

NEAR EASTERN UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

Web Site Design for Car Agent Company

Graduation Project
COMme 400

Student: Osama Alkurd (20021931)

Supervisor: Asst. Prof. Dr
Adil Amirjanov

Nicosia " 2006

ACKNOWLEDGMENT

First I want to thank Dr. Adil Amijanov. Under this guidance, I successfully overcome many difficulties and learn about programming, in each discussion he explained my questions patiently, and I felt progress from his advices. I asked him many questions in computer science and he always answered my questions in detail.

Special thanks to faculty of engineering for their helping me, to get good qualification that helps me in my life.

I also want to thank my friends those helped me for the past 4 years, special thanks to Mohammad Elfawair, Imad al dahdouh for their guidance.

Finally, I want to thank my family, especially my parents. Without their endless support and love for me, I would never achieve my current position. I wish my mother lives happily with my father in the heaven be proud for me.

ABSTRACT

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

According to the changeable events in car technology transmittance over the world, it has been chosen as quick intermedia. The Internet and the world of cars therefore in order to be able to have such an attractive for people over world for who are following the cars website, the design is set to satisfy the satisfactions of users so that the method of display the external design of car web page 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 VBSCRIPT gives ease for designer to design with high.level.of.techniques, .Tobe familiar with the designing fields. enough knowledge should be obtained about the following subjects HTML, ADO, and ASP, whereas ORACLE, SQL, ACCESS should be used to deal with database.

TABLE OF CONTENT

ACKNOWLEDGMENT..	i
ABSTRACT.....	ii
TABLE OF CONTENTS.....	iii
	vi
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 file.....	6
1.13 Text links.....	7
1.14 Image links.....	7
1.15 Tables.....	8
1.15.1 Drawing a table.....	9
1.15.2 Specifying table sizes.....	10
CHAPTER TWO: VBSCRIPT	
2.1 Introduction.....	12
2.2 Statements and Keywords.....	12

2.3 Operators.....	12
2.3 VBScript Functions....	13
2.3.1 Type Checking Functions.....	13
2.3.2 Value Constant Data Type.....	14
2.3.3 Type Casting Functions.....	14
2.3.4 Formatting Functions.....	15
2.3.5 Date Functions...	16
2.3.6 Date Constants.....	17
2.3.7 Day of the Week Constants.....	17
2.3.8 String Functions.....	18
2.3.9 Other functions.....	20
2.4 Control Structures.....	20
2.4.1 Looping Structures., ..	21
2.4.2 Branching Structures.....	23

CHAPTER 3: MICROSOFT ACCESS DATABASE

3.1 Introduction.....	24
3.2 Objects.....	25
3.3 Tables.....	26
3.3.1 Data Types.....	27
3.4 Primary Key	28
3.5 Entering Data.....	28
3.6 Field.....	29
3.7 Relationships	29
3.8 Forms	30
3.9 Reports	30

CHAPTER FURE: ASP AND ADO

INTRODUCTION

My project is about an interaction Car Agent 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 Car pages I decided to be done with display car agents; for car sales, cars rentals, the last car technologies, and according to be able to follow the modem cars from time to another I hade to use database which would enable me to add and improve and delete the cars thus in order to link the database with news pages I hade to use ADO 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 cars 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 vbscript, 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 & ADO 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 Car agent: here a main heading, there a sub heading, there bold text, there a list, ora line etc. Originally I-ITML 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 5 or 6 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 centering etc. cannot be represented in ASCH. 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 Car agent before a word to be printed bold you find the following symbol: , and after it this symbol : . B for bold. The switch-off command is similar to the switch-on command, with the addition of a slash /. So the Car agent for bold is written in HTML as follows: bold word group 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 htrn or html!

1.2 JHTML 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 ` ...` 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 `` tag as below:
` ...,` 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> Car Agent </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 Car Agent:

```
<HTML>
<HEAD>
<TITLE> Car Agent </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 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 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
 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 `
` 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"> <iP>
<P ALIGN="center"> </iP>
<P ALIGN="right"> <iP>
<HTML>
<HEAD>
<TITLE> Car Agent </TITLE>
</HEAD>
<BODY>
<P ALIGN="left"> You can align text in left, </iP>
<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 `` tag. Actually we will need parameters for this tag that specify image file location, file name and other optional parameters.

Look at this Car Agent:

Car Agent

```
<HTML>
```



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

In this Car Agent 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> Car Agent </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.
```

 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> tag to do this. As before we will need extra parameters from this tag. Look at example below:

```
<HTML>
<HEAD>
<TITLE> Car Agent </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 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 with an tag that displays an image file.

```
<HTML>
<HEAD>
<TITLE> Car Agent </TITLE>
</HEAD>
<BODY>
Click on below picture to visit my homepage. <BR><BR>
<A HREF="http://www.angelfire.com/nt/sannadys">
<IMG SRC="me.gif">
</A>
```

