

آدام فریمن

آموزش حرفه ای

ASP.NET Core 6



جلد سوم:

ویژگی های پیشرفته

مروّت گیوی

ترجمه  
Parjomac

آدام فریمن

## آموزش حرفه‌ای 6 ASP.NET Core

### جلد سوم: ویژگی‌های پیشرفته

ترجمه

مرّوت گیوی

انتشارات ترجمک

تابستان ۱۴۰۲

# آموزش حرفه‌ای ASP.NET CORE 6

## توسعه وب با ASP.NET Core

**توجه :**

کتاب حاضر ترجمه **مرّوت گیوی** از انتشارات ترجمک می‌باشد. فایل کتاب حاوی اطلاعات DRM (مدیریت حقوق دیجیتال) است. وقتی برای اولین بار فایل را باز می‌کنید، کد شناسایی کتاب به همراه آدرس IP سیستم شما ذخیره شده و زمانیکه آنلاین شوید، به سرور انتشارات ترجمک انتقال می‌یابد.

خواهشمند است به حقوق مترجمان و گروه فنی ترجمک احترام گذاشته و از توزیع بدون مجوز فایل کتاب اجتناب نمایید. شما با خرید و دانلود این کتاب موافقت نموده‌اید که اطلاعات فایل DRM به سرور انتشارات ترجمک انتقال یابد و در صورت محرز شدن نقض حقوق صاحب اثر، کلیه خسارات حاصله در طی فرآیند حقوقی و مطابق قانون حمایت حقوق مؤلفان و مصنفان و هنرمندان و ناشران جمهوری اسلامی (مصوب دوازده اسفند ۱۳۶۵ یا بعد از آن) از شما دریافت شود.

از اینکه با عرضه مقرون به صرفه کتاب‌های الکترونیک و شکوفایی انتشارات ترجمک همیاری می‌کنید، سپاسگزاریم.

**انتشارات ترجمک**

**برنامه نویسی و وب**

عنوان و نام پدیدآور	: آموزش حرفه‌ای ASP.NET Core 6/آدام فریمن؛ ترجمه مروت گیوی.
مشخصات نشر	: همدان: ترجمک، ۱۴۰۲
مشخصات ظاهری	: جلد ۳: ویژگی‌های پیشرفته
شابک	: دوره: ۱-۵۳-۷۸۵۵-۶۲۲-۹۷۸؛ جلد ۳: ۰-۵۰-۷۸۵۵-۶۲۲-۹۷۸
موضوع	: مایکروسافت دات‌نت فریم‌ورک، Microsoft .NET Framework، وبگاه‌ها - برنامه‌های تالیفی، Web sites -- Authoring programs، علوم کامپیوتر، Computer science، نرم‌افزار - مهندسی، Software engineering
مندرجات	: جلد ۱: ساخت وبسایت تجارت الکترونیک، ۲: کار با داده، ۴: اجرای پروژه مثال
عنوان دیگر	: آموزش کاربردی Pro ASP.NET Core MVC 6 همراه با پیاده‌سازی کامل یک پروژه با MVC و صفحات Razor
رده بندی کنگره	: QA76/76
رده بندی دیویی	: 005/276
شماره کتابشناسی ملی	: ۹۳۰۱۹۶۲

## شناسنامه کتاب



نام کتاب: آموزش حرفه‌ای ASP.NET Core 6: جلد سوم: ویژگی‌های پیشرفته

ترجمه: مروت گیوی

ناشر: انتشارات ترجمک

صفحه آرایشی: انتشارات ترجمک

طراحی جلد: محمدحسین گیوی

نوبت چاپ: چاپ اول، ۱۴۰۲

قیمت: ۹۵۰۰۰ تومان

چاپ: گروه نشر الکترونیک ترجمک

شابک دوره: ۱-۵۳-۷۸۵۵-۶۲۲-۹۷۸

شابک جلد: ۰-۵۰-۷۸۵۵-۶۲۲-۹۷۸

تلفن تماس: ۰۹۱۸۱۵۰۶۱۰۰

تارنمای اینترنتی: <https://tarjomac.ir>

ISBN : 978-622-7855-53-1



ISBN : 978-622-7855-50-0



## مقدمه

کتاب آموزش حرفه‌ای ASP.NET Core 6 آدام فریمن که اکنون چاپ نهم خود را منتشر کرده است یکی از منابع اصلی و قدرتمند فراگیری توسعه وب و نوشتن اپلیکیشن‌های وب کامل و عملیاتی است.

به منظور تسهیل تهیه کتاب بر حسب سطح برنامه نویسی و نیاز هر فرد، کتاب به چهار جلد براساس بخش‌های کتاب اورجینال تقسیم و منتشر شده است. مخاطبان این کتاب برنامه نویسان و توسعه دهندگان وب است که دانش پایه تولید وبسایت و زبان C# را داشته و می‌خواهند کارهای خود را تقویت و به روز رسانی کنند.

نرم افزار ویزوال بیسیک ۲۰۲۳ و تکنولوژی دات نت ۶ در این کتاب استفاده شده و با پیروی گام به گام از فصول کتاب می‌توان اپلیکیشن وب کاملاً حرفه‌ای و قدرتمند تولید کرد. امید است خوانندگان این مجموعه کتاب چهار جلدی بتوانند وبسایت‌های مورد نظر خود را تولید کنند. ساختار کتاب به صورت زیر است:

- جلد اول: ساخت وبسایت تجارت الکترونیک
- جلد دوم: کار با داده‌ها
- جلد سوم: ویژگی‌های پیشرفته
- جلد چهارم: اجرای پروژه نمونه

## پیشگفتار

برنامه نویسان و طراحان وب حرفه ای با استفاده از راهنمایی های این کتاب پرفروش، که اکنون در نسخه نهم آن است و برای .NET 6 ASP.NET Core به روز شده است، برنامه های نابتتری را برای پلتفرم ASP.NET Core تولید خواهند کرد. کتاب شامل توضیحات مفصلی در مورد پلتفرم ASP.NET Core و اپلیکیشن فریمورک های مورد پشتیبانی آن است.

این راهنمای بنیادی به زمینه چینی برای ASP.NET Core 6 می پردازد و ابزارها و تکنیک های مورد نیاز برای ساختن اپلیکیشن های وب مدرن و قابل توسعه را معرفی می کند. ویژگی ها و قابلیت های جدیدی مانند MVC، Blazor Server، Razor Pages و Blazor WebAssembly همراه با مثال هایی از کاربرد آنها پوشش داده شده است.

ASP.NET Core 6 آخرین تحول پلتفرم وب ASP.NET مایکروسافت است و یک فریمورک «آگونیسست میزبان» و یک مدل برنامه نویسی با بهره وری بالا ارائه می کند که معماری کد تمیزتر، توسعه مبتنی بر تست و بسط پذیری قدرتمند را ترویج می کند.

