

آدام فریمن

آموزش حرفه ای

ASP.NET Core 6



جلد اول:

ساخت وبسایت تجارت الکترونیک

مروّت گیوی

ترجمه  
Parjomac

آدام فریمن

**آموزش حرفه‌ای 6 ASP.NET Core**

**جلد اول: ساخت وبسایت تجارت الکترونیک**

ترجمه

مروّت گیوی

انتشارات ترجمک

تابستان ۱۴۰۲

# آموزش حرفه‌ای 6 ASP.NET CORE

## 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 ۱: ساخت وبسایت تجارت الکترونیک
شابک	: دوره: ۱-۵۳-۷۸۵۵-۶۲۲-۹۷۸؛ جلد ۳: ۴-۵۲-۷۸۵۵-۶۲۲-۹۷۸
موضوع	: مایکروسافت دات‌نت فریم‌ورک، Microsoft .NET Framework، وبگاه‌ها - برنامه‌های تالیفی، Web sites -- Authoring programs، علوم کامپیوتر، Computer science، نرم‌افزار - مهندسی، Software engineering
مندرجات	: جلد ۲: کار با داده، ۳: ویژگی‌های پیشرفته 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 اول: ساخت وبسایت تجارت الکترونیک

ترجمه: مروت گیوی

ناشر: انتشارات ترجمک

صفحه آرایشی: انتشارات ترجمک

طراحی جلد: محمدحسین گیوی

نوبت چاپ: چاپ اول، ۱۴۰۲

قیمت: ۷۵۰۰۰ تومان

چاپ: گروه نشر الکترونیک ترجمک

شابک دوره: ۱-۵۳-۷۸۵۵-۶۲۲-۹۷۸

شابک جلد: ۴-۵۲-۷۸۵۵-۶۲۲-۹۷۸

تلفن تماس: ۰۹۱۸۱۵۰۶۱۰۰

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

ISBN : 978-622-7855-53-1



ISBN : 978-622-7855-52-4





## 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 نابتری 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 «آگونیزست میزبان» و یک مدل برنامه نویسی با بهره‌وری بالا ارائه می‌کند که معماری کد تمیزتر، توسعه مبتنی بر تست و بسط پذیری قدرتمند را ترویج می‌کند.

**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.

## آنچه خواهید آموخت:

- کاوش کل پلتفرم ASP.NET Core
- اعمال ویژگی های جدید ASP.NET Core 6 به محیط توسعه خود
- نحوه ایجاد خدمات وب RESTful، اپلیکیشن های وب و اپلیکیشن های سمت کلاینت
- راه اندازی سریع و موثر مدل های برنامه نویسی جدید بر پایه دانش موجود خود

مخاطبان این کتاب توسعه دهندگان وب با دانش اولیه توسعه وب و C# است که می خواهند آخرین پیشرفت ها و عملکردهای ASP.NET Core را در پروژه های خود بگنجانند.

آدام فریمن یک متخصص IT با تجربه است که پست مدیر ارشد را در طیف وسیعی از شرکت ها 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 مسافت های طولانی می کند.

آدام فریمن



## محتوای کتاب در یک نگاه

صفحه	فصل
۱۴	فصل ۱- آشنایی با مجموعه ASP.NET Core
۲۴	فصل ۲- شروع کار با ASP.NET Core
۴۵	فصل ۳- اولین اپلیکیشن ASP.NET Core شما
۷۵	فصل ۴- استفاده از ابزارهای برنامه نویسی
۹۲	فصل ۵- ویژگی های ضروری C#
۱۴۲	فصل ۶- تست اپلیکیشن ASP.NET Core
۱۶۳	فصل ۷- SportsStore: یک اپلیکیشن واقعی
۲۰۳	فصل ۸- SportsStore: منو و سبد خرید
۲۴۱	فصل ۹- SportsStore: تکمیل سبد خرید
۲۷۰	فصل ۱۰- SportsStore: مدیریت
۲۹۴	فصل ۱۱- SportsStore: امنیت و انتقال
Error! Bookmark not defined.	واژه نامه

## فهرست مندرجات کتاب

صفحه	عنوان
.....	مقدمه
.....	پیشگفتار
.....	<b>فصل ۱- آشنایی با مجموعه ASP.NET CORE</b>
.....	آشنایی با .NET CORE ، .NET FRAMEWORK و .NET
.....	آشنایی با فریمورک های اپلیکیشن
.....	آشنایی با فریمورک MVC
.....	چیدن الگوها در جای خود
.....	آشنایی با صفحات ریزور
.....	آشنایی با بلیزور
.....	آشنایی با فریمورک های کاربردی
.....	موضوعاتی برای نسخه های آینده
.....	آشنایی با پلتفرم ASP.NET CORE
.....	آشنایی با این کتاب
.....	برای اجرای مثال های کتاب به چه نرم افزاری نیاز دارم؟
.....	برای دنبال کردن مثال ها به چه پلتفرمی نیاز دارم؟
.....	اگر در اجرای مثال ها مشکل داشته باشم، چکار کنم؟
.....	اگر اشتباهی در کتاب پیدا کنم، چه می شود؟
.....	جایزه گزارش خطا
.....	این کتاب چه چیزی را پوشش می دهد؟
.....	بخش ۱: معرفی ASP.NET Core
.....	بخش ۲: پلتفرم ASP.NET Core
.....	بخش ۳: اپلیکیشن های ASP.NET Core
.....	بخش ۴: ویژگی های پیشرفته ASP.NET Core
.....	چه چیزی در این کتاب پوشش داده نمی شود؟
.....	چگونه با نویسنده تماس بگیرم؟
.....	اگر واقعا از این کتاب لذت بردم، چطور؟

۲۳	..... اگر این کتاب من را عصبانی کرده باشد و من بخواهم شکایت کنم چه می شود؟
۲۳	..... خلاصه فصل

## فصل ۲- شروع کار با ASP.NET CORE..... ۲۴

۲۴	..... بروزرسانی این کتاب
۲۴	..... انتخاب ویرایشگر کد
۲۵	..... نصب ویژوال استودیو
۲۷	..... نصب .NET SDK
۲۸	..... نصب ویژوال استودیو کد
۲۸	..... نصب .NET SDK
۲۹	..... نصب SQL Server LocalDB
۳۱	..... خلق یک پروژه ASP.NET CORE
۳۲	..... باز کردن پروژه در ویژوال استودیو
۳۴	..... باز کردن پروژه در ویژوال استودیو کد
۳۵	..... اجرای اپلیکیشن ASP.NET CORE
۳۶	..... آشنایی با نقاط پایانی
۳۹	..... آشنایی با مسیریابی
۳۹	..... آشنایی با رندر شدن HTML
۳۹	..... خلق و رندر کردن نما
۴۳	..... اضافه کردن خروجی پویا
۴۴	..... نتیجه کلی کار
۴۴	..... خلاصه فصل

