

**PROJECT REPORT ON**  
**ONLINE WEBSITE BUILDER**  
**AT**

**MASTER COMPUTECH PVT. LTD.**  
**DADAR**

**SUBMITTED BY**  
**SHAHIL SHAH**  
**VIKRAM SUTAR**  
**PRIYA PRABHU**  
**SHOME SHARMA**

**DEPARTMENT OF COMPUTER**  
**WATUMULL INSTITUTE OF ELECTRONICS**  
**ENGINEERING AND COMPUTER TECHNOLOGY**  
**WORLI, MUMBAI - 400 018**  
**UNIVERSITY OF MUMBAI**

**2006 - 2007**



PROJECT REPORT ON

## ONLINE WEBSITE BUILDER

SUBMITTED BY ,

- SHAHIL SHAH
- VIKRAM SUTAR
- PRIYA PRABHU
- SHOME SHARMA

GUIDED BY  
MR. ANUPAM WADEKAR  
(MASTER COMPUTECH)

AND

MR. RAHUL JINTURKAR  
(W.I.E.E.C.T)

DEPARTMENT OF COMPUTER  
WATUMULL INSTITUTE OF ELECTRONICS  
ENGINEERING AND COMPUTER TECHNOLOGY  
MUMBAI

2006-2007

**WATUMULL INSTITUTE OF ELECTRONICS  
ENGINEERING & COMPUTER TECHNOLOGY**  
47, R.G THADANI MARG, WORLI, MUMBAI 400018

**CERTIFICATE**

This is to verify that

Names

Exam Seat No.

1. Shahil Shah
2. Vikram Sutar
3. Priya Prabhu
4. Shomee Sharma

Of Final Year (VIII semester) degree course in Computer Engineering have  
completed the specified project report on

**ONLINE WEBSITE BUILDER**

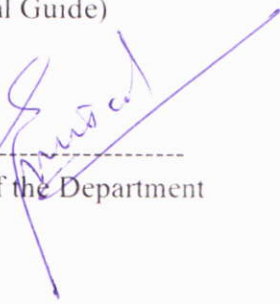
During the academic year July 2006 to May 2007



Prof. Rahul Jinturkar  
(Internal Guide)



External Examiner



Head of the Department



Principal

Date:



## ACKNOWLEDGEMENT

It gives us great pleasure in presenting our final year B.E. project report done with Master Computech Ltd. on an Online Website Builder.

This report includes information on what is an Online Website Builder. It also includes the working of the Website Builder and its functionality and scope in the industry.

We have completed this report from various sources and our research.

We are thankful to our Project Guide at Master Computech, Mr Anupam Wadekar for guiding us and helping us during this project.

We are also very thankful to our HOD Prof (Mr.) Prashant Misal and our Internal Project Guide Prof.(Mr.) Rahul Jinturkar for their support.

We hope this report will be informative and useful to the reader as it is intended to be.

Yours sincerely

Shahil Shah



Vikram Sutar



P. R. Prabhu.

Priya Prabhu

Shomee Sharma

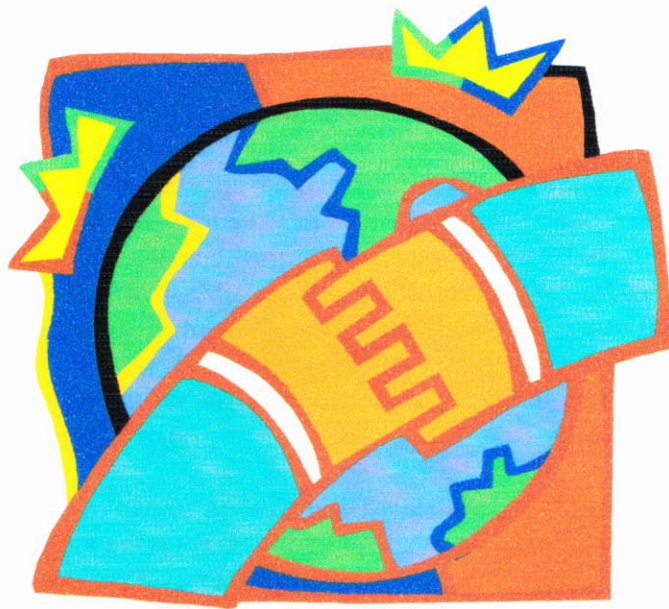




# TABLE OF CONTENTS

	PAGE NUMBER
COVER PAGE	i
COLLEGE CERTIFICATION	ii
COMPANY CERTIFICATION	iii
ACKNOWLEDGEMENT	iv
TABLE OF CONTENTS	v
1. INTRODUCTION	1
2. LITERATURE REVIEW	2
2.1 THE MAIN PAGE	2
2.2 THE INTERFACE	2
2.3 FILE EDITOR	3
2.4 FTP PROGRAM	5
2.5 USER CONTENT MANAGMENT	6
2.6 TECNOLOGIES USED	6
3. SOFTWARE DESIGN	7
3.1 DIAGRAMS	7
3.1.1 DFD LEVEL 0	7
3.1.2 DFD LEVEL 1	8
3.1.3 CFD	9
3.2 THE CODE	10
4. TEST CASES AND TEST RESULTS	14
4.1 UNIT TESTING	14
4.2 TESTING OF IMAGE UPLOADING	14
4.3 TESTING OF UCM MODULE	14
4.4 INTEGRATION TESTING	15
5. APPLICATIONS	16
6. CONCLUSION	17
7. APPENDIX	18
8. BIBLIOGRAPHY	20
9. REFERENCES	21

# ***INTRODUCTION***



# 1. INTRODUCTION

An Online Web Builder provides all the facilities that a person needs to select templates and create his own website. The major advantage being all the work is done online at the disposal of the user. Web site builders can be divided into online and off-line Web site builders. With an off-line application users will edit the site on their local computer and then publish it to the Web server. With online applications users edit their site using a Web browser. A Web Builder can be thought of as an online FrontPage or Publisher, which will help us create Websites by simply Dragging and adding contents onto a Predefined Design Template. The Designer need not worry about the corresponding code. It shall all be taken care off at the Back-end automatically.

A Web site builder, as the name suggests, is a software application or a bundle of applications that lets the user create a Web site without knowledge of the underlying markup and scripting languages like HTML, JavaScript, and ASP and so on. Therefore it is distinguished from an HTML editor. Often it will offer the user a collection of templates, pre-fabricated web pages that can be modified by the user. Some also offer additional features like shopping carts or surveys. Some Web site builders will give the experienced user the opportunity to directly edit the underlying code.

Web site builders can be divided into online and off-line Web site builders. With an off-line application users will edit the site on their local computer and then publish it to the Web server. With online applications users edit their site using a Web browser.

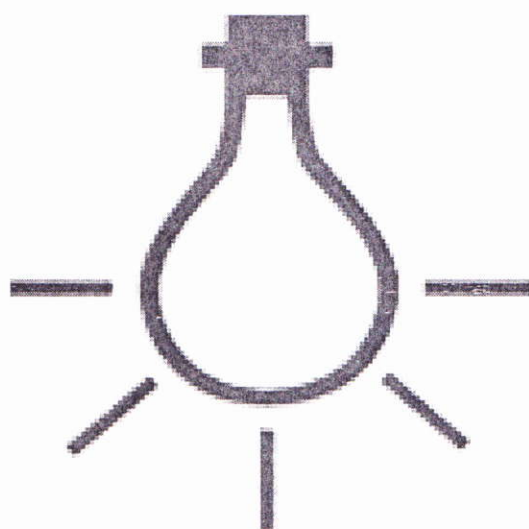
A Web Builder can be thought of as an online FrontPage or Publisher, which will help us create Websites by simply Dragging and adding contents onto a Predefined Design Template. Some Web site builders will give the experienced user the opportunity to directly edit the underlying code.

**WYSIWYG**(IPA Pronunciation [wɪziwɪg] or [wiziwɪg]), is an acronym for *What You See Is What You Get*, used in computing to describe a system in which content during editing appears very similar to the final product. It is commonly used for word processors, but has other applications, such as Web (HTML) authoring. The phrase was originally popularized by comedian Flip Wilson, whose character "Geraldine" would often say this to excuse her quirky behavior.

The Designer need not worry about the corresponding code. It shall all be taken care off at the Back-end automatically.



