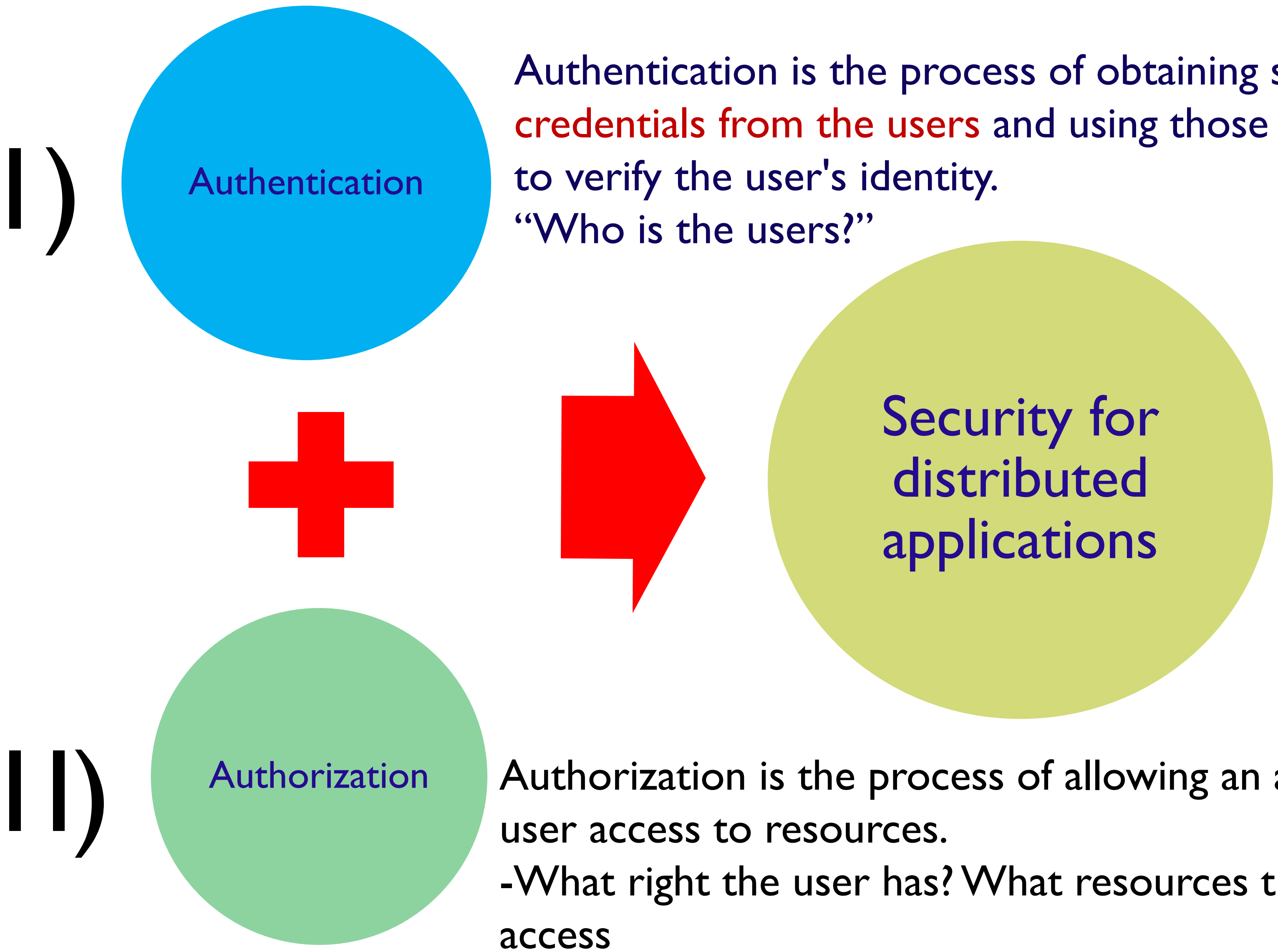
MCSE 541: Visual Programming

Authentication and Authorization in ASP.NET Core

Prof. Dr. Shamim Akhter
shamimakhter@iubat.edu

Two Interlinked Concepts

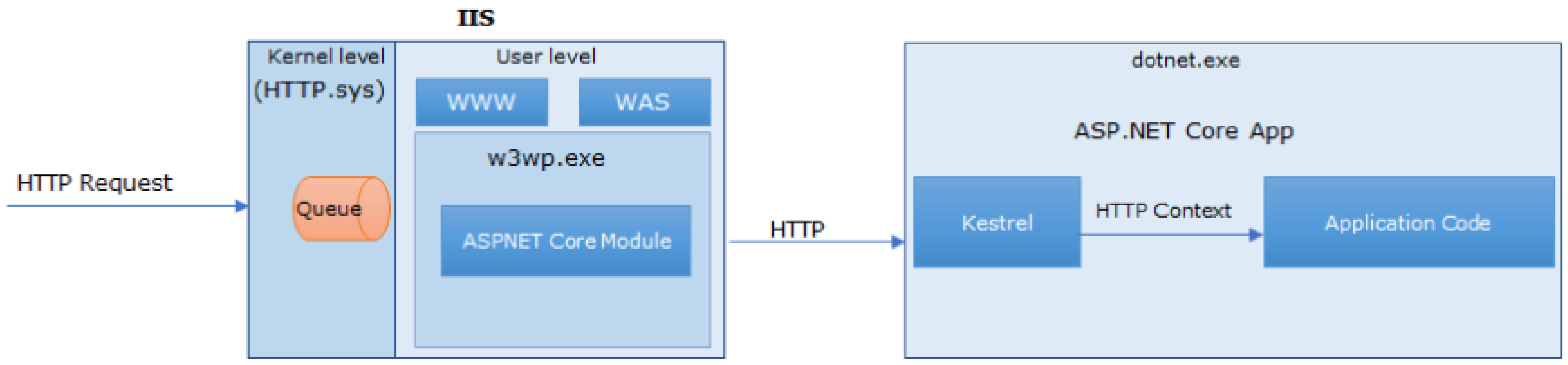


Authentication is the process of obtaining some sort of credentials from the users and using those credentials to verify the user's identity.
“Who is the users?”

Authorization is the process of allowing an authenticated user access to resources.
-What right the user has? What resources the user can access

How about **anonymous users** want to connect and use the application

Two Layers of Authentication – IIS and ASP.NET



How a request is served in this scenario:

1. The request is received by the *HTTP.sys* from the network.
2. If response is cached at *HTTP.sys* then it is sent back from there
   else gets a place the corresponding Application Pool's queue.

3. When a thread is available in the thread pool, it picks up the request and start processing it.

4. The request goes through IIS processing pipeline. Including few native IIS modules and once it reaches to **ANCM(ASPNET Core Module)**, it forwards the request to Kestrel (under dotnet.exe).

How a request is served.

5. ANCM has a responsibility to manage the process as well.
   - It (re)starts the process (if not running or crashed) and
   - IIS integration middleware configures the server to listen the request on port defined in environment variable.

   Server only accepts the requests which originates from ANCM.

   Note -Please do note that in ASP.NET Webforms/MVC the application is hosted under the worker process w3wp.exe which is managed by Windows Activation Service (WAS) which was part of IIS.

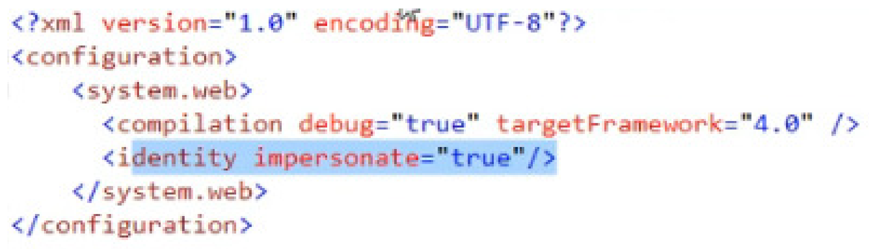
6. Once the request is received by Kestrel, it creates the ***HTTPContext object*** and request is handed over to ASP.NET Core middleware pipeline.

7. The request is passed to routing middleware which invokes the right controller and action method (model binding, various filters almost similar way as earlier versions).

8. Finally, the response is returned from the action and passed to kestrel via Middlewares and later sent back to client via IIS.

Steps in the joint IIS and ASP.net authentication process

1. IIS first checks to make sure the incoming request comes from an IP address that is allowed access to the domain. If not it denies the request.
2. Next IIS performs its own user authentication if it is configured to do so. By **default IIS allows anonymous access**, so requests are automatically authenticated, but you can change this default on a per - application basis with in IIS.



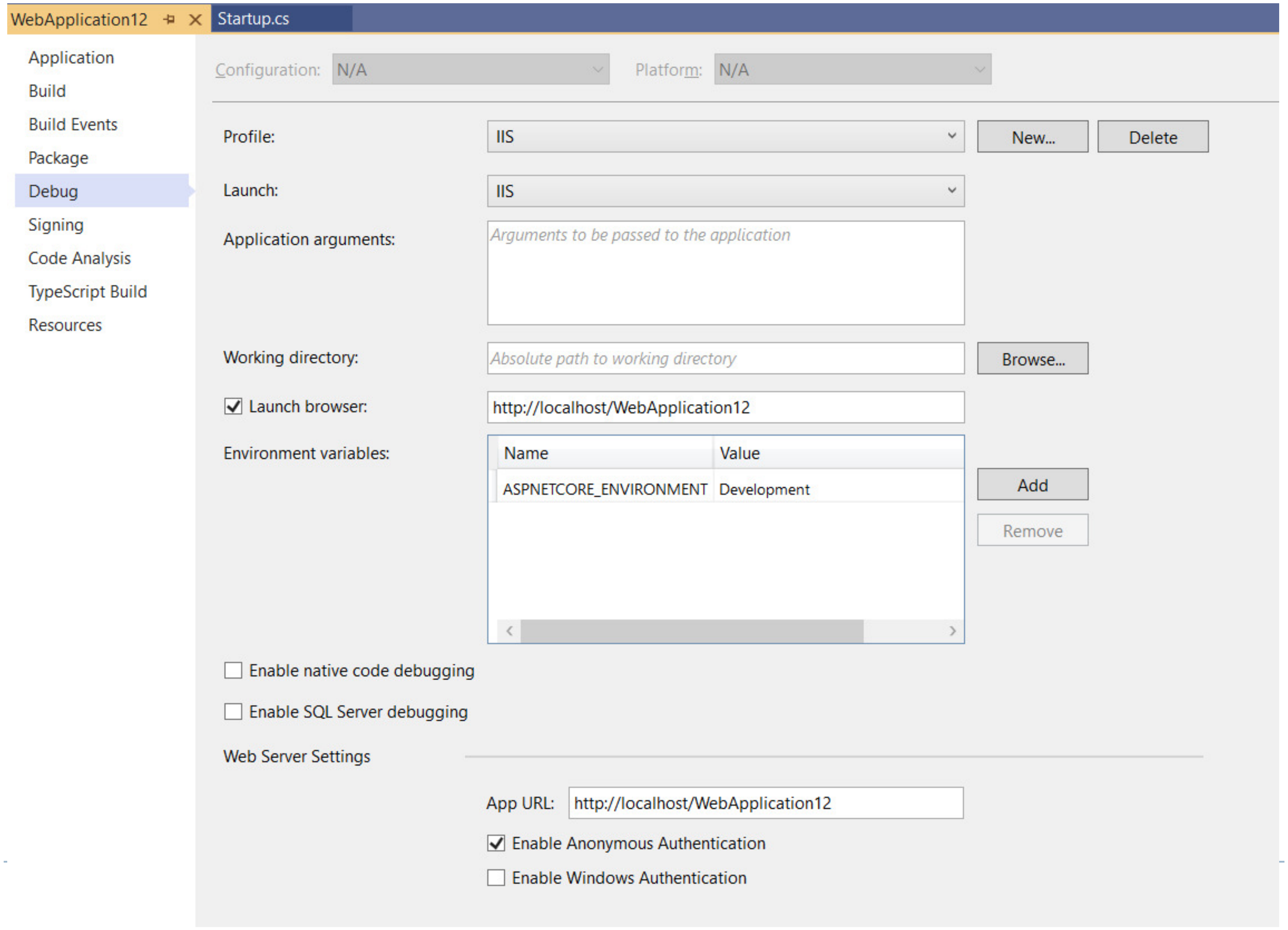
```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.web>
      <compilation debug="true" targetFramework="4.0" />
      <identity impersonate="true"/>
    </system.web>
</configuration>
```

3. If the request is passed to ASP.net with an authenticated user, ASP.net checks to see whether impersonation is enable. If impersonation is enabled, ASP.net acts as though it were the authenticated user using IUSR account. If not ASP.net acts with its own configured account using application pool identity .

4. Finally the identity from step 3 is used to request resources from the operating system. If ASP.net authentication can obtain all the necessary resources it grants the users request otherwise it is denied. Resources can include much more than just the ASP.net page itself you can also use .Net's code access security features to extend this authorization step to disk files, Registry keys and other resources.

Run A Program In IIS server

- Solution->Properties->Debug



Configuration: N/A Platform: N/A

Profile: IIS

Launch: IIS

Application arguments: Arguments to be passed to the application

Working directory: Absolute path to working directory

☑ Launch browser: http://localhost/WebApplication12

Environment variables:

Name	Value
ASPNETCORE_ENVIRONMENT	Development

☐ Enable native code debugging

☐ Enable SQL Server debugging

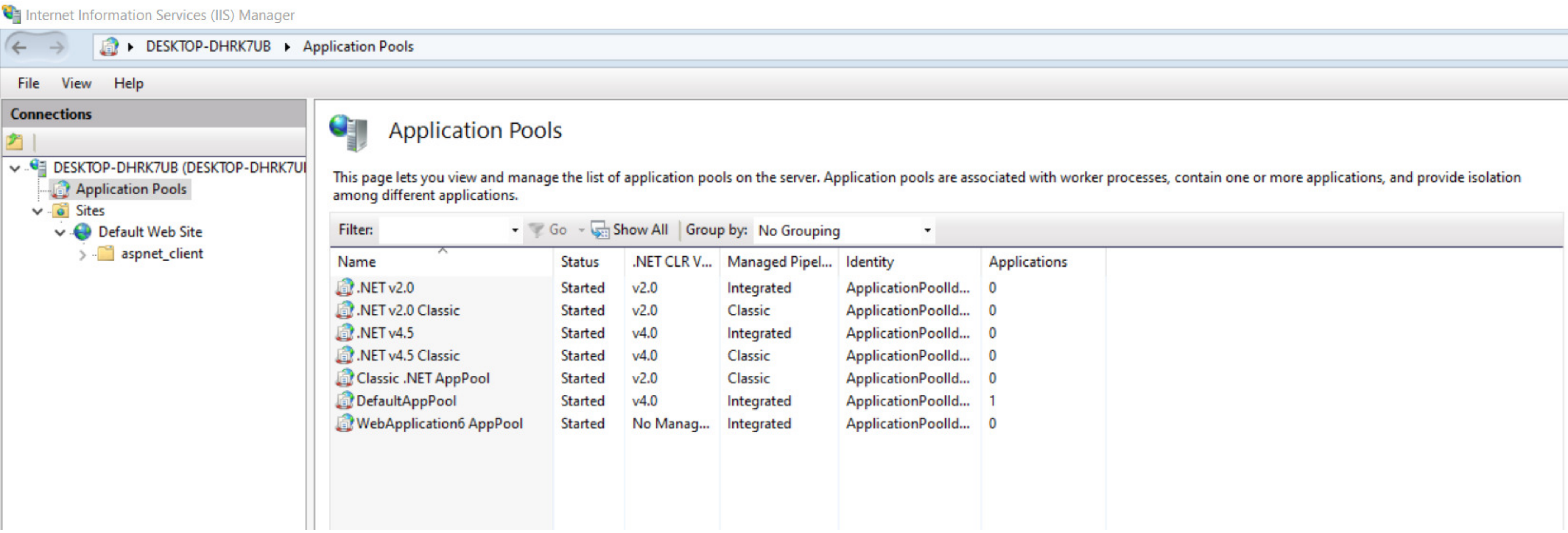
Web Server Settings

App URL: http://localhost/WebApplication12

☑ Enable Anonymous Authentication

☐ Enable Windows Authentication

IIS Application Pools



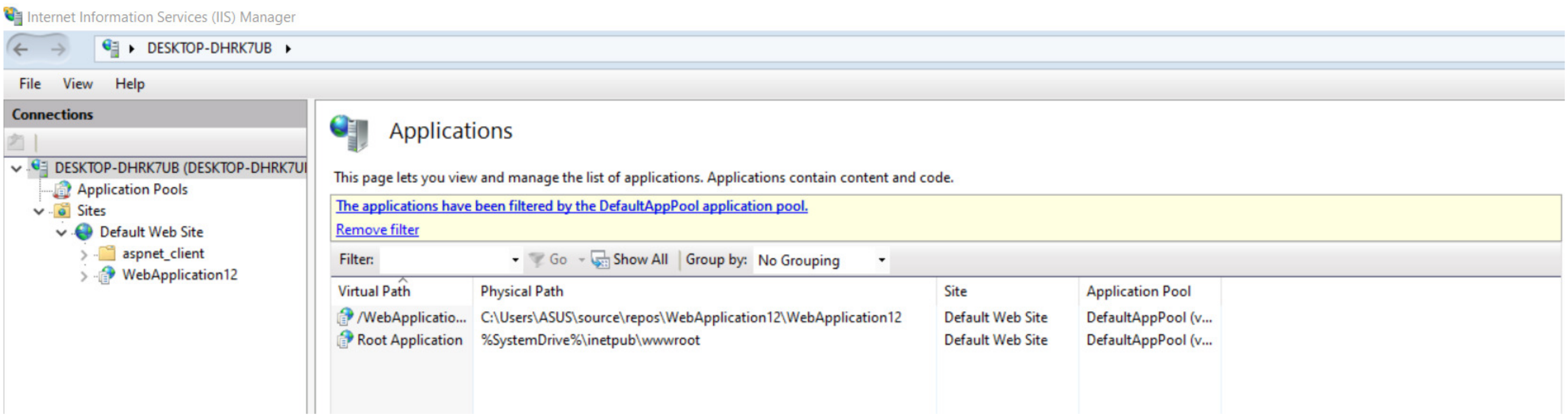
Application Pools

This page lets you view and manage the list of application pools on the server. Application pools are associated with worker processes, contain one or more applications, and provide isolation among different applications.

