DESIGN AND DEVELOPMENT OF SIMULATION TOOL FOR TESTING SEO COMPLIANCE OF A WEB PAGE – A CASE STUDY

Dr. P.G. Naik  
Professor  
Dept. of Computer Studies  
CSIBER, Kolhapur, India

Dr. R.S. Kamath  
Associate Professor  
Dept. of Computer Studies  
CSIBER, Kolhapur, India

Dr. S.S. Jamsandekar  
Assistant Professor  
Dept. of Computer Studies  
CSIBER, Kolhapur, India

Abstract: Efficient search engines enable the end user to avail the target information as early as and as accurately as possible. Search engine optimization conforms to the act of optimizing the webpage in particular and a website in general so as to acquire a highest ranking in search engine results. The ranking is based on some pre-defined criteria which SEO encapsulates for generating right signal to the search engine. The basic methodology adopted for this purpose primarily focuses on enriching the website content and improving readability by making the web pages more search engine friendly. Several optimization tools exist for rendering the website search engine friendly. But all these tools operate like a black box where the internal details are not immediately revealed to an end user. To facilitate this, the current work the authors have designed and developed a simulation tool for testing W3C compliance which is a necessary condition for making a webpage search engine friendly. The browsers do not complain against the violation of W3C compliance rules, however such violations do result in rendering the web page invisible to the search engine which in turn results less network traffic flow to the website. Among other things W3C and CSS compliance are the most basic rules which a website designer cannot afford to ignore. In the current work, the tool is designed and developed for the purpose which accepts an HTML document and computes its W3C compliance score. The tool meticulously analyzes the HTML code and generates several reports conforming to violation of rules, if any. Further, the tool has the provision for converting the document into W3C compliance document based on the specified ruleset.Finally, the tool has been tested by applying it to the case study of a hypothetical organization.

Keywords: CSS, Digital Marketing, HTML, JavaScript, Off-site Optimization, On-site Optimization, W3C Compliance

1. INTRODUCTION

In the era of Internet and worldwide web, digital marketing has emerged as the most promising and powerful trend in marketing which focuses on retaining the existing customers and acquiring the new customers based on the digital content. Academic institutions are not an exception to this. Due to the extensive competition among educational institutions for acquiring quality students, institutions are also adopting these advanced cutting edge technologies in establishing their brand names and reaching the aspirants across the globe. There is vast amount of digital content already present on Internet and more content is constantly generated at an exponential rate. Due to an intense competition, every organization intends to see itself at the top of the content generated by the search engines. It has been predicted that in the coming decade, in this entire competition only those organizations which quickly adopt themselves to these cutting edge technologies will survive which gives them an upper hand over their competitors and will consequently be the emerging winners conforming to the Darwinian principle of survival of the fittest. Currently most of the academic institutions are adopting Search Engine Optimization (SEO) for rendering their sites search engine friendly thereby sailing up to higher positions in search engine results.

Search engine optimization is a digital marketing methodology dealing with the measures taken towards improvement of a ranking of a website in search engine results thereby resulting in an increase in page views. SEO optimization can be achieved with the help of a set of rules pertaining to keyword selection, image optimization, focusing on rich web content to name a few which in turn renders the website search engine friendly. There exist a fistful of freelancing companies providing these facilities. SEO focuses on improving web visibility through a set of specific keywords. SEO in essence is a guiding theory for increasing web traffic. Keyword is the most crucial factor which plays a key role in influencing search engine results to a large extent. The generic optimization principle namely, early termination technique is employed by majority of search engines which involves computation of results based on indexing and employing pruning techniques which reduces the huge computational task involved in exhaustive traversal.

The most challenging task for the search engine is the tremendous workload to be handled by it and on an average the typical search engine needs to address hundreds of millions of queries on per-day basis involving billions of documents. Search engine addresses this issue by involving millions of machines distributed over multiple data centers with an intention to reduce query processing cost. The SEO techniques primarily fall into two distinct categories

- On-site optimization
- Off-site optimization

On-site optimization demands meticulous attention during the phase of website development targeting domain names, file names, directory structure, title, keyword selection and keyword density. Another area which requires meticulous attention is keyword consistency within the headings, anchor tags, alternate tags, descriptions etc.