# *LITERATURE REVIEW*

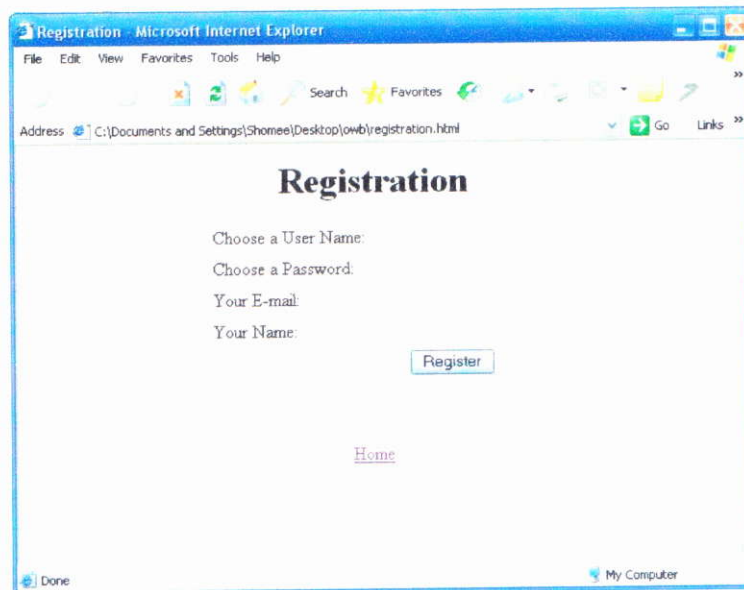


## 2. LITERATURE REVIEW

### 2.1 THE MAIN PAGE.

The main interface consists of a user login page. This consists of a simple login for users. The users are categorized into two types. Mainly the registered users and the new users. The registered users consist of those users who have previously visited the site and have a valid username and a corresponding password. The only thing they need to do is simply login and edit their pages as and when needed as per their needs. The only requirement being continuous and steady access to the internet!

The other types of users are “new users”. These are people who are new to the site. The people who are first time users can also register and get their login. Henceforth they can edit their pages at their convenience.



### 2.2 THE INTERFACE

Once the user has successfully logged in, he can make any changes on the previously uploaded page. This leads the user to a page called “Make a New Page”. This page also contains an option of logging out. When the user finishes his work he can log out as he wishes. This page contains many options for the user. These can be listed as follows.

### 2.2.1 VIEW USER PAGES

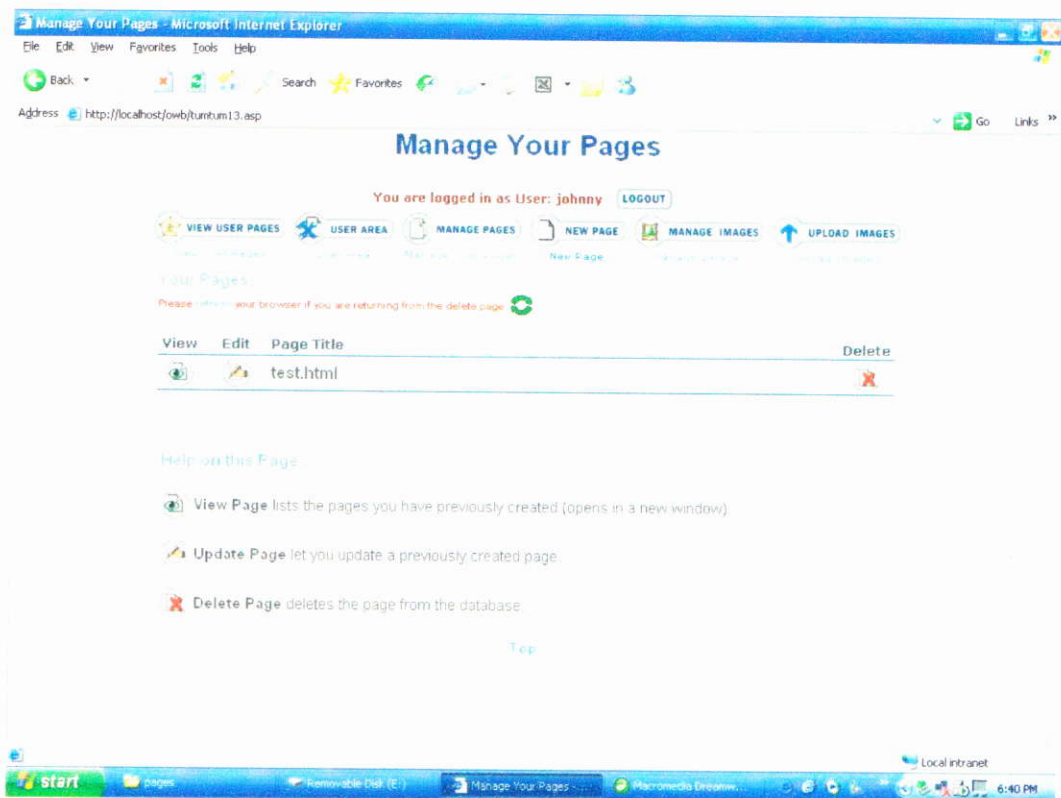
This is a link which leads the user to view the pages that he has already made before. He can view the pages as they would appear finally.

### 2.2.2 USER AREA

This leads the user to the workspace. This is the area where he can start designing his page. There are many options he can choose to modify his pages.

### 2.2.3 MANAGE PAGES

This option is used to store ,edit ,view & delete all the pages created by the user.



### 2.2.4 NEW PAGE

This option brings a brand new page for the user to put to use.

### 2.2.5 MANAGE IMAGES

This link manages all the images that the user wishes to use on his page.



### 2.2.1 VIEW USER PAGES

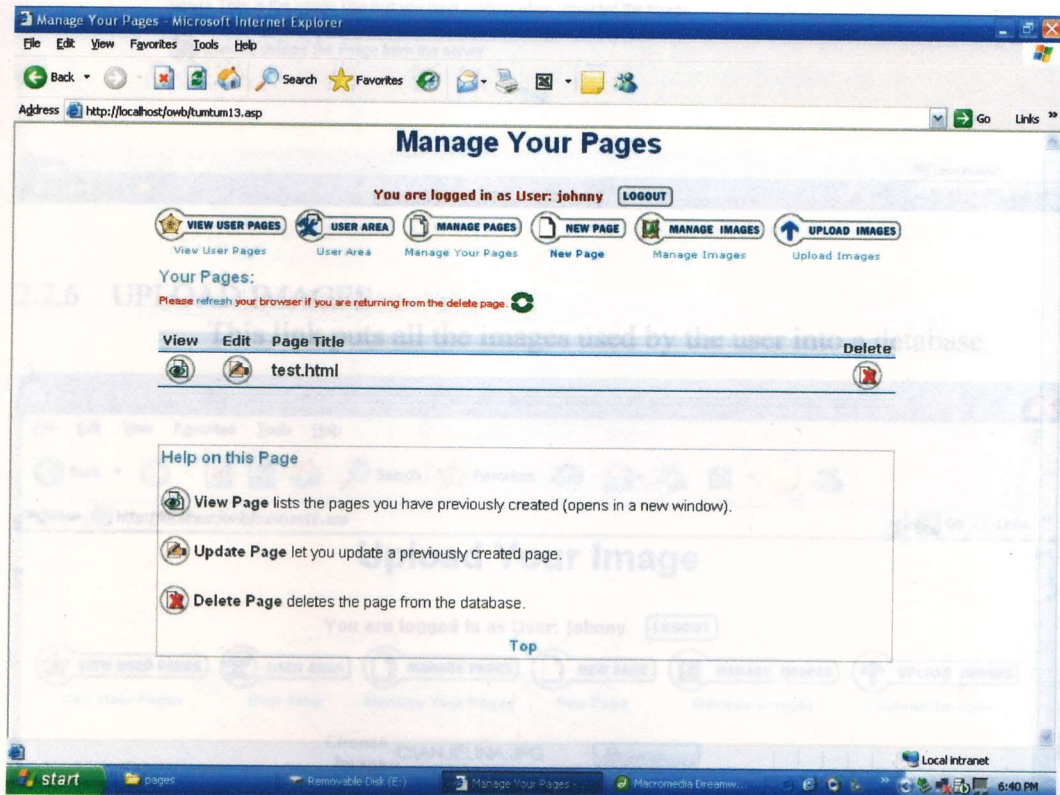
This is a link which leads the user to view the pages that he has already made before. He can view the pages as they would appear finally.

### 2.2.2 USER AREA

This leads the user to the workspace. This is the area where he can start designing his page. There are many options he can choose to modify his pages.

### 2.2.3 MANAGE PAGES

This option is used to store ,edit ,view & delete all the pages created by the user.

