

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

**ONLINE SHOP ACTIVE SERVER PAGES
PROGRAMMING**

**Graduation Project
COM-400**

Student: Hüseyin POLAT

Supervisor: Ümit İLHAN

Nicosia-2003

ACKNOWLEDGEMENT

First, Thank god for all What I have been up to and thank Dr. Suat Günsel for granting me the opportunity reaching this level of education which is almost the most important level of my goals.

Second, I would like to thanks my supervisor Ümit İLHAN. For his invaluable advice and belief in my work anhd myself over thne course at this graduating Project. Helping me usage of ASP and thanks Ayşe helping me.

Third, I thank my family for their constant encouragement, love and support during the preparation of this Project.

ABSTRACT

The people have no choose to live, but the people have to continue the life. Thanks to God that, every person has the different characteristics. According to this, every person has a job. And also some of the people have a lot of jobs. These jobs can take most of the time o a day. For this reason some people try to make the jobs easier then before. We are doing this job. We are trying to earn more time from the day. We are trying to make every operation easily, quickly and rightly.

This Online Shopping program is the example of this idea. Because before the computer spread to world, there was a lot of complex calculations and also more paper using just to take the shopping records. But today's world allows a person about the keeping the more than 10,000 people records at the same time. The keeping, and finding too.

This program makes the every process of the Online Shopping easiest. Take pride for prepare this program to the real business world.

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	I
ABSTRACT.....	II
TABLE OF CONTENTS.....	III
LIST OF TABLE.....	VI
LIST OF PICTURES.....	VII
CHAPTER ONE.....	1
1.1 DATA BASE STRUCTURE.....	1
1.2 WORING WITH HTML AND ASP.....	4
1.3 WHAT IS WEP FORM.....	4
1.4 WHAT ARE ASP.NET WEB FORMS.....	4
1.5 WEB FORM ARCHITECTURE.....	5
1.6 FEATURES OF WEB FORM DESIGN TIME.....	5
1.7 FEATURES OF WEB FORM RUNTIME.....	6
1.8 WEB FORM CONTROLS.....	6
1.9 HTML CONTROLS.....	6
1.10 WEB CONTROLS.....	7
1.11 WEB FORMS AND RAD EXPERIENCE.....	10
1.12DISTRIBUTED TRANSACTION PROCESSING APPLICATIONS.....	11
1.13 ATOMIC.....	12
1.14 CONSISTANT.....	12
1.15 ISOLATED.....	12
1.16 DURABLE.....	12
1.17 MICROSOFT TRANSACTION SERVER (MTS).....	13
1.18 TRANSACTION SUPPORT SETTINGS.....	14
1.19 TRANSACTION CONTROL.....	15
1.20 WHAT IS ASP.....	17
1.22 WHERE CAN I GET ASP.....	18
1.22 WINDOWS NT 4.0 OPTION PACK.....	19
1.23 INSTALLATION AND STUP UNDER WINDOWS NT 4.0.....	19
1.24 SETTING UP ASP UNDER IIS.....	20
1.25 SETTING UP ASP UNDER PWS.....	22

1.27 ASP IS NOT JUST IIS.....	23
1.28 FOUR REASONS TO VALIDATE YOUR HTML.....	23
1.29 OFFLINE HTML.HELP.COM VALIDATOR.....	24
1.30 LICENCE.....	25
1.31 WHAT IS XHTML.....	25
1.32 MISSION OF THE HTML WORKING GROUP.....	25
1.33 RECOMMANDATIONS.....	25
1.34 SUMMARY.....	26
1.35 INTRODUCTION.....	27
CHAPTER TWO.....	28
USER INTERFACE.....	28
2.1 START MENU.....	28
2.2 MAIN MENU.....	29
2.2.1. CATEGORIES COMPUTER PARTS.....	30
2.2.2. ALL STOCKS.....	31
2.2.3. MAIN BOARD.....	33
2.2.3.1. SELECT MOTHERBOARD.....	34
2.2.4. VIDEO ADAPTERS.....	35
2.2.4.1. SELECT VIDEO CARD.....	36
2.2.5. CPU.....	37
2.2.5.1. SELECT CPU.....	38
2.2.6. HARD DRIVES.....	39
2.2.6.1. SELECT HARD DRIVES.....	40
2.2.7. MONITORS.....	41
2.2.7.1. SELECT MONITOR.....	42
2.2.8. CDROM DRIVERS.....	43
2.2.8.1. SELECT CDROM.....	44
2.2.9. SOUND CARDS.....	45
2.2.9.1. SELECT SOUND CARD.....	46
2.2.10. MODEMS.....	47
2.2.10.1. SELECT MODEM.....	48
2.2.11. KEYBOARDS.....	49
2.2.11.1. SELECT KEYBOARD.....	50

2.2.12. PRINTERS.....	51
2.2.12.1. SELECT PRINTER.....	52
2.2.13. SCANNERS.....	53
2.2.13.1. SELECT SCANNER.....	54
CHAPTER THREE.....	55
PURCHASING EVENT.....	55
3.1 ADD BASKET.....	55
3.1.2 BASKET TABLE.....	56
3.1.3 ORDER PAGE.....	57
CHAPTER FOUR.....	59
REGISTER EVENT.....	59
4.1. REGISTER FORM.....	59
CHAPTER FIVE.....	61
MEMBER REGISTER.....	61
5.1.MEMBER FORM.....	61
5.1.1. MY PAGE.....	61
5.1.1.2. EVALUATION.....	62
CONCLUSION.....	63
REFERENCES.....	64
APPENDIX.....	65
CODES.....	65
DATABASE.....	98

LIST OF TABLES

TABLE 1.2. COMPUTER KATEGORİ.....	1
TABLE 1.3 COMPLAR	1
TABLE 1.4. KATEGORİ	1
TABLE.1.5. SİPARİŞ KALEMLERİ.....	2
TABLE 1.6.1. SİPARİŞLER.....	2
TABLE 1.6.2. SİPARİŞLER.....	2
TABLE 1.6.3. SİPARİŞLER.....	2
TABLE 1.6.4. SİPARİŞLER.....	3
TABLE 1.6.5. SİPARİŞLER.....	3
TABLE 1.7. UYELER.....	3
TABLE 1.8. ZİYAREYCI SAYFALARI.....	3

LIST OF PICTURES

PICTURE 1.1.	7
PICTURE 1.2.	8
PICTURE 1.3.	11
PICTURE 1.4.	13
PICTURE 1.4.1.	15
PICTURE 1.5.	21
PICTURE 1.5.1.	21
PICTURE 1.5.2.	22
PICTURE 1.5.3.	23
PICTURE 2.1.....	28
PICTURE 2.2.....	29
PICTURE 2.2.1.	30
PICTURE 2.2.2.	31
PICTURE 2.2.3.	33
PICTURE 2.2.3.1.	34
PICTURE 2.2.4.	35
PICTURE 2.2.4.1.	36
PICTURE 2.2.5.	37
PICTURE 2.2.5.1.	38
PICTURE 2.2.6.	39
PICTURE 2.2.6.1.	40
PICTURE 2.2.7.	41
PICTURE 2.2.7.1.	42
PICTURE 2.2.8.....	43
PICTURE 2.2.8.1.	44
PICTURE 2.2.9.....	45
PICTURE 2.2.9.1.	46
PICTURE 2.2.10.	47
PICTURE 2.2.10.1.	48
PICTURE 2.2.11.	49
PICTURE 2.2.11.1.	50

PICTURE 2.2.12.	51
PICTURE 2.2.12.1.	52
PICTURE 2.2.13.	53
PICTURE 2.2.13.1.	54
PICTURE 3.1.....	55
PICTURE 3.1.1.....	56
PICTURE 3.1.2.	56
PICTURE 3.1.3.	57
PICTURE 3.1.3.1.	58
PICTURE 4.1.	59
PICTURE 4.1.1.	59
PICTURE 4.1.2.	60
PICTURE 5.1.	61
PICTURE 5.1.1.	61
PICTURE 5.1.1.2.	62

CHAPTER ONE

DATABASE STRUCTURE

1.1. Database structure

Program Database includes seven tables. Some tables are given below. Student database table includes about the information student registration.

ALAN ADI	comp_id	kategori_id
VERI TURU	Otomatik Sayı	Otomatik Sayı
alan boyutu	Uzun sayı	Uzun sayı
ondalik deger	Otomatik	Otomatik
varsayilan deger	0	0
gerekli	Hayır	Hayır
siralli	Evet	Evet

Table 1.2. Computer kategori

ALAN ADI	VERI TURU	alan boyutu
comp_id	sayı	Uzun sayı
comp_ismi	Metin	50
kategori	Metin	50
fiyat_TL	sayı	Uzun sayı
sayfa_ismi	Evet	50

Table 1.3 Complar

ALAN ADI	kategori_id	kategori_ismi
VERI TURU	Otomatik Sayı	Metin
alan boyutu	Uzun sayı	50
ondalik deger	Otomatik	Otomatik
varsayilan deger	0	0
gerekli	Hayır	Hayır
siralli	Evet	Evet

Table 1.4. Kategori

ALAN ADI	siparis_id	kategori_id	adet
VERİ TURU	Otomatik Sayı	Otomatik Sayı	Otomatik Sayı
alan boyutu	Uzun sayı	Uzun sayı	Uzun sayı
ondalık deger	Otomatik	Otomatik	Otomatik
varsayılan deger	0	0	0
Gerekli	Hayır	Hayır	Hayır
Siralli	Evet	Evet	Evet

Table.1.5. Sipariş Kalemleri

ALAN ADI	siparis_id	ad	soyad
VERİ TURU	Otomatik Sayı	Metin	Metin
alan boyutu	Uzun sayı	50	50
ondalık deger	Otomatik	-	-
varsayılan deger	0	0	0
gerekli	Hayır	Hayır	Hayır
siralli	Evet	Evet	Evet

Table 1.6.1. Siparişler

ALAN ADI	adres1	adres2
VERİ TURU	Metin	Metin
alan boyutu	50	50
ondalık deger	-	-
varsayılan deger	0	0
Gerekli	Hayır	Hayır
Siralli	Evet	Evet

Table 1.6.2. Siparişler

ALAN ADI	sehir	posta_kodu	tel
VERİ TURU	Metin	Metin	Metin
alan boyutu	50	50	50
ondalık deger	-	-	-
varsayılan deger	0	0	0
Gerekli	Hayır	Hayır	Hayır
Siralli	Evet	Evet	Evet

Table 1.6.3. Siparişler

ALAN ADI	nakliye	toplam	kk_tip
VERİ TURU	Metin	Metin	Metin
alan boyutu	50	50	50
ondalik deger	-	-	-
Varsayılan deger	0	0	0
Gerekli	Hayır	Hayır	Hayır
Siralli	Evet	Evet	Evet

Table 1.6.4. Siparişler

ALAN ADI	kk_num	kk_ustundeki_isim	durum
VERİ TURU	Metin	Metin	Metin
alan boyutu	50	50	50
ondalik deger	-	-	-
Varsayılan deger	0	0	0
Gerekli	Hayır	Hayır	Hayır
Siralli	Evet	Evet	Evet

Table 1.6.5. Siparişler

ALAN ADI	ad	soyad	kullanici_ismi	sifre	email
VERİ TURU	Metin	Metin	Metin	Metin	Metin
alan boyutu	50	50	50	50	50
ondalik deger	-	-	-	-	-
Varsayılan deger	0	0	0	0	0
Gerekli	Hayır	Hayır	Hayır	Hayır	Hayır
Siralli	Evet	Evet	Evet	Evet	Evet

Table 1.7. Uyeler

ALAN ADI	kullanici_ismi	kategori_id
VERİ TURU	Metin	Otomatik Sayı
alan boyutu	50	Uzun sayı
ondalik deger	-	Otomatik
varsayılan deger	0	0
gerekli	Hayır	Hayır
siralli	Evet	Evet

Table 1.8. Ziyareci Sayfaları

1.9. Working with HTML and ASP

ASP.NET is a revolutionary new programming framework that enables the rapid development of powerful web applications and services. Being part of the emerging Microsoft .NET Platform, it provides the easiest and most scalable way to build, deploy and run distributed web applications that can target any browser or device.

1.10 What is Web Form?

Web Forms offer a compelling new way to architect Web software. They bridge the gap between traditional application models and Web programming models by making use of the same form-based programming paradigm. Developer can easily build cross-platform, cross-browser web applications.

1.11. What are ASP.NET Web Forms?

ASP.NET web forms are used to create applications in which the primary user interface is a browser.

Web forms are an ASP.NET technology designed to easily build dynamic, interactive web pages. The ASP.NET Web Forms Page Framework is a scalable Common Language Runtime programming model that can be used on the server to dynamically generate web pages.

Web Forms simplifies the development of Web applications in the following ways:

- Provides an event-based programming model on the server similar to the forms-based development paradigm found in Microsoft® Win32®-based development tools such as the Microsoft Visual Basic® development system.
- Allows complete separation of HTML markup from application logic. The logic, or code behind the page, is compiled and provides much better performance. Furthermore, this code can be written in any Microsoft® .NET language.
- Enables a rich design-time experience. Visual Studio® .NET provides a rapid application development (RAD) experience for creating and managing Web Forms.
- Supports a rich and highly functional set of controls and .NET components that offer a consistent type-safe object model. In addition, the framework naturally lends itself to extensibility through custom and third-party components.
- When POST BACK (values are submitted to the database) takes place, input values are automatically in the controls.

- Web forms use only one source code for all the browsers (IE, Mobile etc).
- Web forms basic functionality is for all the browsers. Web form has a target Schema property which determines which HTML elements can be used. Each target Schema option selects a particular Web browser's Document Object Model (DOM) to be addressed by HTML markup. Available schemas for Beta1 are HTML3.2, HTML 4.0 and Navigator 4.0.

1.12. Web Form Architecture

Web Form Pages are text files with an .aspx filename extension. They can be deployed throughout an IIS virtual root directory tree. A standard Web Forms page consists of a Web Form "markup" file containing the visual, XML-based representation of the page (HTML, WML, XML, etc.), and a source file with event handling code. Web Forms are combinations of HTML, code, and prepackaged server components (controls) that execute on a Web server running Microsoft Internet Information Services (IIS - version 5.0) and they display a user interface by generating—for the most part—ordinary HTML to return to browsers. When a browser client requests .aspx resource, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming request. The .aspx file is only compiled the first time it is accessed. The compiled type instance is reused across multiple requests. The source is compiled into executable code providing fast runtime performance. Both files reside and execute on the server where they generate a Web page that's sent to the client.

1.13. Features of Web Form Design Time

- Use a programming model that is instantly familiar to most developers. Developers can move easily between desktop and Web projects without the need for extensive retraining.
- Separate the HTML layout from the code behind the page. This separation makes it easier to update either piece independent of the other, simplifies code navigation and enables code to be versioned more easily.
- Code can be written in any language that is supported by .NET framework. HTML pages can also be imported from tools like FrontPage and converted into Web Forms.

1.14. Features of Web Form Runtime:

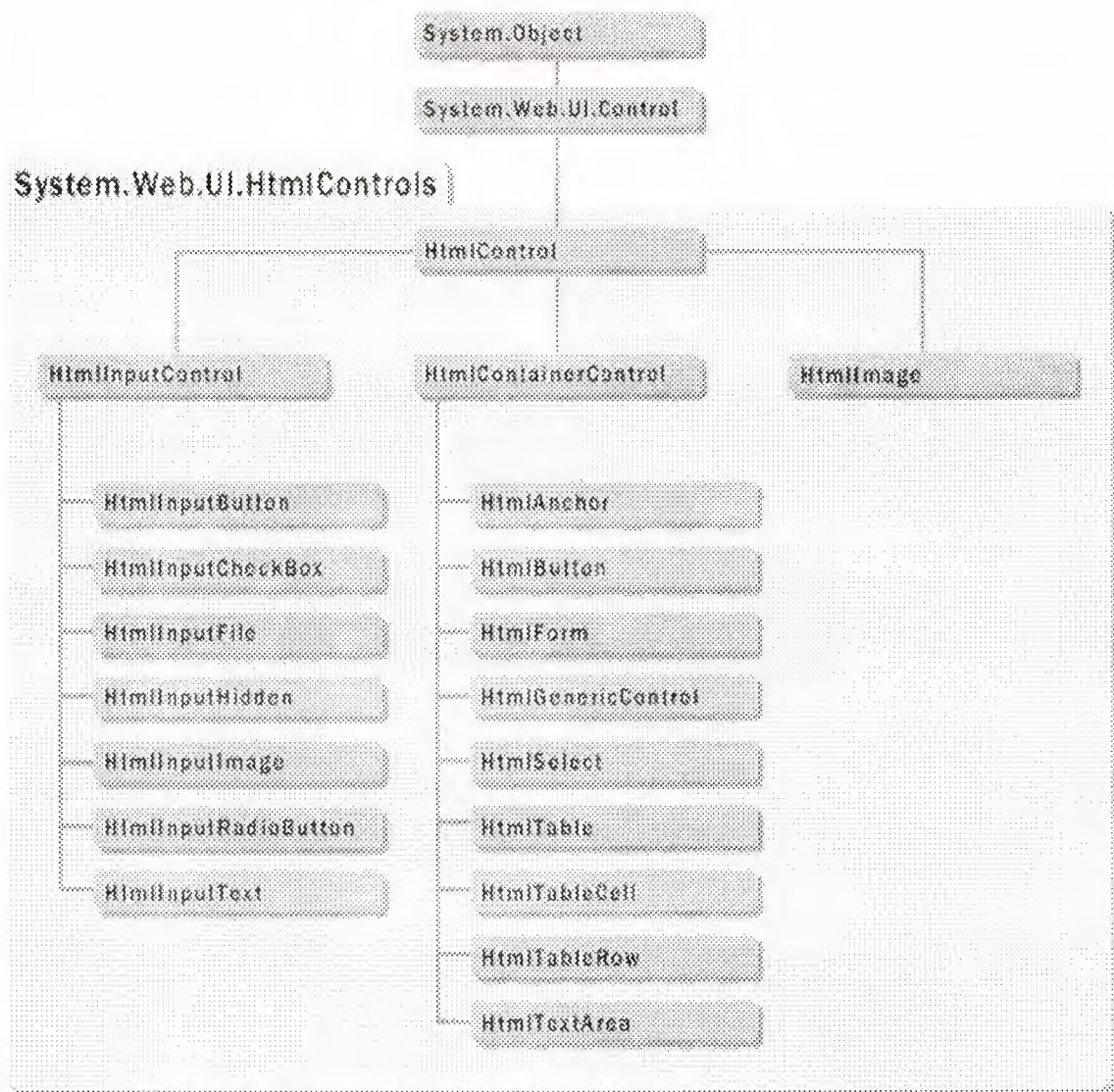
- Deliver a programming model and execution framework that enables developers to implement server-generated HTML pages.
- Greatly enhance runtime performance because the code behind the HTML page is compiled into an executable and not in script.
- Generate HTML pages, which mean the page can be viewed on any platform, with any browser. Alternatively, developers can target the special capabilities of a specific browser or wireless device.

