

**Version C MVC 5 uses PhotoSharingApp – V D uses various apps to show functionality****Mod 1 – watch out for ver c vs. d**

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD01\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD01_LAB_MANUAL.md)

Razor Pages, Web API, and MVC

**Explore PhotoSharingApp** – what does it do and how? What messages if any log to the console F12?

Framework

[https://github.com/MicrosoftLearning/20486C-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD01\\_LAK.md](https://github.com/MicrosoftLearning/20486C-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD01_LAK.md)

Core

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD01\\_DEMO.md](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD01_DEMO.md)

**Mod 2; Designing ASP.NET Core MVC Web Applications**

Planning module parts with Visio

Mod 3 Demo – **Middleware – more like 487**

Dependency Injection pattern to achieve loose coupled classes. IoC Container (a.k.a. DI Container) is a framework for implementing automatic dependency injection. It manages object creation and it's life-time, and also injects dependencies to the class.

The IoC container creates an object of the specified class and also injects all the dependency objects through a constructor, a property or a method at run time and disposes it at the appropriate time. This is done so that we don't have to create and manage objects manually.

<https://www.tutorialsteacher.com/ioc/ioc-container>

UseStaticFiles: [https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD03\\_DEMO.md#demonstration-how-to-work-with-static-files](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD03_DEMO.md#demonstration-how-to-work-with-static-files)

Common types of static files:

HTML files

CSS stylesheets

JavaScript files

Images and other assets

JSON or XML files

---

1. How to add HTML and image files as static files in the wwwroot folder
  2. How to serve static files by using the UseStaticFiles middleware
  3. What is the result of attempting to access a file that does not exist
- 

## Module 4: Developing ASP.NET MVC 5 Controllers (Photosharing) vs. CORE Module 4: Developing Controllers 01\_WorldJourney\_begin

---

Scenario – ver c

You have been asked to *add a controller to the photo sharing application that corresponds to the Photo model class that you have created in an earlier module.*

- The controller should include actions that respond when users upload photos, list all photos, display a single photo, and delete photos from the application.
- You should also add an action that returns the photo as a .jpg file to show on a webpage.

Demo: Optional exercise: Write Controllers and Actions, Action Filters, Add Routes

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD04\\_DEMO.md#demonstration-how-to-add-routes](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD04_DEMO.md#demonstration-how-to-add-routes)

Create an action filter class that logs the details of actions, controllers, and parameters to the Visual Studio Output window, whenever an action is called.

Exercise 3: Using the Photo Controller

Task 1: Create the Index and Display views

Ver D

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD04\\_LAK.md](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD04_LAK.md)

## Module 5: Developing ASP.NET MVC 5 Views vs. CORE RazorSyntaxExample

Scenario PhotoSharingApplication

[https://github.com/MicrosoftLearning/20486C-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD05\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486C-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD05_LAB_MANUAL.md)

You have been asked to add the following views to the **photo sharing application**:

- A Display view for the Photo model objects. This view will display a single photo in a large size, with the title, description, owner, and created date properties.

- A Create view for the Photo model objects. This view will enable users to upload a new photo to the gallery and set the title and description properties.
- A Photo Gallery partial view. This view will display many photos in thumbnail sizes, with the title, owner, and created date properties. This view will be used on the All Photos webpage to display all the photos in the application. In addition, this view will also be used on the home page to display the three most recent photos.

After adding these three views to the photo sharing application, you will also test the working of the web application.

### Objectives

After completing this lab, you will be able to:

- Add Razor views to an MVC application and set properties such as scaffold and model binding.
- Write both HTML markup and C# code in a view by using Razor syntax.
- Create a partial view and use it to display re-usable markup.

\*Use Razor syntax demo: [https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD05\\_DEMO.md#demonstration-how-to-use-the-razor-syntax](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD05_DEMO.md#demonstration-how-to-use-the-razor-syntax)

### HTML Tags and Helpers

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD05\\_DEMO.md#lesson-2-using-html-helpers-and-tag-helpers](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD05_DEMO.md#lesson-2-using-html-helpers-and-tag-helpers)

### Mod 6 Dev Models (ppt) VER D

#### Demo lab: Forms - Use Display and Edit Data Annotations

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD06\\_DEMO.md#lesson-2-working-with-forms](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD06_DEMO.md#lesson-2-working-with-forms)

### Module 6: **Testing and Debugging** ASP.NET MVC 5 Web Applications VER C

#### Scenario

The Photo Sharing application is in the early stages of development. However, frequent errors are hindering the productivity of the development team.

The senior developer advises that you intercept exceptions and other flaws as early as possible.

You have been asked to perform unit tests of the PhotoController to ensure that all scenarios work as expected and to avoid problems later in the web application development life cycle.

You have also been asked to ensure that when critical errors occur, developers can obtain helpful technical information.

#### Objectives

After completing this lab, you will be able to:

- Perform unit tests of the components of an MVC web application.
- Configure an exception handling strategy for an MVC web application.
- Use Visual Studio debugging tools against a web application.

