UBND TỈNH BÀ RỊA – VŨNG TÀU

TRƯỜNG CAO ĐẲNG KỸ THUẬT CÔNG NGHỆ

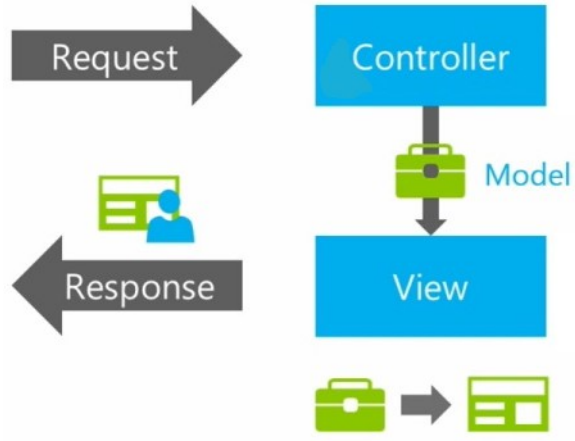
GIÁO TRÌNH

MÔ ĐUN LẬP TRÌNH WEB ASP.NET MVC

NGHỀ: CÔNG NGHỆ THÔNG TIN

TRÌNH ĐỘ: CAO ĐẲNG

(Ban hành kèm theo Quyết định số: ……/QĐ-CĐKTCN, ngày … tháng … năm 20…… của Hiệu trưởng Trường Cao đẳng Kỹ thuật Công nghệ BR-VT)



BM/QT10/P.ĐTSV/04/04
Ban hành lần: 3

BÀ RỊA – VŨNG TÀU, NĂM 2020

TUYÊN BỐ BẢN QUYỀN

Nhằm đáp ứng nhu cầu học tập và nghiên cứu cho giảng viên và sinh viên nghề Công nghệ Thông tin trong trường Cao đẳng Kỹ thuật Công nghệ Bà Rịa – Vũng Tàu, chúng tôi đã thực hiện biên soạn tài liệu Lập trình web ASP.NET MVC.

Tài liệu được biên soạn thuộc loại giáo trình phục vụ giảng dạy và học tập, lưu hành nội bộ trong Nhà trường nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Giáo trình "Lập trình web ASP.NET MVC" được biên soạn dựa trên khung chương trình đào tạo Cao đẳng nghề Công nghệ Thông tin đã được Trường Cao đẳng Kỹ thuật Công nghệ Bà Rịa – Vũng Tàu phê duyệt.

Tác giả đã nghiên cứu một số tài liệu, công nghệ hiện đại kết hợp với kinh nghiệm làm việc thực tế để viết nên giáo trình này. Nội dung được tác giả trình bày cô động, dễ hiểu kèm theo các bước hướng dẫn thực hành chi tiết theo nguyên tắc quan tâm đến kết quả đầu ra, khả năng tự học và kỹ năng cần thiết để HSSV có thể hoàn thành ứng dụng web quảng cáo sản phẩm, thương mại điện tử, … bằng .NET framework theo mô hình MVC, một trong những mô hình ứng dụng phổ biến hiện nay, đáp ứng nhu cầu thực tế của doanh nghiệp.

Nội dung giáo trình được chia thành 10 bài, trong đó:

Bài 1: Tổng quan về ASP.NET MVC

Bài 2: Tạo ứng dụng web ASP.NET MVC

Bài 3: Xây dựng model

Bài 4: Tạo view

Bài 5: Tạo controller

Bài 6: Truyền dữ liệu giữa controller và view

Bài 7: Thiết kế view bằng HTML Helper

Bài 8: Hiển thị dữ liệu trên Web Grid

Bài 9: Kiểm tra dữ liệu trên trang web

Bài 10: Triển khai ứng dụng

Trong quá trình biên soạn, chắc chắn giáo trình còn nhiều thiếu sót. Tác giả rất mong nhận được ý kiến đóng góp của quý thầy/cô và các em học sinh, sinh viên để tiếp tục hoàn thiện hơn.

Xin chân thành cảm ơn quý đồng nghiệp, bạn bè đã có những ý kiến đóng góp trong quá trình biên soạn giáo trình này.

