



NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

WEB DESIGN FOR TRAFFIC ACCIDENT

COM 400 PROJECT

MUSTAFA ŞEVKİ HIZAL (20000404)

AHMET İHSAN ÜNLER (20000863)

SUBMITTED TO: ÜMİT SOYER

NICOSIA 2005

AKNNOWLEDGMENTS

Firstly, I would like to thank our advisors Ümit Soyer and Assist. Prof. Dr. Rahip Abiyev for advises, and also he helped me to carry out this project.

Also we would like to thank our all teachers who has given us these great education.

In addition we would like to thank to all my friends who helped me while I was preparing this project especially Ali Güneri, Şefki Kolozali.

ABSTRACT

The web is a hypertext system that runs over the Internet as one of its services. Users can work on any computer and browse documents that exist anywhere in the world. Furthermore, these documents can be linked to documents from any other computer in the world.

On the other hand the internet allows a client program on one computer to access a server on another computer. Viewing the web as a hypertext means that it needs navigation support features and that users will come to rely on consistent use of these navigation features to move around in hyperspace. In other words, there is a specific user interface that is optimal for the hypertext problem of navigating a huge information space to find and read information. Not that we are anywhere close to an optimal hypertext interface in today's Web browsers but they do have some features like back and bookmarks that users quickly learn to rely on.

It is unlikely that the user interface that is optimal for browsing hypertext will also be optimal for every other thing people want to do with computers. Thus, if the web becomes a single, universal interface to all internet services, we will either end up with a sub-optimal hypertext interface for the browsing tasks or get a sub-optimal user interface for all the non-browsing tasks

There are services to access email through a Web browser, but they have a distinctly second-class feel relative to the use of a smoothly designed email interface like Eudora or Outlook

TABLE OF CONTENTS

INTRODUCTION	1
CHAPTER I: DEFINITION OF ASP	2
1.1 What is ASP ?	2
1.2 ASP Compatibility	3
1.3 What is an ASP File?	4
1.4 What can ASP do for you?	4
1.5 How Does ASP Differ from HTML?	4
1.6 What will you learn?	5
1.7 ASP Syntax	5
1.7.1 The Basic Syntax Rule	5
1.7.2 The response Object	6
1.7.3 VBScript	6
1.7.4 JavaScript	7
1.7.5 Other Scripting Languages	7
1.8 Asp variables	8
1.8.1 Lifetime of Variables	8
1.8.2 Session Variables	8
1.8.3 Application Variables	8
1.9 Asp Procedures	8
1.9.1 Procedures	8
1.9.2 Differences between Javascript and Vbscript	10
1.10 Asp Forms	10

1.10.1 User Input	10
1.10.2 Request QueryString	10
1.10.3 Request Form	11
1.10.4 Form Validity	12
1.11 Asp session Object	13
1.11.1 The Session Object	13
1.11.2 When does a Session Object Start?	13
1.11.3 When does a Session Object End ?	13
1.11.4 Store and Retrieve Session Variables	15
1.11.5 Remove Session Variables	16
1.11.6 Loop Through the Content Connection	16
1.12 Asp Including Files	18
1.12.1 The Include Directive	18
1.12.2 How to Use the Include Directive	18
1.12.3 Syntax for Including Files	19
1.12.4 The Virtual Keyboard	19
1.12.5 The File Keyboard	20
1.12.6 Tips and Notes	20
CHAPTER II: DEFITION OF HTML	22
2.1 What is HTML	22
2.2 A Brief History of HTML	22
2.3 What is an HTML Files?	23
2.4 Titles... and Titles	24
2.5 The Body of The Document	24
2.6 HTML Tags	25

2.7 Headers	26
2.8 Character	26
2.9 Paragraphs	27
2.10 Line Links	27
2.11 HTML Links	27
2.11.1 The Target Attribute	27
2.11.2 The Anchor Tag The Name Attribute	28
2.12 HTML Tables	28
2.12.1 Tables and The Border Attribute	29
2.12.2 Heading in a Table	30
2.12.3 Empty Cells in a Table	30
2.13 Forms	31
2.14 Input	32
2.14.1 Text Fields	32
2.14.2 Radio Buttons	33
2.14.3 Checkboxes	33
2.14.4 Form's Action Attribute and The Submit Button	34
2.15 HTML Lists	34
2.15.1 Unordered Lists	34
2.15.2 Definition Lists	35
2.15.3 Definition Lists	35
2.16 HTML Backgrounds	36
2.16.1 Bgcolor	36
2.16.2 Background	37
2.17 HTML Images	37

INTRODUCTION

Nowadays we observed to increase the number of vehicle in Turkish Republic of North Cyprus. While increasing of number of vehicle, accidents increased. Unfortunately many people were die with accident.

Roads are insufficient despite increasing of the number of vehicle. This situation may cause the accidents.

Mistakes of driver are very important factor for accident. Because many reason of accidents are mistake of driver.

Some mistakes of driver;

- Don't obey rules of traffic
- Drunk during driving
- Driving very speed

Other reason of accidents is traffic light and signboard has not enough in the Turkish Republic of North Cyprus. Traffic light and signboard provides flow of traffic effectively.

In particular time such as morning 8:00 and evening 17:00, traffic is very crowd. Also flow of traffic is stop in some locations such as Dereboyu and Girne Kapısı.

Traffic policies should have new technologic advancements. This technologic advancement allows administration of traffic easily.

Traffic Information System is a technologic advancement. Traffic Information System contains information about driver, vehicle and walker with statistic data. This system stores this information.

Traffic Information System involves e-driver learning, statistics of traffic data, rules of traffic and presentation and program about traffic

CHAPTER I

DEFINITION OF ASP

1.1 WHAT IS 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 effected 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 page with the extension asp is requested by a browser the web server knows to interpret any ASP contained within the web page before sending the HTML produced to the browser. 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 can not 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 way 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 traffic 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 it is micro version Personal Web Server (PWS) comes free with Windows.

For Windows users you can find Internet Information Services (IIS) or Personal Web Server (PWS) in the following places.

- **Windows 2000/XP Pro**-Internet Information Service (IIS) can be found in 'Add/Remove Programs' in the 'Control Panel'.
- **Windows 98**-Personal Web Server (PWS) can be found under 'add-ons' on the Windows 98 CD.
- **Windows NT4/95**-You can get hold of IIS by downloading the NT4 Option Pack from Microsoft (don't be fooled by the name as it also runs on Windows 95).
- **Windows ME**-IIS and PWS are not supported on this operating system.
- **Windows XP Home Edition**-IIS and PWS are not supported on this operating system.

For those of you running other operating systems or web servers Sun produce a product called Chili!Soft ASP which enables ASP to be used on other web servers including , Apache, I-Planet, Zeus, Red Hat Secure Server, etc, using various operating systems including, Linux, Solaris, HP-UX, AIX, etc.

1.2 ASP Compatibility

- ASP is a Microsoft Technology
- To run IIS you must have Windows NT 4.0 or later
- To run PWS you must have Windows 95 or later
- ChiliASP is a technology that runs ASP without Windows OS

- Instant ASP is another technology that runs ASP without Windows

1.3 What is an ASP File?

- An ASP file is just the same as an HTML file
- An ASP file can contain text, HTML, XML, and scripts
- Scripts in an ASP file are executed on the server
- An ASP file has the file extension “.asp”

1.4 What can ASP do for you?

- Dynamically edit, change or add any content of a Web page
- Respond to user queries or data submitted from HTML forms
- Access any data or databases and return the results to a browser
- Customize a Web page to make it more useful for individual users
- The advantages of using ASP instead of CGI and Perl, are those of simplicity and speed
- Provides security since your ASP code can not be viewed from the browser
- Since ASP files are returned as plain HTML, they can be viewed in any browser
- Clever ASP programming can minimize the network traffic.

1.5 How Does ASP Differ from HTML?

- When a browser requests an HTML file, the server runs the file
- When a browser requests an ASP file, IIS passes the request to the ASP engine. The ASP engine reads the ASP file, line by line, and executes the scripts in the file. Finally, the ASP file is returned to the browser as plain HTML

1.6 What will you learn?

Let's consider the creating ASP pages in the VBScript Language. VBScript is a language exactly like many others. This means there are many different categories where you can place all the different commands which you can use.

Mainly it's these categories:

- Functions
- Objects
- Methods
- Operators
- Properties
- Statements

These categories contains many different commands which works and are used in different ways. Alls the commands are most likely used together with other commands from different categories. Therefor we will jump a little between them just to teach you the basics and the most important things. But of course, later we will go deeper in to the commands, methods and objects. In the beginning we will teach you the differents between the categories.

1.7 ASP SYNTAX

1.7.1 The Basic Syntax Rule

An ASP file normally contains HTML tags, just like an HTML file. However, an ASP file can also contain server scripts, surrounded by the delimiters `<%` and `%>` . Server scripts are

executed on the server, and can contain any expressions, statements, procedures, or operators valid for the scripting language you prefer to use.

1.7.2 The response Object

The Write method of the ASP Response Object is used to send content to the browser. For example, the following statement sends the text “Hello World” to the browser:

```
<%  
response.write("Hello World")  
%>
```

1.7.3 VBScript

You may use different scripting languages in ASP files. However, default scripting language is VBScript:

```
<html>  
<body>  
<%  
response.write("Hello World")  
%>  
</body>  
</html>
```


The example above writes "Hello World" into the body of the document.

12.4 JavaScript

To use JavaScript as the default scripting language for a particular page you must insert a language specification at the top of the page:

```
<!DOCTYPE html>  
<html language="javascript"%>
```

```
<body>  
  <script>  
    Write("Hello World!")  
  </script>  
</body>
```

Unlike VBScript – JavaScript is case sensitive. You will have to write your ASP code using uppercase letters and lowercase letters when the language requires it.

12.5 Other Scripting Languages

ASP is shipped with VBScript and JScript (Microsoft's implementation of JavaScript). If you want to script in another language, like PERL, REXX, or Python, you will have to install engines for them.

Important: Because the scripts are executed on the server, the browser that displays the ASP file does not need to support scripting at all!

1.8 ASP VARIABLES

1.8.1 Lifetime of Variables

A variable declared outside a procedure can be accessed and changed by any script in the ASP file.

A variable declared inside a procedure is created and destroyed every time the procedure is executed. No scripts outside the procedure can access or change the variable.

To declare variables accessible to more than one ASP file, declare them as session variables or application variables.

1.8.2 Session Variables

Session variables are used to store information about ONE single user, and are available to all pages in one application. Typically information stored in session variables are name, id and preferences.

1.8.3 Application Variables

Application variables are also available to all pages in one application. Application variables are used to store information about ALL users in a specific application.

1.9 ASP PROCEDURES

1.9.1 Procedures

The ASP source code can contain procedures and functions:

```
<html>

<head>

<%

sub vbproc (num1,num2)

response.write (num1*num2)

end sub

%>

</head>

<body>

<p>Result : <%call vbproc(3, 4)%></p>

</body>

</html>
```

Insert the `<%@ language="language" %>` line above the `<html>` tag to write procedures or functions in another scripting language than default:

```
<%@ language="javascript" %>

<html>
```

```

<head>
<%

function jsproc (num1, num2)

{
Response.write (num1, num2)

}

%>

</head>

<body>

<p>Result : <% jsproc (3, 4) %></p>

</body>

</html>

```

1.9.2 Differences Between VBScript and JavaScript

When calling a VBScript or a JavaScript procedure from an ASP file written in VBScript, you can use the “call” keyword followed by the procedure name. If a procedure requires parameters, the parameter list must be enclosed in parentheses when using the “call” keyword. If you omit the “call” keyword, the parameter list must not be enclosed in parentheses. If the procedure has no parameters, the parentheses are optional.

When calling a JavaScript or a VBScript procedure from an ASP file written in JavaScript, always use parentheses after the procedure name.

1.10 ASP Forms (User Input)

1.10.1 User Input

The request object may be used to retrieve user information from forms:

```
<form method="get" action="simpleform.asp">
```

```
First Name: <input type="text" name="fname">
```

```
<br />
```

```
Last Name: <input type="text" name="lname">
```

```
<br /><br />
```

```
<input type="submit" value="submit">
```

```
</form>
```

User input can be retrieved in two ways: With Request.QueryString or Request.Form.

1.10.2 Request.QueryString

The Request.QueryString command is used to collect values in a form with method="get". information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's address bar) and has limits on the amount of information to send.

If a user typed "Bill" and "Gates" in the form example above, the URL sent to the server would look like this:

```
http://www.3schools.com/simpleform.asp?fname=Bill&lname=gates
```

Assume that the ASP file “simpleform.asp” contains the following script:

```
<body>

Welcome

<%

response.write (request.querystring (“fname”))

response.write (“ “ & request.querystring(“lname”))

%>

</body>
```

The browser will display the following in the body of the document:

Wellcome Bill Gates

1.10.3 Request.Form

The Request.Form command is used to collect values in a form with method=”post”. Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send.

If a user typed “Bill” and “Gates” in the form example above, the URL sent to the server would look like this:

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

Assume that the ASP file “simpleform.asp” contains the follwing script:


```

<body>

welcome

<%

response.write (request.form ("fname"))

response.write (" " & request.form ("lname"))

%>

</body>

```

The browser will display the following in the body of the document:

Wellcome Bill Gates

1.10.4 Form Validation

User input should be validated on the browser whenever possible (by client scripts).