**آدام فریمن**، نویسنده این کتاب پیشروی بازار، کتاب را به طور کامل بازبینی کرده است و توضیح می دهد که چگونه از ASP.NET Core بیشترین بهره را ببرید. او با مباحث مقدماتی شروع می کند، به شما در مورد کامپوننت های میان افزار، سرویس های سرخود، درخواست پیوند مدل و موارد دیگر آموزش می دهد. با کسب دانش و اعتماد به نفس، او موضوعات پیچیده تر و ویژگی های پیشرفته تر، از جمله مسیریابی نقطه پایان و تزریق وابستگی را معرفی می کند. او آنقدر عمقی پیش می رود تا دانش مورد نیاز را به شما بدهد.

این کتاب از همان قالب و سبک نسخه های محبوب قبلی پیروی می کند، اما همه چیز را برای نسخه جدید ASP.NET Core یعنی برای NET 6 بروز می کند و تمرکز مطلب را بسط می دهد تا تمام پلتفرم ASP.NET Core را در بر بگیرد. یک مطالعه موردی کامل بصورت یک اپلیکیشن وب ASP.NET Core ارائه شده که می توانید به عنوان یک الگو برای پروژه های خود از آن استفاده کنید.

کد منبع این کتاب را می توانید در <https://github.com/Apress/pro-asp.net-core-6> بیابید.

## آنچه خواهید آموخت:

- کاوش کل پلتفرم ASP.NET Core
  - اعمال ویژگی‌های جدید ASP.NET Core 6 به محیط توسعه خود
  - نحوه ایجاد خدمات وب RESTful، اپلیکیشن‌های وب و اپلیکیشن‌های سمت کلاینت
  - راه اندازی سریع و موثر مدل‌های برنامه نویسی جدید بر پایه دانش موجود خود
- آدام فریمن یک متخصص IT با تجربه است که پست مدیر ارشد را در طیف وسیعی از شرکت‌ها داشته است و اخیراً به عنوان مدیر ارشد فناوری و مدیر عامل یک بانک جهانی خدمت کرده است. او که اکنون بازنشسته شده است، وقت خود را صرف نوشتن و دویدن در مسافت‌های طولانی می‌کند.

## مخاطبان این کتاب

مخاطبان این کتاب توسعه دهندگان وب با دانش اولیه توسعه وب و C# است که می‌خواهند آخرین پیشرفت‌ها و عملکردهای ASP.NET Core را در پروژه‌های خود بگنجانند.

## اطلاعات کتابشناختی کتاب اورجینال

**عنوان کتاب:** ASP.NET Core ۶ پرو: ساخت اپلیکیشن‌های وب آماده کلاود با استفاده از صفحات MVC، بلیزور و ریزور

Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages

نویسنده: آدام فریمن (Adam Freeman)

کد DOI: 10.1007/978-1-4842-7957-1

ناشر: انتشارات ای پرس، برکلی، کالیفرنیا

تاریخ انتشار: ۲۵ فوریه ۲۰۲۲

شابک (پرینت): 978-1-4842-7956-4

شابک (ایبوک): 978-1-4842-7957-1

نوبت چاپ: ۹ (9th edition)

تعداد صفحات: XXXIII، ۱۲۵۳

## ساختار کتاب

کتاب ASP.NET Core 6 پرو: ساخت اپلیکیشن های وب آماده کلاود با استفاده از صفحات MVC، بلیزور و ریزور در چهار جلد تهیه شده است.

جلد اول کتاب همان بخش اول کتاب اورجینال می باشد که در ۱۱ فصل به مقدمات کار پرداخته است. در این بخش از پروژه مثال SportsStore استفاده شده است و مقدمات توسعه وب از قبیل آماده سازی محیط توسعه آموزش داده شده است. ابتدا نحوه تهیه و نصب نرم افزارهای ویژوال استودیو و ویژوال استودیو کد آموزش داده شده است و طریقه نصب پکیج های میان افزار و فریمورک ها بحث شده است. سپس با استفاده از پروژه فروشگاه ورزشی به آموزش نحوه تولید صفحات، منو، سید خرید و امنیت و انتقال اپلیکیشن پرداخته شده است.

جلد دوم کتاب با معرفی پلتفرم ASP.NET Core شروع می شود. سپس در ۶ فصل متوالی به آموزش نحوه مسیریابی، تزریق وابستگی، استفاده از ویژگی های خاص پلتفرم و نهایتا کار با داده ها پرداخته شده است.

جلد سوم شامل فصل ۱۸ تا فصل ۳۱ است. در این جلد از کتاب که همان بخش سوم کتاب اورجینال است، ابتدا یک پروژه مثال ساخته می شود، سپس به وب سرویس ها، ویژگی های وب سرویس ها، کار با کنترلرها و نماها، استفاده از صفحات ریزور، کار با کامپوننت نما، تگ هلهپرهای سفارشی و تگ هلهپرهای سرخود، پیوند مدل، اعتبارسنجی مدل، استفاده از فیلتر و ساخت یک اپلیکیشن فرم پرداخته می شود.

جلد چهارم کتاب شامل فصل ۳۲ تا فصل ۳۹ است. در این جلد که بخش چهارم کتاب اورجینال است، ابتدا یک پروژه مثال ساخته می شود، سپس به بحث در مورد بلیزور سرور، بلیزور وب اسمبلی، ویژگی های پیشرفته بلیزور، استفاده از آیدنتیتی برای احراز هویت و تعیین مجوزها و نقش پرداخته می شود.

این کتاب شامل یک پکیج آنلاین است که کلیه کدهای مورد استفاده در کتاب، پروژه ها و رفع خطای کتاب در آن گنجانده شده است و از مخزن گیتهاب ناشر قابل دسترسی است.

آدام فریمن



## محتوای کتاب در یک نگاه

صفحه	فصل
۱	فصل ۱۸- ساخت یک پروژه نمونه
۱۱	فصل ۱۹- ساخت وب سرویس RESTFUL
۵۱	فصل ۲۰- ویژگی های پیشرفته وب سرویس
۸۳	فصل ۲۱- استفاده از کنترلر با نما: بخش ۱
۱۲۶	فصل ۲۲- استفاده از کنترلر با نما: بخش ۲
۱۶۳	فصل ۲۳- استفاده از صفحات ریزور
۲۰۰	فصل ۲۴- استفاده از کامپوننت های نما
۲۳۲	فصل ۲۵- استفاده از تگ هلپرها
۲۷۲	فصل ۲۶- استفاده از تگ هلپرهای سرخود
۳۰۳	فصل ۲۷- استفاده از تگ هلپرهای فرم
۳۴۷	فصل ۲۸- استفاده از پیوند مدل
۳۹۱	فصل ۲۹- استفاده از اعتبارسنجی مدل
۴۳۸	فصل ۳۰- استفاده از فیلترها
۴۹۱	فصل ۳۱- خلق اپلیکیشن فرم

