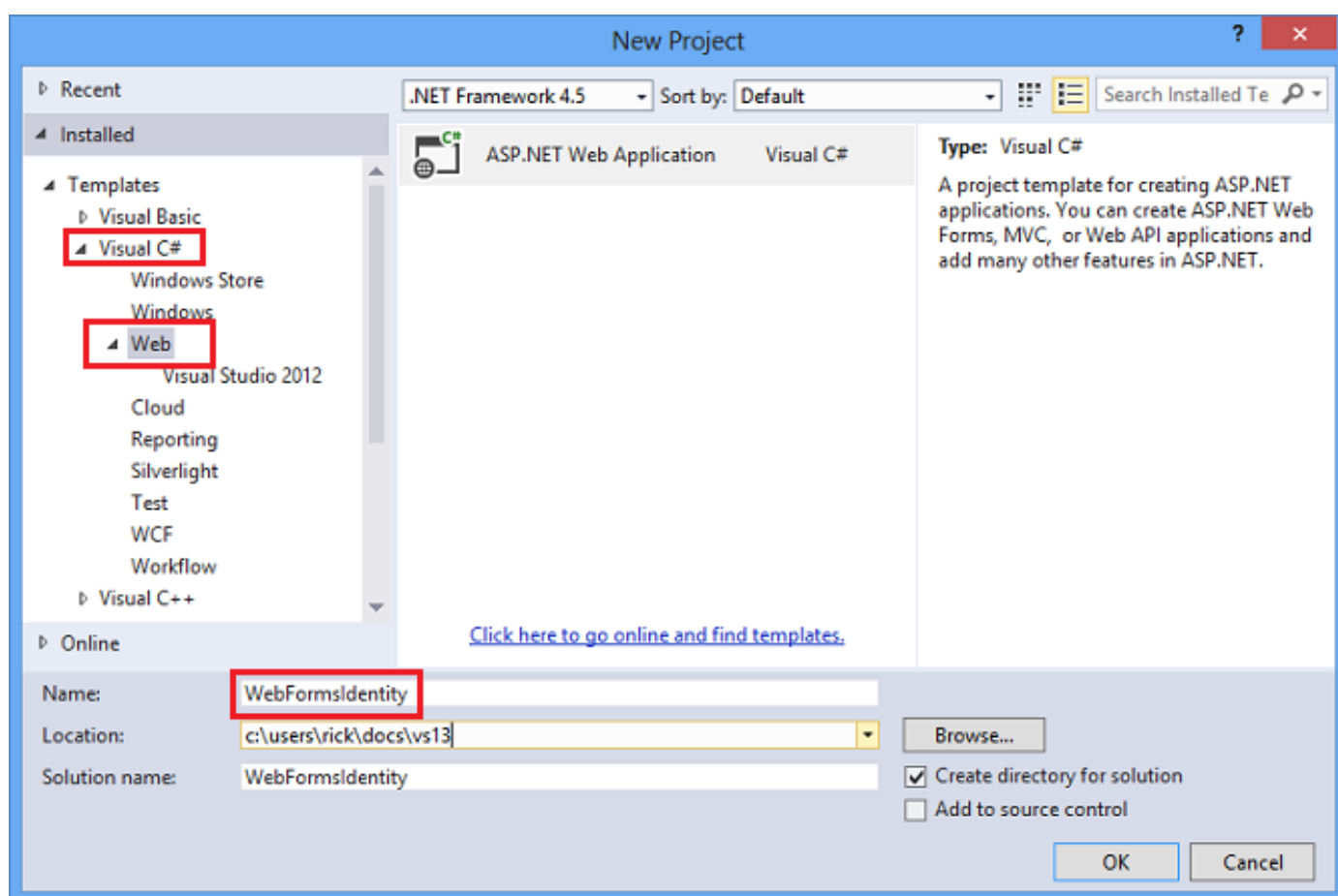


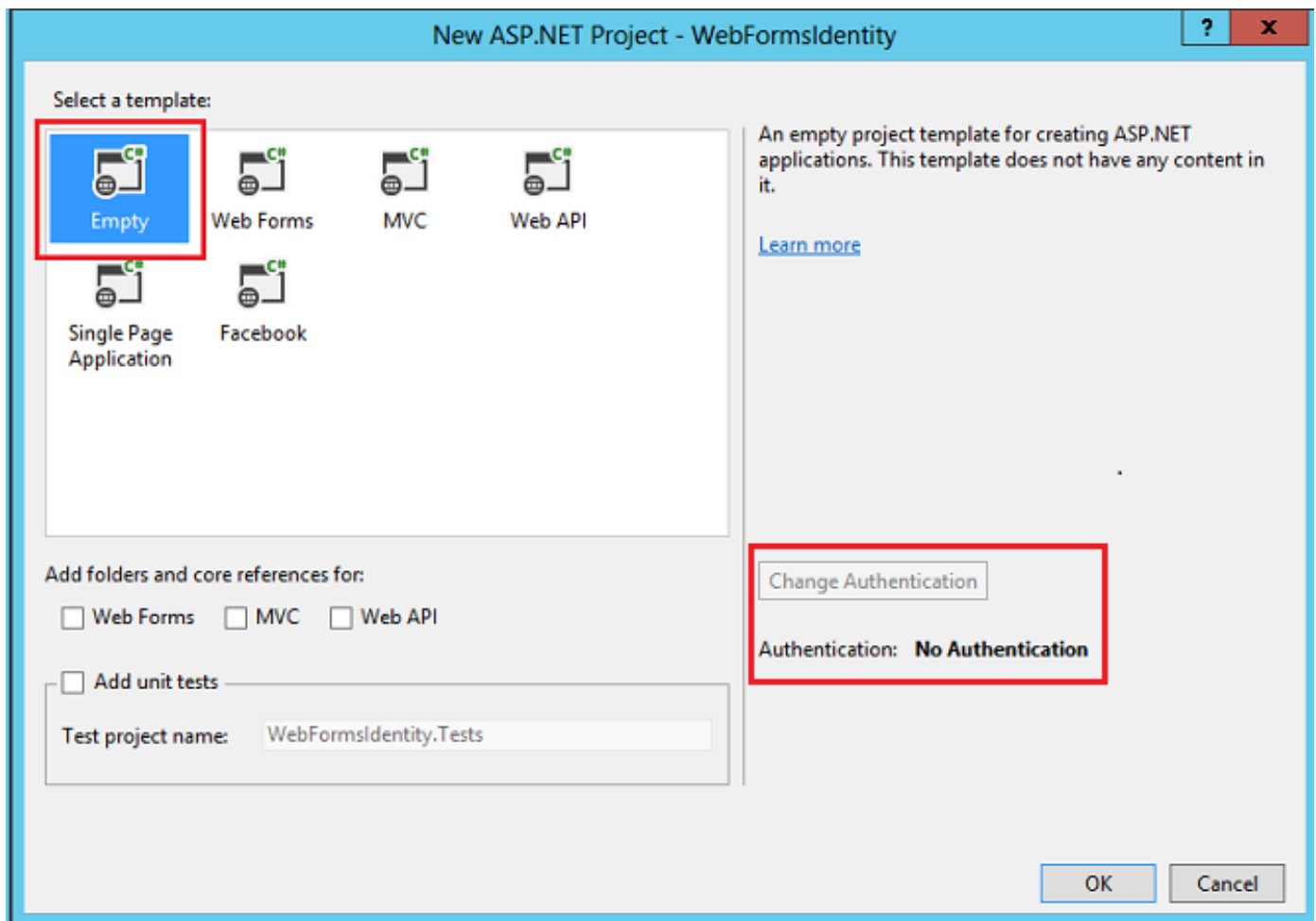
با نصب و اجرای [Visual Studio 2013 Express for Web](#) یا [Visual Studio 2013](#) شروع کنید.

یک پروژه جدید بسازید (از صفحه شروع یا منوی فایل)

گزینه Visual C# و سپس **ASP.NET Web Application** را انتخاب کنید. نام پروژه را به "WebFormsIdentity" تغییر داده و OK کنید.