```
</BODY>
```

```
</HTML>
```

In above Car Agent clicking on picture will bring surfer to the address of `` 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> Car Agent </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 Car Agent 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> Car Agent </TITLE>
</HEAD>
<BODY>
<TABLE WIDTH=50% BORDER=1>
<TR>
<TD>Cell Row 1 Col 1</TD>
<TD>Cell Row1 Col2</TD>
</TR>
<TR>
<TD>Cell Row2 Col 1</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 VBSCRIPT

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 VBScript. The beautiful thing about VBScript is that the knowledge and system-related prerequisites for learning the language are relatively low. You just need to know some HTML.

This section covers the VBScript keywords, operators, functions, and control structures.

2.2 Statements and Keywords

Dim is used to declare variables, VBScript variables are variants, which means that they do not have to have a fixed data type. Const is used to declare constants, which are like variables except that they cannot be changed in the script. Option Explicit is put at the top of a page to force explicit declaration of all variables.

2.3 Operators

In order of precedence, this is a list of operators supported in VBScript.

- Anything in parentheses.
- Exponentiation (^)
- Negation (-)
- Multiplication, Division (*, /)
- Integer Division (\)
- Modulus (Mod)
- Addition, Subtraction (+, -)
- String Concatenation (&)
- Comparison Operators (=, <, <<, >, >>, <=, >=)

- Not
- And
- Or
- Xor
- Eqv
- Imp

2.3 VBScript Functions

This will provide you with a quick look at the more important VBScript functions. They include functions for type checking, typecasting, formatting, math, date manipulation, string manipulation, and more.

2.3.1 Type Checking Functions

These functions allow you to determine the data subtype of a variable or expression.

VarType (expression) returns an integer code that corresponds to the data type.

TypeName (expression) returns a string with the name of the data type rather than a code.

IsNumeric (expression) returns a Boolean value of True if the expression is numeric data, and False otherwise.

IsArray (expression) returns a Boolean value of True if the expression is an array, and False otherwise.

IsDate (expression) returns a Boolean value of True if the expression is date/time data, and False otherwise.

IsEmpty (expression) returns a Boolean value of True if the expression is an empty value (uninitialized variable), and False otherwise.

IsNull (expression) returns a Boolean value of True if the expression contains no valid data, and False otherwise.

IsObject (expression) returns a Boolean value of True if the expression is an object, and False otherwise.

2.3.2 Value Constant Data Type

Table (2.1) Value Constant Data Type

Value	Constant	Data Type
0	VbEmpty	Empty (This is the type for a variable that has not been used yet. In other words, Empty is the default datatype.)
1	VbNull	Null (No valid data)
2	VbInteger	Integer
3	VbLong	Long
4	VbSingle	Single
5	VbDouble	Double
6	VbCurrency	Currency
7	VbDate	Date
8	VbString	String
9	VbObject	Object
10	VbError	Error
11	VbBoolean	Boolean
12	VbVariant	Variant (used with VbArray)
13	VbDataObject	Data Access Object
14	VbDecimal	Decimal
17	VbByte	Byte
8192	VbArray	Array (VBScript uses 8192 as a base for arrays and adds the code for the data type to indicate an array. 8204 indicates a variant array, the only real kind of array in VBScript.)

2.3.3 Type Casting Functions

Typecasting allows you to convert between data subtypes.

- **CInt(expression)** casts expression to an integer. If expression is a floating-point value or a currency value, it is rounded. If it is a string that looks like a number, it is turned into that number and then rounded if necessary. If it is a Boolean value of true, it becomes -1. False becomes 0. It also must be within the range that an integer can store.
- **CByte(expression)** casts expression to a byte value provided that expression falls between 0 and 255. expression should be numeric or something that can be cast to a number.
- **CDbl(expression)** casts expression to a double, expression should be numeric or something that can be cast to a number,

- . CSng(expression) casts expression to a single. It work.s like CDbl(), but must fail within the range represented by a single.
- . CBool(expression) casts expression to a Boolean value. If expression is zero, the result is False. Otherwise, the result is True. Expression should be numeric or something that can be cast to a number.
- . CCur(expression) casts expression to a currency value, expression should be numeric or something that can be cast to a number.
- : CDate(expression) casts expression to a date value, expression should be numeric or something that can be cast to a number, or a string of a commonly used date format. DateValue(expression) or TimeValue (expression) can also be used for this.
- : CStr(expression) casts expression to a string, expression can be any kind of data.

2.3.4 Formatting Functions

FormatDateTime(expression, format) is used to format the date/time <lata in expression. format is an optional argument that should be one of the following:

- . vbGeneralDate Display date, if present, as short date. Display time, if present, as long time. Value is 0. This is the default setting if no format is specified,
- : vbLongDate Display date using the server's Jong date format. Value is 1.
- : vbShortDate Display date using the server's short date format. Value is 2.
- : vbl.ong'Time Display time using the server's long time format. Value is 3.
- : vbShortTime Display time using the server's short time forinat. Value is 4.

FonnatCurrency (value, numdigits, leadingzero, negparen, delimiter) is used to format the rmonetary value specified by value.

- . numdigits specifies the number of digits after the decimal place to display. -1 indicates to use the system default.
- Tristate options have three possible values. If the value is -2, it means use the system default. If it is -1, it means turn on the option. If it is 0, tum off the option.
- leadingzero is a Tristate option indicating whether to include leading zeroes on values less than 1.
- negparen is a Tristate option indicating whether to enclose negative values in parentheses.
- delimeter is a Tristate option indicating whether to use the delimiter specified in the computer's settings to group digits.

FormatNumber is used to format numerical values. It is almost exactly like FormatCurrency, only it does not display a dollar sign.

FormatPercent works like the previous two. The options are the same, but it turns the value it is given into a percentage.

2.3.5 Date Functions

: Date returns the current date on the server.

: Time returns the current time on the server.

: Now returns the current date and time on the server.

: DateAdd(interval, number, date) is used to add to the date specified by date. Interval is a string that represents whether you want to add days, months, years, and so on. Number indicates the number of intervals you want to add; that is, the number of days, months, years, and so on.

-. DateDiff(interval, date1, date2, firstDOW, firstWOY) is used to find the time between two dates, DateDiff returns the number of intervals elapsed between date1 and date2. The optional integer firstDOW specifies what day of the week to treat as the first. The optional firstWOY specifies which week of the year to treat as the first.

: DateSerial(year, month, day) takes the integers year, month, and day and puts them together into a date value. They may be negative.

: TimeSerial(hour, minute, second) is similar to DateSerial. TimeSerial returns the number of seconds elapsed since midnight.

-. DatePart(interval, datetime, firstDOW, firstWOY) allows you to retrieve the part of datetime specified by interval. The optional integer firstDOW specifies what day of the week to treat as the first. The optional firstWOY specifies which week of the year to treat as the first,

