TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET**

**NĂM HỌC 2024-2025**

**XÂY DỰNG WEBSITE BÁN LAPTOP**

*Giảng viên hướng dẫn:*

TS. Đoàn Phước Miền

*Sinh viên thực hiện:*

Họ tên: Vũ Văn Hòa

MSSV: 170124242

Lớp: DK24TTC2

***Vĩnh Long, tháng 11 năm 2025***

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN MÔN HỌC: CHUYÊN ĐỀ ASP.NET**

**NĂM HỌC 2024-2025**

**XÂY DỰNG WEBSITE BÁN LAPTOP**

*Giảng viên hướng dẫn:*

TS. Đoàn Phước Miền

*Sinh viên thực hiện:*

Họ tên: Vũ Văn Hòa

MSSV: 170124242

Lớp: DK24TTC2

***Vĩnh Long, tháng 11 năm 2025***

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*…………, ngày ….. tháng ….. năm ……..*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*……….., ngày …… tháng …… năm …….*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

**LỜI CẢM ƠN**

Tôi xin cảm ơn quý thầy cô Trường Đại học Trà Vinh đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình học tập, cũng như trong quá trình làm báo cáo. Tôi xin cảm ơn thầy, cô trong khoa Kỹ thuật và Công nghệ đã tận tình dạy bảo, giúp đỡ cho tôi về kiến thức chuyên môn lẫn kinh nghiệm sống để trở thành một người có ích cho xã hội.

Cảm ơn những người bạn, anh, chị làm việc chung đã giúp đỡ tôi trong những ngày thực hiện bài báo cáo.

Đặc biệt, tôi gửi lời cảm ơn chân thành đến thầy Đoàn Phước Miền đã tận tình hướng dẫn, truyền đạt kiến thức cũng như tạo điều kiện thuận lợi cho tôi hoàn thành tốt bài báo cáo này.

Với vốn kiến thức đã được truyền tải trong quá trình học tập và tìm hiểu, đặc biệt là được sự hướng dẫn của thầy Đoàn Phước Miền đã giúp cho tôi hoàn thành đồ án cơ sở ngành. Trong quá trình làm bài có thể tôi sẽ mắc phải một số thiếu sót nên rất mong nhận được những đóng góp ý kiến của thầy, cô để có thể khắc phục và phát triển từng bước hoàn thiện hơn.

Xin chân thành cảm ơn!

*Sinh viên ký và ghi rõ họ và tên*

**MỤC LỤC**

**DANH MỤC HÌNH ẢNH**

**DANH MỤC CỤM TỪ VIẾT TẮT**

**MỞ ĐẦU**

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, việc ứng dụng công nghệ thông tin vào hoạt động kinh doanh không còn là xu hướng, mà đã trở thành nhu cầu thiết yếu. Đặc biệt, thương mại điện tử đang ngày càng khẳng định vai trò quan trọng trong việc kết nối doanh nghiệp với khách hàng, mở rộng thị trường và nâng cao hiệu quả kinh doanh.

Bên cạnh đó, việc xây dựng các hệ thống website bán hàng chuyên nghiệp giúp doanh nghiệp không chỉ giới thiệu sản phẩm một cách trực quan, tiện lợi mà còn hỗ trợ quản lý dữ liệu, đơn hàng, và chăm sóc khách hàng hiệu quả hơn. Trong bối cảnh đó, ASP.NET – một nền tảng phát triển web mạnh mẽ do Microsoft cung cấp – trở thành công cụ lý tưởng để xây dựng các ứng dụng web động, bảo mật và dễ bảo trì.

Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài “Xây dựng website bán laptop sử dụng ASP.NET” nhằm mục tiêu tìm hiểu và ứng dụng các kiến thức đã học vào việc thiết kế và xây dựng một hệ thống website thương mại điện tử cơ bản. Thông qua chuyên đề này, chúng em mong muốn có cơ hội tiếp cận sâu hơn với quy trình phát triển web, hiểu rõ cách tổ chức dữ liệu, xử lý nghiệp vụ và tích hợp giao diện người dùng thân thiện.

Báo cáo chuyên đề là kết quả của quá trình học tập, nghiên cứu và thực hành nghiêm túc. Mặc dù đã cố gắng hoàn thiện, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Kính mong quý thầy cô góp ý để em có thể rút kinh nghiệm và nâng cao kiến thức trong thời gian tới.

**CHƯƠNG 1: TỔNG QUAN**

**1.1 Lý do chọn đề tài**

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, công nghệ thông tin đóng vai trò then chốt trong hầu hết các lĩnh vực, đặc biệt là thương mại điện tử. Việc xây dựng các website bán hàng không chỉ giúp doanh nghiệp giới thiệu sản phẩm đến khách hàng một cách nhanh chóng, tiện lợi mà còn hỗ trợ hiệu quả trong quản lý kinh doanh và chăm sóc khách hàng.

Lĩnh vực kinh doanh laptop là một trong những ngành hàng công nghệ phát triển nhanh và có tính cạnh tranh cao. Việc ứng dụng một hệ thống website thương mại điện tử chuyên nghiệp giúp các cửa hàng, doanh nghiệp tăng khả năng tiếp cận khách hàng, mở rộng thị trường, đồng thời tạo điều kiện thuận lợi cho người tiêu dùng trong việc tìm kiếm và mua sắm sản phẩm.

Đối với các công ty thì việc xây dựng các website riêng càng ngày càng trở nên cấp thiết. Việc xây dựng các website giúp cho các công ty phát triển một cách nhanh chống đặc biệt là các công ty bán hàng và quản lý bán hàng. Hoạt động của một công ty, cửa hàng có quy mô khá lớn sẽ càng được tăng cường và mở rộng nếu xây dựng được một website tốt. Bắt nguồn với ý tưởng này, cùng với những gợi ý của Giảng viên Đoàn Phước Miền, chúng em đã thực hiện Bài tập “Xây dựng website bán Laptop”.

**1.2 Mục tiêu nghiên cứu**

Vận dụng kiến thức đã học để xây dựng một website bán hàng hoàn chỉnh: Ứng dụng lý thuyết về lập trình web, cơ sở dữ liệu, mô hình MVC, HTML/CSS, ngôn ngữ C# và công nghệ ASP.NET để phát triển một hệ thống website thương mại điện tử thực tế.

Thiết kế giao diện người dùng thân thiện và dễ sử dụng: Tạo ra giao diện website trực quan, dễ thao tác cho cả người dùng lẫn quản trị viên, giúp nâng cao trải nghiệm người dùng (UX/UI).

Xây dựng các chức năng bán hàng cơ bản: Phát triển các chức năng cần thiết của một website thương mại điện tử như: tìm kiếm sản phẩm, xem chi tiết sản phẩm, giỏ hàng, đặt hàng, quản lý đơn hàng và quản lý thông tin người dùng.

Tìm hiểu và thực hành với công nghệ ASP.NET: Nắm bắt quy trình phát triển ứng dụng web với ASP.NET, hiểu cách kết nối và thao tác với cơ sở dữ liệu SQL Server, cũng như tổ chức và xử lý nghiệp vụ trong môi trường thực tế.

**1.3 Phương pháp nghiên cứu**

* *Phương pháp nghiên cứu tài liệu*

Tìm hiểu, thu thập và tổng hợp kiến thức từ các tài liệu liên quan đến ngôn ngữ lập trình C#, công nghệ ASP.NET, mô hình MVC, HTML/CSS, JavaScript và cơ sở dữ liệu SQL Server.

Tham khảo các giáo trình, sách chuyên ngành, tài liệu online, diễn đàn lập trình, các website mẫu để học hỏi cách tổ chức, xây dựng và triển khai hệ thống website bán hàng.

