**Geraldine Begley**

**S00159150**

**HDipComputing**

In my project proposal I was aiming to produce a website as I do not have one at the moment. I also aimed to produce a database of clients and projects I completed, macros for Word to help make editing more efficient and produce a program to import PDF indexes into my software program. I was able to achieve the first three aims and have produced the website, database and macros.

I did not produce a program to import PDF indexes into Word or my indexing software (SKY) due to time, ability and other restrictions. There are several barriers to converting the PDF index including converting indents for sub- and sub-sub-headings into tabs to produce an output file to import to SKY. Other problems include dealing with a two-column PDF index and retaining style and font information when dealing with scanned PDF documents.

The remainder of this document will deal with the first three parts of the project: website, database and macros.

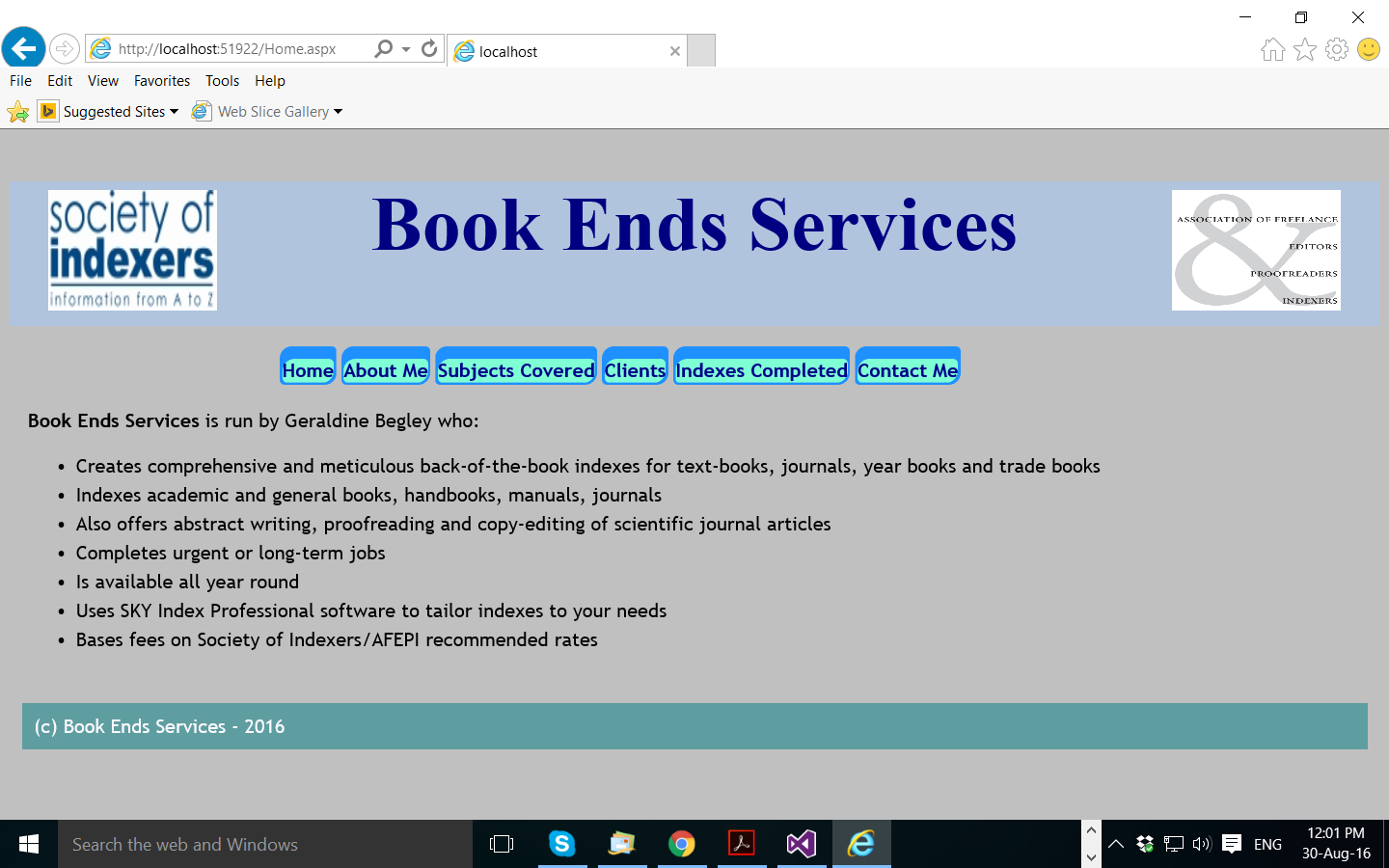
1. **WEBSITE**

I built a website for my small freelance business. The website is in a file called Semester3Project\_Website.

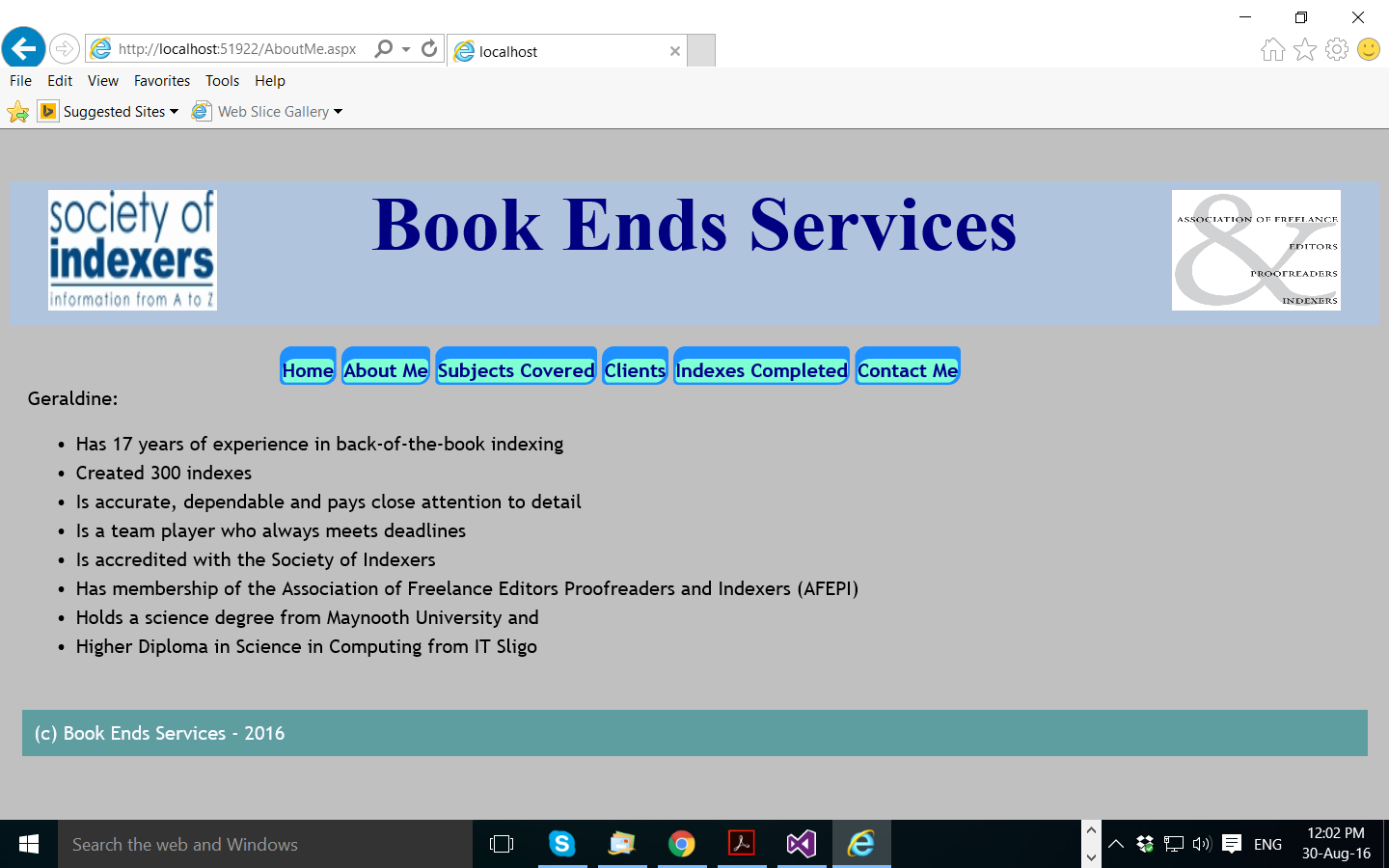
The website has 6 pages: Home, About Me, Subjects Covered, Clients, Indexes Completed and Contact Me.

I called the business “Book Ends Services”. Users can click on the Society of Indexers and AFEPI logos to get more information regarding indexing. On the last page users can contact me with queries etc.

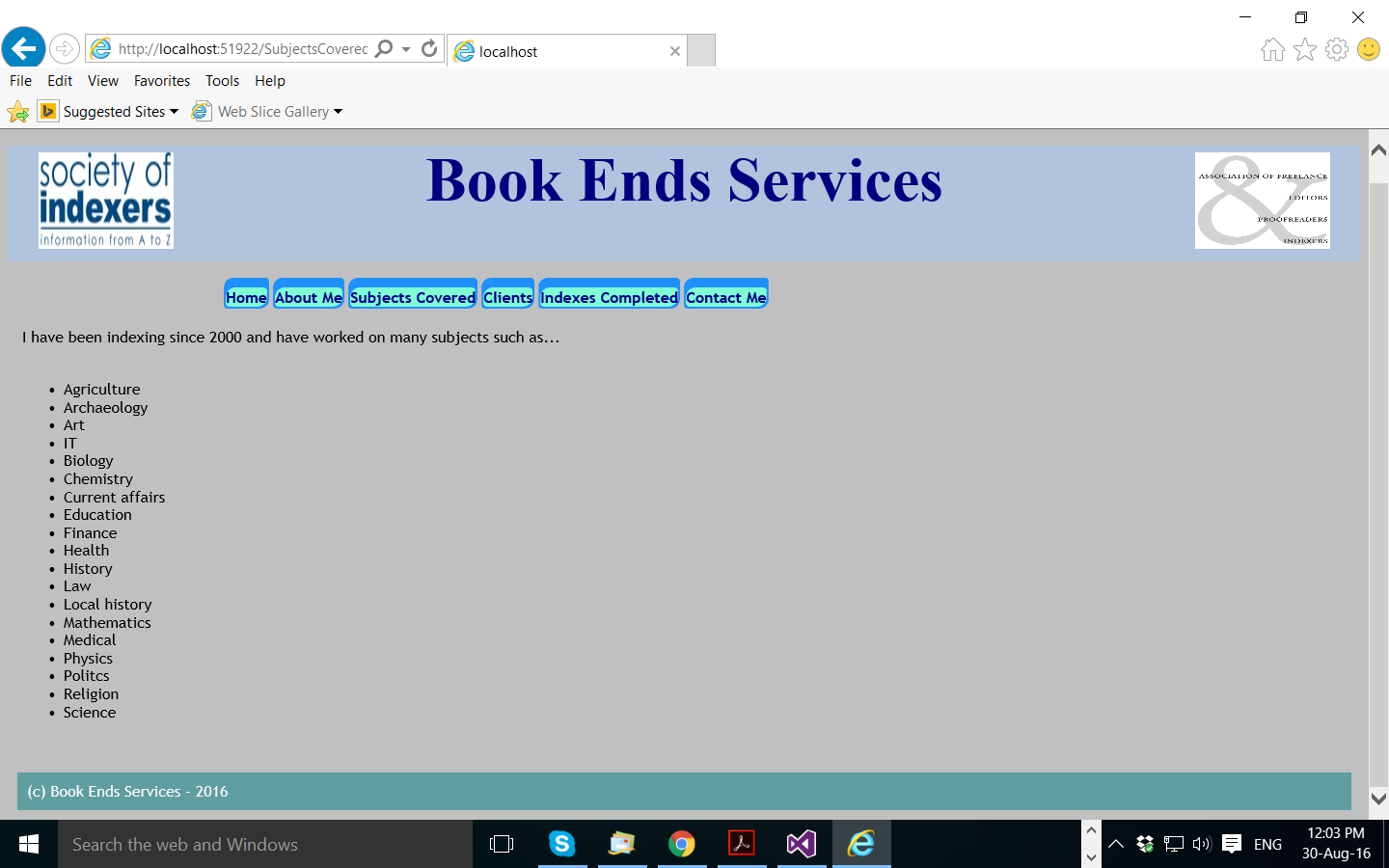
Page 1/Home page:



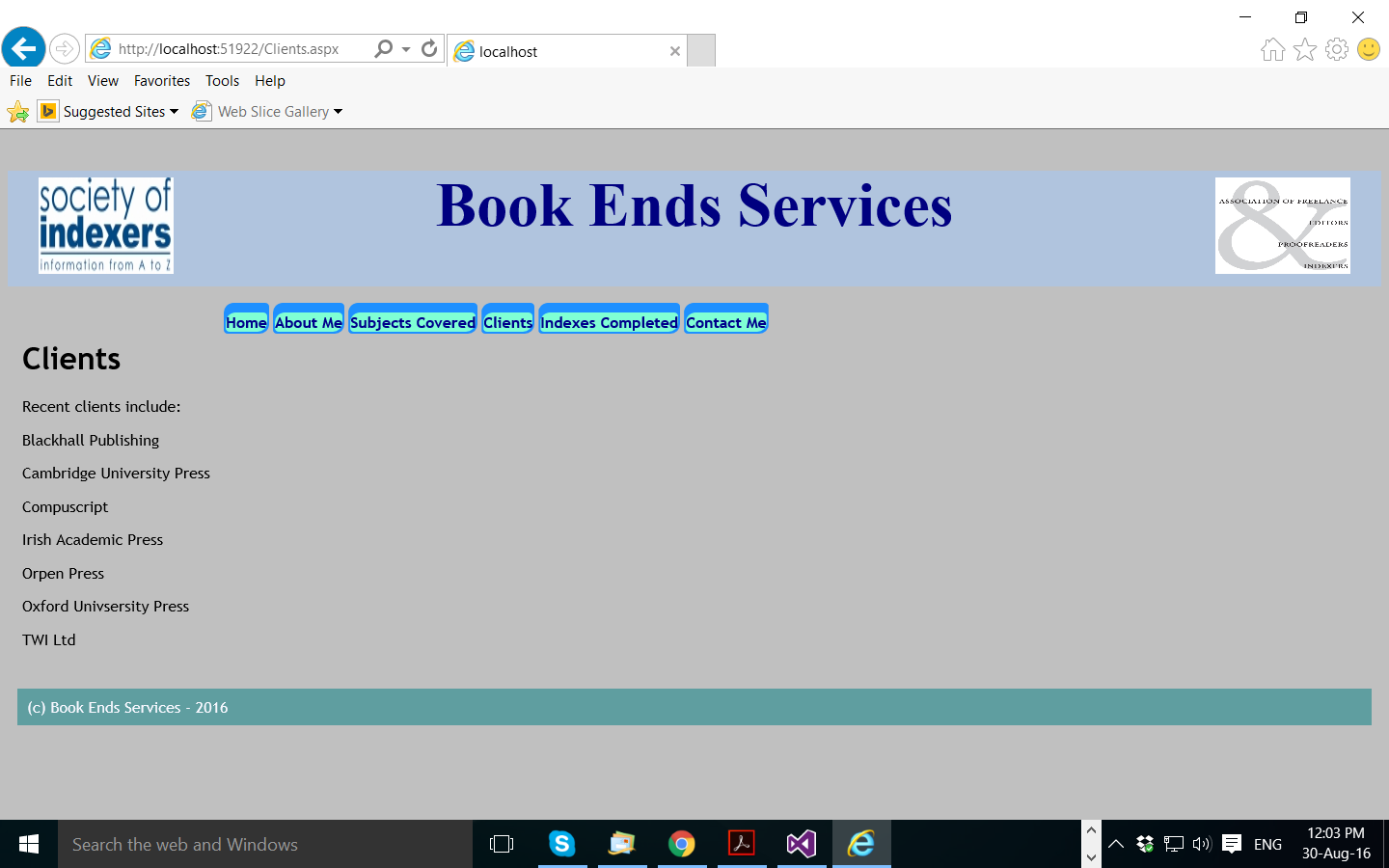
About Me page:



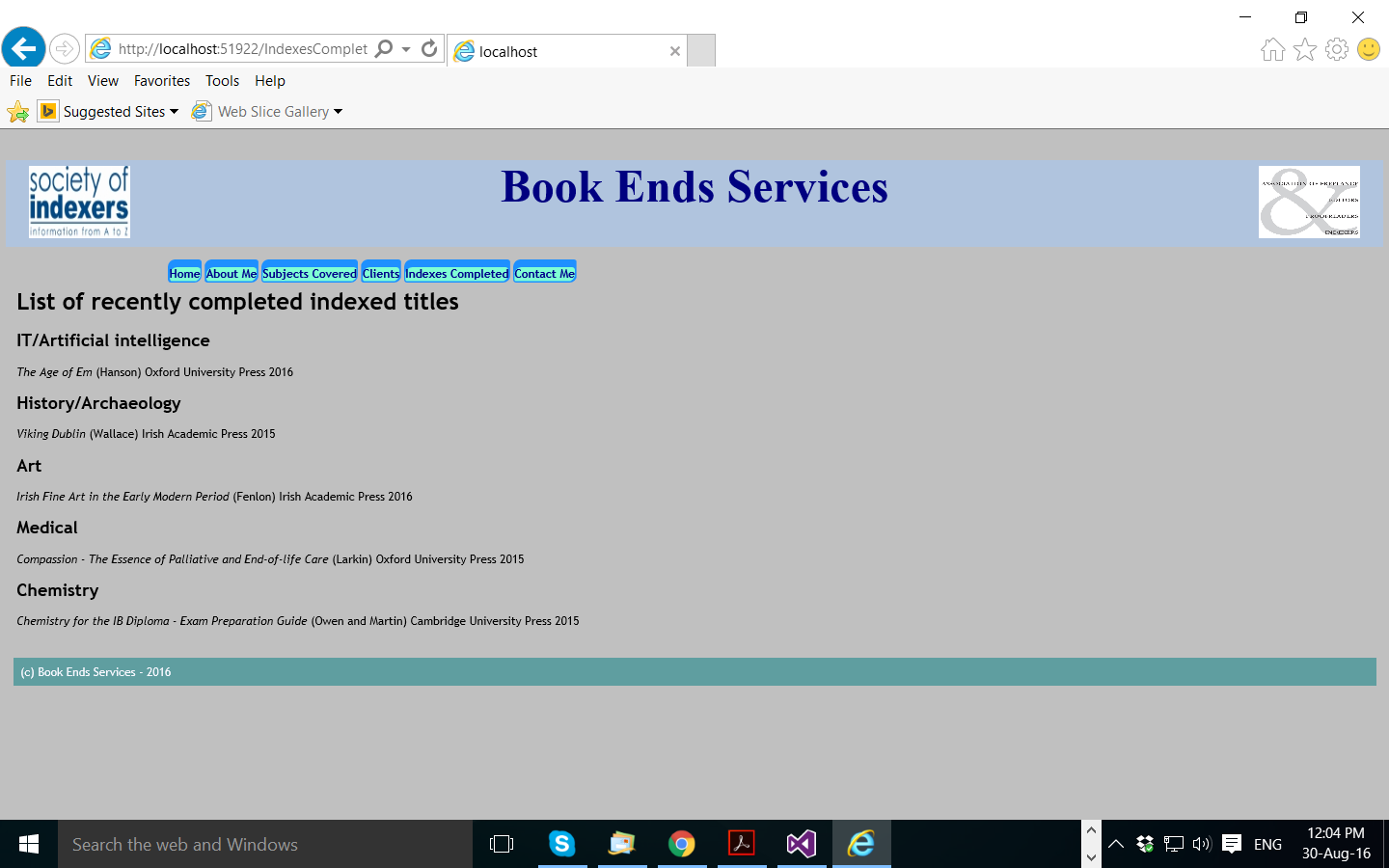
Subjects covered page:



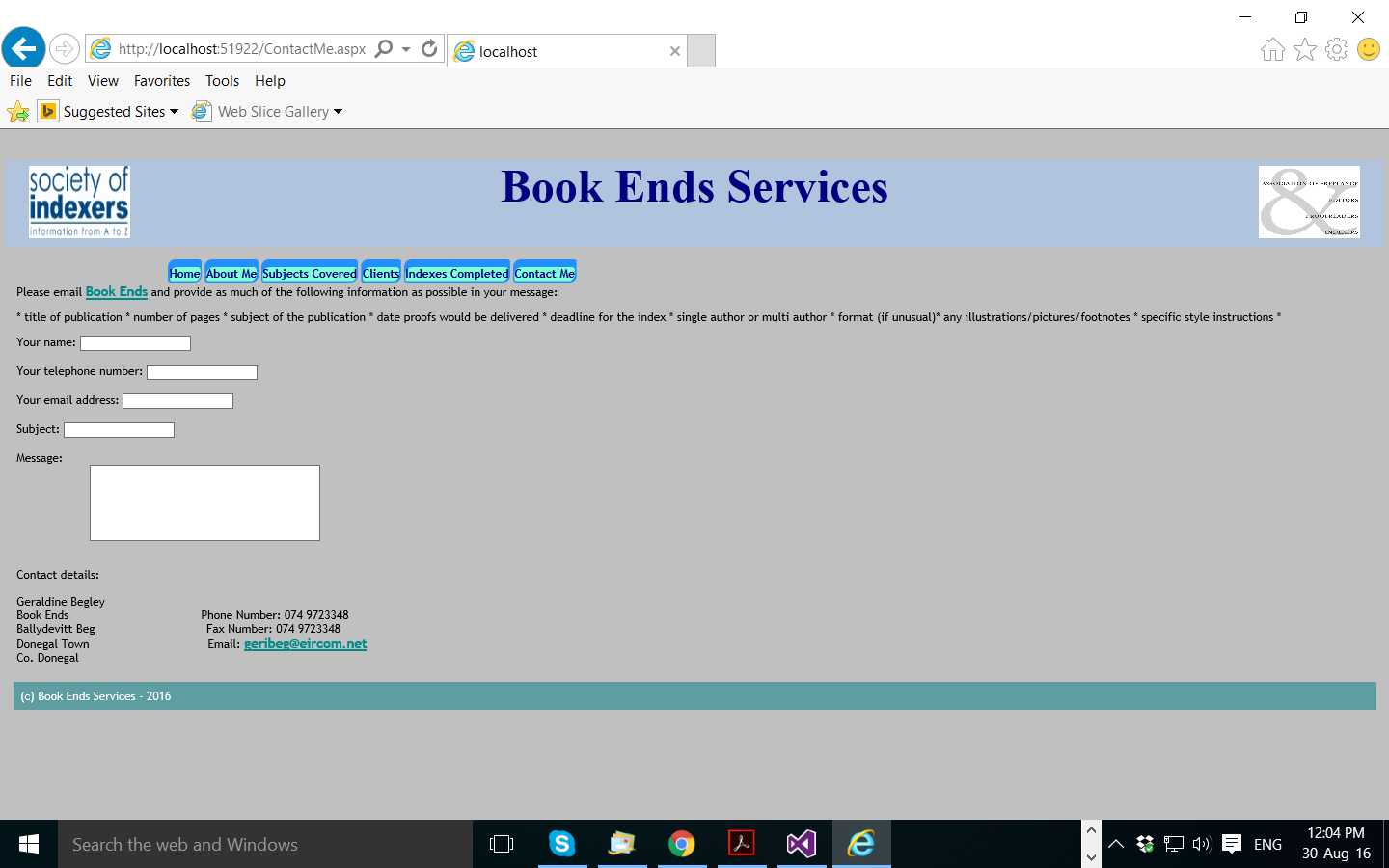
Clients page:



List of completed indexes:



Contact me page:



**2. DATABASE**

I created a database in SQL Server called BookEnds with three tables: BooksIndexed, Clients and ContactNames.

The database is a work-in-progress and data can be filled in when I get more information regarding the books I indexed and as I get more work. More tables can be added for example to keep a record of fees received and number of hours worked.

