

# Developing a Browser Extension for SEO Analysis and Report Generation

CS252 Project Report  
Presented to  
Dr. Thomas Austin  
Department of Computer Science  
San José State University  
Fall 2017

By  
Vikram Deshmukh  
December 9, 2017

## Abstract

Search Engine Optimization (SEO) has become an integral part of publishing any content on the web. The primary goal of SEO is to ensure higher visibility to the content. It is imperative to perform SEO on any web-related content that one wishes to make index-able by search engines. The main goal of this project was to develop a browser extension that performs SEO analysis and displays a report within the browser without requiring the user to leave the page.

Due to its convenience and extremely lucid documentation, Chrome was the browser of choice for developing this extension.

**Key Items**—*Search Engine Optimization, Chrome, Extension, TypeScript, ECMAScript6, Singleton*

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Importance of performing SEO . . . . .	4
1.2	Content Optimization . . . . .	4
1.3	Markup Optimization . . . . .	4
<b>2</b>	<b>Requirement Gathering</b>	<b>4</b>
2.1	The Rubric . . . . .	5
<b>3</b>	<b>Implementation</b>	<b>5</b>
<b>4</b>	<b>Deployment</b>	<b>6</b>
<b>5</b>	<b>Scope &amp; Future Work</b>	<b>7</b>
<b>6</b>	<b>Conclusion</b>	<b>7</b>

# 1 Introduction

Search Engine Optimization (SEO) is an important step when it comes to authoring content for the web. There are many tools and small organizations that are dedicated to performing SEO analysis of existing sites and making suggestions to improve content visibility. There are also some existing extensions in the Chrome Web Store and Firefox Add-ons Gallery. But these are either marred by bad reviews due to functionality issues or charge their users for premium access. The goal of this project is to develop an extension that perform SEO analysis and generates a report within the browser.

## 1.1 Importance of performing SEO

It is imperative nowadays to have any content on the web optimized for search engines. Some of the biggest benefits of ensuring that web pages are search engine optimized are summarized below [2]:

- Higher Page Rank
- Increases visibility
- Increases frequency of appearing in search results
- Increased website traffic (thereby revenue)

The process of SEO primarily involves of 2 actions:

## 1.2 Content Optimization

Content Optimization involves enhancing the quality of the content on the webpage by including internet buzz words like *Free*, *Online*, *Download* while also ensuring that most relevant keywords appear frequently and strategically throughout the page content. This ensures that the search engines give higher ranking to your page when any users search for those particular keywords.

## 1.3 Markup Optimization

Markup Optimization, unlike content optimization, has nothing to do with the content on the page. In fact, it is also about the syntactic details about how the page has been constructed. With the advent of HTML5, web developers can afford to indulge themselves in creating a webpage that uses newer tags that are semantically relevant while also adding some syntaxing sugar. For example, *header*, *footer*, *aside*, *video*, etc. Search engines tend to prefer web pages that adhere to the newer HTML specification.

# 2 Requirement Gathering

The very first step involved gathering information about SEO guidelines, processes, and understanding how various search engines parse web pages. This step also involved understanding the best practices, dos and donts of SEO in order to come up with a rubric to evaluate any given web page. This information was critical as it would be later used to create a rubric for evaluation of the web page.

## 2.1 The Rubric

After reading through Google SEO Guidelines [3] and SEO Analytics and Reporting section on Moz.com [6], here is the rubric that was created:

- Presence of exactly one H1 tag
- Missing meta tags like `description`, `keywords`, etc.
- Presence of broken links
- Page Redirect, if any, should be HTTP status 301 (Server-side redirect)
- Missing closing tags
- `img` tags with empty or missing `alt` attribute
- Empty tags in the document
- `description` or `title` too long/short

After testing the document markup against this rubric, any problem encountered with the markup is classified as *Critical*, *Important*, *recommended*. The page is then given an overall rating in the form of a smiley face: sad, neutral, or happy.

## 3 Implementation

Being an extension for the Chrome browser [4], opting for JavaScript was the obvious choice. But as we had seen during the Intro to JavaScript class [1], there are many ways of declaring a class and writing Object Oriented code in JavaScript with their own pros and cons. So, it was decided to use TypeScript to write clean, type-safe JavaScript.

The steps involved are mentioned below:

1. Read Page Source.
2. Sanitize HTML: Remove comments, scripts tags and inline scripts, link and style tags using the `HTMLSanitizer.js` file.
3. Analyze HTML against the rubric using the `HTMLAnalyzer.js` file.
4. Display the results of analysis in the extension window.

I decided to write HTML String sanitizer code and code that performs SEO analysis as reusable, independent libraries. Also, in order make these truly reusable, the libraries are written using the Singleton pattern to ensure only a single instance of the library is available to the developer for use.

Since there may be instances when a non-tech person needs to share the generated report with the developer so that the developer can fix the reported errors, there is also a feature for downloading the generated report as a PDF file. This nifty little feature was adding using an existing jqPDF library [7].

Given below is a code snippet that shows how Singleton classes are created using TypeScript and the resulting JavaScript.

```
// TypeScript
namespace HTMLAnalyzer {
    export function analyze(str: string) { }
}

// JavaScript version with closure
var HTMLAnalyzer;
(function (HTMLAnalyzer) {
    function analyze(str) { }
    HTMLAnalyzer.analyze = analyze;
})(HTMLAnalyzer || (HTMLAnalyzer = {}));
```

Listing 1: TypeScript code for singleton pattern

Here's a screenshot of the extension in action when triggered on a simple HTML page without an `title` or `meta` tags and with only a few hyperlinks.

