**TRƯỜNG ĐẠI HỌC SÀI GÒN**

**KHOA CÔNG NGHỆ THÔNG TIN**

~~~~~~~~~~~~~

**BÁO CÁO MÔN CÁC CÔNG NGHỆ LẬP TRÌNH HIỆN ĐẠI**

**ĐỀ TÀI: TÌM HIỂU VÀ ỨNG DỤNG CÔNG NGHỆ ASP.NET ĐỂ XÂY DỰNG**



**Giảng viên hướng dẫn:** Phạm Thi Vương

**Nhóm:**

Danh sách sinh viên thực hiện:

Dương Minh Trí – 3122410420

Đinh Bá Phong – 3122410305

Vũ Đăng Khoa – 3122410188

Đoàn Phong Lưu – 3122410227

Thành phố Hồ Chí Minh, tháng 9/2025

**NHẬN XÉT ĐÁNH GIÁ CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**LỜI CẢM ƠN.**

- Lời đầu tiên nhóm em xin gửi lời biết ơn sâu sắc đến thầy **Phạm Thi Vương**, người đã là nguồn động viên và hỗ trợ không ngừng trong quá trình thực hiện đề tài và báo cáo này. Những sự chỉ dạy của thây không chỉ là những lời khuyên có giá trị tận tâm và từ kinh nghiệm, thầy luôn sẵn lòng giải đáp mọi thắc mắc, đưa ra những góp ý xây dựng, và chỉnh sửa linh hoạt để giúp nhóm em vượt qua những khó khăn và hoàn thành đề tài đúng theo tiến độ.

- Ngoài ra nhóm em cũng bày tỏ lòng biết ơn đến Ban Giám Hiệu Trường Đại Học Sài Gòn, người đã tạo điều kiện thuận lợi cho quá trình nghiên cứu và tìm kiếm thông tin. Sự hỗ trợ và tận tâm của quý Thầy Cô trong Khoa Đào Tạo và đặc biệt là ngành Công Nghệ Thông Tin đã đóng góp quan trọng để nhóm em có được nền tảng kiến thức vững chắc để thực hiện đề tài môn Kiểm thử phần mềm.

- Dù đề tài và báo cáo được thực hiện với kiến thức còn hạn chế và nhiều hạn chế kỹ thuật, nhóm đều tự nhận thức được những điều này, tuy nhiên nhóm đã cố gắng cùng nhau hoành thành đồ án một cách tốt nhất và cố gắng bù đắp cho nhau những thiếu sót. Nhóm em mong nhận được sự góp ý quý báu của thầy để kiến thức của chúng em được hoàn thiện hơn và chúng em có thể làm tốt hơn nữa trong những lần sau. Lời cuối cùng, nhóm em xin chúc thầy luôn khỏe mạnh, thành công và hạnh phúc trong mọi công việc.

Nhóm em xin chân thành cảm ơn.

MỤC LỤC chỗ này

**PHẦN 1: TỔNG QUAN VỀ CÔNG NGHỆ ASP.NET**

**Chương 1: Giới thiệu chung**

1. **Lịch sử hình thành và phát triển:**

* Tổ chức tạo ra: Microsoft.
* Thời điểm ra đời: Lần đầu được công bố vào năm 2002, mục đích để thay thế ASP (Active Server Pages).
* Bối cảnh và lý do ra đời:
  + ASP cũ chạy trên nền script engine, hiệu năng thấp, khó mở rộng.
  + Microsoft muốn xây dựng một framework mạnh mẽ hơn, hướng đối tượng, tích hợp chặt chẽ với .NET Framework để phát triển ứng dụng web hiện đại.
  + ASP.NET ra đời nhằm hỗ trợ phát triển web động, bảo mật, dễ bảo trì, có khả năng mở rộng tốt.

1. **Sự thay đổi qua các phiên bản:**

* **ASP.NET 1.0 (2002)**: Ra mắt cùng .NET Framework 1.0, hỗ trợ Web Forms.
* **ASP.NET 2.0 (2005)**: Cải tiến Web Forms, thêm Membership, Master Pages.
* **ASP.NET 3.5 (2007)**: Hỗ trợ AJAX, LINQ.
* **ASP.NET MVC (2009)**: Giới thiệu mô hình MVC, tách biệt logic và giao diện.
* **ASP.NET Web API (2012)**: Tối ưu cho xây dựng RESTful API.
* **ASP.NET Core (2016, tái cấu trúc lớn)**:
  + Mã nguồn mở.
