

CURS 9 MPP

- Validarea campurilor folosind controlul `RequiredFieldValidator`
- Validarea campurilor folosind controlul `RangeValidator`
- Validarea campurilor folosind controlul `RegularExpressionValidator`
- Utilizarea functiilor de validare executate pe server (verificare duplicat pe cheie semantica)
- Utilizarea unei functii de validare executata de browser (JavaScript)
- Vizualizarea si selectia datelor folosind ca sursa de date un fisier XML
- Utilizarea bazelor de date SQL Server

1. Validarea campurilor folosind controlul `RequiredFieldValidator`

Controlul *RequiredFieldValidator* ne efectueaza validarea de obligativitate a completarii unui camp. Vom considera ca exemplu o pagina pt. introducerea datelor despre produse.

Pas 1. Crearea unei noi pagini `RFV.aspx` si setarea ei ca pagina de start.

Pas 2. Plasarea unui control `Label` (id => `LblDProdus` si `Text` => `Produs`) si a unui control `TextBox` (id => `TxDProdus`), ambele avand `Position` => `Absolute`.

Pas 3. Plasarea unui control `RequiredFieldValidator`, din grupa `Validation` de la `ToolBox` (Id => `RFVDProdus` `Position` => `Absolute`.

- Vom opta pt. afisarea mesajelor de eroare intr-un control `ValidationSummary`
- Setam urmatoarele proprietati ale controlului `RFVDProdus`:
 - `Control to validate` => `TxDProdus`
 - `Error Message` => `Completati Produs`
 - `Text` => ` `
 - `SetFocusOnError` => `True`

Pas 4. Plasarea unui control `ValidationSummary` (Id => `VS1`) `Position` => `Absolute`.

- Setam proprietatea `ForeColor` pe valoarea `#FF3300` (Rosu)

Pas 5. Plasarea unui control `Button` (Id => `BtnConfirma`, `Text` => `Confirma`) avand `Position` => `Absolute`

- De observat proprietatea `CausesValidation` setata implicit pe `True`

Pas 6. Plasarea unui control `Button` (Id => `BtnRenunta`, `Text` => `Renunta`) avand `Position` => `Absolute`

- Setam proprietatea `CausesValidation` pe `False`

Pas 7. Se parcurg pasii 2-3 pentru campul `PretVanzare`

Pas 8. Plasarea unui control `Label` (id => `LblUM` si `Text` => `UM`) si a unui control `DropDownList` (id => `DdlUM`), ambele avand `Position` => `Absolute`.

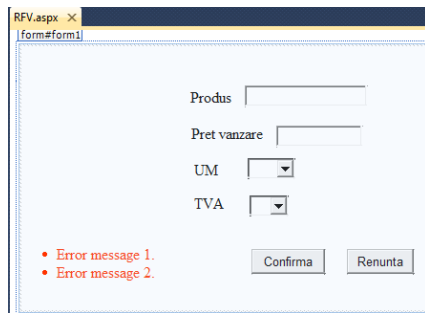
- Configurarea controlului `DdlUM` cu o lista de unitati de masura, prima fiind sirul vid.

Pas 9. Plasarea unui control `RequiredFieldValidator`, din grupa `Validation` de la `ToolBox` (Id => `RFVUM` `Position` => `Absolute`.

- Setam urmatoarele proprietati ale controlului `RFVUM`:
 - `Control to validate` => `DdlUM`
 - `Error Message` => `Completati UM`
 - `Text` => ` `
 - `SetFocusOnError` => `True`

Pas 10. Se parcurg pasii 8-9 pt. campul `CotaTVA`

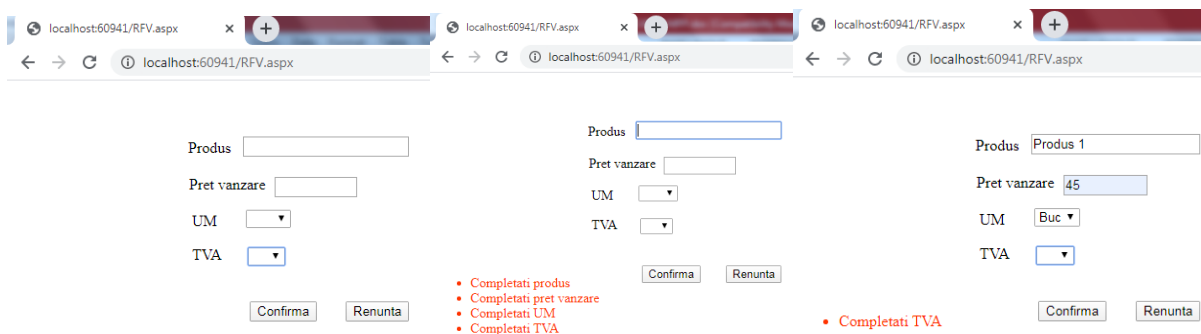
- Structura paginii in modul `Design`:



- Textul paginii in modul Source:

```
<form id="form1" runat="server">
<asp:Label ID="LblDProdus" runat="server" Text="Produs"
style="position: absolute; top: 61px; left: 189px;"></asp:Label>
<asp:TextBox ID="TxtDProdus" runat="server"
style="position: absolute; top: 58px; left: 246px;"></asp:TextBox>
<asp:RequiredFieldValidator ID="RFVProdus" style="position: absolute; top: 58px; left: 406px;"
runat="server" ErrorMessage="Completati produs"
ControlToValidate="TxtDProdus" SetFocusOnError="True">&nbsp;&nbsp;&nbsp;</asp:RequiredFieldValidator>
<asp:Label ID="LblPretVanzare" runat="server" Text="Pret vanzare"
style="position: absolute; top: 99px; left: 190px;"></asp:Label>
<asp:TextBox ID="TxtPretVanzare" runat="server"
style="position: absolute; top: 100px; left: 279px; width: 82px;"></asp:TextBox>
<asp:RequiredFieldValidator ID="RFVPretVanzare" runat="server" ErrorMessage="Completati pret vanzare"
style="position: absolute; top: 100px; left: 387px;"
ControlToValidate="TxtPretVanzare" SetFocusOnError="True">&nbsp;&nbsp;&nbsp;</asp:RequiredFieldValidator>
<asp:Label ID="LblUM" runat="server" Text="UM"
style="position: absolute; top: 137px; left: 193px;"></asp:Label>
<asp:ValidationSummary ID="VS1" runat="server"
style="position: absolute; top: 225px; left: 9px; ForeColor="#FF3300" />
<asp:Button ID="BtnConfirma" runat="server" Text="Confirma"
style="position: absolute; top: 229px; left: 253px;" />
<asp:Button ID="BtnRenunta" runat="server" Text="Renunta"
style="position: absolute; top: 229px; left: 353px; CausesValidation="False" />
<asp:DropDownList ID="DdlUM" runat="server"
style="position: absolute; top: 134px; left: 249px;">
<asp:ListItem></asp:ListItem>
<asp:ListItem>Buc</asp:ListItem>
<asp:ListItem>Kg</asp:ListItem>
<asp:ListItem>M</asp:ListItem>
<asp:ListItem>MP</asp:ListItem>
</asp:DropDownList>
<asp:RequiredFieldValidator ID="RFVUM" runat="server"
ErrorMessage="Completati UM"
style="position: absolute; top: 132px; left: 313px; ControlToValidate="DdlUM"
SetFocusOnError="True">&nbsp;&nbsp;&nbsp;</asp:RequiredFieldValidator>
<asp:Label ID="LblTVA" runat="server" Text="TVA"
style="position: absolute; top: 172px; left: 193px;"></asp:Label>
<asp:DropDownList ID="DdlTVA" runat="server"
style="position: absolute; top: 173px; left: 251px;">
<asp:ListItem></asp:ListItem>
<asp:ListItem>0</asp:ListItem>
<asp:ListItem>5</asp:ListItem>
<asp:ListItem>9</asp:ListItem>
<asp:ListItem>19</asp:ListItem>
</asp:DropDownList>
<asp:RequiredFieldValidator ID="RFVTVA" runat="server"
ErrorMessage="Completati TVA"
style="position: absolute; top: 174px; left: 344px; height: 13px;"
ControlToValidate="DdlTVA" SetFocusOnError="True">&nbsp;&nbsp;&nbsp;</asp:RequiredFieldValidator>
</form>
```

