Lab: Exploring ASP.NET MVC Core

# Scenario

You are working as a junior developer at Adventure Works. You have been asked by a senior developer to investigate the possibility of creating a web-based photo sharing application for your organization’s customers, similar to one that the senior developer has seen on the Internet. Such an application will promote a community of cyclists who use Adventure Works equipment, and the community members will be able to share their experiences. This initiative is intended to increase the popularity of Adventure Works Cycles, and thereby to increase sales. You have been asked to begin the planning of the application by examining an existing photo sharing application and evaluating its functionality. You have also been asked to examine programming models available to ASP.NET developers. To do this, you need to create basic web applications written with three different models: Web Pages, Web Forms, and MVC. Your manager has asked you to report on the following specific questions for each programming model:

* How does the developer set a connection string and data provider?
* How does the developer impose a consistent layout, with Adventure Works branding and menus, on all pages in the web application?
* How does the developer set a cascading style sheet with a consistent set of color, fonts, borders, and other styles?
* How does the developer add a new page to the application and apply the layout and styles to it?

Objectives

After completing this lab, you will be able to:

* Describe the MVC programming model available in ASP.NET Core.
* Describe the structure of a web application developed in the MVC programming model.

Lab Setup

Estimated Time: 45 minutes

# Exercise 1: Exploring a Photo Sharing Application

## Scenario

In this exercise, you will begin by examining the photo sharing application.

The main tasks for this exercise are as follows:

1. Register a user account.
2. Upload and explore photos.
3. Use slideshows.
4. Test the authorization.

## Task 1: Register a user account.

1. On the Start screen, open Visual Studio.
2. Navigate to the following location to open the PhotoSharingSample.sln file:

* Allfiles (C):\Labfiles\Mod01\PhotoSharingSample

1. Run the web application in non-debugging mode.
2. Create a new user account with the following credentials:

* User name: <A user name of your choice>
* Password: <A password of your choice>

## Task 2: Upload and explore photos.

1. Add the following comment to the Orchard image:

* Subject: Just a Test Comment
* Comment: This is a Sample

1. Add a new photo to the application by using the following information:

* Title of the photo: Fall Fungi
* Navigation path to upload the photo: Allfiles (C):\ Labfiles\Mod01\Pictures\fungi.jpg
* Description: Sample Text

1. Verify the description details of the newly added photo.

## Task 3: Use slideshows.

1. Use the Slideshow feature.
2. Add the following images to your list of favorite photos:

* Fall Fungi
* Orchard
* Flower

1. View the slideshow of the images selected as favorites.

## Task 4: Test the authorization.

1. Log off from the application, and then attempt to add a comment for the Fall Fungi image.
2. Attempt to add a new photo to the Photo Index page.
3. Close the Internet Explorer window and the Visual Studio application.

**Results**: At the end of this exercise, you will be able to understand the functionality of a photo sharing application, and implement the required application structure in the Adventure Works photo sharing application.

# Exercise 2: Exploring an MVC Application

## Scenario

In this exercise, you will create a simple MVC application and explore its structure.

The main tasks for this exercise are as follows:

1. Create an MVC Core application.
2. Explore the application structure.
3. Add simple functionality.
4. Apply the site layout.

## Task 1: Create an MVC Core application.

1. Start Visual Studio and create a new MVC project by using the ASP.NET Core Web Application (.NET Core) template. Choose the MVC template with No Authentication.
2. Run the new MVC application in a browser and explore the Contact page.
3. Stop debugging by closing the browser.

## Task 2: Explore the application structure.

1. Open the appsettings.json file and verify whether the presence of a database connection string.
2. Verify that the ~/Views/Shared/\_Layout.cshtml file contains a common layout for all pages on the website, and how pages link to the layout.
3. Verify that the Site.css file is used to apply styles to all pages on the website, and note how the pages link to the style sheet.

## Task 3: Add simple functionality.

1. Add a new view to the application by using the following information:

* Parent folder: /Views/Home
* Name of the view: TestPage.cshtml
* Clear the Use a layout page check box.

1. Add an H1 element to the TestPage.cshtml view by using the following information:

* Content: This is a Test Page

1. Add an action to the HomeController.cs file by using the following information:

* Procedure name: TestPage
* Return type: IActionResult
* Procedure parameters: None
* Return the view "TestPage"

1. Add a link to the Index.cshtml page by using the following information:

* Start tag: <a>
* Attribute: href=” ~/Home/TestPage”
* Content: Test Page
* End tag: </a>

1. Save all the changes.
2. Run the website and view the page you added.
3. Stop debugging by closing the browser.

## Task 4: Apply the site layout.

1. Open the TestPage.cshtml file and remove the code that sets the Layout = null.
2. In the TestPage.cshtml file, remove all the tags except the <h1> tag and its contents.
3. Save all the changes.
4. Run the web application and browse to Test Page.
5. Close all the open applications.

**Results**: At the end of this exercise, you will be able to build a simple MVC application in Visual Studio.