2.3.6 Date Constants

Tabfo (2.2)

Value	Meaning
yyyy	Year
"q"	Quarter
"m"	Month
"y"	Day of year
"D"	Day
"d"	Weekday
"w"	Week of year
"h"	Hour
"i"	Minute
"s"	Second

2.3.7 Day of the Week Constants

Table (2.3) Day of the Week Constants

0	vbUseSystem	National Language Snpport API Setring
1	vbSunday	Sunday (default)
	vb'Monday	Monday
3	vb'Tuesday	Tuesday
4	vb'Wednesday	Wednesday
5	vb'Thursday	Thursday
6	vb'Friday	Friday
7	vb'Saturday	Saturday
	vbUseSystem	National Language Snpport API Setting
	vbFirstJan1	Week of January 1
	vbFirstFourDays	First week with four days of new year
	vbFirstFullWeek	First full week

- Year(date) returns the year portion from date as a number.
- JMonth(date) returns the month portion from date as a number.
- MonthName(date) returns the month portion from date,
- Day(date) returns the day portion from date as a number.
- Weekday(date) returns the day of the week of date as a number.
- Hour(time) returns the hour portion from time.
- Minute(time) returns the minute portion from time.
- Second(time) returns the second portion from time.

2.3.8 String Functions

- UCase(string) returns string with all its lowercase letters converted to uppercase letters.
- LCase(string) returns string with all its uppercase letters converted to lowercase letters.
- LTrim(string) removes all the spaces from the left side of string.
- RTrim(string) removes all the spaces from the right side of string.
- Trim(string) removes spaces from both the left and the right sides.
- Space(number) returns a string consisting of number spaces.
- String(number, character) returns a string consisting of character repeated number times.
- Len(string) returns the number of characters in string.
- Len(variable) returns the number of bytes required by variable.
- LenB(string) returns the number of bytes required to store string.
- StrReverse(string) returns string with the characters in reverse order.
- StrComp(string1,string2,comparetype) is used to perform string comparisons. If comparetype is zero or omitted, the two strings are compared as if uppercase letters come before lowercase letters. If comparetype is one, the two strings are compared as if upper and lowercase letters are the same. StrComp returns -1 if string 1 is less than string2. It returns 0 if they are the same, and 1 if string 1 is greater than string2.
- Right(string,number) returns the number rightmost characters of string.
- RightB(string,number) works like Right, but number is taken to be a number of bytes rather than characters.
- Left(string,number), as you may guess, returns the number leftmost characters of string.
- LeftB(string,number) works like Left, but number is taken to be a number of bytes rather than characters.
- Mid(string,start,length) returns length characters from string, starting at position start. When length is greater than the number of characters left in the string, the rest of the string is returned. If length is not specified, the rest of the string starting at the specified starting position is returned.
- MidB(string,start,length) works like Mid, but start and length are both taken to be byte numbers rather than character numbers.

`InStr(start,string1,string2,comparetype)` is used to check if and where `string2` occurs within `string1`. `start` is an optional argument that specifies where in `string1` to start looking for `string2`. `comparetype` is an optional argument that specifies which type of comparison to perform. If `comparetype` is 0, a binary comparison is performed, and uppercase letters are distinct from lowercase letters. If `comparetype` is 1, a textual comparison is performed, and uppercase and lowercase letters are the same. `InStr` returns zero if `string1` is empty (""), if `string2` is not found in `string1`, or if `start` is greater than the length of `string2`. It returns Null if either string is Null. It returns `start` if `string2` is empty. If `string2` is successfully found in `string1`, it returns the starting position where it is first found.

-. `InStrB` works like `InStr` except that the start position and return value are byte positions, not character positions.

-. `InStrRev(string1,string2,start,comparetype)` starts looking for a match at the right side of the string rather than the left side. `start` is by default -1, which means to start at the end of the string.

-. `Replace(string,find,replace,start,count,comparetype)` is used to replace occurrences of `find` with `replace` in `string`. `start`, `count`, and `comparetype` are optional, but if you want to use `replace`, you must use the ones that come before it. `start` indicates where the resulting string will start and where to start searching for `find`. It defaults to 1. `count` indicates how many times to perform the replacement. By default, `count` is -1, which means to replace every occurrence. If `comparetype` is 0, a binary comparison is performed, and uppercase letters are distinct from lowercase letters. If `comparetype` is 1, a textual comparison is performed, and uppercase and lowercase letters are the same.

-. `Filter(arrStrings,SearchFor,include,comparetype)` searches an array of strings, `arrStrings`, and returns a subset of the array. `include` is a Boolean value. If `include` is `True`, `Filter` searches through all the strings in `arrStrings` and returns an array containing the strings that contain `SearchFor`. If `include` is `False`, `Filter` returns an array of the strings that do not contain `SearchFor`. `include` is optional and defaults to `True`. `comparetype` works the same as in the other string functions we have discussed. If you want to use `comparetype`, you must use `include`.

-. `Split(expression,delimiter,count,comparetype)` takes a string and splits it into an array of strings. `expression` is the string to be split up. If `expression` is zero length, `Split` returns an array of no elements. `delimiter` is a string that indicates what is used to

separate the sub-strings in expression. This is optional; by default the delimiter is the space. If delimiter is zero length (""), an array of one element consisting of the whole string is returned, count is used to specify a maximum number of sub-strings to be created. The default for count is -1, which means no limit. If comparetype is 0, a binary comparison is performed, and uppercase letters are distinct from lowercase letters. If comparetype is 1, a textual comparison is performed, and uppercase and lowercase letters are the same. comparetype is only useful when the delimiter you have chosen is a letter.

Jointstringarray.delimiter) does just the opposite of Split. It takes an array of strings and joins them into one string, using delimiter to separate them. delimiter is optional; the space is the default.

2.3.9 Other functions

: LBound(array) returns the smallest valid index for array.

: UBound(array) returns the largest valid index for array.

: Asc(string) returns the ANSI character code for the first character of string.

-. Chr(integer) returns a string consisting of the character that matches the ANSI character code specified by integer.

-. Array(value1, value2, ..., valueN) returns an array containing the specified values. This is an alternative to assigning the values to array elements one at a time.

2.4 Control Structures

Control structures allow you to control the flow of execution of your scripts.

You can specify that some code should be executed only under certain circumstances, using conditional structures. You can specify that some code should be executed repeatedly, using looping structures. Lastly, you can specify that code from somewhere else in the script should be executed using branching controls.

Conditional Structures

The If..Then ...Else construct allows you to choose which block of code to execute based on a condition or series of conditions.

<%


