



NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**Electronic Commerce Using ASP
(e-commerce)**

**Graduation Project
COM-400**

Student: Buğra DEMİRCİOĞLU (20000048)

Supervisor: Mr. Ümit İLHAN

Nicosia-2005

ACKNOWLEDGEMENTS

“First, I would like to thank my supervisor Mr. Ümit İlhan for his invaluable advice and belief in my work and myself over the course of this Graduation Project..

Second, I would like to express my gratitude to Near East University for the scholarship that made the study possible.

Third, I thank my family for thier constant encouragement and support during the preparation of this project.

Finally, I would like to thank my all friends for their advice and support.”

ABSTRACT

Electronic commerce is a means of enabling and supporting such change on a global scale. It enables companies to be more efficient and flexible in their internal operations, to work more closely with their suppliers, and to be more responsive to the needs and exceptions of their customers. It allows companies to select the best suppliers regardless of their geographical and to sell to a global market.

One special case of electronic commerce is electronic trading, in which a supplier provides goods or services to a customer in return for payment. A special case of electronic trading is electronic retailing, where the customer is an ordinary consumer rather than another company. However, while these special cases are of considerable economic importance, they are just particular examples of the more general case of any form of business operation or transaction conducted via electronic media. Other equally valid examples includes internal transactions within a single company or provision of information to an external organisation without charge.

Simply we can say that the electronic commerce is modern way to make you shopping through the internet.

TABLE OF CONTENTS

ACKNOWLEDGMENT	i
ABSTRACT	ii
TABLE OF CONTENTS	iii
INTRODUCTION	vi
CHAPTER ONE: What is the Internet	1
1.1 What is the Internet?	1
1.2 What is a Browser?	3
1.3 What is the World Wide Web and what makes it work?	3
1.4 Getting Connected to the Internet	4
1.5 New Standard Protocols	5
1.6 Is IP Characteristic	6
CHAPTER TWO: Hypertext Markup Language(HTML)	8
2.1 Introduction to HTML?	8
2.2 Elements in HTML Documents	9
2.2.1 Empty Elements	10
2.2.2 Upper and Lower Case	10
2.2.3 Elements can have Attributes	10
2.3 HTML Document Structure	11
2.3.1 Example of Document Structure	11
2.4 HTML Language(codes)	12

CHAPTER THREE: DATABASE USING ASP	18
3.1 The need for ASP	18
3.2 What is ASP?	19
3.3 Runing ASP	21
3.3.1 Internet Information Services (IIS)	21
3.3.1.1 Installing IIS	22
3.3.2 Personal Web Server	25
3.4 Begining ASP	25
3.4.1 Steps for Installation	26
3.4.2 Creating Virtual Directories	26
3.5 Accessing your Webpages	27
3.5.1 Localhost	27
3.6 The Database	27
3.6.1 DSN	28
3.6.2 Connecting to the Database	28
3.6.2.1 Creating the Guest Book Database	29
3.7 How can I create an SQL Server Database	35
3.7.1 Creating an SQL Server Database	36
3.7.2 Creating an ODBC Connection	37
CHAPTER FOUR: ELECTRONIC COMMERCE(e-commerce)	44
4.1 Introduction	44
4.2 What is Electronic Commerce?	44
4.3. Visitor (Customer) Page	46
4.3.1 Main Page	46
4.3.2 Product Presentation	47

4.3.3 Purchase Section	48
4.3.3.1 Purchase Action	49
4.3.4 Order State	50
4.3.5 Home Link	51
4.3.6 Contact Link	51
4.3.7 Transportation Info Link	52
4.4 Administrator Pages	53
4.4.1 Product List	53
4.4.2 New Product	55
4.4.3 Orders	56
4.5 Database Tables Needed For E-Commerce	57
4.5.1 Product Table	57
4.5.2 Order Table	58
4.5.3 Show Table	58
CONCLUSION	59
REFERENCES	60
APPENDIX – A	61
APPENDIX – B	107

INTRODUCTION

This project provides a brief introduction to electronic commerce: it discusses the nature of electronic commerce, considers its scope and impact, and outlines several examples. It then identifies a number of open issues and the actors responsible for addressing those issues.

Also you will find a lot of information about the ASP (Active Server Page), which is related with the electronic commerce because when you decide to make an electronic commerce web page that means you will need to use the active server page, because you are going to deal with database. For example "Global Tradepoint Network" The Global Tradepoint Network is a huge network of business information, developed under the UN-supported Electronic Trade Efficiency Programme. By interfacing to established national databases, the network aims to supply key trading information, transportation options and prices, insurance facilities, credit availability, customs requirements, and import/export regulations.

The objective of this project is to show you how the e-commerce works, you will get it through four chapters and conclusion.

Chapter one is talking about the internet in generally.

Chapter two talks about the HTML (Hypertext Markup Language), it is a language using to create a web page in the internet.

Chapter three talks about how to create a database using ASP, in the chapter you will find some ASP's codes and some lessons to how to use the ASP.

Chapter four is talking about the electronic commerce, what is it, how does it work and a lot of information about the electronic commerce.

CHAPTER ONE

WHAT IS THE INTERNET

1.1 What is the Internet?

The Internet is a Network of networks, linking computers to computers sharing the TCP/IP protocols. Each runs software to provide or “serve” information and/or to access and view information. The Internet is the transport vehicle for the information stored in files or documents on another computer. It can be compared to an international communications utility servicing computers. It is sometimes compared to a gaint international plumbing system. The Internet itself does not contain information. It is a slight misstatement to say a “document was not found on the Internet.” It would be more correct to say it was found through or using the Internet. What it was found in (or on) is one of the computers linked to the Internet.

Computers on the Internet may use one or all of the following Internet services:

- Electronic mail (e-mail). Permits you to send and receive mail. Provides access to discussion groups often called Listservs® after the software they operate under.
- Telnet or remote login. Permits your computer to log onto another computer and use it as if you were there.
- FTP or File Transfer Protocol. Allows your computer to rapidly retrieve complex files intact from a remote computer and view or save them on your computer.
- Gopher. An early, text-only method for accessing internet documents. Gopher has been almost entirely subsumed in the World Wide Web, but you may still find gopher documents linked to in web pages.

- The World Wide Web (WWW or “the Web”). The largest, fastest growing activity on the Internet.

A commonly asked question is “What is the Internet?” The reason such a question gets asked so often is because there is no agreed upon answer that neatly sums up the Internet. The Internet can be thought about in relation to its common protocols, as a physical collection of routers and circuits, as a set of shared resources, or even as an attitude about interconnecting and intercommunication. Some common definitions given in the past include:

- a network of networks based on the TCP/IP protocols,
- a community of people who use and develop those networks,
- a collection of resources that can be reached from those networks.

Today’s Internet is a global resource connecting millions of users that began as an experiment over 20 years ago by the U.S. Department of Defense. While the networks that make up the Internet are based on a standard set of protocols (a mutually agreed upon method of communication between parties), the Internet also has gateways to networks and services that are based on other protocols.

To help answer the question more completely, the rest of this paper contains an updated second chapter from “The Whole Internet User’s Guide and Catalog” BY Ed Krol (1992) that gives a more thorough explanation. (The excerpt is published through the gracious permission of the publisher, O’Reilly & Associates, Inc.)

1.2 What is a Browser?

A browser is a computer program that resides on your computer enabling you to use the computer to view WWW documents and access the Internet taking advantage of text formatting, hypertext links, images, sounds, motion and other features. Netscape and Internet explorer are currently the leading “graphical browsers” in the world (meaning they facilitate the viewing of graphics such as images and video and more.) There are other browsers (e.g.: Macweb, Opera). Most offer many of the same features and can be successfully used to retrieve documents and activate many kinds of programs.

Browsers all rely on “plug-ins” to handle the fancier files you find on the Web. Plug-ins are sub-programs stored within a browser or elsewhere in your computer especially to support special types of files you may click on. If you click on a link, and your computer does not currently have the plug-in needed for the file you clicked on, you are usually prompted with an opportunity to get the plug-in. Most plug-ins are free, and easy and safe to install on your computer; follow the instructions you are given.

The main way in which browsers differ is in the convenience features they offer for navigating and managing the Web and all the URLs you may want to keep track of. Netscape and Internet Explorer both offer the ability to e-mail documents, download them to diskette, print them, and keep track of where you have been and sites you want to “bookmark”.

1.3 What is the World Wide Web and what makes it work?

The WWW incorporates all of the services above and much more. You can retrieve documents, view images, animation, and video, listen to sound files, speak and hear voice,

and view programs that run on practically any software in the world, providing your computer has the hardware and software to do these things.

When you log onto the Internet using Netscape or Microsoft's Internet Explorer or some other browser, you are viewing documents on the World Wide Web. The current foundation on which the WWW function is the programming language called HTML . it is HTML and other programming imbedded within HTML that make possible Hypertext. Hypertext is the ability to have web pages containing links, which are areas in a page or buttons or graphics on which you can click your mouse button to retrieve another document into your computer. This "clickability" using Hypertext links is the feature which is unique and revolutionary about the Web.

How the hypertext links work? Every document or file or site or movie or soundfile or anything you find on the Web has a unique URL (uniform resource locator) that identifies what computer the thing is on, where it is within that computer, and its specific file name. Every Hypertext link on every web page in the world contains one of the URLs. When you click on a link of any kind on a Web page, you send a request to retrieve the unique document on some computer in the world that is uniquely identified by that URL. URLs are like addresses of web pages. A whole cluster of internationally accepted standards (such as TCP/IP and HTML) make possible this global information retrieval phenomenon that transcends all language boundaries.

1.4 Getting Connected to the Internet

To access the Internet by computer, you need a computer, a modem or other telecommunications link, and software to connect to an Internet Service Provider (links to more about ISPs). If you are not affiliated with the University or wish a private ISP, here is

a site where you can find ISPa of all kinds by area code. This type of technical information is beyond the scope of this tutorial and Teaching Library Workshops.

TV Set-Top Boxes such as SONY's "WEB-TV" are emerging as an alternative to PCs and MACs for viewing the Internet. You may wish to consult Yahoo's links, including opinions, on WebTV computer hardware technology.

Confused by all this jargon? See GLOSSARY of WWW and Netscape Jargon. Want help and instructions? The Teaching Library offers free drop-in classes on the Internet, WWW, Netscape, and finding information using the Internet. Click [here](#) for schedule of Teaching Library Courses.

1.5 New Standard Protocols

When I was talking about how the Internet started, I mentioned the International Standards Organization (ISO) and their set of protocol standards. Well, they finally finished designing it. Now it is an international standard, typically referred to as the ISO/OSI (Open Systems Interconnect) protocol suite. Many of the Internet's component networks allow use of OSI today. There isn't much demand, yet. The U.S. government has taken a position that government computers should be able to speak these protocols. Many have the software, but few are using it now.

It's really unclear how much demand there will be for OSI, notwithstanding the government backing. Many people felt that the current approach isn't broke, so why fix it? They are just becoming comfortable with what they have, why should they have to learn a new set of commands and terminology just because it is the standard?

Currently there are no real advantages to moving to OSI. It is more complex and less mature than IP, and hence doesn't work as efficiently. OSI does offer hope of some additional features, but it also suffers from some of the same problems which will plague IP as the network gets much bigger and faster. It's clear that some sites will convert to the OSI protocols over the next few years.

1.6 Is IP Characteristic

We further choose to define the Internet as begin those networks that use IP to permit users to use both the communication services and at least TELNET and FTP among the interactive services we have listed. This requirement for IP has been questioned by some on the basis that there are now application gateways for other protocol suites such as Novell Netware that permit use of such services. This kind of application gateway is actually nothing new, and is not yet widespread. We choose to think of such networks, at least for the moment, as yet another layer of the onion, outside the core and consumer layers of the Internet.

Other have objected to the use of IP as a defining characteristic of the Internet because they think it's too technical. Actually, we find far fewer people confused about whether a software package or network support IP than about whether it's part of the Internet or not.

Some people point out that services like WWW, Gopher, FTP, TELNET, etc. Could easily be implemented on top of other protocol suites. This is true, and has been done. However, people seem to forget to ask why these services developed on top of IP in the first place. There seems to be something about IP and the Internet that is especially conducive to the development of new protocols. We make no apologies about naming IP, because we think it is important.

There is also the question of IP to where? If you have a UNIX shell login account on a computer run by an Internet access provider, and that system has IP access to the rest of the Internet, then you are an Internet user. However, you will not be able to use the full graphical capabilities of protocols such as WWW, because the provider's system cannot display on a bitmapped screen for you. For that, you need IP to your own computer with a bitmapped screen. These are two different degrees of Internet connectivity that are important to both end users and marketer. Some people refer to them as text-only interactive access and graphical interactive access. Some people have gone so far to say have to have graphical capabilities to have a full service Internet connection. That may or may not be so, but in the interests of keeping the major categories to a minimum, we are simply going to note these degrees and say no more about them in this article. However, we agree that the distinction of graphical access is becoming more important with the spread of WWW and Mosaic.

CHAPTER TWO

HYPERTEXT MARKUP LANGUAGE (HTML)

2.1 Introduction to HTML?

HTML, or Hypertext Markup Language is designed to specify the logical organisation of a document, with important hypertext extensions. It is not designed to be the language of a WYSIWYG word processor such as Word or WordPerfect. This choice was made because the same HTML document may be viewed by many different “browsers”, of very different ability. Thus, for example, HTML allows you to mark selections of text as titles or paragraphs, and then the interpretation of these marked elements up to the browser. For example one browser may indent the beginning of a paragraph, while another may only leave a blank line.

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

The detailed rules for HTML (the names of tags/elements, how they can be used) are defined using another language known as the standard generalized markup language, or SGML. SGML is wickedly difficult, and was designed for massive document collections,

such as repair manuals for F-16 fighters, or maintenance plans for nuclear submarines. Fortunately, HTML is much simpler!

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

HTML is the lingua franca for publishing hypertext on the World Wide Web. It is a non-proprietary format based upon SGML, and can be created and processed by a wide range of tools, from simple plain text editors- you type it in from scratch- to sophisticated WYSIWYG authoring tools. HTML uses tags such as `<h1>` and `</h1>` to structure text into headings, paragraphs, lists, hypertext links etc. Here is a 10-minute guide for newcomers to HTML. W3C's statement of direction for HTML is given on the HTML Activity Statement. See also the page on our work on the next generation of Web forms, and the section on Web history.

2.2 Elements in HTML Documents

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

Most elements mark blocks of the document for particular purpose or formatting: the above `<element_name>` tag marks the beginning of such as section. The end of this section is then marked by the ending tag `</element_name>` -- note the leading slash character "/" that

appears in front of the element name in an end tag. End, or stop tags are always indicated by leading slash character.

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

2.2.1 Empty Elements

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

`<HR>`

2.2.2 Upper and Lower Case

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

2.2.3 Elements can have Attributes

Many elements can arguments that pass parameters to the interpreters handling this element. These arguments are called attributes of the elements of the element. For example, consider the element A, which marks a region of text as the beginning (or end) of a hypertext link. This element can have several attributes. One of them, HREF, specifies the hypertext

document to which the marked piece of text is linked. To specify this in the tag for A you write.