Name	Status	.NET CLR V...	Managed Pipel...	Identity	Applications
.NET v2.0	Started	v2.0	Integrated	ApplicationPoolId...	0
.NET v2.0 Classic	Started	v2.0	Classic	ApplicationPoolId...	0
.NET v4.5	Started	v4.0	Integrated	ApplicationPoolId...	0
.NET v4.5 Classic	Started	v4.0	Classic	ApplicationPoolId...	0
Classic .NET AppPool	Started	v2.0	Classic	ApplicationPoolId...	0
DefaultAppPool	Started	v4.0	Integrated	ApplicationPoolId...	1
WebApplication6 AppPool	Started	No Manag...	Integrated	ApplicationPoolId...	0

Each Application runs their own processes.

Running Program in DefaultAppPool



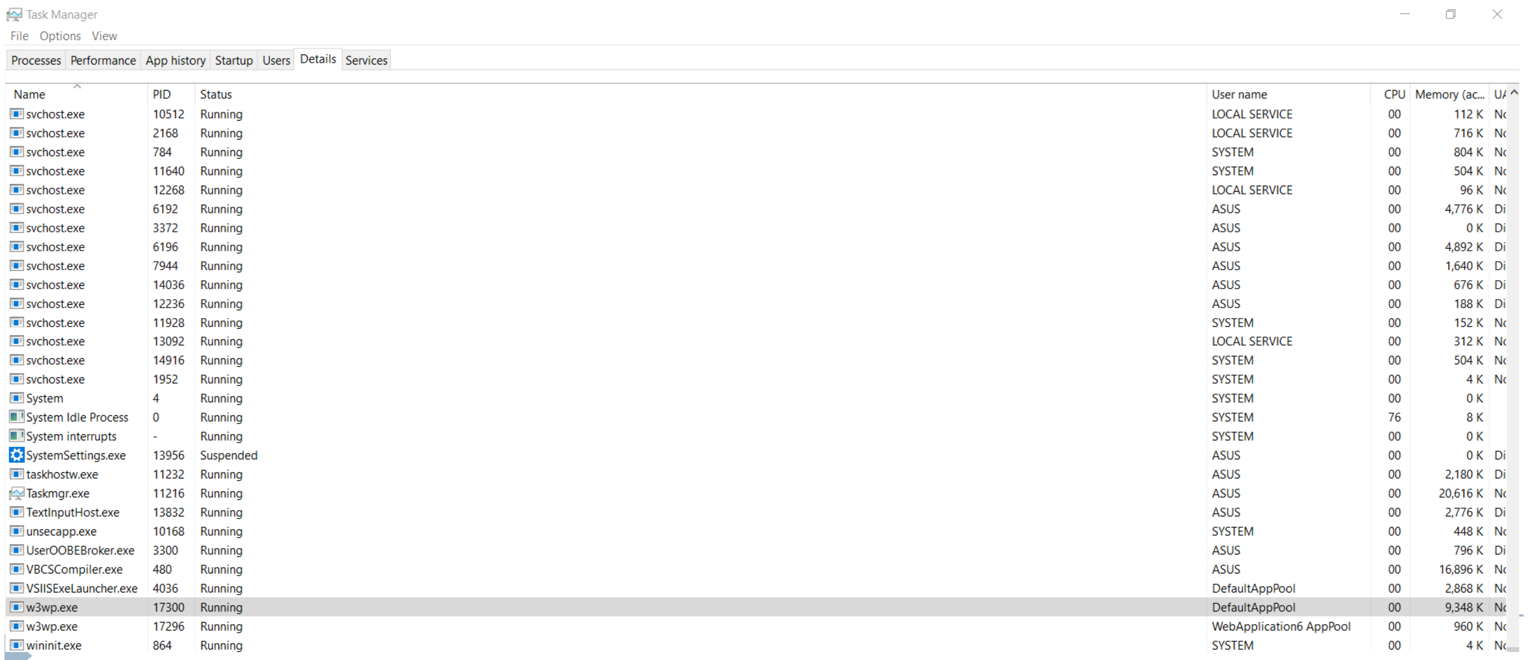
Applications

This page lets you view and manage the list of applications. Applications contain content and code.

The applications have been filtered by the DefaultAppPool application pool.

Remove filter

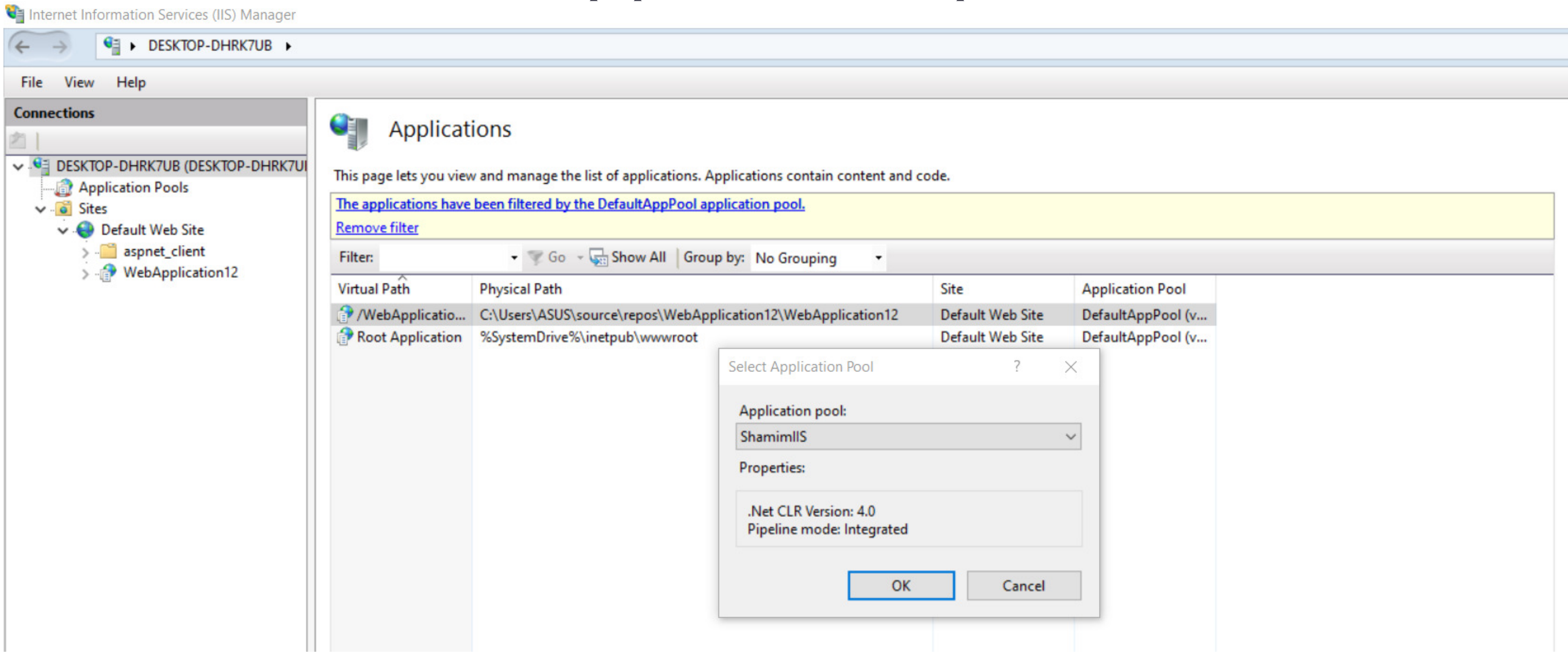
Virtual Path	Physical Path	Site	Application Pool
/WebApplicatio...	C:\Users\ASUS\source\repos\WebApplication12\WebApplication12	Default Web Site	DefaultAppPool (v...
Root Application	%SystemDrive%\inetpub\wwwroot	Default Web Site	DefaultAppPool (v...



Name	PID	Status	User name	CPU	Memory (ac...	UA
svchost.exe	10512	Running	LOCAL SERVICE	00	112 K	Nc
svchost.exe	2168	Running	LOCAL SERVICE	00	716 K	Nc
svchost.exe	784	Running	SYSTEM	00	804 K	Nc
svchost.exe	11640	Running	SYSTEM	00	504 K	Nc
svchost.exe	12268	Running	LOCAL SERVICE	00	96 K	Nc
svchost.exe	6192	Running	ASUS	00	4,776 K	Di
svchost.exe	3372	Running	ASUS	00	0 K	Di
svchost.exe	6196	Running	ASUS	00	4,892 K	Di
svchost.exe	7944	Running	ASUS	00	1,640 K	Di
svchost.exe	14036	Running	ASUS	00	676 K	Di
svchost.exe	12236	Running	ASUS	00	188 K	Di
svchost.exe	11928	Running	SYSTEM	00	152 K	Nc
svchost.exe	13092	Running	LOCAL SERVICE	00	312 K	Nc
svchost.exe	14916	Running	SYSTEM	00	504 K	Nc
svchost.exe	1952	Running	SYSTEM	00	4 K	Nc
System	4	Running	SYSTEM	00	0 K	
System Idle Process	0	Running	SYSTEM	76	8 K	
System interrupts	-	Running	SYSTEM	00	0 K	
SystemSettings.exe	13956	Suspended	ASUS	00	0 K	Di
taskhostw.exe	11232	Running	ASUS	00	2,180 K	Di
Taskmgr.exe	11216	Running	ASUS	00	20,616 K	Nc
TextInputHost.exe	13832	Running	ASUS	00	2,776 K	Di
unsecapp.exe	10168	Running	SYSTEM	00	448 K	Nc
UserOOBEBroker.exe	3300	Running	ASUS	00	796 K	Di
VBCSCompiler.exe	480	Running	ASUS	00	16,896 K	Nc
VSIISExeLauncher.exe	4036	Running	DefaultAppPool	00	2,868 K	Nc
w3wp.exe	17300	Running	DefaultAppPool	00	9,348 K	Nc
w3wp.exe	17296	Running	WebApplication6 AppPool	00	960 K	Nc
wininit.exe	864	Running	SYSTEM	00	4 K	Nc

End the task from Task Manager. The webpage will be closed also.

Create a new application pool



Internet Information Services (IIS) Manager

DESKTOP-DHRK7UB ▸

File View Help

Connections

- DESKTOP-DHRK7UB (DESKTOP-DHRK7UI
  - Application Pools
  - Sites
    - Default Web Site
      - aspnet_client
      - WebApplication12

Applications

This page lets you view and manage the list of applications. Applications contain content and code.

The applications have been filtered by the DefaultAppPool application pool.

Remove filter

Filter: Go Show All | Group by: No Grouping

Virtual Path	Physical Path	Site	Application Pool
/WebApplicatio...	C:\Users\ASUS\source\repos\WebApplication12\WebApplication12	Default Web Site	DefaultAppPool (v...
Root Application	%SystemDrive%\inetpub\wwwroot	Default Web Site	DefaultAppPool (v...

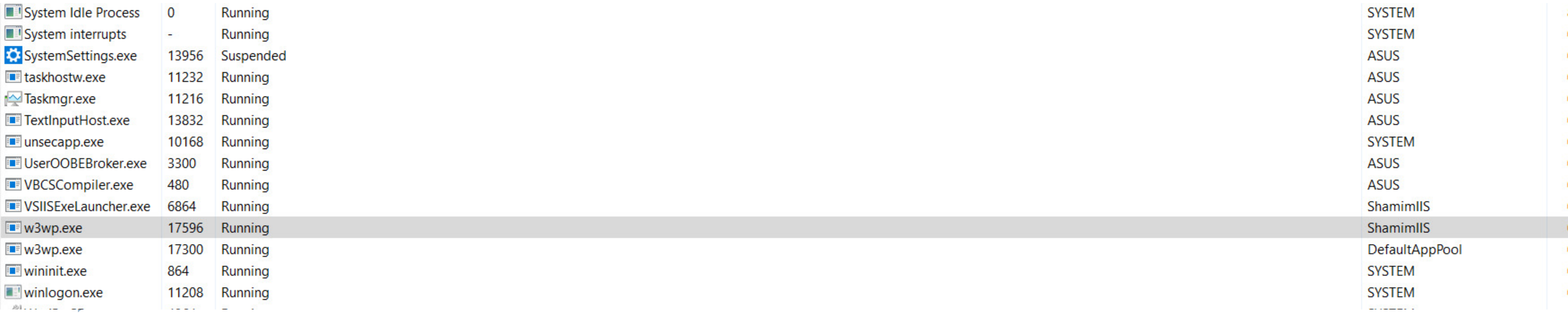
Select Application Pool

Application pool: ShamimIIS

Properties:

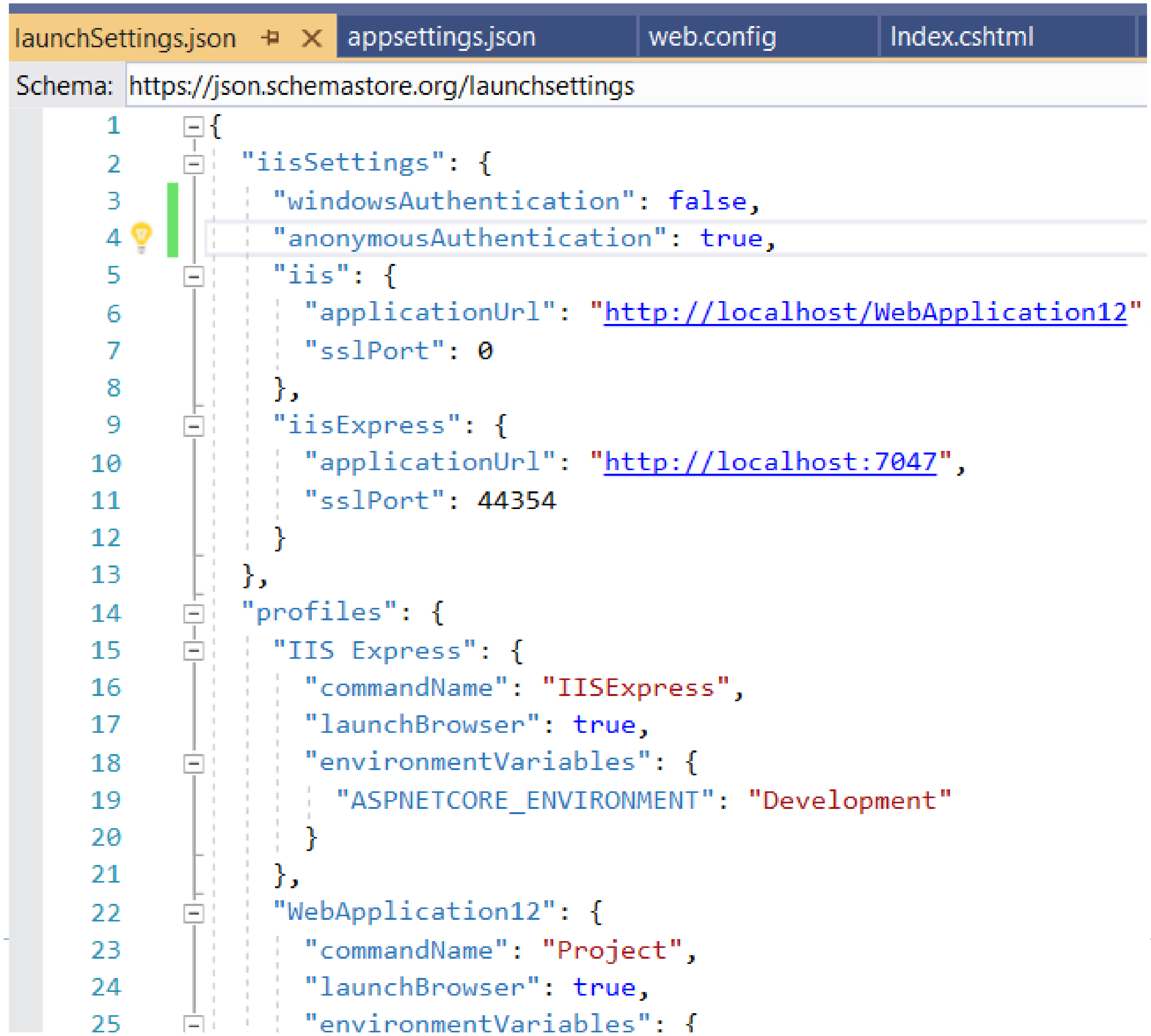
.Net CLR Version: 4.0
Pipeline mode: Integrated

OK Cancel



System Idle Process	0	Running	SYSTEM
System interrupts	-	Running	SYSTEM
SystemSettings.exe	13956	Suspended	ASUS
taskhostw.exe	11232	Running	ASUS
Taskmgr.exe	11216	Running	ASUS
TextInputHost.exe	13832	Running	ASUS
unsecapp.exe	10168	Running	SYSTEM
UserOOBEBroker.exe	3300	Running	ASUS
VBCSCompiler.exe	480	Running	ASUS
VSIISExeLauncher.exe	6864	Running	ShamimIIS
w3wp.exe	17596	Running	ShamimIIS
w3wp.exe	17300	Running	DefaultAppPool
wininit.exe	864	Running	SYSTEM
winlogon.exe	11208	Running	SYSTEM