1.15. Web Form Controls

Server controls are used to create the user interface for your Web application. They are allowed to generate any output that is suitable for the device or browser they are targeting. There are two sets of server controls within the ASP+ framework: HTML controls and Web controls, both of which render HTML that is consumed by Web browsers. In addition to their rendering, these controls also encapsulate the mechanism for preserving state across round-trips and raising server-side events for various client events.

1.16. HTML Controls

The first set of controls within the ASP+ framework is called HTML controls. These controls exist within the `System.Web.UI.HtmlControls` namespace, and derive (directly or indirectly) from the `HtmlControl` base class. HTML controls are instantiated for any tag containing the `runat="server"` attribute. For example, the following HTML creates an instance of `HtmlInputText` named "textBox1": `<input type="text" runat="server" id="textBox1" value="some text">` because HTML controls map directly to HTML tags, they are useful in migrating existing ASP pages to ASP+.



Picture 1.1. Illustrates the class hierarchy for the HTML controls

1.17 Web Controls

The second set of the server controls within the ASP+ Framework are called Web controls. These controls exist within the System.Web.UI.WebControls namespace, and derive (directly or indirectly) from the Web Control base class.

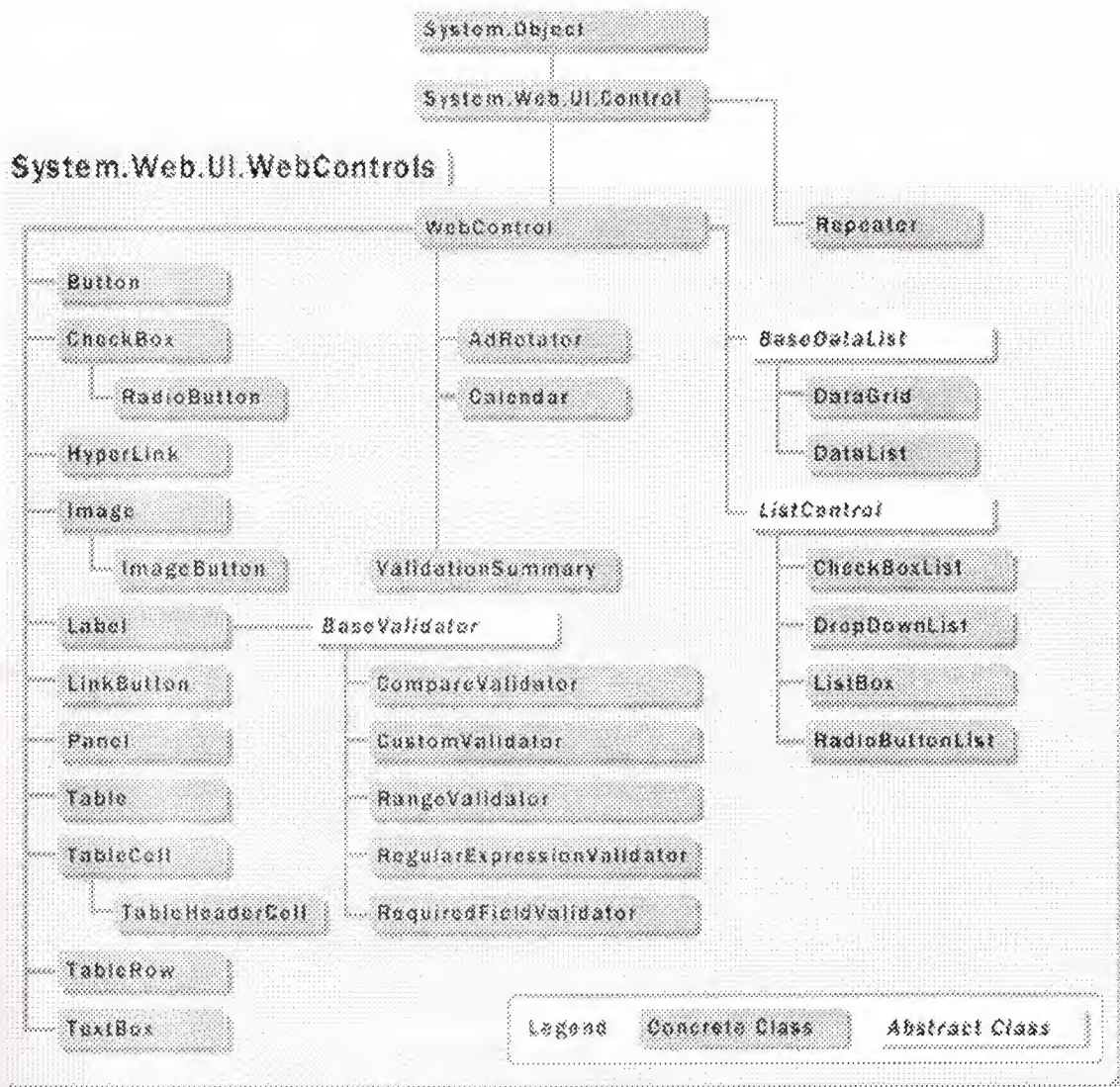
Web controls include traditional form controls such as the Textbox and Button controls as well as other higher-level abstractions such as the Calendar and Data Grid controls.

They provide several features that simplify your development efforts, including:

- A rich and consistent object model. The Web Control base class implements a number of properties that are common to all controls. These include Fore Color, Back Color, Font, Enabled, etc.
- Automatic browser detection. Web controls automatically detect the capabilities of the client browser and can customize their rendering to make best use of those capabilities.
- Data binding. In Web Forms pages, any property of a control can be data bound. In addition, there are several Web controls that can be used to render the contents of a data source.

HTML Controls vs. Web Controls

The two suites of controls can be compared on the basis of various factors.



Picture 1.2. System Web

Sr. No	Factor	HTML Control	Web Control
1	Control Abstraction	The HTML controls do not provide any abstraction. Each control has a one-to-one mapping with an HTML tag.	The Web controls create higher-level abstractions that do not have any HTML equivalents (such as Calendar and Data Grid). They do not map directly to HTML tag.
2	Object Model	The HTML controls provide a very HTML-centric object model. Each control includes an attribute collection, which you can use to manipulate attributes of a tag.	The Web controls provide a forms-based Visual Basic-like programming model. While they provide an attribute collection as well, their focus is on providing a rich type-safe and consistent object model.
3	Target Browser	The HTML controls do not automatically detect the capabilities of the browser requesting the page and do not modify the HTML they render. When you use this set of controls, you are responsible for making sure your page operates in both up-level and down-level clients.	The Web controls provide the functionality to automatically adjust their rendered output to ensure that they work equally well with both up-level and down-level browsers.
4	Control Over Rendering	HTML controls provide you with complete control over what gets rendered and sent to the client browser.	Web controls provide a richer object model and the ability to target multiple browsers. As a result, they do not offer the same level of control over o.p.

1.18. Web Forms and the RAD Experience

Web Forms and the server controls bring the same rapid application development (RAD) experience to Web pages and applications. The toolbox allows dragging and dropping visual components like controls and non-visual components like data sets and other server components onto the design surface.

The design surface itself shows a WYSIWYG view of the controls on the page. The designer also has an associated code-behind window, which allows you to create and edit the logic behind the page. Double-clicking components and controls navigates to the appropriate code in this code-behind window. The Property Browser allows you to customize components on the design surface. The Solution Explorer and the project are used to manage and deploy the components, pages, code-behind files, and other content files present within your Web application.

The development environment also includes a revamped debugger, which can help you debug your Web applications easily.

Creating a Simple Web Form:

Now, we will create a simple web form application, which will retrieve a data from the database and display in data grid. Data grid is list bound control.

Steps for creating the application:

- First create the project using visual basic or C#.
- Drop a data grid control on the web form.
- Write a code in web forms page load event.
- Write a select query.

```
Sqlstr= "select * from employee"
```

Declare a connection with the database.

Execute the select query using the adoCommand.

```
//fill the dataset with the result of the select query.
```

```
MyCommand.filldataset (ds,"Employee")
```

```
//attach data grid to the dataset.
```

```
Datagrid1.datasource=ds
```

```
//binding data to data grid
```

```
Datagrid1.databind ()
```

This event will be fired while page loading. User can see the result of the select query in the data grid.

The Data Grid control creates richly formatted tabular layouts to render the contents of the data source to which it is bound. It provides the ability to support sorting, editing, paging, etc. Data grid can be formatted with the available formatting styles.

```
<asp:DataGrid runat="server" ...>
  <property name="Columns">
    <asp: BoundColumn Header Text="ID" .../>
    ...
  </property>
  ...
</asp: DataGrid>
```

ID	Name	State
154-00-1300	John Doe	CA
000-00-0123	John Millington	CA
172-32-1176	Johnson White	CA

Picture 1.3. Example

1.12. Distributed Transaction Processing Applications

The main concern of a Distributed Transaction processing application is to maintain concurrency and synchronization of the entire transaction taking place. The success of these application lies with dealing of transaction failures. In this article, we will look into how to make distributed transaction processing applications successful.

In most cases, transactional services are provided by the underlying infrastructure. Relational DBMSs such as SQL Server and Oracle provide extensive support for transaction processing. Other types of data sources, such as mainframe applications and message queues, often provide their own forms of transactional support. This built-in support makes writing a transaction much simpler.

Transaction Processing...

What does transaction-processing mean? In simple terms - start a transaction, conduct a series of read and write operations, and finish by committing or aborting the transaction.

Let's have a quick look at the theory behind transaction processing. It is based on ACID rules i.e. a transaction must be atomic, consistent, isolated, and durable.

1.13. Atomic

A transaction must be an all-or-nothing proposition. Everything must be updated successfully or nothing should be updated.

1.14. Consistent

Individual operations within a transaction may leave data in such a state that it violates the system's integrity constraints. Before a transaction completes, the system's data as a whole must be returned to a valid state.

1.15. Isolated

The system must hide the uncommitted changes of a transaction from all the other transactions. The act of hiding or isolating changes is typically accomplished through locking.

1.1. Durable

When a transaction is committed, the data sources involved must keep all changes in stable storage and these changes must be recoverable in the event of a system failure.

Distributed Transactions...

A single database server can enforce the ACID rules when it holds all the data involved in a transaction. For example, SQL Server supplies its own internal transaction manager and can provide commit and rollback behavior. Applications that run distributed transactions are far more challenging to implement. Commit and abort behavior for each transaction must be coordinated across several data sources. Microsoft uses a Windows NT service known as the Distributed Transaction Coordinator (DTC) to run transactions in a distributed environment. Writing a DTC-based transaction involves quite a bit of system-level coding. Furthermore, some of the DTC interfaces are inaccessible to higher-level languages such as Visual Basic. To overcome the DTC programming problem, Microsoft came up with a new technology called Microsoft Transaction Server (MTS). MTS makes it possible to run distributed transactions without having to get involved with the low-level details of the DTC. This same MTS service is a core part of the Windows 2000 operating system with some extra features and is termed as COM+.

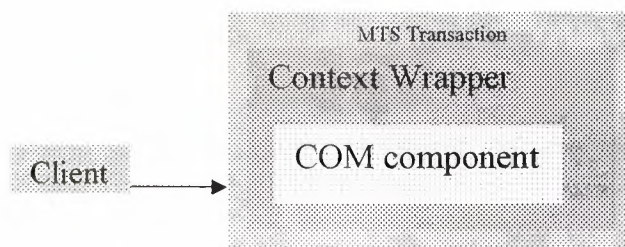
1.17. Microsoft Transaction Server (MTS)

MTS hides all DTC interaction by providing a programming model based on declarative transactions. MTS provides support for in-process server-side components

(ActiveX DLL). MTS can also be defined as Component based programming model. An MTS component is a type of COM component that executes in the MTS run-time environment. All the business logic is built in the component under the scope of MTS.

When a client application calls a MTS component, MTS intercepts the call and wraps the component in an object context wrapper. The object context wrapper takes the place of the object the client application sees.

On creation of the MTS object by the client, the client thinks that it is connected directly to the object, but in reality, it is holding a reference to the context wrapper. The object context wrapper exposes the same methods as the actual component.



Picture 1.4. Transaction

When MTS component calls SetComplete/Setabort, MTS releases all its references to the object. When the component object is deactivated, the client remains connected to the context wrapper. When the client invokes another method call, the MTS runtime creates another instance from the same component and activates it inside the context wrapper. This just-in-time activation policy makes the client think that it's dealing with a single object across method calls. However, behind the context wrapper, the MTS runtime creates and destroys objects on every call.

Every MTS component has a transaction support attribute. When installing (not importing) your DLL with the MTS Explorer, MTS automatically configures your components with the appropriate transaction support attributes. When you invoke a method on transactional objects, MTS interacts with the DTC on your behalf to run distributed transactions.

An MTS object can be created by a client or by another MTS object. When the MTS runtime receives an activation request to create a new object, it determines whether the object's creator is running inside an existing transaction. The MTS runtime also inspects the transaction support attribute of the component. These settings can be

set after the components are hosted on to MTS or when the MTS Components are created using programming language such as Visual Basic 6.0.

1.18. Transaction support settings

- **Requires a Transaction :** The object is created within the context of a transaction. The object is placed inside the transaction of its creator if one exists. If the creator isn't running in a transaction, the MTS runtime creates a new transaction for the object.
- **Requires a New Transaction:** The MTS runtime creates a new transaction for the object.
- **Supports Transactions:** The object is placed inside the transaction of its creator if one exists. If the creator isn't running in a transaction, the object is created without a transaction.
- **Doesn't Support Transactions:** The object isn't created within the context of a transaction.
- **Disabled: (Only for COM+)** The new transaction type, turns off transaction support for the object. Set the transactional property to Disabled for those objects which are not connected to database. Documentation states that objects set to Disabled will see no more overhead from COM+ than a standard object would.

A client can initiate a transaction by activating an MTS object from a component that has the transaction support attribute value as "Requires a Transaction" or "Requires a New Transaction". The MTS runtime determines that the new object must run inside a transaction. Since this is the first object being created, the MTS runtime object creates a new MTS transaction and then creates this new object inside it. The first object created inside a transaction is known as the root.

The root object can create additional objects inside its transaction. If you're writing a method for an MTS object and you want to propagate another object inside that transaction, you must follow two rules. First, you must create the object from components marked, as either "Requires a Transaction" or "Supports Transactions". The MTS runtime will create the new object in the same transaction as its creator with either setting. Second, you must create additional objects with "Create Instance". The scope of every transaction is always contained within the scope of a single MTS

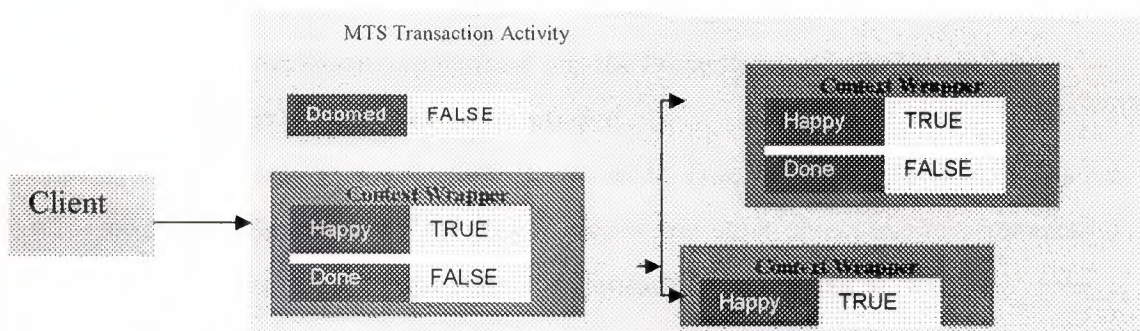
activity. Create Instance always creates the new object inside the same MTS activity and it gives the new object its own context wrapper.

On creation of the root object the MTS runtime calls down to the DTC to create a new transaction on its behalf. Consider the MTS transaction as the logical transaction and the DTC transaction as the physical transaction.

The MTS runtime creates the logical transaction when it creates the root object, but it defers creating the physical transaction until the root object is activated. An MTS object is activated when the client invokes the first method call. The MTS runtime creates the physical transaction with an isolation level Serializable and a timeout interval of 60 seconds.

1.19. Transaction Control

The block diagram shows a diagram of the root object, its context wrapper, and the MTS transaction in which they are running. It also shows some important internal flags maintained by MTS. These variables aren't directly exposed to you, but you should understand how they influence the MTS runtime's decision to commit or abort the transaction.



