**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**(УНИВЕРСИТЕТ ИТМО)**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Монченко Артем Сергеевич | | | | | | |
| (Фамилия, И., О.) | | | | | | | | |
| Факультет | ПИиКТ | | | | | | Группа | P41071 |
| Направление (специальность) | | | | | | 09.04.04 Программная инженерия | | |
| Руководитель | | | | Государев И. Б., преподаватель | | | | |
| (Фамилия, И., О., должность,) | | | | | | | | |
|  | | | | | | | | |
| Дисциплина | | | Проектирование и анализ языков веб-решений | | | | | |
|  | | | | | | | | |
| Наименование темы: | | | | | Анализ работы технического SEO в React и Next.js | | | |
| |  | | --- | | (Наименование сайта) | | | | | | | | | |
|  | | | | | | | | |

Задание. Провести сравнительный анализ работы технического SEO в React и Next.js

Краткие методические указания (задачи работы)

1. Исследовать документацию фреймворков и понятия SEO.  
2. Реализовать проекты для проведения анализа.  
3. Проанализировать работу SEO для первого проекта.

4. Проанализировать работу SEO для второго проекта.

5. Исследовать полученные результаты.

Содержание пояснительной записки

Оглавление. Введение. Теоретическая часть – введение определений, используемых по ходу выполнения исследования. Реализация приложений – описание методов и подходов, использованных для написания приложения. Ход исследования – описание действий, используемых методов, задействованных ресурсов. Выводы. Заключение. Список использованной литературы.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Б. Государев   
 (подпись)

Студент  А.С. Монченко  
 (подпись) (Фамилия И.О.)

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**(УНИВЕРСИТЕТ ИТМО)**

**ГРАФИК КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Монченко Артем Сергеевич | | | | | | |
| (Фамилия, И., О.) | | | | | | | | |
| Факультет | ПИиКТ | | | | | | Группа | P41071 |
| Направление (специальность) | | | | | | 09.04.04 Программная инженерия | | |
| Руководитель | | | | Государев И. Б., преподаватель | | | | |
| (Фамилия, И., О., должность) | | | | | | | | |
|  | | | | | | | | |
| Дисциплина | | | Проектирование и анализ языков веб-решений | | | | | |
|  | | | | | | | | |
| Наименование темы: | | | | | Анализ работы технического SEO в React и Next.js | | | |
| |  | | --- | | (Наименование сайта) | | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование этапа | Дата завершения | | Оценка и подпись руководителя |
| Планируемая | Фактическая |
| 1 | Исследование документации React | 08.09.2021 | 1.11.2021 |  |
| 2 | Исследование документации Next.js | 13.10.2021 | 13.11.2021 |  |
| 3 | Исследование понятия SEO | 15.10.2021 | 15.11.2021 |  |
| 4 | Реализация проектов необходимых для анализа | 17.11.2021 | 21.11.2021 |  |
| 5 | Проведение экспериментов | 06.05.2022. | 07.05.2022 |  |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Б. Государев   
 (подпись)

Студент  А.С. Монченко

**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО"**

**(УНИВЕРСИТЕТ ИТМО)**

**АННОТАЦИЯ К КУРСОВОМУ ПРОЕКТУ (РАБОТЕ)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Монченко Артем Сергеевич | | | | | | |
| (Фамилия, И., О.) | | | | | | | | |
| Факультет | ПИиКТ | | | | | | Группа | P41071 |
| Направление (специальность) | | | | | | 09.04.04 Программная инженерия | | |
| Руководитель | | | | Государев И. Б., преподаватель | | | | |
| (Фамилия, И., О., должность) | | | | | | | | |
|  | | | | | | | | |
| Дисциплина | | | Проектирование и анализ языков веб-решений | | | | | |
|  | | | | | | | | |
| Наименование темы: | | | | | Анализ работы технического SEO в React и Next.js | | | |
| |  | | --- | | (Наименование сайта) | | | | | | | | | |
|  | | | | | | | | |

ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1. Цель и задачи работы** | | Определены руководителем | | Предложены студентом | |
| Цель работы – Провести анализ работы технического SEO в React и Next.js | | | | | |
| Задачи работы:  1. Исследовать документацию фреймворков и понятия SEO. 2. Реализовать проекты для проведения анализа. 3. Проанализировать работу SEO для первого проекта.  4. Проанализировать работу SEO для второго проекта.  5. Исследовать полученные результаты. | | | | | |
| **2. Характер работы** | | | | | |
|  | | | | | |
| Расчет | Конструирование | | Моделирование | | Другое |
| **3. Содержание работы** | | | | | |
| Проведён анализ работы технического SEO в React и Next.js | | | | | |
|  | | | | | |
|  | | | | | |
|  | | | | | |
| **4. Выводы** | |  | | | |
| Требования к проекту реализованы | | | | | |
|  | | | | | |
|  | | | | | |

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Б. Государев   
 (подпись)

Студент  А.С. Монченко

**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**Факультет программной инженерии и компьютерной техники**

Направление (специальность) ⎯ 09.04.04 Программная инженерия

Специализация ⎯ Веб-технологии

Дисциплина — Проектирование и анализ языков веб-решений

Курсовой проект (работа)

ТЕМА: Анализ работы технического SEO в React и Next.js

ВЫПОЛНИЛ

Студент группы   P41071  Монченко А.С.

№  группы подпись, дата ФИО

ПРОВЕРИЛ к.п.н., доцент  Государев И.Б.

ученая степень, должность подпись, дата ФИО

САНКТ-ПЕТЕРБУРГ  
2022 г.

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc102840193)

[Теоретическая часть 4](#_Toc102840194)

[Реализация приложений 8](#_Toc102840195)

[Ход исследования 10](#_Toc102840196)

[Результаты 12](#_Toc102840197)

[Выводы 13](#_Toc102840198)

[Заключение 13](#_Toc102840199)

[Список литературы 15](#_Toc102840200)

Введение

В настоящее время человечество переживает период BANI-мира: хрупкого, тревожного, нелинейного, непостижимого и непонятного, в котором нет времени для размеренного отдыха, а также места спокойному и неторопливому ритму жизни. Именно цифровизация и стала главной причиной, по которой изменился уклад современного общества. В связи с постоянным запросом на производительность цифровых систем в условиях недостатка личного времени пользователя и появились одностраничные приложения (SPA) во главе с такими фреймворками, как React, Vue и Angular.

Сайты, созданные на основе данных технологий, быстро загружаются даже при низкой скорости соединения и не требуют высокой пропускной способности канала, лучше работают на мобильных устройствах и, что самое важное, отличаются высокой производительностью. Основным недостатком SPA являются проблемы с индексацией страниц таких приложений поисковыми роботами, что негативно влияет на SEO, а следовательно, и на поисковую выдачу. Дело в том, что когда поисковая система заходит на сайт, сделанный с помощью Ajax-технологий или какого-нибудь современного фреймворка, то видит пустой HTML-документ без какого-либо содержимого.

Для решения проблем оптимизации поисковой выдачи был изобретён Next.js – основанный на React фреймворк, предназначенный для разработки веб-приложений, обладающих функционалом, выходящим за рамки SPA. В его основе лежит технология SSR позволяющая с помощью Node.js запускать JavaScript код на сервере (а не в браузере, как обычно) и готовый результат отправлять пользователю, избегая лишней нагрузки на его браузер и компьютер.

Целью данного исследования является анализ работы технического SEO в React и Next.js приложениях, в качестве которых буду выступать [сайт-портфолио](https://mondevyat.org/) и [блог](https://blogify-tau.vercel.app/) со статьями IT-тематики, соответственно.

Теоретическая часть

Стек технологий, необходимых для реализации приложений, на основе которых и будет производится анализ, будет состоять из следующих компонентов: React – в качестве JS-фрейморка для фронтэнд-части сайта-портфолио, Tailwind CSS – в качестве CSS-фреймворка, Next.js – в качестве JS-фрейморков для фронтэнд-части блога, GraphCMS – в качестве серверной части блога.

Как следует из вышеописанного стека, оба веб-сайта будут реализованы как одностраничные или SPA-приложения. Они представляют собой веб-сайт или веб-приложение, которое использует один HTML-документ в качестве оболочки для всех веб-страниц и взаимодействующее с пользователем посредством динамически подгружаемых HTML, CSS и JavaScript. Они так же значительно превосходят в производительности и отзывчивости стандартный мультистраничный сайт (МРА).

