



# ASP.net MVC Syllabus

## Overview

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller. The ASP.NET MVC framework provides an alternative to the ASP.NET Web Forms pattern for creating MVC-based Web applications. The ASP.NET MVC framework is a lightweight, highly testable presentation framework that (as with Web Forms-based applications) is integrated with existing ASP.NET features, such as master pages and membership-based authentication. The MVC framework is defined in the System.Web.Mvc namespace and is a fundamental, supported part of the System.Web namespace.

The MVC pattern helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. The pattern specifies where each kind of logic should be located in the application. The UI logic belongs in the view. Input logic belongs in the controller. Business logic belongs in the model. This separation helps you manage complexity when you build an application, because it enables you to focus on one aspect of the implementation at a time. For example, you can focus on the view without depending on the business logic. The course is hands-on with many example programs and lab exercises.

## Course Goals

- Gain an understanding of the philosophy and architecture of WPF
  - Create Windows applications using the classes provided by WPF
  - Understand the principles of XAML and create applications using a combination of code and XAML
  - Use the layout features of WPF to create flexible and attractive user interfaces
  - Implement event and command-driven applications with windows, menus, dialogs, toolbars, and other common user interface features
  - Use more advanced features of WPF such as dependency properties, routed events, logical resources, styles, templates, and data binding
- Understand the MVC design pattern and how it's applied in ASP.NET MVC

- Recognize the benefits and drawbacks of using Asp.Net Become equipped to make good choices about model design and use of Microsoft data access technologies
- Use Asp.Net MVC's routing system to achieve a REST-style architecture
- Building Model
- Routes and URLs
- Controllers
- View and View Templates
- Learn how to build a compelling and maintainable HTML user interface using both the ASP.NET and new Razor view engine
- Filters
- See how to easily add AJAX and jQuery capabilities to an ASP.NET MVC application

✓ Introduction

- ASP and ASP.NET WebForms
- The MVC Design Pattern
- Unit Testing
- Managing Dependencies

✓ ASP.NET MVC Architecture

- ASP.NET Platform
- Controllers, Models, and Views
- URL Routing
- Controller Actions
- Razor View Engine
- Extensibility

✓ Defining the Model

- Object Relational Mapping (ORM)
  - ADO.NET
  - Entity Framework (EF)
  - nHibernate
- Using an IoC Container

✓ Routes and URLs

- Introduction to Routing
- Defining Routes
- Route Evaluation
- The Controller Factory
- Constraints

- Exempting URLs
- Constructing Outgoing URLs
- Unit Testing Routes
  
- ✓ Controllers and Actions
  - IController, ControllerBase, and Controller
  - Defining Actions
  - Action Selectors
  - Action Filters
  - HTTP Verbs
  - HttpContext and RouteData
  - Maintaining Testability
  - Returning Data with ActionResult
  - Parameters and the Model Binder
  - Asynchronous Action Methods
  
- ✓ Views
  - View Engines
  - Templates and Scaffolding
  - Razor Syntax
  - ViewData and ViewBag
  - Strongly-Typed Views
  - Layout Pages
  - Custom Sections
  - Partial Views
  - Child Actions
  - Using a ViewModel Object
  
- ✓ HTML Helpers
  - Basic Helpers
  - Strongly-Typed Helpers
  - Creating Custom Helpers
  - Declarative Helpers
  
- ✓ Data Validation
  - Data Annotations
  - Validation HTML Helpers
  - ModelState
  - IValidatableObject

- Client Side Validation
  
- ✓ Authentication and Authorization
  - Using Windows Authentication
  - Building a Custom Account Controller
  - Configuring Forms Authentication
  - Using the Authorize Attribute
  
- ✓ Ajax
  - Controller Actions for Ajax
  - Ajax Helpers
  - Unobtrusive Ajax
  - Ajax with jQuery
  
- ✓ Web API
  - Using HTTP as an Application Protocol
  - Content Negotiation
  - Routing
  - ApiController
  - MediaTypeFormatters
  - Validation
  - OData Queries
  
- ✓ Customizing and Extending ASP.NET MVC
  - ASP.NET MVC Pipeline
  - Advanced Routing
  - Custom Global Filters
  - Customizing MVC Code Templates
  - Creating a Custom Controller Factory
  
- ✓ Deployment
  - Server Requirements
  - Configuration Options