  + Đa nền tảng (Windows, Linux, macOS).
  + Hiệu năng cao, kiến trúc module gọn nhẹ.
* **ASP.NET Core 5/6/7/8 (2020–2023)**: Hợp nhất với .NET 5+ (không còn tách .NET Framework/Core), hỗ trợ Minimal API, cải thiện hiệu năng, tích hợp container và cloud-native.

1. Hệ sinh thái:

* **Thư viện, framework, công cụ hỗ trợ**:
  + Entity Framework Core (ORM).
  + Blazor (SPA framework chạy trên WebAssembly hoặc server).
  + SignalR (real-time communicastion).
  + Identity (xác thực & phân quyền).
  + Visual Studio, Visual Studio Code.
* **Cộng đồng phát triển**:
  + Rất lớn, chủ yếu tập trung ở Microsoft Docs, GitHub, Stack Overflow.
  + Hỗ trợ chính thức từ Microsoft, cộng đồng đóng góp mạnh mẽ (ASP.NET Core open-source).

**Chương 2: Cơ hội nghề nghiệp và thị trường**

1. **Các vị trí công việc:**

* **ASP.NET Developer / .NET Developer**: Phát triển web/app với ASP.NET Core, Web API, Blazor.
* **Fullstack .NET Developer**: Kết hợp backend ASP.NET với frontend (Angular, React, Blazor).
* **DevOps Engineer (chuyên .NET)**: Triển khai, CI/CD cho ứng dụng ASP.NET Core trên Azure, AWS.
* **Software Engineer (C#/.NET)**: Phát triển sản phẩm tổng hợp (web, desktop, cloud).

1. **Nhu cầu tuyển dụng:**

**Hiện tại (2025):**

* Ở Việt Nam, ASP.NET Core nằm trong top 5 công nghệ backend được tuyển dụng nhiều (theo TopCV, ITviec).