Bà Rịa – Vũng Tàu, ngày …… tháng …… năm ……

Tham gia biên soạn

1. Phan Hữu Phước – Chủ biên

MỤC LỤC

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: Lập trình web ASP.NET MVC

Mã mô đun: MĐ28

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

Vị trí: được giảng dạy sau khi học xong Thiết kế web, Lập trình Windows EF và WCF, SQL Server, Cơ sở dữ liệu, Quản trị CSDL Access.

Tính chất: là mô đun tự chọn

Ý nghĩa và vai trò của mô đun: cung cấp cho người học kiến thức và kỹ năng để xây dựng hoàn chỉnh một website thương mại điện tử, quảng bá cho các tổ chức, cá nhân trên nền .NET Framework theo mô hình MVC.

Mục tiêu của mô đun:

Về kiến thức:

Tìm hiểu được ngôn ngữ lập trình ASP.NET.

Tìm hiểu được mô hình MVC (Model – View – Controller)

Biết quá trình xử lý yêu cầu của mô hình xử lý MVC và ASP.NET MVC.

Biết được các truyền và gọi dữ liệu giữa Controller và View.

Thiết kế được phân trang cho từng trang.

Sử dụng được Bootstrap để hiệu quả và đảm bảo tính đồ họa cho từng trang web.

Thiết kế được những website điện tử đơn giản, các tổ chức, doanh nghiệp, cá nhân đơn giản.

Về kỹ năng:

Khả năng tuy duy, sáng tạo trong việc thiết kế và lập trình các ứng dụng website điện tử cơ bản.

Kết nối và xử lý dữ liệu bằng LINQ và Entity Framework

Xây dựng được ứng dụng web với ASP.Net MVC

Về năng lực tự chủ và trách nhiệm:

Reèn luyện va`nâng cao ky˜năng lập trình theo công nghệ Entity Framework (EF), Windows Communication Foundation (WCF).

Thực hiện các bài thực hành đảm bảo đúng trình tự, an toàn.

Nội dung của mô đun:

BÀI 1: TỔNG QUAN VỀ ASP.NET MVC

Mã bài: 28.1

Giới thiệu:

Model View Controller là một mô hình kiến trúc theo hướng đối tượng, cho phép người phát triển tách một ứng dụng thành 3 thành phần chính: Model, View, Controller.

Mục tiêu:

Trình bày được mô hình, ưu và nhược điểm của MVC, lý do làm ứng dụng trên mô hình MVC

Biết các thành phần và quy trình xử lý yêu cầu của ASP.NET MVC

Nội dung chính:

Mô hình MVC

1.1. Giới thiệu

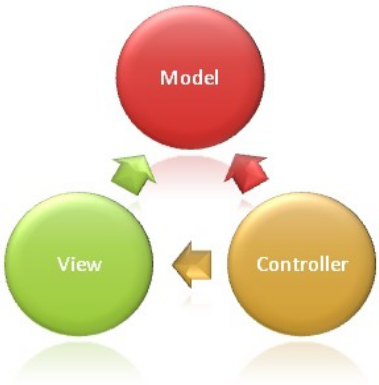
Model View Controller là một mô hình kiến trúc theo hướng đối tượng, cho phép người phát triển tách một ứng dụng thành 3 thành phần chính:

Model: thành phần đại diện cho dữ liệu của ứng dụng, bao gồm cả chức năng kiểm tra tính hợp lệ của dữ liệu.

View: thành phần đảm trách việc hiển thị dữ liệu và các thành phần trong giao diện người dùng.

Controller: thành phần có trách nhiệm tiếp nhận và xử lý các yêu cầu gửi đến cho ứng dụng, làm nhiệm vụ điều phối công việc giữa View và Model.

→ Việc phát triển ứng dụng, nâng cấp, bảo trì và thử nghiệm trở nên đơn giản và dễ dàng hơn.