2.3 HTML Document Structure

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

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

2.3.1 Example of Document Structure

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>Environmental Change Project </TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <H1> Environmental Change Project </H1>
```

```
    <Pre>Welcome to the home page of the Environmental Change Project.
```

```
      This project is different from other projects with similar names.
```

```
      In our case we actually wish to change the climate.
```

```
      For example, we would like hot beaches in Northern Quebec,
```

```
and deserts near Chicago.</Pre>
<p>So how will we do this. Well we do the following </p>
<p>< a href="burn.html"> Burn </a> </p>
</BODY>
</HTML>
```

2.4 HTML Language(codes)

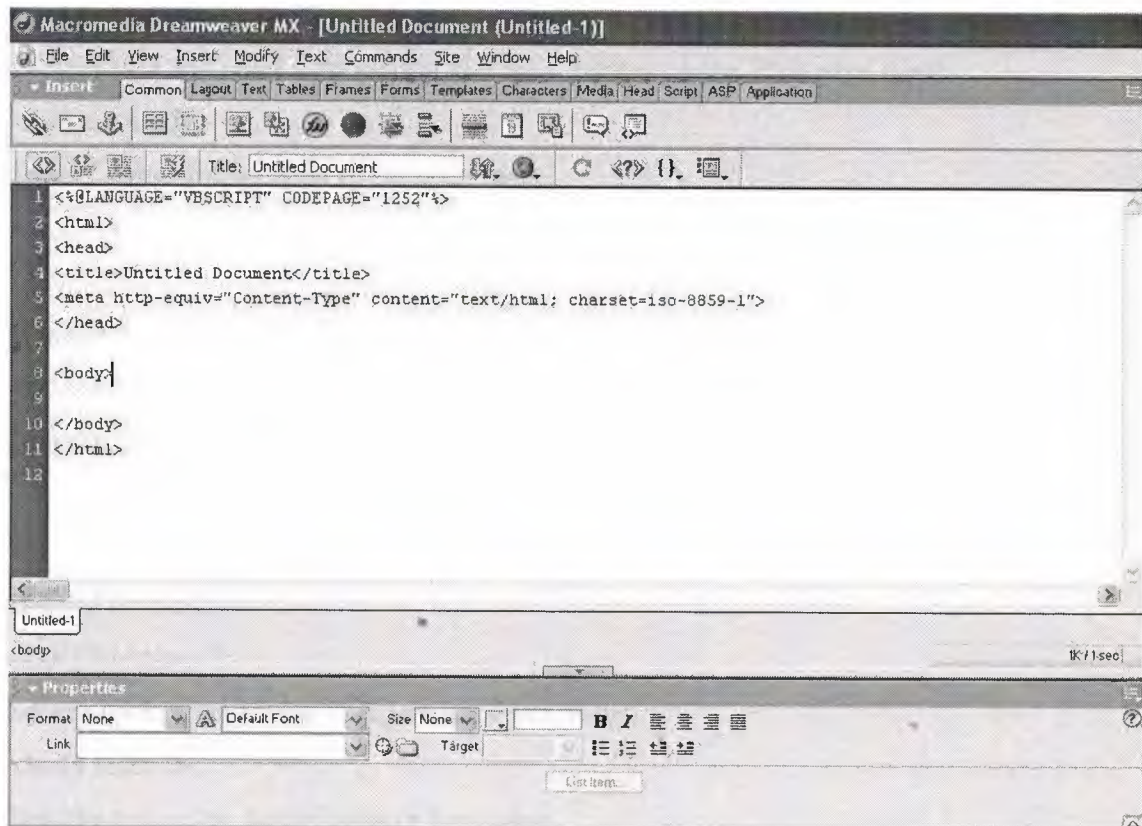


Fig. 2.1 HTML using Macromedia Dreamweaver MX

In fig. It shows that how and the HTML user going to write the codes using macromedia deramweaver MX, it is one of the best programmes in the internet programming.

And now the to show of HTML's codes and what does it means

Basic Tags

`<html></html>`

Creates an HTML document

`<head></head>`

Sets off title and other information that isn't display on the Web page itself

`<body></body>`

Setsoff the visible portion of the document

Header Tags

`<title></title>`

Puts the name of the document in the title bar

Body Attributes

`<body bgcolor=?>`

Set the background color,using name or hex value

`<body text=?>`

Sets the text color, using name or hex value

`<body link=?>`

Sets the color of links, using name or hex value

`<body vlink=?>`

Sets the color of followed links, using name or hex value

`<body alink=?>`

Sets the color of links on click

Text Tags

`<pre></pre>`

Creates preformatted text

`<h1></h1>`

Creates the largest headline

`<h6></h6>`

Creates the smallest headline

``

Creates bold text

`<i></i>`

Creates italic text

``

Emphasizes a word(with italic or bold)

``

Sets size of font, from 1 to 7

``

Sets font color, using name or hex value

Links

``

Creates a hyperlink

``

Creates a mailto link

Formatting

`<p></p>`

Creates a new paragraph

`<p align=?>`

Aligns a paragraph to the left, right, or center

`
`

Inserts a line break

`<d1></d1>`

Creates a definition list

``

Precedes each list item, and adds a number

``

Creates a bulleted list

Graphical Elements

``

Adds an image

``

Aligns an image:left, right, center; bottom, top, middle

``

Sets size of border around an image

`<hr>`

Inserts a horizontal rule

`<hr size=?>`

Sets size (height) of rule

`<hr width=?>`

Sets width of rule, in percentage or absolute value

Tables

`<table></table>`

Creates table

`<tr></tr>`

Sets off each row in a table

`<td></td>`

Sets off each cell in a row

`<th></th>`

Sets off the table header(a normal cell with bold, centered text)

Table Attributes

`<table border=#>`

Sets width of border around table cells

`<table cellspacing=#>`

Sets amount of space between table cells

`<table cellpadding=#>`

Sets amount of space between a cell's border and its contents

`<tr align=?>` or `<td align=?>`

Sets alignment for cell(s) (left, center, or right)

`<tr valign=?>` or `<td valign=?>`

Sets vertical alignment for cell(s) (top, middle, or bottom)

Forms

`<form></form>`

Creates all forms

`<select multiple name="NAME" size=?></select>`

Creates a scrolling menu. Size sets the number of menu items visible before you need to scroll.

`<option>`

Sets off each menu item

`<select name="NAME"></select>`

Creates a pulldown menu

CHAPTER THREE

DATABASES USING ASP

3.1 The need for ASP

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

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

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

On the other hand ASP lacks the platform probability that PERL (Note that with advent of Chili! ASP its not true anymore), enjoys along with resources available to PERL programmers on the Internet but it is much easier to learn and develop in. When this the exception handling in VBScript leaves a bit to be desired – which would be critical if say there was an error inserting data into a database. I did not use the debugger in the development of the code and found that most of run time errors were due to the fact that I had variables spelled wrongly or I did not include the “=” sign as part of a variable when it was embedded in HTML.

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

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

3.2 What is ASP?

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

Active Server Pages (ASP) are Web pages that contain server-side scripts in addition to the usual mixture of HTML tags. Server-side scripts are special commands you include in web pages that are processed before the pages are sent from the server to the web-browser of someone who's visiting your website. When you type a URL in the ADDRESS box or click a link on a webpage, you're asking a web-server on a computer somewhere to send a file to the web-browser (also called a "client") on your web-browser displays its contents as a combination of text, images, and sounds. In the case of an Active Server Page, the process is similar, except there's an extra processing step that takes place just before the server sends the file. ASP is also an abbreviation for application service provider.

An Active Server Page (ASP) is an HTML page that includes one or more scripts (small embedded programs) that are processed on a Microsoft Web server before the page is sent to the user. An ASP is somewhat similar to a server-side include or a common gateway interface (CGI) application in that all involves programs that run on the server, uses input received as the result of the user's request for the page to access data from a database and then build or customizes the page on fly before sending it to the requestor.

ASP is a feature of the Microsoft Internet Information Server (IIS), but, since the server-side script is just building a regular HTML page, it can be delivered to almost any browser. You can create an ASP file by including a script written VBScript or Jscript in an HTML file. You name the HTML file with ".asp" file suffix. Microsoft recommends the use of the server-side ASP rather than a client-side, where there is actually a choice, because the server-side script will result in an easily displayable HTML page. Client-side scripts (for example, with JavaScript) may not work as intended on older browsers.

3.3 Runing ASP

Since the server must do additional processing on the ASP scripts, it must have the ability to do so. The only servers which support this facility are Microsoft Internet Information Services & Microosoft Personal Web Server. Let us look at both in detail, so that you can decide which one is most suitable for you.

3.3.1 Internet Information Services (IIS)

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

IIS (Internet Information Server) is a group of Internet servers (including a Web or Hypertext Transfer Protocol server) with additional capabilities for Microsoft's Windows NT and Windows 200 Server operation system. IIS is Microsoft's entry to compete in the Internet server market that is also addressed by Aphache, sun Microsystems, O'Reilly, and others. With IIS, Microsoft including a set of programs for building and administering Web sites, a search engine, and support for writing Web_based applications that access databases. Microsoft points out that IIS is tightly integrated with the Windows NT and Windows 2000 Servers in a number of ways, resulting in faster Web page serving.

A typical company that buys IIS can create pages for Web sites using Microsoft's Front Page product (with its WYSIWYG user interface). Web developers can use Microsoft's Active Server Page (ASP) technology, which means that applications – including ActiveX controls- can be imbedded in Web pages that modify the content sent back to users. Developers can also write programs that filter requests and get the correct Web pages for different users by using Microsoft's Internet Information Server Application Program

Interface (ISAPI) interface. ASPs and ISAPI programs run more efficiently than common gateway interface (CGI) and server-side include (SSI) programs, two current technologies. (However, there are comparable interfaces on other platforms.)

Microsoft includes special capabilities for server administrators designed to appeal to Internet Service Providers (ISPs). It includes a single window (or "console") from which all service and users can be administered. It's designed to be easy to add components as snap-ins that you didn't initially install. The administrative windows can be customized for access by individual customers.

3.3.1.1 Installing IIS

If you are running Windows XP Professional on your computer you can install Microsoft's web server, Internet Information Server 5.1 (IIS) for free from the Windows XP Professional installation CD and configure it to run on your system by following the instructions below:

1. Place the Windows XP Professional CD-Rom into your CD-Rom Drive.
2. Open 'Add/Remove Windows Components' found in 'Add/Remove Programs' in the 'Control Panel'.
3. Place a tick in the check box for 'Internet Information Services(IIS)' leaving all the default installation settings intact.
4. Once IIS is installed on your computer you can view home pages in a web browser by typing 'http://localhost' (you can substitute 'localhost' for the name of your computer) into

the address bar of your web browser. If you have not placed your web site into the default directory you should now be looking at the IIS documentation.

5. if you are not sure of the name of your computer right- click on the 'My Computer' icon on your desktop, select 'properties' from the shortcut menu, and click on the 'Computer Name' tab.

6. your default web directory to place your web site in is 'C:\Inetpub\wwwroot', but if you don't want to over write the IIS documentation found in this directory you can set up your virtual directory through the 'Internet Information Services' console.

7. the 'Internet Information Services' console can be found in the 'Administrative Tools' in the 'Control Panel' under 'Performance and Maintenance', if you do not have the control panel in Classic View.

8. Double-click on the 'Internet Information Services' icon.



Figure 3.1:Administrative tool

9. Once the 'Internet Information Services' console is open you will see any IIS web services you have running on your computer including the SMRP server and FTP server, if you chose to install them with IIS.

10. To add a new virtual directory right click on 'Default Web Site' and select 'New', followed by 'Virtual Directory', from the drop down list.

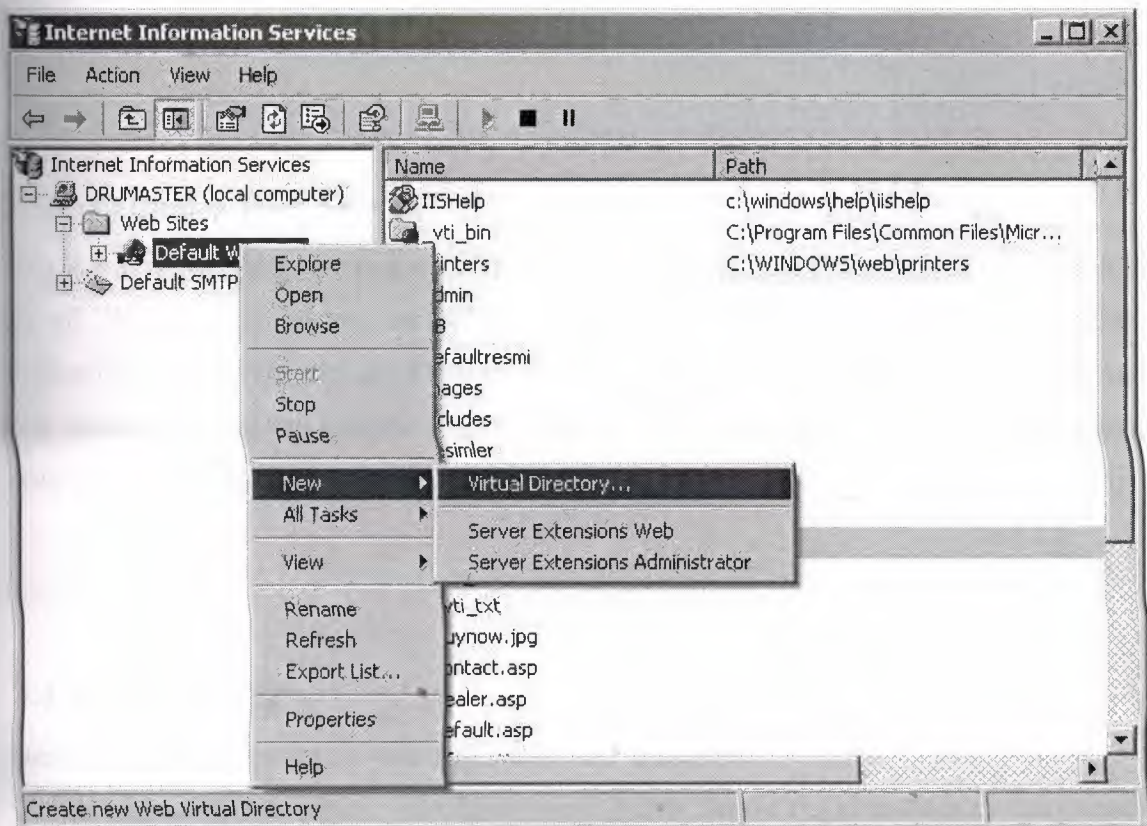


Figure 3.2: Internet Information Services

11. Next you will see the 'Virtual Directory Creation Wizard' from the first screen click the 'Next' button.

12. Next you will see a 'Browse.....' button, click on this to select the directory your web site pages are in on your computer, after which click on the 'next' button to continue.

13. On the final part of the wizard you will see a series of boxes, if you are not worried about security then select them all, if you are and want to run ASP script then check the first two, followed by the 'next' button.

3.3.2 Personal Web Server

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

3.4 Beginning ASP

Here a few quick tips before you begin your ASP session!

Unlike normal HTML pages, you can not view Active Server Pages without running a web-server. To test your own pages, you should save your pages in a directory mapped as a virtual directory, and then use your web-browser to view the page.

3.4.1 Steps for Installation

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

3.4.2 Creating Virtual Directories

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

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

3.5 Accessing your Webpages

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

3.5.1 Localhost

Let us first see, what we mean by a hostname. Whenever you connect to a remote computer using its URL, you are in effect calling it by its hostname. For example when you type in;

<http://www.google.com/>

you are really asking the network to connect to a computer named www.google.com. It is called the “hostname” of that computer. Localhost is a special hostname. It always references your own machine. So what you just did, was to try to access a web page on your own machine (which is what you wanted to do anyway). For testing all your pages, you will need to use localhost as the hostname. By the way, there is also special IP address associated with local host, that is 127.0.0.1 so you could as well have typed: <http://127.0.0.1/> and would have received the same page. To access pages in a virtual directory called [myscripts](http://localhost/myscripts/) for example, you should type in;

<http://localhost/myscripts/> in the address bar.

I hope the concept is now clear...

3.6 The Database

So let's start with the database – I used MS Access to develop the database.

3.6.1 DSN

Once you have designed your database the next step will be to create a DSN entry, UserDB1, to the this:

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

3.6.2 Connecting to the Database

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

The first post-startup request is may to the web server for any *.asp file in an application causes the Global.asa to be read. So the moment a request is may to any *.asp in the

directory in which the intranet application is stored a connection is established with the DSN User DB1. Following that the default document, in this case default.asp is processed.

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

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

You

File

3.6.2.1 Creating the Guest Book 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 same directory as the web page connecting to the database is going to be.

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

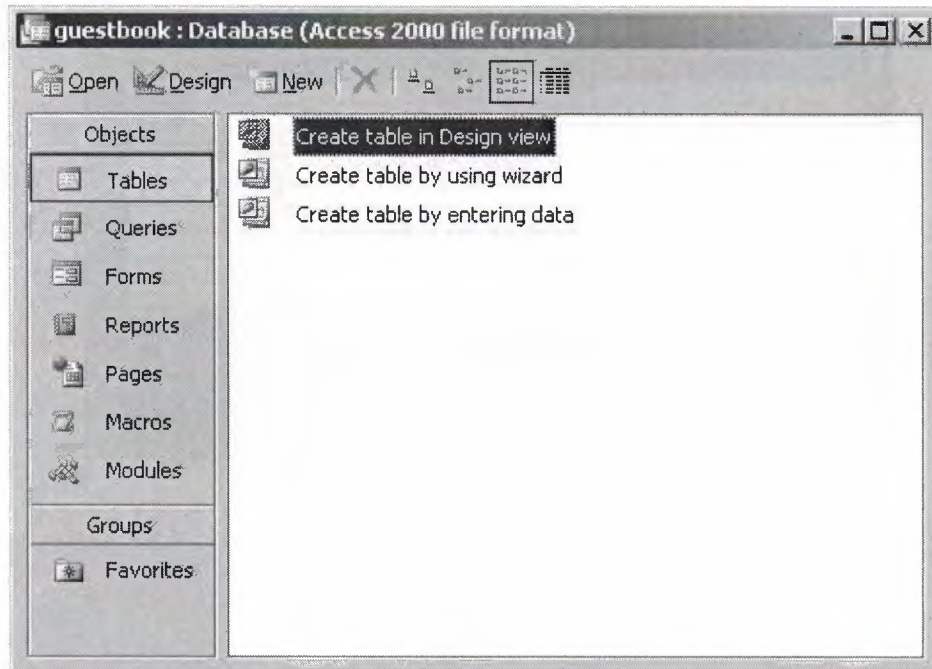


Figure 3.3: Guestbook

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

Field 1 needs to be called 'ID_no' and have the data type of 'auto number'. Also set this field as the 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.

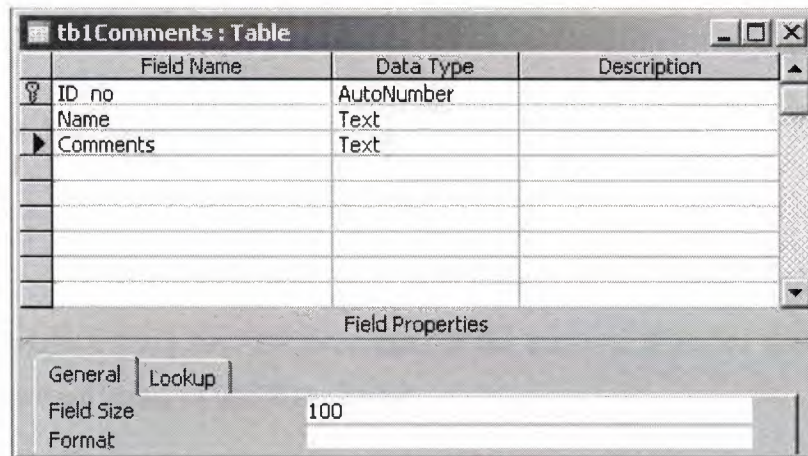


Figure 3.4: Database Table

Once all the field's have been created and the database types and primary key set, save the table as 'tb1Comments'.

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 (tb1Comments) 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.

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 >
```

