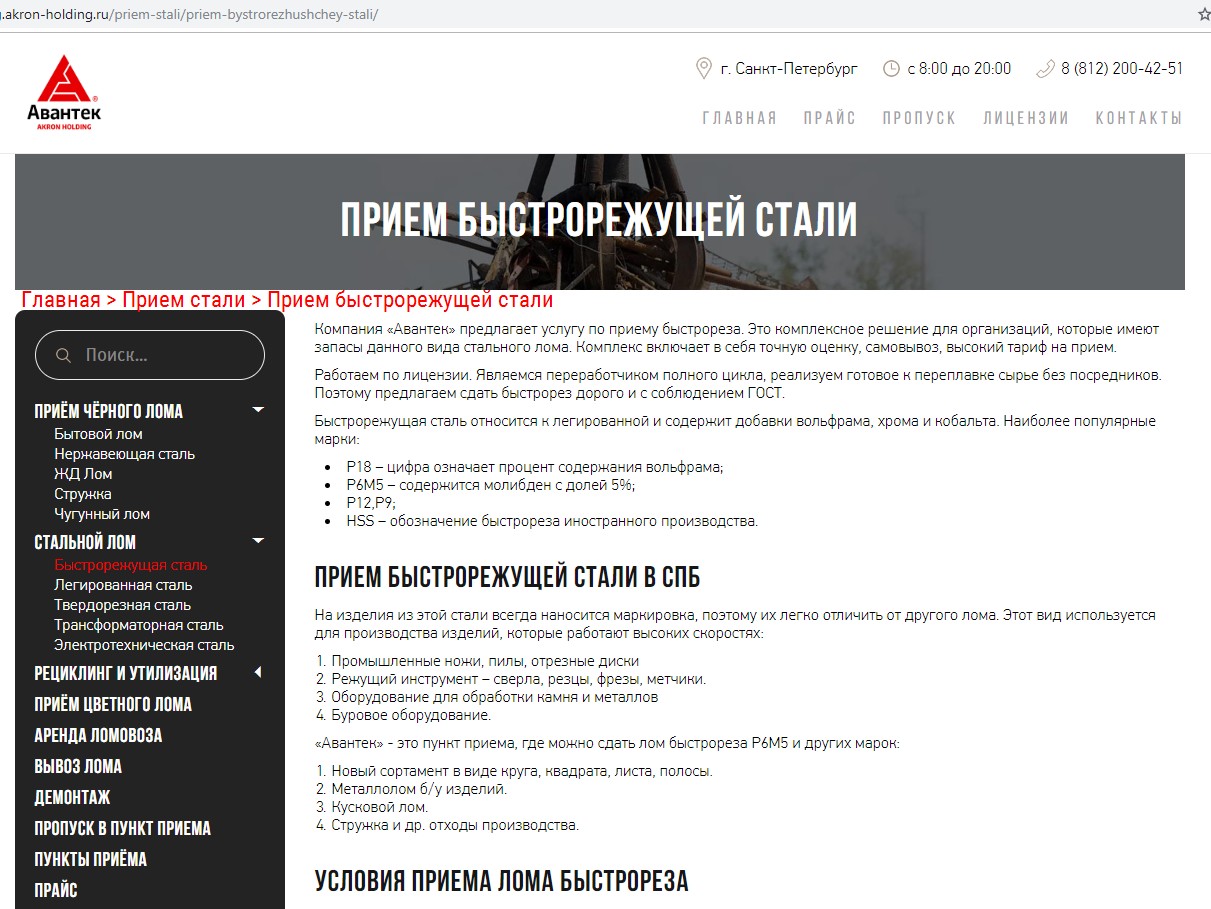
## Хлебные крошки

Реализовать хлебные крошки, где все значения кроме последнего - являются ссылками. Хлебные крошки по умолчанию - заголовок h1 страницы, но должна быть возможность редактировать хлебные крошки без изменения заголовка h1. На всех сайта разместить в одном месте.  


<http://ipic.su/img/img7/fs/kiss_246kb.1562867155.jpg>

## С точки зрения программирования Хлебные крошки необходимо разместить в заголовок (header.php). Чтобы не вставлять в каждую страницу. В этом случае строка будет в этом месте:

## 

Оформить отдельной плашкой. Возможно сдвинуть вправо до границы. Тут дезигнеру нужно поработать.

