

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**INTERACTIVE WEB PAGE DESIGN FOR
COUNTRY (PALESTINE ONLINE)**

**Graduation Project
COM-400**

Student: Muath Ismail (20021511)

Supervisor: Assist. Prof. Dr. Adil Amirjanov

Nicosia - 2006

ACKNOWLEDGEMENT

*Firs of all, and before saying any thing, I would like to thank ALLAH-our creator-
For his blessing and the patience he gave me through out my life and specially during my
education.*

*Secondly, I would like to show my special appreciation to my supervisor
Assist. Prof. Dr. Adil Amirjanov for his invaluable advice and for the generosity he
exhibited with his time and effort over this project.*

*I would like to express my gratitude to Prof. Dr. Senol Bektas(Vice President).
Also I would like to express my gratitude to the members of ABAC Club, specially my
close friend Eng. Muhammad Elfawaier for their endless support to me during my
university life.*

*After that, I love to send my special thanks to my family especially parents and my uncle,
for the support they gave me financially and morally. It was not possible to be what I am
without them. I won't ever forget their encouragement and support as long as I am alive.
Thank you father, you have always been my ideal. Mother, I am so glad to thank you at
this particular moment of my life, you prayers helped me making this day come true.
Thank you Uncle, I won't ever forget your financial and educational support to me. I also
thank my brothers, my sisters. I love you all so much.*

*In the last but not least I would like to thank my entire friends with whom I have shared
great times. I wish you all the best. And may ALLAH bless you all.*

ABSTRACT

Recently, the Internet become one of the most important intermedia and connections in the world, therefore it has a considerable position in the interaction between nations.

According to the important of the information about countries on Internet, especially tourist, political and economical information to illustrate the situation of those countries for the visitors (tourists) and also according to the changeable events in news over the world, it has been chosen as quick intermedia.

TABLE OF CONTENT

ACKNOWLEDGMENT.....	i
ABSTRACT.....	ii
TABLE OF CONTENTS.....	iii
INTRODUCTION.....	viii
CHAPTER ONE: The Hyper Text Markup Language.	
1.1 Introduction to html.....	1
1.2 Versions of HTML.....	2
1.3 Elements in HTML Documents	3
1.3.1 Empty Elements.....	3
1.3.2 Upper and Lower Case.....	3
1.3.3 Elements can have Attributes.....	4
1.4 HTML Document Structure	4
1.5 HTML BODY.....	5
1.6 Organization of Elements in the BODY.....	6
1.6.1 Text Block Elements.....	6
1.6.2 Text Emphasis Elements.....	7
1.6.3 Special Elements Hypertext Anchors	7
1.7 Basic HTML Tags.....	7
1.7.1 Headings.....	7
1.7.2 Paragraphs.....	8
1.7.3 Line Breaks.....	8
1.7.4Comments in HTML.....	8
1.8 HTML Links.....	8
1.8.1The Anchor Tag and the Href Attribute.....	8
1.8.2 The Target Attribute.....	9

1.8.3 The Anchor Tag and the Name Attribute.....	9
1.9 Preformatted Text	10
1.10 Tables.....	11
1.10.1 Tables and the Border Attribute.....	11
1.10.2 Headings in a Table.....	11
1.10.3 Empty Cells in a Table.....	12
1.10.4 Table Tags.....	13
1.11 HTML Forms and Input.....	13
1.11.1 Input.....	14
1.11.2 TEXT FIELDS.....	14
1.12 What do FORMS do?	14
CHAPTER TWO ACTIVE SERVER PAGES	
2.1 The need for ASP.....	16
2.2 What is ASP?.....	17
2.3 Active Server Pages 3.0.....	18
2.3.1 What Can We Do with Active Server Pages?.....	18
2.3.2 Server-Side Scripts.....	19
2.4 Running ASP.....	19
2.4.1 Internet Information Services.....	19
2.4.2 Personal Web Server.....	19
2.5 Beginning ASP.....	19
2.5.1 Steps for Installation.....	20
2.5.2 Creating Virtual Directories.....	20
2.6 Accessing your webpage.....	20
2.6.1 Local host.....	20
2.7 The Database connectivity.....	21

2.7.1 DSN	21
2.7.2 Connecting to the Database.....	21
2.8 Summary.....	22

CHAPTER THREE VISUAL BASIC SCRIPT

3.1 Overview.....	23
3.2 What is VBScript?.....	23
3.3 How does it Work?.....	23
3.4 How to Put VBScript Code in an HTML Document.....	23
3.5 How to Handle Older Browsers.....	24
3.6 Where to Put the VBScript.....	24
3.6.1 Scripts in the head section.....	25
3.6.2 Scripts in the body section.....	25
3.6.3 Scripts in both the body and the head section.....	25
3.7 VBScript Conditional Statements.....	26
3.7.1 Conditional Statements.....	26
3.7.1.1 if... Then...Else.....	26
3.7.1.2 if... Then...Else if.....	27
3.7.1.3 Select Case.....	28
3.8 VB Script and ASP.....	28
3.9 Summary.....	29

CHAPTER FOUR DATABASE USING MICROSOFT ACCESS

4.1 Introduction.....	30
4.2 Create a database using the Database Wizard	31
4.3 Create a database without using the Database Wizard	31
4.3.1 Relationships	32
4.4 Project's DATABASE.....	34

4.4.1 Data Base Design.....	34
4.4.2 Database Tables.....	35
4.4.2.1 Admin's tables.....	35
4.4.2.2 Cities' Table.....	36
4.4.2.3 Hotels' Table.....	36
4.4.2.4 News's Table.....	36
4.4.2.5 World of News's Table.....	37
4.4.2.5 Famous Persons' Table.....	37
4.4.2.5 Customers Information's Tables.....	38
4.4.2.5.1 Customers Hotels Information's Table.....	38
4.4.2.5.2 Customers Rent Cars Information's Table.....	38

CHAPTE FIVE WEB SITE DESIGN

5.1 Introduction.....	40
5.2 System Design of The Web Site.....	40
5.3 The Content of Project.....	41
5.3.1 First page (home page).....	41
5.3.2 News.Asp.....	41
5.3.3 Cities. Asp.....	43
5.3.4 Famous Persons.asp.....	44
5.3.5 Rent Car.asp.....	45
5.3.6 Hotel.asp.....	46

5.4 Administrator Design.....	47
5.4.1 Insert.asp page.....	47
5.4.2 Delete.asp Page.....	48
5.4.3 Update.asp page	49
CONCLUSION.....	50
REFERENCES.....	51

INTRODUCTION

The project is about an interaction country website, this project will show how the all pages are interacting with each other and with other programs, and we will that all this interacting occurs by using the ASP language inside the internal pages of site.

In order to design such Pages as country page was decided to display, first, the news of the country besides the world news. Second the major cities, historical places, famous persons and photo gallery. Third some information about hotels and travel agents to be as a guide for the country's visitors. And to be able to follow the modern information from time to another were decided to use database which would be enable to add and improve and delete the current information, and to able to add the breaking news. In order to link the database with pages, was decided to use ADO system which throw it could make the WebPages interactive during the database, this belongs the practical side of the project whereas the theoretical side of the project is talking about an interactive web site; actually country website, it contains five chapters.

Chapter 1 talks about HTML; a brief description about it, it's tags and elements, HTML Structure, HTML links, creating tables, adding images to the web pages, etc

Chapter 2 talks about the ASP and to connect to database how to connect databases to the Internet and how to interact with users using theses technologies.

Chapter 3 talks about one of the scripting languages, which is VBScript, talks about how to Put VBScript Code in an HTML Document, and about VBScript with ASP

Chapter 4 talks about Ms Access, how to create table, how to insert, delete, update, retrieve data using SQL statements. And it talks about the project database and tables.

Chapter 5 is the practical part of the project that is the web site.

CHAPTER ONE

The Hyper Text Markup Language

1.1 Introduction to html

The Hyper Text Markup Language (HTML) is a simple markup language used to create hypertext documents that are portable from one platform to another. HTML or HyperText Markup Language is designed to specify the logical organization of a document, with important hypertext extensions. It is not designed to be the language of a WYSIWYG word processor such as Word or WordPerfect. This choice was made because many different "browsers", of very different abilities, may view the same HTML document. Thus, for example, HTML allows you to mark selections of text as titles or paragraphs, and then leaves the interpretation of these marked elements up to the browser. For example one browser may indent the beginning of a paragraph, while another may only leave a blank line.

