Министерство образования и науки Республики Казахстан

Торайгыров Университет

Факультет Энергетики и компьютерных наук

Кафедра Информационные технологии

**КУРСОВОЙ ПРОЕКТ**

|  |  |  |  |
| --- | --- | --- | --- |
| **По специальности** | | | «Информационные системы» |
| **дисциплине** | | Проект | |
| **Тема** | Восприятие пользователем цифрового контента и SEO-продвижение | | |
|  | | | |

**Выполнил** студент группы ИС-102(с) Афанасьева Д.Т.

**Проверил** старший преподавательАлимова Ж.С.

Павлодар, 2020

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc69425025)

[ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc69425026)

[1. Психологические аспекты восприятия информации как составная часть SEO-продвижения 4](#_Toc69425027)

[1.1. Целевая аудитория и сегментирование 4](#_Toc69425028)

[1.2. Особенности восприятия различных форматов информации 4](#_Toc69425029)

[1.3. UX/UI как важная составляющая успеха программного продукта 5](#_Toc69425030)

[2. Современные технологии оптимизации и продвижения сайта 6](#_Toc69425031)

[2.1. Кэширование и минификация 6](#_Toc69425032)

[2.2. Настройка метатегов 6](#_Toc69425033)

[2.3. Микроразметка и Open Graph 7](#_Toc69425034)

[2.4. ЧПУ 8](#_Toc69425035)

[2.5. Sitemap.xml и robots.txt 8](#_Toc69425036)

[2.6. Особенности HTTP/HTTPS и заголовки 8](#_Toc69425037)

[3. Инструменты SEO-тестирования сайта 9](#_Toc69425038)

[3.1. Google Developers Console 9](#_Toc69425039)

[3.2. Яндекс Метрика 10](#_Toc69425040)

[3.3. SEO-анализаторы 10](#_Toc69425041)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 11](#_Toc69425042)

[3.1. Проектирование базы данных и создание бэкенда 11](#_Toc69425043)

[3.2. Создание фронтенд-части 15](#_Toc69425044)

[3.3. Настройка SEO 17](#_Toc69425045)

[ЗАКЛЮЧЕНИЕ 19](#_Toc69425046)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 20](#_Toc69425047)

# **ВВЕДЕНИЕ**

В настоящее время все больше информации представлено в виде интернет-ресурсов, поэтому важным становится качество и доступность контента. Для этого необходимо знать об особенностях восприятия информации и технических особенностях продвижения.

В соответствии с этим тема данного курсового проекта: Восприятие пользователем цифрового контента и SEO-продвижение.

Целью проекта является изучение современных средств SEO-оптимизации и основных трендов в UX-дизайне.

Актуальность проекта заключается в том, что зачастую создатели сайтов не анализируют читабельность и индексируемость страниц, что может привести к тому, что информация останется незамеченной, общее впечатление у пользователей будет хуже, либо поисковые роботы не смогут корректно обработать содержимое.

В рамка выполнения данного курсового проекта поставлены следующие задачи:

- описать основные показатели SEO, проверяемые анализаторами (микроразметка, sitemap.xml, robots.txt, кэшируемость, протокол передачи данных, ЧПУ, редиректы, HTTP-заголовки и др.).

- провести сравнительный анализ средств аналитики посещаемости сайтов (Google Analytics, Яндекс Метрика).

- изучить психологические особенности восприятия текстовой, графической, аудио- и видеоинформации

- при создании сайта учесть вышеописанные показатели и провести тестирование.

В данном проекте будут рассмотрены особенности продвижения контента как с технической стороны, так и с социально-психологической. То есть будет подробно описан процесс настройки вышеперечисленных параметров в реальном проекте, а также проанализированы особенности восприятия различных видов информации (текстовой, видео, звуковой, графической) с учетом типа аудитории. Для демонстрации процесса изучения будет создан веб-сайт, тематика которого следующая: платформа объединения пользователей по интересам, областям деятельности и целям в обучении для совместного развития, написания блогов и демонстрации прогресса по личным проектам. В ходе выполнения будут изучены последние тренды в веб-разработке.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1. Психологические аспекты восприятия информации как составная часть SEO-продвижения**

### **1.1. Целевая аудитория и сегментирование**

Целевая группа, целевая аудитория — термин, используемый в маркетинге или рекламе для обозначения группы людей, объединённых общими признаками, или объединённой ради какой-либо цели или задачи. Под общими признаками могут пониматься любые характеристики, требуемые организаторами.

Сегментирование целевой аудитории – это разделение аудитории на группы, где они объединены по признаку схожих потребностей (запросов).

При правильном подходе составляется портрет потенциального клиента, определяется его социальный статус, основные цели и интересы, в соответствии с этим определяется то, как будет преподноситься контент и как настроить рекламные кампании.