* Thường xuất hiện trong job description yêu cầu kết hợp SQL Server, Azure, và frontend framework.

**Tương lai gần:**

* Nhu cầu ổn định, đặc biệt ở các doanh nghiệp lớn, ngân hàng, công ty outsource cho thị trường Âu-Mỹ.
* ASP.NET Core sẽ duy trì do Microsoft tiếp tục đầu tư mạnh mẽ.

1. **Mức lương tham khảo:**

**Ở Việt Nam, năm 2025:**

* Junior .NET Developer: 10–15 triệu VNĐ/tháng.
* Middle .NET Developer: 20–30 triệu VNĐ/tháng.
* Senior .NET Developer: 35–50 triệu VNĐ/tháng, có thể cao hơn tại các công ty quốc tế hoặc Fintech.

1. **Các công ty/sản phẩm tiêu biểu đang sử dụng:**

* **Toàn cầu**: Stack Overflow, GoDaddy, Dell, Visual Studio Team Services.
* **Việt Nam**: Nhiều ngân hàng (Techcombank, Vietcombank), tập đoàn lớn (FPT, Viettel), công ty outsource (TMA Solutions, NashTech).

**PHẦN 2:  NỘI DUNG CỐT LÕI VÀ THỰC HÀNH**

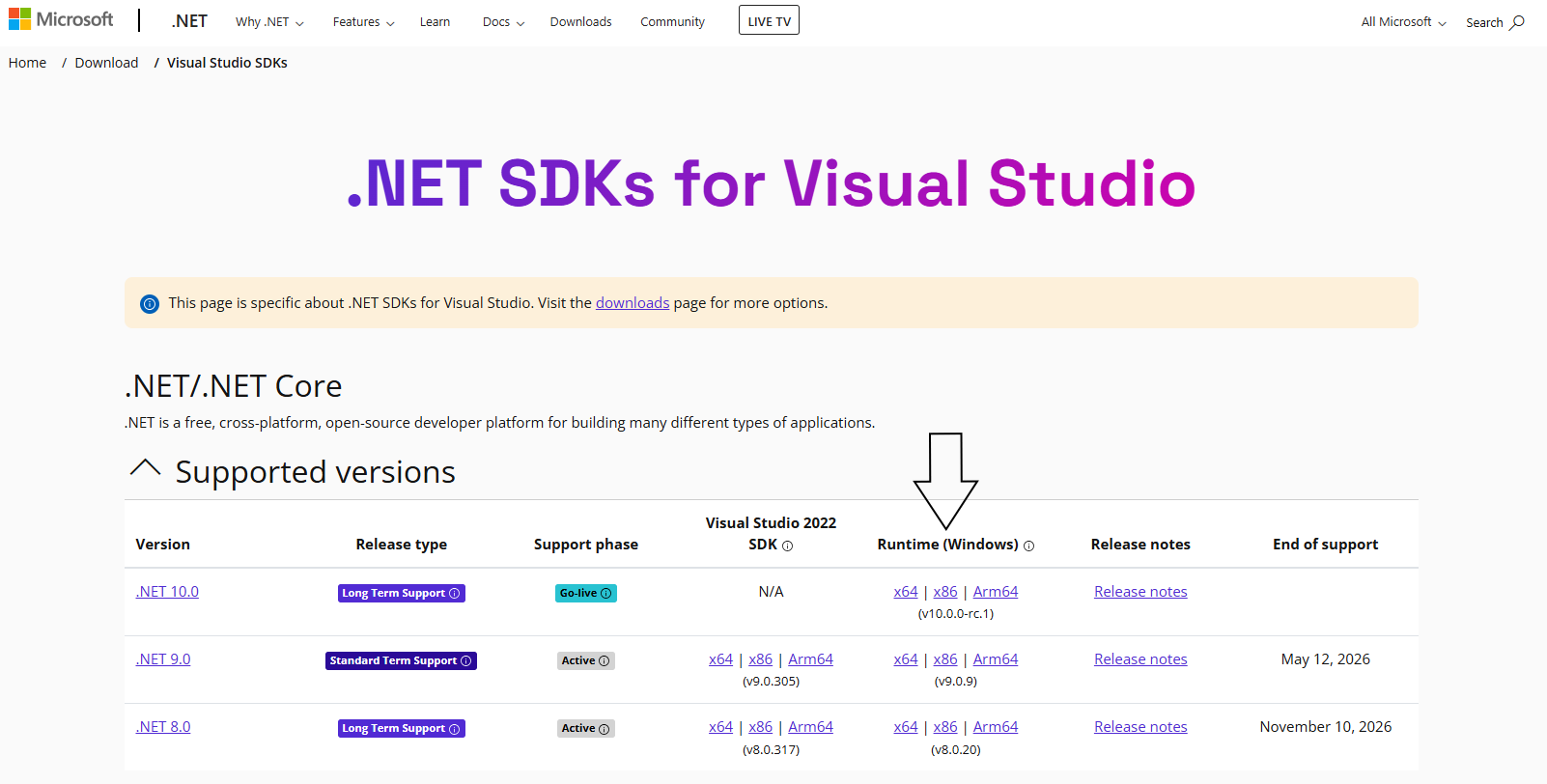
**Chương 3: Cài đặt môi trường và “Hello World”**

1. **Lý thuyết:**

* Các công cụ cần thiết để bắt đầu:
  + .NET SDK
  + Visual Studio Code.
    1. Sau khi cài xong Visual Studio Code, ta sẽ tải các extension cần thiết là: C# Extension, C# Dev Kit và .NET Runtime Install Tool.

1. **Hướng dẫn:**

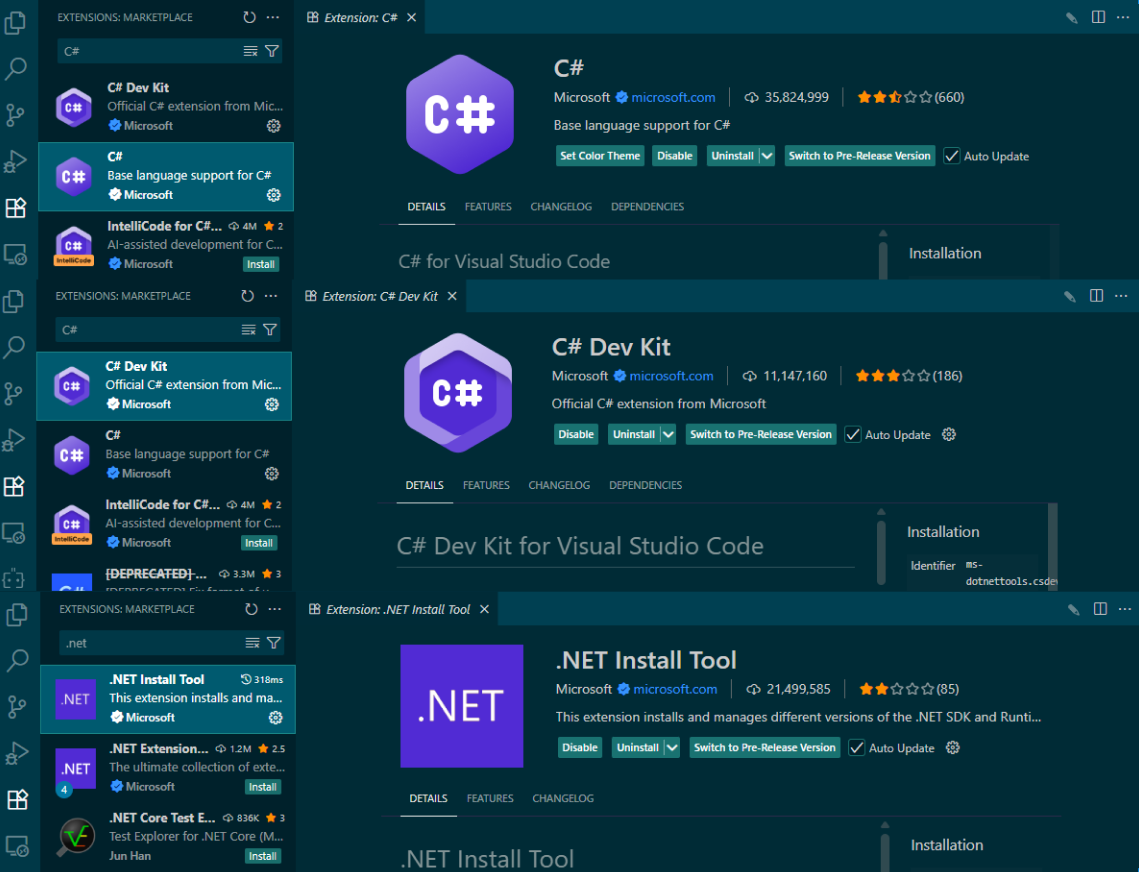