HTML instructions divide the text of a document into blocks called elements. These can be divided into two broad categories -- those that define how the BODY of the document is to be displayed by the browser, and those that define information 'about' the document, such as the title or relationships to other documents. The vocabulary of these elements and a description of the overall design of HTML documents is given in the rest of Section 2. The Last part of the section also describes standard naming schemes for HTML documents and related files.

The detailed rules for HTML (the names of the tags/elements, how they can be used) are defined using another language known as the standard generalized markup language, or SGML. SGML is wickedly difficult, and was designed for massive document collections, such as repair manuals for F-16 fighters, or maintenance plans for nuclear submarines. Fortunately, HTML is much simpler!

However, SGML has useful features that HTML lacks. For this reason, markup language and software experts have developed a new language, called XML (the extensible markup language) that has most of the most useful features of HTML and SGML.

extensible markup language) that has most of the most useful features of HTML and SGML.

1.2 Versions of HTML

HTML is an evolving language, and each new version is given a number. The first definitive version was HTML 2.0 -- this had most of the elements we know and love, but was missing some of the Netscape/Microsoft extensions, and did not support tables, or ALIGN attributes.

HTML 3 (late 1995) was an ambitious effort on the part of Dave Raggett to upgrade the features and utility of HTML. However, it was never completed or implemented, although many features were integrated in the next "official" version of HTML, known as HTML 3.2.

HTML 3.2 was the next official version, integrating support for TABLES, image, heading and other element ALIGN attributes, and a few other finicky details. HTML 3.2 is the current "universal" dialect -- essentially all browsers understand HTML 3.2. It was, however, missing some of the Netscape/Microsoft extensions, such as FRAMES, EMBED and APPLET. Support for these (after a fashion) came in HTML 4.0

HTML 4.01 is the current official standard. It includes support for most of the proprietary extensions, plus support for extra features (Internationalized documents, support for Cascading Style Sheets, extra TABLE, FORM, and JavaScript enhancements), that are not universally supported. Some of these enhancements are not discussed in these online notes. They are discussed in detail in The HTML 4.0 Sourcebook, available in most bookstores.

For more details about the various versions of HTML, you are also referred to the bibliography, which contains references to the various specifications, and to my online resource that defines the structural rules of the different HTML variants, however, the evolution of HTML has now ceased -- HTML 4.01 is the last version of HTML. For the future, a new language, called XHTML -- for the extensible HyperText Markup Language, is replacing HTML. The differences are actually small, but important, as described on the next section.

However, HTML and XHTML are very similar, so the things you learn in this tutorial will still be useful. My newer book, *The XHTML 1.0 Language and Design Sourcebook*, explains in details these differences and how to use the new XHTML language. On the other hand, Section 2.5 of this tutorial covers the most important differences, and costs a whole lot less!

1.3 Elements in HTML Documents

The HTML instructions, along with the text to which the instructions apply, are called HTML *elements*. The HTML instructions are themselves called *tags*, and look like `<element_name>` -- that is, they are simply the element name surrounded by left and right angle brackets.

Most elements mark blocks of the document for particular purpose or formatting: the above `<element_name>` tag marks the beginning of such as section. The end of this section is then marked by the *ending* tag `</element_name>` -- note the leading slash character "/" that appears in front of the element name in an end tag. This leading slash character always indicates end, or stop tags.

For example, the heading at the top of this page is an H2 element, (a level 2 heading) which is written as:

```
<H2> 2.1 Elements in HTML </H2>.
```

1.3.1 Empty Elements

Some elements are *empty* -- that is, they do not affect a block of the document in some way. These elements do not require an ending *tag*. An example is the `<HR>` element, which draws a horizontal line across the page. This element would simply be entered as `<HR>`

1.3.2 Upper and Lower Case

Element names are case *insensitive*. Thus, the horizontal rule element can be written as any of `<hr>`, `<Hr>` or `<HR>`.

1.3.3 Elements can have Attributes

Many elements can have arguments that pass parameters to the interpreter handling this element. These arguments are called *attributes* of the element. For example, consider the element A, which marks a region of text as the beginning (or end) of a hypertext link. This element can have several attributes. One of them, HREF, specifies the hypertext document to which the marked piece of text is linked. To specify this in the tag for A you write:

```
<A HREF="http://www.somewhere.ca/file.html"> marked text </a>.
```

Where the attribute HREF is assigned the indicated value. Note that the A element is not empty, and that it is closed by the tag . Note also that end tags *never* take attributes -- the attributes to an element are always placed in the start tag.

1.4 HTML Document Structure

HTML documents are structured into two parts, the HEAD, and the BODY. Both of these are contained within the HTML element -- this element simply denotes this as an HTML document.

The head contains information about the document that is not generally displayed with the document, such as its TITLE. The BODY contains the body of the text, and is where you place the document material to be displayed. Elements allowed inside the HEAD, such as TITLE, are not allowed inside the BODY, and vice versa.

Example of Document Structure

```
<HTML>
  <HEAD>
    <TITLE> Environmental Change Project </TITLE>
  </HEAD>
  <BODY>
    <h1> Environmental Change Project </h1>
```

Welcome to the home page of the Environmental Change Project.

This project is different from other projects with similar names. In our case we actually *wish* to change the climate. For example, we would like hot beaches in Northern Quebec, and deserts near Chicago.

```
<p>So how will we do this. Well we do the following
<ul>
<li><A HREF="burn.html">Burn</A> more forests.
<li>Destroy the
    <A HREF="http://who.zoo.do/ozone.html">Ozone</A> layer.
<li>Birth more <A HREF="ftp:foo.do.do/cows.gif">cows</a>
</ul>

</BODY>
</HTML>
```

1.5 HTML BODY

The BODY element contains all the *content* of a document, as opposed to the HEAD, which contains *information about* the document. Various mark-up elements are allowed within the body to indicate headings, paragraphs, lists, hypertext links, images, and so on. These are described and illustrated in documents following this one.

The following shows typical use of HEAD and BODY elements, using this document as an example:

```
<HTML>
<HEAD>
<TITLE> BODY element in HTML </TITLE>
<BASE HREF="http://www.utoronto.ca/webdocs/HTMLdocs/NewHTML/body.html">
</HEAD>

<BODY>
```

[ToC]

[Up]

<H1> BODY Element in HTML </H1>

<p> The BODY element contains all the content of the document,

As opposed to the HEAD, which

Contains information about the document. </p>...

</BODY>

</HTML>

1.6 Organization of Elements in the BODY

1.6.1 Text Block Elements

As mentioned on the previous page, the BODY element contains all the displayed content of a document. Structurally, the document content is organized into blocks of text, such as paragraphs, lists, headings, paragraphs, block quotations, and so on. These are generically called block elements, since they "block" chunks of text together into logical units. Block elements can often contain other blocks -- for example, a list item can contain paragraphs or block quotations, so that these elements can often nest together.

The block-level elements are:

- Hn (Headings)
- P
- PRE
- HR
- FORM
- TABLE
- DIV (HTML 3.2 and up)

1.6.2 Text Emphasis Elements

At the next level down are text-level markup elements. These are elements that mark text for special meanings, for example, that a particular piece of text is emphasized (EM) or a citation (CITE), or that specify the desired physical formatting, such as boldface (B) or italics (I). These elements can usually appear anywhere inside a block element, with a few exceptions (you can't have images inside a PRE element).

1.6.3 Special Elements -- Hypertext Anchors

Analogous to the text-level markup is the anchor (A) element. This is the element that marks hypertext links. Obviously you want to know a lot about this one.

D. Character-Level Elements

Then are what I call character-level elements, namely line breaks (BR) and images (IMG). These are treated much like characters, and can appear wherever there is a character in a document.

E. Character References

Finally there are character or entity references. These are special HTML "escape" codes that can be used to enter special characters that are hard to type, such as accented or other non-ascii characters. You also need to use these to type angle brackets or ampersand characters -- as these are otherwise interpreted as HTML tags (< ... >) or as the beginnings of character or entity references (&).

1.7 Basic HTML Tags

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

1.7.1 Headings

Headings are defined with the <h1> to <h6> tags. <h1> defines the largest heading. <h6> defines the smallest heading.

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
<h4>This is a heading</h4>
<h5>This is a heading</h5>
```

```
<h6>This is a heading</h6>
```

HTML automatically adds an extra blank line before and after a heading.

1.7.2 Paragraphs

Paragraphs are defined with the `<p>` tag.