### **1.2. Особенности восприятия различных форматов информации**

Важно понять, как работает фокусировка внимания, как мозг воспринимает различные цвета, формы, объем и формат информации и т.д. Например, текст в несколько колонок легче читается, чем широкие строки, контент лучше группировать по 4-7 элементов, так как это оптимально для запоминания.

В последнее время наблюдается тенденция к тому, что люди не способны долго фокусировать внимание на объемном тексте, а предпочитают небольшие отрывки текста с изображениями либо короткие видео в пределах нескольких минут. Этим обусловлена популярность различных социальных сетей, так как основной их контент не требует большой нагрузки и легок в восприятии.

Цветовые решения в визуальном оформлении имеют немаловажное значение, так как каждый цвет имеет свое психологическое действие. При выборе цвета учитывают, какие действия требуются от пользователя и какие эмоции вызвать: доверие, спокойствие, серьезный настрой, желание рисковать, практичность или тщеславие. При этом важно помнить, что в качественном дизайне не стоит использовать много цветов, это выглядит сумбурно. Как правило, выбирается несколько основных цветов, используемых на страницах. Нужно также помнить о соблюдении контраста для читабельности и удачных цветовых сочетаниях. Здесь могут помочь специальные сервисы для подбора палитры.

В рамках работы над курсовым проектом был проведен опрос касательно восприятия контента, и результаты следующие (таблица 1):

|  |  |
| --- | --- |
| Что вам предпочтительнее? | Читать текст – 83,3%  Смотреть видео – 16,7% |
| С каких устройств вы чаще выходите в интернет? | Смартфон – 50%  Ноутбук – 16,7%  Планшет – 0%  ПК – 33,3% |
| Часто ли вы делитесь ссылками на интернет-источники в социальных сетях? | Часто – 50%  Иногда – 16,7%  Редко – 16,7%  Никогда – 16,7% |
| Сложно ли вам читать текст больше одной страницы? | Сложно, никогда не дочитываю – 16,7%  Не вызывает затруднений – 83,3% |
| На что вы обращаете внимание в первую очередь? | На текст – 83,3%  На иллюстрации – 16,7% |
| Часто ли вы переходите на вторую страницу поисковой выдачи? | Всегда просматриваю только первые ссылки – 16,7%  Если мне очень нужно найти информацию – 33,3%  Всегда смотрю несколько страниц – 50% |
| Что доставляет вам наибольшее неудобство при пользовании различными сайтами? | Медленная загрузка – 33,3%  Непонятный интерфейс – 50%  Неудачно подобранные цвета – 16,7%  Появление всплывающих окон – 66,7%  Реклама – 16,7% |
| Темная тема или светлая тема? | Темная тема – 66,7%  Светлая тема – 33,3% |
| Насколько активно вы публикуете в интернете свой контент (посты, фото в соцсетях, блоги и т.д.)? | Часто – 0%  Иногда – 16,7%  Редко – 66,7%  Никогда – 16,7% |
| Часто ли вы лайкаете или комментируете? | Часто – 16,7%  Иногда – 33,3%  Редко – 33,3%  Никогда – 16,7% |

Таблица 1 – Результаты опроса

### **1.3. UX/UI как важная составляющая успеха программного продукта**

UX (User Experience - опыт пользователя) - восприятие и ответные действия пользователя, возникающие в результате использования и/или предстоящего использования продукции, системы или услуги.

UI – интерфейс пользователя, обеспечивающий непосредственное взаимодействие пользователя с системой.

Эти понятия тесно связаны, так как при проектировании дизайна учитывается удобство и интуитивная понятность конечному пользователю. Так, элемент своим внешним видом, расположением, размером должен описывать свою функциональность и предназначение. Если интерфейс спроектирован хорошо, пользователь будет без дополнительных подсказок понимать, что происходит и какие действия стоит выполнить для достижения своих целей. Если клиенту удобно и приятно работать с продуктом, он более лоялен и программный продукт выполняет свои функции должным образом.

При грамотном подходе проводится usability testing, заключающийся в следующем: тестовая группа потенциальных пользователей работает с прототипом программного продукта, их действия записываются и анализируются, также проводится опрос о впечатлениях. На основании этого вносятся правки и корректировки.

**2. Современные технологии оптимизации и продвижения сайта**

### **2.1. Кэширование и минификация**

Кэш или кеш — промежуточный буфер с быстрым доступом к нему, содержащий информацию, которая может быть запрошена с наибольшей вероятностью. Кэширование — это процесс запоминания браузером данных, которые были переданы на сервер для обработки и использованы в дальнейшем при загрузке страницы.