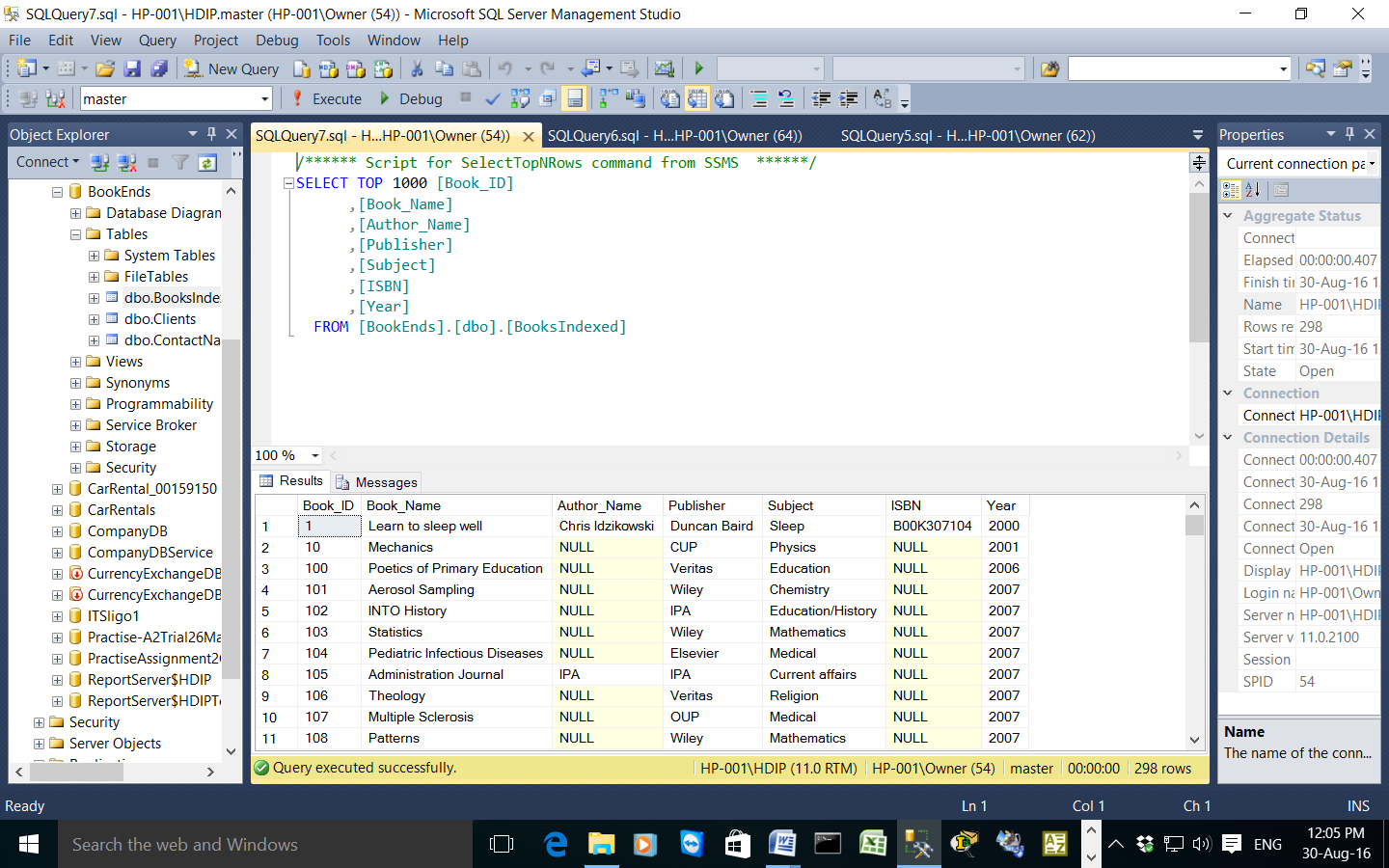
The columns in the BooksIndexed table are: Book\_Name, Author\_Name, Publisher, Subject, ISBN and Year.

The Clients table has the columns: ContactFirstName, ContactLastName, JobTitle, PublisherName, Address1, Address2, City, Country, Post Code, Phone No, Website and Email.

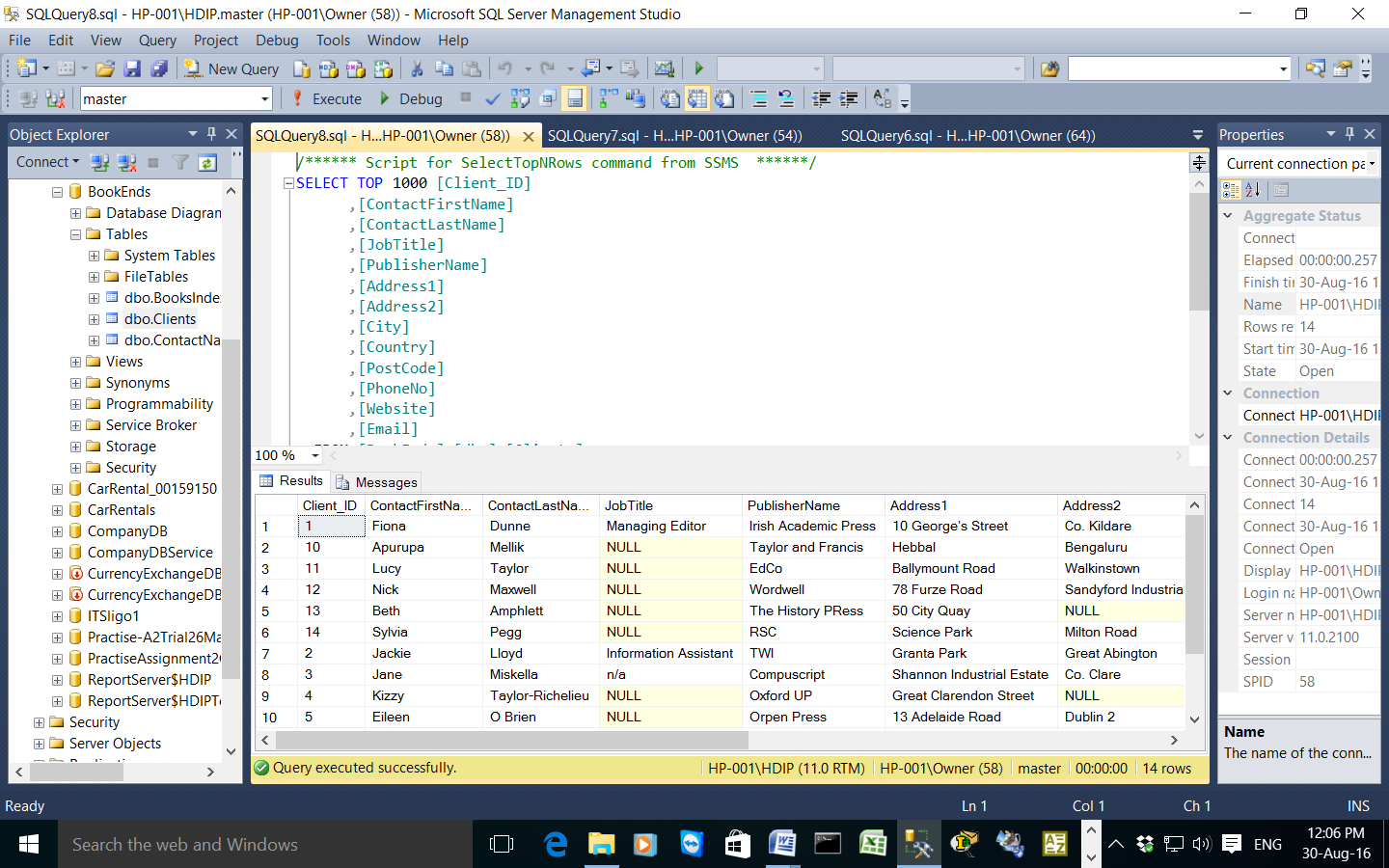
The Contacts table has the columns: First\_Name, Last\_Name, Email, CompanyName and Description (for example job title)

Screen shots of the tables are taken to show the tables.

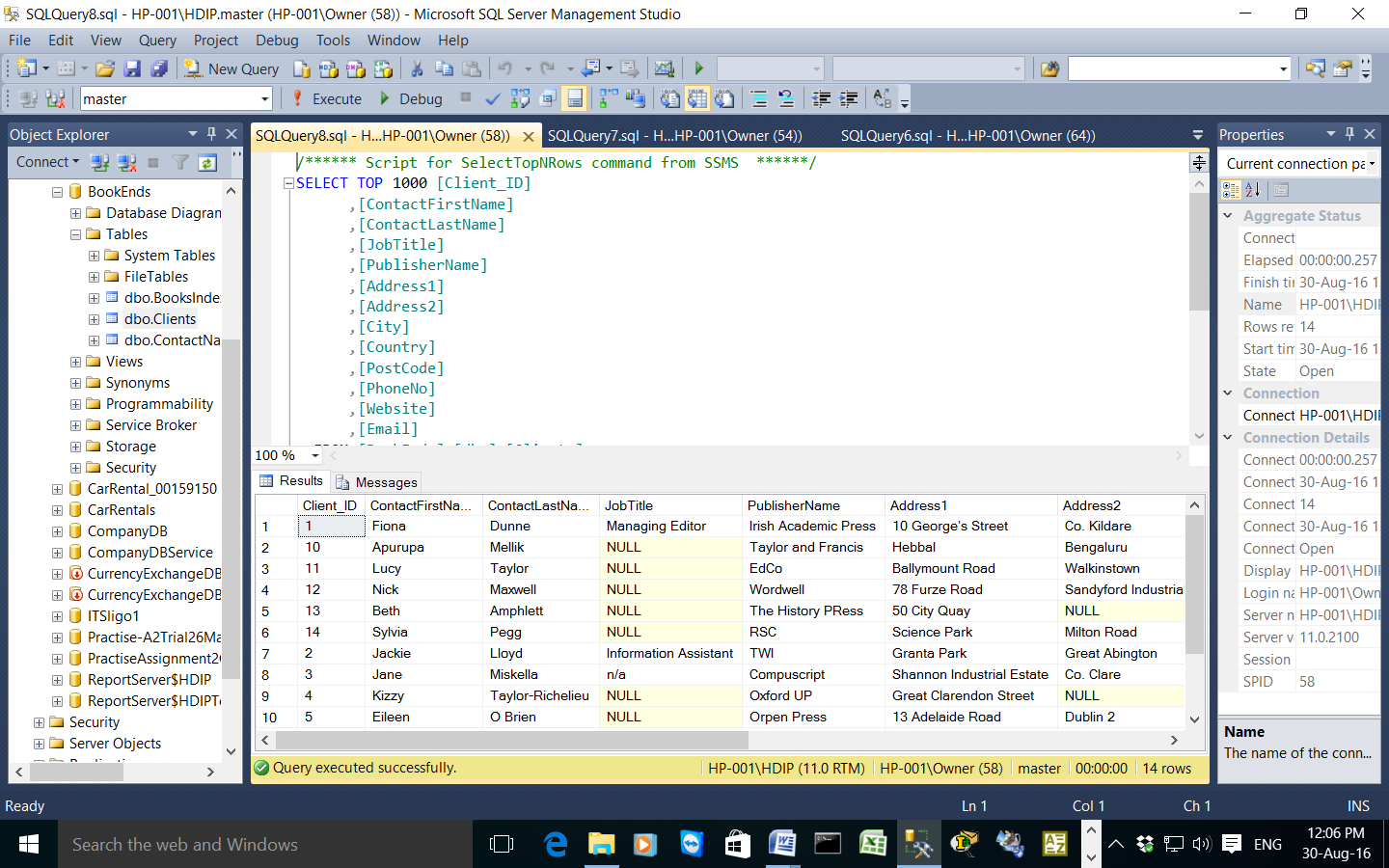
BooksIndexed table:



Clients table:



Contact Names table:



Some of the SQL code used is included below:

To make the Books\_Indexed table

make table Books\_Indexed

To insert data into the BooksIndexed table:

USE BookEnds;

INSERT INTO BooksIndexed(Book\_ID, Book\_Name, Author\_Name, Publisher, Subject, ISBN, Year)

VALUES(298, 'Irish Fine Art', 'Fenlon', 'Irish Academic Press', 'Art',NULL, 2016);

To create the contactNames table:

CREATE TABLE contactNames

(Name\_ID int identity not null,

First\_Name char(30) not null,

Last\_Name char(30) not null,

Email varchar(30) not null,

CompanyName char(30) not null,

Description varchar(80) null);

To create the Clients table:

create table Clients

(

Client\_ID nchar(10),

ContactFirstName nchar(50),

ContactLastName nchar(50),

JobTitle nchar(30),

PublisherName nchar(30),

Address1 nchar(20),

Address2 nchar(20),

City nchar(20),

Country nchar(20),

PostCode nchar(10),

PhoneNo nchar(12),

Website nchar(30),

Email nchar(30)