* *Phương pháp phân tích hệ thống*

Phân tích yêu cầu hệ thống từ góc độ người dùng và quản trị viên để xác định các chức năng cần thiết.

Xây dựng sơ đồ chức năng, sơ đồ luồng dữ liệu (DFD), mô hình thực thể - quan hệ (ERD) để thiết kế hệ thống một cách có tổ chức và dễ triển khai.

* *Phương pháp thiết kế và lập trình*

Áp dụng quy trình phát triển phần mềm để thiết kế giao diện, xây dựng cơ sở dữ liệu và lập trình các chức năng chính của hệ thống theo mô hình phân lớp (3-tier hoặc MVC).

Sử dụng công cụ Visual Studio và hệ quản trị cơ sở dữ liệu SQL Server để phát triển và kiểm thử ứng dụng.

* *Phương pháp thực nghiệm*

Tiến hành chạy thử nghiệm hệ thống, kiểm tra tính đúng đắn của các chức năng như tìm kiếm, đặt hàng, quản lý sản phẩm và xử lý đơn hàng.

Ghi nhận các lỗi phát sinh trong quá trình sử dụng và thực hiện các bước sửa lỗi, cải tiến giao diện và tối ưu hiệu suất hệ thống.

**1.4 Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là quy trình xây dựng và phát triển một website thương mại điện tử bán laptop sử dụng công nghệ ASP.NET. Cụ thể bao gồm:

* *Người dùng cuối (khách hàng mua laptop):*

Là những người truy cập website để tìm kiếm, xem thông tin sản phẩm, đặt hàng, quản lý đơn hàng, v.v.

Nghiên cứu nhu cầu, hành vi sử dụng và trải nghiệm của người dùng để thiết kế giao diện thân thiện, dễ sử dụng.

* *Quản trị viên (admin):*

Là người có quyền quản lý nội dung website, bao gồm thêm/sửa/xóa sản phẩm, quản lý danh mục, đơn hàng và người dùng.

Nghiên cứu các tính năng và yêu cầu cần thiết để xây dựng hệ thống quản trị hiệu quả, dễ thao tác.

* *Hệ thống website bán hàng:*

Nghiên cứu cách thiết kế và xây dựng các chức năng chính của một website thương mại điện tử như: hiển thị sản phẩm, giỏ hàng, đặt hàng, đăng ký/đăng nhập người dùng, quản lý dữ liệu, kết nối cơ sở dữ liệu, v.v.

Tập trung vào kiến trúc phần mềm, tổ chức mã nguồn, giao diện người dùng và tương tác dữ liệu.

* *Công nghệ ASP.NET và các công cụ hỗ trợ:*

Nghiên cứu cách sử dụng ASP.NET (Web Forms hoặc MVC), ngôn ngữ C#, Bootstrap, HTML/CSS, JavaScript và SQL Server để phát triển website.

Khai thác các thư viện, framework và công cụ hỗ trợ nhằm tăng hiệu quả lập trình, bảo mật và tối ưu hiệu suất hệ thống.

**1.5 Phạm vi nghiên cứu**

Đề tài “Xây dựng website bán laptop sử dụng ASP.NET” được thực hiện trong phạm vi nghiên cứu và triển khai như sau:

* *Về chức năng hệ thống*

Website tập trung vào các chức năng cơ bản của một hệ thống bán hàng trực tuyến như:

Hiển thị danh sách và thông tin chi tiết sản phẩm (laptop)

Tìm kiếm và lọc sản phẩm theo tên, giá, hãng sản xuất

Quản lý giỏ hàng và đặt hàng

Đăng ký, đăng nhập và quản lý thông tin người dùng

Quản lý sản phẩm, đơn hàng và người dùng từ phía quản trị viên

* *Về công nghệ sử dụng*

Sử dụng nền tảng ASP.NET (Web Forms hoặc MVC) kết hợp với ngôn ngữ C#, Bootstrap, HTML/CSS, JavaScript và SQL Server để xây dựng hệ thống.

Áp dụng mô hình phân lớp (3-tier) hoặc MVC để tổ chức mã nguồn một cách rõ ràng, dễ bảo trì.

* *Về quy mô hệ thống*

Hệ thống được xây dựng ở mức độ mô phỏng, chưa tích hợp chức năng thanh toán trực tuyến hoặc các công nghệ bảo mật nâng cao.

Dữ liệu sản phẩm và người dùng mang tính thử nghiệm, phục vụ mục đích học tập và trình bày chuyên đề.

* *Về thời gian thực hiện*

Đề tài được thực hiện trong khoảng thời gian quy định của học phần chuyên đề, vì vậy phạm vi nghiên cứu và phát triển được giới hạn ở mức cơ bản, phù hợp với năng lực và thời gian của nhóm thực hiện.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1 Tổng quan về mô hình MVC**

**2.1.1 Khái niệm**

MVC là viết tắt của Model - View - Controller, là một mô hình kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng web. Mô hình này giúp phân tách rõ ràng các thành phần của ứng dụng, từ đó tăng tính tổ chức, dễ bảo trì và dễ mở rộng.

Cụ thể, mô hình MVC bao gồm ba thành phần chính:

* ***Model (M)*** *–* ***Tầng dữ liệu và xử lý nghiệp vụ***

Đại diện cho **dữ liệu** và **logic nghiệp vụ** của ứng dụng.

Là nơi xử lý các thao tác với cơ sở dữ liệu như: truy vấn, thêm, sửa, xóa dữ liệu.

Model không phụ thuộc vào giao diện hay cách hiển thị dữ liệu.

* ***View (V) – Tầng giao diện người dùng***

Là phần hiển thị thông tin ra cho người dùng.

View nhận dữ liệu từ Controller và trình bày dưới dạng HTML, CSS, có thể kèm JavaScript để tạo giao diện thân thiện, trực quan.

View không xử lý logic nghiệp vụ.

* ***Controller (C) – Tầng điều khiển***

Là trung gian giữa Model và View.

Nhận yêu cầu từ người dùng (thông qua URL, nút nhấn, v.v.), xử lý yêu cầu đó, tương tác với Model để lấy dữ liệu, sau đó chuyển dữ liệu đến View để hiển thị.

Ngoài ra, Controller còn thực hiện các xử lý điều hướng hoặc logic cần thiết khi người dùng tương tác.

**2.1.2 Các thành phần**

Ba thành phần chính của mô hình MVC bao gồm:

* ***Model (M) – Mô hình dữ liệu***

*Chức năng chính:*

Đại diện cho dữ liệu và logic nghiệp vụ của ứng dụng.

*Nhiệm vụ:*

Quản lý dữ liệu, trạng thái của ứng dụng.

Tương tác với cơ sở dữ liệu (CRUD: Create, Read, Update, Delete).

Cung cấp dữ liệu cho View thông qua Controller.

* ***Controller (C) – Bộ điều khiển***

*Chức năng chính:*

Là trung gian giữa Model và View. Nhận đầu vào từ người dùng và xử lý logic.

*Nhiệm vụ:*

Nhận yêu cầu từ người dùng (qua giao diện – View).

Gọi đến các phương thức của Model để xử lý dữ liệu.

Chọn View phù hợp để hiển thị kết quả.

* ***View (V) – Giao diện hiển thị***

*Chức năng chính:*

Hiển thị thông tin cho người dùng (UI – User Interface).

*Nhiệm vụ:*

Nhận dữ liệu từ Controller/Model và trình bày ra màn hình.

Không xử lý logic nghiệp vụ.

**2.1.3 Lịch sử phát triển**

MVC (Model – View – Controller) có một lịch sử phát triển khá lâu dài, bắt nguồn từ những năm 1970 bởi Trygve Reenskaug, là một nhà nghiên cứu tại Xerox PARC – một trong những trung tâm công nghệ tiên phong lúc bấy giờ. Ông phát triển mô hình MVC trong quá trình làm việc với ngôn ngữ Smalltalk để hỗ trợ phát triển giao diện người dùng (GUI) một cách rõ ràng, có tổ chức.