Обычно кэшируются такие медиафайлы, как изображения, аудио, видео, файлы скриптов, стилей и шрифтов. Кэширование настраивается путем корректной настройки заголовков в файле .htaccess. Также полезная функция – теги, указывающие, какие ресурсы необходимо подгрузить заранее, тем самым оптимизируют и ускоряют загрузку страницы.

Минификация в языках программирования — процесс, направленный на уменьшение размера исходного кода путём удаления ненужных символов без изменения его функциональности. Минификация особенно полезна для программ на интерпретируемых языках, потому что она уменьшает объём данных, которые должны быть обработаны.

### **2.2. Настройка метатегов**

Метатеги — (X)HTML-теги, предназначенные для предоставления структурированных метаданных о веб-странице. Они располагаются в секции head разметки.

Для корректного отображения и индексирования страницы необходимо правильно заполнить метатеги description и title. Они отвечают за то, по каким поисковым запросам найдется сайт и как будет выглядеть миниатюра в поисковой выдаче.

Также имеют значение различные теги link. Наиболее применимые типы:

- <link rel=”stylesheet”> - используется для подключения внешних стилей

- <link rel=”preload”> - используется для предзагрузки медиафайлов

- <link rel=”prefetch”> - аналогично preload, с тем отличием, что подгружаются в фоне те файлы, которые понадобятся на другой странице

- <link rel=”prerender”> - предварительно загружает и кэширует другую страницу, которая с большой вероятностью понадобится

- <link rel=”canonical”> - канонический URL

- <link rel=”alternate”> - альтернативные версии страницы на других языках

- <link rel=”apple-touch-icon”> - определяет картинку/иконку на устройствах Apple, которая определяет данный сайт или страницу.

Стоит обратить внимание на следующие метатеги:

- <meta charset="utf-8"> - задает кодировку страницы

- <meta name="viewport" content="width=device-width, initial-scale=1"> - необходимо для правильного масштабирования и адаптивности

Также в метатеги часто помещают различную служебную информацию, например, токены и др.

### **2.3. Микроразметка и Open Graph**

Помимо вышеописанных базовых метатегов также широко используются специальные средства разметки, помогающие в продвижении ресурса.

Микроразметка – это разметка страницы сайта с дополнительными тегами и атрибутами, которая помогает поисковикам более четко понимать содержание страниц интернет-ресурса. А это, в свою очередь, позволяет более корректно отображать контент в поисковой выдаче.

С помощью микроразметки можно описать навигацию по сайту, ссылки на контакты, режим работы, оценки, и многие другие полезные функции, которые помогут получить доступ к нужному контенту прямо из поисковой выдачи и повысить SEO. Настройка микроразметки Schema.org наиболее предпочтительна через специальный скрипт application/ld+json.

Open Graph -это протокол, который позволяет управлять и контролировать данные, формирующиеся в превью (структурирует информацию о странице) в посте, когда пользователь делиться ссылкой в социальных сетях. В специальных метатегах прописывается название сайта, URL, описание, логотип, изображение (рисунок 1).

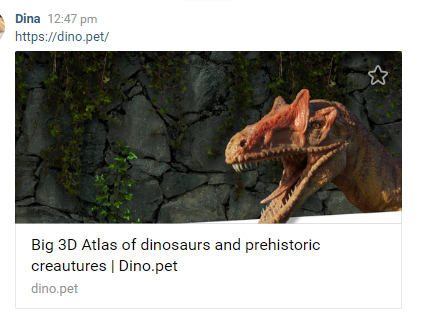


Рисунок 1 – сайт с Open Graph

### **2.4. ЧПУ**

Человекопонятный URL — URL-путь, состоящий из понятных слов, вместо идентификаторов, и отражающий файловую структуру сайта. Например, вместо /cat/3a/47/ или /main.php?q=10&sort=2&id=41 будет /blog/article/the-best-trends/. Преимущество в лучшей читаемости и смысловой нагрузке. Реализовать можно с помощью грамотной настройки маршрутизации и генерации «слагов» для каждого профиля, статьи и других ресурсов, на которые будет генерироваться ссылка.

### **2.5. Sitemap.xml и robots.txt**

Sitemaps — XML-файлы с информацией для поисковых систем о страницах веб-сайта, которые подлежат индексации.

Стандарт исключений для роботов — стандарт ограничения доступа роботам к содержимому на http-сервере при помощи текстового файла robots.txt, находящегося в корне сайта. При этом стоит прописывать отдельные правила для роботов Google и Яндекс.

### **2.6. Особенности HTTP/HTTPS и заголовки**

HTTPS — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов TLS или устаревшего в 2015 году SSL. В отличие от HTTP с TCP-портом 80, для HTTPS по умолчанию используется TCP-порт 443.