);

To create the BooksIndexed table:

create table BooksIndexed

(

Book\_ID nchar(10),

Book\_Name nchar(50),

Author\_Name nchar(50),

Publisher nchar(30),

Subject nchar(30),

ISBN nchar(20),

Year date,

);

I also created 3 stored procedures to search for the books indexed in 2016, IT books indexed and publisher details. The stored procedures are below:

USE BookEnds

GO

CREATE PROCEDURE spGetBooksIndexedIn2016

AS

SELECT Book\_Name, Author\_Name, Publisher, Year

FROM

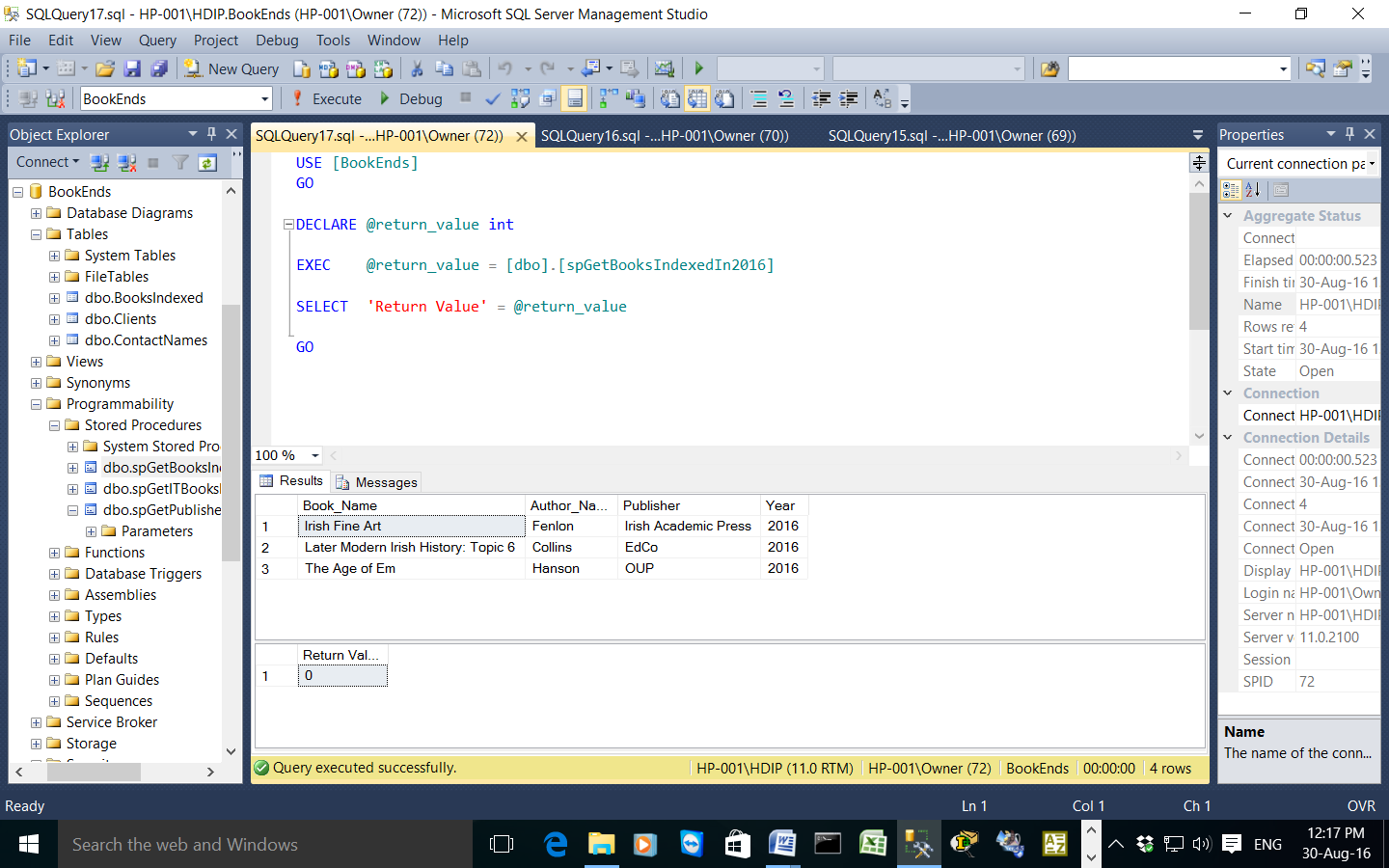
BooksIndexed

WHERE Year = 2016

ORDER BY

Book\_Name;

EXEC spGetBooksIndexedIn2016



USE BookEnds

GO

CREATE PROCEDURE spGetITBooksIndexed

AS

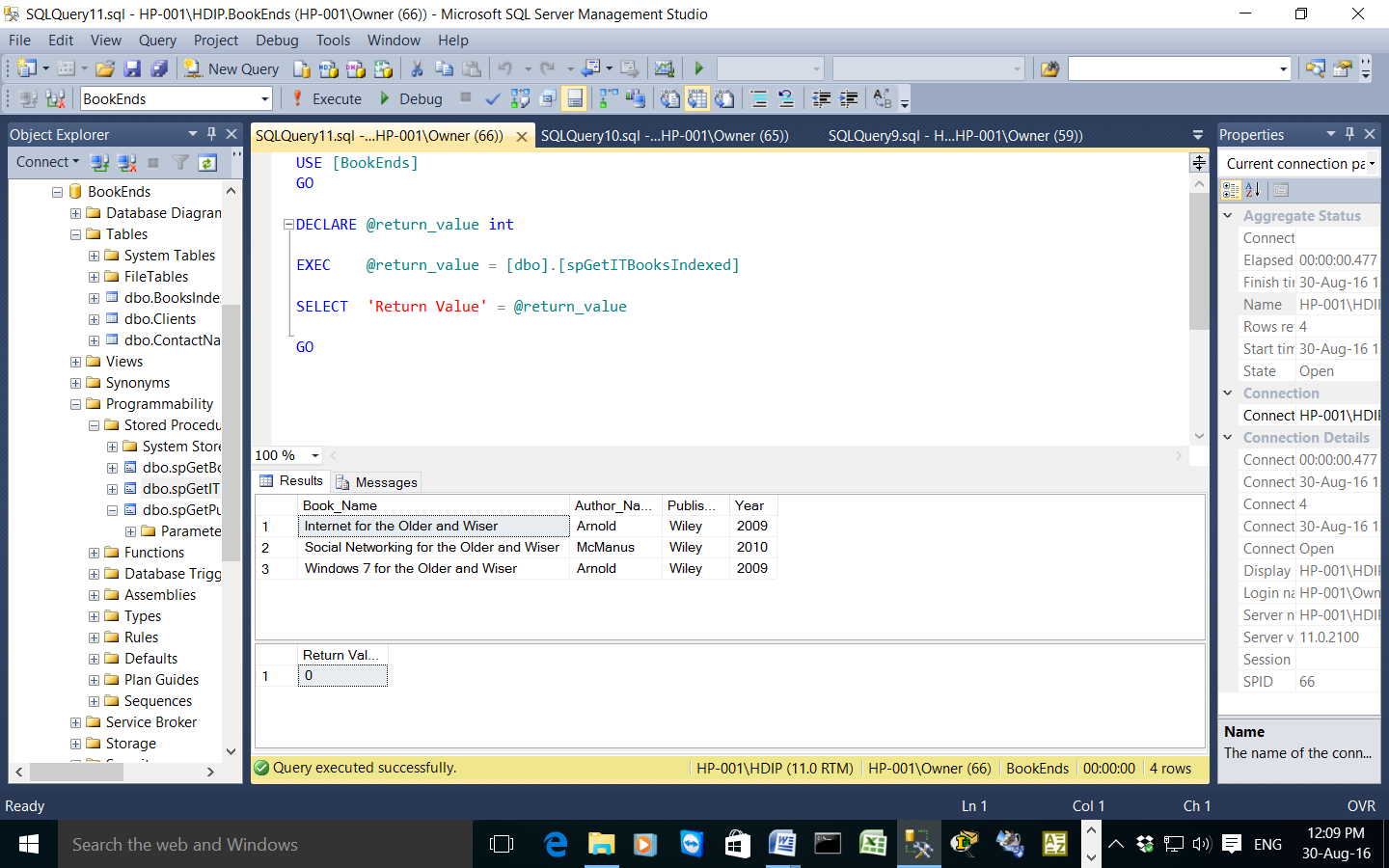
SELECT Book\_Name, Author\_Name, Publisher, Year

FROM BooksIndexed

WHERE Subject = 'IT'

ORDER BY Book\_Name;

EXEC spGetITBooksIndexed



USE BookEnds

GO

CREATE PROCEDURE spGetPublisherDetails

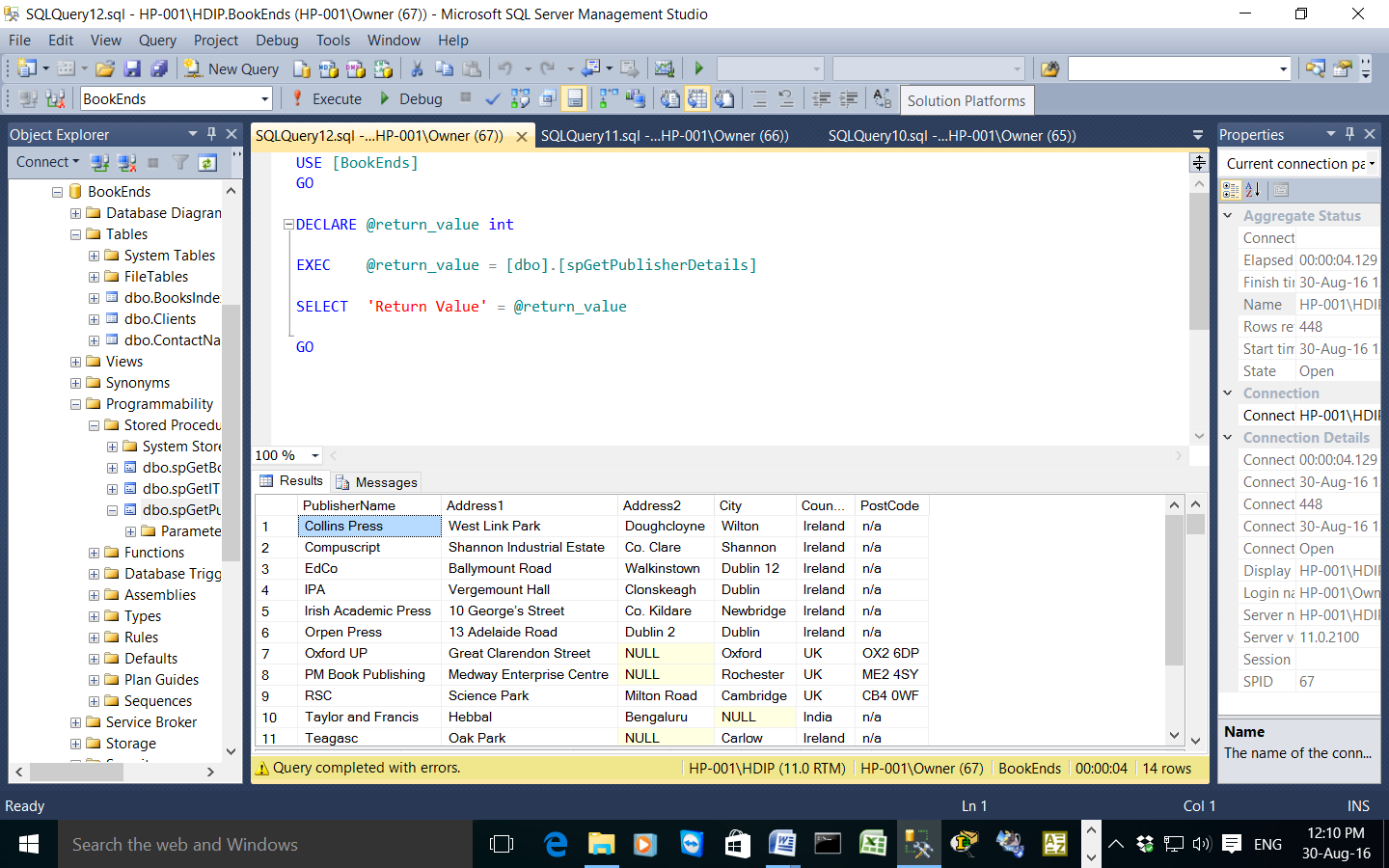
AS

SELECT PublisherName, Address1, Address2, City, Country, PostCode

FROM Clients

ORDER BY PublisherName;

EXEC spGetPublisherDetails



**3. WORD VBA MACROS**

Some of the macros I have used in the project are from recording my own or found on the internet from various sources and adapted for my own use: [www.archivepub.co.uk](http://www.archivepub.co.uk), [www.gmayor.com](http://www.gmayor.com), Microsoft, Youtube and Google Books.

