

Razvan Alexandru Mezei

Introduction to the Development of Web Applications Using ASP .Net (Core) MVC

Synthesis Lectures on Computer Science

The series publishes short books on general computer science topics that will appeal to advanced students, researchers, and practitioners in a variety of areas within computer science.

Razvan Alexandru Mezei

Introduction to the Development of Web Applications Using ASP .Net (Core) MVC

Razvan Alexandru Mezei
Department of Computer Science
Hal and Inge Marcus School of Engineering
Saint Martin's University
Lacey, WA, USA

ISSN 1932-1228 ISSN 1932-1686 (electronic)
Synthesis Lectures on Computer Science
ISBN 978-3-031-30625-9 ISBN 978-3-031-30626-6 (eBook)
<https://doi.org/10.1007/978-3-031-30626-6>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*I would like to dedicate this work to my fiancée
Adriana Cheteles for her great support and
encouragement and to my Ph.D. advisor,
Prof. George Anastassiou, whose mentorship
and guidance opened my professional path.*

Razvan Alexandru Mezei

Preface

This work is intended to be used as a (quick) one-semester introduction to Web Applications development using ASP .Net Core MVC. In particular, it briefly introduces some client-side languages and frameworks (HTML, CSS, JavaScript, and Bootstrap), then it focuses primarily on the server-side portion (C#, Entity Framework Core) using the Model-View-Controller (MVC) pattern. Along the way, you will be introduced to many concepts such as routing, services and dependency injection, object relational mapper, model validation, and authentication. By the end of this book, you will create a web application that stores its data in a database and includes some basic account management functions (register user accounts, login, and logout).

Lacey, WA, USA

Razvan Alexandru Mezei

Contents

1	Introduction	1
2	Prepare the Development Environment	5
2.1	Choose a Web Browser	5
2.2	Install Visual Studio Code	6
2.3	Install Visual Studio	6
2.4	Install DB Browser for SQLite	7
2.5	Miscellaneous/Optional	8
2.5.1	Show File Name Extensions	8
2.6	Microsoft SQL Server	8
2.6.1	Sample Data Generators	8
3	Brief Introduction to Html	9
3.1	Let's Create Our First HTML Page	9
3.2	Add Titles, Paragraphs and Headings	11
3.3	Add a Second Webpage	13
3.4	Add Links and White Spaces to Our Pages	13
3.5	Add Images and White Spaces to Our Pages	15
3.6	Tables and Buttons	17
3.7	A Few Other HTML Elements We'll Use Later	19
3.7.1	Label and Select Elements	19
3.7.2	Input Elements	20
3.8	Form and More on Input Elements and Attributes	21
3.9	GET Versus POST Request, the Action and the Method Attributes	25
4	Brief Introduction to CSS, Javascript, and Bootstrap	29
4.1	Motivation for Using CSS and JavaScript	29
4.2	Our First CSS Example	30
4.3	Introduction to CSS Syntax	31
4.4	CSS Selectors	33
4.5	Conflicting CSS Specifications	37

4.6	Other CSS Selectors	38
4.7	A Few More Examples of Property-Value Pairs for CSS	38
4.7.1	Text Color in CSS	39
4.7.2	Text Alignment in CSS	39
4.7.3	Fonts in CSS	40
4.8	The Box Model and the Developer Tools	43
4.9	The DIV Element	45
4.10	Ways to Add CSS	45
4.10.1	Internal	45
4.10.2	In-line	45
4.10.3	External	47
4.11	First Encounter with Bootstrap	48
4.11.1	Add Bootstrap 5 .css to Our Webpages	48
4.11.2	Bootstrap 5 Tables	49
4.11.3	Bootstrap 5 Buttons and Links	50
4.11.4	Bootstrap 5 Container, Padding	51
4.11.5	Bootstrap 5 Source Code	51
4.11.6	Center Contents with < DIV> and CSS	52
4.12	Introduction to JavaScript	54
4.13	JavaScript Statements	56
4.14	JavaScript Functions	56
4.15	Add JavaScript to Our Webpages	57
4.16	Introduction to the Document Object Model (DOM)	58
4.17	Add Event Handlers	60
4.18	An Example: Toggle Between Dark/Light Mode	61
4.19	The Back Button	62
4.20	External JavaScript	63
4.21	More Introduction to Bootstrap	64
4.22	Ways to Include Bootstrap in Our Projects	64
4.23	Some CDNs for Bootstrap 5	64
4.24	View Bootstrap 5 Source Files	65
4.25	Bootstrap 5 navbar	65
5	Some C# Fundamentals	69
5.1	Hello World in C# (Console Application)	69
5.2	Top-Level Statements	70
5.3	Namespaces, Using Directive, and Global Using Directive	71
5.3.1	Namespaces	71
5.3.2	Using Directives	71
5.3.3	Implicit Using Directives	73
5.3.4	Global Using Directives	73
5.4	Comments	74

