



## Bài 09

# *Authentication và Authorization*



- ◆ Giới thiệu về Authentication
- ◆ Form authentication
- ◆ Authentication Modes
- ◆ ASP.NET Identity
- ◆ Authorization
- ◆ Users và Roles



- ◆ Authentication là tiến trình nhận biết người dùng trước khi gán quyền truy cập vào nguồn tài nguyên của ứng dụng.
- ◆ Khi ứng dụng nhận được request từ người dùng nó sẽ tiến hành việc nhận biết người dùng thông qua username và password mà người dùng cung cấp.
- ◆ ASP.NET MVC Framework cung cấp 3 chế độ nhận biết người dùng
  - ◆ Forms authentication
  - ◆ Windows authentication
  - ◆ OpenId/OAuth



- ◆ Ứng dụng chịu trách nhiệm thu thập thông tin người dùng và xác thực người dùng
- ◆ Ứng dụng xác thực người dùng bằng cách cung cấp một form login trên trang web.
- ◆ Khi người dùng login thành công nó sẽ sinh ra một cookie phục vụ như là một thẻ xác thực người dùng.
- ◆ Cookie sau đó được sử dụng bởi trình duyệt cho những lần request sau của người dùng để validate request đó.



# Authentication Modes

- ◆ Khi bạn tạo ứng dụng ASP.NET MVC trong Visual Studio 2013, ứng dụng không tự động sử dụng bất kỳ loại xác thực nào.
- ◆ Để xác thực người dùng bằng Forms authentication bạn cần tự cấu hình trong phân tử <system.web> của file Web.config

```
<system.web>  
  <authentication mode="None" />  
</system.web>
```

```
<system.web>  
  <authentication mode="Forms" />  
</system.web>
```



- ◆ Windows authentication phù hợp với môi trường intranet, ở đây người dùng được biết đến thông qua việc login vào hệ thống mạng cục bộ.
- ◆ Khi bạn sử dụng cơ chế login này trong ứng dụng của bạn, nó sẽ là bức tường lửa kiểm soát tiến trình xác thực.
- ◆ Trong cơ chế này ứng dụng sẽ sử dụng thông tin người dùng được xác thực bởi firewall của máy tính cho việc kiểm tra bảo mật.
- ◆ Để cấu hình bạn chỉ cần thay đổi thuộc tính mode trong phần tử authentication của file web.config thành Windows



# Authentication Modes

2 references

```
public partial class Startup
```

```
{
```

```
    // For more information on configuring authentication, please visit http://go.microsoft.com/fwlink/?LinkId=301864
```

1 reference

```
    public void ConfigureAuth(IAppBuilder app)
```

```
    {
```

```
        // Enable the application to use a cookie to store information for the signed in user
```

```
        app.UseCookieAuthentication(new CookieAuthenticationOptions
```

```
        {
```

```
            AuthenticationType = DefaultAuthenticationTypes.ApplicationCookie,
```

```
            LoginPath = new PathString("/Account/Login")
```

```
        });
```

```
        // Use a cookie to temporarily store information about a user logging in with a third party login provider
```

```
        app.UseExternalSignInCookie(DefaultAuthenticationTypes.ExternalCookie);
```

```
        // Uncomment the following lines to enable logging in with third party login providers
```

```
        //app.UseMicrosoftAccountAuthentication(
```

```
        //     clientId: "",
```

```
        //     clientSecret: "");
```

```
        //app.UseTwitterAuthentication(
```

```
        //     consumerKey: "",
```

```
        //     consumerSecret: "");
```

```
        //app.UseFacebookAuthentication(
```

```
        //     appId: "",
```

```
        //     appSecret: "");
```

```
        //app.UseGoogleAuthentication();
```

```
    }
```

```
}
```

- ◆ Trước ASP.NET MVC 5, MVC Framework sử dụng Membership API để thực hiện authentication và authorization.
- ◆ ASP.NET MVC 5 sử dụng ASP.NET Identity, đây là một hệ thống membership mới cho các ứng dụng ASP.NET MVC.
- ◆ Bạn có thể sử dụng ASP.NET Identity để bổ sung các tính năng login cho ứng dụng của bạn.
- ◆ ASP.NET Identity sử dụng phương pháp Code First để lưu trữ thông tin người dùng trong database.
- ◆ Khi tạo ứng dụng ASP.NET MVC trong Visual Studio 2017, các lớp quản lý Identity và Account sẽ tự động thêm vào project
- ◆ Tập IdentityModel.cs được tạo để quản lý Identity của ứng dụng





