



LẬP TRÌNH WEBSITE (ASP.NET MVC 5)

CHƯƠNG 7:

KIỂM LỖI & BẢO MẬT WEBSITE



- ✓ Giảng Viên: ThS. Dương Thành Phết
- ✓ Email: phetcm@gmail.com
- ✓ Website: <http://www.thayphet.net>
- ✓ Mobile: 0918158670


NỘI DUNG

1. Giới thiệu
2. Kiểm lỗi trong mvc
3. Thuộc tính
4. Kiểm lỗi bằng tay
5. Kiểm soát yêu cầu giả lập
6. Tấn công website - xss
7. Antiforery
8. Authentication & authorization

1. GIỚI THIỆU

- ✓ Dữ liệu không hợp lệ nhập từ người sử dụng sẽ gây các lỗi khó lường.
- ✓ Vì vậy việc kiểm soát dữ liệu vào luôn đóng vai trò quan trọng.
- ✓ Các lỗi thường gặp
 - Để trống ô nhập...
 - không đúng định dạng email, creditcard, url...
 - sai kiểu số nguyên, số thực, ngày giờ...
 - Không hợp lệ - phải có giá trị tối thiểu, tối đa, trong phạm vi...
 - không đúng như kết quả tính toán trước

1. GIỚI THIỆU

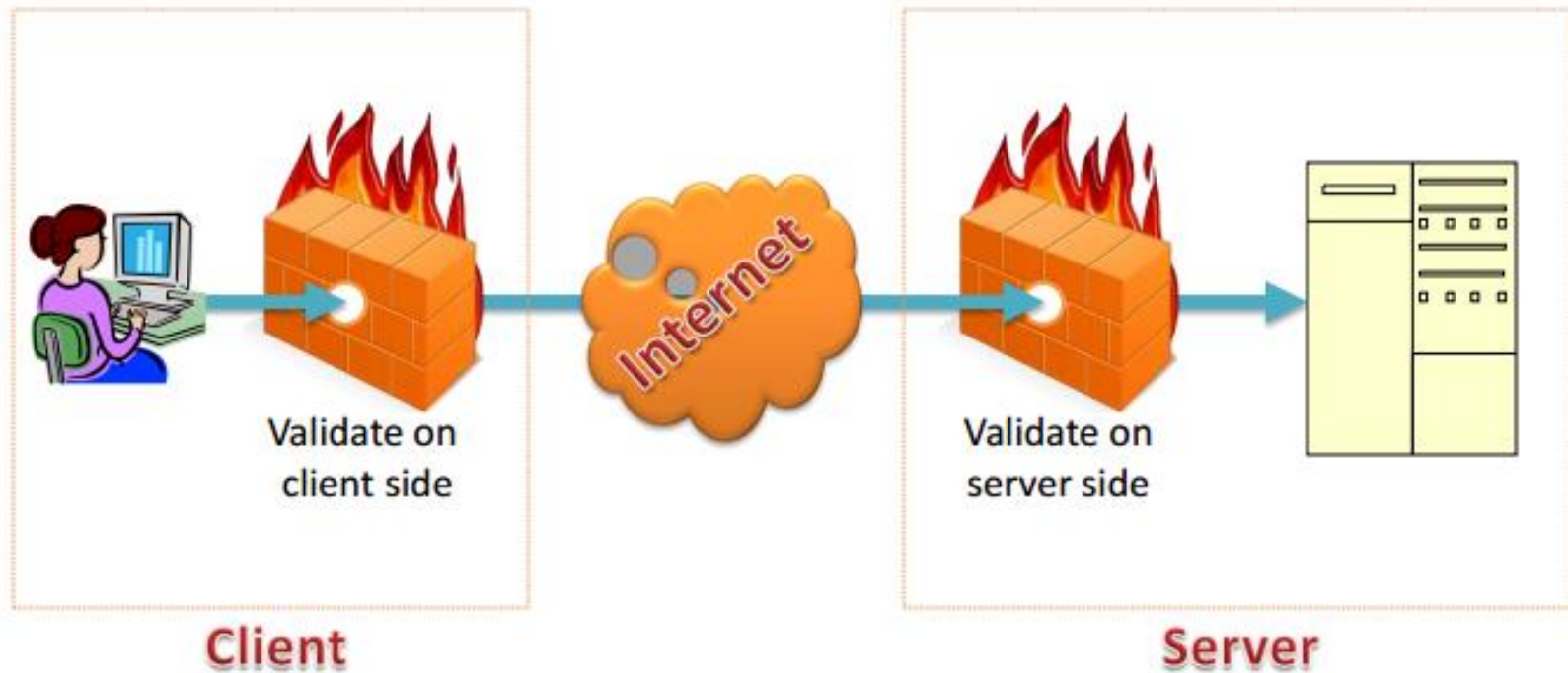
 ĐĂNG KÝ THÀNH VIÊN

Tên đăng nhập	:	<input type="text" value="user"/>	🔥 Mã này đã được sử dụng.
Mật khẩu	:	<input type="password"/>	🔥 Trường bắt buộc.
Nhập lại mật khẩu mới	:	<input type="password" value="abc"/>	🔥 Giá trị nhập không giống.
Họ và tên	:	<input type="text"/>	🔥 Trường bắt buộc.
Giới tính	:	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ	
Thư điện tử	:	<input type="text" value="abc"/>	🔥 Không đúng dạng email.
Điện thoại di động	:	<input type="text"/>	
Ngày sinh	:	<input type="text"/>	🔥 Trường bắt buộc.
Địa chỉ	:	<input type="text"/>	
Hình ảnh	:	<input type="text" value="C:\NetworkCfg.xml"/> <input type="button" value="Browse..."/>	Không chấp nhận loại tập tin này.
Mã bảo mật	:	<input type="text" value="AS"/>	🔥 Sai mã bảo mật. AEBG44

Đăng ký

1. GIỚI THIỆU

Mô hình kiểm lỗi



2. KIỂM LỖI TRONG MVC

- ✓ Kiểm soát dữ liệu có thể thực hiện cả 2 phía là client và server.
 - Kiểm phía client sẽ phản ứng nhanh cho người sử dụng để có thể sửa chữa ngay.
 - Kiểm lỗi phía server sẽ thực hiện các lỗi mà client không thể làm được nếu dữ liệu có liên quan đến tài nguyên server.
- ✓ Với MVC bạn chỉ cần viết 1 lần nhưng kiểm tra cả 2 phía là client và server. Nếu vì một lý do nào đó mà client không thực hiện được thì đã có server thay thế.