```

If condition1 Then
    codeblock1
Elseif condition2 Then
    codeblock2
Else
    codeblock3
End If
%>

```

If condition1 is true, codeblock1 is executed. If it is false, and condition2 is true, codeblock 2 is executed. If condition1 and condition2 are both false, codeblock3 executes. An If-Then construct may have zero or more Elseif statements, and zero or one Else statements.

In place of some really complex If ...Then constructs, you can use a Select Case statement. It takes the following form:

```

Select Case variable
    Case choice 1
        codeblock1
    Case choice2
        codeblock2
    Case choicen
        codeblockn
    Case default
        default code block
End Select

```

This compares the value of variable with choice1, choice2, and so on. If it finds a match, it executes the code associated with that choice. If it does not, it executes the default code.

2.4.1 Looping Structures

Looping structures allow you to execute the same block of code repeatedly. The number of times it executes may be fixed or may be based on one or more conditions.

The For ...Next looping structure takes the following form:

For counter = start to stop

codeblock

Next

codeblock is executed with counter having the value start, then with counter having the value start+1, then start+2, and so forth through the value stop.

Optionally, you may specify a different value to increment counter by. In this case the form looks like this:

For counter = start to stop Step stepvalue

codeblock

Next

Now counter will take the values start+stepvalue, start+stepvalue+stepvalue, and so forth. Notice that if stepvalue is negative, stop should be less than start.

The For Each ...Next looping structure takes the following form:

For Each item In Set

codeblock

Next

codeblock is executed with item taking the value of each member of Set. Set should be an array or a collection.

The Do While-Loop looping structure has the following form:

Do While boolean Value

code block

Loop

codeblock is executed as long as boolean Value is True. If it is False to begin with, the loop is not executed at all.

While ...Wend looping structure has the following form:

While boolean Value

codeblock

Wend

codeblock is executed as long as boolean value is True. If it is False to begin with, the loop is not executed at all.

The Do-Loop While looping structure has the following form:

Do code block

Loop While boolean Value

codeblock is executed as long as booleanvalue is True. The loop is executed at least once no matter what.

The Do Until-Loop looping structure has the following form:

Do Until boolean Value

codeblock

Loop

code block is executed as long as booleanValue is false. If it is true to begin with, the loop is not executed at all.

The Do...Loop Until looping structure has the following form:

Do code block

Loop Until boolean Value

code block is executed as long as booleanValue is false, The loop is executed at least once no matter what.

2.4.2 Branching Structures

Branching structures allow you to jump from one position in the code to another. A subroutine does not return a value. It simply executes. Subroutines look like this:

Sub name (argumentlist)

code block

End Sub

Functions do return values and have the following form:

Function name (argumentlist)

code block

name = expression

End Function

CHAPTER3

MICROSOFT ACCESS DATABASE

3.1 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*, and *Data-type*. Here is the Hierarchy that Microsoft Access uses in breaking down a database.

Database File

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

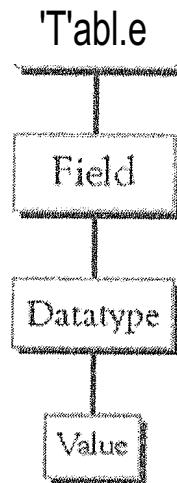


Figure 3.1

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

3.2 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.

3.3 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:

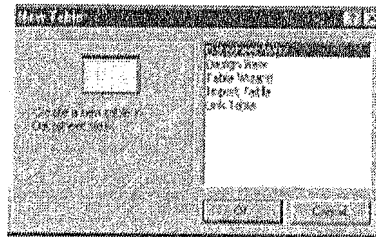


Figure (3.3) New Table

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:

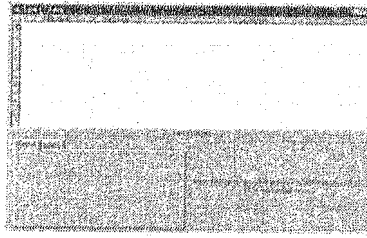


Figure (3.4) Tables Datatype

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