SPA приложение содержит в себе три составляющих:

1. Представление (view) предназначено для отрисовки пользовательского интерфейса. Через представление осуществляется взаимодействие пользователя с функциями системы. Результатом работы представления является обычный HTML. При помощи событий, изменения в представлении сразу же обновляют данные в модели. Представление так же предоставляет пользователю элементы управления для взаимодействия с приложением.
2. Модель (model) предназначена для хранения данных приложения и работы с событиями. При помощи событий, изменения в модели сразу же обновляют данные в представлении. Модель организует связь с сервером через REST API или WebSocket соединение.
3. Сервисы (services) предназначены для получения данных через REST API и WebSocket, а также для реализации логики отдельных компонентов приложения [1].

Для SPA приложений взаимодействие с backend частью в большей степени сводится к использованию REST API и WebSocket соединения. За счет этого, разработчику SPA приложения не требуется реализовывать функционал на backend, и он может полностью сфокусироваться на своей части. Также, процесс создания приложения в целом, может протекать параллельно – backend и frontend части пишутся одновременно.

Вопреки своему настольному характеру, одностраничные приложения по-прежнему разрабатываются с использованием фреймворков JavaScript. Самые популярные – React, Angular и Vue.js.

В данной работе подробнее всего будет рассмотрен фреймворк React - библиотека JavaScript предназначенная для создания пользовательского интерфейса [11]. Библиотека пропагандирует, что всё есть компоненты, которые в свою очередь делятся на функциональные и классовые, и они могут быть вложены друг в друга. Таким образом можно одни компоненты вкладывать в другие и продолжать дополнять данное дерево зависимостей. Фреймворк использует специальный язык разметки JSX со своим синтаксисом. JSX – это не HTML, а написанные на JavaScript для удобства разработки конструкции, которые интерпретатор React визуально превращает в код похожий на HTML.

React использует рендеринг на стороне клиента (CSR), когда сервер отправляет на страницу почти пустой файл HTML, за которым следуют все файлы JavaScript в одном большом пакете, который затем должен быть обработан в браузере для рендеринга DOM-дерева [2].

На страницах, отображаемых на стороне клиента, начальная скорость загрузки низкая, и пользователь видит пустой экран, пока не будет выполнен весь JavaScript и не будут выполнены запросы к API. Однако последующие скорости загрузки будут быстрыми, так как дальнейшие изменения потребуют только обновления соответствующих разделов DOM-дерева [2].

Проблема начальной скорости загрузки в CSR может быть решена с помощью рендеринга на стороне сервера (SSR), при котором сервер извлекает информацию из базы данных и отправляет на страницу подготовленный HTML-файл. При таком подходе на странице есть значимый контент, который может просматривать пользователь. Страница становится интерактивной после выполнения файла JavaScript. Здесь сервер несет нагрузку по выборке и обработке данных, имея более высокую начальную скорость загрузки. Однако на последующие скорости загрузки это не повлияет, так как каждую страницу необходимо отобразить и снова получить с сервера [2].

Один из простых способов комбинировать методы рендеринга на стороне клиента и на стороне сервера, чтобы получить лучшее из обоих - использовать Next.js, фреймворк, построенный на основе React, который позволяет пользователям задействовать широкий спектр функций, таких как рендеринг на стороне сервера, статическая генерация, простая маршрутизация на основе файлов и многое другое [2].

Next.js был вдохновлен подходом языка PHP и его экосистемы к серверной отрисовке приложения (Server-Side Render, SSR), и впервые был упомянут в 2016 году компанией-создателем данной библиотеки – ZEIT [10]. Основным его отличием от React является гибридный (изоморфный) подход: первый запрос в приложении всегда обрабатывается на стороне сервера, все последующие же изменяют отображения привычным для React способом – на стороне клиента. Именно такой подход к построению SPA, основанный на серверном отображении и изоморфности, решает проблему индексации поисковыми роботами. Раньше заполнение контента только после асинхронного выполнения JavaScript-кода затрудняло работу роботов при сканировании страниц, что ожидаемо приводило к снижению SEO-эффективности. В то же самое время реализация изоморфоности не лишает разработчиков привычной отзывчивости и скорости работы одностраничных приложений. [3].

