A picture containing text, poster

Description automatically generated

Phần mở đầu

* **Lý do chọn đề tài:**
* **Mục tiêu:**

Mục lục

Chương 1, Tổng quan về ASP.NET

1. Giới thiệu

* Trên thế giới phát triển ứng dụng và web hiện nay, có rất nhiều framework và ngôn ngữ được sử dụng bởi các nhà phát triển. Mỗi framework đều có những tính năng và lợi ích riêng, và trong số đó, **ASP.NET** đã trở thành một trong những nền tảng mã nguồn mở phổ biến và được ưa chuộng bởi các nhà phát triển vì những ưu điểm vượt trội của nó.
* **ASP.NET** là một framework phát triển ứng dụng web được phát triển bởi Microsoft. **ASP.NET** được xây dựng trên nền tảng **.NET Framework**, cho phép các nhà phát triển tạo ra các ứng dụng web động và tương tác mạnh mẽ với các nguồn dữ liệu.
* Phiên bản **ASP.NET** đầu tiên được triển khai là 1.0 và phiên bản **ASP.NET** mới nhất là phiên bản 4.6. **ASP.NET** được thiết kế để tương thích với giao thức **HTTP**. **HTTP** là giao thức chuẩn được sử dụng trên tất cả các ứng dụng web. cung cấp cho các nhà phát triển các công cụ mạnh mẽ để tạo ra các ứng dụng web chất lượng cao, đáp ứng được các yêu cầu phức tạp của doanh nghiệp. Nó hỗ trợ nhiều ngôn ngữ lập trình như **C#**, **VB.NET**, **F#** và các công nghệ như **HTML**, **CSS** và **JavaScript** để xây dựng các ứng dụng web đáp ứng và thân thiện với người dùng.
* **ASP.NET** cũng cung cấp nhiều tính năng để giúp các nhà phát triển xây dựng các ứng dụng web bảo mật và đáng tin cậy. Nó có thể tích hợp với các công nghệ bảo mật như **OAuth**, **OpenID** và **SSL** để đảm bảo an toàn cho người dùng và dữ liệu.
* Ngoài ra, **ASP.NET** còn có khả năng xử lý tải cao, giúp các ứng dụng web hoạt động mượt mà và hiệu quả hơn, đáp ứng được nhu cầu của các doanh nghiệp với lượng truy cập lớn.

Với các tính năng mạnh mẽ và khả năng tích hợp tốt với các công nghệ khác, **ASP.NET** là một lựa chọn lý tưởng cho các nhà phát triển ứng dụng web mong muốn xây dựng các ứng dụng chất lượng cao và đáp ứng nhu cầu của khách hàng.

ASP viết đầy đủ là **Active Server Pages**, và .NET là viết tắt của **Network Enabled Technologies**.

1. Các thành phần trong ASP.NET

**ASP.NET** là một **framework** phát triển ứng dụng web được phát triển bởi Microsoft. Nó cung cấp cho các nhà phát triển nhiều thành phần để tạo ra các ứng dụng web đáp ứng và thân thiện với người dùng. Dưới đây là một số thành phần quan trọng của **ASP.NET**:

* **ASP.NET Web Forms**: Web Forms là một công nghệ cho phép các nhà phát triển tạo ra các ứng dụng web truyền thống và động. Nó cho phép bạn tạo các trang web động bằng cách sử dụng các control trên trình duyệt như các textbox, label và button.
* **ASP.NET MVC**: MVC (Model-View-Controller) là một kiến trúc phát triển ứng dụng web được sử dụng phổ biến trong ASP.NET. Nó cho phép tách biệt các phần của ứng dụng web như dữ liệu, giao diện và logic xử lý, giúp dễ dàng quản lý và bảo trì ứng dụng.
* **ASP.NET Web API**: Web API là một công nghệ cho phép các nhà phát triển tạo ra các dịch vụ web RESTful (Representational State Transfer) để truyền tải dữ liệu giữa các ứng dụng.
* **ASP.NET SignalR**: SignalR là một thư viện cho phép các ứng dụng web thời gian thực với tính năng đẩy thông báo tới người dùng, kết nối thời gian thực và gửi dữ liệu giữa các ứng dụng.
* **Entity Framework**: Entity Framework là một ORM (Object-Relational Mapping) cho phép các nhà phát triển truy cập vào cơ sở dữ liệu một cách dễ dàng thông qua mã hóa đối tượng. Nó cho phép bạn tương tác với cơ sở dữ liệu như các đối tượng thay vì viết các truy vấn SQL trực tiếp.
* **ASP.NET Identity**: Identity là một khung xác thực và quản lý người dùng giúp cho các nhà phát triển có thể xây dựng tính năng đăng nhập, đăng ký và quản lý người dùng vào ứng dụng web của họ.