* Bước 1: Tải và cài đặt .NET SDK ở trang chủ chính thức của Microsoft: <https://dotnet.microsoft.com/en-us/download/visual-studio-sdks>, tải bộ cài đúng theo phiên bản hệ điều hành của mình.



* Bước 2: Tải và cài đặt Visual Studio Code ở trang chủ chính thức của Microsoft: <https://code.visualstudio.com/Download>, chọn đúng phiên bản theo hệ điều hành của mình.



* Bước 3: Tải các extensions C# Extension, C# Dev Kit và .NET Runtime Install Tool.



1. **Thực hành:**

Các bước chạy dự án Hello World ASP.NET đầu tiên như sau:

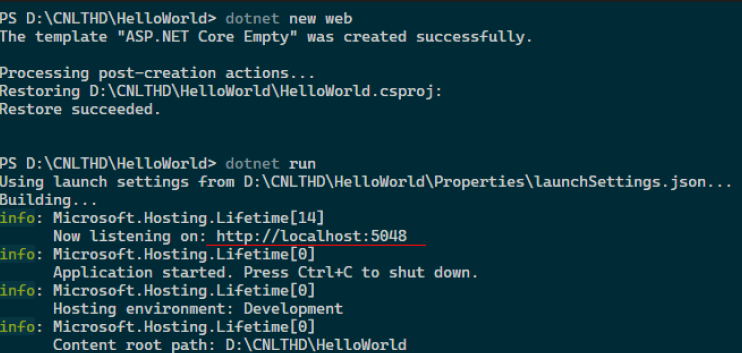
* Chạy các mã lệnh sau ở Terminal:

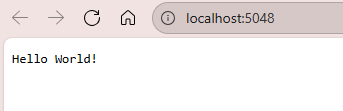
mkdir HelloAspNet && cd HelloAspNet

dotnet new web

dotnet run

* Nếu làm đúng các bước, Visual Studio Code sẽ thông báo building thành công và đưa lại đường dẫn mà khi mở ta có thể thấy được “Hello World!”





* Phân tích cấu trúc thư mục dự án vừa tạo:
  + bin/ và obj/ là nơi chứa các file chạy sau khi build và nơi chứa file tạm và metadata trong quá trình build.
  + Program.cs: Entry point, định nghĩa route, middleware, app.Run().
  + .csproj: Khai báo framework (.NET 9.0), package, tùy chọn build.
  + appsettings.json: Cấu hình mặc định (logging, connection string, host).
  + appsettings.Development.json: Ghi đè khi môi trường Development.
  + Properties/launchSettings.json: Cấu hình debug (port, env, auto mở browser).

