

Dezvoltarea Aplicatiilor Web-Anul 3

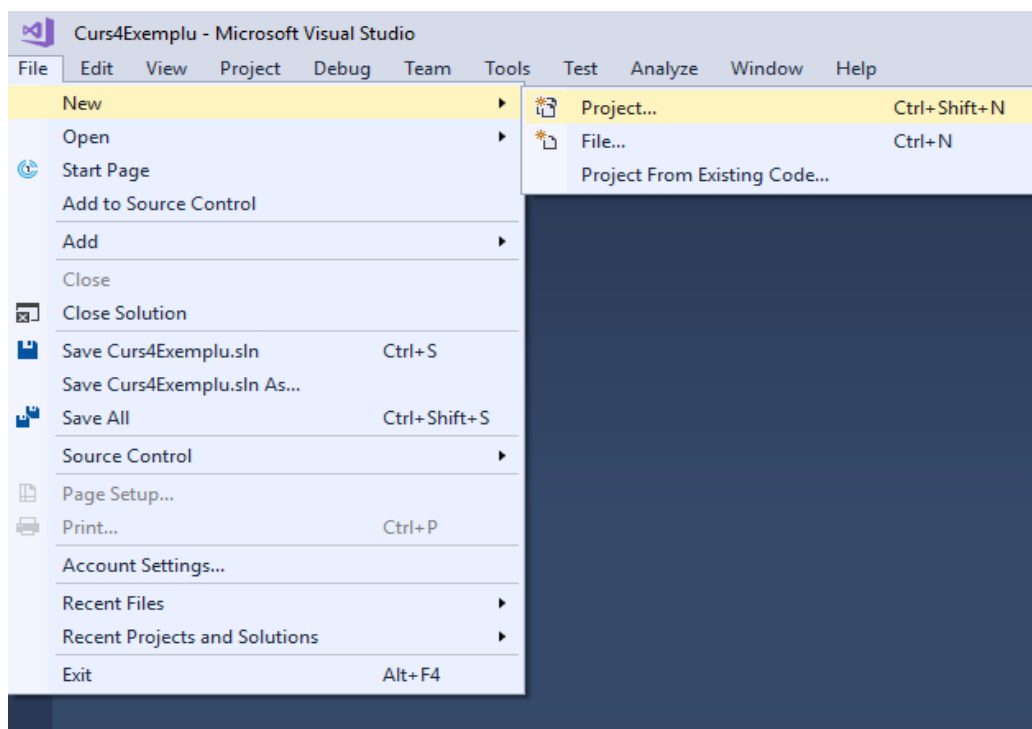
Curs 4

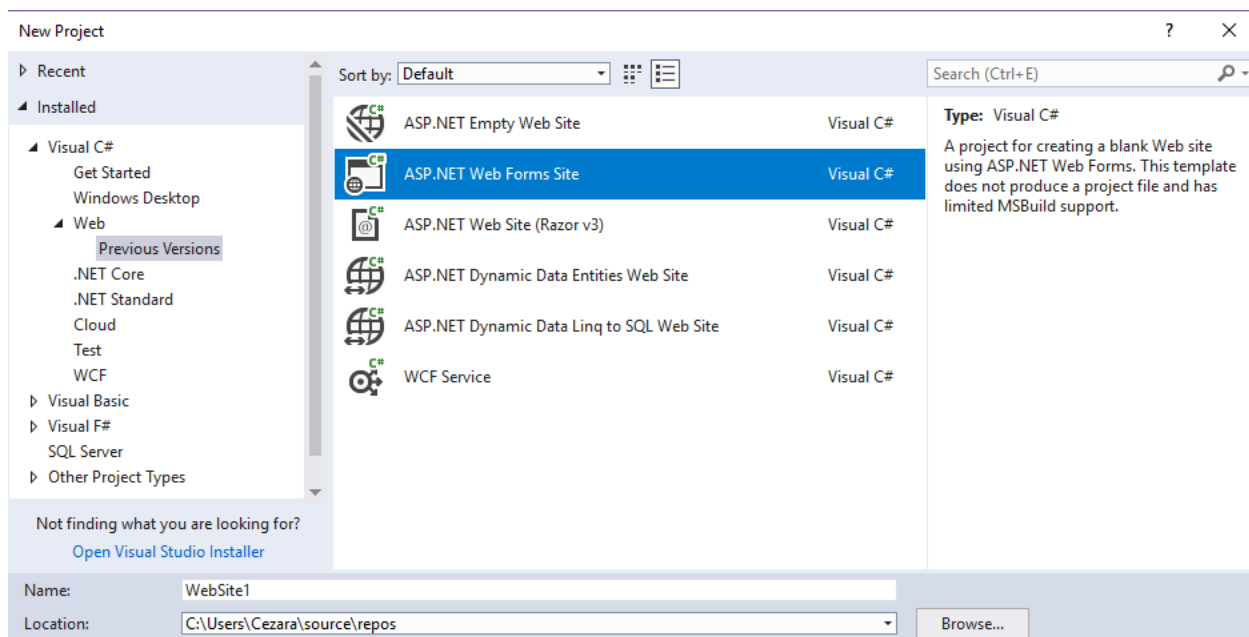
Crearea unui ASP.NET Web Site Project. Crearea folderelor si paginilor cu acces restrictiv. Autentificarea si crearea bazei de date. Atribuirea de roluri utilizatorilor. Modificarea regulilor de acces. Testarea aplicatiei. Sistemul de autentificare Identity. Identity Manager. Alocarea unui rol la inregistrare.

Crearea unui ASP.NET Web Site Project

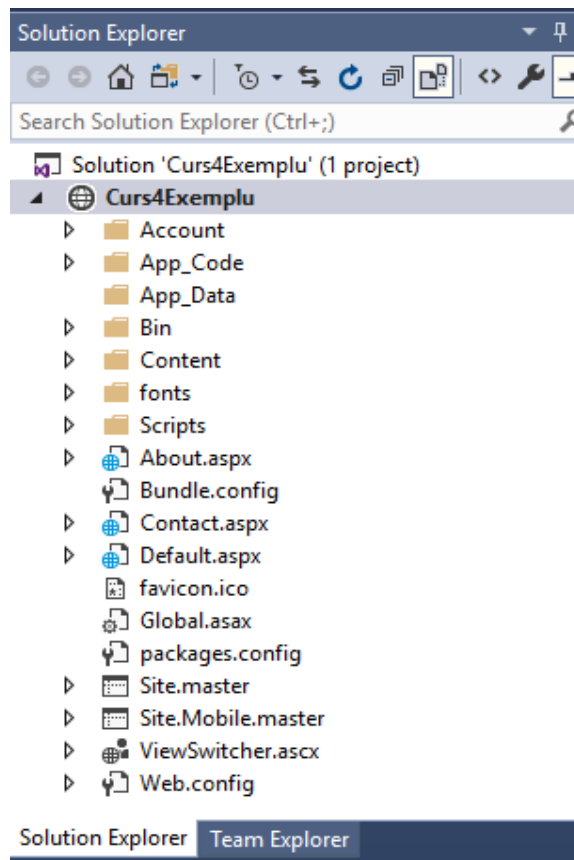
O sa folosim pentru crearea unui proiect nou optiunea **ASP.NET Web Forms Site**. In momentul in care utilizam aceasta optiune, proiectul va include elemente necesare si obligatorii pentru a crea o aplicatie care sa aiba un sistem de autentificare (membership).