Pas 11. Verificarea functionarii paginii



Observatii

- Initial nu se afiseaza mesaje de eroare

- Click pe butonul confirma => afisare mesaje eroare
- Validarea se declanseaza la click pe butonul Confirma (propr. CausesValidation pe True)
- Validarea nu se declanseaza la click pe butonul Renunta (propr. CausesValidation pe False)
- Pe masura ce completam campurile, mesaje de eroare dispar.
- Cursorul este pozitionat pe campul necompletat (propr. SetFocusOnError pe True)

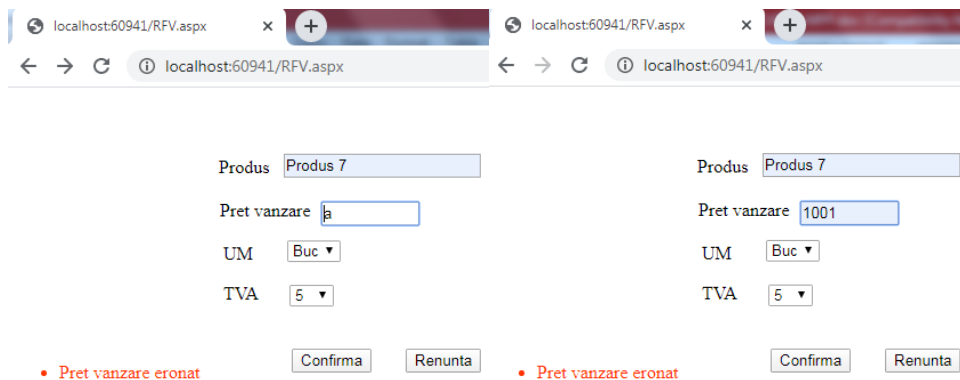
2. Validarea campurilor folosind controlul RangeValidator

Controlul *RangeValidator* realizeaza validarea de tip de date si apartenenta valorii unui camp la un interval. Vom extinde exemplul anterior cu validarea de numericitate a pretului de vanzare si apartenenta lui la intervalul 0 – 1000.

. Pas 1. Plasam un control RangeValidator (id => RVPretVanzare) Position => Absolute.

- FieldToValidate => TxtPretVanzare
- Error Message => Pret vanzare eronat
- Maximum Value => 1000
- MinimumValue => 0
- SetFocusOnError => True
- Text =>
- Type => Currency

Pas 2. Verificare executie pagina

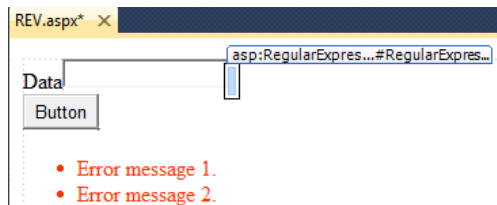


3. Validarea campurilor folosind controlul RegularExpressionValidator

Controlul *RegularExpressionValidator* permite validarea unui format specific pt. valoarea unui camp. Sa consideram ca exemplu validarea unei date calendaristice de forma zz.ll.aaaa.

Pas 1. Crearea unei noi pagini REV.aspx si setarea ei ca pagina de start.

Pas 2. Structurarea paginii:



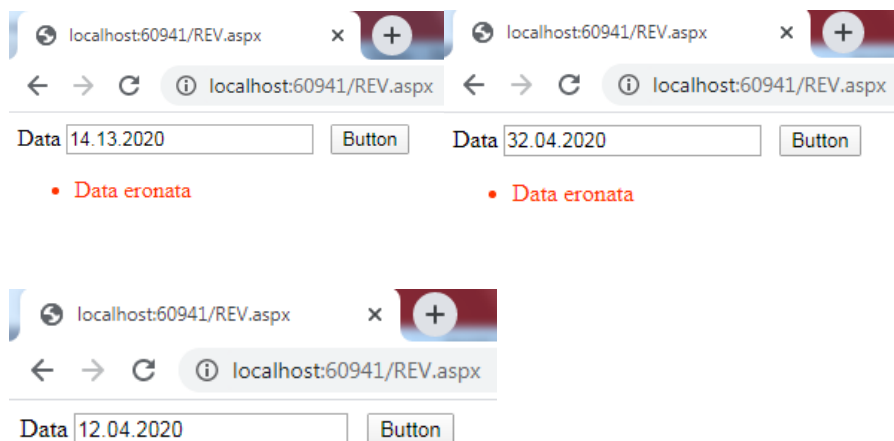
Pas 3. Configurarea controlului RegularExpressionValidator