## فصل ۳- اولین اپلیکیشن ASP.NET CORE شما..... ۴۵

۴۵	..... آماده کردن صحنه
۴۵	..... خلق پروژه
۴۶	..... آماده سازی پروژه
۴۸	..... افزودن مدل داده
۴۹	..... خلق اکشن و نمای دوم
۵۱	..... لینک کردن اکشن متدها
۵۲	..... ساختن فرم
۵۴	..... دریافت اطلاعات فرم
۵۵	..... آشنایی با پیوند مدل
۵۷	..... ذخیره پاسخ ها
۵۸	..... اضافه کردن نمای تشکر

۵۹	..... نمایش پاسخ ها
۶۲	..... افزودن اعتبارسنجی
۶۶	..... هایلایت کردن فیلدهای نامعتبر
۶۸	..... استایل بندی محتوا
۶۸	..... سبک دهی به نمای خوش آمدگویی
۷۰	..... استایل بندی نمای فرم
۷۱	..... استایل بندی نمای تشکر
۷۳	..... استایل بندی نمای لیست
۷۴	..... خلاصه فصل

#### فصل ۴- استفاده از ابزارهای برنامه نویسی..... ۷۵

۷۶	..... خلق پروژه های ASP.NET CORE
۷۶	..... خلق پروژه با استفاده از خط فرمان
۷۷	..... آشنایی با محدودیت های تمپلیت های پروژه
۷۸	..... باز کردن پروژه
۷۹	..... افزودن کد و محتوا به پروژه ها
۸۱	..... آشنایی با اسکفلد کردن آیتم
۸۲	..... ساخت و اجرای پروژه ها
۸۴	..... استفاده از ویژگی Hot Reload
۸۶	..... ساخت و اجرای پروژه ها با استفاده از ویژوال استودیو کد
۸۶	..... ساخت و اجرای پروژه ها با استفاده از ویژوال استودیو
۸۶	..... مدیریت پکیج ها
۸۶	..... مدیریت پکیج های NuGet
۸۷	..... مدیریت پکیج های ابزار
۸۸	..... مدیریت پکیج های سمت کلاینت
۹۰	..... اشکال زدایی پروژه ها
۹۱	..... چگونه من کد خود را دیباگ می کنم
۹۱	..... خلاصه فصل

#### فصل ۵- ویژگی های ضروری C#..... ۹۲

۹۳	..... آماده شدن برای این فصل
۹۳	..... باز کردن پروژه
۹۳	..... فعال کردن فریمورک MVC
۹۴	..... خلق کامپوننت های اپلیکیشن
۹۴	..... خلق مدل داده

۹۵	..... خلق کنترلر و نما
۹۶	..... انتخاب پورت HTTP
۹۶	..... اجرای اپلیکیشن مثال
۹۷	..... آشنایی با دستورات سطح بالا
۹۸	..... آشنایی با دستورات USING همگانی
۹۹	..... آشنایی با دستورات using ضمنی
۱۰۰	..... آشنایی با تجزیه و تحلیل وضعیت نول
۱۰۱	..... اطمینان از تخصیص مقدار به فیلدها و ویژگی ها
۱۰۲	..... استفاده از انواع Nullable
۱۰۳	..... انتخاب نوع NULLABLE مناسب
۱۰۴	..... بررسی از نظر مقادیر نول
۱۰۶	..... استفاده از عملگر شرطی Null
۱۰۶	..... استفاده از عملگر Null-Coalescing
۱۰۷	..... رونویسی تجزیه و تحلیل وضعیت نول
۱۰۸	..... غیرفعال کردن هشدارهای تجزیه و تحلیل وضعیت نول
۱۰۹	..... استفاده از درون یابی رشته ای
۱۱۰	..... استفاده از آغازگر شی و مجموعه
۱۱۱	..... استفاده از Index Initializer
۱۱۲	..... استفاده از عبارات جدید TARGET-TYPED
۱۱۳	..... تطبیق الگو
۱۱۴	..... تطبیق الگو در دستورات switch
۱۱۵	..... استفاده از متدهای افزونه
۱۱۸	..... اعمال متدهای افزونه به یک رابط کاربری
۱۱۹	..... خلق متدهای افزونه فیلترینگ
۱۲۱	..... استفاده از عبارات لامبدا
۱۲۳	..... تعریف توابع
۱۲۶	..... سایر اشکال عبارات لامبدا
۱۲۷	..... استفاده از متدها و ویژگی های عبارات لامبدا
۱۲۸	..... استفاده از استنباط نوع
۱۲۹	..... استفاده از انواع ناشناس
۱۳۱	..... استفاده ی پیاده سازی پیش فرض در واسط های کاربری
۱۳۳	..... استفاده از متدهای آسنکرون
۱۳۳	..... کار مستقیم با وظایف
۱۳۵	..... اعمال کلمات کلیدی await و async

۱۳۶	..... استفاده از برشمارش آسنکرون
۱۴۰	..... بدست آوردن نام ها
۱۴۱	..... خلاصه فصل

## فصل ۶- تست اپلیکیشن ASP.NET CORE ..... ۱۴۲

۱۴۲	..... تصمیم گیری در مورد واحد تست
۱۴۳	..... آماده شدن برای این فصل
۱۴۳	..... باز کردن پروژه
۱۴۴	..... انتخاب پورت HTTP
۱۴۴	..... فعال کردن فریمورک MVC
۱۴۵	..... ایجاد کامپوننت های برنامه
۱۴۵	..... ایجاد مدل داده
۱۴۶	..... ایجاد View و Controller
۱۴۷	..... اجرای اپلیکیشن مثال
۱۴۷	..... ایجاد پروژه تست واحد
۱۴۸	..... حذف کلاس تست پیش فرض
۱۴۸	..... نوشتن و اجرای تست های واحد
۱۵۱	..... اجرای تست ها با کاوشگر تست ویژوال استودیو
۱۵۱	..... اجرای تست در ویژوال استودیو کد
۱۵۲	..... اجرای تست ها از خط فرمان
۱۵۳	..... تصحیح تست واحد
۱۵۴	..... جداسازی کامپوننت ها برای تست واحد
۱۵۶	..... جداسازی یک جزء (کامپوننت)
۱۵۹	..... آشنایی با توسعه تست - محور
۱۶۰	..... استفاده از پکیج ماکت (ساختگی)
۱۶۰	..... ایجاد یک شیء ماکت
۱۶۲	..... خلاصه فصل

## فصل ۷- SPORTSSTORE: یک اپلیکیشن واقعی ..... ۱۶۳

۱۶۳	..... تست کردن واحد
۱۶۴	..... ایجاد پروژه ها
۱۶۴	..... ایجاد پروژه تست واحد
۱۶۵	..... باز کردن پروژه ها
۱۶۵	..... پیکربندی پورت HTTP
۱۶۶	..... ایجاد پوشه های پروژه اپلیکیشن