3.3.1 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).

*t.1.v<hfh~ld in thJ Staden Ruj\Jttt-;>ta.f:ita Jontai~J\$ Hut sr:inw (lr
 m:liio:uJ~t- l;~ar.p r;ün;-lul Hith ası yhl'd~ru-) So:ial
 s,,l,1 #J 1'1IM l; m e:n~ml;it t\A:1 COLU!vl.l\J 1\~i7\Jllt;t;*

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 row is a record in a table. Each row contains all of the information about a student: name, last name, birthday, address, and city. The first row is an example of a ROW.

Figure (3.3)

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.

3.5 Entertng 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:



Figure (3.4) Entering Data

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

3.6 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
- e 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.

3.7 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.

3.8 Forms

A form is nothing more than a graphical representation of a table. You can add, update, and 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.

3.9 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 ADO

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 VBScript.

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, and no active pages.

As ASP was first introduced by Microsoft on its 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 HS or its micro version Personal Web Server (PWS) comes free with Windows.

4.2 What 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 Using ADO objects to connect to databases?

Dim conuection

Dim conuection string

Set connection = Server. CreateObject("ADODB. Connection ")

connection. Openiconuectiou string)

execute the sql statements

connection.c/ose

Set connection=Nothing

4.4 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 '*webdatabases*' 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*'.

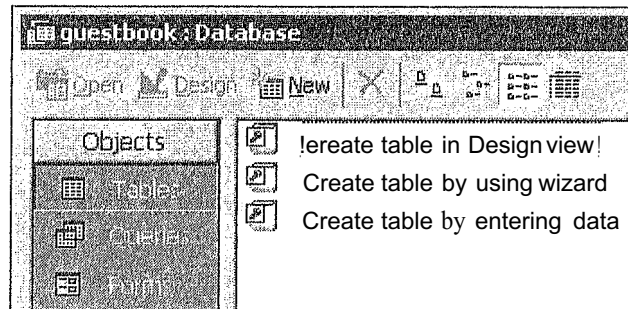


Figure (4.1) Creating Database.

You now need to create 6 tables for the database and select their data types.

Table 1 needs to be called 'admin': This table contains two fields, and also has the data type of each field.

Table 2 needs to be called 'car' and contains six fields. The 'carid' field defines as Primary Key,

Table 3 needs to be called 'Customer' and has 8 fields; the 'cusid' field defines as Primary Key.

Table 4 needs to be called 'rent' and has 5 fields; the 'carid' field defines as Primary Key.

Table 5 needs to be called 'rentcars' and has 7 fields; the 'carid' field defines as Primary Key.

Table 6 needs to be called 'sales' and has 3 fields; the 'carid', 'cusid', and 'date' field defines as Primary Key.

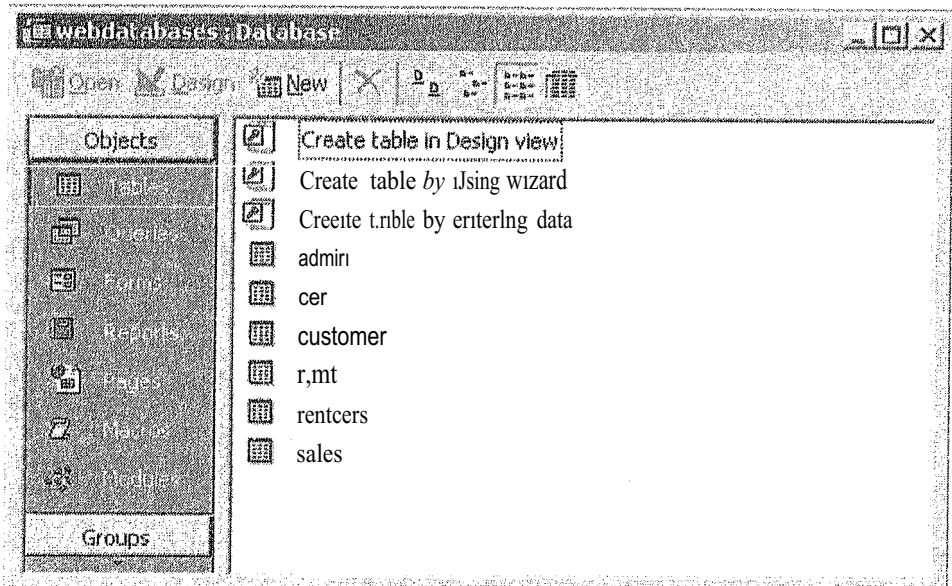


Figure (4.2) Databases Tables

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 any table in the main dialog box. From here you can enter some test data,

4.5 Creating an HTML Page to take Customer Registrations

For example (customer registrations page)

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 8 text boxes. which will add the user input into the database.

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title--Customer Registration-->
</head>
<body>
<form method="post" action="<%=MM_edit/vction'X%" name="form1" >
```

```

<table align="center">
  <tr valign="baseline">
    <td nowrap align="right">Address:</td>
    <td><input type="text" name="laddress" value="" size="32">
  </td>
</tr>
<tr valign="baseline">
  <td nowrap align="right">Cuspassword:</td>
  <td><input type="password" name="cuspassword" value="" size="32">
</td>
</tr>
<tr valign="baseline">
  <td nowrap align="right">Custid:</td>
  <td><input type="text" name="custid" value="" size="32">
</td>
</tr>
<tr valign="baseline">
  <td nowrap align="right">Email:</td>
  <td><input type="text" name="cmail" value="" size="32">
</td>
</tr>
<tr valign="baseline">
  <td nowrap align="right">Fname:</td>
  <td><input type="text" name="fname" value="" size="32">
</td>
</tr>
<tr valign="baseline">
  <td nowrap align="right">License:</td>
  <td><input type="text" name="license" value="" size="32">
</td>
</tr>