Giai đoạn phát triển ban đầu (1980s), MVC được tích hợp sâu vào các môi trường phát triển dựa trên Smalltalk-80.

Lúc này mô hình chủ yếu được dùng trong ứng dụng desktop GUI, chưa được phổ biến rộng rãi ra ngoài cộng đồng nghiên cứu.

Áp dụng vào Web & mở rộng (1990s – 2000s). Khi ứng dụng web phát triển mạnh (HTML, HTTP, trình duyệt...), nhu cầu tổ chức mã nguồn rõ ràng ngày càng cấp thiết.

**2.1.4 Luồng xử lý trong MVC**

Luồng xử lý thể hiện cách dữ liệu và sự kiện di chuyển giữa các thành phần trong hệ thống. Dưới đây là mô tả tuần tự các bước khi người dùng tương tác với hệ thống:

* ***Người dùng tương tác với hệ thống (qua View)***

Ví dụ: Người dùng nhấn nút "Đăng nhập", điền form, chọn menu...

Người dùng gửi **Request** (yêu cầu) đến **Controller.**

* ***Controller xử lý logic & tương tác với Model***

Controller tiếp nhận yêu cầu từ View và quyết định nên làm gì:

Kiểm tra dữ liệu đầu vào.

Gọi các phương thức tương ứng từ Model để xử lý dữ liệu.

* ***Model thao tác dữ liệu***

Model chịu trách nhiệm:

Tương tác với cơ sở dữ liệu.

Xử lý nghiệp vụ, tính toán, kiểm tra logic...

Sau khi hoàn thành, Model trả dữ liệu/ trạng thái cho Controller.

* ***Controller nhận kết quả và chọn View phù hợp***

Dựa trên kết quả xử lý từ Model, Controller quyết định:

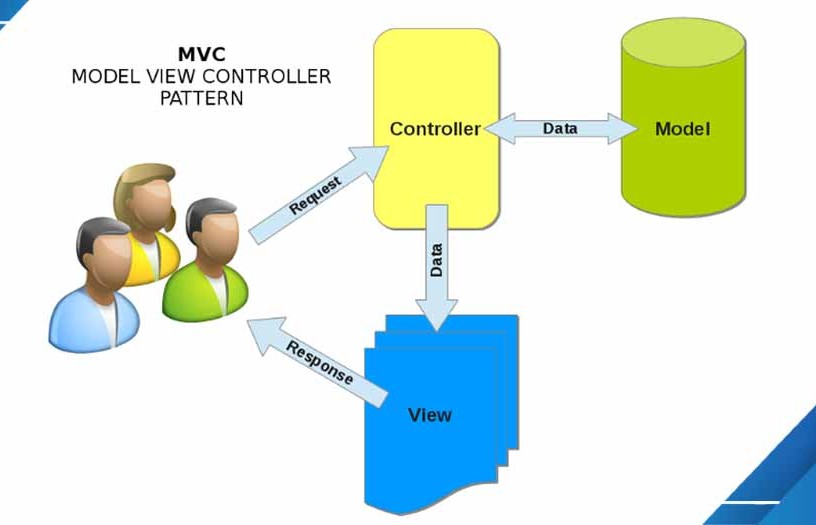
Hiển thị View thành công hay thất bại.

Truyền dữ liệu cần thiết cho View để hiển thị.

* ***View hiển thị dữ liệu cho người dùng***

View nhận dữ liệu từ Controller và hiển thị ra giao diện người dùng.

Người dùng lại tiếp tục tương tác, bắt đầu vòng lặp mới.



Hình 1. Luồng xử lý của mô hình MVC

**2.2 Công nghệ ASP.NET MVC**

**2.2.1 Tổng quan**

**ASP.NET MVC** (Model–View–Controller) là một framework phát triển ứng dụng web được Microsoft giới thiệu, dựa trên nền tảng ASP.NET.

Nó áp dụng mô hình kiến trúc MVC giúp tách biệt:

**Model**: Xử lý dữ liệu, logic nghiệp vụ.

**View**: Giao diện người dùng.

**Controller**: Điều phối luồng xử lý giữa Model và View.

Mục tiêu của ASP.NET MVC là giúp lập trình viên:

Xây dựng ứng dụng web linh hoạt, dễ mở rộng, dễ kiểm thử (test).

Tận dụng được sức mạnh của .NET Framework hoặc .NET Core (về sau).

**2.2.2 Lịch sử phát triễn**

* 2009: Microsoft công bố ASP.NET MVC tại hội nghị MIX, phiên bản 1.0 ra mắt giữa năm.
* 2010: ASP.NET MVC 2 cải tiến khả năng validation, hỗ trợ khuôn mẫu dữ liệu (Data Annotations) và area.
* 2011: ASP.NET MVC 3 giới thiệu Razor View Engine, hỗ trợ dependency injection và filters nâng cao.
* 2012: ASP.NET MVC 4 bổ sung Web API, hỗ trợ phát triển RESTful services và cải thiện mobile templates.
* 2013: ASP.NET MVC 5 tích hợp tính năng ASP.NET Identity, attribute routing và nâng cấp Bootstrap mặc định.
* 2016: ASP.NET Core MVC trong .NET Core 1.0 ra mắt, phát triển đa nền tảng, hiệu năng cao, modular hóa gói NuGet.
* Hiện tại: ASP.NET Core MVC tiếp tục được cập nhật song song với .NET 5/6/7/8, tối ưu hiệu năng và hỗ trợ minimal APIs.

**2.2.3 Tính năng của ASP.NET MVC**

Công nghệ ASP.NET MVC cung cấp nhiều tính năng mạnh mẽ giúp quá trình phát triển ứng dụng web trở nên linh hoạt, hiệu quả và dễ bảo trì hơn. Một trong những tính năng nổi bật nhất là hệ thống định tuyến (Routing), cho phép ánh xạ các URL thân thiện với người dùng đến các hành động (Action) trong Controller, giúp cải thiện khả năng SEO và tổ chức cấu trúc ứng dụng rõ ràng. ASP.NET MVC còn hỗ trợ Model Binding và Validation, giúp tự động chuyển đổi dữ liệu từ các biểu mẫu (form) hoặc tham số URL sang các đối tượng Model, đồng thời kiểm tra và xác thực dữ liệu đầu vào một cách dễ dàng thông qua các thuộc tính (Data Annotation).

Một thành phần quan trọng khác là View Engine Razor, cho phép chèn trực tiếp mã C# vào trong mã HTML bằng cú pháp đơn giản, ngắn gọn và dễ đọc, góp phần nâng cao hiệu quả hiển thị dữ liệu động. Ngoài ra, hệ thống Controller và Action trong ASP.NET MVC giúp điều phối luồng xử lý của ứng dụng: nhận yêu cầu từ người dùng, xử lý logic và trả kết quả về View hoặc dữ liệu JSON. Framework này cũng cung cấp cơ chế Filter (bộ lọc), cho phép lập trình viên can thiệp vào các giai đoạn trước hoặc sau khi Action thực thi, thường dùng để xử lý các tác vụ như xác thực người dùng, ghi log hoặc bộ nhớ đệm (cache).