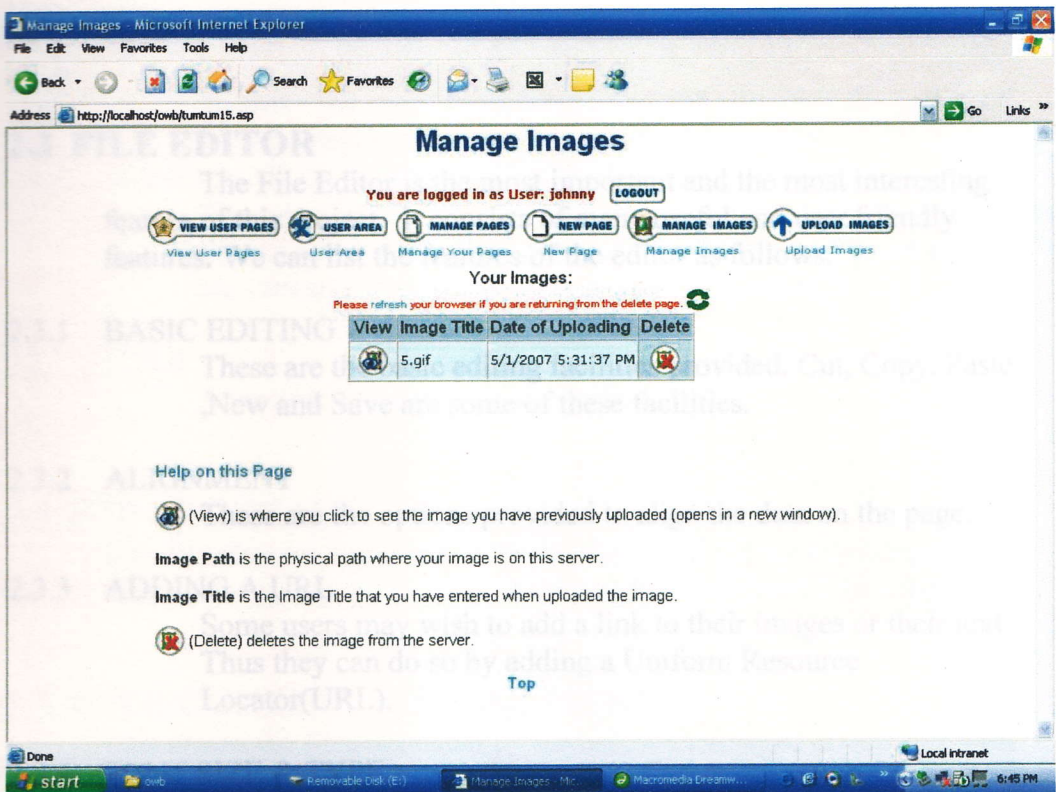


### 2.2.4 NEW PAGE

This option brings a brand new page for the user to put to use.

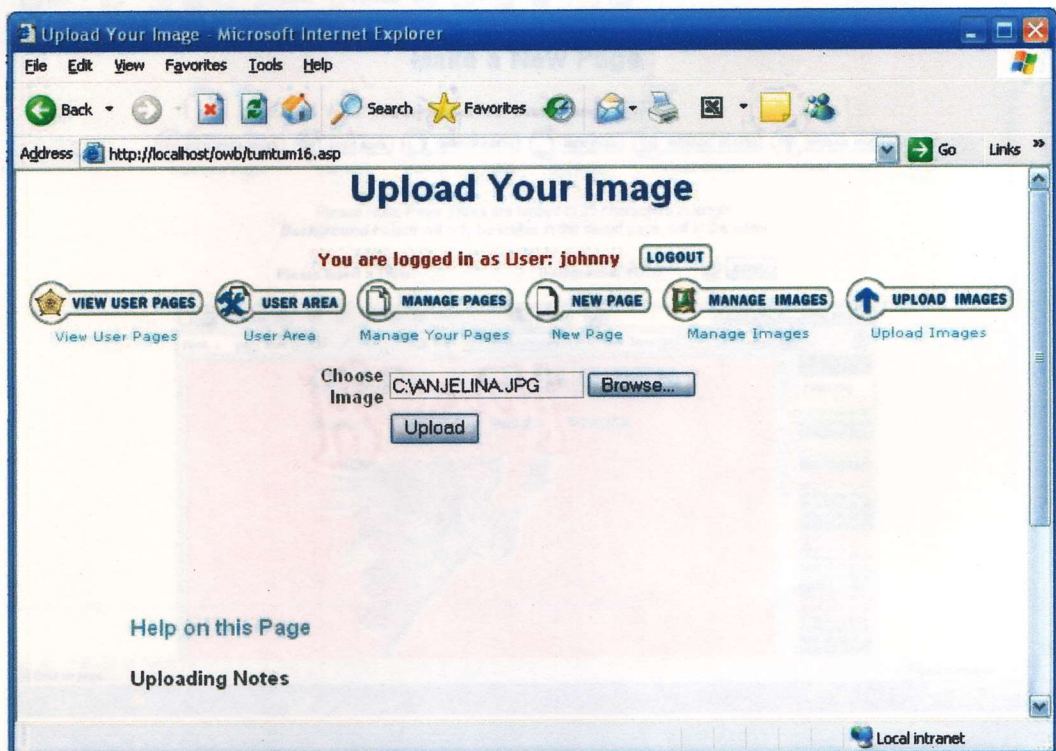
### 2.2.5 MANAGE IMAGES

This link manages all the images that the user wishes to use on his page.



## 2.2.6 UPLOAD IMAGES

This link puts all the images used by the user into a database.





## 2.3 FILE EDITOR

The File Editor is the most important and the most interesting feature of this project. It consists of many useful and user friendly features. We can list the features of the editor as follows.

### 2.3.1 BASIC EDITING

These are the basic editing facilities provided. Cut, Copy, Paste, New and Save are some of these facilities.

### 2.3.2 ALIGNMENT

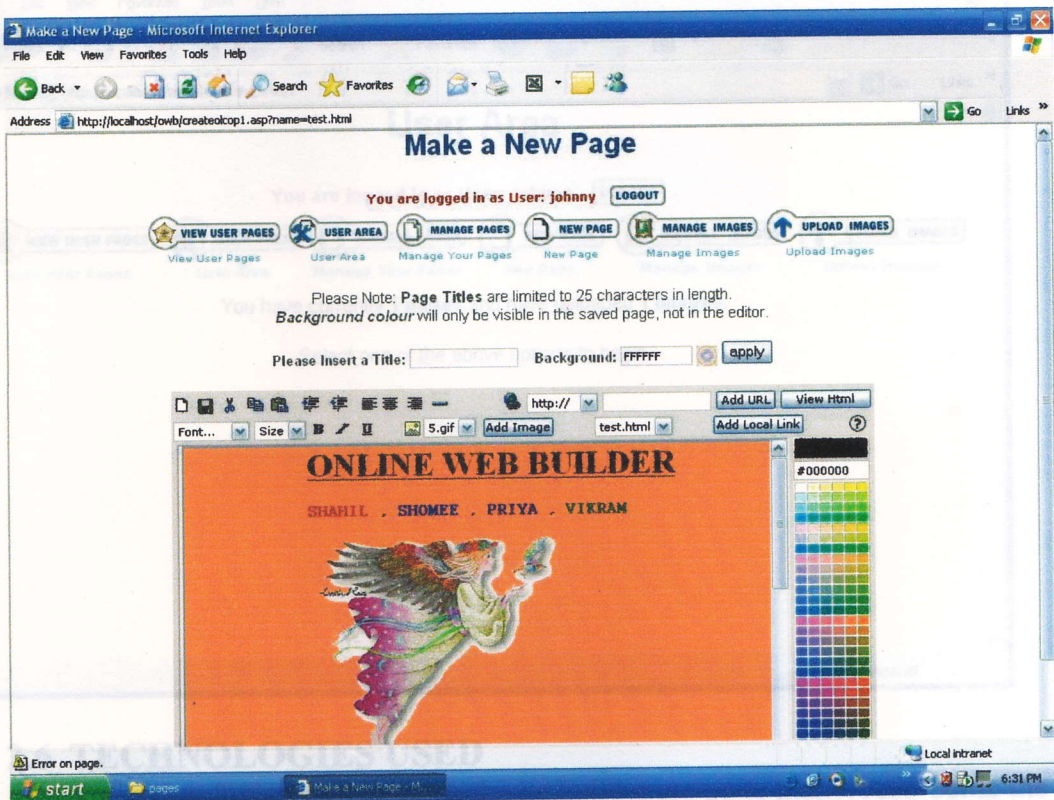
These are the options provided to align the data on the page.

### 2.3.3 ADDING A URL

Some users may wish to add a link to their images or their text. Thus they can do so by adding a Uniform Resource Locator(URL).