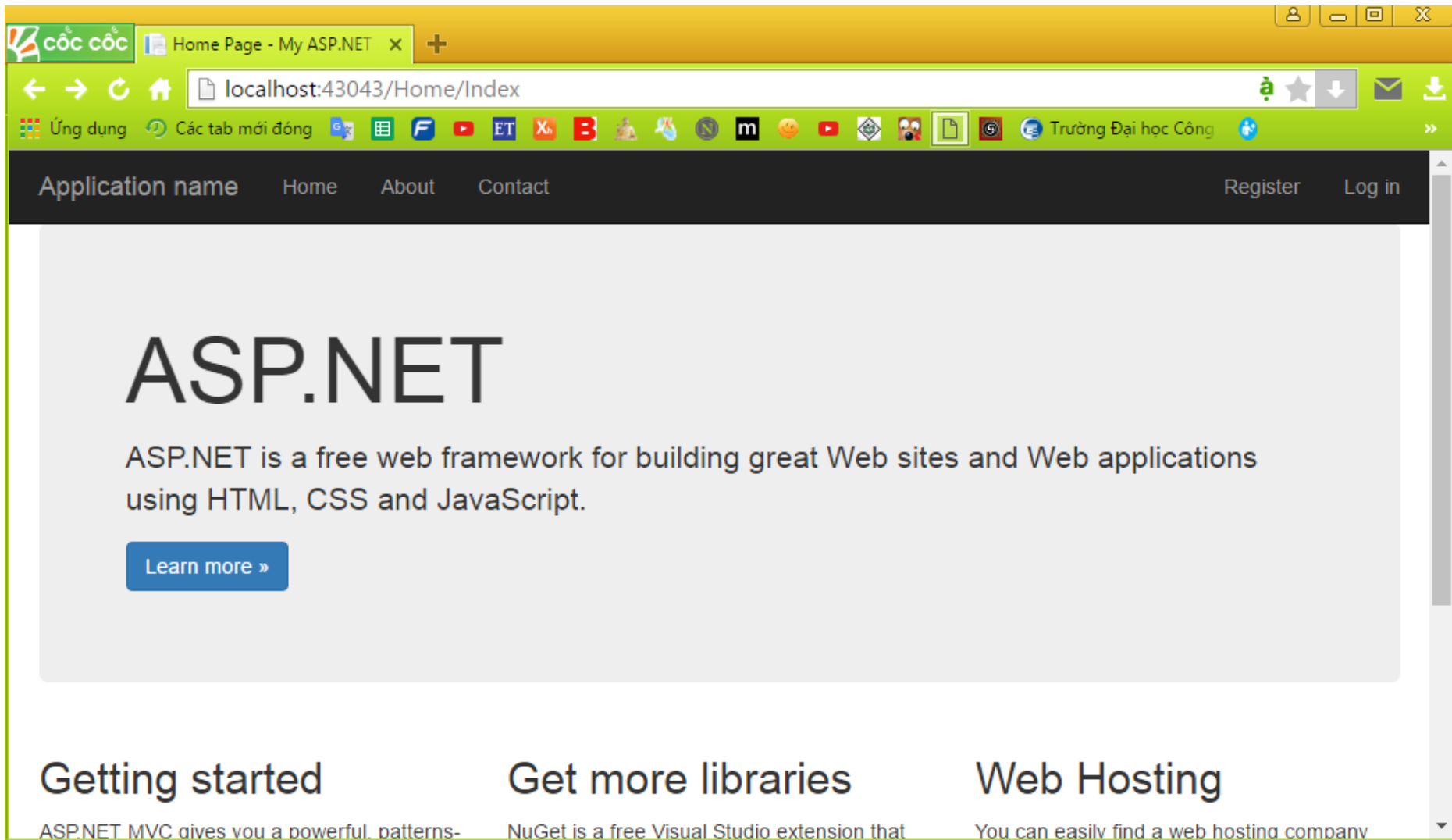
## ◆ Code tệp IdentityModel.cs

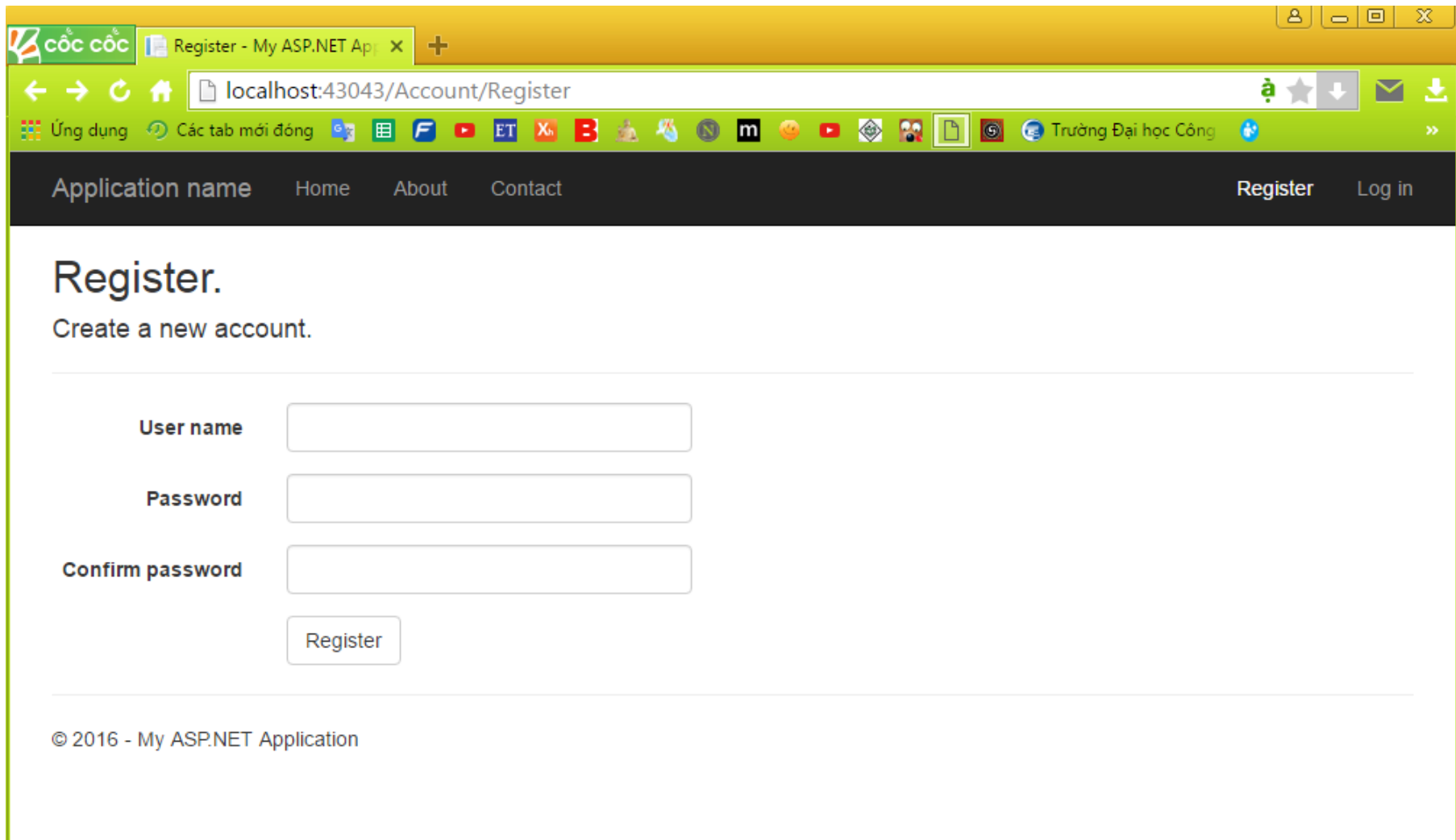
```
namespace WebApplication8.Models
{
    8 references
    public class ApplicationUser : IdentityUser
    {
    }

    2 references
    public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
    {
        1 reference
        public ApplicationDbContext()
            : base("DefaultConnection")
        {
        }
    }
}
```



# ASP.NET Identity





The screenshot shows a web browser window with the title "Register - My ASP.NET Application". The address bar displays "localhost:43043/Account/Register". The browser's taskbar at the bottom shows various application icons, including "Ứng dụng", "Các tab mới đóng", and "Trường Đại học Công".

The web page has a dark navigation bar with the following links: "Application name", "Home", "About", "Contact", "Register", and "Log in".