5.5	Existing Data Types	74
5.6	String Interpolation	76
5.7	Enumerations	77
5.8	Classes	78
5.9	References and Objects	79
5.10	Instance Variables/Non-static Fields	80
5.11	Dot Notation	81
5.12	Methods	82
5.13	The this Keyword	84
5.14	Access Modifiers	84
5.15	Properties	85
5.16	Constructors	87
5.17	Method Overloading	88
5.18	Conditionals, Loops, and Lists	88
5.19	Collections and Generic Collections	90
5.20	Inheritance	90
5.21	The base Keyword and the Constructors	92
5.22	Interfaces	95
	5.22.1 Some Motivation	96
	5.22.2 How to Define an Interface	96
	5.22.3 How to Implement an Interface	96
5.23	How to Use an Interface	97
5.24	Lambda Expressions	98
5.25	LINQ	100
5.26	Working with null Values	102
5.27	Solution Files .sln	102
5.28	Other Resources for Learning C#	103
6	Middleware, Services, Intro to Dependency Injection	105
6.1	What Are ASP .Net (Core) MVC Web Applications?	105
6.2	An Introduction to the MVC Pattern	107
6.3	A Quick Dive into an MVC Example (Optional)	108
6.4	Let's Start Our ASP .Net Core Application Project in Here	113
	6.4.1 The Empty Web Application Starting Point	113
	6.4.2 The MVC Web Application Starting Point	114
6.5	Entry Point to Our Web Application: Program.cs	114
6.6	The Middleware Pipeline	115
	6.6.1 The Current Code in Our Project	115
	6.6.2 Run, Use, and Map	117
	6.6.3 First Example	117
	6.6.4 Second Example	119
	6.6.5 Third Example	119

6.6.6	Other Middleware Components	121
6.7	Static Files Middleware	122
6.7.1	What Are <i>Static Files</i> ?	122
6.7.2	Where Do We Store <i>Static Files</i> ?	122
6.7.3	How Do We Allow Access to <i>Static Files</i> ?	122
6.7.4	How Can We Access <i>Static Files</i> ?	123
6.7.5	Default (Static) Page	124
6.8	Introduction to Services (Optional)	128
6.8.1	Example—Step 1: Define a Class and An Interface	128
6.8.2	Example—Step 2: Register a Service	129
6.8.3	Example—Step 3: Use a Service	129
7	Routing, Models, and Controllers	131
7.1	A Little Cleanup Before We Continue	131
7.2	Some Essential MVC Concepts and the HTTP Request Lifecycle	132
7.3	Introduction to Routing	134
7.3.1	Adding MVC to Our ASP .Net Core Application	135
7.3.2	Default Routing, the Home Controller, and Actions	135
7.4	Add a Model, a Controller, and Views	139
7.4.1	Add a Model Class	139
7.4.2	Add a (Second) Controller Class	140
7.4.3	Add a First View	141
7.4.4	Test Our Code so Far	142
7.5	Various Action Result Types	143
7.6	Conventional Versus Attribute Routing	145
7.6.1	Conventional Routing	146
7.6.2	Attribute Routing	148
7.6.3	Mixing Routings	150
8	More on Controllers and Views, Introduction to Razor Syntax	153
8.1	A Little Cleanup Before We Continue	153
8.2	Some Essential MVC Concepts and the HTTP Request Lifecycle (Revisited)	154
8.3	Another Example of Model, Controller, and Views	156
8.3.1	The Instructor Model	156
8.3.2	The InstructorController Class	157
8.4	The Index Action and View	161
8.4.1	Add a View for Our Index Action	161
8.4.2	Strongly Typed and Weakly Typed Views	163
8.4.3	Introduction to Razor Engine and Razor Syntax	164
8.4.4	Action Using a View with a Different Name	167
8.5	The ShowDetails Action and View	169
8.5.1	The ShowDetails Action	169

8.5.2	The ShowDetails View	170
8.6	A First Look at Tag Helpers and HTML Helpers	173
8.6.1	A First Example of an HTML Helper	173
8.6.2	A First Example of a Tag Helper	174
8.6.3	Add Links to the Index View Using Tag Helpers and HTML Helpers	175
8.6.4	Add Bootstrap to the Index View	175
8.6.5	Add Links to the <i>ShowDetails</i> View	177
9	More on Views, Data Annotations	179
9.1	Introduction to Data Annotations	179
9.1.1	Update the ShowDetails View	179
9.1.2	Update the Index View (Optional)	184
9.2	The Add Action and View	186
9.2.1	The Add Action—GET	186
9.2.2	The Add View	187
9.2.3	The Add Action—POST	192
9.2.4	A Few More Details About the Model Binding	195
9.2.5	A Few More Details About the GET Versus POST	196
9.3	The Edit Action and View	198
9.3.1	The Edit Action—GET	198
9.3.2	Add Edit Links in the Index View	198
9.3.3	The Edit View	199
9.3.4	The Edit Action—POST	201
9.3.5	An Example of a Service	203
9.4	The Delete Action and View	207
9.4.1	The Delete Action—GET	207
9.4.2	Add Delete Links in the Index View	208
9.4.3	The Delete View	208
9.4.4	The DeleteConfirmed Action—POST	209
10	Model Validation	213
10.1	Step 1: Add (Built-in or Custom) Validation Attributes	216
10.2	Step 2: Enforce Validation by Making Use of the ModelState	217
10.3	Step 3: Display Error Messages via Validation Tag Helpers	218
10.3.1	To Display a Summary of All Error Messages	219
10.3.2	To Display In-line Error Messages	219
10.4	Let's Test Our Model Validation	221
10.5	Custom Validation Attributes (Optional)	222
10.5.1	Create a Custom Validation Attribute	223
10.5.2	Use a Custom Validation Attribute	226
10.5.3	Let's Test the Newly Added Custom Validation	226