```

```

<tr valign="top">
  <td nowrap align="right">Lname:</td>
  <td><input type="text" name="lname" value="" size="32">
  </td>
</tr>
<tr valign="top">
  <td nowrap align="right">Phone:</td>
  <td><input type="text" name="phone" value="" size="32">
  </td>
</tr>
<tr valign="top">
  <td nowrap align="right">&nbsp;</td>
  <td><input type="submit" value="Insert record">
  </td>
</tr>
</table>
<input type="hidden" name="MM_insert" value="form1">
</form>
<p>&nbsp;</p>
</body>
</html>

```

And there are more pages like (insert, update, and delete) contents similar codes.

CHAPTER5

CAR AGENT WEB SITE 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 System design of the web site

The following figure illustrates the system design of the web site design with their database relationships

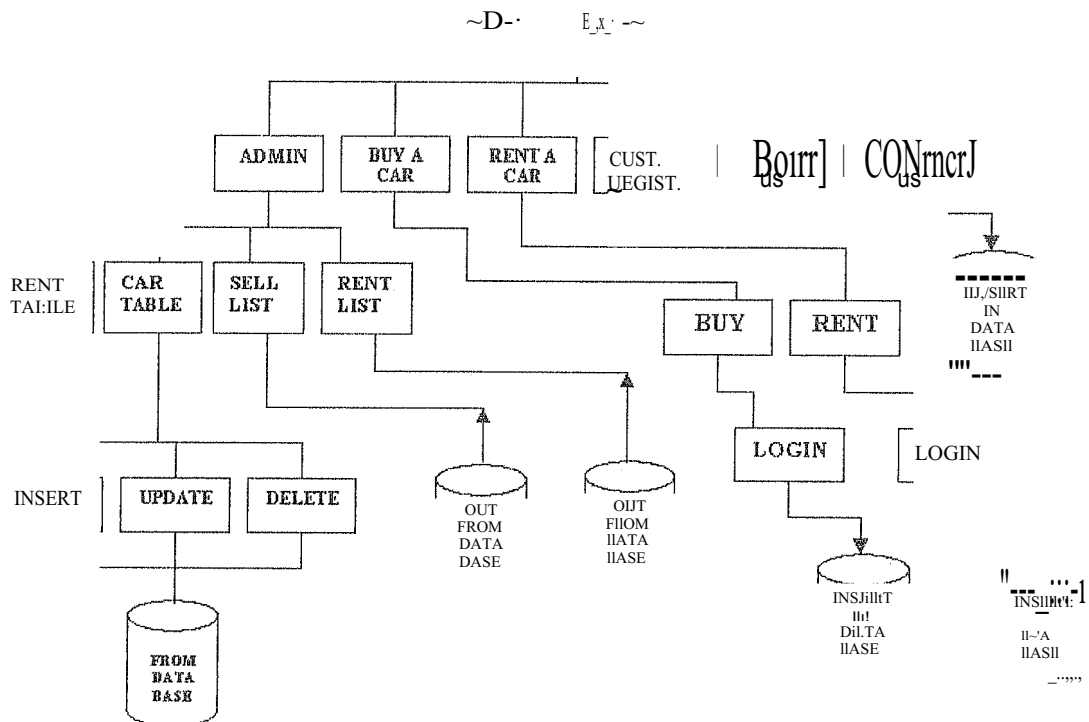


Figure (5.1) Block Diagram

5.3 Database design of the system

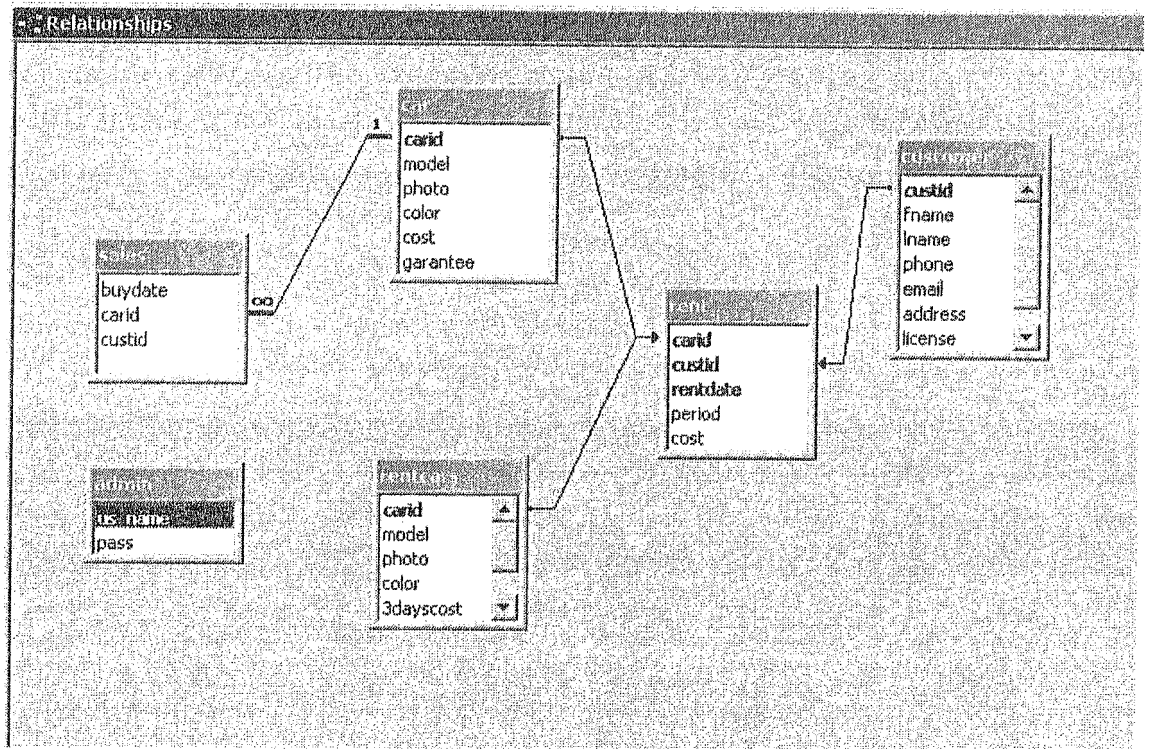


Figure (5.2) Relationships

5.4 Database objects of the web page (Tables)

admin Table	
us name	pass
abuanaas	20021931

Figure (5.3) Administrator Table

car : Table						
	carid	model	photo	color	cost	garantee
+	bmw	bmwz3	E:\karashek\picture\Car	red	€ 3,000.00	3
+	eeewfw	bmw	C:\Documents and Sett	red	€ 30,000.00	3
+	fds	mkjkh	C:\Documents and Sett	nbj	€ 144,242.00	3
+	hdsgbv	vw	C:\Documents and Sett	nbhj	€ 411.00	5
+	ioio	mercedes	C:\Documents and Sett	red	€ 300,000.00	3
+	qqq	dfggd	C:\Documents and Sett	dfgd	€ 45,354.00	2
+	rerer	bmw	C:\Documents and Sett	red	€ 30,000.00	3
+	rtr	bmwz8	D:\Media\Pictures\Spec	red	€ 30,000.00	3
*					€ 0.00	0

Figure (5.4) Car Table

customer : Table								
	custid	fname	lname	phone	email	address	license	cuspassword
+	1	ali	mhamd					200
+	100	123	123	123	123	add	123	111
+	123	hjhju	hjhjhj	154151	jbkhgkldhb		12	
+	444	knknk	knknkj	15151	kbkhbuk	khbyhvbyh	12	
+	7896788	7678678	57	1254	67676	67867	77	8888
+	bcbv	cvbcb	cvbcb	11651	fsdfsdfs	kuuyk	12	
+	fsdfs	dsdfs	sdfsf	fsdfs	fsdfsfs	kyukuy	ds	
+	fsdfs	fsdfs	sdfsfs	115115	fsdfsdfs	,m,lkml	sf	
+	kjk	sdsv	sdsv	5215	,klmkm	kmikmg	12	
+	lmm	jnlkjnk	kjnkknk	515	ljnlkj	ljhinii	12	
+	sadad	asdada	mghkg	21515	gjhghjhj	sdfsf	12	
+	sdfsd	fsdf	sdfsdf	15151	lgmgldf	yukyk	lm	
+	try	sdfsdf	sdkjfn	svkvs	fsdfs	l,l,l	ll	
*								

Figure (5.5) Customer Table

rent : Table					
	carid	custid	rentdate	period	cost
	1		5/27/2006	4	€ 60.00
	1	1	5/27/2006	2	€ 20.00
	1	1	5/28/2006	4	€ 60.00
	1	1	5/29/2006	10	€ 3,230.00
▶				0	€ 0.00

Figure (5.6) Rent Table

rentcars : Table						
	carid	model	photo	color	3dayscost	7dayscost
+	1	12		YELLOW	€ 10.00	\$15.00
▶					€ 0.00	\$0.00

Figure(5.7) Rentcar 'table

sales : Table			
	buydate	carid	custid
	5/27/2006	bmw	1
	5/27/2006	rerer	1
	5/28/2006	rerer	1
	5/29/2006	rerer	1

Figure (5,8) Car Table

5.5 THE APPLICATION OF MY J>ROJECT

My project contain four main website pages besides the home page that contains six main links belongs to administrator, buy a car, rent a car, customer registration, contact us and about us.