Anonymous User Enable



Schema: https://json.schemastore.org/launchsettings

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iis": {
      "applicationUrl": "http://localhost/WebApplication12"
      "sslPort": 0
    },
    "iisExpress": {
      "applicationUrl": "http://localhost:7047",
      "sslPort": 44354
    }
  },
  "profiles": {
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "WebApplication12": {
      "commandName": "Project",
      "launchBrowser": true,
      "environmentVariables": {
```

User Authenticate as PoolIdentity

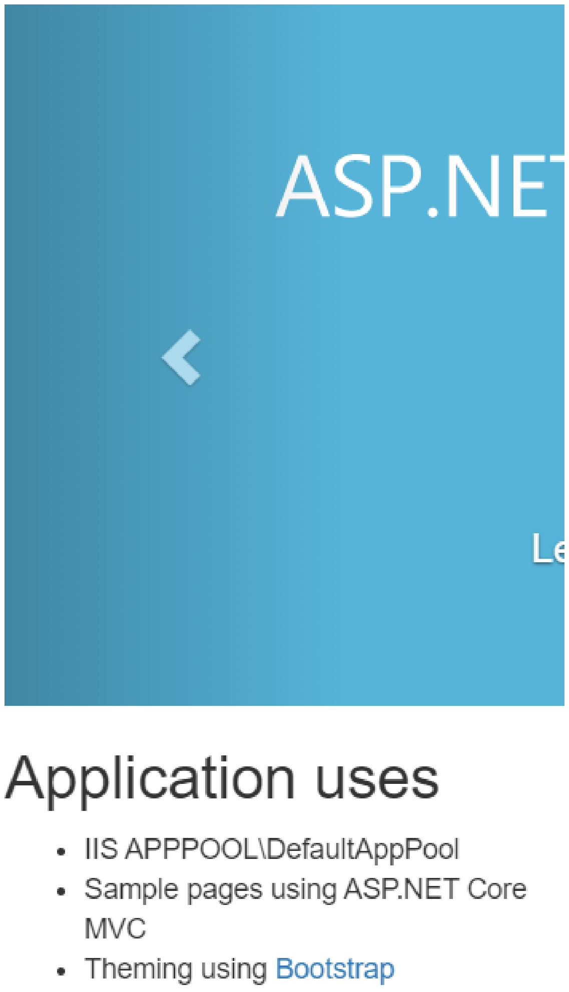
```
ViewBag.Name=System.Security.Principal.WindowsIdentity.GetCurrent().Name;
        ViewBag.Idenity = User.Identity.IsAuthenticated.ToString();
```

ApplicationPoolIdentity

The least privileged one.

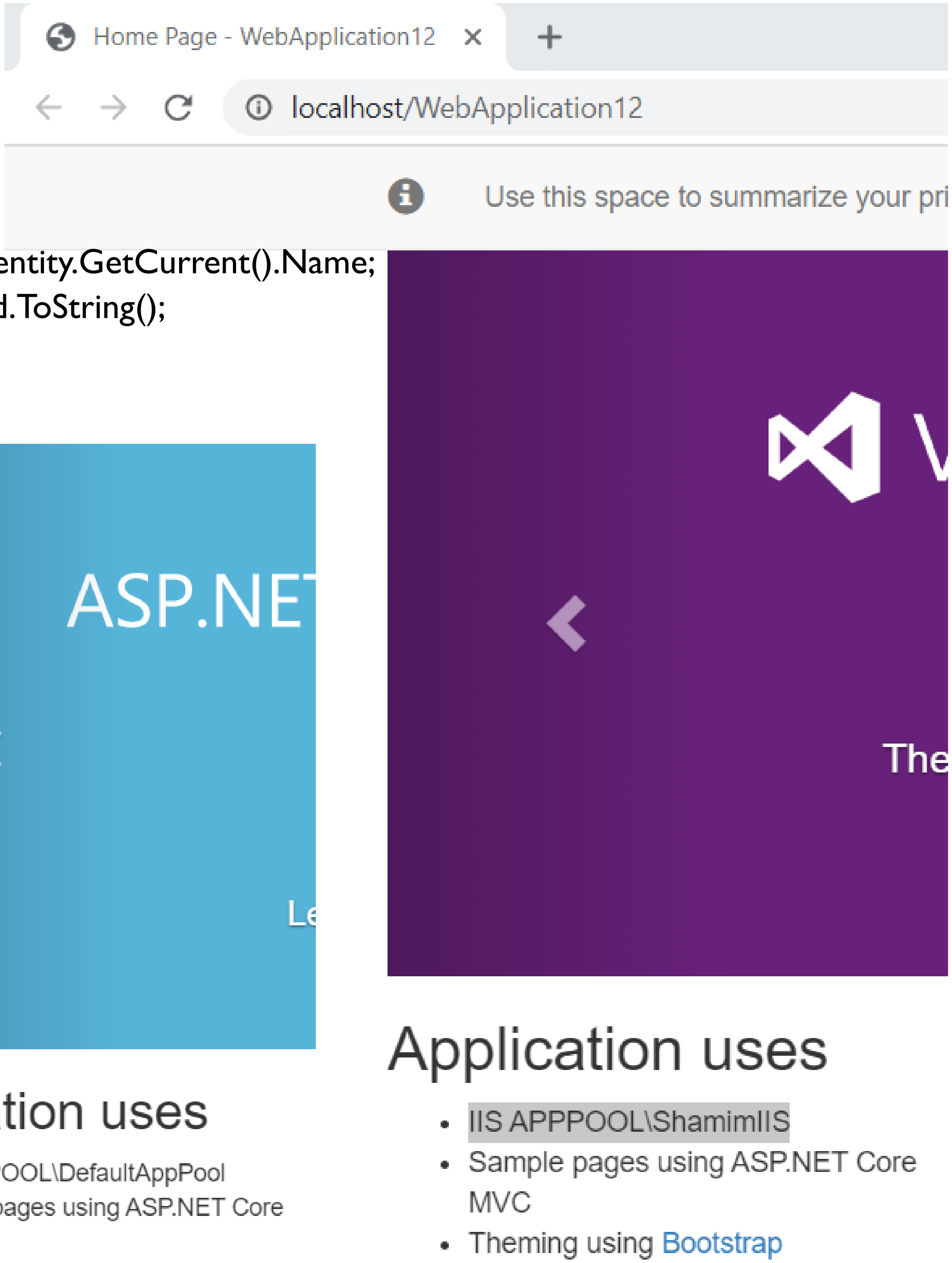
Create a Virtual account for each new application pool and run worker processes under this account.

Running an application with low privileged one is good practice



Application uses

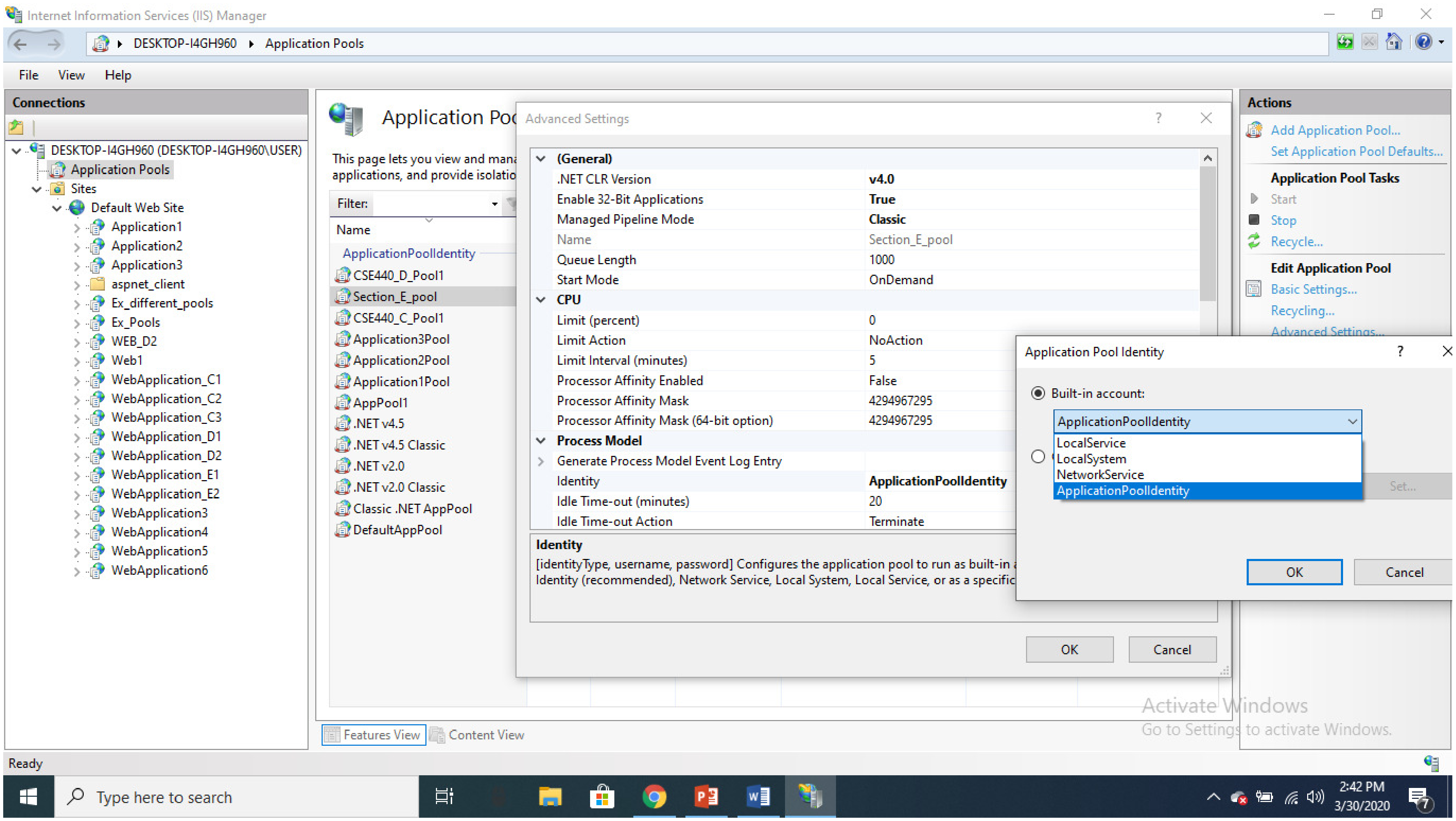
- IIS APPPOOL\DefaultAppPool
- Sample pages using ASP.NET Core MVC
- Theming using Bootstrap


Home Page - WebApplication12
localhost/WebApplication12
Use this space to summarize your pri
The

Application uses

- IIS APPPOOL\ShamimIIS
- Sample pages using ASP.NET Core MVC
- Theming using Bootstrap

Types of Identity with Pool


Internet Information Services (IIS) Manager

DESKTOP-I4GH960 ▸ Application Pools

File View Help

Connections
- DESKTOP-I4GH960 (DESKTOP-I4GH960\USER)
  - Application Pools
  - Sites
    - Default Web Site
      - Application1
      - Application2
      - Application3
      - aspnet_client
      - Ex_different_pools
      - Ex_Pools
      - WEB_D2
      - Web1
      - WebApplication_C1
      - WebApplication_C2
      - WebApplication_C3
      - WebApplication_D1
      - WebApplication_D2
      - WebApplication_E1
      - WebApplication_E2
      - WebApplication3
      - WebApplication4
      - WebApplication5
      - WebApplication6

Application Poo

This page lets you view and mana applications, and provide isolatio

Filter:

Name
- ApplicationPoolIdentity
- CSE440_D_Pool1
- Section_E_pool
- CSE440_C_Pool1
- Application3Pool
- Application2Pool
- Application1Pool
- AppPool1
- .NET v4.5
- .NET v4.5 Classic
- .NET v2.0
- .NET v2.0 Classic
- Classic .NET AppPool
- DefaultAppPool

Advanced Settings

(General)	
.NET CLR Version	**v4.0**
Enable 32-Bit Applications	**True**
Managed Pipeline Mode	**Classic**
Name	Section_E_pool
Queue Length	1000
Start Mode	OnDemand
CPU	
Limit (percent)	0
Limit Action	NoAction
Limit Interval (minutes)	5
Processor Affinity Enabled	False
Processor Affinity Mask	4294967295
Processor Affinity Mask (64-bit option)	4294967295
Process Model	
Generate Process Model Event Log Entry	
Identity	**ApplicationPoolIdentity**
Idle Time-out (minutes)	20
Idle Time-out Action	Terminate

Identity
[identityType, username, password] Configures the application pool to run as built-in a Identity (recommended), Network Service, Local System, Local Service, or as a specific

OK Cancel

Application Pool Identity

● Built-in account:

ApplicationPoolIdentity
- LocalService
- LocalSystem
- NetworkService
- ApplicationPoolIdentity

○

Set...

OK Cancel

Actions
- Add Application Pool...
- Set Application Pool Defaults...

Application Pool Tasks
- Start
- Stop
- Recycle...

Edit Application Pool
- Basic Settings...
- Recycling...
- Advanced Settings...

Features View | Content View

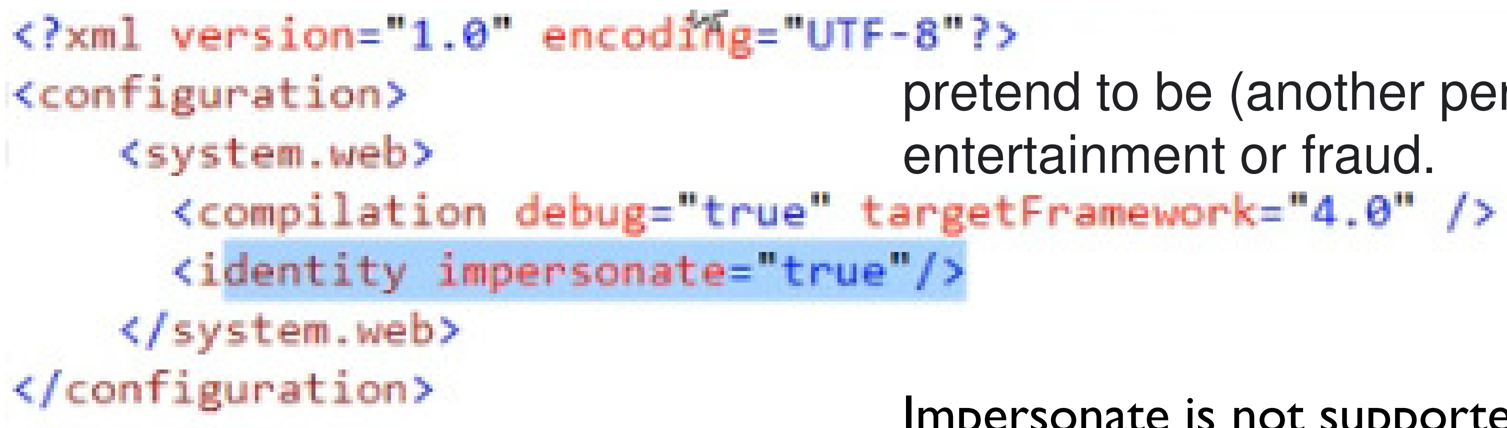
Activate Windows
Go to Settings to activate Windows.

Ready

Type here to search

2:42 PM
3/30/2020

Impersonate enable ASP.NET

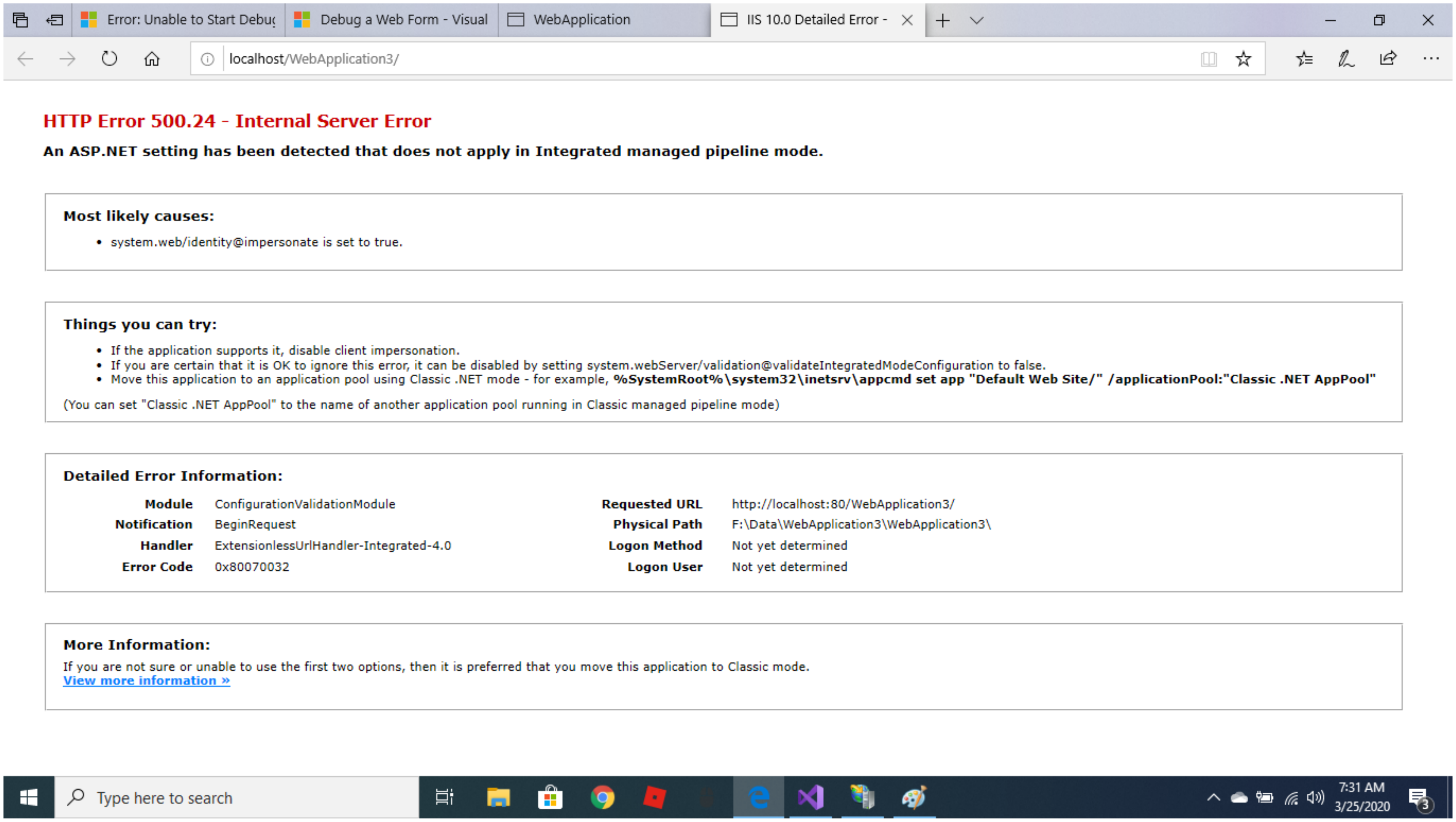


```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.web>
      <compilation debug="true" targetFramework="4.0" />
      <identity impersonate="true"/>
    </system.web>
</configuration>
```

pretend to be (another person) for entertainment or fraud.

Impersonate is not supported in ASP.NET Core