Bên cạnh đó, ASP.NET MVC chú trọng đến tính bảo mật và xác thực (Security & Authentication) thông qua việc hỗ trợ tích hợp hệ thống ASP.NET Identity, OAuth và các phương thức mã hóa hiện đại. Một ưu điểm khác là khả năng kiểm thử (Testability) cao nhờ cấu trúc tách biệt rõ ràng giữa Model, View và Controller, giúp dễ dàng viết và chạy các bài kiểm thử đơn vị (Unit Test). Cuối cùng, ASP.NET MVC còn có khả năng tích hợp linh hoạt với các công nghệ hiện đại như JavaScript, AJAX, jQuery, SignalR hoặc các framework frontend như Angular, React và Vue.js, giúp tạo nên các ứng dụng web động, tương tác cao và đáp ứng tốt yêu cầu của người dùng hiện đại.

**2.3 Giới thiệu ngôn ngữ lập trình C# (C Sharp)**

**2.3.1 Tổng quan**

C# (đọc là “C Sharp”) là một ngôn ngữ lập trình hướng đối tượng được Microsoft phát triển, do Anders Hejlsberg (người từng thiết kế Turbo Pascal và Delphi) khởi xướng vào năm 2000. C# là một phần trong nền tảng .NET Framework, ra đời với mục tiêu tạo ra một ngôn ngữ hiện đại, mạnh mẽ, an toàn và dễ sử dụng, đồng thời khắc phục những hạn chế của các ngôn ngữ lập trình trước đó như C++ hay Visual Basic.

C# được thiết kế dựa trên cú pháp của ngôn ngữ C và C++, đồng thời chịu ảnh hưởng từ Java, nên lập trình viên có thể dễ dàng tiếp cận và học tập. Ngôn ngữ này hỗ trợ đầy đủ các nguyên lý của lập trình hướng đối tượng (OOP) như đóng gói (Encapsulation), kế thừa (Inheritance) và đa hình (Polymorphism). Ngoài ra, C# còn cung cấp nhiều tính năng tiên tiến như thu gom rác tự động (Garbage Collection), quản lý bộ nhớ an toàn, xử lý sự kiện và ngoại lệ, cùng LINQ (Language Integrated Query) giúp thao tác dữ liệu dễ dàng hơn.

Ban đầu, C# chủ yếu được sử dụng để phát triển ứng dụng Windows thông qua .NET Framework. Tuy nhiên, với sự ra đời của .NET Core và hiện nay là .NET (từ phiên bản .NET 5 trở đi), C# đã trở thành ngôn ngữ lập trình đa nền tảng, cho phép xây dựng các ứng dụng web, desktop, mobile, game và cloud chạy trên Windows, Linux và macOS.

Một ưu điểm nổi bật của C# là khả năng tích hợp chặt chẽ với hệ sinh thái Microsoft, bao gồm các công nghệ như ASP.NET (phát triển web), Xamarin (ứng dụng di động), Unity (phát triển game), và Azure (dịch vụ điện toán đám mây). Với cú pháp rõ ràng, hiệu năng cao, cộng đồng lớn mạnh và sự hỗ trợ liên tục từ Microsoft, C# ngày nay là một trong những ngôn ngữ lập trình phổ biến và được ưa chuộng nhất thế giới, đặc biệt trong lĩnh vực phát triển phần mềm doanh nghiệp và ứng dụng hiện đại.

**2.4 Giới thiệu các Controls của ASP.NET**

**2.4.1 Tổng quan**

Trong công nghệ ASP.NET, Control (điều khiển) là các thành phần giao diện được sử dụng để tạo, hiển thị và tương tác với người dùng trên trang web. Các control giúp lập trình viên dễ dàng xây dựng giao diện động mà không cần viết quá nhiều mã HTML hoặc JavaScript thủ công. ASP.NET cung cấp một tập hợp phong phú các server controls, HTML controls và validation controls nhằm đáp ứng đa dạng nhu cầu phát triển web.

**2.4.2 HTML Controls**

HTML Controls là các thẻ HTML thông thường được mở rộng để hoạt động như server controls bằng cách thêm thuộc tính runat="server". Khi đó, ASP.NET có thể truy cập và xử lý các phần tử này trên máy chủ.

Giúp lập trình viên điều khiển nội dung và hành vi của thẻ HTML trực tiếp từ mã C# trên server.

**2.4.3 Web Server Controls**

Đây là nhóm control được ASP.NET cung cấp sẵn, có khả năng tự động sinh HTML và tương tác với người dùng mạnh mẽ hơn HTML Controls.

*Một số Web Server Controls thông dụng gồm:*

| **Nhóm** | **Tên Control** | **Chức năng** |
| --- | --- | --- |
| **Control nhập liệu** | TextBox, Button, CheckBox, RadioButton, DropDownList, ListBox | Cho phép người dùng nhập hoặc chọn dữ liệu. |
| **Control hiển thị** | Label, Image, HyperLink, Literal | Hiển thị nội dung văn bản, hình ảnh, liên kết. |
| **Container Controls** | Panel, PlaceHolder, MultiView | Dùng để nhóm và bố trí các control khác. |
| **Control dữ liệu (Data Controls)** | GridView, DataList, Repeater, FormView, DetailsView | Hiển thị và thao tác với dữ liệu từ cơ sở dữ liệu. |
| **Control điều hướng** | Menu, TreeView, SiteMapPath | Giúp người dùng di chuyển giữa các trang. |
| **Login Controls** | Login, LoginView, LoginStatus, CreateUserWizard | Dùng để xác thực người dùng và quản lý phiên đăng nhập. |

**2.4.4 Validation Controls**

Nhóm Validation Controls dùng để kiểm tra tính hợp lệ của dữ liệu nhập từ người dùng trước khi gửi lên server, giúp giảm lỗi và tăng tính bảo mật.

*Một số control thường dùng gồm:*

| **Control** | **Mục đích** |
| --- | --- |
| **RequiredFieldValidator** | Bắt buộc người dùng phải nhập dữ liệu. |
| **RangeValidator** | Kiểm tra giá trị trong một khoảng cho phép. |
| **CompareValidator** | So sánh giá trị của 2 control. |
| **RegularExpressionValidator** | Kiểm tra dữ liệu theo biểu thức chính quy (regex). |
| **CustomValidator** | Cho phép tự định nghĩa quy tắc kiểm tra riêng. |
| **ValidationSummary** | Hiển thị danh sách các lỗi kiểm tra trên trang. |

**2.5 Giới thiệu ADO.NET và các đối tượng truy cập CSDL**

**2.5.1 Khái niệm về ADO.NET**

ADO.NET (ActiveX Data Objects .NET) là một công nghệ truy cập và xử lý dữ liệu trong nền tảng .NET Framework do Microsoft phát triển.

Nó cho phép các ứng dụng giao tiếp, kết nối và thao tác dữ liệu với nhiều hệ quản trị cơ sở dữ liệu khác nhau như SQL Server, Oracle, MySQL, Access,…

ADO.NET được xây dựng dựa trên mô hình hướng đối tượng, cung cấp một tập hợp các lớp (classes) trong namespace System.Data và các namespace mở rộng như System.Data.SqlClient, System.Data.OleDb,... giúp lập trình viên dễ dàng thực hiện các thao tác như:

* Kết nối đến CSDL.
* Thực thi truy vấn SQL (SELECT, INSERT, UPDATE, DELETE).
* Lấy dữ liệu và hiển thị lên giao diện.
* Cập nhật dữ liệu ngược trở lại CSDL.

Khác với ADO (phiên bản cũ trong VB6), ADO.NET hỗ trợ mô hình ngắt kết nối (disconnected model), giúp tối ưu hiệu năng, giảm tải cho máy chủ cơ sở dữ liệu và phù hợp với các ứng dụng web.

**2.5.2 Mô hình hoạt động của ADO.NET**

*ADO.NET hoạt động dựa trên hai mô hình chính:*

* *Connected Model (mô hình kết nối)*:

Ứng dụng kết nối trực tiếp đến CSDL, thực thi câu lệnh SQL, lấy kết quả và ngắt kết nối ngay sau đó.

