Prelucrarea prin ASP a chestionarelor din paginile Web

Prof.dr. Floarea NASTASE
Catedra de Informatica Economica, A.S.E. Bucuresti

*Internet este un nou mediu de comunicare interactiv. Majoritatea site-urilor Internet, în special cele comerciale, utilizeaza pagini Web interactive, complexe. Aceste pagini, deseori descrise ca aplicati Web, includ programe de prelucrare, denumite scripturi, care preiau informatia din raspunsul utilizatorului. O aplicatie Web poate utiliza un form pentru a testa cerintele utilizatorilor, iar informatiile relevante vor fi stocate într-o baza de date si apoi se trimite un raspuns clientului. **ASP** (**A**ctive **S**erver **P**ages) este una din tehnologiile propuse de Microsoft, care poate fi utilizata pentru crearea paginilor Web dinamice si interactive sau pentru realizarea aplicatiilor tranzactionale securizate pe Web.*
***Cuvinte cheie**: ASP, HTML, CGI, ISAPI, VBScript*.

Comunicarea între browser si un server Web are loc cu ajutorul protocolului HTTP. În functie de tehnica folosita de server-ul HTTP, paginile HTML pot fi generate în mod static sau dinamic. Pentru a genera HTML în mod dinamic pot fi utilizate interfete de tip gateway, cum ar fi **CGI** (**C**ommon **G**ateway **I**nterface) sau **ISAPI** (**I**nternet **S**erver **A**pplication **P**rogramming **I**nterface).

Tehnica de tip ISAPI, propusa de Microsoft, foloseste biblioteci dinamice (DLL), fiind utilizata pe platformele Windows. Acest lucru face ca, în comparatie cu un program de tip CGI, cererea sa fie prelucrata mai rapid, fiindca timpul de procesare aditional, necesar schimbarii contextului între procese de catre sistemul de operare, nu mai este necesar (figura 1.).

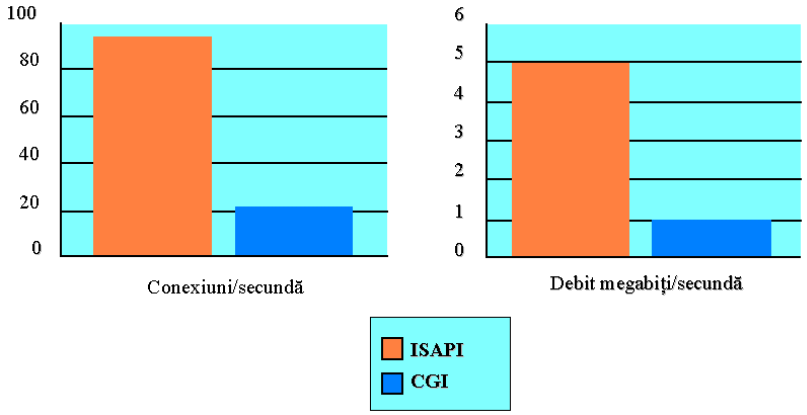

Fig.1. Performante ISAPI comparativ cu CGI (WebStone 1.1)

Prima tehnologie de tip ISAPI, propusa de Microsoft în 1995, a fost **IDC** (**I**nternet **D**atabase **C**onnector). Aceasta a permis interogarea unei baze de date si încorporarea rezultatului în pagina HTML, cu ajutorul unui limbaj scriptic simplu si a unui fisier de tip "*template*" HTML. La aceasta ora, tehnologia IDC este înca prezenta pe server-ele Web ale firmei Microsoft, dar începe sa prezinte un interes mai degraba istoric, pe piata aparând deja instrumente care permit migrarea de la aplicatiile de tip IDC catre ASP.

Active Server Pages este un mediu pentru editarea scripturilor localizate pe server. Tehnologia a fost propusa de Microsoft în 1996, având în prealabil numele de cod

Denali. ASP functioneaza pe urmatoarele server-e Web:

- *Microsoft Internet Information Server* (IIS) versiunea 3.0 sau mai mare, sub Windows NT Server 4.0,Windows 2000;
- *Microsoft Peer Web Services*, versiunea 3.0 sub Windows NT Workstation;
- *Microsoft Personal Web Server*, sub Windows 95/98 (PSW este o versiune mai restrânsa a lui IIS).

ASP-urile se utilizeaza pentru:

- Editarea dinamica, schimbarea sau adaugarea de continut la o pagina Web;
- Generarea de raspunsuri cererilor utilizatorilor sau informatiilor trimise prin formularele HTML;
- Accesarea oricarei date sau baze de date si returnarea de rezultate catre navigator;
- Personalizarea paginilor Web, facându-le mai utile pentru utilizatori particulari;
- Simplitatea si viteza de executie a ASP fata de CGI si Perl;
- Asigurarea securitatii, deoarece codul ASP nu este vizibil prin navigator;
- Posibilitatea de a fi afisate de orice navigator, fisierele ASP sunt returnate în format HTML;
- Minimizarea traficului prin retea.