۱۶۷	آماده سازی سرویس ها و پایپلاین درخواست اپلیکیشن
۱۶۸	پیکربندی موتور نمای ریزور
۱۶۹	خلق کنترلر و نما
۱۷۰	شروع مدل داده
۱۷۱	بررسی و اجرای اپلیکیشن
۱۷۱	افزودن داده به برنامه
۱۷۱	نصب پکیج های Entity Framework Core
۱۷۲	تعریف رشته اتصال
۱۷۳	ایجاد کلاس زمینه ای دیتابیس
۱۷۳	پیکربندی Entity Framework Core
۱۷۴	خلق مخزن
۱۷۵	آشنایی با اینترفیس های IQUERYABLE<T> و IENUMERABLE<T>
۱۷۷	خلق مهاجرت دیتابیس
۱۷۷	ایجاد داده های نمونه
۱۸۰	ریست کردن دیتابیس
۱۸۰	نمایش لیست محصولات
۱۸۰	استفاده از اسکفلد کردن ویژوال استودیو
۱۸۱	آماده سازی کنترلر
۱۸۲	تست واحد: دسترسی به مخزن
۱۸۳	به روز رسانی نما
۱۸۴	اجرای اپلیکیشن
۱۸۴	اضافه کردن صفحه بندی
۱۸۵	تست واحد: صفحه بندی
۱۸۶	نمایش لینک های صفحه
۱۸۷	اضافه کردن View Model
۱۸۸	اضافه کردن کلاس تگ هلپر
۱۸۹	تست واحد: خلق لینک های صفحه
۱۹۱	اضافه کردن نما مدل داده
۱۹۲	تست واحد: داده نمای مدل صفحه
۱۹۴	نمایش لینک های صفحه
۱۹۵	بهینه کردن URLها
۱۹۷	سبک دادن به محتوا
۱۹۷	نصب پکیج بوت استرپ
۱۹۸	استفاده از استایل های بوت استرپ

۲۰۱	..... ایجاد نمای نسبی
۲۰۲	..... خلاصه فصل

### فصل ۸ - SPORTSSTORE: منو و سبد خرید ..... ۲۰۳

۲۰۳	..... اضافه کردن کنترل های ناوبری
۲۰۳	..... فیلتر کردن لیست محصولات
۲۰۵	..... تست واحد: بروزرسانی تست های واحد موجود
۲۰۷	..... تست واحد: فیلتر دسته ها
۲۰۸	..... اصلاح طرح URL
۲۱۲	..... ساخت منوی ناوبری دسته ها
۲۱۲	..... خلق کامپوننت نمای ناوبری
۲۱۴	..... ایجاد لیست دسته ها
۲۱۵	..... تست واحد: تولید لیست دسته ها
۲۱۶	..... ایجاد نما
۲۱۷	..... هایلایت کردن دسته فعال
۲۱۸	..... تست واحد: گزارش کردن دسته منتخب
۲۲۰	..... تصحیح شمارش صفحات
۲۲۲	..... تست واحد: شمارش محصول مختص دسته
۲۲۳	..... ساخت سبد خرید
۲۲۴	..... پیگیربندی صفحات ریزور
۲۲۶	..... ایجاد صفحه ریزور
۲۲۶	..... ایجاد دکمه های افزودن به سبد خرید
۲۲۸	..... فعال کردن جلسات
۲۲۹	..... پیاده سازی ویژگی سبد خرید
۲۳۰	..... تست واحد: تست سبد خرید
۲۳۳	..... تعریف متدهای افزونه وضعیت جلسه
۲۳۴	..... تکمیل صفحه ریزور
۲۳۶	..... آشنایی با صفحات ریزور
۲۳۸	..... تست واحد: صفحات ریزور
۲۴۰	..... خلاصه فصل

### فصل ۹ - SPORTSSTORE: تکمیل سبد خرید ..... ۲۴۱

۲۴۱	..... بهینه سازی مدل سبد خرید با یک سرویس
۲۴۱	..... خلق کلاس سبد خرید آگاه از ذخیره سازی
۲۴۴	..... رجیستر کردن سرویس



۲۴۵	.....	تسهیل صفحه ریزور سبد خرید
۲۴۶	.....	بروزرسانی تست های واحد
۲۴۷	.....	تکمیل عملکرد سبد خرید
۲۴۷	.....	حذف اقلام از سبد خرید
۲۵۰	.....	افزودن ویجت خلاصه سبد خرید
۲۵۰	.....	افزافه کردن پکیج Font Awesome
۲۵۱	.....	خلق کلاس کامپوننت نما و نما
۲۵۳	.....	ارسال سفارشات
۲۵۳	.....	خلق کلاس مدل
۲۵۴	.....	افزودن فرآیند پرداخت
۲۵۵	.....	خلق کنترلر و نما
۲۵۱	.....	پیاده سازی پردازش سفارش
۲۵۱	.....	بسط دیتابیس
۲۵۹	.....	بازنشانی دیتابیس
۲۵۹	.....	خلق مخزن برای سفارشات
۲۶۰	.....	آشنایی با مخزن سفارشات
۲۶۲	.....	تکمیل کنترلر سفارش
۲۶۳	.....	تست واحد: پردازش سفارش
۲۶۵	.....	نمایش خطاهای اعتبارسنجی
۲۶۷	.....	نمایش صفحه خلاصه
۲۶۹	.....	خلاصه فصل

## فصل ۱۰- SPORTSSTORE: مدیریت ..... ۲۷۰

۲۷۰	.....	آماده سازی سرور بلیزور
۲۷۱	.....	خلق فایل Imports
۲۷۲	.....	ایجاد صفحه ریزور استارت آپ
۲۷۲	.....	خلق کامپوننت های مسیریابی و چیدمان
۲۷۴	.....	خلق کامپوننت ریزور
۲۷۴	.....	چک کردن نصب بلیزور
۲۷۵	.....	مدیریت سفارشات
۲۷۶	.....	بهبود کردن مدل
۲۷۷	.....	نمایش سفارشات به مدیر
۲۸۱	.....	افزافه کردن مدیریت کاتالوگ
۲۸۲	.....	گسترش دادن مخزن

۲۸۳	..... اعمال خصیصه های اعتبارسنجی به مدل داده
۲۸۴	..... خلق کامپوننت لیست
۲۸۶	..... خلق کامپوننت جزئیات
۲۸۸	..... خلق کامپوننت ویرایشگر
۲۹۱	..... حذف محصولات
۲۹۳	..... خلاصه فصل