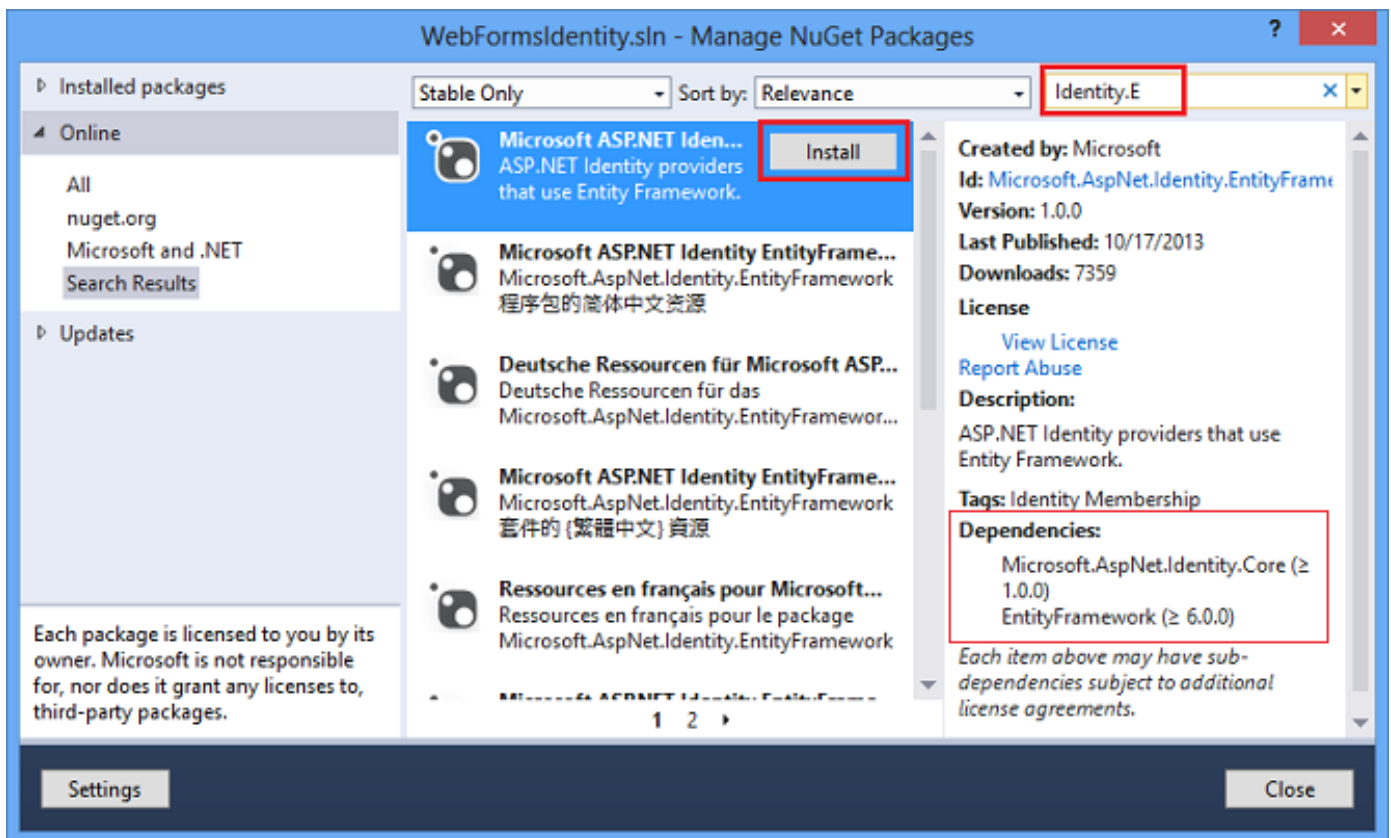
در دیالوگ جدید ASP.NET گزینه **Empty** را انتخاب کنید.



دقت کنید که دکمه **Change Authentication** غیرفعال است و هیچ پشتیبانی‌ای برای احراز هویت در این قالب پروژه وجود ندارد.

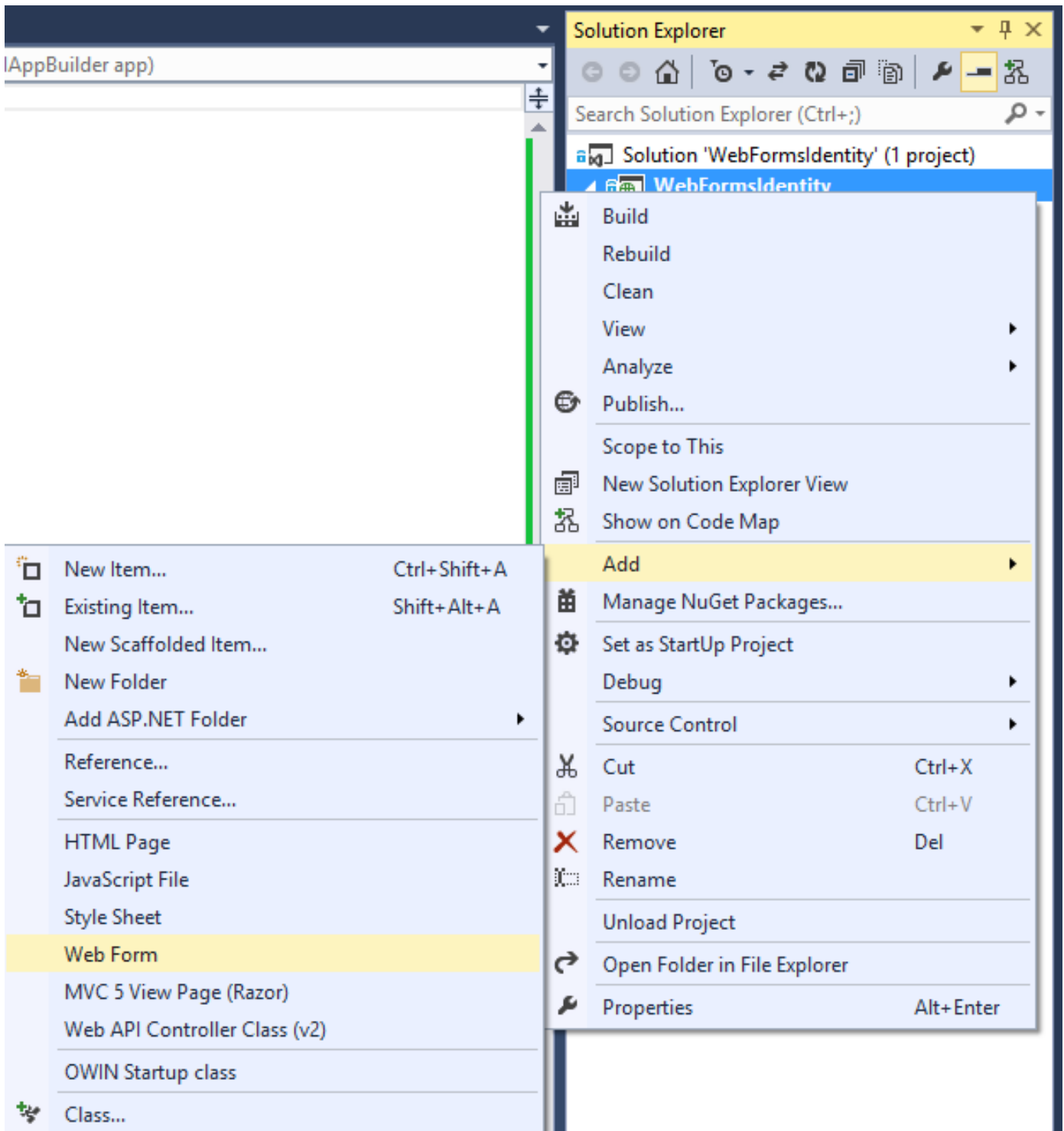
افزودن پکیج‌های ASP.NET Identity به پروژه

روی نام پروژه کلیک راست کنید و گزینه **Manage NuGet Packages** را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "Identity.E" را وارد کرده و این پکیج را نصب کنید.



دقت کنید که نصب کردن این پکیج وابستگی‌ها را نیز بصورت خودکار نصب می‌کند: Entity Framework و ASP.NET Identity Core.

افزودن فرم‌های وب لازم برای ثبت نام کاربران
یک فرم وب جدید بسازید.



در دیالوگ باز شده نام فرم را به **Register** تغییر داده و تایید کنید.

