



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

**INTERACTIVE NEWS WEB PAGE DESIGN**

**Graduation Project  
COM-400**

**Student: Murad ALzu'bi (20011522)**

**Supervisor: DR. Adil Amirjanov**

*Lefkosa-2005*



## ACKNOWLEDGEMENT

*First, I would like to thank my supervisor*

*Assoc. Prof. Dr Adil Amirjanov for his invaluable advice*

*And for the generosity he exhibited with his time and effort over this project*

*Second, I thank my family especially parents and my fiancé, brothers for their continuous encouragement and support during preparation of this project.*

*Third, I would like to express my gratitude to Near East University*

*Finally, I would like to thank all my friends, especially Emad Aldhdoh, Mahmoud Alfourey, Yazan Alzu'bi and Khaled Alwaled for their support and their constant moral sustain.*

**ABSTRACT**

Nowadays the internet become one of the most important intermedia in the world, therefore it has a significant role in the news side.

According to the changeable events in news transmittance over the world, it has been choosen as quick intermedia. The internet and the world of news therefore in order to be able to have such an attractive for people over world for who are following the news website I used to design as fit as possible to satisfy the satisfactions of users so that the method of display the external design of news webpage should be familiar with the ease transmittance between the whole pages, so that it has to make sense that the designer knows how to coordinate the page thus the JSCRIBT gives ease for designer to design with high level of techniques, To be familiar with the designing fields I should have an enough knowledge about the following subjects HTML ODBC and ASP, whereas I must know ORCAL, SQL, ACCESS to deal with database. That is why I have decided to design this website for the field of news.

## TABLE OF CONTENTS

### ACKNOWLEDGMENT

### ABSTRACT

### TABLE OF CONTENTS

### INTRODUCTION

### CHAPTER ONE: HTML

1.1 Introduction .....	1
1.2 HTML code headers.....	2
1.3 Title.....	2
1.4 Web page body.....	2
1.5 Changing size and face of fonts.....	3
1.6 Size of font .....	3
1.7 Changing font colors.....	3
1.8 Combining Fonts attributes .....	4
1.9 Line Breaks, Paragraphs .....	4
1.10 Paragraph alignments in your web page .....	5
1.11 Images in your web page.....	5
1.12 Path of image files .....	6
1.13 Text links.....	6
1.14 Image links .....	7
1.15 Tables.....	8
1.15.1 Drawing a table.....	8
1.15.2 Specifying table sizes .....	9

### Chapter Two: JavaScript

2.1 Introduction.....	11
2.2 Functions.....	11
2.3 The Time-object .....	14
2.4 The Date-object .....	15



### **CHAPTER THREE: MICROSOFT ACCESS DATABASE**

1.1 Introduction .....	19
1.2 Objects .....	20
1.3 Tables .....	21
1.3.1 Data Types .....	22
1.4 Primary Key .....	23
1.5 Entering Data .....	23
1.6 Field .....	24
1.7 Relationships .....	24
1.8 Forms .....	25
1.9 Reports .....	25

### **CHAPTER FOUR: ASP AND ODBC**

4.1 Introduction ASP .....	26
4.2 What is ADO .....	26
4.3 Creating the Database .....	27
4.4 Creating an HTML Page to take User Input .....	28
4.5 Creating a Page to Select the Database Entry to Delete .....	29
4.6 Creating a Page to Select the Database Entry to Update .....	31

### **CHAPTER FIVE: THE APPLICATION OF MY PROJECT**

5.1 Introduction .....	34
5.2 The contents of my project .....	34
5.3 My Database .....	35
5.4 Administration Page .....	37
5.5 Add page .....	38
5.6 Remove and update pages .....	40
5.6.1 Remove page .....	40
5.6.2 Update page .....	44

### **CONCLUSION**

### **REFERENCES**

## **INTRODUCTION**

My project is about an interaction news website, which means that all pages are interacting with each other, and with other programs, thus all that occurs by using ASP system therefore this system creates same instruction of the WebPages units. in order to design my WebPages as a news pages I decided to be done with display the international daily news such as over whole world news , economic news, sport news, the last versions of technology side beside to cars race news, and according to be able to follow the modern news from time to another I hade to use database which would enable me to add and improve and delete the unbeliever news thus in order to link the database with news pages I hade to use ODBC system which throw it I could make my WebPages interactive during the database , this belongs the practical side of my project whereas the theoretical side of my project is talking about an interactive web site; actually news website, it contains five chapters.

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

Chapter 2 talks about one of the scripting languages, which is Jsript, I talked about its functions, objects.

Chapter 3 talks about Ms Access, how to create table, how to insert, delete, update, retrieve data using SQL statements.

Chapter 4 talks about ASP & ODBC technologies; how to connect databases to the internet and how to interact with users using theses technologies.

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

## CHAPTER 1 HTML

### 1.1 Introduction

HTML is inextricably linked with the development of the World Wide Web. Most of the amazing documents in the WWW are based around HTML. This is a code which describes the logical structure of documents. HTML defines the structure of a text document rather than the layout. For World of Today: here a main heading, there a sub heading, there bold text, there a list, or a line etc. Originally HTML was intended for the world-wide representation of scientific texts. Understandably so, seeing that scientists invented the whole thing. Since then new HTML versions have incorporated numerous little .tricks.. How you see these individual elements depends on your browser, which interprets the HTML code and reproduces the document accordingly.

The hottest browsers are *Opera*, *Netscape Communicator* and *Microsoft Internet Explorer*.

The latter is included in Windows 98. It seems currently evident that Microsoft has won the so-called .browser war.. The market share of the Internet Explorer 4 or 5 is double that of the Communicator. HTML is based on ASCII, American Standard Code for Information Interchange. This is a code string which enables text representation. More complex formatting such as bold print, underscoring or centring etc. cannot be represented in ASCII. But then there is HTML.

So called HTML commands, called tags, prescribe the text structure. These commands are place in <> brackets. There is usually one command to switch a characteristic on and another to turn it off. (A few commands are stand alone.)

For World Of Today before a word to be printed bold you find the following symbol: <B>, and after it this symbol : </B>. B for bold . The switch-off command is similar to the switch-on command, with the addition of a slash /. So the World Of Today for bold is written in HTML as follows: <B>bold word group</B> The browser turns this into a bold word group. In either case . it does not matter at all whether the tags are in lower or upper case letters !

So HTML files are just text files with the ending htm or html!



## 1.2 HTML code headers

Every html page must have a header. Header contains important information about the page.

Different tags are used for different sections of a header. Header of an html page is specified by

`<HEAD>` and `</HEAD>` tags.

`<HTML>`

`<HEAD>`

...

`</HEAD>`

`</HTML>`

We will enter header information between `<HEAD>` `</HEAD>` tags.

## 1.3 Title

One of the most important parts of a header is title. Title is the small text that will appear in title bar of viewer's browser. So html document will be as below:

`<HTML>`

`<HEAD>`

`<TITLE>Title of the page</TITLE>`

`</HEAD>`

`</HTML>`

## 1.4 Web page body

Now the web page needs a body in which will enter web page content. As you may guess we will use these tags:

`<BODY>` `</BODY>`

Body will come right after header end tag. So the web page will be something like this

Welcome to our homepage.



## 1.5 Changing size and face of fonts

We can change face and size of fonts using `<FONT>...</FONT>` tags. Also using this tag alone will not change the text. You need to use parameters for this tag. These parameters specify what kind of change you need in text font.

## 1.6 Size of font

To change size of font in a part of text, include it with a `<FONT>` tag as below:  
`<FONT SIZE=n>...</font>` n is size of font. Size of font must be a number between 1 and 7.  
If you insert some text without determining its size default size will be 3.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
<FONT SIZE=1>1This text is bold</FONT><br>
<FONT SIZE=2>2This text is bold</FONT><br>
<FONT SIZE=3>3This text is bold</FONT><br>
<FONT SIZE=4>4This text is bold</FONT><br>
<FONT SIZE=5>5This text is bold</FONT><br>
<FONT SIZE=6>6This text is bold</FONT><br>
<FONT SIZE=7>7This text is bold</FONT><br>
</BODY>
</HTML>
```

## 1.7 Changing font colors

In previous lesson you learned how to change web page background color. Here we will learn how to change text color. Look at this World Of Today:

```
<HTML>
<HEAD>
```

```
<TITLE>World Of Today</TITLE>
</HEAD>
<BODY>
<FONT COLOR="#FF0000">This text is in red color.</FONT><br>
<FONT COLOR="#00FF00">This text is in green color.</FONT><br>
<FONT COLOR="#0000FF">This text is in blue color.</FONT><br>
</BODY>
</HTML>
```

In above text different colors are used. You can change text color by changing color number. If you don't remember how to specify color numbers return to previous lesson and review section on "web page background color".

## 1.8 Combining Fonts attributes

We can obviously combine <FONT> tag parameters as we wish. In this way we will be able to have text with different colors, font faces and font sizes.

You can even use text styles with <FONT> tag by nesting style and font tags.

```
<B><I><FONT SIZE="5" FACE="IMPACT" COLOR="#00FF00">
```

How is this?

```
</FONT></I></B>
```

## 1.9 Line Breaks, Paragraphs

As we saw in previous lesson if we break lines in html code by simply inserting enter keys (new line characters), lines will not break in output result in browser. They will be printed in a single line in browser. We must use <BR> tag to do this as you used it in previous lesson.

You can also divide text using paragraphs. A paragraph starts on a new line with one blank line after previous line. Paragraph tag is <p> </p>

```
<p>First paragraph</p>
```

```
<p>Second paragraph</p>
```

You will nest other tags inside paragraph tag for fonts, styles and other tags that will be used inside a paragraph.

There is another option in forming text that is using `<PRE>` tag. Text between `<PRE>` `</PRE>` tags will be displayed exactly as it is typed in html source. Therefore you will not need `<BR>` tags to break lines. It is enough to enter text in separate lines with enter key (new line character) at their end.

### 1.10 Paragraph alignments in your web page

You can determine how a paragraph will be aligned in a web page. You can align text in left, right or center of a web page. To specify alignment for a paragraph you must use an extra parameter for your paragraph tag. You can use one of these combinations:

```
<P ALIGN="left"> </P>
```

```
<P ALIGN="center"> </P>
```

```
<P ALIGN="right"> </P>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>World Of Today</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<P ALIGN="left">You can align text in left.</P>
```

```
<P ALIGN="center">You can align text in center.</P>
```

```
<P ALIGN="right">You can align text in right.</P>
```

```
</BODY>
```

```
</HTML>
```

### 1.11 Images in your web page

To add an image in a web page. Tag that will be used for this purpose is `<IMG>` tag. Actually we will need parameters for this tag that specify image file location, file name and other optional parameters.

Look at this World Of Today:

### World Of Today 11.1

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY BACKGROUND="image1.gif">
<B>This is an image:</B><BR>
<IMG SRC="abanner.gif">
</BODY>
</HTML>
```

In this World Of Today it used both a background image and an image between texts. Also you may have noticed that this tag is also a single tag that does not need an ending tag.

If you want to show your image in a different size than its real size, you can specify its size as below.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY BACKGROUND="image1.gif">
This is an image:<BR>
<IMG SRC="abanner.gif" WIDTH=234 HEIGHT=30>
</BODY>
</HTML>
```

### 1.12 Path of image file

Image file must be located in the same directory that html file is located. If the image file resides in other directory, we must add a relational path or a complete URL to this image.

See example below:



`<IMG SRC="images/abanner.gif">` Image is located in "images" directory below the directory that html file resides.

`<IMG SRC="../abanner.gif">` Image is located in parent directory of the directory of html file.

### 1.13 Text links

Creating a text link is an easy task. We will use `<A>` `</A>` tag to do this. As before we will need extra parameters from this tag. Look at example below:

```
<HTML>
<HEAD>
<TITLE>World Of Today</TITLE>
</HEAD>
<BODY>
<A HREF="http://www.yahoo.com">Click here to visit Yahoo</A>
</BODY>
</HTML>
```

Above code will create a link that clicking on it will send the user to Yahoo website. We have used

HREF parameter to specify destination web page. Text between `<A>` and `</A>` is link text which user will click on it to go to destination page.

### 1.14 Image links

It is possible to use an image instead of text. To do this, you must replace link text between `<A>` and `</A>` with an `<IMG>` tag that displays an image file.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
```

Click on below picture to visit my homepage.<BR><BR>

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif">
```

```
</A>
```

```
</BODY>
```

```
</HTML>
```

In above World Of Today clicking on picture will bring surfer to the address of `<A HREF="...">` tag. If you see the result in a browser you will notice a blue border around the picture. This blue border is added to image because it is a default for image links. If you don't want this border, use `border=0` parameter.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>World Of Today</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Click on below picture to visit my homepage.<BR><BR>

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif">
```

```
</A>
```

```
<BR><BR>Without link border : <BR><BR>
```

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif" border=0>
```

```
</A>
```

```
</BODY>
```

```
</HTML>
```

## 1.15 Tables

Table is a matrix like object that holds other objects such as text, images, buttons and etc. Even if you don't see them they are present in all professional web pages. Hidden tables hold graphic images and text in their places in these pages.

### 1.15.1 Drawing a table

To draw a table we will use `<TABLE>` tag. We will need two other related tags to make table rows and columns. These are `<TR>` and `<TD>` tags. `<TR>` tag is used to create a row in table. Data that will fit in a row will be enclosed in `<TR>` `</TR>` tags.

Following World of Today produces a table with two rows. We will need `<TD>` tag to create columns in each row.