Picture 1.4.1 Transaction

For understanding how one can control the transaction we will look at some important internal flags maintained by MTS. These flags are as follows

Doomed

Initial Value: False

Purpose: MTS runtime inspects this flag to abort or commit the transaction. If value is "TRUE" then the transaction is aborted if "FALSE" then the transaction is committed and also all the objects are deactivated.

Methods which change the value: MTS inspects the "Happy" flag to set the value and value is same as "Happy" flag.

Happy

Initial Value: True

Purpose: On deactivation of the object the MTS runtime inspects this flag and accordingly sets the doomed flag value

Methods which change the value:

1. Disable Commit method sets the Happy flag to "False"
2. Enable Commit method sets the Happy flag to "True"
3. Set Abort method sets the Happy flag to "False"
4. Set Complete method sets the Happy flag to "True"

Done

Initial Value: False

Purpose: On deactivation of the object the MTS runtime inspects this flag and accordingly sets the doomed flag value

Methods which change the value:

Disable Commit method sets the Done flag to "True"

Enable Commit method sets the Done flag to "False"

Set Abort method sets the Done flag to "True"

Set Complete method sets the Done flag to "False"

1.20. Important points to be remembered

1. Once the transaction's doomed flag is set to True, it can't be reversed. This has a powerful implication: if the root object or any other object inside a transaction is deactivated in an unhappy state, the transaction is doomed to failure.
2. The root object can create secondary objects inside the same transaction by calling Create Instance on components that are marked as either Requires a Transaction or Supports Transactions
3. A secondary object follows the same rules as the root object. When it is deactivated, the MTS runtime inspects its happy flag. If the happy flag is set to False, the MTS runtime sets the transaction's doomed flag to True. When this happens, the transaction is doomed to failure. There is nothing that any other object can do to set the doomed flag back to True.
4. When a secondary object calls Set Abort, the root object should also call Set Abort and halt any attempt to complete additional work. However, there's one small hitch: the root object cannot examine the doomed flag—it can't ask the MTS runtime whether a secondary object has called Set Abort. Therefore, your

secondary objects must communicate with the root object when they call Set Abort.

1.21. What is ASP?

In the dim and distant past, web developers wishing to create more than just static displays of information turned to CGI (Common Gateway Interface) and Perl to introduce some sort of interaction to their pages. While this approach worked (and indeed, many sites still use it today), CGI was not by any means fast and the quest for an alternative means to create a page dynamically continued. ISAPI came and went – primarily because it required more knowledge to create a dynamic filter than web programmers were prepared to learn. Finally came web scripting languages and with them Microsoft's Active Server Pages: a server-side scripting technology for building web pages that are both dynamic and interactive.

An Active Server Page itself is simply a text file script with the extension .asp containing HTML, client- and server-side script. The implementation behind the ASP page was created by Microsoft and intended as an open technology server-side framework, giving web developers the freedom to develop dynamic web sites using information accessed from the many COM-compliant data sources available to them.

The syntax and grammar of ASP is easy to comprehend, and yet powerful enough to:

- Support some interaction between page user and server

- allow web page access to databases and directory services

- Incorporate and make use of high-powered COM components

It's also server-based, and therefore browser-independent, leaving you only the problems of which cross-browser client-side script and style sheet you want to use.

Writing ASP pages is straightforward:

The static content of the page is created using HTML and text. We can also use standard HTML forms to question the client and, using ASP, create a page that incorporates their answers into the text. This is the simplest form of client-server interactivity using ASP.

We make calls to one or several of the ASP intrinsic objects to create the dynamic element of the page. ASP provides a set of intrinsic objects – Response,

Request, Application, Server and Session – that allow us to access information about the request being made of the server, as well as the response that the server will send.

We also have access to the functionality of our standard scripting languages (VBScript, JScript and JavaScript) and the scripting objects they expose.

Finally, we can incorporate any standard COM object (or active server component, as we call them) into a page. There are many third-party companies who specialize in writing such components and if you do use them, then you have the added guarantee that they've been stress-tested and fully debugged. On the other hand, if you need your component to do something unique, COM components are easy to develop in any established language. Visual Studio would be a good investment at this point.

So now that we know what ASP is, where can I get an ASP scripting engine from?

1.22. Where Can I Get ASP?

ASP is a web enhancement tool that was developed and distributed by Microsoft. As such, the ASP scripting host (which translates ASP into HTML) is distributed within Microsoft's web-specific products – its web servers, Internet Information Server (IIS) and Personal Web Server (PWS) and its HTML editors (FrontPage 9x and Visual Interdev, though not Notepad ;-). In the spirit of the open standard that ASP wants to be, the scripting host has also been taken by third parties and rewritten for use with other, non-Microsoft web servers – more on that later.

To date, there have been two versions of ASP for development use. The first, ASP 1.0, appeared in 1996 with Windows NT4 Service Pack 3 as part of IIS 3.0. ASP 2.0 reared its head about nine months later, much improving on its previous incarnation – it is this version which is available for installation with a web server/html editor from quite a few sources. The latest version, ASP 3.0, is due to ship some time in the first half of 1999 with IIS 5.0 (as part of Windows 2000. Since ASP 3.0 is currently not widely available (and is still in beta), we'll concentrate on ASP 2.0 in this book, which you can install from the following sources.

1.23. Windows NT 4.0 Option Pack

Contrary to its rather specific name, the Windows NT 4.0 Option Pack comes in two different flavors (the choice depends on which operating system you are using). If your system set-up is based upon Windows 95 or Windows NT 4 Workstation, then

ASP will be installed with Personal Web Server. If you use Windows NT 4 Server, then ASP is installed with Internet Information Server.

The Option Pack is available from several different sources:

MSDN subscribers will find it on Disk 2 of the BackOffice Test Platform or Disk 13 of the Development Platform.

Other users can download it from the Microsoft home site at <http://www.microsoft.com/ntserver/nts/downloads/recommended/NT4OptPk/default.asp>. The total pack is 27Mb, so be prepared for a long wait – and no, there isn't an option to download just IIS and ASP.

Windows 98 Install Disk

The Windows 98 Install Disk includes a version of Personal Web Server and ASP as an extra to the operating system itself. Simply run setup.exe from the addons/pws directory of the disk and all will be well.

Visual Studio 6

Personal Web Server and IIS, and therefore ASP, are included with this product.

1.24. Installation and Setup under Windows NT 4.0

Windows NT4.0 can be a tricky beast to set up, and getting the latest version of ASP running with the latest hot fixes for security holes and other bugs can be a matter of trial and error unless you know the right order in which to install your software. For an NT 4.0 web server running ASP, it would be preferable to get a copy of the NT 4.0 Option Pack and then install in this order:

Windows NT 4.0 Server/Workstation

NT 4.0 Service Pack 3

Internet Explorer 4.01 (or later)

NT 4.0 Option Pack (or other source of ASP)

NT 4.0 Service Pack 4

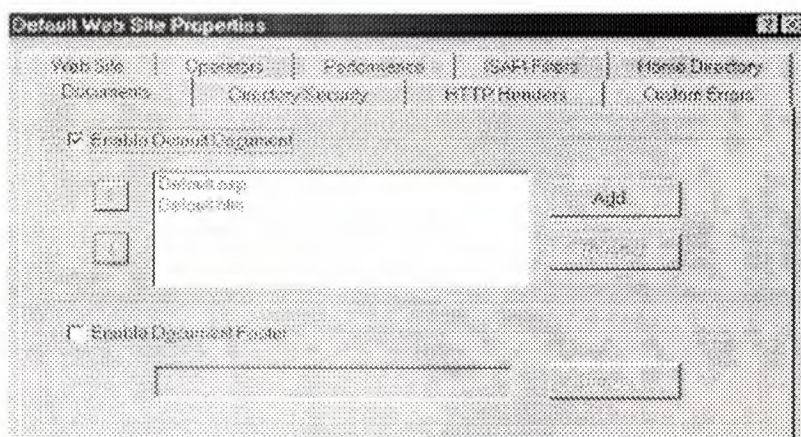
Service Pack 3 is required to install the Option Pack, but in its turn the Option Pack contains older files than SP4 which contains the latest hot fixes (unto November

1998 anyway). Check out the Microsoft security or downloads sites for the fixes after SP4 (see Appendix H for URLs).

1.25. Setting up ASP under IIS

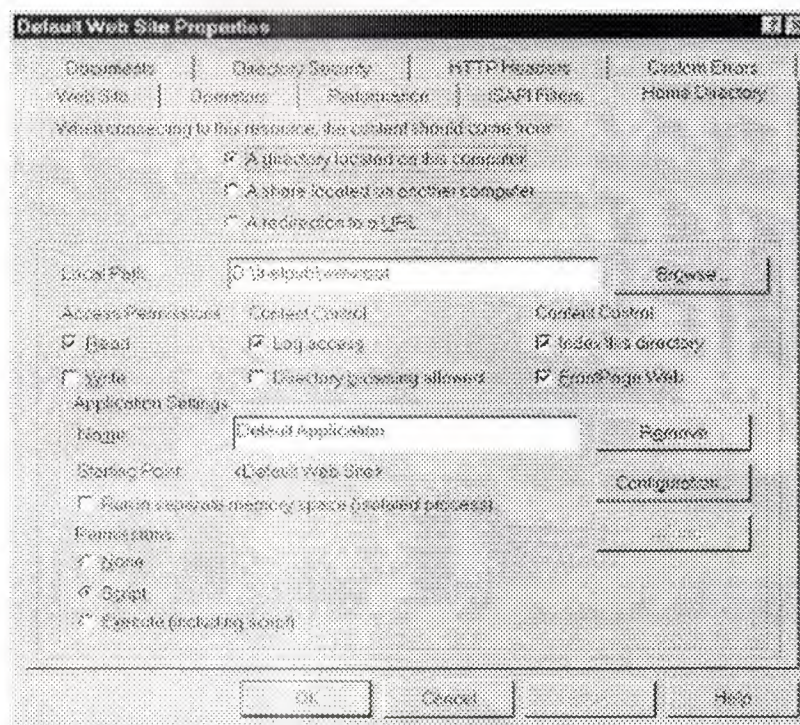
If you've installed Internet Information Server, there are a couple of other things you need to do in order to make sure ASP is running and will run in your test code. Checking that ASP has been installed properly is simplicity itself. IIS documentation comes installed as a set of .asp files – you can test your ASP installation simply by choosing to run those help files from the Start menu (at Programs | NT 4.0 Option Pack | Product Documentation). If the help files display correctly, then ASP is running. If they don't, then it isn't.

The other quick check to make is to ensure that the virtual root for your web site is set up to run .asp files. In the MMC, expand the IIS folder and select either a web site or a virtual directory beneath it. Bring up the Properties dialog for that virtual directory by right-clicking on it and selecting Properties from the menu that appears. Select the Documents tab, add Default.asp to the default documents listbox and promote it above Default.htm using the arrows on the right-hand side.



Picture 1.5. Controller

Finally, click on the Home Directory tab (this will be the Virtual Directory tab if you are setting the properties of a virtual directory instead of a web site). Under the Permissions heading at the bottom of the dialog, make sure the Script radio button is checked – this ensures the ASP scripting host will parse the page when it is requested by a client.

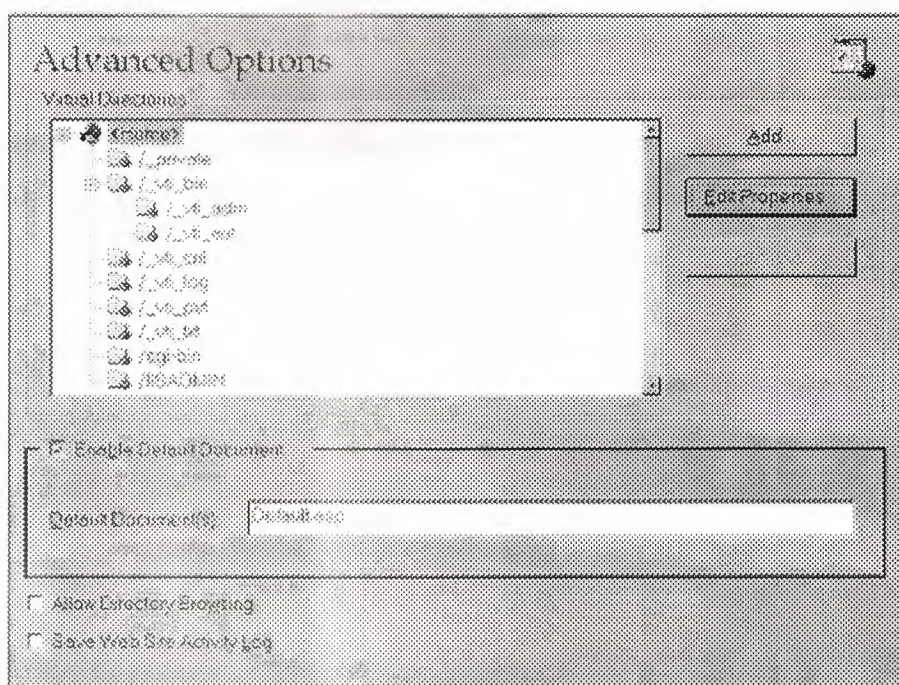


Picture 1.5. 1. Controller

This applies to IIS4.0 only. In IIS3.0 the Script button doesn't exist, so you have to select the Execute button. It was only when IIS3.0 was being used in production setups, that anyone noticed that selecting Execute allows any executable file to be run in that directory – which imposed a security threat on the website . Thus, in IIS4.0 the Script button appears in the Permissions dialog to allow only the 'secure' execution of scripts in the relevant directory.

1.26. Setting Up ASP Under PWS

Personal Web Server is really just a scaled down version of Internet Information Server and to ensure that ASP is running correctly and is available for your site, the checks are the same. The PWS documentation can be run from the Start menu under Programs | Personal Web Server | Product Documentation to confirm that ASP is running, and once you have created a virtual directory for your web site, the default document dialog can be found in the Advanced Options window of PWS.



Picture 1.5.2. Controller

Meanwhile, the Scripts permission button can be found by selecting Edit Properties. Again, make sure that Scripts is checked along with Read.



Picture 1.5.3. Controller

1.27. ASP is not just for IIS

The core of ASP is the ASP Scripting Host, which you can find in `asp.dll` – without it, we would have no working `.asp` files. However, although the vast majority of ASP developers are currently using either Personal Web Server or Internet Information Server to base their pages on, these web servers are no longer the only products that contain a productive ASP engine for us to work with.

The really attractive part of having open standards – even wannabe ones like ASP – for the Internet is that once a standard has been released, it can be implemented by anyone. Soon after Microsoft's release of ASP 1.0, a company called Chili! Soft began to develop a fully-functional parser for other web servers besides IIS and PWS.

On the Windows NT platform, Netscape's Enterprise and Fast track servers are now ASP-enabled thanks to their product Chili! ASP, as are Lotus' Domino and Go Web servers. A version for NT Apache is currently in beta. Chili! ASP also offers ASP and ADO to Netscape servers on the Sun Solaris platform and has plans in the works for more non-PC-based servers too. It will be interesting to see how Chili! Soft react to the release of ASP 3.0. The question of whether or not they'll upgrade their current products is pressing, but whatever the answer there, ASP is no longer just for IIS.

1.28. Four Reasons to Validate your HTML

HTML validation allows authors to check their HTML syntax. Valid HTML is less likely to cause problems with different browsers and more likely to survive the next browser release. There are many reasons to validate your HTML, but here we have chosen to focus on historical examples that show why visually checking your Web pages in browsers is not enough.

Reason 1: Netscape 1.2

When Netscape 1.1 was released, many authors were delighted to find that they could create animated titles and backgrounds by including multiple `<TITLE>` and `<BODY>` tags. Valuators flagged these as errors since only one TITLE and BODY is permitted in HTML.

Netscape 1.2 fixed the bug in Netscape 1.1, and multiple `<TITLE>` and `<BODY>` tags were ignored. The result on some invalid pages varied from an incomplete or incoherent title to unreadable text.

Reason 2: Netscape 2.0

Netscape 2.0 provided the most stark display of the importance of validation. Prior to Netscape 2.0, the Web's most popular browser quietly corrected an author's missing quotation mark, as in `Oops`. Netscape 2.0 began requiring that closing quotation mark, and at the same time parts of many Web pages began disappearing. Valid HTML survived unscathed.

Reason 3: Netscape 3.0

Netscape 3.0 began looking too hard for entities in attribute values. The result was that unescaped ampersands in attribute values could cause Netscape to use the wrong attribute value. For example, the invalid HTML `` worked fine in versions of Netscape prior to 3.0. But Netscape 3.0 would try to fetch "foo.pl? Chapter=1§ion=2" instead of the "right" location. The valid HTML, ``, has worked fine in all versions of Netscape.

Reason 4: Netscape 4.0

Netscape 4.0 began requiring the terminating semicolon on entities where previous versions often had not. For example, some HTML tutorials show their expertise with `<P>Foo` in examples, which Netscape 4.0 shows literally while previous versions had shown "`<P>Foo`". Again, valid HTML worked fine in all versions of Netscape.

1.29. Offline HTML.Help.com Valuator

The Offline HTMLHelp.com Valuator is an offline version of our online validation service. The Valuator checks the syntax of HTML and XHTML documents.

The Offline Valuator is geared toward UNIX users. Windows users may prefer A Real Valuator by the same author.

1.30. License

This program is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

Notes

- Changes in new versions are noted in the change log.
- A README is available. It currently describes how to add your own DTDs.
- I recommend making the "validate" script executable (using `chmod a+x validate`). Otherwise you will need to run `perl validate foo.html` instead of `validate foo.html`.
- The first line of the "validate" script specifies the location of the Perl 5 executable. If your Perl 5 is not in `/usr/bin/perl`, you will need to change the first line of "validate" accordingly.