فایل ایجاد شده جدید را باز کرده و کد Markup آن را با قطعه کد زیر جایگزین کنید.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Register.aspx.cs"
Inherits="WebFormsIdentity.Register" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
```

```

</head>
<body style="
  <form id="form1" runat="server">
    <div>
      <h4 style="Register a new user</h4>
      <hr />
      <p>
        <asp:Literal runat="server" ID="StatusMessage" />
      </p>
      <div style="margin-bottom:10px">
        <asp:Label runat="server" AssociatedControlID="UserName">User name</asp:Label>
        <div>
          <asp:TextBox runat="server" ID="UserName" />
        </div>
      </div>
      <div style="margin-bottom:10px">
        <asp:Label runat="server" AssociatedControlID="Password">Password</asp:Label>
        <div>
          <asp:TextBox runat="server" ID="Password" TextMode="Password" />
        </div>
      </div>
      <div style="margin-bottom:10px">
        <asp:Label runat="server" AssociatedControlID="ConfirmPassword">Confirm
password</asp:Label>
        <div>
          <asp:TextBox runat="server" ID="ConfirmPassword" TextMode="Password" />
        </div>
      </div>
      <div>
        <div>
          <asp:Button runat="server" OnClick="CreateUser_Click" Text="Register" />
        </div>
      </div>
    </div>
  </form>
</body>
</html>

```

این تنها یک نسخه ساده شده *Register.aspx* است که از چند فیلد فرم و دکمه ای برای ارسال آنها به سرور استفاده می‌کند.

فایل کد این فرم را باز کرده و محتویات آن را با قطعه کد زیر جایگزین کنید.

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using System;
using System.Linq;

namespace WebFormsIdentity
{
  public partial class Register : System.Web.UI.Page
  {
    protected void CreateUser_Click(object sender, EventArgs e)
    {
      // Default UserStore constructor uses the default connection string named: DefaultConnection
      var userStore = new UserStore<IdentityUser>();
      var manager = new UserManager<IdentityUser>(userStore);

      var user = new IdentityUser() { UserName = UserName.Text };
      IdentityResult result = manager.Create(user, Password.Text);

      if (result.Succeeded)
      {
        StatusMessage.Text = string.Format("User {0} was created successfully!", user.UserName);
      }
      else
      {
        StatusMessage.Text = result.Errors.FirstOrDefault();
      }
    }
  }
}

```

کد این فرم نیز نسخه ای ساده شده است. فایلی که بصورت خودکار توسط VS برای شما ایجاد می‌شود متفاوت است.

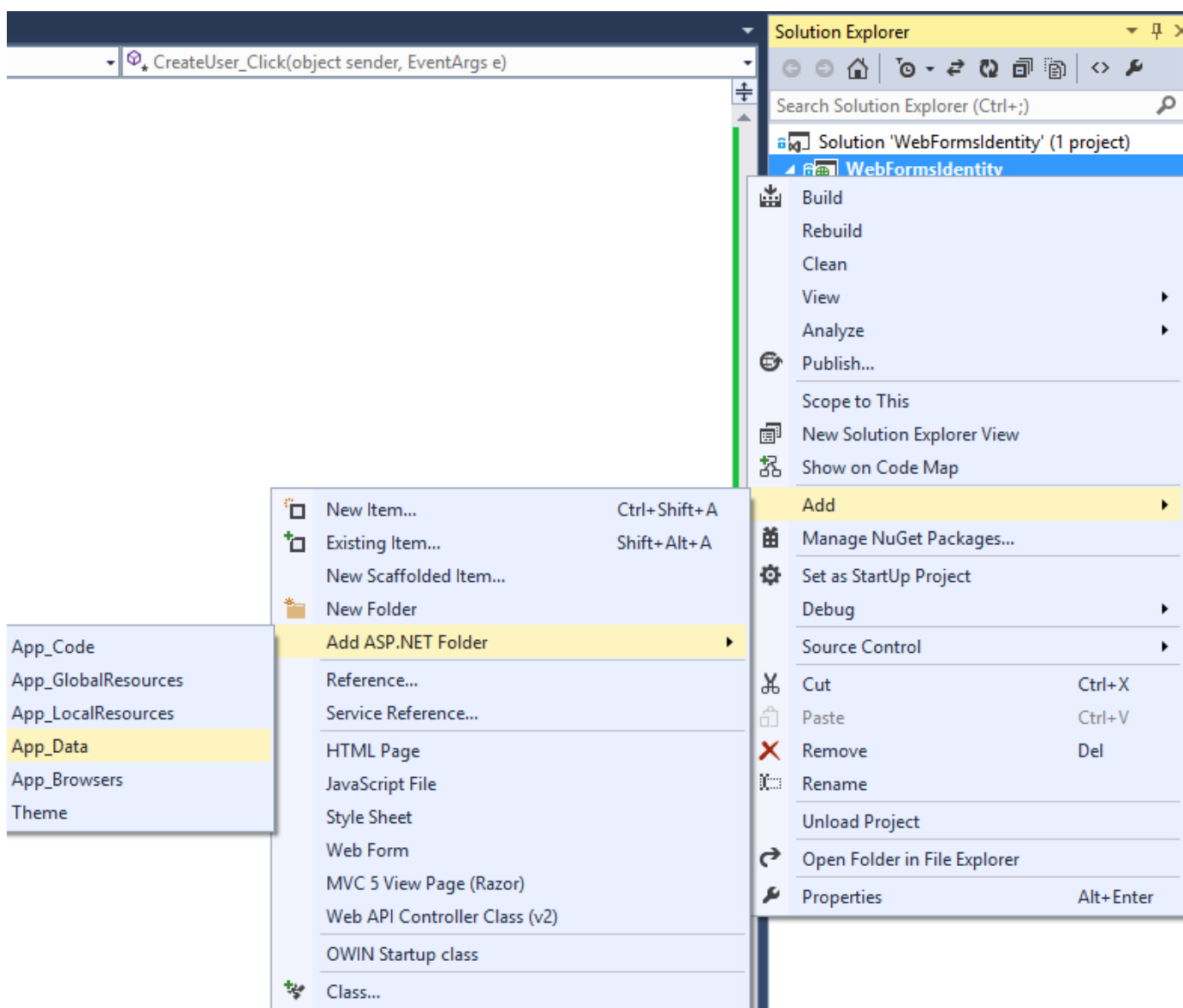
کلاس *IdentityUser* پیاده سازی پیش فرض EntityFramework از قرارداد IUser است. قرارداد IUser تعاریفات حداقلی یک کاربر در ASP.NET Identity Core را در بر می گیرد.

کلاس *UserStore* پیاده سازی پیش فرض EF از یک فروشگاه کاربر (user store) است. این کلاس چند قرارداد اساسی ASP.NET Identity Core را پیاده سازی می کند: *IUserStore*, *IUserRoleStore*, *IUserLoginStore*, *IUserClaimStore*.

کلاس *userManager* دسترسی به API های مربوط به کاربران را فراهم می کند. این کلاس تمامی تغییرات را بصورت خودکار در *UserStore* ذخیره می کند.

کلاس *IdentityResult* نتیجه یک عملیات هویتی را معرفی می کند (identity operations).

پوشه **App_Data** را به پروژه خود اضافه کنید.



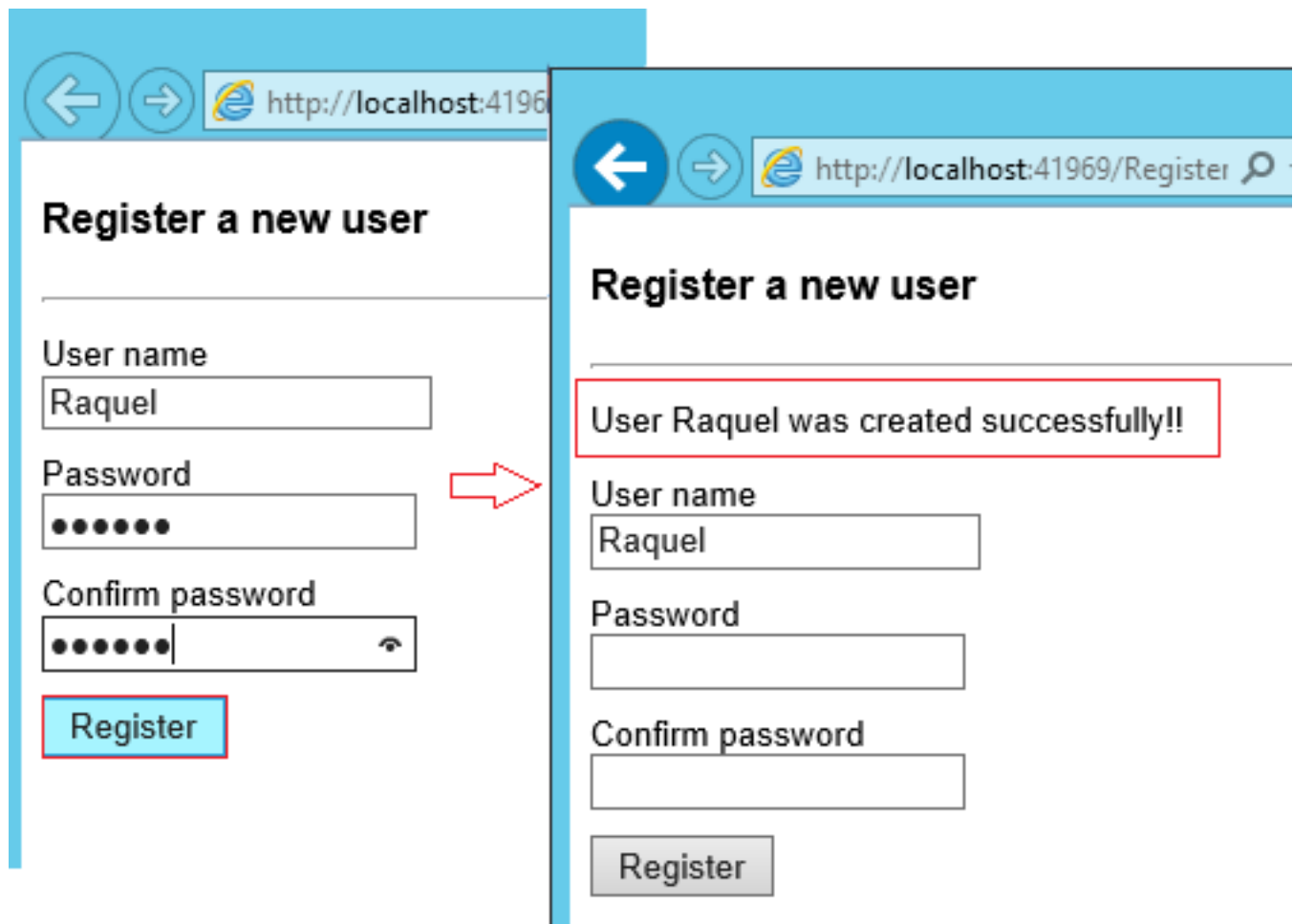
فایل *Web.config* پروژه را باز کنید و رشته اتصال جدیدی برای دیتابیس اطلاعات کاربران اضافه کنید. این دیتابیس در زمان اجرا (runtime) بصورت خودکار توسط EF ساخته می شود. این رشته اتصال شبیه به رشته اتصالی است که هنگام ایجاد پروژه بصورت