```
<p>This is a paragraph</p>  
<p>This is another paragraph</p>
```

HTML automatically adds an extra blank line before and after a paragraph.

1.7.3 Line Breaks

The `
` tag is used when you want to end a line, but don't want to start a new paragraph. The `
` tag forces a line break wherever you place it.

```
<p>This <br> is a para<br>graph with line breaks</p>
```

The `
` tag is an empty tag. It has no closing tag.

1.7.4 Comments in HTML

The comment tag is used to insert a comment in the HTML source code. The browser will ignore a comment. You can use comments to explain your code, which can help you when you edit the source code at a later date.

```
<!-- This is a comment -->
```

Note that you need an exclamation point after the opening bracket, but not before the closing bracket.

1.8 HTML Links

HTML uses a hyperlink to link to another document on the Web.

1.8.1 The Anchor Tag and the Href Attribute

HTML uses the `<a>` (anchor) tag to create a link to another document.

An anchor can point to any resource on the Web: an HTML page, an image, a sound file, a movie, etc.

The syntax of creating an anchor:

```
<a href="url">Text to be displayed</a>
```

The <a> tag is used to create an anchor to link from, the href attribute is used to address the document to link to, and the words between the open and close of the anchor tag will be displayed as a hyperlink.

This anchor defines a link to s:

```
<a href="http://www.w3schools.com/">Visit W3Schools!</a>
```

1.8.2 The Target Attribute

With the target attribute, you can define where the linked document will be opened.

The line below will open the document in a new browser window:

```
<a href="http://www.w3schools.com/"  
target="_blank">Visit W3Schools!</a>
```

1.8.3 The Anchor Tag and the Name Attribute

The name attribute is used to create a named anchor. When using named anchors we can create links that can jump directly into a specific section on a page, instead of letting the user scroll around to find what he/she is looking for.

Below is the syntax of a named anchor:

```
<a name="label">Text to be displayed</a>
```

The name attribute is used to create a named anchor. The name of the anchor can be any text you care to use.

The line below defines a named anchor:

```
<a name="tips">Useful Tips Section</a>
```

You should notice that a named anchor is not displayed in a special way.

To link directly to the "tips" section, add a # sign and the name of the anchor to the end of a URL, like this:

```
<a href="http://www.w3schools.com/html_links.asp#tips">  
Jump to the Useful Tips Section</a>
```

A hyperlink to the Useful Tips Section from WITHIN the file "html_links.asp" will look like this:

```
<a href="#tips">Jump to the Useful Tips Section</a>
```

1.9 Preformatted Text

The PRE element is used to enclose text to be displayed with a fixed width typewriter-like font. This is useful for presenting text that has been formatted for a teletype, or for a fixed width character display. It is also commonly used to present typed computer code examples.

In particular, as HTML 2.0 did not contain a TABLE element, the PRE element was the only way in which structured tables could be properly displayed on most current browsers.

HTML now supports tables, and almost all browsers now support Tables. PRE takes one optional attribute: WIDTH. This *supposedly* specifies the maximum number of characters that can be displayed on a single line. However, most browsers ignore this attribute completely. It is likely that the WIDTH attribute will be dropped in subsequent versions of HTML.

You *can* include character emphasis elements (EM, STRONG, B, I, etc.) as well as hypertext anchors within a PRE.

Things to Avoid

You should NOT use elements that define paragraph formatting within the PRE element. This means you should not use <P>, <ADDRESS>, <Hn> and so on. You should avoid the use of tab characters -- use single blank characters to space text apart.

Things That are OK

You can use Anchor. A typed carriage return will cause a new line in the presented text.

Example of PRE

The following text:

```
<PRE WIDTH=30>
```

```
Hi there. This is a rather <EM>stupid</EM> example.
```

```
Lord only knows what I was thinking at the time. I
```

```
suppose I should get more sleep when I do this stuff. But
```

```
why should I try and write Shakespearean sonnets when it is
```

```
just a stupid example of the <A HREF="pre.html">PRE</A> element!
```

```
</PRE>
```

1.10 Tables

Tables are defined with the `<table>` tag. A table is divided into rows (with the `<tr>` tag), and each row is divided into data cells (with the `<td>` tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

```
<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

1.10.1 Tables and the Border Attribute

If you do not specify a border attribute the table will be displayed without any borders. Sometimes this can be useful, but most of the time, you want the borders to show.

To display a table with borders, you will have to use the border attribute:

```
<table border="1">
<tr>
<td>Row 1, cell 1</td>
<td>Row 1, cell 2</td>
</tr>
</table>
```

1.10.2 Headings in a Table

Headings in a table are defined with the `<th>` tag.

```
<table border="1">
<tr>
<th>Heading</th>
<th>Another Heading</th>
</tr>
```

```

<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>

```

How it looks in a browser:

Heading	Another Heading
row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

1.10.3 Empty Cells in a Table

Table cells with no content are not displayed very well in most browsers.

```

<table border="1">
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td></td>
</tr>
</table>

```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

Note that the borders around the empty table cell are missing (NB! Mozilla Firefox displays the border).

To avoid this, add a non-breaking space () to empty data cells, to make the borders visible:

```

<table border="1">
<tr>

```

```

<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>&nbsp;</td>
</tr>
</table>

```

How it looks in a browser:

row 1, cell 1	row 1, cell 2
row 2, cell 1	

1.10.4 Table Tags

Tag	Description
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table
<thead>	Defines a table head
<tbody>	Defines a table body
<tfoot>	Defines a table footer

1.11 HTML Forms and Input

HTML Forms are used to select different kinds of user input. form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the `<form>` tag.

```
<form>
  <input>
  <input>
</form>
```

1.11.1 Input

The most used form tag is the `<input>` tag. The type of input is specified with the `type` attribute. The most commonly used input types are explained below.

1.11.2 Text Fields

Text fields are used when you want the user to type letters, numbers, etc. in a form.

```
<form>
First name:
<input type="text" name="firstname">
<br>
Last name:
<input type="text" name="lastname">
</form>
```

1.12 What do FORMS do?

The FORM element allows you to create a fill-out form: the user reading the HTML document will see the FORM elements as user input elements -- he or she can then type information into the fields or select from buttons and pull-down menus to input their data. When the user *submits* the FORM, the data are encoded and transmitted to a server, where it must be interpreted and processed by a program. In general, this is done by so-called CGI, or *Common Gateway Interface* programs, residing on an HTTP server.

An example (if rather idiotic) FORM is:

```
<FORM ACTION="http://bla.bla.edu/cgi-bin/some-prgm" METHOD=POST>
```

```
First entry field: <INPUT NAME="entry1"> <BR>
```

Second entry field: `<INPUT NAME="entry2" VALUE="bloop">` `
`

Third entry field: `<INPUT NAME="entry3">` ---

Select Option: `<SELECT NAME="entry4">`

`<OPTION VALUE="no1">`Frogs

`<OPTION VALUE="no2">`Peaches

`<OPTION VALUE="no3">`Cream

`<OPTION VALUE="no4">`Newts

`</SELECT>`

To submit the query, press:

`<INPUT TYPE="submit" VALUE="Submit Query">`. `<P>`

`</FORM>`.

CHAPTER TWO

ACTIVE SERVER PAGES

2.1 The need for ASP

Microsoft's Active Server Pages (ASP) with IIS 3.0 offers the web developer flexible, easy to use, scalable methods to interact with ODBC compliant databases for an Internet site or Intranet application. In this article the basic methods that are needed to interact with a database are illustrated - namely, adding, editing and deleting records.

Using ASP highly interactive pages can be developed independent of the type of browser that will be used to access these pages - from Lynx to Internet Explorer 3.0. ASP encompasses the capabilities of both JavaScript and VBScript with the added bonus that components can be easily added to extend the Internet/intranet application. Using ASP as part of your development not only means that you can initially develop in Microsoft's Access and scale up to a Microsoft SQL Server 6.5 database; but that you can access other vendor databases that are ODBC compliant. Its faster than using Visual Basic and the WinCGI interface - it will be interesting to compare performance with IDC and the use of an ISAPI filter to access a ODBC database. Needless to say, anyone who likes programming in Visual Basic is going to have a ball-using ASP.

In addition using Chili! ASP the functional equivalent of Microsoft's Active Server engine can be used on a range of NT based Web servers, including Netscape, Lotus, O'Reilly's, and some UNIX servers.

On the other hand ASP lacks the platform portability that PERL (Note that with advent of Chili! ASP its not true anymore), enjoys along with vast resources available to PERL programmers on the Internet but its is much easier to learn and develop in. When this the exception handling in VBScript leaves a bit to be desired - which would be critical if say there was an error inserting data into a database. I did not use the debugger in the development of the code and found that most of the run time errors were due to the fact

that I had variables spelled wrongly or I did not include the "=" sign as part of a variable when it was embedded in HTML.