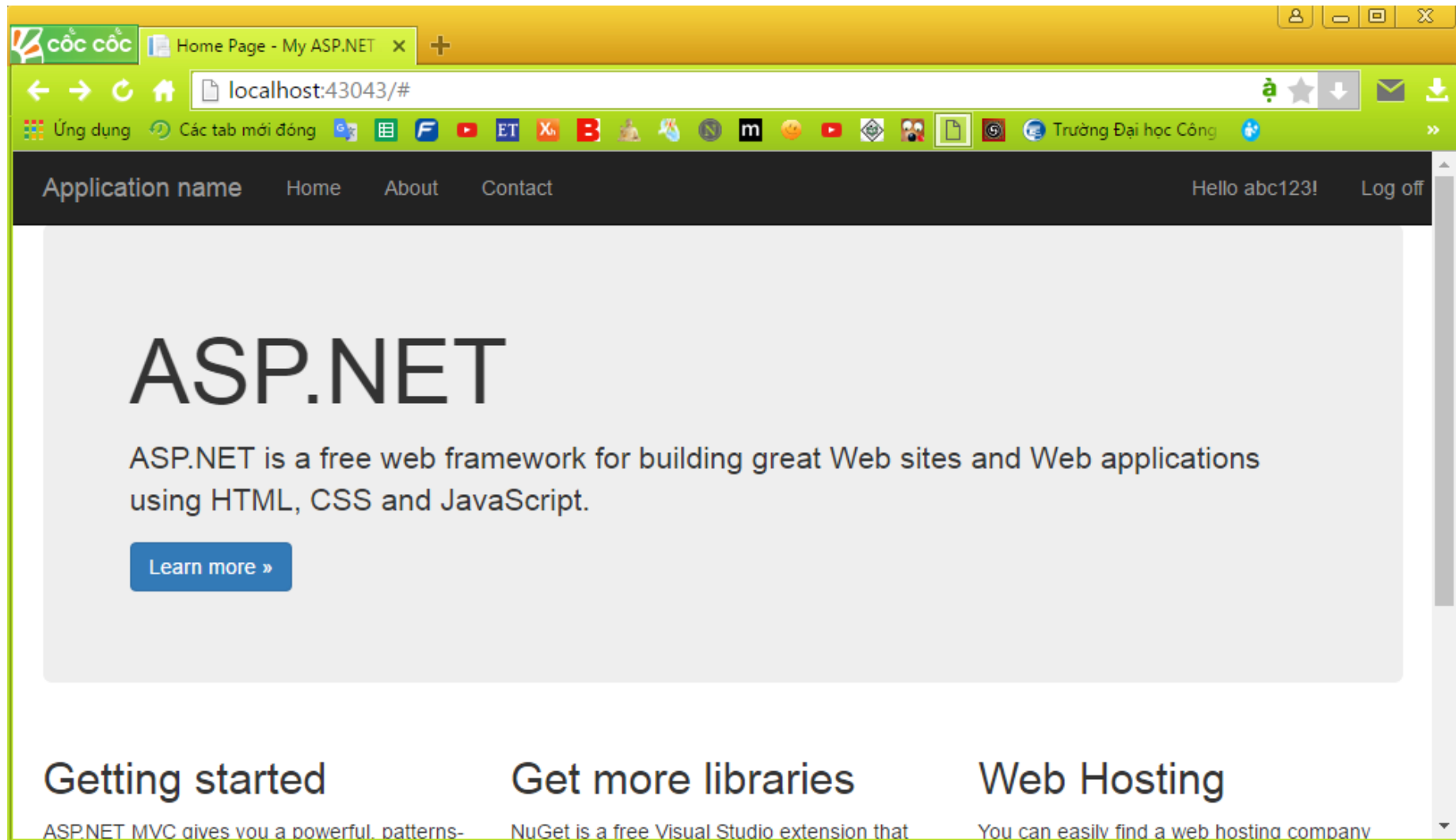
The main content area is titled "Register." and includes the instruction "Create a new account." Below this, there are three input fields for registration:

- User name**:
- Password**:
- Confirm password**:

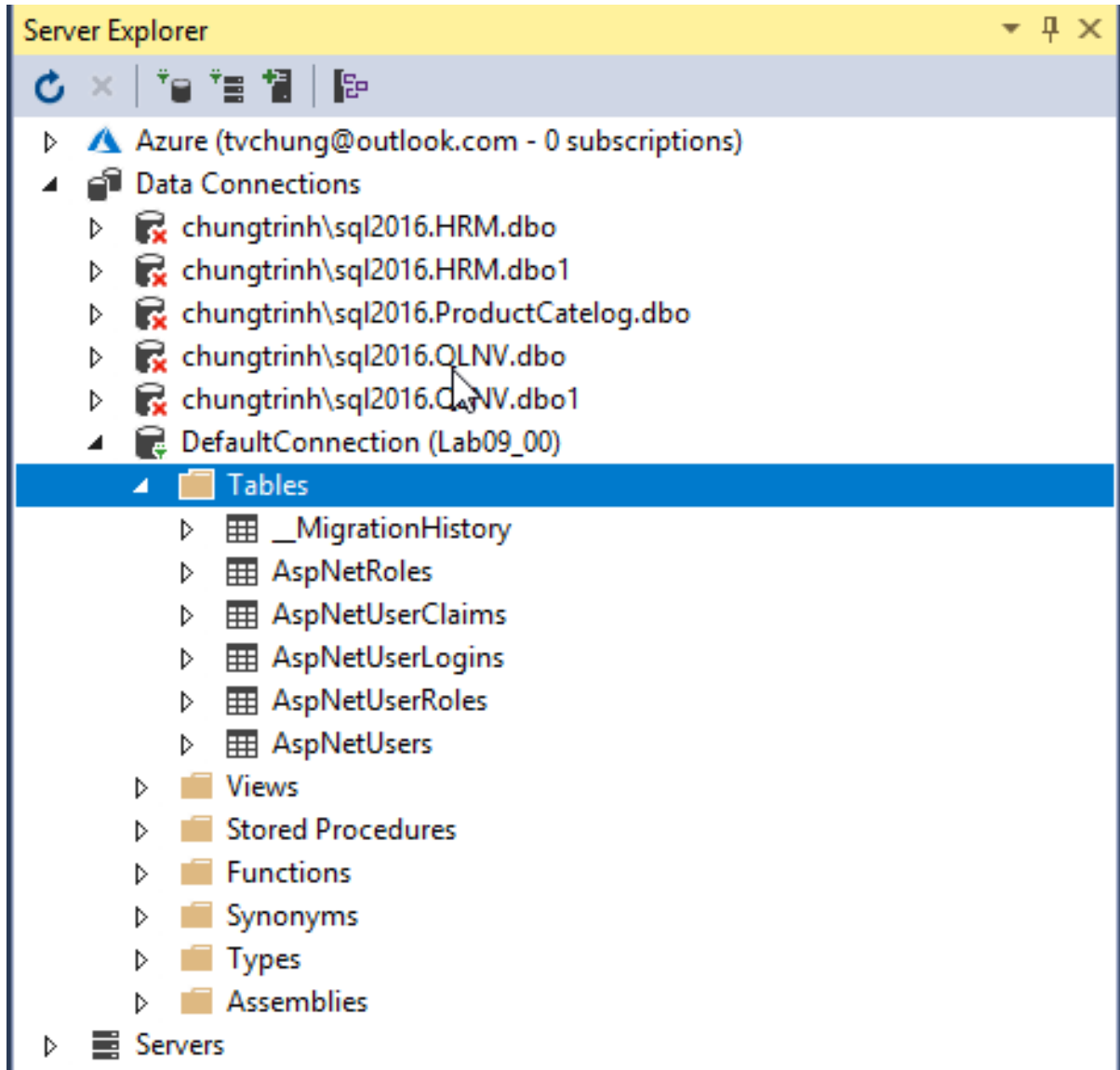
A "Register" button is located below the input fields.

At the bottom of the page, the copyright notice reads: "© 2016 - My ASP.NET Application".

# ASP.NET Identity



# ASP.NET Identity



- ◆ Authorization là tiến trình xác minh xem người dùng đã xác thực được truy cập vào nguồn tài nguyên nào.
- ◆ Sau khi tạo người dùng trong ứng dụng, bạn cần gán quyền truy cập vào tài nguyên của ứng dụng cho người đó



- ◆ Để duy trì một tập các thiết lập khác nhau cho mỗi người dùng là một nhiệm vụ phức tạp. Do vậy bạn cần phải tập hợp những người dùng có cùng chức năng nhiệm vụ vào một nhóm gọi là Role, sau đó định nghĩa quyền hạn cho Role đó
- ◆ Trong ứng dụng ASP.NET MVC bạn có thể xác minh User và Role bằng cách sử dụng thuộc tính Authorize.

```
[Authorize(Users="Mathews, Jones")]  
public class ManagerProduct: Controller  
{  
    // Code to perform some operation  
}
```



# Users và Roles

```
[Authorize(Roles="Administrator")]  
public class ManagerProduct: Controller  
{  
    // Code to perform some operation  
}
```

```
[Authorize(Roles="Manager", Users="Steve, Mathews")]  
public class ManagerProduct: Controller  
{  
    // Code to perform some operation  
}
```





- ◆ ASP.NET MVC Framework cung cấp các loại role như sau:
  - ◆ `ActiveDirectoryRoleProvider`: lớp này cho phép bạn quản lý thông tin vai trò của một ứng dụng trong Active Directory.
  - ◆ `SqlRoleProvider`: lớp này cho phép bạn quản lý thông tin vai trò của một ứng dụng trong SQL database.
  - ◆ `SimpleRoleProvider`: lớp làm việc với các phiên bản của SQL Server khác nhau.
  - ◆ `UniversalProviders`: lớp làm việc với bất kỳ database mà Entity Framework hỗ trợ.