```
<TABLE>
<TR>
<TD>First Row</TD>
</TR>
<TR>
<TD>Second Row</TD>
</TR>
</TABLE>
```

If you browse this code in a browser you may surprise. You will not see any table but two lines of

code. In fact table is there but you cannot see it.

`<TABLE>` Tag will not make table borders. You must use a parameter to add borders to the table.

You can specify a border width for a table by adding a border parameter to `<TABLE>` tag.

```
<TABLE BORDER=1>
<TR>
<TD>First Row</TD>
</TR>
<TR>
<TD>Second Row</TD>
</TR>
</TABLE>
```

As you may guess default border size is 0. When we do not specify sizes for a table it will be in a size that it needs to be able to fit text or any other object that it will hold.

### 1.15.2 Specifying table sizes

To specify width for a table both in percents of page width and in pixels.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
<TABLE WIDTH=50% BORDER=1>
<TR>
<TD>Cell Row1 Col1</TD>
<TD>Cell Row1 Col2</TD>
</TR>
<TR>
<TD>Cell Row2 Col1</TD>
<TD>Cell Row2 Col2</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

If you want you can determine table width in pixels.

```
<TABLE WIDTH=250 BORDER=1>
<TR>
<TD>Cell Row1 Col1</TD>
<TD>Cell Row1 Col2</TD>
</TR>
<TR>
<TD>Cell Row2 Col1</TD>
<TD>Cell Row2 Col2</TD>
</TR>
</TABLE>
```



You can specify table height too. In this way you can determine height and width of table. Width and height of table will be divided between cells in rows and columns so if table width is 100 and there are 2 columns then width of each cell will be 50.

Just pay attention to this important point that if you put a lot of text in a cell of a table it will be expanded to fit the text in it.

## CHAPTER 2 JAVASCRIPT

### 2.1 Introduction

Everyone who is the least bit familiar with the Internet eventually wants to represent him or herself there with a home page. But the common page-building programs like Netscape Composer or Microsoft FrontPage no longer suffice for creating anything more than a very mediocre home page. Anyone who wants to have a really cool home page must know a little more than the countless amateurs who are out there on the Internet. The easiest and best tool for creating a truly attractive and interactive home page is called JavaScript. The beautiful thing about JavaScript is that the knowledge and system-related prerequisites for learning the language are relatively low. You just need to know some HTML

### 2.2 Functions

We will use functions in most of our JavaScript programs. Therefore I will talk about this important concept already now. Basically functions are a way for bundling several commands together. Let's write a script which outputs a certain text three times. Consider the following approach:

Example 4.1

```
<html>
<script language="JavaScript">
<!-- hide
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
// -->
</script>
```

```
</html>
```

This will write out the text

Welcome to my homepage!

This is JavaScript!

Three times. Look at the source code - writing the code three times brings out the right result. But is this very efficiently? No, we can solve this better. How about this code which does the same:

Example 4.2

```
<html>
<script language="JavaScript">
<!-- hide
function myFunction() {
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
}
myFunction();
myFunction();
myFunction();
// -->
</script>
</html>
```

In this script we define a function. This is done through the lines:

```
function myFunction() {
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
}
```

The commands inside the {} belong to the function myFunction(). This means that our two document.write() commands are bundled together and can be executed through a function call. In the example we have three function calls. You can see that we write myFunction() three times just below the definition of the function. These are the three function calls. This

means that the contents of the function are being executed three times. This is a very easy example of a function.

~~You~~ might wonder why functions are so important. While reading this tutorial you will certainly realize the benefits of functions. Especially variable passing makes our scripts really flexible.

Functions can also be used in combination with event-handlers. Please consider this example 4.3

```
<html>
<head>
<script language="JavaScript">
<!-- hide
function calculation() {
var x= 12;
var y= 5;
var result= x + y;
alert(result);
}
// -->
</script>
</head>
<body>
<form>
<input type="button" value="Calculate" onClick="calculation()">
</form>
</body>
</html>
```

(The online version lets you test this script immediately)

The button calls the function calculation(). You can see that the function does certain calculations.

For this we are using the variables x, y and result. We can define a variable with the keyword var. Variables can be used to store different values - like numbers, text strings etc..



The line `var result= x + y;` tells the browser to create a variable `result` and store in it the result of `x + y` (i.e. `5 + 12`). After this operation the variable `result` is 17. The command `alert(result)` is in this case the same as `alert(17)`. This means we get a popup window with the number 17 in it.

## 2.3 The Time-object

With the help of timeouts (or timer) you can let the computer execute some code after a certain period of time. The following script shows a button which opens up a popup window after 3 seconds. The script looks like this:

Example 5.1

```
<script language="JavaScript">
<!-- hide
function timer() {
setTimeout("alert('Time is up!')", 3000);
}
// -->
</script>
...
<form>
<input type="button" value="Timer" onClick="timer()">
</form>
setTimeout()
```

Is a method of the window-object? It sets a timeout - I think you might have guessed that. The first argument is the JavaScript code which shall be executed after a certain time. In our case this argument is `"alert('Time is up!')"`. Please note that the JavaScript code has to be inside quotes. The second argument tells the computer when the code shall be executed. You have to specify the time in number of milliseconds (3000 milliseconds = 3 seconds).

## 2.4 The Date-object

JavaScript lets you use some predefined objects. This is for example the Date-object, the Array object or the Math-object. There are several other objects - please refer to the documentation provided by Netscape for a complete reference.

We are going to have a look at the Date-object first. As the name implies this object lets you work with time and date. For example you can easily calculate how many days are left until next christmas. Or you can add the actual time to your HTML-document.

So let's begin with an example which displays the actual time. First we have to create a new Date-object. For this purpose we are using the new operator. Look at this line of code:

```
today = new Date()
```

This creates a new Date-object called today. If you do not specify a certain date and time when creating a new Date-object the actual date and time is used. This means after executing *today = new Date()* the new Date-object *today* represents the date and time of this specific moment.

The Date-object offers some methods which can now be used with our object today. This is for example *getHours()*, *setHours()*, *getMinutes()*, *setMinutes()*, *getMonth()*, *setMonth()* and so on. which changes the time every second or millisecond automatically.

In order to get another date and time we can use another constructor (this is the *Date()* method which is called through the *new* operator when constructing a new Date-object):

*today = new Date(1997, 0, 1, 17, 35, 23)* This will create a Date-object which represents the first of january 1997 at 17:35 and 23 seconds. So you specify the date and time like this:

```
Date(year, month, day, hours, minutes, seconds)
```

Please note that you have to use 0 for January - and not 1 as you might think. 1 stands for February and so on. Now we will write a script which outputs the actual date and time. The result will look like this:

*Time: 17:53*

*Date: 4/3/2010*

The code looks like this:

Example 6.1

```

<script language="JavaScript">
<!-- hide
now= new Date();
document.write("Time: " + now.getHours() + ":" + now.getMinutes() + "<br>");
document.write("Date: " + (now.getMonth() + 1) + "/" + now.getDate() + "/" +
(1900 + now.getYear()));
// -->
</script>

```

Here we use methods like *getHours()* in order to display the time and date specified in our *Date* object *now*. You can see that we are adding 1900 to the year. The method *getYear()* returns the number of years since 1900. This means if the year is 1997 it will return 97 if the year is 2010 it will return 110 - not 10! If we add 1900 we won't have the year 2000 problem. Remember that we have to increment the number we receive through *getMonth()* by one.

This script does not check whether the number of minutes is less than 10. This means you can get a time which looks like this: 14:3 which actually means 14:03. We will see in the next script how to solve this problem. Now we will have a look at a script which displays a working clock:

#### Example 6.2

```

<html>
<head>
<script Language="JavaScript">
<!-- hide
var timeStr, dateStr;
function clock() {
now= new Date();
// time
hours= now.getHours();
minutes= now.getMinutes();
seconds= now.getSeconds();
timeStr= "" + hours;

```



```

timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;
timeStr+= ((seconds < 10) ? ":0" : ":") + seconds;
document.clock.time.value = timeStr;
// date
date= now.getDate();
month= now.getMonth()+1;
year= now.getFullYear();
dateStr= "" + month;
dateStr+= ((date < 10) ? "/0" : "/") + date;
dateStr+= "/" + year;
document.clock.date.value = dateStr;
Timer= setTimeout("clock()",1000);
}
// -->
</script>
</head>
<body onLoad="clock()">
<form name="clock">
Time:
<input type="text" name="time" size="8" value=""><br>
Date:
<input type="text" name="date" size="8" value="">
</form>
</body>
</html>

```

*(The online version lets you test this script immediately)*

We use the `setTimeout()` method for setting the time and date every second. So we create every second a new Date-object with the actual time. You can see that the function `clock()` is called with the `onLoad` event-handler in the `<body>` tag. In the body-part of our HTML-page we have two text-elements. The function `clock()` writes the time and date into these two form-elements in the right format. You can see that we are using two strings `timeStr`



and *dateStr* for this purpose. We have mentioned earlier that there is a problem with minutes less than 10 - this script solves this problem through this line of code: *timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;*

Here the number of minutes are added to the string *timeStr*. If the minutes are less than 10 we have to add a 0. This line of code might look a little bit strange to you. You could also write it like this which might look more familiar:

```
if (minutes < 10) timeStr+= ":0" + minutes
else timeStr+= ":" + minutes;
```

## CHAPTER 3

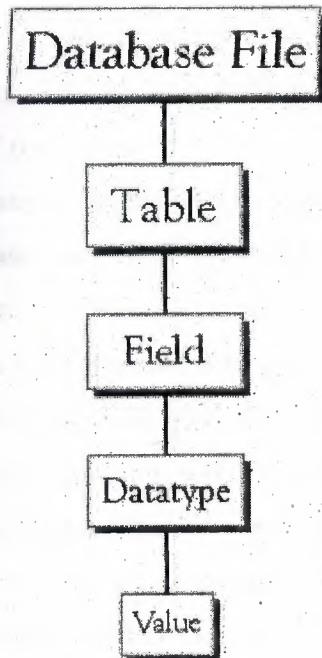
### MICROSOFT ACCESS DATABASE

#### 1.1 Introduction

Access is an interactive, relational database management system. A database is an organized collection of data stored in categories that are accessible in a logical or practical manner.

Relational databases enable data to be stored in multiple tables linked together via data indexes. This makes working with the data faster and easier. Once entered into the database, the data may be manipulated or viewed in various ways such as by sorting or by specially set-up queries and reports.

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, you need to understand how Microsoft Access breaks down a database. Some keywords involved in this process are: *Database File, Table, Record, Field, Data-type*. Here is the Hierarchy that Microsoft Access uses in breaking down a database.



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

**Table:** A table is a collection of data about a specific topic. There can be multiple tables in a database.

**Field:** Fields are the different categories within a Table. Tables usually contain multiple fields.

**Datatypes:** Datatypes are the properties of each field. A field only has 1 datatype.

FieldName) Student LastName

Datatype) Text

## 1.2 Objects

Every database can contain several types of object. The data itself is contained in an object called a table. The data can be used in a wide variety of formats, for example, mailing lists, forms, reports and graphs. Each of these is a type of object. You create objects by clicking on the appropriate tab in the Database window, then clicking on the New button. Once objects such as tables have been created, they will be listed under the object tab whenever it is selected. To open an existing object, click on its name so that it becomes highlighted then click on Open. All objects have properties that can be set to determine how the object appears or operates



## 1.3 Tables

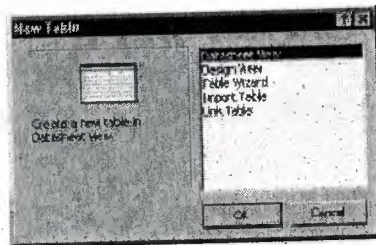
A table is the first type of object to create in a database. It is a way of defining how the data is to be stored. As with all databases, a table consists of 'records' (rows) and 'fields' (columns).

Each record represents one individual item in the database, such as a person in an address book, and each field represents a component of the record such as a surname or part of an address.

Access is a relational database management system. This means that instead of having the data stored in one large file or table, it can be divided into several smaller tables. This reduces the amount of duplication of data and makes it easier to manage. The tables can be related to each other by a common field such as a case identification number. You can work with several tables at the same time.

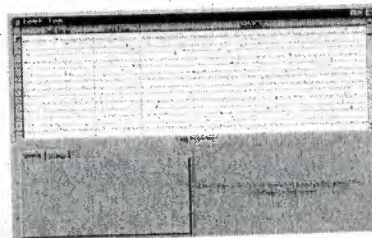
### Creating a Table

To create a table, make sure that the Tables object tab is selected then click on the New button. This presents you with a dialogue box with five choices:



If you select Table Wizard, Access will help you to create a standard table by using pre-defined fields. Selecting Design View, on the other hand allows you to design your own table.

Select Design View and click OK to bring up the Table design window:





In the lower right corner of this window is an area containing a brief explanation of the section containing the cursor? Field Names and

## 1.3.1 Data Types

Each field of the table has three components: the Field Name, which identifies the data stored in the field, the Data Type, which tells Access what kind of data will be stored in the field, such as text, numbers or dates, and the Description, which helps the user remember the purpose of the field. It is very important to choose the right data type for a field at this stage. Access will automatically insert a data type when you name a field but if you need to change it, click on the data type box, and then click on the downward arrow that appears to the right of the box. You can then choose a new data type from the drop-down list.

As you type in the name of each field, a Properties section appears in the lower left corner of the window where you can optionally define several properties for each field. Here you can specify the format of the data to be entered and specify criteria for validating the data as it is entered.

When you have defined the name, data type and properties of each field, save the table by choosing save from the File menu. At this point you will be informed that a Primary key has not been set and you will be asked if you wish Access to create one.

A table is a collection of data about a specific topic. Using a separate table for each topic means that you store that data only once, which makes your database more efficient, and reduces data-entry errors.

Tables organize data into columns (called **fields**) and rows (called **records**).

Each field in the Student Records table contains the same type of information for every student, such as student's Social Security Number (Soc Sec #). This is an example of a **COLUMN**.

Student Records Table					
Soc Sec #	First Name	Last Name	BirthDate	Address	City
123456789	Todd	Jones	1/1/78	312 Wenona Rd	Bay City
315465866	Alan	Craig	2/8/80	123 N Union	Bay City
968585471	Stacy	Evans	3/8/81	RR 5 Box 880	Auburn
848131523	John	Anderson	4/5/80	83 Washington Dr.	Midland

Each record in a Student Records table contains all of the information about one student, such as their First Name, Last Name, Birthday, Address, and City, etc... This is an example of a **ROW**.

### 3.4 Primary Key

One or more fields (columns) whose value or values uniquely identify each record in a table. A primary key does not allow Null values and must always have a unique value. A primary key is used to relate a table to foreign keys in other tables.

A Primary key field is a field that is used uniquely to identify each record. The field can be used by Access to manipulate data more efficiently. You do not have to specify a key. If you wish to use one, you can choose an existing field or, if you answer yes to the prompt for Access to create a Primary key, Access will create an extra field containing an ID number for each case.

### 1.5 Entering Data

Once you have defined the structure of a table, you can start to enter data into it. You created the table in Design view. To enter data, switch to Datasheet view by selecting Datasheet from the View menu. This produces a spreadsheet style window with the name of each field at the top of a column:





The highlighted box or cell of the datasheet denotes the insertion point for data. If you requested Access to create a Primary key, the first cell is a counter called ID, which is automatically filled in by Access. Press TAB to move to the next cell containing your first

## 1.6 Field

- **Saving Data:** You do not have to do anything special to save the data. When you leave a record to go to the next one or close the table, Access automatically saves any changes. To close the table choose Close from the File menu
- **Finding Data:** The simplest way to find a record in a table is: From the database window, open the table in Datasheet view. Select the field to search by clicking on its name box. Choose Find from the Edit menu. Enter the value you want to search for in the Find What? box and click on Find First.

You can also specify whether the search should match for case, in which direction to search and whether to search other fields. When you initiate a search, the cursor moves to the first occurrence of the search string in the table and the string is highlighted. To search for another occurrence of the same value, use the Find Next button.

## 1.7 Relationships

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 have 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 are responsible for by creating a relationship between the teacher's table and the student's table using the TeacherID fields.

## 1.8 Forms

A form is nothing more than a graphical representation of a table. You can add, update, delete records in your table by using a form. NOTE: Although a form can be named different from a table, they both still manipulate the same information and the same exact data. Hence, if you change a record in a form, it will be changed in the table also.

A form is very good to use when you have numerous fields in a table. This way you can see all the fields in one screen, whereas if you were in the table view (datasheet) you would have to keep scrolling to get the field you desire.

## 1.9 Reports

A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.



## **CHAPTER 4 ASP AND ODBC**

### **4.1 Introduction ASP**

Active Server Pages or ASP, as it is more commonly known, is a technology that enables you to make dynamic and interactive web pages. ASP uses server-side scripting to dynamically produce web pages that are not affected by the type of browser the web site visitor is using. The default scripting language used for writing ASP is VBScript, although you can use other scripting languages like JScript (Microsoft's version of JavaScript).

ASP pages have the extension .asp instead of .HTM, when a browser the web server knows to interpret any ASP contained within the web page before sending the HTML produced to the browser requests a page with the extension .asp. This way all the ASP is run on the web server and no ASP will ever be passed to the web browser. Any web pages containing ASP cannot be run by just simply opening the page in a web browser. The page must be requested through a web server that supports ASP, this is why ASP stands for Active Server Pages, no server, no active pages.

As ASP was first introduced by Microsoft on it's web server, Internet Information Services (IIS), that runs on Windows 2000/XP Pro/NT4, it is this web server that ASP pages usually run best on.

For those of you running Windows and wish to play around with ASP on your own system you will need to install Microsoft's Internet Information Services (IIS). Lucky IIS or its micro version Personal Web Server (PWS) comes free with Windows.

### **4.2What is ADO?**

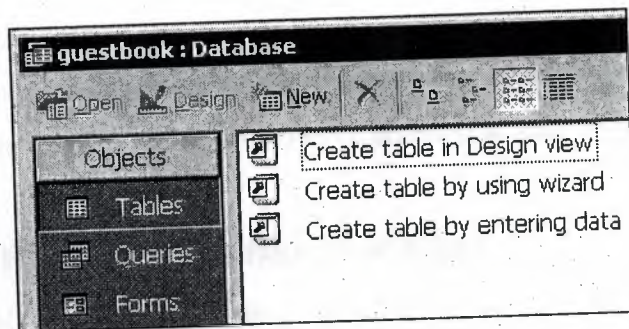
- ADO is a Microsoft technology
- ADO stands for ActiveX Data Objects
- ADO is a Microsoft Active-X component
- ADO is automatically installed with Microsoft IIS

- ADO is a programming interface to access data in a database

### 4.3 Creating the Database

To create a database your first need to open Microsoft Access and choose 'Blank Access Database' from the starting menu. You will be prompted for a name for the database and where you want it saved. Call the database 'World\_news.mdb' and save it in the same directory as the web page connecting to the database is going to be.

You should now see the main Access dialog box, from here select 'Create table in design view'.





You now need to create 3 fields for the database and select there data types.

Field 1 needs to be called 'ID\_no' and have the data type of 'AutoNumber'. Also set this field as the primary key .

Field2 needs to be called 'Name' and have the data type of text. Field 3 needs to be called 'Comments' and also has the data type of text, but this time you need to change the default field size of 50 to 100 characters under the 'General' tab in the 'Field Properties' box at the bottom of the screen..



tblComments : Table		
	Field Name	Data Type
	ID_no	AutoNumber
	Name	Text
	Comments	Text
General   Lookup		
Field Size	100	
Format		

Once the entire fields have been created and the data types and primary key set save the table as 'tblComments'.

Now the table has been created you need to enter some test data into the table. You can do this by double-clicking on the new table (tblComments) in the main dialog box. From here you can enter some test data. I would recommend entering at least 3 pieces of test data.

#### 4.4 Creating an HTML Page to take User Input

First we need to quickly create an HTML page with a form on it to take the input from the user.

In this page we will have two text boxes, one called 'name' and the other called, 'comments', we will then use the post method to send the page to the file, 'add\_to\_World\_news.asp' that we are going to be creating later in this tutorial, which will add the user input into the database.

```
<html>
<head>
<title>World_news Form</title>
</head>
<body bgcolor="white" text="black">
<!-- Begin form code -->
<form name="form" method="post" action="add_to_World_news.asp">
```

```
Name: <input type="text" name="name" maxlength="20">
<br>
Comments: <input type="text" name="comments" maxlength="50">
<input type="submit" name="Submit" value="Submit">
</form>
<!-- End form code -->
</body>
</html>
```

Save the page as *'World\_news\_form.htm'* in the same folder as the *World\_news* database.

#### 4.5 Creating a Page to Select the Database Entry to Delete

First we need to create a page to display the contents of the database so we can select which entry that we want to delete.

I'm not going to go into too much detail on this page as it is almost identical to the page *'World\_news.asp'* we created in the first database tutorial.

The only difference is that we are selecting all fields from the table *'tblComments'* in the *World\_news* database so we are using the SQL query with the wild card, *'tblComments.\*'* to get all the fields from the table.

The other difference is when we are displaying the contents of the database in the web page using the *'Response.Write'* method we are now creating a hyperlink to the *'delete\_entry.asp'* page which we will be creating later in this tutorial to delete the entry.



```
<html>
<head>
<title>Delete Entry Select</title>
</head>
<body bgcolor="white" text="black">
<%
'Dimension variables
Dim adoCon      'Holds the Database Connection Object
Dim rsWorld_news 'Holds the recordset for the records in the database
Dim strSQL      'Holds the SQL query for the database
'Create an ADO connection object
Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection
adoCon.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("World_news.mdb")

'Set an active connection to the Connection object using DSN connection
'adoCon.Open "DSN=World_news"

'Create an ADO recordset object
Set rsWorld_news = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database
strSQL = "SELECT tblComments.* FROM tblComments;"

'Open the recordset with the SQL query
rsWorld_news.Open strSQL, adoCon

'Loop through the recordset
```



```
Do While not rsWorld_news.EOF
    'Write the HTML to display the current record in the recordset
    Response.Write ("<br>")
    Response.Write ("<a href=""delete_entry.asp?ID=" & rsWorld_news("ID_no") &
""">")
    Response.Write (rsWorld_news("Name"))
    Response.Write ("</a>")
    Response.Write ("<br>")
    Response.Write (rsWorld_news("Comments"))
    Response.Write ("<br>")