2. KIỂM LỖI TRONG MVC

❑ Mã trên Model

- ✎ Đính kèm các Attribute kiểm lỗi cho các Property
 - ✓ `[Required]`, `[StringLength]`...

❑ Mã trên View

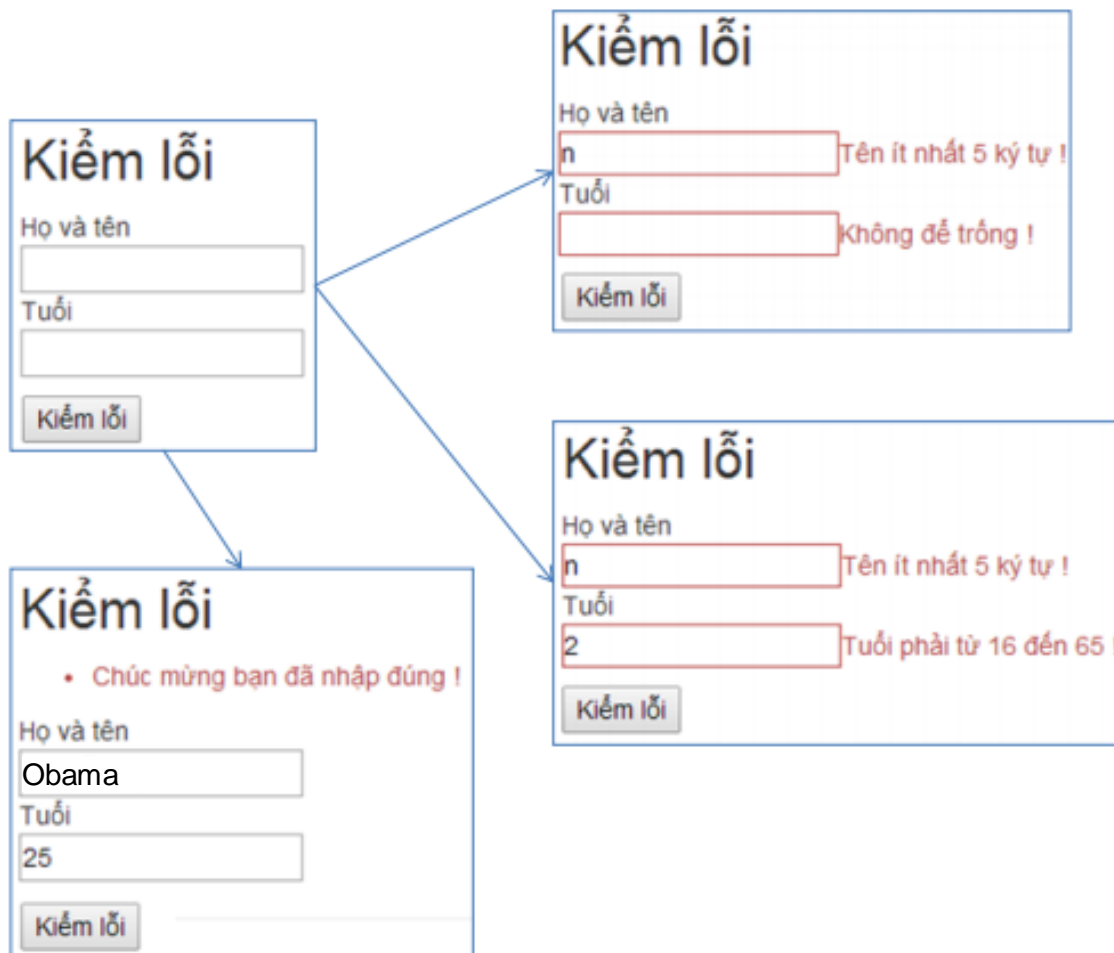
- ✎ Hiển thị lỗi
 - ✓ `@Html.ValidationMessageFor(Property)`
 - ✓ `@Html.ValidationSummary()`
- ✎ Kiểm lỗi phía client
 - ✓ `@Scripts.Render("~/bundles/jquery")`

❑ Mã trên Controller

- ✎ Kiểm lỗi phía server
 - ✓ `ModelState.IsValid`
 - ✓ `ModelState.AddModelError()`

2. KIỂM LỖI TRONG MVC

Ví dụ:



Model?
Controller?
View?

2. KIỂM LỖI TRONG MVC

Lớp Model:

```
public class EmployeeInfo
{
    [MinLength(5, ErrorMessage="Tên ít nhất 5 ký tự !")]
    public String FullName { get; set; }
    [Required(ErrorMessage="Không để trống !")]
    [Range(16, 65, ErrorMessage = "Tuổi phải từ 16 đến 65 !")]
    public int Age { get; set; }
}
```

Annotation	Thuộc tính	Mô tả
[MinLength]	FullName	Giới hạn số lượng ký tự tối thiểu là 5. Nếu không nhập vẫn hợp lệ vì không sử dụng Required
[Required]	Age	Không để trống
[Range]	Age	Giới hạn tuổi từ 16 đến 65

2. KIỂM LỖI TRONG MVC

Lớp Controller

```
public class ValidatorController : Controller
{
```

```
    public ActionResult Index()
    {
        return View();
    }
```

Kiểm lỗi phía server

Kiểm lỗi

Họ và tên

Tuổi

Kiểm lỗi

```
    public ActionResult Validate(EmployeeInfo model)
    {
        if (ModelState.IsValid)
        {
            ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng !");
        }
        return View("Index");
    }
}
```

Bổ sung thông báo lỗi model

2. KIỂM LỖI TRONG MVC

View

```
@model Mvc5CodeDemo.Models.EmployeeInfo
<h2>Kiểm lỗi</h2>
@Html.ValidationSummary(true)
@using (Html.BeginForm("Validate", "Validator"))
{
    <div>Họ và tên</div>
    @Html.TextBoxFor(m => m.FullName)
    @Html.ValidationMessageFor(m => m.FullName)
    <div>Tuổi</div>
    @Html.TextBoxFor(m => m.Age)
    @Html.ValidationMessageFor(m => m.Age)
    <hr />
    <input type="submit" value="Kiểm lỗi" />
}

@section scripts{
    @Scripts.Render("~/bundles/jqueryval")
}
```