The code is to use as a reference example, not a robust application. Conditions such as trying to delete or edit records when there are no records in the database have not been dealt with. The code was developed on Windows NT 4.0, with MS Access 7 as the database. You will need the 32 Bit ODBC Drivers for Microsoft Access 7.

To illustrate how you can put ASP to work on your web pages I am going to show you how to use ASP to interact with a database that contains user information. The example covers the basic methods that would be needed by anyone working with a database. You will be able to add, edit and delete entries into this database.

Why bother with ASP at all, when HTML can serve your needs? If you want to display information, all you have to do is fire up your favorite text editor, Type in a few HTML tags, and save it as an HTML file. Bingo, you're done! But wait – what if you want to display information *that changes*? Supposing You're writing a page that provides constantly changing information to your Visitors, for example, weather reports, stock quotes, a list of your girlfriends, Etc, HTML can no longer keep up with the pace. What you need is a system That can present dynamic information. And ASP fits the bill perfectly.

2.2 What is ASP?

In the language of Microsoft, Active Server Pages is an open, compile-free application environment in which you can combine HTML, scripts, and reusable ActiveX server Components to create dynamic and powerful Web-based business solutions. Active Server Pages enables server side scripting for IIS with native support For both VBScript and JScript.

Active Server Pages (ASPs) are Web pages that contain *server-side scripts* in addition to the usual mixture of text and HTML tags. Server-side scripts are special commands you put in Web pages that are processed before the pages are sent from the server to the web-

browser of someone who's visiting your website. When you type a URL in the Address box or click a link on a web page, you're asking a web-server on a computer somewhere to send a file to the web-browser (also called a "client") on your computer. If that file is a normal HTML file, it looks the same when your web-browser receives it as it did before the server sent it. After receiving the file, your web-browser displays its contents as a combination of text, images, and sounds.

In the case of an Active Server Page, the process is similar, except there's an extra processing step that takes place just before the server sends the file.

2.3 Active Server Pages 3.0

Before the server sends the Active Server Page to the browser, it *runs* all server-side scripts contained in the page. Some of these scripts display the current date, time, and other information. Others process information the user has just typed into a form, such as a page in the website's guestbook. And you can write your own code to put in whatever dynamic information you want.

To distinguish Active Server Pages from normal HTML pages, Active Server Pages are given the ".asp" extension.

2.3.1 What Can We Do with Active Server Pages?

There are many things you can do with Active Server Pages.

- You can display date, time, and other information in different ways.
- You can make a survey form and ask people who visit your site to fill it
- Out, send emails, save the information to a file, etc
- You can have a database, which people can access via the web.
- People can get information from the database as well as update or insert information into it.
- You can password-protect certain sections of your site, and make sure that only authorized users can see that information.
- The possibilities are virtually endless. Most of that you see on web pages nowadays can be easily done using ASP.

2.3.2 Server-Side Scripts

Server-side scripts typically start with `<%` and end with `%>`. The `<%` is called an *opening tag*, and the `%>` is called a *closing tag*. In between these tags are the server-side scripts. You can insert server-side scripts anywhere in your webpage - even inside HTML tags.

2.4 Running ASP

Since the server must do additional processing on the ASP scripts, it must have the ability to do so. The only servers which support this facility are Microsoft Internet Information Services & Microsoft Personal Web Server.

Let us look at both in detail, so that you can decide which one is most suitable for you.

2.4.1 Internet Information Services

This is Microsoft's web server designed for the Windows NT platform. It can only run on Microsoft Windows NT 4.0, Windows 2000 Professional, & Windows 2000 Server. The current version is 5.0, and it ships as a part of the Windows 2000 operating system.

2.4.2 Personal Web Server

This is a stripped-down version of IIS and supports most of the features of ASP. It can run on all Windows platforms, including Windows 95, Windows 98 & Windows Me. Typically, ASP developers use PWS to develop their sites on their own machines and later upload their files to a server running IIS. If you are running Windows 9x or me, your only option is to use Personal Web Server 4.0.

2.5 Beginning ASP

Here a few quick tips before you begin your ASP session! Unlike normal HTML pages, you cannot view Active Server Pages without running a web-server. To test your own pages, you should save your pages in a directory mapped as a virtual directory, and then use your web-browser to view the page.

2.5.1 Steps for Installation

- From the CD, run the SETUP.EXE program for starting the web-server installation.
- After the installation is complete, go to Start > Programs > Microsoft PWS > Personal Web Manager. And click the “Start” button under Publishing.
- Now your web-server is up & running.

2.5.2 Creating Virtual Directories

After you have installed the web-server, you can create virtual directories as follows:

- Right-Click on the folder that you wish to add as a virtual directory.
- Select “Properties” from the context-menu.
- In the second tab titled “Web Sharing,” click “Share this folder,” then “Add Alias”. (If you do not see these options enabled, your web-server is not properly running. Please see the steps above under “Installation.”)

2.6 Accessing your webpage

Now that your server is completely configured and ready to use. Start your web-browser, and enter the following address into the address-bar. `http://localhost/`
You should see a page come up that tells you more about Microsoft IIS (or PWS, as the case may be)

2.6.1 Local host

Let us first see, what we mean by a hostname. Whenever you connect to a remote computer using its URL, you are in effect calling it by its hostname.
For example, when you type in `http://www.google.com/` you are really asking the network to connect to a computer named `www.google.com`. It is called the “hostname” of that computer.
`localhost` is a special hostname. It always references your own machine. So what you just did, was to try to access a webpage on your own machine (which is what you wanted to

do anyway.) For testing all your pages, you will need to use localhost as the hostname. By the way, there is also a special IP address associated with localhost, that is 127.0.0.1 so you could as well have typed:

```
http://127.0.0.1/
```

And would have received the same page.

To access pages in a virtual directory called my scripts for example, you should type in:

```
http://localhost/myscripts/
```

In the address bar. I hope the concept is now clear ...

2.7 The Database connectivity

So lets start with the database - I used MS Access to develop the database.

2.7.1 DSN

Once you have designed you database the next step will be to create a DSN entry, serDB1. To do this:

- Click on your "Start" Button, and go to Control Panel under Settings.
- Click on "32 ODBC", select "System DSN"
- Click "Add" to add a DSN entry, and then on "Microsoft Access Drive". If "Microsoft Access Driver" does not appear on the list, you possibly have not installed Microsoft Access 7's, 32 bit ODBC drivers.

2.7.2 Connecting to the Database

So far we have developed a basic database and added a DSN entry in order that the database is accessed using ODBC - nothing to really write home about. ASP offers two methods to access the database. In the first each access to the database would have first connect to the database; once the connection has been established SQL statements can be seed to manipulate data; once completed, all related objects are closed. There are a number of illustrations using this technique in the samples provided with the Active Server Pages. The snippet of code illustrates a connection to a database with "ADO Samples" as the SN, obtaining a record set based on a SQL query. Once the script has done with the data, the record set and the connection to the database are closed.

The first post-startup request is made to the web server for any *.asp file in an application causes the Global.asa to be read. So the moment a request is made to any *.asp in the directory in which the intranet application is stored a connection is established with the DSN UserDB1. Following that the default document, in this case default.asp is recessed.

In ASP based applications the programming logic, variables and HTML can be maintained in a single file. Commonly used functions across an ASP application can be in one file that is included in different pages using the "include" statement. With regards to the logic of the example I have used a simple state space model to determine the state of the ASP page - i.e. is an addition, deletion or update taking place or not. The information of the current state of the page is dictated by the contents of the form element named "Action". The value the element "Action" is obtained from the form in VBScripts with the statement in.

For developers familiar with Microsoft's Internet related products ASP will possibly be the way to go to develop intranet applications - especially if you are a Visual Basic Programmer. In a couple of months visual tools will be available that will give ASP a more robust development environment.

2.8 Summary

In this chapter the role of using ASP for web development of databases is described and the use of ASP for connecting, manipulating and managing records through the Internet is shown.