Browser validation is faster and you reduce the server load.

You should consider using server validation if the user input will be inserted into a database.

A good way to validate a form on the server is to post the form to itself, instead of jumping to a different page. The user will then get the error messages on the same page as the form. this makes it easier to discover the error.

1.11 ASP Session object

1.11.1 The Session object

When you are working with an application, you open it, do some changes and then you close it. This is much like a session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are and what you do because the http address doesn't maintain state.

ASP solves this problem by creating a unique cookie for each user. The cookie is sent to the client and it contains information that identifies the user. This interface is called the session object.

The session object is used to store information about, or change settings for a user session.

The variables stored in the session object hold information about one single user and are available to all pages in one application. Common information stored in session variables are name, id, and preferences. The server creates a new session object for each new user and destroys the session object when the session expires.

1.11.2 When does a session start?

A session starts when:

- A new user request an ASP file, and the Global.asa file includes a session_on start procedure.
- A value is stored in a Session variable
- A user requests an ASP file, and the Global.asa file uses the <object> tag to instantiate an object with session scope

1.11.3 When does a session End?

A session ends if a user has not requested or refreshed a page in the application for a specified period. By default, this is 20 minutes.

If you want to set a timeout interval that is shorter or longer than the default, you can set the Timeout property.

The example below sets a timeout interval of 5 minutes:

```
<%  
Session.Timeout=5  
%>
```

To end a session immediately, you may use the Abandon method:

```
<%  
Session.Abandon  
%>
```

Note: The main problem with session is WHEN they should end. We don't know if the user's last request was the final one or not. So we do not know how long we should keep the session "alive". Waiting too long for an idle session uses resources on the server, but if the session is deleted too soon the user has to start all over again because the server has deleted all the information. Finding the right timeout interval can be difficult.

Tip: If you are using session variables, store SMALL amounts of data in them.

1.11.4 Store and Retrieve Session Variables

The most important thing about the Session object is that you can store variables in it.

The example below will set the Session variable username to "Donald Duck" and the Session variable age to "50":

```
<%  
Session("username")="Donald Duck"  
Session("age")=50  
%>
```

When the value is stored in a session variable it can be reached from any page in the ASP application:

```
Welcome <%Response.Write(Session("username")) %>
```

The line above returns: "Welcome Donald Duck".

It is possible to store user preferences in the Session object, and then access that preference to choose what page to return to the user.

The example below specifies a text-only version of the page if the user has a low screen resolution:

```
<% If Session ("screenres")=" low" then%>
```

This is the text version of the page

```
<%Else%>
```

This is the multimedia version of the page

```
<%End If%>
```

1.11.5 Remove Session Variables

The Contents collection contains all session variables. It is possible to remove a session variable with the Remove method.

The example below removes the session variable "sale" if the value of the session variable "age" is lower than 18:

```
<%
```

```
If Session.Contents("age")<18 then
```

```
Session.Contents.Remove("sale")
```

```
End If
```

```
%>
```

To remove all variables in a session, use the RemoveAll method:

```
<%
```

```
Session.Contents.RemoveAll()
```

```
%>
```

1.11.6 Loop Through the Contents Collection

The Contents collection contains all session variables. We can loop through the Contents collection, to see what's stored in it:

```
Session("username")="Donald Duck"
```

```
Session("age")=50
```

```
Dim i
```

```
For Each i in Session.Contents
```

```
Response.Write(i & "<br />")
```

```
Next
```

Result:

username

age

If the number of items in the Contents collection is unknown, then we can use the Count property:

```
Dim i
```

```
Dim j
```

```

j=Session.Contents.Count
Response.Write("Session variables: " & j)

For i=1 to j
Response.Write(Session.Contents(i) & "<br />")

Next

```

Result:

```

Session variables: 2
Donald Duck
50

```

1.11.7 Loop Through the Static Objects Collection

Looking through the StaticObjects collection, we can see the values of all objects stored in the Session object:

1.12 ASP Including Files

1.12.1 The #include Directive

You can insert the content of one ASP file into another ASP file before the server executes it, with the #include directive. The #include directive is used to create functions, headers, footers, or elements that will be reused on multiple pages.

1.12.2 How to Use the #include Directive

Here is a file called “mypage.asp”:

```
<html>
<body>
<h3>Words of Wisdom:</h3>
<p><!--#include file="Wisdom.inc"--></p>
<h3>The time is:</h3>
<p><!--#include file="time.inc"--></p>
</body>
</html>
```

Here is the “wisdom.inc” file:

“One should never increase, beyond what is necessary, the number of entities required to explain anything.”

Here is the “time.inc” file:

Response.Write

If you look at the source

```
<html>
<body>
```

```
</h3>Words of wisdom:</h3>
```

```
<!-- The File Include -->
```

```
<p>"one should never, increase, beyond what is necessary, the number of  
files required to explain anything."</p>
```

```
<h3>The time is:</h3>
```

```
<p>11:33:42 AM</p>
```

```
<body>
```

```
</html>
```

and in a browser, it will look something like this:

112.3 Syntax for Including Files

To include a file in an ASP page, place the #include directive inside comment tags:

```
<!--#include virtual="somefilename"-->
```

```
<!--#include file="somefilename"-->
```

112.4 The Virtual Keyboard

Use the virtual keyboard to indicate a path beginning with a virtual directory.

If a file named "header.inc" resides in a virtual directory named/html, the following line would insert the contents of "header.inc":

```
<!--#include virtual="/html/header.inc"-->
```

112.5 The File Keyboard

Use the file keyboard to indicate a relative path. A relative path begins with the directory that contains the including file.

If you have a file in the html directory, and the file "header.inc" resides in html\headers, the following line would insert "header.inc" in your life:

```
<!--#include file ="headers\header.inc"-->
```

Note that the path to the included file (headers\header.inc) is relative to the including file. If the file containing this #include statement is not in the html directory, the statement will not work.

112.6 Tips and Notes

In the section above we have used the file extension ".inc" for included files. Notice that if a user tries to browse an INC file directly, its content will be displayed. If your included file contains confidential information or information you do not want any users to see, it is better to use an ASP extension. The source code in an ASP file will not be visible after the interpretation. An included file can also include other files, and one ASP file can include the same file more than once.

Important: Included files are processed and inserted before the scripts are executed.

The following script will not work because ASP executes the #include directive before it assigns a value to the variable:


```
fname="header.inc"
```

```
<!--#include file="<%=fname%>"-->
```

You can not open or close a script delimiter in an INC file. This script will not work:

```
For I=1 to n
```

```
<!--#include file="count.inc"-->
```

```
Next
```

But this script will work:

```
<% For i = 1 to n %>
```

```
<!--#include file="count.inc"-->
```

```
<% Next %>
```

CHAPTER II

DEFINITION OF HTML

2.1 What is HTML?

To publish information for global distribution, one needs a universal understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from Hyper Text Markup Language).

HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents.

2.2 History of HTML

HTML was originally developed by Tim Berners-Lee while at CERN, and popularized by the Mosaic browser developed at NCSA. During the course of the 1990s it has blossomed with the explosive growth of the Web. During this time, HTML has been extended in a number of ways. The Web depends on Web page authors and vendors sharing the same conventions for HTML. This has motivated joint work on specifications for HTML.

HTML 2.0 (November 1995) was developed under the aegis of the Internet Engineering Task Force (IETF) to codify common practice in late 1994. HTML+ (1993) and HTML 3.0 (1995) proposed much richer versions of HTML. Despite never receiving consensus in standards discussions, these drafts led to the adoption of a range of new features. The efforts of the World Wide Web Consortium's HTML Working Group to codify common practice in 1996 resulted in HTML 3.2 (January 1997).

Most people agree that HTML documents should work well across different browsers and platforms. Achieving interoperability lowers costs to content providers since they must develop only one version of a document. If the effort is not made, there is much greater risk that the Web will devolve into a proprietary world of incompatible formats, ultimately reducing the Web's commercial potential for all participants.

Each version of HTML has attempted to reflect greater consensus among industry players so that the investment made by content providers will not be wasted and that their documents will not become unreadable in a short period of time.

HTML has been developed with the vision that all manner of devices should be able to use information on the Web: PCs with graphics displays of varying resolution and colour depths, cellular telephones, hand held devices, devices for speech for output and input, computers with high or low bandwidth, and so on.

2.3 What is an HTML File?

- HTML stands for Hyper Text Markup Language
- An HTML file is a text file containing small markup tags
- The markup tags tell the Web browser how to display the page

- An HTML file must have an html file extension
- An HTML file can be created using a simple text editor

2.4 Titles.....and Titles

Before the `<Title>` tag you should have a `<HEAD>` tag. This tag says that the information following it is in the HEAD of the document.

The Title is a very important part of every HTML document because it labels the document for the user, and for holists and bookmarks. On most WWW browsers you will find that the Title is shown in a separate space from the document. The title is easy to add and should be one of the first things in an HTML document. The title text should be descriptive of the contents of the document.

Example:

```
<html>
<title>Testing Titles</title>
```

This will place the text "Testing Titles" in the Title space on your HTML browser.

```
</html>
```

After the title tag be sure to end the head of the document with the `</HEAD>` tag. Remember that since the Title can not be seen in your document you may wish to repeat it by placing it as a header at the beginning of your file. The header will appear in the Body of the document.

2.5 The Body of the Document

The body of the document start with a <Body> tag. This is the part of the document where you will put all of the text and multimedia elements you want to see in the browser window.

Inside the <body> tag attributes may be added that will change the background of the document, like <BODY BGCOLOR> and <BODY BACKGROUND>.

<BODY BGCOLOR> is set by putting in the RGB values for a colour in their hexadecimal form to get a specific colour for your background.

Example:

<BODY BGCOLOR="#ffffff">=white.

The <BODY BACKGROUND> tag will allow you to put an image into the background of your document.

To do this type:

<BODY BACKGROUND=filename.gif>

Be careful not to let your backgrounds overwhelm or detract from the content of your page.

Once you have established the canvas, it's time to add information. One of the first bits of info to include is a header. There are several kinds of Headers available...

2.6 HTML Tags

- HTML tags are used to mark-up HTML elements
- HTML tags are surrounded by the two characters < and >
- The surrounding characters are called angle brackets

- HTML tags normally come in pairs like and
- The first tag in a pair is the start tag, the second tag is the end tag
- The text between the start end tags is the element content
- HTML tags are not case sensitive, means the same as

2.7 Header

Header are usually used separate sections of text on page. Header have 6 different sizes, from.

Larger

Than normal text to much

Smaller

Than normal text. All headers come out on the screen as bold faced text. A Header will always be on its own line.

Example:

```
<html>
<title>Headers example</title>
<h1>This largest Text</h1>
<h2>This the Next Size...Header 2</h2>
<h3>This the same size as Normal Bold Text</h3>
<h4>Smaller...and Smaller...H4</h4>
<h5>Second Smakkaest</h5>
<h6>This is the Smallest Header</h6>
</html>
```

2.8 Character

at this time there are 3 styles of text supported by most WWW browser.

When using character types always make sure you close the tags after you have opened them.

The physical Styles are:

- **Bold** – which uses the tag.
- *Italic* –which uses the <I> tag.
- Typewriter Text –which uses the <TT> tag.

2.9 Paragraphs

Paragraphs are defined with the <p> tag.

<p> This is a paragraph</p>

<p> This is another paragraph</p>

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

2.10 Line Breaks

The
 tag is used when you want to end a line, but do not want to start a new paragraph.

The
 tag forces a line break wherever you place it.

Example:

<p> This
 is para
 graph with line breaks </p>

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

2.11 HTML LINKS

2.11.1 The Target Attribute

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

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

```
<a href="http://www.w3schools.com/html/html_links.asp" target="blank">Visit wW3Schools!</a>
```

2.11.2 The Anchor Tag and the Name Attribute

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

Below is the syntax of a named anchor:

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

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

The line below defines a named anchor:

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

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

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

```
<a href="http://www.w3schools.com/html_links.asp#tips">
```

```
Jump to the Useful Tips Section</a>
```

2.12 HTML TABLES

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

```
<table border="1">
```

```
<tr>
```

```
<td>row 1, cell 1</td>
```

```
<td>row 1, cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
<td>row 2, cell 1</td>
```

```
<td>row 2, cell 2</td>
```

How it looks in a browser:

```
row1, cell 1 row 1, cell 2
```

```
row2, cell 1 row 2, cell 2
```

2.12.1 Tables and the Border Attribute

If you do not specify a border attribute the table will be displayed without any borders.

Sometimes this can be useful, but most of the time, you want the borders the

Show. Another Heading

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

```
<table border="1">
```

```
<td>Row 1, cell 1</td>
```

```
<td>Row 1, cell 2</td>
```

```
</tr>
```

```
</table>
```

```
<td>Row 1, cell 1</td>
```

```
<td>Row 1, cell 2</td>
```

2.12.2 Headings in a Table

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

```
<table border="1">
```

```
<tr>
```

```
<th>heading</th>
```

```
</tr>
```

```
<tr>
```

```
<td>row 1, cell 1</td>
```

```
<td>row 1, cell 2</td>
```

```
</tr>
```

```
<tr>
```

```
<td>row 2, cell 1</td>
```

```
<td>row 2, cell 2</td>
```

```
</tr>
```

```
</table>
```

How it looks in a browser:

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

2.12.3 Empty Cells in a Table

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

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

How it looks in a browser:

row 1, cell 1 row 1, cell 2

row 2, cell 1

Note that the borders around the empty table cell are missing.