**Chương 4: Các khái niệm cơ bản**

**4.2. Cấu trúc ứng dụng ASP.NET**

**4.2.1. Sơ đồ thư mục (ASP.NET Core MVC – mẫu điển hình)**

MyApp/

├─ Controllers/

│ ├─ HomeController.cs

│ └─ ProductsController.cs

├─ Models/

│ └─ Product.cs

├─ Views/

│ ├─ Shared/

│ │ ├─ \_Layout.cshtml

│ │ └─ \_ValidationScriptsPartial.cshtml

│ ├─ Home/

│ │ └─ Index.cshtml

│ └─ Products/

│ ├─ Index.cshtml

│ └─ Details.cshtml

├─ wwwroot/ // css, js, img tĩnh

│ ├─ css/site.css

│ └─ js/site.js

├─ appsettings.json

├─ Program.cs // .NET 6+ (Minimal hosting)

├─ Startup.cs // .NET 3.1/5 (kiểu hosting cũ, nếu dùng)

└─ MyApp.csproj

**Ý nghĩa các thư mục/chính**

* **Controllers/**: Lớp điều khiển (nhận request, gọi dịch vụ/Model, trả View/JSON…).
* **Models/**: Lớp miền nghiệp vụ (entity/DTO/ViewModel/Validation).
* **Views/**: Razor \*.cshtml hiển thị UI; cấu trúc theo tên controller.