خودکار برای شما تنظیم می‌شود.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  For more information on how to configure your ASP.NET application, please visit
  http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <configSections>
    <!-- For more information on Entity Framework configuration, visit
    http://go.microsoft.com/fwlink/?LinkId=237468 -->
    <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
  </configSections>
  <connectionStrings>
    <add name="DefaultConnection" connectionString="Data
    Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\WebFormsIdentity.mdf;Initial
    Catalog=WebFormsIdentity;Integrated Security=True"
    providerName="System.Data.SqlClient" />
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.5" />
    <httpRuntime targetFramework="4.5" />
  </system.web>
  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
    EntityFramework">
      <parameters>
        <parameter value="v11.0" />
      </parameters>
    </defaultConnectionFactory>
    <providers>
      <provider invariantName="System.Data.SqlClient"
      type="System.Data.Entity.SqlServer.SqlProviderServices, EntityFramework.SqlServer" />
    </providers>
  </entityFramework>
</configuration>
```

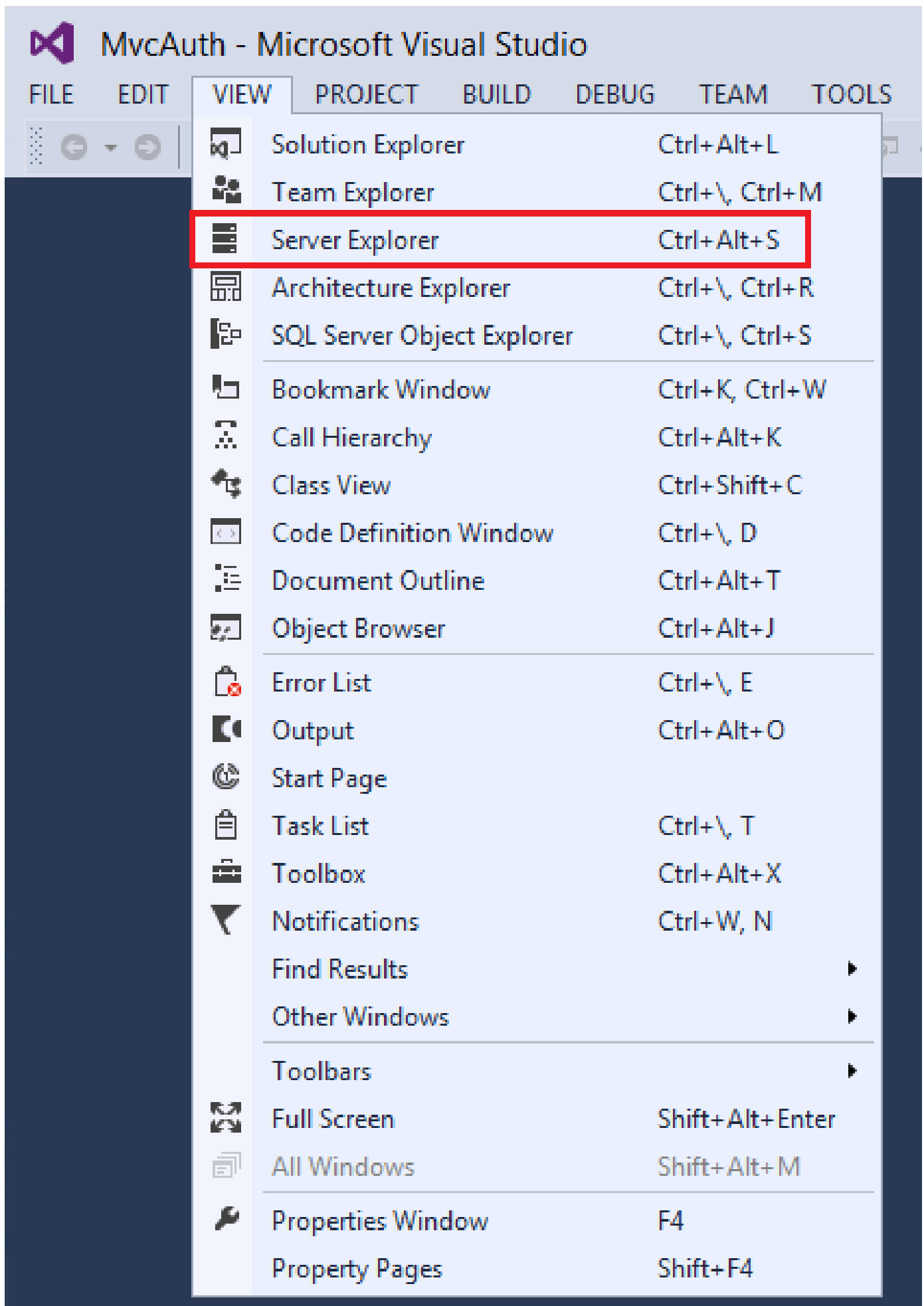
همانطور که مشاهده می‌کنید نام این رشته اتصال *DefaultConnection* است.

روی فایل *Register.aspx* کلیک راست کنید و گزینه **Set As Start Page** را انتخاب کنید. اپلیکیشن خود را با کلیدهای ترکیبی **Ctrl + F5** اجرا کنید که تمام پروژه را کامپایل نیز خواهد کرد. یک نام کاربری و کلمه عبور وارد کنید و روی **Register** کلیک کنید.

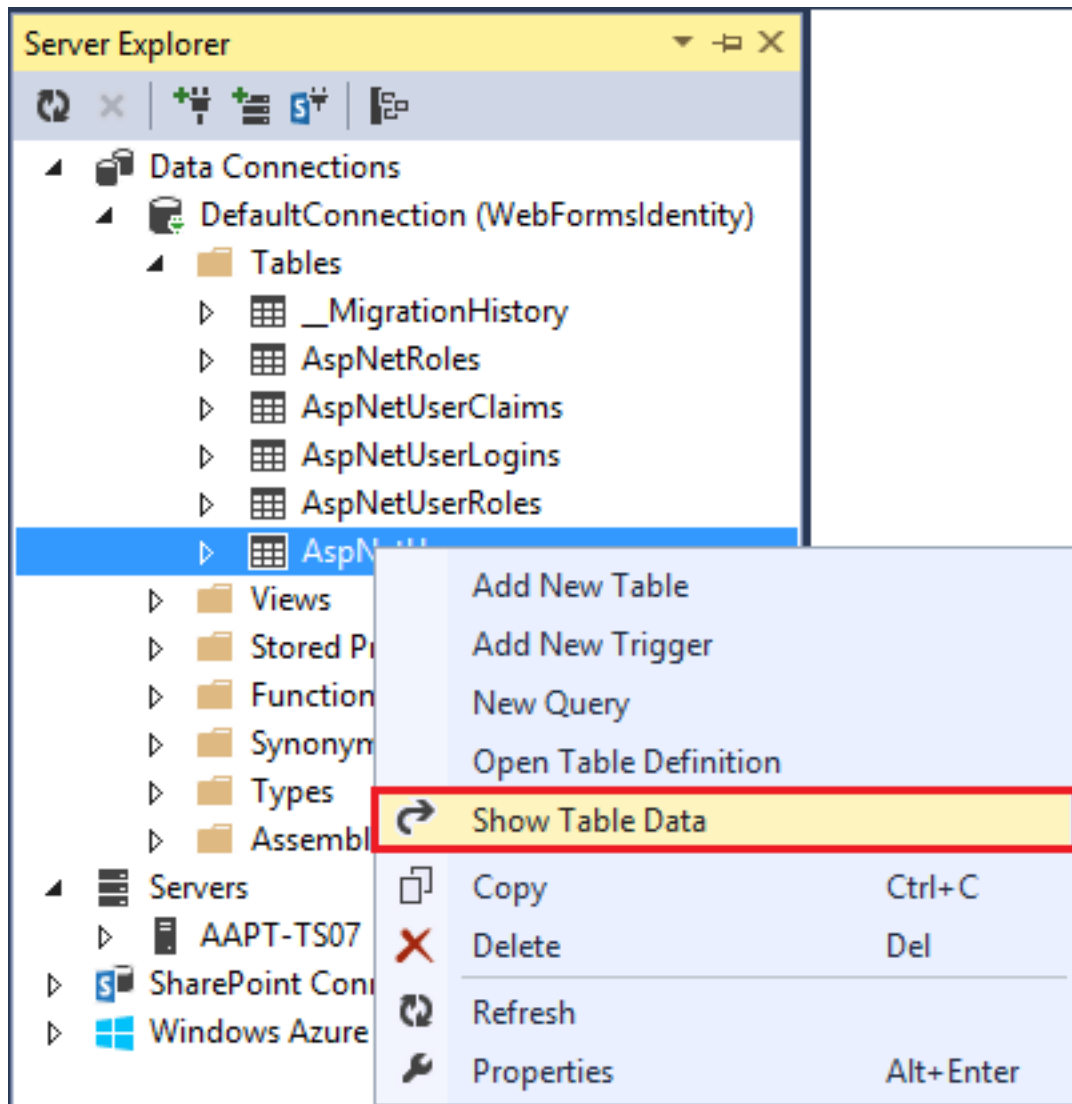


ASP.NET Identity از اعتبارسنجی نیز پشتیبانی می‌کند، مثلاً در این مرحله می‌توانید از اعتبارسنجی‌هایی که توسط ASP.NET Identity Core عرضه می‌شوند برای کنترل رفتار فیلدهای نام کاربری و کلمه عبور استفاده کنید. اعتبارسنجی پیش فرض کاربران (User) که `UserValidator` نام دارد خاصیتی با نام `AllowOnlyAlphanumericUserNames` دارد که مقدار پیش فرضش هم `true` است. اعتبارسنجی پیش فرض کلمه عبور (`MinimumLengthValidator`) اطمینان حاصل می‌کند که کلمه عبور حداقل 6 کاراکتر باشد. این اعتبارسنجی‌ها بصورت `property` در کلاس `userManager` تعریف شده‌اند و می‌توانید آنها را `overwrite` کنید و اعتبارسنجی سفارشی خود را پیاده کنید. از آنجا که الگوی دیتابیس سیستم عضویت توسط `Entity Framework` مدیریت می‌شود، روی الگوی دیتابیس کنترل کامل دارید، پس از `Data Annotations` نیز می‌توانید استفاده کنید.