Off-site optimization on the other hand focuses on inclusion of a website on various social networking sites for increasing network traffic to the website, availing a reputed web hosting service, blog publishing, link building etc. The parameter employed for measuring the off-site optimization
is Click-Through-Rate (CTR) of a website which is computed by dividing the number of times the page containing the link is displayed by number of times the link is clicked. For example if the page containing the link is clicked 10,000 times and the link is clicked 100 times, then CTR for the page is (100/10000)*100 % = 1%.

SEO optimization is based on four optimization techniques: structure optimization, keyword optimization, content optimization and link optimization. In the current work, authors focus on content optimization based on few key rules and validate the HTML document against those rules. Based on the validation process a compliance score is assigned to the document. The tool also contains the provision for converting invalid HTML documents into their valid counterparts. Currently, the tool operates only with a few set of rules which can easily be extended to incorporate additional rules.

2. LITERATURE REVIEW

With the rapid evolution of information technology and explosion in the number of websites, SEO technology is continuously gaining a tremendous importance and is catching the attention of every website developer. SEO techniques focus on few parameters pertaining to keyword selection, back linking, rich dynamic content and target towards gaining a better ranking in the search engine results [1, 2]. Website promotion is just a single application of SEO technology. The application of search engine optimization is many fold. Apart from improving the page ranking, there are numerous applications which exploit search engine optimization techniques such as spamming application, six sigma management application for which SEO provides a conceptual background [2-4]. The guidelines for building search engine friendly websites resulting in optimization of search queries have been presented [5-7]. Further, the authors present a review of search engine optimization tool as a part of digital marketing regime. Different categories of SEO tools are highlighted to aid website promotion. Search engine filtering system based on two-tier link extractor is presented to decrease the traffic of irrelevant pages in search result [8]. Search engine optimization based on soft computing technique is presented by Wang et.al [9] where a back propagation learning method of a neural network is employed for speedy retrieval of data from the web. Web caching based on semantic web technique is employed for designing a cluster of search engines with a goal to reduce load on the web server and latency time [10]. The different algorithms, lowest relative value algorithm, LUV algorithm and least weighted usage algorithm are employed to study their relative merits and demerits.

Research Gaps

Most of the work on search engine optimization techniques focus on on-site and off-site optimization techniques and primary focus is on keyword optimization. There are no simulation tools available for revealing the basic operation of these optimization tools. To account for this the current research focuses on design and development of simulation tool for testing SEO compliance of a website. The scope of the tool is rather too restricted to account for only the partial content optimization. The paper provides the first hand information for new researchers in this area.

3. CONCEPTUAL MODEL DESIGN

A. Problem Definition

XYZ institute is a leading management institute in south western Maharashtra accredited by NACC with “A” grade. The institute has in-house website development team which is actively involved in maintenance of the website. The management of the institute is interested in improving the website ranking in search engine queries.

B. Proposed Solution

The problem can be addressed at two levels. As a first alternative, hiring a freelancer, who works for 25% cheaper than SEO company. But at the same time it can be risky and time consuming as freelancer many not be aware of the specific website strategy and may struggle to understand the exact needs. Finding a good freelancer is time consuming and is more often a trial and error process.

Current Status

There are plethora of free and open source tools available on Internet which enable the analysis of the website content and give SEO ranking of the website. The tools generate the report pertaining to

- W3C Compliance Test
- CSS Compliance Test
- Keyword Analysis
- Broken links analysis

and suggest improvements for making the website SEO affine.

As a second alternative, the in-house tool can be designed and developed for verifying W3C and CSS Compliance which can then be input to available tools for improving it further. This enables the website to rank top in search engine results thereby improving the volume of traffic towards website by search engine. This demands the technical expertise which may be expected from XYZ institute due to the availability of in-house website development team.

In a simpler manner, at a first level search engine optimization can be achieved by targeting search terms which commonly include title, description, keywords, headings and alternate text. Few prominent rules for SEO are listed below:

C. Few Prominent Rules for SEO

Optimization rules exist on HTML filename and several HTML elements such as title, anchor element etc.

- Matching filenames with a page title
- Using hyphens rather than keyword for separating keywords.
- File size restriction to 101K, since text above it is chopped by most of the search engines.
- No use of frames and JavaScript embedded in HTML
- Usage of more text than HTML elements
- Using meta tag with the attribute “name” set to “keywords” and the “content” attribute listing about 25-30 keywords.
• Using Meta tag with the attribute “name” set to “description” and the “content” attribute listing about 150 characters.
• Limiting title size to maximum 9 words or 60 characters.
• Adding title to anchor element.