В настоящее время наблюдается тенденция к тому, что сайты без SSL-сертификата (работающие на HTTP) в принципе не имеют доверия и помечаются как опасные источники. Например, в консоли разработчиков Google и Facebook не удастся настроить авторизацию через их сервисы без HTTPS.

Заголовки – это специальные параметры, которые несут определенную служебную информацию о соединении по HTTP. Заголовки передают информацию о статусе ответа, типе передаваемых данных, авторизации и много другой необходимой браузеру информации.

Коды ответа сгруппированы в 5 классов:

- Информационные 100 - 199

- Успешные 200 - 299

- Перенаправления 300 - 399

- Клиентские ошибки 400 - 499

- Серверные ошибки 500 - 599

|  |  |
| --- | --- |
| Код ответа | Название |
| 100 | Continue |
| 101 | Switching Protocol |
| 102 | Processing |
| 103 | Early Hints |
| 200 | OK |
| 201 | Created |
| 202 | Accepted |
| 203 | Non-Authoritative Information |
| 204 | No Content |
| 205 | Reset Content |
| 206 | Partial Content |
| 300 | Multiple Choice |
| 301 | Moved Permanently |
| 302 | Found |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |
| 306 | Switch Proxy |
| 307 | Temporary Redirect |
| 308 | Permanent Redirect |
| 400 | Bad Request |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | Forbidden |
| 404 | Not Found |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |
| 408 | Request Timeout |
| 409 | Conflict |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 414 | Request-URI Too Long |
| 415 | Unsupported Media Type |
| 416 | Requested Range Not Satisfiable |
| 417 | Expectation Failed |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 502 | Bad Gateway |
| 503 | Service Unavailable |
| 504 | Gateway Timeout |
| 505 | HTTP Version Not Supported |

Таблица 2 - HTTP – коды ответа

## **3. Инструменты SEO-тестирования сайта**

### **3.1. Google Developers Console**

В консоли разработчика Google можно зарегистрировать свой проект, получить ключи API для различных сервисов, таких как Youtube, OAuth.

Search Console – это бесплатный инструмент от компании Google для разработчиков и владельцев сайтов, а также специалистов по поисковой оптимизации. Он позволяет собирать данные об эффективности сайта в Google Поиске. С помощью данного инструмента можно провести тестирование микроразметки.

### **3.2. Яндекс Метрика**

«Яндекс. Метрика» — бесплатный интернет-сервис компании Яндекс, предназначенный для оценки посещаемости веб-сайтов и анализа поведения пользователей (время на сайте, повторные посещения, источники трафика, процент отказов). На данный момент Яндекс. Метрика является третьей по размеру системой веб-аналитики в Европе. «Яндекс. Метрика» стала общедоступной 24 апреля 2009 года. Счетчик интегрируется на сайт посредством специального кода и собирает данные о посещаемости, целевых событиях, при включенном Вебвизоре даже запись действий пользователя на странице.