تایید دیتابیس `LocalDbIdentity` که توسط EF ساخته می‌شود
از منوی `View` گزینه `Server Explorer` را انتخاب کنید.



گره (DefaultConnection (WebFormsIdentity و سپس Tables را باز کنید. روی جدول **AspNetUsers** کلیک راست کرده و **Show Table Data** را انتخاب کنید.



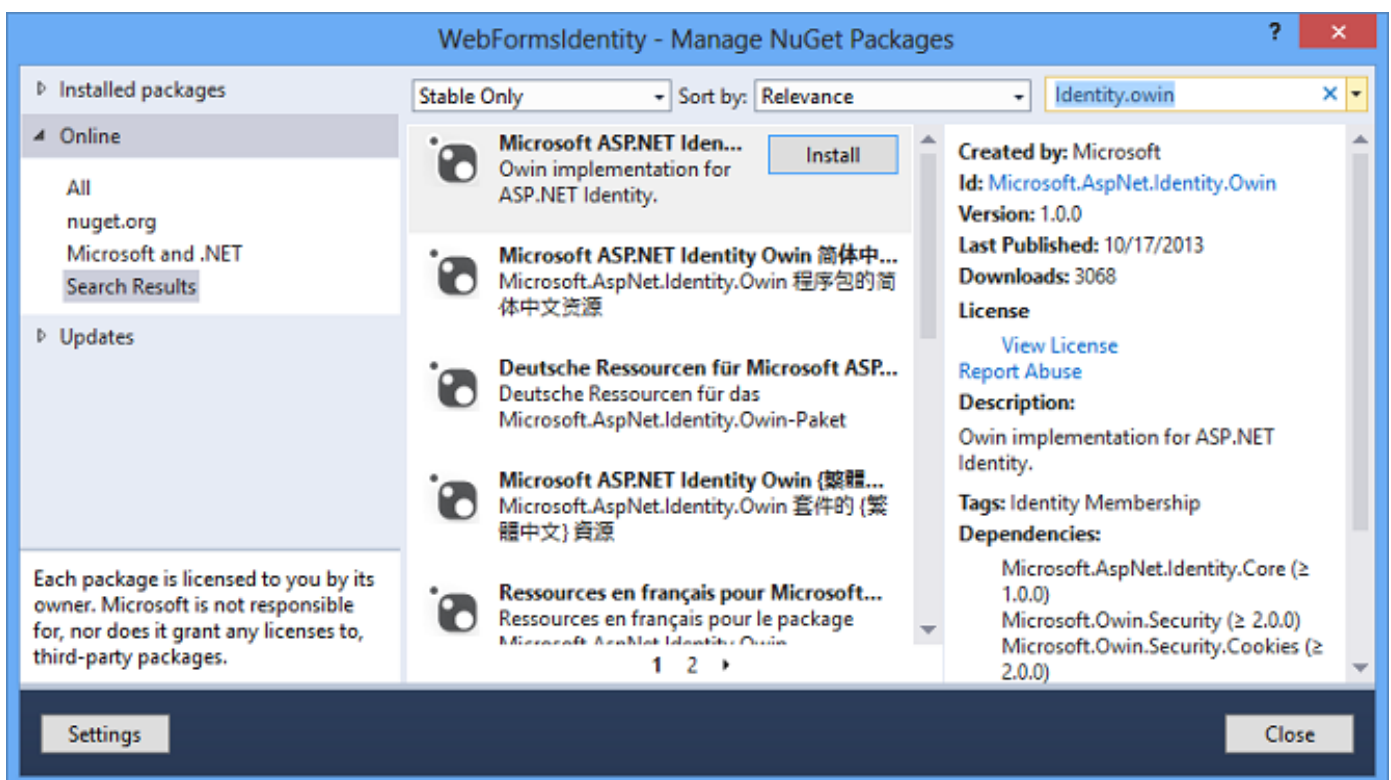
dbo.AspNetUsers [Data]				
Max Rows: 1000				
	Id	UserName	PasswordHash	SecurityStamp
	ca0-b79ad701433c	Raquel	AMmwIk9DNJf...	e12b7b1c-2aac...
*	NULL	NULL	NULL	NULL

پیکربندی اپلیکیشن برای استفاده از احراز هویت OWIN

تا این مرحله ما تنها امکان ایجاد حساب‌های کاربری را فراهم کرده ایم. حال نیاز داریم امکان احراز هویت کاربران برای ورود آنها به سایت را فراهم کنیم. ASP.NET Identity برای احراز هویت مبتنی بر فرم (forms authentication) از OWIN Authentication استفاده می‌کند. OWIN Cookie Authentication مکانیزمی برای احراز هویت کاربران بر اساس cookie ها و claim ها است (claims-based). این مکانیزم می‌تواند توسط Entity Framework روی [OWIN](#) یا IIS استفاده شود. با چنین مدلی، می‌توانیم از پکیج‌های احراز هویت خود در فریم ورک‌های مختلفی استفاده کنیم، مانند ASP.NET MVC و ASP.NET Web Forms. برای اطلاعات بیشتر درباره پروژه Katana و نحوه اجرای آن بصورت Host Agnostic به لینک [Getting Started with the Katana Project](#) مراجعه کنید.

نصب پکیج‌های احراز هویت روی پروژه

روی نام پروژه خود کلیک راست کرده و **Manage NuGet Packages** را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "Identity.Owin" را وارد کنید و این پکیج را نصب کنید.



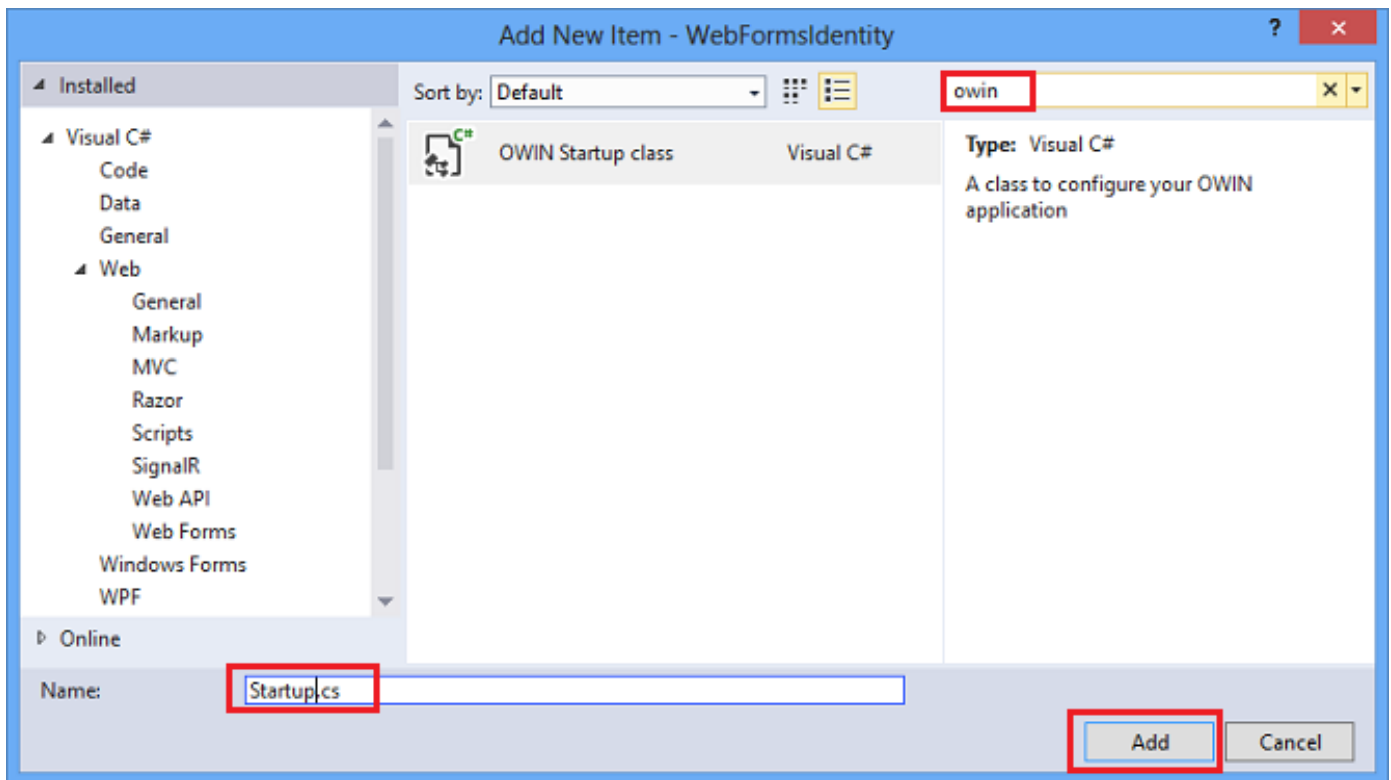
به دنبال پکیجی با نام *Microsoft.Owin.Host.SystemWeb* بگردید و آن را نیز نصب کنید.

پکیج **Microsoft.AspNet.Identity.Owin** حاوی یک سری کلاس *Owin Extension* است و امکان مدیریت و پیکربندی OWIN Authentication در پکیج‌های ASP.NET Identity Core را فراهم می‌کند.

پکیج **Microsoft.Owin.Host.SystemWeb** حاوی یک سرور OWIN است که اجرای اپلیکیشن‌های مبتنی بر OWIN را روی IIS و استفاده از ASP.NET Request Pipeline ممکن می‌سازد. برای اطلاعات بیشتر به [OWIN Middleware in the IIS integrated pipeline](#) مراجعه کنید.

افزودن کلاس‌های پیکربندی Startup و Authentication

روی پروژه خود کلیک راست کرده و گزینه **Add** و سپس **Add New Item** را انتخاب کنید. در قسمت جستجوی دیالوگ باز شده عبارت "owin" را وارد کنید. نام کلاس را "Startup" تعیین کرده و تایید کنید.



فایل Startup.cs را باز کنید و قطعه کد زیر را با محتویات آن جایگزین کنید تا احراز هویت OWIN Cookie Authentication پیکربندی شود.