La dezvoltarea unui site care utilizeaza ASP-uri pot interactiona aproape toate tehnologiile Web dinamice existente. În figura 2 este prezentat modul în care se integreaza diferitele componente de sistem cu aplicatiile.

Caracteristica majora a ASP consta în capacitatea sa de includere a scripturilor direct într-un fisier pe care-l acceseaza navigatorul, generând astfel *pagini dinamice*. Dar, fata de celelalte scripturi de pe server, fisierele ASP pot sa contina cod HTML sau XML, incluzând chiar scripturi pentru client si referiri la componentele COM (Component Object Model) prin care se realizeaza o serie de sarcini, cum ar fi conectarea la o baza de date.

Creatorii de pagini Web, folosind HTML si facilitatile de script din ASP, vor putea foarte usor sa realizeze pagini interactive, ca de exemplu preluarea si prelucrarea datelor dintr-un formular si înregistrarea lor într-o baza de date. Limbajul în care se scriu sripturile nu va crea probleme. Mediul ASP nu restrictioneaza folosirea unui anumit limbaj de script. ASP-ul înglobeaza compilatoare pentru limbajele VBScript, JScript, ActiveX, dar foarte usor se vor instala pentru PERL, REXX si Python .

Pentru programatorii de aplicatii Web în limbajele Vizual Basic, C++ sau Java, mediul ASP ofera modalitati extrem de simple pentru a încapsula componente ActiveX, pentru a crea module reutilizabile de program sau pentru a crea componente proprii. MTS (Microsoft Transaction Server), inclus in Windows NT Option Pack, reduce complexitatea si costul realizarii de aplicatii pentru server, aplicatii care au partea de securitate bine dezvoltata si implementata.

În mod obisnuit, când utilizatorul tasteaza un URL în zona de adresa a navigatorului sau când activeaza o legatura hipertext, o cerere este trimisa catre server. Pagina ceruta este reprezentata printr-un fisier stocat pe discul server-ului Web, iar acesta va începe sa o încarce în memorie. Daca este o pagina HTML statica server-ul adauga doar informatiile utilizate pentru transmisie, cum ar fi tipul documentului, codificarea pentru transmiterea prin HTTP si emite ansamblul catre navigator. Utilizatorul vede continutul paginii HTML, iar codul sursa corespunde continutului fisierului stocat pe discul de pe masina server.

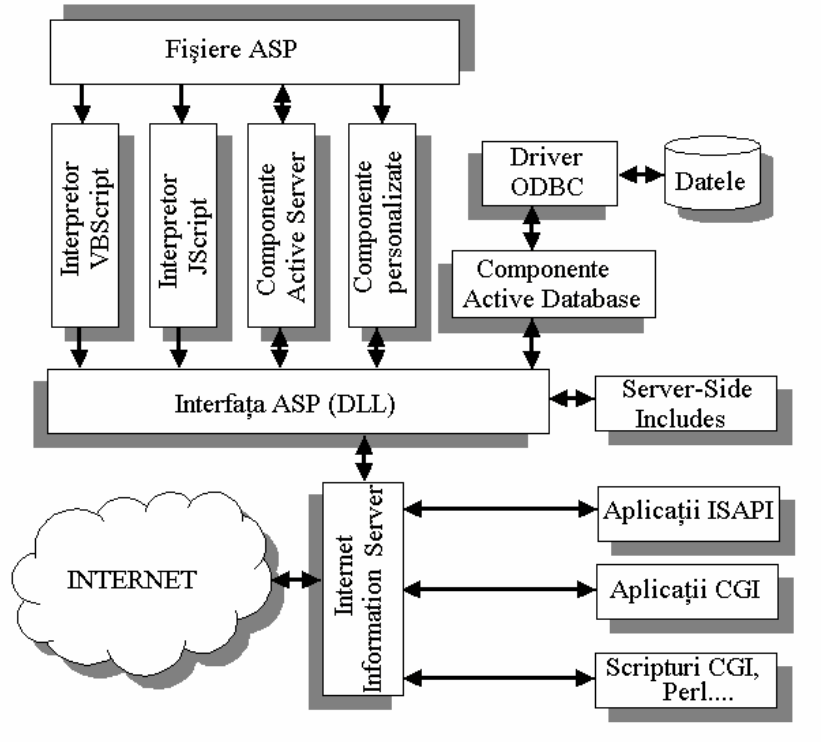

Fig.1. Integrarea componentelor de sistem cu aplictiile

În cazul în care este ceruta o pagina ASP, navigatorul are acces la aceasta într-un mod similar. Pagina este încarcata în memorie, ca o pagina statica banala, dar înainte de a fi trimisa catre navigator, server-ul verifica daca pagina contine scripturi care trebuie gestionate si executate. Aceste scripturi pot genera si insera valori în pagina, pot crea text sau cod HTML suplimentar, în functie de context. Odata terminata modificarea paginii, aceasta este emisa catre navigator prin HTTP. La destinatie, continutul fisierului este constituit din text si cod HTML, ca o pagina statica normala (figura 3). Acesta este o copie a fisierului de pe server, dar modificata conform rezultatelo r obtinute în urma executiei scripturilor.