Другим важным преимуществом SSR перед CSR является поисковая оптимизация (SEO). Поскольку страницы, отображаемые на сервере, уже подготовлены с соответствующими метаданными, роботы, используемые поисковыми системами, такими как Google, могут легко классифицировать эти страницы, что приводит к более высокому поисковому рейтингу веб-сайта. Однако страницы, использующие CSR, полагаются на выполнение JavaScript для заполнения метаданных. Часто, если выполнение занимает более 300–400 мс, бот обрабатывает пустую страницу, что негативно влияет на SEO показатели [2].

SEO подразумевает под собой реализацию мероприятий, направленных на продвижение сайта в поисковых системах. Если учесть то, что прибыльность некоторых сайтов зависит от их посещаемости, а посещаемость напрямую связана с позицией в поисковых системах, то вопрос с поисковой оптимизацией является весьма и весьма актуальным [4]. SEO – комплекс мероприятий по внутренней и внешней оптимизации для поднятия позиций сайта в результатах выдачи поисковых систем по определённым запросам пользователей, с целью увеличения сетевого трафика и потенциальных клиентов, и последующей монетизации этого трафика. SEO может быть ориентировано на различные виды поиска, включая поиск информации, товаров, услуг, изображений, видеороликов, новостей и специфические отраслевые поисковые системы.

Техническое SEO определяется производительностью веб-приложения и представляет собой процесс оптимизации сайта в соответствии с современными требованиями поисковых систем. Мало создать качественный по наполнению сайт, если с технической точки зрения сайт несовершенен, то он не будет ранжироваться в поисковиках. На базовом уровне системы должны иметь возможность находить, сканировать, отображать и индексировать страницы вашего ресурса. Для полной технической оптимизации, страницы:

* должны быть безопасны;
* адаптированы для мобильных устройств;
* не должны содержать дублирующуюся информацию;
* должны быстро загружаться.

Главной по важности технических работ является планирование структуры сайта. Именно из-за плохо спроектированной архитектуры ресурса возникают проблемы с индексацией. Более того, структура сайта влияет и на все остальные рекомендации для оптимизации — от URL-адресов до robots.txt. Под структурой подразумевается то, как именно организованы страницы веб-сайта. Она должны быть «плоской»: при ней все страницы сайта находятся в нескольких кликах друг от друга. Только такая структура поможет Google и другим поисковикам просканировать абсолютно весь веб-ресурс. В противном случае, если сайт содержит «оторванные страницы», на которые не указывает ни одна ссылка, то их обход для бота-краулера будет весьма затруднительным. В этом вопросе помогут сервисы подобные [Ahrefs](https://ahrefs.com/) или [Semrush](https://ru.semrush.com/), они помогают получить представление о структуре сайта.

Итак, из теоретической части были выделены следующие критерии для анализа технологического SEO в React и Next.js приложениях: соответствие основам технического SEO, структура и навигация, распознавание, дублирующийся контент, скорость загрузки содержимого и отклик, особенности оптимизации.

Реализация приложений

Был произведён собственный эксперимент, в ходе которого реализован блог по тематике IT. В ходе разработки данного приложения было задействовано множество особенностей Next.js, среди которых: компонент Link, реализующий роутинг между «страницами»; компонент Head, добавляющий метатеги в контейнер <head> и позволяющий избежать дублирования мета-тегов на сайте, что существенно улучшает результаты SEO; компонент Image, значительно повышающий скорость отклика сайта, что является одним из самых важнейших показателей для пользовательского опыта и поискового рейтинга Google. Вышеописанные практики положительно влияют на производительность сайта, помогая пользователю тратить гораздо меньше времени на ожидание загрузки содержимого страницы.

В качестве фреймворка для CSS было решено взять Tailwind CSS. Данная библиотека упрощает стилизацию HTML, тем же путем, как это делает Bootstrap, – добавляя огромное количество разнообразных классов. Но, в отличие от Bootstrap, который добавляет уже готовые к употреблению компоненты, такие как кнопки, алерты и навбары, классы TailwindCSS нацелены на конкретное свойство. В TailwindCSS нет заранее написанной кнопки, её нужно написать самому [12].

