**Проект**: сайт для компании, по металлообработке.

Страница проекта: https://polifortmash.ru

В ходе разработки проекта я использовал множество современных технологий и инструментов, которые позволили мне расширить свои знания и навыки в области веб-разработки. Основная цель проекта заключалась в создании полнофункционального веб-приложения с современным интерфейсом и эффективной производительностью. Проект охватывает как фронтенд, так и бэкенд аспекты, и включает в себя клиент-серверные взаимодействия, обработку и оптимизацию данных, а также использование лучших практик для стилизации и форматирования кода.

Я научился эффективно использовать компонентный подход, который позволяет создавать повторно используемые элементы интерфейса и улучшает структуру кода.

Основной стек технологий:

PHP, Laravel, Vue 3, Typescript, Vite, SCSS, Bootstrap, Tailwind, Axios, Prettier, Autoprefixer, ES Lint. IDE: VS Code.

**Laravel**: PHP-фреймворк с открытым исходным кодом для разработки веб-приложений. Он обеспечивает мощные инструменты для работы с базами данных, маршрутизацией, аутентификацией и многими другими задачами.

**Typescript**: Статическая типизация предотвращает множество ошибок на этапе компиляции, поддержка современных возможностей JavaScript, улучшенная интеграция с IDE.

**Bootstrap**: Быстрое создание прототипов, кроссбраузерная совместимость, поддержка адаптивного дизайна, большой набор готовых компонентов.

**SCSS**: Поддержка переменных и функций, упрощение написания сложных стилей, улучшенная структура CSS-кода.

**Vue**: Легкость в освоении, высокая производительность, поддержка компонентного подхода, гибкость и расширяемость.

**Vite**: современный инструмент сборки для фронтенд-разработки, который обеспечивает быструю перезагрузку страниц и минимальную конфигурацию. Он поддерживает ES-модули и современный синтаксис JavaScript.

**Sharp**: это высокопроизводительная библиотека для обработки изображений в Node.js. Она поддерживает изменение размера, сжатие, конвертацию форматов и другие операции.

**Prettier:** это форматировщик кода, который обеспечивает единообразие стиля кода в проекте. Он автоматически форматирует код в соответствии с заданными правилами.

**Autoprefixer**: это плагин для PostCSS, который автоматически добавляет вендорные префиксы к CSS-коду, обеспечивая кроссбраузерную совместимость.

**ESLint:** ESLint помогает соблюдать стандарты кодирования, определенные в проекте, что делает код более читабельным и поддерживаемым.

**GIT:** распределенная система контроля версий, которая позволяет отслеживать изменения в коде, работать с несколькими ветками и безопасно интегрировать изменения, внесенные разными разработчиками.

**Оптимизация производительности**

Была произведена оптимизация производительности проекта, в результате оригинальные большие изображения были подогнаны под соответствующие размеры и были сжаты без потери качества. Так же был заменён формат с jpg на webp, в результате проделанной работы удалось уменьшить размер главной страницы с 12.2 мб до 3.8 мб. Так же была добавлена ленивая загрузка изображений, скорость загрузки страницы резко увеличилось на мобильных устройствах.

**SEO Оптимизация**

В ходе разработки проекта я также уделил значительное внимание SEO оптимизации, чтобы обеспечить лучшее ранжирование сайта в поисковых системах и повысить его видимость для потенциальных пользователей.

**Оптимизация через Яндекс.Wordstat**

Одним из ключевых этапов оптимизации стало использование сервиса Яндекс.Wordstat для анализа ключевых слов. Этот инструмент позволил мне:

**Анализировать популярные запросы**: Я изучил, какие запросы наиболее часто используют пользователи, интересующиеся услугами, предлагаемыми на сайте.

**Подбор ключевых слов**: На основе анализа я выбрал наиболее релевантные ключевые слова и фразы, которые помогли улучшить релевантность содержимого сайта запросам пользователей.

**Анализ конкурентов**: Используя Wordstat, я также провел анализ конкурентов, чтобы понять, какие ключевые слова они используют для продвижения своих сайтов.

В результате я смог улучшить структуру контента на сайте, сделать его более привлекательным для поисковых систем и увеличить шансы на высокие позиции в выдаче.

**Автоматическое развёртывание на хостинге**

В проекте было добавлено автоматическое развёртывание на хостинг через FTP с помощью скрипта на Node.js. Это решение обеспечивает удобство и скорость публикации изменений, так как все файлы автоматически загружаются на сервер после сборки. Процесс значительно ускоряется за счёт отправки только новых и изменённых файлов. Автоматизация развёртывания снижает риск ошибок и упрощает управление приложением.

**Заключение**

Проект дал мне возможность глубже погрузиться в современные технологии веб-разработки и расширить свои знания в различных областях. Использование этих технологий позволило мне создать полнофункциональное, эффективное и современное веб-приложение, которое отвечает всем требованиям текущего проекта.