Thông báo lỗi chung không bao gồm lỗi đã thông báo cho từng thuộc tính

Thông báo lỗi cho từng thuộc tính

Thực hiện kiểm lỗi phía client

3. THUỘC TÍNH KIỂM LỖI

Annotation	Mô tả	Ví dụ
[Required]	Bắt buộc	[Required] public String Name{get;set}
[Range(Min, Max)]	Giới hạn số trong khoảng	[Range(16, 65)] public String Age{get;set}
[StringLength(Max)]	Giới hạn độ dài chuỗi	[StringLength (20, MinimumLength=5)] public String Password{get;set}
[EmailAddress]	Định dạng email	[EmailAddress] public String Email{get;set}
[CreditCard]	Định dạng số thẻ tín dụng	[CreditCard] public String CardNumber{get;set}
[Url]	Định dạng url	[Url] public String Website{get;set}
[Compare(Property)]	So sánh giá trị	[Compare("Password")] public String ConfirmPassword{get;set}
[RegularExpression(Regex)]	So khớp chuỗi	[RegularExpression("\d{9}")] public String IdCard{get;set}
[MinLength(Min)]	Giới hạn tối thiểu chuỗi, mảng	[MinLength(1)] public String[] Hobbies{get;set}
[MaxLength (Max)]	Giới hạn tối đa chuỗi, mảng	[MaxLength (255)] public String Description{get;set}

3. THUỘC TÍNH KIỂM LỖI

HTML 5 ELEMENT

❑ [DataType(DataType.Password, ErrorMessage = "")]

~~DataType.CreditCard~~

~~DataType.MultilineText~~

~~DataType.Currency~~

~~DataType.Password~~

~~DataType.Date~~

~~DataType.PhoneNumber~~

~~DataType.DateTime~~

~~DataType.PostalCode~~

~~DataType.Duration~~

~~DataType.Text~~

~~DataType.EmailAddress~~

~~DataType.Time~~

~~DataType.Html~~

~~DataType.Upload~~

~~DataType.ImageUrl~~

~~DataType.Url~~

4. KIỂM LỖI BẰNG TAY

```
public ActionResult Validate(String FullName, int Age)
{
    if (String.IsNullOrEmpty(FullName))
    {
        ModelState.AddModelError("FullName", "Không để trống họ và tên");
    }
    else if (FullName.Length < 5)
    {
        ModelState.AddModelError("FullName", "Ít nhất 5 ký tự !");
    }

    if (Age < 16 && Age > 65)
    {
        ModelState.AddModelError("Age", "Tuổi phải từ 16 đến 65 !");
    }

    if (ModelState.Count == 0) // không có lỗi nào
    {
        ModelState.AddModelError("", "Chúc mừng bạn đã nhập đúng !");
    }
    return View("Index");
}
```

4. KIỂM LỖI BẰNG TAY

Thuộc tính kiểm lỗi tùy biến

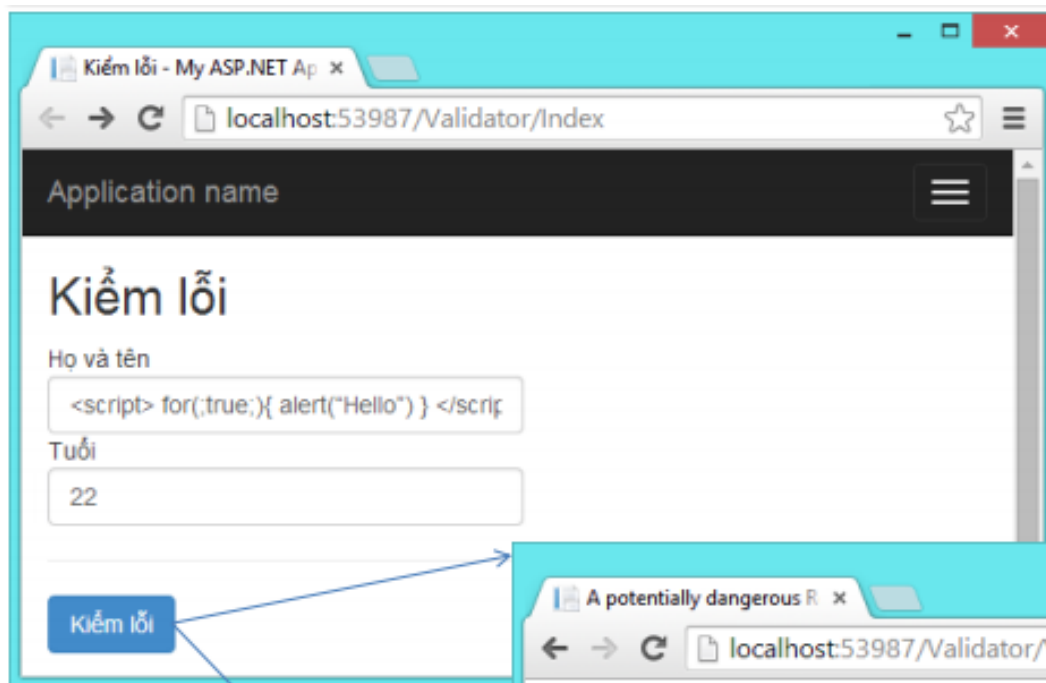
```
public sealed class EvenNumberAttribute: ValidationAttribute
{
    public EvenNumberAttribute() : base("Vui lòng nhập số chẵn !") { }

    public override bool IsValid(object value)
    {
        if (value == null)
        {
            return true;
        }
        return Convert.ToInt64(value) % 2 == 0;
    }
}
```

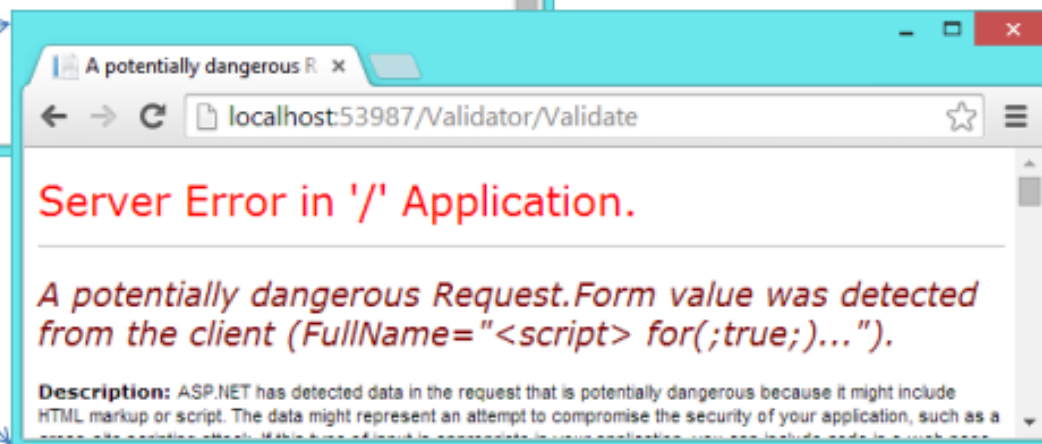
[EvenNumber]
public String Age{get;set}

4. KIỂM LỖI BẰNG TAY

SSX



Tại sao chặn
HTML?