### 2.3.4 FONT SIZE & TYPE.

There are options available to enhance the font size and the type of font we may wish to use.





### 2.3.5 ADDING AN IMAGE & A LOCAL LINK

To make the pages more interesting and attractive the option of adding images and local links has been provided.

### 2.3.6 ADDING COLOURS.(BACKGROUND AND CONTENT)

The colors of the font as well as the background can be changed as per our needs using this option. A separate page pops up to specify the various colors and their corresponding codes.

### 2.3.7 TITLE.

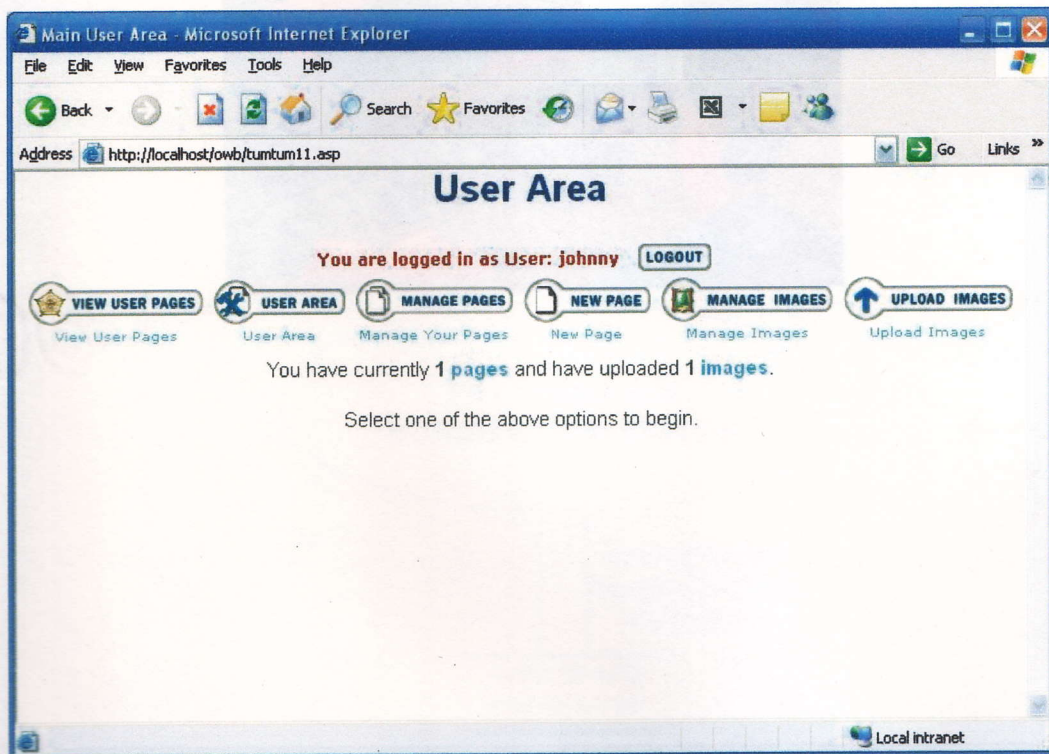
A suitable title can be given to the page.

## 2.4 FTP PROGRAMS

The files that are uploaded by the user to use to enhance his pages , need to be stored somewhere. These are uploaded and stored into a database.

## 2.5 USER CONTENT MANAGEMENT

This means the files uploaded by the user are managed effectively with the help of the backend.



## 2.6 TECHNOLOGIES USED

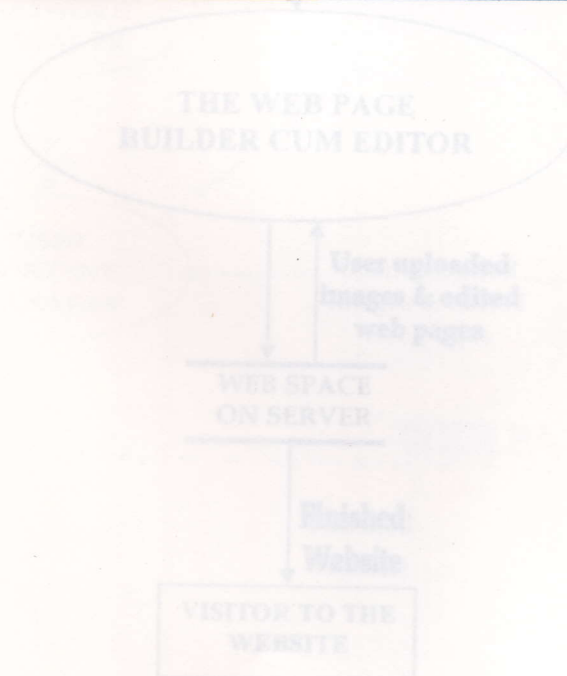
The technologies used in the web site builder are quite many. ASP being the heart of the project, JavaScript was also used. MS SQL was used to support the backend.