    'Move to the next record in the recordset
    rsWorld_news.MoveNext
Loop

'Reset server objects
rsWorld_news.Close
Set rsWorld_news = Nothing
Set adoCon = Nothing
%>
</body>
</html>
```

Save this page as '*delete\_select.asp*' in the same folder as the World\_news database.

#### 4.6 Creating a Page to Select the Database Entry to Update

First we need to create a page to display the contents of the database so we can select which entry that we want to update.

I'm not going to go into any detail about this page as it is almost identical to the page we



**NEAR EAST UNIVERSITY**

**Faculty of Engineering**

**Department of Computer Engineering**

**INTERACTIVE NEWS WEB PAGE DESIGN**

**Graduation Project  
COM-400**

**Student: Murad ALzu'bi (20011522)**

**Supervisor: DR. Adil Amirjanov**

*Lefkosa-2005*





## ACKNOWLEDGEMENT

*First, I would like to thank my supervisor*

*Assoc. Prof. Dr Adil Amirjanov for his invaluable advice*

*And for the generosity he exhibited with his time and effort over this project*

*Second, I thank my family especially parents and my fiancé, brothers for their continuous encouragement and support during preparation of this project.*

*Third, I would like to express my gratitude to Near East University*

*Finally, I would like to thank all my friends, especially Emad Aldhdoh, Mahmoud Alfourey, Yazan Alzu'bi and Khaled Alwaled for their support and their constant moral sustain.*



**ABSTRACT**

Nowadays the internet become one of the most important intermedia in the world, therefore it has a significant role in the news side.

According to the changeable events in news transmittance over the world, it has been choosen as quick intermedia. The internet and the world of news therefore in order to be able to have such an attractive for people over world for who are following the news website I used to design as fit as possible to satisfy the satisfactions of users so that the method of display the external design of news webpage should be familiar with the ease transmittance between the whole pages, so that it has to make sense that the designer knows how to coordinate the page thus the JSCRIBT gives ease for designer to design with high level of techniques, To be familiar with the designing fields I should have an enough knowledge about the following subjects HTML ODBC and ASP, whereas I must know ORCAL, SQL, ACCESS to deal with database. That is why I have decided to design this website for the field of news.

## TABLE OF CONTENTS

### ACKNOWLEDGMENT

### ABSTRACT

### TABLE OF CONTENTS

### INTRODUCTION

### CHAPTER ONE: HTML

1.1 Introduction .....	1
1.2 HTML code headers.....	2
1.3 Title.....	2
1.4 Web page body.....	2
1.5 Changing size and face of fonts.....	3
1.6 Size of font .....	3
1.7 Changing font colors.....	3
1.8 Combining Fonts attributes .....	4
1.9 Line Breaks, Paragraphs .....	4
1.10 Paragraph alignments in your web page .....	5
1.11 Images in your web page.....	5
1.12 Path of image files .....	6
1.13 Text links.....	6
1.14 Image links .....	7
1.15 Tables.....	8
1.15.1 Drawing a table.....	8
1.15.2 Specifying table sizes .....	9

### Chapter Two: JavaScript

2.1 Introduction.....	11
2.2 Functions.....	11
2.3 The Time-object .....	14
2.4 The Date-object .....	15

## **CHAPTER THREE: MICROSOFT ACCESS DATABASE**

1.1 Introduction .....	19
1.2 Objects .....	20
1.3 Tables .....	21
1.3.1 Data Types .....	22
1.4 Primary Key .....	23
1.5 Entering Data .....	23
1.6 Field .....	24
1.7 Relationships .....	24
1.8 Forms .....	25
1.9 Reports .....	25

## **CHAPTER FOUR: ASP AND ODBC**

4.1 Introduction ASP .....	26
4.2 What is ADO .....	26
4.3 Creating the Database .....	27
4.4 Creating an HTML Page to take User Input .....	28
4.5 Creating a Page to Select the Database Entry to Delete .....	29
4.6 Creating a Page to Select the Database Entry to Update .....	31

## **CHAPTER FIVE: THE APPLICATION OF MY PROJECT**

5.1 Introduction .....	34
5.2 The contents of my project .....	34
5.3 My Database .....	35
5.4 Administration Page .....	37
5.5 Add page .....	38
5.6 Remove and update pages .....	40
5.6.1 Remove page .....	40
5.6.2 Update page .....	44

## **CONCLUSION**

## **REFERENCES**

## **INTRODUCTION**

My project is about an interaction news website, which means that all pages are interacting with each other, and with other programs, thus all that occurs by using ASP system therefore this system creates same instruction of the WebPages units. in order to design my WebPages as a news pages I decided to be done with display the international daily news such as over whole world news , economic news, sport news, the last versions of technology side beside to cars race news, and according to be able to follow the modern news from time to another I hade to use database which would enable me to add and improve and delete the unbeliever news thus in order to link the database with news pages I hade to use ODBC system which throw it I could make my WebPages interactive during the database , this belongs the practical side of my project whereas the theoretical side of my project is talking about an interactive web site; actually news website, it contains five chapters.

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

Chapter 2 talks about one of the scripting languages, which is Jsript, I talked about its functions, objects.

Chapter 3 talks about Ms Access, how to create table, how to insert, delete, update, retrieve data using SQL statements.

Chapter 4 talks about ASP & ODBC technologies; how to connect databases to the internet and how to interact with users using theses technologies.

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



## CHAPTER 1 HTML

### 1.1 Introduction

HTML is inextricably linked with the development of the World Wide Web. Most of the amazing documents in the WWW are based around HTML. This is a code which describes the logical structure of documents. HTML defines the structure of a text document rather than the layout. For World of Today: here a main heading, there a sub heading, there bold text, there a list, or a line etc. Originally HTML was intended for the world-wide representation of scientific texts. Understandably so, seeing that scientists invented the whole thing. Since then new HTML versions have incorporated numerous little .tricks.. How you see these individual elements depends on your browser, which interprets the HTML code and reproduces the document accordingly.

The hottest browsers are *Opera*, *Netscape Communicator* and *Microsoft Internet Explorer*. The latter is included in Windows 98. It seems currently evident that Microsoft has won the so-called .browser war.. The market share of the Internet Explorer 4 or 5 is double that of the Communicator. HTML is based on ASCII, American Standard Code for Information Interchange. This is a code string which enables text representation. More complex formatting such as bold print, underscoring or centring etc. cannot be represented in ASCII. But then there is HTML.

So called HTML commands, called tags, prescribe the text structure. These commands are place in <> brackets. There is usually one command to switch a characteristic on and another to turn it off. (A few commands are stand alone.)

For World Of Today before a word to be printed bold you find the following symbol: <B>, and after it this symbol : </B>. B for bold . The switch-off command is similar to the switch-on command, with the addition of a slash /. So the World Of Today for bold is written in HTML as follows: <B>bold word group</B> The browser turns this into a bold word group. In either case . it does not matter at all whether the tags are in lower or upper case letters !

So HTML files are just text files with the ending htm or html!

## 1.2 HTML code headers

Every html page must have a header. Header contains important information about the page.

Different tags are used for different sections of a header. Header of an html page is specified by

`<HEAD>` and `</HEAD>` tags.

`<HTML>`

`<HEAD>`

...

`</HEAD>`

`</HTML>`

We will enter header information between `<HEAD>` `</HEAD>` tags.

## 1.3 Title

One of the most important parts of a header is title. Title is the small text that will appear in title bar of viewer's browser. So html document will be as below:

`<HTML>`

`<HEAD>`

`<TITLE>Title of the page</TITLE>`

`</HEAD>`

`</HTML>`

## 1.4 Web page body

Now the web page needs a body in which will enter web page content. As you may guess we will use these tags:

`<BODY>` `</BODY>`

Body will come right after header end tag. So the web page will be something like this

Welcome to our homepage.

## 1.5 Changing size and face of fonts

We can change face and size of fonts using `<FONT>...</FONT>` tags. Also using this tag alone will not change the text. You need to use parameters for this tag. These parameters specify what kind of change you need in text font.

## 1.6 Size of font

To change size of font in a part of text, include it with a `<FONT>` tag as below:  
`<FONT SIZE=n>...</font>` n is size of font. Size of font must be a number between 1 and 7.  
If you insert some text without determining its size default size will be 3.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
<FONT SIZE=1>1This text is bold</FONT><br>
<FONT SIZE=2>2This text is bold</FONT><br>
<FONT SIZE=3>3This text is bold</FONT><br>
<FONT SIZE=4>4This text is bold</FONT><br>
<FONT SIZE=5>5This text is bold</FONT><br>
<FONT SIZE=6>6This text is bold</FONT><br>
<FONT SIZE=7>7This text is bold</FONT><br>
</BODY>
</HTML>
```

## 1.7 Changing font colors

In previous lesson you learned how to change web page background color. Here we will learn how to change text color. Look at this World Of Today:

```
<HTML>
<HEAD>
```



```
<TITLE>World Of Today</TITLE>
</HEAD>
<BODY>
<FONT COLOR="#FF0000">This text is in red color.</FONT><br>
<FONT COLOR="#00FF00">This text is in green color.</FONT><br>
<FONT COLOR="#0000FF">This text is in blue color.</FONT><br>
</BODY>
</HTML>
```

In above text different colors are used. You can change text color by changing color number. If you don't remember how to specify color numbers return to previous lesson and review section on "web page background color".

## 1.8 Combining Fonts attributes

We can obviously combine <FONT> tag parameters as we wish. In this way we will be able to have text with different colors, font faces and font sizes.

You can even use text styles with <FONT> tag by nesting style and font tags.

```
<B><I><FONT SIZE="5" FACE="IMPACT" COLOR="#00FF00">
```

How is this?

```
</FONT></I></B>
```

## 1.9 Line Breaks, Paragraphs

As we saw in previous lesson if we break lines in html code by simply inserting enter keys (new line characters), lines will not break in output result in browser. They will be printed in a single line in browser. We must use <BR> tag to do this as you used it in previous lesson.

You can also divide text using paragraphs. A paragraph starts on a new line with one blank line after previous line. Paragraph tag is <p> </p>

```
<p>First paragraph</p>
```

```
<p>Second paragraph</p>
```

You will nest other tags inside paragraph tag for fonts, styles and other tags that will be used inside a paragraph.

There is another option in forming text that is using `<PRE>` tag. Text between `<PRE>` `</PRE>` tags will be displayed exactly as it is typed in html source. Therefore you will not need `<BR>` tags to break lines. It is enough to enter text in separate lines with enter key (new line character) at their end.

### 1.10 Paragraph alignments in your web page

You can determine how a paragraph will be aligned in a web page. You can align text in left, right or center of a web page. To specify alignment for a paragraph you must use an extra parameter for your paragraph tag. You can use one of these combinations:

```
<P ALIGN="left"> </P>
<P ALIGN="center"> </P>
<P ALIGN="right"> </P>
<HTML>
<HEAD>
<TITLE>World Of Today</TITLE>
</HEAD>
<BODY>
<P ALIGN="left">You can align text in left.</P>
<P ALIGN="center">You can align text in center.</P>
<P ALIGN="right">You can align text in right.</P>
</BODY>
</HTML>
```

### 1.11 Images in your web page

To add an image in a web page. Tag that will be used for this purpose is `<IMG>` tag. Actually we will need parameters for this tag that specify image file location, file name and other optional parameters.

Look at this World Of Today:

### World Of Today 11.1

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY BACKGROUND="image1.gif">
<B>This is an image:</B><BR>
<IMG SRC="abanner.gif">
</BODY>
</HTML>
```

In this World Of Today it used both a background image and an image between texts. Also you may have noticed that this tag is also a single tag that does not need an ending tag.

If you want to show your image in a different size than its real size, you can specify its size as below.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY BACKGROUND="image1.gif">
This is an image:<BR>
<IMG SRC="abanner.gif" WIDTH=234 HEIGHT=30>
</BODY>
</HTML>
```

### 1.12 Path of image file

Image file must be located in the same directory that html file is located. If the image file resides in other directory, we must add a relational path or a complete URL to this image.

See example below:



`<IMG SRC="images/abanner.gif">` Image is located in "images" directory below the directory that html file resides.

`<IMG SRC="../abanner.gif">` Image is located in parent directory of the directory of html file.

### 1.13 Text links

Creating a text link is an easy task. We will use `<A>` `</A>` tag to do this. As before we will need extra parameters from this tag. Look at example below:

```
<HTML>
<HEAD>
<TITLE>World Of Today</TITLE>
</HEAD>
<BODY>
<A HREF="http://www.yahoo.com">Click here to visit Yahoo</A>
</BODY>
</HTML>
```

Above code will create a link that clicking on it will send the user to Yahoo website. We have used

HREF parameter to specify destination web page. Text between `<A>` and `</A>` is link text which user will click on it to go to destination page.

### 1.14 Image links

It is possible to use an image instead of text. To do this, you must replace link text between `<A>` and `</A>` with an `<IMG>` tag that displays an image file.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
```

Click on below picture to visit my homepage.<BR><BR>

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif">
```

```
</A>
```

```
</BODY>
```

```
</HTML>
```

In above World Of Today clicking on picture will bring surfer to the address of `<A HREF="...">` tag. If you see the result in a browser you will notice a blue border around the picture. This blue border is added to image because it is a default for image links. If you don't want this border, use `border=0` parameter.