- FieldToValidate => TextBox1
- Error Message => Data eronata
- SetFocusOnError => True
- Text =>
- ValidationExpression => (0[1-9]|[12][0-9]|3[01])[.](0[1-9]|1[012])[.]\d{4}

Textul sursa al paginii:

```
<form id="form1" runat="server">
<asp:Label ID="Label1" runat="server" Text="Data"></asp:Label>
<asp:TextBox ID="TextBox1" runat="server" Width="115px"></asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
    ErrorMessage="Data eronata" ControlToValidate="TextBox1"
    ValidationExpression="(0[1-9]|[12][0-9]|3[01])[.](0[1-9]|1[012])[.]\d{4}">&nbsp;</asp:RegularExpressionValidator>
<br />
<asp:Button ID="Button1" runat="server" Text="Button" />
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
    ForeColor="#FF3300" />
</form>
```

Pas 4. Executie pagina



- Limbajul pentru descrierea expresiilor de validare

Specificatori

.	Orice caracter diferit de Enter
[lista caractere]	Orice caracter individual ce apare in lista ex. [abc]
[^lista caractere]	Orice caracter individual ce nu apare in lista ex. [^abc]
[dom ₁ , dom ₂ , ... , dom _n]	Orice caracter individual dintr-unul din domeniile specificate Ex. [3-7,a-d,A-D]
\w	Orice caracter alfanumeric
\W	Orice caracter non alfanumeric
\s	Orice caracter spatiu (spatiu, tab, formfeed, newline)

\S	Orice caracter non spatiu
\d	Orice caracter cifra
\D	Orice caracter non cifra

Cuantificatori

*	zero sau mai multe aparitii
+	una sau mai multe aparitii
?	zero sau o aparitie
{N}	N aparitii
{N,}	N sau mai multe aparitii
{N,M}	Intre N si M aparitii

- Exemplu: considerand un camp de tip CNP si am vrea sa impunem formatul de 13 cifre ar trebui sa avem expresia: \d{13}

4. Utilizarea functiilor de validare executate pe server (verificare duplicat pe cheie semantica)

Pas 1. Consideram o baza de date continand tabela:

Produse(IdProdus, DProdus, PretVanzare, UM, CotaTVA, IdCategorie, SpPoza)

in care DProdus este cheie semantica

Pas 2. Cream o pagina noua SV.aspx si o setam ca pagina de start

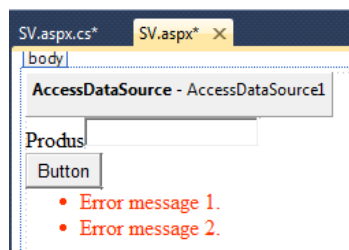
Pas 3. Plasam si configuram un control AccessDataSource. Plasam o eticheta si un TextBox pt. denumirea produsului.

Pas 4. Se plaseaza pe pagina un control CustomValidator si se configureaza:

- ErrorMessage => Produs deja existent
- Text =>
- FieldToValidate => TextBox1
- SetFocusOnError => True

Pas 5. Plasam deasemenea un control ValidationSummary

Structura paginii:



Sursa paginii:

```
<form id="form1" runat="server">
<asp:AccessDataSource ID="AccessDataSource1" runat="server"
    DataFile="~/App_Data/BD1.accdb"
    SelectCommand="SELECT [IdProdus], [DProdus], [UM], [CotaTVA], [PretVanzare], [IdCategorie], [SpPoza] FROM [Produse]">
</asp:AccessDataSource>
<asp:Label ID="Label1" runat="server" Text="Produs"></asp:Label>
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:CustomValidator ID="CustomValidator1" runat="server"
    ErrorMessage="Produs deja existent" ControlToValidate="TextBox1"
    OnServerValidate="CustomValidator1_ServerValidate" SetFocusOnError="True">&nbsp;</asp:CustomValidator>
<br />
<asp:Button ID="Button1" runat="server" onclick="Button1_Click" Text="Button" />
<asp:ValidationSummary ID="ValidationSummary1" runat="server"
    ForeColor="#FF3300" Width="267px" />
</form>
```

Pas 6. Se creaza procedura eveniment CustomValidator1_ServerValidate (dublu click pe controlul CustomValidator1)