1.31. What is XHTML?

The Extensible HyperText Markup Language (XHTML™) is a family of current and future document types and modules that reproduce, subset, and extend HTML, reformulated in XML. XHTML Family document types are all XML-based, and ultimately are designed to work in conjunction with XML-based user agents. XHTML is the successor of HTML, and a series of specifications has been developed for XHTML.

1.32. Mission of the HTML Working Group

To develop the next generation of HTML as a suite of XML tag sets with a clean migration path from HTML 4. Some of the expected benefits include: reduced authoring costs, an improved match to database & workflow applications, a modular solution to the increasingly disparate capabilities of browsers, and the ability to cleanly integrate HTML with other XML applications. For further information, see the Charter for the HTML Working Group (members only).

1.33. Recommendations

W3C produces what are known as "Recommendations". These are specifications, developed by W3C working groups, and then reviewed by Members of the Consortium. A W3C Recommendation indicates that consensus has been reached among the Consortium Members that a specification is appropriate for widespread use.

1.34 Summary:

Distributed Transaction processing Applications can have their business logic embedded in MTS components. This will ensure the concurrency and synchronization of the entire transaction-taking place.

This is a brief overview of the ASP.NET Web form. It has many advantages over the traditional designing system. Some of them are as follows

- After running the form first time on the server will compile it and keep cached copy in memory results in increasing the performance.
- Configuration file is a plain text file.
- Uploading or changing can be done while application is running.
- Deployment is easy. All the requests are redirected to new code.

1.35. INTRODUCTION

Online Shopping Program will be the helpful to all members whose are ASP student, and also whose interested in ASP programming language, interesting of software.

Online Shopping program is easier and quicken way of the recording, searching, deleting and editing of the record that the Online Shopping.

ACKNOWLEDGEMENT

First, Thank god for all What I have been up to and thank Dr. Suat Günsel for granting me the opportunity reaching this level of education which is almost the most important level of my goals.

Second, I would like to thanks my supervisor Ümit İLHAN. For his invaluable advice and belief in my work anhd myself over thne course at this graduating Project. Helping me usage of ASP and thanks Ayşe helping me.

Third, I thank my family for their constant encouragement, love and support during the preparation of this Project.

CHAPTER TWO

USER INTERFACE

2.1 Start Menu



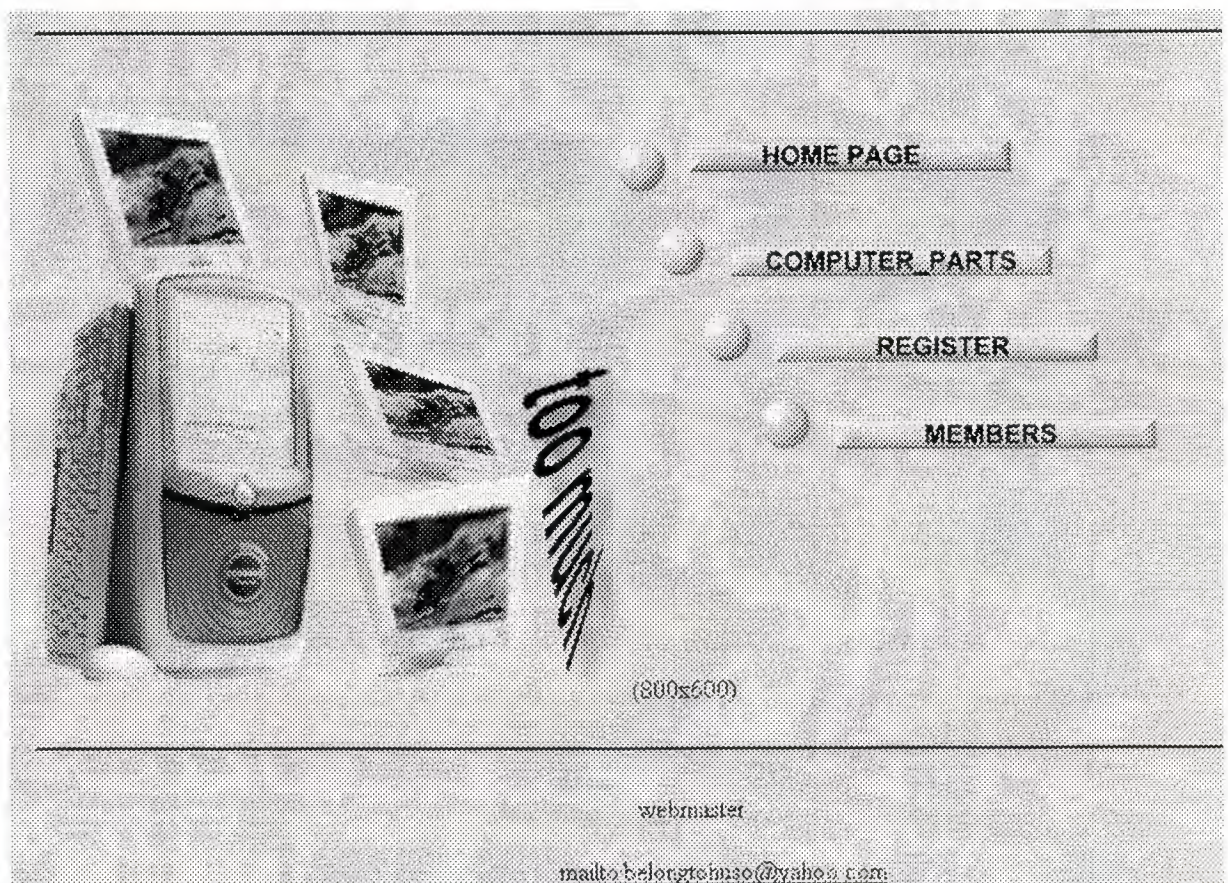
Picture 2.1

Online shopping page is the easier way of the recording searching, deleting and editing of the record that the internet has. I design as simple as possible user. It's user friendly form. Which has designed in ASP. Properties and events were used to create, user-friendly menu.

I tried had to make menu simple and easy understandable as much as. I tried because in these years basic and simple and not complex programs are deving sense and

as I am engineering student. I know this and I am trying to have competitive programs which others are also doing in market actually this is making us to care move on that project. So I can also survive in market Polat Company want me to make simple easy program to be understood. Easier and howmany. I am explaining what kind of menu they wish.

2.2 Main Menu

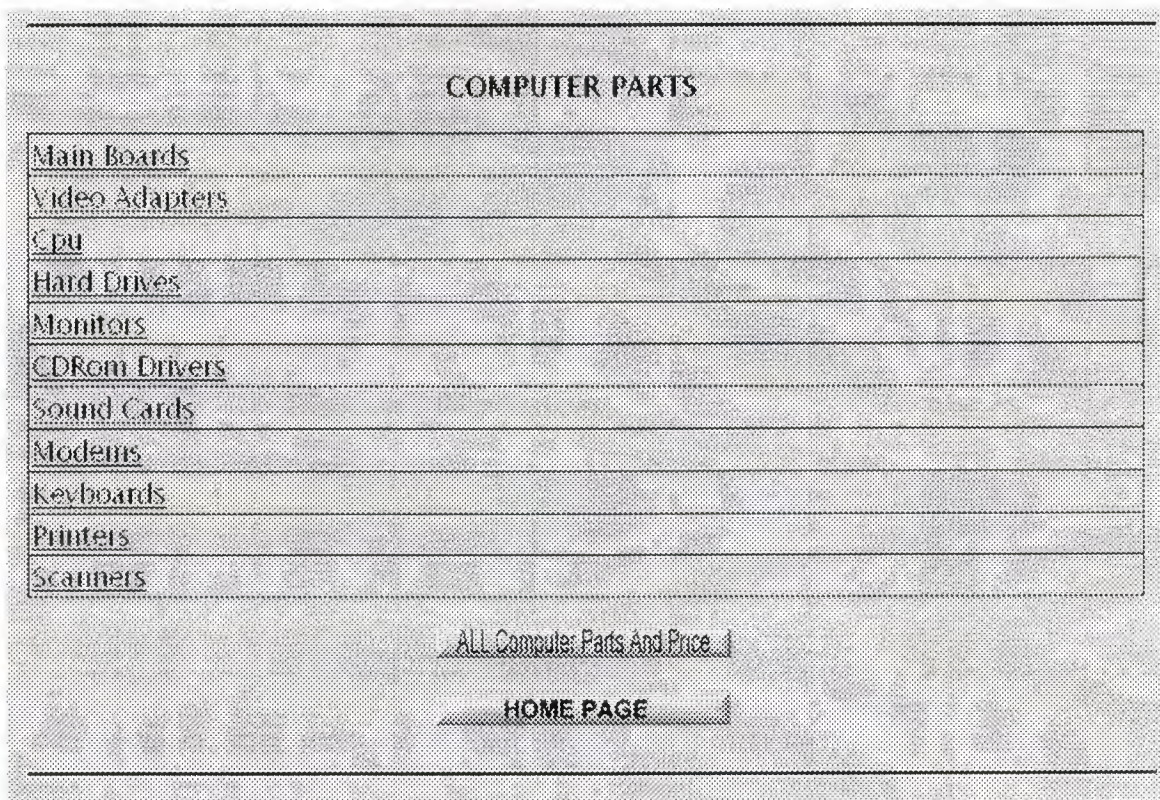


Picture 2.2

When user enter home page on menu on form then we will see user interface menu one by one.

This is the main window of the Online Shopping program. Every application can occur from here. As shown on the main window there are four command buttons to make easier. Access for the user. There command buttons are sequentially home page button, computer parts button, register button, members button.

2.2.1. Categories Computer Parts



Picture 2.2.1.

When user clicks on this menu on form we will see computer parts. These parts are taken from categorie database when we click on it. As I mentioned in other menus in form we also see how many stock to customers and we again keep their parts.

2.2.2. All Stocks

Computer Part Name	Price
100Gb Seagate/W D/7200rpm ATA100	31\$
15" Acer	32\$
17" Acer V771 (27)	33\$
17" Mitsubishi 1770G (27)	34\$
19" Hitachi CM721 Flat (21)	36\$
19" Proview	35\$
20Gb Seagate/W D/5400rpm ATA100	27\$
21 " Mitsubishi 2107 (26)	37\$
32Mb Daytona G2 MX200 TV out	16\$
40Gb Seagate/W D/5400rpm ATA100	28\$
52X Creative CDROM	41\$
52X Diamond Data	39\$
52X Samsung CDROM	38\$
52X Sony CDROM	40\$
56K Data/Fax/Voice External	49\$
56k Data/Fax/Voice Internal	47\$
56K Netcomm Internal	48\$
56K Netcomm Mega External	50\$
60Gb Seagate/W D/7200rpm ATA100	29\$
64Mb Asus V7100 MX400 TV Out	19\$
64Mb Daytona G2 MX200	17\$
64Mb Eagle Geforce 2 MX400-TV OUT	18\$
64Mb Powercolor GeForce3	20\$
80Gb Seagate/W D/7200rpm ATA100	30\$
AMD Athlon XP 1600+ OEM	22\$
AMD Athlon XP 1700+ OEM	23\$
AMD Duron 1000 OEM	21\$
Asus A7V133 Socket A With Raid	15\$
Asus CUSL2-C S370	11\$
Asus TUSL-M S370	10\$
Canon N640P	59\$
Canon N650U	60\$
Creative S/B Audigi 5.1 Digital	45\$
Creative S/B Audigi Platinum 5.1	46\$
Creative S/B Live D E 5.1 Retail Box	44\$
Creative S/B Live! Value! with software	43\$
Creative Vibra 128 PCI with software	42\$

Epson 1250 Photo	62\$
Epson 1650 Photo	63\$
Epson Perfection 1250	61\$
Epson Photo 1290	58\$
Gigabyte 71XE4 Socket A	12\$
Gigabyte 7ZXE Socket A	13\$
Gigabyte 7ZXR-C Socket A	14\$
HP 1200 Laser	56\$
HP 640C	54\$
HP 840C	55\$
HP Multimedia Internet Keyboard	52\$
Intel Pentium 4 1.5 (S478)	34\$
Intel Pentium 4 1.6 (S478)	25\$
Intel Pentium 4 1.7 (S478)	26\$
Keyboard PSII/Senial	51\$
Microsoft Internet Keyboard & Mouse Retail	53\$
Panasonic KXP6100 Laser	57\$

[Back to Computer Parts](#)

Picture 2.2.2.

When you click all computer parts and price button are we will see again stock table.

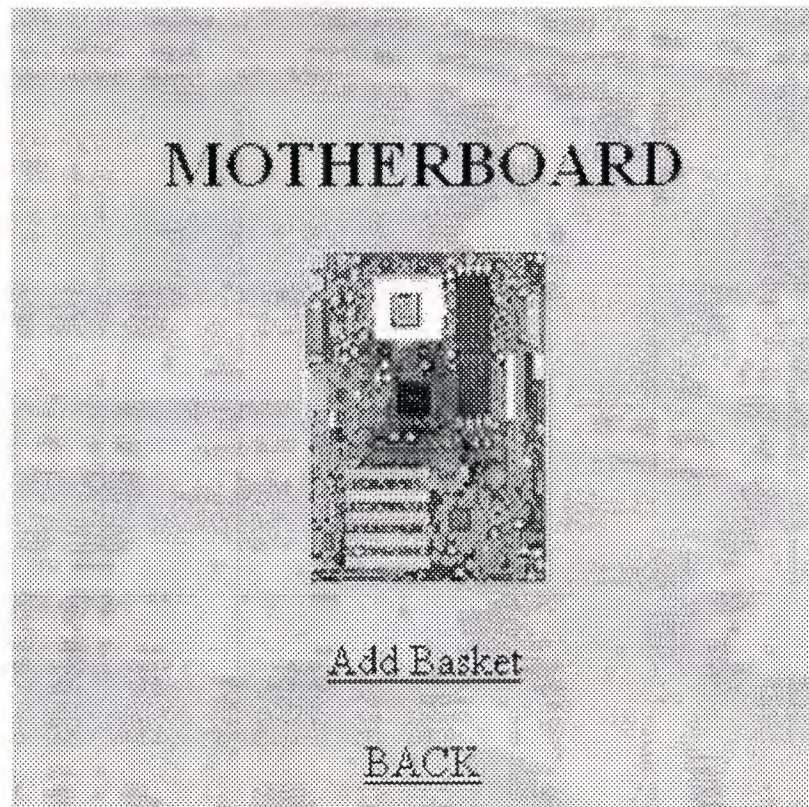
2.2.3. Main Board

Main Boards		
Code	Computer Part Name	Price
101	Asus TUSL-M S370	10
		\$
102	Asus CUSL2-C S370	11
		\$
103	Gigabyte 7IXE4 Socket A	12
		\$
104	Gigabyte 7ZXE Socket A	13
		\$
105	Gigabyte 7ZXR-C Socket A	14
		\$
106	Asus A7V133 Socket A With Raid	15
		\$
<div>BACK</div> <div>HOME PAGE</div>		

Picture 2.2.3.

When user come this page, they will see kinds of main board. And also they can select which one they want.

2.2.3.1. Select Motherboard



Picture 2.2.3.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

2.2.4. Video Adapters

Video Adapters		
Code	Computer Part Name	Price
201	32Mb Daytona G2 MX200 TV out	16
		\$
202	64Mb Daytona G2 MX200	17
		\$
203	64Mb Eagle GeForce 2 MX400 TV OUT	18
		\$
204	64Mb Asus V7100 MX400 TV Out	19
		\$
205	64Mb Powercolor GeForce3	20
		\$

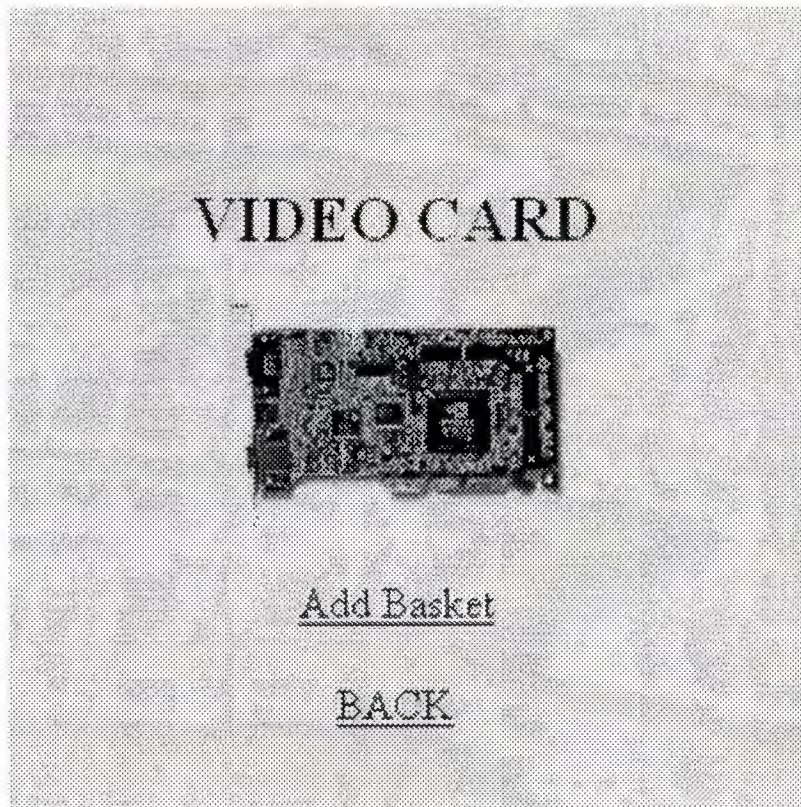
BACK

HOME PAGE

Picture 2.2.4.

When user come this page, they will see kinds of video Adapters. And also they can select which one they want

2.2.4.1. Select Video Card



Picture 2.2.4.1.

After when user select parts, it is seeing parts Picture then user select if they want to buy this parts then click add basket button.