Безусловно засорять HTML инлайн-стилями – это строго против правил, но Tailwind нарушает эту аксиому благодаря своим особенностям. Как уже говорилось раньше, Tailwind – это не Bootstrap, их концепция, согласно которой классы выступают именно как css-свойства, а не как компоненты, меняет всё. Здесь сходство лишь внешнее, а так Tailwind – это новый подход к написанию CSS. По факту, код Tailwind CSS пишется в формате похожем на популярный плагин Emmet, что значительно ускоряет разработку. Переиспользуемость и никаких лишних стилей, благодаря удобно настраиваемому файлу конфигурации и подробной документации.

Как уже упоминалось выше, в качестве серверной стороны приложения будет выступать GraphCMS – безголовый CMS, основанный на GraphQL, языка запросов для API-интерфейсов и среда, в которой они выполняются [9]. Он позволяет создавать размещенный API-интерфейс контента GraphQL для сайтов и приложений, в то же время предоставляя редакторам контента все инструменты, необходимые для выполнения этой работы [8].

Лучший способ развернуть Next.js в рабочей среде – использовать плат- форму Vercel от создателей Next.js. Данная облачная платформа предназначена для статических сайтов и гибридных приложений [2]. Имея простой в установке и использовании интерфейс командой строки, Vercel позволяет автоматически получить проект с SSL-сертификатом, а также автоматически обновляет содержимое страниц сайта при изменении программного кода в среде разработки [13]. Подобные особенности данной облачной платформы имеют своё положительное влияние на поисковую оптимизацию и индексацию веб-приложения поисковыми роботами.

Второй проект представляет собой сайт-портфолио, реализованный как SPA-приложение. Для его написания понадобился следующий стек технологий: React – в качестве JS-фреймворка для фронтэнд части приложения, CSS-модули, Three.js – для работы с 3d-объектами и прочие несущественные библиотеки.

Ход исследования

Как уже было выделено в теоретической части исследования, анализ технического SEO у React и Next.js приложений будет проходить по шести основным критериям:

1. Соответствие основам технического SEO – проверка страниц приложения на безопасность, адаптивность, отсутствие дублирующейся информации, а также отзывчивость интерфейса.
2. Структура и навигация – проверка страниц на наличие «плоской», организованной структуры сайта, согласованной структуры URL и, если понадобится, навигации.
3. Распознавание роботами – критерий, посвящённый упрощению поиска сайта в интернете и индексации всех его страниц.
4. Дублирующийся контент – критерий, имеющий особую важность для блога со статьями и проверяющий любой «некачественный» или повторяющийся контент, как например, несколько копий одной и той же страницы по разным URL-адресам в CMS.
5. Скорость сайта – критерий, напрямую влияющий на ранжирование веб-ресурса, и грамотная реализация которого приведёт к увеличению трафика. На неё могут влиять лишние скрипты, уменьшение размера страниц или эксперимент с загрузкой сайта с CDN или без.
6. Особенности оптимизация – критерий, подразумевающий разные способы оптимизации, не относящиеся к разделам выше, например, hreflang для международных сайтов, обработка битых ссылок, мобильное юзабилити и много другое.

В ходе проверки приложений на соответствие первому критерию оптимизации было выявлено, что React-приложение при наличии безопасных страниц, адаптивности для мобильных устройств и отсутствию дублирующейся информации, недостаточно быстро загружает медиаконтент, такой как картинки или 3d-объекты. Next.js перекрывает этот недостаток за счёт специального встроенного компонента Image, в том числе значительно повышающего скорость отклика сайта.

Касаемо проверки на соответствие второму критерию выяснилось, что приложения имеют «плоскую» структуру сайта, согласованную структуру URL, а блог располагает элементами навигации, реализованных в виде списка категорий статей, исходя из этого можно сделать вывод о том, что приложения в полной мере соответствуют критериям структуры и навигации. Аудит сайта был проведён при помощи сервиса [Ahrefs](https://ahrefs.com/), помогающего получить представление о структуре сайта.

Благодаря функции «Site Explorer» сервиса [Ahrefs](https://ahrefs.com/) было выявлено, что распознавание веб-ресурса поисковыми роботами (краулерами) происходит оптимальнее при работе с блогом на Next.js нежели на сайте-портфолио, реализованном при помощи фреймворка React, поэтому соответствие у данного проекта третьему критерию оптимизации частичное, нежели у его оппонента.