## فهرست مندرجات کتاب

صفحه	عنوان
أ	مقدمه
ب	پیشگفتار
۱	<b>فصل ۱۸- ساخت یک پروژه نمونه</b>
۱	خلق پروژه
۲	افزودن مدل داده
۲	افزودن پکیج های <i>NuGet</i> به پروژه
۳	ایجاد مدل داده
۵	آماده سازی داده های اولیه
۶	پیکربندی سرویس ها و میان افزار <i>Entity Framework Core</i>
۸	خلق و اعمال مهاجرت
۸	اضافه کردن فریمورک <i>CSS</i>
۹	پیکربندی پایپلاین درخواست
۱۰	اجرای اپلیکیشن مثال
۱۰	خلاصه فصل
۱۱	<b>فصل ۱۹- ساخت وب سرویس RESTFUL</b>
۱۲	آماده شدن برای این فصل
۱۳	آشنایی با وب سرویس RESTFUL
۱۳	آشنایی با <i>URL</i> ها و متدهای درخواست
۱۴	آشنایی با <i>JSON</i>
۱۴	آشنایی با بدیل های وب سرویس های RESTFUL
۱۴	خلق وب سرویس با استفاده از <i>API</i> کمینه
۱۷	خلق وب سرویس با استفاده از کنترلر
۱۸	اوج و افول الگوی <i>MVC</i> در <i>ASP.NET CORE</i>
۱۸	فعال کردن فریمورک <i>MVC</i>
۱۹	خلق یک کنترلر

۲۰	آشنایی با کلاس پایه
۲۱	آشنایی با خصوصیات کنترلرها
۲۳	GET و POST: انتخاب گزینه درست
۲۳	آشنایی با نتایج اکشن متد
۲۴	استفاده از تزریق وابستگی در کنترلرها
۲۵	چرخه عمر سرویس زمینه ای ENTITY FRAMEWORK CORE
۲۶	استفاده از پیوند مدل برای دسترسی به داده های مسیر
۲۷	پیوند مدل از تنه درخواست
۲۹	افزودن اکشن های بیشتر
۳۱	بهبود وب سرویس
۳۱	پشتیبانی از درخواست های منبع-مقاطع
۳۲	استفاده از اکشن آسنکرون
۳۳	جلوگیری از پیوند مدل بیش از حد
۳۵	استفاده از نتایج اکشن
۳۹	انجام تغییر مسیر
۴۰	ریدایرکت به یک اکشن متد
۴۱	ریدایرکت کردن با استفاده از مقادیر مسیر
۴۲	اعتبارسنجی داده ها
۴۴	اعمال خصیصه کنترلر API
۴۵	اجتناب از نول شدن ویژگی ها
۴۶	طرح ریزی ویژگی های منتخب
۴۷	پیکربندی سریالی ساز JSON
۵۰	خلاصه فصل
<b>۵۱</b>	<b>فصل ۲۰- ویژگی های پیشرفته وب سرویس</b>
۵۲	آماده شدن برای این فصل
۵۳	حذف دیتابیس
۵۳	اجرای اپلیکیشن مثال
۵۴	رسیدگی به داده های مرتبط
۵۶	قطع مراجع چرخشی در داده های مرتبط
۵۷	پشتیبانی از متد HTTP PATCH
۵۸	آشنایی با JSON Patch
۵۸	نصب و پیکربندی پکیج JSON Patch
۶۰	تعریف اکشن متد

۶۱	..... آشنایی با قالب بندی محتوا
۶۲	..... آشنایی با سیاست محتوای پیش فرض
۶۳	..... آشنایی با مذاکره محتوا
۶۵	..... فعال کردن قالب بندی XML
۶۷	..... احترام کامل به هدر Accept
۶۹	..... تعیین فرمت نتیجه اکشن
۷۰	..... درخواست فرمت در URL
۷۲	..... محدود کردن قالب‌های دریافتی یک اکشن متد
۷۴	..... مستندسازی و کاوش وب سرویس ها
۷۴	..... رفع تناقض اکشن ها
۷۵	..... نصب و پیکربندی پکیج Swashbuckle
۷۸	..... تنظیم دقیق توصیف API
۷۹	..... اجرای API Analyzer
۸۰	..... اعلام نوع نتیجه اکشن متد
۸۲	..... خلاصه فصل

## فصل ۲۱- استفاده از کنترلر با نما: بخش ۱..... ۸۳

۸۴	..... آماده شدن برای این فصل
۸۵	..... حذف دیتابیس
۸۵	..... اجرای اپلیکیشن مثال
۸۶	..... شروع کار با نماها
۸۶	..... پیکربندی اپلیکیشن
۸۷	..... خلق یک کنترلر HTML
۸۸	..... آشنایی با مسیریابی قراردادی
۹۰	..... آشنایی با قواعد نمای ریزور
۹۰	..... خلق نمای ریزور
۹۲	..... تعدیل نمای ریزور
۹۴	..... انتخاب نما بر حسب نام
۹۶	..... استفاده از نماهای مشترک
۹۸	..... مشخص کردن مکان نما
۹۸	..... کار با نماهای ریزور
۹۹	..... مشاهده خروجی کامپایل شده از نماهای ریزور
۱۰۱	..... کش کردن پاسخ ها
۱۰۲	..... تنظیم نوع مدل نما

۱۰۷	..... استفاده از یک فایل <i>View Imports</i>
۱۱۰	..... آشنایی با سینتاکس ریزور
۱۱۰	..... آشنایی با دستورالعمل ها
۱۱۱	..... آشنایی با عبارات محتوا
۱۱۲	..... تنظیم محتوای عنصر
۱۱۲	..... آشنایی با استفاده از اپراتور شرطی NULL در نماها
۱۱۴	..... تنظیم مقادیر خصیصه
۱۱۶	..... استفاده از دستورات شرطی
۱۲۰	..... برشمارش توالی ها
۱۲۳	..... استفاده از بلوک های کد ریزور
۱۲۵	..... خلاصه فصل

## فصل ۲۲ - استفاده از کنترلر با نما: بخش ۲ ..... ۱۲۶

۱۲۶	..... آماده شدن برای این فصل
۱۲۹	..... حذف دیتابیس
۱۲۹	..... اجرای اپلیکیشن مثال
۱۳۰	..... استفاده از کیف نما
۱۳۱	..... چه موقع از کیف نما استفاده کنیم
۱۳۲	..... استفاده از داده موقت
۱۳۴	..... استفاده از خصیصه داده موقت
۱۳۵	..... کار با لی اوت ها
۱۳۷	..... پیکریندی لی اوت ها با استفاده از <i>View Bag</i> کیف نما
۱۳۹	..... شناخت حق تقدم کیف نما
۱۳۹	..... استفاده از فایل <i>View Start</i>
۱۴۱	..... رونویسی لی اوت پیش فرض
۱۴۲	..... انتخاب لی اوت از طریق کدنویسی
۱۴۵	..... استفاده از بخش های لی اوت
۱۴۹	..... استفاده از بخش های اختیاری در لی اوت
۱۵۱	..... تست بخش های چیدمان (لی اوت)
۱۵۳	..... استفاده از نماهای نسبی
۱۵۳	..... فعال کردن نماهای نسبی
۱۵۴	..... ایجاد نمای نسبی
۱۵۴	..... بکارگیری نمای نسبی
۱۵۶	..... انتخاب مدل نمای نسبی با استفاده از یک دستور

۱۵۷	..... استفاده از نماینده تمپلیت
۱۵۸	..... آشنایی با رمزگذاری - محتوا
۱۵۸	..... آشنایی با رمزگذاری HTML
۱۶۰	..... آشنایی با کدگذاری JSON
۱۶۱	..... خلاصه فصل

