Lab: Controlling Access to ASP.NET MVC Core Web Applications

# Scenario

A large part of the functionality for your proposed Photo Sharing application is in place. However, stakeholders are concerned about security because there are no restrictions on the tasks that users can complete. The following restrictions are required:

* Only site members should be able to add photos.
* A photo should only be deleted by its owner.
* Only site members should be able to add comments.
* Views should only show links to actions that the user is authorized to perform.

You have been asked to resolve these concerns by creating a policies for the Photo Sharing application. Visitors should be able to register as users of the web application and create user accounts for themselves. After registration, when the users log on to the application, they will have access to actions such as adding and deleting photos. Anonymous users will not have access to perform these actions. Additionally, registered users should also be able to reset their own password.

# Objectives

After completing this lab, you will be able to:

* Provide access to resources in a web application.

Estimated Time: 45 minutes

### Exercise 1: Add and Test Identity

You decide to scaffold Microsoft Identity into your project, using a ASP .NET Core 2.1 feature.

**Results**: At the end of this exercise, you will create a Photo Sharing application in which users can register for an account, log on, and log off.

# Exercise 2: Authorizing Access to Resources

## Scenario

You should ensure that:

* Only site members can add photos.
* A photo can only be deleted by its owner.
* Only site members can add comments.
* Views only show links to actions that the user is authorized to perform.

The main tasks for this exercise are as follows:

1. Restrict Access to the Create actions
2. Create the Authorization Handlers.
3. Add the Authorization Policies.
4. Restrict access to Photo actions.
5. Restrict access to the Comment actions.
6. Check authorization in the views.
7. Test authorization.

**Results**: At the end of this exercise, you will have authorized authenticated users to access resources in your web application.

**Question**: When you tried to add a photo before logging on to the application, why did ASP.NET display the Login view?