D. W3C Compliance Test
Browsers do not consider any violation against W3C compliance but search engines do. On useful link to validate the webpage is [validator.w3.org](http://validator.w3.org)

Rules of W3C Compliance
1. Every tag must be closed explicitly. Empty tag must use terminating slash.
2. The head tag is mandatory.
3. All tags must be in lower case.
4. All attributes must be enclosed in double quotes.
5. Form tags cannot be nested.
6. Use entity references wherever necessary.
7. CSS rules should be in lower case.

E. Application Architecture
The application architecture and operation of SEO analysis tool is presented in Figure 1.

![Figure 1. Application Architecture of SEO Analysis Tool](image)

The various steps involved in testing SEO compliance are depicted below:
1. Accept a webpage from the end user.
2. Check no. of significant rules (A rule is considered to be significant if the corresponding element exists in an input webpage)
3. Distribute the percentage among the significant rules equally.
4. Analyze the webpage and generate SEO compliance report.
5. Modify the webpage to render it more SEO friendly.

The tool operates by accepting an HTML document from an end user. HTML document is then analyzed to check the presence of some prominent tags. Seven rules for W3C compliance stated above are considered. Out of the seven rules stated above, some rules may be insignificant under the context based on the structure of HTML document input by an end user. The
primary focus is in locating un-terminated tags, unquoted attributes, missing head element etc. The score is equally distributed among the tags present and finally the SEO compliance score for the current ruleset is computed.

F. Implementation of Rules
All the seven rules carry equal priority and the percentage score reserved for each rule is given by,

\[ \text{Percentage} = \frac{100}{N} \]

where, N is the number of significance rules.
The net score is computed using the formula

\[ \sum S_i \]

where \(1 \leq i \leq N\)

The computation of a score for first four rules is depicted below:

**Implementation of Rule 1:**
The rule for the matching tags (un-terminated tags) is,

\[ S = E + T \]

where, S, E and T refer to the no of start tags, end tags and terminating tags, respectively.
The score awarded after testing rule 1 is

\[ \text{Score}_1 = (\text{Percentage}) \times \frac{(E+T)}{S} \]

**Implementation of Rule 2:**
Implementation of rule 2 is trivial since the full/zero score is awarded based on the presence/absence of a head node.

**Implementation of Rule 3:**
Score3= (Percentage) * L / T
where T has the same meaning as above and L is the number of tags in lower case.

**Implementation of Rule 4:**
Score4= (Percentage) * Q / A
where Q is the total number of quoted attributes and A is the total number of attributes.
The final SEO compliance score is then given by,

\[ S = S_1 + S_2 + \ldots + S_7 \]

4. RESULTS AND DISCUSSIONS
The model proposed above is implemented in Visual Basic 6 and is applied to the case study of XYZ educational institute stated above. The partial source code for implementation of tool is listed in Appendix A. The tool is tested for different test cases to check its validity and scope. The tool is also compared with the markup validation service provided by W3C by employing the website validator.w3c.org. The results are depicted in Figure 2(a) - Figure 2(d)
Figure 2(a)-2(d) Reports Generated by W3C Validator Tool

Figure 3 and 4 shows the keyword analysis and SEO score computed by employing open source SEO tools freely available on Internet.

Figure 3. Keyword Analysis Tool
Figure 4. Open Source Software for Computation of SEO Score

Figure 5(a)- Figure 5(d) shows the GUI of W3C validator tool implemented by the authors. Figure 5(a) depicts the ruleset currently employed by the tool. On analyzing the HTML document entered by the end user for the presence and absence of certain tags, the evaluation criteria computed by the tool is depicted in Figure 5(b). Figure 5(c) depicts in-depth analysis of HTML tag set. Based on the compliance of the rules, the score is computed and is displayed to an end user as depicted in Figure 6(a) and 6(b) for two different HTML documents.
Figure 5(a)-5(d) GUI of Custom W3C Validator Tool
Figure 6(a)-6(b) SEO Compliance Score Computed by Custom W3C Validator Tool