Pasii pe care ii urmam pentru crearea acestui tip de proiect:



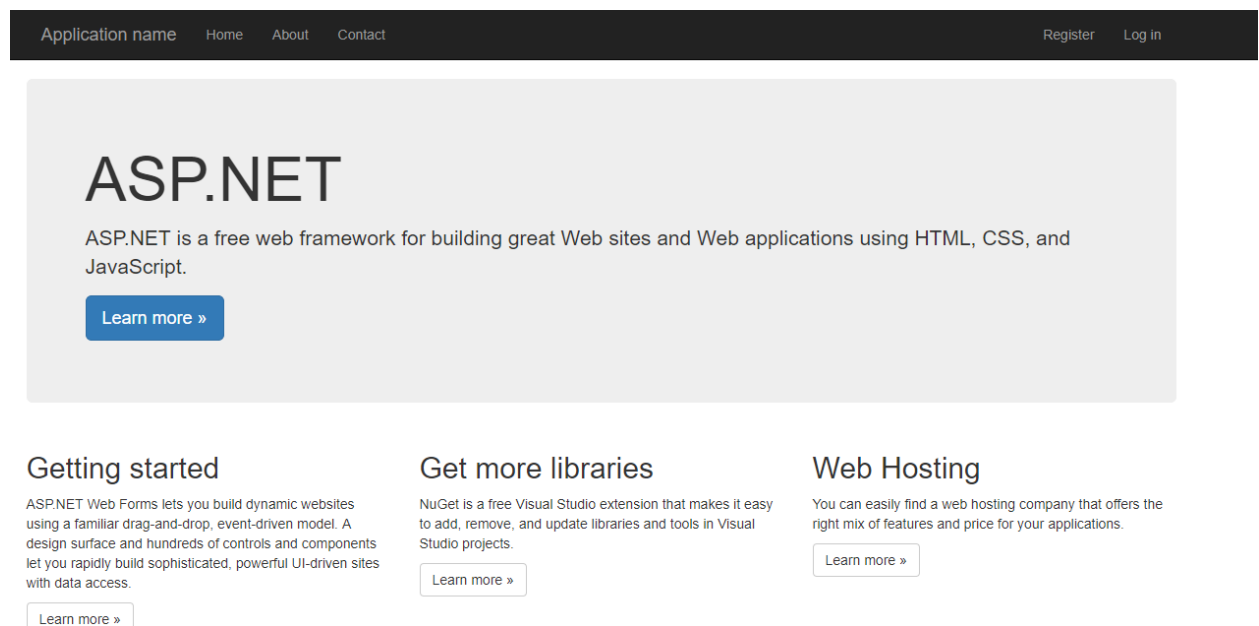


Se pot observa folderele, fisierele si paginile create:



Visual Studio creeaza automat paginile: **Default.aspx**, **About.aspx**, **Contact.aspx**, **Site.master**, fisierul **Web.config**, folderul **Account**, etc.

Dupa rularea paginii **Default.aspx**:



cezara@codesilk.com / www.cezarabenegui.com/DAW

Curs 4 – Dezvoltarea Aplicatiilor Web

Page 3

Pagina About

Application name Home About Contact

About.

Your application description page.

Use this area to provide additional information.

© 2018 - My ASP.NET Application

Pagina Contact

Application name Home About Contact

Contact.

Your contact page.

One Microsoft Way
Redmond, WA 98052-6399
P: 425.555.0100

Support: Support@example.com
Marketing: Marketing@example.com

© 2018 - My ASP.NET Application

Pagina Register

Application name Home About Contact

Register.

Create a new account.

User name

Password

Confirm password

Register

Pagina Login

Application name Home About Contact

Log in.

Use a local account to log in.

User name

Password

☐

Remember me?

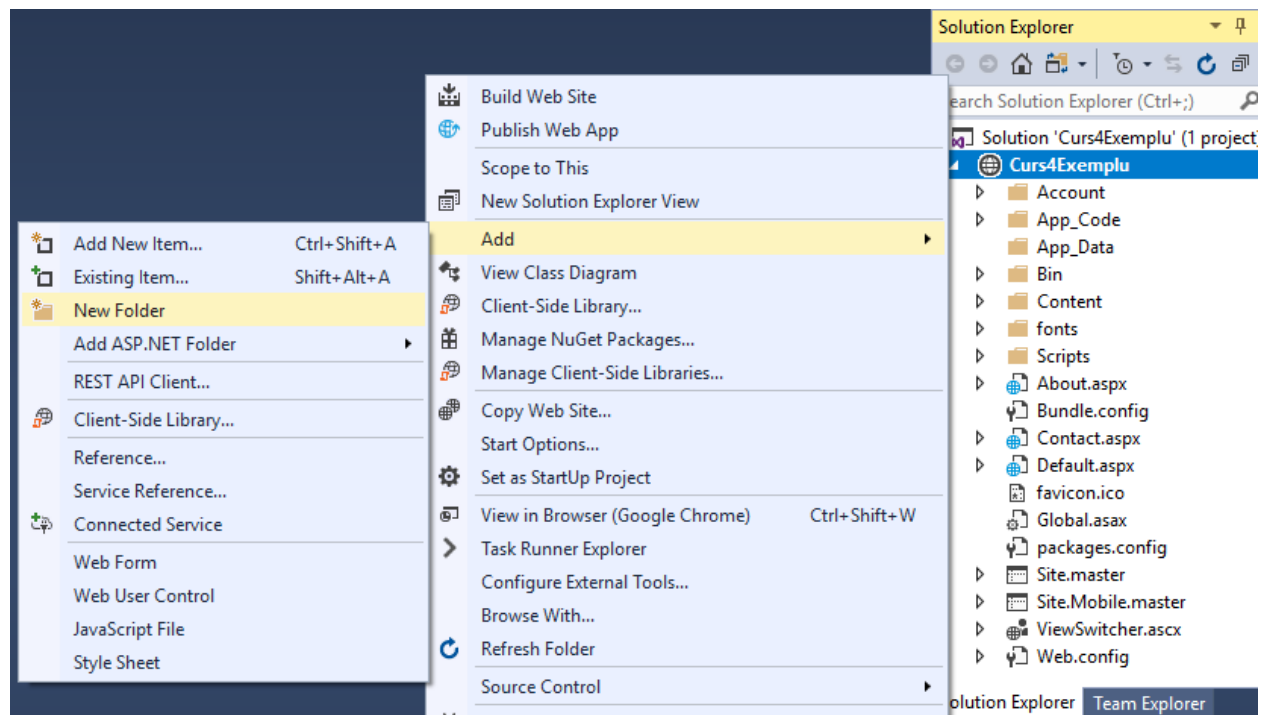
Log in

[Register](#) if you don't have a local account.