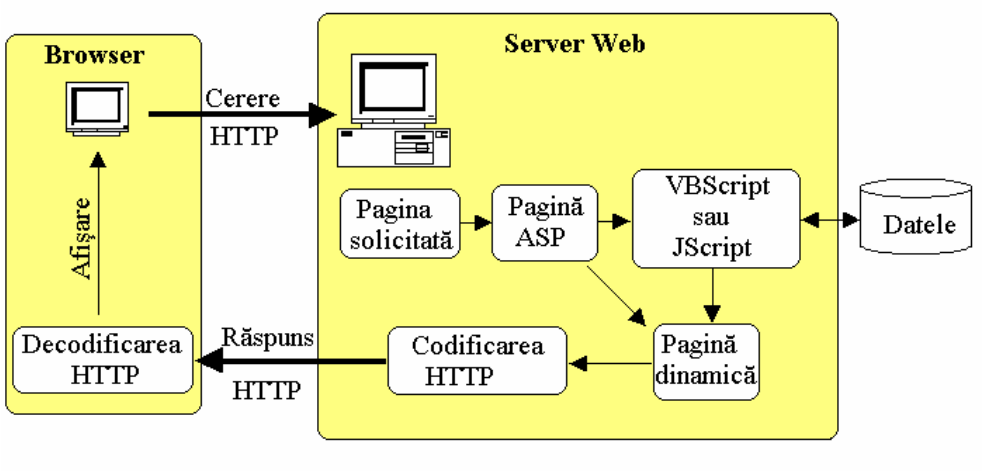

Fig.2. Apelarea unui fisier .asp

Deoarece scripturile sunt executate pe server, acesta realizeaza toate operatiile necesare generarii paginilor Web care sunt emise catre navigator. Scripturile de pe server nu pot fi copiate, deoarece numai rezultatele scriptului sunt transmise catre navigator.

Crearea paginilor ASP

Un fisier ASP este un fisier text care are extensia .asp si contine (figura 4):

- text;
- tag-uri HTML sau XML;
- comenzi ASP.

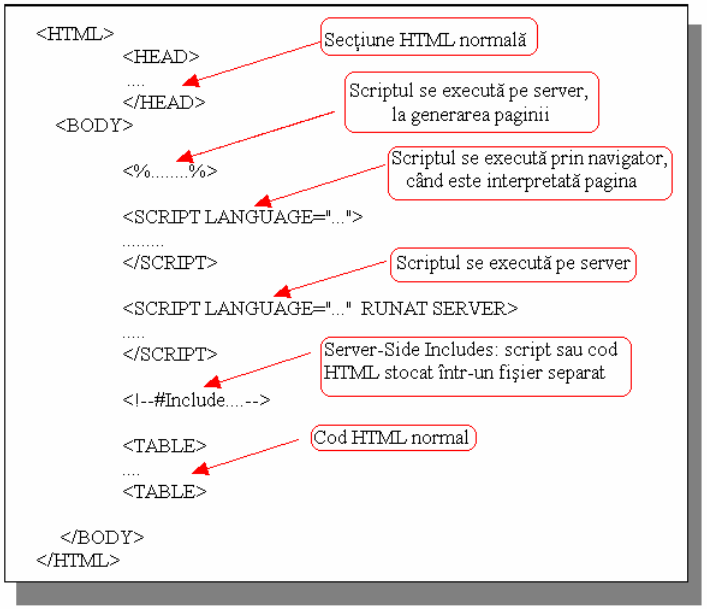

Fig.3. Continutul unui fisier .asp

Obiecte ASP predefinite

Pentru a utiliza marea majoritate a obiectelor existente, un program trebuie mai întâi sa creeze o *instanta* a acelui obiect. ASP furnizeaza cinci obiecte care nu necesita instantierea. Aceste obiecte sunt:

- Request furnizeaza informatia venita de la clientul HTTP care a initiat cererea;
- Response trimite informatie catre client;
- Server controleaza mediul de executie ASP;
- Session contine informatii legate de sesiunea de lucru cu server-ul Web, deschisa de un client;
- Application contine informatii globale despre aplicatia care poate fi accesate de catre mai multi clienti.