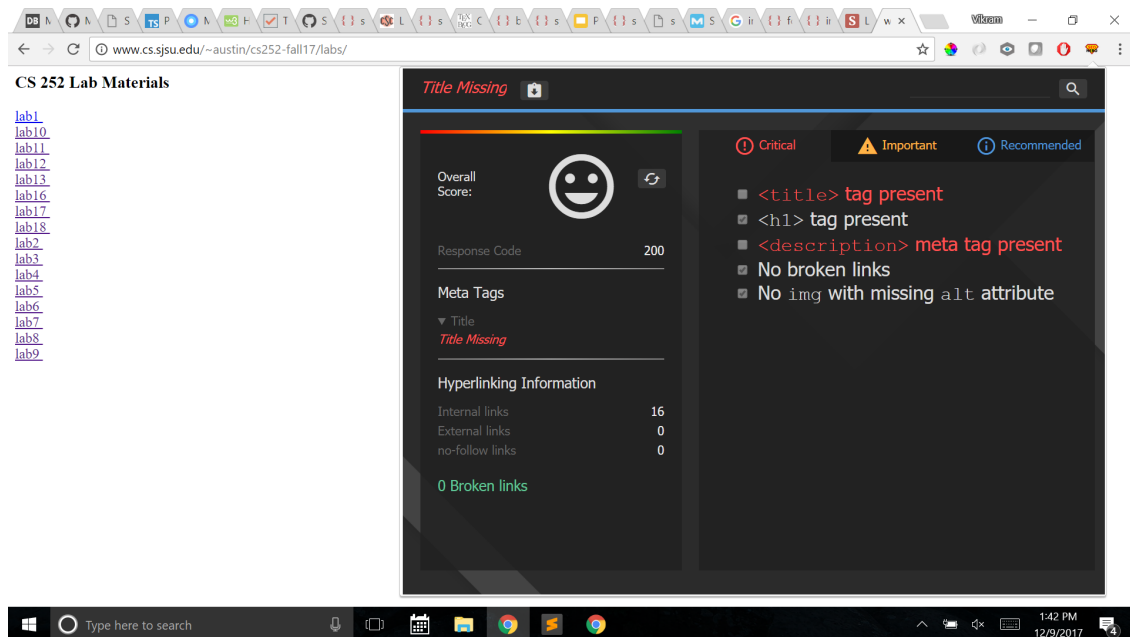


Figure 1: Screenshot of the extension in action

## 4 Deployment

Deploying to Chrome Web Store is an extremely simple and easy process. To publish your app to the Chrome Web Store, follow these steps [5]:

1. Create your app zip file
2. Create a developer account

3. Upload your app
4. Get app constraints (like app ID, OAuth token) and finish your app code
5. Pay the developer signup fee
6. Publish your app

Once the extension has been submitted for publishing, it gets deployed in the Chrome Web Store in about an hour, provided it passes preliminary, automated security checks.

## 5 Scope & Future Work

The current version of the extension only checks the content and links on the current page. The extension UI contains a search box in the top-right corner. The intention was to allow the user to analyze any other URL from within the extension, without having to open the URL in a separate tab.

Functionality can also be added to the extension to consume the `sitemap` of the particular domain, crawl through all the pages listed in the `sitemap`, and then generate a comprehensive list of all SEO related problems encountered by the extension.

Interested developers are welcome to contribute to the project by becoming a collaborator. Please request repository access by writing an email to [vikram.deshmukh@sjsu.edu](mailto:vikram.deshmukh@sjsu.edu).

## 6 Conclusion

This project provided an opportunity to learn the Chrome Browser Extension API. It also provided exposure to writing code that runs within the browser's sandbox. Additionally, there were ample opportunities to apply a bunch of concepts that are part of the course syllabus like Scoping in JavaScript, Event-based and Object-Oriented programming, in a real project.

## References

- [1] Thomas Austin. Introduction to JavaScript. <http://www.cs.sjsu.edu/~austin/cs252-fall17/slides/CS252-Day12-JavaScript.pdf>, 2017. [Online; accessed 07-Nov-2017].
- [2] Google Webmaster Central. Google Webmaster SEO Blog. <https://webmasters.googleblog.com/2008/11/googles-seo-starter-guide.html>, 2008. [Online; accessed 05-Oct-2017].
- [3] Google Developer. Search Engine Optimization: Starter Guide. <https://static.googleusercontent.com/media/www.google.com/en//webmasters/docs/search-engine-optimization-starter-guide.pdf>, 2017. [Online; accessed 05-Nov-2017].
- [4] Chromium Developers. Getting Started: Building a Chrome Extension. <https://developer.chrome.com/extensions/getstarted>, 2017. [Online; accessed 05-Oct-2017].
- [5] Google Developers. Publish in the Chrome Web Store - Google Chrome. <https://developer.chrome.com/webstore/publish>, 2017. [Online; accessed 05-Dec-2017].

- [6] Christopher Ferris, Chance Casey, and Coates Coates. SEO Analytics and Reporting — Moz. <https://moz.com/learn/seo/analytics-reporting>, 2017. [Online; accessed 15-Nov-2017].
- [7] James Hall. jqPDF: Client-side JavaScript PDF generation for everyone. <https://github.com/MrRio/jspdf>, 2017. [Online; accessed 15-Nov-2017].