Hình 1.1. Mô hình 3 thành phần của MVC

1.2. Ưu và nhược điểm

Ưu điểm

Phát triển phần mềm một cách chuyên nghiệp hóa, có thể chia công việc cho nhiều nhóm chuyên môn khác nhau

Nhóm thiết kế

Nhóm lập trình

Nhóm tổ chức dữ liệu

→ Giúp phát triển ứng dụng nhanh, đơn giản, dễ dàng nâng cấp

Bảo trì: với các lớp được phân chia, các thành phần của một hệ thống dễ dàng được thay đổi nhưng có thể được cô lập từng lớp, hoặc chỉ ảnh hưởng đến lớp ngay gần kề nó, không làm ảnh hưởng đến cả ứng dụng.

Mở rộng: việc thêm chức năng vào từng lớp sẽ dễ dàng hơn.

Nhược điểm

Thích hợp với các dự án vừa và lớn

Mất nhiều thời gian trong quá trình phát triển

Mấ thời gian trung chuyển dữ liệu giữa các lớp

1.3. Lý do nên làm trên mô hình MVC

Mô hình MVC đã chia ứng dụng thành các thành phần Model, View và Controller nên người phát triển có thể tạo ra nhiều View và nhiều Controller cho các Model mà không phải đối mặt với việc thay đổi trong thiết kế Model.

→ Giúp cho việc duy trì, di chuyển và tổ chức ứng dụng dễ dàng hơn.

Đối với những người mới thì việc xây dựng ứng dụng dựa trên mô hình MVC phức tạp và lãng phí vì công việc này như là xây dựng một dự án lớn, tuy nhiêu "bí mật" của MVC không nằm ở chỗ viết code mà ở chỗ duy trì nó.

Cho phép sửa code mà không ảnh hưởng nhiều đến các thành phần khác.

Cho phép làm việc nhóm trở nên dễ dàng hơn vì mỗi nhóm sẽ làm việc dựa trên thế mạnh của mình.

Nhóm View: chịu trách nhiệm về xây dựng giao diện tương tác với người dùng

Nhóm Model: chịu trách nhiệm về việc xây dựng các lớp xử lý trên dữ liệu

Nhóm Controller: có cái nhìn tổng thể về các luồng của ứng dụng, quản lý các yêu cầu, làm việc với các Model và lựa chọn View hiển thị cho người dùng.

ASP.NET MVC

2.1. ASP.NET MVC là gì?

ASP.NET MVC là một Framework hỗ trợ đầy đủ cho việc xây dựng ứng dụng web ASP.NET theo mô hình MVC.

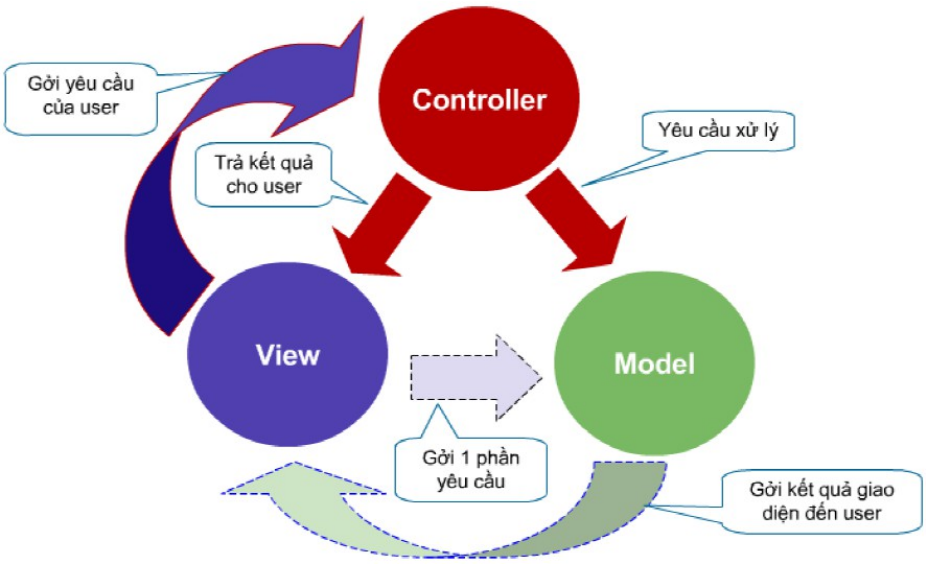
Giống như ASP.NET Web Forms, ASP.NET MVC được xây dựng dựa trên ASP.NET Framework. Điều này có nghĩa là người phát triển có thể sử dụng các API trong ứng dụng ASP.NET Web Forms truyền thống vào trong ứng dụng ASP.NET MVC.