2.2.5. CPU

Cpu

Code	Computer Part Name	Price
301	AMD Duron 1000 OEM	21
		\$
302	AMD Athlon XP 1600+ OEM	22
		\$
303	AMD Athlon XP 1700+ OEM	23
		\$
304	Intel Pentium 4 1.5 (S478)	24
		\$
305	Intel Pentium 4 1.5 (S478)	25
		\$
306	Intel Pentium 4 1.7 (S478)	26
		\$

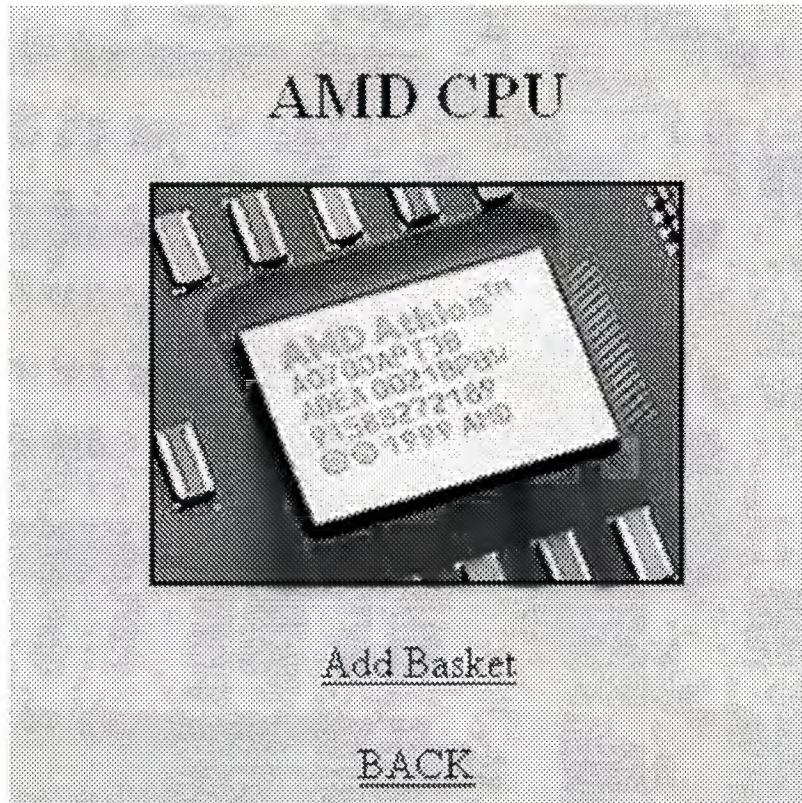
BACK

HOME PAGE

Picture 2.2.5.

When user come this page, they will see kinds of Cpu. And also they can select which one they want.

2.2.5.1. Select CPU



Picture 2.2.5.1.

After when user select parts, it is seeing parts Picture then user select if they want to buy this parts then click add basket button

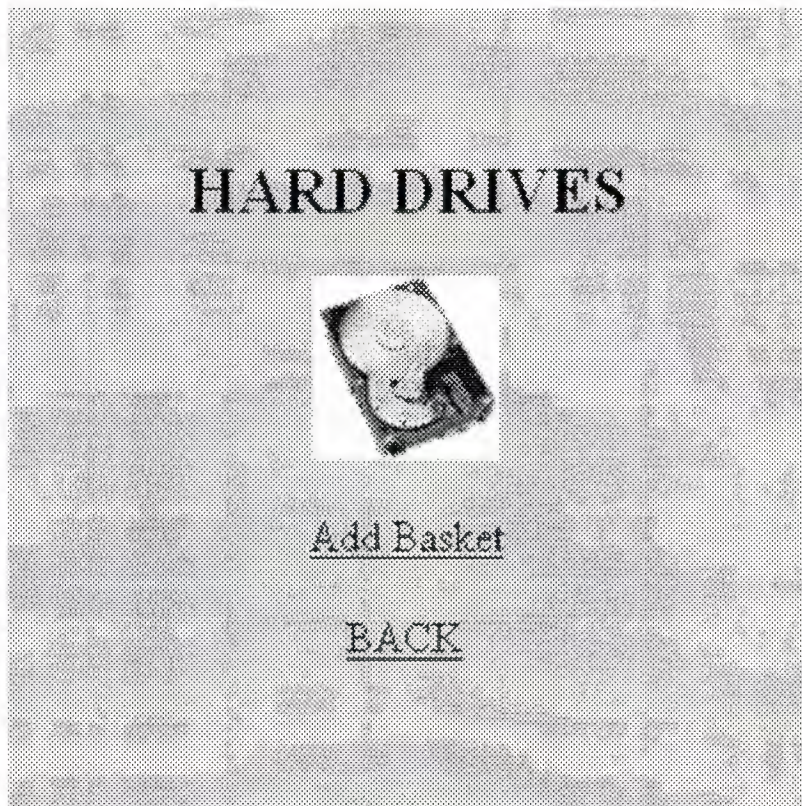
2.2.6. Hard Drives

Hard Drives		
Code	Computer Part Name	Price
401	20Gb Seagate/W.D/5400rpm ATA100	27
		\$
402	40Gb Seagate/W.D/5400rpm ATA100	28
		\$
403	60Gb Seagate/W.D/7200rpm ATA100	29
		\$
404	80Gb Seagate/W.D/7200rpm ATA100	30
		\$
405	100Gb Seagate/W.D/7200rpm ATA100	31
		\$
BACK		
HOME PAGE		

Picture 2.2.6.

When user come this page, they will see kinds of hard drives. And also they can select which one they want.

2.2.6.1. Select Hard Drives



Picture 2.2.6.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

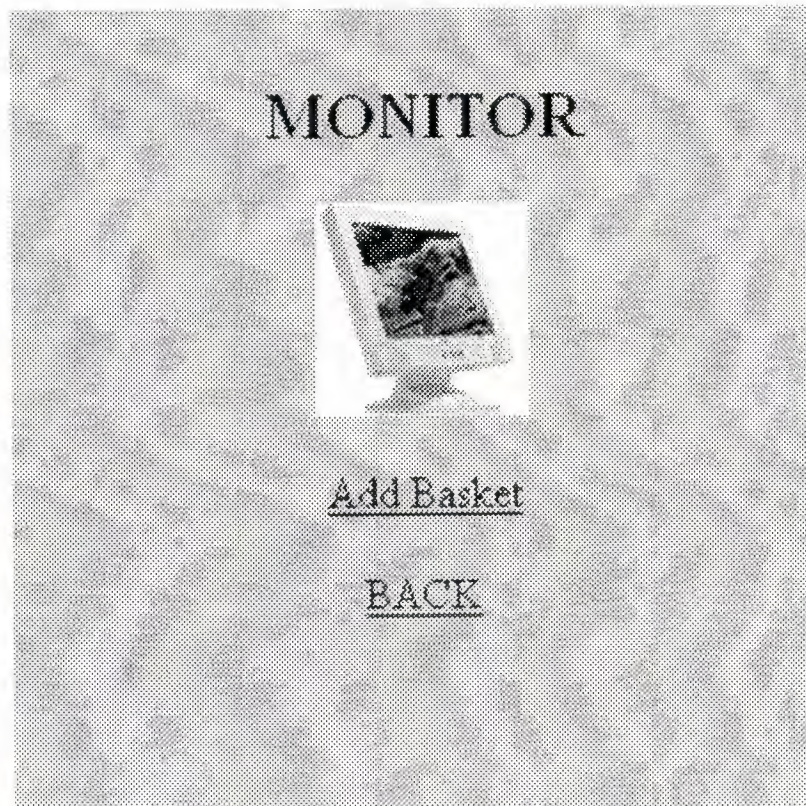
2.2.7. Monitors

Monitors		
Code	Computer Part Name	Price
501	15" Acer	32
		\$
502	17" Acer V771 (27)	33
		\$
503	17" Mitsubishi 1770G (27)	34
		\$
504	19" Proview	35
		\$
505	19" Hitachi CM721 Flat (21)	36
		\$
506	21 " Mitsubishi 2107 (26)	37
		\$

Picture 2.2.7.

When user come this page, they will see kinds of monitors. And also they can select which one they want.

2.2.7.1. Select Monitor



Picture 2.2.7.1.

After when user select parts, it is seeing parts Picture then user select if they want to buy this parts then click add basket button

2.2.8. CDROM Drivers

CDRom Drivers		
Code	Computer Part Name	Price
601	52X Samsung CDROM	38
		\$
602	52X Diamond Data	39
		\$
603	52X Sony CDEOM	40
		\$
604	52X Creative CDROM	41
		\$

[BACK](#)

[HOME PAGE](#)

Picture 2.2.8

When user come this page, they will see kinds of CDROM drivers. And also they can select which one they want

2.2.8.1. Select CDRom



Picture 2.2.8.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

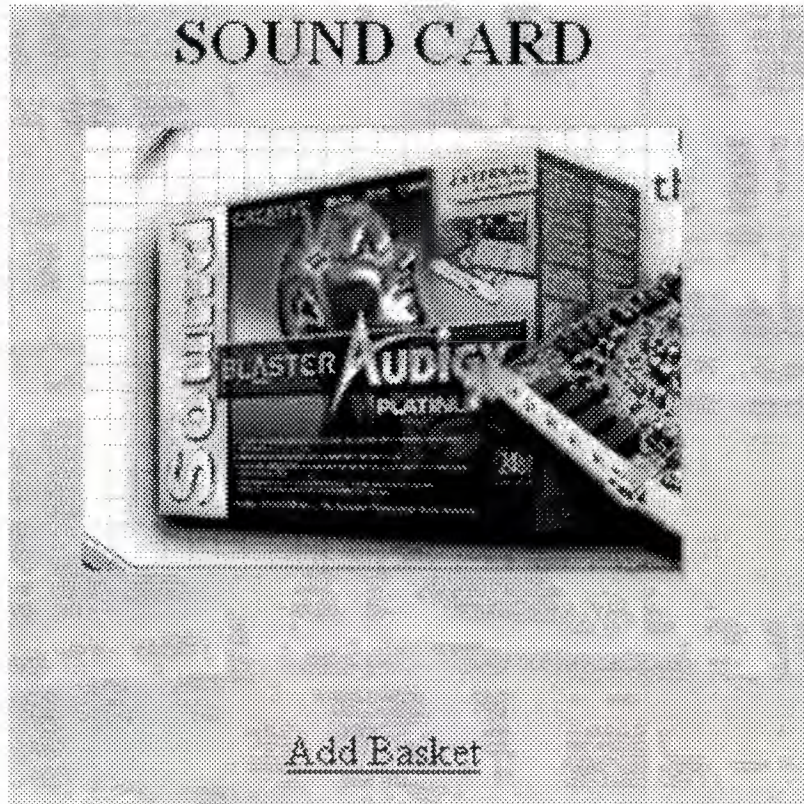
2.2.9. Sound Cards

Sound Cards		
Code	Computer Part Name	Price
701	<u>Creative Vibra 128 PCI with software</u>	42
		\$
702	<u>Creative S/B Live! Value! with software</u>	43
		\$
703	<u>Creative S/B Live D E S 1 Retail Box</u>	44
		\$
704	<u>Creative S/B Audigy 5.1 Digital</u>	45
		\$
705	<u>Creative S/B Audigy Platinum 5.1</u>	46
		\$
BACK		
HOME PAGE		

Picture 2.2.9

When user come this page, they will see kinds of sound cards. And also they can select which one they want.

2.2.9.1. Select Sound Card



Picture 2.2.9.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

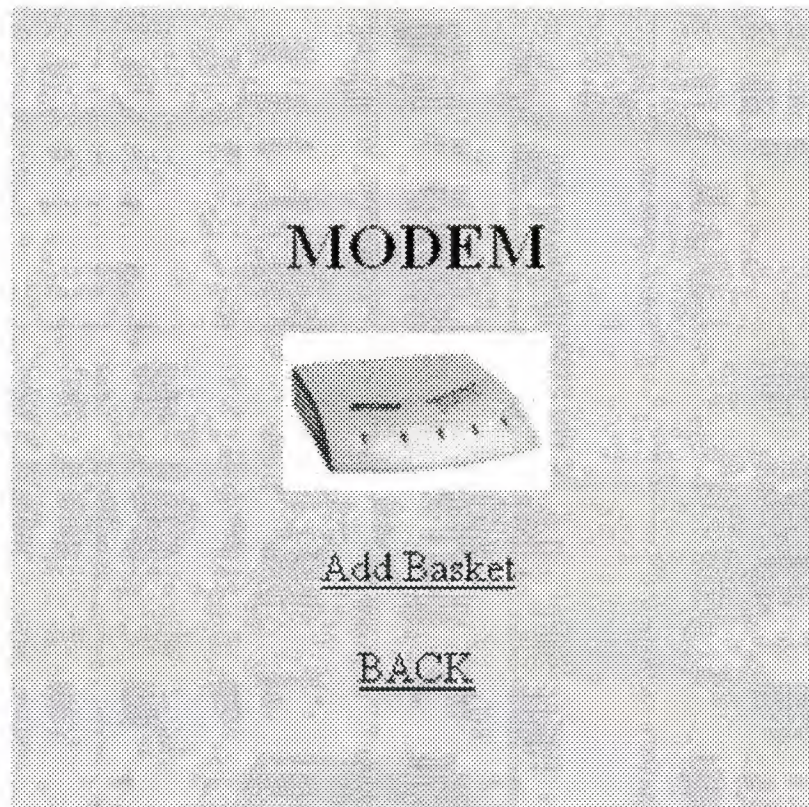
2.2.10. Modems

Modems		
Code	Computer Part Name	Price
801	56K Data/Fax/Voice Internal	47
		\$
802	56K Netcomm Internal	48
		\$
803	56K Data/Fax/Voice External	49
		\$
804	56K Netcomm Mega External	50
		\$
<div>BACK</div> <div>HOME PAGE</div>		

Picture 2.2.10.

When user come this page, they will see kinds of modems. And also they can select which one they want.

2.2.10.1. Select Modem



Picture 2.2.10.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

2.2.11. Keyboards

Keyboards		
Code	Computer Part Name	Price
901	<u>Keyboard PS/2/Serial</u>	51
		\$
902	<u>HP Multimedia Internet Keyboard</u>	52
		\$
903	<u>Microsoft Internet Keyboard & Mouse Retail</u>	53
		\$

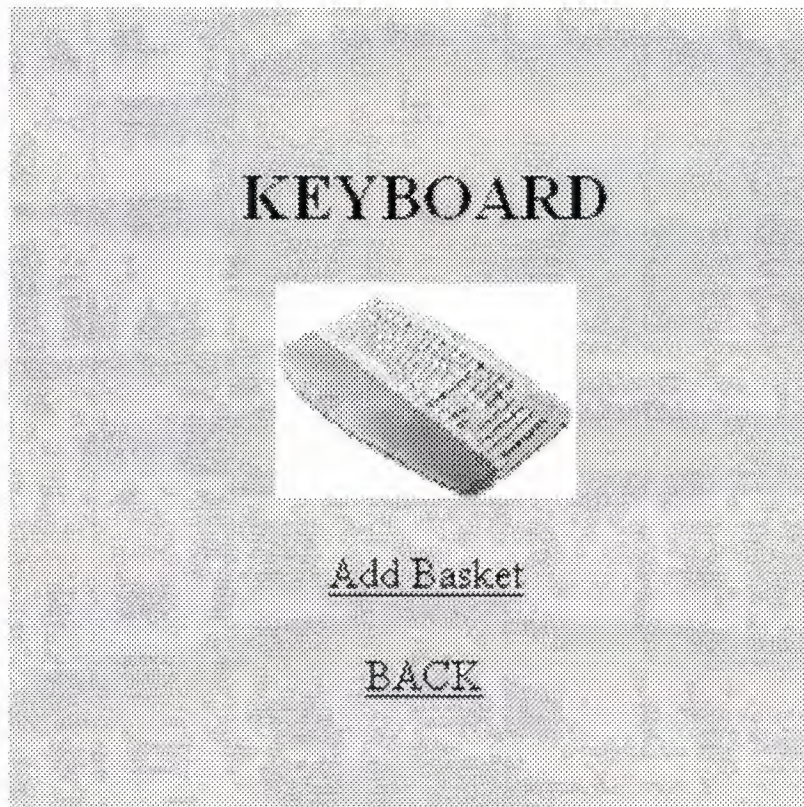
BACK

HOME PAGE

Picture 2.2.11.

When user come this page, they will see kinds of keyboards. And also they can select which one they want.

2.2.11.1. Select Keyboard



Picture 2.2.11.1.

After when user select parts, it is seing parts Picture then user select if they want to buy this parts then click add basket button.

2.2.12. Printers

Printers		
Code	Computer Part Name	Price
1001	<u>HP 640C</u>	54
		\$
1002	<u>HP 840C</u>	55
		\$
1003	<u>HP 1200 Laser</u>	56
		\$
1004	<u>Panasonic KXP5100 Laser</u>	57
		\$
1005	<u>Epson Photo 1290</u>	58
		\$

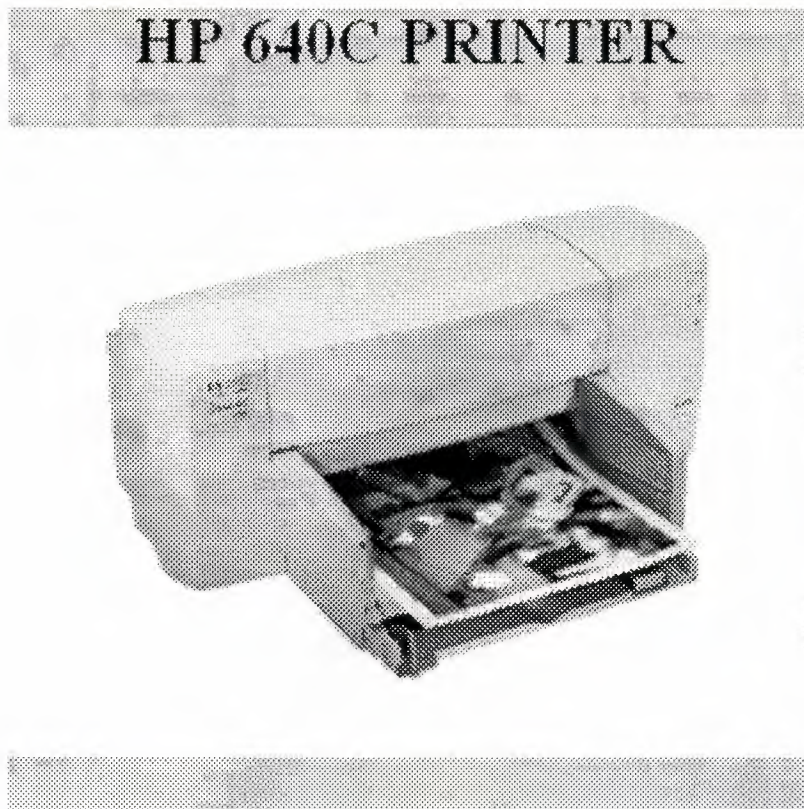
BACK

HOME PAGE

Picture 2.2.12.