Figures 7(a)-(c) report the analysis of HTML document, seo2.html, with Figure 8. depicting the report generated by the tool in text format.
### HTML Page Analysis for D:\seo2.html

| HTML Element       | Total | Correct | Incorrect |
|--------------------|-------|---------|-----------|
| Tags               | 17 Tags | 16 Tags | 1 Tag      |
| Attributes         | 4 Attributes | 2 Attributes | 2 Attributes without Quotes |
| Short Tags         | 17 Short Tags |           |           |
| End Tags           | 7 End Tags |           |           |
| Terminated Tags    | 1 Terminated Tag |       |           |
| Non-Terminated Tag | 2 Non-Terminated Tags |     |           |

### Evaluation Scheme

| Rule | Significant (Yes/No) | Percentage (%) | Percentage Assigned (%) |
|------|----------------------|----------------|-------------------------|
|      | 1. Yes               | 25.00%         | 10.00%                  |
|      | 2. Yes               | 25.00%         | 00.00%                  |
|      | 3. Yes               | 25.00%         | 20.00%                  |
|      | 4. Yes               | 25.00%         | 30.00%                  |
|      | 5. Yes               | 00.00%         | 00.00%                  |
|      | 6. No                | 00.00%         | 00.00%                  |
|      | 7. No                | 00.00%         | 00.00%                  |

**SEO Compliance Score**: 35.42%
Converting the Invalid HTML document into Valid HTML Document.

The structure of HTML document, “seo.html” input to the tool is shown in Figure 9. As revealed from the HTML content head tag is missing, some tags (hr and html end tag) are in uppercase. Br tags are not terminated.

Figure 10(a) and (b) depict the report generated on performing the tag analysis by the tool in grid and text format, respectively.
On selecting the “Convert” option from the main menu, a new file with the name `<filename>_new.html` is created taking care of all the rules, where `<filename>` is the primary filename of the HTML file entered by an end user. The structure of the new document created by the tool is shown in Figure 11.
On analyzing the converted HTML document, the reports generated by the tool are depicted in Figure 12.
Figure 12. Reports Generated on Analyzing HTML Document Generated by Tool
5. CONCLUSIONS AND SCOPE FOR FUTURE WORK

Search engine optimization conforms to the act of optimizing the website so as to acquire a highest ranking in search engine results. The ranking is based on some predefined criteria which SEO encapsulates for generating right signal to the search engine. Hence knowledge of these criteria enables in rendering the website visible to most of the search engines and being listed at the top of the search results generated by a search engine. In the current research, the authors have considered a set of validation rules for W3C compliance and a tool is designed and implemented to generate a W3C Compliance score for a web page. The tool also enables converting an invalid HTML document to its valid counterpart by incorporating the necessary changes to the document by referring to the rule set. The tool acts as a first-hand simulation tool to understand search engine optimization issues. The scope of the tool can be further broadened to incorporate few more rules corresponding to page ranking, back linking, keyword density calculation etc. to widen its scope.

REFERENCES

1. Constantinos Dimopoulos, Sergey Nepomnyachiy, Torsten Suel, “Optimizing Top-k Document Retrieval Strategies for Block-Max Indexes”, IEEE Conference, ACM, 2013, 113-122.

2. Lihong Zhang, Jianwei Zhang, Yanbin Ju, in “The Research on Search Engine Optimization Based on Six Sigma Management”, in IEEE conference 2011, 101-103

3. Antriksha Somani, Ugrasen Suman, “Counter Measures against Evolving Search Engine Spamming Techniques”, IEEE Conference, 2011, 214-217.

4. H. L. Hsing, H. J. Chen, L. Me. Huang and H. H. Yi, “The application of search engine optimization for internet marketing: An example of the motel web sites”, ICCAE Chung Hua University Taiwan, vol. 1, (2010), pp. 380-383.

5. P. T. Chung, S. H. Chung and C. K. Hui, “A web server design using search engine optimization techniques for web intelligence for small organizations”, LISAT IEEE Long Island University Brooklyn NY USA, (2012), pp. 1-6.

6. M. Cui and S. Hu, “Search Engine Optimization Research for Website promotion”, ICM Transp. Manage., China, vol. 4, (2011), pp. 100-103.

7. Killoran, J.B., "How to Use Search Engine Optimization Techniques to Increase Website Visibility", Professional Communication, IEEE Transactions on (Volume:56, Issue: 1)

8. S. M. Kumar, P. Revathy and K. Vijayalakshmi, “Implementation of two-tier link extractor in optimized search engine filtering system”, IEEE IMSAA IT Dept. Hyderabad, India, (2009), pp. 1-4.