* **Views/Shared/**: layout, partial, view dùng chung.
* **wwwroot/**: web root cho file tĩnh (CSS/JS/img).

**4.2.2. Program.cs / Startup.cs / Global.asax**

**a) Program.cs (ASP.NET Core .NET 6+ – Minimal hosting)**



**b) Startup.cs (ASP.NET Core 3.1/5 – kiểu cũ)**



**c) Global.asax (ASP.NET MVC5 – .NET Framework)**



**4.2.3. appsettings.json và đọc cấu hình**



**Đọc cấu hình qua IConfiguration hoặc IOptions<T>**



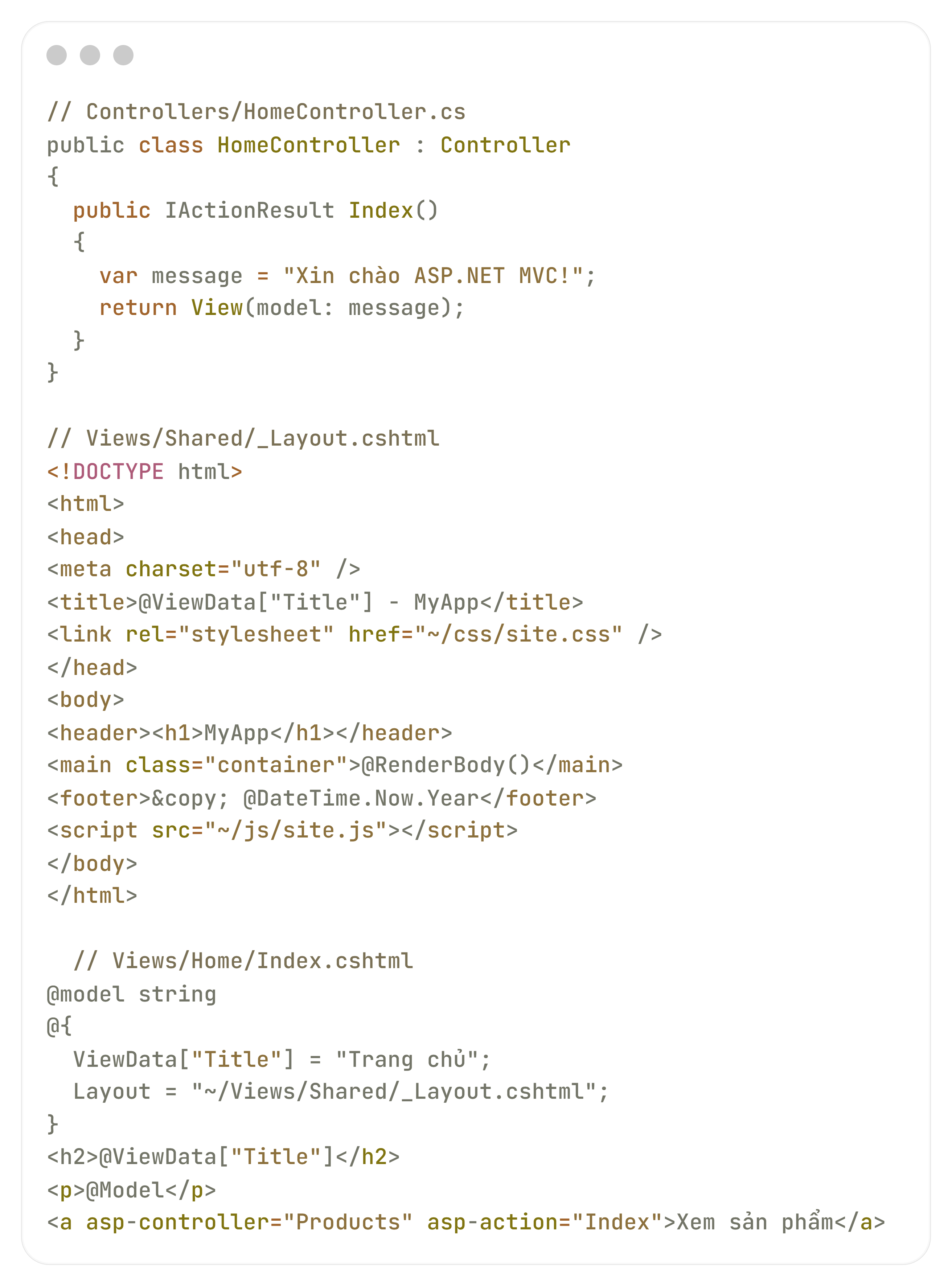
**4.2.4. Middleware.**

* **Middleware**: thành phần trong pipeline xử lý HTTP; có thể can thiệp trước/sau khi gọi next().
* **Thứ tự** là **bắt buộc** (ví dụ UseRouting phải trước UseAuthorization).



Đặt middleware logging **sau** UseRouting nếu muốn log thông tin endpoint; **trước** UseStaticFiles nếu không cần log file tĩnh.

**4.2.5. View, Layout và Tag Helper**



**4.3. Routing cơ bản**

**4.3.1. Khái niệm & thuật ngữ**

* **Route**: mẫu ánh xạ URL → **endpoint** (Controller/Action hoặc handler).