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>World Of Today</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Click on below picture to visit my homepage.<BR><BR>

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif">
```

```
</A>
```

```
<BR><BR>Without link border : <BR><BR>
```

```
<A HREF="http://www.angelfire.com/nt/sarmadys">
```

```
<IMG SRC="me.gif" border=0>
```

```
</A>
```

```
</BODY>
```

```
</HTML>
```

## 1.15 Tables

Table is a matrix like object that holds other objects such as text, images, buttons and etc. Even if you don't see them they are present in all professional web pages. Hidden tables hold graphic images and text in their places in these pages.

### 1.15.1 Drawing a table

To draw a table we will use `<TABLE>` tag. We will need two other related tags to make table rows and columns. These are `<TR>` and `<TD>` tags. `<TR>` tag is used to create a row in table. Data that will fit in a row will be enclosed in `<TR>` `</TR>` tags.

Following World ~~Of~~ Today produces a table with two rows. We will need `<TD>` tag to create columns in each row.

```
<TABLE>
<TR>
<TD>First Row</TD>
</TR>
<TR>
<TD>Second Row</TD>
</TR>
</TABLE>
```

If you browse this code in a browser you may surprise. You will not see any table but two lines of

code. In fact table is there but you cannot see it.

`<TABLE>` Tag will not make table borders. You must use a parameter to add borders to the table.

You can specify a border width for a table by adding a border parameter to `<TABLE>` tag.

```
<TABLE BORDER=1>
<TR>
<TD>First Row</TD>
</TR>
<TR>
<TD>Second Row</TD>
</TR>
</TABLE>
```

As you may guess default border size is 0. When we do not specify sizes for a table it will be in a size that it needs to be able to fit text or any other object that it will hold.



### 1.15.2 Specifying table sizes

To specify width for a table both in percents of page width and in pixels.

```
<HTML>
<HEAD>
<TITLE>World Of Today </TITLE>
</HEAD>
<BODY>
<TABLE WIDTH=50% BORDER=1>
<TR>
<TD>Cell Row1 Col1</TD>
<TD>Cell Row1 Col2</TD>
</TR>
<TR>
<TD>Cell Row2 Col1</TD>
<TD>Cell Row2 Col2</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

If you want you can determine table width in pixels.

```
<TABLE WIDTH=250 BORDER=1>
<TR>
<TD>Cell Row1 Col1</TD>
<TD>Cell Row1 Col2</TD>
</TR>
<TR>
<TD>Cell Row2 Col1</TD>
<TD>Cell Row2 Col2</TD>
</TR>
</TABLE>
```

You can specify table height too. In this way you can determine height and width of table. Width and height of table will be divided between cells in rows and columns so if table width is 100 and there are 2 columns then width of each cell will be 50.

Just pay attention to this important point that if you put a lot of text in a cell of a table it will be expanded to fit the text in it.

## CHAPTER 2 JAVASCRIPT

### 2.1 Introduction

Everyone who is the least bit familiar with the Internet eventually wants to represent him or herself there with a home page. But the common page-building programs like Netscape Composer or Microsoft FrontPage no longer suffice for creating anything more than a very mediocre home page. Anyone who wants to have a really cool home page must know a little more than the countless amateurs who are out there on the Internet. The easiest and best tool for creating a truly attractive and interactive home page is called JavaScript. The beautiful thing about JavaScript is that the knowledge and system-related prerequisites for learning the language are relatively low. You just need to know some HTML

### 2.2 Functions

We will use functions in most of our JavaScript programs. Therefore I will talk about this important concept already now. Basically functions are a way for bundling several commands together. Let's write a script which outputs a certain text three times. Consider the following approach:

Example 4.1

```
<html>
<script language="JavaScript">
<!-- hide
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
// -->
</script>
```



```
</html>
```

This will write out the text

Welcome to my homepage!

This is JavaScript!

Three times. Look at the source code - writing the code three times brings out the right result. But is this very efficiently? No, we can solve this better. How about this code which does the same:

Example 4.2

```
<html>
<script language="JavaScript">
<!-- hide
function myFunction() {
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
}
myFunction();
myFunction();
myFunction();
// -->
</script>
</html>
```

In this script we define a function. This is done through the lines:

```
function myFunction() {
document.write("Welcome to my homepage!<br>");
document.write("This is JavaScript!<br>");
}
```

The commands inside the {} belong to the function myFunction(). This means that our two document.write() commands are bundled together and can be executed through a function call. In the example we have three function calls. You can see that we write myFunction() three times just below the definition of the function. These are the three function calls. This

means that the contents of the function are being executed three times. This is a very easy example of a function.

~~You~~ might wonder why functions are so important. While reading this tutorial you will certainly realize the benefits of functions. Especially variable passing makes our scripts really flexible.

Functions can also be used in combination with event-handlers. Please consider this example 4.3

```
<html>
<head>
<script language="JavaScript">
<!-- hide
function calculation() {
var x= 12;
var y= 5;
var result= x + y;
alert(result);
}
// -->
</script>
</head>
<body>
<form>
<input type="button" value="Calculate" onClick="calculation()">
</form>
</body>
</html>
```

(The online version lets you test this script immediately)

The button calls the function calculation(). You can see that the function does certain calculations.

For this we are using the variables x, y and result. We can define a variable with the keyword var. Variables can be used to store different values - like numbers, text strings etc..

The line `var result= x + y;` tells the browser to create a variable `result` and store in it the result of `x + y` (i.e. `5 + 12`). After this operation the variable `result` is 17. The command `alert(result)` is in this case the same as `alert(17)`. This means we get a popup window with the number 17 in it.

## 2.3 The Time-object

With the help of timeouts (or timer) you can let the computer execute some code after a certain period of time. The following script shows a button which opens up a popup window after 3 seconds. The script looks like this:

Example 5.1

```
<script language="JavaScript">
<!-- hide
function timer() {
setTimeout("alert('Time is up!')", 3000);
}
// -->
</script>
...
<form>
<input type="button" value="Timer" onClick="timer()">
</form>
setTimeout()
```

Is a method of the window-object? It sets a timeout - I think you might have guessed that. The first argument is the JavaScript code which shall be executed after a certain time. In our case this argument is `"alert('Time is up!')"`. Please note that the JavaScript code has to be inside quotes. The second argument tells the computer when the code shall be executed. You have to specify the time in number of milliseconds (3000 milliseconds = 3 seconds).



## 2.4 The Date-object

JavaScript lets you use some predefined objects. This is for example the Date-object, the Array object or the Math-object. There are several other objects - please refer to the documentation provided by Netscape for a complete reference.

We are going to have a look at the Date-object first. As the name implies this object lets you work with time and date. For example you can easily calculate how many days are left until next christmas. Or you can add the actual time to your HTML-document.

So let's begin with an example which displays the actual time. First we have to create a new Date-object. For this purpose we are using the new operator. Look at this line of code:

```
today= new Date()
```

This creates a new Date-object called today. If you do not specify a certain date and time when creating a new Date-object the actual date and time is used. This means after executing *today= new Date()* the new Date-object *today* represents the date and time of this specific moment.

The Date-object offers some methods which can now be used with our object today. This is for example *getHours()*, *setHours()*, *getMinutes()*, *setMinutes()*, *getMonth()*, *setMonth()* and so on. which changes the time every second or millisecond automatically.

In order to get another date and time we can use another constructor (this is the *Date()* method which is called through the *new* operator when constructing a new Date-object):

*today= new Date(1997, 0, 1, 17, 35, 23)* This will create a Date-object which represents the first of january 1997 at 17:35 and 23 seconds. So you specify the date and time like this:

```
Date(year, month, day, hours, minutes, seconds)
```

Please note that you have to use 0 for January - and not 1 as you might think. 1 stands for February and so on. Now we will write a script which outputs the actual date and time. The result will look like this:

*Time: 17:53*

*Date: 4/3/2010*

The code looks like this:

Example 6.1

```

<script language="JavaScript">
<!-- hide
now= new Date();
document.write("Time: " + now.getHours() + ":" + now.getMinutes() + "<br>");
document.write("Date: " + (now.getMonth() + 1) + "/" + now.getDate() + "/" +
(1900 + now.getYear()));
// -->
</script>

```

Here we use methods like *getHours()* in order to display the time and date specified in our *Date* object *now*. You can see that we are adding 1900 to the year. The method *getYear()* returns the number of years since 1900. This means if the year is 1997 it will return 97 if the year is 2010 it will return 110 - not 10! If we add 1900 we won't have the year 2000 problem. Remember that we have to increment the number we receive through *getMonth()* by one.

This script does not check whether the number of minutes is less than 10. This means you can get a time which looks like this: 14:3 which actually means 14:03. We will see in the next script how to solve this problem. Now we will have a look at a script which displays a working clock:

#### Example 6.2

```

<html>
<head>
<script Language="JavaScript">
<!-- hide
var timeStr, dateStr;
function clock() {
now= new Date();
// time
hours= now.getHours();
minutes= now.getMinutes();
seconds= now.getSeconds();
timeStr= "" + hours;

```

```

timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;
timeStr+= ((seconds < 10) ? ":0" : ":") + seconds;
document.clock.time.value = timeStr;
// date
date= now.getDate();
month= now.getMonth()+1;
year= now.getFullYear();
dateStr= "" + month;
dateStr+= ((date < 10) ? "/0" : "/") + date;
dateStr+= "/" + year;
document.clock.date.value = dateStr;
Timer= setTimeout("clock()",1000);
}
// -->
</script>
</head>
<body onLoad="clock()">
<form name="clock">
Time:
<input type="text" name="time" size="8" value=""><br>
Date:
<input type="text" name="date" size="8" value="">
</form>
</body>
</html>

```

(The online version lets you test this script immediately)

We use the `setTimeout()` method for setting the time and date every second. So we create every second a new Date-object with the actual time. You can see that the function `clock()` is called with the `onLoad` event-handler in the `<body>` tag. In the body-part of our HTML-page we have two text-elements. The function `clock()` writes the time and date into these two form-elements in the right format. You can see that we are using two strings `timeStr`



and *dateStr* for this purpose. We have mentioned earlier that there is a problem with minutes less than 10 - this script solves this problem through this line of code: *timeStr+= ((minutes < 10) ? ":0" : ":") + minutes;*

Here the number of minutes are added to the string *timeStr*. If the minutes are less than 10 we have to add a 0. This line of code might look a little bit strange to you. You could also write it like this which might look more familiar:

```
if (minutes < 10) timeStr+= ":0" + minutes
else timeStr+= ":" + minutes;
```

## **CHAPTER 3**

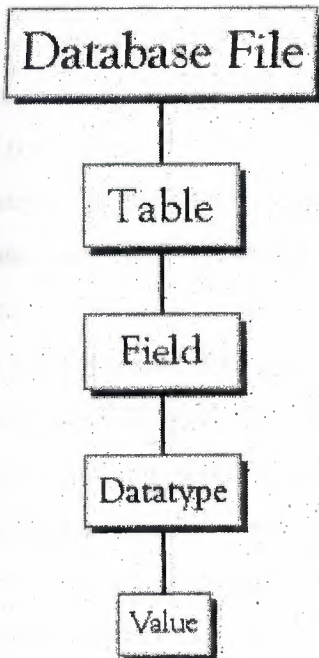
### **MICROSOFT ACCESS DATABASE**

#### **1.1 Introduction**

Access is an interactive, relational database management system. A database is an organized collection of data stored in categories that are accessible in a logical or practical manner.

Relational databases enable data to be stored in multiple tables linked together via data indexes. This makes working with the data faster and easier. Once entered into the database, the data may be manipulated or viewed in various ways such as by sorting or by specially set-up queries and reports.

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, you need to understand how Microsoft Access breaks down a database. Some keywords involved in this process are: *Database File, Table, Record, Field, Data-type*. Here is the Hierarchy that Microsoft Access uses in breaking down a database.