## 

## Формирование HTML карты сайта

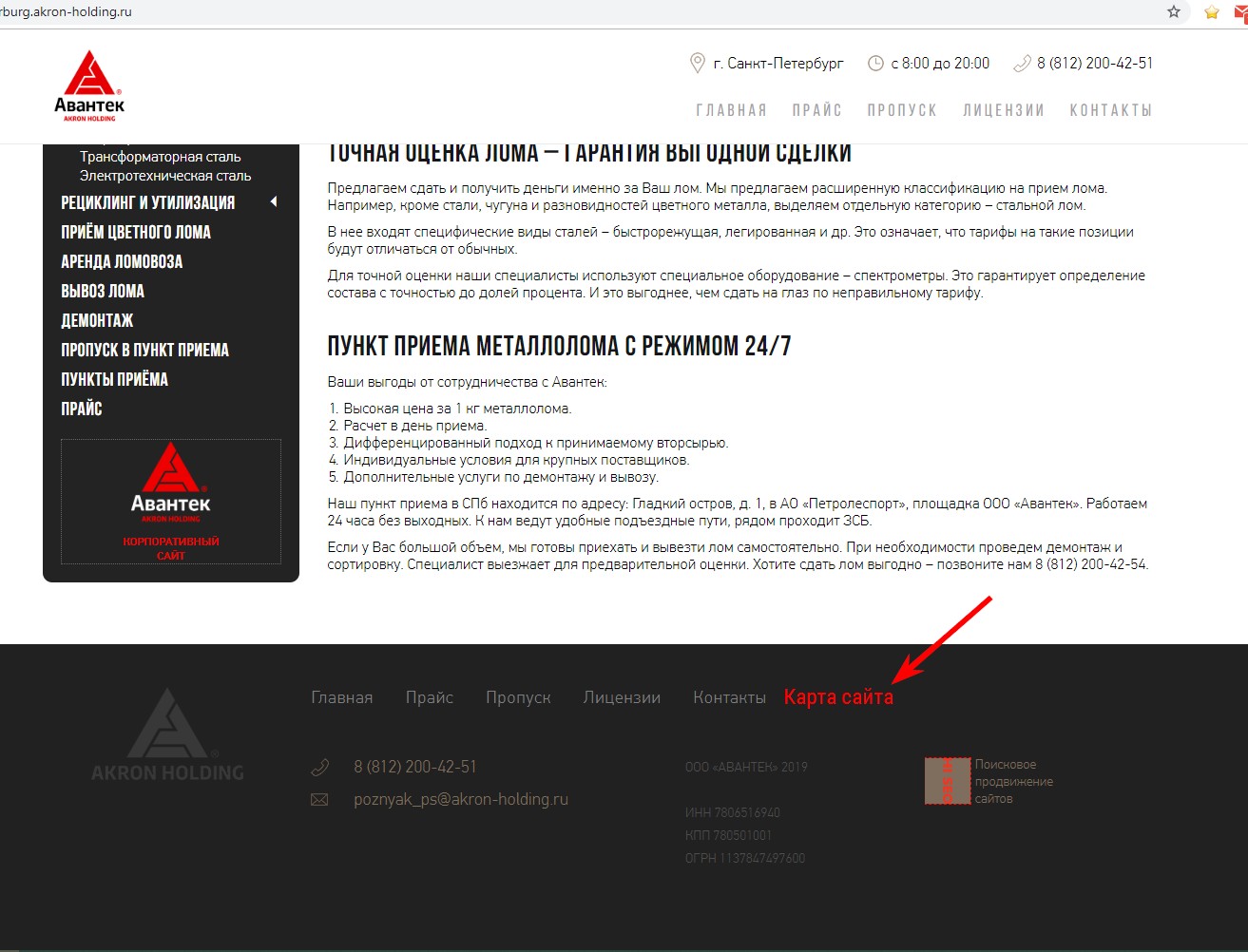
Корректно сформирована HTML-карта сайта приводит к улучшению поисковой индексации, а также помогает пользователю детальнее понять структуру сайта и найти необходимую страницу.