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

```
<%
```

```
'Dimension variables
```

```
Dim adoCon      'Holds the Database Connection Object
```

```
Dim rsGuestbook 'Holds the recordset for the records in the database
```

```
Dim strSQL      '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. 'Create an ADO connection object set
adoCon=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 we created above to open the database by telling the connection object to use the 'Microsoft Access Driver' to open the database 'guestbook.mdb'.

You'll 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 to

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

If on the other hand 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 system DSN you will need to read my tutorial on, Setting up a system DSN 'Set an active connection to the Connection object using DSN connection `adoCon.Open "DSN=guestbook"`. Next create an ADO recordset object `Set rsGuestbook=Server.CreateObject("ADODB.Recordset")`. To query a database we need to use SQL (Structured Query Language). In the next line we initialise the variable 'strSQL' query to read in the fields 'Name' and 'Comments' from the 'tb1Comments' table. 'Initialise the strSQL variable with an SQL statement to query the database `strSQL="SELECT tb1Comments. Name, tb1Comments. Comments FROM tb1Comments;"`

Now we can open the recordset and run the SQL query on the database returning the results of 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 record in the 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 ("
")

'Move to the next record in the recordset

rsGuestbook.MoveNext

Loop

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

%>

</body>

</html>

Now call the file you have created 'guestbook.asp' and directory folder as the database, don't forget the '.asp' extension.

And that's about it, you have now created a connection to a database and display your Guestbook in a web page, now to find out how add comments to the Guestbook through a web form read the next tutorial on, Adding Data to an Access Database.

If you find that you are getting errors connecting to the database then please read through the Access Database Errors FAQ's, practically make sure you have the correct 'ODBC Drivers' installed on your system and if you are using the, 'NTFS file system', make sure the permissions are correct for the database and the directory the database in.

3.7 How can I create an SQL Server Database

In order to illustrate the process of connecting to a data source with ASP, we will need to do three things:

- Create a database
- Create an ODBC data source name (DSN)
- Create an ASP page

3.7.1 Creating an SQL Server Database

First we need a database. Since this article isn't about design, we will a very simple SQL Server database – one table! We'll name this database 15Seconds, and we will name our table t_articles.

To create the database:

1. Open SQL Server 7.0 Enterprise Manager (Start(Programs(SQL Server 7.0 (Enterprise Manager)).
2. Expand the Enterprise Manager tree, selecting the SQL Server to which you would lik to add the database, until you see the "Database" node.
3. Right click on the "Database" node and select "New Database".
4. On the "Database Properties" dialog box, enter "15Seconds" in the "Name" field.
5. Click the "OK" button.

We now have a database named "15Seconds" to which we can add our table.

To create the table, t_articles, perform the following:

1. In Enterprise Manager, expand the "Database" node.
2. Right click on the "15Seconds" node.
3. Select "Table....."
4. On the "Choose Name" dialog, Enter "t_articles" in the "Enter a name for the table:" textbox.
5. On the "Choose Name" dialog, Enter "t_articles" in the "Enter a name for the table:" textbox.

6. click the "OK" button.
7. On the "Add Table" dialog box, enter the following information.
8. On the "New Table" dialog box.

Now we have a database and table. Let's add some sample data. We'll add one record. To add data to t_articles, perform the following:

1. In Enterprise Manager, expand the "15Seconds" database node.
2. Double click on "Tables".
3. On the right side of Enterprise Manager, right click on "t_articles".
4. Select "Open Table".
5. Select "Return all rows".
6. On the "Data in Table t_articles" enter the following data.

So, there is the entire database. Pretty impressive, huh? Now that we database and a table, we need to create an ODBC connection to our database.

3.7.2 Creating an ODBC Connection

To create an ODBC connection, perform the following task:

1. Open Control Panel (start(Setting(Control Panel).
2. Double click on the "Data Sources (ODBC)" icon in the Control Panel.
3. Select the "System DSN" tab on the "ODBC Data Source Administrator" dialog box.
4. Click the "Add" button.
5. On the "Create New Data Source" dialog box, highlight "SQL Server" and click "Finish".

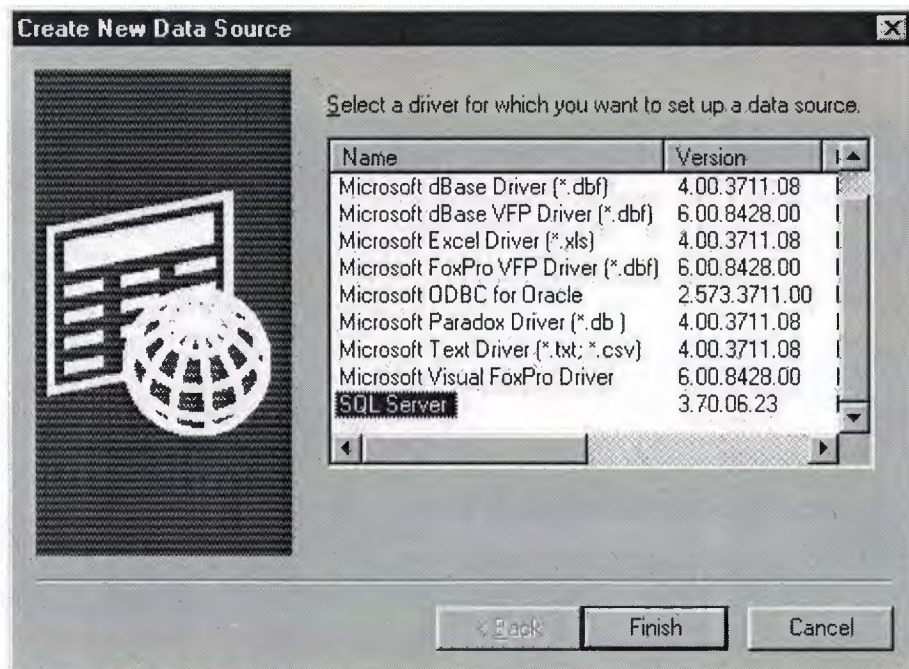


Figure 3.5: Creating New Database

6. On the “Create a New Data Source to SQL Server” dialog box :
 - Enter “15Seconds” in the “Name” field. This is not the name of the database, but the name of the DSN. I kept it the same just for simplicity, however, this is not good practice for security reasons.
 - In the “Description” field enter a brief description for the DSN. I entered “15Seconds Sample DSN”.
 - From the “Server” drop-down box, select the SQL Server to which you would like to connect. Since, my instance of SQL Server to which you would like to connect.

Since, my instance of SQL Server resides on the same machine where I am creating the DSN, I selected “(local)”.

- Click the “Next” button.

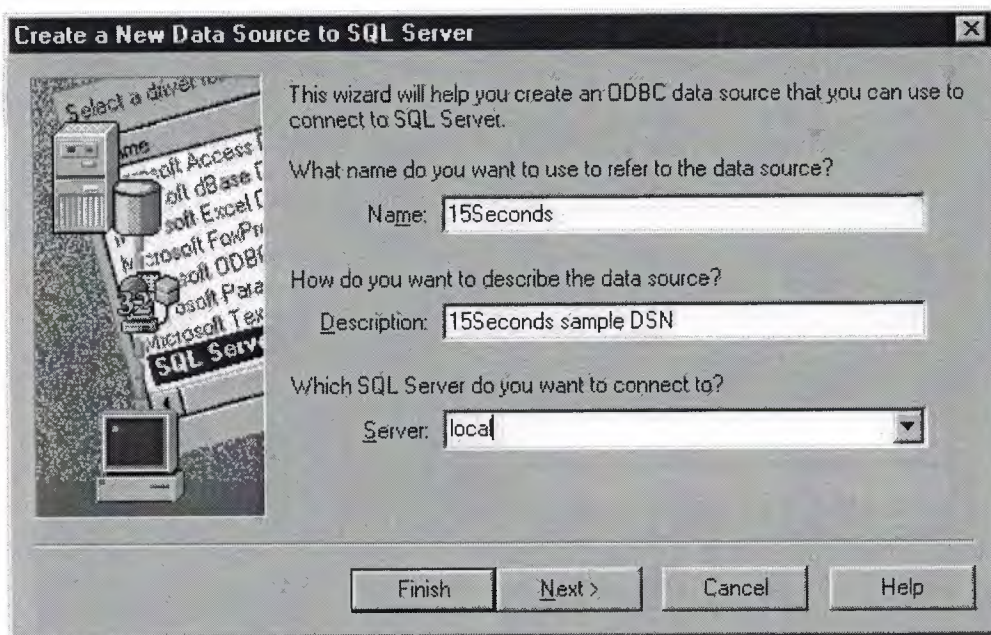


Figure 3.6: Create a New Data Source to SQL Server

7. On the second “Create a New Data Source to SQL Server” dialog box:

- Select the “With SQL Server authentication using a login ID and password entered by the user” radio button to indicate that database security will be implemented by SQL Server rather than Windows NT.
- Select the “Connect to SQL Server to obtain default setting for the additional configuration options” checkbox.
- In the Login ID textbox, enter “bu”.
- Click the “Next” button.