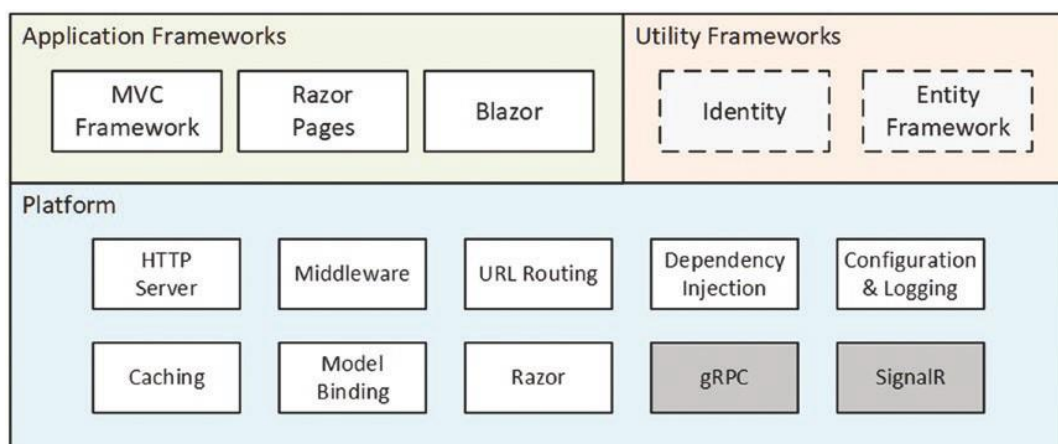
## فصل ۱۱ - SPORTSSTORE: امنیت و انتقال ..... ۲۹۴

۲۹۴	..... ایجاد دیتابیس آیدنتیتی
۲۹۴	..... نصب پکیج Identity برای Entity Framework Core
۲۹۵	..... خلق کلاس زمینه ای
۲۹۵	..... تعریف رشته اتصال
۲۹۶	..... پیکربندی اپلیکیشن
۲۹۸	..... خلق و اعمال مهاجرت دیتابیس
۲۹۸	..... تعریف داده های نمونه
۳۰۱	..... حذف و خلق مجدد دیتابیس ASP.NET CORE IDENTITY
۳۰۱	..... افزودن یک ویژگی مدیریت متعارف
۳۰۲	..... اعمال سیاست تعیین هویت پایه
۳۰۴	..... ایجاد کنترلر و نمای حساب کاربری
۳۰۸	..... تست امکانات امنیتی اپلیکیشن
۳۰۸	..... آماده سازی اپلیکیشن ASP.NET CORE برای استقرار
۳۰۹	..... پیکربندی رسیدگی به خطاها
۳۱۱	..... خلق تنظیمات پیکربندی تولید
۳۱۲	..... خلق ایمیج داکر
۳۱۲	..... نصب Docker Desktop
۳۱۲	..... خلق فایل های پیکربندی Docker
۳۱۳	..... انتشار و تهیه ایمیج اپلیکیشن
۳۱۵	..... اجرای اپلیکیشن کانتینری شده
۳۱۶	..... خلاصه فصل

# آشنایی با مجموعه ASP.NET Core

مجموعه ASP.NET Core پلتفرم برنامه نویسی وب مایکروسافت می باشد. ASP.NET اولیه در سال ۲۰۰۲ معرفی شد و از آن زمان تحت چندین تحول و تجدید مکرر قرار گرفته است تا وقتی که به ASP.NET Core نسخه ۶ تبدیل شده است که موضوع این کتاب است.

ASP.NET Core شامل یک پلتفرم برای پردازش درخواست های HTTP، یک مجموعه فریمورک های اصلی<sup>۱</sup> برای خلق اپلیکیشن ها و فریمورک های کاربردی<sup>۲</sup> ثانویه برای فراهم کردن سرویس های پشتیبان می باشد، همانطور که در شکل ۱-۱ بطور شماتیک ترسیم شده است.



شکل ۱-۱ ساختار ASP.NET Core

## آشنایی با ASP.NET CORE، ASP.NET FRAMEWORK و ASP.NET

اگر هرگز برای یک سازمان بزرگ کار نکرده اید، ممکن است تصور کنید مایکروسافت یک سازمان منظم با استراتژی روشن است که یک لشکر از برنامه نویسان بزرگ برای آن کار می کنند تا محصولات پیچیده ای چون ASP.NET Core را به بازار عرضه کند.

در واقعیت چنین نیست، مایکروسافت مجموعه ای پر هرج و مرج از قبایل غیرکارکردی است که بطور مدام در حال سرکوب و زیرآب زنی همدیگر هستند تا پرستیژ و ارتقای شغلی برای خود دست و پا کنند. محصولات معمولاً

<sup>1</sup> Principal Frameworks

<sup>2</sup> Utility Frameworks

در اوج دعوی آنها منتشر می شوند و موفقیت ها اغلب کاملاً غیرمنتظره است. این مختص مایکروسافت نیست و در بسیاری از سازمان های بزرگ مصداق دارد، اما در مورد ASP.NET Core و نامگذاری گیج کننده آن که مایکروسافت خلق کرده است، کاملاً صحیح است.

چندین سال پیش، بخشی از مایکروسافت که مسئول ASP.NET بود یک نسخه مخصوص خود از پلتفرم .net ایجاد کرد و نسبت به سایر بخش های دات نت بیشتر آپدیت می شد. ASP.NET Core و .NET Core خلق شدند، امکان برنامه نویسی چند پلتفرمی فراهم شد، هرچند با استفاده از زیرمجموعه ای از API های اولیه دات نت، که بسیاری از آنها مخصوص ویندوز بود. این یک انتقال دردناک بود، اما بدین معنی بود که توسعه وب بطور مستقل از محیط توسعه تنها ویندوز اولیه خود، دچار تحول و تکامل شده است و تحت عنوان دات نت فریمورک به کار ادامه داد.

اما هیچ کس دوست ندارد میراث خوار یک قوم باشد، زیرا افتخاری در روشن نگهداشتن چراغ ها در مایکروسافت وجود نداشت. روشن بود که .NET Core آینده است و گروه دات نت یکی مدعی شدند که تکنولوژی و API های آنها بایستی بخشی از .NET Core باشد. API های .NET Core بتدریج بسط داده شد و نتیجه اش یک به هم ریختگی بدون پیوستگی، تلاش های خودخواهانه برای تمایز بین .NET Core و .NET Framework و استاندارد کردن API ها بود.

برای رفع این مشکل مایکروسافت .NET Core و .NET Framework را با هم ادغام کرد و به .NET تبدیل شد، یعنی بخش Core از نام آن حذف گردید. فکر کنم همین نام .NET را نیز هنگام خارج شدن از اتاق به قصد تعطیلات آخر هفته و بصورت گذری انتخاب کردند اما من شک دارم که نتیجه ماههای زیادی بحث داغ باشد.

مشکل حذف بخش Core از نام آن این است که نمی توان بطور یکنواخت آن را اجرا کرد. نام ASP.NET Core از ابتدا به نسخه .NET Core مربوط به ASP.NET دلالت داشت و برگشت به آن نام می توانست خیلی سردرگم کننده تر باشد.

در نتیجه مایکروسافت نمی توانست تصمیم بگیرد که از کدام نام استفاده کند. شما نام ASP.NET Core 6 را در راهنما و اسناد زیادی می بینید و این نامی است که من در این کتاب استفاده می کنم، اما خواهید دید که ASP.NET Core در .NET 6 نیز استفاده می شود بخصوص در مطبوعات و مطالب بازاریابی. روشن نیست که کدام نام برنده خواهد شد، اما تا رسیدن به شفافیت نامی، بایستی به دقت مشخص کنید که قصد دارید از .NET Framework استفاده کنید، از .NET Core و یا از .net استفاده کنید.