## فصل ۲۳ - استفاده از صفحات ریزور ..... ۱۶۳

۱۶۴	..... آماده شدن برای این فصل
۱۶۵	..... اجرای اپلیکیشن مثال
۱۶۵	..... آشنایی با صفحات ریزور
۱۶۷	..... ایجاد صفحات ریزور
۱۶۹	..... آشنایی با قواعد قراردادی مسیریابی URL
۱۶۹	..... آشنایی با مدل صفحه
۱۷۰	..... آشنایی با نمای صفحه
۱۷۱	..... آشنایی با کلاس C# تولید شده
۱۷۲	..... آشنایی با مسیریابی صفحات ریزور
۱۷۴	..... آشنایی با رسیدگی پیش فرض به URL
۱۷۵	..... تعیین الگوی مسیریابی در صفحه ریزور
۱۷۷	..... اضافه کردن مسیرها برای صفحه ریزور
۱۷۸	..... آشنایی با کلاس مدل صفحه
۱۷۹	..... استفاده از یک فایل کلاس Code-Behind
۱۸۱	..... افزودن فایل ایمپورت نما
۱۸۲	..... آشنایی با نتیجه اکشن صفحات ریزور
۱۸۴	..... استفاده از نتیجه اکشن
۱۸۷	..... رسیدگی به چندین متد HTTP
۱۸۹	..... آشنایی با ریدایرکت کردن POST
۱۹۰	..... انتخاب متد رسیدگی کننده
۱۹۳	..... آشنایی با نمای صفحه ریزور
۱۹۳	..... ایجاد لی اوت برای صفحات ریزور
۱۹۵	..... استفاده از نماهای نسبی در صفحات ریزور
۱۹۶	..... آشنایی با مسیر جستجوی متد نسبی
۱۹۹	..... خلاصه فصل

## فصل ۲۴ - استفاده از کامپوننت های نما ..... ۲۰۰

۲۰۱	..... آماده شدن برای این فصل
-----	------------------------------

۲۰۳	حذف دیتابیس
۲۰۴	اجرای اپلیکیشن مثال
۲۰۴	آشنایی با کامپوننت های نما
۲۰۵	خلق و استفاده از کامپوننت نما
۲۰۶	اعمال کامپوننت نما
۲۰۹	استفاده از کامپوننت های نما در صفحات ریزور
۲۱۰	آشنایی با نتایج کامپوننت نما
۲۱۰	برگرداندن یک نمای نسبی
۲۱۴	برگرداندن قطعات HTML
۲۱۶	دریافت داده های زمینه
۲۱۸	ارائه زمینه از نمای والد با استفاده از آرگومان ها
۲۲۱	ارائه مقدار با استفاده از هلیپرهای کامپوننت
۲۲۲	استفاده از مقدار پارامتر پیش فرض
۲۲۳	ایجاد کامپوننت نمای آسنکرون
۲۲۵	خلق کلاس های کامپوننت نما
۲۲۸	خلق کلاس کنترلر هیبرید (پیوندی)
۲۳۱	خلاصه فصل

## فصل ۲۵- استفاده از تگ هلیپرها ..... ۲۳۲

۲۳۳	آماده شدن برای این فصل
۲۳۵	حذف دیتابیس
۲۳۵	اجرای اپلیکیشن مثال
۲۳۶	خلق تگ هلیپر
۲۳۶	تعریف کلاس تگ هلیپر
۲۳۷	دریافت داده زمینه
۲۳۸	تولید خروجی
۲۳۹	رجیستر کردن تگ هلیپر
۲۴۰	استفاده از تگ هلیپر
۲۴۲	محدود کردن حوزه یک تگ هلیپر
۲۴۳	بسط حوزه تگ هلیپر
۲۴۵	تعیین ترتیب اجرای تگ هلیپرها
۲۴۶	ویژگی های پیشرفته تگ هلیپر
۲۴۶	خلق کوتاه نویسی عناصر
۲۴۹	ایجاد عناصر به صورت برنامه نویسی

۲۵۰	..... اضافه کردن و الحاق کردن محتوا و عناصر
۲۵۰	..... درج محتوا در اطراف عنصر خروجی
۲۵۲	..... درج محتوا در داخل عنصر خروجی
۲۵۴	..... دریافت داده زمینه نما
۲۵۷	..... کار با دستورات مدل
۲۵۹	..... کار با مدل صفحه
۲۶۲	..... هماهنگی بین تگ هلیرها
۲۶۴	..... سرکوب عنصر خروجی
۲۶۶	..... استفاده از کامپوننت های تگ هلیپر
۲۶۷	..... ایجاد کامپوننت راهنمای برجسب
۲۶۹	..... گسترش انتخاب عنصر کامپوننت تگ هلیپر
۲۷۱	..... خلاصه فصل
<b>۲۷۲</b>	<b>..... فصل ۲۶- استفاده از تگ هلیرهای سرخود</b>
۲۷۳	..... آماده شدن برای این فصل
۲۷۴	..... اضافه کردن یک فایل تصویر
۲۷۵	..... نصب پکیج سمت کلاینت
۲۷۵	..... حذف دیتابیس
۲۷۵	..... اجرای اپلیکیشن مثال
۲۷۶	..... فعال کردن تگ هلیرهای سرخود
۲۷۶	..... تبدیل عناصر لنگر
۲۷۹	..... استفاده از عناصر Anchor برای صفحات ریزور
۲۸۰	..... تولید URLها (و نه لینک ها)
۲۸۱	..... استفاده از تگ هلیپر جاوا اسکریپت و CSS
۲۸۱	..... مدیریت فایل های جاوا اسکریپت
۲۸۱	..... انتخاب فایل های جاوا اسکریپت
۲۸۳	..... آشنایی با نقشه های منبع
۲۸۴	..... محدود کردن الگوی گلوبینگ
۲۸۶	..... آشنایی با تقویت کش
۲۸۸	..... کار با شبکه های تحویل محتوا
۲۹۰	..... مدیریت استایل شیت های CSS
۲۹۱	..... انتخاب استایل شیت
۲۹۲	..... کار با شبکه های تحویل محتوا
۲۹۴	..... کار با عناصر تصویر