CHAPTER THREE VISUAL BASIC SCRIPT

3.1 Overview

This chapter deals with VBScript. VBScript is derived from Visual Basic, which has origins in GW-Basic that was shipped with all early versions of MS-DOS. GW-Basic evolved from the original Dartmouth BASIC (Beginners All-purpose Symbolic Instruction Code) developed in the 1960's

3.2 What is VBScript?

- VBScript is a scripting language
- A scripting language is a lightweight programming language
- VBScript is a light version of Microsoft's programming language Visual Basic

3.3 How does it Work?

When a VBScript is inserted into a HTML document, the Internet browser will read the HTML and interpret the VBScript. The VBScript can be executed immediately, or at a later event.

3.4 How to Put VBScript Code in an HTML Document

```
<html>
<head>
</head>
<body>
<script type="text/vbscript">
document.write("Hello from VBScript!")
</script>
</body>
</html>
```

And it produces this output:

```
Hello from VBScript!
```

To insert a script in an HTML document, use the `<script>` tag. Use the `type` attribute to define the scripting language.

```
<script type="text/vbscript">
```

Then comes the VBScript: The command for writing some text on a page is **document.write**:

```
document.write("Hello from VBScript!")
```

The script ends:

```
</script>
```

3.5 How to Handle Older Browsers

Older browsers that do not support scripts will display the script as page content. To prevent them from doing this, you can use the HTML comment tag:

```
<script type="text/vbscript">  
<!--  
  some statements  
-->  
</script>
```

3.6 Where to Put the VBScript

Scripts in a page will be executed immediately while the page loads into the browser. This is not always what we want. Sometimes we want to execute a script when a page loads, other times when a user triggers an event.

3.6.1 Scripts in the head section

Scripts to be executed when they are called or when an event is triggered go in the head section. When you place a script in the head section you will assure that the script is loaded before anyone uses it:

```
<html>
<head>
<script type="text/vbscript">
  some statements
</script>
</head>
```

3.6.2 Scripts in the body section

Scripts to be executed when the page loads go in the body section. When you place a script in the body section it generates the content of the page:

```
<html>
<head>
</head>
<body>
<script type="text/vbscript">
  some statements
</script>
</body>
```

3.6.3 Scripts in both the body and the head section

You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
<script type="text/vbscript">
  some statements
</script>
</head>
<body>
<script type="text/vbscript">
```

```
some statements
</script>
</body>
```

3.7 VBScript Conditional Statements

3.7.1 Conditional Statements

Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In VBScript we have three conditional statements:

- **If statement** - use this statement if you want to execute a set of code when a condition is true
- **If...then...else statement** - use this statement if you want to select one of two sets of lines to execute
- **If...then...else if statement** - use this statement if you want to select one of many sets of lines to execute
- **Select case statement** - use this statement if you want to select one of many sets of lines to execute

3.7.1.1 if.... Then...Else

You should use the If...Then...Else statement if you want to

- Execute some code if a condition is true
- Select one of two blocks of code to execute

If you want to execute only **one** statement when a condition is true, you can write the code on one line:

```
if i=10 Then msgbox "Hello"
```

There is no. Else. In this syntax. You just tell the code to perform **one action** if the condition is true (in this case if i=10).

If you want to execute **more than one** statement when a condition is true, you must put each statement on separate lines and end the statement with the keyword "End If":

```
if i=10 Then
  msgbox "Hello"
  i = i+1
end If
```

There is no ..else.. in this syntax either. You just tell the code to perform **multiple actions** if the condition is true.

If you want to execute a statement if a condition is true and execute another statement if the condition is not true, you must add the "Else" keyword:

```
if i=10 then
  msgbox "Hello"
else
  msgbox "Goodbye"
end If
```

The first block of code will be executed if the condition is true, and the other block will be executed otherwise (if i is not equal to 10).

3.7.1.2 if.... Then...Else if

You can use the if...then...else if statement if you want to select one of many blocks of code to execute:

```
if payment="Cash" then
  msgbox "You are going to pay cash!"
elseif payment="Visa" then
  msgbox "You are going to pay with visa."
elseif payment="AmEx" then
  msgbox "You are going to pay with American Express."
else
  msgbox "Unknown method of payment."
end If
```

3.7.1.3 Select Case

You can also use the SELECT statement if you want to select one of many blocks of code to execute:

```
select case payment
case "Cash"
    msgbox "You are going to pay cash"
case "Visa"
    msgbox "You are going to pay with visa"
case "AmEx"
    msgbox "You are going to pay with American Express"
case Else
    msgbox "Unknown method of payment"
end select
```

This is how it works: First we have a single expression (most often a variable) that is evaluated once. The value of the expression is then compared with the values for each Case in the structure. If there is a match, the block of code associated with that Case is executed.

3.8 VB Script and ASP

Define at the top of the page to be saved as .asp the default script language as follows:

```
<%@ LANGUAGE = "VBScript"%>
```

Active Server Pages recognize script commands entered between the pair <% %> . In this format it handles code, but not necessarily it will produce output, you need to explicitly use output commands in order to produce output. When you use the pair <%= %> any variable, function will produce automatically output. Here are two simple examples displaying Hello! in the browser:

```
<% Response.Write("Hello!") %> ----> explicit output
```

```
<%= "Hello!" %> ----> implicit output
```

We will see and discuss four examples in order to understand how VBScript is used within the ASP tags.

example is our "old friend" Hello World! script:

hello.asp

```
<%@ LANGUAGE="VBSCRIPT"%>
<html>
<head>
<title>Hello World</title> </head>

<body bgcolor="White">
<div align="center"><h1>Hello World</h1></div>
<br><p><hr><p>
<%Response.Write("Hello World!")%>