Crearea folderelor si paginilor cu acces restrictiv

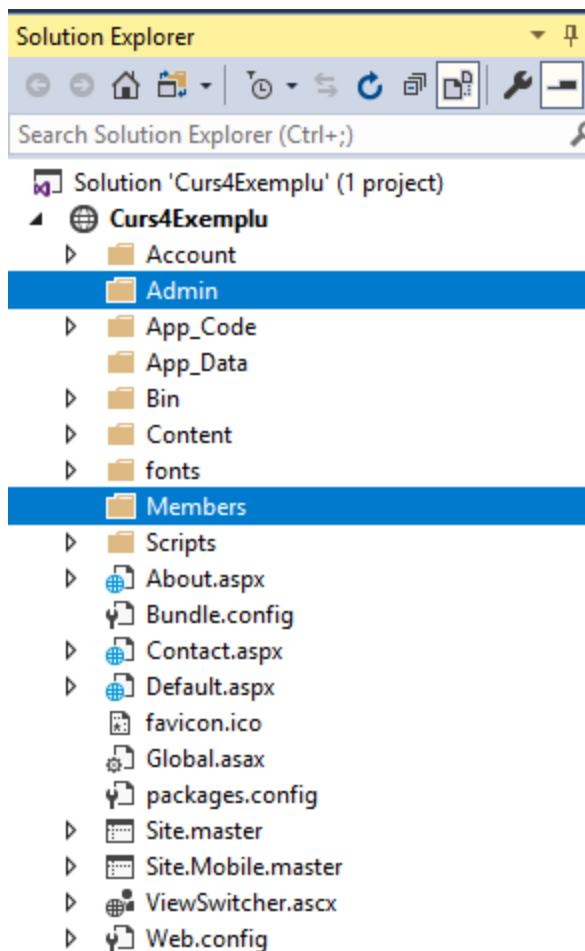
Pentru a exemplifica utilizarea rolurilor o sa cream doua foldere care o sa contina paginile cu acces restrictiv. Un folder va contine paginile care pot fi accesate doar de utilizatorii autentificati (acestea nu vor putea fi accesate de catre utilizatorii neautentificati – **anonymous users**). Al doilea folder va contine paginile care pot fi accesate de utilizatorii care au rol de administrator. Utilizatorii anonimi si utilizatorii care au cont, dar nu au rolul de administrator nu pot accesa paginile din al doilea folder. La final o sa aplicam regulile de acces acestor doua foldere pentru a specifica drepturile de acces pentru fiecare categorie de utilizatori.

Crearea folderelor:



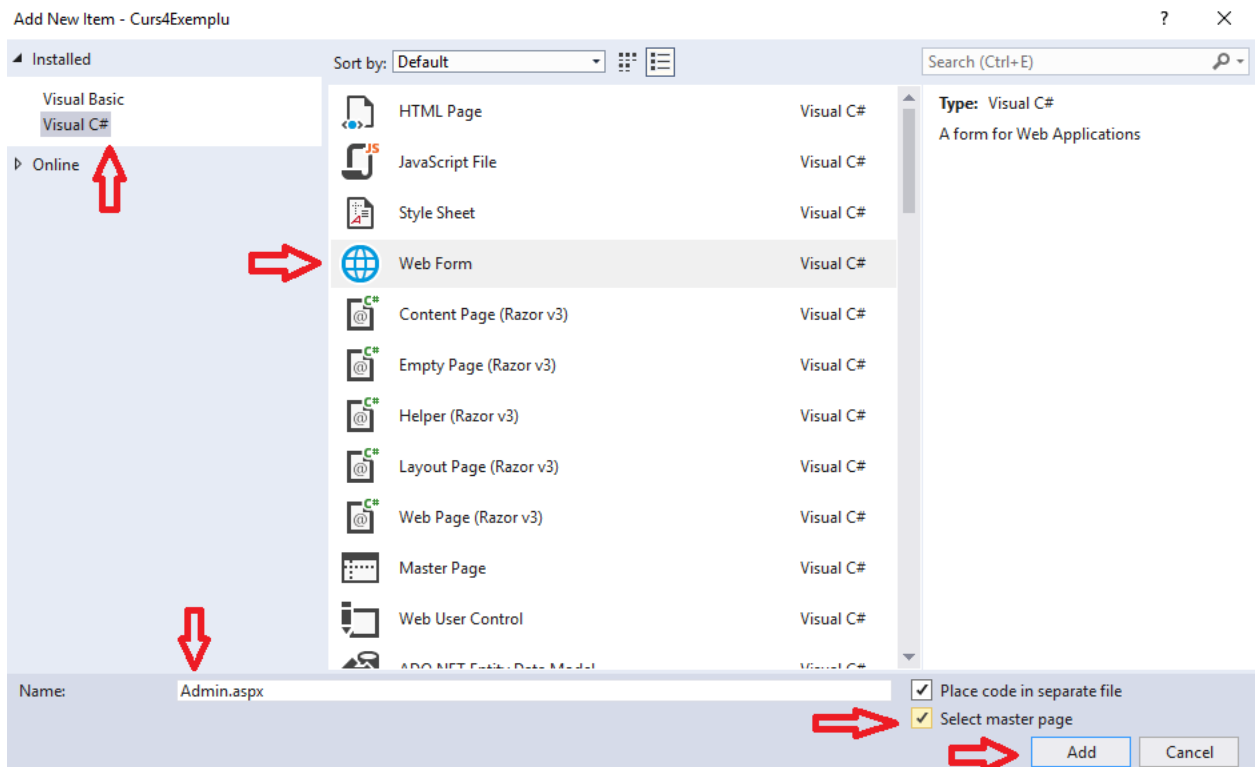
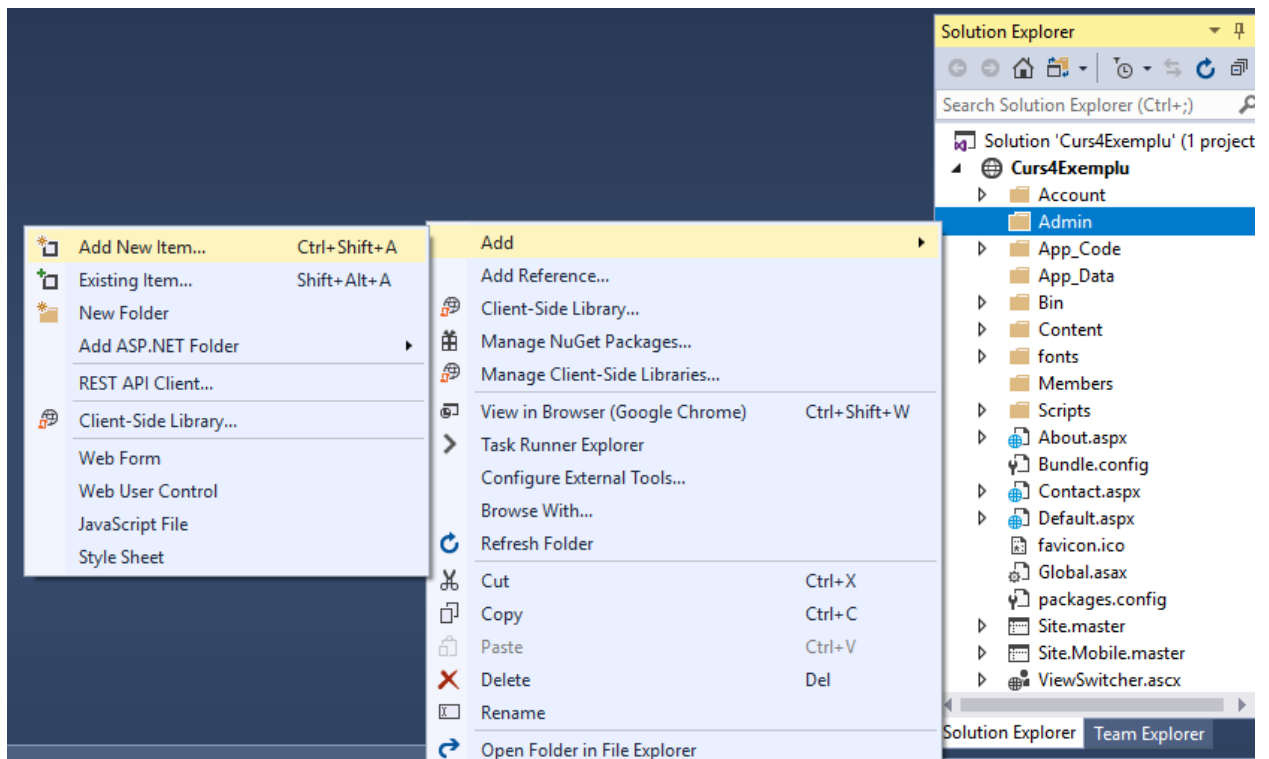
Folderul Admin o sa contina paginile accesibile doar utilizatorilor care vor fi asignati rolului de Admin.