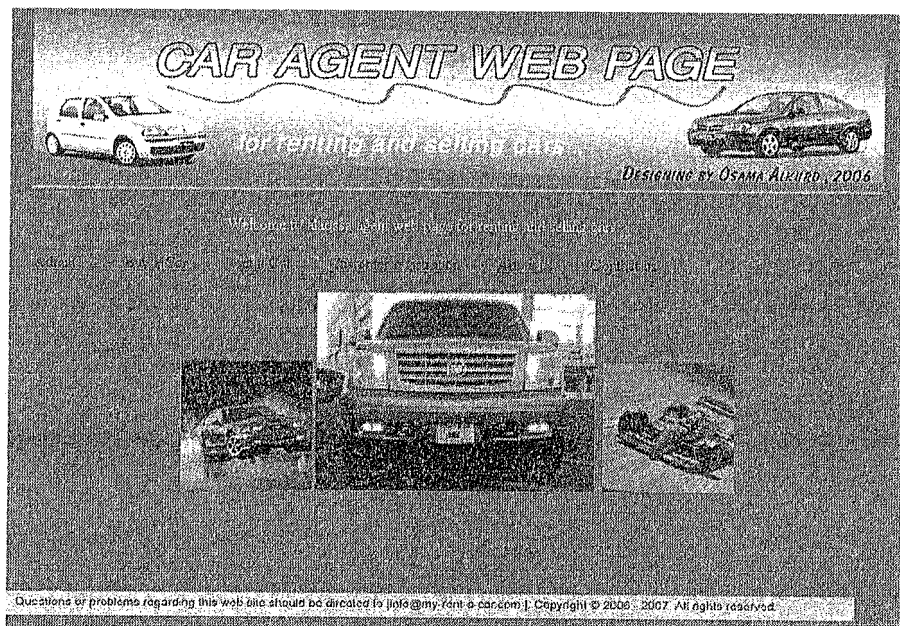


Figure (5.9) Index Page

5.5.1 Administrator design

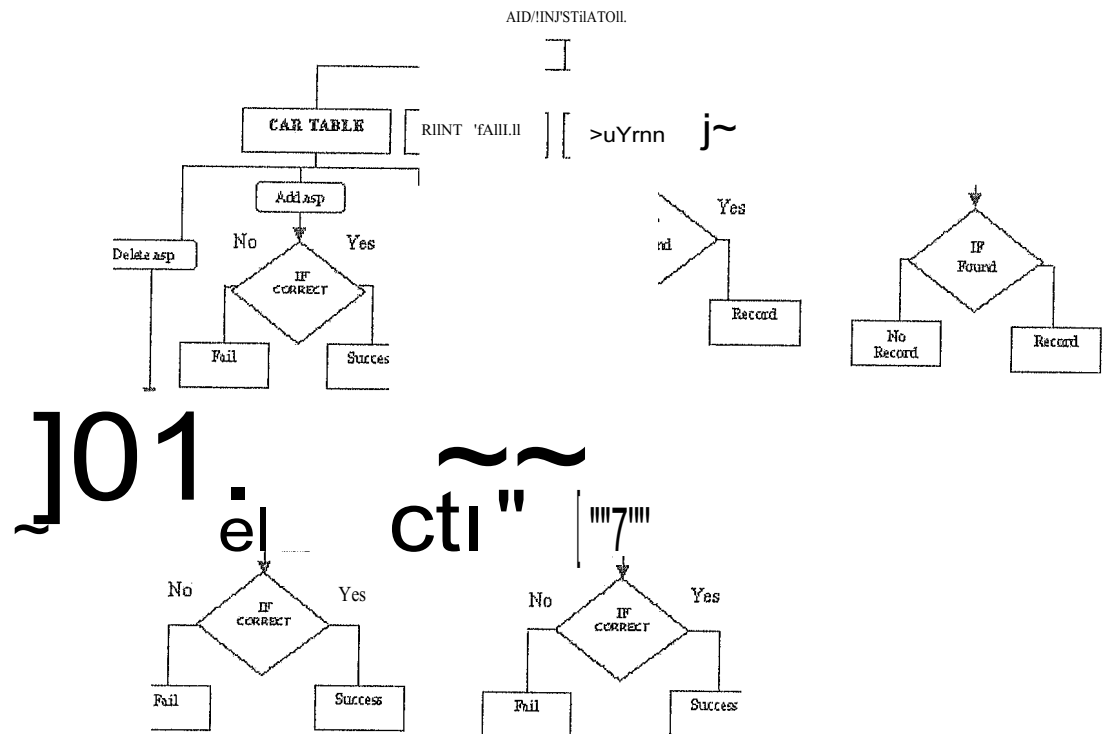


Figure (5.10) Admin Block Diagram

The first main page is the administrator page that contains username and password for the administrator.

After you login successfully, directly you will go to main admin page that contain four main links, sell car, rent car, today rent car list (report), today sales car list (report).



Figure (5.11) Main Admin Page

Sell car web page contains three options for insert, update or delete a car.

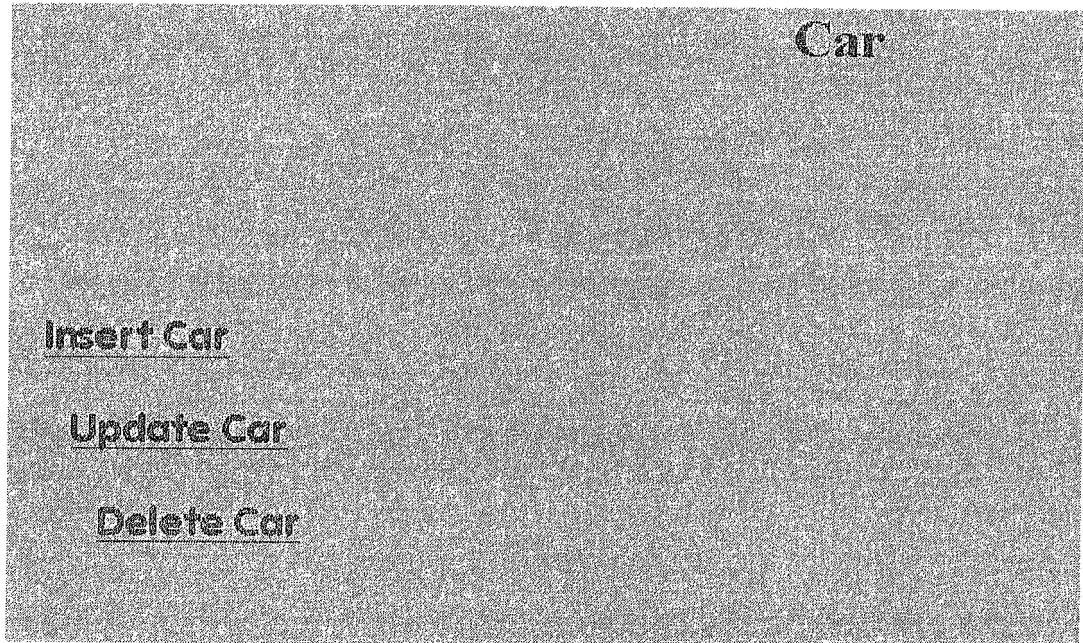


Figure (5.12) Car Page

Rent car also has same three options (insert, update, delete).

