Module 1

Exploring ASP.NET Core MVC

Module Overview

- Overview of Microsoft Web Technologies
- Overview of ASP.NET 4.x
- Introduction to ASP.NET Core MVC

Lesson 1: Overview of Microsoft Web Technologies

- Introduction to Microsoft Web Technologies
- Overview of ASP.NET
- Client-Side Web Technologies
- Hosting Technologies

Introduction to Microsoft Web Technologies

Develop

- Visual Studio
- Visual Studio Code

Host

- IIS
- Microsoft Azure

Execute

Server-Side

- ASP.NET Core
- ASP.NET 4.x

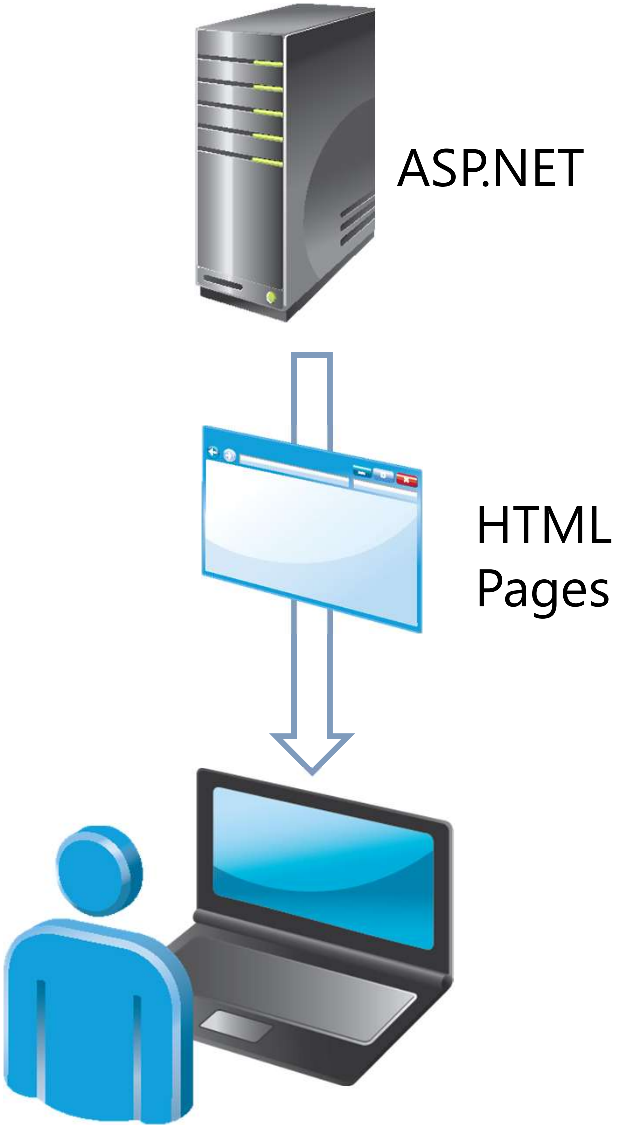
Client-Side

- JavaScript
- jQuery
- Angular
- React
- AJAX

Overview of ASP.NET

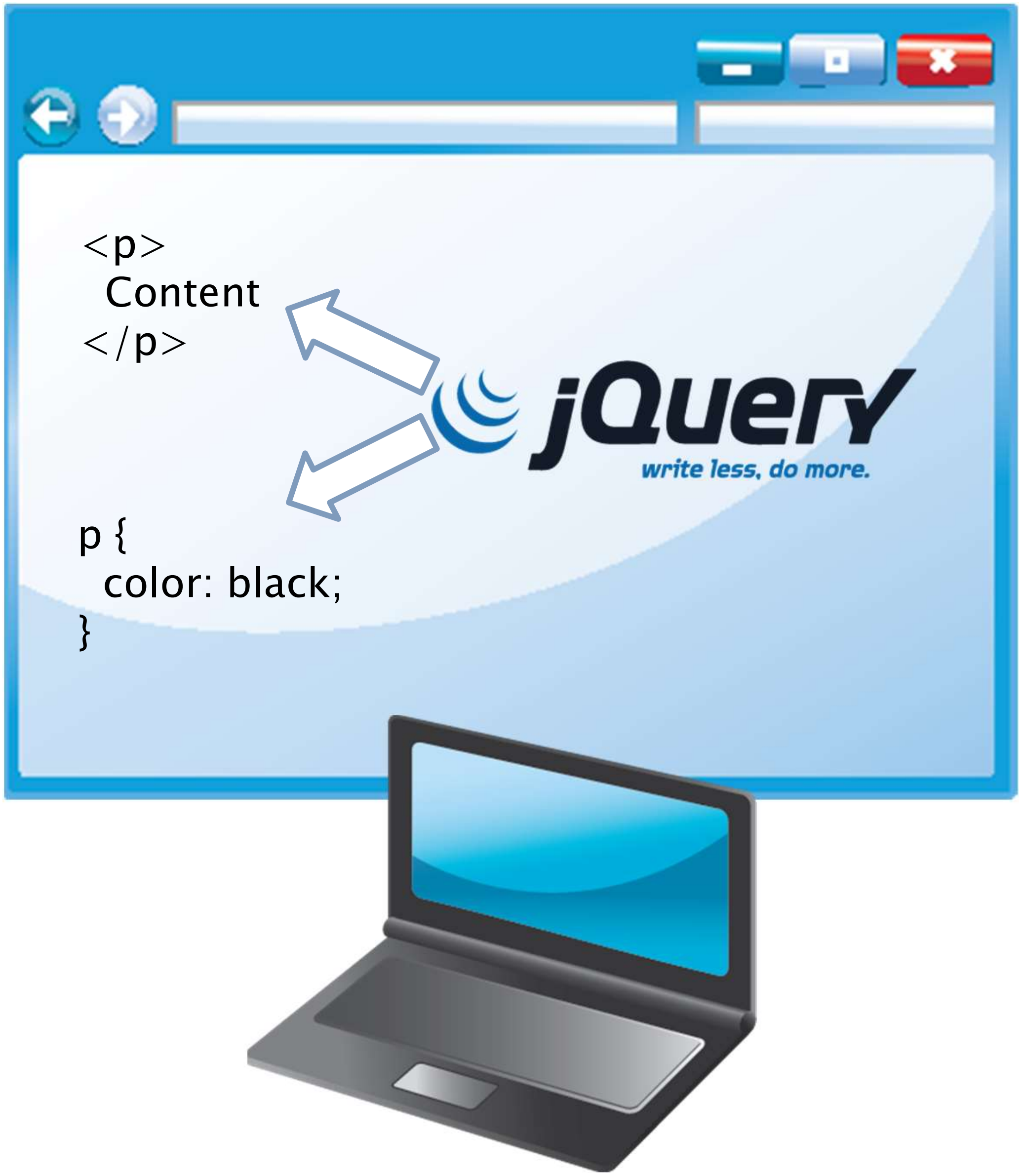
Programming Models:

- ASP.NET 4.x:
  - Web Pages
  - Web Forms
- ASP.NET 4.x and ASP.NET Core:
  - MVC
  - Web API
- ASP.NET Core:
  - Razor Pages



Client-Side Web Technologies

- JavaScript
- jQuery
- AJAX
- Angular
- React
- And more



```
<p>
  Content
</p>
```

```
p {
  color: black;
}
```

Hosting Technologies

- IIS
  - Features
  - Scaling
  - Perimeter Networks
- IIS Express
- Other Web Servers

Microsoft Azure

- Cloud computing provides scalability, flexibility, security, and reliability
- The Microsoft Azure platform includes:
  - Web Apps
  - Databases (Azure SQL Database, Cosmos DB)
  - Virtual Machines