10.6	Validation Text Styling	227
11	Persistent Data: Entity Framework Core	231
11.1	Introduction	231
11.2	Classes Involved: Providers, DbContext, and DbSet	232
11.3	Add Entity Framework Core to Our Web Application	232
11.3.1	Step 1: Create/Choose Your Entity Classes	233
11.3.2	Step 2: Install NuGet Packages	233
11.3.3	Step 3: Create a Class Derived from DbContext	234
11.3.4	Step 4: Data Seeding	235
11.3.5	Step 5: Register Our DbContext as a Service, and Use a Connection String	236
11.3.6	Test Our Database	239
11.4	Use Entity Framework Core in Our Web Application, Dependency Injection Revisited	240
11.4.1	Inject Entity Framework Core in InstructorController	240
11.4.2	Update the Actions to Use Entity Framework Core	240
11.4.3	Important: Automated Id Generation	242
11.4.4	Let's Test That We Have Persistent Data	243
11.4.5	EnsureDeleted	244
11.5	Practice: Update the StudentController Class	245
11.5.1	Inject Entity Framework in StudentController	245
11.5.2	Use Entity Framework Core in StudentController Actions	245
11.6	How to Use Microsoft SQL Server Instead of SQLite (Optional)	247
11.6.1	Install SQL Server Express LocalDB Database on Your Machines	247
11.6.2	Make Changes so Entity Framework Core Now Works with a Microsoft SQL Server Database	248
12	Consistent Look: Layouts, Friendly Error Pages, and Environments	251
12.1	Filter Results	251
12.1.1	Update the Index View	251
12.1.2	Update the Index Action	253
12.1.3	Implement the <i>Clear the Filter</i> Button	254
12.2	Filter Results Using a Dropdown List (Optional)	255
12.2.1	Create the Dropdown List Items in the Index Action	256
12.2.2	Display the Dropdown List Items in the Index View	256
12.2.3	Use of the Dropdown List to Filter Our Results	256
12.2.4	The Code	258
12.3	Consistent Webpages—Using Razor Layouts	258
12.3.1	Create a Layout	259
12.3.2	Use the Layout in Our Views	259

12.3.3	Add a Bootstrap 5 Navbar to Our Layout	265
12.3.4	Add Navigation Links to Various Actions and Controllers ...	265
12.4	Layout Sections (Optional)	267
12.4.1	Define a Section	267
12.4.2	Make Use of a Section	269
12.5	Make Use of Bootstrap 5 Buttons	270
12.5.1	The Index View	270
12.5.2	The ShowDetails View	273
12.5.3	Use Bootstrap for Styling Validation Errors	273
12.6	Configure a Friendly Error Page	277
12.6.1	Introduction	277
12.6.2	Work with Multiple Environments	278
12.6.3	The Developer Exception Page	279
12.6.4	The Friendly Error Page	280
13	Work with Images (Optional)	285
13.1	Add a New Property for the Image to the Model/Entity Class	285
13.2	Modify the Add View, so It Allows a User to Upload an Image	286
13.3	Modify the Add Action so the File Uploaded Gets Saved into the Database	287
13.4	Modify the ShowDetails Action to Transform the Byte Array Back into an Image	289
13.5	Modify the ShowDetails View so It Displays the Profile Image	289
13.6	Bootstrap Card Deck for the Index Action and View (Optional)	291
14	Introduction to Authentication. User Login, Logout, and Registration ...	293
14.1	Introduction to Some Security Concepts	293
14.2	Introduction to ASP .Net (Core) Identity	294
14.2.1	Step 1: Install NuGet Packages	295
14.2.2	Step 2: Define Our User Class (Derived from IdentityUser)	295
14.2.3	Step 3: Update Our DbContext Derived Class to Use Identity	296
14.2.4	Step 4: Register the Identity Services	296
14.2.5	Step 5: Add Authentication and Authorization Middleware Components	297
14.2.6	Test Your Work	298
14.2.7	Step 6: Register, Login, and Logout	298
14.2.8	Step 7: Add Simple Authorization to Our Web Application (Optional)	306
	References	311