1. **To toggle track changes on and off**

When editing I have to turn track changes on and it can be useful to have a quick macro to turn it on and off.

Sub ToggleTrackChange()

'

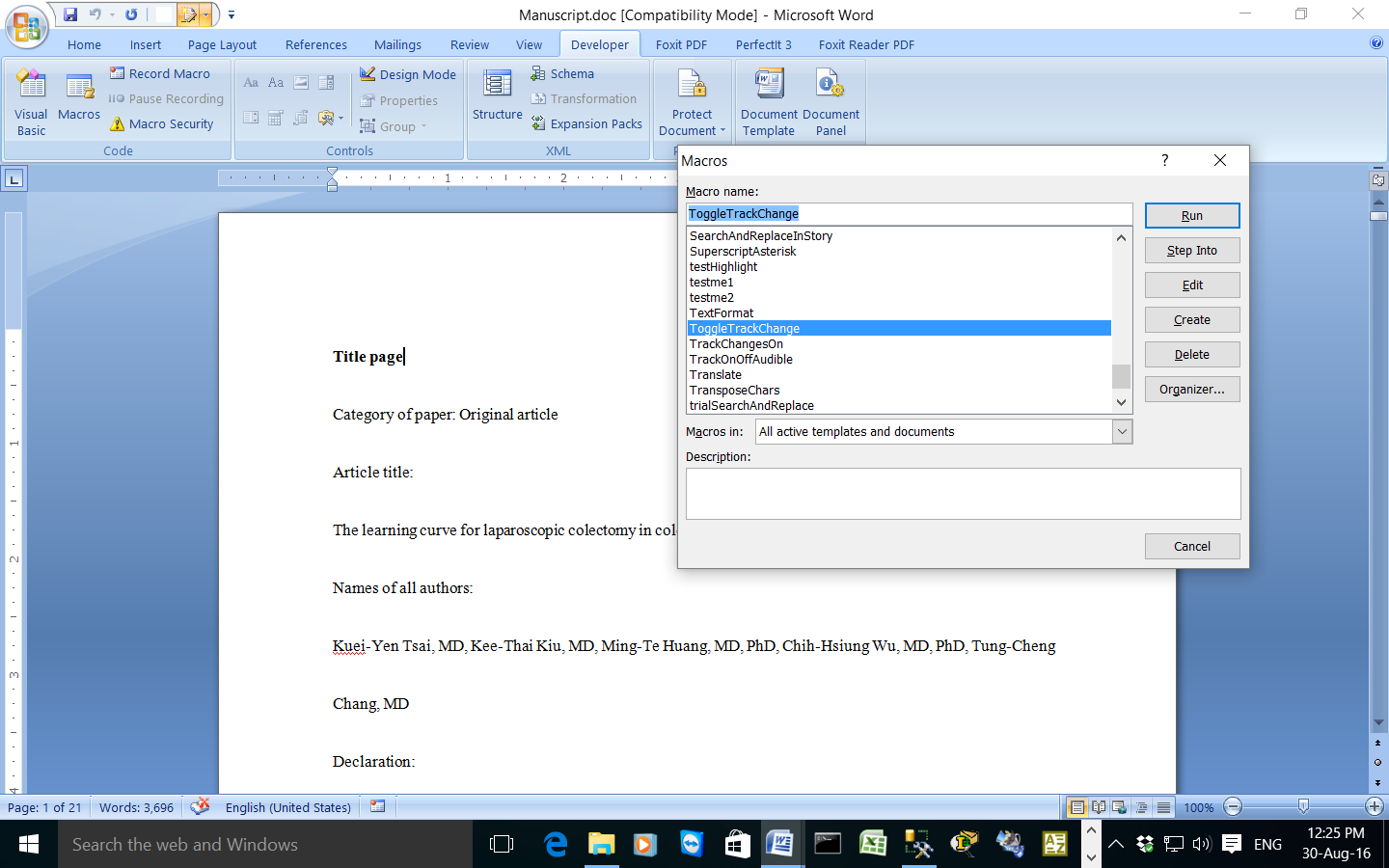
' ToggleTrackChange Macro

'

'

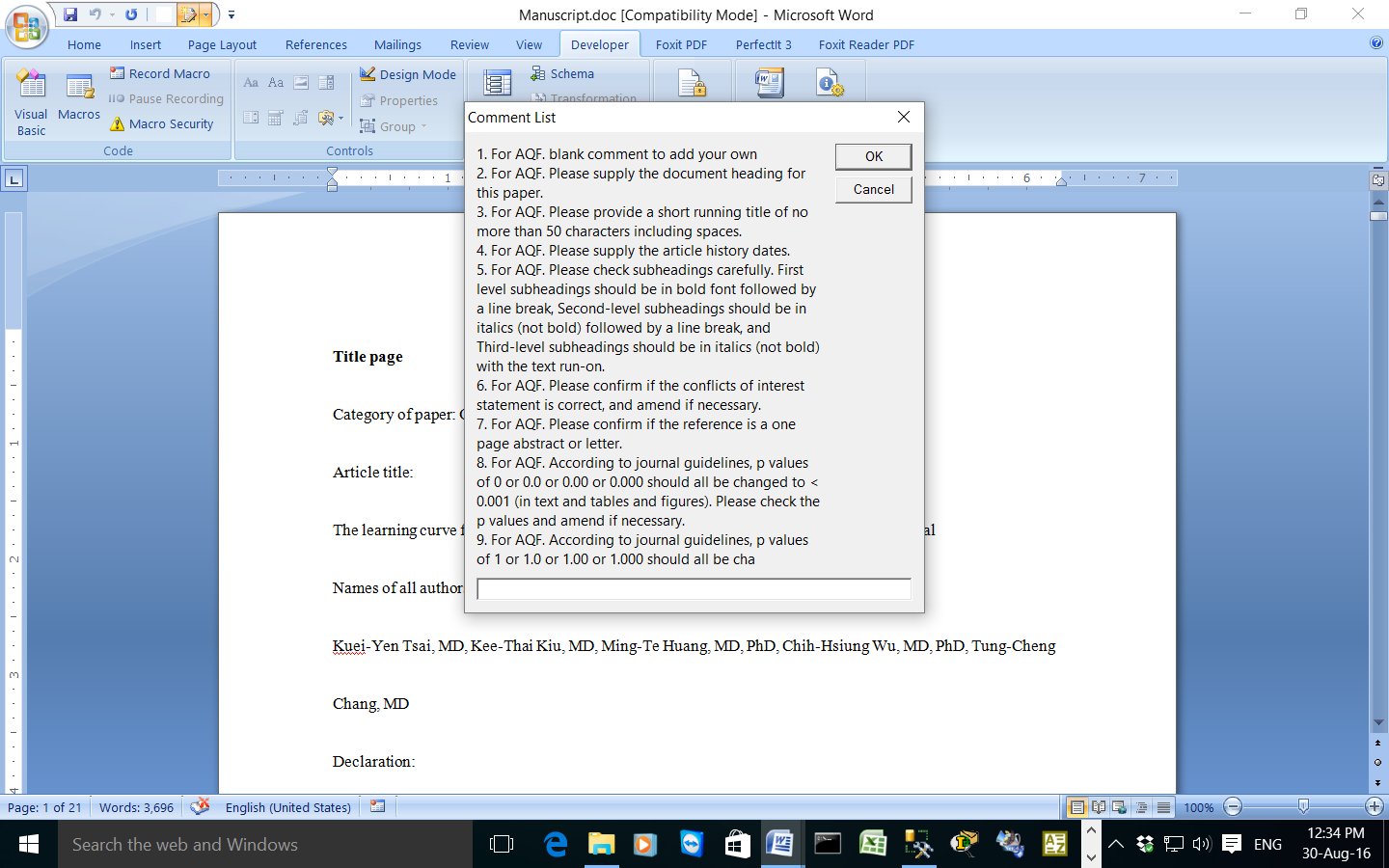
ActiveDocument.TrackRevisions = Not ActiveDocument.TrackRevisions

End Sub



1. **To choose comments from a menu**

I can use this macro to insert comments into articles. Usually I have these open in another file and copy and paste them in or I write them out. This should save me time having the comments in a list. It can also be added to for future use. This macro is adapted from macros on the archivepub and gmayor websites and contains separate macros that I recorded for inserting the individual text and comments.



Sub ProjectCommentMenu()

'

' ProjectCommentMenu Macro

'

'

Dim sPrompt As String

Dim sUserResp As String

Dim iUR As Integer

sPrompt = "1. For AQF. blank comment to add your own " & vbCrLf

sPrompt = sPrompt & "2. For AQF. Please supply the document heading for this paper." & vbCr

sPrompt = sPrompt & "3. For AQF. Please provide a short running title of no more than 50 characters including spaces." & vbCr

sPrompt = sPrompt & "4. For AQF. Please supply the article history dates." & vbCr

sPrompt = sPrompt & "5. For AQF. Please check subheadings carefully. First level subheadings should be in bold font followed by a line break, Second-level subheadings should be in italics (not bold) followed by a line break, and Third-level subheadings should be in italics (not bold) with the text run-on." & vbCr

sPrompt = sPrompt & "6. For AQF. Please confirm if the conflicts of interest statement is correct, and amend if necessary." & vbCr

sPrompt = sPrompt & "7. For AQF. Please confirm if the reference is a one page abstract or letter." & vbCr

sPrompt = sPrompt & "8. For AQF. According to journal guidelines, p values of 0 or 0.0 or 0.00 or 0.000 should all be changed to < 0.001 (in text and tables and figures). Please check the p values and amend if necessary." & vbCr

sPrompt = sPrompt & "9. For AQF. According to journal guidelines, p values of 1 or 1.0 or 1.00 or 1.000 should all be changed to > 0.99 (in text and tables and figures). Please check the p values and amend if necessary" & vbCr

iUR = 0

While iUR < 1 Or iUR > 9

sUserResp = InputBox(sPrompt, "Comment List")

If StrPtr(sUserResp) = 0 Then Exit Sub

iUR = Val(sUserResp)

Wend

Select Case iUR

Case 1

'inserts blank comment

Selection.Comments.Add Range:=Selection.Range

Case 2

'supply document heading comment

' Add comment to AQF to supply the article category

'

'ActiveDocument.TrackRevisions = Not ActiveDocument.TrackRevisions

'Selection.TypeParagraph

'Selection.MoveUp Unit:=wdLine, Count:=1

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. Please supply the article category heading."

Selection.EscapeKey

Case 3

'running title comment

' AddRunningTitle Macro

' add running title to document

'

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:="For AQF. Please provide a short running title of no more than 50 characters including spaces."

Selection.TypeBackspace

Selection.EscapeKey

Case 4

'add article history dates and comment

' HistoryDates Macro

'

'

Selection.TypeText Text:="Received, revised, accepted"

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. Please supply the article history dates."

Selection.EscapeKey

Case 5

'subheadings comment

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. Please check subheadings carefully. First level subheadings should be in bold font followed by a line break, Second-level subheadings should be in italics (not bold) followed by a line break, and Third-level subheadings should be in italics (not bold) with the text run-on."

Selection.EscapeKey

Case 6

'conflicts of interest

' ConflictsOfInterest Macro

' add conflicts of interest statement and comment

'

' Add conflicts of interest statement and comment

'

Selection.Font.Bold = wdToggle

Selection.TypeText Text:="Conflicts of interest"

Selection.TypeParagraph

Selection.Font.Bold = wdToggle

Selection.TypeText Text:= \_

"All authors have no conflicts of interest to declare."

Selection.Comments.Add Range:=Selection.Range

' ActiveWindow.ActivePane.HorizontalPercentScrolled = 18

Selection.TypeText Text:= \_

"For AQF. Please confirm if the conflicts of interest stateme"

Selection.TypeText Text:="nt is correct, and amend if necessary."