4. KIỂM LỖI BẰNG TAY

SSX

Kiểm lỗi - My ASP.NET Ap x

localhost:53987/Validator/Validate

Application name

Kiểm lỗi

- Chúc mừng bạn đã nhập đúng !

Họ và tên

Tuổi

Kiểm lỗi

```
[ValidateInput(false)]
public ActionResult Validate(EmployeeInfo model)
{
    if (ModelState.IsValid)
    {
        ModelState.AddModelError("",
            "Chúc mừng bạn đã nhập đúng !");
    }
    return View("Index");
}
```

5. KIỂM SOÁT YÊU CẦU GIẢ LẬP

- Bổ sung **@Html.AntiForgeryToken()** vào form để tránh các request giả mạo

```
@using (Html.BeginForm("Withdraw", "Bank")) {  
    @Html.AntiForgeryToken()  
    <fieldset>  
        <legend>Fields</legend>  
        <p>  
            <label for="Amount">Amount:</label>  
            @Html.TextBox("Amount")  
        </p>  
        <p>  
            <input type="submit" value="Withdraw" />  
        </p>  
    </fieldset>  
}
```

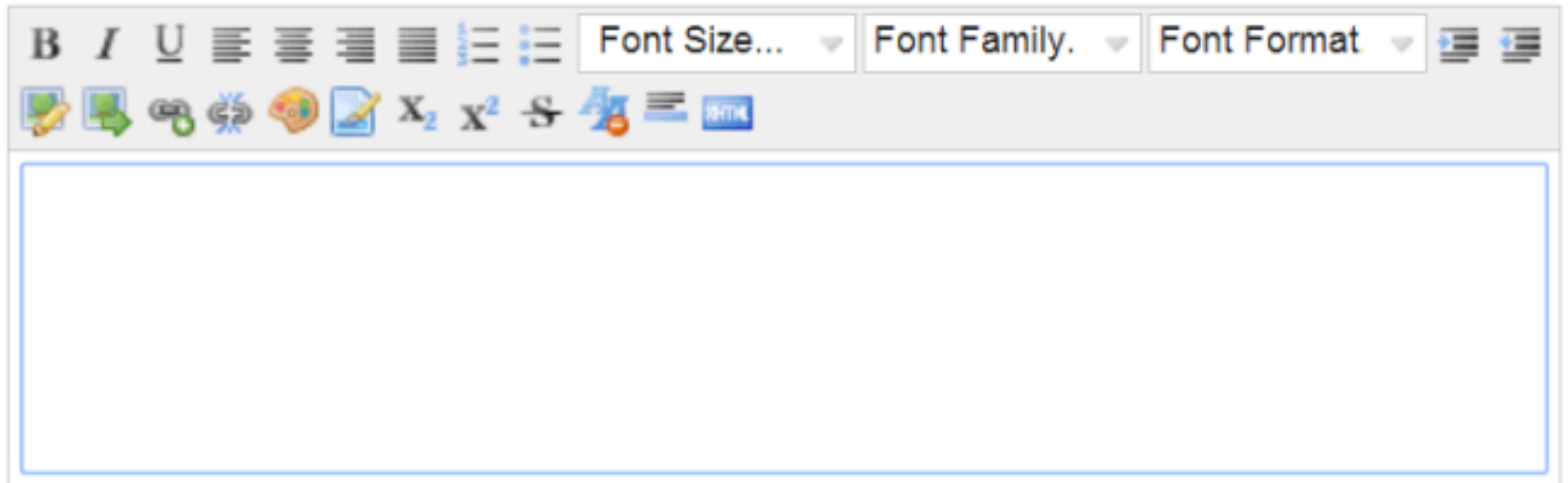
~~
<link href="http://.../Bank/Withdraw?Amount=9999">~~

THỰC HÀNH

Email	<input type="text" value="u"/>	The Email field is not a valid e-mail address.
ConfirmEmail	<input type="text" value="e"/>	'ConfirmEmail' and 'Email' do not match.
Age	<input type="text" value="-1"/>	The field Age must be between 16 and 65.
Salary	<input type="text"/>	The Salary field is required.
CreditCard	<input type="text" value="e"/>	The CreditCard field is not a valid credit card number.
Website	<input type="text" value="e"/>	The Website field is not a valid fully-qualified http, https, or ftp URL.
Photo	<input type="text" value="e"/>	
SaigonMotoNumber	<input type="text" value="e"/>	The field SaigonMotoNumber must match the regular expression '5\d-[A-Z]\d-((\d{4}))(\d{3}\d{2})'.
Description	<div><div>ApplicationMy ASP.NET MVC5</div><div>ApplicationMy ASP.NET MVC5</div><div>ApplicationMy ASP.NET MVC5</div><div>Application</div></div>	The field Description must be a string with a maximum length of 255.
<div>Save</div>		