```
<configuration>
  <system.web>
    <identity impersonate="true"/>
    <compilation debug="true" targetFramework="4.7.2" />
    <httpRuntime targetFramework="4.7.2" />
  </system.web>
  <system.codedom>
    <compilers>
      <compiler language="c#;cs;csharp" extension=".cs"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.CSharpCodeProvider
, Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:1659;1699;1701" />
      <compiler language="vb;vbs;visualbasic;vbscript" extension=".vb"
type="Microsoft.CodeDom.Providers.DotNetCompilerPlatform.VBCodeProvider,
Microsoft.CodeDom.Providers.DotNetCompilerPlatform, Version=2.0.1.0,
Culture=neutral, PublicKeyToken=31bf3856ad364e35" warningLevel="4"
compilerOptions="/langversion:default /nowarn:41008
/define:_MYTYPE=\&quot;Web\&quot; /optionInfer+" />
    </compilers>
  </system.codedom>
</configuration>
```



Error: Unable to Start Debu
Debug a Web Form - Visual
WebApplication
IIS 10.0 Detailed Error -
localhost/WebApplication3/
Type here to search
7:31 AM
3/25/2020

HTTP Error 500.24 - Internal Server Error

An ASP.NET setting has been detected that does not apply in Integrated managed pipeline mode.

Most likely causes:

- system.web/identity@impersonate is set to true.

Things you can try:

- If the application supports it, disable client impersonation.
- If you are certain that it is OK to ignore this error, it can be disabled by setting system.webServer/validation@validateIntegratedModeConfiguration to false.
- Move this application to an application pool using Classic .NET mode - for example, **%SystemRoot%\system32\inetsrv\appcmd set app "Default Web Site/" /applicationPool:"Classic .NET AppPool"**

(You can set "Classic .NET AppPool" to the name of another application pool running in Classic managed pipeline mode)

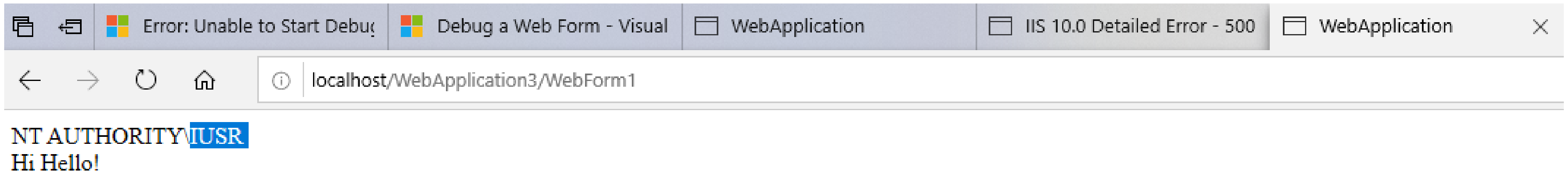
Detailed Error Information:

Module	ConfigurationValidationModule	**Requested URL**	http://localhost:80/WebApplication3/
Notification	BeginRequest	**Physical Path**	F:\Data\WebApplication3\WebApplication3\
Handler	ExtensionlessUrlHandler-Integrated-4.0	**Logon Method**	Not yet determined
Error Code	0x80070032	**Logon User**	Not yet determined

More Information:

If you are not sure or unable to use the first two options, then it is preferred that you move this application to Classic mode.
View more information »

Change pool to classical mode


Error: Unable to Start Debu
Debug a Web Form - Visual
WebApplication
IIS 10.0 Detailed Error - 500
WebApplication
localhost/WebApplication3/WebForm1

```
NT AUTHORITY\IUSR
Hi Hello!
```

Authentication providers

- The ASP.net architecture includes
  - two(2) different authentication providers
    - Windows Authentication provider
      - Authenticates users based on their windows accounts.
      - This provider uses IIS to perform the authentication and then passes the authenticated identity to the code.
      - This is the default provided for ASP.net.
    - Forms authentication provider
      - uses custom HTML forms to collect authentication information and lets you use your own logic to authenticate users.
      - The user's credentials are stored in a cookie for use during the session.

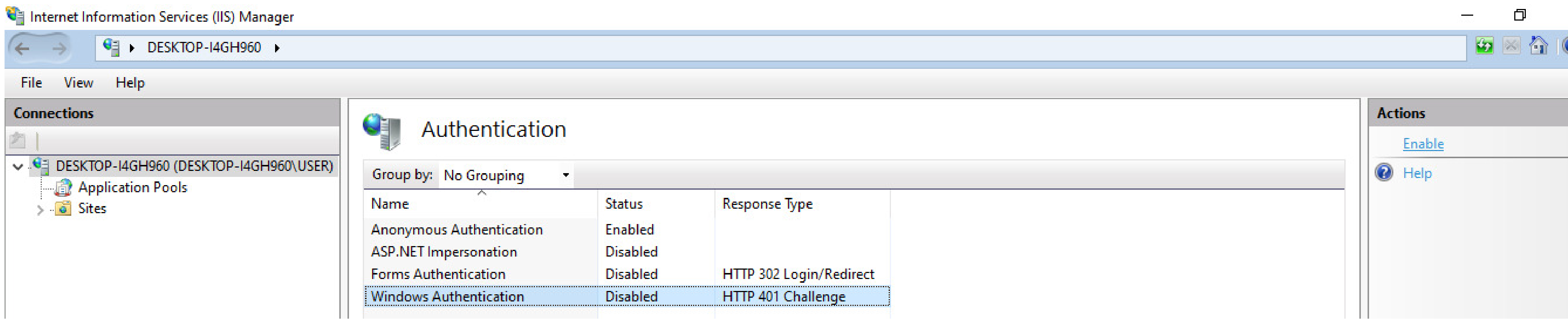
Selecting an authentication provider is as simple as making an entry in the web.config file for the application.

```
<authentication mode="windows">
```

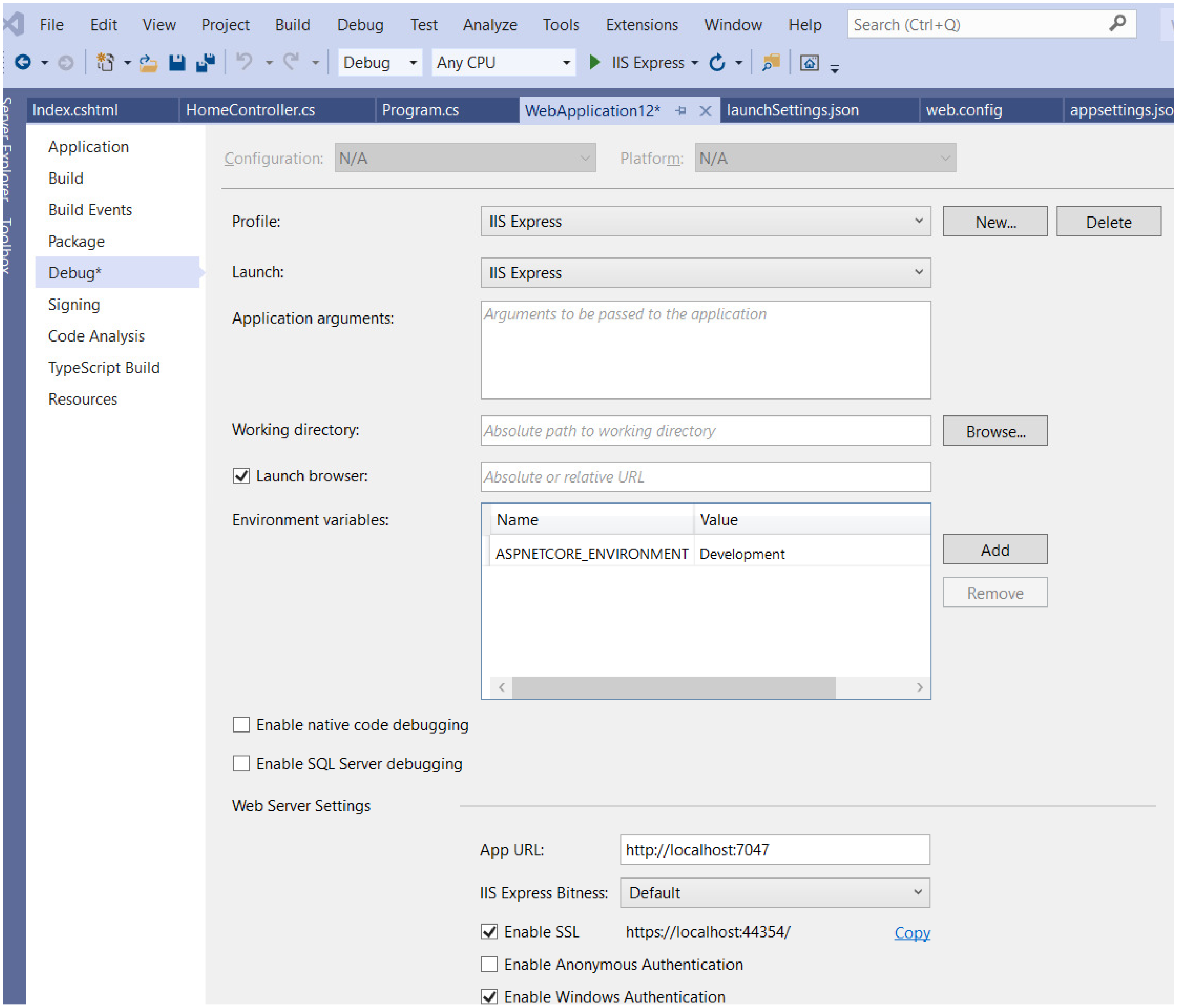
```
<authentication mode="forms">
```

Windows Authentication

- Enable it in IIS



Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Disabled	HTTP 401 Challenge



Configuration: N/A   Platform: N/A

Profile: IIS Express

Launch: IIS Express

Application arguments: Arguments to be passed to the application

Working directory: Absolute path to working directory

[✓] Launch browser: Absolute or relative URL

Environment variables:

Name	Value
ASPNETCORE_ENVIRONMENT	Development

[ ] Enable native code debugging

[ ] Enable SQL Server debugging

Web Server Settings

App URL: http://localhost:7047

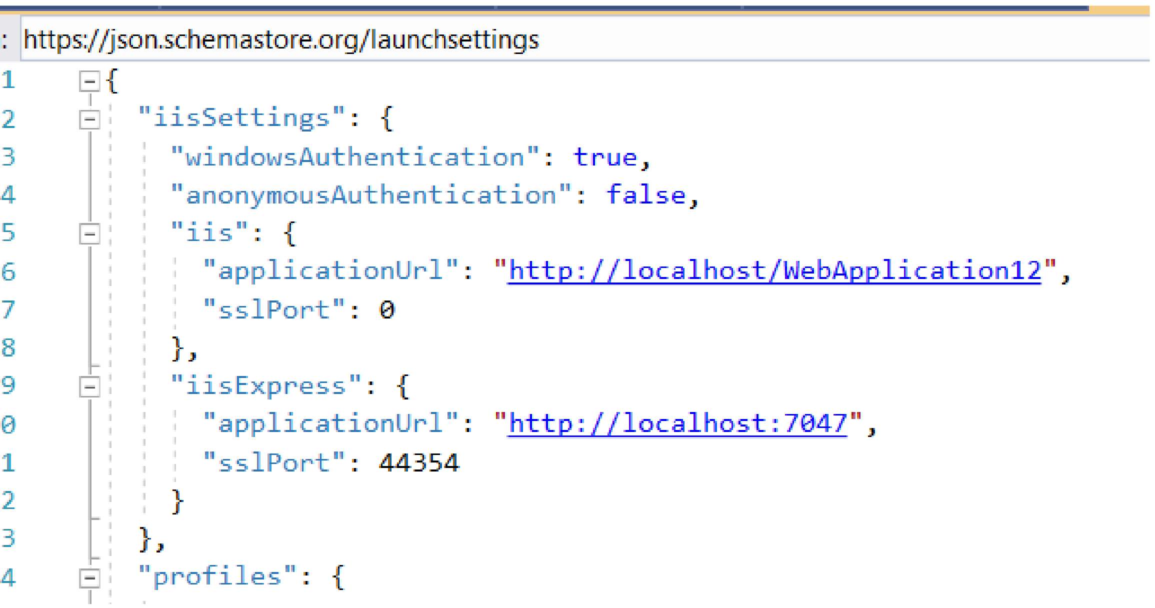
IIS Express Bitness: Default

[✓] Enable SSL   https://localhost:44354/

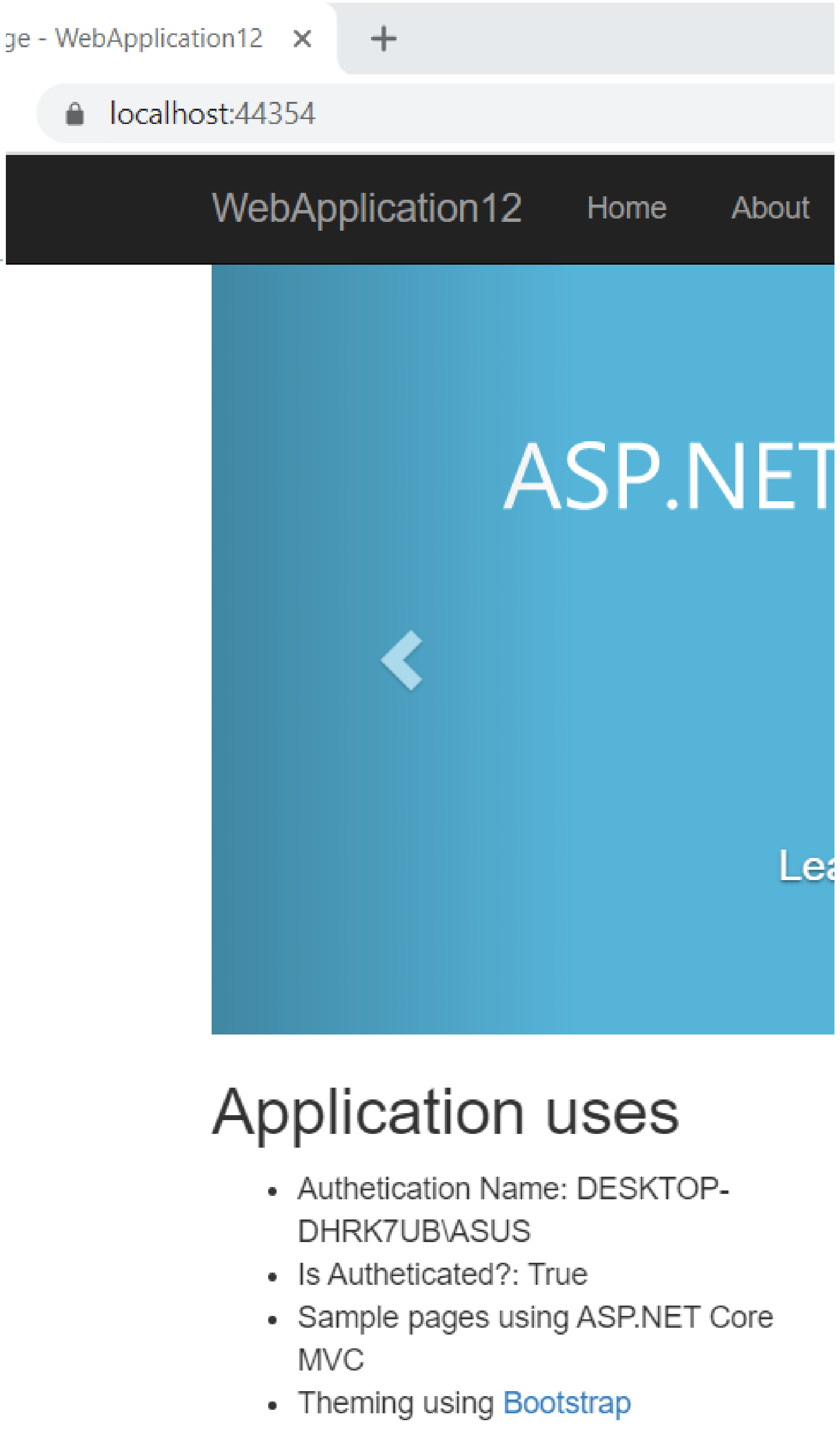
[ ] Enable Anonymous Authentication

[✓] Enable Windows Authentication

Windows Authentication IIS Express