۲۹۵	..... استفاده از کش داده
۲۹۷	..... استفاده از کش توزیع شده برای محتوا
۲۹۸	..... تنظیم انقضای کش
۲۹۸	..... تنظیم یک نقطه انقضای ثابت
۲۹۹	..... تنظیم یک دوره انقضای آخرین استفاده
۳۰۰	..... استفاده از گونه های مختلف کش کردن
۳۰۱	..... استفاده از تگ هلپر محیط میزبانی
۳۰۲	..... خلاصه فصل
۳۰۳	..... فصل ۲۷- استفاده از تگ هلپرهای فرم
۳۰۴	..... آماده شدن برای این فصل
۳۰۵	..... حذف دیتابیس
۳۰۵	..... اجرای اپلیکیشن مثال فصل
۳۰۶	..... آشنایی با الگوی رسیدگی به فرم
۳۰۶	..... خلق یک کنترلر برای رسیدگی به فرم ها
۳۰۹	..... خلق یک صفحه ریزور برای رسیدگی به فرم ها
۳۱۲	..... استفاده از تگ هلپرهای برای بهبود فرم های HTML
۳۱۲	..... کار با عناصر فرم
۳۱۲	..... ست کردن هدف فرم
۳۱۴	..... تبدیل دکمه های فرم
۳۱۶	..... کار با عناصر ورودی
۳۱۷	..... انتخاب ویژگی های مدل در صفحات ریزور
۳۱۸	..... تبدیل خصیصه نوع عنصر ورودی
۳۲۰	..... قالب بندی مقادیر عناصر ورودی
۳۲۲	..... اعمال قالب بندی از طریق کلاس مدل
۳۲۴	..... نمایش مقادیر داده های مرتبط در عناصر ورودی
۳۲۹	..... کار با عنصر برچسب (لیبل)
۳۳۱	..... کار با عناصر SELECT و OPTION
۳۳۳	..... تکمیل عنصر select
۳۳۶	..... کار با محدوده متنی (TEXTAREA)
۳۳۸	..... استفاده از ویژگی ضد جعل
۳۴۰	..... فعال کردن ویژگی ضد جعل در یک کنترلر
۳۴۱	..... فعال کردن ویژگی ضد جعل در صفحه ریزور
۳۴۳	..... استفاده از توکن های ضد جعل با کلاینت های جاوا اسکریپت

۳۴۶ ..... خلاصه فصل

## فصل ۲۸ - استفاده از پیوند مدل ..... ۳۴۷

۳۴۸ ..... آماده شدن برای این فصل

۳۴۹ ..... حذف دیتابیس

۳۴۹ ..... اجرای اپلیکیشن مثال فصل

۳۵۰ ..... آشنایی با پیوند مدل

۳۵۲ ..... پیوند دادن انواع داده های ساده

۳۵۳ ..... پیوند انواع داده ساده در صفحات ریزور

۳۵۵ ..... آشنایی با مقادیر پیش فرض پیوند

۳۵۸ ..... پیوند انواع پیچیده

۳۶۰ ..... پیوند به یک ویژگی

۳۶۲ ..... اتصال انواع پیچیده تو در تو

۳۶۴ ..... تعیین پیشوندهای سفارشی برای انواع پیچیده تو در تو

۳۶۷ ..... ویژگی های دارای پیوند انتخابی

۳۶۹ ..... پیوند انتخابی در کلاس مدل

۳۷۰ ..... پیوند به آرایه ها و مجموعه ها

۳۷۰ ..... پیوند به آرایه ها

۳۷۳ ..... تعیین شماره ایندکس مقادیر آرایه

۳۷۶ ..... پیوند مدل به دیکشنری

۳۷۸ ..... پیوند به مجموعه های انواع پیچیده

۳۸۰ ..... تعیین منبع پیوند مدل

۳۸۳ ..... انتخاب منبع پیوند برای یک ویژگی

۳۸۴ ..... استفاده از هدرها برای پیوند مدل

۳۸۶ ..... استفاده از تنه درخواست به عنوان منابع پیوند

۳۸۸ ..... پیوند مدل دستی

۳۹۰ ..... خلاصه فصل

## فصل ۲۹ - استفاده از اعتبارسنجی مدل ..... ۳۹۱

۳۹۲ ..... آماده شدن برای این فصل

۳۹۴ ..... حذف دیتابیس

۳۹۴ ..... اجرای اپلیکیشن مثال فصل

۳۹۵ ..... آشنایی با لزوم اعتبارسنجی مدل

۳۹۵ ..... اعتبارسنجی کردن داده ها

۴۰۰ ..... نمایش پیام های اعتبارسنجی

۴۰۱	.....	آشنایی با بررسی های اعتبارسنجی ضمنی
۴۰۳	.....	اجرای اعتبارسنجی صریح
۴۰۶	.....	بیکربندی پیام های خطای اعتبارسنجی پیش فرض
۴۰۹	.....	نمایش پیام های اعتبار سنجی سطح-ویژگی
۴۱۱	.....	نمایش پیام های سطح-مدل
۴۱۳	.....	اعتبارسنجی صریح داده در صفحه ریزور
۴۱۷	.....	تعیین قوانین اعتبارسنجی با استفاده از فراداده
۴۲۰	.....	اقدامات اعتبارسنجی
۴۲۱	.....	آشنایی با اعتبارسنجی کنترلر وب سرویس
۴۲۱	.....	خلق خصیصه اعتبارسنجی ویژگی سفارشی
۴۲۲	.....	اعتبارسنجی مجدد داده ها
۴۲۲	.....	خلق یک خصیصه سفارشی اعتبارسنجی مدل
۴۲۵	.....	استفاده از ویژگی اعتبارسنجی سفارشی مدل در صفحه ریزور
۴۲۸	.....	اجرای اعتبارسنجی سمت کلاینت
۴۳۰	.....	اجتناب از تناقض با مرورگر
۴۳۱	.....	بسط اعتبارسنجی سمت کلاینت
۴۳۱	.....	انجام اعتبارسنجی ریموت
۴۳۵	.....	انجام اعتبارسنجی ریموت در صفحات ریزور
۴۳۷	.....	خلاصه فصل

### فصل ۳۰- استفاده از فیلترها..... ۴۳۸

۴۳۹	.....	آماده شدن برای این فصل
۴۴۱	.....	فعال کردن اتصالات HTTPS
۴۴۳	.....	حذف دیتابیس
۴۴۳	.....	اجرای اپلیکیشن مثال فصل
۴۴۴	.....	استفاده از فیلترها
۴۴۷	.....	استفاده از فیلترها در صفحات ریزور
۴۴۹	.....	آشنایی با فیلترها
۴۵۱	.....	ایجاد فیلترهای سفارشی
۴۵۲	.....	آشنایی با فیلترهای تعیین مجوز
۴۵۲	.....	خلق یک فیلتر مجوزدهی
۴۵۴	.....	آشنایی با فیلترهای منابع
۴۵۶	.....	خلق فیلتر منبع
۴۵۶	.....	فیلترها و تزریق وابستگی

۴۵۱	..... خلق فیلتر منبع آسنکرون
۴۵۹	..... آشنایی با فیلترهای اکشن
۴۶۱	..... ایجاد فیلتر اکشن
۴۶۲	..... پیاده سازی فیلتر اکشن با استفاده از کلاس پایه خصیصه
۴۶۳	..... استفاده از متدهای فیلتر کنترلر
۴۶۴	..... آشنایی با فیلترهای صفحه
۴۶۶	..... خلق فیلتر صفحه
۴۶۸	..... استفاده از متدهای فیلتر مدل صفحه
۴۶۹	..... آشنایی با فیلترهای نتیجه
۴۷۰	..... آشنایی با فیلترهای نتیجه همیشه-اجرا
۴۷۱	..... خلق فیلتر نتیجه
۴۷۳	..... پیاده سازی فیلتر نتیجه با استفاده از کلاس پایه Attribute
۴۷۴	..... آشنایی با فیلترهای استثنا
۴۷۵	..... خلق یک فیلتر استثناء
۴۷۸	..... مدیریت چرخه حیات فیلتر
۴۸۰	..... ایجاد کارخانه های فیلتر
۴۸۲	..... استفاده از حوزه های تزریق وابستگی برای مدیریت چرخه عمر فیلتر
۴۸۳	..... استفاده از فیلترها به عنوان سرویس بدون واسط IFILTERFACTORY
۴۸۳	..... ایجاد فیلترهای گلوبال
۴۸۵	..... آشنایی با ترتیب فیلترها و تغییر آن
۴۸۸	..... تغییر ترتیب فیلترها
۴۹۰	..... خلاصه فصل
<b>۴۹۱</b>	<b>..... فصل ۳۱- خلق اپلیکیشن فرم</b>
۴۹۱	..... آماده شدن برای این فصل
۴۹۴	..... حذف دیتابیس
۴۹۴	..... اجرای اپلیکیشن مثال فصل
۴۹۵	..... خلق یک اپلیکیشن MVC FORMS
۴۹۵	..... آماده سازی مدل نما و نما
۴۹۸	..... خواندن داده ها
۵۰۰	..... خلق داده
۵۰۴	..... ویرایش داده ها
۵۰۷	..... حذف داده ها
۵۰۹	..... خلق اپلیکیشن فرم های صفحات ریزور