Selection.EscapeKey

' ActiveWindow.ActivePane.HorizontalPercentScrolled = 1

Case 7

'one page journal reference

'Selection.PasteAndFormat (wdPasteDefault)

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. Please confirm if the reference is a one page abstract or letter. "

Selection.EscapeKey

Case 8

'p = 0.000

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. According to journal guidelines, p values of 0 or 0.0 or 0.00 or 0.000 should all be changed to < 0.001 (in text and tables and figures). Please check the p values and amend if necessary."

Selection.EscapeKey

Case 9

'p=1

Selection.Comments.Add Range:=Selection.Range

Selection.TypeText Text:= \_

"For AQF. According to journal guidelines, p values of 1 or 1.0 or 1.00 or 1.000 should all be changed to < 0.99 (in text and tables and figures). Please check the p values and amend if necessary."

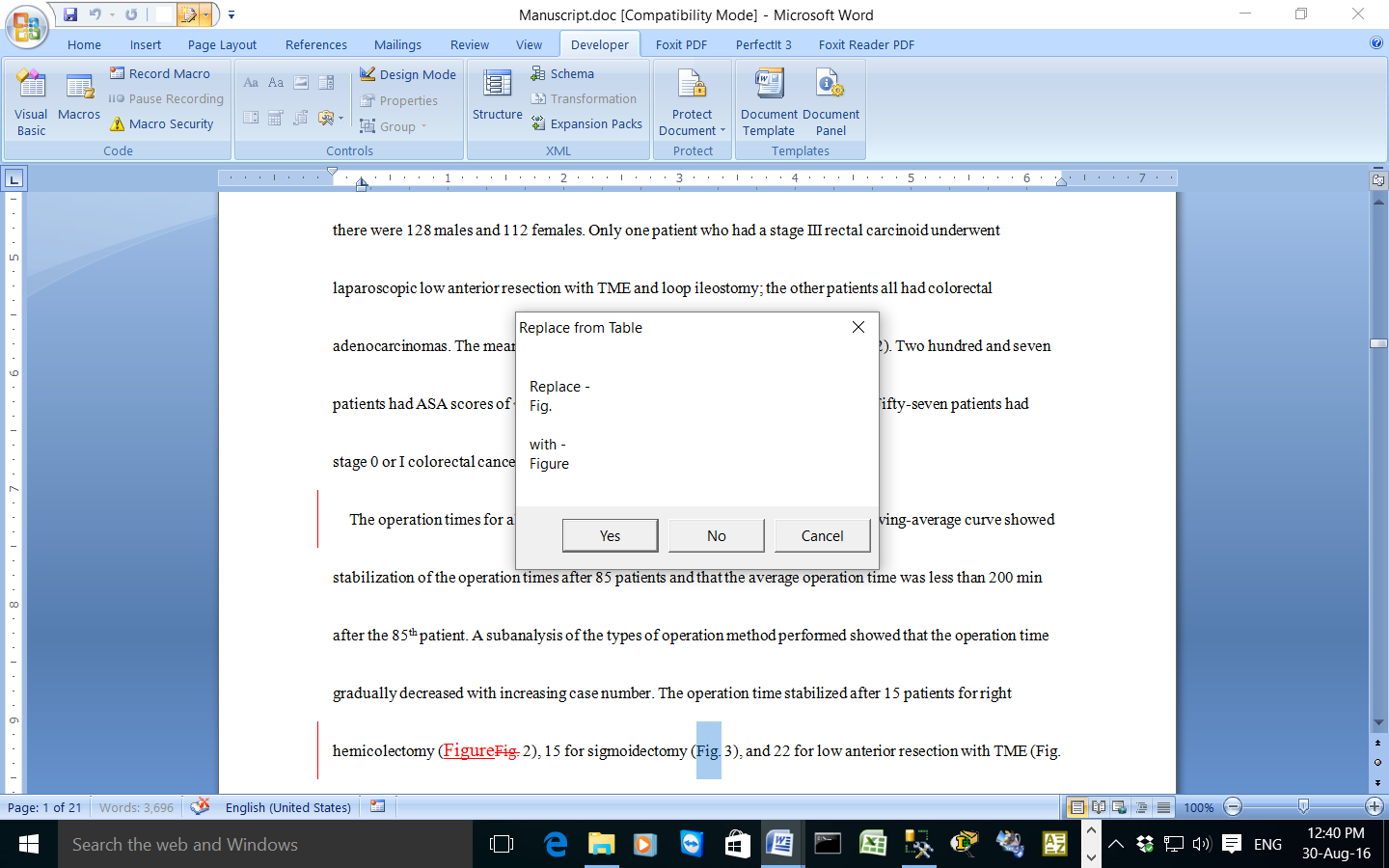
Selection.EscapeKey

End Select

End Sub

1. **Macro to search and replace text from a table.**

This macro is adapted from one on the gmayor.com website. The user is asked if they wish to replace a word/phrase/number with something else. This is useful because some text is correct in some situations. A drawback of this macro is changing italics to roman for example, the font does not show up in the menu box.

****

The macro begins with a message box to ask the user if they would like to run the search and replace macro. It ends with a message box telling the user the macro has finished working as the macro is slow and has a lot to check. It also allows the user to cancel out of the macro.

Sub ProjectSearchandReplaceFromTableList()

'add a message box Yes or No

Dim Answer As VbMsgBoxResult

Answer = MsgBox("Do you want to run the search and replace macro?", vbYesNo, "Run Macro")

If Answer = vbYes Then

Dim oChanges As Document, oDoc As Document

Dim oTable As Table

Dim oRng As Range

Dim oStory As Range

Dim rFindText As Range, rReplacement As Range

Dim i As Long

Dim sFname As String

Dim sAsk As String

sFname = "C:\Users\Owner\Documents\changes.docx"

Set oDoc = ActiveDocument

Set oChanges = Documents.Open(FileName:=sFname, Visible:=False)

Set oTable = oChanges.Tables(1)

For i = 1 To oTable.Rows.Count

For Each oStory In oDoc.StoryRanges

Set oRng = oStory

Set rFindText = oTable.Cell(i, 1).Range

rFindText.End = rFindText.End - 1

Set rReplacement = oTable.Cell(i, 2).Range

rReplacement.End = rReplacement.End - 1

With oRng.Find

.ClearFormatting

.Replacement.ClearFormatting

Do While .Execute(findText:=rFindText, \_

MatchWholeWord:=True, \_

MatchWildcards:=False, \_

Forward:=True, \_

Wrap:=wdFindStop) = True

oRng.Select

sAsk = MsgBox("Replace - " & vbCr & oRng & vbCr + vbCr & \_

"with - " & vbCr & rReplacement, vbYesNoCancel, \_

"Replace from Table")

If sAsk = vbCancel Then Exit Sub

If sAsk = vbYes Then

oRng.FormattedText = rReplacement.FormattedText

End If

oRng.Collapse wdCollapseEnd

Loop

End With

If oStory.StoryType <> wdMainTextStory Then

While Not (oStory.NextStoryRange Is Nothing)

Set oStory = oStory.NextStoryRange

Set oRng = oStory

Set rFindText = oTable.Cell(i, 1).Range

rFindText.End = rFindText.End - 1

Set rReplacement = oTable.Cell(i, 2).Range

rReplacement.End = rReplacement.End - 1

With oRng.Find

.ClearFormatting

.Replacement.ClearFormatting

Do While .Execute(findText:=rFindText, \_

MatchWholeWord:=True, \_

MatchWildcards:=False, \_

Forward:=True, \_

Wrap:=wdFindStop) = True

oRng.Select

sAsk = MsgBox("Replace - " & vbCr & oRng & vbCr + vbCr & \_

"with - " & vbCr & rReplacement, vbYesNo, \_

"Replace from Table")

If sAsk = vbCancel Then Exit Sub

If sAsk = vbYes Then

oRng.FormattedText = rReplacement.FormattedText

End If

oRng.Collapse wdCollapseEnd

Loop

End With

Wend

End If

Next oStory

Next i

If ActiveWindow.View.SplitSpecial <> wdPaneNone Then

ActiveWindow.Panes(2).Close

End If

If ActiveWindow.ActivePane.View.Type = wdNormalView Or ActiveWindow. \_

ActivePane.View.Type = wdOutlineView Then

ActiveWindow.ActivePane.View.Type = wdPrintView

End If

ActiveWindow.ActivePane.View.SeekView = wdSeekCurrentPageHeader

If ActiveWindow.View.SplitSpecial = wdPaneNone Then

ActiveWindow.ActivePane.View.Type = wdPrintView

Else

ActiveWindow.View.Type = wdPrintView

End If

ActiveWindow.ActivePane.View.SeekView = wdSeekMainDocument

If ActiveWindow.View.SplitSpecial = wdPaneNone Then

ActiveWindow.ActivePane.View.Type = wdPrintView

Else

ActiveWindow.View.Type = wdPrintView

End If

oChanges.Close wdDoNotSaveChanges

lbl\_Exit:

Set oStory = Nothing

Set oRng = Nothing

Set rFindText = Nothing

Set rReplacement = Nothing

MsgBox "The search and replace macro has finished running"