When user come this page, they will see kinds of printers. And also they can select which one they want.

2.2.12.1. Select Printer



Picture 2.2.12.1.

After when user select parts, it is seeing parts Picture then user select if they want to buy this parts then click add basket button.

2.2.13. Scanners

Scanners		
Code	Computer Part Name	Price
1101	Canon I4640P	59
		\$
1102	Canon I4650U	60
		\$
1103	Epson Perfection 1250	61
		\$
1104	Epson 1250 Photo	62
		\$
1105	Epson 1650 Photo	63
		\$

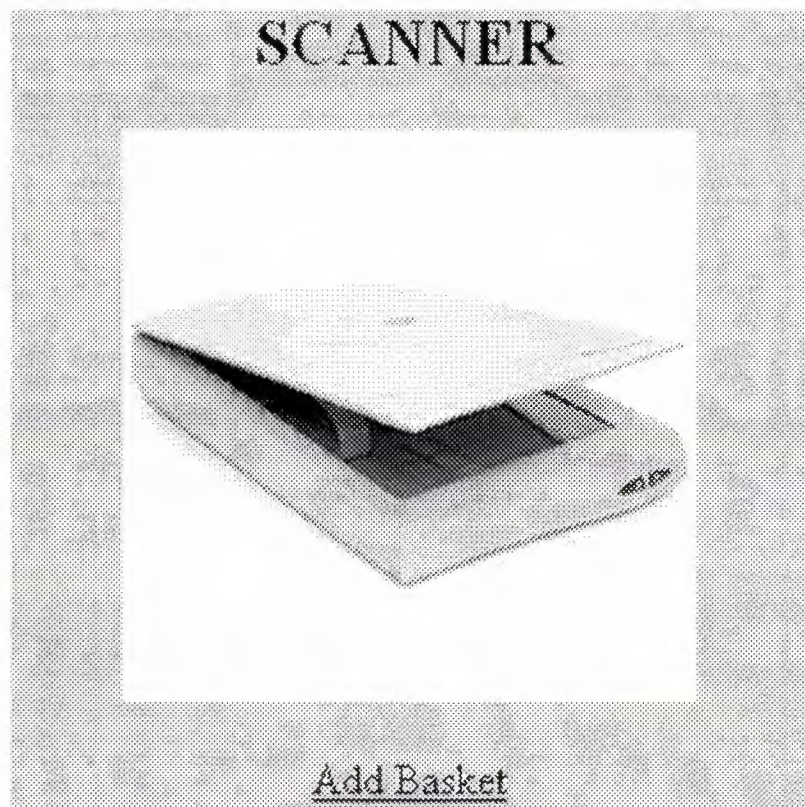
BACK

HOME PAGE

Picture 2.2.13.

When user come this page, they will see kinds of scanners. And also they can select which one they want.

2.2.13.1. Select Scanner



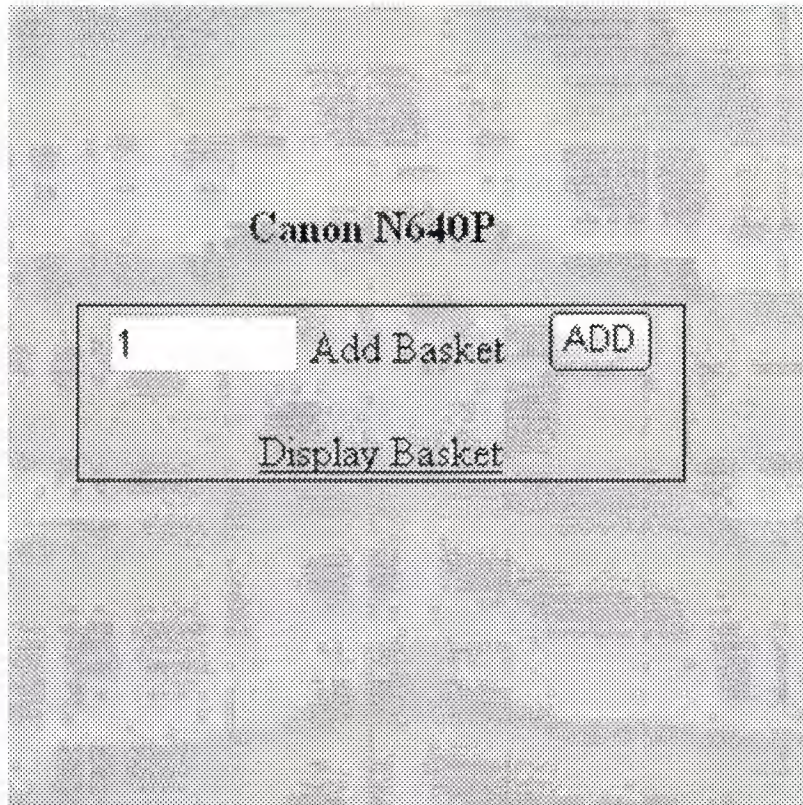
Picture 2.2.13.1.

After when user select parts, it is seeing parts Picture then user select if they want to buy this parts then click add basket button.

CHAPTER THREE

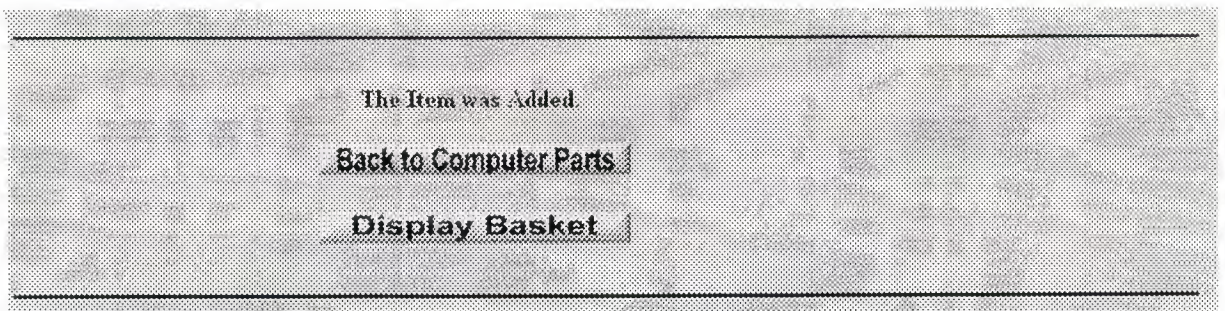
PURCHASING EVENT

3.1 Add Basket



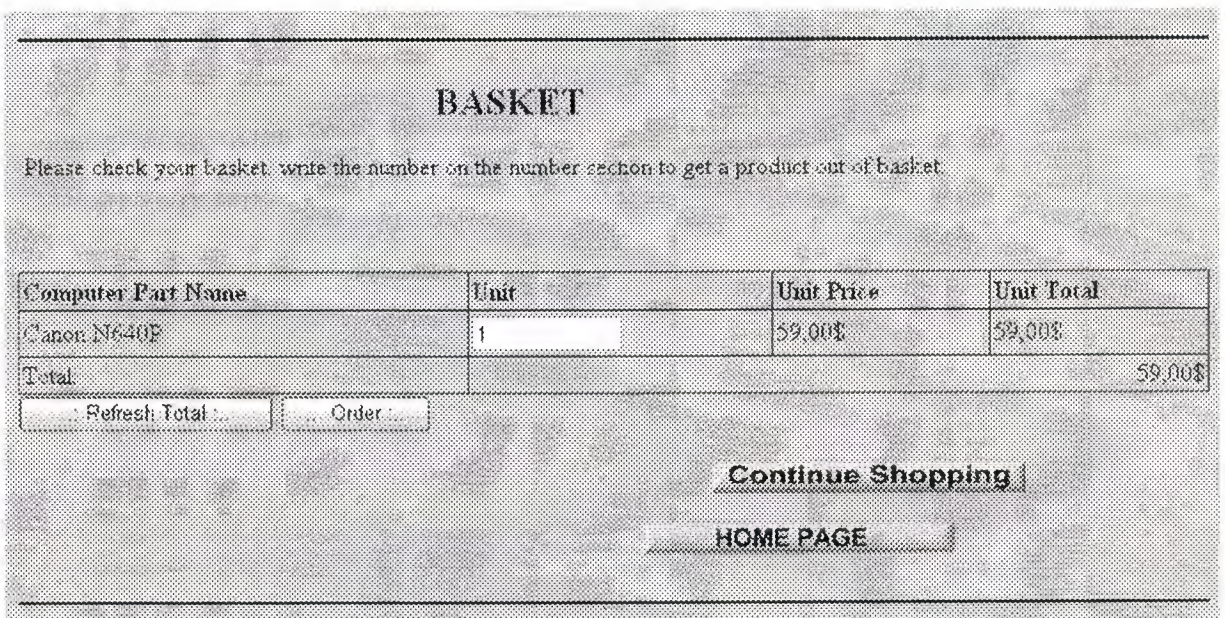
Picture 3.1

This section when user or members select parts and they want to buy this parts, they can see how many parts they chose. And adding to basket.



Picture 3.1.1

3.1.2 Basket Table



Picture 3.1.2.

Here again user they can see all operation which they can did and user can delete of any parts. Which they don't want. They can go back to the purchasing page.

3.1.3 Order Page

ORDER
[Change Order](#)

Code	Computer Part Name	Unit	Unit Price	Unit Total
1101	Canon N640P	1	\$59,00	\$59,00
Bottom Total				\$59,00
Transport (5%)				\$2,95
Total :				\$61,95

Order Info
Name :
Surname:
Address1:
Address2:
City :
Zip :
Code :
Telephone Number :

☐ MasterCard ☐ Visa ☐ American Express
Card Number :
On Card Name :

[BACK](#)

Picture 3.1.3.

When user finish all operations, they will enter order information. Example name, surname, card number...

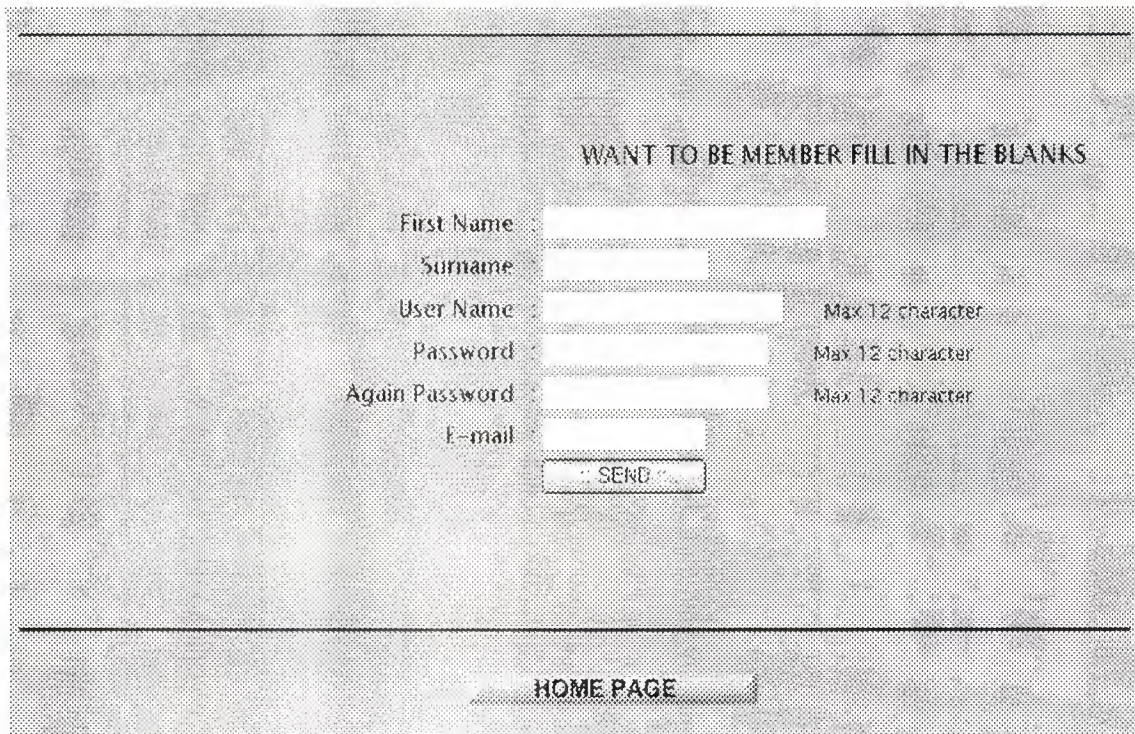


Picture 3.1.3.1.

CHAPTER FOUR

REGISTER EVENT

4.1. REGISTER FORM

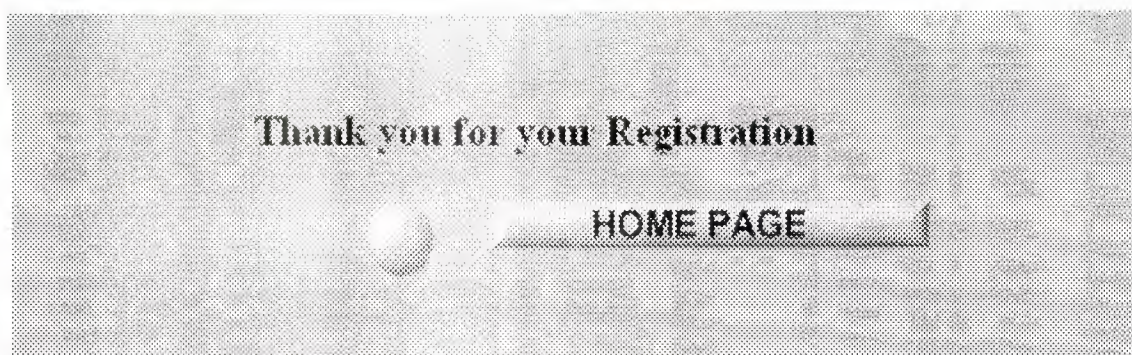


WANT TO BE MEMBER FILL IN THE BLANKS

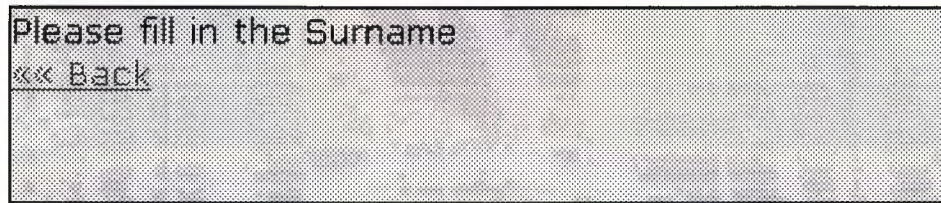
First Name	<input type="text"/>	
Surname	<input type="text"/>	
User Name	<input type="text"/>	Max 12 character
Password	<input type="text"/>	Max 12 character
Again Password	<input type="text"/>	Max 12 character
E-mail	<input type="text"/>	
<input type="button" value="SEND"/>		

Picture 4.1.

When user click register button they will write name surname password, email.
After click send button new user register in online shopping site.



Picture 4.1.1.



Picture 4.1.2.

if user can enter wrong informations it will come massage. I will shown the message box in picture.

CHAPTER FIVE

MEMBER REGISTER

5.1.MEMBER FORM

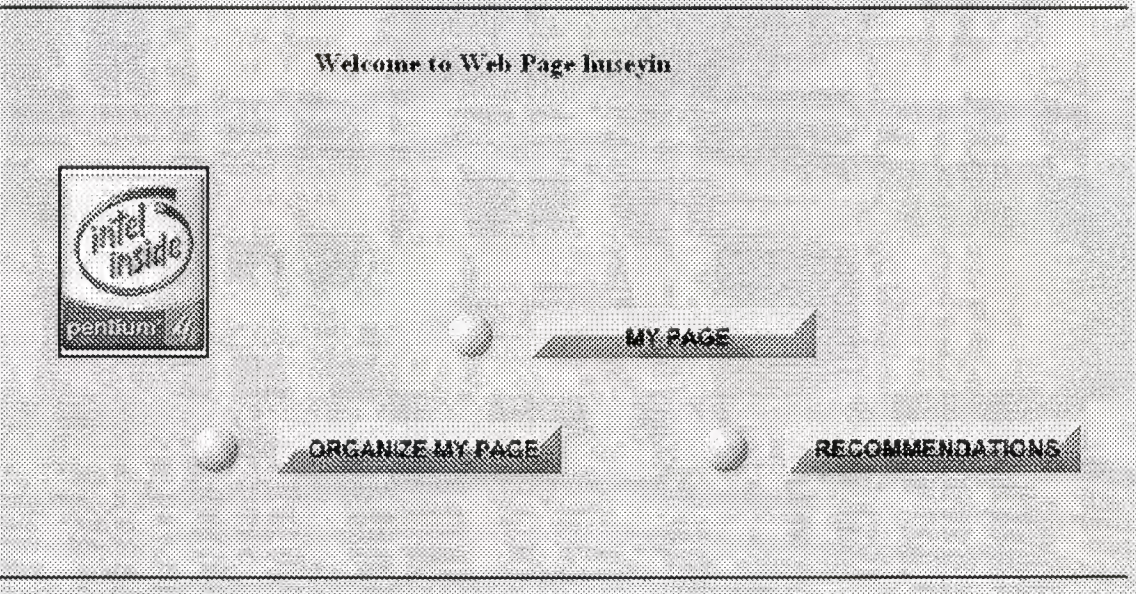
PLEASE ENTER USER NAME AND PASSWORD

User Name :	<input type="text"/>
Password :	<input type="password"/>
<input type="button" value="... SEND ..."/>	

Picture 5.1.