Trong ASP.NET MVC, nhiều cải tiến về ASP.NET đã được đưa vào Framework.

Mục đích chính của mẫu thiết kế này là cô lập xử lý nghiệp vụ từ giao diện người dùng để tập trung vào khả năng bảo trì, cải tiến, kiểm thử tốt hơn và làm cho ứng dụng có cấu trúc gọn hơn.

MVC Framework được định nghĩa trong namespace System.Web.Mvc.

2.2. Kiến trúc



Hình 1.2. Mô hình kiến trúc 3 thành phần trong MVC

Models là các thành phần có nhiệm vụ:

Đọc và ghi dữ liệu

Lưu trữ thông tin, trạng thái của các đối tượng

Tất cả các nghiệp vụ logic đều được thực thi ở Model. Dữ liệu được nhập từ người dùng sẽ thông qua View để kiểm tra ở Model trước khi lưu vào CSDL. Việc truy xuất, xác nhận và lưu dữ liệu là phần việc của Model.

Views

Là các thành phần chịu trách nhiệm hiển thị thông tin cho người dùng thông qua giao diện

Các thông tin cần hiển thị trên View được tạo ra từ các thành phần Models trong mô hình dữ liệu

Controllers

Là các thành phần xử lý tương tác với người dùng, làm việc với Model, chọn một View thích hợp để hiển thị thông tin giao diện người dùng

Trong một ứng dụng MVC, View hiển thị thông tin, còn điều khiển dòng nhập xuất của người dùng vẫn do Controller đảm trách.

2.3. Quá trình xử lý yêu cầu

Một trong những khái niệm quan trọng nhất để hiểu về ứng dụng MVC là không có mối quan hệ tồn tại giữa một yêu cầu và một tập tin vật lý bên trong Web Server.

Trong ứng dụng ASP.NET Web Forms truyền thống, mỗi yêu cầu được dịch thành một lời gọi đến một tập tin trong Web Server. Ví dụ: nếu yêu cầu là một url http://myapp/mypage.aspx thì Web Server sẽ thông dịch yêu cầu này bằng cách tìm kiếm trong thư mục của ứng dụng một tập tin có tên là mypage.aspx. Sau đó sẽ xử lý tập tin này và trả về kết quả dạng HTML.

Đối với ứng dụng MVC, khi tiếp nhận một yêu cầu (ví dụ: http://myapp/product/list) thì một thành phần được gọi là ***"routing engine"*** sẽ so sánh yêu cầu với route cụ thể.

Route sẽ xác định các yêu cầu bằng cách sử dụng một chuỗi nền, thiết lập Controller và phương thức bên trong Controller cần xử lý yêu cầu.

Khi Route được nhận diện, công cụ route tạo ra bộ giám sát yêu cầu, bộ này lần lượt sẽ tạo ra các đối tượng Controller để xử lý yêu cầu (trong ví dụ trên thì controller là ***product*** và phương thức xử lý là ***list***).

Hình 1.3. Quá trình xử lý yêu cầu của MVC

2.4. Một số đặc điểm

Tiếp tục hỗ trợ các tính năng trong ASP.NET

Hỗ trợ sử dụng các tập tin .ASPX, ASCX, .Master như là thành phần View

Hỗ trợ đầy đủ các tính năng bảo mật của ASP.NET

Form/Windows authenticate, URL authorization, Membership/Roles, Output và data caching, secion/profile state, configuration system, provider architecture, …

Tách rõ ràng các mối liên quan, mở ra khả năng test TDD (Test Driven Developer)

Có thể test unit trong ứng dụng mà không cần phải chạy Controllers cùng với tiến trình của ASP.NET và có thể dùng bất kỳ một unit testing framework nào như Nunit, MBUnit, MS Test, …

Có khả năng mở rộng ứng dụng, mọi thứ trong MVC được thiết kế để dễ thay thế, dễ dàng tùy biến.