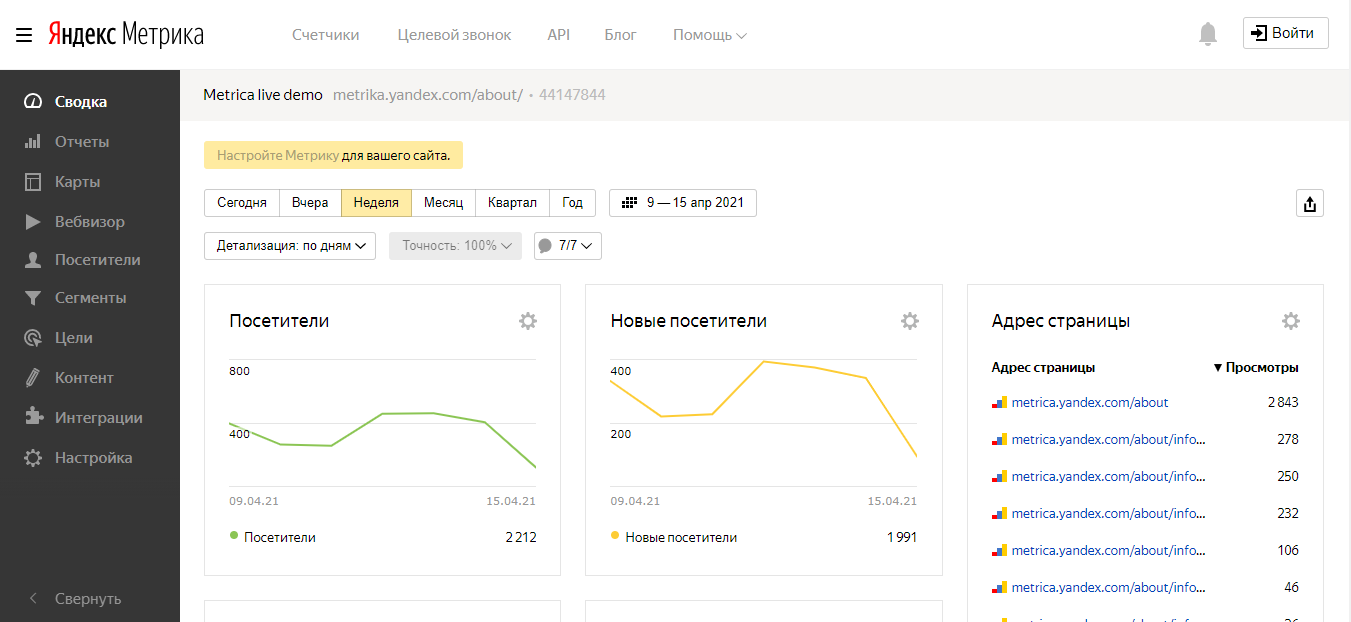


Рисунок – Интерфейс консоли Яндекс Метрики

### **3.3. SEO-анализаторы**

Также в Интернете можно найти множество онлайн-сервисов и плагинов для SEO-тестирования (sitemap, robots.txt, скорость загрузки, и др.). Многие плагины позволяют получить подробный отчет о состоянии сайта и указанием ошибок и предупреждений, что помогает в настройке сайта.

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

### **3.1. Проектирование базы данных и создание бэкенда**

В данном проекте использовалась СУБД MySQL и PHP-фреймворк Laravel с архитектурой MVC. Laravel имеет мощный инструмент для работы с базой данных – Eloquent. При данном подходе записи БД представляют собой объекты соответствующих классов, что позволяет задавать связи между моделями, работать с массовым присвоением, настраивать вычисляемые свойства, прописывать хуки жизненного цикла и многие другие полезные функции.

Структура проекта примерно такова:

- модели (соответствуют таблицам в БД);

- контроллеры (непосредственно логика работы с данными, рендеринг представлений);

- файлы роутов (определение правил маршрутизации, разграничение доступа к различным ресурсам);

- представления (шаблоны blade, в которых рендерится содержимое веб-страниц);

- медиафайлы, стили, скрипты;

- конфигурационные файлы.

Также с помощью роутинга и настройки middleware была реализована мультиязычность на сайте: при парсинге URL устанавливается текущая локаль приложения.

В данном проекте использовались следующие модели:

- User.php

- Category.php

- Project.php

- Article.php

- Comment.php

- Rate.php

- Subscription.php

Между моделями реализованы различные типы отношений:

- belongsTo – отношение «многие-к-одному». Например, статья блога принадлежит одному пользователю;

- hasMany – отношение «один-ко-многим»: пользователь создает множество проектов;

- belongsToMany – «многие-ко-многим»: статья имеет несколько тегов, при этом к каждому тегу относится много статей. Реализуется через промежуточною таблицу;

- morphMany – полиморфическое отношение «один-ко-многим»: и статьи, и проекты имеют комментарии. Обратное отношение morphTo.

Пример модели проекта:

*<?php*

*namespace App;*

*use Illuminate\Database\Eloquent\Model;*

*class Project extends Model*

*{*

*public static function boot()*

*{*

*parent::boot();*

*static::deleting(function ($project) {*

*$project->categories()->sync([]);*

*$project->comments()->delete();*

*$project->rates()->delete();*

*});*

*}*

*protected $fillable = ['title', 'subtitle', 'user\_id', 'slug'];*

*public function user()*

*{*

*return $this->belongsTo('App\User')->withDefault();*

*}*

*public function categories()*

*{*

*return $this->morphToMany('App\Category', 'taggable');*

*}*

*public function getDateAttribute()*

*{*

*$time = strtotime($this->created\_at);*

*if(time() - $time < 60) return \_\_('Just now');*

*if(time() - $time < 3600) return ceil((time() - $time) / 60).' '.\_\_('minutes ago');*

*if(time() - $time < 24 \* 3600) return ceil((time() - $time) / 3600).' '.\_\_('hours ago');*

*if(time() - $time < 7 \* 24 \* 3600) return ceil((time() - $time) / (24 \* 3600)).' '.\_\_('days ago');*

*return date('d.m.Y H:i', $time);*

*}*

*public function likes()*

*{*

*return $this->morphMany('App\Rate', 'rateble')->where('rate', 1);*

*}*

*public function dislikes()*

*{*

*return $this->morphMany('App\Rate', 'rateble')->where('rate', 0);*

*}*

*public function rates()*

*{*

*return $this->morphMany('App\Rate', 'rateble');*

*}*

*public function comments()*

*{*

*return $this->morphMany('App\Comment', 'commentable');*

*}*

*public function generateSlug()*

*{*

*if(!trim($this->title)) $this->slug = $this->id;*

*else {*

*$this->slug = \Str::slug($this->title);*

*$i = 0;*

*while(static::where('slug', $this->slug)->where('id', '<>', $this->id)->exists())*

*$this->slug = \Str::slug($this->title).'\_'.(++$i);*

*}*

*$this->save();*

*}*

*}*

Как правило, каждый метод контроллера соответствует определенному маршруту, обрабатывающему GET и POST запросы. Любой метод принимает объект класса Request, содержащий входные параметры и заголовки и возвращает Response (рендеринг представления, файл, перенеправление, ответ в формате json и т.д.).