  - Mobile Apps
  - Media Services

Lesson 2: Overview of ASP.NET 4.x

- Overview of Web Pages
- Overview of Web Forms
- Overview of MVC
- Discussion: ASP.NET 4.x Application Scenarios
- Shared ASP.NET 4.x Features
- Overview of Web API

Overview of Web Pages

- Code in .CSHTML files
- Precise control over HTML

```
<h2>Special Offers</h2>
<p>Get the best possible value on Northwind specialty
   foods by taking advantage of these offers:</p>
@foreach (var item in offers) {
  <div class="offer-card">
    <div class="offer-picture">
      @if (!String.IsNullOrEmpty(item.PhotoUrl)){
        <img src="@Href(item.PhotoUrl) alt="@item.Title"  />
      }
    </div>
  </div>
}
```

Overview of Web Forms

- Code in .aspx files and code-behind files
- Create a UI by dragging controls onto a page
- Controls provide rich properties and events
- Bind controls to data

Overview of MVC

- Models encapsulate objects and data
- Views generate the user interface
- Controllers interact with user actions
- Code in .cshtml and .cs files
- Precise control of HTML and URLs
- Easy to use unit tests

Discussion: ASP.NET 4.x Application Scenarios

Which programming model will you use in the following scenarios? (Choose one of the following in each scenario: MVC, Web Forms, Web Pages)

