



**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII  
AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software i Automatică**

## **Raport**

### **Lucrarea de laborator №5**

**Disciplina: Tehnologii Web.**

**Tema: Implementarea restricțiilor privind accesul  
utilizatorilor la anumite pagini ale site-ului (admin,  
utilizator).**

**Realizat:**

**Bunescu Gabriel, TI-207**

**Profesor:**

**asis. univ. Rusu Cristian.**

## Sarcina lucrari:

Implementarea restrictionarii accesului utilizatorului la anumite pagini ale site-ului(admin,user).

## Elaborarea lucrării:

Pentru a implementa mecanismul de restricționare a anumitor grupuri de utilizatori la anumite acțiuni disponibile pe site, este necesar să se utilizeze mecanisme de atribute, și anume atributul `ActionFilterAttribute`.

Atributul `Admin Mod` va permite să restricționăm accesul nu la toate acțiunile controlerului și numai la anumite acțiuni ale controlerului. Pentru a utiliza atributul `Admin Mod`, trebuie să adăugați o adnotare înaintea numelui metodei sau clasei. În același timp, deoarece metodele `OnActionExecuting` ale acestor atribute trebuie să fie executate secvențial, mai întâi de către `AdminMod`.

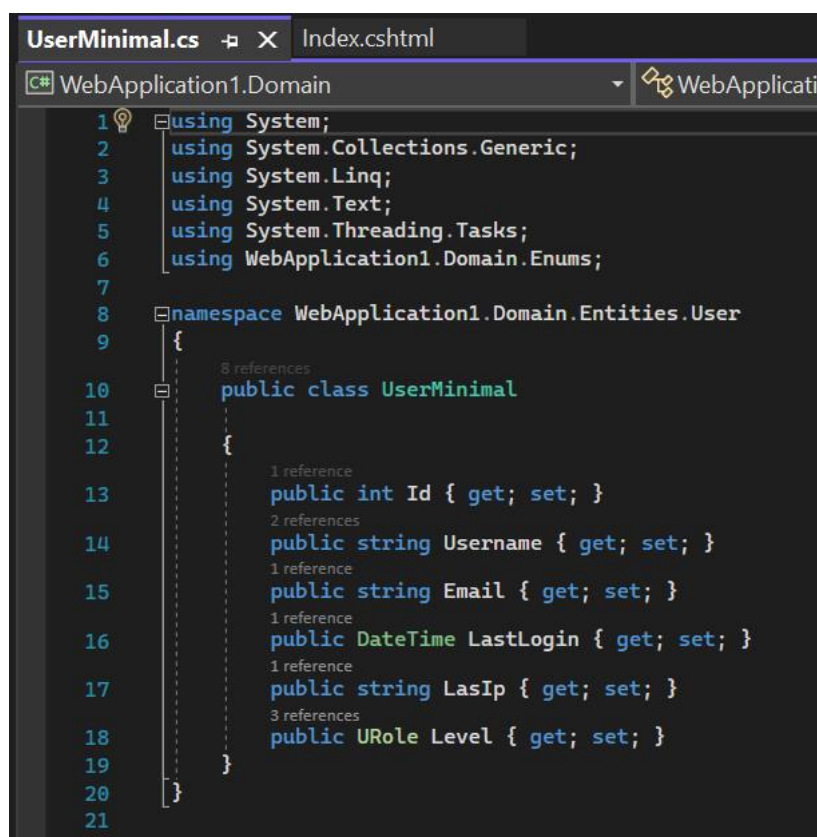


Figura 1 – Clasa `UserMinimal`

## Crearea clasei `AdminAttributes`:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;
using WebApplication1.BussinesLogic.Interfaces;
using WebApplication1.Domain.Enums;
using WebApplication1.Web.Controllers;

namespace WebApplication1.Attributes
{
    public class AdminMode : ActionFilterAttribute
```

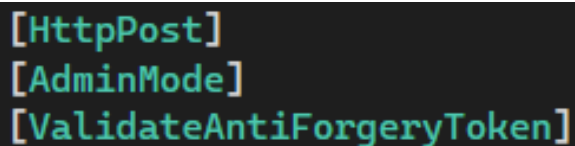
```

{
    private readonly ISession _sessionBusinessLogic;

    public AdminMode()
    {
        var businessLogic = new BussinesLogic.BussinesLogic();
        _sessionBusinessLogic = businessLogic.GetSessionBL();
    }

    public override void OnActionExecuting(ActionExecutingContext filterContext)
    {
        var apiCookie = HttpContext.Current.Request.Cookies["X-KEY"];
        if (apiCookie != null)
        {
            var profile = _sessionBusinessLogic.GetUserByCookie(apiCookie.Value);
            if (profile != null && profile.Level == URole.Admin)
            {
                HttpContext.Current.SetMySessionObject(profile);
            }
            else
            {
                filterContext.Result = new System.Web.Mvc.RedirectToRouteResult(new
                    RouteValueDictionary(new { controller = "Home", action = "Error" }));
            }
        }
        else
        {
            filterContext.Result = new System.Web.Mvc.RedirectToRouteResult(new
                RouteValueDictionary(new { controller = "Home", action = "Error" }));
        }
    }
}

```



```

[HttpPost]
[AdminMode]
[ValidateAntiForgeryToken]

```

Figura 2 – Utilizarea adnotarii AdminMode

## Concluzie:

În urma acestui laborator am studiat posibilitatea de restrictionare a accesului la anumite pagini pentru utilizatori cu ajutorul atributelor utilizate pe platforma ASP.NET.