When user registering online shopping site the other days user can enter easily.

5.1.1. My page





Picture 5.1.1.

In this page seeing only member users.

5.1.1.2. Evaluation

Please specify your evaluations and recommendations about our page.

	Name: <input type="text"/>		
	Surname: <input type="text"/>		
	Recommendations: <input type="text"/>		
	<input type="button" value="SEND"/>		

Please specify your evaluations and recommendations about our page.

Previous Ideas and Recommendations

HUSEYİN POLAT
15.06.2003
ÇOK GÜZEL BİRSAYFA

Picture 5.1.1.2.

In this page users write ideas about the page.

CONCLUSION

This Project obtain me to increase we knowledge about ASP and thinking complexly with solutions to some specific and complex cases of the information flow when writing and developing program on the computers as software. Before we have learned programming in traditional languages, We was writing lots of codes and procedures to develop similar program that includes data functions. This type of programming languages obligates the programmer to write lots of codes.

In the ASP language these are allowed by program so it is very easy to compile and execute the program.

These kinds of languages, makes the Active Server Pages, Pronto Page, Gift Movie Gear, Photo Impact vs. programs to use friendly.

REFERENCES

1. ASP (First Edition) by Türkmen Kitabevi (İhsan KARAGÜLLE).
2. NEU Library
3. Com 340 Lecture codes.
4. www.aspogreniyorum.com
5. www.hepsiburada.com
6. www.bilgisayardershanesi.com.tr
7. www.turkmenkitabevi.com
8. www.indirin.com

APPENDIX

CODES

```
<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title> .....: POLATCOMPUTER | Online Shop
::.....</title>
</head>
<% @ LANGUAGE=VBScript CODEPAGE=1254 %>
<style fprolloverstyle>A: hover {font-weight: bold}
</style>
</head>
</head>
<body bgcolor="#CCCCCC" topmargin="10"
leftmargin="10" link="#000000" vlink="#000000"
alink="#000000">
<div align="left">
<table border="2" width="758" height="1"
id="AutoNumber1" bordercolorlight="#000000" cellspacing="1"
bordercolor="#000000">
<tr>
<td width="211" height="60" colspan="2"><p
align="center"></p>
</td> </tr> <tr>
<td width="208" height="1">
<palign="center"><font
size="7">ONLINE</font></p>
```


[illegible]

```

align="right"></p><body bgcolor=#CCCCCC link="#000000"
vlink="#000000" alink="#000000" text="#000000"
topmargin="10" leftmargin="25">
<p align="right"></p>
<body topmargin="10" leftmargin="25">
<hr color="#000000" width="758" align="left">
<imgborder="0" src="IMAGES/gorunum.gif"
width="350" height="350" align="left"></p>
<p>
&nbsp;&nbsp; 
<a href="default.asp"></a></p>
<p>&nbsp;&nbsp;  <a href="comp_kategorileri.asp"></a></p>
<p><imgborder="0"
src="benim_sayfam/IMAGES/NOKTA.gif" width="30"
height="30">&nbsp;&nbsp;  <a href="KAYIT.ASP"></a></p>
<p><imgborder="0"
src="benim_sayfam/IMAGES/NOKTA.gif" width="30"
height="30">&nbsp;&nbsp;  <a href="uye_giris.asp"></a></p>
<p align="left">
</p>
<p>

```



```

</p>
<p align="left">&nbsp;</p>
<p align="left">&nbsp;&nbsp;&nbsp;(800x600)</p>
<hr color="#000000" width="758" align="left">
<p>webmaster</p>
<p><a href="mailto:belongtohuso@yahoo.com">mailto:b
elongtohuso@yahoo.com</a></p>

```

```

<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: Members Login :.....</title>
</head>
<body link="#000000" vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC"
topmargin="0" leftmargin="0">
<% if Request.QueryString("donus")="sifre" then%>
<body link="#000000" vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC"
topmargin="0" leftmargin="0">
<h4 align="center"> <font color="#FF0000"><span
style="font-weight: 400">
<font face="Lucida Sans Unicode">Wrong
Password</font></span></h4>
<% Elseif
Request.QueryString("donus")="kullanici_ismi" then%>
<h4 align="center"> <font color="#FF0000"><span
style="font-weight: 400">
<body link="#000000" vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC"
topmargin="0" leftmargin="0">

```

```
<font face="Lucida Sans Unicode">Wrong User
Name</font></span></font></h4>
```

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

```
<table border="0" cellpadding="0" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111"
width="100%" id="AutoNumber3">
```

```
<tr>
```

```
<td width="100%">
```

```
&nbsp;</td>
```

```
</tr>
```

```
</table>
```

```
<% end if %>
```

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

```
<B>
```

```
<hr color="#000000">
```

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

```
<table border="0" cellpadding="0" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111"
width="100%" id="AutoNumber2" height="1">
```

```
<tr>
```

```
<td width="100%" height="1" colspan="3"></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="30%" height="20">&nbsp;</td>
```

```
<td width="43%" height="20">
```

```
<font face="Lucida Sans Unicode" style="font-size:
13pt">PLEASE ENTER USER NAME AND
PASSWORD</font></td>
```

```
<td width="27%" height="20">&nbsp;</td>
```

```
</tr>
```

```
</table>
```

```
<p style="margin-top: 0; margin-bottom: 0">&nbsp;</p>
```

```
<table border="0" cellpadding="0" cellspacing="0"
style="border-collapse: collapse" bordercolor="#111111"
```



```
width="100%" id="AutoNumber1" bordercolordark="#000080"
bordercolorlight="#000080">
```

```
<tr>
```

```
<td width="33%">&nbsp;</td>
```

```
<td width="33%">
```

```
<TABLE border=1 cellpadding=2 cellspacing=0
bgcolor=#CCCCCC width="298" style="border-collapse:
collapse" bordercolor="#000000" >
```

```
<TR>
```

```
<TD align="right" width="155">
```

```
<p align="center"><font face="Lucida Sans Unicode"
style="font-size: 11pt">User Name :</font></TD>
```

```
<B>
```

```
<TD width="130" bgcolor="#CCCCCC">
```

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

```
<INPUT maxlength=12 name=kullanici_ad
size=22></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD align="right" width="155">
```

```
<p align="center"><font face="Lucida Sans Unicode"
style="font-size: 11pt">Password<B>
:</font></TD>
```

```
<TD width="130">
```

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

```
&nbsp;<!--webbot bot="Validation" b-value-
required="TRUE" i-minimum-length="4" i-maximum-
length="12" --><INPUT type=password maxlength=12
name=sifre size=22></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD width="155">
```

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

```
<TD width="130">
```

```

        <p align="center">
            <INPUT type=submit value="... SEND ..."
style="color: #000080; font-family: Lucida Sans
Unicode"></TD>
        </TR>
    </TABLE>
</td>
<td width="34%">
    <p align="center">&nbsp;</td>
</tr>
</table>
<p align="center" style="margin-top: 0; margin-bottom:
0">&nbsp;</p>
<p align="center" style="margin-top: 0; margin-bottom:
0">&nbsp;</p>
<p align="center" style="margin-top: 0; margin-bottom:
0">&nbsp;</p>
<hr color="#000000">
</FORM>
</b>
</BODY>
</HTML>

```

```

<html>
<head>
    <meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
    <meta http-equiv="Content-Type"
content="text/html; charset=windows-1254">
    <title> ..... Welcome Register .....</title>
</head>
<br><br>
<% @ LANGUAGE=VBScript CODEPAGE=1254 %>

```



```

<%
ad=Request.Form("ad")
soyad=Request.Form("soyad")
kullanici_ismi=Request.Form("kullanici_ad")
sifre=Request.Form("sifre")
sifre2=Request.Form("sifre2")
email=Request.Form("email")
%>
<%
if ad="" then
hata = "Please fill in the Name"
TEKRAR
Response.End
End if
if soyad="" then
hata = "Please fill in the Surname"
TEKRAR
Response.End
End if
if kullanici_ismi="" then
hata = "Please fill in the User Name"
TEKRAR
Response.End
End if
if sifre="" then
hata = "Please fill in the Password"
TEKRAR
Response.End
End if
if sifre2="" then
hata = "Please fill in the Again Password"
TEKRAR
Response.End
End if

```



```

end if
else%>
<h3 align="left"> The passwords you entered are different
from each other, Please try again</h3>
<p align="left"><b></b><a href="KAYIT.ASP"></a></p>
<%end if%>
<% SUB TEKRAR %>
<br><br>
<div align="left">
<table width="20" border="1" cellpadding="0"
cellpadding="0" bgcolor="#C0C0C0" bordercolor="#000000"
style="border-collapse: collapse">
<tr>
<td width="350">
<table width="350" border="0" cellpadding="2">
<tr>
<font face="Verdana, Arial, Helvetica, sans-serif">
<td width="350" bgcolor="#888888" align="left"
height="100%" valign="top"><font
size="2"><%=hata%><br><a
href="javascript:history.back()">«« Back </a></font></td>
</tr>
</table>
</div>
<p><br></p>
</div>
<% END SUB %><body link="#000000"
vlink="#000000" alink="#000000"
bgcolor="#C0C0C0"></html>
<%uye_isim=Request.Form("kullanici_ad")
sifre=Request.Form("sifre")

```

```

set
veritabani=Server.CreateObject("ADODB.Connection")
veritabani.open "computer"
sorgu="select      *      from      uyeler      where
kullanici_ismi='"&uye_isim&'"
set uye_ismi_seti=veritabani.Execute(sorgu)
if uye_ismi_seti.EOF then
Response.Redirect "uye_giris.asp?donus=kullanici_ismi"
else
gercek_sifre=trim(uye_ismi_seti("sifre"))
if sifre=gercek_sifre then
Response.Cookies("giris")("kullanici_ismi")=uye_isim
else
Response.Redirect "uye_giris.asp?donus=sifre"
end if
end if
%>

```

```

<html>
<head>
<meta      http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta      http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: My Page :.....</title>
</head>
<body      link="#000000"      vlink="#000000"
alink="#000000"      topmargin="10"      leftmargin="25"
bgcolor="#C0C0C0">
<br><br><br>
<%if      Request.Cookies("giris")("kullanici_ismi")=""
then%>

```

Özür dileriz, bu sayfaya girebilmek için üye olmanız gereklidir.

[illegible]


```

        <body          link="#000000"          vlink="#000000"
alink="#000000"      bgcolor="#CCCCCC"      topmargin="10"
leftmargin="25">
        <B>
        <hr color="#000000" align="left" width="758">
        </b>
        <br><br><br>
        </head><%          @          LANGUAGE=VBScript
CODEPAGE=1254 %>
        <%
        if Request.Cookies("giris")("kullanici_ismi")="" then%>
        &nbsp;<p align="center">Sorry ! .
        <a      href=" ../kayit.asp">If      you      want      to
member</a><br><br><br>
        Already member <a href=" ../uye_giris.asp">For login
page </a>click here.
        <%else%> You want to display item.<br>
        </p>
        <FORM          ACTION="sayfami_duzenle_islem.asp"
method="post">
        <%
        kullanici_ad=Request.Cookies("giris")("kullanici_ismi")
        set
        veritabani=Server.CreateObject("ADODB.Connection")
        veritabani.open "computer"
        set ziyaretc_i_say=veritabani.Execute("select * from
ziyaretc_i_sayfalari where kullanici_ismi=""&kullanici_ad&""
order by kategori_id")
        if ziyaretc_i_say.EOF then
        ilk="insert                                into
ziyaretc_i_sayfalari(kategori_id,kullanici_ismi)          values
(0,"&kullanici_ad&"" )
        veritabani.Execute (ilk)
        end if

```

```

        set ziyaretc_i_say=veritabani.Execute("select * from
ziyaretc_i_sayfalari where kullanici_ismi='"&kullanici_ad&"
order by kategori_id")

        set kategori=veritabani.Execute("select * from kategori
order by kategori_id")

        ziyaretc_i_say.movenext
        while not kategori.EOF
        %>
        <INPUT TYPE="checkbox" NAME="kategori"
VALUE="<%=kategori("kategori_id")%>"
        <% if not ziyaretc_i_say.EOF then
        if kategori("kategori_id") = ziyaretc_i_say("kategori_id")
then%>
                CHECKED
        <%
        ziyaretc_i_say.MoveNext
        end if
        end if
        %>
        >
        <%=kategori("kategori_ismi")%><br>
        <%kategori.MoveNext
        wend
        set onay=veritabani.Execute("SELECT * from
ziyaretc_i_sayfalari where kullanici_ismi='"&kullanici_ad&" and
kategori_id=0")
        %>
        <br>
        <input type=submit value="... Display ..." style="color:
#000080">
        </form>
        <center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
        <%

```



```
kategori.close  
set kategori=nothing  
onay.close  
set onay=nothing  
veritabani.Close  
set veritabani=nothing  
end if  
%>  
<B>  
  
<hr color="#000000" align="left" width="758">  
    </BODY>  
</HTML>
```

```
<%  
if Request.Cookies("gis")("kullanici_ismi")="" then%>  
    <body          link="#000000"          vlink="#000000"  
alink="#000000" bgcolor="#CCCCCC">  
        <a  
href="../KAYIT.ASP">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;p;  
  
        </a>; <br>  
        Already Member <a href="../uye_giris.asp">Login Web  
Page </a>  
  
    <%else  
        adresler=Request.Form("adres_defteri")  
        yeni_comp=Request.Form("yeni_complar")  
        linkler=Request.Form("linklerim")  
        kullanici_ad=Request.Cookies("gis")("kullanici_ismi")  
        set  
        veritabani=Server.CreateObject("ADODB.Connection")  
            veritabani.open "computer"  
            veritabani.Execute ("delete from ziyaretci_sayfalari where  
kullanici ismi='"&kullanici_ad&"")
```

```

set kayit_set=Server.CreateObject("ADODB.Recordset")
kayit_set.Open "ziyaretci_sayfalari" ,veritabani,3,3
for each kategori in Request.Form("kategori")
kayit_set.AddNew
kayit_set("kullanici_ismi")=kullanici_ad
kayit_set("kategori_id")=kategori
kayit_set.Update
next
kayit_set.AddNew
kayit_set("kullanici_ismi")=kullanici_ad
kayit_set.Update
veritabani.Close
set veritabani=nothing
end if
Response.Redirect "sayfam.asp"
%>

```

```

<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....&nbsp; POLAT COMPUTER :.....</title>
</head>
<br><br><br><br><br>
<body link="#000000" vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">
<p></p>
<p></p>
<p></p>
<p align="center"><b><font
size="5">MOTHERBOARD</font></b></p>
</BODY>

```



```

        align="center"></p>
        <center><!--#include          virtual="../../satis/sepet.inc"--
></center>
        <center><!--#include
virtual="../../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>
<head>
<meta          http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta          http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....&nbsp; POLAT COMPUTER :.....</title>
</head>
<br><br><br><br><br>
<body          link="#000000"          vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">
<p          align="center"><font
size="5"><b>CDRom</b></font></p>
</BODY>
<p          align="center"></p>
<center><!--#include          virtual="../../satis/sepet.inc"--
></center>
<center><!--#include
virtual="../../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

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

```

```

        <meta                                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
        <title>.....: POLAT COMPUTER:.....</title>
</head>
<B>
<hr color="#000000" align="left" width="758">
        </b>
<br><br><br><br><br>
        <body                                link="#000000"                vlink="#000000"
alink="#000000"                text="#000000"                bgcolor="#CCCCCC"
topmargin="0" leftmargin="0">
        <p></p>
        <p></p>
        <p align="center"><font                size="5"><b>AMD
CPU</b></font></p>
        </BODY>
        <p align="center"></p>
        <center><!--#include                virtual="../satis/sepet.inc"--
></center>
        <center><!--#include
virtual="../comp_sayfalari/back.inc"--></center>
        <B>
        <hr color="#000000" align="left" width="758">

</HTML>

```

```

<html>
<head>
        <meta                                http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
        <meta                                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
        <title>.....:&nbsp; POLAT COMPUTER:.....</title>

```



```

</head>
<br><br><br><br><br>
<body      link="#000000"      vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">
<p></p>
<p></p>
<p      align="center"><font      size="5"><b>INTEL
CPU</b></font></p>
</BODY>
<p      align="center"></p>
<center><!--#include      virtual=" ../satis/sepet.inc"--
></center>
<center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>
<head>
<meta      http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta      http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: POLAT COMPUTER :.....</title>
</head>
<br><br><br><br><br>
<body      link="#000000"      vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC">
<p></p><p></p>
<p      align="center"><b><font      size="5">VIDEO
CARD</font></b></p>
</BODY>
<p      align="center"></p>

```

```

        <center><!--#include          virtual=" ../satis/sepet.inc"--
></center>

        <center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
        </HTML>