```
using Microsoft.AspNet.Identity;
using Microsoft.Owin;
using Microsoft.Owin.Security.Cookies;
using Owin;

[assembly: OwinStartup(typeof(WebFormsIdentity.Startup))]

namespace WebFormsIdentity
{
    public class Startup
    {
        public void Configuration(IAppBuilder app)
        {
            // For more information on how to configure your application, visit
            http://go.microsoft.com/fwlink/?LinkID=316888
            app.UseCookieAuthentication(new CookieAuthenticationOptions
            {
                AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
                LoginPath = new PathString("/Login")
            });
        }
    }
}
```

این کلاس حاوی خاصیت OwinAttribute است که کلاس راه انداز OWIN را نشانه گذاری می‌کند. هر اپلیکیشن OWIN یک کلاس راه انداز (startup) دارد که توسط آن می‌توانید کامپوننت‌های application pipeline را مشخص کنید. برای اطلاعات بیشتر درباره این مدل، به [OWIN Startup Class Detection](#) مراجعه فرمایید.

افزودن فرم‌های وب برای ثبت نام و ورود کاربران

فایل Register.cs را باز کنید و قطعه کد زیر را وارد کنید. این قسمت پس از ثبت نام موفقیت آمیز کاربر را به سایت وارد می‌کند.

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using System;
using System.Linq;
using System.Web;

namespace WebFormsIdentity
{
    public partial class Register : System.Web.UI.Page
    {
        protected void CreateUser_Click(object sender, EventArgs e)
        {
            // Default UserStore constructor uses the default connection string named: DefaultConnection
            var userStore = new UserStore<IdentityUser>();
            var manager = new UserManager<IdentityUser>(userStore);
            var user = new IdentityUser() { UserName = UserName.Text };

            IdentityResult result = manager.Create(user, Password.Text);

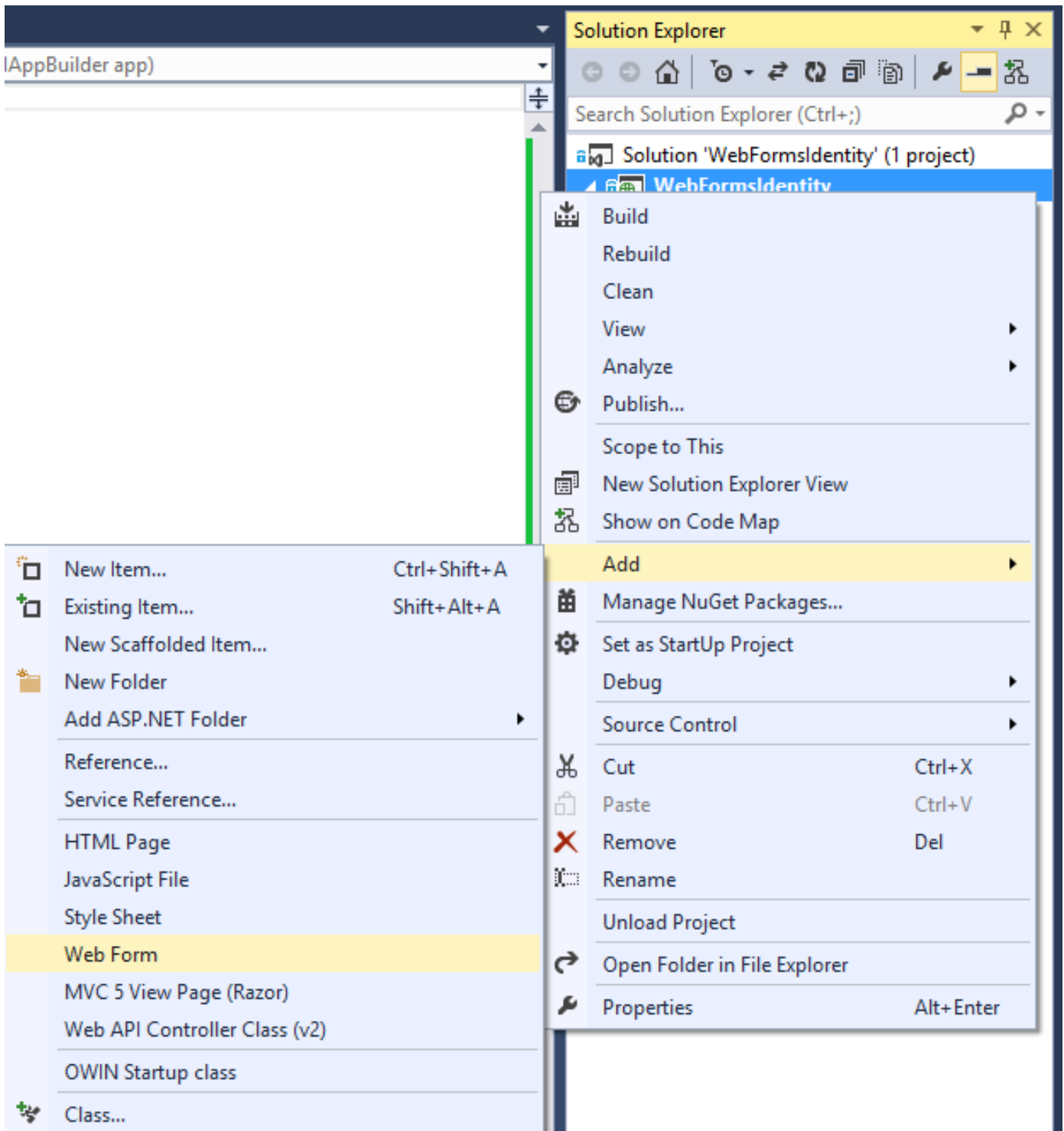
            if (result.Succeeded)
            {
                var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
                var userIdentity = manager.CreateIdentity(user,
DefaultAuthenticationTypes.ApplicationCookie);
                authenticationManager.SignIn(new AuthenticationProperties() { }, userIdentity);
                Response.Redirect("~/Login.aspx");
            }
            else
            {
                StatusMessage.Text = result.Errors.FirstOrDefault();
            }
        }
    }
}

