ASP.NET is a framework, or tool kit, for creating web applications. These are applications that operate over a network, largely (but not exclusively) where a user employs a web browser to communicate with a server over an intranet or the Internet. In this chapter, I will give you a high-level overview of ASP.NET and explain how I have mapped the key elements into the different parts of this book.

An Overview of ASP.NET

Web applications have special demands. The way that browsers work, the nature of HTTP (the protocol over which most web application operate), and the separation of the server from the client all affect the nature of web applications and web application programming. ASP.NET provides the features we need to meet those special demands and create an application that we can deliver using web technology.

ASP.NET is not a new technology. ASP.NET 1.0 was released as part of the wider .NET framework more than a decade ago. At the time that ASP.NET 1.0 was introduced, Visual Basic was a dominant force in the programming world, and Microsoft created ASP.NET to bring the Visual Basic programming model to the web development world, including concepts such as drag-and-drop controls, events, and design surfaces. These features are wrapped up in a feature set called Web Forms (which I cover in Part III of this book), predicated on the idea that the developer doesn’t need to have direct knowledge of or control over the underlying HTML and HTTP.

This may seem like an odd concept today, when almost every developer has at least a basic knowledge of HTTP and HTML, but it made perfect sense at the time. There was a huge population of Visual Basic developers, and Microsoft wanted to protect this market segment by giving them web development tools that built on their existing experience.

Note When I refer to the Visual Basic model, I don’t mean the language itself; rather, I mean the approach, tools, and environment that Visual Basic programmers use. The Visual Basic language has struggled since the introduction of .NET. Many programmers have moved to C#, leaving the market segment for Visual Basic .NET much reduced.
ASP.NET 1 introduced some core themes that have underpinned all versions of ASP.NET, including the most recent version, ASP.NET 4:

- **It has close integration with the .NET Framework**: ASP.NET is very tightly bound to the .NET Framework, such that ASP.NET applications are hosted in the .NET runtime and all the features of the .NET Framework (LINQ, the Entity Framework, automatic garbage collection, and so on) are available to ASP.NET programmers.

- **ASP.NET applications are compiled**: We build ASP.NET applications using a mix of annotated HTML and C# classes. But everything, including the marked-up HTML, is compiled into .NET classes to improve performance.

- **Visual Studio provides comprehensive support for ASP.NET**: We create ASP.NET applications just as we would any other kind of Visual Studio project. There is support for IntelliSense, debugging, and packaging and deployment, just as there is for other application types. You could choose to build an ASP.NET application outside of Visual Studio, but it would be much harder to do.

- **ASP.NET supports all .NET Framework languages**: Even though C# has emerged as the dominant .NET language, we can write ASP.NET applications using any of the languages supported by .NET. This includes Visual Basic .NET and F#. Microsoft had a brief foray into supporting other languages on .NET, including Ruby and Python, but that seems to have died recently.

- **ASP.NET is tightly integrated into IIS**: To deploy and run an ASP.NET application, you really need to use IIS, which is Microsoft’s web application server. IIS is available only for Windows and works best on Windows Server. In embracing ASP.NET, you are adopting an entire stack of tools from Microsoft. They are pretty good tools, but there isn’t any diversity available.

If you have done any other kind of .NET development, these themes will make it easier for you to learn ASP.NET programming. The tools are very similar, the approach to creating and managing projects is the same, and, of course, C# is a familiar language when it comes to writing the code segments of a web application.

### The Structure of ASP.NET

ASP.NET has evolved since it was first released, and the importance of the Web Forms programming model has lessened. We can tease ASP.NET into two sections: Web Forms and the core platform, as illustrated in Figure 3-1.

![Figure 3-1. Separating ASP.NET into the core platform and Web Forms](image)