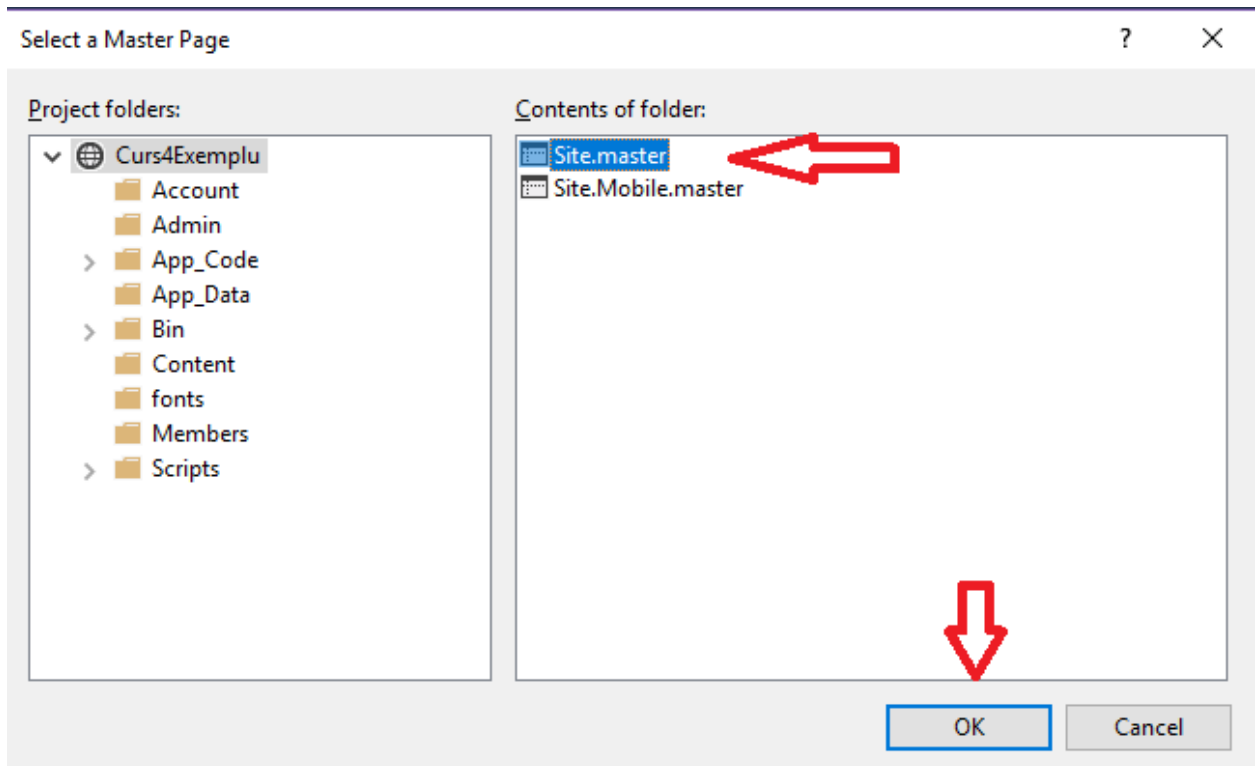
Folderul Members o sa contina paginile accesibile tuturor utilizatorilor logati.



Adaugarea paginilor in folderele create:

Se adauga pagina Admin.aspx in folderul Admin si pagina Members.aspx in folderul Members.





```
Server Explorer  Toolbox
Admin.aspx.cs  Admin.aspx
1  <%@ Page Title="" Language="C#" MasterPageFile="~/
   Site.master" AutoEventWireup="true"
   CodeFile="Admin.aspx.cs" Inherits="Admin_Admin" %>
2
3  <asp:Content ID="Content1"
   ContentPlaceHolderID="MainContent" Runat="Server">
4      <h1>Welcome to the administrators page</h1>
5  </asp:Content>
```

Se procedeaza in acelasi mod si pentru adaugarea paginii Members.aspx

Se creeaza legaturi catre paginile create (Admin, Members):

In pagina Master – **Site.master** – se adauga legaturile catre noile pagini adaugate, Admin si Members.



```
57         </ul>
58         <asp:LoginView runat="server" ViewStateMode="Disabled">
59             <AnonymousTemplate>
60                 <ul class="nav navbar-nav navbar-right">
61                     <li><a runat="server" href="~/Account/Register">Register</a></li>
62                     <li><a runat="server" href="~/Account/Login">Log in</a></li>
63                 </ul>
64             </AnonymousTemplate>
65             <LoggedInTemplate>
66                 <ul class="nav navbar-nav navbar-right">
67                     <li><a runat="server" href="~/Account/Manage" title="Manage your
68 account">Hello, %!: Context.User.Identity.GetUserName() %!</a></li>
69                     <li><a runat="server" href="~/Admin/Admin.aspx">Admin</a></li>
70                     <li><a runat="server" href="~/Members/Members.aspx">Members</a></li>
71                 </ul>
72             </LoggedInTemplate>
73         </asp:LoginView>
```

Controlul **LoginView** afiseaza anumite secvente de cod in functie de starea utilizatorilor (autentificati/neautentificati) si in functie de rolul acestora.

