

# Instructor's Summary for *Murach's ASP.NET Core MVC*

The instructor's materials for *Murach's ASP.NET Core MVC* will help any college instructor or corporate trainer run an effective course based on the book. This summary introduces you to these materials and helps you get started using them.

At the least, we recommend that you read the topics under *What's included in the instructor's materials* because they not only describe the components but also our underlying instructional philosophy. We also recommend that you read *How to get started with our materials* because that provides the installation details that you'll need and gives you charts that summarize the components at a glance.

But first, some thoughts about the modular structure of this book that you should be aware of. This structure is important because it gives you instructional options that you just don't have with other books.

<b>About the modular structure of the book .....</b>	<b>2</b>
Section 1: Get off to a fast start .....	2
Section 2: Master the essential skills .....	2
Section 3: Add more skills as you need them .....	2
<b>What's included in the student download .....</b>	<b>3</b>
Book applications .....	3
Exercise starts .....	3
Exercise solutions .....	3
<b>What's included in the instructor's materials .....</b>	<b>4</b>
The student download files .....	4
Objectives .....	4
Test banks .....	4
Projects and solutions .....	5
Case study and section-by-section solutions .....	5
PowerPoint slides .....	5
<b>How to get started with our materials .....</b>	<b>6</b>
The student download files that get installed .....	7
The instructor materials that get installed .....	7
<b>Any comments? .....</b>	<b>7</b>



**Mike Murach & Associates, Inc.**

1-800-221-5528 • (559) 440-9071 • Fax: (559) 440-0963

[murachbooks@murach.com](mailto:murachbooks@murach.com) • [www.MurachForInstructors.com](http://www.MurachForInstructors.com)

Copyright © 2020 Mike Murach & Assoc. Inc. All rights reserved.

## About the modular structure of the book

---

*Murach's ASP.NET Core MVC* is designed to give you flexibility in presenting the content that's right for your course and your students. For example, after your students complete section 1, you can continue by teaching any chapter in section 2. In other words, the chapters in section 2 are written as independent modules that require only section 1 as a prerequisite. That's what we mean by *modularity*, and that lets you choose the subjects you want to teach, as well as the sequence in which you teach them.

Beyond that, you have some options as to which chapters you assign in each section of the book. Those options are described in the topics that follow. Once you understand your options, you should have no problem selecting the chapters and teaching sequence that's best for your course.

### Section 1: Get off to a fast start

---

Section 1 presents a 5-chapter course that's designed to get your students off to a great start. It shows how to use ASP.NET Core MVC to develop single-page MVC web apps by the end of chapter 2, and it shows how to develop multi-page MVC web apps that use a database by the end of chapter 4. Along the way, it shows how to use Bootstrap to create responsive web apps, and it shows how to test and debug web apps, a part of the job that many books treat too lightly or too late. At that point, your students will be ready for rapid progress in the sections that follow.

If you want to use Visual Studio Code (VS Code) instead of the Visual Studio IDE, you can integrate chapter 17 into this section. It shows how to use VS Code to perform all of the same tasks that chapters 2-5 show how to perform with Visual Studio.

### Section 2: Master the essential skills

---

Section 2 presents essential skills that every ASP.NET Core MVC developer should have. To do that, this section reviews and expands on some skills presented in section 1 such as routing, Razor views, model binding, data validation, and EF (Entity Framework) Core. It also adds new skills like working with session state and cookies. Then, it finishes by presenting a realistic website that shows how all these skills fit together.

We recommend teaching the chapters in this section in sequence, but you can change the sequence if you want. In addition, if you're short on time, you can skip or skim chapter 13 and move on to any of the chapters in section 3.

### Section 3: Add more skills as you need them

---

Section 3 presents more skills that are commonly needed by professional ASP.NET MVC programmers. That includes how to work with dependency injection (DI), unit testing, custom tag helpers, partial views, view components, authentication, and authorization. As we mentioned earlier, it also includes the Visual Studio Code editor, an increasingly popular alternative to the Visual Studio IDE.

Again, we recommend teaching these chapters in sequence, although you may not want to cover all the topics in each chapter. As we said before, you can also use the material in chapter 17 early in the course, if you want your students to use VS Code instead of Visual Studio from the start. And if you don't have time to go into this section at all, the material will still be available to your students who become professional developers or who are driven by their own curiosity to learn more. Sections 1 and 2 will provide the foundation they need to explore these chapters on their own.