Scripturile acceseaza obiectele utilizând *metodele* si *proprietatile* lor. Sintaxa de accesare a obiectelor depinde de limbajul folosit, dar în general este de forma:

Obiect.Metoda parametri

sau

Obiect.Proprietate parametri

Cele cinci obiecte ASP predefinite, utilizabile în schimburile client-server sunt organizate ierarhic. În vîrf se situeaza obiectul Server. Metodele si proprietatile sale permit executia functiilor utilitare valabile pentru toate scripturile. Acest obiect reprezinta mediul în care se executa paginile. Celelalte obiecte Application, Session, Request si Response se integreaza pentru a realiza o aplicatie Active Server. Figura 5 prezinta modul de integrare a acestor obiecte.

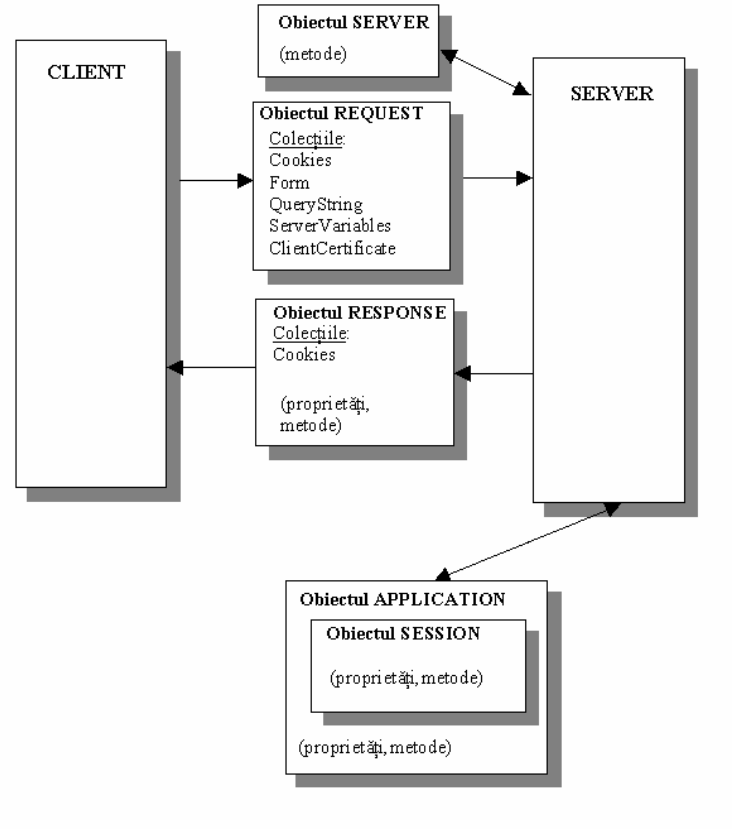

Fig.4. Obiectele ASP

Obiectul **Request** contine informatiile referitoare la cererile utilizatorului, cum ar fi tipul de browser folosit, informatia continuta într-un chestionar HTML trimisa catre server etc. Utilizând acest obiect, accesarea acestor informatii este foarte simpla. Interfata obiectului Request este sub forma a cinci *colectii* de variabile.

- QueryString - pastreaza valorile variabilelor în "query string";
- Form - pastreaza valorile din formular. Formularul trebuie sa utilizeze metoda POST;
- Cookies - pastreaza valorile pentru cookie;
- ServerVariables - pastreaza valorile variabilelor server;
- ClientCertificate – pastreaza valorile câmpurilor memorate în certificatul client, pentru a se asigura securitatea tranzactiilor.

Accesul la informatia continuta de acest obiect se face în general cu ajutorul sintaxei:

```
Request.collection
Request.property
```

```
Request.method
```

În mod frecvent se utilizeaza:

```
Request.NumeColectie(Variabila)
```

unde NumeColectie este numele uneia din cele cinci colectii indicate mai sus, iar Variabila este numele variabilei din colectie, care se doreste a fi accesata.

Obiectul Request ofera informatiile utilizabile în scripturi. Fiecare din proprietati este o colectie de valori de acelasi tip. Informatia continuta în obiectul Request provine de la client si este trimisa prin cererea HTTP catre server, care o decodifica si o face disponibila prin obiectul Request.

În afara informatiilor standard continute în antetul HTTP al cererii, navigatorul trimite informatii server-ului în alte doua moduri:

- printr-o sectiune <FORM>
- ca un sir de cereri adaugate la sfârsitul URL-ului.

Transmiterea informatiei dintr-un formular

Formularul HTML este în general cea mai utilizata metoda de trimitere a informatiei de la client catre server. Formularele

HTML contin elemente de interfata standard, grafice si text. Dupa completarea unui formular HTML de catre utilizator, trebuie sa se prelucreze informatia. În momentul în care clientul selecteaza butonul *Submit*, întreaga informatie din câmpurile chestionarului este trimisa la server. Aici, scripturile ASP pot avea acces la aceasta informatie, pentru a o analiza si manipula, utilizând numele elementelor din formular.
Folosind ASP-ul, întâlnim trei metode de baza pentru colectarea informatiei din formulare HTML:

- un fisier .htm (sau .html) static contine un formular care trimite valorile sale catre un fisier .asp (figura 6);
- un fisier .asp poate sa creeze un formular care sa trimita informatia catre un alt fisier .asp;
- un fisier .asp poate sa creeze un formular care sa trimita informatia chiar catre fisierul .asp care contine formularul.

Primele doua metode functioneaza în acelasi fel, un formular interactioneaza cu alte programe de pe server. Prin ASP, sarcina de colectare si prelucrare a informatiei din formular este simplificata considerabil. A treia metoda este frecvent utilizata.

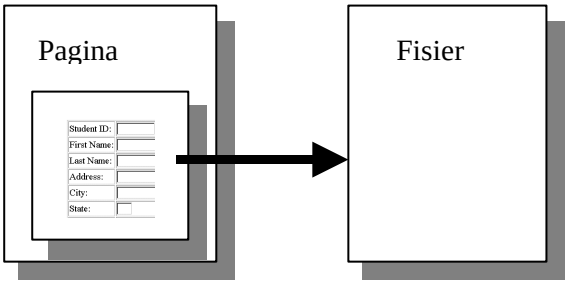

Fig.1. Fisierul .asp preia datele dintr-o pagina .html

Datele dintr-o sectiune <FORM> a unei pagini Web pot fi transferate catre server utilizând:

- numele fisierului .asp ca valoare pentru atributul ACTION;
- valoarea GET sau POST pentru atributul METHOD.

Daca se utilizeaza valoarea GET datele sunt adaugate URL-ului sub forma unui sir constituit din perechile nume_câmp/valoare (datele urmeaza URL-ului, separatorul fiind caracterul ?):

http://acasa/tutorial/test1.asp**?Data_Ord=12+mai+2000&Nr_Contr=50&Tip_Tranz=Cumparare&Cod_Titlu=actiune&Nr_Titlu=1000&Mod_Neg=Vedere&Tip_Curs=CEL+MAI+BUN&Comentariu=Vom+comunica+telefonic%0D%0A++X1&Term_Lim=12+dec+2001**

Acelasi efect se obtine prin utilizarea marcatorului <A>, sub forma:

<A HREF="nume_script.asp**?nume_câmp1=valoare1&nume_câmp2=valoare2…">text de legatura<A>**

Când se utilizeaza metoda GET, pentru transmiterea unei mari cantitati de informatie catre server, exista riscul de a se pierde o parte din date. Unele server-e au tendinta de a restrictiona lungimea maxima a datelor transferate în acest mod (la 1000 de caractere), iar cererile lungi vor fi trunchiate. Daca se doreste transmiterea de informatii în cantitate mai mare, atunci este recomandat sa se foloseasca metoda POST. În cazul în care se utilizeaza metoda POST pentru trimiterea datelor, acestea sunt incluse în antetul HTTP si nu mai sunt vizibile în zona de adresa a navigatorului. De aceasta data, valorile informatiilor nu mai sunt accesibile prin colectia QueryString, dar ASP pune la dispozitie colectia Form, care contine datele trimise prin metoda POST.
Indiferent de metoda utilizata, datele sunt receptionate sub forma unui sir de caractere ce trebuie decodificat. În acest sir de caractere spatiile sunt înlocuite cu (+), iar

diferitele caractere speciale prin (%) urmat de codul lor în hexadecimal. În plus, numele variabilelor este despartit de valoarea lor prin semnul (=). Perechile nume/valoare sunt despartite prin (&).

Utilizarea colectiei QueryString

Pentru a întelege utilizarea colectiei QueryString, se va considera formularul din figura 7.

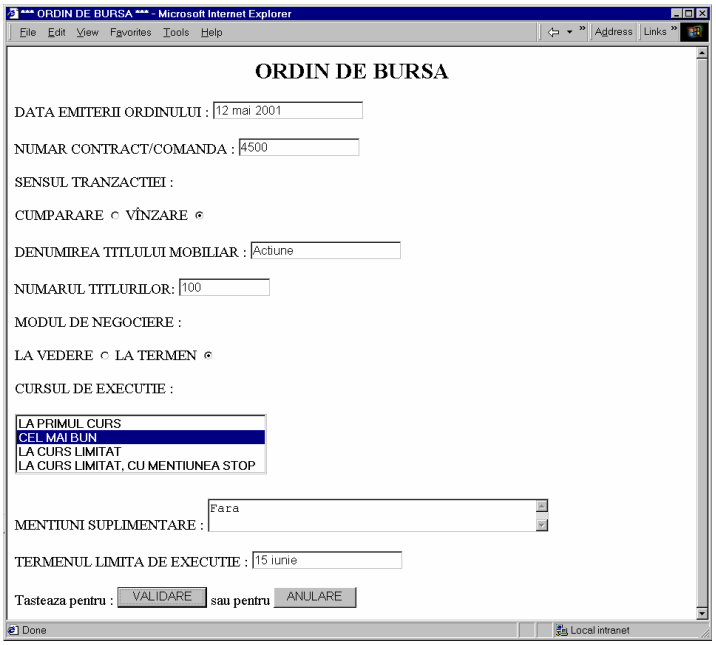

Fig.5. Pagina care include un formular

Cod HTML prin care se genereaza intrarea Tip_Tranz din formular este:

```
<P> SENSUL TRANZACTIEI :
<P>CUMPARARE
  <INPUT NAME=Tip_Tranz TYPE=radio
CHECKED VALUE=Cumparare>
  VÎNZARE
 <INPUT NAME=Tip_Tranz TYPE=radio
VALUE=Vinzare>
```

Pentru a avea acces la un element al colectiei este suficient sa se utilizeze numele respectivului element, în cazul de fata Tip_Tranz. În acest caz, codul scriptului poate fi:

```
<% @Language=VBScript %>
<B> Formularul include: </B><P>

Intrarea  <B>'<% = "Tip_Tranz" %>'</B>
cu valoarea: <I>'<% =
 Request.QueryString("Tip_Tranz") %>'</I>
```

Utilizând un navigator se obtine (figura 8):

Fig.6. Utilizarea colectiei QueryString

Valorile din colectie nu pot fi accesate printr-un index numeric. Prin urmare, este dificila parcurgerea colectiei prin instructiunea For...Next si un contor numeric. Totusi, colectia accepta o instructiune de forma For...Each cu sintaxa generala:

```
For Each element in colectie
....
Next
```

Astfel, pentru a afisa datele trimise din pagina HTML anterioara, se poate utiliza codul urmator, care parcurge colectia QueryString:

```
<% @Language=VBScript %>
<B> Formularul include: </B><P>
<%
For Each Item in Request.QueryString %>
Intrarea  <B>'<% = Item %>'</B>
cu valoarea: <I>'<% = Request.QueryString(Item) %>'</I><BR>
<% Next %>
```

Rezultatul este redat în figura 9:

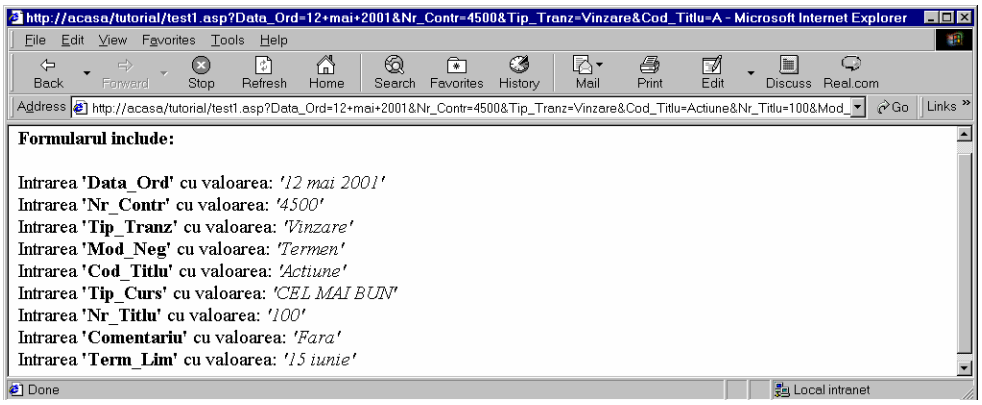

Figura 7. Parcurgerea colectiei QueryString

Utilizarea colectiei Form

Când valoarea atributului METHOD este POST, datele care se trimit catre server sunt incluse în antetul HTTP, colectia QueryString este vida iar valorile transmise vor fi accesibile prin colectia Form a obiectului Request. Este posibil ca aceasta colectie sa fie parcursa în acelasi fel ca QueryString. Pentru a parcurge întreaga colectie Form se va utiliza codul:

```
<% @Language=VBScript %>
<B> Formularul include: </B><P>
<%
For Each Item in Request.Form %>

Intrarea  <B>'<% = Item %>'</B>
cu valoarea: <I>'<% = Request.Form(Item) %>'</I><BR>
<% Next %>
```

Elemente de colectie care detin mai multe valori

Daca pentru majoritatea câmpurilor de intrare dintr-un formular exista o corespondenta între nume si valoare, în unele situatii se utilizeaza un ansamblu de intrari cu acelasi nume. De data aceasta unui nume poate sa-i corespunda mai multe valori. În mod normal, va fi transmisa valoarea selectata.

Proprietatea *Count* a colectiei receptioneaza numarul de elemente ale unei colectii, ceea ce permite sa se stie daca au fost parcurse. Din formularul prezentat anterior se pot recupera toate valorile grupelor de optiuni prin codul:

```
<% @Language=VBScript %>
<B> Formularul include: </B><P>

<%For Each Item in Request.Form
   If Request.Form(Item).Count Then
    For intLoop = 1 to Request.Form(Item).Count
%>
<% = Item & " : Index = " & intLoop & "
Valoare = " & Request.Form(Item)(intLoop) %>
<BR>
<% Next
Else %>
<% = Request.Form(Item) %>
<% End If
Next %>
```

Colectia Form este parcursa prin structura repetitiva For..Each. Cum anumite elemente ale colectiei pot fi reprezentate printr-un grup de valori, aceasta se verifica prin proprietatea Count. În formularul an-

terior valorile sunt unice, rezultatul este (figura 10):

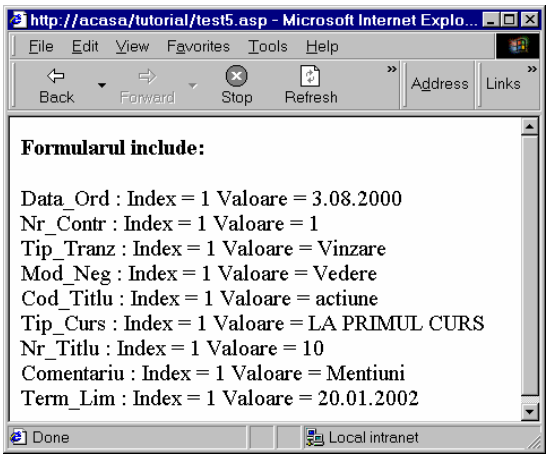

Fig.8. Utilizarea proprietatii Count a colectiei Form (1)

Vom considera un alt formular, cu selectie multipla (figura 11):

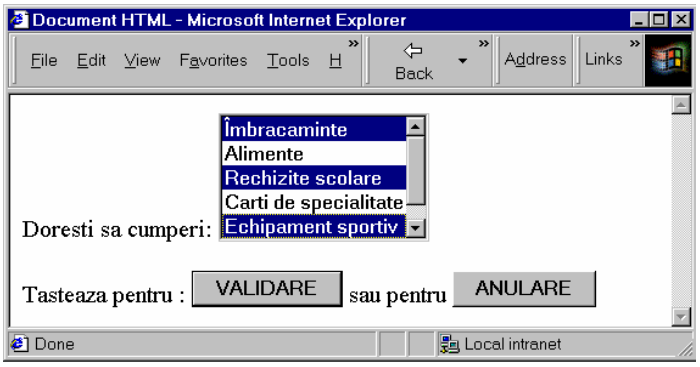

Fig.9. Formular cu selectie multipla

Daca se utilizeaza acelasi script, rezultatul este redat în figura 12 (au fost selectate trei optiuni):

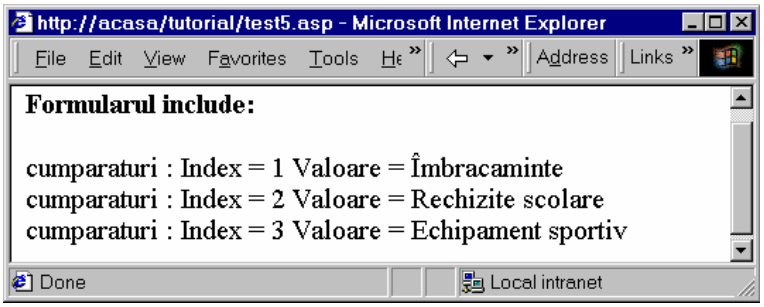

Fig.10. Utilizarea proprietatii Count a colectiei Form (2)