9. S. Wang, K. Xu, Y. Zhang and F. Lie, “Search Engine Optimization based on Algorithm of BP Neural networks”, CIS Inst. of Eng. & Tech., China, (2011), pp. 390-394

10. M. Rajaram and S. L. S. Vadvivu, “Web caching in Semantic Web based multiple search engines”, ICCIE IEEE Dept. of EEE Tirunelveli, India, (2010), pp.1-7.

Appendix A

Partial Source Code for Implementation of SEO Compliance Test Tool in Visual Basic 6

```vbscript
head = False
form1 = False
css = False
Tagupper = False
uppercnt = 0
cattr = 0
tattr = 0
total = 0

mnuEvaluation_Click

If (head = True) Then
    total = total + percentage
End If

Open filename For Input As #1
While (EOF(1) = False)
    Input #1, Line1
    n = InStr(1, Line1, ”<”)
    m = InStr(1, Line1, ”>”)
    While (n > 0 And m > 0)
        Tag = Mid(Line1, n + 1, m - 2)
        If (Tag = UCase(Tag)) Then
            Tagupper = True
        End If
        uppercnt = uppercnt + 1
    End If
    n = InStr(n + 1, Line1, ”<”)
    m = InStr(n + 1, Line1, ”>”)
    total = total + percentage
Wend
Close #1

Open filename For Input As #1
While (EOF(1) = False)
    Input #1, Line1
    n = InStr(1, Line1, ”=”)
    While (n > 0)
        n = InStr(n + 1, Line1, ”=”)
        tattr = tattr + 1
    Wend
    m = Mid(Line1, n + 1, 1)
    cattr = cattr + 1
Wend
Close #1

score2 = cattr / tattr * percentage
```
count = 4

Open filename For Input As #1
While (EOF(1) = False)
    Input #1, Line1
    n = InStr(1, ",""<head>"", Line1)
    m = Mid(Line1, n + 1, 1)
    If (n > 0 And m > 0) Then
        head = True
        count = count + 1
    End If
End While

n = InStr(1, ",""form"", Line1)
If (n > 0) Then
    form1 = True
    count = count + 1
End If

n = InStr(1, ",""style"", Line1)
If (n > 0) Then
    css = True
    count = count + 1
End If

n = InStr(1, "&", Line1)
If (n > 0) Then
    entity = True
    count = count + 1
End If

Wend
Close #1
percentage = 100 / count

n = countstarttags
m = countemptyendtags
x = n - m
y = countendtags
lowercnt = n - uppercnt
totalcnt = n
score1 = lowercnt / totalcnt * percentage

If (x = 2 * y) Then
    MsgBox "Tags match.."
Else
    MsgBox.Abs(x - 2 * y) & " Tag(s) of " & (x - y + m) & " total tags do not match.."
End If

diff = (x - y + m) - Abs(x - 2 * y)
score = diff / (x - y + m) * percentage

If (head = True) Then
    MSFlexGrid1.TextMatrix(2, 3) = Format(CStr(percentage), ",##.00") & "%"
Else
    MSFlexGrid1.TextMatrix(2, 3) = "00.00%"
End If

If (Tagupper = True) Then
    MSFlexGrid1.TextMatrix(3, 3) = Format(CStr(score1), ",##.00") & "%"
Else
    MSFlexGrid1.TextMatrix(3, 3) = Format(CStr(percentage), ",##.00") & "%"
End If

Private Sub mnuConvert_Click()
    Dim Line1 As String
    fname = Left(filename, Len(filename) - 5)
    Open filename For Input As #3
    While (EOF(3) = False)
        Input #3, Line1
        n = InStr(1, Line1, ",=")
        If (Trim(LCase(Line1)) = "<html>") Then
            Print #2, Line1
            If (head = False) Then
                Print #2, "<head>
                Print #2, vbTab & ",""<title>" & filename & ",</title>"
                Print #2, "</head>"
            End If
            ElseIf (Trim(LCase(Line1)) = "<hr>"") Then
                Print #2, "<hr/>
            ElseIf (Trim(LCase(Line1)) = "<br>"") Then
                Print #2, "<br/>
            ElseIf (n > 0) Then
                m = Mid(Line1, n + 1, 1)
                While (n > 0 And m > 0)
                    If (m <> """") Then
                        s1 = InStr(n, Line1, ","" "")
                        s2 = InStr(n, Line1, ",""""")
                        If (s1 > s2) Then
                            Min = s2
                        Else
                            Min = s1
                        End If
                        newline = Mid(Line1, 1, n) & ","" & Mid(Line1, n + 1, Min - n + 1)
                        Line1 = newline
                    End If
                End While
                Print #2, LCase(Line1)
                Else
                    Print #2, LCase(Line1)
                End If
            End If
        End If
    Wend
    Close #3
    Close #2
    MsgBox "Converted Successfully and Saved As " & fname
    Shell ("notepad " & fname)
End Sub