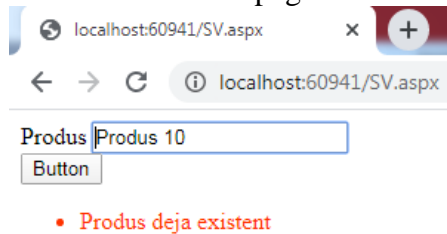
```
protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args) {
    OleDbConnection con = new OleDbConnection();
    con.ConnectionString = AccessDataSource1.ConnectionString;
    OleDbCommand cmd = new OleDbCommand();
    cmd.Connection = con;
    cmd.CommandText = "Select IdProdus from Produse where DProdus='" + TextBox1.Text + "'";

    OleDbDataReader rd;

    con.Open();

    rd = cmd.ExecuteReader();
    if (rd.Read()) args.IsValid = false;
    else args.IsValid = true;
    con.Close();
}
```

Pas 7. Verificam functionarea paginii



5. Utilizarea unei functii de validare executata de browser (JavaScript)

Vom lua ca exemplu o functie de validare scrisa in JavaScript pt. validarea si extragerea datei din CNP

Pas 1. Cream o pagina noua CV.aspx si o setam ca pagina de start

Pas 2. Se creaza o eticheta cu textul CNP si un textBox cu id-ul TxtCNP.

Pas 3. Se creaza un control CustomValidator:

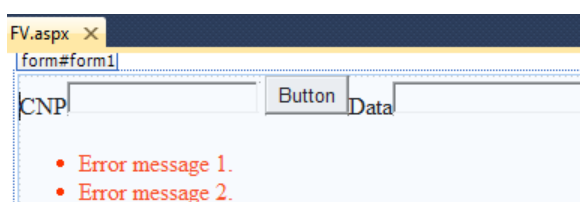
- ControlToValidate => txtCNP
- ErrorMessage => CNP Eronat
- Text =>
- ClientValidationFunction => f
- SetFocusOnError => True

Pas 4. Se creaza un control de tip Button.

Pas 5. Se creaza o eticheta cu textul Data si un textBox cu id-ul TxtDN.

Pas 6. Se creaza un control ValidationSummary.

Structura paginii (Design) este:



Pas 7. Definirea functiei f in codul HTML.

```
function f(ct1, args) {
    var s = args.Value;
    var an, luna, zi;

    an = s.substr(1, 2);
    luna = s.substr(3, 2);
    zi = s.substr(5, 2);

    if (luna < 1 || luna > 12)
    { args.IsValid = false; document.getElementById("TxtDN").value=""; return; }

    document.getElementById("TxtDN").value = luna + "/" + zi + "/" + an;
    args.IsValid = true;
}
```

Textul paginii (Source) este:

```
<head runat="server">
    <script type="text/javascript" >
        function f(ct1, args) {
            var s = args.Value;
            var an, luna, zi;

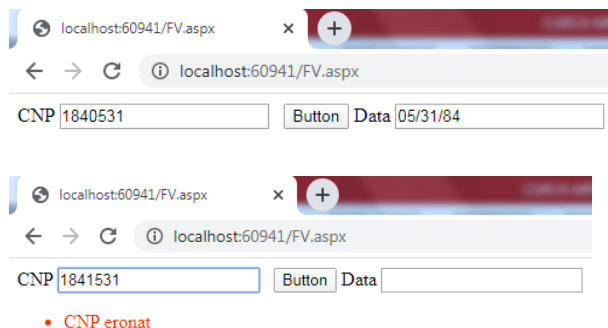
            an = s.substr(1, 2);
            luna = s.substr(3, 2);
            zi = s.substr(5, 2);

            if (luna < 1 || luna > 12)
            { args.IsValid = false; document.getElementById("TxtDN").value=""; return; }

            document.getElementById("TxtDN").value = luna + "/" + zi + "/" + an;
            args.IsValid = true;
        }
    </script>
    <title></title>
</head>

<body>
    <form id="form1" runat="server">
        <asp:Label ID="Label1" runat="server" Text="CNP"></asp:Label>
        <asp:TextBox ID="TxtCNP" runat="server"></asp:TextBox>
        <asp:CustomValidator ID="CustomValidator1" runat="server"
            ErrorMessage="CNP eronat" ClientValidationFunction="f"
            ControlToValidate="TxtCNP" SetFocusOnError="True">&nbsp;</asp:CustomValidator>
        <asp:Button ID="Button1" runat="server" Text="Button" />
        <asp:Label ID="Label2" runat="server" Text="Data"></asp:Label>
        <asp:TextBox ID="TxtDN" runat="server"></asp:TextBox>
        <asp:ValidationSummary ID="ValidationSummary1" runat="server"
            ForeColor="#FF3300" />
    </form>
</body>
```

Executie:



6. Vizualizarea si selectia datelor folosind ca sursa de date un fisier XML

Pas 1. Includerea in aplicatie a unui fisier XML: Add => New Item => XML File.