```
: https://json.schemastore.org/launchsettings
{
  "iisSettings": {
    "windowsAuthentication": true,
    "anonymousAuthentication": false,
    "iis": {
      "applicationUrl": "http://localhost/WebApplication12",
      "sslPort": 0
    },
    "iisExpress": {
      "applicationUrl": "http://localhost:7047",
      "sslPort": 44354
    }
  },
  "profiles": {
```



Application uses

- Authetication Name: DESKTOP-DHRK7UB\ASUS
- Is Autheticated?: True
- Sample pages using ASP.NET Core MVC
- Theming using Bootstrap

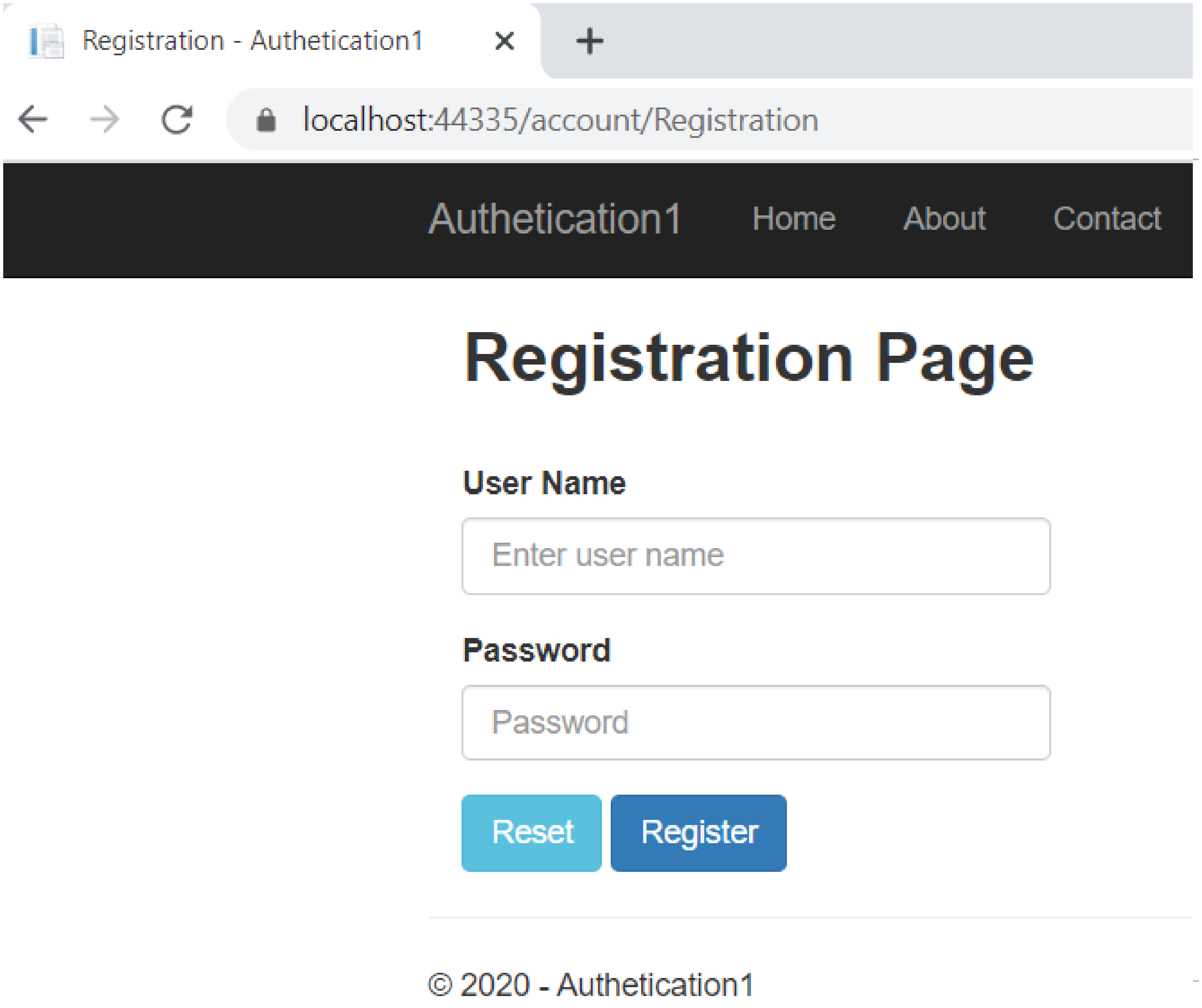
Forms Authentication

- Is used for Internet Web Applications.
- User do not have to be member of domain-based network (Windows Authentication)
- Authenticate users by using their own code and then maintain an authentication token in a cookie or in the page URL.
- Gmail.com, Amazon.com, Facebook.com are examples

How to use forms authentication?

- Create a registration page

- Create a login page
- Collects the credentials from the users use predefine code to authenticate the credentials.


Registration - Authetication1
localhost:44335/account/Registration
Authetication1
Home
About
Contact

Registration Page

User Name

Enter user name

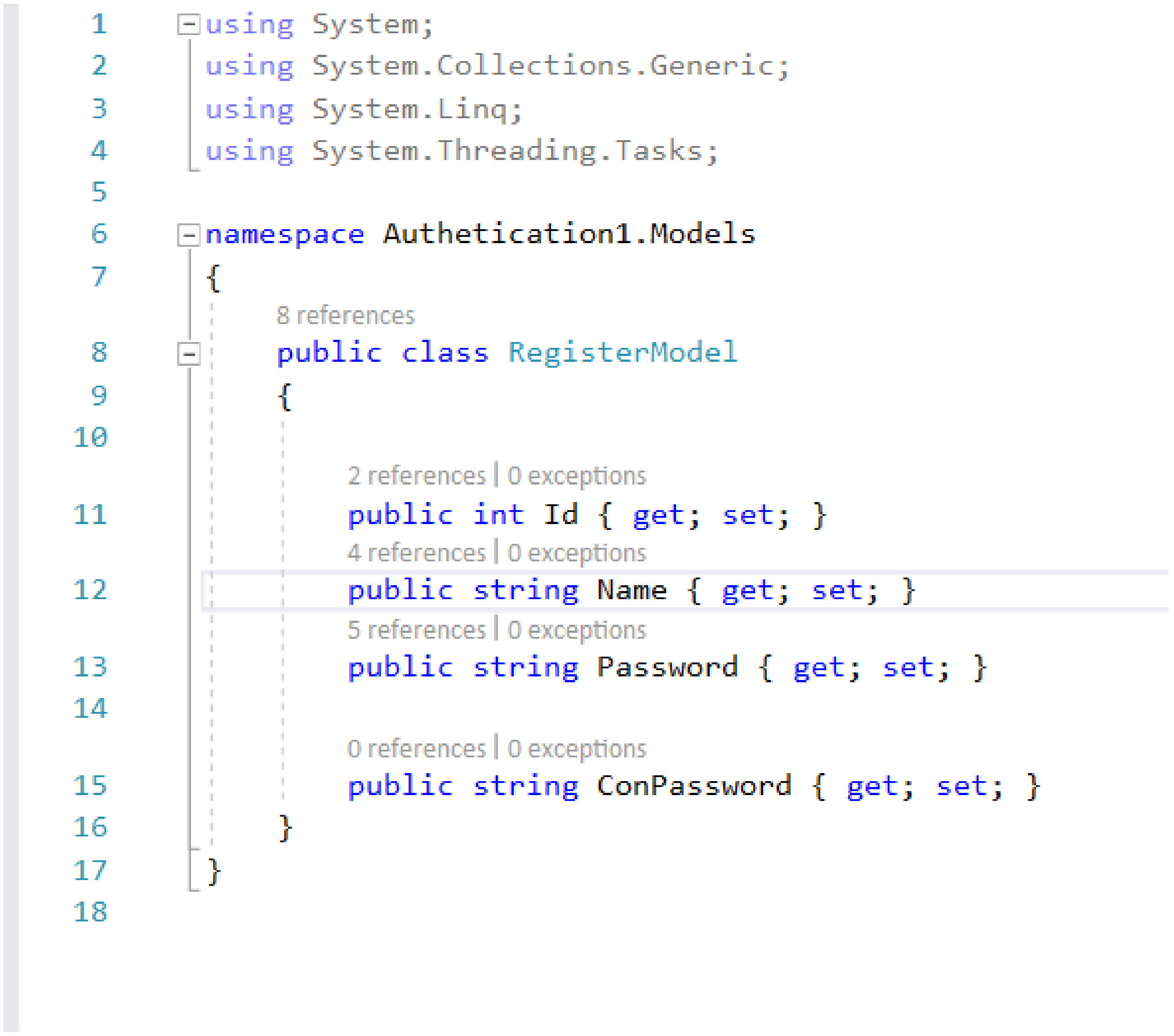
Password

Password

Reset Register

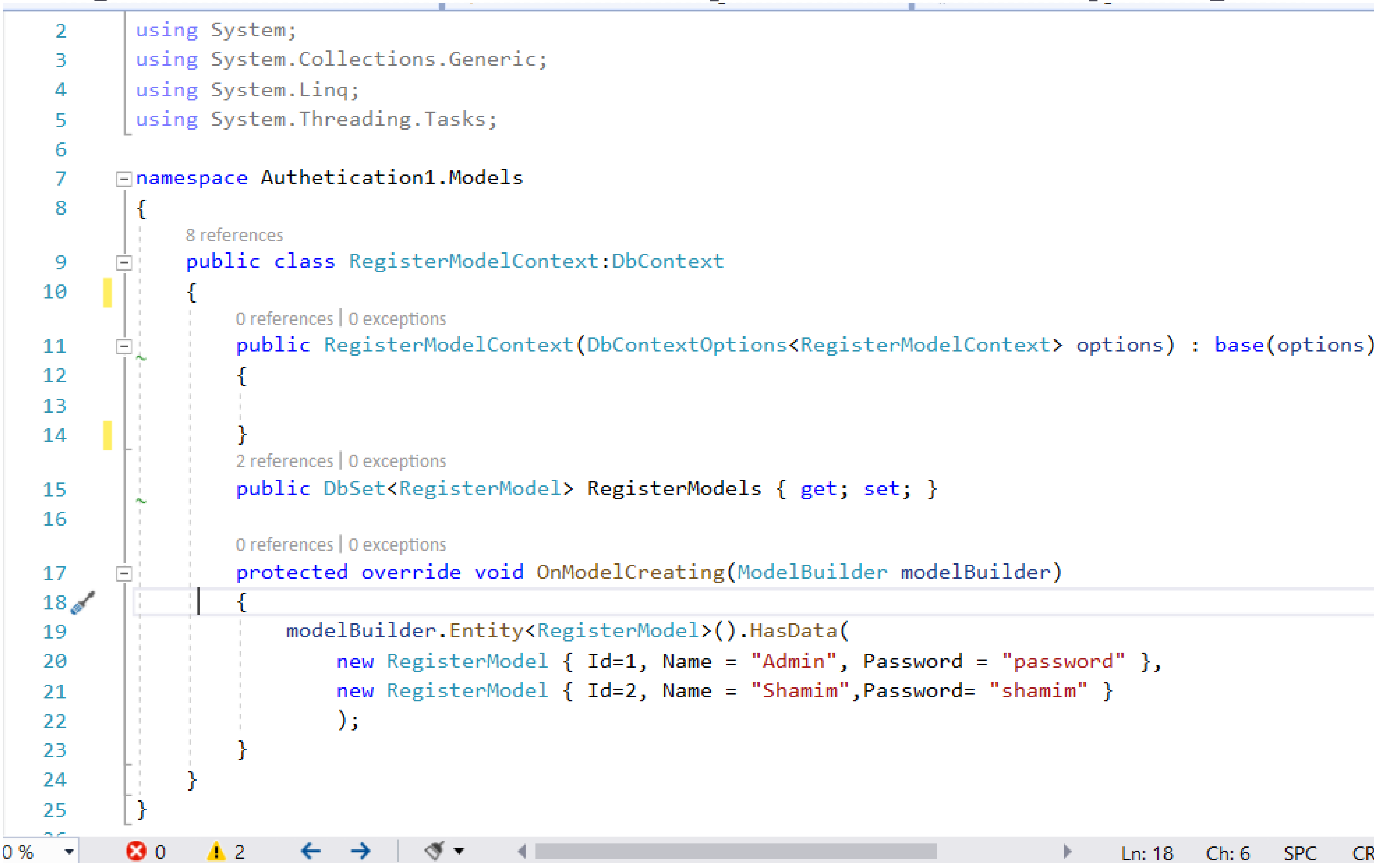
© 2020 - Authetication1

RegisterModel.cs [Step1]



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Autheticationl.Models
{
    8 references
    public class RegisterModel
    {

        2 references | 0 exceptions
        public int Id { get; set; }
        4 references | 0 exceptions
        public string Name { get; set; }
        5 references | 0 exceptions
        public string Password { get; set; }

        0 references | 0 exceptions
        public string ConPassword { get; set; }
    }
}

```

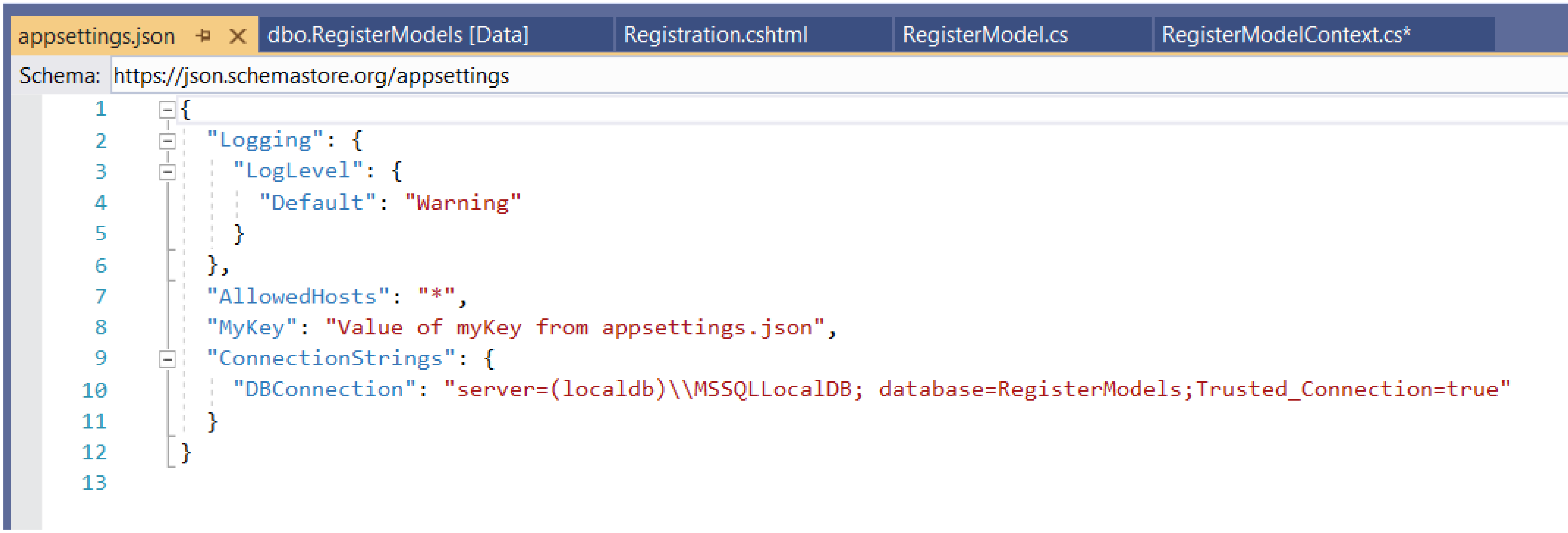
RegisterModelContext.cs [Context class] Step-2



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Authetication1.Models
{
    8 references
    public class RegisterModelContext:DbContext
    {
        0 references | 0 exceptions
        public RegisterModelContext(DbContextOptions<RegisterModelContext> options) : base(options)
        {

        }
        2 references | 0 exceptions
        public DbSet<RegisterModel> RegisterModels { get; set; }