## آشنایی با فریمورک های اپلیکیشن

زمانی که استفاده از ASP.NET Core را شروع می کنید، وقتی متوجه می شوید که فریمورک های اپلیکیشن<sup>۱</sup> متفاوتی وجود دارد، دچار سردرگمی می شوید. همانطور که می آموزید، این فریمورک ها مکمل بوده و مشکلات مختلف را حل می کنند یا بخاطر بعضی ویژگی ها، یک مشکل را به شیوه ای متفاوت حل می کنند. آشنایی با رابطه بین این فریمورک ها به معنی درک الگوهای طراحی در حال تحولی است که مایکروسافت پشتیبانی می کند، همانطور که در بخش های آینده توضیح خواهم داد.

<sup>1</sup> Application Frameworks

## آشنایی با فریمورک MVC

فریمورک MVC در نسخه های قبلی ASP.NET و خیلی قبل از .NET Core و .NET 6. معرفی گردید. ASP.NET اولیه به یک مدل توسعه بنام وب پیچ متکی بود که تجربه نوشتن برنامه های دسکتاپ را باز تولید می کرد اما به پروژه های وب گسسته ای ختم می شد که بخوبی مقیاس پذیر نبود. فریمورک MVC همراه با فریمورک وب پیچ با یک مدل توسعه وب معرفی شد که ماهیت HTTP و HTML را می پذیرفت بجای آنکه سعی کند آن را مخفی کند.

MVC سرنام مدل-نما-کنترلر<sup>۱</sup> است، یک الگوی طراحی که شکل یک اپلیکیشن را توصیف می کند. در الگوی MVC روی جداسازی نگرانی ها تاکید می شود، جایی که عملکرد هر بخش بطور مستقل تعریف می شود، که یک آنتی دوت موثر برای معماری مشکل دار منسوخ شده وب پیچ ها است.

نسخه های مختلف فریمورک MVC براساس بنیان های ASP.NET ساخته شده است که از ابتدا برای وب پیچ ساخته شده بود، که منجر به ویژگی ها و روند کاری نامناسبی می شد. با حرکت به سمت .NET Core، خود ASP.NET Core به ASP.NET تبدیل شد و فریمورک MVC براساس بنیانی باز، قابل بسط و بین پلتفرمی بازسازی شد.

فریمورک MVC به عنوان بخش مهمی از ASP.NET Core باقی مانده است، اما نحوه استفاده عمومی آن با معرفی اپلیکیشن های تک صفحه ای (SPAs)<sup>۲</sup> تغییر کرده است. در SPA مرورگر یک درخواست HTTP منفرد ارسال می کند و یک سند HTML دریافت می کند که کلاینت غنی تحویل می دهد و معمولا به زبان های کلاینت جاوااسکریپت از قبیل آنگولار یا ری اکت<sup>۳</sup> نوشته شده است. حرکت به سمت SPA بدین معنی است که جداسازی تمیز و مشخص، چیزی که فریمورک MVC از ابتدا به این نیت ایجاد شد و تاکید بر این نکته می کند که دیگر الگوی MVC نیاز نیست، گرچه فریمورک MVC هنوز مفید است (و در وب سرویس ها برای پشتیبانی از SPA استفاده می شود، چیزی که در فصل ۱۹ کتاب بحث خواهد شد).

### چیدن الگوها در جای خود

الگوهای طراحی واکنش های قوی را بر می انگیزد، همانطور که من ایمیل هایی از مخاطبان دریافت می کنم که شاهد آن است. بخش مهمی از پیام هایی که دریافت می کنم شکایاتی است که من الگویی را صحیح اعمال نکرده ام.

الگوها جواب افراد دیگر به مشکلاتی است که در پروژه های دیگر با آن مواجه می شوند. اگر خود را در مقابل همان مشکلات یافتید، درک نحوه حل کردن آن در گذشته می تواند کمک کننده باشد. اما این بدین معنی نیست که باید همان الگوها را دقیقا رعایت کنید یا اصلا از این الگوها تبعیت کنید، اگر عواقب آن را بخوبی بشناسید. مثلا اگر الگویی قرار است پروژه را قابل اداره تر کند، شما انحراف از آن الگو را انتخاب می کنید، پس باید بپذیرید که پروژه شما ممکن است سخت تر اداره شود. اما الگویی که درهم ریخته است می تواند بدتر از الگو نداشتن باشد و هیچ الگویی برای همه پروژه ها مناسب نیست.

<sup>1</sup> Model-View-Controller

<sup>2</sup> Single-Page Applications

<sup>3</sup> Angular or React

توصیه من این است که از الگوها آزدانه استفاده کنید، طبق ضرورت آنها را اتخاذ کرده، کسانی که با الگوها موجب سردرگمی شما می شوند را نادیده بگیرید.

## آشنایی با صفحات ریزور

یکی از معایب فریمورک MVC این است که کارهای آمادگی قبلی زیادی دارد تا بتوان اپلیکیشن را راه اندازی و محتوا تولید کرد. علیرغم مشکلات ساختاری آن، یکی از مزایای الگوی وب پیچ ساده بودن اپلیکیشن آن بود که در طی چند ساعت قابل ساخت و اجرا بود.

صفحات ریزور<sup>۱</sup> الگوی ساخت وب پیچ را گرفته و آن را با استفاده از ویژگی های پلتفرم پیاده می کند، چیزی که از ابتدا برای فریمورک MVC نوشته شده بود. کد و محتوا مخلوط می شود تا یک صفحه خود-کفا ایجاد شود، این موجب سرعت داشتن نوشتن وب پیچ ها می شود، بدون اینکه بعضی مشکلات فنی زمینه ای رخ دهد (گرچه مشکل مقیاس پذیری بالای پروژه های پیچیده هنوز باقی می ماند).

صفحات ریزور را می توان همراه با فریمورک MVC استفاده کرد، کاری که من قصد انجام آن را دارم. من بخش اصلی اپلیکیشن را با استفاده از فریمورک MVC می نویسم و از صفحات ریزور برای ویژگی های ثانویه از قبیل ابزارهای مدیریت و گزارش دهی استفاده می کنم. این رویکرد را در فصول ۷ تا ۱۱ خواهید دید، که من یک اپلیکیشن ASP.NET Core واقعی به نام SportsStore را می نویسم.

## آشنایی با بلیزور

محبوب شدن فریمورک های سمت کلاینت جاوااسکریپت می تواند برای توسعه دهندگان با C# مانعی باشد، که بایستی یک زبان برنامه نویسی متفاوت (و تا حدودی ایدیوسنکراتیک) را بیاموزند. من خودم جاوااسکریپت را دوست دارم زیرا به اندازه C# سیال و نمایشی است. اما تسلط بر یک زبان برنامه نویسی جدید مستلزم زمان و تعهد زیادی است، بخصوص زبانی که بطور بنیادین با C# تفاوت هایی دارد.