# SOFTWARE DESIGN

## 3. SOFTWARE DESIGN

### 3.1 DIAGRAMS

Fig 3.1.1 DATA FLOW DIAGRAM: LEVEL 0





### 3. SOFTWARE DESIGN

Fig 3.1.2 DATA FLOW DIAGRAM : LEVEL 1

#### 3.1 DIAGRAMS

Fig 3.1.1 DATA FLOW DIAGRAM: LEVEL 0

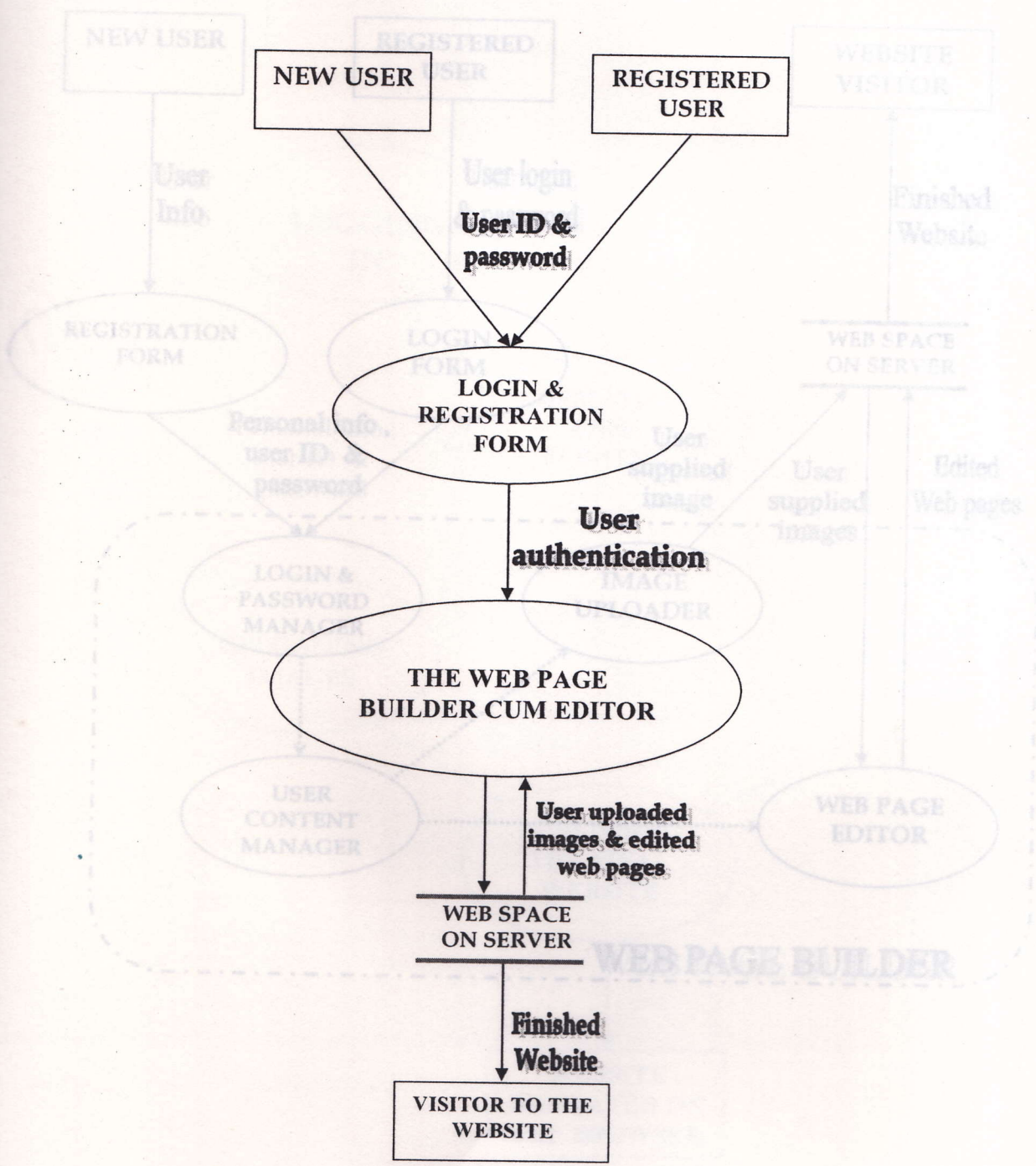
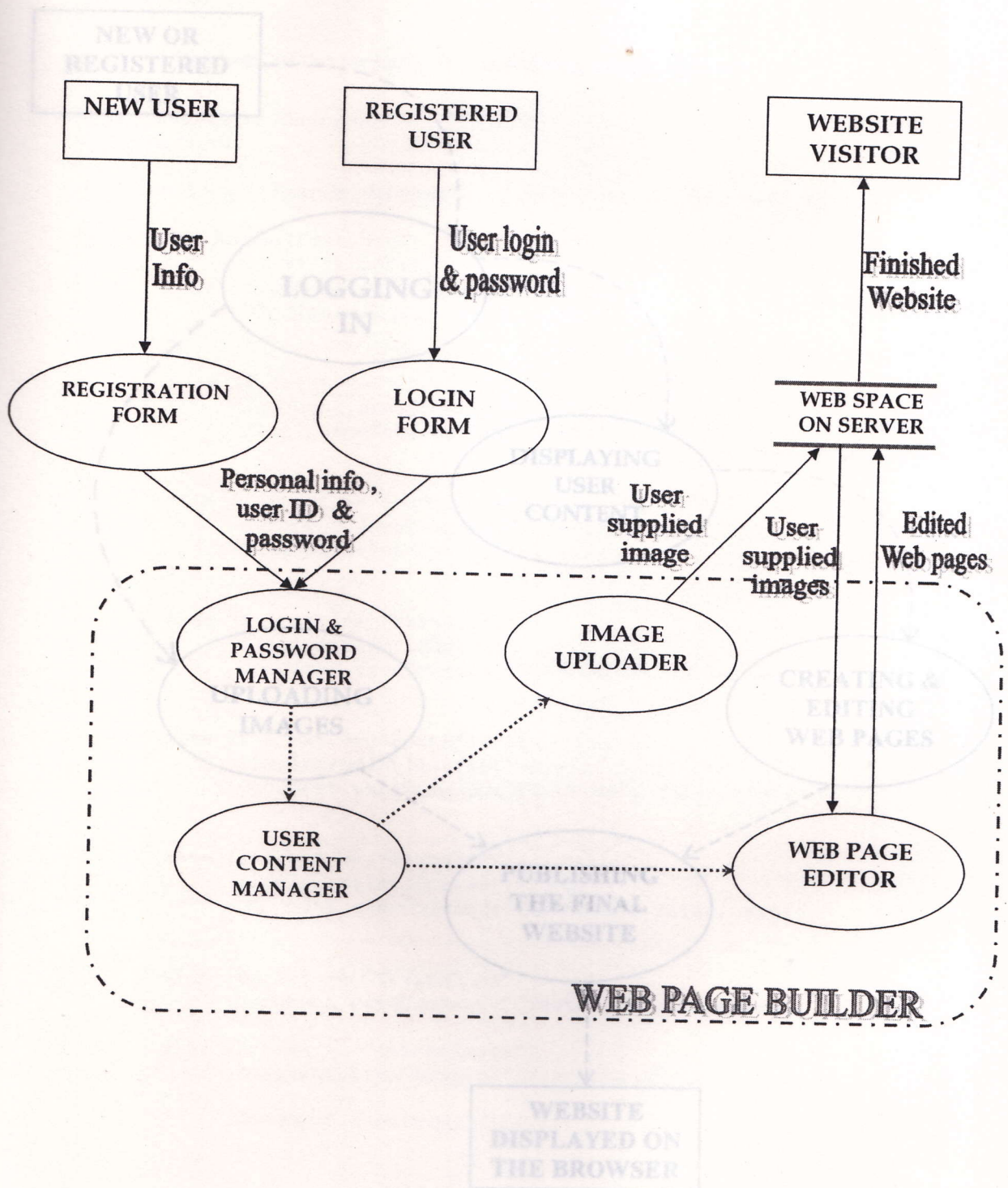
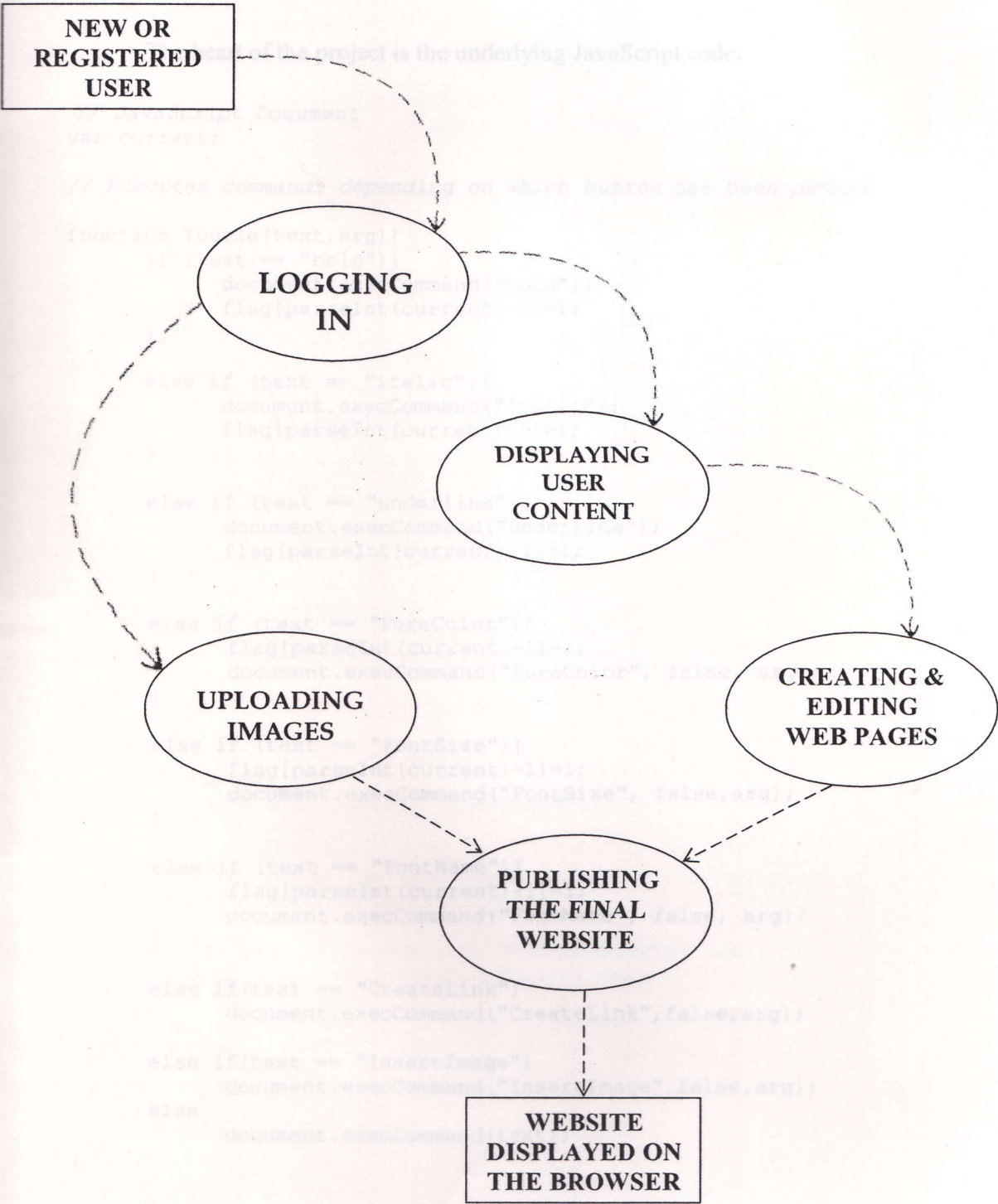