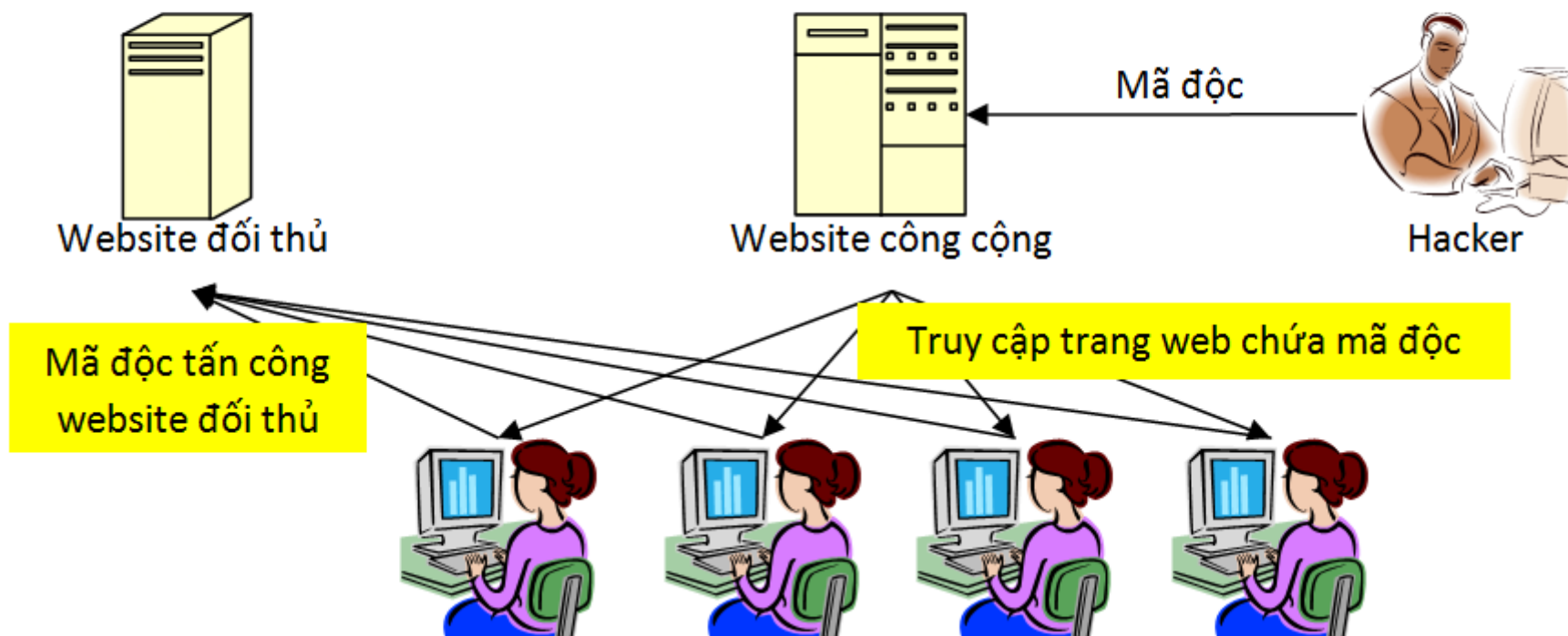
THỰC HÀNH

☐ NicEditor



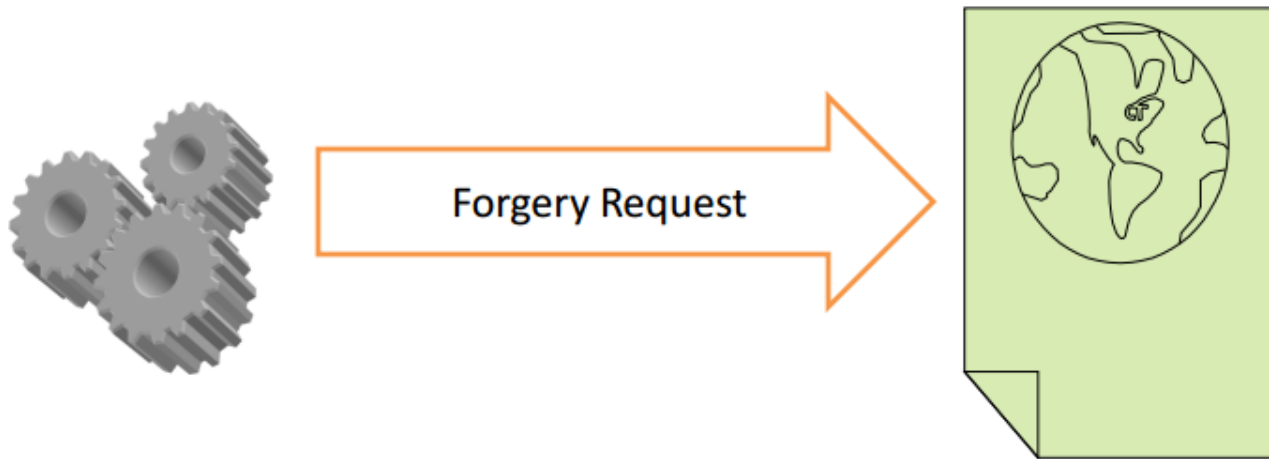
6. XSS

- ❑ Tấn công website bằng cách nhúng mã script vào một trang web của site khác



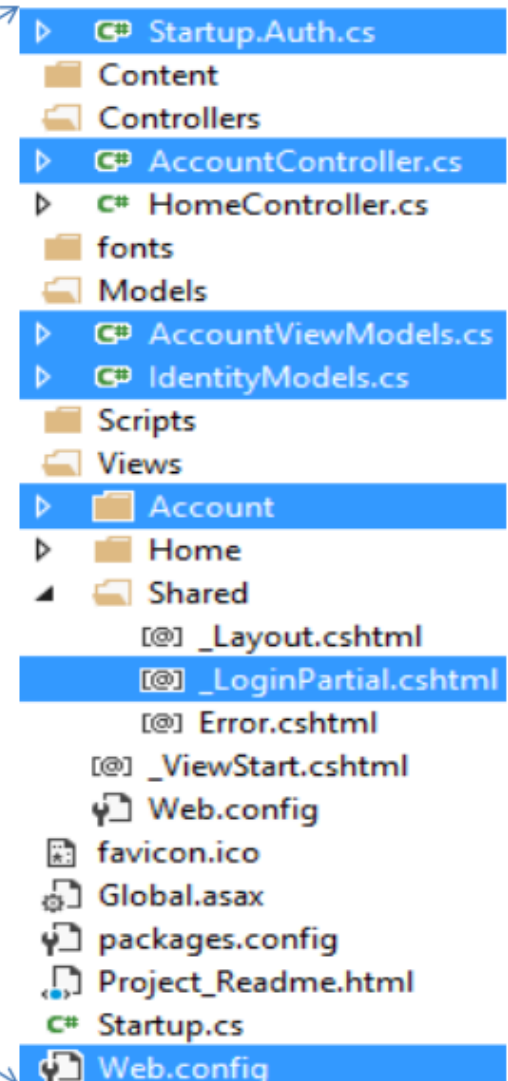
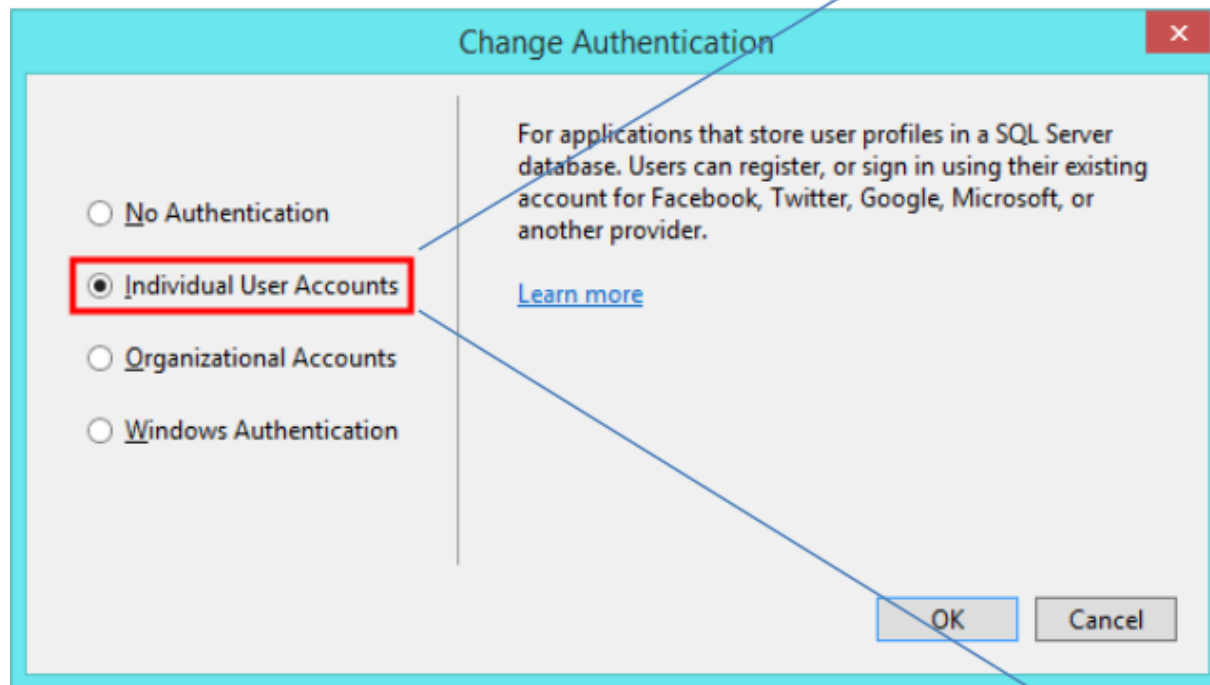
7. ANTIFORERY

- ❑ Giả mạo request để gửi dữ liệu tự động đến server để thực hiện hành động một cách tự động



8. AUTHENTICATION & AUTHORIZATION

- ☐ Kiểm soát đăng nhập
- ☐ Kiểm soát vai trò



8. AUTHENTICATION & AUTHORIZATION

Các thành phần Security

- ❑ **Startup.Auth.cs:**
 - ✂ Cấu hình trang đăng nhập và các nguồn đăng nhập bên ngoài
- ❑ **AccountController.cs:**
 - ✂ Định nghĩa các action đăng nhập, đăng xuất, đăng ký, đổi mật khẩu
- ❑ **AccountViewModels.cs:**
 - ✂ Các model buộc dữ liệu với giao diện
- ❑ **IdentityModels.cs:**
 - ✂ Model dữ liệu security như User, Role, UserInRoles
- ❑ **Account/*.cshtml:**
 - ✂ các view liên quan đến các action trong AccountController
- ❑ **Shared/_LoginPartial.cshtml:**
 - ✂ view thành phần nhúng vào layout
- ❑ **Web.config:**
 - ✂ chứa khai báo chuỗi kết nối đến CSDL

8. AUTHENTICATION & AUTHORIZATION

Tổ chức AccountController

Buộc phải đăng nhập mới sử dụng các action của controller này

[Authorize]

```
public class AccountController : Controller
{
    public AccountController()
        : this(new UserManager<ApplicationUser>(
            new UserStore<ApplicationUser>(new ApplicationDbContext())))
    {
    }

    public AccountController(UserManager<ApplicationUser> userManager)
    {
        UserManager = userManager;
    }

    public UserManager<ApplicationUser> UserManager { get; private set; }

    ...các action và mã hỗ trợ khác...
}
```