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

**Table:** A table is a collection of data about a specific topic. There can be multiple tables in a database.

**Field:** Fields are the different categories within a Table. Tables usually contain multiple fields.

**Datatypes:** Datatypes are the properties of each field. A field only has 1 datatype.

FieldName) Student LastName

Datatype) Text

## 1.2 Objects

Every database can contain several types of object. The data itself is contained in an object called a table. The data can be used in a wide variety of formats, for example, mailing lists, forms, reports and graphs. Each of these is a type of object. You create objects by clicking on the appropriate tab in the Database window, then clicking on the New button. Once objects such as tables have been created, they will be listed under the object tab whenever it is selected. To open an existing object, click on its name so that it becomes highlighted then click on Open. All objects have properties that can be set to determine how the object appears or operates



## 1.3 Tables

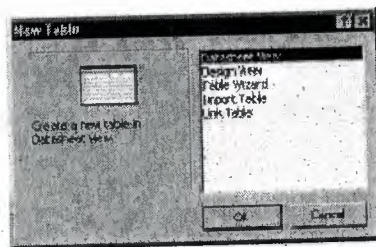
A table is the first type of object to create in a database. It is a way of defining how the data is to be stored. As with all databases, a table consists of 'records' (rows) and 'fields' (columns).

Each record represents one individual item in the database, such as a person in an address book, and each field represents a component of the record such as a surname or part of an address.

Access is a relational database management system. This means that instead of having the data stored in one large file or table, it can be divided into several smaller tables. This reduces the amount of duplication of data and makes it easier to manage. The tables can be related to each other by a common field such as a case identification number. You can work with several tables at the same time.

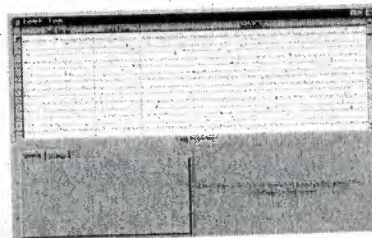
### Creating a Table

To create a table, make sure that the Tables object tab is selected then click on the New button. This presents you with a dialogue box with five choices:



If you select Table Wizard, Access will help you to create a standard table by using pre-defined fields. Selecting Design View, on the other hand allows you to design your own table.

Select Design View and click OK to bring up the Table design window:



In the lower right corner of this window is an area containing a brief explanation of the section containing the cursor? Field Names and

## 1.3.1 Data Types

Each field of the table has three components: the Field Name, which identifies the data stored in the field, the Data Type, which tells Access what kind of data will be stored in the field, such as text, numbers or dates, and the Description, which helps the user remember the purpose of the field. It is very important to choose the right data type for a field at this stage. Access will automatically insert a data type when you name a field but if you need to change it, click on the data type box, and then click on the downward arrow that appears to the right of the box. You can then choose a new data type from the drop-down list.

As you type in the name of each field, a Properties section appears in the lower left corner of the window where you can optionally define several properties for each field. Here you can specify the format of the data to be entered and specify criteria for validating the data as it is entered.

When you have defined the name, data type and properties of each field, save the table by choosing save from the File menu. At this point you will be informed that a Primary key has not been set and you will be asked if you wish Access to create one.

A table is a collection of data about a specific topic. Using a separate table for each topic means that you store that data only once, which makes your database more efficient, and reduces data-entry errors.

Tables organize data into columns (called **fields**) and rows (called **records**).



Each field in the Student Records table contains the same type of information for every student, such as student's Social Security Number (Soc Sec #). This is an example of a **COLUMN**.

Student Records Table					
Soc Sec #	First Name	Last Name	BirthDate	Address	City
123456789	Todd	Jones	1/1/78	312 Wenona Rd	Bay City
315465866	Alan	Craig	2/8/80	123 N Union	Bay City
968585471	Stacy	Evans	3/8/81	RR 5 Box 880	Auburn
848131523	John	Anderson	4/5/80	83 Washington Dr.	Midland

Each record in a Student Records table contains all of the information about one student, such as their First Name, Last Name, Birthday, Address, and City, etc... This is an example of a **ROW**.

### 3.4 Primary Key

One or more fields (columns) whose value or values uniquely identify each record in a table. A primary key does not allow Null values and must always have a unique value. A primary key is used to relate a table to foreign keys in other tables.

A Primary key field is a field that is used uniquely to identify each record. The field can be used by Access to manipulate data more efficiently. You do not have to specify a key. If you wish to use one, you can choose an existing field or, if you answer yes to the prompt for Access to create a Primary key, Access will create an extra field containing an ID number for each case.

### 1.5 Entering Data

Once you have defined the structure of a table, you can start to enter data into it. You created the table in Design view. To enter data, switch to Datasheet view by selecting Datasheet from the View menu. This produces a spreadsheet style window with the name of each field at the top of a column:





The highlighted box or cell of the datasheet denotes the insertion point for data. If you requested Access to create a Primary key, the first cell is a counter called ID, which is automatically filled in by Access. Press TAB to move to the next cell containing your first

## 1.6 Field

- **Saving Data:** You do not have to do anything special to save the data. When you leave a record to go to the next one or close the table, Access automatically saves any changes. To close the table choose Close from the File menu
- **Finding Data:** The simplest way to find a record in a table is: From the database window, open the table in Datasheet view. Select the field to search by clicking on its name box. Choose Find from the Edit menu. Enter the value you want to search for in the Find What? box and click on Find First.

You can also specify whether the search should match for case, in which direction to search and whether to search other fields. When you initiate a search, the cursor moves to the first occurrence of the search string in the table and the string is highlighted. To search for another occurrence of the same value, use the Find Next button.

## 1.7 Relationships

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 have 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 are responsible for by creating a relationship between the teacher's table and the student's table using the TeacherID fields.

## 1.8 Forms

A form is nothing more than a graphical representation of a table. You can add, update, delete records in your table by using a form. NOTE: Although a form can be named different from a table, they both still manipulate the same information and the same exact data. Hence, if you change a record in a form, it will be changed in the table also.

A form is very good to use when you have numerous fields in a table. This way you can see all the fields in one screen, whereas if you were in the table view (datasheet) you would have to keep scrolling to get the field you desire.

## 1.9 Reports

A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.

## **CHAPTER 4 ASP AND ODBC**

### **4.1 Introduction ASP**

Active Server Pages or ASP, as it is more commonly known, is a technology that enables you to make dynamic and interactive web pages. ASP uses server-side scripting to dynamically produce web pages that are not affected by the type of browser the web site visitor is using. The default scripting language used for writing ASP is VBScript, although you can use other scripting languages like JScript (Microsoft's version of JavaScript).

ASP pages have the extension .asp instead of .HTM, when a browser the web server knows to interpret any ASP contained within the web page before sending the HTML produced to the browser requests a page with the extension .asp. This way all the ASP is run on the web server and no ASP will ever be passed to the web browser. Any web pages containing ASP cannot be run by just simply opening the page in a web browser. The page must be requested through a web server that supports ASP, this is why ASP stands for Active Server Pages, no server, no active pages.

As ASP was first introduced by Microsoft on it's web server, Internet Information Services (IIS), that runs on Windows 2000/XP Pro/NT4, it is this web server that ASP pages usually run best on.

For those of you running Windows and wish to play around with ASP on your own system you will need to install Microsoft's Internet Information Services (IIS). Lucky IIS or its micro version Personal Web Server (PWS) comes free with Windows.

### **4.2What is ADO?**

- ADO is a Microsoft technology
- ADO stands for ActiveX Data Objects
- ADO is a Microsoft Active-X component
- ADO is automatically installed with Microsoft IIS

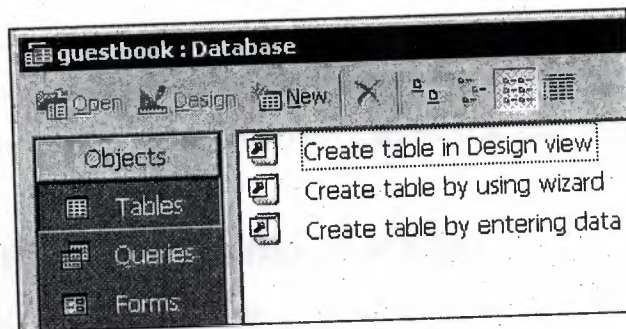


- ADO is a programming interface to access data in a database

### 4.3 Creating the Database

To create a database your first need to open Microsoft Access and choose 'Blank Access Database' from the starting menu. You will be prompted for a name for the database and where you want it saved. Call the database 'World\_news.mdb' and save it in the same directory as the web page connecting to the database is going to be.



You should now see the main Access dialog box, from here select 'Create table in design view'.



You now need to create 3 fields for the database and select there data types.

Field 1 needs to be called 'ID\_no' and have the data type of 'AutoNumber'. Also set this field as the primary key .

Field2 needs to be called 'Name' and have the data type of text. Field 3 needs to be called 'Comments' and also has the data type of text, but this time you need to change the default field size of 50 to 100 characters under the 'General' tab in the 'Field Properties' box at the bottom of the screen..

tblComments : Table		
	Field Name	Data Type
	ID_no	AutoNumber
	Name	Text
	Comments	Text
General   Lookup		
Field Size	100	
Format		

Once the entire fields have been created and the data types and primary key set save the table as 'tblComments'.

Now the table has been created you need to enter some test data into the table. You can do this by double-clicking on the new table (tblComments) in the main dialog box. From here you can enter some test data. I would recommend entering at least 3 pieces of test data.

#### 4.4 Creating an HTML Page to take User Input

First we need to quickly create an HTML page with a form on it to take the input from the user.

In this page we will have two text boxes, one called 'name' and the other called, 'comments', we will then use the post method to send the page to the file, 'add\_to\_World\_news.asp' that we are going to be creating later in this tutorial, which will add the user input into the database.

```
<html>
<head>
<title>World_news Form</title>
</head>
<body bgcolor="white" text="black">
<!-- Begin form code -->
<form name="form" method="post" action="add_to_World_news.asp">
```



```
Name: <input type="text" name="name" maxlength="20">
<br>
Comments: <input type="text" name="comments" maxlength="50">
<input type="submit" name="Submit" value="Submit">
</form>
<!-- End form code -->
</body>
</html>
```

Save the page as *'World\_news\_form.htm'* in the same folder as the *World\_news* database.

#### 4.5 Creating a Page to Select the Database Entry to Delete

First we need to create a page to display the contents of the database so we can select which entry that we want to delete.

I'm not going to go into too much detail on this page as it is almost identical to the page *'World\_news.asp'* we created in the first database tutorial.

The only difference is that we are selecting all fields from the table *'tblComments'* in the *World\_news* database so we are using the SQL query with the wild card, *'tblComments.\*'* to get all the fields from the table.

The other difference is when we are displaying the contents of the database in the web page using the *'Response.Write'* method we are now creating a hyperlink to the *'delete\_entry.asp'* page which we will be creating later in this tutorial to delete the entry.



```
<html>
<head>
<title>Delete Entry Select</title>
</head>
<body bgcolor="white" text="black">
<%
'Dimension variables
Dim adoCon      'Holds the Database Connection Object
Dim rsWorld_news 'Holds the recordset for the records in the database
Dim strSQL      'Holds the SQL query for the database
'Create an ADO connection object
Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection
adoCon.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("World_news.mdb")

'Set an active connection to the Connection object using DSN connection
'adoCon.Open "DSN=World_news"

'Create an ADO recordset object
Set rsWorld_news = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database
strSQL = "SELECT tblComments.* FROM tblComments;"

'Open the recordset with the SQL query
rsWorld_news.Open strSQL, adoCon

'Loop through the recordset
```



```
Do While not rsWorld_news.EOF
    'Write the HTML to display the current record in the recordset
    Response.Write ("<br>")
    Response.Write ("<a href=""delete_entry.asp?ID=" & rsWorld_news("ID_no") &
""">")
    Response.Write (rsWorld_news("Name"))
    Response.Write ("</a>")
    Response.Write ("<br>")
    Response.Write (rsWorld_news("Comments"))
    Response.Write ("<br>")

    'Move to the next record in the recordset
    rsWorld_news.MoveNext
Loop

'Reset server objects
rsWorld_news.Close
Set rsWorld_news = Nothing
Set adoCon = Nothing
%>
</body>
</html>
```

Save this page as 'delete\_select.asp' in the same folder as the World\_news database.

#### 4.6 Creating a Page to Select the Database Entry to Update

First we need to create a page to display the contents of the database so we can select which entry that we want to update.