End If

Exit Sub

End Sub

**The table used in the search and replace:**

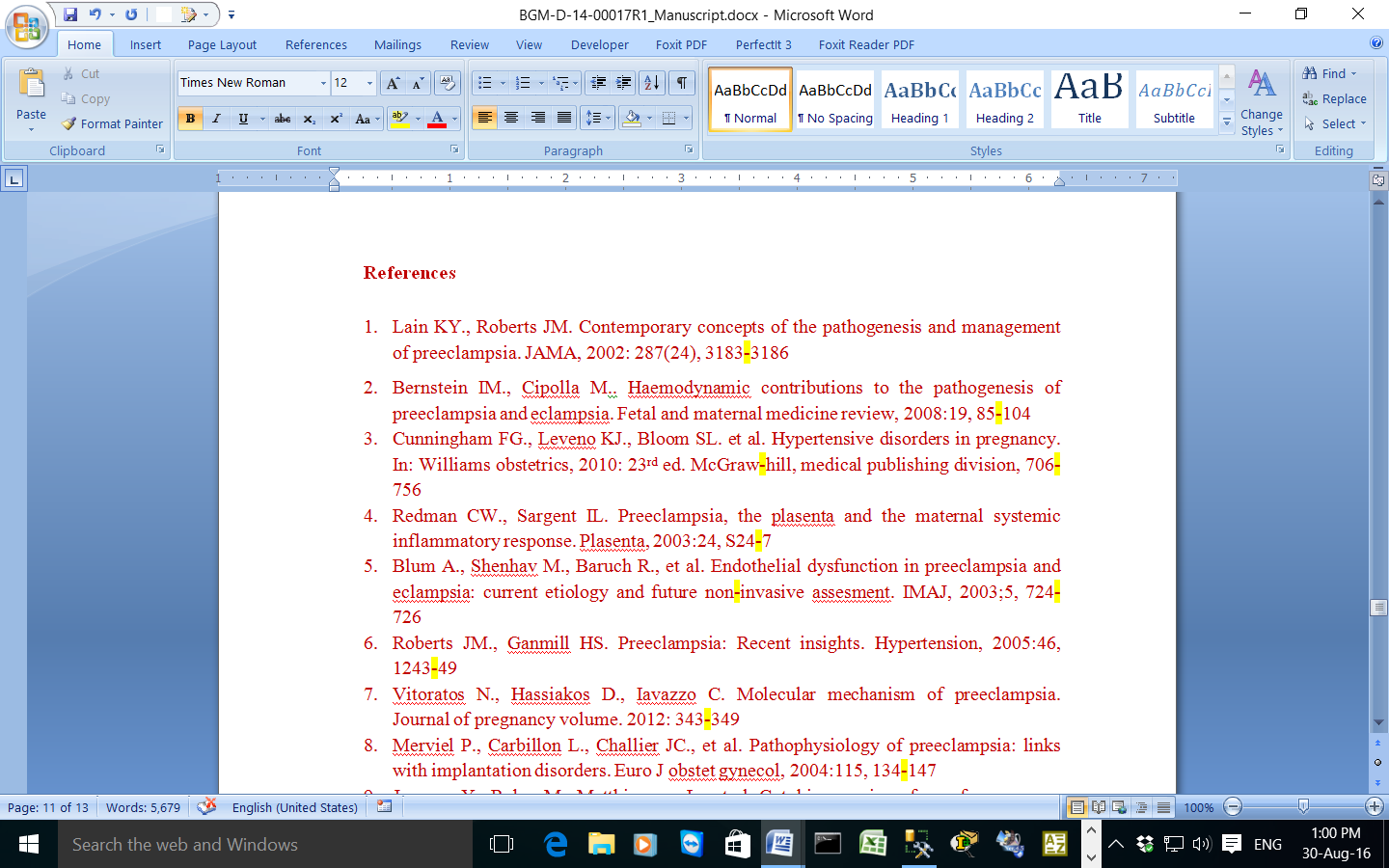
|  |  |
| --- | --- |
| © |  |
| ® |  |
| ™ |  |
| another 2 | two more |
| as evidence by | as demonstrated by |
| assessed by | assessed using the |
| Reference | References |
| bulla | bullae |
| by now | up to now |
| by the year | by |
| cm H2O | cmH2O |
| and colleagues | et al |
| colored by | shown by |
| compared to | compared with |
| computer aided | computer-aided |
| couldn’t | could not |
| 1 cycle | one cycle |
| day 1 | Day 1 |
| eg | e.g., |
| *et al* | et al |
| et al. | et al |
| eleven | 11 |
| twelve | 12 |
| thirteen | 13 |
| fourteen | 14 |
| fifteen | 15 |
| sixteen | 16 |
| seventeen | 17 |
| eighteen | 18 |
| nineteen | 19 |
| twenty | 20 |
| Fig. | Figure |
| first minute | 1st minute |
| first year | 1st year |
| 2-fold | twofold |
| \* | \* |
| Conclusions | Conclusion |
| Key words | Keywords |
| g/dl | g/dL |
| grade 3 tumor | Grade 3 tumor |
| Gram negative bacteria | Gram-negative bacteria |
| 4 groups | four groups |
| has estimated | estimates |
| health care | healthcare |
| health-care | healthcare |
| herein |  |
| gram | g |
| one hour | 1 hour |
| two hours | 2 hours |
| However | However, |
| human beings | humans |
| human subjects | patients |
| ie | i.e., |
| in contrast | by contrast |
| information about | information regarding |
| it showed that | it was shown that |
| it’s | It is |
| de novo | *de novo* |
| post hoc | *post hoc* |
| a priori | *a priori* |
| in situ | *in situ* |
| in utero | *in utero* |
| in vivo | *in vivo* |
| en bloc | *en bloc* |
| in vitro | *in vitro* |
| less than | < |
| more than | > |
| like | such as |
| mg/dl | mg/dL |
| millijoules | kilocalories |
| μl | μL |
| mm Hg | mmHg |
| one month | 1 month |
| two months | 2 months |
| three months | 3 months |
| n= | *n* = |
| needle no. 1 | needle No. 1 |
| New Zealand White Rabbits | New Zealand white rabbits |
| percent | % |
| on the other hand | However, |
| one thousand | 1000 |
| p< | *p* < |
| patient 1 | Patient 1 |
| % | % |
| relationship to | relationship with |
| representation is given in | represented by |
| resistant mechanisms | resistance mechanisms |
| scrutinize | examine |
| sec | seconds |
| since | because |
| structure based | structure-based |
| subject | individual |
| ten | 10 |
| 2-sided test | two-sided test |
| U test | U-test |
| the elderly | elderly people |
| This case report presented | This case report presents |
| type 1 error | Type 1 error |
| Using | The use of |
| one hundred | 100 |
| Vitamin | vitamin |
| v.s. | vs. |
| was seldom reported | has seldom been reported |
| one week | 1 week |
| two weeks | 2 weeks |
| western countries | Western countries |
| while | although |
| while the | although |
| WHO | the WHO |
| with medication of | with treatment of |
| X-ray | x-ray |
| one year | 1 year |
| **Corresponding author:** | \* Corresponding author. |
| Sec. | Section |
| Rd. | Road |
| Email: | *E-mail address*: |

This table can be amended and added to when I come across phrases and words to look out for.

1. **Hyphen replace menu**

I wanted a macro to help make checking if hyphens needed to be changed into other formats (endash/minus/emdash), deleted or left as they were. The search and replace macro developed in the [www.archivepub.co.uk](http://www.archivepub.co.uk) made replacing numbers with endashes difficult as Word would insert the en-dash in the wrong position, creating errors in the page ranges for example. I liked the menu style of the search and replace in the previous macro but I couldn’t develop it the way I would have liked. In the end I come up with a solution which suits me. First of all I highlight the hyphens to make them stand out on the screen, then each time I come across a hyphen I run the macro. Ideally I would like the macro to alert me when it finds a hyphen. I developed the macro menu from combining the separate macros I recorded when doing search and replace and other macros mainly from [www.archivepub.co.uk](http://www.archivepub.co.uk) and [www.gmayor.com](http://www.gmayor.com).

First of all I highlight the hyphens:



Sub HighlightHyphen()

'

'

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

Selection.Find.Replacement.Highlight = True

With Selection.Find

.Text = "-"

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

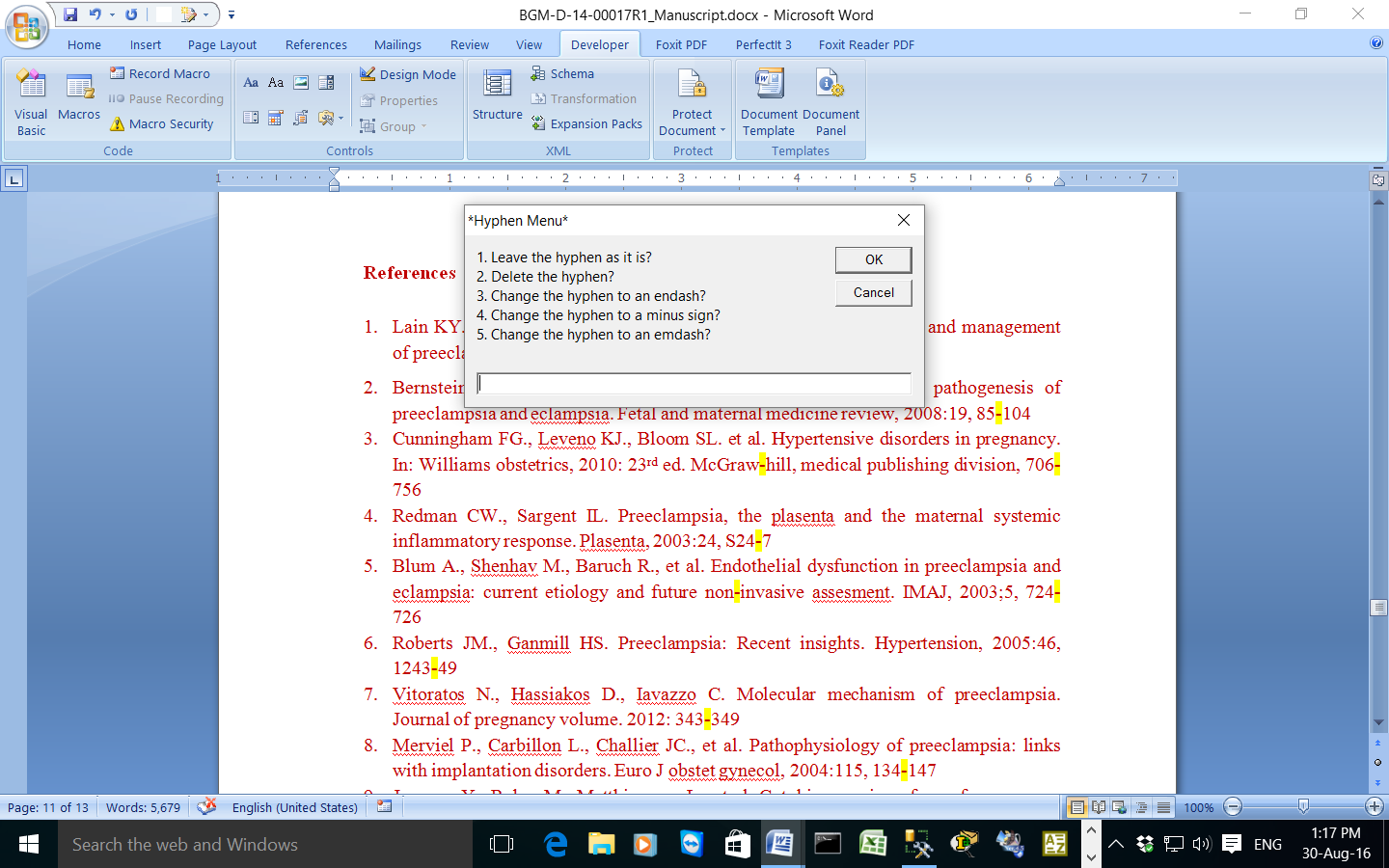
Selection.Find.Execute Replace:=wdReplaceAll

Options.DefaultHighlightColorIndex = wdYellow

Selection.Range.HighlightColorIndex = wdYellow

End Sub

Then I run the Hyphen menu macro and scan the page for hyphens which need changing or deleting. The user can select 1 to exit the macro so I did not need to add a statement for Cancelling out of the macro.



Sub ProjectHyphenMenu()

'

'

'

Dim sPrompt As String

Dim sUserResp As String

Dim iUR As Integer

sPrompt = "1. Leave the hyphen as it is?" & vbCrLf

sPrompt = sPrompt & "2. Delete the hyphen?" & vbCrLf

sPrompt = sPrompt & "3. Change the hyphen to an endash?" & vbCrLf

sPrompt = sPrompt & "4. Change the hyphen to a minus sign?" & vbCrLf

sPrompt = sPrompt & "5. Change the hyphen to an emdash?" & vbCrLf

iUR = 0

While iUR < 1 Or iUR > 5

sUserResp = InputBox(sPrompt, "\*Hyphen Menu\*")

iUR = Val(sUserResp)

Wend

Select Case iUR

Case 1

'exits the routine and doesn't change anything

Exit Sub

Case 2

'deletes the hyphen

'

' DeleteHyphen2 Macro

'

'

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "-"

.Replacement.Text = ""

'.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

With Selection

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseStart

Else

.Collapse Direction:=wdCollapseEnd

End If

.Find.Execute Replace:=wdReplaceOne

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseEnd

Else

.Collapse Direction:=wdCollapseStart

End If

.Find.Execute

End With

Case 3

'change the hyphen to an endash

' HyphenToEndash Macro

'

'

'

' HyphenToEndash Macro

'

'

Selection.InsertSymbol CharacterNumber:=8211, Unicode:=True, Bias:=0

Selection.Delete Unit:=wdCharacter, Count:=1

'from recorded macro

'With Selection

' If .Find.Forward = True Then

' .Collapse Direction:=wdCollapseStart

' Else

' .Collapse Direction:=wdCollapseEnd

'End If

'.Find.Execute Replace:=wdReplaceOne

'If .Find.Forward = True Then

' .Collapse Direction:=wdCollapseEnd

'Else

' .Collapse Direction:=wdCollapseStart

'End If

'End With

Case 4

'change the hyphen to a minus sign

'

' HyphenToMinus Macro

'

'

Selection.InsertSymbol CharacterNumber:=8722, Unicode:= \_

True

Selection.Delete Unit:=wdCharacter, Count:=1

Selection.Find.Execute

Case 5

'change hyphen to an emdash

'

' replace hyphen with emdash

'

Selection.Find.ClearFormatting

Selection.Find.Replacement.ClearFormatting

With Selection.Find

.Text = "-"

.Replacement.Text = "—"