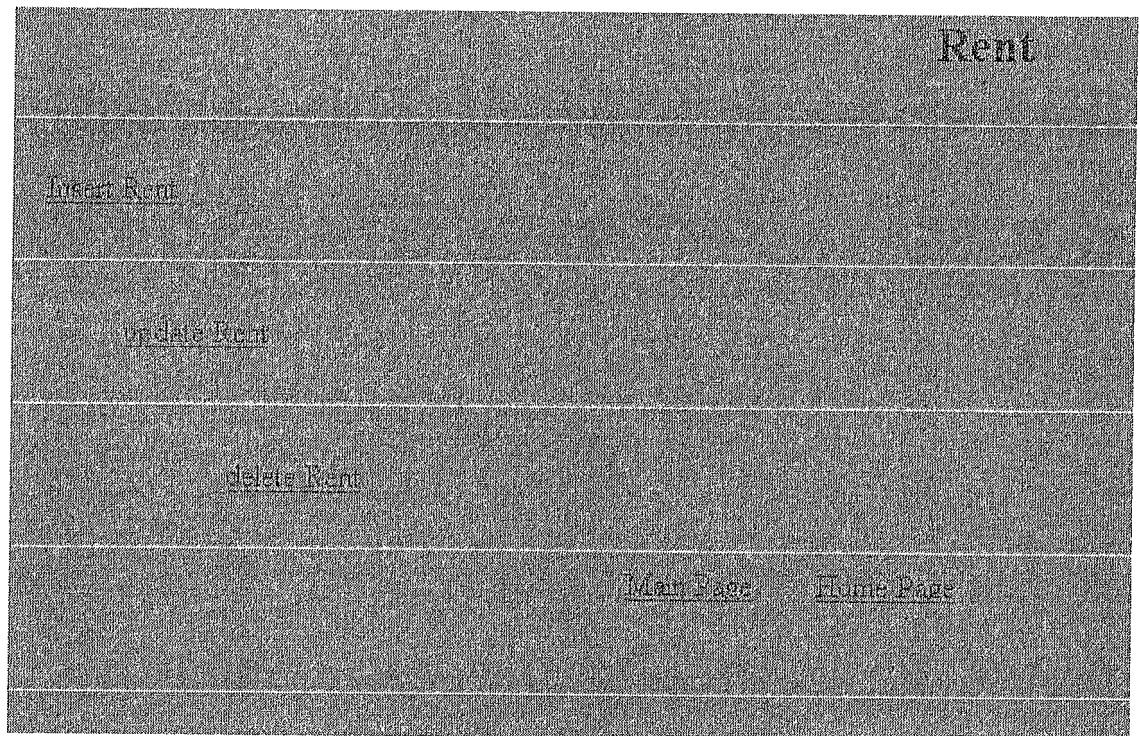


Figure (5J2) Rent Page

The other two pages show me daily reports for the rented and sold cars,

TODAY RENT LIST

Address <http://localhost/rentlist.asp>

TODAYKENT 11ST


CAR ID	COST	PERIOD	CUSTOMER ID
1	30	3	1

[Main Page](#)

[Home Page](#)

Figure (5.13) Rent List Page

TODAY SELL LIST

Address  http://localhost/saleslist.asp

TODAY SIJ-EUST

re{ } M


rerer J

[Main Page](#)

[Home Page](#)


Figure (5.14) Car List Page

The second main page is buy a car web page that has carid, model, photo, cost, and guarantee and buy button.


PHOTO	COLOR	COST	GARANTEE	BUY
	red	30000	3	Buy

bmw

---Ji..

PHOTO	COLOR	COST	GARANTEE	BUY
	red	30000	3	Buy

bmw

PHOTO	COLOR	COST	GARANTEE	BUY
	red	3000	3	Buy

bmw

bmwz3

Figure (5.15) Buy Customer Page

When you press buy button, if you are register before directly you will find the company account as in the figure,

Else have to register username and password before then you will go back to buy a car web page.

Username	
Password	
<input type="button" value="Login"/>	

Figure (5.16) Login Page

The third main page is the rent a car web page that contains a table with eight columns and one button belongs to the record of cars for renting, the eight columns are carid, model, photo, color, three days cost, seven days cost, more cost, period and the button rent,


CAR ID	MODEL	PHOTO	COLOR	3 DAYS COST	7 DAYS COST	MORE COST	PERIOD	RENT
1	12		YELLOW	10	15	323	3	<input type="button" value="Rent"/>

Figure (5.17) Rent Customer Page

And as it is in the buy page you have register before renting.

The fourth main page is customer registration where customers have to register before doing any transactions, and it has eight fields, the customer has to fill them, address, cuspassword, custid, email, fname, license, lname, phone,

Address:	<input type="text"/>
Cuspassword:	<input type="text"/>
Custid:	<input type="text"/>
Email:	<input type="text"/>
Friarne:	<input type="text"/>
Licrise:	<input type="text"/>
Lname:	<input type="text"/>
Phone:	<input type="text"/>
<input type="button" value="Insert record"/>	

Figure (5.18) Customer Registration Page

CONCLUSIONS

The Internet web site has become one of the most important things in the life. the Internet has entered mostly all fields; politics, education, economy, health, industrial etc.

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

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

This project is about an interaction Car Agent 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 Car pages I decided to be done with display car agents; for car sales, cars rentals, the last car technologies, and according to be able to follow the modern cars from time to another I had to use database which would enable me to add and improve and delete the cars thus in order to link the database with news pages I had to use ADO system which throw it I could make my WebPages interactive during

REFERENCES

References of Books

- [1] *HTML for the World Wide Web with XHTML and CSS: Visual Quick Start Guide*, Fifth Edition -- by Elizabeth Castro.
- [2] *How to Do Everything with HTML* -- by James H. Pence.
- [3] *HTML for Dummies* by Ed Tittle, Stephen J. James (. June 23, 1997)
- [4] *How to Do Everything with HTML* by James H. Pence
- [5] *Beginning Active Server Pages 3.0 (Programmer to Programmer)* -- by David Buser, et al: .A..SPiitaNutshell 2nd Edition -- by A. Keyton Weissinger;
- [6] *Sams Teach Yourself Active Server Pages 3.0 in 21 Days* -- by Scott Mitchell, James Atkinson.
- [7] *VBScript in a Nutshell*, by Paul Lomax, Matt Childs, and Ron Petruska
- [8] *A Practical Guide to Microsoft Active Server Pages 3.0*, By Manas Tungare

References from Internet

- 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