→ Phù hợp cho thao tác nhanh, dữ liệu nhỏ.

* *Disconnected Model (mô hình ngắt kết nối):*

Dữ liệu được tải về bộ nhớ tạm (DataSet hoặc DataTable), cho phép thao tác, xử lý offline và đồng bộ lại khi cần.

→ Phù hợp cho ứng dụng web và hệ thống nhiều người dùng.

**2.5.3 Các đối tượng chính trong ADO.NET**

|  |  |  |
| --- | --- | --- |
| **Đối tượng** | **Namespace** | **Chức năng** |
| **Connection** (SqlConnection, OleDbConnection) | System.Data.SqlClient / System.Data.OleDb | Tạo và quản lý kết nối giữa ứng dụng và CSDL. |
| **Command** (SqlCommand, OleDbCommand) | Cùng namespace với Connection | Thực thi các câu lệnh SQL hoặc stored procedure (lệnh lưu trữ). |
| **DataReader** (SqlDataReader, OleDbDataReader) | Cùng namespace | Đọc dữ liệu theo dòng (forward-only), tốc độ nhanh, ít tốn bộ nhớ (dùng cho mô hình Connected). |
| **DataAdapter** (SqlDataAdapter, OleDbDataAdapter) | Cùng namespace | Là cầu nối giữa DataSet và CSDL, giúp lấy dữ liệu vào DataSet và cập nhật ngược lại. |
| **DataSet** | System.Data | Tập hợp dữ liệu trong bộ nhớ (bao gồm nhiều DataTable), làm việc ngắt kết nối với CSDL. |
| **DataTable** | System.Data | Đại diện cho một bảng dữ liệu (giống như bảng trong SQL). |
| **DataView** | System.Data | Cho phép lọc, sắp xếp dữ liệu trong DataTable mà không thay đổi dữ liệu gốc. |
| **Parameter** | System.Data | Dùng để truyền tham số vào câu lệnh SQL hoặc stored procedure. |

**2.5.4 Quy trình truy cập dữ liệu với ADO.NET**

*Quy trình cơ bản gồm 5 bước:*

* Tạo chuỗi kết nối (Connection String) đến CSDL.
* Mở kết nối (Open Connection) bằng đối tượng SqlConnection.
* Khởi tạo và thực thi câu lệnh (Command) thông qua SqlCommand.
* Đọc hoặc lấy dữ liệu bằng SqlDataReader hoặc DataAdapter + DataSet.
* Đóng kết nối (Close Connection) sau khi hoàn tất.

**2.6 Giới thiệu về SQL Server**

**2.6.1 Tổng quan**

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) do Microsoft phát triển.

Hệ thống này cho phép lưu trữ, quản lý và truy xuất dữ liệu một cách an toàn, hiệu quả và linh hoạt thông qua ngôn ngữ SQL (Structured Query Language).

SQL Server được thiết kế để phục vụ cho cả ứng dụng nhỏ lẫn hệ thống doanh nghiệp quy mô lớn, đồng thời tích hợp chặt chẽ với các công nghệ của Microsoft như .NET, Visual Studio, Azure,...

**2.6.2 Lịch sử phát triển**

Năm 1989: Phiên bản đầu tiên của SQL Server được phát hành, phát triển chung giữa Microsoft, Sybase và Ashton-Tate, chạy trên hệ điều hành OS/2.

Năm 1993: Microsoft tách riêng, phát triển SQL Server cho nền tảng Windows NT.

Các phiên bản quan trọng:

* SQL Server 2000 – hỗ trợ XML và giao diện quản lý cải tiến.
* SQL Server 2005 – ra đời nền tảng .NET Integration, hỗ trợ CLR.
* SQL Server 2012 – thêm chức năng AlwaysOn và Data Tools.
* SQL Server 2016 – cải tiến bảo mật, hỗ trợ JSON, In-Memory OLTP.
* SQL Server 2019 – hỗ trợ Big Data Cluster và tích hợp với Linux.
* SQL Server 2022 – tăng cường khả năng kết nối với Azure và tối ưu hiệu năng truy vấn.

**2.6.3 Cấu trúc của SQL Server**

*SQL Server gồm ba lớp chính:*

**Database Engine**: Thành phần cốt lõi, thực hiện lưu trữ, xử lý và bảo mật dữ liệu.

**SQL Server Agent**: Quản lý và tự động hóa các tác vụ (backup, bảo trì, lịch trình công việc).

**SQL Server Management Studio (SSMS)**: Giao diện đồ họa cho phép người dùng quản trị, viết truy vấn SQL, và theo dõi hoạt động của hệ thống.

**2.6.4 Kết luận**

SQL Server là một nền tảng quản lý dữ liệu toàn diện, đáng tin cậy và mạnh mẽ, phù hợp cho cả ứng dụng doanh nghiệp, website động, và các hệ thống phân tích dữ liệu lớn.

Nhờ khả năng tích hợp sâu với hệ sinh thái Microsoft và bảo mật cao, SQL Server hiện là một trong những hệ quản trị CSDL phổ biến nhất thế giới.

**2.7 Giới thiệu về Bootstrap**

**2.7.1 Khái niệm về Bootstrap**

Bootstrap là một **framework mã nguồn mở** (open-source framework) được phát triển nhằm hỗ trợ thiết kế giao diện người dùng (UI – User Interface) cho các ứng dụng web một cách nhanh chóng và nhất quán. Framework này được xây dựng chủ yếu dựa trên **HTML, CSS và JavaScript**, giúp lập trình viên dễ dàng tạo ra các website có giao diện đẹp, tương thích với nhiều kích thước màn hình và thiết bị khác nhau.

**Bootstrap** ban đầu được phát triển bởi **Mark Otto** và **Jacob Thornton** tại **Twitter** vào năm 2011, với mục tiêu chuẩn hóa các công cụ thiết kế nội bộ. Sau khi được phát hành công khai, **Bootstrap** nhanh chóng trở thành một trong những framework front-end phổ biến nhất trên thế giới.

**2.7.2 Mục tiêu và ưu điểm của Bootstrap**

**2.7.2.1. Mục tiêu phát triển**

Bootstrap được xây dựng với các mục tiêu chính sau:

* Tăng tốc quá trình phát triển giao diện web.
* Đảm bảo tính thống nhất về thiết kế trên nhiều trình duyệt và thiết bị.
* Hỗ trợ thiết kế web responsive (giao diện thích ứng với mọi kích thước màn hình).
* Cung cấp một bộ công cụ và thành phần UI phong phú, dễ sử dụng.

**2.7.2.2. Ưu điểm nổi bật**

* Dễ sử dụng: chỉ cần nắm vững HTML và CSS cơ bản là có thể áp dụng.
* Responsive Design: tự động điều chỉnh bố cục phù hợp với các kích thước màn hình (máy tính, máy tính bảng, điện thoại).
* Tính nhất quán cao: các thành phần giao diện được định nghĩa sẵn theo cùng một chuẩn thiết kế.
* Tích hợp dễ dàng: tương thích với hầu hết các framework front-end và thư viện JavaScript khác.
* Cộng đồng lớn: tài liệu phong phú, nhiều ví dụ và hỗ trợ từ cộng đồng lập trình viên toàn cầu.

**2.7.3 Cấu trúc và thành phần của Bootstrap**

Bootstrap bao gồm các thành phần chính sau:

* **CSS:** Cung cấp các lớp định dạng sẵn cho bố cục, kiểu chữ, bảng, biểu mẫu, nút, hình ảnh, v.v.
* **Grid System:** Hệ thống lưới 12 cột giúp xây dựng bố cục web linh hoạt và responsive.
* **Components:** Tập hợp các thành phần giao diện có sẵn như Navbar, Modal, Dropdown, Alert, Carousel, v.v.
* **Utilities:** Các lớp tiện ích (utility classes) hỗ trợ điều chỉnh nhanh các thuộc tính như margin, padding, màu sắc, kích thước.
* **JavaScript Plugins:** Cung cấp các hiệu ứng động như Tooltip, Collapse, Modal, và các thành phần tương tác khác.