* **Token**: {controller}, {action}, {id?}…;
* **Constraints**: ràng buộc kiểu/định dạng, ví dụ int, guid, min, regex…
* **Conventional routing** vs **Attribute routing**: định nghĩa mẫu chung một lần **so với** gắn attribute trực tiếp lên controller/action.

**4.3.2. Conventional Routing (route mặc định & route bổ sung)**

**Định tuyến mặc định (Program.cs – .NET 6+)**

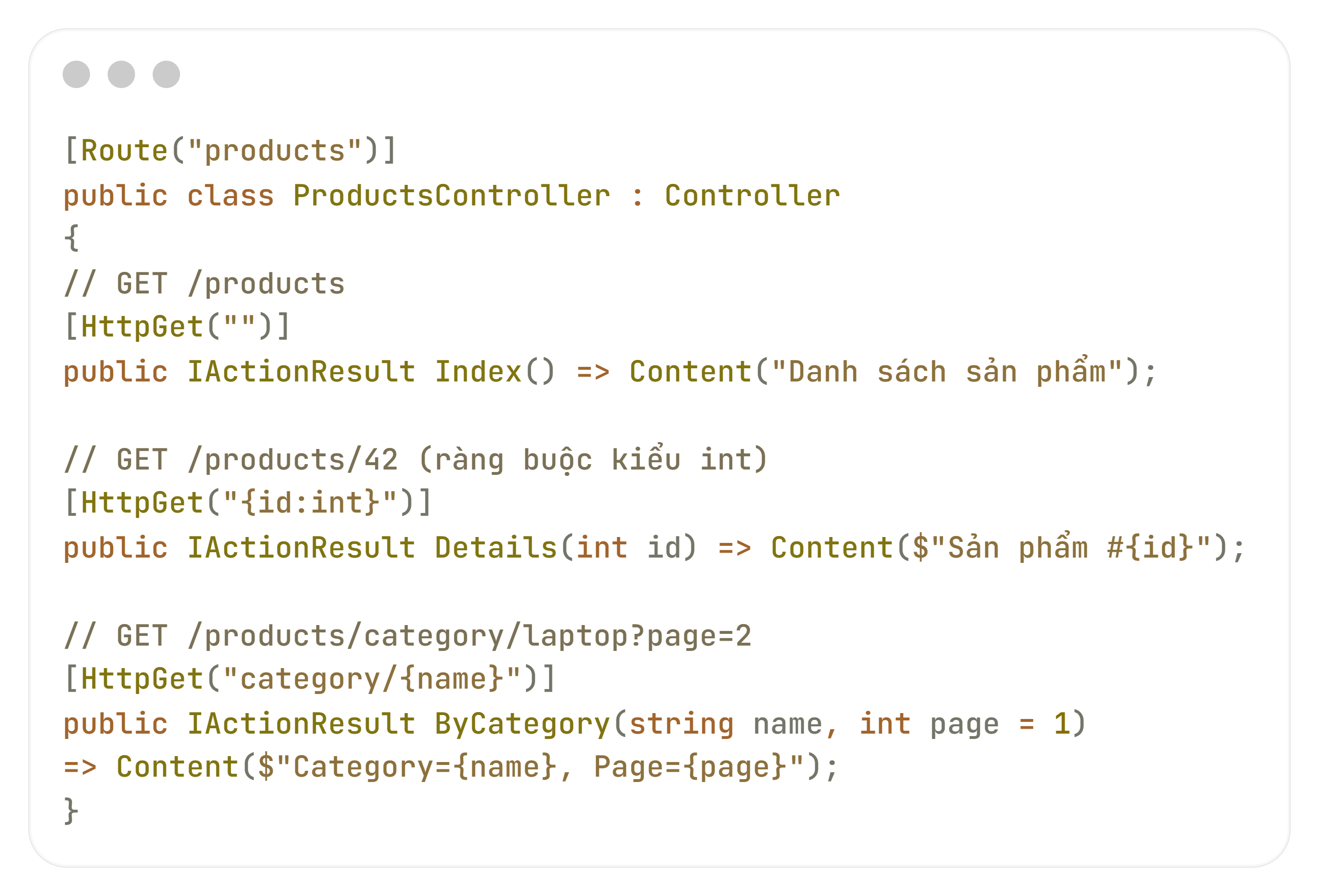
* / → HomeController.Index()
* /Products/Details/5 → ProductsController.Details(5)

**Thêm route tùy biến (đặt TRƯỚC route mặc định để ưu tiên)**

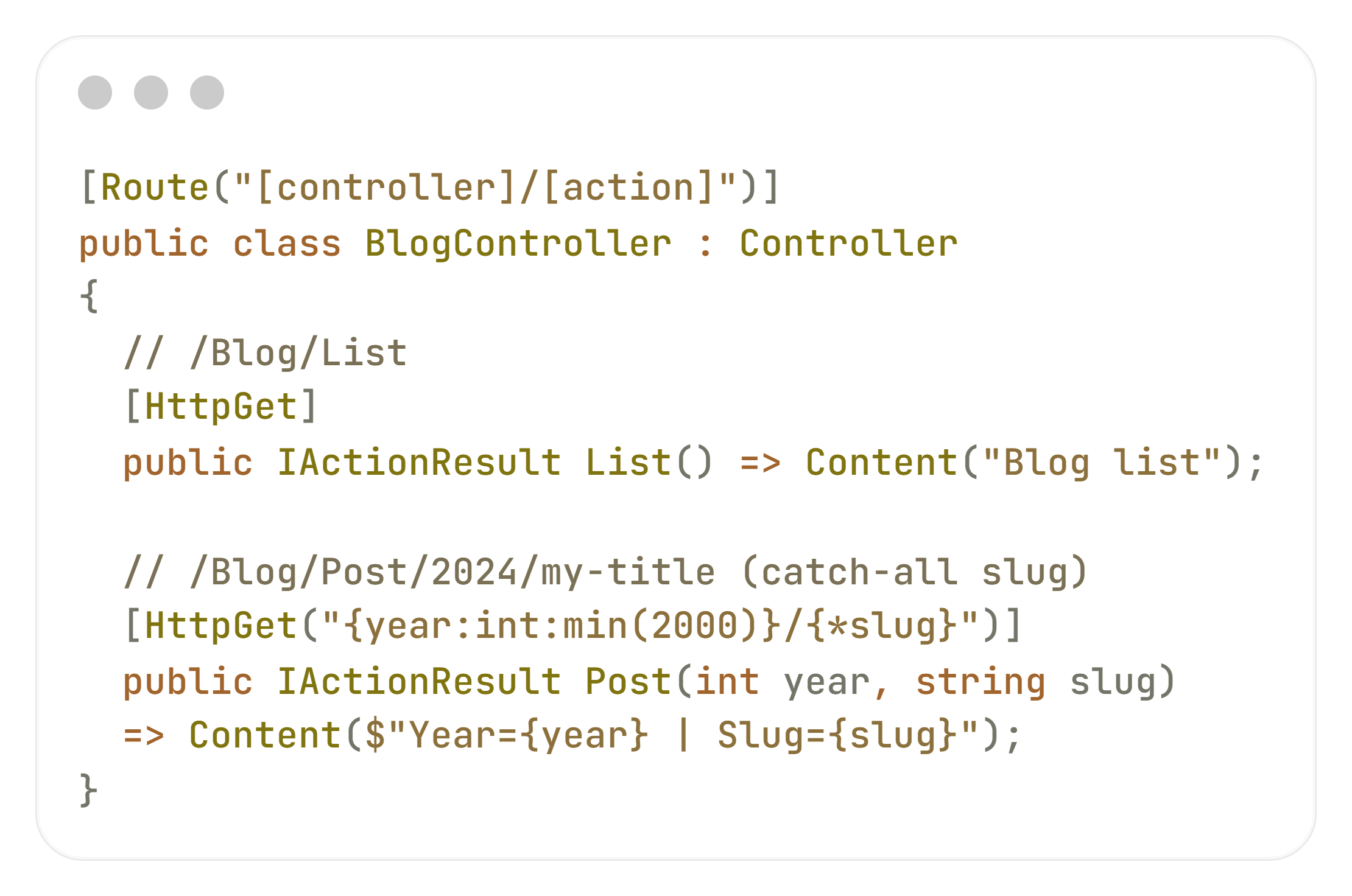
**Nguyên tắc**: Conventional routes được xét **theo thứ tự khai báo**. Đặt các route cụ thể nằm **trước** route tổng quát.

**4.3.3. Attribute Routing (linh hoạt theo controller/action)**

**Khai báo ở mức controller + action**



**Sử dụng token tự động**

**Bật attribute