Пример контроллера:

*<?php*

*namespace App\Http\Controllers;*

*use Illuminate\Http\Request;*

*use App\Article;*

*use App\Category;*

*class BlogController extends Controller*

*{*

*public function index($locale = 'en')*

*{*

*$articles = Article::withCount('likes')->withCount('dislikes')->withCount('comments')->get();*

*return view('blog/index', [*

*'articles' => $articles,*

*'title' => 'Blog',*

*]);*

*}*

*public function view($locale = 'en', $id)*

*{*

*$article = Article::where('slug', $id)->withCount('likes')->withCount('dislikes')->firstOrFail();*

*return view('blog/view', [*

*'article' => $article,*

*'title' => $article->title,*

*]);*

*}*

*public function create($locale = 'en')*

*{*

*$categories = Category::all();*

*return view('blog/create', [*

*'categories' => $categories,*

*'title' => 'New article',*

*]);*

*}*

*public function store(Request $request, $locale = 'en')*

*{*

*$article = \Auth::user()->articles()->create($request->all());*

*$article->categories()->sync($request->categories ?? []);*

*$article->generateSlug();*

*return redirect()->route('blog.view', [app()->getLocale(), $article->slug])->with('success', 'Article created');*

*}*

*public function edit($locale = 'en', $id)*

*{*

*$article = Article::where('slug', $id)->firstOrFail();*

*$categories = Category::all();*

*return view('blog/edit', [*

*'article' => $article,*

*'categories' => $categories,*

*'title' => $article->title,*

*]);*

*}*

*public function update(Request $request, $locale = 'en', $id)*

*{*

*$article = Article::where('slug', $id)->firstOrFail();*

*$article->update($request->all());*

*$article->categories()->sync($request->categories ?? []);*

*$article->generateSlug();*

*return redirect()->route('blog.view', [app()->getLocale(), $article->slug])->with('success', 'Article updated');*

*}*

*public function delete($locale = 'en', $id)*

*{*

*$article = Article::where('slug', $id)->firstOrFail();*

*$article->delete();*

*return redirect()->route('blog', app()->getLocale())->with('success', 'Article deleted');*

*}*

*}*

### **3.2. Создание фронтенд-части**

В Laravel для рендеринга представлений используется движок Blade. Он позволяет передавать в представление различные данные из контроллеров, использовать циклы, условия, различные хелперы. При этом в браузере отобразится код, скомпилированный в HTML. Полезная функция – вынесение повторно используемых элементов в шаблоны и последующее подключение, либо наследование от базового шаблона.

Пример файла шаблона:

*@extends('layouts.app')*

*@section('content')*

*<div class="bg-light py-5 shadow">*

*<div class="container">*

*<div class="d-sm-flex align-items-start justify-content-center justify-content-sm-between">*

*<h1 class="mb-4 mb-md-0">{{ \_\_('Projects') }}</h1>*

*<a href="{{ route('projects.create', app()->getLocale()) }}" class="btn btn-success btn-lg">{{ \_\_('Start new project') }}</a>*

*</div>*

*</div>*

*</div>*

*<div class="container py-5">*

*<div class="row">*

*@foreach($projects as $project)*

*<div class="col-lg-4 col-sm-6">*

*<div class="card h-100 shadow">*

*<a href="{{ route('projects.view', [app()->getLocale(), $project->slug]) }}" class="card-body text-dark text-decoration-none">*

*<h3 class="mb-2">{{ $project->title }}</h3>*

*<small class="d-block text-muted mb-4">{{ $project->user->name }}, {{ $project->date }}</small>*

*<p>{!! $project->subtitle !!}</p>*

*<div class="d-flex flex-wrap">*

*@foreach($project->categories as $category)*

*<div class="mr-3 text-primary">*

*#{{ $category->name }}*

*</div>*

*@endforeach*

*</div>*

*</a>*

*<div class="card-footer bg-white d-flex justify-content-between">*

*<div class="d-flex">*

*<div class="c-pointer d-flex align-items-center mr-4 like-btn" data-href="{{ route('project.like', $project->id) }}" data-likes=".likes-count{{ $project->id }}" data-dislikes=".dislikes-count{{ $project->id }}">*