Tất cả các thành phần trên đều là những công nghệ quan trọng của **ASP.NET**, giúp cho các nhà phát triển xây dựng các ứng dụng web đáp ứng và thân thiện với người dùng, và làm cho **ASP.NET** trở thành một trong những nền tảng phát triển ứng dụng web phổ biến nhất trên thế giới.

Và ở dự án này, tôi chọn **ASP.NET MVC** để hoàn thành dự án của mình

1. Mô hình MVC

3.1 Giới thiệu về MVC

**MVC** là viết tắt của cụm từ **“Model-View-Controller“**. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. **MVC** là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. **MVC** chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

* **Model (dữ liệu)**: Quản lí xử lí các dữ liệu.
* **View (giao diện)**: Nới hiển thị dữ liệu cho người dùng.
* **Controller (bộ điều khiển)**: Điều khiển sự tương tác của hai thành phần **Model** và **View**

**MVC** được tiến sĩ **Trygve Reenskaug** đưa vào [ngôn ngữ lập trình](https://vietnix.vn/ngon-ngu-lap-trinh/) **Smalltalk-76** khi ông đến trung tâm Nghiên cứu **Xerox Palo Alto (PARC)** vào giữa năm 1970. Sau đó, việc triển khai trở nên phổ biến trong các phiên bản khác của **Small- Talk**. Năm 1988, các bài báo *“The Journal of Object Technology” – JOT* mang lại bước tranh toàn cảnh về **MVC** mang liệu sự hiệu quả tốt nhất.

**Luồng xử lý trong của mô hình MVC**, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, **Controller** xử lý **input** của user rồi giao tiếp với**Model** trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

*Ở đây,****View*** *không giao tiếp trực tiếp với* ***Model****. Sự tương tác giữa****View*** *và* ***Model****sẽ chỉ được xử lý bởi****Controller.***

* 1. Ưu và nhược điểm MVC
     1. Ưu điểm:
* Đầu tiên, nhắc tới ưu điểm mô hình **MVC** thì đó là băng thông ([**Bandwidth**](https://vietnix.vn/bandwidth-la-gi/)) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
* Một lợi thế chính của **MVC** là nó tách biệt các phần **Model**, **Controller** và **View** với nhau.
* Sử dụng mô hình **MVC** chức năng **Controller** có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
* Hỗ trợ [**TTD**](https://en.wikipedia.org/wiki/Test-driven_development) (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won [**test case**](https://vietnix.vn/test-case-la-gi/).
* Phiên bản mới nhất của **MVC** hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ **View** của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.
  + 1. Nhược điểm:
* Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.
* MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.
  1. Vì sao nên sử dụng mô hình MVC
     1. Quy trình phát triển nhanh hơn
* **MVC** hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một **mô hình MVC** được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên **View** và một developer khác có thể làm việc với **Controller** để tạo logic nghiệp vụ cho ứng dụng web đó.
* Do đó, **ứng dụng mô hình MVC** có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.
  + 1. Khả năng cung cấp nhiều chế độ View
* Trong **mô hình MVC**, bạn có thể tạo nhiều View cho **chỉ một mô hình**. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời.
* Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.
  + 1. Các sửa đổi không ảnh hưởng đến toàn bộ mô hình
* Đối với bất kỳ ứng dụng web nào, người dùng có xu hướng thay đổi thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng…
* Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.
  + 1. MVC Model trả về dữ liệu mà không cần định dạng
* **MVC pattern** có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.
* ***Ví dụ:*** tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng **Macromedia Flash** hay **Dream Viewer.**

1. Các bước để xây dựng Website bằng MVC 5

Chương 2, Xây dựng Website quản lý khách sạn

1. Phát biểu bài toán
2. Phân tích thiết kế và cài đặt
   1. Phân tích
   2. Thiết kế
   3. Cài đặt
3. Hướng dẫn sử dụng

Phần Kết luận