I'm not going to go into any detail about this page as it is almost identical to the page we



```

Do While not rsWorld_news.EOF
    'Write the HTML to display the current record in the recordset
    Response.Write ("<br>")
    Response.Write ("<a href=""delete_entry.asp?ID=" & rsWorld_news("ID_no") &
"">")
    Response.Write (rsWorld_news("Name"))
    Response.Write ("</a>")
    Response.Write ("<br>")
    Response.Write (rsWorld_news("Comments"))
    Response.Write ("<br>")

    'Move to the next record in the recordset
    rsWorld_news.MoveNext
Loop

'Reset server objects
rsWorld_news.Close
Set rsWorld_news = Nothing
Set adoCon = Nothing
%>
</body>
</html>

```

Save this page as '*delete\_select.asp*' in the same folder as the World\_news database.

## 4.6 Creating a Page to Select the Database Entry to Update

First we need to create a page to display the contents of the database so we can select which entry that we want to update.

I'm not going to go into any detail about this page as it is almost identical to the page we



created in the last database tutorial on selecting an entry to delete. The only difference is the hyperlink created is to the form, 'update\_form.asp', that we are going to create next in this tutorial.

```
<html>
<head>
<title>Update Entry Select</title>
</head>
<body bgcolor="white" text="black">
<%
'Dimension variables
Dim adoCon      'Holds the Database Connection Object
Dim rsWorld_news 'Holds the recordset for the records in the database
Dim strSQL      'Holds the SQL query for the database
'Create an ADO connection object
Set adoCon = Server.CreateObject("ADODB.Connection")

'Set an active connection to the Connection object using a DSN-less connection
adoCon.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("World_news.mdb")

'Set an active connection to the Connection object using DSN connection
'adoCon.Open "DSN=World_news"

'Create an ADO recordset object
Set rsWorld_news = Server.CreateObject("ADODB.Recordset")

'Initialise the strSQL variable with an SQL statement to query the database
strSQL = "SELECT tblComments.* FROM tblComments;"

'Open the recordset with the SQL query
```



```
rsWorld_news.Open strSQL, adoCon

'Loop through the recordset
Do While not rsWorld_news.EOF
    'Write the HTML to display the current record in the recordset
    Response.Write ("<br>")
    Response.Write ("<a href=""update_form.asp?ID=" & rsWorld_news("ID_no") &
""">")
    Response.Write (rsWorld_news("Name"))
    Response.Write ("</a>")
    Response.Write ("<br>")
    Response.Write (rsWorld_news("Comments"))
    Response.Write ("<br>")

    'Move to the next record in the recordset
    rsWorld_news.MoveNext
Loop

'Reset server objects
rsWorld_news.Close
Set rsWorld_news = Nothing
Set adoCon = Nothing

%>
</body>
</html>
```

Save this page as '*update\_select.asp*' in the same folder as the World\_news database.



## **CHAPTER 5**

### **NEWS PAGE DESIGN**

#### **5.1 Introduction**

In the beginning I would like to declare what is going on my practical project which belongs the news WebPages that could be updatable from time to another, therefore the changeable daily news require such a specific area to be followed sequentially, so that I decided to design this webpage to be one of optimum choices that may I will get job within, so during my studying I found that there are too many alternatives fields that can be my future choice, whereas I detected that I will be professional and familiar with WebPages designing.

#### **5.2 The contents of my project**

My project contains five news website pages thus the first page belongs the main page which called (world of today), the second page refers to the economic side of news which called (world of economy), the third page related with the technology side of modern life and the last development occurs in the dominative changes life, the fourth page deals with sport throw its all related field and the fifth page which satisfies the young's race cars satisfactions. And these pages would be updatable during the event changes from time to another accordingly and also to go more in the details about my project in my project I have webpage administration page which allows me to control about many things inside my project for example if I need to add any information to the World Of Today it will travel me to addition page which is one of three links inside this page (ADMINISTRATION PAGE), the second link goes to the delete page which allows me to delete any information in my project and finally the third link get me on the updating page which achieves me to keep continue if I need to update any information immediately in my website and just I want to press your attention that all these pages which inside the administration page are connected by ODBC and ASP (Active Server Pages) by using the java script and Database.



### 5.3 My Database

firstly; I use the Microsoft access to design my database inside which include particular specification for my data as a table which includes six fields and this is my explanation about each field alone and here figure(5\_1) here express the way that I use it input my database.

ID	ADD_DATE	SUBJECT	BODEY	PICTURE	TYPE
2	1/1/2000	sub1	BBBBBBBBBB		1
3	1/1/2000	sub3	111111111111		1
4	1/1/2000	sub4	as'tnx 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
16	5/5/2005	phylistin	fertryrt	plan.jpg	5
19	5/5/2005	alla	trytghfghfghfghf	hp5-28-05zz.jpg	4
21	8/5/2005	&#1581;&#1585	&#1578; &#1605	btn.JPG	1
66	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
70	2/2/2005	kjhk	khkj	top.sat2.iraq.ap	2
77	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
88	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
99	2/2/2002	fdsgsdgsg	tretargf	btn.JPG	2
101	5/28/2005	murad	safdddddffsfss	cars.jpg	2
*	0				0

Figure 1: Microsoft access



- Primary key:

(actually this application is used here just to combine the value or values uniquely identify each record in a table with other values in other fields to manipulate data more efficiently, I mean here that I can use it in case that I need to update or delete any information by using the primary key I can specify any information that I need depending on the property of this field.

- ADD\_DATE:

This field expresses the date of the news which is inserted in the website.

- SUBJECT:

During this field I can show the head line of the news in the website.

- Body:

From this field I can publish my news into the website.

- Picture:

This field allows me to give more explanation about the news by inserting the picture which is related with the news in the website.

- TYPE:

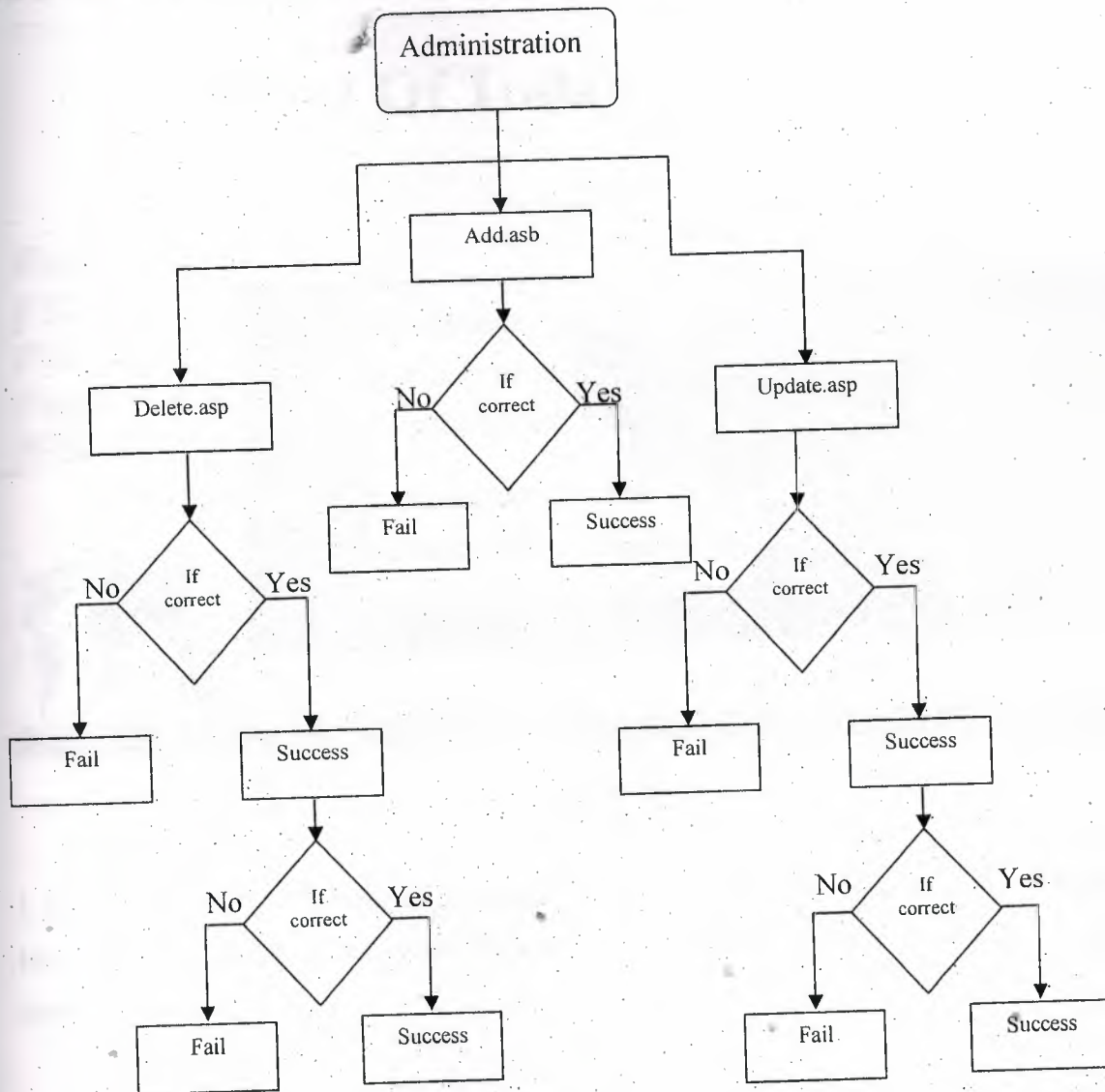
this field specify in which page that I can make these operations (ADD\_DATE, SUBJECT, BODY, PICTURE) only for example if I enter (number 1 )which corresponds to the home page in this field it means that all the values in the same row in special fields will display at the home page, and I code it as below

- 1 corresponds to home page
- 2 corresponds to economy page
- 3 corresponds to technology page
- 4 corresponds to sport page
- 5 corresponds to car page



## 5.4 Administration Page

As we told before the administration page is the first page that I constructed which gives me the ability to go in too many things in my project to become the control panel of my project. It has three main links, each link arrives you to specification page. In addition, the flowchart below explains what the construction of the home page.





## 5.5 Add page

This page is responsible for inserting data –news- to the web site. Each news has its own ID number, date, subject (the title of news), body (details of the news), type (what kind of news is this) and picture, as it shown below.

**World Of Today**

CONTENT MANAGER

Home  
Economy  
Technology  
Sports  
Cars

11/05/2015

ID	<input type="text"/>
DATE	<input type="text"/>
SUBJECT	<input type="text"/>
BODY	<input type="text"/>
TYPE	<input type="text"/>
Picture	<input type="text"/>
<input type="button" value="Insert Record"/>	

Copyright World Of Today 2005

Figure 2: add page

I used ODBC and ASP(using Jscript) techniques to link the data base with the web site, here there are the codes that is used to insert news into my database , and to be viewed later.

```
// *** Insert Record: construct a sql insert statement and execute it
```

```
if (String(Request("MM_insert")) != "undefined") {
```

```
// create the sql insert statement
```

```
var MM_tableValues = "", MM_dbValues = "";
for (var i=0; i+1 < MM_fields.length; i+=2) {
    var formVal = MM_fields[i+1];
    var MM_typesArray = MM_columns[i+1].split(",");
    var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
    var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
    var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
    if (formVal == "" || formVal == "undefined") {
        formVal = emptyVal;
    } else {
        if (altVal != "") {
            formVal = altVal;
        } else if (delim == "") { // escape quotes
            formVal = "" + formVal.replace(/'/g, "'") + "";
        } else {
            formVal = delim + formVal + delim;
        }
    }
    MM_tableValues += ((i != 0) ? "," : "") + MM_columns[i];
    MM_dbValues += ((i != 0) ? "," : "") + formVal;
}
MM_editQuery = "insert into " + MM_editTable + " (" + MM_tableValues + ") values ("
+ MM_dbValues + ")";

if (!MM_abortEdit) {
    // execute the insert
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
    MM_editCmd.ActiveConnection.Close();
}
```

```
if (MM_editRedirectUrl) {  
    Response.Redirect(MM_editRedirectUrl);  
}  
}  
  
%>  
<%
```

## 5.6 Remove and update pages

These two pages are connected with search page, which allows me to specify any information that I need to remove or update by calling the ID of the information itself. As shown in figure 3

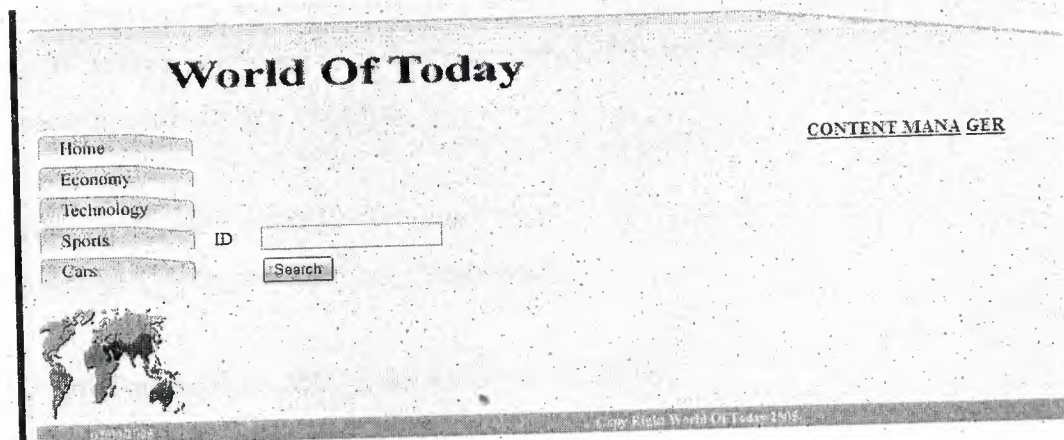


Figure 3: Search page

### 5.6.1 Remove page

it used for delete a record i.e. – a news – from the web site ,first of all you have to get the news ID number to be removed ,then: administrating page will search about that



record if it is founded it will be deleted ,then it will give you success operation, otherwise it will give fail operation as shown in figure 4.