Fig 3.1.2 DATA FLOW DIAGRAM : LEVEL 1



**Fig 3.1.3 CONTROL FLOW DIAGRAM : LEVEL 1**





## 3.2 THE CODE:

### 3.2.1 execCommand code for Online Web Page Editor functions

The heart of the project is the underlying JavaScript code.

```
// JavaScript Document
var current;

// Executes commands depending on which button has been pushed

function Toggle(text, arg){
    if (text == "bold"){
        document.execCommand("bold");
        flag[parseInt(current)-1]=1;
    }
    else if (text == "italic"){
        document.execCommand("Italic");
        flag[parseInt(current)-1]=1;
    }
    else if (text == "underline"){
        document.execCommand("underline");
        flag[parseInt(current)-1]=1;
    }
    else if (text == "ForeColor"){
        flag[parseInt(current)-1]=1;
        document.execCommand("ForeColor", false, arg);
    }
    else if (text == "FontSize"){
        flag[parseInt(current)-1]=1;
        document.execCommand("FontSize", false, arg);
    }
    else if (text == "FontName"){
        flag[parseInt(current)-1]=1;
        document.execCommand("FontName", false, arg);
    }
    else if (text == "CreateLink")
        document.execCommand("CreateLink", false, arg);
    else if (text == "InsertImage")
        document.execCommand("InsertImage", false, arg);
    else
        document.execCommand(text);
}
```

### 3.2.2 Image Uploading Code (ASP Code)

//everytime you press enter key "<p>....</p>" tag is inserted between two lines so this function replaces "<p>....</p>" tag with ".....<br>"tag so that text appears line after line.

```
function fun1(e){
    var div1, div2;
    var current = window.event.srcElement.id;

    // this part removes white background of text which is typed
    after pressing enter key if any of the command buttons are not
    used.

    if(flag[parseInt(current)-1]==0&&init[parseInt(current)-
1]==1){
        init[parseInt(current)-1]=0;
        var c = window.event.srcElement.innerHTML;
        var d = c.replace(/\</font>/gi, "");
        c = "";
        c = d.replace(/\<font style="background-color:
#ffffff">/gi, "");
        window.event.srcElement.innerHTML = "";
        window.event.srcElement.innerHTML = c;
    }
    if(e.keyCode==13){
        init[parseInt(current)-1]=1;
        var s =
window.event.srcElement.innerHTML.replace(/\<p>&nbsp;</p>/gi, "");
        var c = s.replace(/\<p>/gi, "");
        s = "";
        s = c.replace(/\</P>/gi, "<br>");
        window.event.srcElement.innerHTML = "";
        window.event.srcElement.innerHTML = s;
    }
}

//function to start a new page
function newdoc(){
    document.all("ledit").innerHTML="";
}

//function to change the background colour of the web page.
function changebackgroundcolour(col){
    col = "#" + col;
    var element_div = document.getElementById("ledit");
    element_div.style.backgroundColor= col;
}

//end of code
```



### 3.2.2 Image Uploading Code (ASP Code)

```

<%
dim logoutPage
loginPage = "login1.html"
If (session.Contents("valid") <> "true") Then
Response.Redirect(loginPage)
%>
<%
Class FileUploader
Public Files
Private mcolFormElem

Public Default Sub Upload()
    Dim biData,
sInputName,i,p,nPosEnd1,nPosEnd2,data,oFs,oFile
    Dim nPosBegin, nPosEnd, nPos, vDataBounds,
nDataBoundPos
    Dim nPosFile, nPosBound,nIndex,npos1
    dim lk
    p=Request.TotalBytes
    biData = Request.BinaryRead(p)
    nPosEnd1 = InstrB(1, biData, CByteString("Content-
Type"))
    nPosEnd2 = InstrB(nPosEnd1, biData,
CByteString(Chr(13)))
    npos1 = InstrB(1, biData, CByteString("-----
-----7d61281170216--"))
    lk=p
    lk=lk-44
    Dim a,b,c,d,f,g,h,k,j
    c = InstrB(1, biData, CByteString("filename"))
    a= InstrB(c, biData, CByteString(ChrB(34)))

    b = InstrB(a, biData, CByteString("Content-Type"))
    b=b-chrB(51)
    d = MidB(biData,a, b-a)
    f = CWideString(d)

    g=Instr(1, f, ".")
    h=InstrRev(f,"\\",g)
    k=Mid(f,h+1,c)
    data = MidB(biData, nPosEnd2+4, lk-nPosEnd2-5)

    Set oFs =
Server.CreateObject("Scripting.FileSystemObject")
    If Not
oFS.FolderExists("c:\inetpub\wwwroot\user/"&session.Contents("use
r")&"/images/") Then Exit Sub
    Set oFile =
oFS.CreateTextFile("c:\inetpub\wwwroot\user/"&session.Contents("u
ser")&"/images/"&k, True)
    Response.Write "final<BR>"
    For nIndex = 1 to LenB(data)
        oFile.Write Chr(AscB(MidB(data,nIndex,1)))

```

## TESTING AND TEST CASES

```
Next

oFile.Close

response.Redirect("ok[1].asp")

End Sub

'String to byte string conversion
Private Function CByteString(sString)
    Dim nIndex
    For nIndex = 1 to Len(sString)
        CByteString = CByteString &
ChrB(AscB(Mid(sString,nIndex,1)))
    Next
End Function

'Byte string to string conversion
Private Function CWideString(bsString)
    Dim nIndex
    CWideString = ""
    For nIndex = 1 to LenB(bsString)
        CWideString = CWideString &
Chr(AscB(MidB(bsString,nIndex,1)))
    Next
End Function
End Class

%>

? refer appendix
```

# TESTING AND TEST CASES

A software project remains incomplete and vulnerable if it is not properly tested. Keeping this in mind, we have tested the web site builder accordingly. The various areas of the project were tested by all the members of the group.

## 4.1 UNIT TESTING

Every module of the code was tested separately to verify the smooth functioning of the entire system.



## 4.2 TESTING OF UPLOAD MODULE

The maximum file size was tested by the uploading form.



## 4.3 TESTING OF USER CONTENT MANAGEMENT MODULE

All the images & web pages uploaded by the user were successfully stored on the server.



## 4. TESTING AND TEST CASES.

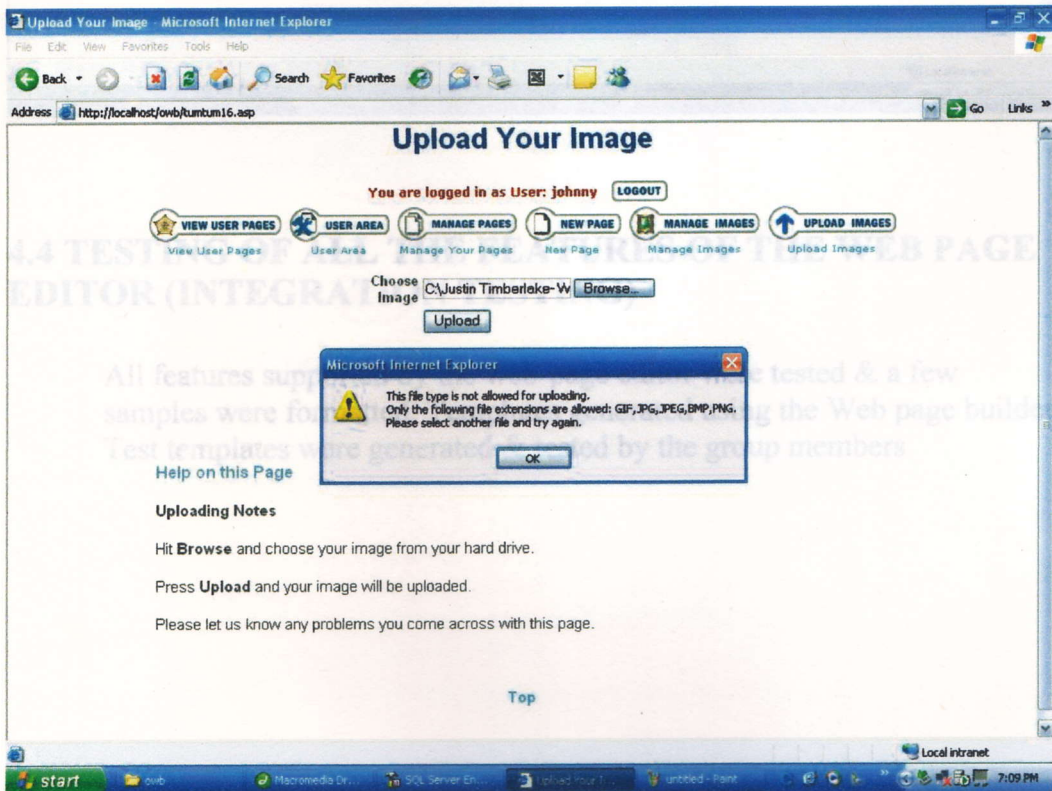
A software project remains incomplete and vulnerable if it is not properly tested. Keeping this in mind, we have tested the web site builder accordingly. The various areas of the project were tested by all the members of the group.