Проверка веб-сервисов на наличие «некачественного» или дублирующегося контента показала, что оба приложения, и на React и на Next.js, полностью удовлетворяют четвёртому критерию SEO-оптимизации. Раздел «Качество контента» всё того же [Ahrefs](https://ahrefs.com/) показывает содержится ли на сайте страницы с повторяющимся контентом или несколько копий одной и той же страницы, но по разным URL-адресам.

Скорость мобильной и десктопной версий приложений были получены при помощи веб-сервиса [PageSpeed Insights](https://pagespeed.web.dev). Выходные результаты говорят о частичном соответствии пятому критерию с небольшим преимуществом десктопной React-версии и мобильной Next.js-версии.

Переходя к последнему критерию, отвечающему за особенности оптимизации, можно выделить наличие обработки битых ссылок у блога на Next.js в то время, как подобные особенности у сайта-портфолио на React отсутствуют, что говорит о полном несоответствии шестому критерию SEO-оптимизации.

Результаты

В ходе исследования были выявлены основные особенности при работе с техническим SEO в React и Next.js приложениях. Окончательные результаты сравнительного анализа приложений по шести основным выделенным SEO-критериям предоставлены в Таблица таблице 1.

Таблица Результаты анализа приложений по SEO-критериям

|  |  |  |
| --- | --- | --- |
| **Наименование критерия** | **React-приложение** | **Next.js-приложение** |
| Соответствие основам технического SEO | Частично соответствует | Полностью соответствует |
| Структура и навигация | Полностью соответствует | Полностью соответствует |
| Распознавание роботами | Частично соответствует | Полностью соответствует |

***Продолжение табл. 1***

|  |  |  |
| --- | --- | --- |
| Дублирующийся контент | Полностью соответствует | Полностью соответствует |
| Скорость сайта | Частично  соответствует | Частично  соответствует |
| Особенности оптимизации | Не соответствует | Частично соответствует |

Выводы

Для проведения сравнительного анализа технологического SEO в приложениях, реализованных при помощи одной из технологий: React или Next.js, были разработаны два проекта в виде сайта-портфолио и блога со статьями. Первый проект выполнен с использованием JavaScript-фреймворка React, а второй при помощи основанной на предыдущем фреймворке технологии под названием Next.js, решающая проблемы оптимизации поисковой выдачи благодаря функционалу, выходящему за рамки одностраничного приложения.

Как можно заметить из вышеописанных результатов, в вопросах SEO-оптимизации с большим преимуществом в соответствии основным критериям оптимизации поисковой выдачи лидирует фреймворк Next.js.

Заключение

В ходе выполнения работы были выбраны и реализованы два проекта, различающиеся по изначально выбранному стеку технологий. Была проанализирована область исследования в сфере SEO и выделены шесть основных критериев, по которым и проходило сравнение вышеописанных приложений. На основании полученных результатов был проведён сравнительный анализ и выявлена та технология, которая выдаёт более подходящие результаты при работе с поисковой оптимизацией.

Таким образом, исходя из всего того, что было изложено ранее, можно сделать следующий вывод: в случае, если пользователю необходимо создать небольшой проект, как например, сайт-портфолио, и ему не особо важна индексация данного проекта, его ранжирование в международной паутине для привлечения траффика на свой ресурс, то выбор остаётся за CSR-технологиями, а именно фреймворком React, в противном же случае стоит обратить внимание на технологию Server-Side Rendering и сделать выбор в пользу Next.js.