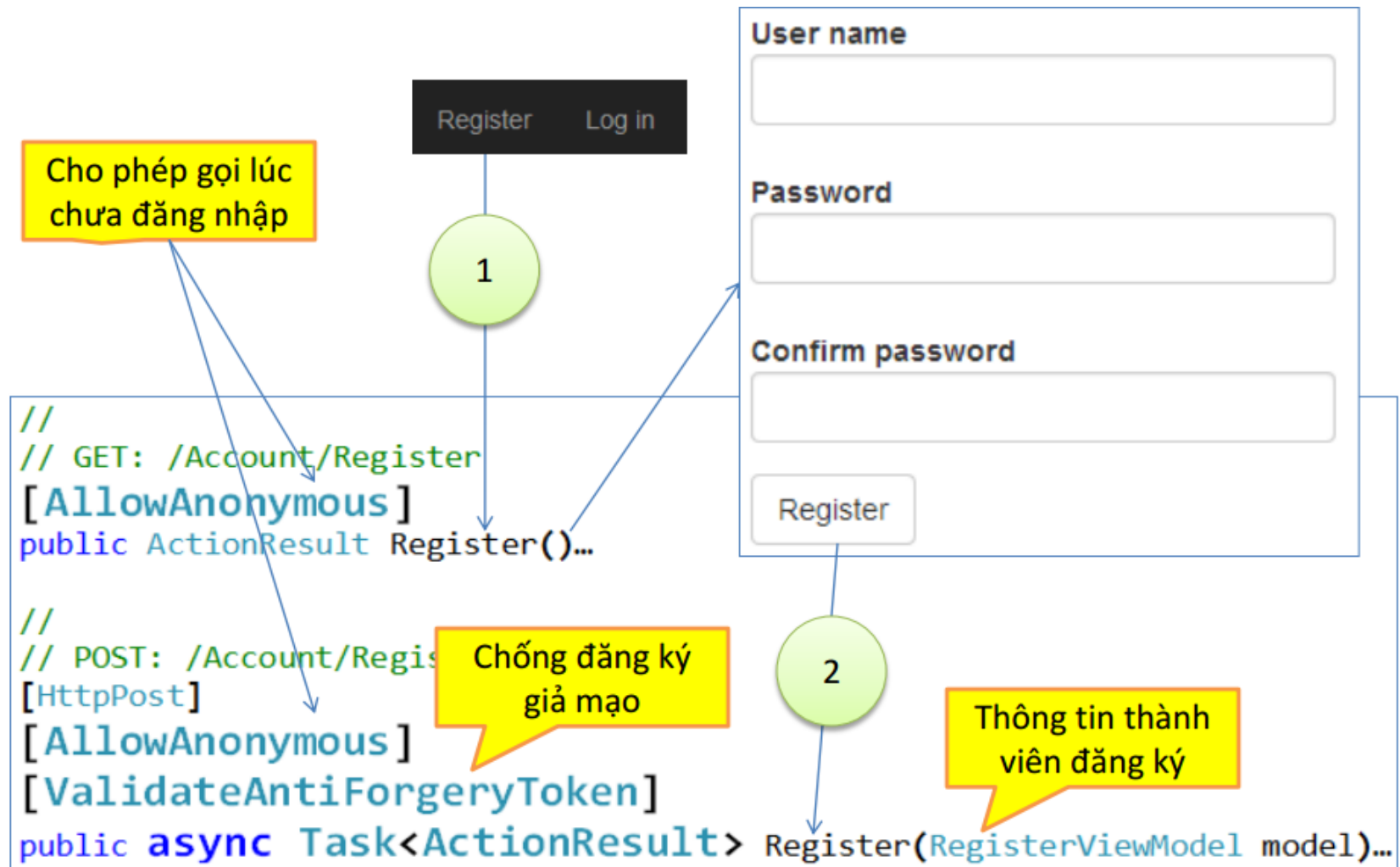
DbContext làm việc với CSDL thành viên

Sử dụng để quản lý thành viên

Các action được bảo vệ

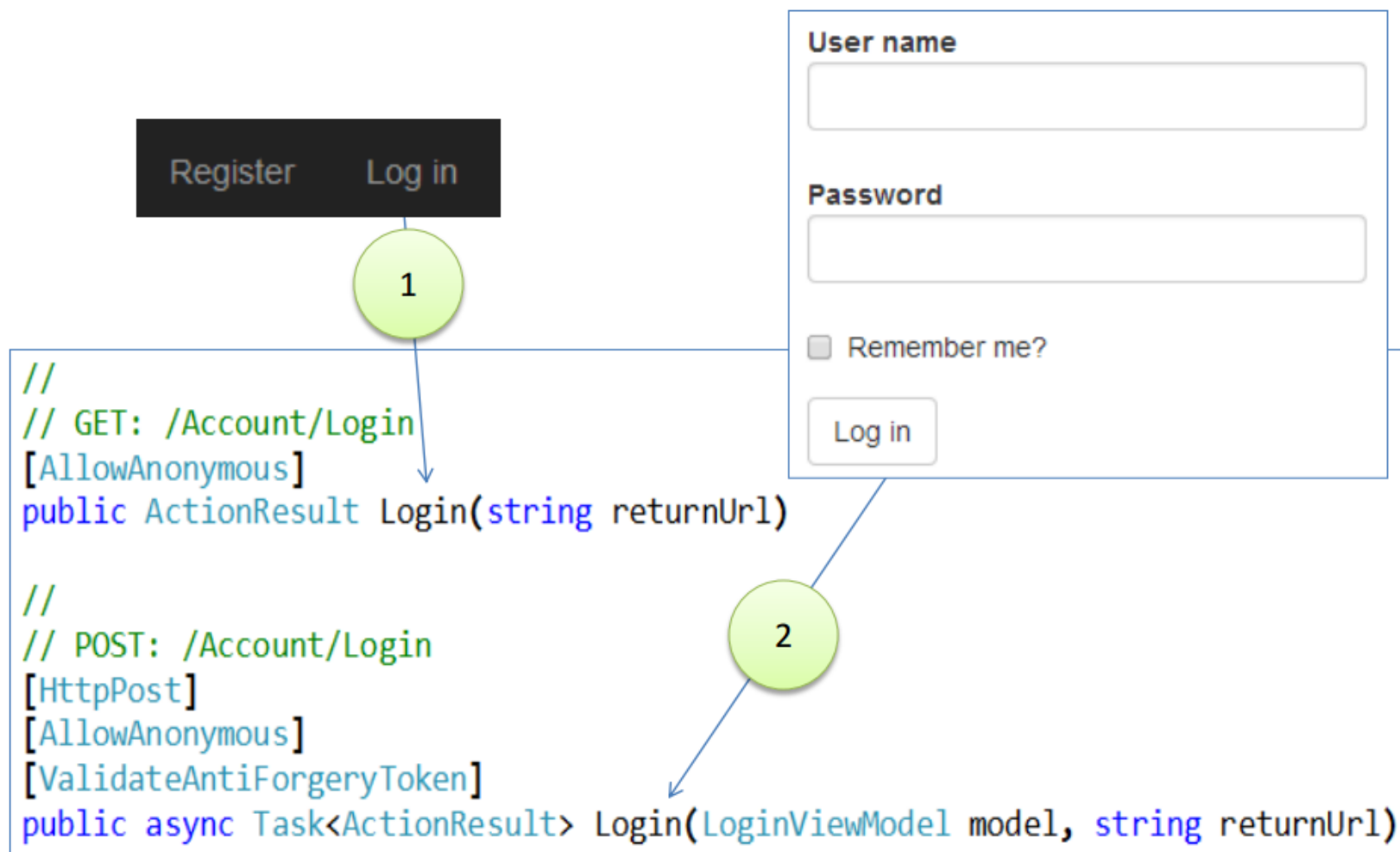
8. AUTHENTICATION & AUTHORIZATION

RegisterAction



8. AUTHENTICATION & AUTHORIZATION

Login Action



8. AUTHENTICATION & AUTHORIZATION

External Account

Use a local account to log in.

User name

Password

☐ Remember me

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

Use another service to log in.