In aplicatie va fi inclus un fisier XMLFile1.xml. Se deschide de asemenea o fereastră de editare pentru fisierul respectiv, ce contine linia:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Pas 2. Se editeaza continutul fisierului xml:

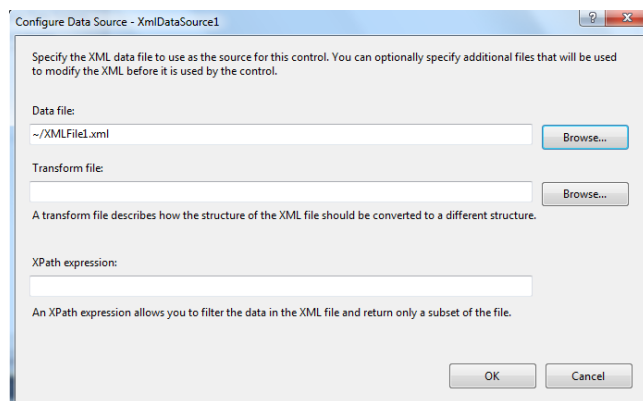
```
<?xml version="1.0" encoding="utf-8" ?>
<oferta>
  <Produs Id="1" Denumire="Produs 01" Um="Buc" Pret="15" />
  <Produs Id="2" Denumire="Produs 02" Um="Buc" Pret="20" />
  <Produs Id="3" Denumire="Produs 03" Um="Kg" Pret="10" />
  <Produs Id="4" Denumire="Produs 04" Um="Mp" Pret="30" />
</oferta>
```

Pas 3. Se creaza o noua pagina si se declara ca pagina de start.

Pas 4. Se plaseaza pe pagina un control XmlDataSource:

Pas 5. Se configureaza sursa de date completand Data file cu:

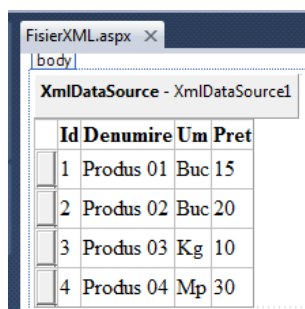
~/XMLFile1.xml



Pas 6. Se plaseaza pe pagina un control GridView

Pas 7. Se configureaza controlul GridView : Data Source, AutoFormat, etc.

- Structura paginii (Design):



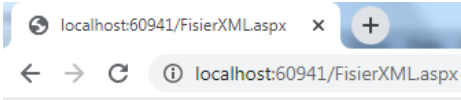
- Codul sursa al paginii:


```

<form id="form1" runat="server">
<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile="~/XMLFile1.xml"></asp:XmlDataSource>
<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataSourceID="XmlDataSource1">
    <Columns>
        <asp:CommandField ButtonType="Button" SelectText="" ShowSelectButton="True" />
        <asp:BoundField DataField="Id" HeaderText="Id" SortExpression="Id" />
        <asp:BoundField DataField="Denumire" HeaderText="Denumire"
            SortExpression="Denumire" />
        <asp:BoundField DataField="Um" HeaderText="Um" SortExpression="Um" />
        <asp:BoundField DataField="Pret" HeaderText="Pret" SortExpression="Pret" />
    </Columns>
    <SelectedRowStyle BackColor="#CCCCFF" />
</asp:GridView>
</form>

```

Pas 8. Executie



	Id	Denumire	Um	Pret
<input type="checkbox"/>	1	Produs 01	Buc	15
<input type="checkbox"/>	2	Produs 02	Buc	20
<input type="checkbox"/>	3	Produs 03	Kg	10
<input type="checkbox"/>	4	Produs 04	Mp	30

7. Utilizarea bazelor de date SQL Server

Pas 1. Crearea unei baze de date cu SGBD-ul SQL Server:

- Solution explorer => Click dreapta pe numele aplicatiei => Add => New Item => SQL Server DataBase
- Ne propune numele Database1.mdf pentru baza de date nou creata
- Acceptam numele si apasam butonul Add
- Ne intreaba daca sa creeze baza de date in folderul App_Data al aplicatiei
- Acceptam si verificam in Solution Explorer crearea acestei baze de date

Pas 2. Cream tabela Produse

- Click dreapta pe Database1.mdf => Open
- Se deschide fereastra Database Explorer peste fereastra Solution Explorer
- Click dreapta pe optiunea Tables => Add New Table
- Se deschide o fereastra pentru specificarea structurii noii tabele
- Prima linie va contine IdProdus (Column Name) Int (Data Type)
- Click dr pe linia IdProdus si selectia optiunii Set Primary Key
- In sectiunea Column Properties (din partea inferioara a ecranului) deschidem (+) gruparea Identity Specification si setam Yes pentru optiunea (Is Identity). In acest mod realizam echivalenta cu tipul Autonumber de la SGBD-ul Access.
- Continuum cu specificarea celorlalte atribute: DProdus nchar(10), Um nchar(10), CotaTVA int, PretVanzare numeric(18,0), Poza nchar(20)
- Click dr. pe tab-ul tabelei => Save Table1 si specificam numele Produse pentru tabela creata
- Click dr. pe tab-ul tabelei produse => Close
- Click dr. pe tabela Produse din fereastra Database Explorer => Show Table Data
- Se deschide o fereastra de editare pentru Tabela Produse
- Completam cateva inregistrari pentru tabela Produse
- Click dr. pe tab-ul tabelei => close

Column Name	Data Type	Allow Nulls
IdProdus	int	<input type="checkbox"/>
DProdus	nchar(10)	<input checked="" type="checkbox"/>
UM	nchar(10)	<input checked="" type="checkbox"/>
CotaTVA	int	<input checked="" type="checkbox"/>
PretVanzare	numeric(18, 0)	<input checked="" type="checkbox"/>
Poza	nchar(20)	<input checked="" type="checkbox"/>

Pas 3. Crearea unei noi pagini si setarea ei ca pagina de start.

Pas 4. Se plaseaza pe pagina un control SQLDataSource

Pas 5. Se configureaza controlul SDS1:

- Choose your data connection => deschidem combo box-ul si selectam Database1.mdf
- Confirmam optiunea pentru salvarea conexiunii in fisierul de configurare
- Configuram cererea SQL => Specify columns for a table or view
- Test query

Pas 6. Plasam un control GridView (GV1)

Pas 7. Configuram controlul GV1: Autoformat, DataSource, etc

- Structura paginii (modul Design):

IdProdus	DProdus	UM	CotaTVA	PretVanzare	Poza
0	abc	abc	0	0	abc
1	abc	abc	1	0.1	abc
2	abc	abc	2	0.2	abc
3	abc	abc	3	0.3	abc
4	abc	abc	4	0.4	abc

- Codul sursa al paginii:

```
<form id="form1" runat="server">
  <asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%= ConnectionStrings.ConnectionString %>"
    SelectCommand="SELECT [IdProdus], [DProdus], [UM], [CotaTVA], [PretVanzare], [Poza] FROM [Produse]">
  </asp:SqlDataSource>
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataSourceID="SqlDataSource1">
    <Columns>
      <asp:BoundField DataField="IdProdus" HeaderText="IdProdus"
        SortExpression="IdProdus" />
      <asp:BoundField DataField="DProdus" HeaderText="DProdus"
        SortExpression="DProdus" />
      <asp:BoundField DataField="UM" HeaderText="UM" SortExpression="UM" />
      <asp:BoundField DataField="CotaTVA" HeaderText="CotaTVA"
        SortExpression="CotaTVA" />
      <asp:BoundField DataField="PretVanzare" HeaderText="PretVanzare"
        SortExpression="PretVanzare" />
      <asp:BoundField DataField="Poza" HeaderText="Poza" SortExpression="Poza" />
    </Columns>
  </asp:GridView>
</form>
```

Pas 8. Executie

IdProdus	DProdus	UM	CotaTVA	PretVanzare	Poza
1	Produs 1	Buc	19	100	~/Poze/P1.jpg
2	Produs 2	Kg	9	20	~/Poze/P2.jpg