

ПУТИ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ WEB-ПРИЛОЖЕНИЙ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шаграй А.М.

Карпович С.Е. – д.т.н., профессор

Целью данной работы является рассмотрение способов и средств повышения эффективности и надежности функционирования Web-приложений, в сфере маркетинга и продвижения медицинского оборудования.

Актуальность работы: внедрение правильных подходов и методов разработки сервисов и интерфейсов является важной задачей для отечественных Web-технологий. Использование такого подхода позволит привлекать все большее количество потенциальных клиентов на ресурсы, созданные отечественными разработчиками.

Разработка методологии проведения оценки сервисов и рекомендаций по устранению типовых ошибок Web-ресурсов осуществлялась с использованием опыта российских, белорусских и зарубежных специалистов в области Web-дизайна, дизайна интерфейсов, а также интернет-маркетинга: А. Лебедев, М. Пэймер, С. Смолл, С. Шарипо, Д. Уайт, Э. Петрова, А. Федоров, Т. Кудашкина, Д. Сирокер, Т. Уоккер, К. Уиссен и других.

Предложенное исследование направлено на обобщение критериев оценки качества ресурсов и разработку единой методологии устранения недостатков на данных ресурсах.

Была проведена проверка скорости загрузки сайта с помощью сервиса SeoLik.ru. Полученные данные говорят об относительно высокой скорости загрузки сайта, что является очень важным аспектом, так как зачастую пользователю необходимо быстро получать данные, размещенные на посещаемых им страницах. Но, тем не менее, такой процент скорости загрузки Web-сайта не гарантирует сохранение потенциальных клиентов на ресурсе. Проверка скорости загрузки сайта показана на рисунке 1.

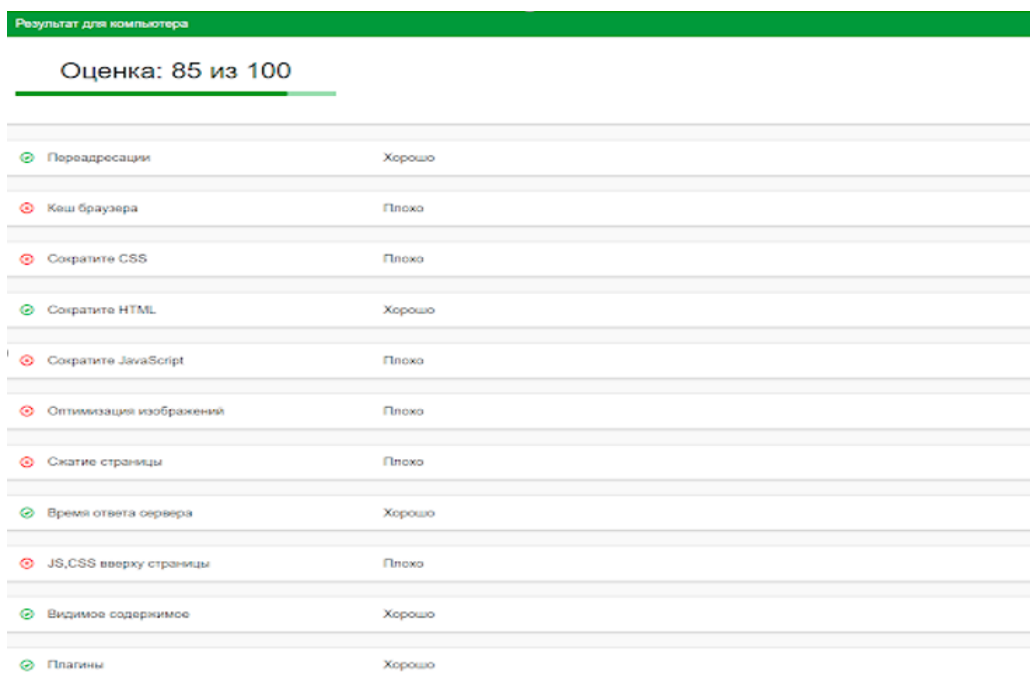


Рисунок. 1 - Проверка скорости загрузки сайта

Данный анализ показывает, что существуют недостатки, которые представляют собой:

1. Большие размеры CSS и JavaScript файлов, что существенно замедляет процесс загрузки страниц на компьютере пользователя.
2. Необходимость оптимизации изображений из-за их больших размеров. Большие изображения также замедляют загрузку страниц.
3. Сжатие страницы, что мешает браузеру быстро считывать программный код страницы.

