**Работа 18 Подбор площадок для размещения ссылок**

Я выбрал тематику площадок и выбрал тематические прощадки по колличеству анкоров. В приложение к работе есть сис.

Вот скриншот из систематичского ядра.

Задание 1. В отчете письменно ответить на вопросы:

**1. Что такое кэш браузера?**

Кэш браузера - это папка с копиями некоторых данных со страниц, которые вы посещали. Обычно в кеше сохраняются те элементы страницы, которые вряд ли успели измениться за промежуток времени между двумя запросами, — музыка, видео, изображения, стили, скрипты.

**2. Как работает браузерное кэширование?**

Браузерное кэширование работает следующим образом: когда браузер загружает страницу или ресурс, он проверяет, есть ли в кэше копия этого ресурса с тем же URL. Если копия есть, браузер использует ее, а не загружает ресурс из интернета. Если копии нет, браузер загружает ресурс из интернета и сохраняет его в кэше для последующих запросов.

**2. Какие ресурсы нуждаются в кэшировании?**

Ресурсы, которые часто запрашиваются, например, изображения, стили, скрипты, и другие статические файлы, часто кэшируются, чтобы уменьшить количество запросов к серверу и увеличить скорость загрузки страницы.

**3. Перечислите HTTP заголовки, относящиеся к кэшированию.**

Cache-Control - Поле Cache-Control общего заголовка HTTP/1.1 используется для задания инструкций по механизму кеширования как в запросах, так и в ответах. Применяется для задания политик кеширования.

Expires - Самой важной здесь является директива "max-age=<seconds>" — максимальное время, в течение которого ресурс считается "свежим". В отличие от директивы Expires, она привязана к моменту запроса. К неизменяющимся файлам приложения обычно можно применять "агрессивное" кеширование. Примером таких статических файлов могут быть изображения, файлы стилей (CSS) или скриптов (JavaScript).

Pragma - Pragma является заголовком HTTP/1.0. Он не описан для HTTP-ответов и, таким образом, не может служить надёжной заменой общему заголовку Cache-Control протокола HTTP/1.1, хотя его поведение аналогично "Cache-Control: no-cache" когда поле заголовка Cache-Control опущено в запросе. Использовать его следует только для совместимости с клиентами HTTP/1.0.

ETag - Заголовок ответа ETag является непрозрачным для клиентского приложения (агента) значением, которое можно использовать в качестве сильного валидатора. Суть в том, что клиент, например, браузер, не знает, что представляет эта строка и не может предсказать, каким будет её значение. Если в ответе присутствует заголовок ETag, клиент может транслировать его значение через заголовок If-None-Match (en-US) будущих запросов для валидации кешированного ресурса.

Last-Modified - Заголовок ответа Last-Modified можно использовать в качестве слабого валидатора. Слабым он считается из-за того, что имеет 1-секундное разрешение. Если в ответе присутствует заголовок Last-Modified, то для валидации кешированного документа клиент может выводить в запросах заголовок If-Modified-Since.

**4. Опишите правильную схему кэширования ресурсов.**

Правильная схема кэширования ресурсов включает использование HTTP заголовков, таких как Cache-Control и Expires, чтобы указать браузеру, как долго кэшировать ресурсы. Также можно использовать ETag и Last-Modified для проверки, изменился ли ресурс с момента последнего кэширования. Если изменений не было, браузер может использовать кэшированный ресурс без дополнительных запросов к серверу.

**Задание 2.**

**Для своего сайта помотреть заголовки, относящиеся к кэшированию и их значения для разных типов файлов - изображения, стили, скрипты, шрифты.**

Заголовок Cache-control   
Поле Cache-Control общего заголовка HTTP/1.1 используется для задания инструкций по механизму кеширования как в запросах, так и в ответах. Применяется для задания политик кеширования.

Полное отсутствие кеширования

В кеше не должно сохраняться ничего — ни по запросам клиента, ни по ответам сервера. Запрос всегда отправляется на сервер, ответ всегда загружается полностью.

Cache-Control: no-store

Cache-Control: no-cache, no-store, must-revalidate

**Сделать вывод о схеме кэширования, принятой на вашем сайте.**

Вывод: Браузерное кэширование. Работа такого кэширования основана на том, что при повторном посещении, браузеру отдаётся заголовок 304 Not Modified, а сама страница или картинка загружаются из локального пользовательского кэша. Получается, что вы экономите на трафике между браузером посетителя и хостингом сайта. Соответственно, страница вашего сайта начинает загружаться быстрее.