۵۱۱	.....	خلق کارکرد مشترک
۵۱۱	.....	استفاده از چندین صفحه
۵۱۴	.....	تعریف صفحات برای عملیات <i>CRUD</i>
۵۱۷	.....	خلق اشیاء داده مرتبط جدید
۵۱۷	.....	ارائه داده های مرتبط در همان درخواست
۵۲۲	.....	تلاش برای ایجاد داده های جدید
۵۲۷	.....	خلاصه فصل



## ساخت یک پروژه نمونه

در این فصل، شما یک پروژه نمونه برای این بخش از کتاب ایجاد خواهید کرد. این پروژه شامل یک مدل داده ساده، یک پکیج سمت کلاینت برای قالب‌بندی محتوای HTML و یک پایپلاین درخواست ساده است.

### خلق پروژه

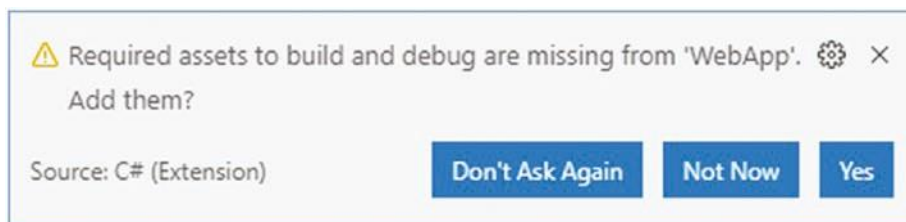
خط فرمان PowerShell را باز کنید و فرامین ذکر شده در لیست ۱۸-۱ را اجرا کنید.

■ نکته: می‌توانید پروژه مثال این فصل (و سایر فصول کتاب) را از <https://github.com/apress/pro-asp.net-core> دانلود کنید. برای نحوه دریافت کمک در صورت مشکل در اجرای مثال‌ها به فصل ۱ مراجعه کنید.

### لیست ۱۸-۱ خلق پروژه

```
dotnet new globaljson --sdk-version 6.0.100 --output WebApp
dotnet new web --no-https --output WebApp --framework net6.0
dotnet new sln -o WebApp
dotnet sln WebApp add WebApp
```

اگر از ویژوال استودیو استفاده می‌کنید، فایل WebApp.sln را در پوشه WebApp باز کنید. اگر از ویژوال استودیو کد استفاده می‌کنید، پوشه WebApp را باز کنید. همانطور که در شکل ۱۸-۱ نشان داده شده است، هنگامی که از شما خواسته شد تا دارایی‌های مورد نیاز برای ساخت و اشکال‌زدایی پروژه را اضافه کنید، روی دکمه Yes کلیک کنید.



شکل ۱۸-۱ افزودن دارایی‌های پروژه

فایل launchSettings.json را از پوشه WebApp/Properties باز کنید، پورت HTTP را تغییر دهید و راه‌اندازی خودکار مرورگر را غیرفعال کنید، همانطور که در لیست ۱۸-۲ نشان داده شده است.

## لیست ۱۸-۲ تنظیم پورت HTTP در فایل launchSettings.json از پوشه Properties

```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:5000",
      "sslPort": 0
    }
  },
  "profiles": {
    "WebApp": {
      "commandName": "Project",
      "dotnetRunMessages": true,
      "launchBrowser": false,
      "applicationUrl": "http://localhost:5000",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

## افزودن مدل داده

مدل داده به ما روش‌های مختلفی را نشان می‌دهد که اپلیکیشن وب را می‌توان با استفاده از ASP.NET Core ساخت و اینکه چگونه پاسخ‌های پیچیده را می‌توان ترکیب کرد و چگونه داده‌ها را می‌توان توسط کاربر ارسال کرد. در بخش‌های بعدی، یک مدل داده ساده ایجاد می‌کنم و از آن برای ایجاد طرح دیتابیس (دیتابیس) استفاده می‌کنم که برای ذخیره داده‌های برنامه استفاده می‌شود.

## افزودن پکیج‌های NuGet به پروژه

مدل داده از Entity Framework Core برای ذخیره و کوئری داده‌ها در دیتابیس SQL Server LocalDB استفاده می‌کند. برای افزودن پکیج‌های NuGet برای Entity Framework Core، از یک خط فرمان PowerShell برای اجرای فرامین ذکر شده در لیست ۱۸-۳ در پوشه پروژه WebApp استفاده کنید.



### لیست ۱۸-۳ اضافه کردن پکیج‌ها به پروژه

```
dotnet add package Microsoft.EntityFrameworkCore.Design --version 6.0.0
dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 6.0.0
```

اگر از ویژوال استودیو استفاده می‌کنید، می‌توانید بسته‌ها را با انتخاب **Project ► Manage NuGet Packages** اضافه کنید. دقت کنید که نسخه صحیح بسته‌ها را برای افزودن به پروژه انتخاب کنید.

اگر از مثال‌های فصل‌های قبلی پیروی نکرده‌اید، باید پکیج ابزار گلوبال را که برای ایجاد و مدیریت مهاجرت‌های Entity Framework Core استفاده می‌شود، نصب کنید. دستورات نشان داده شده در لیست ۱۸-۴ را اجرا کنید تا هر نسخه موجود از پکیج را حذف کنید و نسخه مورد نیاز این کتاب را نصب کنید (اگر این نسخه از پکیج ابزار را در فصل‌های قبلی نصب کرده‌اید، می‌توانید از این دستورات صرف نظر کنید).

### لیست ۱۸-۴ نصب پکیج ابزار گلوبال

```
dotnet tool uninstall --global dotnet-ef
dotnet tool install --global dotnet-ef --version 6.0.0
```

## ایجاد مدل داده

مدل داده این بخش از کتاب شامل سه کلاس به هم مربوط است: `Product`، `Supplier` و `Category`. یک پوشه جدید به نام `Models` ایجاد کنید و یک فایل کلاس به نام `Category.cs` با محتوای نشان داده شده در لیست ۱۸-۵ به آن اضافه کنید.

### لیست ۱۸-۵ محتوای فایل `Category.cs` از پوشه `Models`