بلیزور<sup>۲</sup> تلاش دارد این خلا را بپوشاند و به C# امکان استفاده از اپلیکیشن های سمت کلاینت را بدهد. بلیزور دو نسخه دارد: بلیزور سرور و بلیزور وب اسمبلی. بلیزور سرور<sup>۳</sup> نسخه پایدار است و به عنوان بخشی از ASP.NET Core ساپورت می شود و با اتصال مقاوم HTTP به سرور ASP.NET Core کار می کند، جایی که کد C# اپلیکیشن اجرا می شود. بلیزور وب اسمبلی<sup>۴</sup> یک نسخه آزمایشی است که یک مرحله فراتر رفته و کد C# اپلیکیشن را در مرورگر اجرا می کند. هیچکدام از نسخه های بلیزور برای همه موقعیت ها مناسب نیست، همانطور که در فصل ۳۳ بحث می کنم، اما مسیر را برای آینده توسعه وب با ASP.NET Core مشخص می کنند.

<sup>1</sup> Razor Pages

<sup>2</sup> Blazor

<sup>3</sup> Blazor Server

<sup>4</sup> Blazor WebAssembly

## آشنایی با فریمورک های کاربردی

دو فریمورک با ASP.NET Core ارتباط نزدیکی دارند اما مستقیماً در تولید محتوای HTML یا داده درگیر نیستند. Entity Framework Core یک فریمورک نقشه کشی رابطه شیئی (ORM)<sup>۱</sup> مایکروسافت می باشد که داده های ذخیره شده در یک بانک اطلاعاتی رابطه ای را بصورت شیئی NET در اختیار قرار می دهد. Entity Framework Core را می توان در هر اپلیکیشن دات نت استفاده کرد و معمولاً برای دسترسی به پایگاه داده در برنامه های ASP.NET Core استفاده می شود.

ASP.NET Core Identity فریمورک تعیین هویت و مجوز مایکروسافت است و برای تأیید کردنشیال کاربر در برنامه های ASP.NET Core و محدود کردن دسترسی به ویژگی های مختلف اپلیکیشن استفاده می شود.

من فقط ویژگی های کلی و پایه این دو فریمورک را در این کتاب شرح می دهم و روی قابلیت های مورد نیاز در اکثر برنامه های ASP.NET Core تمرکز خواهم داشت. اما اینها هر دو فریمورک های پیچیده ای هستند که برای توصیف جزئیات آنها در کتابی که بیشتر در مورد ASP.NET Core است، خیلی وسیع هستند.

### موضوعاتی برای نسخه های آینده

من در این کتاب فضای کافی برای پوشش همه ویژگی های ASP.NET Core، Entity Framework Core و ASP.NET Core Identity ندارم، بنابراین روی جنبه هایی تمرکز کرده ام که اکثر پروژه ها به آن نیاز دارند. اگر موضوعاتی وجود دارد که فکر می کنید باید در نسخه بعدی یا در کتاب های عمقی تر اضافه کنم، لطفاً پیشنهادات خود را به آدرس adam@adam-freeman.com برای من ارسال کنید.

## آشنایی با پلتفرم ASP.NET Core

پلتفرم ASP.NET Core شامل ویژگی های سطح پایین مورد نیاز برای دریافت و پردازش درخواست های HTTP و ایجاد پاسخ است. یک سرور HTTP جامع، سیستمی از اجزای میان افزاری برای رسیدگی به درخواست ها و ویژگی های اصلی که فریمورک های برنامه به آن ها وابسته هستند، مانند مسیریابی URL و موتور نمایش ریزور وجود دارد.

بیشتر زمان برنامه نویسی شما صرف کار با فریمورک های برنامه می شود، اما استفاده موثر از ASP.NET Core مستلزم درک قابلیت های قدرتمندی است که پلتفرم ارائه می دهد، قابلیت هایی که فریمورک های سطح بالاتر بدون آنها نمی توانند کار کنند. من نحوه عملکرد پلتفرم ASP.NET Core را با جزئیات در بخش ۲ این کتاب نشان می دهم و توضیح می دهم که چگونه ویژگی های ارائه شده توسط این پلتفرم زیربنای هر جنبه ای از برنامه نویسی با ASP.NET Core است.

من دو ویژگی قابل توجه این پلتفرم را در این کتاب بحث نکرده ام: SignalR و gRPC. از SignalR برای ایجاد کانال های ارتباطی کم-تاخیر بین اپلیکیشن ها استفاده می شود. این پایه و اساسی برای فریمورک Blazor Server که در بخش ۴ این کتاب توضیح می دهم، فراهم می کند. هرچند SignalR به ندرت به طور مستقیم استفاده می شود

<sup>1</sup> Object-Relational Mapping

و جایگزین‌های بهتری برای آن در محدود پروژه‌هایی که به پیام‌رسانی کم - تأخیر نیاز دارند، مانند Azure Event Grid یا Azure Service Bus وجود دارد.

gRPC یک استاندارد نوظهور برای فراخوانی ریموت رویه (RPCs)<sup>۱</sup> بصورت بین پلتفرمی از طریق HTTP است که در ابتدا توسط گوگل (g در gRPC) ایجاد شد و مزایای کارایی و مقیاس‌پذیری را ارائه می‌دهد. gRPC ممکن است استاندارد آینده برای خدمات وب باشد، اما نمی‌توان از آن در برنامه‌های کاربردی وب استفاده کرد زیرا به کنترل سطح پایین پیام‌های HTTP که ارسال می‌کند نیاز دارد، که مرورگرها اجازه نمی‌دهند (یک کتابخانه مرورگر وجود دارد که به gRPC اجازه می‌دهد از طریق یک پراکسی سرور استفاده شود، اما مزایای استفاده از gRPC را تضعیف می‌کند). تا زمانی که gRPC را نتوان در مرورگر استفاده کرد، گنجاندن آن در ASP.NET Core فقط به نفع پروژه‌هایی است که از آن برای ارتباط بین سرورهای بک‌اند استفاده می‌کنند، که پروتکل‌های جایگزین زیادی برای آن وجود دارد. من ممکن است gRPC را در نسخه‌های بعدی این کتاب پوشش دهم، اما نه تا زمانی که بتوان از آن در مرورگر استفاده کرد یا به پروتکل غالب دیتاستر تبدیل شود.

## آشنایی با این کتاب

برای بردن بیشترین بهره از این کتاب، باید با اصول توسعه وب آشنا باشید، نحوه کار HTML و CSS را بدانید و دانش کاری سی شارپ (C#) داشته باشید. اگر تا کنون توسعه و برنامه‌نویسی سمت کلاینت مانند جاوا اسکریپت انجام نداده‌اید، نگران نباشید. تاکید در این کتاب بر C# و ASP.NET Core است و هر آنچه را که باید در طول فصل‌ها یاد بگیرید را قادر خواهید بود یاد بگیرید. در فصل ۵، مهمترین ویژگی‌های C# برای برنامه‌نویسی در ASP.NET Core خلاصه می‌شود.