8. AUTHENTICATION & AUTHORIZATION

Change Pass & LogOff

```
//  
// POST: /Account/LogOff  
[HttpPost]  
[ValidateAntiForgeryToken]  
public ActionResult LogOff()
```

Hello nnghiem! Log off

1

```
//  
// GET: /Account/Manage  
public ActionResult Manage(ManageMessageId? message)  
  
//  
// POST: /Account/Manage  
[HttpPost]  
[ValidateAntiForgeryToken]  
public async Task<ActionResult> Manage(ManageUserViewModel model)
```

Current password

New password

Confirm new password

Change password

2

8. AUTHENTICATION & AUTHORIZATION

IdentityModel.cs

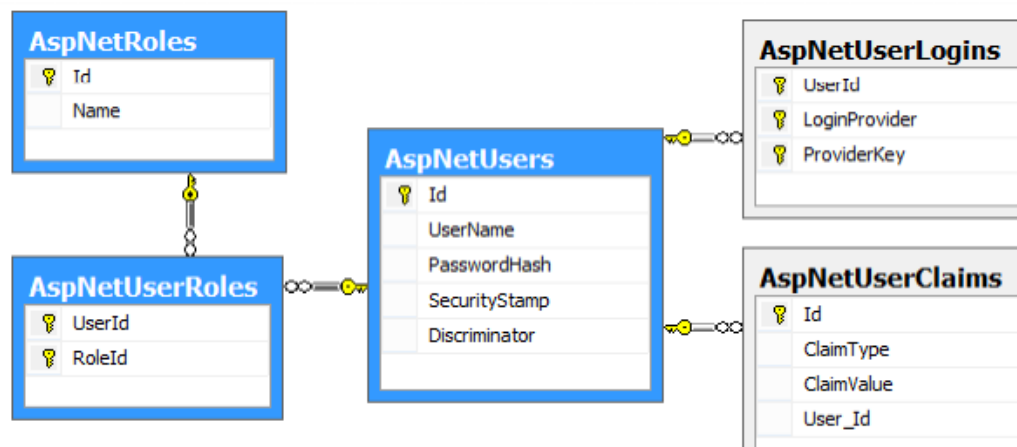
- ❑ Mô hình CSDL security
- ❑ ApplicationUser thông tin thành viên
- ❑ ApplicationDbContext là DbContext làm việc với CSDL thông qua kết nối DefaultConnection được khai trong Web.config.

```
public class ApplicationUser : IdentityUser
{
}

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

8. AUTHENTICATION & AUTHORIZATION

CSDL Thành viên




Bảng	Mô tả	Thực thể
AspNetUsers	Quản lý thông tin thành viên	IdentityUser
AspNetRoles	Quản lý vai trò	IdentityRole
AspNetUserRoles	Phân quyền – user nào có vai trò gì	IdentityUserRole
AppNetUserLogins	Thông tin thêm của tài khoản ngoài	IdentityUserLogin
AspNetUserClaims	Thông tin thêm của tài khoản Active Directory	IdentityUserClaim

8. AUTHENTICATION & AUTHORIZATION

Các thực thể

Phương thức/Thuộc tính	Mô tả
Id: string	Mã chuỗi tự tăng
UserName: string	Tên đăng nhập
Email: string	Email
PhoneNumber: string	Số điện thoại
Roles: ICollection<IdentityUserRole>	Danh sách roles của user



Thuộc tính/Phương thức	Mô tả
UserId	Mã thành viên
RoleId	Mã vai trò

Thuộc tính	Mô tả
Id: string	Mã vai trò
Name: string	Tên vai trò
Users: ICollection<IdentityUserRole>	Danh sách user thuộc vai trò này

8. AUTHENTICATION & AUTHORIZATION

Security API

☐ IdentityDbContext

✎ Quản lý CSDL thành viên

☐ UserManager<IdentityUser>

✎ Quản lý thành viên

☐ RoleManager<IdentityRole>

✎ Quản lý vai trò

☐ IAuthenticationManager

✎ Thông báo cho hệ thống về việc đăng nhập và đăng xuất của thành viên

8. AUTHENTICATION & AUTHORIZATION

Khởi tạo

❑ DB = new **IdentityDbContext()**

✍ Tạo DbContext

❑ UM = new **UserManager**<IdentityUser>(new
UserStore<IdentityUser>(db))

✍ Tạo UserManager

❑ RM = new **RoleManager**<IdentityRole>(new
RoleStore<IdentityRole>(db))

✍ Tạo RoleManager

❑ AU = **HttpContext.GetOwinContext().Authentication**

✍ Lấy authenticate

8. AUTHENTICATION & AUTHORIZATION

UserManger<IdentityUser> API

❑ Các thao tác quản lý thành viên đăng nhập

Phương thức/Thuộc tính	Mô tả
Users	Danh sách thành viên
Create()	Tạo thành viên mới
Delete()	Xóa thành viên
Update()	Cập nhật thông tin thành viên
Find()	Tìm theo thông tin đăng nhập (username, password)
FindById()	Tìm theo mã
FindByEmail()	Tìm theo email
FindByName()	Tìm theo username
ChangePassword()	Đổi mật khẩu
AddPassword()	Cấp mật khẩu mới
RemovePassword()	Xóa mật khẩu
CreateIdentity()	Tạo một ClaimIdentity kết hợp với một user
GetRoles()	Lấy danh sách vai trò của một user
AddToRole()	Thêm vai trò cho 1 user
RemoveFromRole()	Xóa vai trò khỏi 1 user
IsUserInRole()	Kiểm tra một vai trò của một user

8. AUTHENTICATION & AUTHORIZATION

RoleManager<IdentityRole> API

- ❑ Các thao tác quản lý vai trò thành viên

Thuộc tính/Phương thức	Mô tả
Roles	Danh sách vai trò
Create(), CreateAsync()	Tạo vai trò mới
Delete(), DeleteAsync()	Xóa vai trò
Update(), UpdateAsync()	Cập nhật vai trò
FindById(), FindByIdAsync()	Tìm vai trò theo mã
FindByName(), FindByNameAsync()	Tìm vai trò theo tên
RoleExists(), RoleExistsAsync()	Kiểm tra sự tồn tại của vai trò

8. AUTHENTICATION & AUTHORIZATION

IAuthenticationManager API

❑ Thao tác chính của IAuthenticationManager

Thuộc tính/Phương thức	Mô tả
SignIn()	Thông báo đăng nhập đến hệ thống
SignOut()	Thông báo đăng xuất đến hệ thống
GetExternalLoginInfoAsync()	Lấy thông tin đăng nhập bên ngoài

HẾT CHƯƠNG 7