</body>
</html>
```

As you can see we used the explicit ASP tags to have Hello World! Sent to the browser and everything else is HTML.

3.9 Summary

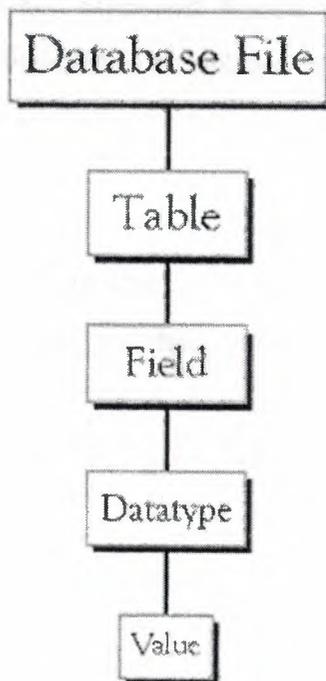
This chapter has taught you how to add VBScript to your HTML pages, to make your web site more dynamic and interactive. You have learned how to create variables and functions, and how to make different scripts run in response to different scenarios.

CHAPTER FOUR

DATABASE USING MICROSOFT ACCESS

4.1 Introduction

Microsoft Access is a powerful program to create and manage your databases. It has many built in features to assist you in constructing and viewing your information. Access is much more involved and is a more genuine database application than other programs such as Microsoft Works. First of all you need to understand how Microsoft Access breaks down a database. Some keywords involved in this process are: Database File, Table, Record, Field, and Data-type. Here is the Hierarchy that Microsoft Access uses in breaking down a database.



Database File: This is main file that encompasses the entire database and that is saved to your hard-drive or floppy disk.
Example) StudentDatabase.mdb

Table: A table is a collection of data about a specific topic. There can be multiple tables in a database.
Example #1) Students
Example #2) Teachers

Field: Fields are the different categories within a Table. Tables usually contain multiple fields.
Example #1) Student LastName
Example #2) Student FirstName

Datatypes: Datatypes are the properties of each field. A field only has 1 datatype.
FieldName) Student LastName
Datatype) Text

Table 4.1 Database Design In Access

4.2 Create a database using the Database Wizard

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click Access Database Wizards, pages, and projects and then click ok. If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click New Database on the toolbar.
2. On the Databases tab, double-click the icon for the kind of database you want to create.
3. Specify a name and location for the database.
4. Click Create to start defining your new database

4.3 Create a database without using the Database Wizard

1. When Microsoft Access first starts up, a dialog box is automatically displayed with options to create a new database or open an existing one. If this dialog box is displayed, click Blank Access Database, and then click OK. If you have already opened a database or closed the dialog box that displays when Microsoft Access starts up, click New Database on the toolbar, and then double-click the Blank Database icon on the General tab.
2. Specify a name and location for the database and click Create. (Below is the screen that shows up following this step)

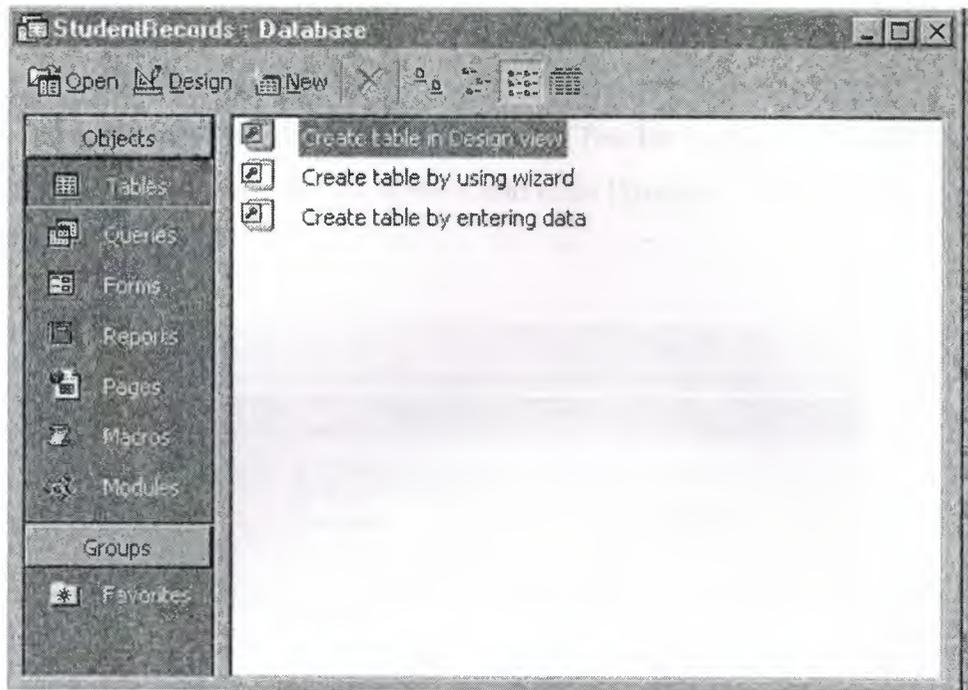


Figure 4.1 Creating Database in Access

4.3.1 Relationships

3. After you've set up multiple tables in your Microsoft Access database, you need a way of telling Access how to bring that information back together again. The first step in this process is to define relationships between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once. A relationship works by matching data in key fields - usually a field with the same name in both tables. In most cases, these matching fields are the primary key from one table, which provides a unique identifier for each record, and a foreign key in the other table. For example, teachers can be associated with the students they're responsible for by creating a relationship between the teacher's table and the student's table using the TeacherID fields. Having met the criteria above, follow these steps for creating relationships between tables.
4. In the database window view, at the top, click on Tools ---> Relationships

5. Select the Tables you want to link together, by clicking on them and selecting the Add Button
6. Drag the primary key of the Parent table (Teacher in this case), and drop it into the same field in the Child table (Student in this case.)

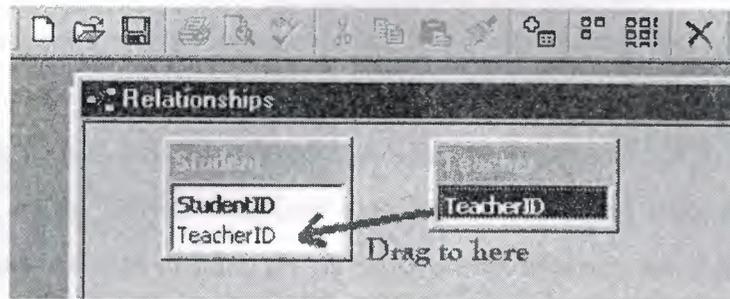


Figure 4.2 Relationships in Access

7. Select Enforce Referential Integrity

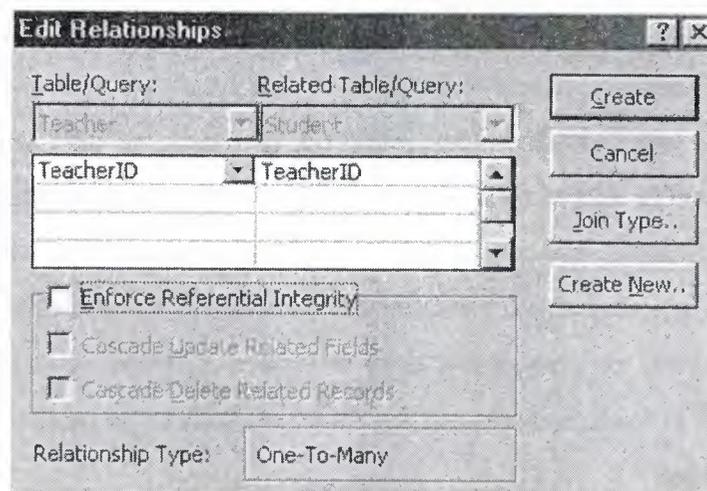


Figure 4.3 Edit Relationships in Access

8. When the Cascade Update Related Fields check box is set, changing a primary key value in the primary table automatically updates the matching value in all related records.

9. When the Cascade Delete Related Records check box is set, deleting a record in the primary table deletes any related records in the related table
10. Click Create and Save the Relationship
11. Microsoft Access Database is a relational desktop database system from Microsoft, packaged with Microsoft Office Professional, which combines the Jet relational database engine with a graphical interface. The development environment provides productivity-enhancing features. It can use data stored in Microsoft Access/Jet, SQL Server, Oracle, or any ODBC-compliant data container. Skilled users and data architects use it to develop powerful, complex applications. Relatively unskilled programmers and non-programmer "power users" can use it to build simple applications without having to deal with features they don't understand. It supports substantial object-oriented (OO) techniques but falls short of being a fully OO development tool.

Figure 4.4 Project's Database

Microsoft Access is widely used by small businesses and hobby programmers to create ad hoc customized systems for handling small tasks. Its ease of use and powerful design tools give the non-professional programmer a lot of power for little effort. However, this ease of use can be misleading. This sort of power user is often an office worker with little or no training in application or data design. Because Ms Access it possible even for such developers to create usable systems, many are misled into thinking that the tool itself is limited to such applications.

4.4 Project's DATABASE

4.4.1 Data Base Design

Project 's database is muath.mdb that contains nine table belongs to the website. The database tables are in figure 4.4 below

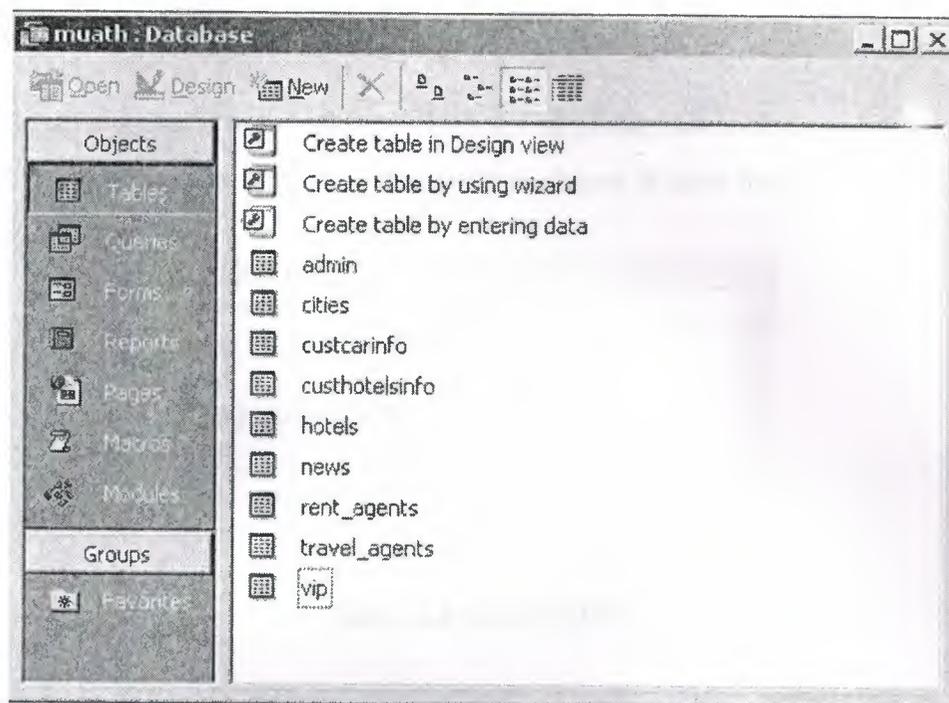


Figure4.4 Project's database

4.4.2 Database Tables

The tables that were created are nine table , admin, cities, custcarinfo, cushotlesinfo, hotels, news, rent_agent, travel_agent vip tables.

4.4.2.1 Admin's tables

The admin table contains us_name username and pass password columns belong to username and password respectively as shown in figure 4.5 below

admin : Table	
us_name	pass
abac	ayman123

Table 4.4 Administrator Table

4.4.2.2 Cities' Table

This table contains three column which are id, photo, desc belong to city id, and its photo and description respectively as shown in table below

descr	photo	id	
uraselim	knknk	1	
one of the most	C:\netpub\www	22	
safad	C:\netpub\www	66	
rkjnjknknknknk	c:\kmmk\	kn	
ereh	C:\netpub\www	sds	
rafah	C:\netpub\www	sdfs	

Table 4.4 Cities' table

4.4.2.3 Hotels' Table

This table contains four column which are id, name, tel, address belong to hotel's id, hotel's name and hotel's tel and address. And it shown below

address	tel	name	id	
gaza	21454154	radison sas	4	

Table 4.4 Hotels' table

4.4.2.4 News's Table

This table contains four column belong to news's id, photo, title and details.

The table is shown below

details	title	photo	id	
jurascalim is libe	jurascalim liberat	jklkkkkh		
bjkbfskjbkbjsjjs	mknij	b:\kkl\	2323	
dfgvf hbhbh bh	gaza	jjjjjjjj	44	

Table 4.5 Palestine news's table

4.4.2.5 World of News's Table

This table contain six column belong to world news's Id, add_date, subject, body, picture and type. The table is shown below

ID	ADD_DATE	SUBJECT	BODEY	PICTURE	TYPE
1	5/5/2055	my project	murad nayef alz	add.asp	
2	1/1/2000	sub1	BBBBBBBBBB		1
3	1/1/2000	sub3	111111111111		1
4	1/1/2000	sub4	as\znx assssss		1
6	2/2/2005	TEST	AAAA		2
8	1/1/2007	NEW CAR	BMW Z5		5
9	1/1/2007	NEW CAR	BMW Z5 BMW		5
10	1/1/2007	NEW CAR	imad ahmed dal		5
12	12/12/2012	12	12	1.jpg	1
13	8/5/1980	NIDAL	نض	world.gif	1
16	5/5/2005	phylistin	fertryrt	plan.jpg	5
19	5/5/2005	alla	trytghfghfghfghfj	hp5-28-05zz.jpg	4
66	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
70	2/2/2005	kjhk	khkj	top.sat2.iraq.ap	2
85	12/25/1983	hi....	horydty	plan.jpg	1
88	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
99	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
101	5/28/2005	murad	safdddddffss	cars.jpg	2
*	0				0

Table 4.6 World's Table

4.4.2.5 Famous Persons' Table

This table contains four columns that are person's id, photo and details, and his citiesid. And it's shown below

citiesid	photo	details	id
nablus	kkkjk	ahmad yassen	1
beit lba	C:\netpub\www	rantesi	2
ramallah	kjljk,l	gghjjj	56
burqah	jhjjn	one of the pales	66

Table 4.7 Vip's Table

4.4.2.5 Customers Information's Tables

In these tables the customer has to insert information into tables (personal details). And there are two tables:

4.4.2.5.1 Customers Hotels Information's Table

This table contains four column belong to customer details and they are first, last, phone and email. And it's shown below

email	phone	last	first
mos@	kkkjk	ahmad yassen	1
nos@	C:\netpub\www	rantesi	2
pos@	kjljk,l	gghjjj	56
tos@	jhjjn	one of the pale	66

Table 4.8 Customers Hotel Information's Table

4.4.2.5.2 Customers Rent Cars Information's Table

This table contains four column belong to customer details and they are first, last, phone and email. And it's shown below

email	phone	last	first
mu@	kkkjk	ahmad yass	1
cvdfgd@c	C:\netpub\www	rantesi	2
cdcdf@cd	kjlk,l	gghjii	56
	jhjijn	one of the	66

Table 4.9 Customers Rent Cars Information's Table

CHAPTE FIVE

WEB SITE DESIGN

5.1 Introduction

In this chapter I am going to declare the practical part of project (Palestine Online). The practical part contains almost 35 pages both html and asp besides to more than 10 database tables which belongs to the Palestine Online WebPages that could be updateable from time to another, therefore the changeable daily news in Palestine require such a specific area to be followed sequentially, so that this page was designed.

5.2 System Design of The Web Site

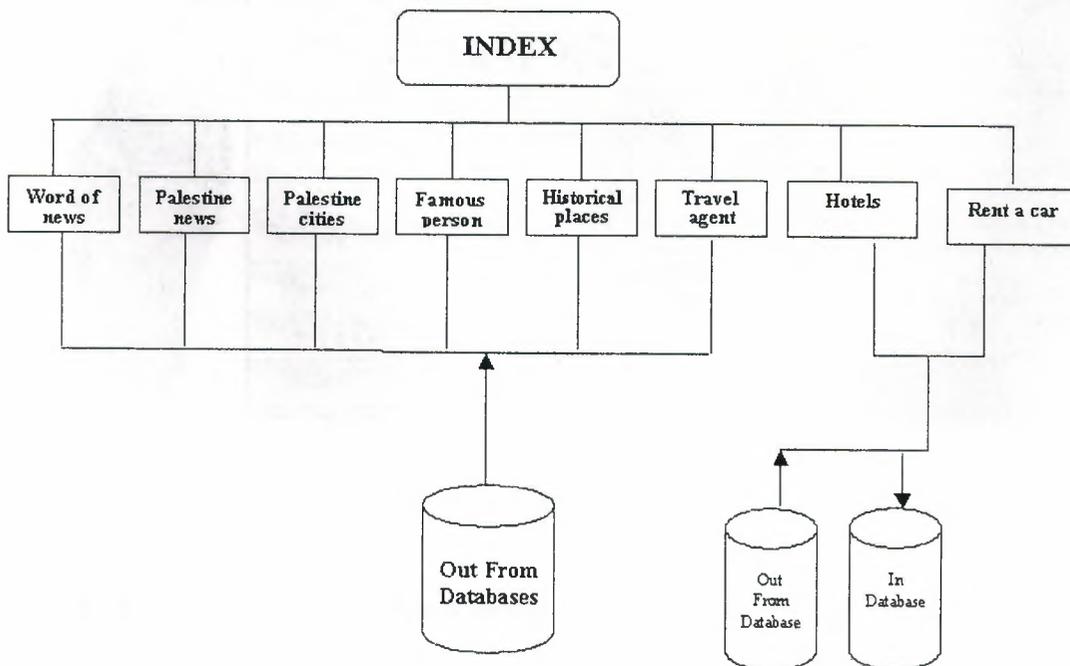


Figure 5.1 System Block Diagram

5.3 The Content of Project

The project contain ten main website pages besides the home page that contains 8 main links belongs to administrator, world of news, Palestine news, Palestinian Cities, Famous persons, historical places, Alaqsa mosque, contact us, about us, and three links as a guide for the Palestine's visitors, hotels, travel agent and rent a car.

5.3.1 First page (home page)

The first page is an html page that contains links to other pages and some picture. The page is below in figure 5.2.



Figure 5.2 Home Page

5.3.2 News.Asp

In this page VBScript and asp language were used to show the last news in Palestine. And below is the code that is used.