## برای اجرای مثال‌های کتاب به چه نرم‌افزاری نیاز داریم؟

شما به یک ویرایشگر کد (ویژوال استودیو یا ویژوال استودیو کد)، کیت توسعه نرم‌افزار .NET Core و SQL Server LocalDB نیاز دارید. همه اینها در سایت مایکروسافت برای استفاده بدون پرداخت هزینه در دسترس هستند، و فصل ۲ حاوی دستورالعمل‌هایی برای نصب همه چیزهایی است که نیاز دارید.

## برای دنبال کردن مثال‌ها به چه پلتفرمی نیاز داریم؟

این کتاب برای ویندوز نوشته شده است. من از Windows 10 Pro استفاده کرده‌ام، اما هر نسخه‌ای از ویندوز که توسط Visual Studio Code، Visual Studio و NET Core پشتیبانی می‌شود، باید کار کند. ASP.NET Core بر روی پلتفرم‌های دیگر پشتیبانی می‌شود، اما مثال‌های موجود در این کتاب به ویژگی SQL Server LocalDB، که مخصوص ویندوز است، متکی هستند. اگر می‌خواهید از پلتفرم دیگری استفاده کنید، می‌توانید از طریق adam@adam-freeman.com با من تماس بگیرید و من به شما نکات کلی برای تطبیق مثال‌ها ارائه می‌دهم، البته با این اخطار که نمی‌توانم راهنمایی دقیق و مفصلی ارائه کنم، اگر گیر کرده باشید.

<sup>1</sup> Remote Procedure Calls



## اگر در اجرای مثال‌ها مشکل داشته باشیم، چکار کنیم؟

اولین کاری که باید انجام دهید این است که به ابتدای فصل برگردید و دوباره شروع کنید. بیشتر مشکلات به دلیل از دست دادن یک مرحله یا دنبال نکردن کامل یک لیست ایجاد می‌شود. به تاکید در فهرست کدها توجه زیادی داشته باشید، که تغییرات مورد نیاز در آنها هایلایت شده است.

در مرحله بعد، فهرست اشتباهات/اصلاحات را که در مخزن GitHub کتاب موجود است، بررسی کنید. کتاب‌های فنی پیچیده هستند و با وجود تلاش‌های من و ویراستارانم، اشتباه اجتناب‌ناپذیر است. لیست خطاها را برای فهرست خطاهای شناخته شده و دستورالعمل رفع آنها را بررسی کنید.

اگر همچنان مشکل دارید، پروژه را برای فصلی که در حال خواندن آن هستید از مخزن GitHub کتاب، <https://github.com/apress/pro-asp.net-core-6> دانلود کنید و آن را با پروژه خود مقایسه کنید. من کدهای موجود در مخزن GitHub را بصورت کار کردن در هر فصل ایجاد کرده‌ام، بنابراین شما باید فایل‌های مشابه با محتویات یکسان در پروژه خود داشته باشید.

اگر هنوز نمی‌توانید مثالی را اجرا کنید، می‌توانید برای کمک با من در [adam@adam-freeman.com](mailto:adam@adam-freeman.com) تماس بگیرید. لطفاً در ایمیل خود روشن کنید که کدام کتاب را می‌خوانید و در کدام فصل/مثال مشکل دارید. لطفاً به یاد داشته باشید که من ایمیل‌های زیادی دریافت می‌کنم و ممکن است فوراً پاسخ ندهم.

## اگر اشتباهی در کتاب پیدا کنیم، چه می‌شود؟

می‌توانید خطاها را از طریق ایمیل به آدرس [adam@adam-freeman.com](mailto:adam@adam-freeman.com) به من گزارش دهید، اگر چه از شما می‌خواهم که ابتدا فهرست اشتباهات/اصلاحات این کتاب را بررسی کنید، که می‌توانید آن را در مخزن GitHub کتاب در <https://github.com/apress/pro-asp.net-core-6> ببینید، ممکن است قبلاً گزارش شده باشد.

من خطاهایی را که احتمالاً باعث سردرگمی خوانندگان می‌شود، به ویژه مشکلات مربوط به کد مثال را به فایل خطا/تصحیح موجود در مخزن GitHub اضافه می‌کنم، با تشکر از اولین خواننده‌ای که آنها را گزارش کرده است. من فهرستی از مسائل کمتر جدی را نگه می‌دارم، که معمولاً به معنای خطاهایی در متن پیرامون مثال‌ها است و هنگام نوشتن یک نسخه جدید، آنها را برطرف می‌کنم.

## جایزه گزارش خطا

شرکت apress موافقت کرده است که یک کتاب الکترونیکی رایگان به خوانندگانی بدهد که برای اولین بار خطا یا اشتباهی را گزارش می‌کنند و آن را در فهرست خطای github برای این کتاب گزارش خواهند کرد. خوانندگان می‌توانند هر کدام از کتاب‌های الکترونیکی apress موجود را از طریق [Springerlink.com](http://Springerlink.com) انتخاب کنند، نه فقط کتاب‌های من.

این یک برنامه کاملاً اختیاری و آزمایشی است. اختیاری به این معنی است که فقط من تصمیم می‌گیرم که کدام خطاها در لیست خطاها فهرست شوند و کدام خواننده اولین کسی است که گزارش می‌دهد. به معنای تجربی، apress ممکن است تصمیم بگیرد که به هر دلیلی در هر زمانی کتاب دیگری را تقدیم نکند. هیچ درخواست تجدید

نظری وجود ندارد و این یک قول یا قرارداد یا هر نوع پیشنهاد رسمی یا رقابتی نیست. یا به عبارت دیگر، این یک روش خوب و غیررسمی برای تشکر و تشویق خوانندگان است تا اشتباهاتی را که هنگام نوشتن این کتاب از قلم انداخته ام، گزارش کنند.

## این کتاب چه چیزی را پوشش می دهد؟

من سعی کرده ام ویژگی هایی را که برای اکثر پروژه های ASP.NET Core نیاز است پوشش دهم. این کتاب به چهار بخش تقسیم شده است که هر بخش مجموعه ای از موضوعات مرتبط را پوشش می دهد.

### بخش ۱: معرفی ASP.NET Core

این بخش از کتاب - که شامل این فصل نیز هست - ASP.NET Core را معرفی می کند. علاوه بر راه اندازی محیط توسعه و ایجاد اولین برنامه خود، با مهم ترین ویژگی های #C برای توسعه وب در ASP.NET Core و نحوه استفاده از ابزارهای توسعه ASP.NET Core آشنا خواهید شد. اما بیشتر بخش ۱ به نوشتن پروژه ای به نام SportsStore اختصاص یافته است، که از طریق آن یک فرآیند برنامه نویسی واقع گرایانه از آغاز تا استقرار را به شما نشان می دهم، تمام ویژگی های اصلی ASP.NET Core را لمس کرده و نشان می دهم که چگونه آنها با هم هماهنگ می شوند - چیزی که می تواند در فصل های عمقی ادامه کتاب گم شود.