Note: I used “bu” for login ID with a blank password for convenience. This is the default SQL Server administrator account. Again, this is not a good idea for security reasons.

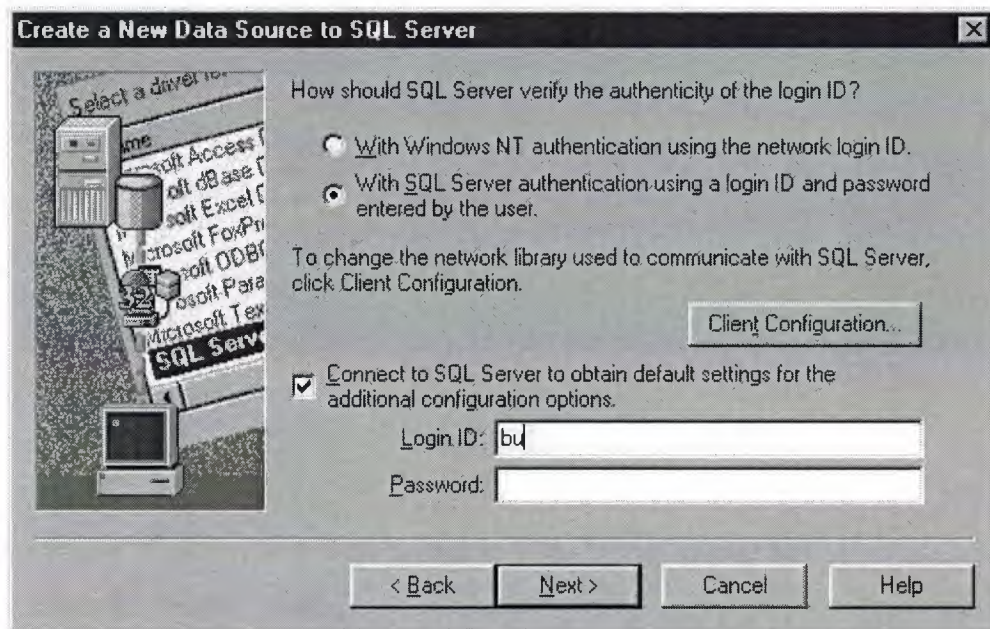


Figure 3.7

8. On the third “Create a New Data Source to SQL Server” dialog box
 - Select the “Change the default database to” checkbox and select “15Secons” from the accompanying select box.
 - Click the “Next” button.

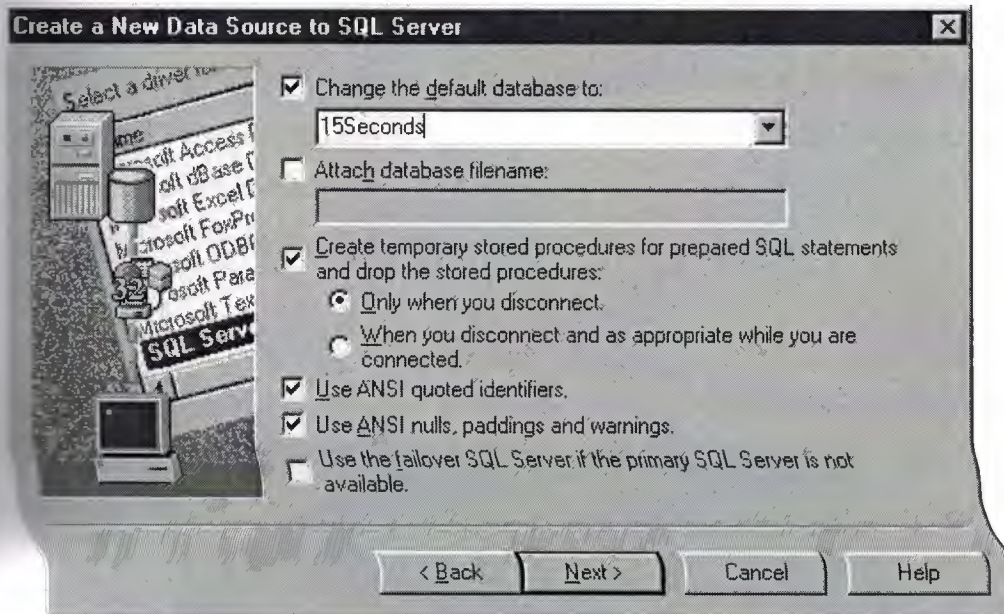


Figure 3.8

9. On the fourth “Create a New Data Source to SQL Server” dialog box
 - Click the “Finish” button.

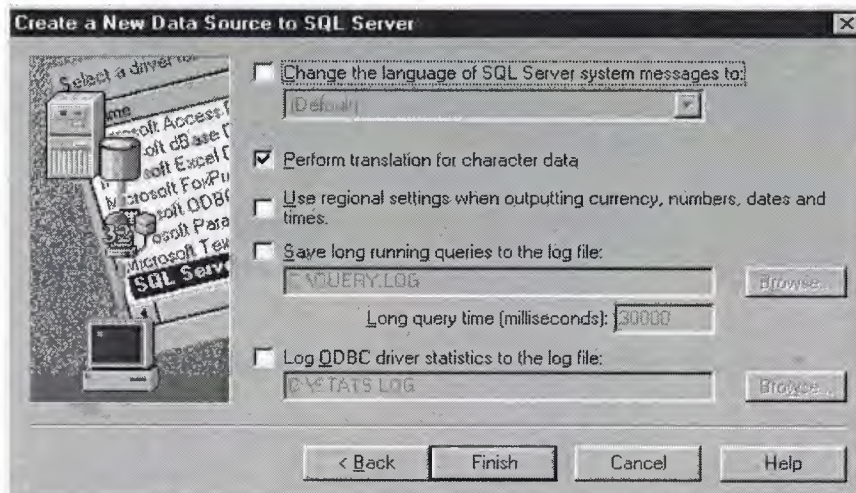


Figure 3.9

10. On the “ODBC Microsoft SQL Server Setup” dialog box, do one of the following:

- Click the “Test Data Source...” button to ensure that the ODBC connection has been created successfully.
- Click the “OK” button to complete the ODBC DSN setup process.



Figure 3.10: ODBC Microsoft SQL Server setup

We have a database, a table, some data and an ODBC DSN. The next step is to create an ASP page to access the data.

CHAPTER FOUR

Electronic Commerce (E-Commerce)

4.1 Introduction

This paper provides a brief introduction to electronic commerce. It discusses the nature of electronic commerce, considers its scope and impact, and outlines several examples. It then identifies a number of open issues and the actors responsible for addressing those issues. Finally, it gives a brief overview of the G-7 Pilot Project "A Global Marketplace for SMEs".

4.2 What is Electronic Commerce?

One possible definition of electronic commerce would be: "any form of business transaction in which the parties interact electronically rather than by physical exchanges or direct physical contact". However, while accurate, such a definition hardly captures the spirit of electronic commerce, which in practice is far better viewed as one of those rare cases where changing needs and new technologies come together to revolutionise the way in which business is conducted.

Modern business is characterised by ever-increasing supply capabilities, ever-increasing global competition, and ever-increasing customer expectations. In response, businesses throughout the world are changing both their organisations and their operations. They are flattening old hierarchical structures and eradicating the barriers between company divisions. They are lowering the barriers between the company and its customers and

suppliers. Business processes are being re-designed so that they cross these old boundaries. We now see many examples of processes that span the entire company and even processes that are jointly owned and operated by the company and its customers or suppliers.

Electronic Commerce is a means of enabling and supporting such change on a global scale. It enables companies to be more efficient and flexible in their internal operations, to work more closely with their suppliers, and to be more responsive to the needs and exceptions of their customers. It allows companies to select the best suppliers regardless of their geographical location and to sell to a global market.

One special case of electronic commerce is electronic trading, in which a supplier provides good or services to a customer in return for payment. A special case of electronic trading is electronic retailing, where the customer is an ordinary consumer rather than another company. However, while these special cases are of considerable economic importance, they are just particular examples of the more general case of any form of business operation or transaction conducted via electronic media. Other equally valid examples include internal transactions within a single company or provision of information to an external organisation without charge.

Electronic Commerce is technology for change. Companies that choose to regard it only as an "add on" to their existing ways of doing business will gain only limited benefit. The major benefit will accrue to fully exploit the opportunities offered by electronic commerce.

4.3 Visitor (Customer) Page

4.3.1 Main Page

When user enters site, first of all, main page is displayed. At the top of the page, in banner place, name of the site, site logo and slogan appears. At the left bottom of banner, category links for products can be found. Just after category links, order state system is placed. In the body part of the page (right-side of categories link) pictures of new products are shown. At the bottom of the page, information links, which helps users to use the web page in most efficient way, is placed.



Figure 4.4: default.asp

4.3.2 Product Presentation

In product categories section, when visitor (user) click on any desired category, all products which belongs to that category, appears as an icon-picture with prices at the bottom of each icon-picture. These icon-pictures are links to details of the related product. When user clicks on these icon-pictures, detailed information and a full picture of the product can be seen.

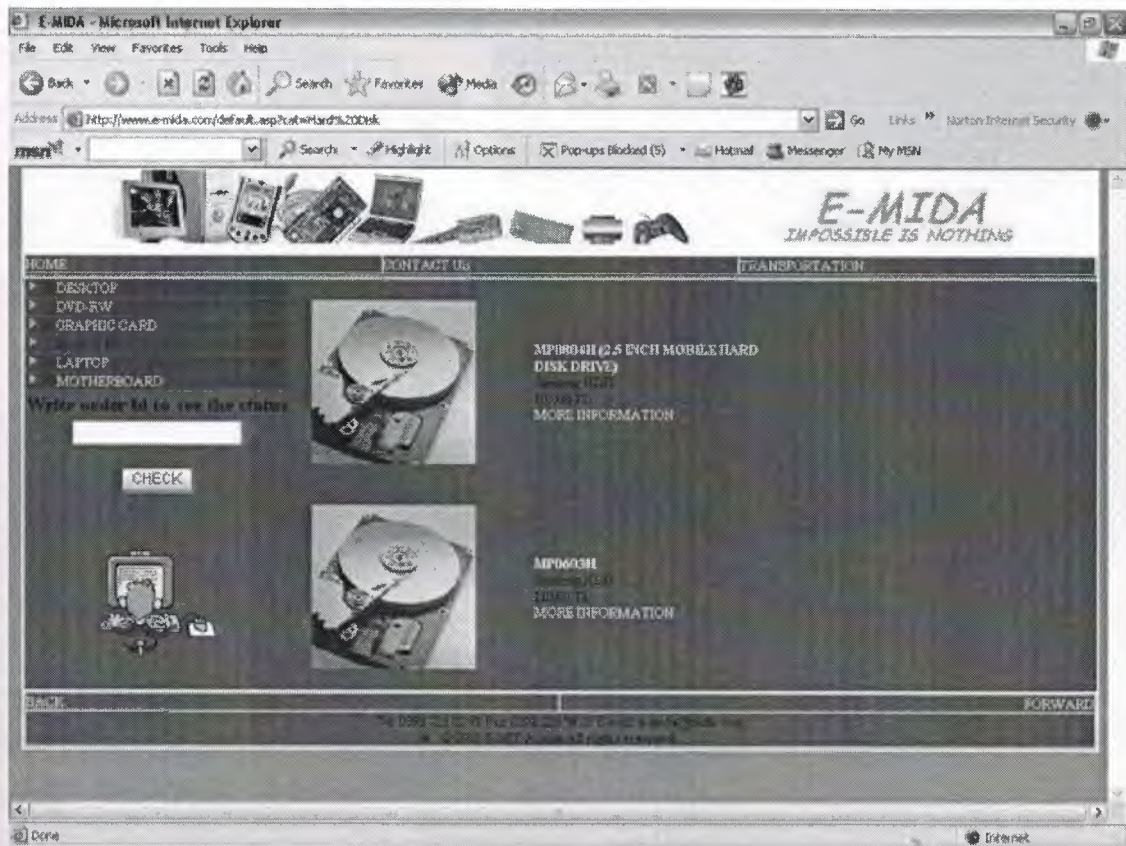


Figure 4.5: product.asp



Figure 4.6: showproduct.asp

4.3.3 Purchase Section

If the user wants to buy a product, just clicks on “Buy Now” button which is placed at the bottom of the detailed information about the product. Mentioned button (Buy Now) leads user to “Purchase Form”. If there are not enough products in stocks, instead of “Buy Now” button, “Not Enough Product in Our Stock” sentence appears.

4.3.3.1 Purchase Action

Just after clicking on “Buy Now” button, a form, which must be filled in order to complete purchasing operation, appears. Via this form, some personal information for purchasing and cargo, which cannot be accessed by any unrelated people, is taken from the user. Just after completing form, in order to finish operation, “Submit” button, at the bottom of the form is clicked. Just after “Submit” operation, an “Order ID” is given to the customer with the information submitted by the user.

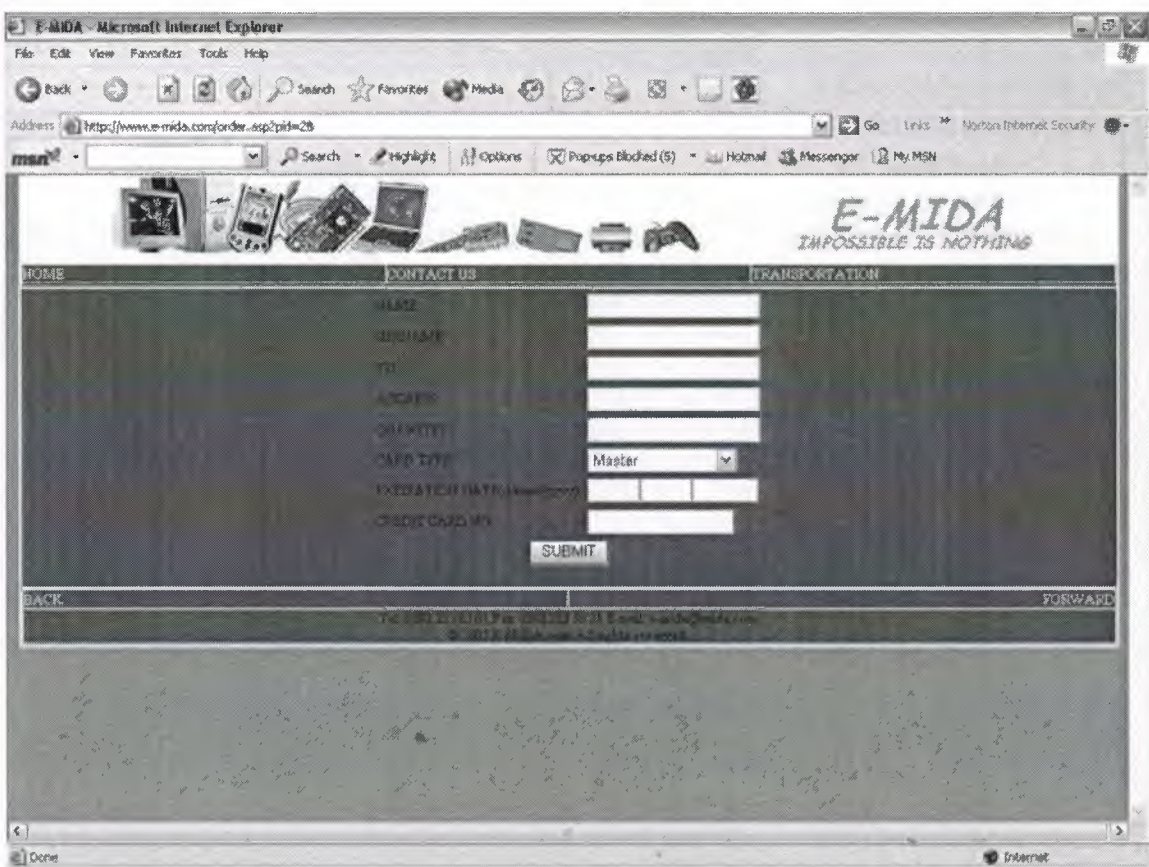


Figure 4.7: order.asp

4.3.4 Order State

As its name implies, this section is used to check current state of the order. In the main page, there is a text-box which is named as “Order State”. User can check the state of an order by entering “Order ID” to that box and clicking on “Check” button, at the bottom of the text-box.

