Authentication 3 – Roles

Building on the previous Authentication Labs we will be adding Authorization as the final piece of our system. This may be easier or harder depending on whether you added the Role bits mentioned in previous assignments.

1. Make sure you have a *Role* property in your Registration Form/Database.
2. Create two Users, One in an ***Admin*** role, another in a ***User*** Role.
3. Add a Role to the Authorize Tag in Privacy.cshtml.cs:

   ```
   [Authorize(Roles = "Admin")]
   ```

   Authorize tags are known as **Declarative Security**.

4. Try Accessing the Privacy page as an ***Admin*** and you should be permitted access. Try again as a ***User*** role and you should get a ***Forbidden*** redirect.
5. Edit your Index.cshtml page to test **Imperative Security** (applying Security through code).

   Test the functionality while not logged in, logged in as a User, finally as an Admin.

```
@if (User.Identity.IsAuthenticated)
    {
    <div class="alert alert-secondary">
        <div>This information is accessible to Employees only </div>
        <div>Hello, @User.Identity.Name, the time is @DateTime.Now</div>
        @if (User.IsInRole("Admin"))
            {
            <button type="button" class="btn btn-danger">Launch Missiles</button>
            }
    </div>
    }
else
    {
    <div class="alert alert-primary">General Public information</div>
    }
```

6. Lockdown the Admin Folder by adding a Policy to ConfigureServices() in Startup.cs:

```
            cookieOptions.Cookie.HttpOnly = true;
        });

        services.AddAuthorization(options =>
        {
            options.AddPolicy("RequireAdmin", policy =>
                   policy.RequireRole("Admin"));
        });

        services.AddRazorPages(options =>
        {
            options.Conventions.AuthorizeFolder("/Common/Admin", "RequireAdmin");
        });
        }
```

   You shouldn't Lockdown the Admin Folder until you have created at Least 1 Admin user.

7. You can also perform declarative security using policies:

   ```
   [Authorize(Policy = "RequireAdmin")]
   ```

   Policies are really powerful, this is just a taste.

8. To make the system useable you should add functionality to Maintain Users (Edit, Delete, Lockout …) this is left as an exercise for the student.
9. Submit whatever screen shots are needed to demonstrate your system.