*Техническое задание*

Создать HTML карту сайта соответствующую следующим требованиям:

* ссылки на страницы в карте сайта расположены в логичной форме, в соответствии с иерархией страниц в виде маркированного списка;
* перечислены ссылки на все страницы сайта, которые открыты для индексации.

Ссылку на HTML карту разместить на всех страницах в футере сайта.



<http://ipic.su/img/img7/fs/kiss_229kb.1562867391.jpg>

Пример реализации карты сайта: <https://kortstroy.com/sitemap>===========================================

## Не очень уверен что это клиентам необходимо. У нас примитивная одно-двух уровневая структура сайта . Фактически на карте сайта будет повтор левого меню в более примитивном виде. Это дело нужно обсудить.

============================================

## Формирование XML карты сайта

*Техническое задание*

Cоздать файл sitemap.xml, отвечающий требованиям [sitemaps.org](http://www.sitemaps.org/ru/protocol.php), а именно:

* используются только обязательные теги <urlset>, <url> и <loc> и не используются необязательные – <lastmod>, <changefreq> и <priority>;
* включает в себя только канонические, открытые для индексации и сканирования страницы, отдающие код ответа 200;
* используется кодировка UTF-8;
* содержит не более 50 000 URL;
* размером не более 10 MB.

Карта сайта(xml) должна иметь расширение /sitemap.xml

Ссылку на XML карту сайта или на файл индекса разместить в Robots.txt.

## =========================================================

## На сайтах РП эта функция включена. Адрес включен в robots.txt.

=======================================================

## 

## Формирование XML карты изображений

*Техническое задание*

Создать файл sitemap-images.xml, отвечающий требованиям [Sitemaps для изображений](https://support.google.com/webmasters/answer/178636?hl=ru), а именно:

* указано пространство имен IMAGE таким образом:

xmlns:image="http://www.google.com/schemas/sitemap-image/1.1";

* используются все обязательные теги <urlset>, <url> и <loc>, <image:image> и <image:loc>;
* содержит только канонические, открытые для индексации и сканирования страницы, отдающие код ответа 200;
* используется кодировка UTF-8;
* содержит не более 50 000 URL;
* содержит не более 1000 изображений для каждого URL;
* имеет размер не более 10 MB.

Ссылку на карту-сайта для изображений добавить в документ.

## ===========================================

## На сайтах РП эта функция включена. Адрес включен в robots.txt.

==========================================

## 

## Настройка 301 редиректов

Необходимо, чтобы каждой странице соответствовал только лишь один URL. Любое его видоизменение, добавление лишних слешей, динамических приставок и т.п. должно приводить к 301 редиректу или возникновению 404 страницы.

Проверять:

* редирект на основное зеркало (c www или без);
* редирект страниц без слеша на конце URL, на страницы со слешем (или наоборот);
* index.php и подобные вариации;
* редирект с заглавных букв на строчные (при этом на сайте все URL должны быть со строчных букв);
* перестановка составляющих URL местами должна приводить к 301 или отдаче 404 кода ответа.

|  |  |  |
| --- | --- | --- |
| **Вид редиректа** | **Откуда** | **Куда** |
| Обработка слеша в конце URL | Не требуется | Не требуется |
| Редирект с дублей главной страницы | Не требуется | Не требуется |
| Лишние слеши в URL | Не требуется | Не требуется |
| URL только в нижнем регистре | <https://chermet-st-peterburg.akron-holding.ru/priem-chernyh-MEtallov/> | <https://chermet-st-peterburg.akron-holding.ru/priem-chernyh-metallov/> |

## Файл Robots.txt

*Техническое задание*

Разместить данный файл, который является универсальным для данных сайтов, при этом изменить директиву “Sitemap”, подставив туда текущую xml-карту сайта для данного сайта.

User-agent: \*

Disallow: /cgi-bin

Disallow: /?

