

MVC Applications with Entity Framework and Security

Duration: 5 days

Course	Developing ASP.NET with MVC Application
Duration	5 days
Format	Day-wise modules with hands-on labs

Day 1: Introduction to MVC and ASP.NET Application Setup

Module 1: Introduction to Web Application Architecture

- Traditional Web Forms vs MVC architecture
- Why MVC is used in enterprise applications
- Separation of concerns
- Request-response lifecycle
- Role of Model, View, and Controller

Module 2: ASP.NET MVC Project Setup

- Creating an ASP.NET MVC project
- Understanding project folder structure
- Controllers folder
- Models folder
- Views folder
- wwwroot/static files
- Program.cs and application startup configuration

Module 3: Controllers and Actions

- Creating controllers
- Action methods
- Returning different result types
- ViewResult
- RedirectResult
- JsonResult
- ContentResult
- FileResult
- Passing data from controller to view

Hands-on Labs

- Create a new ASP.NET MVC application
- Create HomeController and multiple action methods
- Return views and basic content from controller actions
- Build a simple navigation flow between pages

Day 2: Views, Razor Syntax, Layouts, and Routing

Module 1: Working with Views

- Introduction to Razor View Engine
- Razor syntax basics
- HTML helpers
- Tag helpers
- Strongly typed views
- Partial views
- ViewBag, ViewData, and TempData

Module 2: Layouts and UI Structure

- Creating shared layouts
- Sections in Razor views
- Reusable UI components
- Static files: CSS, JavaScript, images
- Bootstrap integration

Module 3: Routing in ASP.NET MVC

- Default route pattern
- Conventional routing
- Attribute routing
- Route parameters
- Optional parameters
- Route constraints

Hands-on Labs

- Create Razor views for application pages
- Design a common layout page
- Use ViewBag, ViewData, and TempData
- Configure controller-based and attribute-based routing

Day 3: Models, Model Binding, and Validation

Module 1: Working with Models

- Purpose of models in MVC
- Creating domain models
- ViewModels vs Models
- Passing model data to views
- Displaying model data in Razor views

Module 2: Model Binding

- What is model binding?
- Binding form data to action parameters
- Binding form data to model objects
- Binding query string values
- Binding route values
- Binding complex objects
- Binding collections
- Include and exclude properties during binding
- Overposting and security considerations

Module 3: Model Validation

- Client-side vs server-side validation
- Data annotation attributes
- Required
- StringLength
- Range
- RegularExpression
- EmailAddress
- Compare
- ModelState and validation lifecycle
- Displaying validation messages
- Custom validation attributes
- Remote validation overview

Hands-on Labs

- Create a Product or Employee model
- Build a form using Razor
- Bind form data to a model object

- Apply validation using data annotations
- Display validation errors on the view
- Prevent invalid data submission using ModelState

Day 4: Database Operations and CRUD Application Development

Module 1: Entity Framework Integration

- Introduction to Entity Framework
- Code First approach
- DbContext and DbSet
- Configuring database connection
- Creating database using migrations
- Understanding migration commands

Module 2: CRUD Operations in MVC

- Create operation
- Read/List operation
- Details page
- Update operation
- Delete operation
- Form submission flow
- Redirect after post pattern

Module 3: Repository and Service Layer Overview

- Why use repository pattern?
- Creating service classes
- Separating business logic from controllers
- Dependency Injection overview
- Clean MVC application structure

Hands-on Labs

- Create database model and DbContext
- Configure Entity Framework
- Generate database using migrations
- Build complete CRUD functionality
- Add search and basic filtering to list page

Day 5: Filters, Security, Error Handling, and Final Application

Module 1: Filters in ASP.NET MVC

- Purpose of filters
- Filter execution pipeline
- Types of filters:
 - Authorization Filters
 - Resource Filters
 - Action Filters
 - Exception Filters
 - Result Filters
- Built-in filters
- Creating custom filters
- Global filters
- Controller-level filters
- Action-level filters
- Filter ordering

Module 2: Authentication, Authorization, and Security

- Authentication vs authorization
- Authorize attribute
- Role-based authorization overview

- Protecting controller actions
- Anti-forgery token
- Preventing CSRF attacks
- Input validation and secure form handling
- Overposting protection

Module 3: Error Handling and Application Finalization

- Exception handling in MVC
- Custom error pages
- Logging overview
- Environment-based error handling
- Application configuration
- Deployment overview

Hands-on Labs

- Apply built-in authorization filter
- Create a custom action filter for logging
- Create a custom exception filter
- Add anti-forgery protection in forms
- Complete final MVC application with CRUD, validation, filters, and basic security