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

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

When the user clicks on the "Submit" button, the content of the form is sent to another file. The `action` attribute defines the name of the file to send the content to. The file defined in the `action` attribute usually does something with the received input.

```
<input name="input" action="html_form_action.asp" method="get">
```

```
<input type="text" name="user">
```

```
<input type="Submit" value="Submit">
```

How it looks in a browser:

name:

When you type some characters in the text field above, and click the submit button, you will send your input to a page called "html_form_action.asp". That page will show you the received

2.5 HTML LISTS

2.5.1 Unordered Lists

An unordered list is a list of items. The list items are marked with bullets (typically small black circles).

An unordered list starts with the `` tag. Each list item starts with the `` tag.

```
<li>ahmet</li>
```

```
<li>mustafa</li>
```

links in a browser:

active link

visited link

For definition (the <dd> tag) you can put paragraphs, line breaks, images,

etc.

SET THE BACKGROUNDS

The tag has two attributes where you can specify background can be a color or an

attribute sets the background to a color. The value of this attribute can be a

number, an RGB value, or a color name.

```
background="000000">
```

```
background="rgb(0,0,0)">
```

```
background="black">
```

will set the background color to black.

Background

The `background` attribute sets the background to an image. The value of this attribute is the image you want to use. If the image is smaller than the browser window, the image is repeated until it fills the entire browser window.

```
background="clouds.gif">
```

```
background="http://www.w3schools.com/clouds.gif">
```

The value can be relative (as in the first line above) or absolute (as in the second line above).

If you want to use a background image, you should keep in mind:

- Does the background image increase the loading time too much? Tip: Image files should be as small as possible.

- Does the background image look good with other images on the page?

- Does the background image look good with the text colors on the page?

- Does the background image look good when it is repeated on the page?

- Does the background image take away the focus from the text?

IMAGES

The Image Tag and the Src Attribute

An image is defined with the `` tag.

The `` tag is an empty tag, which means that it contains attributes only and it has no closing tag.

To insert an image on a page, you need to use the `src` attribute. `src` stands for "source". The

value of the `src` attribute is the URL of the image you want to display on your page.

Syntax of defining an image:

```

```

The URL points to the location where the image is stored. An image named "boat.gif" located



<http://www.w3schools.com/images/boat.gif>.

The browser puts the image where the image tag occurs in the document. If you put an image tag between two paragraphs, the browser shows the first paragraph, then the image, and then the second paragraph.

2.17.2 The Alt Attribute

The alt attribute is used to define an "alternate text" for an image. The value of the alt attribute is an author-defined text:

```

```

The "alt" attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the "alt" attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers.

CHAPTER III

DATABASE STRUCTURE OF THE PROJECT

What is a database?

Once upon a time, in the primitive and barbarian days before computers, the amount of information shepherded by a group of people could be collected in the wisdom and the stories of its order members. In this world, storytellers, magicians, and were considered great and honored. The houses for all was known.

Apparently, and according to vast archeological data, campfires were used (like commandline interface) by the younger members of the community to access the information stored in the minds of the elders using API's such as public String TellUsAboutTheTimeWhen(String

And then of course, like a sweeping and rapidly-encompassing viral infection, came agriculture, over-production of foodstuffs, and the origins of modern-day commerce.

Dealing with vast storehouses of wheat, rice, and maize became quite a chore for the monarchs and emperors that developed along with the new economy. There was simply too much data to be managed in the minds of the elders (who by now were feeling the effects of hardware obsolescence as they were being pushed quietly into the background).

And so, in order to store all the new information, humanity invented the technology of writing. And though great scholars like Aristotle warned that the invention of the alphabet

lead to the subtle but total demise of the creativity and sensibility of humanity, data
to be stored in voluminous data repositories, called books.

we know, eventually books propagated with great speed and soon, whole communities of
migrated to the first real "databases", libraries.

previous versions of data warehouses (people and books), that might be considered the
ancestors of the database lineage, libraries crossed over into the modern-day species,
which were incredibly primitive of course. Specifically, libraries introduced "standards"
by which data could be stored and retrieved.

After all, without standards for accessing data, libraries would be like my closet, endless and
confusing swarms of chaos. Books, and the within books, had to be quickly accessible by
name if they were to be useful.

In fact, the usefulness of a library, or any base of data, is proportional to its data storage and
retrieval efficiency. This one corollary would drive the evolution of databases over the next
2000 years to its current state.

Thus early librarians defined standardized filing and retrieval protocols. Perhaps, if you have
ever made it off the web, you will have seen an old library with its cute little indexing system
(card catalog) and pointers (Dewey decimal system).

And for the next couple thousand years libraries grew, and grew, and grew along with
associated storage/retrieval technologies such as the filing cabinet, colored tabs, and three ring
binders.

until one day about half a century ago, some really bright folks including Alan Turing, working for the British government were asked to invent an advanced tool for breaking German cryptographics "Enigma" codes.

That day the world changed again. That day the computer was born.

The computer was applied to the age-old problem of information storage and retrieval. After all, by World War Two, information was already accumulating at rates beyond the space available in publicly supported libraries. And besides, it seemed somehow cheap and tawdry to store the entire archives of "The Three Stooges" in the Library of Congress. Information was seeping out of every crack and pore of modern day society.

Thus, the first attempts at information storage and retrieval followed traditional lines and metaphors. The first systems were based on discrete files in a virtual library. In this file-oriented system, a bunch of files would be stored on a computer and could be accessed by a computer operator. Files of archived data were called "tables" because they looked like tables used in traditional file keeping. Rows in the table were called "records" and columns were called "fields".

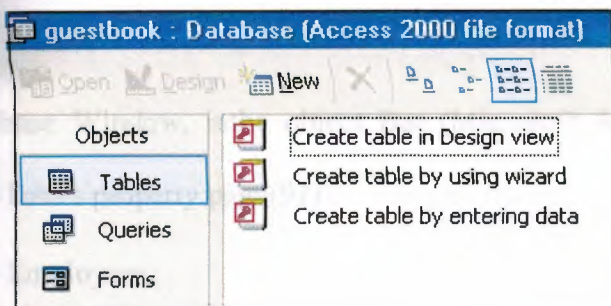
3.2 Connecting To An Access Database

If you are reading this page then I shall assume that you already know a little bit about ASP and running ASP scripts. To make this tutorial more interesting and the following database materials on, Adding, Deleting, and Updating, data from a Microsoft Access database, we are going to use these tutorials to make a simple Guestbook. Before we can connect to a database we need a database to connect too.

Creating the Guestbook 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 'guestbook.mdb' and save it in the directory as the web page connecting to 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 their data types.

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

Field 2 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.

Once all the fields have been created and the data types and primary key set, save the table as

'comments.'

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.

If you are having trouble creating the database then you can download this tutorial containing the Access Database with test data already entered.

3.4 Opening a Table

1. On the Database Window, in the object Bar (Microsoft Access 2000) click the Tables button or the Tables property page(97)
2. Double-click Employees.

employees : Table						
	Employee ID	Employee#	Date Hired	Title	First Name	MI
▶		1 65-4386	26.09.1992	Mrs	Laurentine	D
		2 35-7690	30.12.1990	Mr	jeremiah	
		3 48-9349	14.04.1995	Ms	Monique	P
		4 34-2908	10.05.1996	Mrs	Christine	H
*		0				

Record: 1 of 4

3. To close the table, click its Close button.

3.5 Opening a Query

1. On the Database Window, in the Object (2000) click the Queries button or the Queries property page (97).
2. Double-click list of Registrants By T-Shirt Size to open it.

List of Registrants By T-Shirt Size: Select Query : Table					
	First Name	Last Name	DOB	Gender	TShirtSize
	Charles	Gridley	10.05.1987	Male	Large
	Alicia	Kelley	07.12.1983	Female	Large
	Frency	Seats	27.10.1983	Male	Large
	Brigitte	Blue	01.04.1983	Female	Large
	Willy	Banneker	20.01.1982	Male	Large
	Robert	Wilson	12.02.1985	Male	Large
	Brigitte	Andersen	01.10.1984	Female	Large
	Cynthia	Nightingale	10.02.1986	Female	Large
	Gerard	Beirs	30.04.1982	Male	Large
	David	Ginsberg	27.10.1982	Male	Large
	Christopher	Newmann	04.11.1984	Male	Large
	Franky	Estamos	02.01.1985	Male	Large
▶	John	Compton	05.10.1983	Male	Large
Record: 13 of 13					

3. After viewing the query, to close it, click its Close button.

3.6 Connecting to the Guestbook Database

Now that the database is created and test data entered we can get on with creating the web page to display the data from the database.

First we need to start web page, open up your favourite text editor and type the following HTML.

```
<html>
<head>
<title>My First ASP Page</title>
</head>
<body bgcolor="white" text="black">
```

Next we can begin writing the ASP to connect to the database. First we need to create the variable that we are going to use in the script.

Dimension variables

<code>Dim adocon</code>	Holds the database connection object
<code>Dim rsGuestbook</code>	Holds the recordset for the records on the database
<code>Dim strSQL</code>	Holds the SQL query to query the database

Next we need to create a database connection object on the server using the ADO Database connection object.

```
Set adocan=Server.CreateObject("ADODB.Connection")
```

Now we need to open a connection to the database. There are a couple of ways of doing this either by using a system DSN or a DSN-less connection. First I am going to show you how to make a DSN-less connection as this is faster and simpler to set up than a DSN connection.

To create a DSN-less connection to an Access database we need tell the connection object created above to open the database by telling the connection object to use the 'Microsoft Access Driver' to open the database 'guestbook.mdb'.

We will notice the ASP method 'Server.MapPath' in front of the database. This is used as we need to get the physical path to the database. Server.MapPath returns the physical path the database, e.g. 'c:\website\' , as long as the database is in the same folder as the script it now has the physical path to the database and the database name.

Get an active connection to the connection object using a DSN-less connection

```
Con.Open "DRIVER={Microsoft Access Driver  
(.mdb)};DBQ=Server.MapPath("guestbook.mdb")"
```

On the other hand you want to use a slower DSN connection to the database then you will need to replace the line above with the one below. Also if you don't know how to setup a DSN you will need to read my tutorial on, Setting up a system DSN.

Get an active connection to the connection object using DSN connection

```
Con.Open "DSN=guestbook"
```

Next create an ADO recordset object which will hold the records from the database.

Create an ADO recordset object

```
Set rsGuestbook=Server.Create.Object("ADODB.recordset")
```

Query a database we need to use SQL (Structured Query Language). In the next line we initialise the variable 'strSQL' with an SQL query to read in the fields 'Name' and 'Comments' from the 'tblComments' table.

initialise the strSQL variable with an SQL statement to query the database

```
strSQL = "SELECT tblComments.Name, Comments FROM tblComments;"
```

Now we can open the recordset and run the SQL query on the database returning the results of the query to the recordset.

open the recordset with the SQL query

```
rsGuestbook.Open strSQL, adoCon
```

Using a 'Do While' loop we can loop through the recordset returned by the database while the recordset is not at the end of file (EOF). The 'Response.Write' method is used to output the recordset to the web page. The 'MoveNext' method of the recordset object is used to move to the next recordset before looping back round to display the next record.

Loop through the recordset

Do While not rsGuestbook.EOF

Write the HTML to display the current record in the recordset

Response.Write ("
")

Response.Write (rsGuestbook ("Name"))

Response.Write ("
")

Response.Write (rsGuestbook ("Comments"))

Response.Write ("
")

And finally we need to close the recordset, reset the server objects, close the server side scripting tag, and close the html tags.

Reset server objects

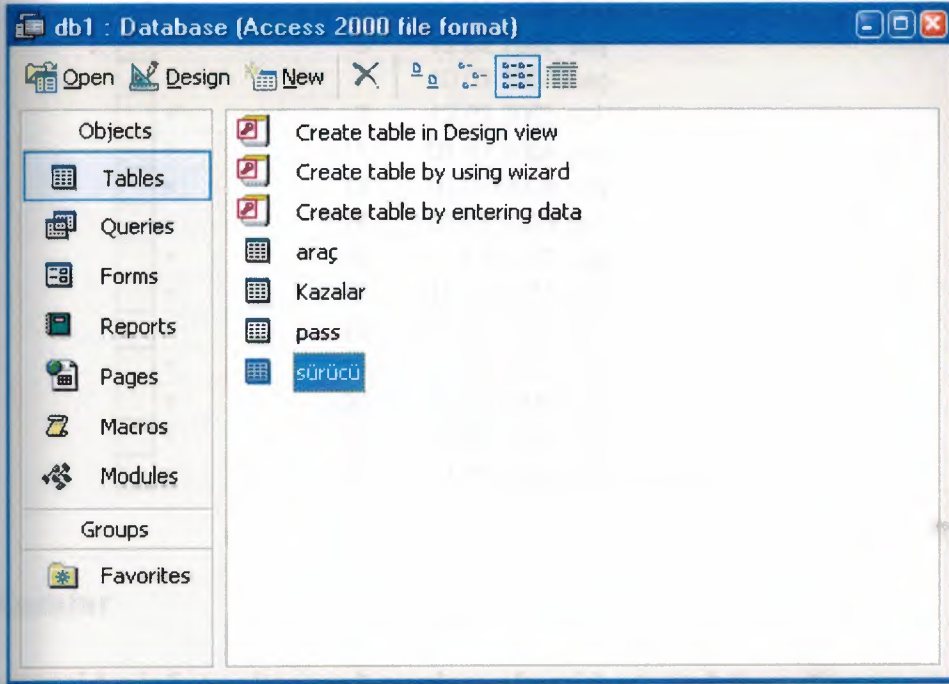
rsGuestbook.Close

set rsGuestbook=Nothing

Set adoCon=Nothing

%>

the necessary database tables as I explained above. The reason that I made these tables is that to make easy to add new information.