**2.7.4 Cấu trúc và thành phần của Bootstrap**

Bootstrap được sử dụng rộng rãi trong:

* Thiết kế website cá nhân và doanh nghiệp.
* Phát triển giao diện ứng dụng web (web app).
* Tạo prototype nhanh cho dự án.
* Xây dựng hệ thống quản trị (Admin Dashboard).

Nhờ cấu trúc rõ ràng và tài nguyên phong phú, Bootstrap giúp rút ngắn thời gian phát triển mà vẫn đảm bảo tính thẩm mỹ và hiệu năng của sản phẩm.

**2.8 Giới thiệu về Entity Framework**

**2.8.1 Giới thiệu về Entity Framework**

Entity Framework (EF) là một ORM (Object-Relational Mapping Framework) do Microsoft phát triển, được sử dụng để hỗ trợ các lập trình viên .NET trong việc tương tác với cơ sở dữ liệu. Thay vì phải viết thủ công các câu lệnh SQL, lập trình viên có thể thao tác trực tiếp với các đối tượng (objects) trong ngôn ngữ lập trình C# hoặc VB.NET.

Entity Framework giúp chuyển đổi dữ liệu giữa cơ sở dữ liệu quan hệ (Relational Database) và các đối tượng hướng đối tượng (Objects) trong ứng dụng, qua đó giảm thiểu sự phức tạp trong quản lý và truy xuất dữ liệu.

**2.8.2 Mục tiêu và vai trò của Entity Framework**

Entity Framework được Microsoft phát triển nhằm:

* Giảm khối lượng mã SQL mà lập trình viên phải viết thủ công.
* Tự động hóa quá trình ánh xạ (mapping) giữa cơ sở dữ liệu và các lớp trong ứng dụng.
* Tăng tính linh hoạt và khả năng mở rộng trong phát triển phần mềm.
* Đảm bảo tính nhất quán dữ liệu thông qua mô hình hướng đối tượng.

Entity Framework đóng vai trò trung gian giữa lớp ứng dụng (Application Layer) và lớp cơ sở dữ liệu (Database Layer), giúp việc truy xuất, thêm, sửa, xóa dữ liệu trở nên dễ dàng và an toàn hơn.

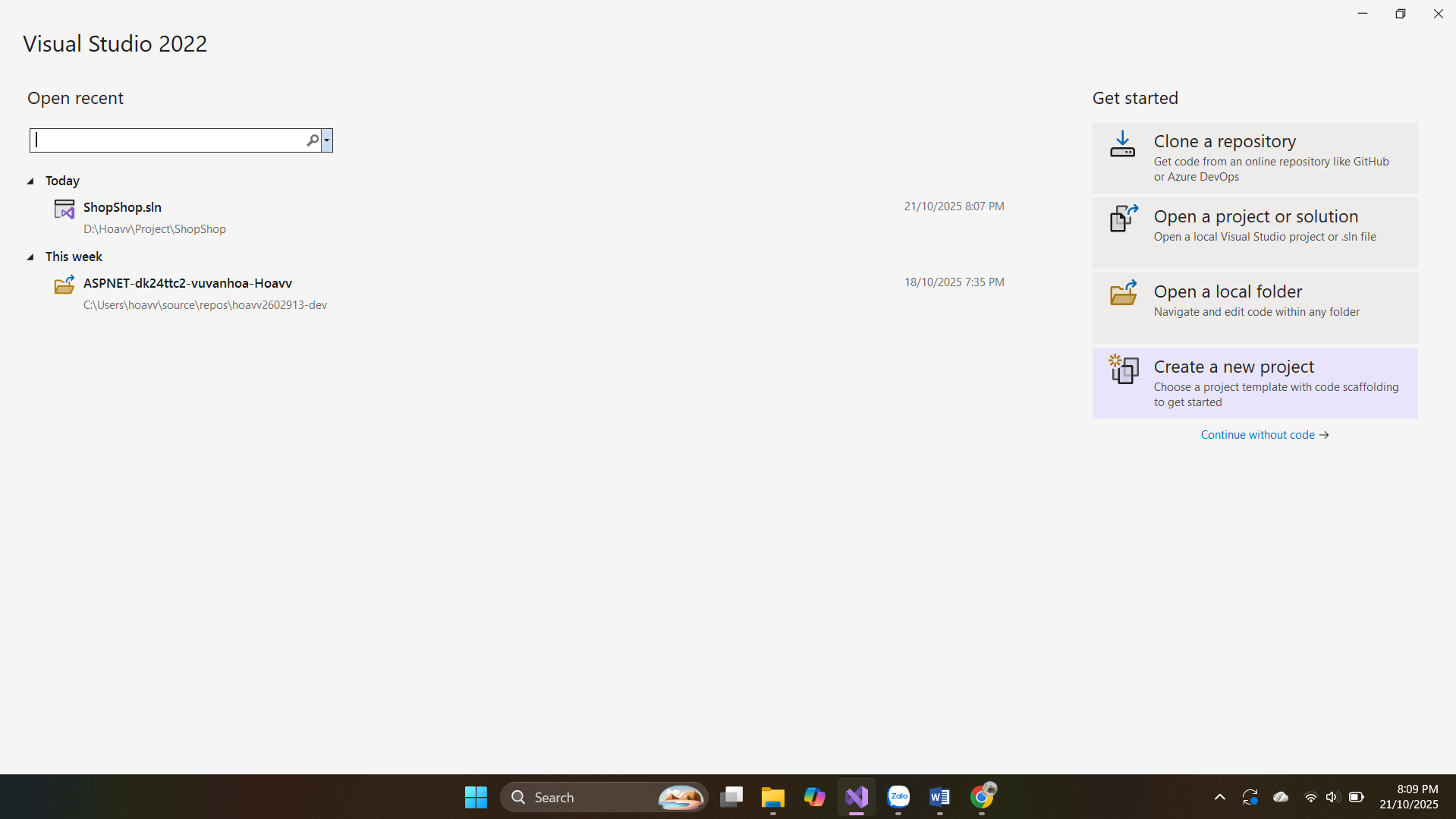
Framework này đặc biệt hữu ích trong các dự án sử dụng kiến trúc 3 lớp (Three-Tier Architecture) hoặc MVC (Model – View – Controller), giúp tăng khả năng tái sử dụng và bảo trì mã nguồn.