        0 references | 0 exceptions
        protected override void OnModelCreating(ModelBuilder modelBuilder)
        {
            modelBuilder.Entity<RegisterModel>().HasData(
                new RegisterModel { Id=1, Name = "Admin", Password = "password" },
                new RegisterModel { Id=2, Name = "Shamim",Password= "shamim" }
                );
        }
    }
}
```

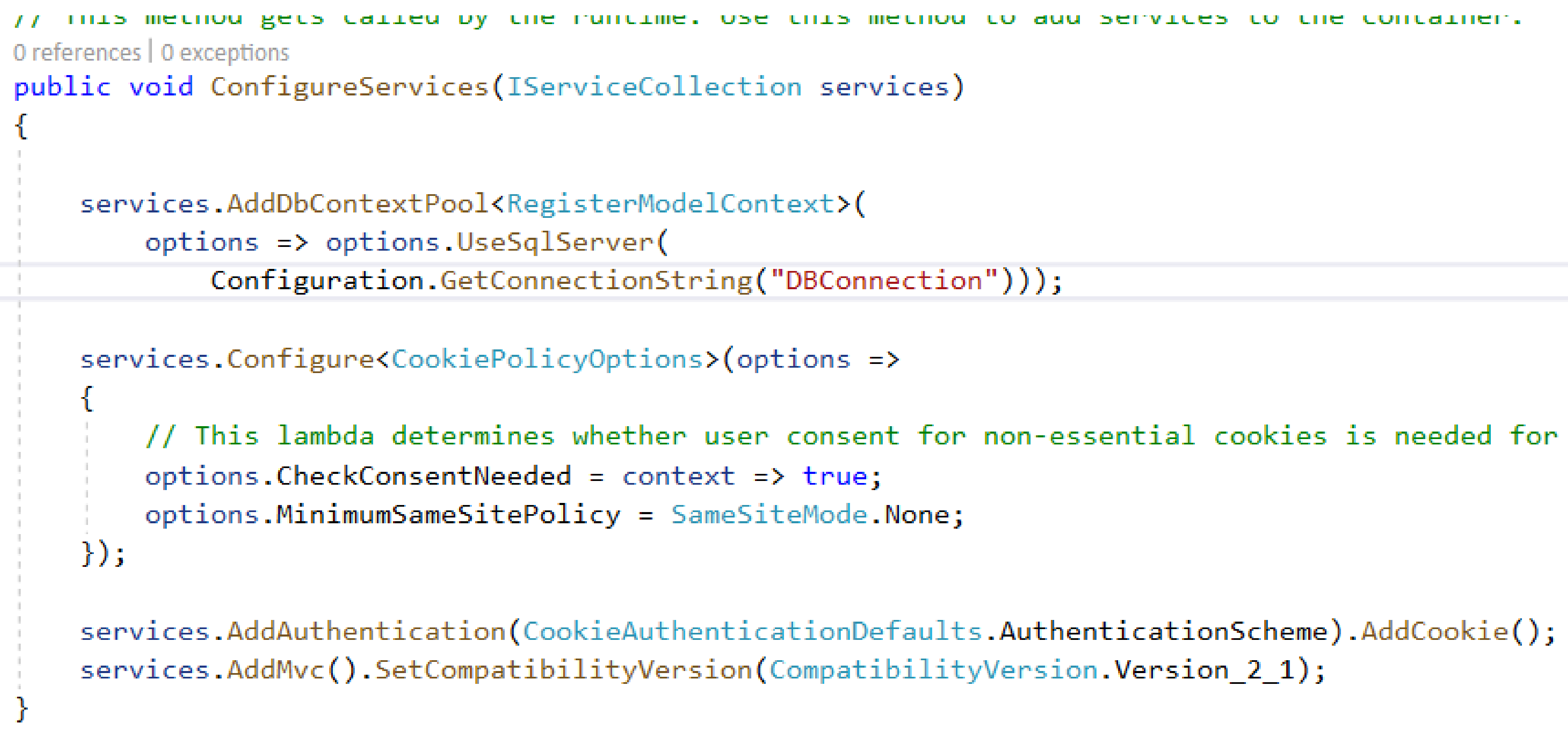
0 % ⊗ 0 ⚠ 2 Ln: 18 Ch: 6 SPC CR

Appsettings.json [Step-3]



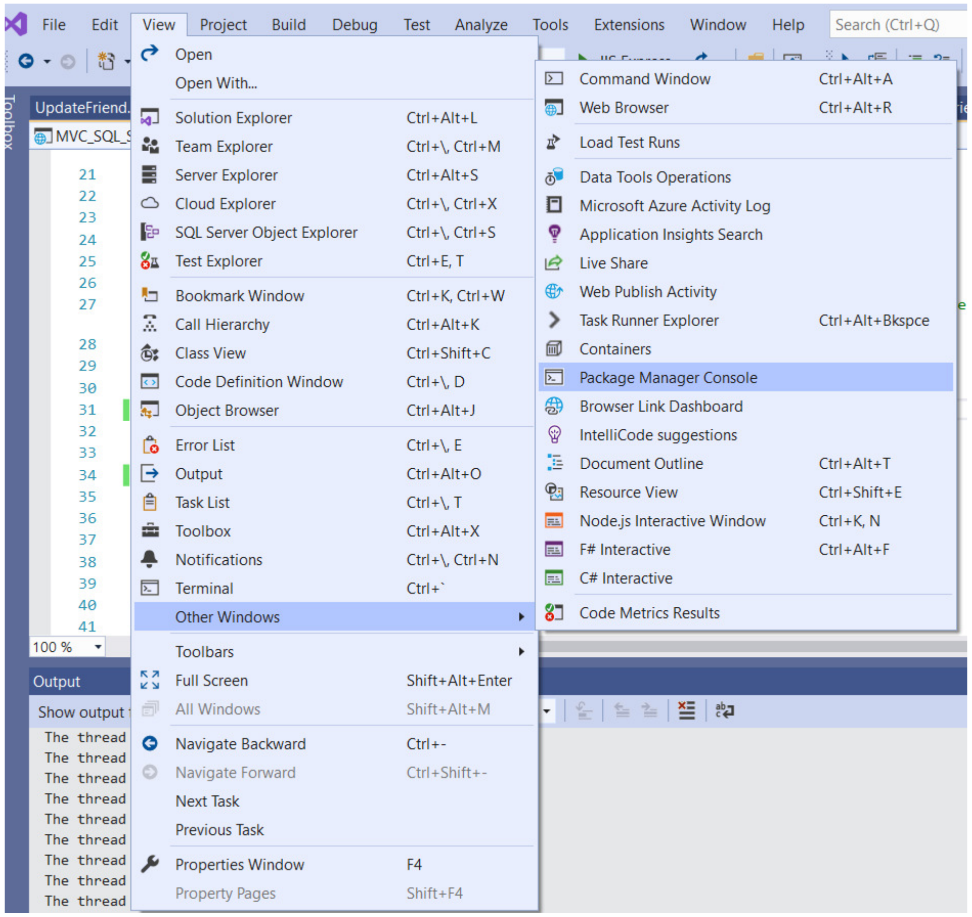
```json
{
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*",
  "MyKey": "Value of myKey from appsettings.json",
  "ConnectionStrings": {
    "DBConnection": "server=(localdb)\\MSSQLLocalDB; database=RegisterModels;Trusted_Connection=true"
  }
}
```

Configuration in the Startup class [Step-4]



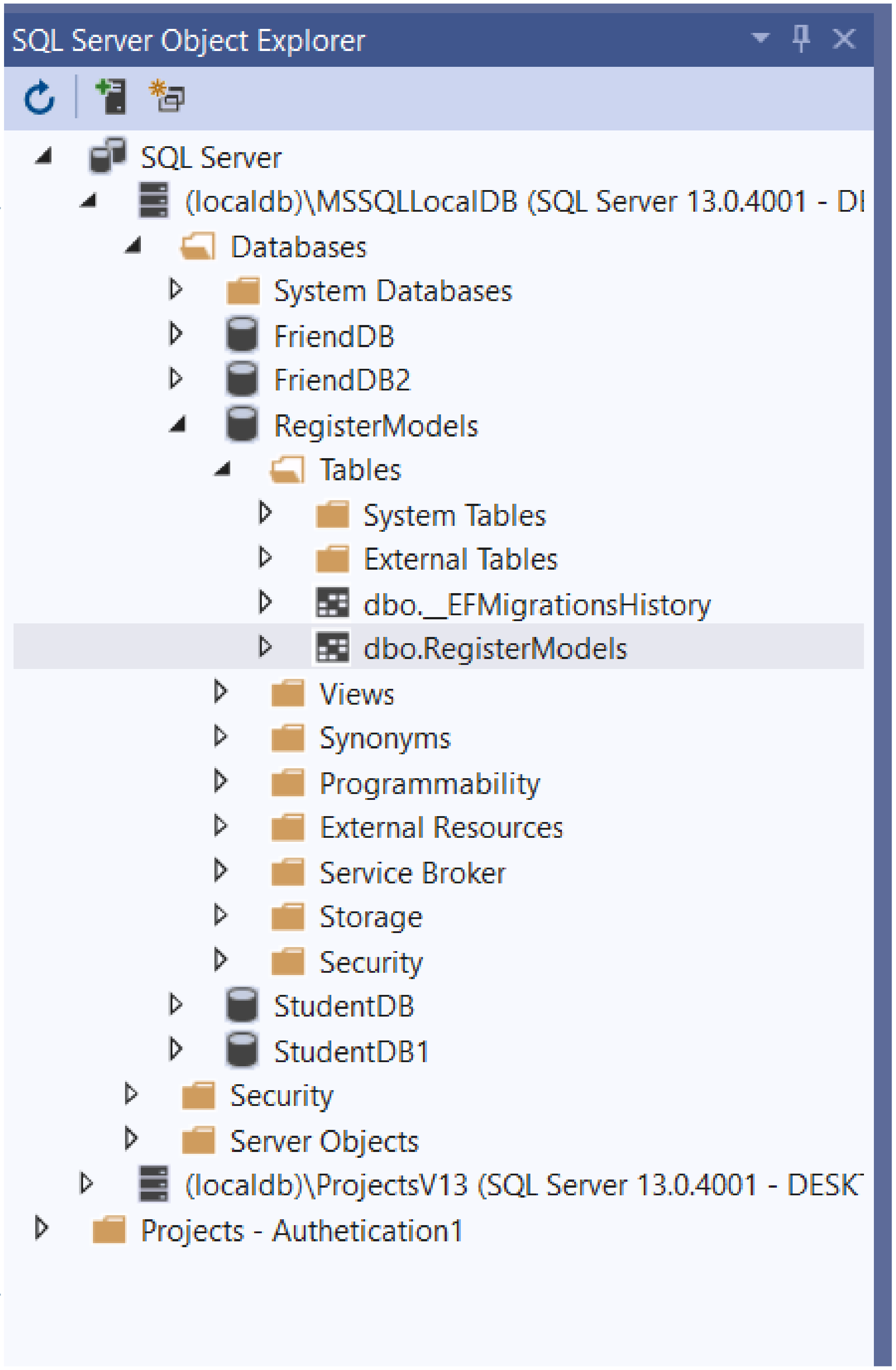
```
// This method gets called by the runtime. Use this method to add services to the container.
0 references | 0 exceptions
public void ConfigureServices(IServiceCollection services)
{

    services.AddDbContextPool<RegisterModelContext>(
        options => options.UseSqlServer(
            Configuration.GetConnectionString("DBConnection")));

    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies is needed for
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });

    services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme).AddCookie();
    services.AddMvc().SetCompatibilityVersion(CompatibilityVersion.Version_2_1);
}
```

Do Migration [Step-6]

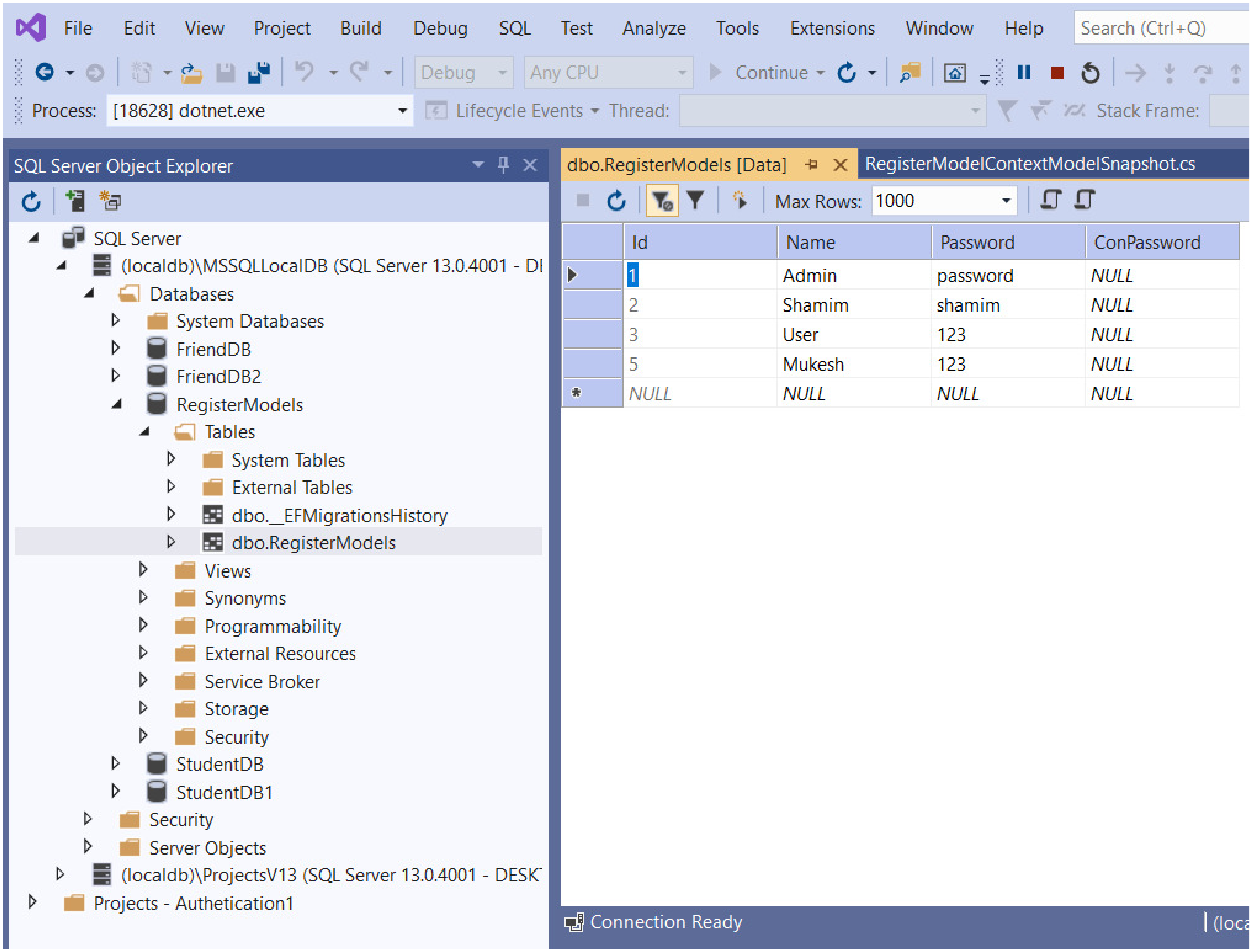


```
The thread
The thread
The thread
The thread
The thread
The thread
The thread
The thread
The thread
```

- **Add-Migration Initialize**
  - A folder will appear
  - Current and future migration will appear there.
- **Update-Database**


SQL Server Object Explorer

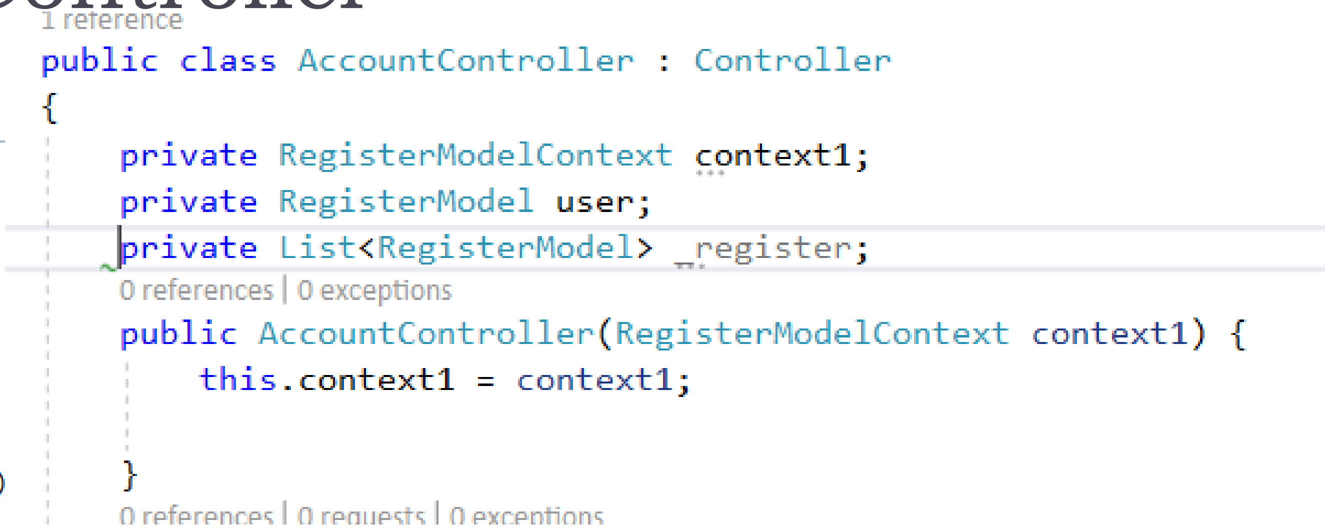
- SQL Server
  - (localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - DI
    - Databases
      - System Databases
      - FriendDB
      - FriendDB2
      - RegisterModels
        - Tables
          - System Tables
          - External Tables
          - dbo.__EFMigrationsHistory
          - dbo.RegisterModels
        - Views
        - Synonyms
        - Programmability
        - External Resources
        - Service Broker
        - Storage
        - Security
      - StudentDB
      - StudentDB1
    - Security
    - Server Objects
  - (localdb)\ProjectsV13 (SQL Server 13.0.4001 - DESK
- Projects - Authetication1



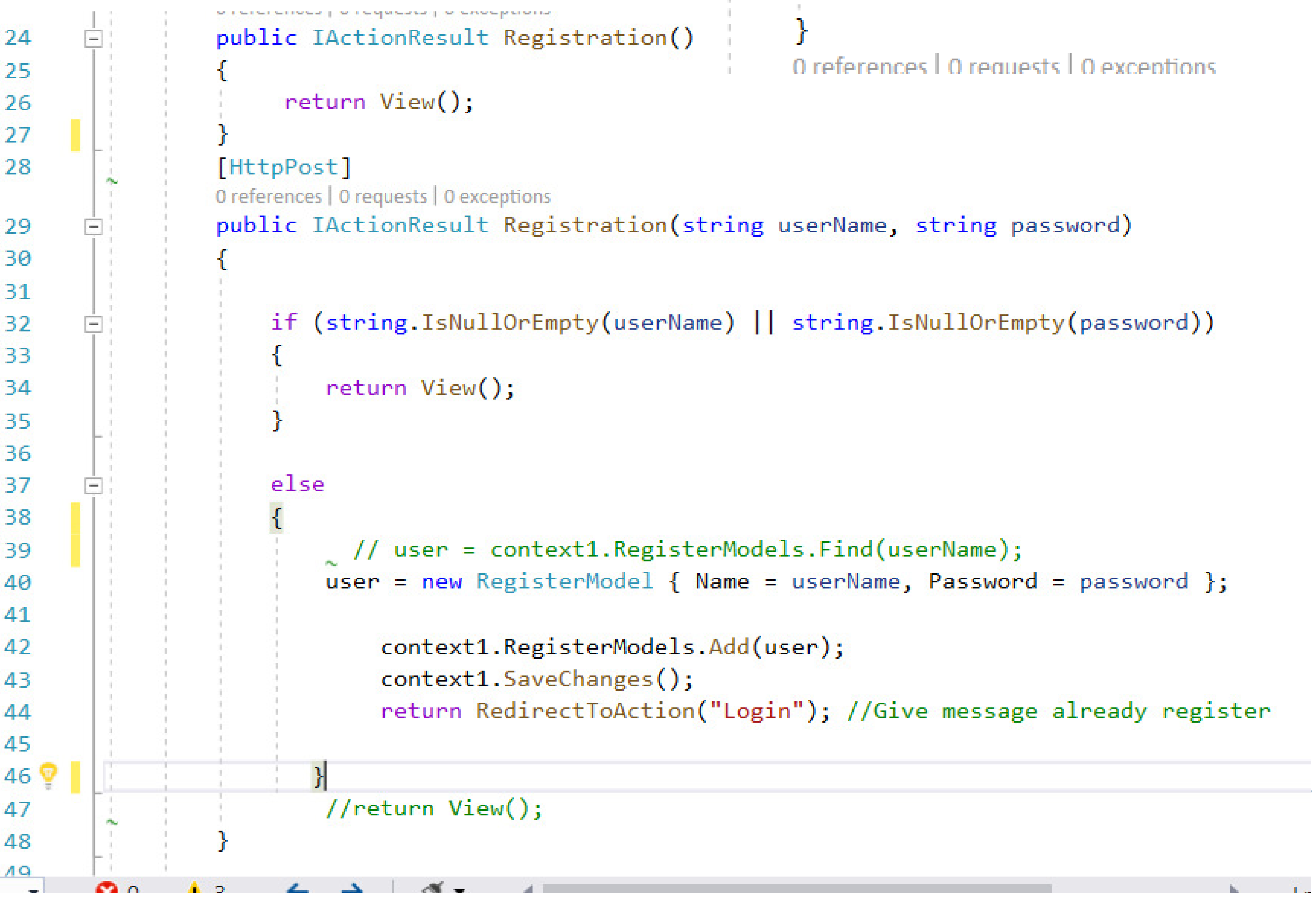
- SQL Server
  - (localdb)\MSSQLLocalDB (SQL Server 13.0.4001 - DI
    - Databases
      - System Databases
      - FriendDB
      - FriendDB2
      - RegisterModels
        - Tables
          - System Tables
          - External Tables
          - dbo.__EFMigrationsHistory
          - dbo.RegisterModels
        - Views
        - Synonyms
        - Programmability
        - External Resources
        - Service Broker
        - Storage
        - Security
      - StudentDB
      - StudentDB1
    - Security
    - Server Objects
  - (localdb)\ProjectsV13 (SQL Server 13.0.4001 - DESK
- Projects - Authetication1

Id	Name	Password	ConPassword
1	Admin	password	NULL
2	Shamim	shamim	NULL
3	User	123	NULL
5	Mukesh	123	NULL
NULL	NULL	NULL	NULL

AuthenticationController

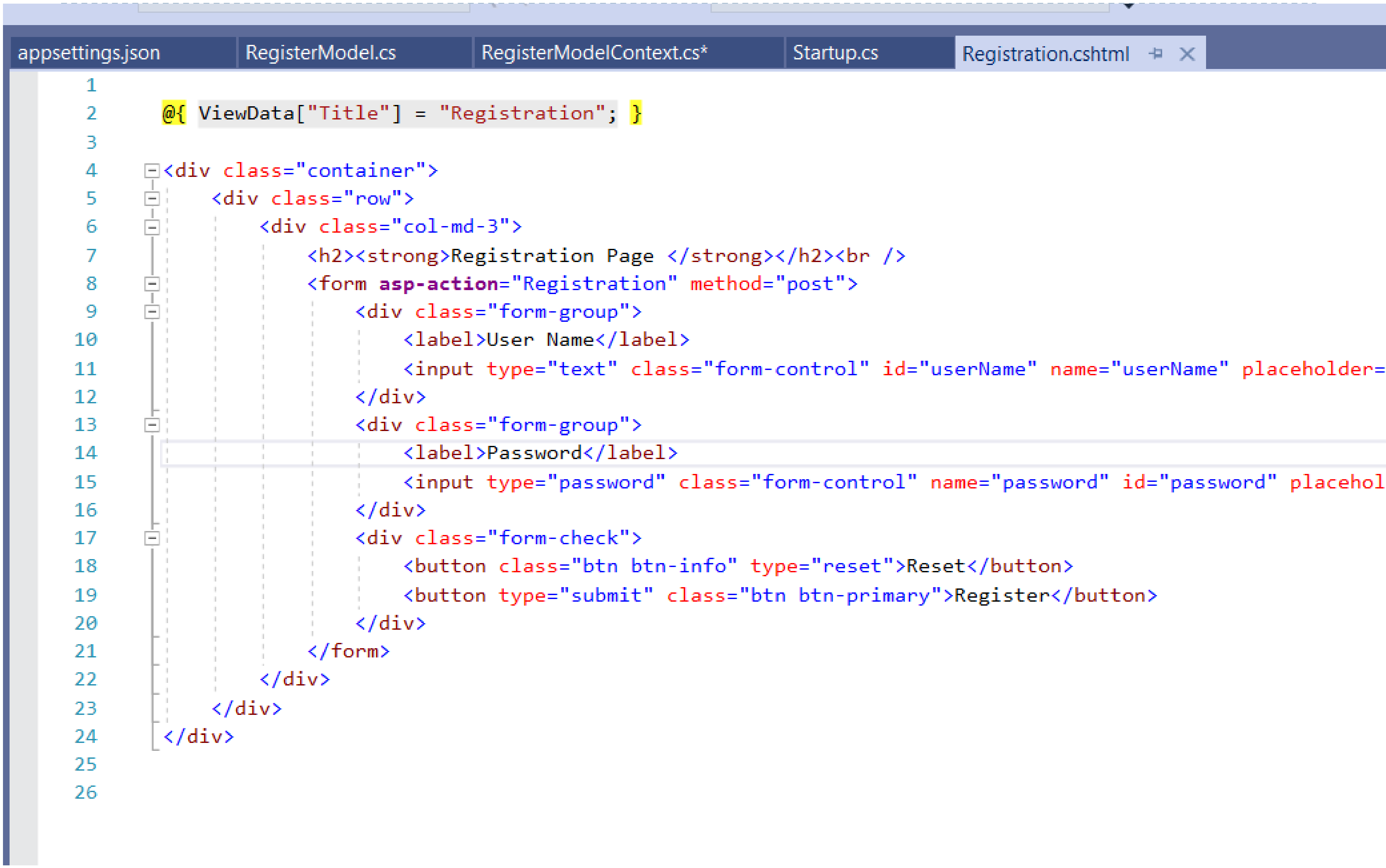