In cadrul elementului **LoginView** codul trebuie scris utilizand una sau mai multe din urmatoarele proprietati:

- **AnonymousTemplate** – este folosit pentru a afisa continutul corespunzator utilizatorilor neautentificati. Utilizatorii autentificati nu pot vedea continutul acestui template
- **LoggedInTemplate** – afiseaza continut utilizatorilor care sunt autentificati, dar care nu au niciun rol atribuit
- **RoleGroups** – afiseaza continutul utilizatorilor autentificati care sunt membrii unui rol existent

```
1 <%@ Page Title="" Language="C#" MasterPageFile="~/Site.master"
  AutoEventWireup="true" CodeFile="Admin.aspx.cs"
  Inherits="Admin_Admin" %>
2
3 <asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
  Runat="Server">
4     <h1>Welcome to the administrators page</h1>
5     <asp:LoginView runat="server">
6
7         <h1>Admin Page</h1>
8
9     </asp:LoginView>
10 </asp:Content>
```

Validation (HTML5): The element 'h1' cannot be nested within the element 'loginview'.

Se pot observa sugestiile oferite in momentul in care folosim un control LoginView

```
1 <%@ Page Title="" Language="C#" MasterPageFile="~/Site.master"
  AutoEventWireup="true" CodeFile="Admin.aspx.cs"
  Inherits="Admin_Admin" %>
2
3 <asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
  Runat="Server">
4     <h1>Welcome to the administrators page</h1>
5     <asp:LoginView runat="server">
6
7         <
8
9     </asp:
10 </asp:Content>
```

AnonymousTemplate
LoggedInTemplate
RoleGroups

Autentificarea si crearea bazei de date

In momentul inregistrarii in aplicatie framework-ul genereaza in mod automat baza de date care contine informatii referitoare la utilizatori, roluri si asocierile dintre acestea.

Application name Home About Contact

Register.

Create a new account.

User name

user

Password

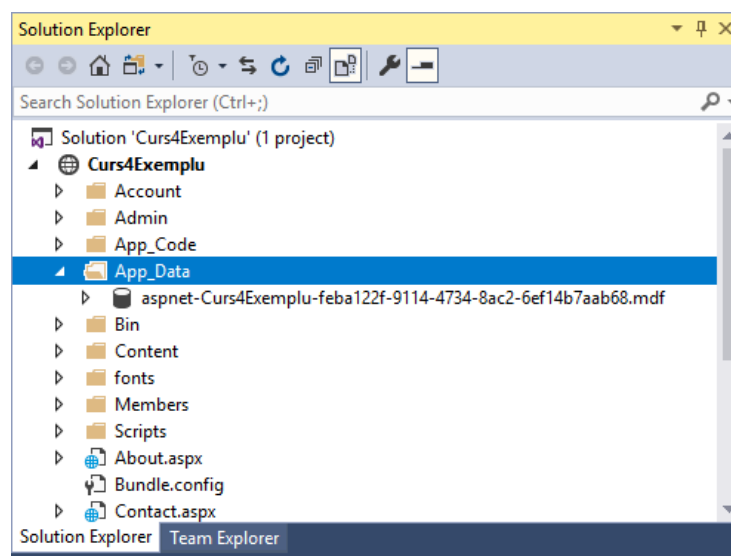
.....

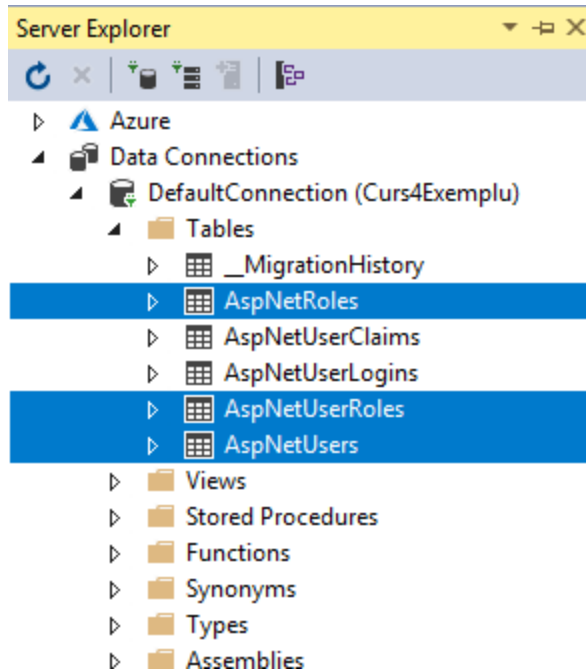
Confirm password

.....

Register

Dupa apasarea butonului “Register”, folderul App_Data va contine baza de date.