3.7 Araç

the database takes the names. This database involves years for vehicles and place of steering-wheel (right or left)

araç : Table				
		Kaza_Id	Araç_yili	Direksiyon
+		1	12.02.2000	Sağ
+		2	13.12.2001	Sol
+		3	03.05.2005	Sağ
+		4	12.02.2004	sol
+		5	03.04.2003	Sağ
+		6	12.02.2002	sag
+		7	10.02.2004	sag
+		9	12.02.2002	sol
+		11	01.01.2004	asdf
+		13	01.01.2004	yok
+		14	01.01.1999	sol
+		16	12.02.2000	sag
+		17	16.11.1973	sag
+		18	13.08.1994	sol
+		19	31.05.1993	sag
+		20	12.03.1999	sag
+		21	23.04.1990	sag
+		22	14.09.1967	sag direksiyon

3.8 Kazalar

This table provides information such as place of accidents and time of accidents.

Table									
Kaza_Id	Sehir	lc_Dis	Saat	Gün	Isiklandirma	Faktör	Hasar	Hava_durumu	Neden
1	Lefkosa	ici	12:00-14:00	Pazartesi	Gündüz	Sürücü Hatası	.000.000,00 TL	Güneşli	Dikkatsiz Sür
2	Gazimagusa	disi	13:00	Cumartesi	Gündüz	Yaya Hatası	.000.000,00 TL	Yağmurlu	Dikkatsiz Sür
3	Girne	disi	14:30	Salı	Gece	Araç Anzası	.000.000,00 TL	Sisli	Aşırı Sürat
4	Lefkosa	ici	13:42	Perşembe	Gündüz	Aşırı Sürat	.000.000,00 TL	Güneşli	Sürücü Hatası
5	lefkosa	disi	23:23	cuma	gündüz	hatalı saglama	.000.000,00 TL	güneşli	sürücü hatası
6	lefkosa	disi	2.2.2004	cuma	var	var	.000.000,00 TL	iyi	yok
7	girne	ici	3.3.2005	pazar	gunduz	yok	.000.000,00 TL	iyi	yok
8	lefkosa	ici	10.2.2004	salı	gece	YOK	.000.000,00 TL	gunesli	yok
9	gazimagusa	disi	12.2.2004	Cuma	gece	nedensiz	.000.000,00 TL	gunesli	hatalı saglam
10	asdf	asdf	12.2.2005	adfs	asdf	asdf	222.222,00 TL	asdf	asdf
11	lefkosa	ic	12.2.2004	salı	yok	yok	.000.000,00 TL	iyi	yok
12	asdf	asdf	1.1.2004	asdf	asdf	asdf	3.234,00 TL	asdf	asdf
13	girne	ic	1.1.2004	pazar	yok	asdf	10.000,00 TL	yok	yk
14	girne	asdf	1.2.2005	adfs	asdf	yok	200.000,00 TL	asdf	asdf
15	girne	iç		salı	var		200.000,00 TL		
16	lefkosa	diş	13:26	pazar	var		.000.000,00 TL		
17	gazimagosa	iç	15:36	pazartesi	var		200.000,00 TL	açık	
18	girne	diş	13:26	çarşamba	var		.770.000,00 TL	yağmurlu	

3.9 Sürücü

This table provides information such as education of drivers and alcohol rate of drivers.

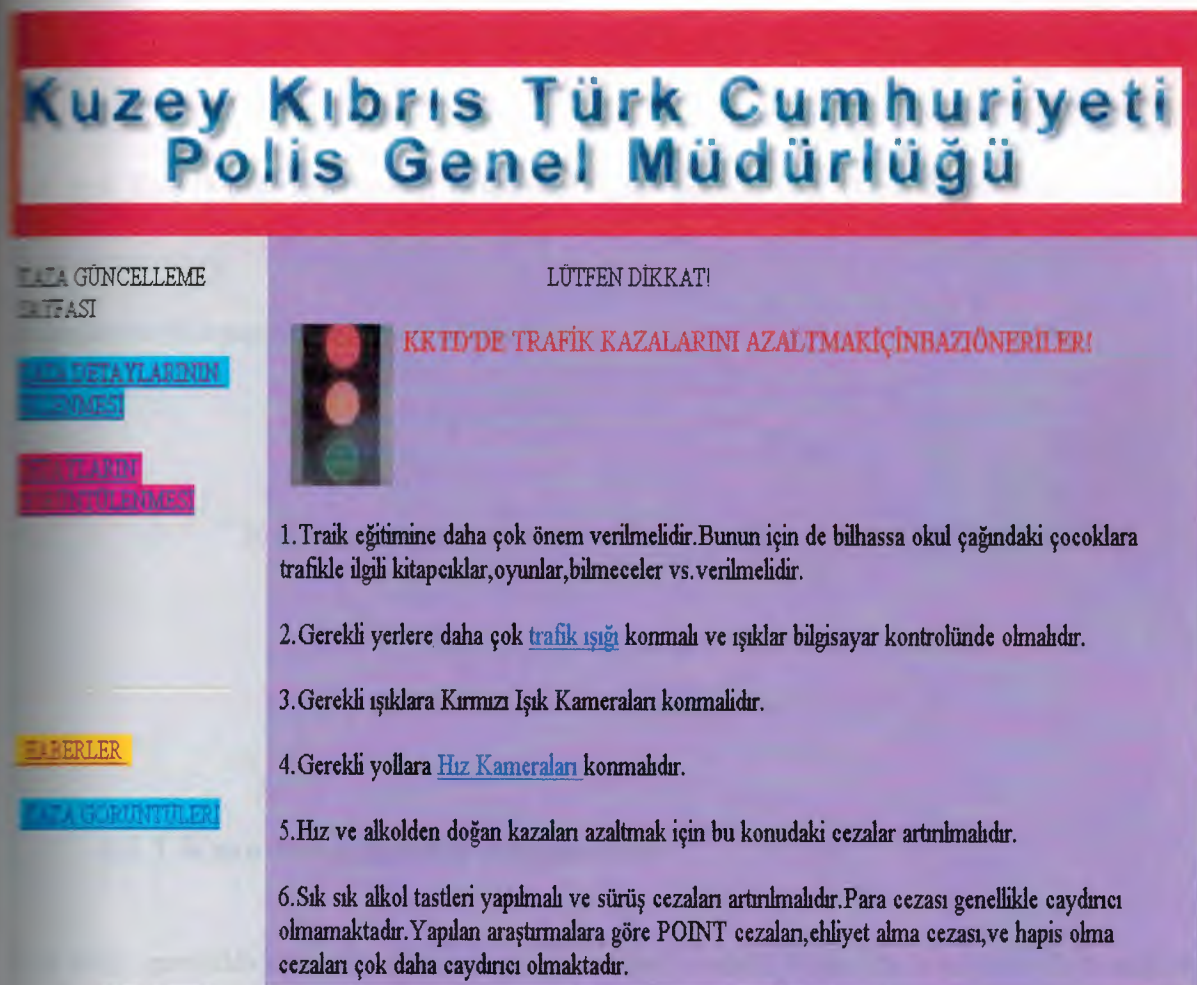
sürücü : Table							
	Kaza_Id	Sürücü_Id	Tahsil	Ehliyet	Emniyet_Keme	Alkol_Orani	Görüs_Mesafes
+	1	100	Üniversite Öğre	Var	Takılı	100-125	Yeterli
+	2	101	İlkokul	Var	Takılmadı	0	Yeterli
+	3	102	Üniversite Mezu	Var	Takılı	200	Yetersiz
+	4	103	ilkokul	var	takılı	0	yeterli
+	5	104	lise	var	takılı	0	yeterli
+	6	6	yok	var	var	yok	yeterli
+	7	402	yok	var	var	yok	yeterli
+	8	2354	yok	var	var	yok	yeterli
+	9	32	ilkokul	var	var	var	yeterli
+	10	12	asdf	asdf	asdf	asdf	asdf
+	11	134	yok	yok	yok	yok	yok
+	12	23	yok	yok	yok	yok	yok
+	13	1223	yok	yok	yok	yok	yok
+	14	12234	yok	yok	yok	yok	yok
+	15	23	lise	var	takılı	12-123	az
+	16	27	lise	var	takılı	15-56	az
+	17	14					
+	18	198					
+	19	177	lise	var	takılı		
+	20	356	ilkokul	yok	yok	78-167	az
+	21	127	lise	1 yıllık	takılı	11-130	yetersiz

CHAPTER IV

STRUCTURE OF PROGRAM

is the first page seen when the page is opened. Our program's language is Turkish.

Therefore the name of link and content of link is Turkish.



This home page has two parts. First part is at right and second part is at left. So first part provides to update accidents (it shows silver color) and another part provides information about the traffic (it shows levander color).

4.1 Kaza Güncelleme Sayfası (The Page of Update Accident)

This section involves four links. These four links :

- Kaza detaylarının eklenmesi
- Detayların görüntülenmesi
- Haberler
- Kaza görüntüleri

The first link (Kaza detaylarının eklenmesi) and second link (Detayların görüntülenmesi) needs a password.

If you know this password, you will open links.

Kullanıcı Adı

Sifre

Giriş

4.1.1 Kaza Detaylarının Eklenmesi

This page provides data about accident, driver and vehicle. It has three steps. In the first step we enter the data about accident such as time and day of accident, reason of accident.

Step shows as below

Kuzey Kıbrıs Türk Cumhuriyeti Polis Genel Müdürlüğü

Trafik Şubesi

Kaza Detayı Ekleme Sayfası

Kaza_Id:	21	Faktör:	Sürücü Hatası
Şehir:	Lefkoşa	Hasar:	1654000000
İc_Dis:	Şehir içi	Hava_durumu:	Bulutlu
Saat:	17:01-08:30	Neden:	Yakın Takip
Gün:	Pazar	Ölü Sayısı:	2
Isiklandırma:	Gece	Yol_Durumu:	Düz yol
<input type="button" value="Detayları Gönder"/>			

After we fill all data about accident and send with “Detayları Gönder” then second page will

open.

In second step we enter the data about driver such as alcohol rate of driver, driver licensing and education.

as below

Kuzey Kıbrıs Türk Cumhuriyeti

Polis Genel Müdürlüğü

Trafik Şubesi

Sürücü Detayı Ekleme Sayfası

Kaza_Id:	21
Sürücü_Id:	127
Tahsil:	lise
Ehliyet:	1 yıllık
Emniyet_Kemer:	takılı
Alkol_Oranı:	11-130
Görüş_Msafesi:	yetersiz
<input type="button" value="Detayları Gönderiniz"/>	

After we fill all data about driver and send with “Detayları Gönderiniz” then third page will open.

In third step we enter the data about vehicle. It involves the place of steering-wheel (right or left) and years of wrecked vehicle.

Third step shows as below

Kuzey Kıbrıs Türk Cumhuriyeti Polis Genel Müdürlüğü

Trafik Şubesi

Araç Detayı Ekleme Sayfası

Kaza_Id:	<input type="text" value="22"/>
Araç_yılı:	<input type="text" value="14.09.1967"/>
Direksiyon:	<input type="text" value="sağ direksiyon"/>
<input type="button" value="Detayları Gönderiniz"/>	

After we fill all data about vehicle and send with “Detayları Gönderiniz” then process will carry out.

It shows as below.

Kuzey Kıbrıs Türk Cumhuriyeti Polis Genel Müdürlüğü

Trafik Şubesi

Detaylar başarıyla yenilenmiştir...

Teşekkür ederiz

4.1.2 Detayların Görüntülenmesi

This links provide to show entered data about accident, driver, vehicle detaily.

It shows as below.

Kaza_Id	Sehir	Ic_Dis	Saat	Gün	Isiklandirma	Faktör	Hasar	Hava_durumu	Neden	Olü_sayisi
1	Lefkosa	ici	12.00-14:00	Pazartesi	Gündüz	Sürücü Hatası	1000000000	Güneşli	Dikkatsiz Sürüş	3
2	Gazimagusa	disi	13:00	Cumartesi	Gündüz	Yaya Hatası	500000000	Yagmurlu	Dikkatsiz Sürüş	2
3	Girne	disi	14:30	Salı	Gece	Araç Arızası	150000000	Sisli	Asiri Sürat	4
4	Lefkosa	ici	13:42	Perşembe	Gündüz	Asiri Sürat	2000000000	Güneşli	Sürüc Hatası	3
5	lefkosa	disi	23:23	cuma	gündüz	hatalı saglama	210000000	güneşli	sürücü hatası	1
6	lefkosa	disi	2.2.2004	cuma	var	var	200000000	iyi	yok	2
7	girne	ici	3.3.2005	pazar	gunduz	yok	20000000	iyi	yok	2
8	lefkosa	ici	10.2.2004	salı	gece	YOK	100000000	gunesli	yok	1
9	gazimagusa	disi	12.2.2004	Cuma	gece	nedensiz	3020000000	gunesli	hatalı saglama	1

Each entered data has a link.If we use link, this entered data show detaily.