Как можно видеть из рисунка 1 – JavaScript и CSS файлы подключены вначале страниц и имеют большие размеры, что мешает загрузке самого кода страницы. Код страницы не может подгрузиться в кэш браузера пока не будут загружены все подключенные файлы.

Для устранения данного недостатка необходимо разбить JavaScript и CSS код на отдельные блоки. Например, не стоит в одном и том же файле обрабатывать сценарии для заголовка сайта и помещать туда скрипты обрабатывающие слайдшоу. Точно так же не стоит задавать стили для таких блоков в одном файле.

Будет лучше, если за обработку событий и стилизацию таких блоков будут отвечать отдельные файлы. Такой подход позволяет существенно ускорять загрузку страницы.

Также для ускорения загрузки не следует подключать в начале страницы все стилевые файлы, которые относятся к другим страницам.

Загрузка файлов JavaScript осуществляется последовательно с кодом HTML страницы. То есть обрабатывая HTML код, как только браузер встречает JavaScript-файл, по умолчанию загрузка страницы приостанавливается и не продолжается пока не будет загружен скриптовой файл. Очень часто такая работа замедляет загрузку страницы, особенно если файлы больших размеров. Чтобы избежать такой ситуации можно подключить файлы обрабатывающие сценарии при помощи следующих атрибутов:

- атрибут `async`;
- атрибут `defer`.

В первом случае скрипт выполняется полностью асинхронно. То есть, при обнаружении `<script async src="...">` браузер не останавливает обработку страницы, а работает дальше. Когда скрипт будет загружен – он выполнится.

Скрипт также выполняется асинхронно, не заставляет ждать страницу, но есть два отличия от `async`:

- браузер гарантирует, что относительный порядок скриптов с `defer` будет сохранён;
- скрипт с `defer` сработает, когда весь HTML-документ будет обработан браузером.

Необходимо свести размер изображений к минимуму: это ускорит загрузку ресурсов. Правильный формат и сжатие изображений позволяет сократить их объем. Благодаря этому пользователи смогут сэкономить время и деньги. Следует провести базовую и расширенную оптимизацию всех изображений. В рамках базовой оптимизации обрезаются ненужные поля, уменьшается глубина цвета (до минимально приемлемого значения), удаляются комментарии и изображение сохраняется в подходящем формате.

Многие Web-серверы могут перед отправкой сжимать файлы в формат GZIP, используя собственные процедуры или сторонние модули. Это позволяет ускорить загрузку ресурсов, необходимых для отображения Web-сайта. Сжатие Web-страниц подразумевает удаление всех поставленных разработчиком в коде страницы пробелов. Данный подход позволяет браузеру быстрее обрабатывать код страницы, так как не приходится тратить ресурсы на дополнительный парсинг (распознавание) кода.

Результаты теста после устранения некоторых ошибок показаны на рисунке 2.

Результат для компьютера	
Оценка: 95 из 100	
Перезадресация	Хорошо
Кеш браузера	Плохо
Сократите CSS	Хорошо
Сократите HTML	Хорошо
Сократите JavaScript	Хорошо
Оптимизация изображений	Плохо
Сжатие страницы	Хорошо
Время ответа сервера	Хорошо
JS,CSS сверху страницы	Хорошо
Видимое содержимое	Хорошо
Плагины	Хорошо

Рисунок 2 - Результаты теста после устранения ошибок

Список использованных источников:

1. Ваховский Р. Что нужно знать о CSS. Преимущества CSS. Особенности и простота использования каскадных таблиц стилей [Электронный ресурс]. – / Р. Ваховский // Учебник WordPress. – 2016. Режим доступа – <https://wordpress-book.ru/poleznye-dopolneniya/chto-nuzhno-znatj-o-css/>. – Дата доступа: 08.02.2020
2. Роббинс Дж. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Дженнифер Роббинс; [пер. с англ. М. А. Райтман]. – 4-е издание. – М.: Эксмо, 2014. – 528 с.