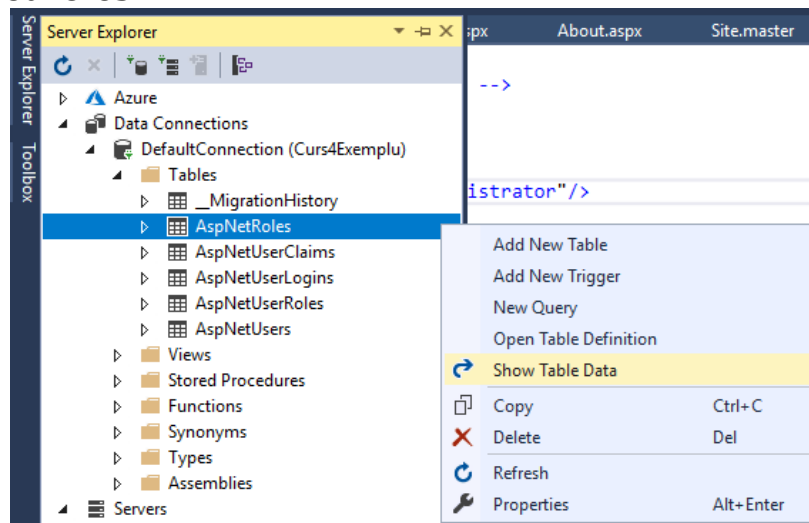




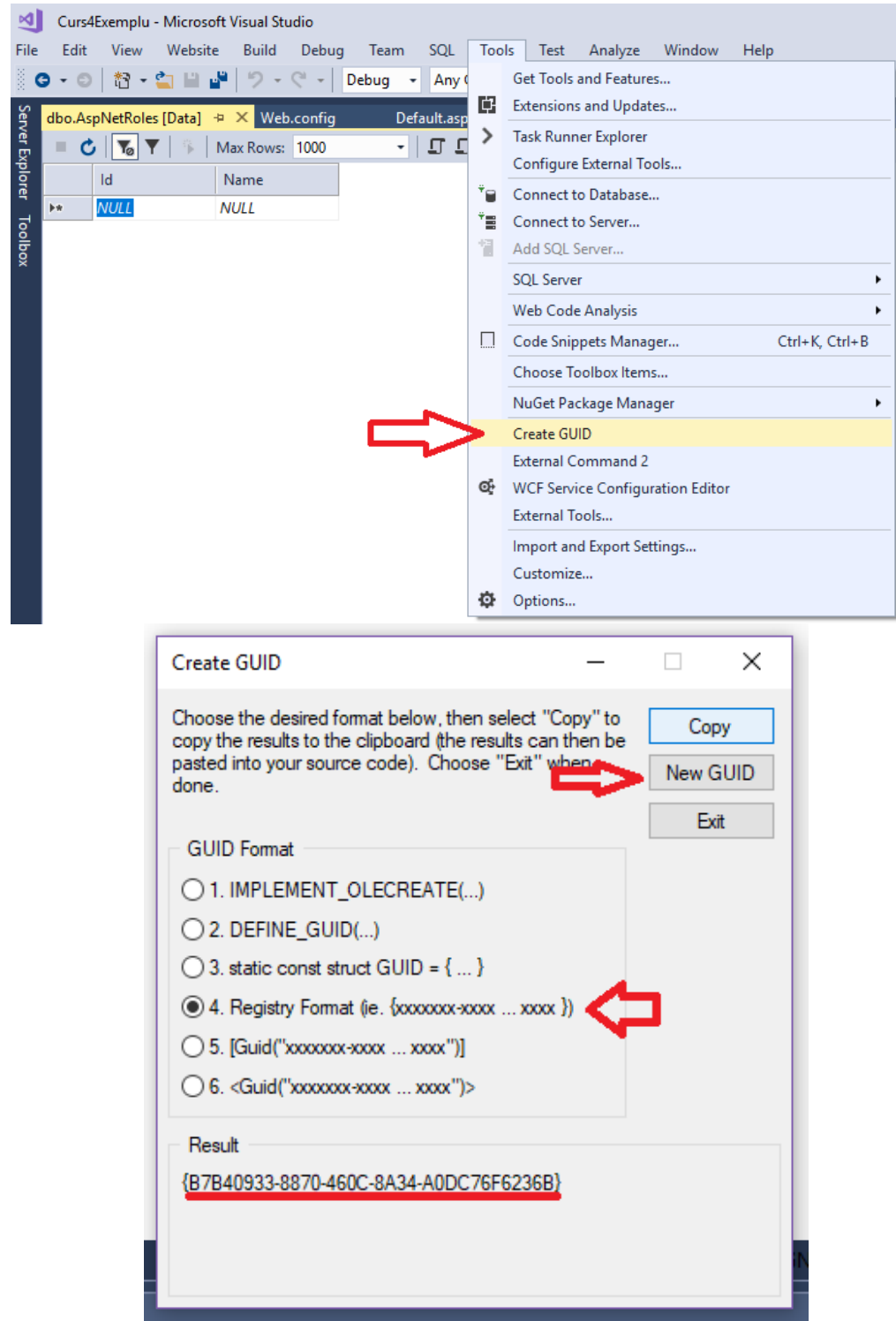
Atribuirea de roluri utilizatorilor

Pentru a configura in mod corect aplicatia, trebuie sa populam baza de date cu rolurile necesare acesteia (in acest exemplu Administrator si Membru).

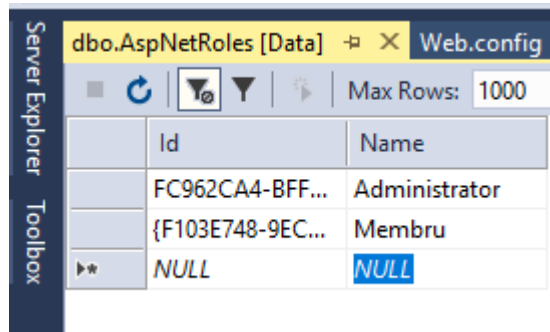
Acest lucru se face introducand informatiile in tabelul **AspNetRoles**.



Tabelele generate utilizeaza pentru cheia primara ID un string sub forma GUID (Globally Unique Identifier). Aceste string-uri pot fi generate din IDE folosind din meniul **Tools** optiunea **Create GUID**:

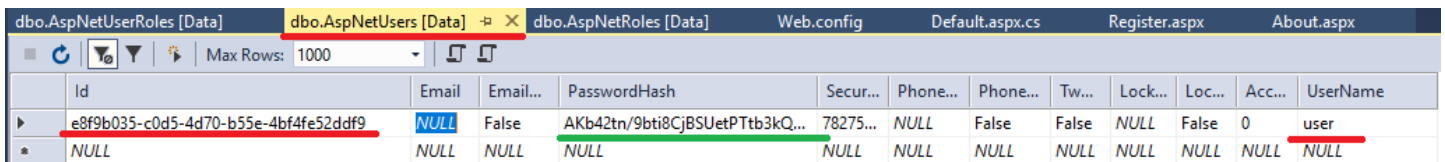


Folosind GUID-ul nou generat introducem datele in tabel pentru fiecare rol necesar aplicatiei.