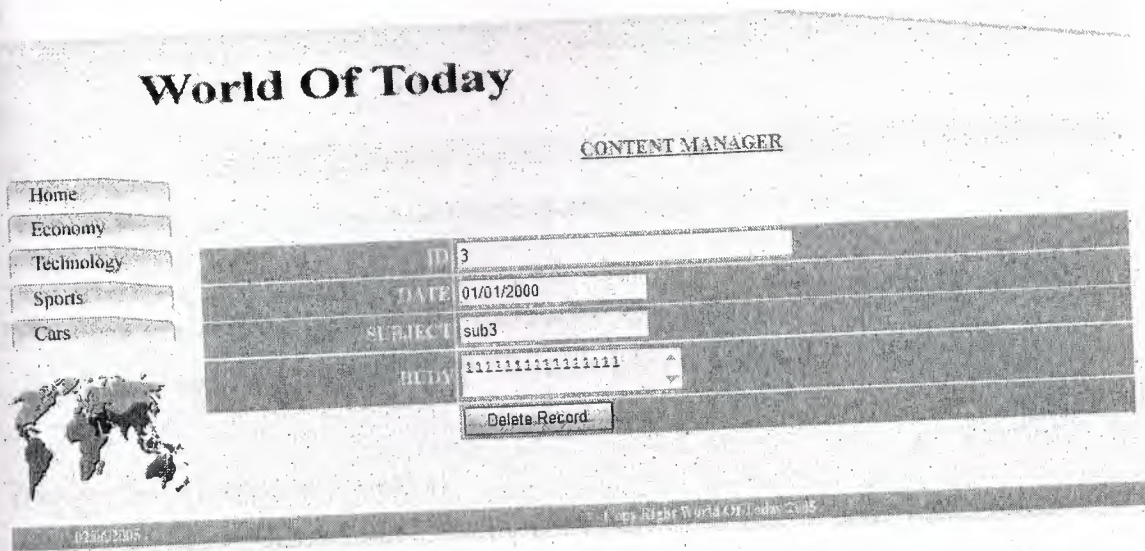


Figure 4: Remove page

By using the delete record code I can delete any news from my project

```
// *** Delete Record: declare variables
```

```
if (String(Request("MM_delete")) == "form1" &&  
    String(Request("MM_recordId")) != "undefined") {
```

```
    var MM_editConnection = MM_DatabaseConn_STRING;  
    var MM_editTable = "world_news";  
    var MM_editColumn = "ID";  
    var MM_recordId = "" + Request.Form("MM_recordId") + "";  
    var MM_editRedirectUrl = "success.htm";
```

```
// append the query string to the redirect URL  
if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.Count > 0) {
```

```
MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)? "?" : "&") +  
Request.QueryString;  
}  
}  
%>  
<%  
// *** Delete Record: construct a sql delete statement and execute it  
  
if (String(Request("MM_delete")) != "undefined" &&  
String(Request("MM_recordId")) != "undefined") {  
  
    // create the sql delete statement  
    MM_editQuery = "delete from " + MM_editTable + " where " + MM_editColumn + " = "  
+ MM_recordId;  
  
    if (!MM_abortEdit) {  
        // execute the delete  
        var MM_editCmd = Server.CreateObject('ADODB.Command');  
        MM_editCmd.ActiveConnection = MM_editConnection;  
        MM_editCmd.CommandText = MM_editQuery;  
        MM_editCmd.Execute();  
        MM_editCmd.ActiveConnection.Close();  
  
        if (MM_editRedirectUrl) {  
            Response.Redirect(MM_editRedirectUrl);  
        }  
    }  
}  
  
%>  
<%
```

```
var Recordset1__MMColParam = "1";
if (String(Request.QueryString("ID")) != "undefined" &&
    String(Request.QueryString("ID")) != "") {
    Recordset1__MMColParam = String(Request.QueryString("ID"));
}
%>
<%
var Recordset1 = Server.CreateObject("ADODB.Recordset");
Recordset1.ActiveConnection = MM_DatabaseConn_STRING;
Recordset1.Source = "SELECT * FROM world_news WHERE ID = "+
Recordset1__MMColParam.replace("/g, """) + "";
Recordset1.CursorType = 0;
Recordset1.CursorLocation = 2;
Recordset1.LockType = 1;
Recordset1.Open();
var Recordset1_numRows = 0;
%>
```



### 6.5.2 Update page

During this page, I can update any information in my database and the aim of this step to give me the ability to make any update in many kinds of information like (picture, date, titles, details news) as shown in Figure 5

**World Of Today**

CONTENT MANAGER

ID	1
DATE	05/05/2055
SUBJECT	my project
BODY	murad nayer alzu'bi
TYPE	4
Picture	add.asp

Copyright World Of Today 2005

Figure 6: Update page

and just I want to press your attention that by using the update record code which is shown below we can change any thing in my data base.

<%

// \*\*\* Update Record: set variables

```
if (String(Request("MM_update")) == "form1" &&
String(Request("MM_recordId")) != "undefined") {
```

```
var MM_editConnection = MM_DatabaseConn_STRING;
var MM_editTable = "world_news";
```

```
var MM_editColumn = "ID";
var MM_recordId = "" + Request.Form("MM_recordId") + "";
var MM_editRedirectUrl = "success.htm";
var MM_fieldsStr = "textfield1|value|textfield2|value|textfield3|value|textarea|value|textfield7|value|textfieldpicture|value";

varMM_columnsStr="ID|none,none,NULL|ADD_DATE|',none,NULL|SUBJECT|',none,"|
BODEY|',none,"|TYPE|none,none,NULL|PICTURE|',none,"";

// create the MM_fields and MM_columns arrays
var MM_fields = MM_fieldsStr.split("|");
var MM_columns = MM_columnsStr.split("|");

// set the form values
for (var i=0; i+1 < MM_fields.length; i+=2) {
    MM_fields[i+1] = String(Request.Form(MM_fields[i]));
}

// append the query string to the redirect URL
if (MM_editRedirectUrl && Request.QueryString && Request.QueryString.Count > 0) {
    MM_editRedirectUrl += ((MM_editRedirectUrl.indexOf("?") == -1)? "?" : "&") +
Request.QueryString;
}
}

%>
<%

// *** Update Record: construct a sql update statement and execute it

if (String(Request("MM_update")) != "undefined" &&
String(Request("MM_recordId")) != "undefined") {
```

```
// create the sql update statement
MM_editQuery = "update " + MM_editTable + " set ";
for (var i=0; i+1 < MM_fields.length; i+=2) {
    var formVal = MM_fields[i+1];
    var MM_typesArray = MM_columns[i+1].split(",");
    var delim = (MM_typesArray[0] != "none") ? MM_typesArray[0] : "";
    var altVal = (MM_typesArray[1] != "none") ? MM_typesArray[1] : "";
    var emptyVal = (MM_typesArray[2] != "none") ? MM_typesArray[2] : "";
    if (formVal == "" || formVal == "undefined") {
        formVal = emptyVal;
    } else {
        if (altVal != "") {
            formVal = altVal;
        } else if (delim == "") { // escape quotes
            formVal = "" + formVal.replace(/'/g, "'") + "";
        } else {
            formVal = delim + formVal + delim;
        }
    }
    MM_editQuery += ((i != 0) ? ", " : "") + MM_columns[i] + " = " + formVal;
}
MM_editQuery += " where " + MM_editColumn + " = " + MM_recordId;

if (!MM_abortEdit) {
    // execute the update
    var MM_editCmd = Server.CreateObject('ADODB.Command');
    MM_editCmd.ActiveConnection = MM_editConnection;
    MM_editCmd.CommandText = MM_editQuery;
    MM_editCmd.Execute();
    MM_editCmd.ActiveConnection.Close();
}
```



```
if (MM_editRedirectUrl) {  
    Response.Redirect(MM_editRedirectUrl);  
}  
  
}  
  
%>  
<%  
var Recordset1__MMColParam = "1";  
if (String(Request.QueryString("ID")) != "undefined" &&  
    String(Request.QueryString("ID")) != "") {  
    Recordset1__MMColParam = String(Request.QueryString("ID"));  
}  
%>  
<%  
var Recordset1 = Server.CreateObject("ADODB.Recordset");  
Recordset1.ActiveConnection = MM_DatabaseConn_STRING;  
Recordset1.Source = "SELECT * FROM world_news WHERE ID = "+  
    Recordset1__MMColParam.replace(/'/g, "") + "";  
Recordset1.CursorType = 0;  
Recordset1.CursorLocation = 2;  
Recordset1.LockType = 1;  
Recordset1.Open();  
var Recordset1_numRows = 0;  
%>
```

## REFERENCES

### References of Books

- [1] *HTML for the World Wide Web with XHTML and CSS: Visual QuickStart Guide*, Fifth Edition -- by Elizabeth Castro.
- [2] *How to Do Everything with HTML* -- by James H. Pence.
- [3] *HTML for Dummies* by Ed Tittel, Stephen J. James (. June 23, 1997).
- [4] *How to Do Everything with HTML* by James H. Pence
- [5] *Teach Yourself ODBC Programming in 21 Days* -- by Bill Whiting, et al; Paperback
- [6] *The ODBC Solution: Open Database Connectivity in Distributed Environments/Book and Disk* (Mcgraw-Hill Series on Computer Communications) -- by Robert Signore, et al;
- [7] *ODBC 3.5 Developer's Guide* by Roger E. Sanders
- [8] *Inside ODBC* (Microsoft programming series) by Kyle Geiger
- [9] *Beginning Active Server Pages 3.0* (Programmer to Programmer) -- by David Buser, et al; ASP in a Nutshell, 2nd Edition -- by A. Keyton Weissinger;
- [10] *Sams Teach Yourself Active Server Pages 3.0 in 21 Days* -- by Scott Mitchell, James Atkinson;

### References from Internet

- [http://www.katsueydesignworks.com/tutorials\\_databases.htm](http://www.katsueydesignworks.com/tutorials_databases.htm)
- <http://www.functionx.com/access/>
- <http://www.bcschools.net/staff/AccessHelp.htm>
- <http://cisnet.baruch.cuny.edu/holowczak/classes/2200/access/accessall.html>
- <http://www.w3schools.com/ado/default.asp>
- <http://www.tutorial-web.com/asp/database/printable.asp>
- [http://www.webwizguide.info/asp/tutorials/deleting\\_data\\_from\\_database.asp](http://www.webwizguide.info/asp/tutorials/deleting_data_from_database.asp)
- <http://www.simongibson.com/intranet/odbc/>
- <http://www.simongibson.com/intranet/odbc/>
- <http://www.sun/jscribt/tutorial.com>