### بخش ۲: پلتفرم ASP.NET Core

فصل های این بخش از کتاب ویژگی های کلیدی پلتفرم ASP.NET Core را پوشش می دهند. من نحوه پردازش درخواست های HTTP، نحوه ایجاد و استفاده از اجزای میان افزار، نحوه ایجاد مسیرها، نحوه تعریف و بکارگیری سرویس ها و نحوه کار با Entity Framework Core را توضیح می دهم. این فصول پایه های ASP.NET Core را توضیح می دهند و درک آنها برای برنامه نویسی موثر با ASP.NET Core ضروری است.

### بخش ۳: اپلیکیشن های ASP.NET Core

فصول این بخش از کتاب نحوه ایجاد انواع مختلف برنامه ها از جمله وب سرویس های RESTful و برنامه های HTML با استفاده از کنترلرها و صفحات ریزور را شرح می دهد. این فصول همچنین ویژگی هایی را شرح می دهند که تولید HTML را آسان می کند، از جمله نماها، کامپوننت های نما، و تگ هلمپرها.

### بخش ۴: ویژگی های پیشرفته ASP.NET Core

بخش پایانی کتاب نحوه ایجاد اپلیکیشن با استفاده از Blazor Server، نحوه استفاده از Blazor WebAssembly، آزمایشی و نحوه احراز هویت کاربران و مجوز دسترسی با استفاده از ASP.NET Core Identity را بحث می کند.

## چه چیزی در این کتاب پوشش داده نمی‌شود؟

در این کتاب مطالب پایه توسعه وب از قبیل HTML و CSS پوشش داده نمی‌شود و #C آموزش داده نمی‌شود (گرچه در فصل ۵ ویژگی‌های مفید سی شارپ که برای برنامه نویسی با ASP.NET Core لازم است، توضیح داده می‌شود؛ زیرا ممکن است برای توسعه‌دهندگانی که از نسخه‌های قدیمی‌تر NET استفاده می‌کنند آشنا نباشند).

همان‌قدر که من دوست دارم در کتاب‌هایم به جزئیات بپردازم، همه ویژگی‌های ASP.NET Core برای جریان اصلی توسعه وب و برنامه نویسی مفید نیستند و باید کتاب‌هایم را در اندازه قابل چاپ نگه دارم. وقتی تصمیم می‌گیرم یک ویژگی را حذف کنم، به این دلیل است که فکر نمی‌کنم مهم باشد یا به این دلیل است که می‌توان با استفاده از تکنیکی که پوشش می‌دهم، به همان نتیجه رسید.

همانطور که قبلاً ذکر شد، من پشتیبانی از ASP.NET Core از SignalR و gRPC را توضیح نداده‌ام و در فصل‌های بعدی به ویژگی‌های دیگری اشاره می‌کنم که توضیح نمی‌دهم، یا به این دلیل که به طور گسترده قابل اجرا نیستند یا به این دلیل که جایگزین‌های بهتری برای آنها در دسترس است. در هر مورد، توضیح می‌دهم که چرا توضیحی را حذف کرده‌ام و به مستندات مایکروسافت برای آن موضوع اشاره می‌کنم.

## چگونه با نویسنده تماس بگیرم؟

می‌توانید به من ایمیل بزنید adam@adam-freeman.com. چند سالی می‌شود که از اولین انتشار آدرس ایمیل در کتاب‌هایم می‌گذرد. من کاملاً مطمئن نبودم که ایده خوبی است، اما خوشحالم که آن را انجام دادم. من ایمیل‌هایی از سراسر جهان دریافت کرده‌ام، از خوانندگانی که در هر صنعتی کار می‌کنند یا در حال تحصیل هستند، و - در اکثر موارد - ایمیل‌ها مثبت، مؤدبانه و لذت‌بخش هستند.

سعی می‌کنم بی‌درنگ پاسخ بدهم، اما ایمیل‌های زیادی دریافت می‌کنم، و گاهی اوقات تاخیر دارم، به خصوص زمانی که سرم پایین است و سعی می‌کنم نوشتن یک کتاب را به پایان برسانم. من همیشه سعی می‌کنم به خوانندگانی که با مثالی در کتاب گیر کرده‌اند کمک کنم، اگرچه از شما می‌خواهم قبل از تماس با من مراحل را که قبلاً در این فصل توضیح داده شد، دنبال کنید.

در حالی که از ایمیل‌های خوانندگان استقبال می‌کنم، اما برخی از سؤالات رایج وجود دارد که پاسخ آنها همیشه منفی خواهد بود. متأسفانه کد فروشگاه جدید شما را نمی‌توانم بنویسم، از انجام تکالیف دانشگاهی شما معذورم، در بحث طراحی تیم توسعه تان شرکت نمی‌کنم، یا به شما نحوه برنامه نویسی را آموزش نمی‌دهم.

## اگر واقعا از این کتاب لذت بردم، چطور؟

لطفاً به من ایمیل بزنید adam@adam-freeman.com و به من اطلاع دهید. آگاهی از یک خواننده خوشحال همیشه لذت بخش است و من از زمانی که برای ارسال آن ایمیل‌ها صرف می‌شود قدردانی می‌کنم. نوشتن این کتاب‌ها می‌تواند دشوار باشد، و این ایمیل‌ها انگیزه‌ای ضروری برای ادامه فعالیت‌هایی هستند که گاهی اوقات ممکن است غیرممکن به نظر برسد.

## اگر این کتاب من را عصبانی کرده باشد و من بخواهم شکایت کنم چه می شود؟

شما همچنان می توانید به من ایمیل بزنید به adam@adam-freeman.com، و من همچنان سعی خواهم کرد به شما کمک کنم. به خاطر داشته باشید که من فقط در صورتی می توانم کمک کنم که توضیح دهید مشکل چیست و می خواهید چه کاری انجام دهم. باید درک کنید که گاهی اوقات تنها نتیجه این است که بپذیرید من نویسنده شما تنها نیستم و تنها راه حل این است که شما این کتاب را برگردانید و کتاب دیگری را انتخاب کنید. من به هر چیزی که شما را ناراحت کرده است فکر می کنم، اما بعد از ۲۵ سال نوشتن کتاب، فهمیدم که همه از خواندن کتاب هایی که من دوست دارم بنویسم، لذت نمی برند.

## خلاصه فصل

در این فصل، صحنه را برای بقیه کتاب چیدم. من یک نمای کلی از ASP.NET Core ارائه دادم، الزامات و محتوای این کتاب را توضیح دادم و طریقه تماس با من را نشان دادم. در فصل بعدی، من به شما نشان می دهم که چگونه برای توسعه و برنامه نویسی با ASP.NET Core آماده شوید.