*<span class="mr-1 likes-count{{ $project->id }}">{{ $project->likes\_count }}</span>*

*<span class="text-success">+</span>*

*</div>*

*<div class="c-pointer d-flex align-items-center like-btn" data-href="{{ route('project.dislike', $project->id) }}" data-likes=".likes-count{{ $project->id }}" data-dislikes=".dislikes-count{{ $project->id }}">*

*<span class="mr-1 dislikes-count{{ $project->id }}">{{ $project->dislikes\_count }}</span>*

*<span class="text-danger">-</span>*

*</div>*

*</div>*

*<small class="text-dark">{{ $project->comments\_count }} {{ \_\_('comments') }}</small>*

*</div>*

*</div>*

*</div>*

*@endforeach*

*</div>*

*</div>*

*@endsection*

*@section('scripts')*

*<script src="/js/likes.js"></script>*

*@endsection*

Для стилизации был использован CSS-фреймворк Bootstrap 4, встроенный в проект, а также созданы свои файлы стилей. Дизайн сайта простой, использованы светлые тона и блоки с тенью (рисунок 2).

Для клиентского программирования использовался VueJS – библиотека для создания интерактивных динамических страниц. С помощью данной библиотеки реализованы блоки обсуждения: ветка ответов выводится с помощью рекурсивного компонента, взаимодействие с серверной частью происходит через fetch-запросы к контроллеру API.

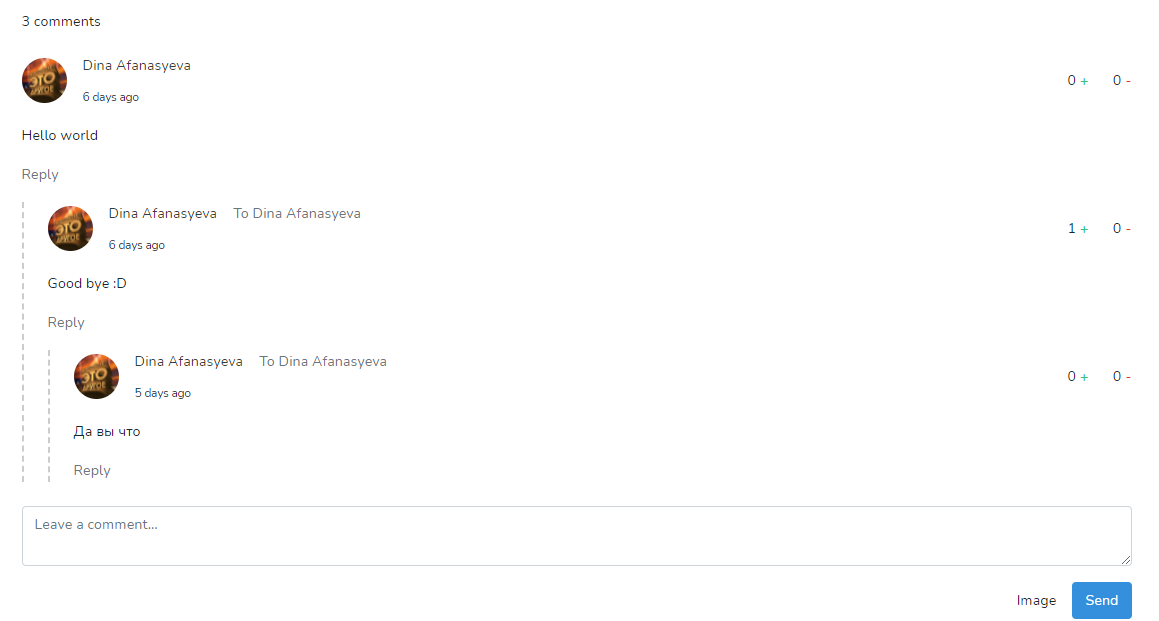


Рисунок 2 – Обсуждение статьи

Некоторые функции, такие как лайки, подписки, интерактивность элементов управления были написаны на чистом JavaScript.

### **3.3. Настройка SEO**

В секции head главного файла app.blade.php настроена разметка OpenGraph, канонические URL и предзагрузка:

*<meta property="og:url" content="{{ route($current, array\_merge($params, ['locale' => app()->getLocale()])) }}" />*

*<meta property="og:type" content="article" />*

*<meta property="og:title" content="{{ $metaTitle }}" />*

*<meta property="og:description" content="{{ $metaDescription }}" />*

*<meta property="og:image" content="{{ asset('images/logo.png') }}"/>*

*<meta property="og:image:secure\_url" content="{{ str\_replace('https', 'http', asset('images/logo.png')) }}" />*

*<meta property="og:image:type" content="image/png" />*

*<meta name="image" content="{{ asset('images/logo.png')}}" property="og:image" />*