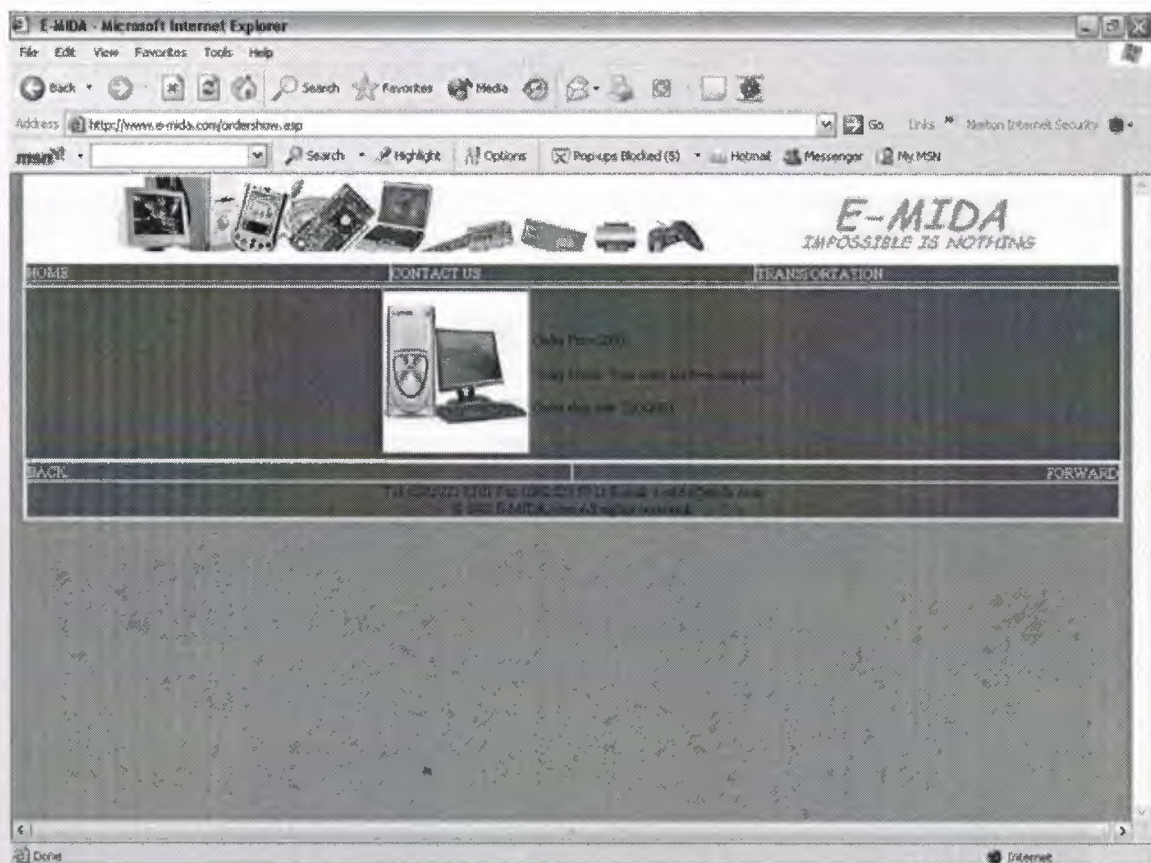


Figure 4.8: ordershow.asp

4.3.5 Home Link

This link can be found at the bottom of each page. By using this link visitor can go back to the main page any time.

4.3.6 Contact Link

“Contact” link can be found at the bottom of the page. Via this link, all questions, problems and comments of the visitors can be handled.

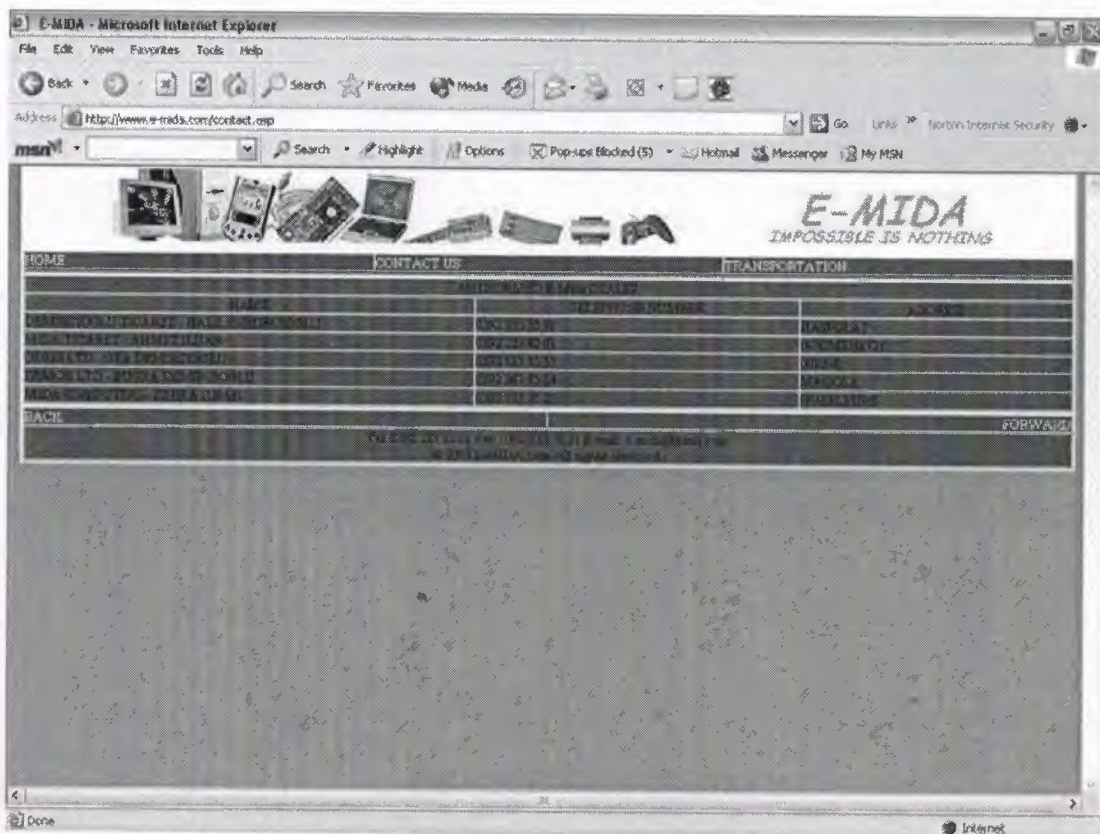


Figure 4.10: contact.asp

4.3.7 Transportation Info Link

All transportation information and process of a purchased product (order) can be followed with this link.

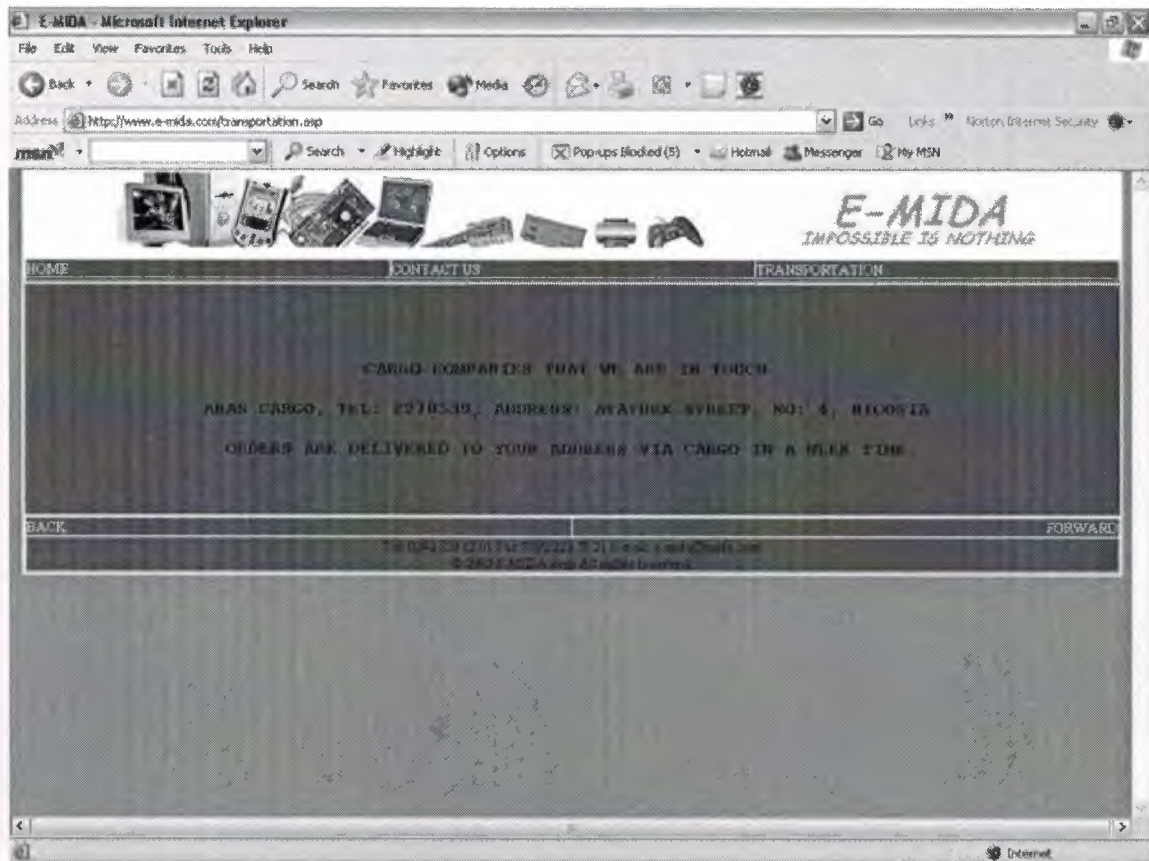


Figure 4.12:transportation.asp

4.3.7 Transportation Info Link

All transportation information and process of a purchased product (order) can be followed with this link.

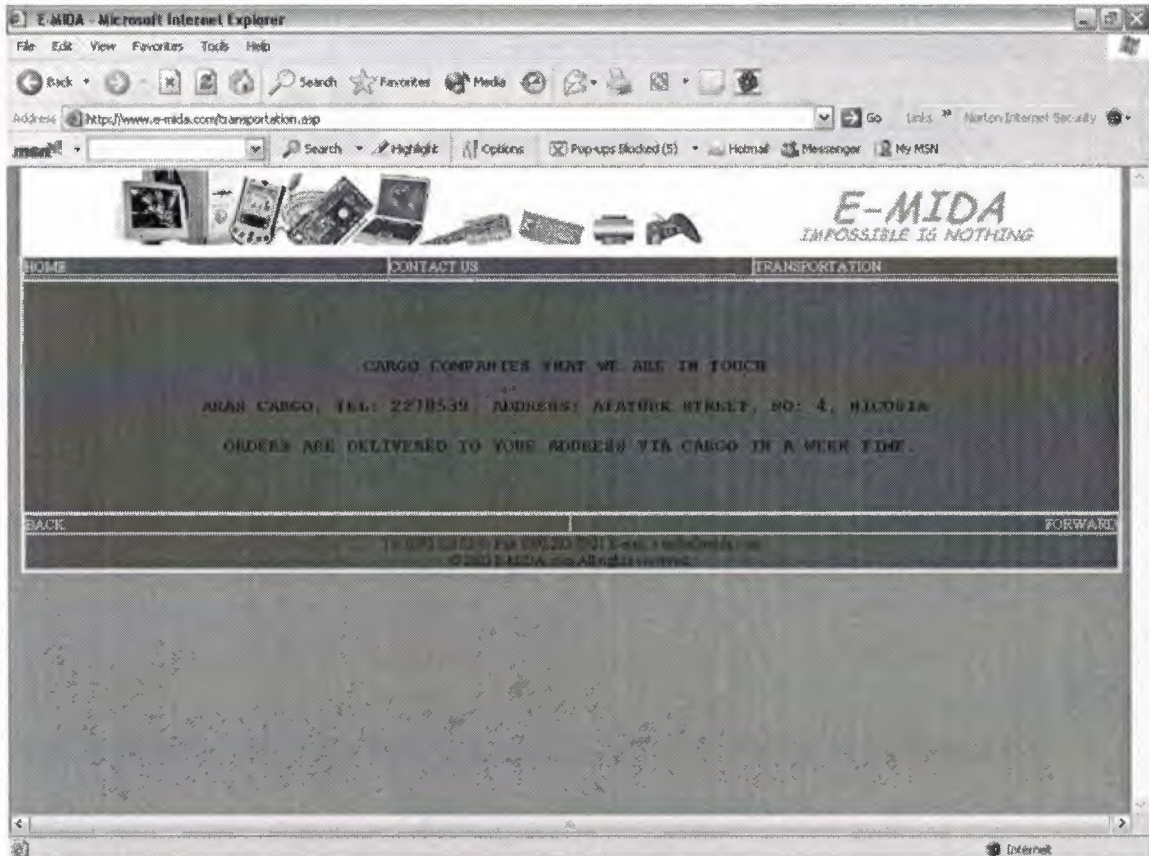


Figure 4.12:transportation.asp

4.4 Administrator Pages



Figure 4.13: administrator.asp

4.4.1 Product List

Administrator of the site can change or delete information about a product via this page. By using “Change” and “Delete” buttons, which is placed in the “Product List” table, administrator can apply changes.

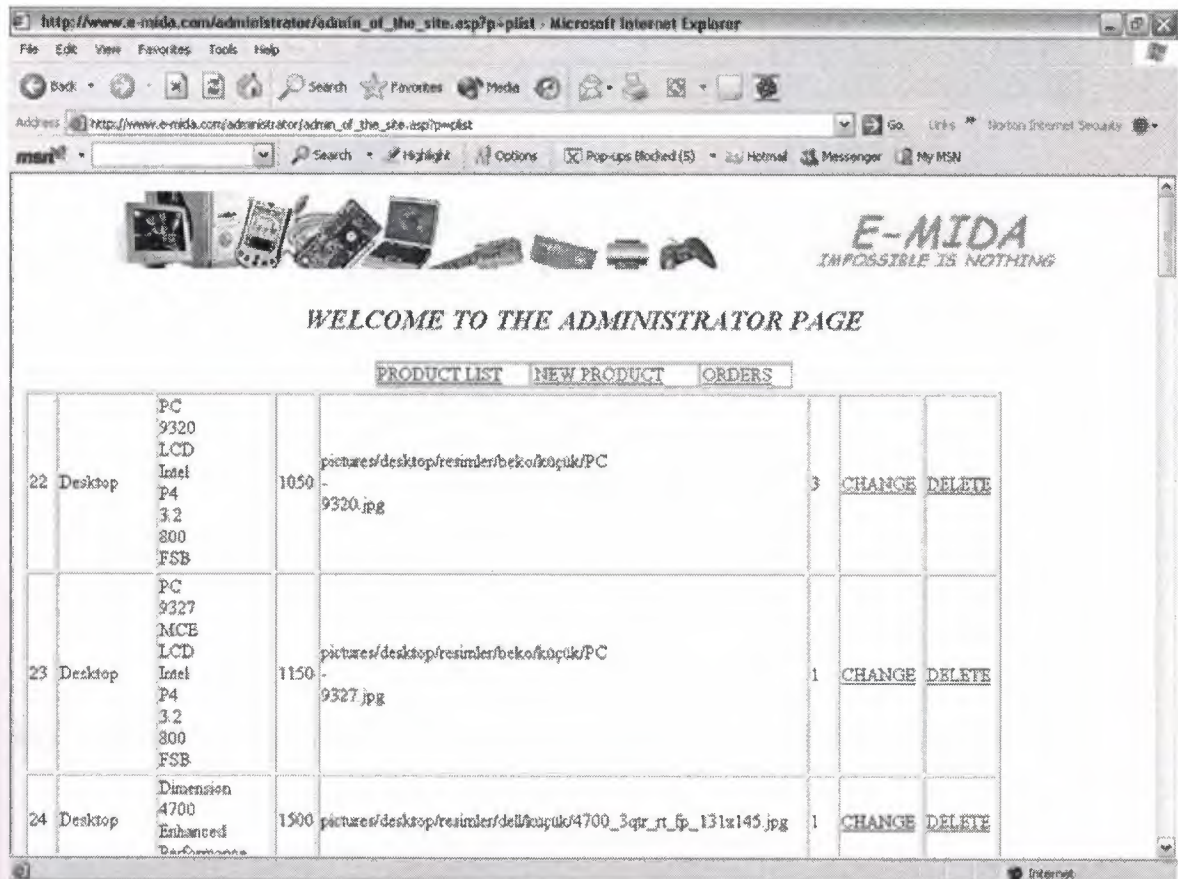


Figure 4.14: administrator.asp

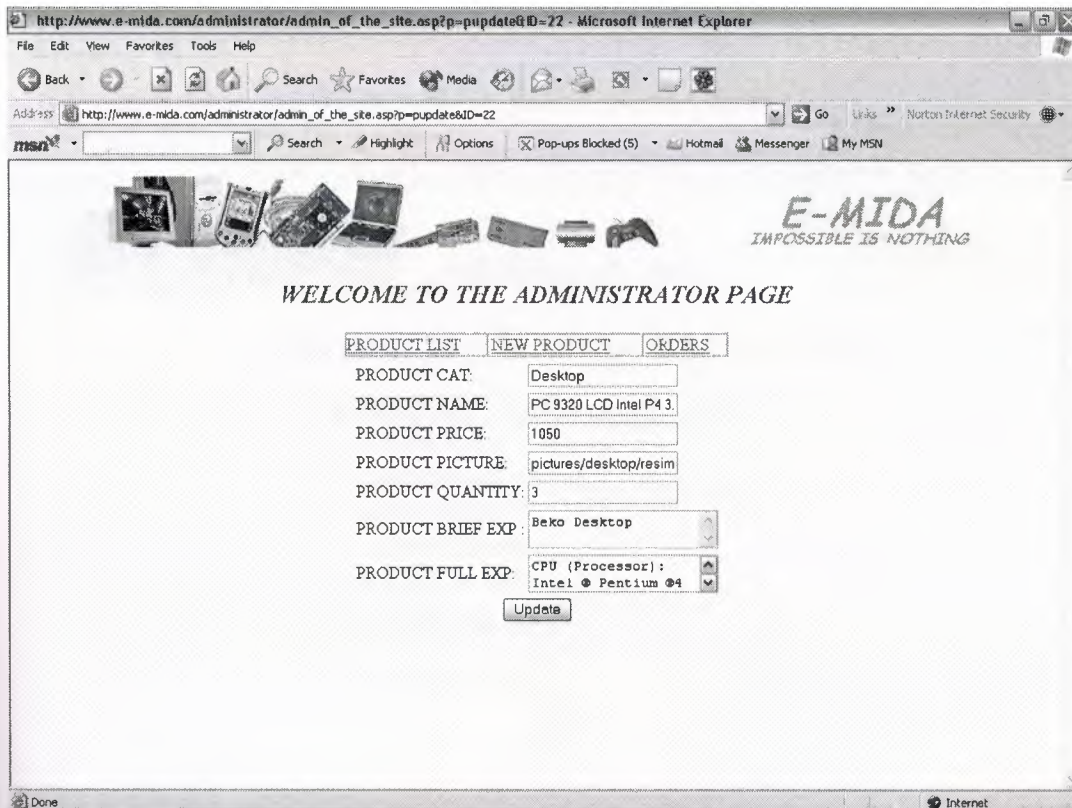


Figure 4.15: administrator.asp

4.4.2 New Product

Addition of a new product to site is carried by this section. By clicking “New Product” link, a form which is used to add a new product to the database of site, appears. Via this form all required information about desired product can be entered. After filling this form by clicking “Add” button, new product can be added to the database.

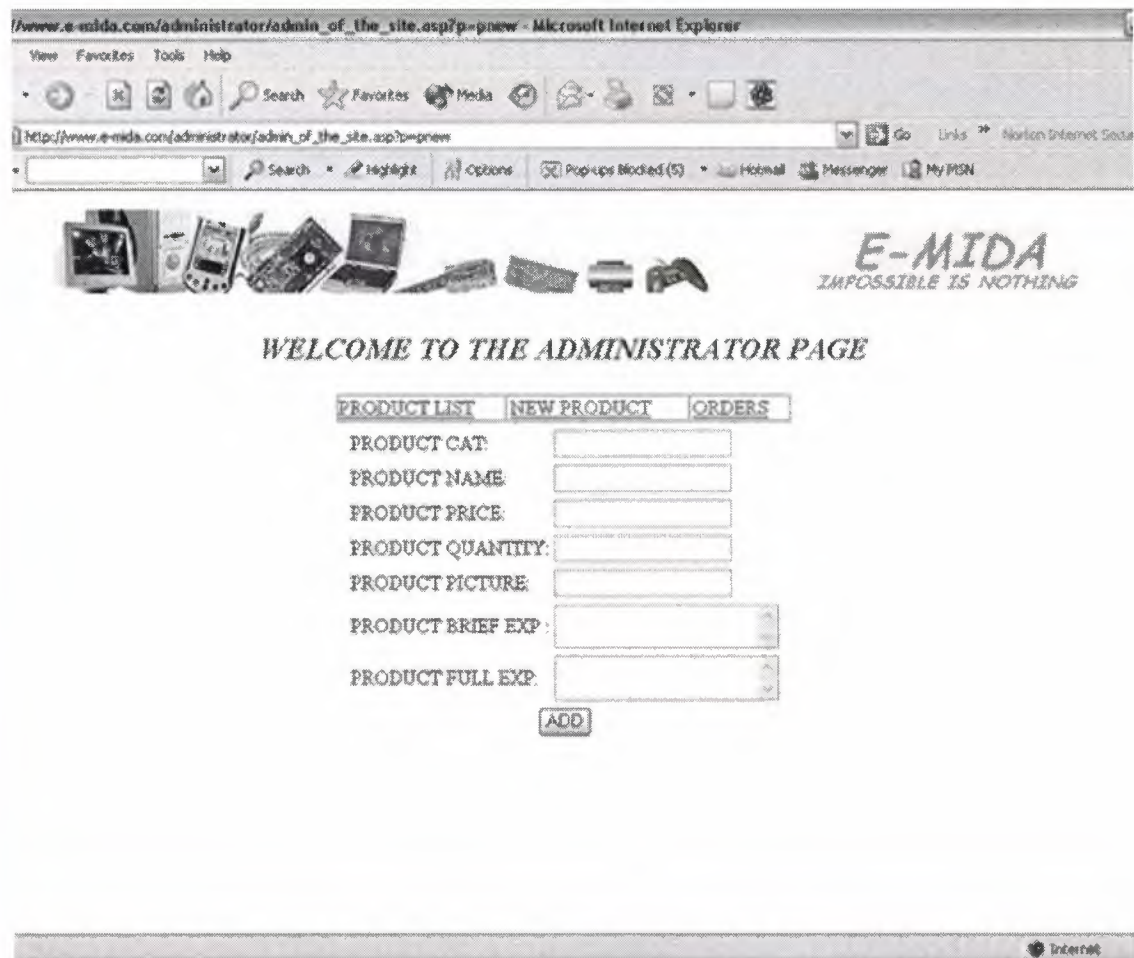


Figure 4.16: administrator.asp

4.4.3 Orders

All information about customers can be accessed via this link. If the price of order is paid by the customer, site administrator clicks on “Ship” button, which is placed at the bottom of the customer and order information, and sends the product to the cargo company for delivery.

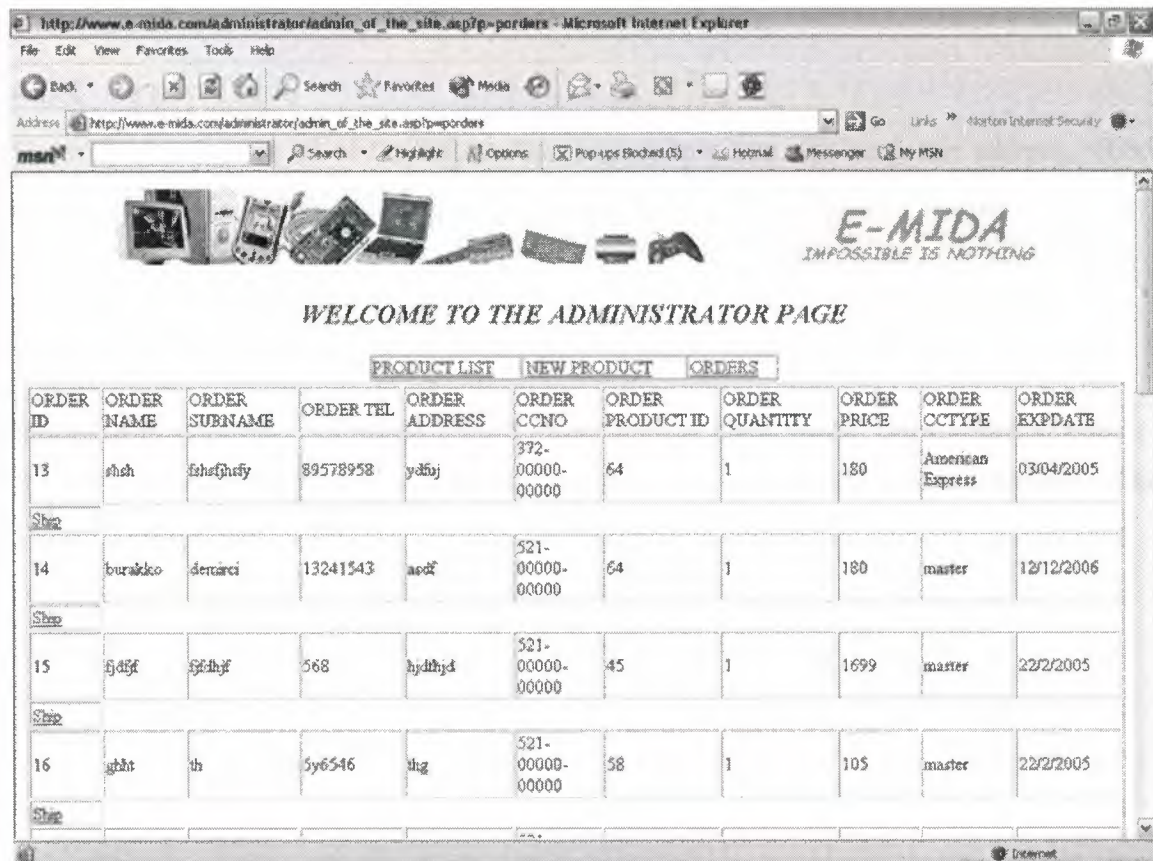


Figure 4.17: administrator.asp

4.5 Database Tables Needed For E-Commerce

4.5.1 Product Table

Product Table contains all of the information (eg: product model, ID, quantity, price, picture, explanation, etc.) about products. (See Figure 1 in Appendix-B)

4.5.2 Order Table

Order Table contains all of the information (eg: order name, ID, surname, address, tel.no, prod_ID, etc.) about orders. (See Figure 2 in Appendix-B)

4.5.3 Show Table

Show Table required for Product List in Administrator Page. Show Table contains some of the information (eg: prod_ID, category, explanation, picture, etc.) about product. (See Figure 3 in Appendix-B)

Conclusion

This introductory paper has emphasized the broad view of electronic commerce as a vehicle for business revolution. It has stressed the importance of taking a global perspective. And it has suggested that the impact of electronic commerce will be pervasive, both on companies and on society as a whole.

Despite a number of open issues that are yet to be resolved, electronic commerce is happening today and happening fast. It is essentially a “bottom up” revolution. Companies world-wide are establish a basic electronic presence on a global open network, learning from the experience, and gradually becoming more sophisticated in their use of the technologies. While the more advanced levels of electronic commerce present substantial challenges, the more basic levels are now well established and supported by “off the shelf” solutions. The best way of gaining the mastery of electronic commerce that will be vital in tomorrow’s markets is to try it today.

The information in this article does not necessarily reflect the opinion of the European Commission. Examples are included for illustrative purpose only and are assumed to be correct. However, the Web sites mentioned or linked to as well as products, unless specifically indicated, are not under the control of the European Commission. For this reason, the Commission can make no representation concerning such sites or products to you: nor does the fact that the Commission has provided reference to these serve as an endorsement by the Commission of any organization or individual or providing content for any of the sites or products listed. The Commission has not tested any of the sites or products listed. The Commission has not tested any software found on the sites and is not in a position to make any representation regarding the quality, safety or suitability of any software found on them or retrievable from them.

References

- 1 E-Commerce Programming
(by Kevin Hoffman)**
- 2 E-Commerce Pro
(by Faisal Khan)**
- 3 E-Commerce Programming with ASP
(by Cristian Darie & Karli Watson)**
- 4 <http://oracle.com>**
- 5 <http://www.ge.com/plastics/index.html>**
- 6 <http://www.asp.net/webmatrix/default.aspx?tabIndex=4&tabId=46>**
- 7 <http://www.iisanswers.com/articles/IIS51.htm>**
- 8 www.asp101.com**
- 9 www.aspfree.com**
- 10 www.ecommercetimes.com**
- 11 www.wilsonweb.com**
- 12 www.asp.net**

APPENDIX – A

“top.asp”

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<!--#include virtual="includes/ADOVBS.INC"-->
<html>
<head>
<title>E-MIDA</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="includes/Stylesheet.css" rel="stylesheet" type="text/css">
</head>
<!--#include virtual="includes/commonfunctions.asp"-->
<body bgcolor="#999999" topmargin="0">
<table border="1" cellspacing="0" cellpadding="0" align="left" width="100%"
bgcolor="#0066CC" bordercolor="#FFFFFF">
<tr>
<td colspan="3"> </td>
</tr>
<tr>
<td colspan="3">
<table border="1" cellpadding="0" cellspacing="0" width="100%">
<tr>
<td width="20%">
<a href="default.asp">Home</a>
</td>
```

```

        <td width="20%">
            <a href="contact.asp">Contact Us</a>
        </td>
        <td width="20%">
            <a href="transportation.asp">Transportation</a>
        </td>

    </tr>
    <tr>
        <td colspan="5" align="center">
            <%if Session("name")<>" " or Session("Surname") <> " " or
Session("username")<>" " then%>
                Welcome <%=Session("name")%>, <%=Session("Surname")%>
            <%end if%>
        </td>
    </tr>
</table>
</td>
</tr>