```
@ %>language="vbscript<%"  
>html<  
>head<<title>news</title></head<
```

```

>body<
>h1 align="center">News List</h1<
>table align="center" width=750 border=1<
%>
dim connection
dim sSQL,sConnString

sSQL="select * from news "

sConnString="PROVIDER=Microsoft.Jet.OLEDB.4.0 _ & ";
"Data Source=" & Server.MapPath("muath.mdb ("

Set connection = Server.CreateObject("ADODB.Connection("
Set recordset = Server.CreateObject("ADODB.Recordset ("

connection.Open(sConnString(

recordset.Open sSQL,connection

while Not recordset.EOF<%

>tr> <td align="center"><img src=<%response.write
recordset.Fields("photo")%>><br<
>    b><%response.write recordset.Fields("title")%></b><br<
%>    response.write recordset.Fields("details")%></td<

%>
recordset.MoveNext

wend

<%

/>table<
%>
recordset.Close
Set recordset = Nothing
Connection.Close
Set Connection = Nothing
<%
/>body<
/>html>

```

5.3.3 Cities. Asp

This page displays the major cities in Palestine and its photos, also here the VBScript was used and it is related to database to be able for change. The code is shown below

```
@ %>language="vbscript<%"  
>html<  
>head<<title>Cities</title></head<  
>body<  
>h1 align="center">Palestine Cities</h1<  
>table align="center" width=750 border=1<  
  