*<link rel="canonical" href="{{ $current == 'home' ? url('/') : route($current, array\_merge($params, ['locale' => 'en'])) }}" />*

*<link rel="alternate" hreflang="en" href="{{ route($current, array\_merge($params, ['locale' => 'en'])) }}" />*

*<link rel="alternate" hreflang="ru" href="{{ route($current, array\_merge($params, ['locale' => 'ru'])) }}" />*

*<link rel="apple-touch-icon" sizes="120x120" href="/apple-touch-icon-120x120-precomposed.png" />*

*<link rel="apple-touch-icon" sizes="152x152" href="/apple-touch-icon-152x152-precomposed.png" />*

*<link rel="preload" href="{{ asset('css/app.css') }}" as="style" type="text/css">*

*<link rel="preload" href="{{ asset('css/custom.css') }}" as="style" type="text/css">*

*<link rel="preload" href="{{ asset('js/app.js') }}" as="script" type="text/javascript">*

*<link rel="preload" href="{{ asset('js/jquery.min.js') }}" as="script" type="text/javascript">*

*<link rel="preload" href="{{ asset('js/bootstrap.min.js') }}" as="script" type="text/javascript">*

*<link rel="{{ $current == 'home' ? 'preload' : 'prefetch' }}" href="{{ asset('images/logo.png') }}" as="image" type="image/png">*

*<link rel="{{ $current == 'home' ? 'preload' : 'prefetch' }}" href="{{ asset('images/beginner.jpg') }}" as="image" type="image/jpeg">*

*<link rel="{{ $current == 'home' ? 'preload' : 'prefetch' }}" href="{{ asset('images/expert.jpg') }}" as="image" type="image/jpeg">*

Всем изображениям был прописан alt тег.

Для настройки кэширования в .htaccess добавлено:

*Header set Cache-Control "max-age=290304000, public"*

Рендеринг sitemap.xml происходит динамически: пути к основным маршрутам статично заданы, а ссылки на статьи блога и проектов, профили пользователей генерируются по циклу вывода из БД.

ЧПУ реализованы с помощью генерации slug для статей, заметок, проектов и профилей, т.е. ресурсов, имеющих свой адрес страницы, например: http://new-project/ru/profile/dina-afanasyeva.

Robots.txt в корне сайта имеет следующее содержание:

*User-agent: \**

*User-agent: Googlebot*

*User-agent: Yandex*

*Allow: /*

*Host: http://new-project*

*Sitemap: http://new-project/sitemap.xml*

# **ЗАКЛЮЧЕНИЕ**

В ходе работы над курсовым проектом были изучены основные способы продвижения информационных ресурсов в сети Интернет. К ним относится как творческая и маркетинговая работа, так и технические настройки, которые должен выполнить веб-разработчик. Копирайтеры, маркетологи, дизайнеры работают над качеством контента, ориентированного на клиента. Задача разработчика – создать сайт должным образом, чтобы он выполнял свои функции стабильно, индексировался в поисковых системах и был доступен и удобен в пользовании. При этом каждый момент важен: будь то проектирование архитектуры бэкенда или стилизация элементов управления.

Исходя из результатов проведенного опроса, большой объем трафика в наше время идет с мобильных устройств. Из этого следует, что стоит большое внимание уделять адаптивном дизайну и оптимизации загрузки сайтов на различных устройствах.

Был создан веб-сайт со следующим функционалом:

- создание личного профиля, добавление фото, указание своих интересов, целей и сфер деятельности;

- написание и чтение блогов;

- создание проектов и их обсуждение;

- система оценок и рейтинга;

- добавление пользователей в подписки;

- глобальный поиск по сайту.

Также проведен опрос по соответствующей теме, изучен теоретический материал и на реальном проекте продемонстрированы базовые SEO- настройки. Выполнение данного курсового проекта также помогло улучшить свои навыки в веб-разработке, в частности, в работе с современными фреймворками. Таким образом, поставленные задачи выполнены и цели достигнуты.

К счастью, современные веб-технологии предоставляют разработчикам большие возможности, нужно только уметь мастерски пользоваться нужными технологиями и инструментами.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Eloquent – Relationships

<https://laravel.com/docs/8.x/eloquent-relationships>

2. Кэширование

<https://serpstat.com/ru/blog/kak-vkljuchit-i-nastroit-kjeshirovanie-sajta/>

3. Что такое UX/UI дизайн на самом деле?

<https://habr.com/ru/post/321312/>

4. Коды ответа HTTP

<https://developer.mozilla.org/ru/docs/Web/HTTP/Status>

5. Стандарт исключений для роботов

<https://ru.wikipedia.org/>

6. Bootstrap color utility classes

<https://getbootstrap.com/docs/4.0/utilities/colors/>