routing**: khi dùng AddControllers() hoặc AddControllersWithViews(), chỉ cần ánh xạ endpoint phù hợp:

* Dùng **conventional**: app.MapControllerRoute(...) (có thể **đồng thời** dùng attribute routing).
* Hoặc ánh xạ tất cả controller theo attribute: app.MapControllers();

**4.3.4. Ràng buộc (Constraints), tùy chọn, mặc định, catch-all**

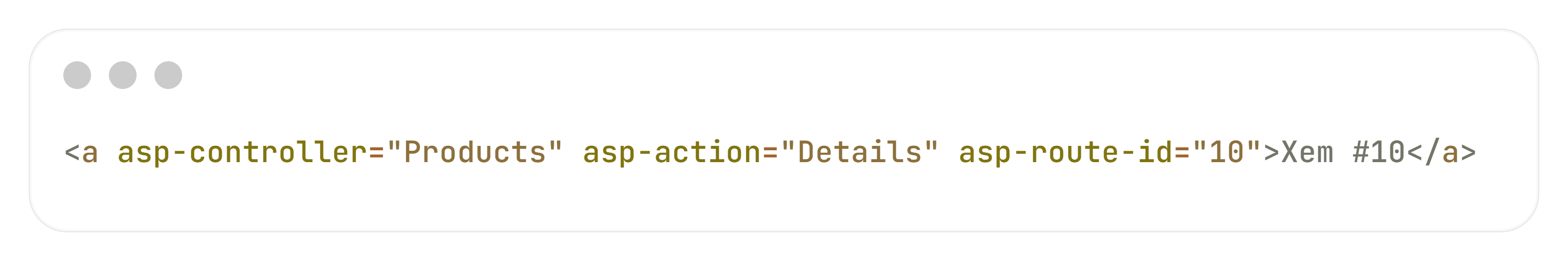
* **Kiểu**: {id:int}, {key:guid}, {code:length(6)}
* **Giới hạn**: {year:int:min(2000):max(2100)}
* **Regex**: {slug:regex(^[a-z0-9-]+$)}
* **Tùy chọn**: {id?}
* **Giá trị mặc định**: defaults: new { page = 1 } hoặc [HttpGet("{page:int=1}")]
* **Bắt tất cả phần còn lại**: {\*path} hoặc {\*\*slug} (từ .NET 8 có {\*\*rest} thân thiện hơn cho nested path)

**4.3.5. Sinh URL & Tag Helpers (liên kết đúng route)**

* **Url.Action** (server-side):



* **Anchor Tag Helper** (trong Razor):



* **Đặt tên route** để sinh URL ổn định:

