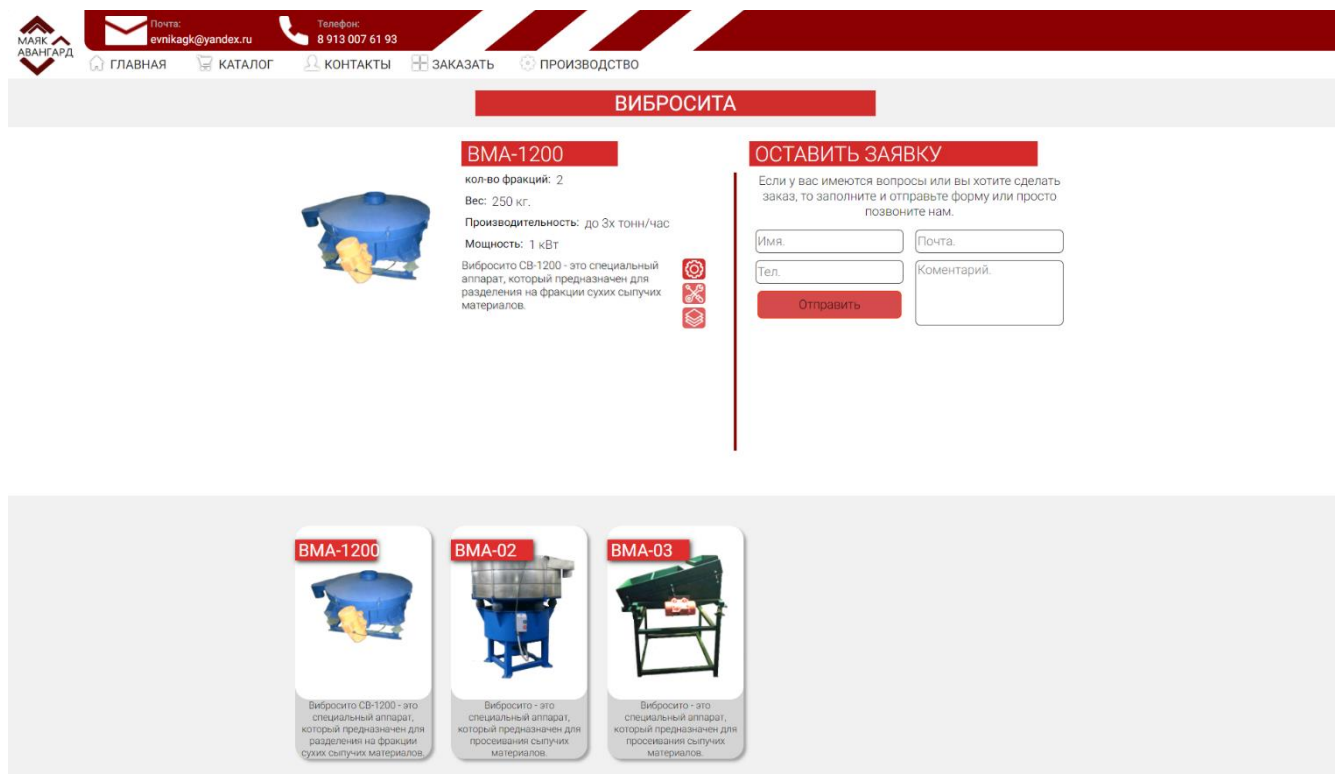


Рисунок 17 – HTML-код выбора товара

HTML-код веб-элемента для отображения характеристик по выбранному оборудованию представлен на рисунке 18.

```
<div class="info">
  <div class="title" id="title">СМА-01</div>

  <div class="props">
    <div class="property">кол-во фракций:</div>
    <div class="text" id="size"></div>
  </div>
  <div class="props">
    <div class="property">Вес:</div>
    <div class="text" id="mass"></div>
  </div>
  <div class="props">
    <div class="property">Производительность:</div>
    <div class="text" id="impact"></div>
  </div>
  <div class="props">
    <div class="property">Мощность:</div>
    <div class="text" id="moshnost"></div>
  </div>
</div>
```

Рисунок 18 – HTML-код шаблона выбранной позиции

Смена отображаемых характеристик происходит по нажатию на другую позицию. Код для смены этих данных представлен на рисунке 19.

```
function setStart() {
    products = document.querySelectorAll(".card");

    title = document.getElementById("title");
    size = document.getElementById("size");
    mass = document.getElementById("mass");
    impact = document.getElementById("impact");
    moshnost = document.getElementById("moshnost");
    desc = document.getElementById("desc");
    photo = document.getElementById("photo");

    if (products[0] != null) {
        card = products[0];

        haract_text = card.childNodes[3].textContent;
        dop_text = card.dataset.dop;
        osob_text = card.dataset.osob;

        title.textContent = card.childNodes[1].textContent;
        size.textContent = card.dataset.size;
        mass.textContent = card.dataset.mass;
        impact.textContent = card.dataset.impact;
        moshnost.textContent = card.dataset.moshnost;
        desc.textContent = card.childNodes[3].textContent;
        photo.style.backgroundImage = card.childNodes[1].style.backgroundImage;
    } else {
    }
}
```

Рисунок 19 – Код для смены товара на панели характеристик

Для просмотра информации о товаре была написана система получения информации из свойств веб-элементов, которые содержали в себе характеристики, изображение и описание товара. Такой веб-элемент изображен на рисунке 20.

```
<div
  class="card"
  data-size="под заказ"
  data-mass="300 кг."
  data-impact="18 об/мин"
  data-moshnost="под заказ"
  data-osob="Основное предназначение реакторов – выработка конечного продукта из исходных компонентов. Основной материал изготовления реактора это нержавеющий металл.
  data-dop="экономичность и высокая надежность в работе;
  простота освоения и технического обслуживания;
  высокая производительность и универсальность к затариваемому материалу;
  использование надежных комплектующих импортного и отечественного производства:"
>
  <div
    class="photo"
    style="
      background-image: url(/assets/products/reaktor/RKMA-01.png);
    "
  >
  <div class="name">РКМА-01</div>
</div>
```

Рисунок 20 – Веб-элемент одной из позиций категории «Реактор»

Чтобы получать данные о посещениях и статистику по сайту была сверстана админ-панель, которая состоит из панели управления с кнопками различного функционала, полем параметров и рабочим окном, где отображаются запрошенные данные. Данная страница представлена на рисунке 21.

Номер телефона	Пользователи	Заказы	Статистика			
Посетителей:	22					
Заказов:	1					
Посещений:	59					
Конверсия:	0.045454545454545					
№:	ИМЯ:	РЕГИОН:	Браузер:	Система:	ДАТА:	Посещений:
number	mustang_nokia_34	Novosibirsk Oblast	Safari	iPhone,	2021-03-21	2
number	mazda_nokia_56	Novosibirsk Oblast	Chrome	Linux,	2021-03-21	2
number	acer_tomato_16	Novosibirsk Oblast	Chrome	Windows	2021-03-21	2
number	hyper_onion_71	Novosibirsk Oblast	Chrome	Windows	2021-03-21	1
number	fox_beans_18	Novosibirsk Oblast	Safari	iPhone,	2021-03-21	1
number	lamborghini_pepper_11	Unknown	Safari	iPhone,	2021-03-21	3
number	doge_pumpkin_33	Novosibirsk Oblast	Safari	Windows	2021-03-21	6
number	intel_ig_79	Novosibirsk Oblast	Safari	Windows	2021-03-21	1
number	aoc_nokia_3	Novosibirsk Oblast	Safari	Windows	2021-03-21	16
number	mustang_dodge_72	Novosibirsk Oblast	Chrome	Windows	2021-03-21	6
number	toyota_folk_52	Novosibirsk Oblast	Chrome	Windows	2021-03-21	1

Рисунок 21 – Веб-элемент одной из позиций категории «Реактор»

Код, отвечающий за отрисовку функционального меню, представлен на рисунке 22.

```

<body>
  <div class="menu">
    <form method="POST">
      <input class="textLabel" type="text" name="option" placeholder="Номер телефона">
      <input type="submit" name="getUsers" value="Пользователи">
      <input type="submit" name="getOrders" value="Заказы">
      <input type="submit" name="getStatistics" value="Статистика">
    </form>
  </div>

```

Рисунок 22 – Код для отображения функционального меню

Для реализации окна данных был использован серверный язык программирования PHP. В коде реализованы модульные функции данных элементов, которые получают данные только из базы данных и пространство имен столбцов, помещая их в массив. Данные функции путем перебора

информации из массива – модульно рисуют каждый элемент таблицы. Код данной реализации представлен на рисунке 23.

```
function drawTable_Orders($col_names, $rows)
{
    echo "<div style=\" display: flex; width: 100%; margin:7px;";
    echo drawCol_name("№");
    foreach ($col_names as $col_name) {
        drawCol_name($col_name);
    }
    echo "</div>";

    foreach ($rows as $row) {
        drawRow_Orders($row);
    };
}

function drawCol_name($name)
{
    echo "<p style=\" margin-left: 10px; width: 150px;\">";
    " . $name . ":";
    "</p>";
}

function drawRow_Orders($rows)
{
    echo "<div style=\"display: flex; width: 100%; margin:7px;";

    foreach ($rows as $row) {
        echo drawRow_var($row);
    };
    echo "
    </div>";
}
```

Рисунок 23 – Код модульных функций элементов таблицы

6.2 Реализация клиентской части сайта

На веб-сервисе расположены блоки с отправкой форм на почту компании, где предоставляется возможность заполнить личные данные и комментарий к запросу, после чего будет выполнена отсылка данных в РНР-скрип посредством

POST-метода, затем скрипт сформирует объект, содержащий HTML-код с данными клиента и отправит его на почту компании.

На рисунке 24 представлен JavaScript-код, который формирует объект для отправки. Он состоит из объекта, который забирает данные из веб-элементов, используя DOM-модель страницы, после чего помещает их в объект отправки, и производит отправку по адресу /php/mail.php. Также тут присутствует проверка на завершенность, и, если она не проходит, то пользователь видит окно ошибки, которое сигнализирует о неисправности при отправке.

```
document.querySelector('#form').addEventListener('submit', async function(event) {  
    event.preventDefault();  
  
    const formData = Object.fromEntries(new FormData(event.target).entries());  
    const {message, mail, phone, name} = formData;  
    const body = `name=${name}&mail=${mail}&phone=${phone}&message=${message}`;  
  
    url = '/php/mail.php';  
    options = {  
        method : 'POST',  
        body,  
        headers: {  
            "Content-type": "application/x-www-form-urlencoded; charset=UTF-8"  
        }  
    };  
  
    const response = await fetch(url, options);  
    const status = await response.text();  
  
    if (+status === 1) {  
        alert('Заявка успешно отправлена!');  
        document.querySelectorAll('.input').forEach(input => input.value = '');  
        document.querySelector('textarea').value = '';  
    }  
    else alert('Произошла ошибка, пожалуйста попробуйте позже');  
});
```

Рисунок 24 – Скрипт, отвечающий за отправку объекта PHP-скрипту

Похожий скрипт используется при отправке заявки на определённое оборудование со страницы категории, представленной на рисунке 17. Тут он не просто формирует объект на основе данных пользователя, но также присоединяет данные о выбранном товаре (название, категория, картинка), чтобы человеку, просматривающему эту заявку, было сразу понятно, на какой

товар она была отправлена. После формирования, информации отсылается на корпоративную почту. Код скрипта изображен на рисунке 25.

```
function sendForm_test() {  
    type = "2";  
    document  
        .querySelector("#form")  
        .addEventListener("submit", async function (event) {  
            category = document.querySelectorAll(".title")[2].textContent;  
            title = document.getElementById("title").textContent;  
  
            console.log(document.getElementById("photo").style.backgroundImage.replace(/(url\(|\)|")/g, ''));  
            event.preventDefault();  
  
            site = " http://mayak-avangard.ru" + document.getElementById("photo").style.backgroundImage.replace(/(url\(|\)|")/g, '' );  
            const formData = new FormData();  
            formData.append('type', type.toString());  
            formData.append('category', category.toString());  
            formData.append('title', title.toString());  
            formData.append('message', document.getElementById("comment").value );  
            formData.append('mail', document.querySelectorAll('.input')[2].value);  
            formData.append('phone', document.querySelectorAll('.input')[1].value);  
            formData.append('name', document.querySelectorAll('.input')[0].value );  
            formData.append('img', site );  
        })  
    ;  
}
```

Рисунок 25 – Код скрипта, отвечающий за формирование объекта со страницы «Категория».

6.3 Реализация серверной части сайта

На рисунке 26 изображена функция принятия данных, формирования сообщения и отправки его на почту. Задачей данного скрипта является отправка сообщения на почтовый адрес компании, а также занесения личных данных пользователя в базу данных. Помимо всего перечисленного туда помещаются данные о каждой заявке и каждом заказе с сайта.

```
if(isset($phone) && $phone != ''){  
    $from = "MayakAvangard";  
    $headers = "From: $from\r\nContent-type: text/html; charset=utf-8\r\n";  
  
    if (mail($to, $subject, $message, $headers)) {  
  
        $con = mysqli_connect("localhost","a316809_1","mayakavangard","a316809_1");  
        $result = mysqli_query($con , "insert into users(name,mail,number) values ('$name','$mail','$phone')");  
        $get_id = mysqli_query($con , "select id from users where number = '$phone'");  
  
        while ($row = $get_id->fetch_assoc()) {  
            $id = $row['id'];  
            $today = date("y-m-j");  
            $result = | mysqli_query($con , "insert into orders(product,item,user_id,message,date) values ('$categ
```

Рисунок 26 – Скрипт, отвечающий за отставку сообщения на почту

Далее идет похожий, но расширенный скрипт на отправку заявки на почту компании. Он изображен на рисунке 27. Данный код примечателен тем, что тут формируется HTML-код в сообщении, а не записывается обычный текст.

```
$type_str = "<strong>" . "$type" . "</strong>" . "\r\n\n" . "<br><br>";
$volume_str = "Объем: " . "<strong>" . "$obem" . "</strong>" . "\r\n\n" . "<br>";
$material_str = "Материал: " . "<strong>" . "$metal" . "</strong>" . "\r\n\n" . "<br>";
$form_str = "Форма: " . "<strong>" . "$form" . "</strong>" . "\r\n\n" . "<br>";
$extra_str = "Дополнительно: " . "<strong>" . "$dops" . "</strong>" . "\r\n\n" . "<br>";
$forWhat_str = "Предназначение: " . "<strong>" . "$goal" . "</strong>" . "\r\n\n" . "<br>";
$verthorizont_str = "Положение: " . "<strong>" . "$category" . "</strong>" . "\r\n\n" . "<br>";
$dno_str = "Днище: " . "<strong>" . "$dnishe" . "</strong>" . "\r\n\n" . "<br>";

//Вибросити формирование сообщения
if ($type == "Вибросито") {
    $message =
    "Новая заявка на " . $type_str . "\r\n" . "<br>"
    . "Email заказчика: " . "<strong>" . "$mail" . "</strong>" . "\r\n" . "<br>"
    . "Телефон заказчика: " . "<strong>" . "$phone" . "</strong>" . "\r\n" . "<br>"
    . "\r\n\n" . "<br>"
    . $volume_str
    . $numDek_str
    . $fraktionSize_str
    . $forWhat_str
    . $extra_str;
} ;
```

Рисунок 28 – Скрипт, отвечающий за отправку заявки на почту

Также теперь на сайте присутствует сборщик данных посетителей для расчета статистики посещений и продаж. Этим занимается скрипт из файла dataCollector.php, в нём описаны основные функции, которые разбирают и собирают следующие данные:

- IP-адрес пользователя;
- Регион проживания;
- Браузер;
- Операционная система;
- Дата посещения;
- Количество посещений.

Основные данные предоставляет переменная окружения HTTP_USER_AGENT, которая содержит информацию о браузере, системе и проч. А для определения геолокации приходится получать IP-адрес пользователя

и «прогонять» его через API сайта <http://www.ip-api.com>. Данный процесс изображен на рисунке 29.

```
function getUser_location()
{
    $ip = getUser_ip();
    $geo = unserialize(file_get_contents('http://ip-api.com/php/' . $ip . '?lang=ru'));
    $city = $geo['city'];
    $region = $geo['regionName'];
    $string = "$city $region";
    if ($string == "") {
        $string = "Unknown";
    };

    return $string;
}
```

Рисунок 29 – Функция для получения местоположения

Основная функция для получения и обработки данных представлена на рисунке 30.

```
function setUser_data()
{
    $browser = getUser_browser();
    $loc = getUser_location();
    $system = getUser_system();
    $date = date("y-m-j");
    $ip = getUser_ip();
    $con = mysqli_connect("localhost", "a316809_1", "mayakavangard", "a316809_1");

    $amount = 1;
    $name = generateName();
    echo $browser;
    if (checkIp() == "new") {
        $result = mysqli_query($con, "insert into visitors(name,location,ip,browser,system,date,amount) values");
    }else{
        $newAmount = mysqli_query($con, "select amount from visitors where ip = '$ip'");
        $newAmount = $newAmount->fetch_array();
        $newAmount = $newAmount[0] + 1;
        echo $newAmount ;
        $result = mysqli_query($con, "update visitors set amount = '$newAmount' where ip = '$ip'");
    };
};
```

Рисунок 30 – Функция для получения и записи данных о пользователе

6.4 Реализация админ-панели сайта

Для реализации админ-панели использовался серверный скриптовый язык программирования PHP. Сама страница предоставляет несколько функций:

- Отображение пользователей;

- Отображение заказов;
- Отображение статистики;
- Параметризация группировки.

Страница админ-панели представлена на рисунке 31.



Рисунок 31 – Админ-панель

Для отображения всех пользователей, которые отправляли какие-либо данные на почту компании, имеется специальная кнопка, которая выдает список пользователей. Данная страница изображена на рисунке 32.

ИМЯ	ПОЧТА	ТЕЛЕФОН
anton	mail@mail.com	89514875092
sdf	dfgd@mail.ru	55gd5
nikola	nikakola@gmail.com	89546754890
fgd	dfgd@mail.ru	5464564
dfg	dfgd@mail.ru	rttyty
lol	dfgd@mail.ru	657567
lol	dfgd@mail.ru	6575675465
lol	dfgd@mail.ru	6575675465343
lol	dfgd@mail.ru	657567546534311
gvvbn	dfgd@mail.ru	6575
gvvbn	dfgd@mail.ru	6575
gvvbn	dfgd@mail.ru	6575
gvvbn	dfgd@mail.ru	6575
gvvbn	dfgd@mail.ru	6575
gvvbn	dfgd@mail.ru	6575

Рисунок 32 – Страница пользователей

Для отображения используется простой PHP-код, формирующий HTML-элементы в зависимости от информации, которая получена из базы данных. Часть скрипта показана на рисунке 33.

```

    foreach ($rows as $row) {
        echo
        "
        <div style=\"display: flex; width: 100%; margin: 7px; background-color: rgb(200, 200, 200);\" >
        --

        <p style=\"margin-left: 10px; width: 170px;\">
        |   |   |   |   . $row['name'] . "
        |
        </p>

        <p style=\"margin-left: 10px; width: 170px;\">
        |   |   |   |   . $row['mail'] . "
        |
        </p>

        <p style=\"margin-left: 10px; width: 100px;\">
        |   |   |   |   . $row['number'] . "
        |
        </p>
        </div>
        ";
    }

```

Рисунок 33 – Часть скрипта, формирующая элемент с данными

Чтобы получить данные из БД, сначала нужно подключиться к ней и произвести правильный запрос, именно этим занимается часть скрипта, представленного на рисунке 34.

```

$query_option = "SELECT
id,
name,
mail,
number
FROM
users WHERE number = '$option'";

if($option == ""){
    $result = mysqli_query($con, $query);
    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
}else{
    $result = mysqli_query($con, $query_option);
    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
}

```

Рисунок 34 – Часть скрипта, получающая данные из БД

Страница для отображения заказов представлена на рисунке 35. Она формируется по тем же правилам, что и получение пользователей.

ПользователиЗаказыСтатистикаНомер телефон

№	ИМЯ:	ПОЧТА:	ТЕЛЕФОН:	КАТЕГОРИЯ:	ТОВАР:	Сообщение:	Дата(ГГГГ-ММ-ДД):
1	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
2	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
3	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
4	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
5	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
6	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
7	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
8	rtu	rtu	rtu	Станции заправки	СЗМА-01		2010-03-21
9	rtu32423	rtu45654	rtu435	Станции заправки	СЗМА-01		2010-03-21
10	rtu32423	rtu45654	rtu435	Станции заправки	СЗМА-01		2021-03-10

Рисунок 35 – Страница заказов

Данные для формирования элементов страницы также берутся из БД, но текст запроса выглядит немного по-другому. Часть PHP-скрипта, отвечающая за получение данных о заказах, представлена на рисунке 36.

```
$query_option = "SELECT
    name,
    mail,
    number,
    product,
    item,
    message,
    date
FROM
    users INNER JOIN orders
    ON users.id = orders.user_id
    WHERE number = '$option';

if ($option == '') {
    $result = mysqli_query($con, $query_prod);
    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
}
else {
    $result = mysqli_query($con, $query_option);
    $rows = mysqli_fetch_all($result, MYSQLI_ASSOC);
}
```

Рисунок 36 – Часть скрипта, отвечающая за получение данных о заказах

Для получения данных с уточнением используется запрос с параметром, в данном случае справа расположено поле ввода, куда вписывается нужный

телефон, и в админ-панель выводится вся информация по выбранному клиенту. Результат работы функции приведён на рисунке 37.

Пользователи Заказы Статистика 89648765045

№	ИМЯ:	ПОЧТА:	ТЕЛЕФОН:	КАТЕГОРИЯ:	ТОВАР:	Сообщение:	Дата(ГГГГ-ММ-ДД):
1	Антон89648765045	gunger@mail.ru	89648765045	сушка/рассев	СМА-01		2021-03-10
2	Антон89648765045	gunger@mail.ru	89648765045	сушка/рассев	СМА-01		2021-03-11
3	Антон89648765045	gunger@mail.ru	89648765045	сушка/рассев	СМА-01		2021-03-11
4	Антон89648765045	gunger@mail.ru	89648765045	сушка/рассев	СМА-01		2021-03-11
5	Антон89648765045	gunger@mail.ru	89648765045	сушка/рассев	СМА-01		2021-03-11

Рисунок 37 – Результат работы поиска данных по номеру телефона

Страница для отображения статистических данных сайта представлена на рисунке 38.

Номер телефон Пользователи Заказы Статистика						
Посетителей:	18					
Заказов:	1					
Посещений:	54					
Конверсия:	0.0555555555555556					
№:	ИМЯ:	РЕГИОН:	Браузер:	Система:	ДАТА:	Посещений:
number	mustang_nokia_34	Novosibirsk Oblast	Safari	iPhone;	2021-03-21	2
number	mazda_nokia_56	Novosibirsk Oblast	Chrome	Linux;	2021-03-21	2
number	acer_tomato_16	Novosibirsk Oblast	Chrome	Windows	2021-03-21	2
number	hyper_onion_71	Novosibirsk Oblast	Chrome	Windows	2021-03-21	1
number	fox_beans_18	Novosibirsk Oblast	Safari	iPhone;	2021-03-21	1
number	lamborghini_pepper_11	Unknown	Safari	iPhone;	2021-03-21	3
number	doge_pumpkin_33	Novosibirsk Oblast	Safari	Windows	2021-03-21	6
number	intel_lg_79	Novosibirsk Oblast	Safari	Windows	2021-03-21	1
number	aoc_nokia_3	Novosibirsk Oblast	Safari	Windows	2021-03-21	16
number	mustang_dodge_72	Novosibirsk Oblast	Chrome	Windows	2021-03-21	5
number	toyota_folk_52	Novosibirsk Oblast	Chrome	Windows	2021-03-21	1

Рисунок 38 – Страница статистики

Для вывода полной статистики используется большое количество графических методов, но основные данные страница получает при помощи функции getStatistics.

В функции происходит подключение к таблице «visitors» и формирование запросов. После чего в переменные записываются данные и помещаются в ассоциативный массив. В конце концов эти данные отправляются через параметры в графические функции, которые отображают их на странице.

Данная функция написана на PHP и представлена на рисунке 39.

```
function getStatistics($con)
{
    $query = "select name,location,browser,system,date,amount from visitors";

    $getSUsers = mysqli_query($con, "select COUNT(*) as couter from visitors");
    $getSOrders = mysqli_query($con, "select COUNT(*) as couter from orders");
    $getVisits = mysqli_query($con, "select sum(amount) as amount from visitors");

    $Users_amount = mysqli_fetch_array($getSUsers)['couter'];
    $Orders_amount = mysqli_fetch_array($getSOrders)['couter'];
    $Visisters_amount = mysqli_fetch_array($getVisits)['amount'];

    $conversion = $Orders_amount / $Users_amount;
    $visitors = mysqli_query($con, $query);
    $statistics = array(
        "users"=>$Users_amount,
        "visits"=>$Visisters_amount,
        "orders"=>$Orders_amount,
        "conver"=>$conversion
    );

    drawRow_Statistics("Посетителей:",$statistics['users']);
    drawRow_Statistics("Заказов:",$statistics['orders']);
    drawRow_Statistics("Посещений:",$statistics['visits']);
    drawRow_Statistics("Конверсия:",$statistics['conver']);
    drawTable_Visitors($visitors);
}
```

Рисунок 39 – Метод для получения статистики

Описанный ранее графический метод представлен на рисунке 40.

```
function drawTable_Visitors($rows)
{
    echo "<div style=\" display: flex; width: 100%; margin:7px;";
    echo drawCol_name("№") . "
    " . drawCol_name("ИМЯ") . "
    " . drawCol_name("РЕГИОН") . "
    " . drawCol_name("Браузер") . "
    " . drawCol_name("Система") . "
    " . drawCol_name("ДАТА") . "
    " . drawCol_name("Посещений") . "
    </div>";

    foreach ($rows as $row) {
        drawRow_Visitors($row);
    };
}
```

Рисунок 40 – Метод для отображения данных на странице

Для визуализации статистических данных было использовано API от компании Google, которое было интегрировано в PHP-код. Результат работы этого модуля представлен на рисунке 41.

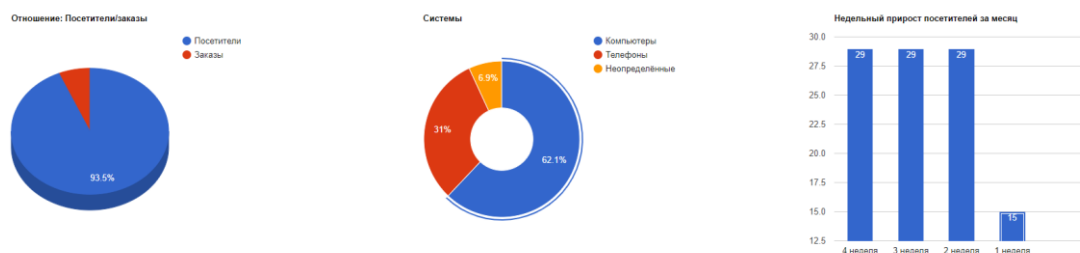


Рисунок 41 – Диаграммы в разделе статистика

Для построения стоит лишь отправить запрос с нашими параметрами из базы данных и получить в ответе готовый график. Код представлен на рисунке 42.

```
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);

function drawChart() {

    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Topping');
    data.addColumn('number', 'Slices');
    data.addRows([";
        foreach($data as $dat => $val){
            echo "[".$dat."', ".$val."],";
        }
        echo "]);

    var options = {'title':'Отношение: Посетители/заказы',
        \is3D\": true,
        \width\":400,
        \height\":300};
    var chart = new google.visualization.PieChart(document.getElementById('graphs'));
    chart.draw(data, options);
}
```

Рисунок 42 – Код для построения круговой диаграммы

7 ТЕСТИРОВАНИЕ И РАЗМЕЩЕНИЕ

7.1 Тестирование взаимодействия БД и сайта

В результате реализации сервиса и админ-панели было проведено тестирование основных функций:

- Запись пользователя;
- Запись информации о заявке;
- Запись информации о заказе;
- Получение данных о пользователях;
- Получение информации о заявках;
- Получение информации о заказах;
- Получение данных по параметрам;
- Запрос статистики.

Заполнение полей контактными данными и комментарием для связи с менеджерами компании изображено на рисунке 43.

СВЯЖИТЕСЬ С НАМИ

Имя.

Почта

Тел.

Директор
Яковлев Максим Сергеевич

Телефон
8 913 007 61 93

Фактический адрес
производства:
Переулок Промышленный
2/1

Отправить

Рисунок 43 – Заполнение полей данными

				id	name	mail	number
<input type="checkbox"/>				77	АНТОН	gunger@mail.tu	89648765045
<input type="checkbox"/>				78	rtyrty	fghvbn	789789789
<input type="checkbox"/>				79	redrt	tyuyt	tyuty
<input type="checkbox"/>				80	redrt56765	tyuyt678	tyuty7897
<input type="checkbox"/>				81	redrt56765	tyuyt678	tyuty7897
<input type="checkbox"/>				82	uiui	ghj	ty
<input type="checkbox"/>				83	uiui566	ghj67u8gh	ty768uyyu
<input type="checkbox"/>				84	АНТОН	gunger@gmail.com	89503456061

☐
 Отметить все
 С отмеченными:

А комментарий помещается в другую таблицу, и можно его наблюдать на рисунке 45 под `id = 53`.

Рисунок 45 – Сообщение пользователя в БД

Пользователи

Заказы

Статистика

89503456061

--

ИМЯ:

ПОЧТА:

ТЕЛЕФОН:

--

Антон

gunger@gmail.com


89503456061

result:

51

На рисунке 47 изображен процесс заполнения полей для отправки заявки на выбранное оборудование.

СТАНЦИИ ЗАТАРИВАНИЯ



РКМА-02

Объем: 1000 л.
Вес: 850 кг.
Мешалка: магнитная
Давление: до 32 бар.

Данный реактор является самой производительной и эффективной моделью. также наши реакторы имеют возможность установки магнитной мешалки.

ОСТАВИТЬ ЗАЯВКУ

Если у вас имеются вопросы или вы хотите сделать заказ, то заполните и отправьте форму или просто позвоните нам.

Можно увеличить объем??

Рисунок 47 – Отправка заявки на выбранный реактор

В базу данных записывается пользователь, а также данные о заявке:

- Категория;
- Товар;
- Клиент;
- Сообщение;
- Дата произведения заявки.

После этого менеджер может получить данные об этом заказе через админ-панель сайта. Достаточно просто ввести номер телефона клиента, и серверная часть сгенерирует запрос и выдаст результат. Данный процесс представлен на рисунке 48.

Пользователи		Заказы	Статистика	89758904758			
№	ИМЯ:	ПОЧТА:	ТЕЛЕФОН:	КАТЕГОРИЯ:	ТОВАР:	Сообщение:	Дата(ГГГГ-ММ-ДД):
1	Максим 89758904758	mksW@gmial.com	89758904758	Станции затаривания	РКМА-02	Можно увеличить объем??	2021-03-18

Рисунок 48 – Поиск данных о заказе по номеру телефона

Также имеется возможность получения статистики за месяц, она показывает количество пользователей, количество заказов и конверсию. Результат выполнения функции можно увидеть на рисунке 49.

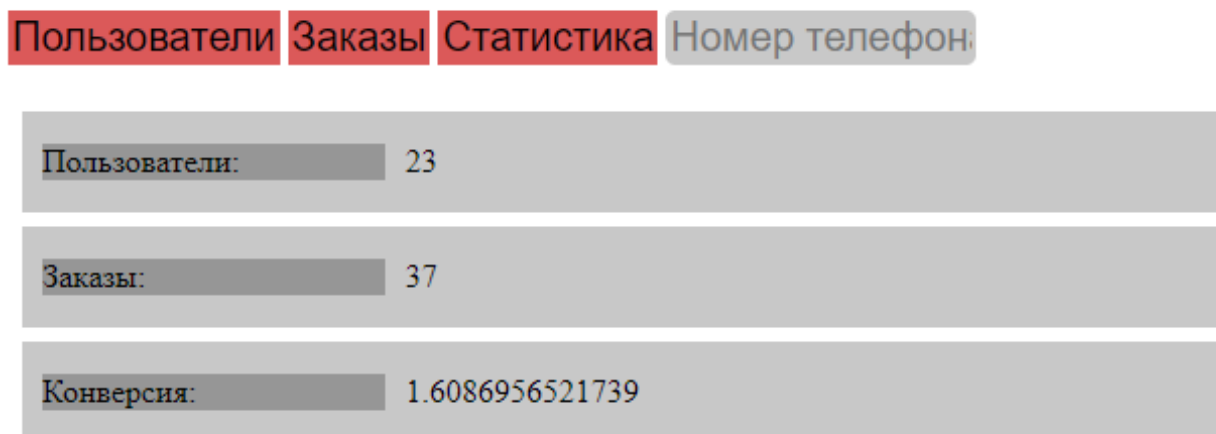


Рисунок 49 – Статистика сайта

7.2 Тестирование SMTP отправки почтовых сообщений

Оставленная ранее заявка с главной страницы – отправляется на почту, и теперь менеджер компании может наблюдать её в новых сообщениях в корпоративной почте. Данное сообщение можно наблюдать на рисунке 50.

Заявка с главной страницы!
Имя: **Антон**
Email: gunger@gmail.com
Телефон: **89503456061**
Сообщение: **Добрый день, связываюсь с вами с целью обсуждения оптового заказа ёмкостного оборудования.**


 Без вирусов. www.avast.ru

Рисунок 50 – Сообщение пользователя

Для отправки использовался протокол SMTP и серверный язык программирования PHP с использованием встроенной функции отправки

почтовых сообщений. А результат в виде сообщения в корпоративной почте представлен на рисунке 51.

Заявка! Категория: **Станции затаривания**
Продукт: **РКМА-02**



Имя: **Максим**
Email: mksW@gmial.com
Телефон: **89758904758**
Сообщение: **Можно увеличить объем??**


 Без вирусов. www.avast.ru

Рисунок 51 – Сообщение о заявке на товар


Как видно из рисунка 51, менеджер получает заявку, которая дает понять, что клиент хочет заказать и какие параметры поменять.

7.3 Тестирование сайта

Для просмотра информации о товаре была написана система получения информации из свойств веб-элементов, которые содержали в себе характеристики, изображение и описание товара.




На странице присутствуют два товара, при нажатии на один из товаров в панели просмотра должно отображаться изображение выбранного товара и данные о нём. Такая панель с информацией представлена на рисунке 52.

СТАНЦИИ ЗАТАРИВАНИЯ



РКМА-01

Объём: под заказ
Вес: 300 кг.
Мешалка: 18 об/мин
Давление: под заказ
Основное предназначение реакторов – выработка конечного продукта из исходных компонентов. Основной материал изготовления реактора это нержавеющая сталь.



ОСТАВИТЬ ЗАЯВКУ

Если у вас имеются вопросы или вы хотите сделать заказ, то заполните и отправьте форму или просто позвоните нам.

Имя.


Почта.

Тел.

Комментарий.


Отправить

РКМА-01



Основное предназначение реакторов – выработка конечного продукта из исходных компонентов. Основной материал изготовления реактора это нержавеющая сталь.

РКМА-02



Данный реактор является самой производительной и эффективной моделью. Также наши реакторы имеют возможность установки магнитной мешалки.

Рисунок 52 – Страница с товарами и информацией о них

За смену этих данных отвечает скрипт, написанный на JavaScript.

Для оформления заказа, который после отправки отправится на корпоративную почту менеджеру, используются поля ввода и выборочное меню.

Сайт предоставляет выбор следующих типов оборудования:

- Вибросито;
- Ёмкость;
- Дозатор;
- Конвейер;
- Диссоolver/Реактор.

Пользователь выбирает нужную ему категорию, а после заполняет предложенную форму. Подобная операция представлена на рисунке 53.

Уведомление от сайта newmayak
Заявка успешно отправлена!

Закреть

Вибросит
Ёмкость
Дозатор
Конвейер
Диссоolver/Реактор

Маяк-авангард предлагает емкости для хранения жидкостей любой вязкости и едкости (Вода, различные реагенты, аммиак и т.д.). Разрабатываемые емкости обеспечивают простоту монтажа, бесперебойную работу и удобство обслуживания. Также используются в многоступенчатых станциях изготавливающих пищевую и парфюмерную продукцию. Цена на все емкости договорная.

Цилиндрическая
34
нержавейка
конусное
подача зерна
gunger@mail.com
86259653265
ручки с боков

Отправить

Рисунок 53 – Страница оформления заявки по параметрам

Для каждого типа оборудования предоставляются индивидуальные параметры для заполнения.

Далее серверный скрипт, написанный на PHP, принимает данные от клиента и на их основе формирует сообщение, а после отправляет его на корпоративную почту. Отправленная заявка отправляется на почту и ожидает прочтения и дальнейшей обработки. Результат работы отправления заявки представлен на рисунке 54.

Новая заявка на **Ёмкость**

Email заказчика: gunger@mail.com

Телефон заказчика: **86259653265**

Положение: **Горизонтальная емкость**

Объем: **34**

Материал: **нержавейка**

Днище: **конусное**

Предназначение: **подача зерна**

Форма: **Цилиндрическая**

Дополнительно: **ручки с боков**

Рисунок 54 – Сообщение о заявке по категории

7.4 Размещение веб-сервиса

7.4.1 Веб-сервер

Одним из важных аспектов веб-программирования является веб-сервер, который отвечает за прием, обработку и отправку HTTP-сообщений.

HTTP – протокол для передачи произвольных данных и является основным средством общения клиент-сервер.

В первую очередь веб-сервер – это компьютерная программа и ее можно установить на любой компьютер. Он общается с клиентом, которым в большинстве случаев является браузер, отправляющий запросы на получение ресурсов с помощью URL-адреса.

- Под ресурсами могут подразумеваться:
- HTML-страницы;
- Изображения;
- Файлы;
- Медиа-потoki.

Соответственно веб-сервер отправляет эти данные клиенту посредством протокола HTTP.

7.4.2 Хостинг

Хостинг – услуга, которая предоставляет пространство для хранения информации на сервере, постоянно работающая и имеющая доступ к сети Интернет. Именно там размещаются файлы HTML, CSS, JS, PHP, БД и другие. Данная услуга является очень важной в среде веб-сайтов, ведь практически весь трафик Интернета обрабатывается и хранится именно на серверах хостинг-компаний, поэтому можно получить доступ к сайтам в любой момент времени.

В качестве хостинга сервиса был выбран продукт компании «Макхост». Он идеально подходит и удовлетворяет всем нужным требованиям для размещения веб-сервиса.

В качестве доменного имени было выбрано имя `mayak-avangard.ru`.

Для полной работы сайта на хостинге была настроена и заполнена база данных [11].

Размещенный веб-сервис компании «Маяк-авангард» изображен на рисунке 55.

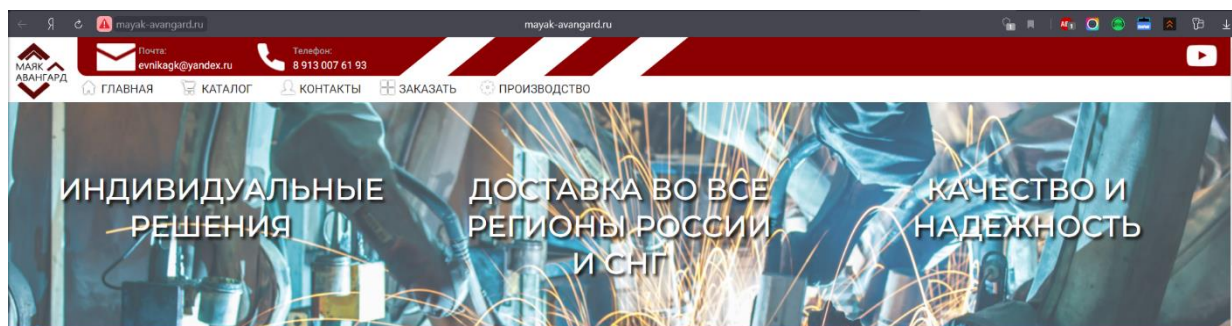


Рисунок 55 – Размещенный сайт

ЗАКЛЮЧЕНИЕ

В соответствии с целью выпускной квалификационной работы бакалавра была произведена модернизация сайта компании «Маяк-авангард» с применением современных инструментов и технологий веб-разработки. Среди прочего были добавлены и протестированы новые функции и страницы, а также админ-панель, позволяющая просматривать и редактировать данные, оставленные действующими и потенциальными клиентами.

Были рассмотрены современные средства разработки и технологии для разработки веб-сервисов и были выбраны средства, наиболее подходящие для модернизации сайта.

В ходе выполнения выпускной работы был выполнен анализ существующей версии веб-сервиса, после была проведена работа по формированию требований к модернизации сайта. С учетом всех сформированных требований были выбраны наиболее подходящие инструменты реализации для разработки веб-сервиса. Для реализации интерфейса клиентской части был выбран язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS, а для реализации функциональной части сайта был выбран скриптовый язык JavaScript. Серверная часть веб-сервиса была выполнена с использованием серверного скриптового языка PHP. В качестве хранилища данных была выбрана СУБД MySQL, под управлением PhpMyAdmin.

Итогом выпускной квалификационной работы является модернизированный веб-сайт компании «Маяк-авангард», который стал более привлекательным по сравнению с предыдущей версией с точки зрения дизайна. Также веб-сервис получил современный функционал, базу данных для хранения информации о клиентах и админ-панель со статистическим модулем сбора и анализа данных о посетителях.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Denisov. Статистика Интернета 2020: сайты, домены, хостинг, трафик [Электронный ресурс] // URL: <https://sdvv.ru/articles/elektronnaya-kommertsiya/statistika-interneta-2020-sayty-domeny-khosting-trafik/> (дата обращения 02.03.2021)
2. HostIQ. Разница между HTTP и HTTPS [Электронный ресурс] // URL: <https://hostiq.ua/wiki/http-https/> (дата обращения 02.03.2021)
3. CoderLessons. Веб-сервисы – Краткое руководство [Электронный ресурс] // URL: <https://coderlessons.com/tutorials/veb-razrabotka/izuchite-veb-servisy/veb-servisy-kratkoe-rukovodstvo> (дата обращения 02.03.2021)
4. Lolz. Тема, которая поможет тебе выбрать язык программирования [Электронный ресурс] // URL: [https://lolz.guru/threads/1675763/#:~:text=1-.%20JavaScript%20Мультипарадигменный%20язык%20программирования.,реализацией%20языка%20ECMAScript%20\(стандарт%20ECMA-262\)](https://lolz.guru/threads/1675763/#:~:text=1-.%20JavaScript%20Мультипарадигменный%20язык%20программирования.,реализацией%20языка%20ECMAScript%20(стандарт%20ECMA-262)) (дата обращения 02.03.2021)
5. Tproger. Простыми словами о «фронтенде» и «бэкенде»: что это такое и как они взаимодействуют [Электронный ресурс] // URL: <https://tproger.ru/translations/frontend-backend-interaction/> (дата обращения 02.03.2021)
6. Otus. Взаимодействие backend и frontend [Электронный ресурс] // URL: <https://otus.ru/nest/post/946/> (дата обращения 02.03.2021)
7. Комьюнити. Что такое MySQL-сервер [Электронный ресурс] // URL: <https://timeweb.com/ru/community/articles/chto-takoe-mysql-server> (дата обращения 02.03.2021)
8. Википедия. JSON [Электронный ресурс] // URL: <https://ru.wikipedia.org/wiki/JSON> (дата обращения 02.03.2021)
9. Хабр. Введение в fetch [Электронный ресурс] // URL: <https://habr.com/ru/post/252941/> (дата обращения 02.03.2021)

10. Хабр. Самые популярные языки программирования. Хабратоп 2020 года [Электронный ресурс] // URL:
<https://habr.com/ru/company/skillfactory/blog/531360/> (дата обращения 02.03.2021)
11. Mchost. Виртуальный хостинг [Электронный ресурс] // URL:
<https://mchost.ru/services/virtual-hosting/> (дата обращения 15.03.2021)
12. Wikipedia. HTML [Электронный ресурс] // URL:
<https://ru.wikipedia.org/wiki/HTML> (дата обращения 17.03.2021)
13. Wikipedia. CSS [Электронный ресурс] // URL:
<https://ru.wikipedia.org/wiki/CSS> (дата обращения 17.03.2021)
14. Wikipedia. JavaScript [Электронный ресурс] // URL:
<https://ru.wikipedia.org/wiki/JavaScript> (дата обращения 17.03.2021)
15. Wikipedia. PHP [Электронный ресурс] // URL:
<https://ru.wikipedia.org/wiki/PHP> (дата обращения 17.03.2021)

ПРИЛОЖЕНИЕ

Исходные коды разработанного веб-сервиса по продаже оборудования приведены на приложенном диске. Кроме того, коды доступны по адресу: <https://github.com/gunger28/newAvangard>.