Список литературы

1. Демчук В.Ю. Одностраничные приложения (SPA) // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2018). 2018. №3. C. 38-43. URL: https://elibrary.ru/item.asp?id=42755722 (дата обращения: 03.05.2022).
2. Князев И.В. Продвинутое кеширование и оптимизация веб-приложений с помощью технологии SWR // Sciences of Europe. 2021. №73. С. 47-49. URL: https://cyberleninka.ru/article/n/prodvinutoe-keshirovanie-i-optimizatsiya-veb-prilozheniy-s-pomoschyu-tehnologii-swr (дата обращения: 03.05.2022).
3. Мытников А.Н., Капитонов А.М. Разработка изоморфных приложений на React с помощью Next.js // Актуальные проблемы современной науки: взгляд молодых ученых. 2020. №5. С. 675-677. URL: https://elibrary.ru/item.asp?id=43065990 (дата обращения: 03.05.2022).
4. Скотт, Адам Д. Разработка на JavaScript. Построение кроссплатформенных приложений с помощью GraphQL, React, React Native и Electron / Адам Д. Скотт. – Cпб.: Питер, 2021. – 320 c. – ISBN 978-5-4461-1462-7.
5. Хорстман, К. Современный JavaScript для нетерпеливых / Кэй Хорстман – М.: ДМК-Пресс, 2021. – 288 c. – ISBN 978-5-97060-177-8.
6. Якушин А.Ю., Муковозов А.М., Исмоилов М.И. Сравнительный анализ реляционной базы данных и документоориентированной NoSQL базы данных в разрезе их применения при создании локального чата/мессенджера // Инновационная наука. 2018. №1. С. 12-16. URL: https://cyberleninka.ru/article/n/sravnitelnyy-analiz-relyatsionnoy-bazy-dannyh-i-dokumentoorientirovannoy-nosql-bazy-dannyh-v-razreze-ih-primeneniya-pri-sozdanii (дата обращения: 03.05.2022).
7. Eisenman, B. Learning React Native / B. Eisenman – E.: O’Reilly Media, 2019. – 450 c. – ISBN 978-1-49198-914-2.
8. GraphCMS [Электронный ресурс] URL: https://graphcms.com/ (дата обращения: 03.05.2022).
9. GraphQL [Электронный ресурс] URL: https://graphql.org/ (дата обращения: 03.05.2022).
10. NextJS [Электронный ресурс] URL: https://nextjs.org/ (дата обращения: 03.05.2022).
11. ReactJS [Электронный ресурс] URL: https://reactjs.org/ (дата обращения: 03.05.2022).
12. Tailwind CSS [Электронный ресурс] URL: https://tailwindcss.com/ (дата обращения: 03.05.2022).
13. Vercel [Электронный ресурс] URL: https://vercel.com/ (дата обращения: 21.11.2021).
14. Zammetti, F. Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker / Frank Zammetti – Е.: Apress, 2020. – 398 c. – ISBN 978-1-48425-737-1.

**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**«Национальный исследовательский университет ИТМО»**

**(Университет ИТМО)**

**ОТЗЫВ РУКОВОДИТЕЛЯ**

**о выполнении курсового проекта (работы)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Студент | | Монченко Артем Сергеевич | | | | | | |
| (Фамилия, И., О.) | | | | | | | | |
| Факультет | ПИиКТ | | | | | | Группа | Р41071 |
| Направление (специальность) | | | | | | 09.04.04 «Программная инженерия» | | |
| Руководитель | | | | Государев И.Б., ИТМО ПИиКТ, доцент, кандидат педагогических наук | | | | |
| (Фамилия, И., О., место работы, должность, ученое звание, степень) | | | | | | | | |
|  | | | | | | | | |
| Дисциплина | | | Проектирование и анализ языков веб-решений | | | | | |
|  | | | | | | | | |
| Наименование темы: | | | | | Анализ работы технического SEO в React и Next.js | | | |
| |  | | --- | |  | | | | | | | | | |
|  | | | | | | | | |

**ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Показатели** | **Оценка** | | | |
| **п/п** | **5** | **4** | **3** | **0** |
| 1. | Проект создан обучающимся самостоятельно |  |  |  |  |
| 2. | Созданные элементы сайта раскрывают тематику и название фирмы |  |  |  |  |
| 3. | Проект технологически грамотный |  |  |  |  |
| 4. | Оформление отвечает требованиям к отчету |  |  |  |  |
| 5. | Во время защиты обучающийся показал умение кратко, доступно представить результаты работы, умение анализировать, аргументировать свою точку зрения, делать обобщение и выводы, адекватно ответить на поставленные вопросы. |  |  |  |  |
| **ИТОГОВАЯ ОЦЕНКА** | |  | | | |

**Отмеченные достоинства:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Отмеченные недостатки:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Заключение:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.Б. Государев   
 (подпись)

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_