Disallow: /wp-

Disallow: \*?s=

Disallow: \*&s=

Disallow: /search

Disallow: /author/

Disallow: \*?attachment\_id=

Disallow: \*/feed

Disallow: \*/rss

Disallow: \*/embed

Disallow: \*/page/

Disallow: /\*.pdf

Allow: \*/uploads

Allow: /\*/\*.js

Allow: /\*/\*.css

Allow: /wp-\*.png

Allow: /wp-\*.jpg

Allow: /wp-\*.jpeg

Allow: /wp-\*.gif

Allow: /wp-\*.svg

Allow: /wp-\*.pdf

Sitemap: https://chermet-st-peterburg.akron-holding.ru/sitemap.xml

## 

## 

## Формирование корректных http-заголовков

Http-заголовку несут в себе специальную информацию для браузеров пользователей. Необходимо настроить все http-заголовки сайта, чтобы сайт верно кешировался и корректно воспринимался поисковыми системами. Настраиваются такие заголовки:

* Last-Modified;
* Cache-Control;
* Expires;
* ETag;
* Vary;
* Date;
* Корректный ответ сервера при существующей/несуществующей страницах.

Проверять в инструменте <http://bertal.ru>

*Техническое задание*

Реализовать отдачу обязательных http-заголовков и ответов сервера со следующей информацией:

Last-Modified – отдавать реальную дату последнего изменения текущей страницы сайта.

Например,

Last-Modified: Sun, 04 Feb 2018 19:51:13 GMT

Проверять данный заголовок можно при помощи [сервиса](https://last-modified.com/ru/if-modified-since.html).

If-Modified-Since - на запрос с таким заголовком сервер должен возвращать код 200, если в Last-Modified указана дата новее, чем отправлено в запросе If-Modified-Since. Если содержимое с указанной даты не менялось (дата в Last-Modified старее), то сервер должен возвращать код ответа 304.

Cache-Control – отдавать для страниц сайта max-age=86400, для всех подключаемых ресурсов (CSS, JS, шрифты, изображения) – max-age=604800.

Например,

Cache-Control: max-age=86400, must-revalidate

или

Cache-Control: max-age=604800, must-revalidate

Expires – отдавать дату устаревания информации: для страниц сайта - текущая + 24 часа, для всех подключаемых ресурсов (CSS, JS, шрифты, изображения) – текущая + 168 часов.

Например,

Expires: Mon, 19 Oct 2015 02:28:12 GMT

ETag – отдавать идентификатор версии запрашиваемого объекта вместе со всеми статическими объектами (html, текст, CSS, JS, шрифты, изображения).

Vary – отдавать сообщение браузеру о том, что отдаваемый вариант кода сайта зависит от браузера клиента.

А именно,

Vary: User-Agent

Date – отдавать текущую дату загрузки страницы сайта/объекта.

Проверять все заголовки стоит при помощи сервиса <http://bertal.ru>.

## 

## Определение канонических URL

Необходимо внедрить атрибут “canonical” на все страницы сайта. В качестве значения — вписать текущий URL без добавления меток из контекстной рекламы и прочих элементов. Данная задача призвана решать 2 задачи:

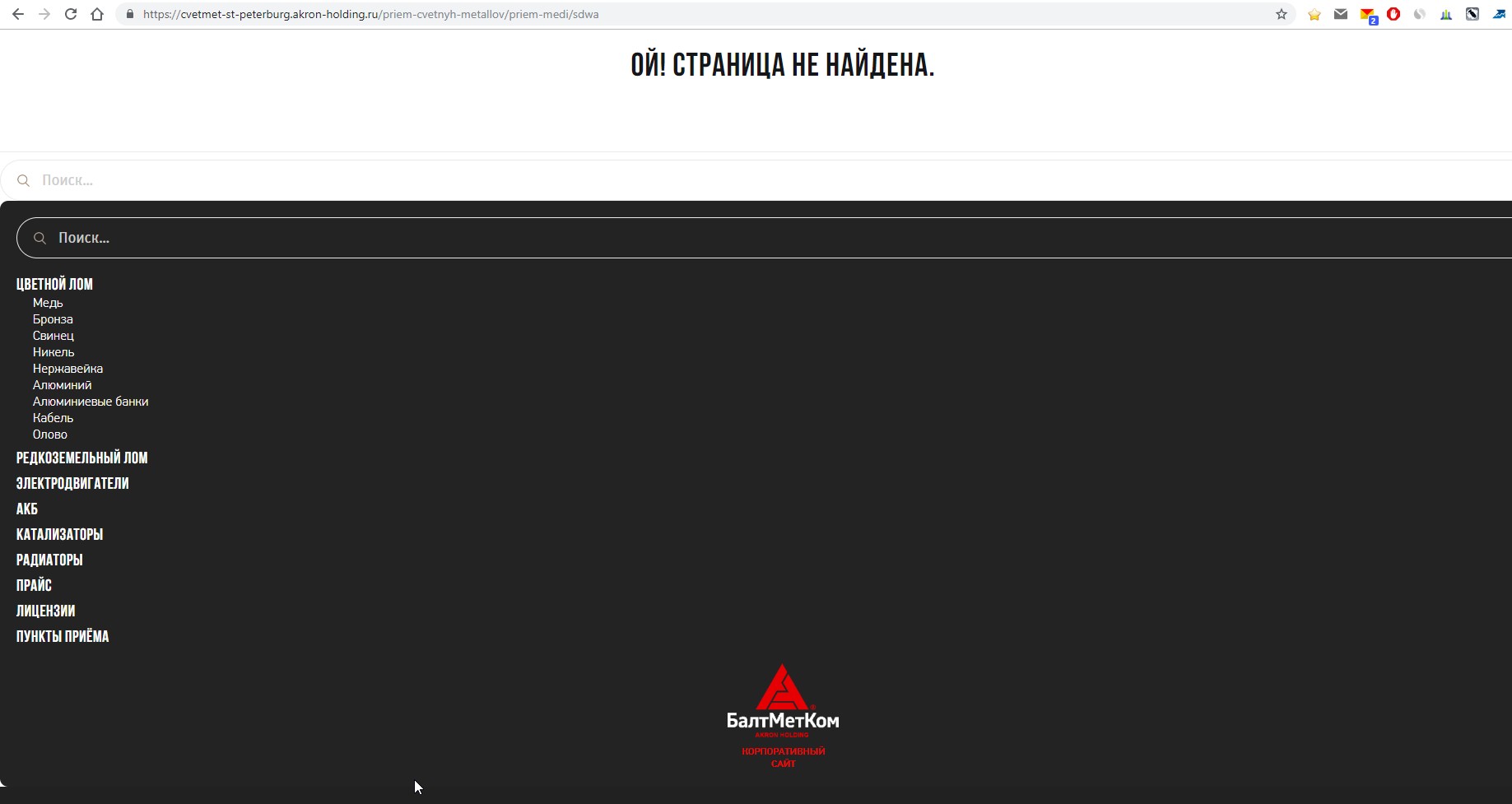
1. Дублирование контента внутри сайта.
2. Дублирование контента в случае, если сайт спарсят или закешируют внешние ресурсы.

<link rel="canonical" href="<https://cvetmet-st-peterburg.akron-holding.ru/priem-cvetnyh-metallov/priem-medi/>" />

для страницы: <https://cvetmet-st-peterburg.akron-holding.ru/priem-cvetnyh-metallov/priem-medi/>

## 

### Страница 404 ошибки

При вводе несуществующего url отдает код 404, но при этом страница не оформлена, и имеет вид “мобильной” версии:  


*Техническое задание*

Отдавать страницу 404 ошибки по запрашиваемому URL без переадресаций на другой URL.

Возвращать код ответа сервера “404 Not Found” при попадании пользователя на такую страницу.

Оформить страницу 404 ошибки в том же стиле, что и весь сайт, включая средства навигации.

Выводить на такой странице сообщение в явной и вежливой форме, что страница не доступна.

Например,

“Извините, запрашиваемая вами страница не найдена”.

Под сообщением о том, что страница не доступна, выводить ссылку на главную страницу, а также ссылки на самые популярные страницы сайта (прайс-лист, контакты).

При смене вида любого URL (удаление из URL части URI подкатегории или что-то в этом роде) - измененный URL должен отдавать ошибку 404.