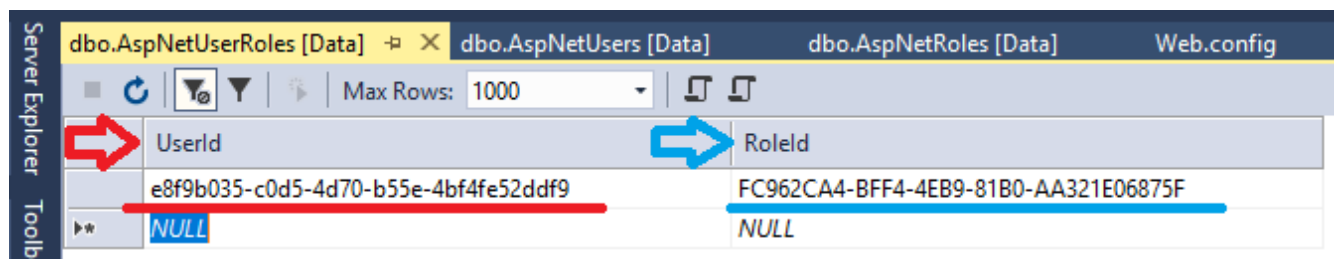


Id	Name
FC962CA4-BFF...	Administrator
{F103E748-9EC...	Membru
NULL	NULL

Urmatorul pas pentru a finaliza sistemul de roluri este asocierea utilizatorilor cu unul dintre rolurile stabilite. Acest lucru se face prin identificarea ID-ului utilizatorului si ID-ului rolului cu care acesta va fi asociat si introducerea acestor detalii in tabelul asociativ **AspNetUserRoles**



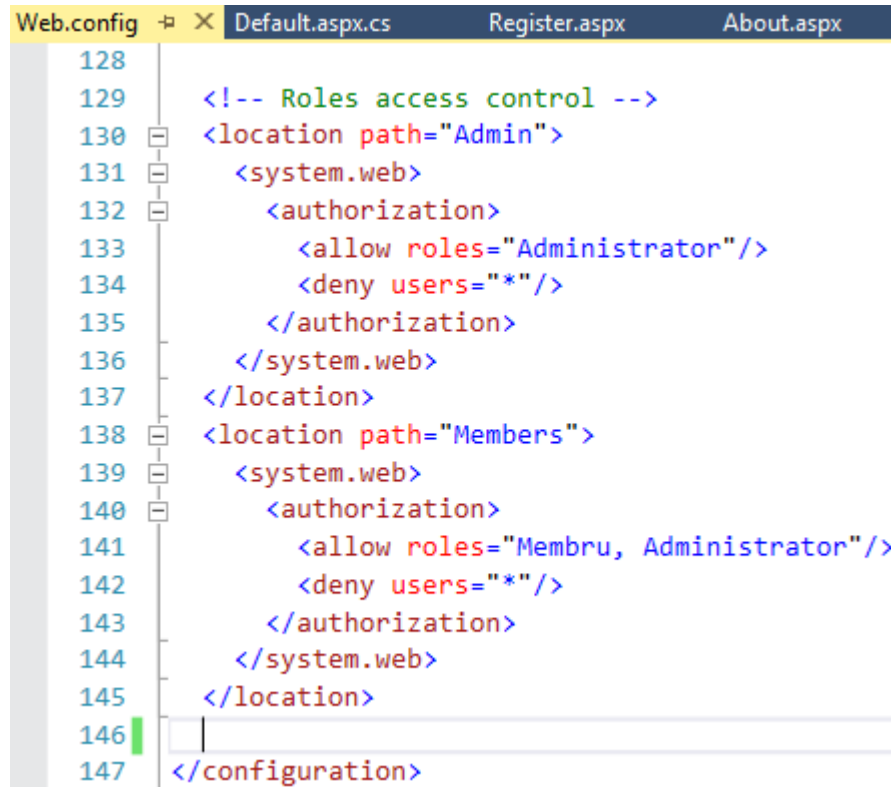
Id	Email	Email...	PasswordHash	Secur...	Phone...	Phone...	Tw...	Lock...	Loc...	Acc...	UserName
e8f9b035-c0d5-4d70-b55e-4bf4fe52ddf9	NULL	False	AKb42tn/9bti8CjBSUetPTtb3kQ...	78275...	NULL	False	False	NULL	False	0	user
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



UserId	RoleId
e8f9b035-c0d5-4d70-b55e-4bf4fe52ddf9	FC962CA4-BFF4-4EB9-81B0-AA321E06875F
NULL	NULL

Modificarea regulilor de acces

Pentru a restrictiona accesul la anumite pagini sau sectiuni ale aplicatiei in functie de rolul utilizatorului, in fisierul **Web.config**, in cadrul tag-ului **<configuration>** trebuie sa definim regulile necesare.

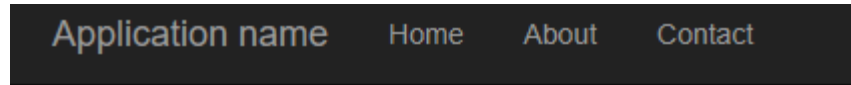


The screenshot shows a code editor with four tabs: Web.config, Default.aspx.cs, Register.aspx, and About.aspx. The Web.config tab is active, displaying XML code for roles access control. The code is as follows:

```
128
129 <!-- Roles access control -->
130 <location path="Admin">
131   <system.web>
132     <authorization>
133       <allow roles="Administrator"/>
134       <deny users="*/>
135     </authorization>
136   </system.web>
137 </location>
138 <location path="Members">
139   <system.web>
140     <authorization>
141       <allow roles="Membru, Administrator"/>
142       <deny users="*/>
143     </authorization>
144   </system.web>
145 </location>
146
147 </configuration>
```


Testarea aplicatiei

Autentificare ca Membru



Log in.

Use a local account to log in.

User name

membru

Password

.....

☐

Remember me?

Log in

[Register](#) if you don't have a local account.



Welcome to Home Page

ASP.NET is a free web framework for building great Web sites and Web applications using HTML, CSS, and JavaScript.

[Learn more »](#)

Welcome to the Members page

© 2018 - My ASP.NET Application

Autentificare ca Administrator

Log in.

Use a local account to log in.

User name

user



Password

.....



Remember me?

Log in

Welcome to the administrators page

© 2018 - My ASP.NET Application

Sistemul de autentificare Identity

Sistemul de autentificare prezentat mai sus se numeste “Identity” si inlocuieste sistemul vechi de autentificare ASP.NET numit “Membership”. Acest mecanism a fost implementat incepand cu versiunea Visual Studio 2013 in defavoarea “Membership”.

Pe langa autentificarea folosind baza de date, prezentata in acest curs, Identity ofera posibilitatea autentificarii cu un serviciu extern cum ar fi: Facebook, Google, Microsoft si Twitter.