```

“catlist.asp”

```
<%
```

```

set conn=dbconnexion
cat=trim(request.QueryString("cat"))
Set catRS = Server.CreateObject( "ADODB.Recordset" )
catRS.ActiveConnection = Conn

```

```

sqlString = "SELECT distinct product_category from products order by
product_category"
catRS.Open sqlString
while not catrs.eof
%>
<table border="0" cellspacing="1" cellpadding="0">
<% IF catrs("product_category") = cat THEN%>
    <tr>
        <td width="25" bgcolor="#C60000" valign="middle">
            
        </td>
        <td width="210px" align="left" bgcolor="#C60000">
            <font color="black"><%=catrs("product_category")%></font>
        </td>
    </tr>
    <% ELSE %>
    <tr>
        <td width="25" valign="middle" bgcolor="#C60000">
            
        </td>
        <td width="210px" align="left" bgcolor="#C60000">
            <a
href="default.asp?cat=<%=server.HTMLEncode(catrs("product_category"))%>"><
%=catrs("product_category")%></a>
        </td>
    </tr>
<% END IF %>
<%

```

```
catrs.movenext
```

```
wend
```

```
%>
```

```
<%
```

```
catrs.close
```

```
set catrs = nothing
```

```
conn.close
```

```
set conn = nothing
```

```
%>
```

```
</TABLE>
```

```
<TABLE>
```

```
<tr>
```

```
  <td>
```

```
    <font size="+1" color="#000000"><b>Write order id to see the  
status</b></font>
```

```
  </td>
```

```
</tr>
```

```
<FORM action="ordershow.asp" method="post">
```

```
  <tr>
```

```
    <td align="center">
```

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

```
    </td>
```

```
  </tr>
```

```
  <tr>
```

```
    <td align="center"><br>
```

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

```

        </FORM>
    </td>
</tr>
</TABLE>

```

“productlist.asp”

```

<%
' Get the Current Page
pg = TRIM( Request( "pg" ) )
IF pg = "" THEN pg = 1
' Open the Recordset
set conn = dbconnexion()
sqlcategorivemarka="Select distinct "
Set prodRS = Server.CreateObject( "ADODB.Recordset" )
prodRS.ActiveConnection = Conn
prodRS.CursorType = adOpenStatic
prodRS.PageSize = 5
if cat="" then
    sqlString = "SELECT * FROM Products ORDER BY product_ID desc "
elseif cat <> "" then
    sqlstring = "Select * from products where product_category= '" & cat & "' order
by product_ID desc"
end if

prodRS.Open sqlString
prodRS.AbsolutePage = pg
%>

```

```

<table width="455" border=0 cellpadding=15 cellspacing=0>
<%
WHILE NOT prodRS.EOF AND rowCount < prodRS.PageSize
rowCount = rowCount + 1
%>
<tr>
<td width="178">
<% IF prodRS( "product_picture" ) <> "?????" THEN %>
    <a href=" ../product.asp?pid=<%=prodRS("product_id")%>">" width="150px" height="150px" hspace=4 vspace=4 border=0
align="center"></a>
    <% elseif prodRS("product_picture")="?????" THEN%>
    <a href=" ../product.asp?pid=<%=prodrs("Product_ID")%>"><IMG
SRC="/img/nopicture.jpg" HSPACE=4 VSPACE=4 BORDER=0 align="center"></a>
    <% END IF %>
</td>
<td width="217">
<a href=" ../product.asp?pid=<%=prodRS( "Product_id" )%>">
<b><%=prodRS( "product_name" )%></b></a>
<br><%=prodRS( "product_briefDesc" )%>
<br><%=formatcurrency(prodrs("Product_Price"))%>
<br><a href=" ../product.asp?pid=<%=prodRS( "Product_id" )%>">
More Information</a>
</td>
</tr>
<%
prodRS.MoveNext
wend

```

```

%>
<%
IF prodRS.PageCount > 1 THEN
%>
<font color="#D7D7D7">
<tr>
    <td align="center" colspan="2">
<%
    FOR i = 1 to prodRS.PageCount
    IF i <> cINT( pg ) THEN
%>
<a href=" ../default.asp?cat=<%=cat%>&pg=<%=i%>">
<%=i%></a>&nbsp;
<% ELSE %>
<b><%=i%></b>&nbsp;
<% END IF %>
<%
    NEXT
%>
</font>
</center>
<%
END IF
%>
</td>
</tr>
</table>

```

“commonfunctions.asp”

<%

FUNCTION CleanCCNum(ccnumber)

FOR i = 1 TO LEN(ccnumber)

IF isNumeric(MID(ccnumber, i, 1)) THEN

CleanCCNum = CleanCCNum & MID(ccnumber, i, 1)

END IF

NEXT

END FUNCTION

FUNCTION validCCNumber(ccnumber)

ccnumber = CleanCCNum(ccnumber)

IF ccnumber = "" THEN

validCCNumber = FALSE

ELSE

isEven = False

digits = ""

for i = Len(ccnumber) To 1 Step -1

if isEven Then

digits = digits & CINT(MID(ccnumber, i, 1)) * 2

Else

digits = digits & CINT(MID(ccnumber, i, 1))

End If

isEven = (Not isEven)

Next

checkSum = 0

For i = 1 To Len(digits) Step 1

```

    checkSum = checkSum + CINT( MID( digits, i, 1 ) )
Next
validCCNumber = ( ( checkSum Mod 10) = 0 )
END IF
End Function

```

```

FUNCTION fixQuotes( theString )
    fixQuotes = REPLACE( theString, "'", "" )
END FUNCTION

```

```

Function DBConnexion()
    Dim conn
    Set conn = Server.CreateObject("ADODB.Connection")
    sconnstring="DRIVER={Microsoft Access Driver (*.mdb)}; " & _
    "DBQ=" & Server.MapPath("db/storeDB.mdb")
    On Error Resume Next
    conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; " & _
    "DBQ=" & Server.MapPath("db/storeDB.mdb")
    If Err.number <> 0 Then
        Response.Write "<br>" & GetTranslation("LANG_SQL_ERROR")
        If IsAuthorized(ROLE_ADMIN) Then
            Response.Write " : " & Err.Description & "<br>"
            Response.Write "Connexion String : " & sConnString & "<br>"
        End If
        Err.Clear
    End If
    On Error Goto 0
    set DBConnexion = conn

```

End Function

FUNCTION checke_mail(thestring)

if not instr(1,thestring,"@")=0 and not instr(1,thestring,".")=0 then

 checke_mail= 0

else

 checke_mail= 1

end if

end function

Function check_empty(thestring)

if trim(thestring)="" then

 check_empty=1

else

 check_empty=0

end if

End Function

Function check_equal(thestring,thestring2)

if thestring=thestring2 then

 check_equal=0

else

 check_equal=1

end if

End Function

Function check_user_exists(thestring)

set conn=DBCONNEXION()

```

sql="select username from usertbl where username= '" &thestring& "'"
set rs=conn.execute(sql)
if rs.eof then
    check_user_exists=0
else
    check_user_exists=1
end if
rs.close
set rs=nothing
conn.close
set conn=nothing
End Function

```

```

Function check_email_exists(thestring)
set conn=Server.CreateObject("ADODB.Connection")
conn.provider="MICROSOFT.JET.OLEDB.4.0"
conn.connectionstring=Server.MapPath("db/info.mdb")
conn.open
sql="select e_mail from usertbl where e_mail= '" &thestring& "'"
set rs=conn.execute(sql)
if rs.eof then
    check_email_exists=0
else
    check_email_exists=1
end if
rs.close
set rs=nothing
conn.close

```

```
set conn=nothing
```

```
End Function
```

```
Function checkunandpsw (username,password)
```

```
set conn=server.CreateObject("ADODB.Connection")
```

```
conn.provider="Microsoft.JET.OLEDB.4.0"
```

```
conn.connectionstring=server.MapPath("db/info.mdb")
```

```
conn.open
```

```
sql="Select UserId,username,passwords from usertbl where username= '" & username & "'
```

```
and passwords= '" & password & "'"
```

```
set rs=conn.execute(sql)
```

```
if not rs.eof then
```

```
    response.Cookies("UserId").expires="1/1/2006"
```

```
    response.Cookies("UserId")=rs("UserID")
```

```
    checkunandpsw=1
```

```
else
```

```
    checkunandpdw=0
```

```
end if
```

```
rs.close
```

```
set rs=nothing
```

```
conn.close
```

```
set conn=nothing
```

```
end function
```

```
Function chgclr(x)
```

```
if x=0 then
```

```
    response.Write("#0066CC")
```

```
else
```

```
        response.Write("#6C91E6")
    end if
End Function
```

```
public sub chgcolor(i)
    if i=0 then
        i=1
        response.Write("#33ccff")
    else
        i=0
        response.Write("#006699")
    end if
end sub
%>
```

“stylesheet.css”

```
A
{
    FONT-SIZE: 13px;
    TEXT-TRANSFORM: uppercase;
    COLOR: white;
    TEXT-DECORATION: none
}
A:hover
{
    COLOR: powderblue;
    TEXT-DECORATION: underline
```

```

}
TD
{
    FONT-WEIGHT: lighter;
    FONT-SIZE: 12px
}

```

“bottom.asp”

```

<tr>
    <td colspan="3">
        <table border="1" cellspacing="0" cellpadding="0" width="100%">
            <tr>
                <td width="50%">
                    <a
                        HREF="javascript:history.go(-1)"
                        onMouseOver="window.status='Click Here to Go Back One Page';return true">Back</a>
                </td>

                <td width="50%">
                    <div align="right"><a
                        HREF="javascript:history.go(-1)"
                        onMouseOver="window.status='Click Here to Go Back One Page';return
                        true">Forward</a>
                    </div></td>

            </tr>
            <tr>
                <td colspan="5" align="center">

```

Tel: 0392 223 82 01 Fax: 0392 223 50 21 E-mail: e-
mida@mida.com

© 2005 E-MIDA.com All rights reserved.