```
namespace WebApp.Models
{
    public class Category
    {
        public long CategoryId { get; set; }
        public string Name { get; set; } = string.Empty;

        public IEnumerable<Product>? Products { get; set; }
    }
}
```

یک کلاس به نام `Supplier.cs` را به پوشه `Models` اضافه کنید و از آن برای تعریف کلاس نشان داده شده در لیست ۱۸-۶ استفاده کنید.

## لیست ۶-۱۸ محتوای فایل Supplier.cs از پوشه Models

```
namespace WebApp.Models
{
    public class Supplier
    {

        public long SupplierId { get; set; }
        public string Name { get; set; } = string.Empty;
        public string City { get; set; } = string.Empty;

        public IEnumerable<Product>? Products { get; set; }
    }
}
```

بعد، یک کلاس به نام Product.cs را به پوشه Models اضافه کنید و از آن برای تعریف کلاس نشان داده شده در لیست ۷-۱۸ استفاده کنید.

## لیست ۷-۱۸ محتوای فایل Product.cs از پوشه Models

```
using System.ComponentModel.DataAnnotations.Schema;

namespace WebApp.Models
{
    public class Product
    {

        public long ProductId { get; set; }

        public string Name { get; set; } = string.Empty;
        [Column(TypeName = "decimal(8, 2)")]
        public decimal Price { get; set; }

        public long CategoryId { get; set; }
        public Category? Category { get; set; }

        public long SupplierId { get; set; }
        public Supplier? Supplier { get; set; }
    }
}
```

هر یک از سه کلاس مدل داده یک ویژگی کلیدی را تعریف می‌کند که مقدار آن زمان ذخیره اشیای جدید توسط دیتابیس تخصیص می‌یابد. همچنین ویژگی‌های ناوبری وجود دارد که برای کوئری داده‌های مرتبط استفاده می‌شود، به طوری که به عنوان مثال، امکان کوئری برای همه محصولات در یک دسته خاص وجود خواهد داشت. ویژگی Price با خصیصه Column تزئین شده است که دقت مقادیر ذخیره شده در دیتابیس را مشخص می‌کند. نداشت یک به یک بین انواع عددی C# و SQL وجود ندارد و ویژگی Column به Entity Framework Core می‌گوید که کدام نوع SQL باید در دیتابیس برای ذخیره مقادیر Price استفاده شود. در این مورد، نوع decimal(8, 2) در مجموع هشت رقم، از جمله دو رقم پس از نقطه اعشار را مجاز می‌کند.

برای ایجاد کلاس زمینه Entity Framework Core که دسترسی به دیتابیس را فراهم می کند، فایل به نام DataContext.cs را به پوشه Models اضافه کنید و کد نشان داده شده در لیست ۸-۱۸ را اضافه کنید.

### لیست ۸-۱۸ محتوای فایل DataContext.cs از پوشه Models

```
using Microsoft.EntityFrameworkCore;

namespace WebApp.Models
{
    public class DataContext : DbContext
    {

        public DataContext(DbContextOptions<DataContext> opts)
            : base(opts) { }

        public DbSet<Product> Products => Set<Product>();
        public DbSet<Category> Categories => Set<Category>();
        public DbSet<Supplier> Suppliers => Set<Supplier>();
    }
}
```

کلاس context ویژگی هایی را تعریف می کند که برای کوئری از دیتابیس برای داده های Product، Category و Supplier استفاده می شود.

## آماده سازی داده های اولیه

یک کلاس به نام SeedData.cs را به پوشه Models اضافه کنید و کد نشان داده شده در لیست ۹-۱۸ را اضافه کنید تا داده های اولیه را که برای پر کردن دیتابیس استفاده می شود، تعریف کنید.

### لیست ۹-۱۸ محتوای فایل SeedData.cs از پوشه Models

```
using Microsoft.EntityFrameworkCore;

namespace WebApp.Models
{
    public static class SeedData
    {

        public static void SeedDatabase(DataContext context)
        {
            context.Database.Migrate();
            if (context.Products.Count() == 0 && context.Suppliers.Count() == 0
                && context.Categories.Count() == 0)
            {

                Supplier s1 = new Supplier
                { Name = "Splash Dudes", City = "San Jose" }; Supplier s2 = new
Supplier
```

```

        { Name = "Soccer Town", City = "Chicago" }; Supplier s3 = new
Supplier
        { Name = "Chess Co", City = "New York" };

        Category c1 = new Category { Name = "Watersports" };
        Category c2 = new Category { Name = "Soccer" };
        Category c3 = new Category { Name = "Chess" };

        context.Products.AddRange(
        new Product { Name = "Kayak", Price = 275, Category = c1,
Supplier = s1 },
        new Product { Name = "Lifejacket", Price = 48.95m, Category =
c1, Supplier = s1 },
        new Product { Name = "Soccer Ball", Price = 19.50m, Category =
c2, Supplier = s2 },
        new Product { Name = "Corner Flags", Price = 34.95m, Category =
c2, Supplier = s2 },
        new Product { Name = "Stadium", Price = 79500, Category = c2,
Supplier = s2 },
        new Product { Name = "Thinking Cap", Price = 16, Category = c3,
Supplier = s3 },
        new Product { Name = "Unsteady Chair", Price = 29.95m, Category
= c3, Supplier = s3 },
        new Product { Name = "Human Chess Board", Price = 75, Category =
c3, Supplier = s3 },
        new Product { Name = "Bling-Bling King", Price = 1200, Category
= c3, Supplier = s3 }
        );
        context.SaveChanges();
    }
}
}
}

```

متد استاتیک SeedDatabase تضمین می کند که همه مهاجرت های معلق به دیتابیس اعمال شده باشد.

اگر دیتابیس خالی باشد، با دسته ها، تامین کنندگان و محصولات نمونه (دانه پاشی) پر می شود. Entity Framework Core از نداشت اشیاء در جداول دیتابیس مراقبت می کند و هنگامی که داده ها ذخیره می شوند، ویژگی های کلیدی به طور خودکار اختصاص داده می شود.

## بیکربندی سرویس ها و میان افزار Entity Framework Core

تغییرات نشان داده شده در لیست ۱۸-۱۰ را در فایل Program.cs اعمال کنید، که Entity Framework Core را بیکربندی می کند و سرویس های DataContext را تنظیم می کند که در این قسمت از کتاب برای دسترسی به دیتابیس استفاده می شود.

**لیست ۱۸-۱۰** آماده سازی سرویس ها و میان افزارها در فایل Program.cs از پوشه WebApp

```
using Microsoft.EntityFrameworkCore;
using WebApp.Models;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDbContext<DataContext>(opts =>
{
    opts.UseSqlServer(builder.Configuration[
        "ConnectionStrings:ProductConnection"]);
    opts.EnableSensitiveDataLogging(true);
});

var app = builder.Build();
app.MapGet("/", () => "Hello World!");
var context = app.Services.CreateScope().ServiceProvider
    .GetRequiredService<DataContext>();
SeedData.SeedDatabase(context);

app.Run();
```

سرویس DataContext حوزه دار<sup>۱</sup> است، به این معنی که برای دریافت سرویس مورد نیاز متد SeedDatabase باید یک حوزه ایجاد کنم.

برای تعریف رشته اتصال که برای داده های اپلیکیشن استفاده می شود، تنظیمات پیکربندی نشان داده شده در لیست ۱۸-۱۱ را در فایل appsettings.json اضافه کنید. رشته اتصال باید در یک خط وارد شود.

**لیست ۱۸-۱۱** تعریف رشته اتصال در فایل appsettings.json از پوشه WebApp

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning",
      "Microsoft.EntityFrameworkCore": "Information"
    }
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "ProductConnection":
      "Server=(localdb)\\MSSQLLocalDB;Database=Products;MultipleActive
      ResultSets=True"
  }
}
```

---

<sup>1</sup> Scoped

علاوه بر رشته اتصال، لیست ۱۸-۱۱ جزئیات ورود به سیستم را برای Entity Framework Core تنظیم می‌کند تا کوئری های SQL ارسال شده به دیتابیس وارد شوند.