Kaza_Id	4
Sehir	Lefkosa
Ic_Dis	ici
Saat	13:42
Gün	Perşembe
Isiklandirma	Gündüz
Faktör	Asiri Sürat
Hasar	2000000000
Hava_durumu	Güneşli
Neden	Sürüc Hatası
Olü_sayisi	3
Yol_Durumu	Çember

4.1.3 Haberler

This link provide information meeting and symposium about traffic.

It shows as below.

Haberler

- Kuzey Kıbrıs'taki trafiğin durumu ve sorunları konulu sempozyum 30 Mayıs 2005 Pazartesi Günü YDÜ AKM Salonunda gerçekleştirilecektir
- YDÜ öğretim üyesi Dr. Ahmet Ertugan polislerin bilgilendirilmesi konulu bir eğitim vermeye başlamıştır.
-

4.1.4 Kaza Görüntüleri

After the accident traffic police photo pictures about accidents. For example wrecked vehicle.

It shows as below.



4.2 Trafik Işığı

Traffic light take place right of home page.If you use link you will see as below.

It shows as below.



TRAFİK IŞIKLARI İLE İLGİLİ TAVSİYELER...

- KKTC'deki trafik ışıkları teknik bir kurumun kontrolüne verilmeli ve 24 saat bakım ve onarımı yapılmalıdır.
- Mevcut ışıkların sayısı gerektiği gibi sırtırılmalıdır.
- Işıkların zamanlama çalışmaları sık sık yapılmalı ve ışık zamanlamaları gerektiği gibi düzeltilmelidir.
- Işıklar adaptiv bilgisayar kontroluna bağlanmalıdır.

1

4.3 Hız Kameraları

Equipment of speed control.If you use link you will see as below.

HIZ KAMERALARI

SABİT KAMERALAR

TAŞINABİLİR KAMERALAR



ahmus.asp

trafik.asp

<? Language="VBScript"%>

<html>

<head>

```
<meta http-equiv="Content-Language" content="tr">
```

<meta name="GENERATOR" content="Microsoft FrontPage 5.0">

```
meta name="ProgId" content="FrontPage.Editor.Document">
```

```
meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
```

title>TRAFIK</title>

```
base target="contents">
```

```
head>
```

body bgcolor="#FF0066">

[illegible]

<p> </p>

<p> </p>

*** Valıkan Beyazıt'ın Haberleri ***

<hr> [Giriş](#) [Gizlilik](#) [Hakkımızda](#) [İletişim](#)

<p>
 HABERLER </p>

 </p>

<p> KAZA GÖRÜNTÜLERİ</p>

</body> [Giriş](#) [Gizlilik](#) [Hakkımızda](#) [İletişim](#)

</html> [Giriş](#) [Gizlilik](#) [Hakkımızda](#) [İletişim](#)

Giriş.asp

<%@LANGUAGE="VBSCRIPT"%>

<!--#include file="Connections/pasword.asp" -->

<%

Dim password

Dim password_numRows

Set password = Server.CreateObject("ADODB.Recordset")

password.ActiveConnection = MM_pasword_STRING

password.Source = "SELECT * FROM pass"

password.CursorType = 0

password.CursorLocation = 2

password.LockType = 1

password.Open()

password_numRows = 0


```

%>

<%

' *** Validate request to log in to this site.

MM_LoginAction = Request.ServerVariables("URL")

If Request.QueryString <> "" Then MM_LoginAction = MM_LoginAction + "?" +
Request.QueryString

MM_valUsername=CStr(Request.Form("textfield"))

If MM_valUsername <> "" Then

    MM_fldUserAuthorization=""

    MM_redirectLoginSuccess="Kazalar_ekleme.asp"

    MM_redirectLoginFailed="tesekkur.htm"

    MM_flag="ADODB.Recordset"

    set MM_rsUser = Server.CreateObject(MM_flag)

    MM_rsUser.ActiveConnection = MM_password_STRING

    MM_rsUser.Source = "SELECT username, password"

    If MM_fldUserAuthorization <> "" Then MM_rsUser.Source = MM_rsUser.Source & "," &
MM_fldUserAuthorization

    MM_rsUser.Source = MM_rsUser.Source & " FROM pass WHERE username='" &
Replace(MM_valUsername,"'", "''") & "' AND password='" &
Replace(Request.Form("textfield2"), "'", "''") & "'"

    MM_rsUser.CursorType = 0

    MM_rsUser.CursorLocation = 2

    MM_rsUser.LockType = 3

    MM_rsUser.Open

    If Not MM_rsUser.EOF Or Not MM_rsUser.BOF Then

```

```

' username and password match - this is a valid user

Session("MM_Username") = MM_valUsername

If (MM_fldUserAuthorization <> "") Then

    Session("MM_UserAuthorization") =

CStr(MM_rsUser.Fields.Item(MM_fldUserAuthorization).Value)

Else

    Session("MM_UserAuthorization") = ""

End If

if CStr(Request.QueryString("accessdenied")) <> "" And false Then

    MM_redirectLoginSuccess = Request.QueryString("accessdenied")

End If

MM_rsUser.Close

Response.Redirect(MM_redirectLoginSuccess)

End If

MM_rsUser.Close

Response.Redirect(MM_redirectLoginFailed)

End If

%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Untitled Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

</head>

<body>

```


%>

Kazalar ekleme.asp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>Polis Genel Müdürlüğü Trafik Şubesi</title>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
```

```
</head>
```

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

```
<!--#include file="Connections/recordset.asp" -->
```

```
<%
```

```
*** Edit Operations: declare variables
```

```
Dim fp_editAction
```

```
Dim fp_abortEdit
```

```
Dim fp_editQuery
```

```
Dim fp_editCmd
```

```
Dim fp_editConnection
```

```
Dim fp_editTable
```

```
Dim fp_editRedirectUrl
```

```
Dim fp_editColumn
```

```
Dim fp_recordId
```

```
Dim fp_fieldsStr
```

```
Dim fp_columnsStr
```

```
Dim fp_fields
```



```

Dim fp_columns
Dim fp_typeArray
Dim fp_formVal
Dim fp_delim
Dim fp_altVal
Dim fp_emptyVal
Dim fp_i
fp_editAction = CStr(Request.ServerVariables("SCRIPT_NAME"))
If (Request.QueryString <> "") Then
    fp_editAction = fp_editAction & "?" & Request.QueryString
End If
'boolean to abort record edit
fp_abortEdit = false
'query string to execute
fp_editQuery = ""
%>
<%
*** Insert Record: set variables
If (CStr(Request("fp_insert")) = "form1") Then
    fp_editConnection = fp_recordset_STRING
    fp_editTable = "Kazalar"
    fp_editRedirectUrl = "sürücü.asp"
    fp_fieldsStr =
    "Kaza_Id|value|Sehir|value|Ic_Dis|value|Saat|value|Gn|value|Isiklandirma|value|Faktr|value|H
asar|value|Hava_durumu|value|Neden|value|OlmIm|value|Yol_Durumu|value"

```

_columnsStr =

aza_Id|none,none,NULL|Sehir|',none,"|Ic_Dis|',none,"|Saat|',none,"|Gün|',none,"|Isiklandirm
none,"|Faktör|',none,"|Hasar|none,none,NULL|Hava_durumu|',none,"|Neden|',none,"|Olü_sa
i|',none,"|Yol_Durumu|',none,""

create the fp_fields and fp_columns arrays

fp_fields = Split(fp_fieldsStr, "|")

fp_columns = Split(fp_columnsStr, "|")

set the form values

for fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2

fp_fields(fp_i+1) = CStr(Request.Form(fp_fields(fp_i)))

next

append the query string to the redirect URL

If (fp_editRedirectUrl <> "" And Request.QueryString <> "") Then

If (InStr(1, fp_editRedirectUrl, "?", vbTextCompare) = 0 And Request.QueryString <> "")

then

fp_editRedirectUrl = fp_editRedirectUrl & "?" & Request.QueryString

Else

fp_editRedirectUrl = fp_editRedirectUrl & "&" & Request.QueryString

End If

End If

End If

>

&

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

fp_tableValues

```
Dim fp_dbValues
```

```
If (CStr(Request("fp_insert")) <> "") Then
```

```
    'create the sql insert statement
```

```
    fp_tableValues = ""
```

```
    fp_dbValues = ""
```

```
    For fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2
```

```
        fp_formVal = fp_fields(fp_i+1)
```

```
        fp_typeArray = Split(fp_columns(fp_i+1),",")
```

```
        fp_delim = fp_typeArray(0)
```

```
        If (fp_delim = "none") Then fp_delim = ""
```

```
        fp_altVal = fp_typeArray(1)
```

```
        If (fp_altVal = "none") Then fp_altVal = ""
```

```
        fp_emptyVal = fp_typeArray(2)
```

```
        If (fp_emptyVal = "none") Then fp_emptyVal = ""
```

```
        If (fp_formVal = "") Then
```

```
            fp_formVal = fp_emptyVal
```

```
        Else
```

```
            If (fp_altVal <> "") Then
```

```
                fp_formVal = fp_altVal
```

```
            ElseIf (fp_delim = "") Then ' escape quotes
```

```
                fp_formVal = "" & Replace(fp_formVal,"","'") & ""
```

```
            Else
```

```
                fp_formVal = fp_delim + fp_formVal + fp_delim
```

```
            End If
```

```

End If

If (fp_i <> LBound(fp_fields)) Then

    fp_tableValues = fp_tableValues & ","

    fp_dbValues = fp_dbValues & ","

End If

fp_tableValues = fp_tableValues & fp_columns(fp_i)

fp_dbValues = fp_dbValues & fp_formVal

Next

fp_editQuery = "insert into " & fp_editTable & " (" & fp_tableValues & ") values (" &
fp_dbValues & ")"

If (Not fp_abortEdit) Then

    ' execute the insert

    Set fp_editCmd = Server.CreateObject("ADODB.Command")

    fp_editCmd.ActiveConnection = fp_editConnection

    fp_editCmd.CommandText = fp_editQuery

    fp_editCmd.Execute

    fp_editCmd.ActiveConnection.Close

    If (fp_editRedirectUrl <> "") Then

        Response.Redirect(fp_editRedirectUrl)

    End If

End If

End If

%>

<%

Dim Recordset1

```



```

Dim Recordset1_numRows

Recordset1.Open "SELECT * FROM Kazalar"

Set Recordset1 = Server.CreateObject("ADODB.Recordset")

Recordset1.ActiveConnection = fp_recordset_STRING

Recordset1.Source = "SELECT * FROM Kazalar"

Recordset1.CursorType = 0

Recordset1.CursorLocation = 2

Recordset1.LockType = 1

Recordset1.Open()

Recordset1_numRows = 0

' *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

Dim Recordset1_total

Dim Recordset1_first

Dim Recordset1_last

'set the record count

Recordset1_total = Recordset1.RecordCount

'set the number of rows displayed on this page

If (Recordset1_numRows < 0) Then

    Recordset1_numRows = Recordset1_total

Elseif (Recordset1_numRows = 0) Then

    Recordset1_numRows = 1

End If

'set the first and last displayed record

```

```

Recordset1_first = 1

Recordset1_last = Recordset1_first + Recordset1_numRows - 1

'we have the correct record count, check the other stats
If Recordset1_total <> -1) Then

    If Recordset1_first > Recordset1_total) Then

        Recordset1_first = Recordset1_total

    End If

    If Recordset1_last > Recordset1_total) Then

        Recordset1_last = Recordset1_total

    End If

    If (Recordset1_numRows > Recordset1_total) Then

        Recordset1_numRows = Recordset1_total

    End If

End If

'*** Recordset Stats: if we don't know the record count, manually count them
If (Recordset1_total = -1) Then

    'count the total records by iterating through the recordset
    Recordset1_total=0

    While (Not Recordset1.EOF)

        Recordset1_total = Recordset1_total + 1

        Recordset1.MoveNext

    Wend

    'reset the cursor to the beginning

```

```

If(Recordset1.CursorType > 0) Then
    Recordset1.MoveFirst
Else
    Recordset1.Requery
End If

'set the number of rows displayed on this page
If(Recordset1_numRows < 0 Or Recordset1_numRows > Recordset1_total) Then
    Recordset1_numRows = Recordset1_total
End If

'set the first and last displayed record
Recordset1_first = 1
Recordset1_last = Recordset1_first + Recordset1_numRows - 1

If (Recordset1_first > Recordset1_total) Then
    Recordset1_first = Recordset1_total
End If

If (Recordset1_last > Recordset1_total) Then
    Recordset1_last = Recordset1_total
End If
End If

%>

<script language="JavaScript" type="text/JavaScript">
<!--

function fp_reloadPage(init) { //reloads the window if Nav4 resized
    if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4))
    {

```