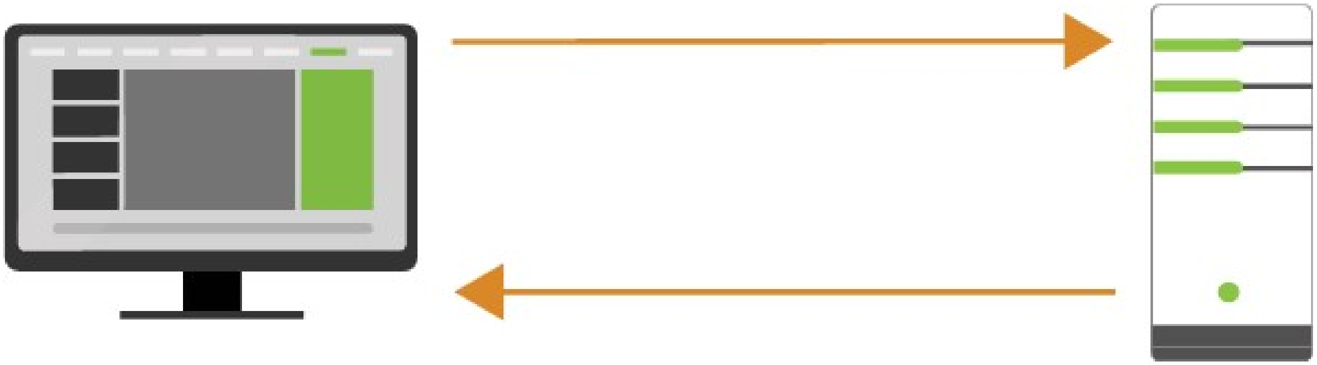
- A database front-end to be hosted on an intranet
- An e-commerce site for a large software organization
- A website for a small charitable trust

Shared ASP.NET 4.x Features

- Configuration
- Authentication
- Membership and Roles
- State Management
- Caching

Overview of Web API

- Helps creating RESTful APIs
- Enables external systems to use your application's business logic
- Accessible to various HTTP clients
- Helps to obtain data in different formats such as JSON and XML
- It supports create, read, update and delete (CRUD) actions
- Ideal for mobile application integration



Lesson 3: Introduction to ASP.NET Core MVC

- Introduction to ASP.NET Core
- Discussion: Choose between ASP.NET 4.x and ASP.NET Core
- Choose between .NET Core and .NET Framework
- Models, Views, and Controllers
- Demonstration: How to Explore an ASP.NET Core MVC Application

Introduction to ASP.NET Core

- ASP.NET Core is:
  - Modern redesign of ASP.NET 4.x
  - Cross-platform and open source
  - Lean, high-performance and modular framework
  - Cloud-ready
- Supports the following programming models:
  - MVC
  - Razor pages
  - Web API

Razor Pages

- Alternative to the MVC programming model
- Starts with the **@page** directive

```
 @page
@model HomePageModel

<h1>@Model.Title</h1>
<h2>@Model.Description</li>
```

