

Laboratorul 9

Aplicație ASP.NET cu baze de date

În cadrul acestui laborator vom învăța să creăm o aplicație ASP.NET cu baze de date SQL Server sau cu MySQL. Vom realiza o aplicație pentru o companie care să poată monitoriza angajații și proiectele acestora.

1. Baza de date

Vom crea o nouă bază de date cu numele "Firma" în Microsoft SQL Server. Baza de date va conține următoarele tabele:

- Users
- Angajati
- Proiecte
- AlocareProiecte

Cu următoarele câmpuri:

OANASORINA\SQLEX...irma - dbo.Users			
Column Name	Data Type	Allow Nulls	
NumeUtilizator	nvarchar(50)	<input type="checkbox"/>	
Parola	nchar(10)	<input type="checkbox"/>	

OANASORINA\SQLEX...a - dbo.Angajati			
Column Name	Data Type	Allow Nulls	
CNP	nchar(13)	<input type="checkbox"/>	
Nume	nvarchar(50)	<input type="checkbox"/>	
Prenume	nvarchar(50)	<input type="checkbox"/>	
Departament	nvarchar(50)	<input type="checkbox"/>	

OANASORINA\SQLEX...a - dbo.Proiecte			
Column Name	Data Type	Allow Nulls	
numeProiect	nvarchar(50)	<input type="checkbox"/>	
numarOreAlocate	int	<input type="checkbox"/>	
departament	nchar(10)	<input type="checkbox"/>	

OANASORINA\SQLEX...AlocareProiecte			
Column Name	Data Type	Allow Nulls	
cnp	nchar(13)	<input type="checkbox"/>	
numeProiect	nvarchar(50)	<input type="checkbox"/>	

2. Pagina de Autentificare

Vom adăuga proiectului o nouă clasă (ConexiuneBD.cs) în care vom inițializa conexiunea la baza de date. Codul sursă al acestei clase este următorul:

```
using Microsoft.Win32;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;

namespace Aplicatie_ASP.NET_cu_Baze_de_date
{
    public class ConexiuneBD
    {
        public static SqlConnection conn = new SqlConnection("Data Source="+GetDataSources()+";Initial
        Catalog=Firma;Integrated Security=True");

        private static string GetDataSources()
        {
            string ServerName = Environment.MachineName;
            string data_source = "";
            RegistryView registryView = Environment.Is64BitOperatingSystem ? RegistryView.Registry64 :
            RegistryView.Registry32;
            using (RegistryKey hkml = RegistryKey.OpenBaseKey(RegistryHive.LocalMachine, registryView))
            {
                RegistryKey instanceKey = hkml.OpenSubKey(@"SOFTWARE\Microsoft\Microsoft SQL Server\Instance
                Names\SQL", false);
                if (instanceKey != null)
                {
                    foreach (var instanceName in instanceKey.GetValueNames())
                    {
                        data_source = data_source + (ServerName + "\\\" + instanceName);
                    }
                }
            }
            return data_source;
        }
    }
}
```

În pagina de autentificare (WebForm1.aspx) vom avea un câmp DropDownList în care se vor încărca din tabelul "Utilizatori" din baza de date toți utilizatorii care au acces la aplicație, și un câmp textBox în care utilizatorul va putea completa parola.

Figura 1. Pagina de autentificare

Pentru a putea avea acces la variabile corespunzătoare conexiunii la baza de date, vom declara aceste variabile în cadrul clasei Formei: `public partial class WebForm1 : System.Web.UI.Page {}`:

```
SqlCommand cmd;
```

```
SqlDataReader dr;
```

La încărcarea paginii (Page_Load) vom adăuga următorul cod sursă:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        try
        {
            ConexiuneBD.conn.Open();
            cmd = new SqlCommand("select NumeUtilizator from Users", ConexiuneBD.conn);

            dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                DropDownList1.Items.Add(dr[0].ToString());
            }
        }
        catch (Exception ex)
        {
            Label1.Text = "Nu se poate realiza conexiunea " + ex.Message;
        }
        finally
        {
            ConexiuneBD.conn.Close();
        }
    }
}
```

La click pe butonul "Autentificare" adăugăm următorul cod sursă:

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {
        ConexiuneBD.conn.Open();
        cmd = new SqlCommand("select Parola from Users where NumeUtilizator='" + DropDownList1.Text +
        "'", ConexiuneBD.conn);

        dr = cmd.ExecuteReader();
        if (!dr.Read())
        {
            Label1.Text = "Datele sunt gresite!";
        }
        else
        {
            string url;

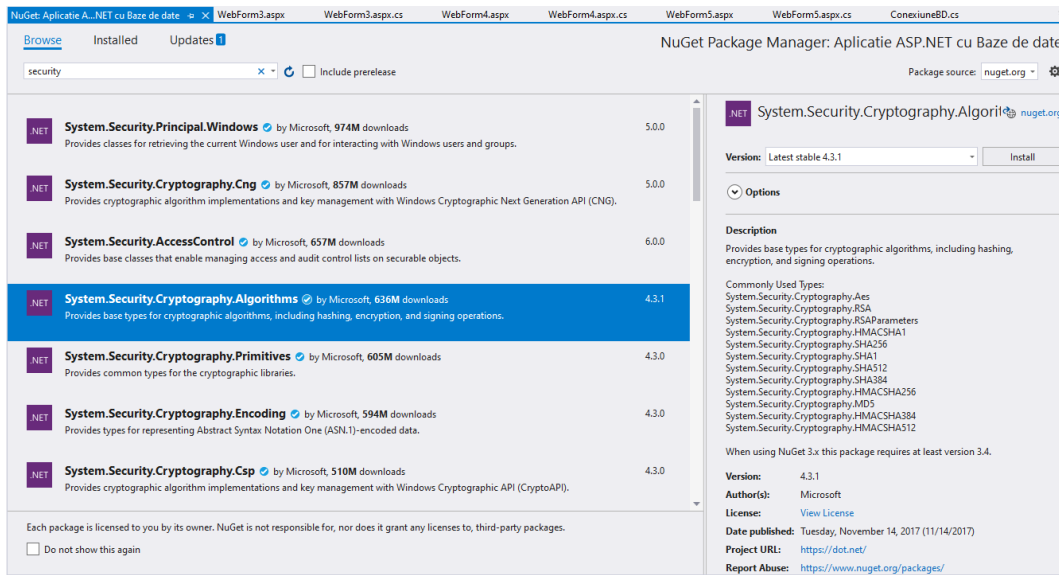
            if (EncDec.Decrypt(dr[0].ToString().Trim()) == TextBox1.Text.Trim())
            {
                Application["numeUser"] = DropDownList1.Text;
                url = "WebForm2.aspx";
                Response.Redirect(url);
            }
            else
            {
                Label1.Text = "Parola gresita!";
            }
        }
    }
    catch (Exception ex)
    {
        Label1.Text = "Nu se poate realiza conexiunea " + ex.Message;
    }
    finally
    {
        ConexiuneBD.conn.Close();
    }
}
```

```
}
}
```

La click pe butonul ”Inregistrare” adăugăm următorul cod sursă:

```
protected void Button2_Click(object sender, EventArgs e)
{
    //deschidem forma de înregistrare
    Response.Redirect("WebForm6.aspx");
}
```

Pentru a putea cripta/decripta parola vom crea o nouă clasă (EncDec.cs) în care vom crea 2 funcții pentru criptarea și decriptarea parolei cu ajutorul algoritmului AES din biblioteca:



Codul sursă al clasei EncDec.cs este:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Web;

namespace Aplicatie_ASP.NET_cu_Baze_de_date
{
    public static class EncDec
    {
        public static string Encrypt(string clearText)
        {
            string EncryptionKey = "MAKV2SPBNI99212";
            byte[] clearBytes = Encoding.Unicode.GetBytes(clearText);
            using (Aes encryptor = Aes.Create())
            {
                Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
                encryptor.Key = pdb.GetBytes(32);
                encryptor.IV = pdb.GetBytes(16);
                using (MemoryStream ms = new MemoryStream())
                {
                    using (CryptoStream cs = new CryptoStream(ms, encryptor.CreateEncryptor(), CryptoStreamMode.Write))
                    {

```