```

از آنجا که ASP.NET Identity و OWIN Cookie Authentication هر دو مبتنی بر Claims هستند، فریم ورک از برنامه نویسی اپلیکیشن انتظار دارد تا برای کاربر یک آبجکت از نوع [ClaimsIdentity](#) تولید کند. این آبجکت تمام اطلاعات اختیارات کاربر را در بر می‌گیرد، مثلاً اینکه کاربر به چه نقش‌هایی تعلق دارد. همچنین در این مرحله می‌توانید اختیارات (Claims) جدیدی به کاربر اضافه کنید.

شما با استفاده از AuthenticationManager که متعلق به OWIN است می‌توانید کاربر را به سایت وارد کنید. برای این کار شما متد SignIn را فراخوانی می‌کنید و آبجکتی از نوع ClaimsIdentity را به آن پاس می‌دهید. این کد کاربر را به سایت وارد می‌کند و یک کوکی برای او می‌سازد. این فراخوانی معادل همان [FormAuthentication.SetAuthCookie](#) است که توسط ماژول [FormsAuthentication](#) استفاده می‌شود.

روی پروژه خود کلیک راست کرده، فرم وب جدیدی با نام **Login** بسازید.



فایل *Login.aspx* را باز کنید و کد Markup آن را مانند قطعه کد زیر تغییر دهید.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Login.aspx.cs"
Inherits="WebFormsIdentity.Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title></title>
</head>
<body style="font-family: Arial, Helvetica, sans-serif; font-size: small">
<form id="form1" runat="server">
<div>
```

```

<h4 style="font-size: medium">Log In</h4>
<hr />
<asp:PlaceHolder runat="server" ID="LoginStatus" Visible="false">
    <p>
        <asp:Literal runat="server" ID="StatusText" />
    </p>
</asp:PlaceHolder>
<asp:PlaceHolder runat="server" ID="LoginForm" Visible="false">
    <div style="margin-bottom: 10px">
        <asp:Label runat="server" AssociatedControlID="UserName">User name</asp:Label>
        <div>
            <asp:TextBox runat="server" ID="UserName" />
        </div>
    </div>
    <div style="margin-bottom: 10px">
        <asp:Label runat="server" AssociatedControlID="Password">Password</asp:Label>
        <div>
            <asp:TextBox runat="server" ID="Password" TextMode="Password" />
        </div>
    </div>
    <div style="margin-bottom: 10px">
        <div>
            <asp:Button runat="server" OnClick="SignIn" Text="Log in" />
        </div>
    </div>
</asp:PlaceHolder>
<asp:PlaceHolder runat="server" ID="LogoutButton" Visible="false">
    <div>
        <div>
            <asp:Button runat="server" OnClick="SignOut" Text="Log out" />
        </div>
    </div>
</asp:PlaceHolder>
</div>
</form>
</body>
</html>

```

محتوای فایل *Login.aspx.cs* را نیز مانند لیست زیر تغییر دهید.

```

using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;
using Microsoft.Owin.Security;
using System;
using System.Web;
using System.Web.UI.WebControls;

namespace WebFormsIdentity
{
    public partial class Login : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (User.Identity.IsAuthenticated)
                {
                    StatusText.Text = string.Format("Hello {0}!", User.Identity.GetUserName());
                    LoginStatus.Visible = true;
                    LogoutButton.Visible = true;
                }
                else
                {
                    LoginForm.Visible = true;
                }
            }
        }

        protected void SignIn(object sender, EventArgs e)
        {
            var userStore = new UserStore<IdentityUser>();
            var userManager = new UserManager<IdentityUser>(userStore);
            var user = userManager.Find(UserName.Text, Password.Text);

            if (user != null)
            {
                var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
                var userIdentity = userManager.CreateIdentity(user,

```

```

DefaultAuthenticationTypes.ApplicationCookie);
        authenticationManager.SignIn(new AuthenticationProperties() { IsPersistent = false },
userIdentity);
        Response.Redirect("~/Login.aspx");
    }
    else
    {
        StatusText.Text = "Invalid username or password.";
        LoginStatus.Visible = true;
    }
}