</td>

</tr>

</table>

</td>

</tr>

</table>

</body>

</html>

“Default.asp”

```
<!--#include virtual="includes/top.asp"-->
```

```
<tr>
```

```
<td colspan="3">
```

```
<table border="0" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td width="25%" valign="top" align="center">
```

```
<%set conn=dbconnexion%>
```

```
<!--#include virtual="includes/catlist.asp"-->
```

```
<br>
```

```
<br>
```

```

```

```
</td>
```

```
<td>
```

```
<!--#include virtual="includes/productlist.asp"-->
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</td>
```

```
</tr>
```

```
<!--#include virtual="includes/bottom.asp"-->
```

“Product.asp”

```
<!--#include virtual="includes/top.asp"-->
<tr>
  <td colspan="3">
    <table border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td width="25%" valign="top" align="center">
          <%set conn=dbconnexion%>
          <!--#include virtual="includes/catlist.asp"-->
          <br>
          <br>
          
        </td>
        <td>
          <%
            set conn1 = dbconnexion
            pid = request.QueryString("pid")
            sql = "SELECT * from Products WHERE product_id = " & pid
            set rs = conn1.execute(sql)
            %>
          <table cellspacing="0" cellpadding="0" width="100%" border="0">
            <tr>
              <td width="30%" align="center" valign="top">"></td>
              <td width="70%"><%=rs("product_briefdesc")%><br><br><a
href="order.asp?pid=<%=rs("product_id")%>"></a><br>
            </tr>
            <tr>
              <td colspan="2"><%=rs("product_fulldesc")%>
            </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </td>
</tr>
</tr>
```

```

<!--#include virtual="includes/bottom.asp"-->
<%
conn1.close
set conn1 = nothing
%>

```

“Order.asp”

```

<!--#include virtual="includes/top.asp"-->
<%
pid=request.QueryString("pid")
%>
<tr align="center">
    <td>
        <form method="post" action="orderreg.asp?prod_ID=<%=pid%>">
        <table>
            <tr>
                <td>
                    NAME:
                </td>
                <td>
                    <input type="text" name="Order_name" maxlength="50">
                </td>
            </tr>
            <tr>
                <td>
                    SURNAME:
                </td>
                <td>
                    <input type="text" name="Order_surname" maxlength="50">
                </td>
            </tr>
            <tr>
                <td>
                    TEL:
                </td>
                <td>

```

```

        <input type="text" name="Order_tel" maxlength="12">
    </td>
</tr>
<tr>
    <td>
        ADDRESS:
    </td>
    <TD>
        <input type="text" name="Order_address" maxlength="500">
    </TD>
</tr>
<tr>
    <td>
        QUANTITY:
    </td>
    <td>
        <input type="text" name="Order_quantity" maxlength="3">
    </td>
</tr>
<tr>
    <td>
        CARD TYPE:
    </td>
    <td>
        <select name="cc_type">
            <option value="master">Master</option>
            <option value="Visa">Visa</option>
            <option value="American Express">American Express</option>
            <option value="Discover">Discover</option>
        </select>
    </td>
</tr>
<tr>
    <td>
        EXPIRATION DATE(dd/mm/yyyy):
    </td>
    <td>
        <input type="text" name="ccexp_day" size="2" maxlength="2"><input
type="text" name="ccexp_month" size="2" maxlength="2"><input type="text"
name="ccexp_year" size="4" maxlength="4">
    </td>
</tr>

```

```

        <tr>
            <td>
                CREDIT CARD NO:
            </td>
            <TD>
                <input type="text" name="Order_ccno" size="16" maxlength="16">
            </TD>
        </tr>
    </table>
    <input type="Submit" align="center" value="SUBMIT">
</form>
        </td>
    </tr>

<!--#include virtual="includes/bottom.asp"-->

```

“Orderreg.asp”

```

<!--#include virtual="includes/top.asp"-->
<%
pid = request.QueryString("prod_ID")
order_name = request.Form("order_name")
order_surname = request.Form("order_surname")
order_tel = request.Form("order_tel")
order_address = request.Form("order_address")
order_ccno = request.Form("order_ccno")
control=true
if request.form("order_name")="" then
    control=false
end if
if request.form("order_surname")="" then

```

```

        control=false
    end if
    if request.form("order_tel")="" then
        control=false
    end if
    if request.form("order_address")="" then
        control=false
    end if
    if request.Form("order_quantity") = "" then
        order_quantity = 0
    else
        order_quantity = cint(request.Form("order_quantity"))
    end if
    cc_type = request.Form("cc_type")
    ccexp_day = request.Form("ccexp_day")
    ccexp_month = request.Form("ccexp_month")
    ccexp_year = request.Form("ccexp_year")
    order_expdate = ccexp_day + "/" + ccexp_month + "/" + ccexp_year
    card_valid = IsCreditCard(cc_type,order_ccno)
    prod_ID = request.QueryString("prod_ID")
    set conn = server.CreateObject("ADODB.Connection")
    conn.provider = "Microsoft.JET.OLEDB.4.0"
    conn.connectionstring = Server.MapPath("DB/storedb.mdb")
    conn.open
    sql="SELECT * FROM products WHERE product_ID=" & pid
    set rs=conn.execute(sql)
    if control then
        if order_quantity<=rs("product_quantity") then

```

```

if IsCreditCard(cc_type,order_ccno)then
sql="SELECT * FROM products WHERE product_ID =" & pid
set rs=conn.execute(sql)
order_price = cdbl(rs("product_price")) * cint(order_quantity)
order_price = cdbl(order_price)
sql
=
"insert
into
orders(order_name,order_surname,order_tel,order_address,order_quantity,order_ccno,order
_prod_ID,order_price,"&_
"order_cctype,order_expdate) Values (" & order_name &
","&order_surname&","&order_tel&","&order_address&","&order_quantity&""&_
","&order_ccno&","&prod_ID&","&order_price&","&cc_type&","&order_expdate&")"
conn.execute(sql)
sql="update products set product_quantity=product_quantity - " & order_quantity & "
where product_ID=" & pid
conn.execute(sql)
function IsCreditCard(ByRef asCardType, ByRef anCardNumber)
' Performs a Mod 10 check To make sure the credit card number
' appears valid
' Developers may use the following numbers as dummy data:
' Visa: 430-00000-00000
' American Express: 372-00000-00000
' Mastercard: 521-00000-00000
' Discover: 620-00000-00000

Dim lsNumber ' Credit card number stripped of all spaces, dashes, etc.
Dim lsChar ' an individual character
Dim lnTotal ' Sum of all calculations
Dim lnDigit ' A digit found within a credit card number

```

```

Dim lnPosition      ' identifies a character position In a String
Dim lnSum           ' Sum of calculations For a specific Set

' Default result is False
IsCreditCard = False

' =====
' Strip all characters that are Not numbers.
' =====

' Loop through Each character inthe card number submitted
For lnPosition = 1 To Len(anCardNumber)
    ' Grab the current character
    lsChar = Mid(anCardNumber, lnPosition, 1)
    ' if the character is a number, append it To our new number
    if IsNumeric(lsChar) Then lsNumber = lsNumber & lsChar

Next ' lnPosition

' =====
' The credit card number must be between 13 and 16 digits.
' =====
' if the length of the number is less Then 13 digits, then Exit the routine
if Len(lsNumber) < 13 Then Exit function

' if the length of the number is more Then 16 digits, then Exit the routine
if Len(lsNumber) > 16 Then Exit function

```

```

' =====
' The credit card number must start with:
' 4 For Visa Cards
' 37 For American Express Cards
' 5 For MasterCard
' 6 For Discover Cards
' =====

' Choose action based on Type of card
Select Case LCase(asCardType)
' VISA
Case "visa", "v"
' if first digit Not 4, Exit function
if Not Left(lsNumber, 1) = "4" Then Exit function
' American Express
Case "american express", "americanexpress", "american", "ax", "a"
' if first 2 digits Not 37, Exit function
if Not Left(lsNumber, 2) = "37" Then Exit function
' Mastercard
Case "mastercard", "master card", "master", "m"
' if first digit Not 5, Exit function
if Not Left(lsNumber, 1) = "5" Then Exit function
' Discover
Case "discover", "discovercard", "discover card", "d"
' if first digit Not 6, Exit function
if Not Left(lsNumber, 1) = "6" Then Exit function

```

```

Case Else
End Select ' LCase(asCardType)

' =====
' if the credit card number is less Than 16 digits add zeros
' To the beginning to make it 16 digits.
' =====
' Continue Loop While the length of the number is less Than 16 digits
While Not Len(lsNumber) = 16

    ' Insert 0 To the beginning of the number
    lsNumber = "0" & lsNumber

Wend ' Not Len(lsNumber) = 16

' =====
' Multiply Each digit of the credit card number by the corresponding digit of
' the mask, and sum the results together.
' =====

' Loop through Each digit
For lnPosition = 1 To 16

    ' Parse a digit from a specified position In the number
    lnDigit = Mid(lsNumber, lnPosition, 1)

    ' Determine if we multiply by:
    '     1 (Even)

```

```

' 2 (Odd)
' based On the position that we are reading the digit from
lnMultiplier = 1 + (lnPosition Mod 2)

' Calculate the sum by multiplying the digit and the Multiplier
lnSum = lnDigit * lnMultiplier

' (Single digits roll over To remain single. We manually have to Do this.)
' if the Sum is 10 or more, subtract 9
if lnSum > 9 Then lnSum = lnSum - 9

' Add the sum To the total of all sums
lnTotal = lnTotal + lnSum

Next ' lnPosition

' =====
' Once all the results are summed divide
' by 10, if there is no remainder Then the credit card number is valid.
' =====
IsCreditCard = ((lnTotal Mod 10) = 0)

```

```

End function ' IsCreditCard

sql="select max(order_ID) as order_ID from Orders"
set rs=conn.execute(sql)
order_ID=rs("order_ID")

```

%>

<tr>

<td>

<table width="900" cellspacing="0" cellpadding="0" align="center" border="1">

<tr align="center"><td>

<table border="1" cellspacing="5" cellpadding="5" width="500"

bgcolor="#FFFFFF">

<tr>

<td>

NAME:

</td>

<td>

<%response.write(order_name)%> </td>

</tr>

<tr>

<td>

SURNAME:

</td>

<td>

<%response.write(order_surname)%> </td>

</tr>

<tr>

<td>

TEL:

</td>

<td>

<%response.write(order_tel)%> </td>

</tr>

<tr>

```

        <td>
            ADDRESS:
        </td>
        <TD>
            <%response.write(order_address)%> &nbsp;</TD>
    </tr>
    <TR>
        <td>
            QUANTITY:
        </td>
        <td>
            <%response.write(order_quantity)%> &nbsp;</td>
    </tr>
    <tr>
        <td>ORDER_ID:
        </td>
        <td>
            <%response.write(order_ID)%> &nbsp;</td>
    </tr>

```

```

</table>

```

```

</form>

```

```

</td>

```

```

</tr>

```

```

<%

```

```

else

```

```

'ccno hatalidir tekrar giriniz

```

```

%>
<tr>
<td colspan="3">
<strong><font color="#FF0000">INVALID CREDIT CARD NUMBER
ENTRY</font></strong>
</td>
</tr>
<%
end if
else
'üründen yeterli sayıda yoksa hata msg verecek
%>
<tr>
<td colspan="3">
<table align="center" width="900" border="1" cellpadding="0" cellspacing="0">
<tr>
<td align="center" height="400px" valign="middle">
<strong><font color="#FF0000"> NOT ENOUGH PRODUCT
AVAILABLE</font></strong>
</td>
</tr>
</table>
</td>
</tr>
<%
end if
else
%>
<tr>

```

```

        <td colspan="3">
            <table align="center" width="900" border="1" cellpadding="0"
cellspacing="0">
                <tr>
                    <td align="center" height="400px" valign="middle">
                        <strong><font color="#FF0000">ONE OF THE REQUIRED
FIELDS IS MISSING GO BACK AND FILL ALL THE FIELDS</font></strong>
                    </td>
                </tr>
            </table>
        </td>
    </tr>
<%
end if
conn.close
set conn=nothing
%><!--#include virtual="includes/bottom.asp"-->

```

“Ordershow.asp”

```

<!--#include virtual="includes/top.asp"-->
<%
order_ID=request.form("Order_ID")
set conn=Server.CreateObject("ADODB.connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("DB/storedb.mdb")

```

```
conn.open
```

```
sql="select * from products,orders where product_ID=order_prod_ID and order_ID = " &  
order_ID
```

```
set rs1=conn.execute(sql)
```

```
if rs1.eof then
```

```
%>
```

```
<tr>
```

```
<td align="center">
```

```
<table align="center" width="900" border="1" cellpadding="0" cellspacing="0">
```

```
<tr>
```

```
<td align="center" height="400px" valign="middle"> <strong><font  
color="#FF0000">INVALID
```

```
ORDER ID</font></strong> </td>
```

```
</tr>
```

```
</table>
```

```
</td>
```

```
</tr>
```

```
<%else
```

```
%>
```

```
<tr>
```

```
<td align="center">
```

```
<table>
```

```
<tr>
```

```
<td valign="top">">
```

```
</td>
```

```
<td>Order Price:<%=rs1("order_price")%> <br><br>
```

```

        Order Status: <%if rs1("order_shipped") then response.Write("Your order has
been shipped<br><br>Order ship date:" & rs1("order_ship_date")) else
Response.Write("Waiting") end if%>
        </td>
    </tr>
</table>
</td>
</tr>
<%end if%>
<!--#include virtual="includes/bottom.asp"-->

```

“contact.asp”

```

<!--#include virtual="includes/top.asp"-->
<tr>
    <td><table width="100%" border="1" cellspacing="0" cellpadding="0">
    <tr>
        <td colspan="3" align="center"> AUTHORIZED E-Mida DEALER </td>
    </tr>
    <tr>
        <td width="43%"><div align="center">NAME</div></td>
        <td width="31%"><div align="center">TELEPHONE NUMBER</div></td>
        <td width="26%"><div align="center">ADDRESS</div></td>
    </tr>
    <tr>
        <td>DEMIRCIOGLU TICARET - HALIL DEMIRCIOGLU</td>
        <td>0392 233 55 91</td>
        <td>HASPOLAT</td>
    </tr>
    <tr>
        <td>MIDA TICARET - BUGRA DEMIRCIOGLU</td>

```

```

<td>0392 223 82 01</td>
<td>GOCMENKOY</td>
</tr>
<tr>
<td>ODEM LTD - OYA DEMIRCIOLU</td>
<td>0392 813 55 55</td>
<td>GIRNE</td>
</tr>
<tr>
<td>DEMOK LTD - BURAK DEMIRCIOLU</td>
<td>0392 363 75 54</td>
<td>MAGOSA</td>
</tr>
<tr>
<td>MIDA COMPUTERS - OZDEN DENIZKURDU</td>
<td>0392 753 75 25</td>
<td>GUZELYURT</td>
</tr>
</table> </td>
</tr>
<!--#include virtual="includes/bottom.asp"-->

```

“Transportation.asp”

```

<table width="900" cellspacing="0" cellpadding="0" align="center" border="1">
<!--#include virtual="includes/top.asp"-->
<tr>
<td align="center"><BR><BR><BR><PRE><h4><strong>
<font color="#000000">CARGO COMPANIES THAT WE ARE IN TOUCH<BR>
ARAS CARGO, TEL: 2278539, ADDRESS: ATATÜRK STREET, NO: 4,
NICOSIA<BR>
ORDERS ARE DELIVERED TO YOUR ADDRESS VIA CARGO IN A WEEK
TIME.</font></strong></h4></PRE><BR><BR>

</td>