## خلق و اعمال مهاجرت

برای ایجاد مهاجرتی که طرحواره دیتابیس<sup>۱</sup> را تنظیم می‌کند، از یک خط فرمان PowerShell برای اجرای فرامین ذکر شده در لیست ۱۸-۱۲ در پوشه پروژه WebApp استفاده کنید.

### لیست ۱۸-۱۲ ایجاد یک مهاجرت در Entity Framework Core

```
dotnet ef migrations add Initial
```

پس از ایجاد مهاجرت، با استفاده از فرمان ذکر شده در فهرست ۱۸-۱۳، آن را به دیتابیس اعمال کنید.

### لیست ۱۸-۱۳ اعمال مهاجرت به دیتابیس

```
dotnet ef database update
```

پیام‌های گزارش‌گیری (لاگ) که توسط اپلیکیشن نمایش داده می‌شوند، دستورات SQL را که به دیتابیس ارسال می‌شوند نشان می‌دهند.

■ **توجه:** اگر به ریست دیتابیس نیاز دارید، دستور `dotnet ef --force drop` و بعد فرمان موجود در لیست ۱۸-۱۳ را اجرا کنید.

## اضافه کردن فریمورک CSS

فصل‌های بعدی راه‌های متفاوتی را که پاسخ‌های HTML را می‌توان تولید کرد، نشان خواهد داد. دستورات نشان داده شده در لیست ۱۸-۱۴ را اجرا کنید تا هر نسخه موجود از LibMan را حذف کنید و نسخه استفاده شده در این کتاب را نصب کنید (اگر این نسخه از LibMan را در فصل‌های قبلی نصب کرده اید، می‌توانید از این دستورات صرف نظر کنید).

### لیست ۱۸-۱۴ نصب پکیج ابزار LibMan

```
dotnet tool uninstall --global Microsoft.Web.LibraryManager.Cli  
dotnet tool install --global Microsoft.Web.LibraryManager.Cli --version 2.1.113
```

برای افزودن فریم ورک Bootstrap CSS به طوری که بتوان پاسخ‌های HTML را استایل‌بندی کرد، دستورات نشان‌داده‌شده در لیست ۱۸-۱۵ را در پوشه پروژه WebApp اجرا کنید.

<sup>1</sup> Database Schema

## لیست ۱۵-۱۸ نصب فریم ورک Bootstrap CSS

```
libman init -p cdnjs
libman install bootstrap@5.1.3 -d wwwroot/lib/bootstrap
```

## پیکربندی پایپلاین درخواست

برای تعریف یک کامپوننت میان‌افزار ساده که برای اطمینان از راه‌اندازی صحیح پروژه نمونه استفاده می‌شود، یک فایل کلاس به نام TestMiddleware.cs را به پوشه WebApp اضافه کنید و کد ذکر شده در لیست ۱۶-۱۸ را اضافه کنید.

### لیست ۱۶-۱۸ محتوای فایل TestMiddleware.cs از پوشه WebApp

```
using WebApp.Models;

namespace WebApp
{
    public class TestMiddleware
    {
        private RequestDelegate nextDelegate;

        public TestMiddleware(RequestDelegate next)
        {
            nextDelegate = next;
        }

        public async Task Invoke(HttpContext context, DataContext dataContext)
        {
            if (context.Request.Path == "/test")
            {
                await context.Response.WriteAsync(
                    $"There are {dataContext.Products.Count()} products\n");
                await context.Response.WriteAsync(
                    $"There are {dataContext.Categories.Count()} categories\n");
                await context.Response.WriteAsync(
                    $"There are {dataContext.Suppliers.Count()} suppliers\n");
            }
            else
            {
                await nextDelegate(context);
            }
        }
    }
}
```

همانطور که در لیست ۱۷-۱۸ نشان داده شده است، کامپوننت میان‌افزار را به پایپلاین درخواست اضافه کنید.

**لیست ۱۷-۱۸** افزودن یک کامپوننت میان افزار به فایل Program.cs از پوشه WebApp

```
using Microsoft.EntityFrameworkCore;
using WebApp.Models;

var builder = WebApplication.CreateBuilder(args);
builder.Services.AddDbContext<DataContext>(opts =>
{
    opts.UseSqlServer(builder.Configuration[
        "ConnectionStrings:ProductConnection"]);
    opts.EnableSensitiveDataLogging(true);
});

var app = builder.Build(); app.UseMiddleware<WebApp.TestMiddleware>();
app.MapGet("/", () => "Hello World!");
var context = app.Services.CreateScope().ServiceProvider
    .GetRequiredService<DataContext>();
SeedData.SeedDatabase(context);

app.Run();
```

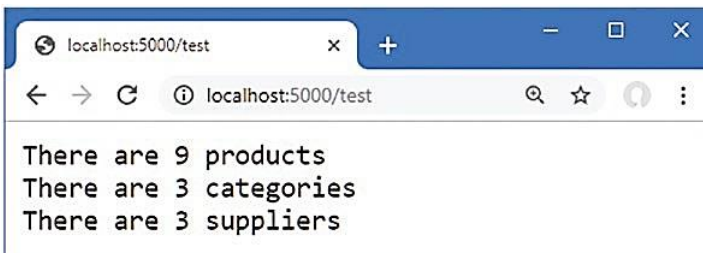
## اجرای اپلیکیشن مثال

اپلیکیشن را با اجرای دستور نشان داده شده در لیست ۱۸-۱۸ در پوشه پروژه WebApp شروع کنید.

### لیست ۱۸-۱۸ اجرای اپلیکیشن مثال

```
dotnet run
```

از یک برگه مرورگر جدید استفاده کنید و <http://localhost:5000/test> را درخواست کنید، پاسخ نشان داده شده در شکل ۱۸-۲ را خواهید دید.



شکل ۱۸-۲ اجرای اپلیکیشن مثال

## خلاصه فصل

در این فصل، من اپلیکیشن مثال را ایجاد کردم که در سراسر این بخش از کتاب استفاده می شود. این پروژه با قالب Empty ایجاد شده است، شامل یک مدل داده است که به Entity Framework Core متکی است و با یک پایپلاین درخواست که شامل یک کامپوننت میان افزار آزمایشی ساده است، پیکربندی شده است. در فصل بعدی، نحوه ایجاد وب سرویس با استفاده از ASP.NET Core را به شما نشان می دهم.