protected void SignOut(object sender, EventArgs e)
{
    var authenticationManager = HttpContext.Current.GetOwinContext().Authentication;
    authenticationManager.SignOut();
    Response.Redirect("~/Login.aspx");
}
}
}

```

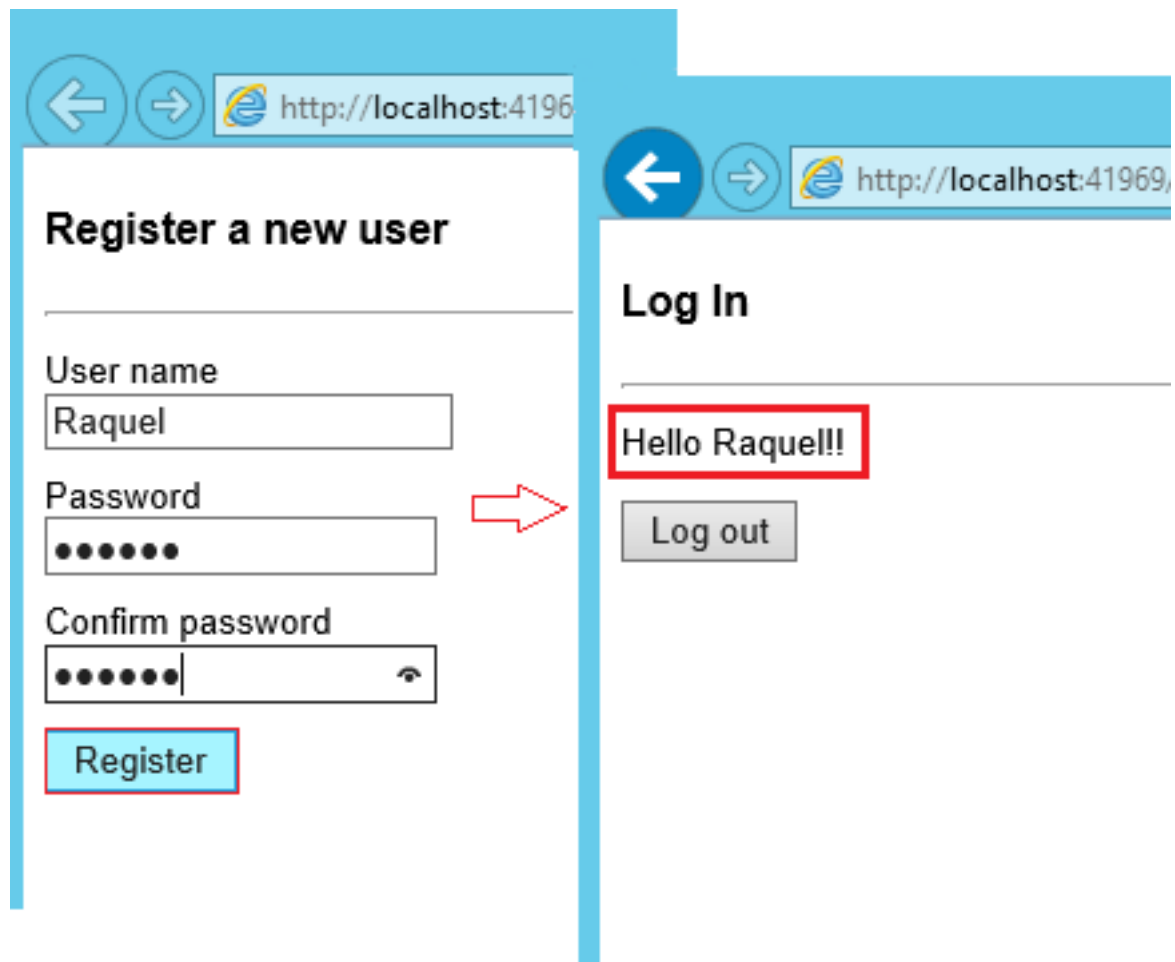
متد Page_Load حالا وضعیت کاربر جاری را بررسی می‌کند و بر اساس وضعیت Context.User.Identity.IsAuthenticated تصمیم‌گیری می‌کند.

نمایش نام کاربر جاری: فریم ورک ASP.NET Identity روی [System.Security.Principal.Identity](#) متدهایی نوشته است که به شما امکان دریافت نام و شناسه کاربر جاری را می‌دهد. این متدها در اسمبلی Microsoft.AspNet.Identity.Core وجود دارند. این متدها جایگزین [HttpContext.User.Identity.Name](#) هستند.

متد SignIn

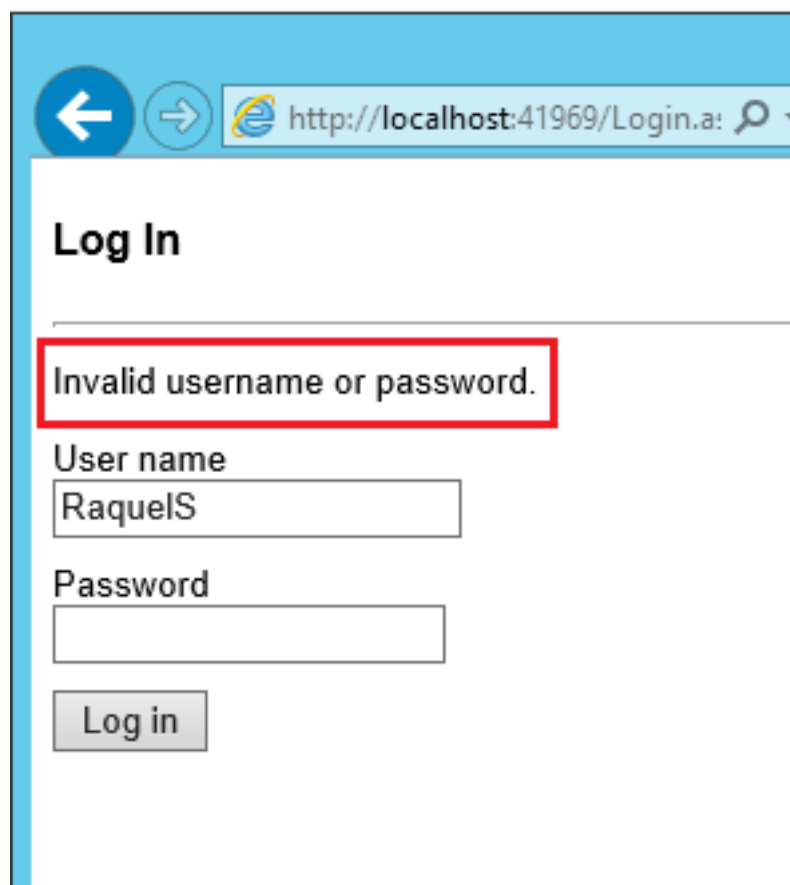
این متد، متد CreateUser_Click را که پیشتر بصورت خودکار ایجاد شده جایگزین می‌کند و پس از ایجاد موفقیت آمیز حساب کاربری، کاربر جاری را به سایت وارد می‌کند. فریم ورک OWIN متدهایی روی System.Web.HttpContext افزوده است که به شما این امکان را می‌دهند که یک ارجاع از نوع IOwinContext بگیرید. این متدها در اسمبلی Microsoft.Owin.Host.SystemWeb وجود دارند. کلاس OwinContext خاصیتی از نوع IAuthenticationManager دارد که امکانات احراز هویت موجود برای درخواست جاری را معرفی می‌کند.

پروژه را با Ctrl + F5 اجرا کنید و کاربر جدیدی بسازید. پس از وارد کردن نام کاربری و کلمه عبور و کلیک کردن دکمه Register باید بصورت خودکار به سایت وارد شوید و نام خود را مشاهده کنید.



همانطور که مشاهده می‌کنید در این مرحله حساب کاربری جدید ایجاد شده و به سایت وارد شده اید. روی **Log out** کلیک کنید تا از سایت خارج شوید. پس از آن باید به صفحه ورود هدایت شوید. حالا یک نام کاربری یا کلمه عبور نامعتبر وارد کنید و روی **Log in** کلیک کنید.

متد `UserManager.Find` مقدار `null` بر می‌گرداند، بنابراین پیام خطای `"Invalid username or password"` نمایش داده خواهد شد.



The screenshot shows a web browser window with a light blue header. The address bar displays the URL `http://localhost:41969/Login.a`. Below the header, the page title is "Log In". A red rectangular box highlights the error message "Invalid username or password." which is displayed in black text. Below the error message, there are two input fields: "User name" containing the text "RaquelS" and "Password" which is empty. At the bottom of the form is a "Log in" button.

Log In

Invalid username or password.

User name
RaquelS

Password

Log in

نظرات خوانندگان

نویسنده: Programmer

تاریخ: ۱۴:۴۱ ۱۳۹۲/۱۰/۱۹

با عرض سلام و تشکر بابت ترجمه روانتون. خیلی وقت بود که منتظر همچین پستی بودم. اینکه بشه با EF عملیات احراز هویت رو با مکانیسمی قویتر از Membership سابق انجام داد. اگر ممکنه همین مثال رو در قالب پروژه MVC انجام بدید. ممنون

نویسنده: آرمین ضیاء

تاریخ: ۱۸:۰۹ ۱۳۹۲/۱۰/۱۹

سلام، متشکرم.

ASP.NET Identity بصورت پیش فرض در قالب پروژه‌های VS 2013 استفاده میشه. در پست‌های قبلی بیشتر درباره این فریم بحث شده که می‌تونید مراجعه کنید. برای اطلاعات بیشتر به [ASP.NET Identity](#) سر بزنید.

نویسنده: کامران

تاریخ: ۱۶:۰۰ ۱۳۹۲/۱۲/۲۶

سلام.

من اگر بخوام لیست کاربرایی که یک نقش خاص مثلا "Admin" رو دارن لیست کنم چطور میتونم دسترسی داشته باشم؟

نویسنده: غلامرضا

تاریخ: ۳:۴۹ ۱۳۹۳/۰۵/۳۰

سلام و میبخشید آیا به غیر از روش ایداعی [این سایت](#) روش دیگه ای برای استفاده از asp.net identity با EF db first وجود داره؟ اگه هست لطفا راهنمایی کنید.

نویسنده: غلامرضا

تاریخ: ۱۶:۳۲ ۱۳۹۳/۰۵/۳۱

جواب رو پیدا کردم اینم لینکش. [فیلم](#)