```

```

</tr>
<!--#include virtual="includes/bottom.asp"-->
</table>

```

“Administrator.asp”

```

<td colspan="3"> </td>

```

```

<p align="center"><font size="5" align="center"><font
color="#0033FF"><em><strong>WELCOME
TO THE ADMINISTRATOR PAGE</strong></em></font></font></p>
<table align="center">
<tr>
<td>
<table width="366" border="1" align="center" cellpadding="0" cellspacing="0">
<tr>
<td width="110">
<div align="center"><a
href="admin_of_the_site.asp?p=plist">PRODUCT
LIST</a> </div></td>
<td width="120">
<div align="center"><a
href="admin_of_the_site.asp?p=pnew">NEW
PRODUCT</a> </div></td>
<td width="65">
<div align="center"><a
href="admin_of_the_site.asp?p=porders">ORDERS</a>
</div></td>
</tr>

```

```

        </table>
    </td>
</tr>
</table>
<%
p=request.QueryString("p")
if p="plist" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
sql="select * from products"
set rs=conn.execute(sql)
%>
<table border="1">
<%
while not rs.eof
%>
    <tr>
        <td width="21">
            <%=rs("Product_id")%>
        </td>
        <td width="22">
            <%=rs("Product_category")%>
        </td>
        <td width="21">
            <%=rs("Product_name")%>
        </td>

```

```

        <td width="20">
            <%=rs("Product_price")%>
        </td>
        <td width="21">
            <%=rs("Product_picture")%>
        </td>
        <td width="21">
            <%=rs("product_quantity")%>
        </td>
        <td width="69">
            <a href="?p=pupdate&ID=<%=rs("Product_id")%>">CHANGE</a>
        </td>
        <td width="60">
            <a href="?p=pdelete&ID=<%=rs("Product_id")%>">DELETE</a>
        </td>
    </tr>
<%
rs.movenext
wend
elseif p="pupdate" then
ID=request.QueryString("ID")
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
sql="select * from products where product_ID="&ID
set rs1 = conn.execute(sql)
%>

```

```

<table align="center">
<form action="?p=update&ID=<%=ID%>" method="post">
  <tr>
    <td>
      PRODUCT CAT:
    </td>
    <td>
      <input
        type="text"
        name="prod_cat"
        value="<%=rs1("product_category")%>"
      </td>
    </tr>
    <tr>
    <td>
      PRODUCT NAME:
    </td>
    <td>
      <input
        type="text"
        name="prod_name"
        value="<%=rs1("product_name")%>"
      </td>
    </tr>
    <tr>
    <td>
      PRODUCT PRICE:
    </td>
    <td>
      <input
        type="text"
        name="prod_price"
        value="<%=rs1("product_price")%>"

```

```

        </td>
    </tr>
    <tr>
        <td>
            PRODUCT PICTURE:
        </td>
        <TD>
            <input
                                type="text"
                                name="prod_picture"
value="<%=rs1("product_picture")%>">
        </TD>
    </tr>
    <tr>
        <td>
            PRODUCT QUANTITY:
        </td>
        <TD>
            <input
                                type="text"
                                name="prod_quantity"
value="<%=rs1("product_quantity")%>">
        </TD>
    </tr>
    <tr>
        <td>
            PRODUCT BRIEF EXP :
        </td>
        <td>
            <textarea
name="prod_brief_exp"><%=rs1("product_briefdesc")%></textarea>
        </td>
    </tr>

```

```

</tr>
<tr>
    <td>
        PRODUCT FULL EXP:
    </td>
    <td>
        <textarea name="prod_full_exp"><%=rs1("product_fulldesc")%></textarea>

    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <input type="submit" value="Update">
    </td>
</tr>
</form>
</table>
<%
elseif p="update" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
prod_ID = request.QueryString("ID")
prod_cat=request.form("prod_cat")
prod_name=request.form("prod_name")
prod_price=request.form("prod_price")
prod_quantity=request.form("prod_quantity")

```

```

prod_picture=request.form("prod_picture")
prod_brief_exp=request.form("prod_brief_exp")
prod_full_exp=request.form("prod_full_exp")
sql = "update products set product_category = " & prod_cat & ",product_name = " &
prod_name & ",product_price = " & prod_price & " "&_
" , product_picture = " & prod_picture & " , product_briefdesc= " & prod_brief_exp & ""
&_
" , product_fulldesc = " & prod_full_exp & " , product_quantity= "& prod_quantity&"
WHERE product_ID = " & prod_ID
conn.execute(sql)
response.Write("<CENTER>The product has been updated.</CENTER>")
conn.close
elseif p="pdelete" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
sql = "DELETE from products WHERE product_ID = " & request.QueryString("ID")
conn.execute(sql)
response.Write("<CENTER>The product is deleted from the database.</CENTER>")
elseif p="pnew" then
%>
<table align="center">
<form action="?p=new&ID=<%=ID%>" method="post">
    <tr>
        <td>
            PRODUCT CAT:
        </td>

```

```

    <td>
        <input type="text" name="prod_cat">
    </td>
</tr>
<tr>
    <td>
        PRODUCT NAME:
    </td>
    <td>
        <input type="text" name="prod_name">
    </td>
</tr>
<tr>
    <td>
        PRODUCT PRICE:
    </td>
    <td>
        <input type="text" name="prod_price">
    </td>
</tr>
<tr>
    <td>
        PRODUCT QUANTITY:
    </td>
    <TD>
        <input type="text" name="prod_quantity">
    </TD>
</tr>

```

```

<tr>
  <td>
    PRODUCT PICTURE:
  </td>
  <TD>
    <input type="text" name="prod_picture">
  </TD>
</tr>
<tr>
  <td>
    PRODUCT BRIEF EXP :
  </td>
  <td>
    <textarea name="prod_brief_exp"></textarea>
  </td>
</tr>
<tr>
  <td>
    PRODUCT FULL EXP:
  </td>
  <td>
    <textarea name="prod_full_exp"></textarea>
  </td>
</tr>
<tr>
  <td colspan="2" align="center">
    <input type="submit" value="ADD">
  </td>

```

```

        </tr>
    </form>
</table>
<%
elseif p="new" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
prod_name=request.form("prod_name")
prod_cat=request.form("prod_cat")
prod_price=request.form("prod_price")
prod_picture=request.form("prod_picture")
prod_quantity=request.form("prod_quantity")
prod_brief_exp=request.form("prod_brief_exp")
prod_full_exp=request.form("prod_full_exp")
sql="INSERT INTO PRODUCTS
(product_name,product_category,product_price,product_picture,product_quantity,product_
briefdesc,product_fulldesc) VALUES ('" & prod_name & "'&_
"," & prod_cat & "," & prod_price & "," & prod_picture & "," & prod_quantity & "," &
prod_brief_exp & "," & prod_full_exp & "')"

conn.execute(sql)
response.Write("<CENTER>The insert operation has been made.</CENTER>")
elseif p="porders" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")

```

```
conn.open
```

```
sql = "SELECT * from orders"
```

```
set rs = conn.execute(sql)
```

```
%>
```

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

```
  <tr>
```

```
    <td>
```

```
      ORDER ID
```

```
    </td>
```

```
    <td>
```

```
      ORDER NAME
```

```
    </td>
```

```
    <td>
```

```
      ORDER SURNAME
```

```
    </td>
```

```
    <td>
```

```
      ORDER TEL
```

```
    </td>
```

```
    <td>
```

```
      ORDER ADDRESS
```

```
    </td>
```

```
    <td>
```

```
      ORDER CCNO
```

```
    </td>
```

```
    <td>
```

```
      ORDER PRODUCT ID
```

```
    </td>
```

```
    <td>
```

```

ORDER QUANTITY
</td>
<td>
ORDER PRICE
</td>
<td>
ORDER CCTYPE
</td>
<td>
ORDER EXPDATE
</td>
</tr>
<%
while not rs.eof
%>
<tr>
<td>
<%=rs("order_ID")%>
</td>
<td>
<%=rs("order_name")%>
</td>
<td>
<%=rs("order_surname")%>
</td>
<td>
<%=rs("order_tel")%>
</td>

```

```

<td>
    <%=rs("order_address")%>
</td>
    <td>
        <%=rs("order_ccno")%>
    </td>
<td>
    <%=rs("order_Prod_ID")%>
</td>
    <td>
        <%=rs("order_quantity")%>
    </td>
<td>
    <%=rs("order_price")%>
</td>
    <td>
        <%=rs("order_cctype")%>
    </td>
<td>
    <%=rs("order_expdate")%>
</td>
</tr>
<tr>
    <td>
        <a href="?p=ordershipped&ID=<%=rs("order_ID")%>">Ship</a>
    </td>
</tr>
<%

```

```

rs.movenext
wend
%>
</table>
<%
elseif p="ordershipped" then
set conn = server.CreateObject("ADODB.Connection")
conn.provider = "Microsoft.JET.OLEDB.4.0"
conn.connectionstring = Server.MapPath("../db/storedb.mdb")
conn.open
order_ID = request.QueryString("ID")
sql = "update orders set order_shipped=true,order_ship_date=" & date & " where
order_ID=" & order_ID
response.Write("<center>The update operation is completed. For the order ID = " &
order_ID & "</center>")
conn.execute(sql)
end if
%>
</table>

```

APPENDIX – B

Microsoft Access - [Products : Table]						
File Edit View Insert Format Records Tools Window Help						
Type a question for help						
product_id	product_name	product_price	product_picture	product_category	product briefdesc	
21	PC 9305 FLAT Intel P4 3.0 Ghz 800 M	1.100,00 €	pictures/desktop/resimler/beko/kucuk/PC-9305.j	Desktop	Beko Desktop Pc	
22	PC 9320 LCD Intel P4 3.2 800 FSB	1.050,00 €	pictures/desktop/resimler/beko/kucuk/PC-9320.j	Desktop	Beko Desktop	
23	PC 9327 MCE LCD Intel P4 3.2 800 F	1.150,00 €	pictures/desktop/resimler/beko/kucuk/PC-9327.j	Desktop	Beko Desktop Pc	
24	Dimension 4700 Enhanced Performan	1.500,00 €	pictures/desktop/resimler/dell/kucuk/4700_3qtr	Desktop	Dell Desktop Pc	
25	Dimension 4700C Enhanced Performa	1.600,00 €	pictures/desktop/resimler/dell/kucuk/4700cfp_6o	Desktop	Dell Desktop Pc	
26	Dimension 8400 Cutting Edge Techno	1.700,00 €	pictures/desktop/resimler/dell/kucuk/8400_3qtr	Desktop	Dell Desktop Pc	
27	Dimension 8400 Media Center Edition	1.800,00 €	pictures/desktop/resimler/dell/kucuk/8400_3qtr	Desktop	Dell Desktop Pc	
28	Dimension XPS Gen 4 Ultimate Gamir	2.000,00 €	pictures/desktop/resimler/dell/buyuk/xps4_front	Desktop	Dell Desktop Pc	
29	A1000N SERIES	1.200,00 €	pictures/desktop/resimler/hp/kucuk/a1000chassi	Desktop	HP Desktop Pc	
30	A850Y	1.000,00 €	pictures/desktop/resimler/hp/kucuk/a1000chassi	Desktop	HP Desktop Pc	
31	Inspiron XPS Gen 2 Extreme Gaming	1.500,00 €	pictures/laptop/resimler/dell/kucuk/inspiron XPS	Laptop	Dell Laptop	
32	Inspiron 9900 Entertainment Powerho	1.400,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 9900	Laptop	Dell Laptop	
33	Inspiron 700m Wide Screen Traveler	1.350,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 700m	Laptop	Dell Laptop	
34	Inspiron 600m Compact Design	1.250,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 600m	Laptop	Dell Laptop	
35	Inspiron 6000 Series Versatile Affordab	1.200,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 6000	Laptop	Dell Laptop	
36	Inspiron 2200 Essential Features , aff	1.100,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 2200	Laptop	Dell Laptop	
37	Inspiron 1000 Value Solution	1.000,00 €	pictures/laptop/resimler/dell/kucuk/inspiron 1000	Laptop	Dell Laptop	
38	Z06000	2.200,00 €	pictures/laptop/resimler/hp/kucuk/z06000chassi	Laptop	HP Laptop	
39	DV4000	2.000,00 €	pictures/laptop/resimler/hp/kucuk/dv4000chassi	Laptop	HP Laptop	
40	DV1000	1.800,00 €	pictures/laptop/resimler/hp/kucuk/dv1000chassi	Laptop	HP Laptop	
41	ZV5000T	1.700,00 €	pictures/laptop/resimler/hp/kucuk/zv5000t_gd	Laptop	HP Laptop	
42	ZE2000	1.650,00 €	pictures/laptop/resimler/hp/kucuk/ze2000chassi	Laptop	HP Laptop	
43	SATELLITE M45-S351	1.999,00 €	pictures/laptop/resimler/toshiba/buyuk/prod_sath	Laptop	Toshiba Laptop	
44	SATELLITE M45-S3511	1.899,00 €	pictures/laptop/resimler/toshiba/buyuk/prod_sath	Laptop	Toshiba Laptop	
45	SATELLITE P35-S611	1.699,00 €	pictures/laptop/resimler/toshiba/buyuk/prod_sath	Laptop	Toshiba Laptop	
46	SATELLITE P35-S629	1.499,00 €	pictures/laptop/resimler/toshiba/buyuk/prod_sath	Laptop	Toshiba Laptop	
47	DVR-1096K	50,00 €	pictures/dvd-rw/resimler/kucuk/1105441685609d	DVD-RW	Pioneer DVD-RW	
48	External Slim USB/Firewire DVD-R/R	60,00 €	pictures/dvd-rw/resimler/kucuk/106596006712d	DVD-RW	Pioneer DVD-RW	
49	CREATIVE 3D BLASTER 5 R3900 P	40,00 €	pictures/graphic card/resimler/creative/n3900pro	Graphic Card	Creative GC	
50	CREATIVE 3D BLASTER 5 R3900 S	50,00 €	pictures/graphic card/resimler/creative/n3900sa	Graphic Card	Creative GC	
51	CREATIVE 3D BLASTER X700 PRO	60,00 €	pictures/graphic card/resimler/creative/x700pro.j	Graphic Card	Creative GC	
52	CREATIVE 3D BLASTER X300	50,00 €	pictures/graphic card/resimler/creative/x300.jpg	Graphic Card	Creative GC	
53	GV-R3850S6V-B (ATI RADEON X850C	60,00 €	pictures/graphic card/resimler/gigabyte/kucuk/gb	Graphic Card	GigaByte GC	
54	GV-R370P256V (ATI RADEON X700 F	55,00 €	pictures/graphic card/resimler/gigabyte/kucuk/gb	Graphic Card	GigaByte GC	
55	GV-N366126VP (NVIDIA GeForce 66	70,00 €	pictures/graphic card/resimler/gigabyte/kucuk/gb	Graphic Card	GigaByte GC	

Figure 1: Product Table

Microsoft Access - [Orders : Table]

File Edit View Insert Format Records Tools Window Help

Type a question for help

Order ID	Order Name	Order Surname	Order tel	Order address	Order conn	Order prod ID	Order quantity	Order price	Order cctype	Order exp
1	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180		
2	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180		
3	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180	master	
4	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180	master	10/10/
5	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180	master	10/10/200
6	Bura	Demirci	05466548655	asdf	521-00000-0000	64	1	180	master	10/10/200
7	Bura	Demirci	05466548655	asdf	521-00000-0000	64	3	540	master	10/10/200
8	Halil	Demircioglu	05428519400	Gocmenkoy	521-00000-0000	30	1	1000	master	25/05/200
9	Halil	Demircioglu	05428519400	Gocmenkoy	521-00000-0000	58	1	105	master	25/05/200
10	gnf	jgfd	467	slt	521-00000-0000	64	1	180	master	2/15/2005
11	Bugra	Demircioglu	05336662780	Lelikosa	521-00000-0000	56	1	60	master	25/05/200
12	Kamil	Ugurlu	05336612160	Ortakoy	521-00000-0000	63	1	200	master	25/05/200
13	shsh	fehshshsfy	89678958	ydfuj	372-00000-0000	64	1	180	American Expr	03/04/200
14	burakko	demirci	13241543	asdf	521-00000-0000	64	1	180	master	12/12/200
15	jgdf	gdfhgf	568	hgdhgd	521-00000-0000	45	1	1639	master	22/2/2005
16	ghht	th	5y6546	thg	521-00000-0000	58	1	105	master	22/2/2005
17	umit	ilhan	23234	teawter	521-00000-0000	48	1	60	master	28/05/200
18	jgdf	hffd	6675	d/fjgdf	521-00000-0000	28	1	2000	master	22/2/2005
* (AutoNumber)							0	0	0	

Record: 14 of 18

Datasheet View

Figure 2: Order Table