```
1 reference
public class AccountController : Controller
{
    private RegisterModelContext context1;
    private RegisterModel user;
    private List<RegisterModel> _register;
    0 references | 0 exceptions
    public AccountController(RegisterModelContext context1) {
        this.context1 = context1;

    }
    0 references | 0 requests | 0 exceptions
```

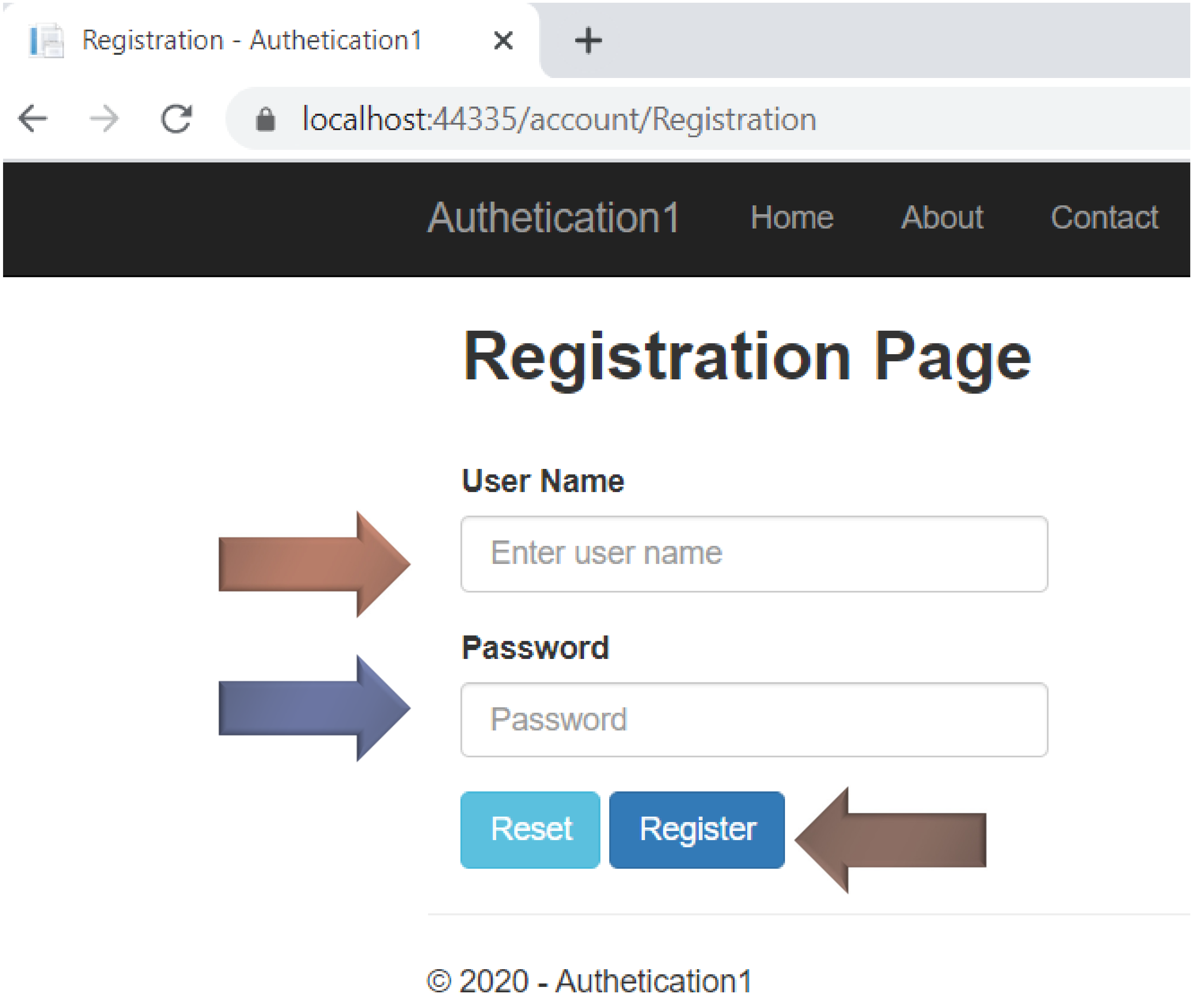


```
        0 references | 0 requests | 0 exceptions
        public IActionResult Registration()
        {
            return View();
        }
        [HttpPost]
        0 references | 0 requests | 0 exceptions
        public IActionResult Registration(string userName, string password)
        {

            if (string.IsNullOrEmpty(userName) || string.IsNullOrEmpty(password))
            {
                return View();
            }

            else
            {
                // user = context1.RegisterModels.Find(userName);
                user = new RegisterModel { Name = userName, Password = password };

                    context1.RegisterModels.Add(user);
                    context1.SaveChanges();
                    return RedirectToAction("Login"); //Give message already register

                }
                //return View();
        }
```

Registration View



```

@{ ViewData["Title"] = "Registration"; }

<div class="container">
    <div class="row">
        <div class="col-md-3">
            <h2><strong>Registration Page </strong></h2><br />
            <form asp-action="Registration" method="post">
                <div class="form-group">
                    <label>User Name</label>
                    <input type="text" class="form-control" id="userName" name="userName" placeholder=
                </div>
                <div class="form-group">
                    <label>Password</label>
                    <input type="password" class="form-control" name="password" id="password" placehol
                </div>
                <div class="form-check">
                    <button class="btn btn-info" type="reset">Reset</button>
                    <button type="submit" class="btn btn-primary">Register</button>
                </div>
            </form>
        </div>
    </div>
</div>


```


Registration - Authetication1
localhost:44335/account/Registration
Authetication1
Home
About
Contact

Registration Page

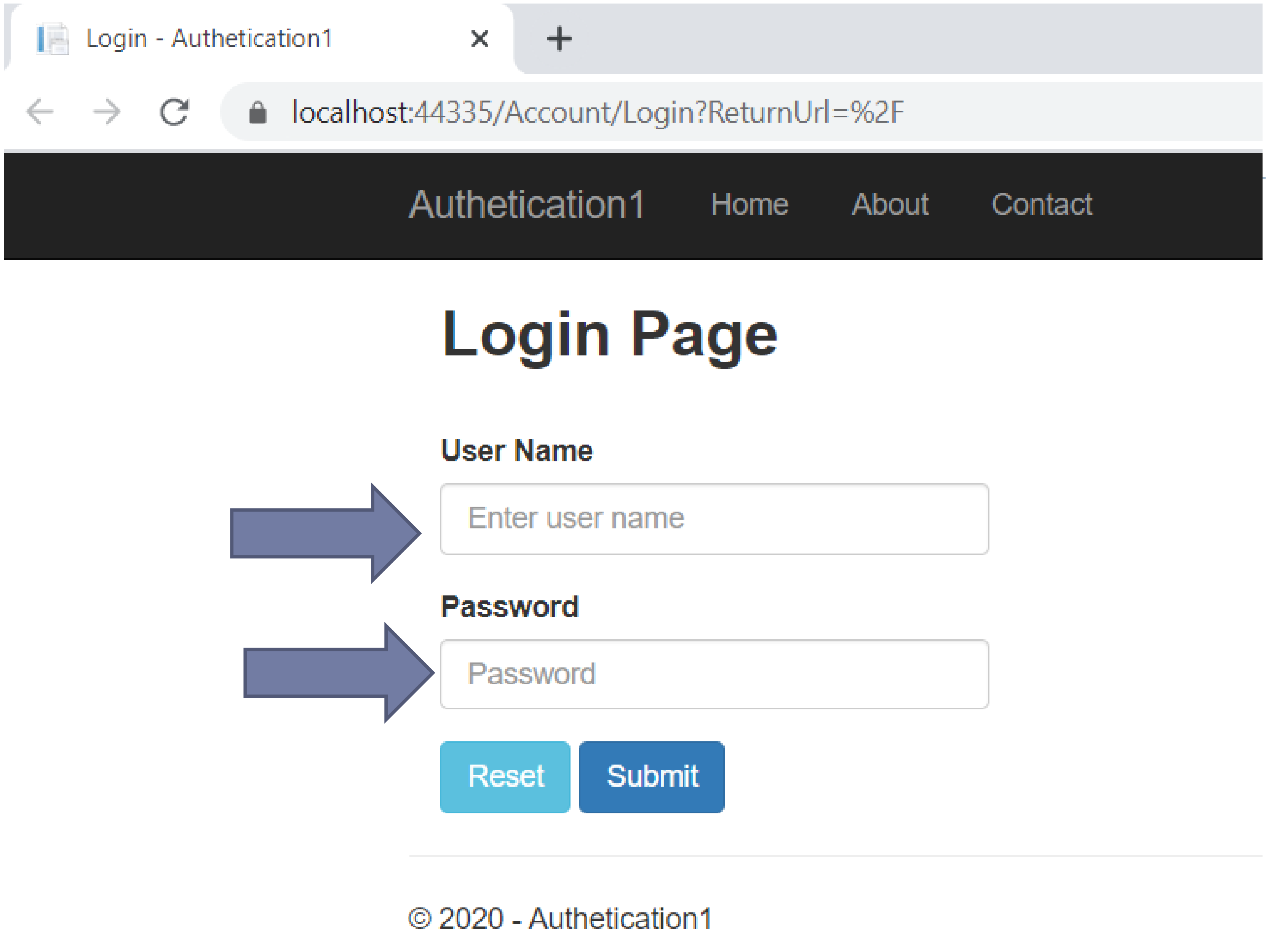
User Name

Enter user name

Password

Password

© 2020 - Authetication1



Login - Authetication1

localhost:44335/Account/Login?ReturnUrl=%2F

Authetication1 Home About Contact

Login Page

User Name

Enter user name

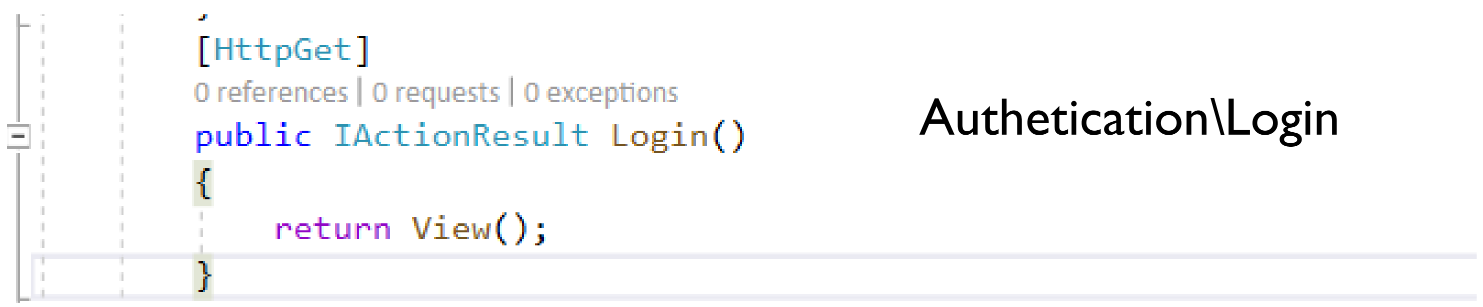
Password

Password

Reset Submit

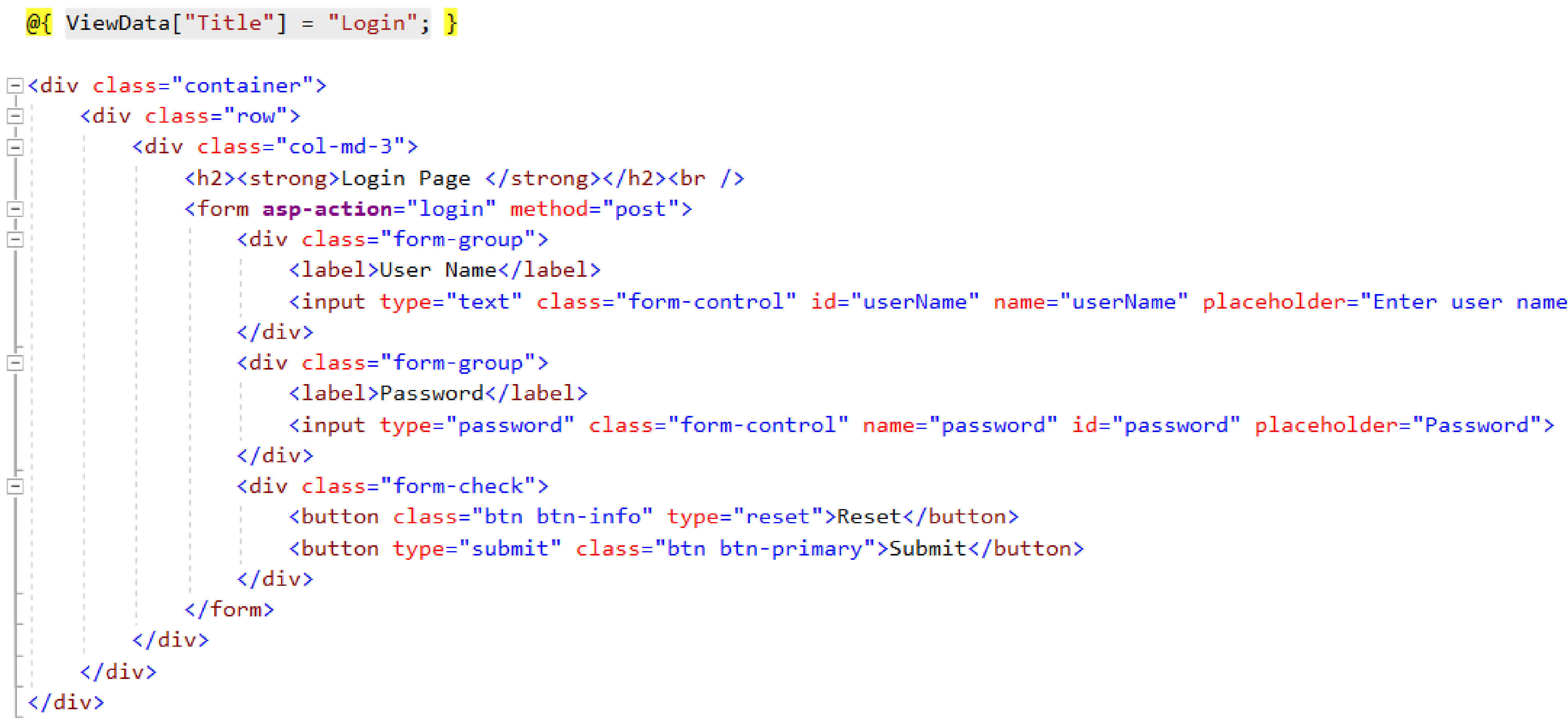
© 2020 - Authetication1

Login View



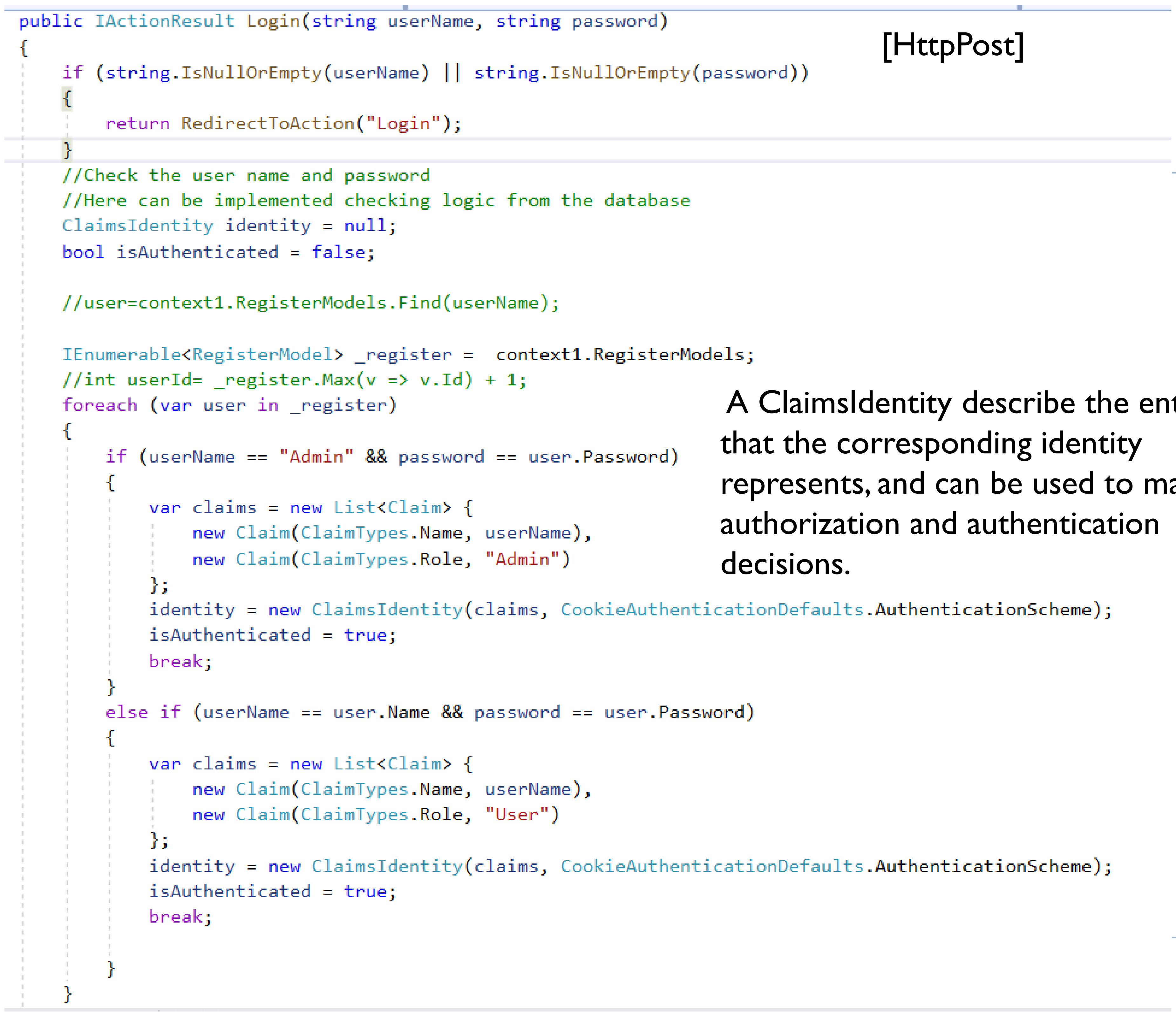
```
[HttpGet]
0 references | 0 requests | 0 exceptions
public IActionResult Login()
{
    return View();
}
```

Authetication\Login



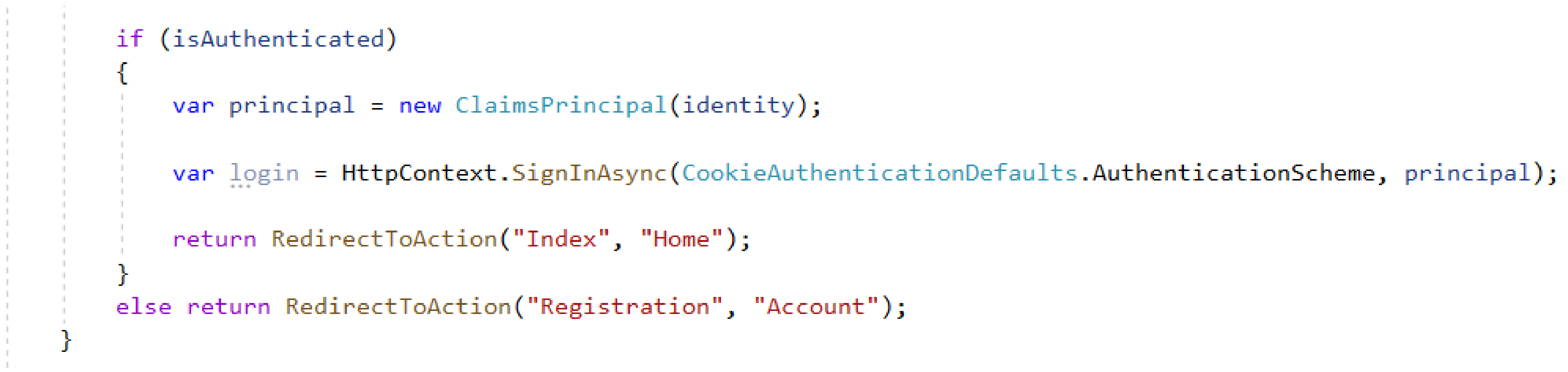
```
@{ ViewData["Title"] = "Login"; }