### 

### Страница 50X ошибки

*Техническое задание*

Создать страницу, на которой заметно и вежливо по центру страницы написано “Извините, произошла ошибка сервера”.

Оформить страницу не в шаблоне сайта – темными шрифтами на светлом фоне.

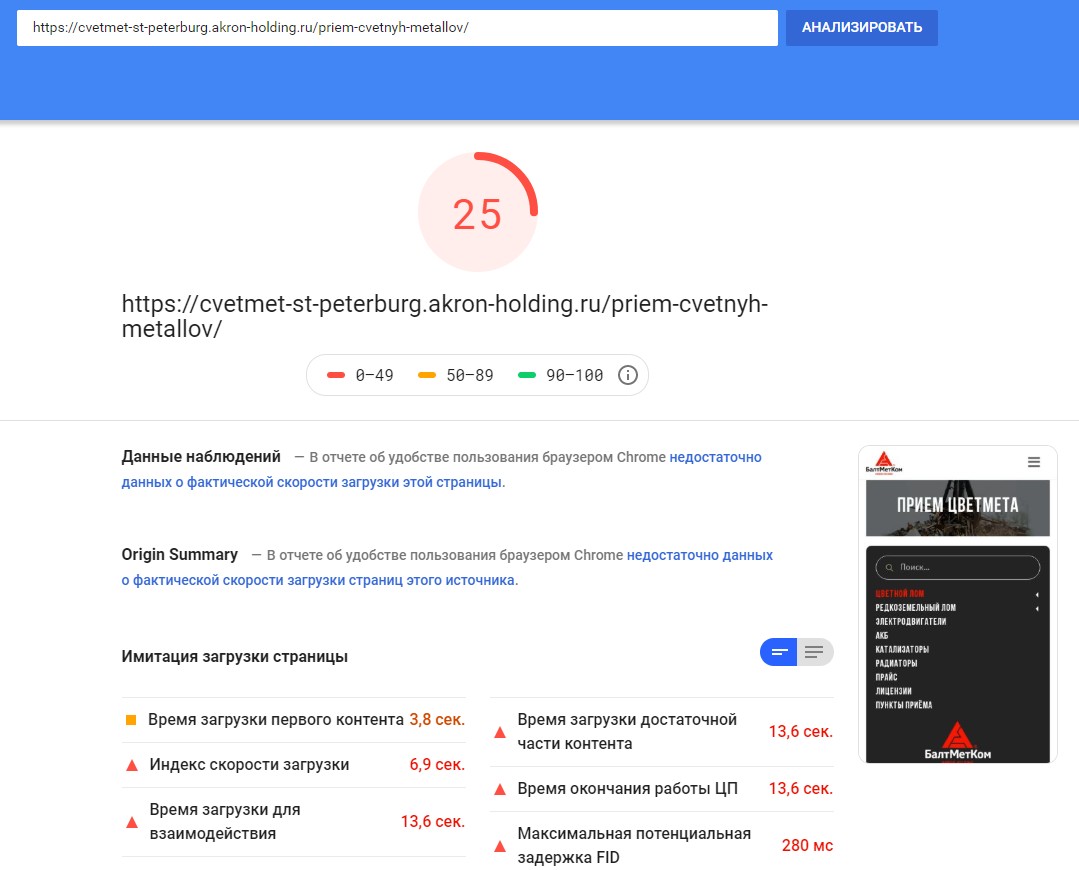
На странице разместить две ссылки/кнопки: “Вернуться на главную страницу” и “Обновить страницу”.

Под ссылками/кнопками разместить контактные телефоны, по которым пользователь может связаться, если сервер продолжает отдавать ошибку.

При возвращении сервером 500, 501, 502, 503, 504, 505, 507 и 510 ошибок – показывать пользователю эту страницу.

## 

## Оптимизация кода и скорости загрузки сайта



*Техническое задание*

Объединить внешние CSS и JavaScript файлы в минимальное количество файлов (чем меньше файлов – тем выше скорость и меньше дополнительных запросов к серверу; результат объединения – не более 3-х файлов каждого типа).

Внутренние фрагменты JavaScript и CSS кода вынести из кода в объединенные внешние файлы, при этом это не должно ухудшать скорость загрузки сайта.