Ánh xạ URL mạnh mẽ, cho phép xây dựng ứng dụng với URL gọn gàng, thân thiện với các Search Engine

Không sử dụng mô hình PostBack từ giao diện gửi đến Server. Thay vào đó, chủ động đưa những PostBack từ View đến thẳng Controller tùy ý.

Không còn ViewState hay Page Lifecycle tồn tại trong mô hình MVC

Hỗ trợ công cụ tạo View (Support for Multiple View Engine)

Cho phép chọn công cụ tạo View

Hộp thoại New Project cho phép xác định View Engine mặc định cho một project

Các lọa View Engine

Web Forms (ASPX)

Razor

Hay mọt View Engine nguồn mở như Spark, Nhaml, Ndjango.

Hỗ trợ định tuyến

ASP.NET MVC Framework có một bộ máy ánh xạ URL thật sự mạnh mẽ

Bộ máy này cung cấp phương pháp rất linh hoạt trong việc ánh xạ URLs sang các Controller.

Có thể dễ dàng định ra các quy luật cài đặt đường đi. ASP.NET MVC dựa vào các quy luật đó để xác định Controller và phương thức (Action) cần phải thực thi.

ASP.NET còn có khả năng phân tích URL, chuyển các thông số trong URL thành các tham số trong lời gọi tới phương thức trong Controller.

Model Binding

Model Binding là tính năng thế mạnh của ASP.NET MVC

Hỗ trợ viết phương thức nhận một đối tượng tùy biến như là một ahm số

Với sự hỗ trợ của Model Binding, bây giờ người phát triển chỉ cần tập trung vào việc cài đặt các nghiệp vụ logic, không cần phải bận tâm về việc suy nghỉ làm cách nào để ánh xạ dữ liệu từ người dùng sang các đối tượng .NET).

Tính năng Bunding được áp dụng mặc định. Người phát triển không cần phải gọi các script hoặc các tập tin css riên biệt bởi vì tất cả sẽ được đóng gói và nén lại.

Filters: tính năng mạnh trong ASP.NET MVC, hỗ trợ cho việc kiểm tra tính hợp lệ trước khi một phương thức hành động được gọi hoặc sau khi thi hành.

Razor View

Cú pháp Razor gọn gàng và xúc tích, đòi hỏi một số lượng tối thiểu các tổ hợp phím.

Tìm hiểu Razor tương đối dễ dàng vì nó dựa trên C# và HTML.

Visual Studio bao gồm IntelliSense và mã cú pháp Razor được màu hóa.

Với Razor View có thể kiểm tra từng đơn vị mà không đòi hỏi phải chạy ứng dụng hoặc phải chạy Web Server.

Từ những đặc điểm đã nêu, có thể tạm tóm tắt lại các ưu và khuyết điểm của ASP.NET MVC như sau:

Ưu điểm

Có tính mở rộng do thay thế từng thành phần một cách dễ dàng.

Không sử dụng ViewState để điều khiển ứng dụng.

Bổ sung mới hệ thống định tuyến (Routers).

Hỗ trợ tối đa cho việc kiểm thử bằng việc tạo và cài đặt các unitests tự động.

Hỗ trợ kết hợp rất tốt giữa người lập trình và thiết kế giao diện.

Khuyết điểm

Thay đổi cách lập trình của Web Forms nên gây ra nhiều khó khăn cho những người chuyên sử dụng Web Forms chuyển sang mô hình này (không còn hướng đến các sự kiển của các control).

Người lập trình phải biết nhiều về HTTP, HTML, CSS và Javascript.

Áp dụng cho dự án nhỏ thì quá cồng kềnh và phức tạp

2.5. So sánh MVC và Web Form

ASP.NET Web Forms sử dụng ViewState để quản lý → khi có sự tương tác với người dùng thì xử lý chậm.

ASP.NET MVC chia làm 3 thành phần: Model, View, Controller.

Mọi tương tác của người dùng với Views sẽ được xử lý thông qua việc thực hiện các phương thức hành động trong Controllers, không còn PostBack, Lifecycle và events.

Việc kiểm tra (test) và gỡ lỗi (debug)

ASP.NET Web Forms chạy tất cả các tiến trình, sự thay đổi ID của bất kỳ control nào cũng ảnh hưởng đến ứng dụng.