```

document.fp_pgW=innerWidth; document.fp_pgH=innerHeight; onresize=fp_reloadPage; }}

if (innerWidth!=document.fp_pgW || innerHeight!=document.fp_pgH) location.reload();

fp_reloadPage(true);

</script>

<form method="post" action="<%=fp_editAction%>" name="form1">

<table align="center" width="75%" border="1">

<tr>

<td height="60" align="center"></td>

</tr>

</table>

<table align="center" width="75%" border="1">

<tr>

<td align="center"></td>

</tr>

</table>

<p>&nbsp;</p>

<table width="75%" border="0" align="center">

<tr>

<td align="center"><font color="#000066" size="4" face="Times New Roman, Times,

serif"></font></td>

</tr>

```



```

<table><br>
<table align="center">
<tr valign="baseline">
<td width="88" align="right" nowrap>Kaza_Id:</td>
<td width="192" align="left" nowrap>
<p align="center">
<input name="Kaza_Id" value='<%= (Recordset1_total)+1%>' size="32" style="float:
left">
</td>
<td width="112">Faktör:</td>
<td width="112"><input type="text" name="Faktr" value="" size="32"></td>
</tr>
<tr valign="baseline">
<td nowrap align="right">Sehir:</td>
<td nowrap align="right"><input type="text" name="Sehir" value="" size="32"></td>
<td>Hasar:</td>
<td><input type="text" name="Hasar" value="" size="32"> </td>
</tr>
<tr valign="baseline">
<td nowrap align="right">Ic_Dis:</td>
<td nowrap align="right"><input type="text" name="Ic_Dis" value="" size="32"></td>
<td>Hava_durumu:</td>
<td><input type="text" name="Hava_durumu" value="" size="32"> </td>
</tr>
<tr valign="baseline">

```

```

<td nowrap align="right">Saat:</td>

<td nowrap align="right"><input type="text" name="Saat" value="" size="32"></td>

<td>Neden:</td>

<td><input type="text" name="Neden" value="" size="32"> </td>

<tr>

<td valign="baseline">

<td nowrap align="right">Gün:</td>

<td nowrap align="right"><input type="text" name="Gn" value="" size="32"></td>

<td>Olü Sayısı:</td>

<td><input type="text" name="Olmlm" value="" size="32"> </td>

<tr>

<td valign="baseline">

<td nowrap align="right">Isiklandirma:</td>

<td nowrap align="right"><input type="text" name="Isiklandirma" value=""

size="32"></td>

<td>Yol_Durumu:</td>

<td><input type="text" name="Yol_Durumu" value="" size="32"> </td>

<tr>

<table>

<table align="center" width="75%" border="0">

<tr>

<td align="center"><a href="tesekkur.htm"><input type="submit" name="submit"

value="Detaylari G&ouml;nder"></a></td>

</tr>

</table>

```

 </p>

 </p>

<input type="hidden" name="fp_insert" value="form1">

</form>

 </p>

</body>

Recordset1.Close()

Set Recordset1 = Nothing

</body>

</html>

Sirücü.asp

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Polis Genel Müdürlüğü Trafik Şubesi</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">

</head>

<@LANGUAGE="VBSCRIPT"%>

<!--#include file="Connections/recordset.asp" -->

<%

*** Edit Operations: declare variables

Dim fp_editAction

Dim fp_abortEdit

Dim fp_editQuery

Dim fp_editCmd

```

Dim fp_editConnection
Dim fp_editTable
Dim fp_editRedirectUrl
Dim fp_editColumn
Dim fp_recordId
Dim fp_fieldsStr
Dim fp_columnsStr
Dim fp_fields
Dim fp_columns
Dim fp_typeArray
Dim fp_formVal
Dim fp_delim
Dim fp_altVal
Dim fp_emptyVal
Dim fp_i

fp_editAction = CStr(Request.ServerVariables("SCRIPT_NAME"))

If (Request.QueryString <> "") Then
    fp_editAction = fp_editAction & "?" & Request.QueryString
End If

'boolean to abort record edit
fp_abortEdit = false

'query string to execute
fp_editQuery = ""

>
<%

```



```

Insert Record: set variables

If Request("fp_insert") = "form1" Then
    fp_editConnection = fp_recordset_STRING
    fp_editTable = "sürücü"
    fp_editRedirectUrl = "arac.asp"
    fp_fieldsStr = "Kara_Id|value|Src_Id|value|Tahsil|value|Ehliyet|value|Emniyet_Kemeri|value|Alkol_Orani|va
    lue|Gors_Mesafesi|value"
    fp_columnsStr = "Kara_Id|none,none,NULL|Sürücü_Id|none,none,NULL|Tahsil|none,"|Ehliyet|none,"|Emniy
    et_Kemeri|none,"|Alkol_Orani|none,"|Görüs_Mesafesi|none,""
    'create the fp_fields and fp_columns arrays
    fp_fields = Split(fp_fieldsStr, "|")
    fp_columns = Split(fp_columnsStr, "|")
    'set the form values
    For fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2
        fp_fields(fp_i+1) = CStr(Request.Form(fp_fields(fp_i)))
    Next
    'append the query string to the redirect URL
    If (fp_editRedirectUrl <> "" And Request.QueryString <> "") Then
        If (InStr(1, fp_editRedirectUrl, "?", vbTextCompare) = 0 And Request.QueryString <> "")
            Then
                fp_editRedirectUrl = fp_editRedirectUrl & "?" & Request.QueryString
            Else

```

```

fp_editRedirectUrl = fp_editRedirectUrl & "&" & Request.QueryString

End If

End If

End If

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

Dim fp_tableValues

Dim fp_dbValues

If CStr(Request("fp_insert")) <> "" Then

    'create the sql insert statement

    fp_tableValues = ""

    fp_dbValues = ""

    For fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2

        fp_formVal = fp_fields(fp_i+1)

        fp_typeArray = Split(fp_columns(fp_i+1),",")

        fp_delim = fp_typeArray(0)

        If (fp_delim = "none") Then fp_delim = ""

        fp_altVal = fp_typeArray(1)

        If (fp_altVal = "none") Then fp_altVal = ""

        fp_emptyVal = fp_typeArray(2)

        If (fp_emptyVal = "none") Then fp_emptyVal = ""

        If (fp_formVal = "") Then

            fp_formVal = fp_emptyVal

        Else

```

```

If (fp_altVal <> "") Then
    fp_formVal = fp_altVal
ElseIf (fp_delim = "") Then ' escape quotes
    fp_formVal = "" & Replace(fp_formVal,"","'") & ""
Else
    fp_formVal = fp_delim + fp_formVal + fp_delim
End If

End If

If (fp_i <> LBound(fp_fields)) Then
    fp_tableValues = fp_tableValues & ","
    fp_dbValues = fp_dbValues & ","
End If

fp_tableValues = fp_tableValues & fp_columns(fp_i)
fp_dbValues = fp_dbValues & fp_formVal

Next

fp_editQuery = "insert into " & fp_editTable & " (" & fp_tableValues & ") values (" &
fp_dbValues & ")"

If (Not fp_abortEdit) Then
    ' execute the insert
    Set fp_editCmd = Server.CreateObject("ADODB.Command")
    fp_editCmd.ActiveConnection = fp_editConnection
    fp_editCmd.CommandText = fp_editQuery
    fp_editCmd.Execute
    fp_editCmd.ActiveConnection.Close
    If (fp_editRedirectUrl <> "") Then

```

```

    Response.Redirect(fp_editRedirectUrl)

End If

End If

End If

%>

<%

Dim Recordset1

Dim Recordset1_numRows

Set Recordset1 = Server.CreateObject("ADODB.Recordset")

Recordset1.ActiveConnection = fp_recordset_STRING

Recordset1.Source = "SELECT * FROM sürücü"

Recordset1.CursorType = 0

Recordset1.CursorLocation = 2

Recordset1.LockType = 1

Recordset1.Open()

Recordset1_numRows = 0

%>

<%

' *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

Dim Recordset1_total

Dim Recordset1_first

Dim Recordset1_last

' set the record count

Recordset1_total = Recordset1.RecordCount

' set the number of rows displayed on this page

```


If (Recordset1_numRows < 0) Then

Recordset1_numRows = Recordset1_total

Elseif (Recordset1_numRows = 0) Then

Recordset1_numRows = 1

End If

*set the first and last displayed record

Recordset1_first = 1

Recordset1_last = Recordset1_first + Recordset1_numRows - 1

*if we have the correct record count, check the other stats

If (Recordset1_total <> -1) Then

If (Recordset1_first > Recordset1_total) Then

Recordset1_first = Recordset1_total

End If

If (Recordset1_last > Recordset1_total) Then

Recordset1_last = Recordset1_total

End If

If (Recordset1_numRows > Recordset1_total) Then

Recordset1_numRows = Recordset1_total

End If

End If

%>

<%

' *** Recordset Stats: if we don't know the record count, manually count them

If (Recordset1_numRows = -1) Then

If (Recordset1_total = -1) Then

' count the total records by iterating through the recordset

Recordset1_total=0

While (Not Recordset1.EOF)

Recordset1_total = Recordset1_total + 1

Recordset1.MoveNext

Wend

' reset the cursor to the beginning

If (Recordset1.CursorType > 0) Then

Recordset1.MoveFirst

Else

Recordset1.Requery

End If

' set the number of rows displayed on this page

If (Recordset1_numRows < 0 Or Recordset1_numRows > Recordset1_total) Then

Recordset1_numRows = Recordset1_total

End If

' set the first and last displayed record

Recordset1_first = 1

Recordset1_last = Recordset1_first + Recordset1_numRows - 1

If (Recordset1_first > Recordset1_total) Then

Recordset1_first = Recordset1_total

End If

If (Recordset1_last > Recordset1_total) Then

Recordset1_last = Recordset1_total

```

End If
End If
%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Poli Genel Müdürlüğü Trafik Şubesi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<form method="POST" action="%=fp_editAction%" name="form1">
<table align="center" width="75%" border="1">
<tr>
<td height="60" align="center"></td>
</tr>
</table>
<table align="center" width="75%" border="1">
<tr>
<td align="center"></td>
</tr>
</table>
<p>&nbsp;</p><table width="75%" border="0" align="center">
<tr>

```

```
<td align="center"><font color="#000066" size="4" face="Times New Roman, Times, serif"></font></td>
```

```
</tr>
```

```
</table><br>
```

```
<table align="center">
```

```
<tr valign="baseline">
```

```
<td nowrap align="right">Kaza_Id:</td>
```

```
<td> <input type="text" name="Kaza_Id" value="<%= (Recordset1_total)+1%>" size="32">
```

```
</td>
```

```
</tr>
```

```
<tr valign="baseline">
```

```
<td nowrap align="right">Sürücü_Id:</td>
```

```
<td> <input name="Src_Id" type="text" size="32"> </td>
```

```
</tr>
```

```
<tr valign="baseline">
```

```
<td nowrap align="right">Tahsil:</td>
```

```
<td> <input type="text" name="Tahsil" value="" size="32"> </td>
```

```
</tr>
```

```
<tr valign="baseline">
```

```
<td nowrap align="right">Ehliyet:</td>
```

```
<td> <input type="text" name="Ehliyet" value="" size="32"> </td>
```

```
</tr>
```

```
<tr valign="baseline">
```

```
<td nowrap align="right">Emniyet_Kemeri:</td>
```



```

        <td> <input type="text" name="Emniyet_Kemeri" value="" size="32"> </td>

</tr>

<tr valign="baseline">

        <td nowrap align="right">Alkol_Oran&#305;:</td>

        <td> <input type="text" name="Alkol_Orani" value="" size="32"> </td>

</tr>

<tr valign="baseline">

        <td nowrap align="right">Görüs_Mesafesi:</td>

        <td> <input type="text" name="Gors_Mesafesi" value="" size="32"> </td>

</tr>

</table>

<table align="center" width="75%" border="0">

        <tr align="center">

                <td align="center"><input name="submit" type="submit" value="Detaylar&yacute;

G&ouml;nderiniz"></td>

        </tr>

</table>

<input type="hidden" name="fp_insert" value="form1">

</form>

<p><a href="../asp/sag.asp.htm">GERI</a></p>

</body>

</html>

<%

Recordset1.Close()

Set Recordset1 = Nothing

```

%>

</html>

Araç.asp

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Polis Genel Müdürlüğü Trafik Şubesi</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">

</head>

<%@LANGUAGE="VBSCRIPT"%>

<!--#include file="Connections/recordset.asp" -->

<%

' *** Edit Operations: declare variables

Dim fp_editAction

Dim fp_abortEdit

Dim fp_editQuery

Dim fp_editCmd

Dim fp_editConnection

Dim fp_editTable

Dim fp_editRedirectUrl

Dim fp_editColumn

Dim fp_recordId

Dim fp_fieldsStr

Dim fp_columnsStr

Dim fp_fields

```

Dim fp_columns
Dim fp_typeArray
Dim fp_formVal
Dim fp_delim
Dim fp_altVal
Dim fp_emptyVal
Dim fp_i

fp_editAction = CStr(Request.ServerVariables("SCRIPT_NAME"))

If (Request.QueryString <> "") Then

    fp_editAction = fp_editAction & "?" & Request.QueryString

End If

' boolean to abort record edit

fp_abortEdit = false

' query string to execute

fp_editQuery = ""