### 4.1 UNIT TESTING

Every module of the code was tested separately to verify the smooth functioning of the entire project.

### 4.2 TESTING OF IMAGE UPLOADING

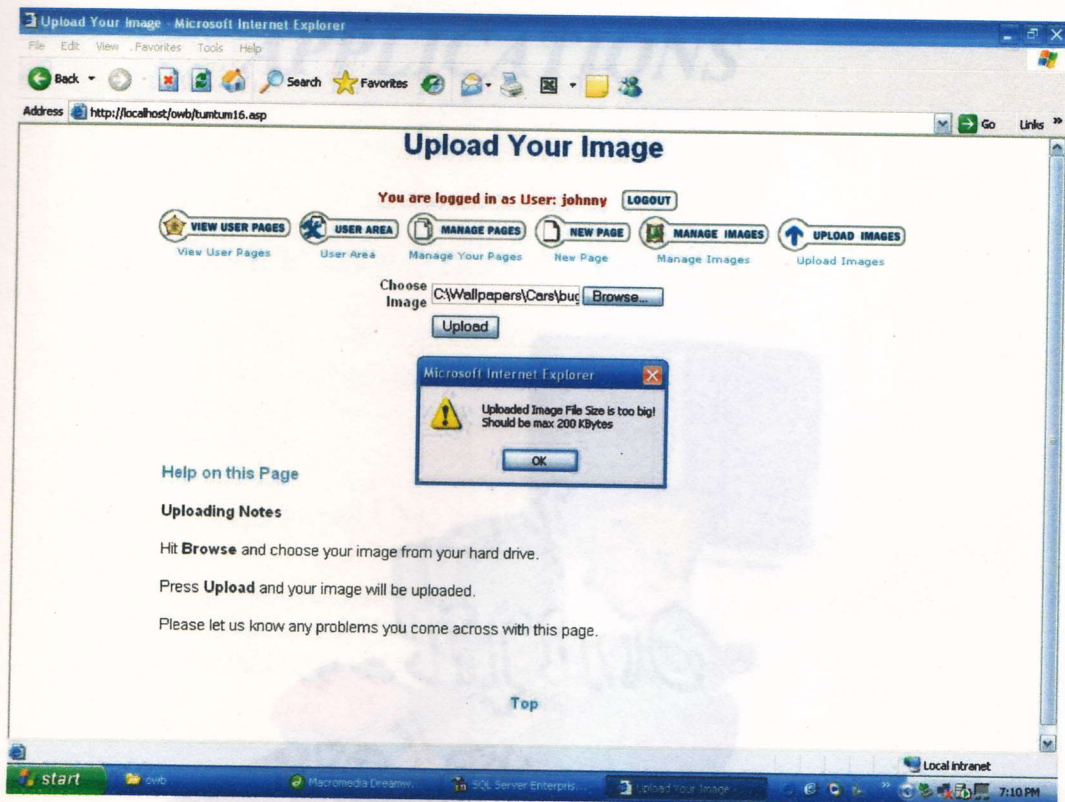
The max size & the different file formats allowed by the uploading form.



### 4.3 TESTING OF USER CONTENT MANAGEMENT MODULE

All the images & web pages uploaded by the user were successfully stored on the server





#### 4.4 TESTING OF ALL THE FEATURES OF THE WEB PAGE EDITOR (INTEGRATION TESTING)

All features supported by the web-page editor were tested & a few samples were formatted. Templates generated using the Web page builder Test templates were generated & tested by the group members

# APPLICATIONS

## 5. APPLICATIONS

### 5.1 CREATING PAGES ONLINE.

Using our online web page builder, a user can create web pages online, even while using a computer that has no supporting softwares like Dreamweaver installed on it.

### 5.2 INTERLINKING PAGES.

Just an internet connection is enough to enable the user to create a web page. The pages created by the user can be interlinked with other pages.

### 5.3 LINKS.

The user can create links to other websites, worldwide.

### 5.4 FAST.

Whenever someone is in urgent need of developing or posting a full-fledged commercially deployable web-site, he can use this web page builder and without requiring any technical expertise.

## ADVANTAGES

Upon logging into the website, we can see, create, edit, and delete web pages stored on the server without using any additional FTP program. Since the entire user data is stored on the server it is secure and can be accessed very quickly. Web pages created using our builder are Application-independent (no external web page editing software is required).



# CONCLUSION

## 5. APPLICATIONS

### 5.1 CREATING PAGES ONLINE.

Using our online web-page editing software, a user can create web pages online, even while working on a machine that has no supporting softwares like Dream Weaver, Front Page, Publisher installed on it

### 5.2 INTERLINKING MULTIPLE PAGES.

Just an internet connection is sufficient to enable the user to create a web Page. Multiple web-pages created by the user can be interlinked using our web-page editor.

### 5.3 LINKS

Also, links can be given to other posted websites, worldwide.

### 5.4 FAST.

Whenever someone is in urgent need of developing or posting a full-fledged commercially deployable web-site, he can use this web page builder and without requiring any technical expertise.

## ADVANTAGES

Upon logging into the website, we can see, create, edit, and delete web pages stored on the server without using any additional FTP program. Since the entire user data is stored on the server it is secure and can be accessed very quickly. Web pages created using our builder are Application-Independent (no external web page editing software is required)



# CONCLUSION

## 6. CONCLUSION

As computer engineers we always need to come up with new softwares. There are hordes of softwares existing in the market and many more would come up in the near future. But keeping all these things in mind, we have tried to design something different and offbeat. At the same time we have strived to keep it simple and user friendly - which is the basic purpose underlining all latest softwares.

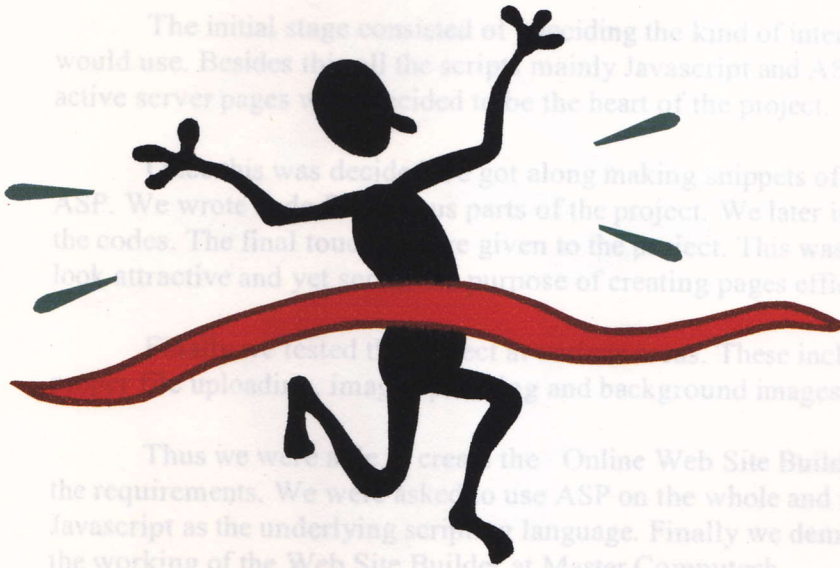
The initial stage consisted of deciding the kind of interface we would use. Besides that all the scripts - mainly Javascript and ASP that is active server pages - were decided to be the heart of the project.

This was decided and got along making snippets of code in ASP. We wrote the various parts of the project. We later integrated the codes. The final touch was given to the project. This was to make it look attractive and yet serve the purpose of creating pages efficiently.

We tested the project. These including uploading images and background images.

Thus we were able to create the Online Web Site Builder as per the requirements. We were able to use ASP on the whole and to use Javascript as the underlying scripting language. Finally we demonstrated the working of the Web Site Builder at Master Computech.

To sum it all, it has been a very rewarding and enriching experience for all four of us.



# APPENDIX

## 6. CONCLUSION

As computer engineers we always need to come up with new softwares. There are hordes of softwares existing in the market and many more would come up in the near future. But keeping all these things in mind, we have tried to design something different and offbeat. At the same time we have strived to keep it simple and user friendly which is the basic purpose underlining all latest softwares.

The initial stage consisted of deciding the kind of interface we would use. Besides this all the scripts mainly Javascript and ASP that is active server pages were decided to be the heart of the project.

Once this was decided we got along making snippets of code in ASP. We wrote code for various parts of the project. We later integrated the codes. The final touches were given to the project. This was to make it look attractive and yet server the purpose of creating pages efficiently.