În mod similar, proprietatea Count se poate folosi si pentru colectia QueryString.
Prin intermediul unui formular se pot insera noi înregistrari într-un tabel al unei baze de date. De exemplu, informatiile tastate în formularul din figura 13 vor fi preluate si incluse în tabelul Stud al bazei de date Evid_stud. Printr-un singur fisier de tip .asp se poate obtine afisarea formularului si preluarea datelor care vor fi trimise în baza de date. Continutul acestui fisierului este:

```
<% @Language=VBScript %>
<html
><head><TITLE>Exemplu</TITLE></head>
<body bgcolor="#FFFFFF" text="#000000">
<%
IF request.form ("Mesaj")="True" THEN
strSID=request.form("txtSID")
strNume=request.form("txtPrenume")
strPrenume=request.form("txtNume")
strAdresa=request.form("txtAdresa")
strOras=request.form("txtOras")
strTara=request.form("txtTara")
strCod=request.form("txtCod")

strProvider = "Driver={Microsoft Access Driver
(*.mdb)};
DBQ=C:\Inetpub\Wwwroot\Proiect\Evid_stud.mdb
;"

set objConn =
 server.createobject("ADODB.Connection")

objConn.Open strProvider

set cm =
 Server.CreateObject("ADODB.Command")
cm.ActiveConnection = objConn

cm.CommandText ="INSERT INTO Stud
(SID,Prenume,Nume,Adresa,Oras,Tara,Cod)
VALUES (?,?,?,?,?,?,?)"

set objparam=cm.createparameter(, 200, , 255,
 strSID)
cm.parameters.append objparam
set objparam=cm.createparameter(, 200, , 255,
 strPrenume)
cm.parameters.append objparam
set objparam=cm.createparameter(, 200, , 255,
strNume)
cm.parameters.append objparam
set objparam=cm.createparameter(, 200, , 255,
 strAdresa)
cm.parameters.append objparam
set objparam=cm.createparameter(, 200, , 255,
strOras)
cm.parameters.append objparam
set objpara m=cm.createparameter(, 200, , 255,
 strTara)
cm.parameters.append objparam
set objparam=cm.createparameter(, 201, , 255
, strCod)
cm.parameters.append objparam
cm.execute
response.write("OK!")
ELSE%>
<h1>Adaugarea unui nou student</h1>
<form name=student1.asp action="student1.asp"
method="POST">
<input type="HIDDEN" name="Mesaj"
 value="True">
<table>
   <tr><td><div class="lsnFig"><table
 border="1">
      <tr>
        <td><span class="normal">ID
 student:</span></td>
        <td><input type="text" size="8"
 name="txtSID"></td>
      </tr>
      <tr>
        <td><span
class="normal">Prenume:</span></td>
        <td><input type="text" size="15"
 name="txtPrenume"></td>
      </tr>
      <tr>
        <td><span
class="normal">Nume:</span></td>
        <td><input type="text" size="20"
 name="txtNume"></td>
      </tr>
      <tr>
        <td><span
class="normal">Adresa:</span></td>
        <td><input type="text" size="20"
 name="txtAdresa"></td>
      </tr>
      <tr>
        <td><span
class="normal">Oras:</span></td>
        <td><input type="text" size="15"
name="txtOras"></td>
      </tr>
      <tr>
        <td><span
class="normal">Tara:</span></td>
        <td><input type="text" size="2"
 name="txtTara"></td>
      </tr>
      <tr>
        <td><span
class="normal">Cod:</span></td>
        <td><input type="text" size="9"
 name="txtCod"></td>
      </tr>
      <tr>
        <td colspan="2" align="center">
      <input type="submit" value="Submit">
      <input type="reset"  value="Reset"> </td>
       </tr>
     </table></form></center>
<%End if%>
</body></html>
```

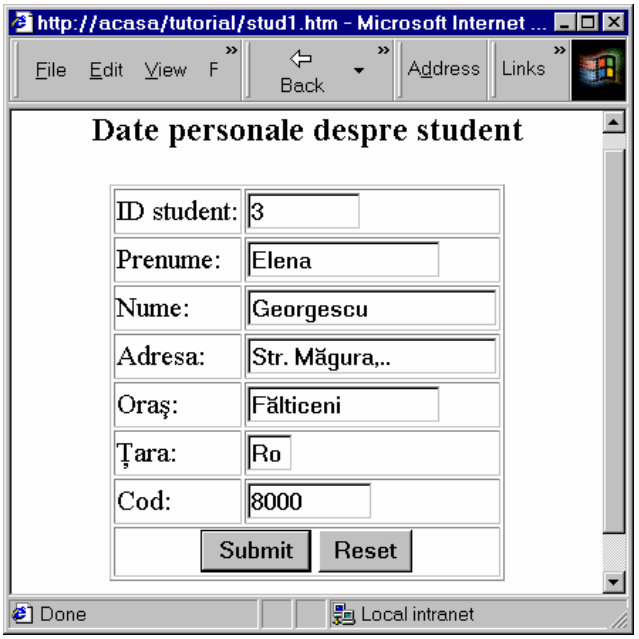

Fig.11. Preluarea datelor prin formular

Un formular bine conceput include un script client care valideaza informatia introdusa înainte ca aceasta sa fie trimisa catre server. Scripturile de validare pot verifica: daca utilizatorul a introdus un numar valid, daca o intrare text a fost completata sau, mai mult, sa se efectueze unele calcule elementare.

În general e bine sa se efectueze cât mai multe validari pe partea clientului. Nu numai ca se poate avertiza utilizatorul mult mai rapid asupra erorilor, dar se reduce timpul de raspuns, si se evita supraîncarcarea inutila a server-ului, eliberând latimea de banda pentru alte aplicatii.

Bibliografie:

[1] ******* www.asp101.com;
[2] *** www.microsoft.com;
[3] Homer A., Gill D., Jakab S., Interface entre Web et bases de donnees**,** Eyrolles, 1998.