[https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD06\\_LAK.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD06_LAK.md)

## Module 7: Structuring ASP.NET MVC 5 Web Applications

Scenario PhotoShareApp

[https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD07\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD07_LAB_MANUAL.md)

An important design priority for the Photo Sharing application is that the visitors should be able to easily and logically locate photographs. Additionally, photo galleries and photos need to appear high in search engine results. To implement these priorities, you have been asked **to configure routes** that enable the entry of user-friendly URLs to access photos.

You have been asked to ensure that the URLs of the following forms work to display a photo:

~/photo/display/Photoid. In this form of URL, Photoid is the database ID of the photo object. This form of URL already works because it matches the default route.

~/photo/Photoid. In this form of URL, Photoid is the database ID of the photo object. This is the logical URL to enter when you know the ID of the photo that you want to access.

~/photo/title/PhotoTitle. In this form of URL, PhotoTitle is the title of the photo object. This is the logical URL to enter when you know the title of the photo that you want to access.

You have also been asked to implement the following navigation controls in the Photo Sharing application:

A menu with links to the main site areas

- A breadcrumb control
- These navigation controls will be added to the menu after the completion of the main site areas.

### Objectives

After completing this lab, you will be able to:

- Add routes to the ASP.NET Routing Engine in an ASP.NET MVC application.
- Build navigation controls within ASP.NET views.

## Module 7: Using Entity Framework Core in ASP.NET Core - Migrations

[https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D\\_MOD07\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486D-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486D_MOD07_LAB_MANUAL.md)

### Objectives

After completing this lab, you will be able to:

- Add new models to the application and add properties to the models.
- Add a class that derives from the DbContext class.
- Use a repository for the application to connect to the database.
- Use Entity Framework Core to retrieve and store data.
- Use Migrations to create a database and to update the schema of the database.

### Using LINQ to Entities

### Loading Related Data

## Manipulating Data by Using Entity Framework

**Module 8: Using Layouts, CSS and JavaScript in ASP.NET Core MVC**PhotoShare: [https://github.com/MicrosoftLearning/20486-](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD08_LAB_MANUAL.md)[DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD08\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD08_LAB_MANUAL.md)**Mod 9**

use partial page updates. You also want to return pages in quick time, while updated information is displayed, for which you decide to configure caching in your application.

[https://github.com/MicrosoftLearning/20486-](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD09_LAB_MANUAL.md)[DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD09\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD09_LAB_MANUAL.md)**Module 10: Using JavaScript and jQuery for Responsive MVC 5 Web Applications:**[https://github.com/MicrosoftLearning/20486-](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD10_LAB_MANUAL.md)[DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD10\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD10_LAB_MANUAL.md)

- Render and execute JavaScript code in the browser.
- Use the jQuery script library to update and animate page components.
- Use jQueryUI widgets in an MVC application.

**Module 11: Controlling Access to ASP.NET MVC 5 Web Applications**[https://github.com/MicrosoftLearning/20486-](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD11_LAB_MANUAL.md)[DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD11\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD11_LAB_MANUAL.md)

- Configure a web application to use ASP.NET Form Authentication with accounts stored in Microsoft Azure SQL database.
- Write models, controllers, and views to authenticate users in a web application.
- Provide access to resources in a web application.
- Enable users to reset their own password.

**Mod 12****Module 12: Building a Resilient ASP.NET MVC 5 Web Application**[https://github.com/MicrosoftLearning/20486-](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD12_LAB_MANUAL.md)[DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD12\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD12_LAB_MANUAL.md)**Scenario**

The senior developer has asked you to implement the following functionality in your Photo Sharing web application.

- Any visitor of the application, including anonymous users, should be able to mark a photograph as a favorite.
- If a user has marked a favorite, a link should be available to display the favorite photo.
- Favorite photos should be displayed in the slideshow view.

**Objectives**

After completing this lab, you will be able to:

- Store a setting for an anonymous or authenticated user in session state.
- Check a user preference when rendering an action link.
- Render a webpage by checking state values in the application.

## Mod 13

## Module 13: Implementing Web APIs in ASP.NET MVC 5 Web Applications

[https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD13\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD13_LAB_MANUAL.md)

## Scenario

Your manager wants to ensure that the photos and information stored in the Photo Sharing application can be integrated with other data in web mash-ups, mobile applications, and other locations. To re-use such data, while maintaining security, you need to implement a RESTful Web API for the application. You will use this Web API to display the locations of photos on a Bing Maps page.

## Objectives

After completing this lab, you will be able to:

- Create a Web API by using the new features of ASP.NET MVC 5.
- Add routes and controllers to an application to handle REST requests.
- Call a REST Web API from jQuery client-side code.

**Module 14: Handling Requests in ASP.NET MVC 5 Web Applications**

Add chat functionality to the application – SignalR

[https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD14\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD14_LAB_MANUAL.md)

## Module 15: Deploying ASP.NET MVC 5 Web Applications

[https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C\\_MOD15\\_LAB\\_MANUAL.md](https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD15_LAB_MANUAL.md)

- Prepare an MVC web application for deployment.
- Deploy an MVC web application to Microsoft Azure.

space