.Forward = True

.Wrap = wdFindContinue

.Format = False

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

End With

Selection.Find.Execute

With Selection

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseStart

Else

.Collapse Direction:=wdCollapseEnd

End If

.Find.Execute Replace:=wdReplaceOne

If .Find.Forward = True Then

.Collapse Direction:=wdCollapseEnd

Else

.Collapse Direction:=wdCollapseStart

End If

.Find.Execute

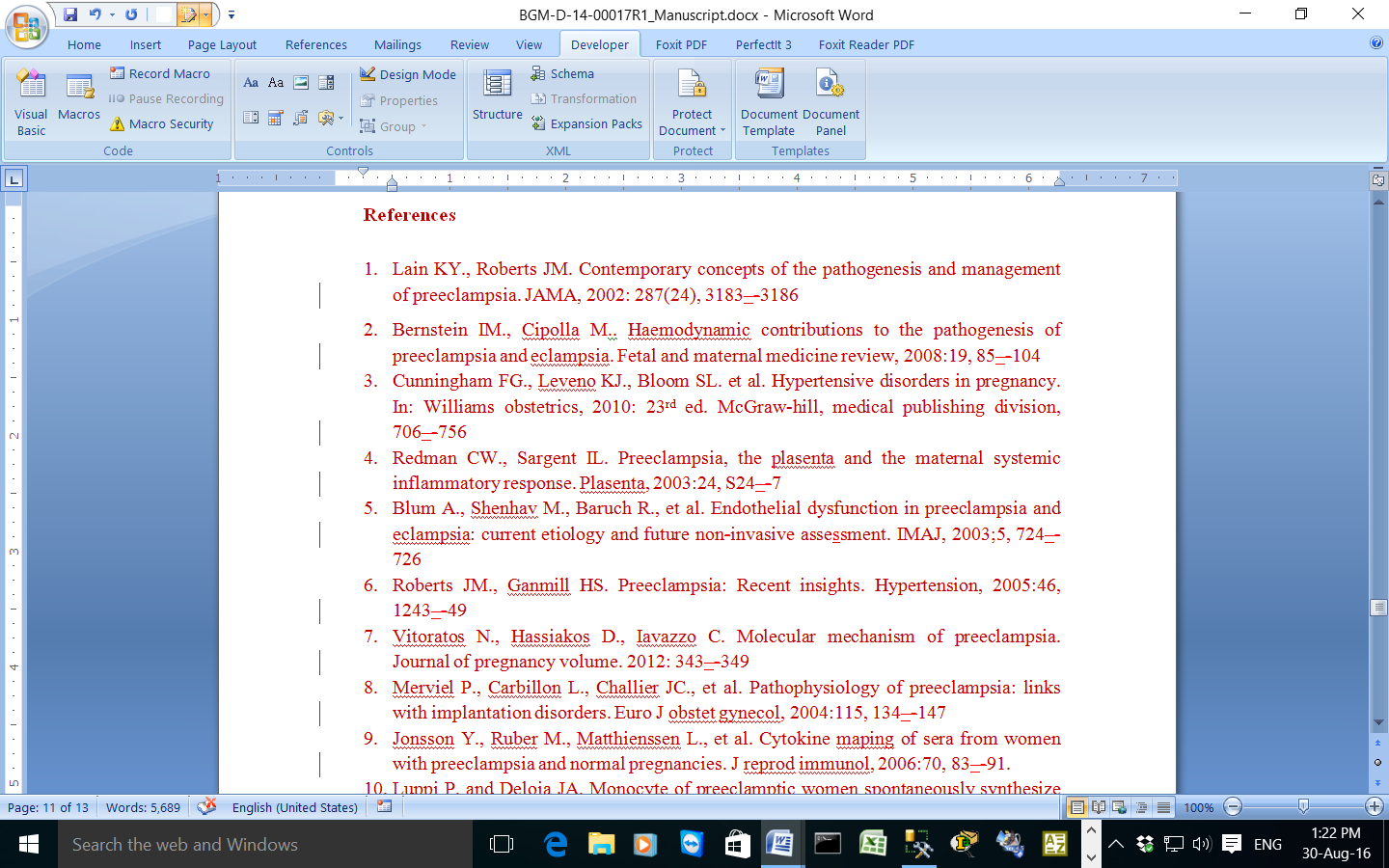
End With

End Select

End Sub

The user has to go to each hyphen as necessary to change them.

Finally there is another macro to unhighlight the text.



Sub UnhighlightText()

'

' UnhighlightText Macro

'

'

Selection.Find.ClearFormatting

Selection.Find.Highlight = True

Selection.Find.Replacement.ClearFormatting

Selection.Find.Replacement.Highlight = False

With Selection.Find

.Text = ""

.Replacement.Text = ""

.Forward = True

.Wrap = wdFindContinue

.Format = True

.MatchCase = False

.MatchWholeWord = False

.MatchWildcards = False

.MatchSoundsLike = False

.MatchAllWordForms = False

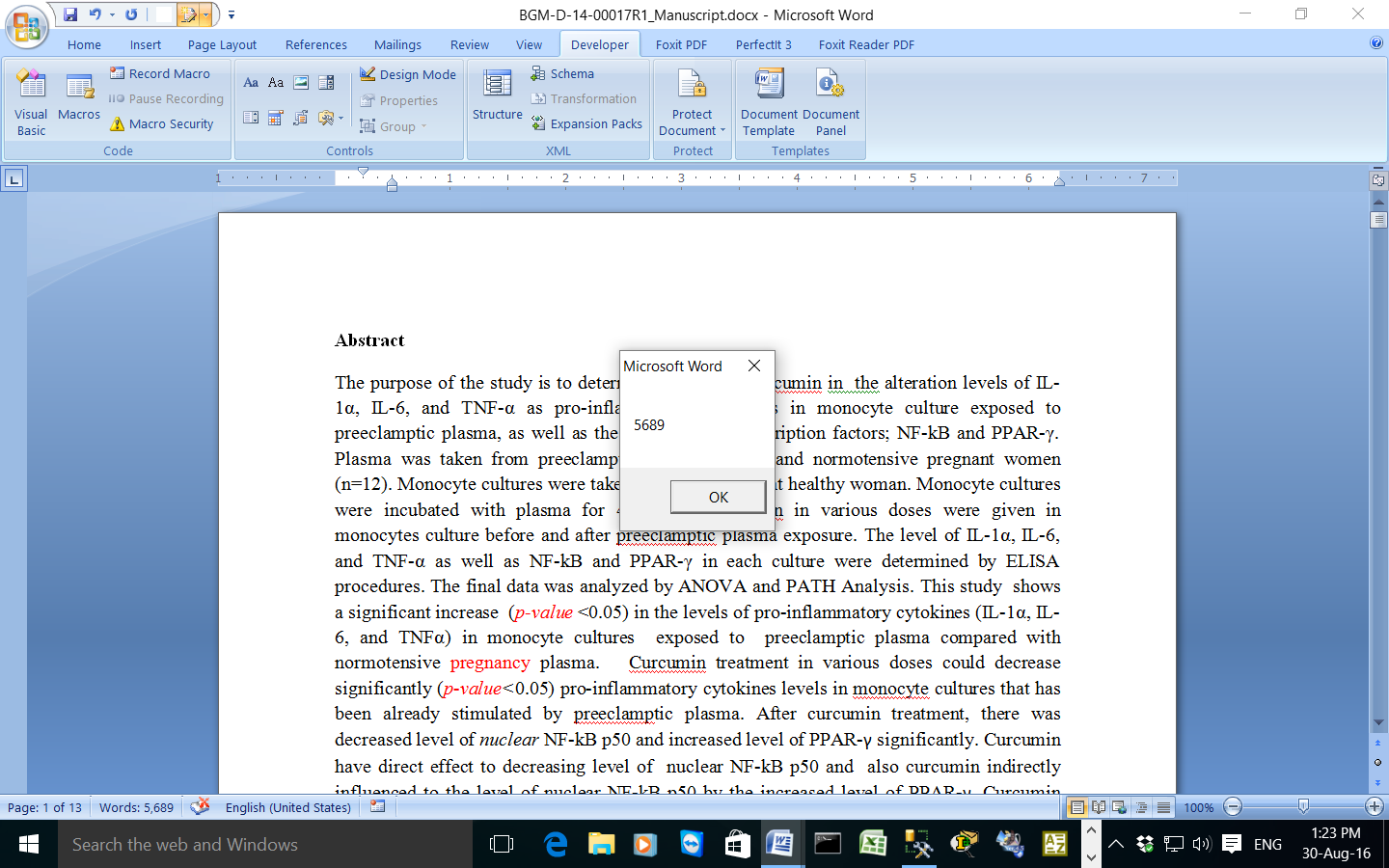
End With

Selection.Find.Execute Replace:=wdReplaceAll

End Sub

1. **Number of Words**

Although I look at the bottom row to see the number of words in a document I have a macro to return the number of words.



Sub NumberOfWords()

'

' NumberOfWords Macro

'

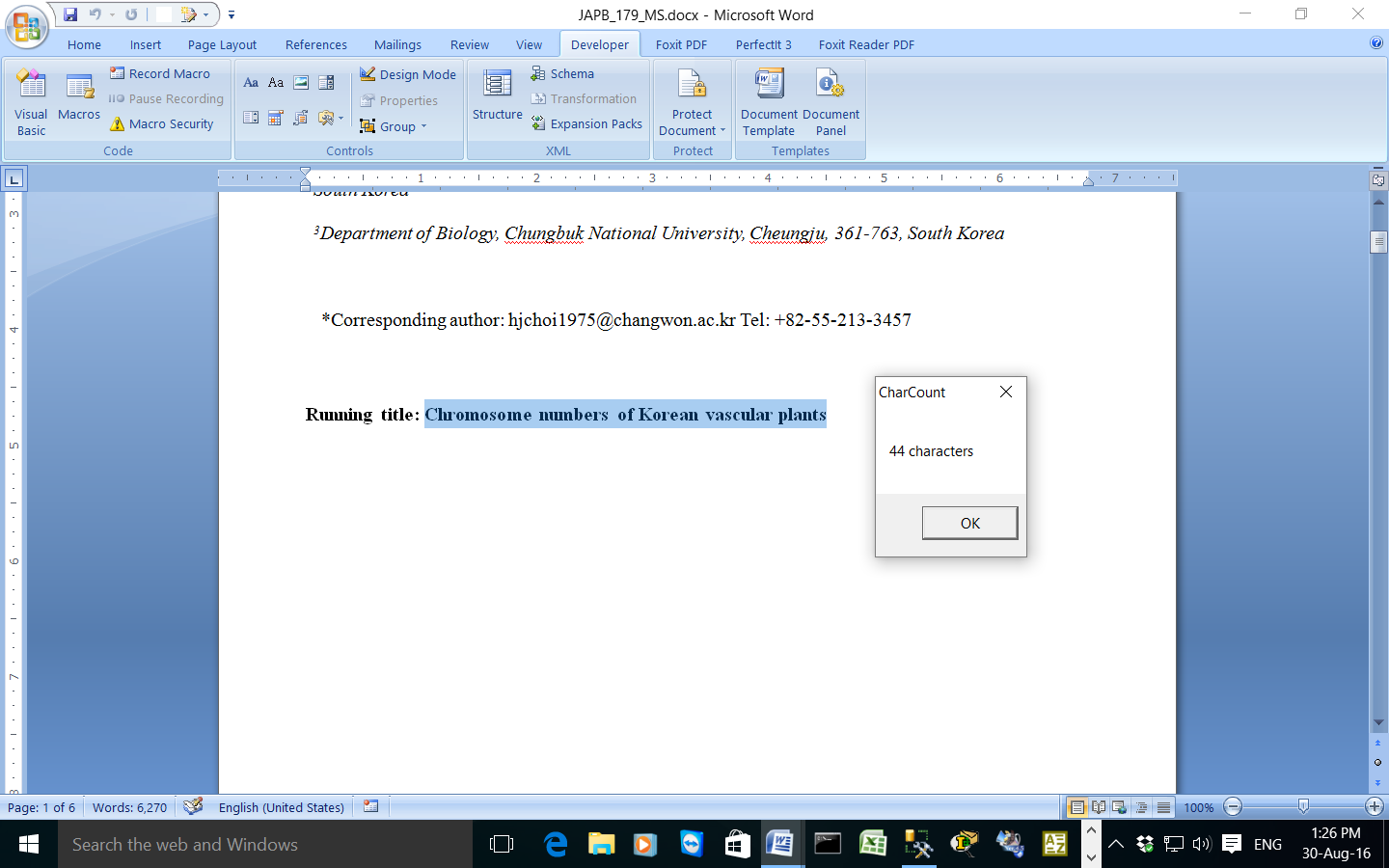
'

MsgBox ActiveDocument.Range.ComputeStatistics(wdStatisticWords)

End Sub

1. **Number of characters**

I have a macro to return the number of characters in a phrase, which would be useful for checking if for example the running title does not exceed its limit.



Sub CountNoCharacters()

'

' CountNoCharacters Macro

'

'

Dim Title As String

Dim CharCount As Integer

Dim Message As String

Title = "CharCount"

CharCount = Len(Selection)

Message = LTrim(Str(CharCount)) + " character"

If CharCount <> 1 Then Message = Message + "s"

MsgBox Message, vbOKOnly, Title

End Sub

I was able to add a keyboard short cut key for all the macros except for the word count macro. I usually look at the number of words in the bottom panel so that issue doesn’t really matter. The macros should help make me more efficient and less likely to miss text which should be edited. When developing the macros I tested them out on some articles received from Compuscript.