Private Sub mnuEvaluation_Click()
Dim Line1 As String

Label1.Visible = False
MSFlexGrid1.Visible = False

CommonDialog1.Action = 1
filename = CommonDialog1.filename
count = 4
Open filename For Input As #1

While (EOF(1) = False)
    Input #1, Line1
    n = InStr(1, "<head>", Line1)
    If (n > 0) Then
        head = True
        End If

    n = InStr(1, "form", Line1)
    If (n > 0) Then
        form1 = True
        count = count + 1
        End If

    n = InStr(1, "style", Line1)
    If (n > 0) Then
        css = True
        count = count + 1
        End If

    n = InStr(1, "&", Line1)
    If (n > 0) Then
        entity = True
        count = count + 1
        End If

    Wend

Close #1
percentage = 100 / count=
End If
End If
End If
End Sub

Private Sub mnuSeoCompliance_Click()
Label1.Caption = "HTML Page Analysis for " & filename
uppercnt = totalcnt - lowercnt
wattr = tattr - cattr

n = countemptyendtags
t = n
n = countstarttags - 2 * countendtags - countemptyendtags

Open filename & ".txt" For Output As #2
Print #2, "************************************************************
"&vbCrLf&vbCrLf&"SEO Compliance Report" &vbCrLf&vbCrLf&"************************************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&"SEO Compliance Score : " & Label3.Caption &vbCrLf&vbCrLf&"***********************************************
"&vbCrLf&vbCrLf&Close #2
MsgBox "Report saved as " & filename & ".txt"
End Sub

Public Function countstarttags() As Integer
    Dim Line1 As String
    Dim cnt As Integer
cnt = 0
Open filename For Input As #1
While EOF(1) = False
    Input #1, Line1
    n = InStr(1, Line1, "<")
    While (n > 0)
        cnt = cnt + 1
        n = InStr(n + 1, Line1, "<")
    Wend
Wend
Close #1
countstarttags = cnt
End Function

Public Function countemptyendtags() As Integer
    Dim Line1 As String
    Dim cnt As Integer
cnt = 0
Open filename For Input As #1
While EOF(1) = False
    Input #1, Line1
    m = InStr(1, Line1, "<")
    While (m > 0)
        cnt = cnt + 1
        m = InStr(m + 1, Line1, ">")
    Wend
Wend
Close #1
countemptyendtags = cnt
End Function

Public Function countendtags() As Integer
    Dim Line1 As String
    Dim cnt As Integer
Open filename For Input As #1
cnt = 0
Open filename For Input As #1
While EOF(1) = False
    Input #1, Line1
    n = InStr(1, Line1, ">")
    While (n > 0)
        cnt = cnt + 1
        n = InStr(n + 1, Line1, ">")
    Wend
Wend
Close #1
countendtags = cnt
End Function

Print #2, "************************************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
Print #2, " ***********************************************
"&vbCrLf&vbCrLf&If (head = False) Then
    Print #2, "Head tag is missing" &vbCrLf
End If
Print #2, "Total Tags : " &totalcnt&vbCrLf
Print #2, "Uppercase Tags : " &uppercnt&vbCrLf
Print #2, "Lowercase Tags : " &lowercnt&vbCrLf
Print #2, "Start Tags : " &countstarttags&vbCrLf
Print #2, "End Tags : " &countendtags&vbCrLf
Print #2, "Terminated Tags : " &t &vbCrLf
Print #2, "Non-Terminated Tags : " & n &vbCrLf
cnt = 0

Open filename For Input As #1
While EOF(1) = False
    Input #1, Line1
    n = InStr(1, Line1, "</")
    While (n > 0)
        cnt = cnt + 1
        n = InStr(n + 1, Line1, "</")
    Wend
Wend
Close #1

countendtags = cnt
End Function