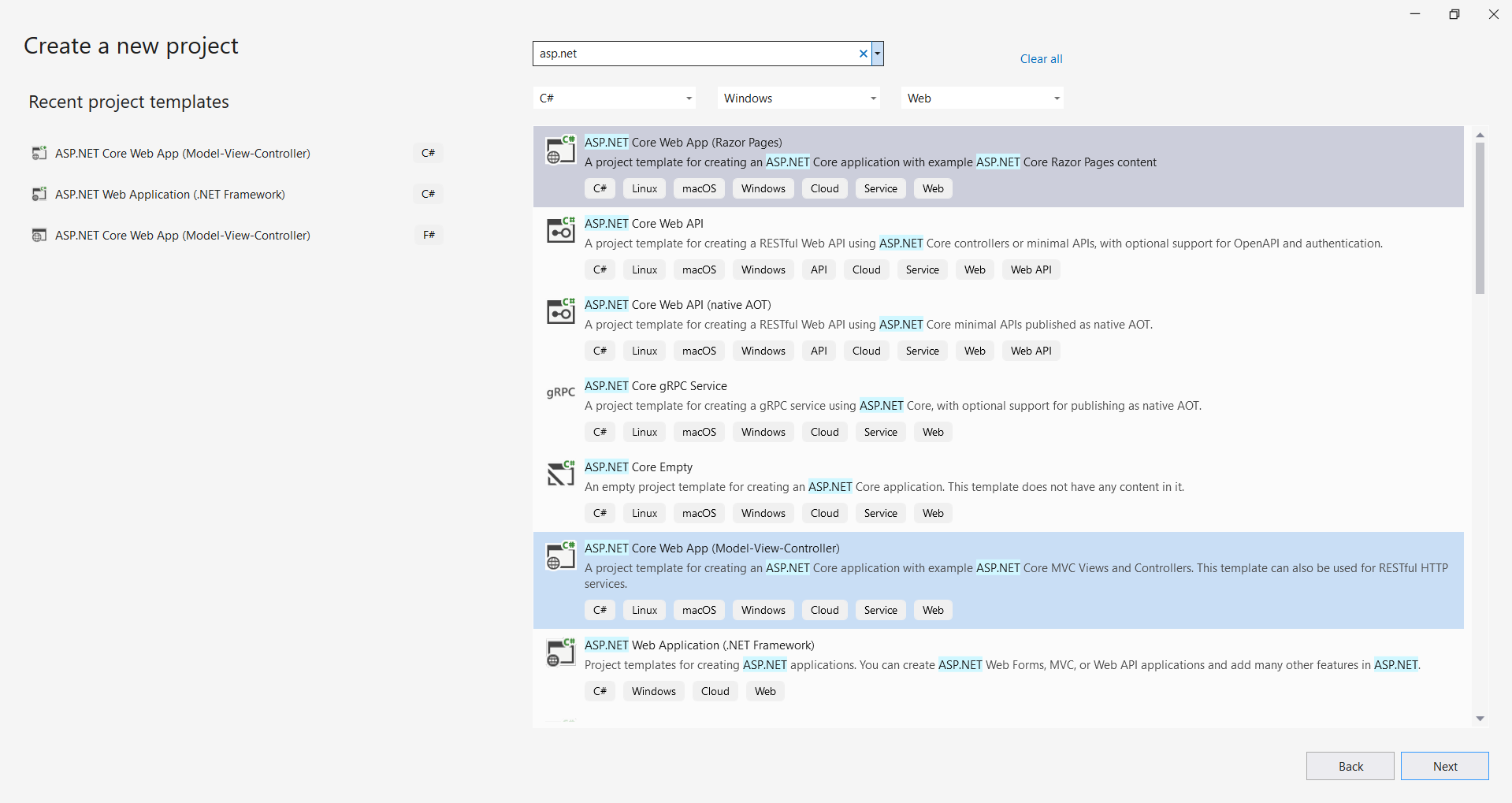
**CHƯƠNG 3: KẾT QUẢ NGHIÊN CỨU**

**3.1 Các công nghệ sử dụng trong đề tài**

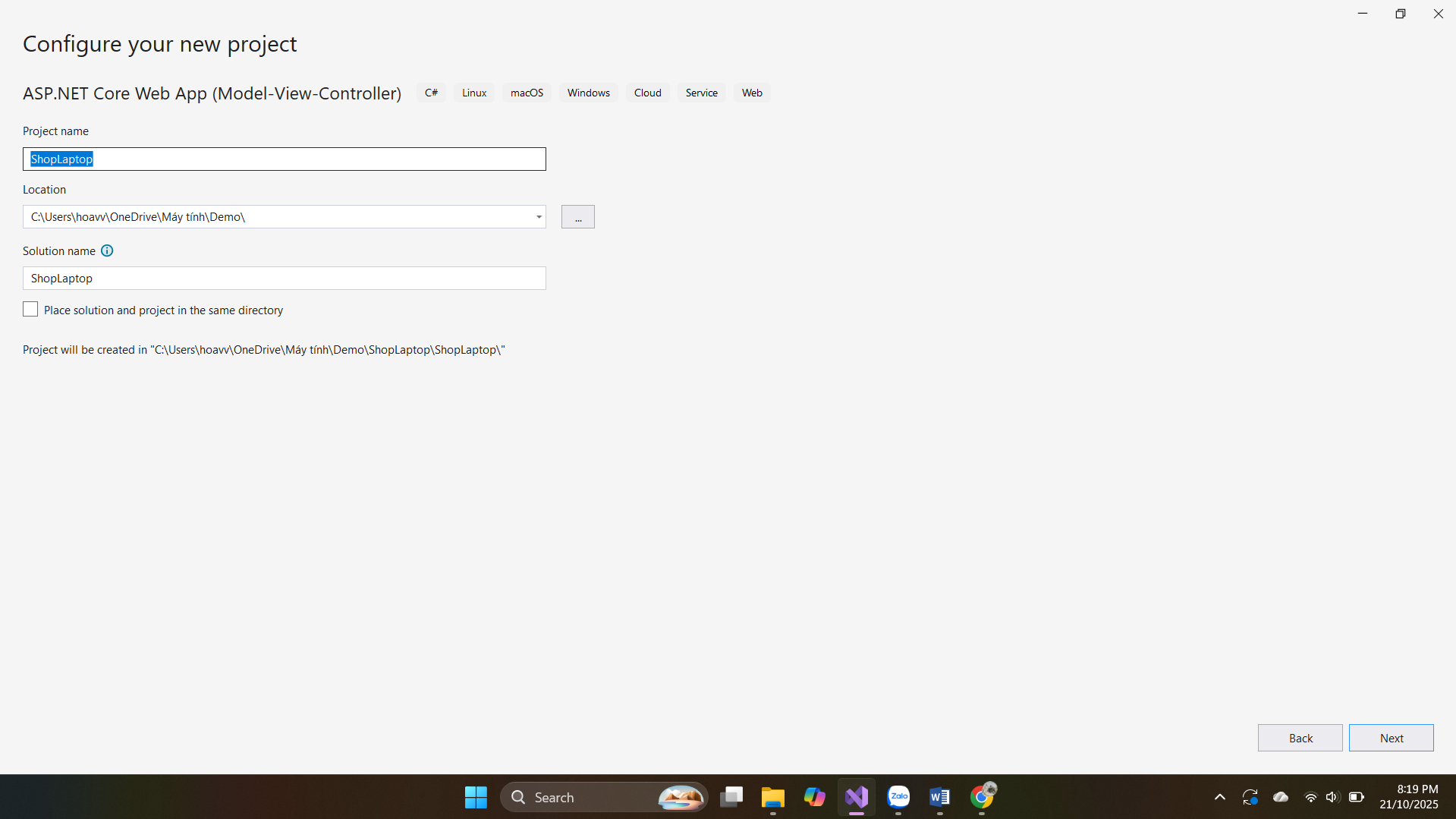
**3.1.1 ASP.NET MVC**

**3.1.1.1 Tạo dự án với ASP.NET MVC**

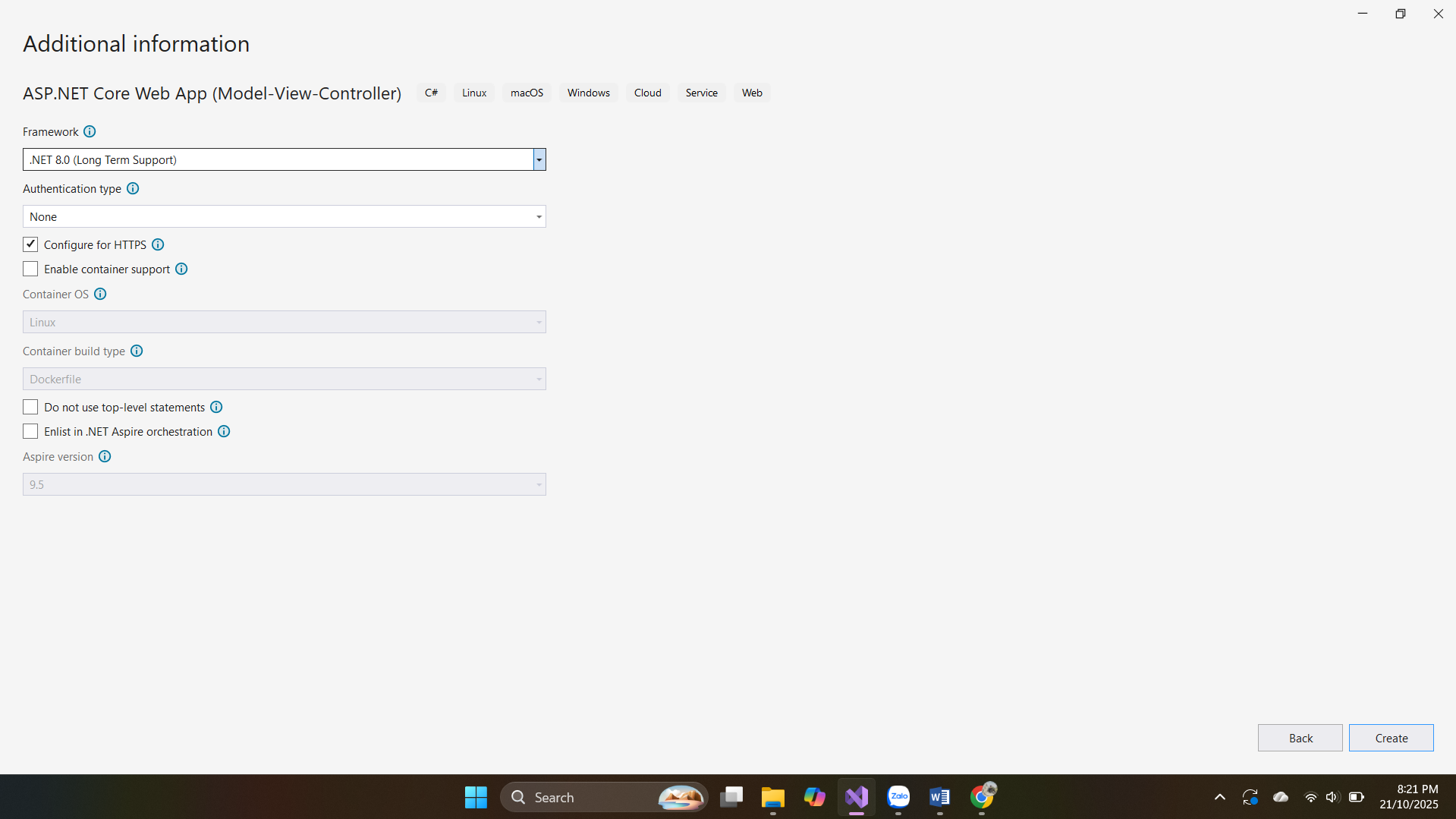
Bước 1: Mở Visual Studio Create a New Project

Bước 2: Chọn ASP.NET Core Web App (Model-View-Controller)

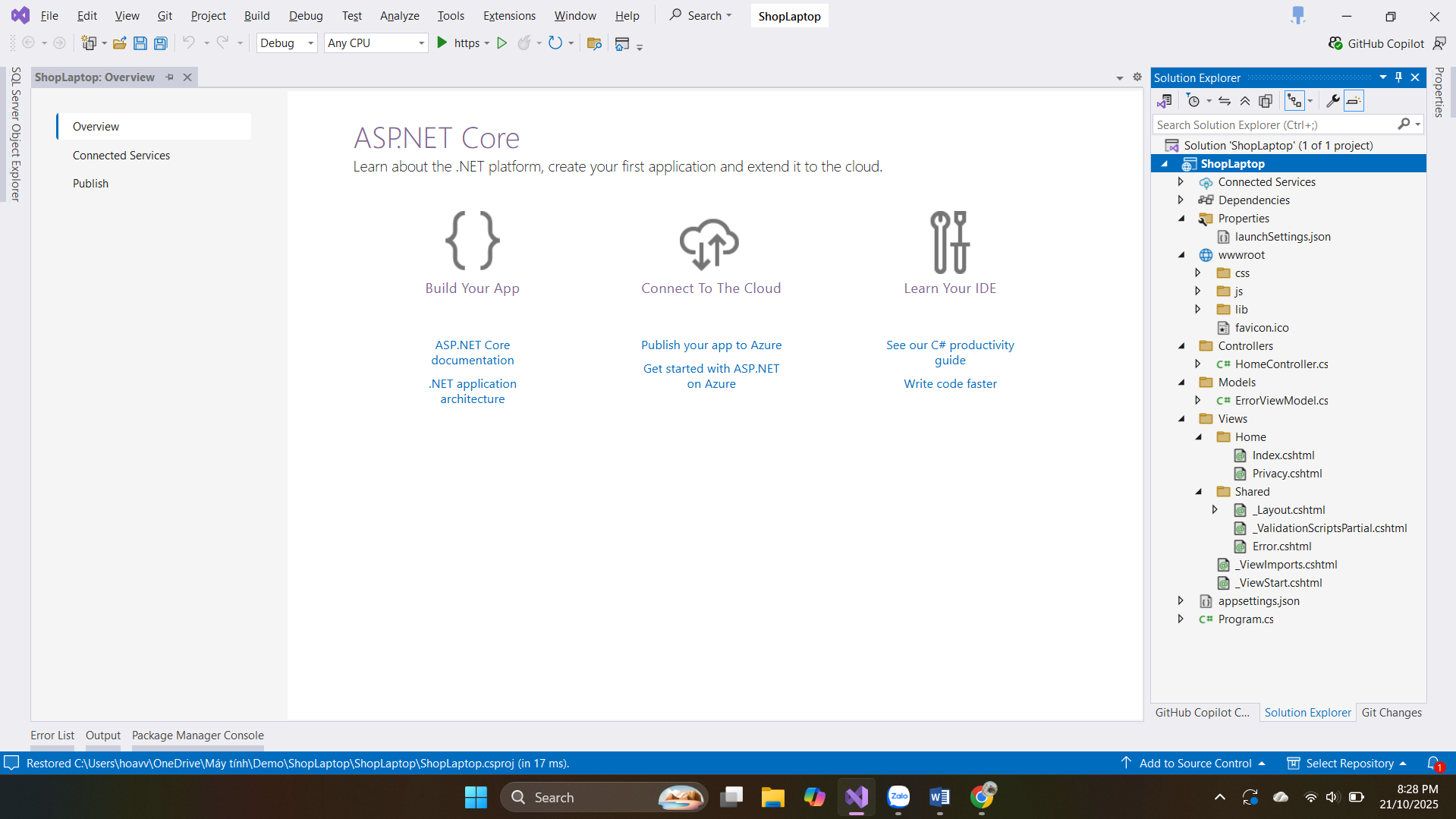
Bước 3: Đặt tên Project, đường dẫn thư mục lưu và nhấn “Next”



Bước 4: Nhấn “Create” để tiếp tục



**3.1.1.2 Cấu trúc cây thư mục của ASP.NET MVC**



Hình 2: Giao diện làm việc và cấu trúc cây thư mục của ASP.NET MVC

**3.1.2 Entity Framework**

**3.1.2 Cài đặt Entity Framework cho dự án**

Vào Tools NuGet Package Manager Package Manager Console để có thể sử dụng dòng lệnh để cài đặt EF Core và provider cho SQL Server.