%>  
  
dim connection  
dim sSQL,sConnString  
  
sSQL="select * from cities "  
  
sConnString="PROVIDER=Microsoft.Jet.OLEDB.4.0 _ & ";  
"Data Source=" & Server.MapPath("muath.mdb ("  
  
Set connection = Server.CreateObject("ADODB.Connection(")  
Set recordset = Server.CreateObject("ADODB.Recordset ("  
  
connection.Open(sConnString(  
  
recordset.Open sSQL,connection  
  
while Not recordset.EOF<%  
  
>tr> <td align="center"><img src=<%response.write  
recordset.Fields("photo")%>><br<  
  
%>   response.write recordset.Fields("descr")%></td<  
  
%>  
recordset.MoveNext  
  
wend  
  
<%  
  
>/table<  
%>  
  
recordset.Close
```

```

Set recordset = Nothing

Connection.Close
Set Connection = Nothing
<%
/>body<
/>html>

```

5.3.4 Famous Persons.asp

This page demonstrates the famous persons in Palestine and their photos and it's related in database with vip table. The page is shown below in figure 5.3.



Figure 5.3 Famous Persons Page

And asp codes are shown below

```

<% "language="vbscript @ %
<html>
<head><title>famous persons</title></head>
<body>
<h1 align="center">Famous list</h1>
<table align="center" width=750 border=1>
%>
dim connection
dim sSQL,sConnString

"sSQL="select * from vip

```

```
& ";sConnString="PROVIDER=Microsoft.Jet.OLEDB.4.0  
("Data Source=" & Server.MapPath("muath.mdb"  
  
("Set connection = Server.CreateObject("ADODB.Connection  
("Set recordset = Server.CreateObject("ADODB.Recordset  
  
(connection.Open(sConnString  
  
recordset.Open sSQL,connection  
  
<%while Not recordset.EOF  
tr> <td align="center"><img src=<%response.write >  
<recordset.Fields("photo")%>><br  
  
<response.write recordset.Fields("details")%></td%>  
  
%>  
recordset.MoveNext  
  
wend  
  
<%  
  
<table/>  
%>  
  
recordset.Close  
Set recordset = Nothing  
Connection.Close  
Set Connection = Nothing  
<%  
<body/>  
html/>
```

5.3.5 Rent Car.asp

The other main page is rent a car page that contains first name, second name, phone no and email address, the visitor has to fill them if he want to rent a car. The page is shown below in figure 5.4.



Figure 5.4 Rent a car page

5.3.6 Hotel.asp

The last main page is hotel page that the visitor can make his reservation for any hotel after he inserts his personal detail besides email address and phone. The page is shown below in figure 5.5.



Figure 5.5 Hotels page

5.4 Administrator Design

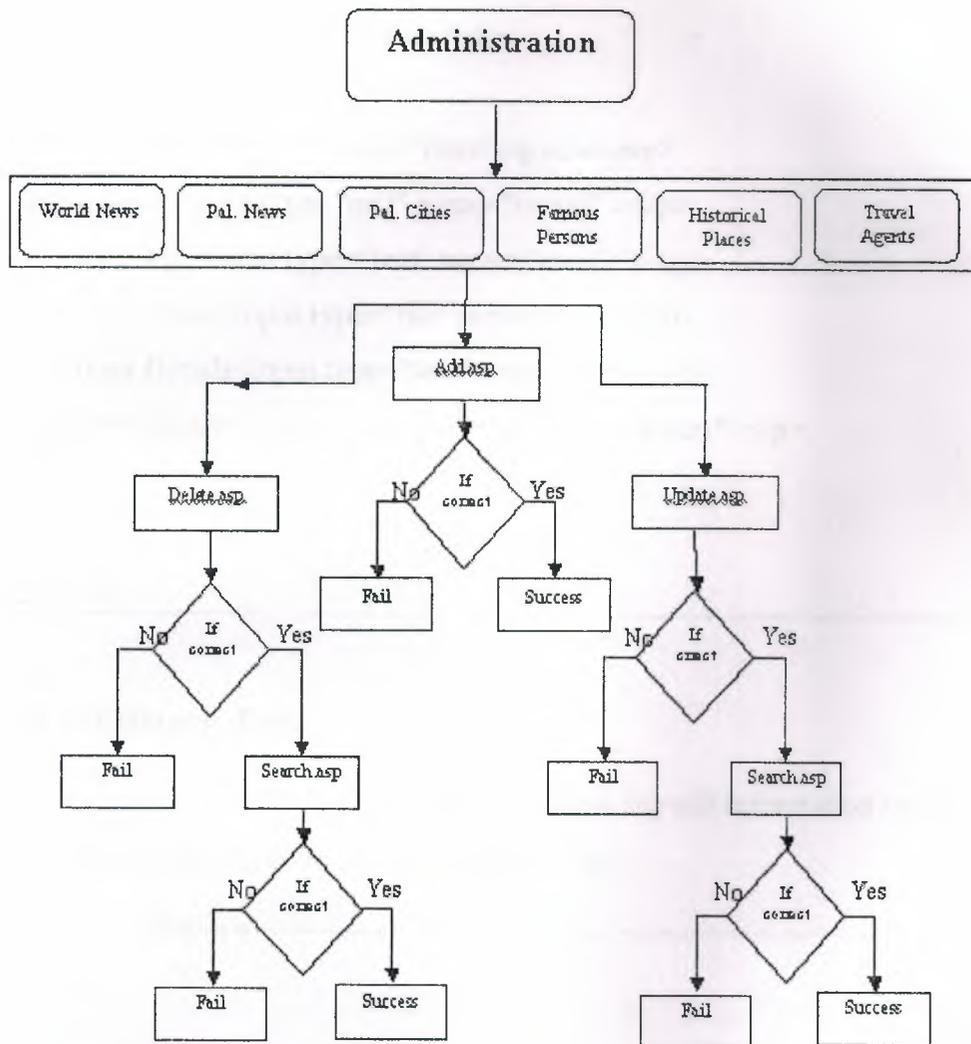


Figure 5.6 Administrator Block Diagram

The administrator page contains five main links belongs to world of news, Palestine news, famous persons, hotel and rent a car. The main important of administrator is to make an insert, update and delete for the pages

5.4.1 Insert.asp page

In this page the administrator can insert new information to database and then to be display to user. And asp code is shown below


```
</body/>
```

```
</html/>
```

5.4.3 Update.asp page

In this page the administrator can update the database tables and then to show the new information to the user. The code that used is shown in the below

```
<head>
```

```
<title>update agent</title>
```

```
</head>
```

```
<body>
```

```
<p align="left"><font size="5"><b>Update Agents</b></font></p>
```

```
<p align="center"></p>
```

```
<form method="post" action="updating agent.asp">
```

```
<p align="left"><b>ID<span lang="ar-sa">:<input type="text" name="id"
size="10"></span> <span lang="ar-sa">
```

```
</span></b></p>
```

```
<p align="left">&nbsp;&nbsp;&nbsp;<input type="Submit" value=" Search "></p>
```

```
</form>
```

```
</body>
```

```
</html>
```

CONCLUSIONS

The Internet web site has become one of the most important things in this life, the Internet has entered mostly all fields; politics, education, economy, health, industrial etc. So that the web page design has to be efficient and attractive, to be able to follow the changeable daily events over the world in different fields. And to display the events in the best way to simplify the job of user.

This project is a country website contains seven main fields (political news, public news, major cities, famous persons, rent cars and hotels); these pages are linked to database to be able for change.

The project contains administration page through which insert, update, and delete are possible for administrator.

In this project the Ms Access is used for database and it is linked with website pages using (ASP & ADO) technologies, and also the VBScript was used.

In future, web page design can be developed to be more useful in all field, and more attractive for the user.

REFERENCES

References from Internet

http://en.wikipedia.org/wiki/Main_Page

<http://www.w3schools.com/default.asp>

<http://htmldog.com/reference/htmltags/>

http://www.w3schools.com/vbscript/vbscript_intro.asp

<http://microsoft.com/office/access/default.htm>

References of Books

[1] *How to Do Everything with HTML* -- by James H. Pence.

[2] *Beginning Active Server Pages 3.0 (Programmer to Programmer)* -- by David Buser,

[3] *VBscript in a Nutshell*, by Paul Lomax, Matt Childs, and Ron Petrussha

[4] *A Practical Guide to Microsoft Active Server Pages 3.0*, By Manas Tungare