%>

<%

'**** Insert Record: set variables

If (CStr(Request("fp_insert")) = "form1") Then

    fp_editConnection = fp_recordset_STRING

    fp_editTable = "araç"

    fp_editRedirectUrl = "tesekkur.htm"

    fp_fieldsStr = "Kaza_Id|value|Ara_yili|value|Direksiyon|value"

    fp_columnsStr = "Kaza_Id|none,none,NULL|Araç_yili|none,NULL|Direksiyon|none,""

    ' create the fp_fields and fp_columns arrays

```

```

fp_fields = Split(fp_fieldsStr, "|")

fp_columns = Split(fp_columnsStr, "|")

' set the form values

For fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2

    fp_fields(fp_i+1) = CStr(Request.Form(fp_fields(fp_i)))

Next

' append the query string to the redirect URL

If (fp_editRedirectUrl <> "" And Request.QueryString <> "") Then

    If (InStr(1, fp_editRedirectUrl, "?", vbTextCompare) = 0 And Request.QueryString <> "")

Then

        fp_editRedirectUrl = fp_editRedirectUrl & "?" & Request.QueryString

    Else

        fp_editRedirectUrl = fp_editRedirectUrl & "&" & Request.QueryString

    End If

End If

End If

%>

<%

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

Dim fp_tableValues

Dim fp_dbValues

If (CStr(Request("fp_insert")) <> "") Then

' create the sql insert statement

fp_tableValues = ""

```



```

fp_dbValues = ""

For fp_i = LBound(fp_fields) To UBound(fp_fields) Step 2

    fp_formVal = fp_fields(fp_i+1)

    fp_typeArray = Split(fp_columns(fp_i+1),",")

    fp_delim = fp_typeArray(0)

    If (fp_delim = "none") Then fp_delim = ""

    fp_altVal = fp_typeArray(1)

    If (fp_altVal = "none") Then fp_altVal = ""

    fp_emptyVal = fp_typeArray(2)

    If (fp_emptyVal = "none") Then fp_emptyVal = ""

    If (fp_formVal = "") Then

        fp_formVal = fp_emptyVal

    Else

        If (fp_altVal <> "") Then

            fp_formVal = fp_altVal

        ElseIf (fp_delim = "") Then ' escape quotes

            fp_formVal = "" & Replace(fp_formVal,"","'") & ""

        Else

            fp_formVal = fp_delim + fp_formVal + fp_delim

        End If

    End If

    fp_tableValues = fp_tableValues & ","

    fp_dbValues = fp_dbValues & ","

End If

```

```

fp_tableValues = fp_tableValues & fp_columns(fp_i)

fp_dbValues = fp_dbValues & fp_formVal

Next

fp_editQuery = "insert into " & fp_editTable & " (" & fp_tableValues & ") values (" &
fp_dbValues & ")"

If (Not fp_abortEdit) Then
    ' execute the insert

    Set fp_editCmd = Server.CreateObject("ADODB.Command")

    fp_editCmd.ActiveConnection = fp_editConnection

    fp_editCmd.CommandText = fp_editQuery

    fp_editCmd.Execute

    fp_editCmd.ActiveConnection.Close

    If (fp_editRedirectUrl <> "") Then

        Response.Redirect(fp_editRedirectUrl)

    End If

End If

End If

%>

<%

Dim Recordset1__fpColParam

Recordset1__fpColParam = "1"

If (Request.QueryString("Kaza_Id") <> "") Then

    Recordset1__fpColParam = Request.QueryString("Kaza_Id")

End If

%>

```

<%

Dim Recordset1

Dim Recordset1_numRows

Set Recordset1 = Server.CreateObject("ADODB.Recordset")

Recordset1.ActiveConnection = fp_recordset_STRING

Recordset1.Source = "SELECT * FROM araç WHERE Kaza_Id = " +

Replace(Recordset1__fpColParam, "", "") + " ORDER BY Kaza_Id ASC"

Recordset1.CursorType = 0

Recordset1.CursorLocation = 2

Recordset1.LockType = 1

Recordset1.Open()

Recordset1_numRows = 0

%>

<%

Dim Recordset2

Dim Recordset2_numRows

Set Recordset2 = Server.CreateObject("ADODB.Recordset")

Recordset2.ActiveConnection = fp_recordset_STRING

Recordset2.Source = "SELECT * FROM Kazalar"

Recordset2.CursorType = 0

Recordset2.CursorLocation = 2

Recordset2.LockType = 1

Recordset2.Open()

Recordset2_numRows = 0

%>

<%

' *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

Dim Recordset2_total

Dim Recordset2_first

Dim Recordset2_last

' set the record count

Recordset2_total = Recordset2.RecordCount

' set the number of rows displayed on this page

If (Recordset2_numRows < 0) Then

Recordset2_numRows = Recordset2_total

Elseif (Recordset2_numRows = 0) Then

Recordset2_numRows = 1

End If

' set the first and last displayed record

Recordset2_first = 1

Recordset2_last = Recordset2_first + Recordset2_numRows - 1

' if we have the correct record count, check the other stats

If (Recordset2_total < -1) Then

If (Recordset2_first > Recordset2_total) Then

Recordset2_first = Recordset2_total

End If

If (Recordset2_last > Recordset2_total) Then

Recordset2_last = Recordset2_total

End If


```

If (Recordset2_numRows > Recordset2_total) Then
Recordset2_numRows = Recordset2_total
End If

If
>
%
** Recordset Stats: if we don't know the record count, manually count them

(Recordset2_total = -1) Then

count the total records by iterating through the recordset

Recordset2_total=0

While (Not Recordset2.EOF)

Recordset2_total = Recordset2_total + 1

Recordset2.MoveNext

Vend

reset the cursor to the beginning

If (Recordset2.CursorType > 0) Then

Recordset2.MoveFirst

Else

Recordset2.Requery

End If

set the number of rows displayed on this page

If (Recordset2_numRows < 0 Or Recordset2_numRows > Recordset2_total) Then

Recordset2_numRows = Recordset2_total

End If

set the first and last displayed record

```

```

Recordset2_first = 1

Recordset2_last = Recordset2_first + Recordset2_numRows - 1

If (Recordset2_first > Recordset2_total) Then

Recordset2_first = Recordset2_total

End If

If (Recordset2_last > Recordset2_total) Then

Recordset2_last = Recordset2_total

End If

End If

%>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

```

```

<html>

```

```

<head>

```

```

<title>Untitled Document</title>

```

```

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

```

```

</head>

```

```

<body>

```

```

<form method="POST" action="<%=fp_editAction%>" name="form1">

```

```

<table align="center" width="75%" border="1">

```

```

<tr>

```

```

<td height="60" align="center"></td>

```

```

</tr>

```

```

</table>

```

```

<table align="center" width="75%" border="1">

```

```
</table>
```

```
<table width="75%" border="0" align="center">
```

```
<tr>
```

```
<td align="center" ><a href="Kazalar_ekleme.asp"><input name="submit"
```

```
type="submit" value="Detaylari G&ouml;nderiniz"></a></td>
```

```
</tr>
```

```
</table>
```

```
<input type="hidden" name="fp_insert" value="form1">
```

```
</form>
```

```
<p>&nbsp;</p>
```

```
</body>
```

```
</html>
```

```
<%
```

```
Recordset1.Close()
```

```
Set Recordset1 = Nothing
```

```
%>
```

```
<%
```

```
Recordset2.Close()
```

```
Set Recordset2 = Nothing
```

```
%>
```

```
</html>
```

```
Tesekkür.asp
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

```
<html>
```

```
<head>
```

```

<title>Untitled Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

</head>

<body>

<table width="75%" border="0" align="center">

  <tr>

    <td height="60" align="center"></td>

  </tr>

  <tr>

    <td height="35" align="center">

      <p></p></td>

    </tr>

  </table>

  <p>&nbsp;</p>

  <p>&nbsp;</p>

  <table width="75%" border="0" align="center">

    <tr>

      <td></td>

    </tr>

    <tr>

      <td></td>

    </tr>

  </table>

  <p align="center">&nbsp;</p>

```


</body>

</html>

Kaza detay.asp

<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>

<!--#include file="Connections/recordset.asp" -->

<%

Dim Recordset1

Dim Recordset1_numRows

Set Recordset1 = Server.CreateObject("ADODB.Recordset")

Recordset1.ActiveConnection = fp_recordset_STRING

Recordset1.Source = "SELECT * FROM Kazalar ORDER BY Kaza_Id ASC"

Recordset1.CursorType = 0

Recordset1.CursorLocation = 2

Recordset1.LockType = 1

Recordset1.Open()

Recordset1_numRows = 0

%>

<%

Dim Repeat1__numRows

Dim Repeat1__index

Repeat1__numRows = -1

Repeat1__index = 0

Recordset1_numRows = Recordset1_numRows + Repeat1__numRows

%>

<%

' *** Recordset Stats, Move To Record, and Go To Record: declare stats variables

Dim Recordset1_total

Dim Recordset1_first

Dim Recordset1_last

' set the record count

Recordset1_total = Recordset1.RecordCount

' set the number of rows displayed on this page

If (Recordset1_numRows < 0) Then

Recordset1_numRows = Recordset1_total

Elseif (Recordset1_numRows = 0) Then

Recordset1_numRows = 1

End If

' set the first and last displayed record

Recordset1_first = 1

Recordset1_last = Recordset1_first + Recordset1_numRows - 1

' if we have the correct record count, check the other stats

If (Recordset1_total < -1) Then

If (Recordset1_first > Recordset1_total) Then

Recordset1_first = Recordset1_total

End If

If (Recordset1_last > Recordset1_total) Then

Recordset1_last = Recordset1_total

End If

If (Recordset1_numRows > Recordset1_total) Then

Recordset1_numRows = Recordset1_total

End If

1 If

Recordset1 (If I = 0 - 1) Then

* Recordset Stats: if we don't know the record count, manually count them

Recordset1_total = -1) Then

Count the total records by iterating through the recordset

Recordset1_total=0

While (Not Recordset1.EOF)

Recordset1_total = Recordset1_total + 1

Recordset1.MoveNext

End

Reset the cursor to the beginning

(Recordset1.CursorType > 0) Then

Recordset1.MoveFirst

Else

Recordset1.Requery

End If

Set the number of rows displayed on this page

(Recordset1_numRows < 0 Or Recordset1_numRows > Recordset1_total) Then

Recordset1_numRows = Recordset1_total

End If

Set the first and last displayed record

Recordset1_first = 1

```
Recordset1_last = Recordset1_first + Recordset1_numRows - 1
```

```
If (Recordset1_first > Recordset1_total) Then
```

```
Recordset1_first = Recordset1_total
```

```
End If
```

```
If (Recordset1_last > Recordset1_total) Then
```

```
Recordset1_last = Recordset1_total
```

```
End If
```

```
End If
```

```
%>
```

```
<%
```

```
Dim fp_paramName
```

```
%>
```

```
<%
```

```
' *** Move To Record and Go To Record: declare variables
```

```
Dim fp_rs
```

```
Dim fp_rsCount
```

```
Dim fp_size
```

```
Dim fp_uniqueCol
```

```
Dim fp_offset
```

```
Dim fp_atTotal
```

```
Dim fp_paramIsDefined
```

```
Dim fp_param
```

```
Dim fp_index
```

```
Set fp_rs = Recordset1
```



```

fp_rsCount = Recordset1_total

fp_size = Recordset1_numRows

fp_uniqueCol = ""

fp_paramName = ""

fp_offset = 0

fp_atTotal = false

fp_paramIsDefined = false

If (fp_paramName <> "") Then

    fp_paramIsDefined = (Request.QueryString(fp_paramName) <> "")

End If

%>

<%

' *** Move To Record: handle 'index' or 'offset' parameter

if (Not fp_paramIsDefined And fp_rsCount <> 0) then

    ' use index parameter if defined, otherwise use offset parameter

    fp_param = Request.QueryString("index")

    If (fp_param = "") Then

        fp_param = Request.QueryString("offset")

    End If

    If (fp_param <> "") Then

        fp_offset = Int(fp_param)

    End If

    ' if we have a record count, check if we are past the end of the recordset

    If (fp_rsCount <> -1) Then

        If (fp_offset >= fp_rsCount Or fp_offset = -1) Then ' past end or move last

```

```

If ((fp_rsCount Mod fp_size) > 0) Then      ' last page not a full repeat region
    fp_offset = fp_rsCount - (fp_rsCount Mod fp_size)
Else
    fp_offset = fp_rsCount - fp_size
End If
End If
End If

' move the cursor to the selected record
fp_index = 0

While ((Not fp_rs.EOF) And (fp_index < fp_offset Or fp_offset = -1))
    fp_rs.MoveNext
    fp_index = fp_index + 1
Wend

If (fp_rs.EOF) Then
    fp_offset = fp_index ' set fp_offset to the last possible record
End If
End If

%>

<%

*** Move To Record: if we dont know the record count, check the display range

If (fp_rsCount = -1) Then
    ' walk to the end of the display range for this page
    fp_index = fp_offset
    While (Not fp_rs.EOF And (fp_size < 0 Or fp_index < fp_offset + fp_size))
        fp_rs.MoveNext
    
```

```

    fp_index = fp_index + 1

Wend

' if we walked off the end of the recordset, set fp_rsCount and fp_size

If (fp_rs.EOF) Then

    fp_rsCount = fp_index

    If (fp_size < 0 Or fp_size > fp_rsCount) Then

        fp_size = fp_rsCount

    End If

End If

' if we walked off the end, set the offset based on page size

If (fp_rs.EOF And Not fp_paramIsDefined) Then

    If (fp_offset > fp_rsCount - fp_size Or fp_offset = -1) Then

        If ((fp_rsCount Mod fp_size) > 0) Then

            fp_offset = fp_rsCount - (fp_rsCount Mod fp_size)

        Else

            fp_offset = fp_rsCount - fp_size

        End If

    End If

End If

' reset the cursor to the beginning

If (fp_rs.CursorType > 0) Then

    fp_rs.MoveFirst

Else

    fp_rs.Requery

End If