```

```

<html>
<head>
        <meta          http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
        <meta          http-equiv="Content-Type"
content="text/html; charset=windows-1254">
        <title>.....&nbsp; POLAT COMPUTER ::.....</title>
</head>
<br><br><br><br><br>
        <body          link="#000000"          vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">
        <p></p>
        <p></p>
        <p align="center"><font          size="5"><b>HARD
DRIVES</b></font></p>
        </BODY>
        <p align="center"></p>
        <center><!--#include          virtual=" ../satis/sepet.inc"--
></center>

        <center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
        </HTML>

```

```

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

```



```

        <meta                                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
        <title>.....&nbsp; POLAT COMPUTER:.....</title>
</head>
<br><br><br><br><br>
<body                                link="#000000"                                vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC">
<p></p>
<p></p>
<p                                align="center"><font
size="5"><b>KEYBOARD</b></font></p>
</BODY>
<p                                align="center"></p>
<center><!--#include                                virtual="../../satis/sepet.inc"--
></center>
<center><!--#include
virtual="../../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>
<head>
<meta                                http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta                                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....&nbsp; POLAT COMPUTER:.....</title>
</head>
<br><br><br><br><br>
<body                                link="#000000"                                vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC">
<p></p>
<p></p>

```

```

        align="center"><font
size="5"><b>KEYBOARD</b></font></p>
</BODY>

        align="center"></p>
        <center><!--#include          virtual=" ../satis/sepet.inc"--
></center>

        <center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>
<head>
        <meta          http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
        <meta          http-equiv="Content-Type"
content="text/html; charset=windows-1254">
        <title>.....: POLATCOMPUTER :.....</title>
</head>
        <br><br><br><br><br>
        <body          link="#000000"          vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC">
        <p></p>
        <p></p>
        <p          align="center"><font
size="5"><b>MODEM</b></font></p>
</BODY>
        align="center"></p>
        <center><!--#include          virtual=" ../satis/sepet.inc"--
></center>

        <center><!--#include
virtual=" ../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>

<head>

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

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

<title>.....:&nbsp; POLAT COMPUTER :.....</title>

</head>

<br><br><br><br><br>

<body                                link="#000000"                vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">

<p></p>

<p></p>

<p                                align="center"><font
size="5"><b>MONITOR</b></font></p>

</BODY>

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

<center><!--#include                virtual="../satis/sepet.inc"--
></center>

<center><!--#include
virtual="../comp_sayfalari/back.inc"--></center>

</HTML>

```

```

<html>

<head>

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

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

<title>.....:&nbsp; POLAT COMPUTER :.....</title>

```

```

</head>
<br><br><br><br><br>
<body      link="#000000"      vlink="#000000"
alink="#000000" text="#000000" bgcolor="#CCCCCC">
<p></p>
<p></p>
<p align="center"><font
size="5"><b>SCANNER</b></font></p>
</BODY>
<p align="center"></p>
<center><!--#include virtual="../satis/sepet.inc"--
></center>
<center><!--#include
virtual="../comp_sayfalari/back.inc"--></center>
</HTML>

```

```

<html>
<head>
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: POLAT COMPUTER::.....</title>
</head>
<br><br><br><br><br>
<body      link="#000000"      vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0">
<p></p>
<p></p>
<p align="center"><font size="5"><b>SOUND
CARD</b></font></p>
</BODY>

```



```

        align="center"></p>
</BODY>
<p align="center">&nbsp;</p>
<center><!--#include                virtual="../satis/sepet.inc"--
></center>
<center><!--#include
virtual="../comp_sayfalari/back.inc"--></center>
</HTML>
</HTML>

```

```

<%comp_id=Request.QueryString("comp_id")
Response.Write comp_id
%><body                link="#000000"                vlink="#000000"
alink="#000000"                topmargin="0"                leftmargin="0"
bgcolor="#C0C0C0">

```

```

<html>
<head>
<meta                http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: Add to the Basket :.....</title>
</head><%                @                LANGUAGE=VBScript
CODEPAGE=1254 %>
<br><br><br><br><br><br>
<body                link="#000000"                vlink="#000000"
alink="#000000" bgcolor="#C0C0C0"><center>
<%
comp_id=Request.QueryString("comp_id")
set
veritabani=Server.CreateObject("ADODB.Connection")
veritabani.open "computer"

```

```
comp=veritabani.Execute("select * from complar where
cstr(comp_id)='"+&comp_id&'")

Response.Write "<h4>&comp("comp_ismi")&" "
%>&nbsp;</center></p>
</h4>
<div align="center">
    <center>
        <TABLE   WIDTH="100%"   style="border-collapse:
collapse"          bordercolor="#111111"           cellpadding="0"
cellspacing="0">
            <TR><TD WIDTH="75%">
                <FORM       ACTION="sepete_ekle_islem.asp"
METHOD="post">
                    <div align="center">
                        <center>
                            <TABLE      border=1      bgcolor=#C0C0C0
style="border-collapse: collapse"      bordercolor="#111111"
width="228">
                                <TR><TD ALIGN="center" width="224">
                                    <INPUT TYPE="hidden" NAME="comp_id"
VALUE="<%=comp_id%>">
                                    <INPUT TYPE="text" SIZE="6" NAME="adet"
VALUE="1"      maxlength="10">      Add
Basket&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
                                    <INPUT TYPE="submit" VALUE="ADD">
                                    <P>
                                    <a              href="sepete_gozat.asp">Display
Basket</a>
                                </TD></TR>
                                </TABLE>
                                </center>
                                </div>
                                </FORM>
                            </TD></TR>
```



```

        </TABLE>
    </center>
</div>
</body></html>

```

```

<html>
<head>
<meta                                http-equiv="Content-Type"
content="text/html; charset=ISO-8859-9">
<meta                                http-equiv="Content-Type"
content="text/html; charset=windows-1254">
<title>.....: Add to the Basket :.....</title>
</head><%                                @                LANGUAGE=VBScript
CODEPAGE=1254 %>
<body                                link="#000000"                vlink="#000000"
alink="#000000" bgcolor="#C0C0C0">
<hr color="#000000" width="758" align="left">
<br>
<%
Sub YeniSiparis
    Application.Lock
    Session("siparis_id")=Application("siparis_id")
    s_siparis_id=ctr(Session("siparis_id"))
    Application("siparis_id")=Application("siparis_id")+1
    Application.Unlock
end sub
Sub SiparisEkle(d_siparis_id,d_comp_id,d_adet)
    SQLsorgu="insert                into                siparis_kalemleri
(siparis_id,comp_id,adet)                values                ("&d_siparis_id&",
"&d_comp_id&", "&d_adet&")"
    veritabani.Execute(SQLsorgu)
end sub
comp_id=Request.Form("comp_id")
adet=Request.Form("adet")

```

```

set                                veritabani                                =
Server.CreateObject("ADODB.Connection")
    veritabani.Open "computer"
        s_siparis_id=cstr(Session("siparis_id"))
        if s_siparis_id="" then
            YeniSiparis
        end if
        set siparis_tek=veritabani.Execute("select * from
siparis_kalemleri where cstr(siparis_id)='"&s_siparis_id&"' and
cstr(comp_id)='"&comp_id&"'")
        if not siparis_tek.EOF then%>
            This item has already put in to your basket. If you want to
            buy more from
            this item<br>
            Please <a href="sepete_gozat.asp"> after clicking
            here</a> identify the number you want to buy on the
            number section.<br>
            <a href=" ../comp_kategorileri.asp">Click for Computer
            Categories</a>
            <%else
            SiparisEkle s_siparis_id,comp_id,adet

            'veritabani.Execute ("insert into siparisler (siparis_id)
            values ("&s_siparis_id&")")
            %>
            <TABLE WIDTH="821">
            <TR>
            <TD ALIGN="center" width="578">
                <h4>The Item was Added.</h4>
                <P>

```



```
        &nbsp;<a href=" ../comp_kategorileri.asp"></a><P>
```

```
        &nbsp;<a href="sepete_gozat.asp"></a><P>
```

```
</TD>
```

```
<TD width="233">
```

```
<P>
```

```
</TD></TR>
```

```
</TABLE>
```

```
<% end if
```

```
veritabani.Close
```

```
set veritabani = Nothing
```

```
%>
```

```
<hr color="#000000" width="758" align="left">
```

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

```
<html>
```

```
<head>
```

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

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

```
<title>.....: Display Basket :.....</title>
```

```
</head><% @ LANGUAGE=VBScript
CODEPAGE=1254 %>
```

```
<body link="#000000" vlink="#000000"
alink="#000000" text="#000000" bgcolor="#C0C0C0"
topmargin="10" leftmargin="25">
```

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

```
<br>&nbsp;<B><hr color="#000000" align="left"
width="758">
```

```
</b>
```

```
<p>
```

```

<%
    set          veritabani          =
Server.CreateObject("ADODB.Connection")
    veritabani.Open "computer"
    if cstr(Session("siparis_id")) = "" then%>
        <body          link="#000000"          vlink="#000000"
alink="#000000"          text="#000000"          bgcolor="#C0C0C0"
topmargin="10" leftmargin="25">
            There is still no available order or your session
expired.</p>
        <p>
            &nbsp;<a href=" ../comp_kategorileri.asp">select an item
for your order.</a>
        <% else
            s_siparis_id = cstr(Session("siparis_id"))

            sqlText = "select complar.comp_id, comp_ismi, " _
                & "fiyat_TL,adet from complar, " _
                & "siparis_kalemleri where " _
                & "complar.comp_id = siparis_kalemleri.comp_id
" _
                & "and siparis_kalemleri.siparis_id = " &
s_siparis_id
            %>

            <H2 align="left">BASKET</H2>
            &nbsp;<p>Please check your basket. write the number
on the number section to get a product out of basket.
            <p>&nbsp;</p>
            <FORM          ACTION="sepete_gozat_islem.asp"
METHOD="post">
                <TABLE          border=1          WIDTH="758"
bgcolor=#C0C0C0          style="border-collapse: collapse"
bordercolor="#111111"          cellpadding="2"
bordercolorlight="#000000" bordercolordark="#000000">

```



```

<TR>
  <TH ALIGN="left">Computer Part Name</TH>
  <TH ALIGN="left">Unit</TH>
  <TH ALIGN="left">Unit Price</TH>
  <TH ALIGN="left">Unit Total</TH>
</TR>
<%
  set gozatSet = veritabani.Execute(sqlText)
  while not gozatSet.EOF
    d_comp_id = gozatSet("comp_id")
    d_comp_ismi = gozatSet("comp_ismi")
    d_fiyat = gozatSet("fiyat_TL")
    d_adet = gozatSet("adet")
    d_toplam = d_toplam + (d_adet * d_fiyat)
  %>

  <TR>
    <TD>
      <%=d_comp_ismi%>
      &nbsp;</TD>
      <TD>
        <INPUT NAME="quant">
        <%=d_comp_id%>
        TYPE="text"
        SIZE=10
        VALUE="<%=d_adet%>" maxlength="10">
        </TD>
        <TD align=ALIGN="right">
          <%= formatNumber(d_fiyat, 2)
          %>$
          </TD><TD align=ALIGN="right">
            <%= formatNumber((d_adet *
            d_fiyat)) %>$</TD>
          </TR>
        <%
          gozatSet.MoveNext
        wend
      gozatSet.Close

```

```

set gozatSet = Nothing

%>

<TR>
  <TD>Total:</TD>
  <TD COLSPAN="3" ALIGN="right">
    <%= formatNumber(d_toplam,2) %>$
  </TD>
</TR>

</TABLE>

<INPUT TYPE="submit" NAME="buton"
VALUE="...: Refresh Total ..." style="color: #000080">
<INPUT TYPE="submit" NAME="buton"
VALUE="...: Order ..." style="color: #000080"></FORM>

<p align="center"><a
href=" ../comp_kategorileri.asp">
  </a></p>

<%
  end if
  veritabani.Close
  set veritabani = Nothing
%>

<p>
  <a href=" ../default.asp"></a></p>
<B>
<hr color="#000000" align="left" width="758">
</b>
</BODY>
</HTML>

```

Database

complar				
comp_id	comp_ismi	kategori	fiyat_TL	sayfa_ismi
101	Asus TUSI-M S370	Main Board	10	asus1.asp
102	Asus CUSL2-C S370	Main Board	11	asus1.asp
103	Gigabyte 71XE4 Socket A	Main Board	12	asus1.asp
104	Gigabyte 7ZXE Socket A	Main Board	13	asus1.asp
105	Gigabyte 7ZXR-C Socket A	Main Board	14	asus1.asp
106	Asus A7V133 Socket A With Raid	Main Board	15	asus1.asp
201	32Mb Daytona G2 MX200 TV out	Video Card	16	ekran.asp
202	64Mb Daytona G2 MX200	Video Card	17	ekran.asp
203	64Mb Eagle Geforce 2 MX400-TV OUT	Video Card	18	ekran.asp
204	64Mb Asus V7100 MX400 TV Out	Video Card	19	ekran.asp
205	64Mb Powercolor GeForce3	Video Card	20	ekran.asp
301	AMD Duron 1000 OEM	AMD Cpu	21	cpu.asp
302	AMD Athlon XP 1600+ OEM	AMD Cpu	22	cpu.asp
303	AMD Athlon XP 1700+ OEM	AMD Cpu	23	cpu.asp
304	Intel Pentium 4 1.5 (S478)	Intel Cpu	24	cpu1.asp
305	Intel Pentium 4 1.6 (S478)	Intel Cpu	25	cpu1.asp
306	Intel Pentium 4 1.7 (S478)	Intel Cpu	26	cpu1.asp
401	20Gb Seagate/W.D/5400rpm ATA100	Hard Disk	27	harddisk.asp
402	40Gb Seagate/W.D/5400rpm ATA100	Hard Disk	28	harddisk.asp
403	60Gb Seagate/W.D/7200rpm ATA100	Hard Disk	29	harddisk.asp
404	80Gb Seagate/W.D/7200rpm ATA100	Hard Disk	30	harddisk.asp
405	100Gb Seagate/W.D/7200rpm ATA100	Hard Disk	31	harddisk.asp
501	15" Acer	Monitor	32	monitor.asp
502	17" Acer V771 (.27)	Monitor	33	monitor.asp
503	17" Mitsubishi 1770G (.27)	Monitor	34	monitor.asp
504	19" Proview	Monitor	35	monitor.asp
505	19" Hitachi CM721 Flat (.21)	Monitor	36	monitor.asp
506	21 " Mitsubishi 2107 (.26)	Monitor	37	monitor.asp
601	52X Samsung CDROM	CD-Rom	38	cdrom.asp
602	52X Diamond Data	CD-Rom	39	cdrom.asp
603	52X Sony CDROM	CD-Rom	40	cdrom.asp
604	52X Creative CDROM	CD-Rom	41	cdrom.asp
701	Creative Vibra 128 PCI with software	Sound Card	42	sound.asp
702	Creative S/B Live! Value! with software	Sound Card	43	sound.asp
703	Creative S/B Live D.E 5.1 Retail Box	Sound Card	44	sound.asp
704	Creative S/B Audigi 5.1 Digital	Sound Card	45	sound.asp
705	Creative S/B Audigi Platinum 5.1	Sound Card	46	sound.asp
801	56k Data/Fax/Voice Internal	Modem	47	modem.asp
802	56K Netcomm Internal	Modem	48	modem.asp
803	56K Data/Fax/Voice External	Modem	49	modem.asp
804	56K Netcomm Mega External	Modem	50	modem.asp
901	Keyboard PSII/Serial	Keyboard	51	keyboard.asp

complar				
comp_id	comp_ismi	kategori	fiyat_TL	sayfa_ismi
902	HP Multimedia Internet Keyboard	Keyboard	52	keyboard.asp
903	Microsoft Internet Keyboard & Mouse Retail	Keyboard	53	keyboard.asp
1001	HP 640C	Printer	54	printer.asp
1002	HP 840C	Printer	55	printer1.asp
1003	HP 1200 Laser	Printer	56	printer2.asp
1004	Panasonic KXP6100 Laser	Printer	57	printer3.asp
1005	Epson Photo 1290	Printer	58	printer4.asp
1101	Canon N640P	Scanner	59	scanner.asp
1102	Canon N65OU	Scanner	60	scanner.asp
1103	Epson Perfection 1250	Scanner	61	scanner.asp
1104	Epson 1250 Photo	Scanner	62	scanner.asp
1105	Epson 1650 Photo	Scanner	63	scanner.asp

kategori	
kategori_id	kategori_ismi
1	Main Boards
2	Video Adapters
3	Cpu
4	Hard Drives
5	Monitors
6	CDRom Drivers
7	Sound Cards
8	Modems
9	Keyboards
10	Printers
11	Scanners

siparis_kalemleri		
siparis_id	comp_id	adet
1	804	2
2	101	1
2	1101	1
3	301	1
3	101	1
3	704	1
4	201	1
4	101	1

siparis_kalemleri		
siparis_id	comp_id	adet
5	101	1
6	704	1
7	901	2

siparis_id	ad	soy_ad	adres1	adres2	sehir	posta_kodu
1	a.kenen	kosger	cumra		üçüyük	42310
6	ALIRIZA	KASABALI	ovakavağı kasal		konya	42310
0						
tel	nakliye	toplam	kk_tip	kk_num	kk_ustundeki_isim	
033288788555			Visa	123123132132	master	
033288788555			Visa	132132123	expres	

uyeler				
ad	soyad	kullanici_ismi	sifre	email
huseyin	POLAT	huso	12345	BELONGTOHUS@YAHOO.COM
ALIRIZA	KASABALI	ARIZA	123	ARIZA@YAHOO.COM
AHMET	KOSGER	AHMET	123	CALDI@YAHOO.COM
a.kenen	kosger	kel	1453	CALDI@YAHOO.COM
asdaas	Asd	asd	asd	asd
irfan	Atik	irfan	123	bn@com
cihan	Kaya	cihan	123	zx@com

ziyaretci_sayfalari	
kullanici_ismi	kategori_id
huso	0
huso	11
irfan	0
ali	0
cihan	0