<div class="container">
    <div class="row">
        <div class="col-md-3">
            <h2><strong>Login Page </strong></h2><br />
            <form asp-action="login" method="post">
                <div class="form-group">
                    <label>User Name</label>
                    <input type="text" class="form-control" id="userName" name="userName" placeholder="Enter user name
                </div>
                <div class="form-group">
                    <label>Password</label>
                    <input type="password" class="form-control" name="password" id="password" placeholder="Password">
                </div>
                <div class="form-check">
                    <button class="btn btn-info" type="reset">Reset</button>
                    <button type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>
    </div>
</div>
```

[HttpPost]



```
public IActionResult Login(string userName, string password)
{
    if (string.IsNullOrEmpty(userName) || string.IsNullOrEmpty(password))
    {
        return RedirectToAction("Login");
    }
    //Check the user name and password
    //Here can be implemented checking logic from the database
    ClaimsIdentity identity = null;
    bool isAuthenticated = false;

    //user=context1.RegisterModels.Find(userName);

    IEnumerable<RegisterModel> _register =  context1.RegisterModels;
    //int userId= _register.Max(v => v.Id) + 1;
    foreach (var user in _register)
    {
        if (userName == "Admin" && password == user.Password)
        {
            var claims = new List<Claim> {
                new Claim(ClaimTypes.Name, userName),
                new Claim(ClaimTypes.Role, "Admin")
            };
            identity = new ClaimsIdentity(claims, CookieAuthenticationDefaults.AuthenticationScheme);
            isAuthenticated = true;
            break;
        }
        else if (userName == user.Name && password == user.Password)
        {
            var claims = new List<Claim> {
                new Claim(ClaimTypes.Name, userName),
                new Claim(ClaimTypes.Role, "User")
            };
            identity = new ClaimsIdentity(claims, CookieAuthenticationDefaults.AuthenticationScheme);
            isAuthenticated = true;
            break;

        }
    }
```

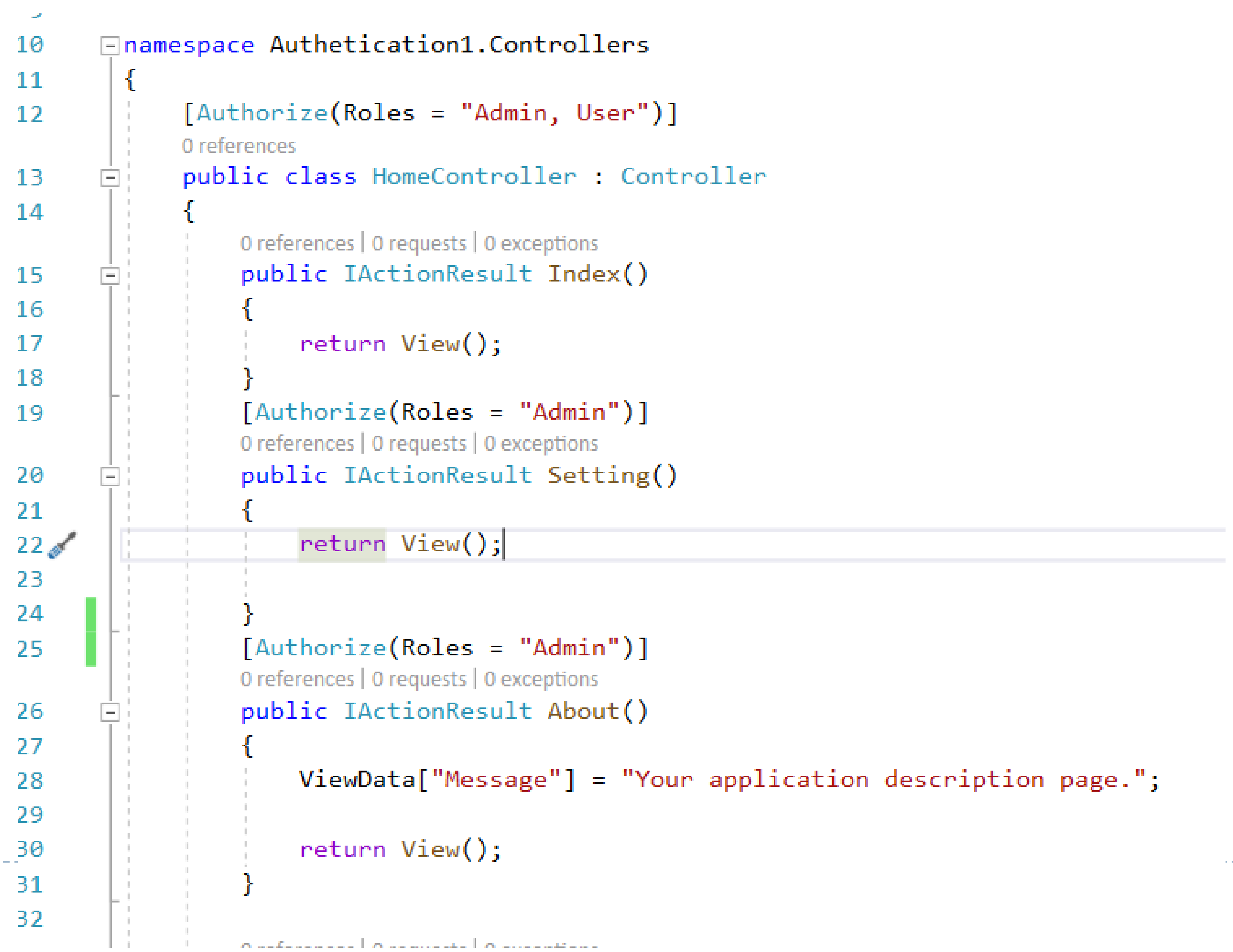
A ClaimsIdentity describe the entity that the corresponding identity represents, and can be used to make authorization and authentication decisions.



```
    if (isAuthenticated)
    {
        var principal = new ClaimsPrincipal(identity);

        var login = HttpContext.SignInAsync(CookieAuthenticationDefaults.AuthenticationScheme, principal);

        return RedirectToAction("Index", "Home");
    }
    else return RedirectToAction("Registration", "Account");
}
```

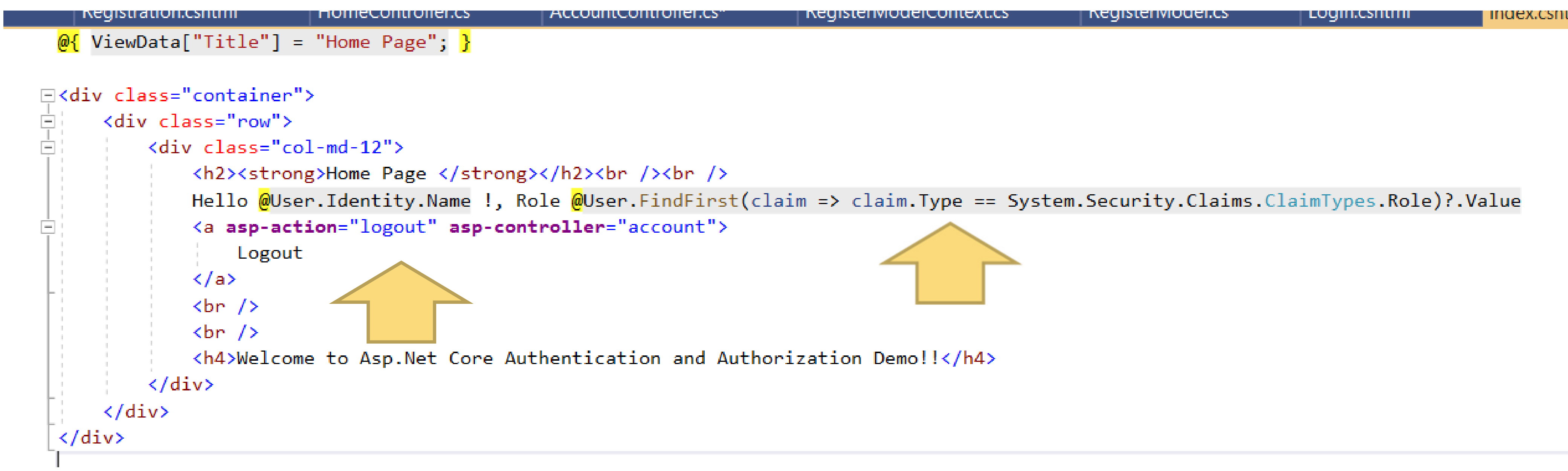
ClaimsPrincipal exposes a collection of identities, each of which is a ClaimsIdentity. In the common case, this collection, which is accessed through the Identities property, will only have a single element.

Home Controller



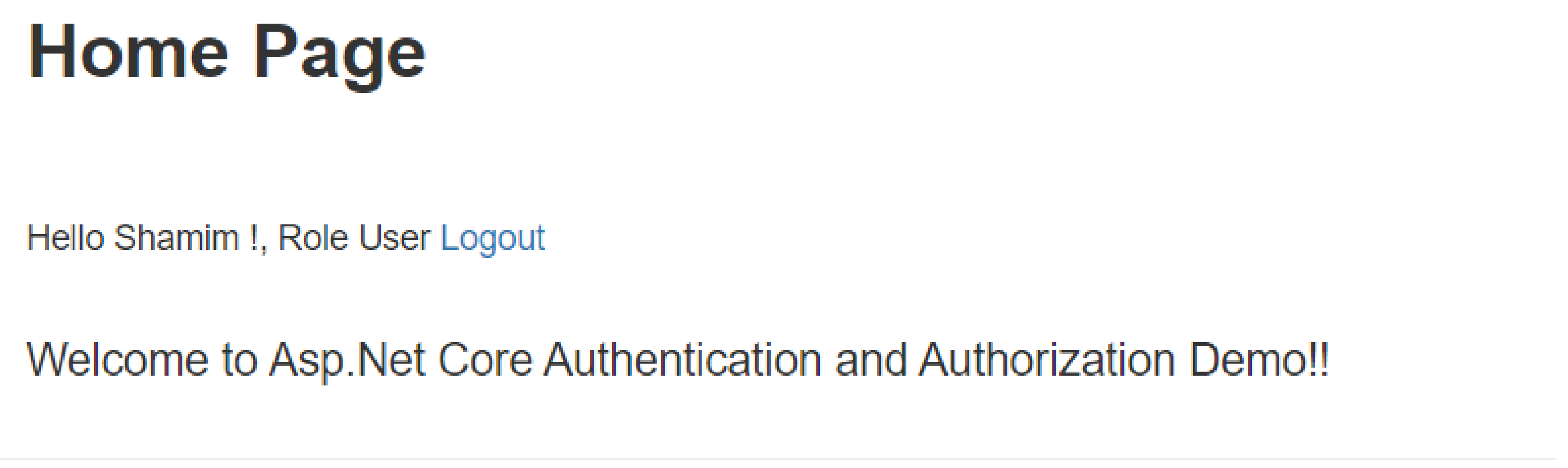
```
namespace Authetication1.Controllers
{
    [Authorize(Roles = "Admin, User")]
    0 references
    public class HomeController : Controller
    {
        0 references | 0 requests | 0 exceptions
        public IActionResult Index()
        {
            return View();
        }
        [Authorize(Roles = "Admin")]
        0 references | 0 requests | 0 exceptions
        public IActionResult Setting()
        {
            return View();

        }
        [Authorize(Roles = "Admin")]
        0 references | 0 requests | 0 exceptions
        public IActionResult About()
        {
            ViewData["Message"] = "Your application description page.";

            return View();
        }

```

Index View



Registration.cshtml HomeController.cs AccountController.cs* RegisterModelContext.cs RegisterModel.cs Login.cshtml Index.csht

```
@{ ViewData["Title"] = "Home Page"; }

<div class="container">
    <div class="row">
        <div class="col-md-12">
            <h2><strong>Home Page </strong></h2><br /><br />
            Hello @User.Identity.Name !, Role @User.FindFirst(claim => claim.Type == System.Security.Claims.ClaimTypes.Role)?.Value
            <a asp-action="logout" asp-controller="account">
                Logout
            </a>
            <br />
            <br />
            <h4>Welcome to Asp.Net Core Authentication and Authorization Demo!!</h4>
        </div>
    </div>
</div>
```

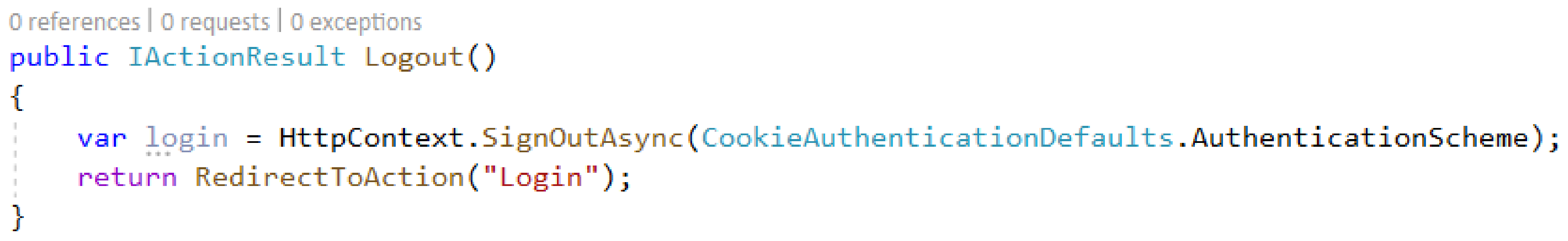


Home Page

Hello Shamim !, Role User Logout

Welcome to Asp.Net Core Authentication and Authorization Demo!!

Logout



```
0 references | 0 requests | 0 exceptions
public IActionResult Logout()
{
    var login = HttpContext.SignOutAsync(CookieAuthenticationDefaults.AuthenticationScheme);
    return RedirectToAction("Login");
}
```