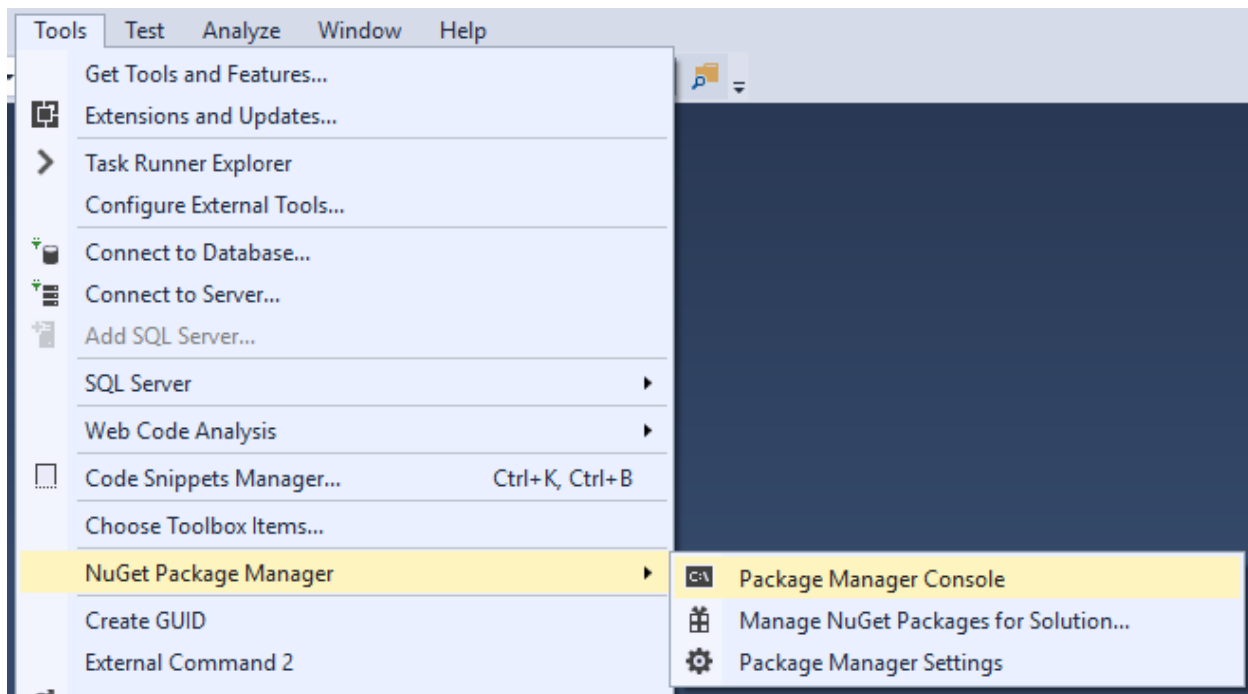
Prin crearea unui nou WebSite (**File > New > Project > ASP.NET Web Forms Site**) in Visual Studio 2017, mecanismul de autentificare este inclus in proiect in mod automat.

Identity Manager

In cadrul Visual Studio 2010, pentru sistemul de autentificare Membership IDE-ul ofera posibilitatea de a face management-ul rolurilor, utilizatorilor si asocierile dintre acestea folosind un panou de administrare web.

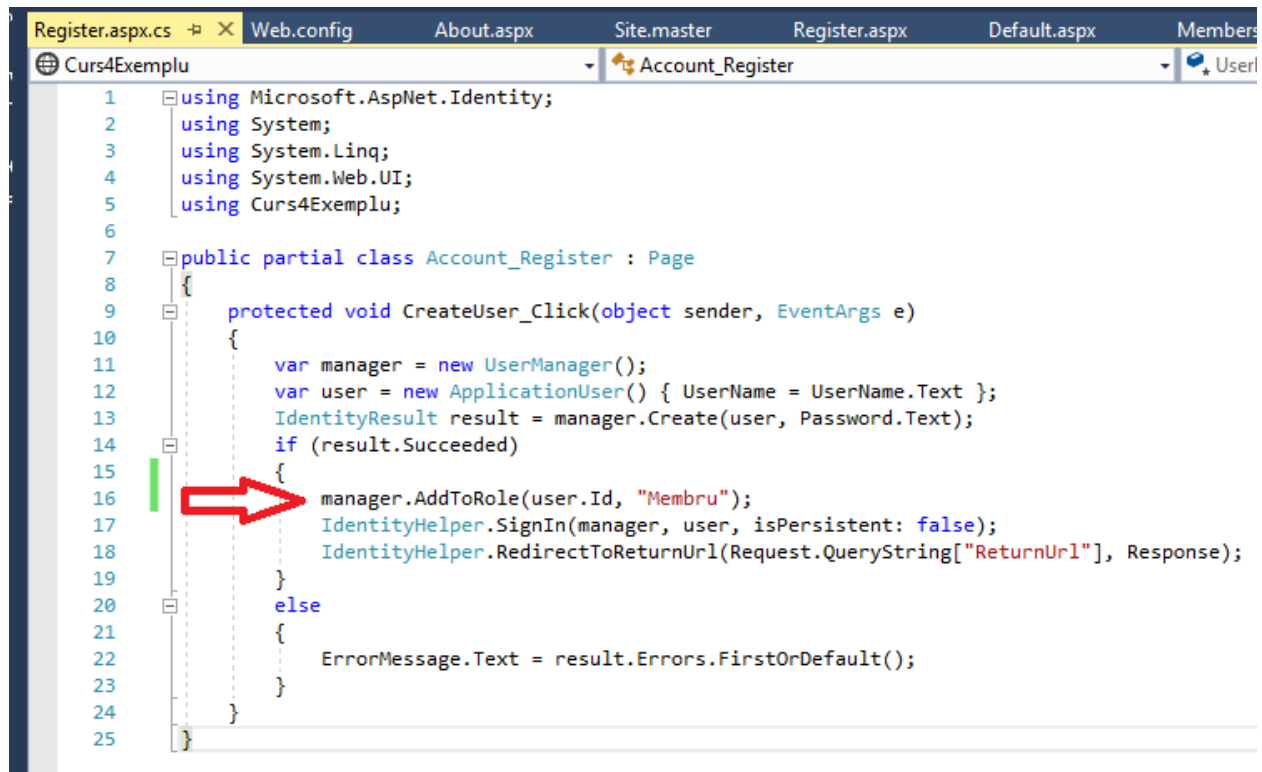
Versiunile de Visual Studio incepand cu 2013 au eliminat acest pachet deoarece nu este compatibil cu sistemul Identity. Pentru a instala un astfel de sistem, exista posibilitatea utilizarii unui pachet extern precum “Identity Manager”.

Pachetele se pot instala in Visual Studio astfel:



Alocarea unui rol la inregistrare

Pentru a alocă în mod automat un rol utilizatorilor la autentificare, se poate folosi următoarea secvență de cod în fișierul Register.aspx



```
1 using Microsoft.AspNet.Identity;
2 using System;
3 using System.Linq;
4 using System.Web.UI;
5 using Curs4Exemplu;
6
7 public partial class Account_Register : Page
8 {
9     protected void CreateUser_Click(object sender, EventArgs e)
10    {
11        var manager = new UserManager();
12        var user = new ApplicationUser() { UserName = UserName.Text };
13        IdentityResult result = manager.Create(user, Password.Text);
14        if (result.Succeeded)
15        {
16            manager.AddToRole(user.Id, "Membru");
17            IdentityHelper.SignIn(manager, user, isPersistent: false);
18            IdentityHelper.RedirectToReturnUrl(Request.QueryString["ReturnUrl"], Response);
19        }
20        else
21        {
22            ErrorMessage.Text = result.Errors.FirstOrDefault();
23        }
24    }
25 }
```