```

        cs.Write(clearBytes, 0, clearBytes.Length);
        cs.Close();
    }
    clearText = Convert.ToBase64String(ms.ToArray());
}
}
return clearText;
}

public static string Decrypt(string cipherText)
{
    string EncryptionKey = "MAKV2SPBNI99212";
    byte[] cipherBytes = Convert.FromBase64String(cipherText);
    using (Aes encryptor = Aes.Create())
    {
        Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61,
0x6e, 0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        encryptor.Key = pdb.GetBytes(32);
        encryptor.IV = pdb.GetBytes(16);
        using (MemoryStream ms = new MemoryStream())
        {
            using (CryptoStream cs = new CryptoStream(ms, encryptor.CreateDecryptor(),
CryptoStreamMode.Write))
            {
                cs.Write(cipherBytes, 0, cipherBytes.Length);
                cs.Close();
            }
            cipherText = Encoding.Unicode.GetString(ms.ToArray());
        }
    }
    return cipherText;
}
}
}

```

3. Pagina de înregistrare

La click pe butonul "Înregistrare" adăugăm următorul cod sursă:

```

protected void Button1_Click(object sender, EventArgs e)
{
    SqlCommand cmd;

    try
    {
        ConexiuneBD.conn.Open();
        cmd = new SqlCommand("insert into Users (NumeUtilizator,Parola) values(@nume,@pass) ",
ConexiuneBD.conn);
        cmd.Parameters.AddWithValue("@nume", TextBox1.Text.Trim());
        cmd.Parameters.AddWithValue("@pass", EncDec.Encrypt(TextBox2.Text.Trim()));

        int rowsAffected = cmd.ExecuteNonQuery();
        if (rowsAffected == 1)
        {
            string url = "WebForm1.aspx";

```

```

        Response.Redirect(url);
    }
    else
        LabelEroare.Text = "Eroare inserare!";
    }
    catch (Exception ex)
    {
        //log error
        LabelEroare.Text = "Eroare la deschidere baza date " + ex.Message;
    }
    //adaugarea datelor
    finally
    {
        ConexiuneBD.conn.Close();
    }
}

```

4. Pagina Principală

În cadrul paginii principale avem următoarele funcționalități (Figura 2):

- Vom avea un tabel cu toți angajații firmei
- Vom putea căuta un angajat după nume
- Vom putea adăuga un nou angajat
- Vom putea alocă un proiect unui angajat
- Vom putea completa, la alegere un chestionar despre anumite probleme legate de firmă

Managementul unei firme

Angajati Proiecte Contact

[Label1][Label2]

Numele căutat

Cauta

Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

Selectați un chestionar:

Unbound

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Figura 2. Pagina principală

La încărcarea paginii (Page_Load) vom adăuga următorul cod sursă:

```

protected void Page_Load(object sender, EventArgs e)
{
    Label2.Text = "User logat: " + (string)Application["numeUser"];
    try

```

```
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT cnp as [CNP], nume as [Nume],
prenume as [Prenume], departament as [Departament] FROM Angajati ORDER BY nume,prenume",
ConexiuneBD.conn);
    DataSet ds = new DataSet();
    da.Fill(ds);
    GridView1.DataSource = ds.Tables[0];
    GridView1.DataBind();

    //pentru chestionare
    string[] filePaths = Directory.GetFiles(Server.MapPath("~/Formulare/"));
    foreach (string filePath in filePaths)
    {
        DropDownList1.Items.Add(new ListItem(Path.GetFileNameWithoutExtension(filePath)));
    }
}
catch (Exception ex)
{
    Label1.Text = "Nu se poate realiza conexiunea la baza de date: " + ex.Message;
}
finally
{
    ConexiuneBD.conn.Close();
}
}
```

Codul sursă care suprascrive evenimentele la nivelul GridView:

```
protected override void Render(System.Web.UI.HtmlTextWriter writer)
{
    foreach (GridViewRow row in GridView1.Rows)
    {
        if (row.RowType == DataControlRowType.DataRow)
        {
            row.Attributes["onmouseover"] =
"this.style.cursor='hand';this.style.textDecoration='underline';";
            row.Attributes["onmouseout"] = "this.style.textDecoration='none';";
            row.Attributes["onclick"] = ClientScript.GetPostBackClientHyperlink(GridView1,
"Select$" + row.DataItemIndex, true);
            row.Style.Add(HtmlTextWriterStyle.Cursor, "pointer");
        }
    }
    base.Render(writer);
}
```

La evenimentul click pe butonul ”Caută” adăugăm următorul cod sursă:

```
protected void Button1_Click(object sender, EventArgs e)
{
    foreach (GridViewRow rx in GridView1.Rows)
    {
        rx.BackColor = Color.White;
        rx.ForeColor = Color.Black;
    }
    foreach (GridViewRow row in GridView1.Rows)
    {

```

```
        if (row.Cells[1].Text != null && row.Cells[1].Text.Trim() == TextBox1.Text.Trim())
        {
            GridView1.SelectedIndex = row.RowIndex;
            Button3.Enabled = true; //butonul de alocare proiect la angajat
            Button4.Enabled = true; //butonul de chestionare
            row.BackColor = Color.DarkRed;
            row.ForeColor = Color.White;
        }
    }
}
```

La evenimentul click pe butonul ”Adaugă angajat” adăugăm următorul cod sursă:

```
protected void Button2_Click(object sender, EventArgs e)
{
    string url;
    url = "WebForm3.aspx";
    Response.Redirect(url);
}
```

La evenimentul click pe butonul ”Alocare proiect angajat” adăugăm următorul cod sursă:

```
protected void Button3_Click(object sender, EventArgs e)
{
    string url;
    string cnp = GridView1.SelectedRow.Cells[0].Text.ToString().Trim();
    url = "WebForm4.aspx?cnp=" + cnp;
    Response.Redirect(url);
}
```

La evenimentul Comandă pe un rând (RowCommand) la DataGrid adăugăm următorul cod sursă:

```
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
    if (e.CommandName == "Select")
    {
        Button3.Enabled = true;
        Button4.Enabled = true; //butonul de chestionare
    }
}
```

La evenimentul click pe butonul ”Completează chestionar” adăugăm următorul cod sursă:

```
protected void Button4_Click(object sender, EventArgs e)
{
    string url;
    string cnp = GridView1.SelectedRow.Cells[0].Text.ToString().Trim();
    url = "WebForm5.aspx?cnp=" + cnp + "&chestionar="+DropDownList1.Text;
    Response.Redirect(url);
}
```

La evenimentul *GridView1_SelectedIndexChanged* adăugăm următorul cod sursă:


```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    foreach (GridViewRow rx in GridView1.Rows)
    {
        rx.BackColor = Color.White;
        rx.ForeColor = Color.Black;
    }
    GridView1.SelectedRow.BackColor = System.Drawing.Color.DarkRed;
    GridView1.SelectedRow.ForeColor = System.Drawing.Color.White;
}
```

5. Pagina Adăugare Angajat

În cadrul acestei pagini vom putea adăuga un nou angajat în cadrul firmei, în tabelul Angajati (Figura 3).

Figura 3. Adăugare angajat

La evenimentul click pe butonul "Adaugă" adăugăm următorul cod sursă:

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlCommand cmd;
    if (TextBox1.Text.ToString().Length != 13)
    {
        Label1.Visible = true;
    }
    else
    {
        try
        {
            conn.Open();
            cmd = new SqlCommand("insert into Angajati (cnp,nume,prenume,departament)
values(@cnp,@nume,@prenume,@departament) ", ConexiuneBD.conn);
            cmd.Parameters.AddWithValue("@cnp", TextBox1.Text.Trim());
            cmd.Parameters.AddWithValue("@nume", TextBox2.Text.Trim());
```

```
cmd.Parameters.AddWithValue("@prenume", TextBox3.Text.Trim());
cmd.Parameters.AddWithValue("@departament", TextBox4.Text.Trim());

int rowsAffected = cmd.ExecuteNonQuery();
if (rowsAffected == 1)
{
    string url = "WebForm2.aspx";
    Response.Redirect(url);
}
else
    LabelEroare.Text = "Eroare inserare!";
}
catch (Exception ex)
{
    //log error
    LabelEroare.Text = "Eroare la deschidere baza date " + ex.Message;
}
//adaugarea datelor
finally
{
    ConexiuneBD.conn.Close();
}
}
```

6. Pagina Alocare proiect la un angajat

În cadrul acestei pagini vom putea alocă un nou proiect unui angajat selectat din GridView, în tabelul AlocareProiecte (Figura 4).