```

```

' move the cursor to the selected record

fp_index = 0

While (Not fp_rs.EOF And fp_index < fp_offset)

    fp_rs.MoveNext

    fp_index = fp_index + 1

Wend

End If

%>

<%

' *** Move To Record: update recordset stats

' set the first and last displayed record

Recordset1_first = fp_offset + 1

Recordset1_last = fp_offset + fp_size

If (fp_rsCount <> -1) Then

    If (Recordset1_first > fp_rsCount) Then

        Recordset1_first = fp_rsCount

    End If

    If (Recordset1_last > fp_rsCount) Then

        Recordset1_last = fp_rsCount

    End If

End If

' set the boolean used by hide region to check if we are on the last record

fp_atTotal = (fp_rsCount <> -1 And fp_offset + fp_size >= fp_rsCount)

%>

```



```

<%
' *** Go To Record and Move To Record: create strings for maintaining URL and Form
parameters

Dim fp_keepNone
Dim fp_keepURL
Dim fp_keepForm
Dim fp_keepBoth
Dim fp_removeList
Dim fp_item
Dim fp_nextItem

' create the list of parameters which should not be maintained
fp_removeList = "&index="

If (fp_paramName <> "") Then
    fp_removeList = fp_removeList & "&" & fp_paramName & "="
End If

fp_keepURL=""
fp_keepForm=""
fp_keepBoth=""
fp_keepNone=""

' add the URL parameters to the fp_keepURL string
For Each fp_item In Request.QueryString
    fp_nextItem = "&" & fp_item & "="

    If (InStr(1,fp_removeList,fp_nextItem,1) = 0) Then
        fp_keepURL = fp_keepURL & fp_nextItem &
Server.URLEncode(Request.QueryString(fp_item))
    End If
Next fp_item

```

```

End If

Next

' add the Form variables to the fp_keepForm string

For Each fp_item In Request.Form

    fp_nextItem = "&" & fp_item & "="

    If (InStr(1,fp_removeList,fp_nextItem,1) = 0) Then

        fp_keepForm = fp_keepForm & fp_nextItem &

Server.URLEncode(Request.Form(fp_item))

    End If

Next

' create the Form + URL string and remove the initial '&' from each of the strings

fp_keepBoth = fp_keepURL & fp_keepForm

If (fp_keepBoth <> "") Then

    fp_keepBoth = Right(fp_keepBoth, Len(fp_keepBoth) - 1)

End If

If (fp_keepURL <> "") Then

    fp_keepURL = Right(fp_keepURL, Len(fp_keepURL) - 1)

End If

If (fp_keepForm <> "") Then

    fp_keepForm = Right(fp_keepForm, Len(fp_keepForm) - 1)

End If

' a utility function used for adding additional parameters to these strings

Function fp_joinChar(firstItem)

    If (firstItem <> "") Then

        fp_joinChar = "&"
    
```

```

Else
    fp_joinChar = ""
End If

End Function

%>

<%

' *** Move To Record: set the strings for the first, last, next, and previous links

Dim fp_keepMove

Dim fp_moveParam

Dim fp_moveFirst

Dim fp_moveLast

Dim fp_moveNext

Dim fp_movePrev

Dim fp_urlStr

Dim fp_paramList

Dim fp_paramIndex

Dim fp_nextParam

fp_keepMove = fp_keepBoth

fp_moveParam = "index"

' if the page has a repeated region, remove 'offset' from the maintained parameters

If (fp_size > 1) Then

    fp_moveParam = "offset"

    If (fp_keepMove <> "") Then

        fp_paramList = Split(fp_keepMove, "&")

        fp_keepMove = ""
    
```

```

For fp_paramIndex = 0 To UBound(fp_paramList)

    fp_nextParam = Left(fp_paramList(fp_paramIndex),
InStr(fp_paramList(fp_paramIndex), "=") - 1)

    If (StrComp(fp_nextParam, fp_moveParam, 1) <> 0) Then

        fp_keepMove = fp_keepMove & "&" & fp_paramList(fp_paramIndex)

    End If

Next

If (fp_keepMove <> "") Then

    fp_keepMove = Right(fp_keepMove, Len(fp_keepMove) - 1)

End If

End If

End If

set the strings for the move to links

If (fp_keepMove <> "") Then

    fp_keepMove = fp_keepMove & "&"

End If

fp_urlStr = Request.ServerVariables("URL") & "?" & fp_keepMove & fp_moveParam & "="

fp_moveFirst = fp_urlStr & "0"

fp_moveLast = fp_urlStr & "-1"

fp_moveNext = fp_urlStr & CStr(fp_offset + fp_size)

If (fp_offset - fp_size < 0) Then

    fp_movePrev = fp_urlStr & "0"

Else

    fp_movePrev = fp_urlStr & CStr(fp_offset - fp_size)

End If

```


%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Untitled Document</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">

</head>

<body>

<table align="center" border="1">

<tr>

<td align="default" width="8%"> Kaza_Id </td>

<td align="default" width="8%"> Sehir </td>

<td align="default" width="8%"> Ic_Dis </td>

<td align="default" width="8%"> Saat </td>

<td align="default" width="8%"> Gün </td>

<td align="default" width="8%"> Isiklandirma </td>

<td align="default" width="8%"> Faktör </td>

<td align="default" width="8%"> Hasar </td>

<td align="default" width="8%"> Hava_durumu </td>

<td align="default" width="8%"> Neden </td>

<td align="default" width="8%"> Olü_sayisi </td>

<td align="default" width="8%"> Yol_Durumu </td>

</tr>

<%

```
While ((Repeat1__numRows < 0) AND (NOT Recordset1.EOF))

%>

<tr>

<td align="default" width="8%">

<a href="kaza_detay_sayfasi.asp?<%= fp_keepBoth & fp_joinChar(fp_keepBoth) &
"Kaza_Id=" & Recordset1.Fields.Item("Kaza_Id").Value
%>"><%= (Recordset1.Fields.Item("Kaza_Id").Value) %></a> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Sehir").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Ic_Dis").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Saat").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Gün").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Isiklandirma").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Faktör").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Hasar").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Hava_durumu").Value) %> &nbsp;</td>

<td align="default" width="8%">

<%= (Recordset1.Fields.Item("Neden").Value) %> &nbsp;</td>
```

```

<td align="default" width="8%">

    <%=Recordset1.Fields.Item("Olü_sayisi").Value)%> &nbsp;  </td>

<td align="default" width="8%">

    <%=Recordset1.Fields.Item("Yol_Durumu").Value)%> &nbsp;  </td>

</tr>

<%

Repeat1__index=Repeat1__index+1

Repeat1__numRows=Repeat1__numRows-1

Recordset1.MoveNext()

Wend

%>

</table>

<br>

<table border="0" width="50%" align="center">

<tr>

    <td width="23%" align="center"> <% If fp_offset < 0 Then %>

        <a href="<%=fp_moveFirst%>">First</a>

    <% End If ' end fp_offset < 0 %> </td>

    <td width="31%" align="center"> <% If fp_offset < 0 Then %>

        <a href="<%=fp_movePrev%>">Previous</a>

    <% End If ' end fp_offset < 0 %> </td>

    <td width="23%" align="center"> <% If Not fp_atTotal Then %>

        <a href="<%=fp_moveNext%>">Next</a>

    <% End If ' end Not fp_atTotal %> </td>

    <td width="23%" align="center"> <% If Not fp_atTotal Then %>

```

```

<a href="<%=fp_moveLast%>">Last</a>

<% End If ' end Not fp_atTotal %> </td>

/tr>

able>

records <%= (Recordset1_first)%> to <%= (Recordset1_last)%> of

<%= (Recordset1_total)%>

><a href=" ../asp/sag.asp.htm">GER&#304;</a></p>

body>

html>

%

Recordset1.Close()

t Recordset1 = Nothing

>

```

Kaza Detay.Sayfasi.asp

```

% @LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>

--#include file="Connections/recordset.asp" -->

%

m Recordset1__fpColParam

Recordset1__fpColParam = "1"

(Request.QueryString("Kaza_Id") <> "") Then

Recordset1__fpColParam = Request.QueryString("Kaza_Id")

nd If

>

%

im Recordset1

```



```
dset1.ActiveConnection = fp_recordset_STRING
```

```
uce(Recordset1__fpColParam, "", "") + ""
```

rdset1.CursorLocation = 2

rdset1.Open()

```
rdset1_numRows = 0
```

d>

<title>Untitled Document</title>

```
ta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

ad>

ble align="center" border="1">

>

 Kaza_Id</td> |

```
align="default"
```

h="50%"><%= (Recordset1.Fields.Item("Kaza_Id").Value)%> </td>

 $r >$ \triangleright

align="default" width="50%">Sehir</td>

align="default"

h="50%"><%=Recordset1.Fields.Item("Sehir").Value)%> </td>

>

>

align="default" width="50%">Ic_Dis</td>

align="default"

h="50%"><%=Recordset1.Fields.Item("Ic_Dis").Value)%> </td>

>

>

align="default" width="50%">Saat</td>

align="default"

h="50%"><%=Recordset1.Fields.Item("Saat").Value)%> </td>

>

>

align="default" width="50%">Gün</td>

align="default"

h="50%"><%=Recordset1.Fields.Item("Gün").Value)%> </td>

>

>

align="default" width="50%">Isiklandirma</td>

align="default"

h="50%"><%=Recordset1.Fields.Item("Isiklandirma").Value)%> </td>

>

>

align="default" width="50%">Faktör</td>

align="default"

width="50%"><%= (Recordset1.Fields.Item("Faktör").Value)%> </td>

align="default" width="50%">Hasar</td>

align="default"

width="50%"><%= (Recordset1.Fields.Item("Hasar").Value)%> </td>

align="default" width="50%">Hava_durumu</td>

align="default"

width="50%"><%= (Recordset1.Fields.Item("Hava_durumu").Value)%> </td>

align="default" width="50%">Neden</td>

align="default"

width="50%"><%= (Recordset1.Fields.Item("Neden").Value)%> </td>

align="default" width="50%">Olü_sayisi</td>

align="default"

width="50%"><%= (Recordset1.Fields.Item("Olü_sayisi").Value)%> </td>

```

align="default" width="50%">Yol_Durumu</td>

align="default"

="50%"><%=Recordset1.Fields.Item("Yol_Durumu").Value)%>&nbsp;  </td>

le>

y>

l>

Recordset1.Close()

Recordset1 = Nothing

```

er.asp

```

LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>

l>

d>

a http-equiv="Content-Language" content="tr">

a name="GENERATOR" content="Microsoft FrontPage 5.0">

a name="ProgId" content="FrontPage.Editor.Document">

a http-equiv="Content-Type" content="text/html; charset=windows-1254">

>Haberler</title>

ad>

y>

font face="Comic Sans MS"><b><u>Haberler</u></b></font></p>

```


border="5" cellpadding="0" cellspacing="0" style="border-collapse: collapse"

color="#000080" width="100%" id="AutoNumber1" height="114"

or="#FFFFFF">

width="100%" height="114">

>

i>Kuzey Kıbrıs'taki trafiğin durumu ve sorunları konulu sempozyum 30

ayıs 2005 Pazartesi Günü YDÜ AKM Salonunda gerçekleştirilecektir

i>YDÜ öğretim üyesi Dr.Ahmet Ertugan polislerin biliçlendirilmesi konulu

r eğitim vermeye başlamıştır.

i>.....

/ul>

/td>

>

le>

znbsp;</p>

znbsp;</p>

znbsp;</p>

a href="sag.asp.htm">GERİ</p>

dy>

nl>

a resimleri.asp

ul>

d>

ta name="GENERATOR" content="Microsoft FrontPage 5.0">

target="_parent" href="ahmus.asp.htm">GERI</p>

 </p>

ody>

ml>

CONCLUSION

The Internet is much more than the Web, and usability will suffer if people try to make the Web do things it is not suited for and make it the only user interface to the Internet.

The Internet allows any computer in the world to exchange data with any other computer in the world. As a result, a client program on one computer can access a server on another computer.

The Web is a hypertext system that runs over the Internet as one of its services. As a result, users can sit at any computer and browse documents that live anywhere in the world; furthermore, these documents can link to documents from any other place in the world.

The reason that we chosen the web design was the great interest on the applications of Internet and focusing on the web design.

The computer programs such as swishmax, dreamweaver, flash, access, firework and it provides to develop and focus on these programs efficiently.