Với MVC, việc đó có thể sử dụng các unit test để thẩm định rất dễ dàng các Controller thực hiện thế nào.

Bảng so sánh các tính năng giữa ASP.NET Web Forms và ASP.NET MVC

Tính năng	ASP.NET Web Forms	ASP.NET MVC
Kiến trúc	Mô hình Web form → Bussiness → Database	Sử dụng việc phân chia ứng dụng thành Models, Views, Controllers
Cú pháp	Sử dụng cú pháp của Web Form, tất cả các sự kiện và controls do Server quản lý	Các sự kiện được điều khiển bởi các controller, các controller không do Server quản lý
Truy cập dữ liệu	Sử dụng hầu hết các công nghệ truy cập dữ liệu trong	Phần lớn dùng LINQ to SQL và Entity Framework để tạo

	ứng dụng	mô hình truy cập đối tượng
Debug	Phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, hiển thị, điều khiển các control	Có thể sử dụng các unit test để kiểm tra các phương thức trong các controller
Tốc độ phân tải	Chậm khi trong trang có quá nhiều control vì ViewState quá lớn	Nhanh hơn do không quản lý ViewState
Tương tác với Javascript	Khó khăn vì các control được điều khiển bởi Server	Dễ dàng vì các đối tượng không do Server quản lý
URL	Không thân thiện	Thân thiện

CÂU HỎI, BÀI TẬP

1.1. Cho biết tên và chức năng của từng thành phần chính trong ứng dụng web ASP.NET MVC

BÀI 2: TẠO ỨNG DỤNG WEB ASP.NET MVC

Mã bài: 28.2

Giới thiệu:

Bài này cung cấp cho người học những kiến thức và kỹ năng để tạo mới một ứng dụng ASP.NET MVC sử dụng mã Razor và cài đặt Entity Framework.

Mục tiêu:

Trình bày cú pháp mã Razor

Tạo được ứng dụng web ASP.NET MVC

Cài đặt Entity Framework cho ứng dụng

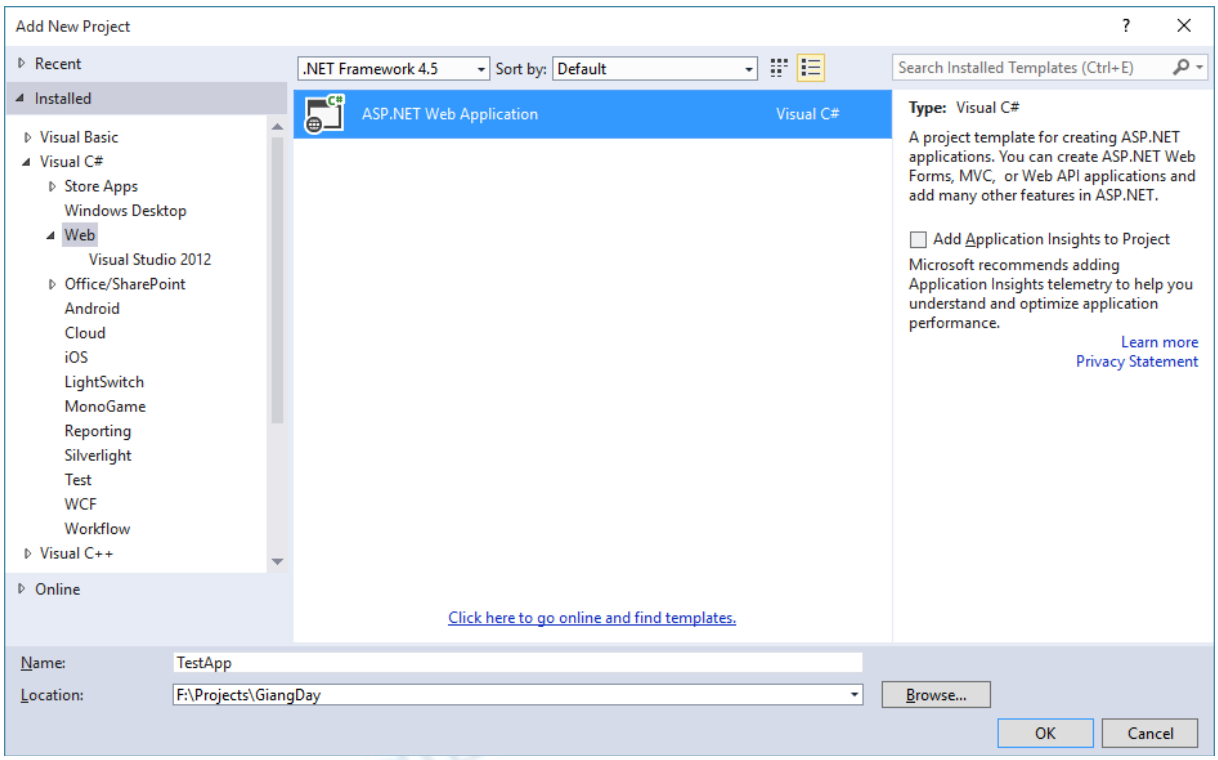
Cẩn thận thực hiện đúng các bước, chọn đúng ngôn ngữ, đường dẫn khi tạo ứng dụng