Включить сжатие на веб-сервере:

Apache: используйте [mod\_deflate](http://httpd.apache.org/docs/current/mod/mod_deflate.html).

Nginx: используйте [HttpGzipModule](http://nginx.org/en/docs/http/ngx_http_gzip_module.html).

IIS: [настройте сжатие HTTP](https://technet.microsoft.com/en-us/library/cc771003(v=WS.10).aspx).

Сократить код JavaScript и CSS.

Для оптимизации и сокращения кода JS и CSS используйте инструмент: [YUI Compressor](http://yui.github.io/yuicompressor/).

Оптимизировать время ответа сервера до 200 ms или менее по данным [Google Console](https://www.google.com/webmasters/tools/crawl-stats).

Желательно довести все типы страниц сайта до 80-100 баллов по [Google Page Speed](https://developers.google.com/speed/pagespeed/insights/).

## 

### Атрибуты alt и title для изображений

*Техническое задание*

Ко всем изображениям на сайте(кроме тех, что используются для верстки) должна быть возможность прописать атрибуты alt и title в ручную.

### Микроразметка хлебных крошек

*Техническое задание*

После внедрения хлебных крошек, необходимо реализовать семантическую разметку хлебных крошек на всех страницах сайта с помощью Microdata, используя схему [BreadcrumbList](http://schema.org/BreadcrumbList) библиотеки [Schema.org](http://schema.org/) в следующем формате:

<ol itemscope itemtype="http://schema.org/BreadcrumbList">

<li itemprop="itemListElement" itemscope

itemtype="http://schema.org/ListItem">

<a itemprop="item" href="[url\_main]">

<span itemprop="name">Главная</span></a>

<meta itemprop="position" content="1" />

</li>

›

<li itemprop="itemListElement" itemscope

itemtype="http://schema.org/ListItem">

<a itemprop="item" href="[url\_2]">

<span itemprop="name">Категория 2 уровня</span></a>

<meta itemprop="position" content="2" />

</li>

›

<li itemprop="itemListElement" itemscope

itemtype="http://schema.org/ListItem">

<a itemprop="item" href="[url\_3]">

<span itemprop="name">Категория 3 уровня</span></a>

<meta itemprop="position" content="3" />

</li>

</ol>

Где

[url\_main] – URL главной страницы сайта;

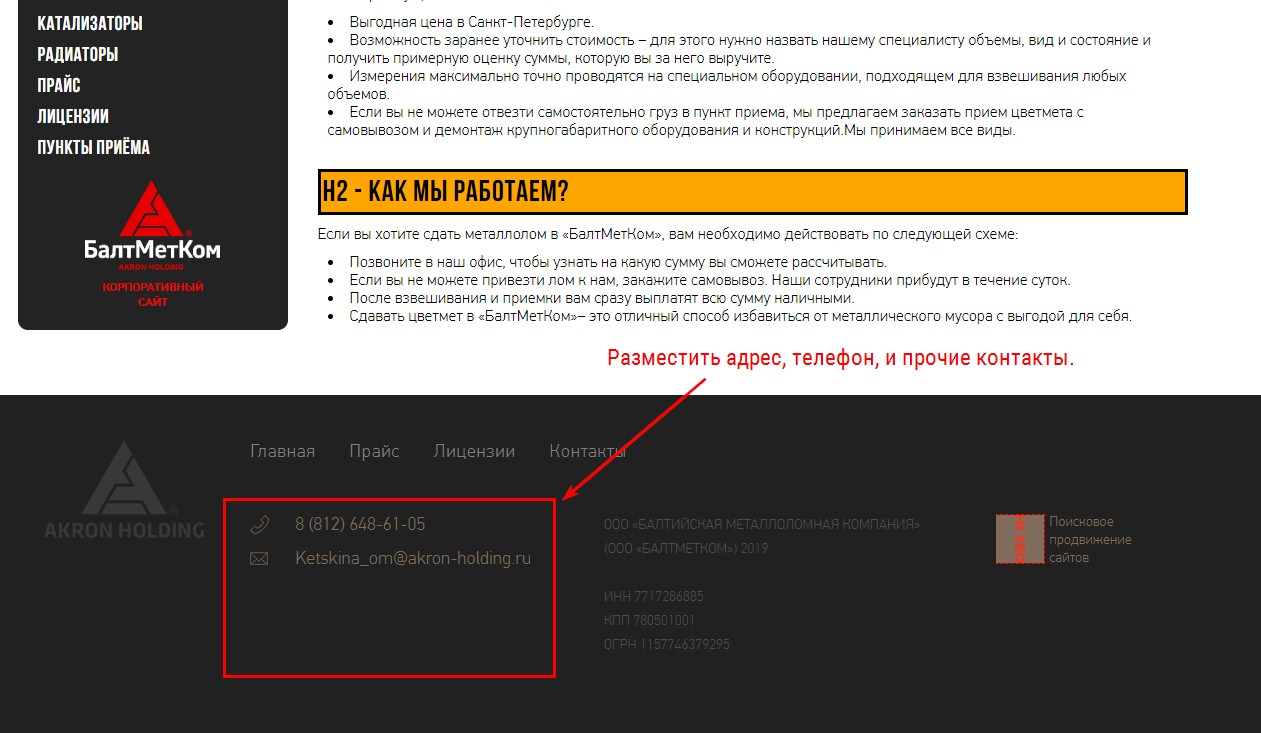
[url\_2], [url\_3] – URL категорий 2 и 3 уровня соответственно.