Finally we tested the project at various areas. These including proper file uploading, image uploading and background images.

Thus we were able to create the Online Web Site Builder as per the requirements. We were asked to use ASP on the whole and to use Javascript as the underlying scripting language. Finally we demonstrated the working of the Web Site Builder at Master Computech.

To sum it all, it has been a very rewarding and enriching experience for all four of us.

# APPENDIX

## 7. APPENDIX

### • execCommand Method

Executes a command on the current document, current selection, or the given range.

Syntax

```
Object.execCommand(sCommand [, oInterface] [, vValue])
```

Parameters

*sCommand*

Command to execute. This is a string of one or more command identifiers that can be

*oInterface*

Optional. An object that implements the `Command` interface.

*vValue*

Optional. A value that is passed to the command. The string, number, or other value is passed to the command if the command supports one.

Return Value

Returns True if the command is successful.

Remarks

Do not invoke the `execCommand` method until after the page loads.

The `oInterface` and `vValue` parameters might be required depending on the command being executed.

### • MS SQL

Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) used by both Microsoft and Sybase. SQL Server is commonly used by businesses for small- to medium-sized databases, but the past five years have seen greater adoption of the product for larger enterprise databases.



## 7. APPENDIX

### • **execCommand Method**

Executes a command on the current document, current selection, or the given range.

Syntax

```
bSuccess = object.execCommand(sCommand [, bUserInterface] [, vValue])
```

Parameters

<i>sCommand</i>	Required. <b>String</b> that specifies the command to execute. This command can be any of the <u>command identifiers</u> that can be executed in script.				
<i>bUserInterface</i>	Optional. <b>Boolean</b> that specifies one of the following values. <table><tr><td>false</td><td>Default. Does not display a user interface.</td></tr><tr><td>true</td><td>Displays a user interface, if the command supports one.</td></tr></table>	false	Default. Does not display a user interface.	true	Displays a user interface, if the command supports one.
false	Default. Does not display a user interface.				
true	Displays a user interface, if the command supports one.				
<i>vValue</i>	Optional. <b>Variant</b> that specifies the string, number, or other value to assign. Possible values depend on <i>sCommand</i> .				

Return Value

Returns True if the command is successful.

Remarks

Do not invoke the **execCommand** method until after the page loads.

The *bUserInterface* and *vValue* parameters might be required depending on the command being executed.

### • **MS SQL**

**Microsoft SQL Server** is a relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) used by both Microsoft and Sybase. SQL Server is commonly used by businesses for small- to medium-sized databases, but the past five years have seen greater adoption of the product for larger enterprise databases.

## • Binary Read

The BinaryRead method is used to retrieve the data sent to the server from the client as part of a POST request. It will store the data in a safe array (an array that stores information about the number of dimensions and the bounds of its dimensions).

**Note:** A call to Request.Form after a call to BinaryRead, and vice-versa, will cause an error.

### Syntax

```
Request.BinaryRead(count)
```

Parameter	Description
count	Required. Specifies how many bytes to read from the client

## • UCM (User Content Management)

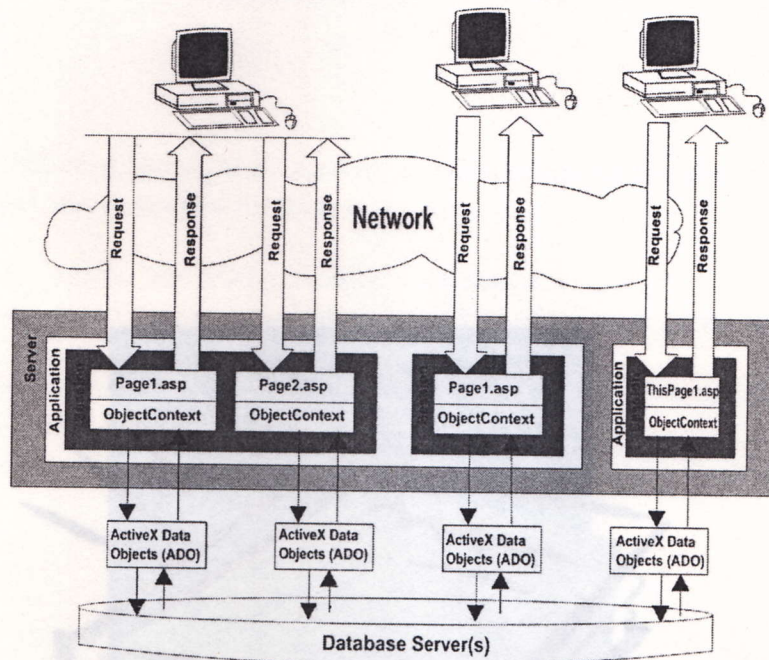
This module is used to manage the images and content uploaded by the user, onto his web space.

### • ASP Objects

1. The ASP Response object is used to send output to the user from the server.
2. The ASP Request object is used to get information from the user.
3. A group of ASP files that work together to perform some purpose is called an application. The Application object in ASP is used to tie these files together.
4. The Session object is used to store information about, or change settings for a user session. Variables stored in the Session object hold information about one single user, and are available to all pages in one application
5. The ASP Server object is used to access properties and methods on the server.
6. The ASPError object is used to display detailed information of any error that occurs in scripts in an ASP page.



# BIBLIOGRAPHY



- **Web Page Editor**

It is used to make all the modifications of the user's content, starting from basic editing, to higher level modifications.

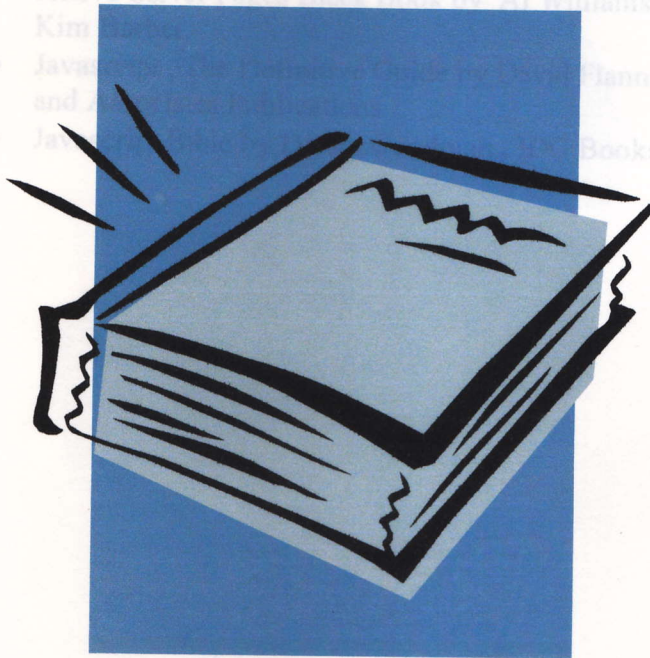


# ***BIBLIOGRAPHY***

## **8. BIBLIOGRAPHY**

The following books were used by us in the course of completing this project and compiling the project report..

- Active Server Pages 3.0 by Ruth Maran .
- Active Server Pages Black Book by Al Williams, Paul Newkirk and Kim L. Johnson .
- Java: The Complete Reference, 2nd Edition by David Flanagan , O'Reilly .
- Java: The Complete Reference, 2nd Edition by David Flanagan , O'Reilly .
- Java: The Complete Reference, 2nd Edition by David Flanagan , O'Reilly .



# REFERENCES

## 8. BIBLIOGRAPHY

The following books were used by us in the course of completing this project and compiling the project report..

- Active Server Pages 3.0 by Ruth Maran .
- Active Server Pages Black Book by Al Williams, Paul Newkirk and Kim Barber.
- Javascript , The Definitive Guide by David Flannagan , O'Reilly and Associates Publications.
- Javascript Bible by Danny Goodman , IDG Books Worldwide.

# ***REFERENCES***

## **REFERENCES**

The following Web sites and Online References helped us in the process of completing our project and compiling our project report.

- [www.lcmnsp.com](http://www.lcmnsp.com)
- [www.ty3schools.com](http://www.ty3schools.com)
- [www.cynit.org](http://www.cynit.org)
- [www.javascripthall.com](http://www.javascripthall.com)
- [www.cs.tut.fi](http://www.cs.tut.fi)





## 9. REFERENCES

The following Web sites and Online References helped us in the process of completing our project and compiling our project report.

- [www.learnasp.com](http://www.learnasp.com)
- [www.w3schools.com](http://www.w3schools.com)
- [www.evolt.org](http://www.evolt.org)
- [www.javascriptmall.com](http://www.javascriptmall.com)
- [www.cs.tut.fi](http://www.cs.tut.fi)