Nội dung chính:

1. Thao tác tạo

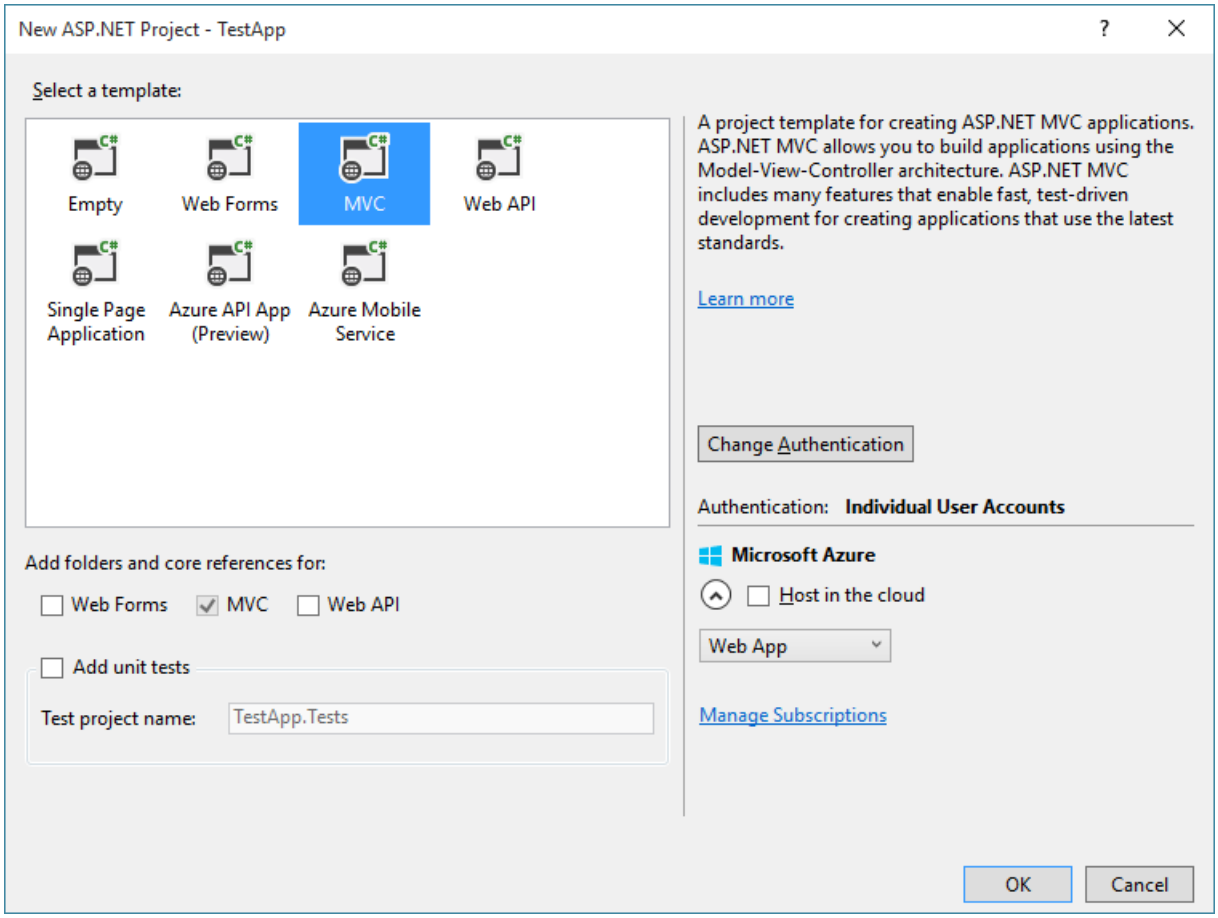
Khởi động Visual Studio 2015/2017/2019

Tạo mới project, chọn .NET Framework, thư mục và tên project theo hộp thoại dưới rồi chọn OK



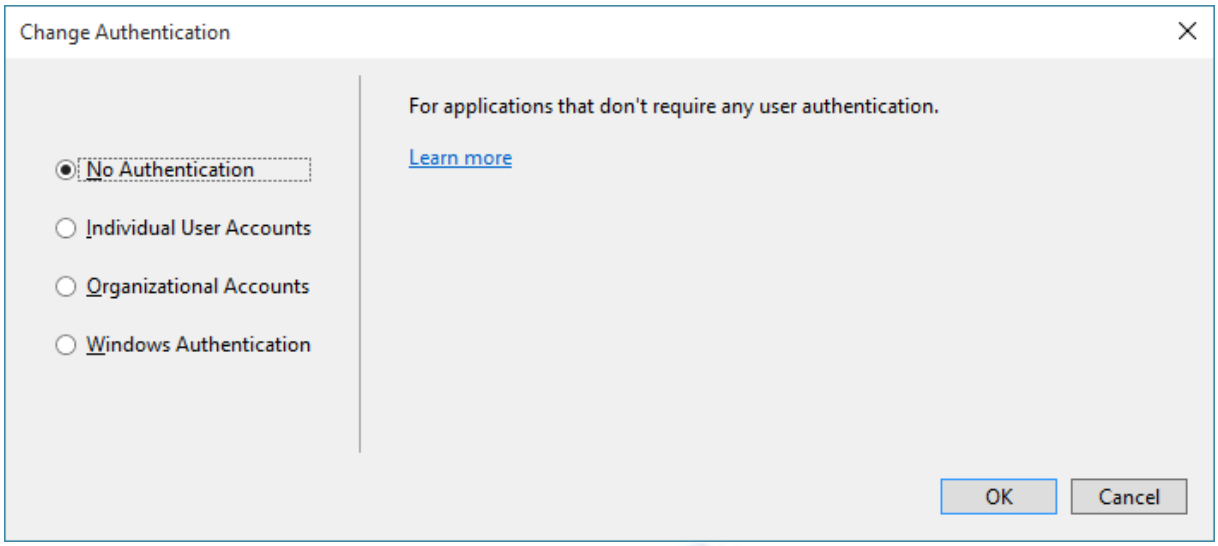
Hình 2.1. Tạo ứng dụng ASP.NET

Chọn MVC trong hộp thoại Select a template, chọn Change Authentication



Hình 2.2. Chọn mô hình MVC cho ứng dụng web

Chọn No Authentication



Hình 2.3. Chọn loại chứng thực quản lý người dùng

Tổ chức lưu trữ

Cấu trúc thư mục của ứng dụng ASP.NET MVC gồm 3 thực mục chính:

Controllers: lưu trữ các class đóng vai trò Controller

Models: lưu trữ các class Model

Views: lưu trữ các View

Các thư mục khác

Content: chứa các tập tin CSS, hình ảnh liên quan đến giao diện

Scripts: chứa các thư viện Javascript, jquery

App_Start: chứa các tập tin liên quan đến việc cấu hình cho các tính năng như: routes, filters, bundles

App_Data: lưu trữ tập tin dữ liệu XML hoặc local database, SQLite

Bin: lưu trữ các compiled của ứng dụng

Phân biệt ASPX và Razor View Engine

Giống nhau

Đều dùng để hiện thị giao diện trên Web Browser, có khái niệm về Master Page và User Control được định nghĩa với Layout (View để kế thừa Layout) và PartialView (subView)

Khác nhau