Managementul unei firme

Angajati Proiecte Contact

ContentPlaceHolder1 (Custom) form#form1

[Label1]

Alocare proiect la angajat

CNP

Selecteaza proiect Unbound

Adauga

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Figura 4. Alocare proiecte

Pentru a putea avea acces la variabile corespunzătoare conexiunii la baza de date, vom declara/inițializa aceste variabile în cadrul clasei Formei: `public partial class WebForm1 : System.Web.UI.Page {}`:

```
SqlCommand cmd;
SqlDataReader dr;
```

La încărcarea paginii (Page_Load) vom adăuga următorul cod sursă:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        string Cnp = Request.QueryString["cnp"];
        TextBox1.Text = Cnp;
        TextBox1.Enabled = false;

        try
        {
            con.Open();
            cmd = new SqlCommand("select numeProiect from Proiecte", ConexiuneBD.conn);

            dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                DropDownList1.Items.Add(dr[0].ToString());
            }
        }
        catch (Exception ex)
        {
            Label1.Text = "Nu se poate realiza conexiunea " + ex.Message;
        }
        finally
        {
            ConexiuneBD.conn.Close();
        }
    }
}
```

La evenimentul click pe butonul "Adaugă" adăugăm următorul cod sursă:

```
protected void Button1_Click(object sender, EventArgs e)
{
    SqlCommand cmd;

    try
    {
        conn.Open();
        cmd = new SqlCommand("insert into AlocareProiecte (cnp,numeProiect)
values(@cnp,@numeP) ", ConexiuneBD.conn);
        cmd.Parameters.AddWithValue("@cnp", TextBox1.Text.Trim());
        cmd.Parameters.AddWithValue("@numeP", DropDownList1.Text.Trim());

        int rowsAffected = cmd.ExecuteNonQuery();
        if (rowsAffected == 1)
        {
            string url = "WebForm2.aspx";
            Response.Redirect(url);
        }
    }
}
```

```
    }  
    else  
        Label1.Text = "Eroare inserare!";  
    }  
    catch (Exception ex)  
    {  
        //log error  
        Label1.Text = "Eroare la deschidere baza date " + ex.Message;  
    }  
    //adaugarea datelor  
    finally  
    {  
        ConexiuneBD.conn.Close();  
    }  
}
```

7. Pagina Completare chestionar

În cadrul acestei pagini vom putea completa diferite chestionare legate de activitatea în firmă (Figura 5).

Chestionarul va fi generat automat în funcție de fișierul selectat în pagina anterioară, controalele generându-se automat dintr-un fișier xml.

The screenshot shows a web application interface. At the top, a dark red header bar contains the text 'Managementul unei firme' in a light blue font. Below this, a dark red navigation bar includes the links 'Angajati', 'Proiecte', and 'Contact'. The main content area has a light gray background. On the left, there's a label 'Chestionar' and a button 'Salvează chestionar'. To the right of the button is a large text input field. Below the input field, there's a footer bar with the text 'Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET'.

Figura 5. Chestionare

Structura fișierului XML este următoarea:

```
<?xml version="1.0" encoding="utf-8" ?>
<!-- Fisa pentru chestionar proiect-->
<Chestionar>
  <textbox nume="functii" detalii="Care sunt functiile implementate de dvs. in cadrul proiectului ?"/>
  <radio nume = "dificultate" detalii="Pe o scara de la 1 la 5 care a fost dificultatea proiectului?">
    <val>1</val>
    <val>2</val>
    <val>3</val>
    <val>4</val>
    <val>5</val>
  </radio>
  <radio nume = "implementare" detalii="Care a fost intervalul de ore alocat pentru terminarea proiectului?">
    <val>50-100</val>
    <val>100-300</val>
    <val>300-500</val>
  </radio>
  <radio nume = "limbaj" detalii="Care a fost limbajul de programare cel mai folosit?">
    <val>C++</val>
    <val>C#</val>
    <val>Java</val>
  </radio>
</Chestionar>
```

La încărcarea paginii (Page_Load) vom adăuga următorul cod sursă:

```
protected void Page_Load(object sender, EventArgs e)
{
    XmlDocument xmlSursa = new XmlDocument();
    string nume_fisier = Request.QueryString["chestionar"];
    xmlSursa.Load(Server.MapPath("~/Formulare/" + nume_fisier + ".xml"));