Discussion: Choose between ASP.NET 4.x and ASP.NET Core

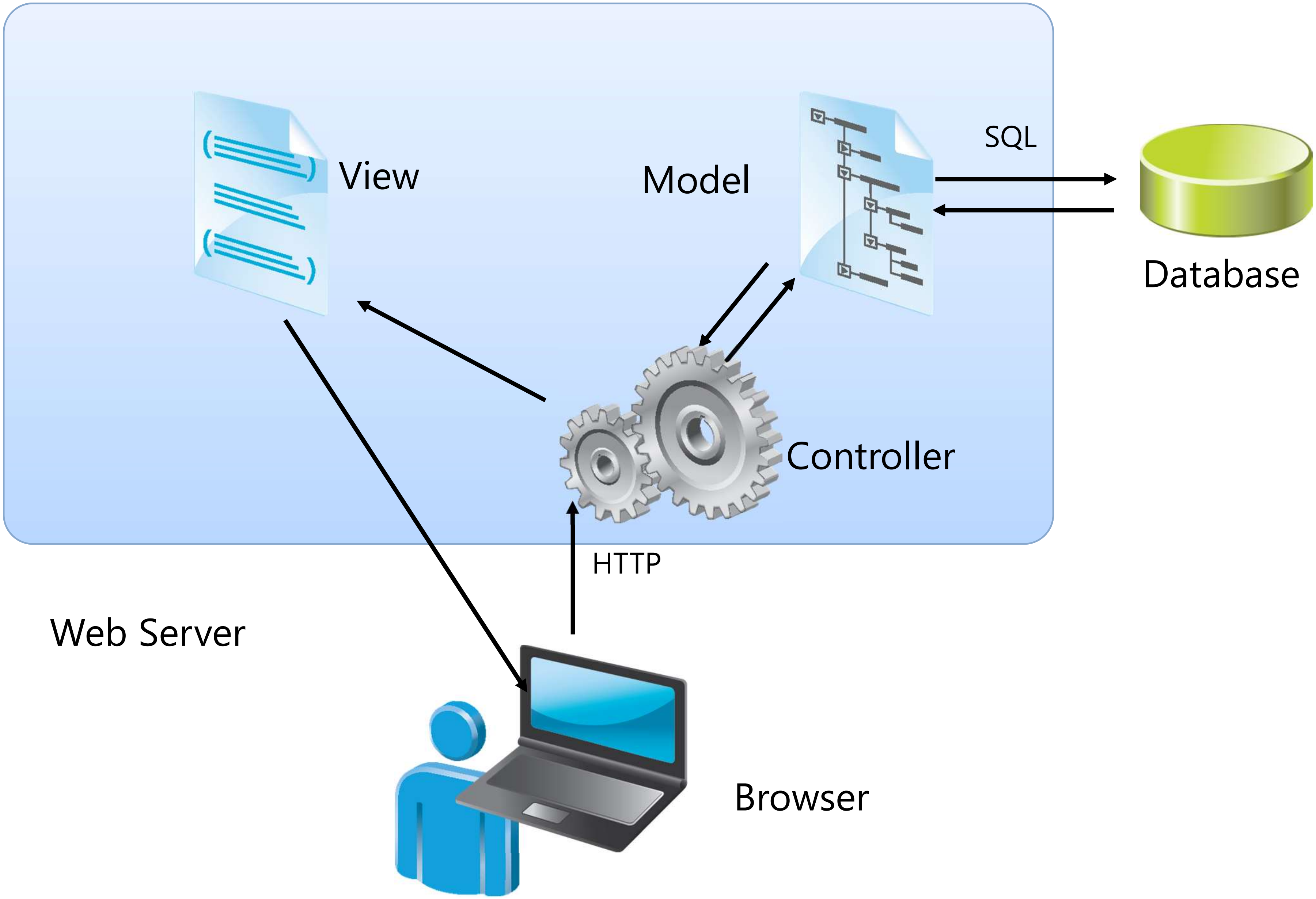
Which version of ASP.NET (ASP.NET 4.x or ASP.NET Core) will you use in the following scenarios?

- High performing application for the stock market
- Update an existing website written in Web Forms
- Open-source project running on macOS

Choose between .NET Core and .NET Framework

- You should use .NET Core when:
  - You want your code to run cross-platform
  - You want to create microservices
  - You want to use Docker containers
  - You want to achieve a high-preforming scalable system
- You should use .NET Framework when:
  - You want to extend an existing application that uses .NET Framework
  - You want to use NuGet packages or third-party .NET libraries that are not supported in .NET Core
  - You want to use .NET technologies that aren't supported in .NET Core
  - You want to use a platform that doesn’t support .NET Core

Models, Views, and Controllers



Demonstration: How to Explore an ASP.NET Core MVC Application

In this demonstration, you will:

1. Examine an ASP.NET Core MVC application that renders the default home page
2. Examine the Index View code
3. Examine the default route used to map requests to the controller
4. Examine the Home Controller code
5. Examine the Photo Model code
6. Examine the photo list rendered as a result of models, controllers, and views working together

Lab: Exploring ASP.NET Core MVC

- Exercise 1: Exploring a Razor Pages Application
- Exercise 2: Exploring a Web API Application
- Exercise 3: Exploring an MVC Application

Estimated Time: 90 minutes

Lab Scenario

You are working as a junior developer at Adventure Works. You have been asked by a senior developer to investigate the possibility of creating a web-based ASP.NET Core MVC application for your organization's customers, similar to the one that the senior developer has seen on the internet. Such an application will promote a community of cyclists who use Adventure Works equipment, and the community members will be able to share their experiences. This initiative is intended to increase the popularity of Adventure Works Cycles, and thereby to increase their sales. You have been asked to begin the planning of the application. You have also been asked to examine programming models available to ASP.NET Core developers. To do this, you need to create basic web applications using three different models: Razor Pages, Web API, and MVC.

Lab Review

- Which of the three programming models has the simplest method of applying a single layout across multiple pages?
- A member of your team replaced the line return View(model); in the Details action of the AnimalController class with the line return View();. What will happen when the Details action is called?
- When you run the CakeStoreApi application, the browser displays value1 and value2. You want to display your first name and your last name instead. What will you have to do to achieve your goal?

Module Review and Takeaways

- Review Questions
- Best Practices
- Common Issues and Troubleshooting Tips