**[!]** Последнюю хлеюную крошку, которая выступает не ссылкой, [не размечать](http://ipic.su/img/img7/fs/kiss_159kb.1501845437.jpg).

### Микроразметка контактов организации

*Техническое задание*

Разместить все имеющиеся контактные данные в футере сайта:

Реализовать семантическую разметку контактных данных организации на всех страницах, которые их содержат, с помощью Microdata, используя схему [Organization](http://schema.org/Organization) библиотеки [Schema.org](http://schema.org/) в следующем формате:

<div itemscope itemtype="http://schema.org/Organization">

<span itemprop="name">[company\_name]</span>

<div itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">

<span itemprop="streetAddress">[address]</span>

<span itemprop="postalCode">[postal\_code]</span>

<span itemprop="addressLocality">[location]</span>

</div>

<span itemprop="telephone">[phone\_number]</span>,

<span itemprop="email">[email]</span>

</div>

Где:

[company\_name] – название компании;

[address] – адрес;

[postal\_code] – почтовый индекс;

[location] – город, страна;

[phone\_number] - телефонный номер (можно несколько номеров размечать);

[email] - e-mail (можно несколько e-mail размечать).

### 

### Микроразметка логотипа компании

*Техническое задание*

Разместить следующий код (предварительно загрузив изображение логотипа на хостинг) на всех страницах сайта в любом месте между тегами <body></body>:

<script type="application/ld+json">

{

"@context": "http://schema.org",

"@type": "Organization",

"url": "https://akron.ru",

"logo": "https://akron/logo.png"

}

</script>

### 

### 

## Протокол Open Graph

Частично реализован!

*Техническое задание*

Добавить перед тегом <head> на всех страницах сайта следующий код:

<html prefix="og: http://ogp.me/ns#">

Добавить между тегами <head></head> на всех страницах сайта следующее мета-описание:

<meta property="og:locale" content="ru\_RU" />

<meta property="og:title" content="[title]" />

<meta property="og:description" content="[description]" />

<meta property="og:url" content="[url]" />

<meta property="og:image" content="[img\_url]" />

Где

[title] – title данной страницы;

[description] – description данной страницы;

[url] – URL данной страницы;

[img\_url] – основное изображение данной страницы (если изображение выделить не представляется возможным - подставлять логотип сайта).

## 

## Like/share-кнопки социальных сетей

*Техническое задание*

Необходимо внедрить на карточках товара кнопки “Поделиться” от Яндекс. Визуально внедрить под картинками товара:  
<script src="//yastatic.net/es5-shims/0.0.2/es5-shims.min.js"></script>

<script src="//yastatic.net/share2/share.js"></script>

<div class="ya-share2" data-services="vkontakte,facebook,odnoklassniki,gplus,twitter"></div>