## What's included in the student download

---

To help your students get the most from our book, our retail website ([www.murach.com](http://www.murach.com)) lets your students download (1) the source code for the book applications, (2) the starting points for the exercises that are at the end of each chapter, and (3) the *solutions* to those exercises. Appendixes A (Windows) and B (macOS) show how to download and set up these files.

### Book applications

---

All of the web applications in this book are stored in a folder named *book\_apps*. That way, your students can run these applications to see how they work. They can review all of the code in any application when the book doesn't present it all. And they can copy and paste code from the book applications into their own applications.

If an application requires a database, the migration files for the database are included in the application. That way, your students can create the database by running the Update-Database command as described in appendix A (Windows) or the database update command as described in appendix B (macOS).

### Exercise starts

---

To help your students master the programming skills that are presented, the book provides exercises at the end of each chapter. For most of these exercises, the students start from an existing web app. That way, your students get the most practice in the least time. These starting points are stored in a folder named *ex\_starts*.

If you review the exercises in the book, you'll see that they guide the students through the process of building and modifying web apps. They also guide the student in using the critical skills for web application development. Because of that, if your students can successfully do all of the exercises, they will be well on their way to a professional level of competence.

### Exercise solutions

---

To help students get over any learning obstacles when they're working on their own, the student download also provides the solutions to the book exercises in a folder named *ex\_solutions*. That way, the students can check the solutions to see how something is done whenever they're wasting time on what is likely to be a trivial coding mistake. We think that providing the solutions is the right approach didactically because it helps students learn faster and better.

We realize, however, that this makes it difficult for an instructor to use the exercises to test their students. That's why the instructor's materials include a set of projects as well as an extensive case study that can be used for testing the competency of students. The instructor's materials include solutions for these projects and the case study, but these solutions are available only to instructors, not to students.

## What's included in the instructor's materials

---

The instructor's materials for this book are designed to save you time in preparing and running an effective course based on the text so that your students gain the programming skills they'll need on the job. So besides the materials in the student download, we provide instructional objectives, test banks, projects, a case study, and PowerPoint slides. A summary of these materials follows.

### The student download files: Book applications, exercise starts, and exercise solutions

---

These are the same materials that your students can download from our retail website. We've included them in the instructor's materials so that you can easily demonstrate and review the book examples and exercise solutions in class, without having to download them separately.

### Objectives

---

We believe that instructional objectives should be the start of any educational methodology, so we provide a set of objectives for each chapter in the book. We developed these based on the principles presented by Robert F. Mager in his classic book, *Preparing Instructional Objectives*. As a result, our objectives describe the skills that your students or trainees should have when they complete a chapter, and you should be able to test whether they can apply those skills.

Beyond that, we've tried to make sure that each objective describes a skill that a professional programmer should have. This gives our objectives a real-world context that you usually won't find in the objectives for other books. So, if your students or trainees can do what the objectives state when the course is over, you can be sure they've learned the skills that they will actually need on the job.

If you review the objectives, you'll see that the first objectives for each chapter are what we refer to as *applied objectives*. These ask the students to apply what they've learned as they develop ASP.NET MVC applications. These of course are the critical objectives of a programming course, and they are best tested by having the students or trainees do the projects and case study problems that we provide.

After the applied objectives for each chapter, you'll find what we refer to as *knowledge objectives*. These objectives define skills like identifying, describing, and explaining the required concepts, terms, and procedures. These objectives determine whether your students are able to talk intelligently about the topics that are presented. And these objectives can be tested by the questions in our test banks.

To help you get the most from the instructional objectives, we have included them at the start of the PowerPoint slides for each chapter. As we see it, if you can convince your students that they only need to have the skills that are described by the objectives, their study becomes far more focused and efficient.

### Test banks

---

To test comprehension, the instructor's materials include one test bank for each chapter in the book. We developed these test banks in ExamView, and we provide them in multiple formats, including those that can be used in various LMSs (like Blackboard, D2L, and Canvas) as well as Rich Text Format (for Word).

Each test bank provides questions that are designed to test the skills that are described by the objectives for that chapter, and each test question is designed to test the skill described by one objective. This keeps the promise to the students that they will only be expected to have the skills that are described by the objectives.

In our test banks, we use only multiple-choice questions because they have the highest validity. To us, that means that the students or trainees who get the best scores are also the ones with the best knowledge and skills. In contrast, matching and true/false questions have low validity, so we don't use them.

Besides matching our questions to the objectives, we use this guideline to check the validity of each question: *A professional ASP.NET Core MVC developer should be able to get the right answer.* This guideline eliminates questions that test the knowledge of trivial details that no one should be expected to remember. This guideline also forces us to focus on questions that test the concepts and skills that are required on the job.