    ///luam toate nodurile textBox
    XmlNodeList nodes2 = xmlSursa.SelectNodes("//Chestionar/textbox");
    foreach (XmlNode text in nodes2)
    {
        string idr = text.Attributes["detalii"].Value;
        Placeholder1.Controls.Add(new LiteralControl("<p/>" + idr));
        ///textBox generat
        TextBox txt = new TextBox();
        txt.ID = text.Attributes["nume"].Value;
        txt.Text = "";
        Placeholder1.Controls.Add(new LiteralControl("<p/>"));
        Placeholder1.Controls.Add(txt);
    }

    ///luam toate nodurile radioButton
    XmlNodeList nodes = xmlSursa.SelectNodes("//Chestionar/radio");
    foreach (XmlNode radio in nodes)
    {
        string idr = radio.Attributes["detalii"].Value;
        Placeholder1.Controls.Add(new LiteralControl("<p/>" + idr));
        RadioButtonList new_radio = new RadioButtonList();
        new_radio.ID = radio.Attributes["nume"].Value;
        XmlNodeList valori = radio.ChildNodes;
        foreach (XmlNode valoare in valori)
        {
            string val = valoare.InnerText;
            new_radio.Items.Add(val);
        }
        new_radio.RepeatDirection = RepeatDirection.Horizontal;
        Placeholder1.Controls.Add(new_radio);
    }
}
```

La evenimentul click pe butonul ”Salvează chestionar” adăugăm următorul cod sursă:

```
protected void Button1_Click(object sender, EventArgs e)
{
    foreach (TextBox txt in Placeholder1.Controls.OfType<TextBox>())
    {
        string nume_Control = txt.ID;
        string valoare_Control = txt.Text;
        TextBox1.Text = TextBox1.Text + "Nume control: " + nume_Control + " Valoare: " +
        valoare_Control + Environment.NewLine;
    }

    foreach (RadioButtonList rb in Placeholder1.Controls.OfType<RadioButtonList>())
    {
        string nume_Control = rb.ID;
        string valoare_Control = rb.SelectedValue;
        TextBox1.Text = TextBox1.Text + "Nume control: " + nume_Control + " Valoare: " +
        valoare_Control + Environment.NewLine;
    }
}
```

Capturi ecran – rulare

Autentificare

Utilizator

Parola

Înregistrare

Utilizator

Parola

Managementul unei firme

[Angajati](#) [Proiecte](#) [Contact](#)

User logat: Oana Chirila

Numele căutat

CNP	Nume	Prenume	Departament
6150408019332	Ionescu	Ana	It
6120214078396	Popescu	Dan	IT

Selectați un chestionar:

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Managementul unei firme

[Angajati](#) [Proiecte](#) [Contact](#)

Adaugare angajat

CNP

23432

CNP INCORECT!!!

Nume

Popescu

Prenume

Dana

Departament

we

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Managementul unei firme

[Angajati](#) [Proiecte](#) [Contact](#)

Alocare proiect la angajat

CNP

1234567890987

Selecteaza proiect

Aplicatia2

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Managementul unei firme

[Angajati](#) [Proiecte](#) [Contact](#)

Chestionar

Salvează chestionar

Care sunt funcțiile implementate de dvs. in cadrul proiectului ?

adaugare

Pe o scara de la 1 la 5 care a fost dificultatea proiectului?

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

Care a fost intervalul de ore alocat pentru terminarea proiectului?

☒ 50-100 ☐ 100-300 ☐ 300-500

Care a fost limbajul de programare cel mai folosit?

☐ C++ ☐ C# ☒ Java

Nume control: functii Valoare: adaugare

Nume control: dificultate Valoare: 4

Nume control: implementare Valoare: 50-100

Nume control: limbaj Valoare: Java

Chirila Oana Sorina - Medii si tehnologii de programare - ASP.NET

Probleme propuse

La aplicația prezentată mai sus adăugați următoarele funcționalități:

- Crearea unui nou tabel "Log_proiecte" în care se pot adăuga orele lucrate în fiecare zi ale unui anumit angajat la un proiect alocat.
- Crearea unei pagini în care se pot introduce coordonatele privind lucrul angajatului la un anumit proiect
- Crearea unei pagini în care se pot vizualiza proiectele firmei
- Crearea unei pagini în care se poate vizualiza munca unui anumit angajat în luna respectivă și se pot calcula orele lucrate.