## Projects and solutions

---

For all of the chapters that teach programming skills (chapters 2-16), the instructor's materials include one or more projects that you can assign at the end of the chapter. Most of these projects show the user interface for an app with some specifications and ask the student to build the app from scratch. This is a great way to test whether students have learned the skills they need to build their own web apps.

Some of these projects are short enough that they can be done in an hour or two, so you may want to use them in computer lab for practice or as tests. Others are more extensive and take longer, so you may want to have your students start these projects in lab but finish them outside of class, or assign them as take-home tests.

The instructor's materials also include the solutions to these projects. This allows you to demonstrate the web apps in class so your students have an idea of how they work. In addition, after your students finish a project, you can present the code that we've developed for the solution and compare it to the solutions that the students develop. That can help your students see other options for coding various parts of the app.

## Case study and section-by-section solutions

---

To provide a more extensive way to test your students' skills, the instructor's materials include a case study that has your students build a website called SportsPro Technical Support. This case study is a series of related assignments that allows your students to build a realistic website. Since most chapters include an assignment, your students can add to the website after each chapter that they finish.

The instructor's materials also include the solutions for this case study at important points in its development. For example, there's a solution for where the website should be by the end of chapter 4 (section 1), by the end of chapter 12 (section 2), and by the end of chapter 16 (section 3).

## PowerPoint slides

---

The PowerPoint slides present all of the critical information from the figures of the book. That includes all of the screenshots, diagrams, tables, and code examples that you may want to go over in class. As a result, these slides make it easy for you to review any of the skills that your students have difficulty with. Beyond the book information, the slides for each chapter start with the instructional objectives so you can review them in class.

## How to get started with our materials

---

You can request the instructor's materials for our book from our instructor website ([www.murachforinstructors.com](http://www.murachforinstructors.com)) and download them from your account page there. The download is available as an executable file (for Windows systems only) or as a zip file. Then, you can install the materials on your computer.

- When you double-click on the executable file, it will install the folders and files onto your C drive in a folder structure that starts with  
`c:\murach\aspnet_core_mvc`
- When you use the zip file, you'll want to make sure you have a folder named *murach* on your C drive and unzip into that. Again, the folder structure will start with `c:\murach\aspnet_core_mvc`.

The table on the next page shows all the components and where to find them in the file structure.

Once the installation is done, you can do a thorough review of all of the materials. In particular, you'll want to run some of the book apps, exercise solutions, project solutions, and case study solutions to see the level of competence that our book develops. You'll also want to click through some of the PowerPoint slides to see how they can help you review and reinforce the information that's presented in the book. And you'll want to take a look at the objectives to see all the skills that this book covers.

## The student download files that get installed

murach\aspnet_core_mvcl student_download\	Contents
book_apps\ 	The Visual Studio projects for the web apps presented throughout the book.
ex_starts\ 	The Visual Studio projects that are the starting points for the exercises at the ends of the chapters.
ex_solutions\ 	The Visual Studio projects that are the solutions for the chapter exercises.

## The instructor materials that get installed

murach\aspnet_core_mvcl instructors\	Contents
Instructor's summary.pdf Objectives.docx Objectives.pdf	This instructor's summary in PDF format. The instructional objectives for all chapters in both Word and PDF formats.
student_projects\ Projects.docx Projects.pdf project_starts\ project_solutions\ 	The specifications for the chapter-by-chapter student projects in both Word and PDF. The starting points for some of the projects. The solutions for all the projects.
case_study\ Case study.docx Case study.pdf case_study_start\ case_study_solutions\ 	The specifications for the SportsPro Technical Support case study in both Word and PDF formats. The starting point for the case study. The solutions for the case study at the end of each section of the book.
Slides\ Test banks\ 	One PowerPoint file for each chapter. One test bank for each chapter in various formats: ExamView, RTF (Word), Blackboard (which can be imported into Canvas and D2L), Respondus, and IMS QTI (the standard test bank format).

## Any comments?

If you have any comments about our book or its instructor's materials, we would be delighted to hear from you. If you discover any errors in our applications or solutions, we would appreciate hearing about them. And if you want to let us know that you're going to adopt our book for your course, that would make our day.

Just email us at the addresses below. But whether or not we hear from you, we want to thank you for your interest in our ASP.NET Core MVC book.

Joel Murach, Author  
[joel@murach.com](mailto:joel@murach.com)

Judy Taylor, Educational Liaison  
[judy@murach.com](mailto:judy@murach.com)