**Buổi 1-2**

1. **ASP.NET là gì ?**

* Là một Framework miễn phí để xây dựng trang web và ứng dụng trên .NET Framework sử dụng HTML, CSS và JavaScript

1. **ASP.NET MVC là gì ?**

* Là một Framework được cung cấp bởi microsoft dựa trên kiến trúc Model-View-Controller để xây dựng ứng dụng và trang web

1. **Ưu / Nhược điểm WebForm ?**

* Không có sự tách biệt rõ ràng 3 tầng trong mô hình lập trình web.
* Trong WebForm sử dụng 1 đối tượng ViewState mà đối tượng này làm cho web bị tăng kích thước vì vậy làm giảm hiệu năng của ứng dụng.
* Giao diện được thiết kế dựa trên những toolbox có sẵn mà webform cung cấp mà hiện nay designer thường sử dụng HTML, CSS để thiết kế giao diện web. Vì webform gây khó khăn trong việc thiết giao diện web.
* Việc thiết kế giao diện trở nên mất linh hoạt.

1. **MVC là gì ?**

* MVC là viết tắt (Model-View-Controller)
* Mô hình MVC giúp tách biệt 3 tầng trong mô hình lập trình web
* Giúp tối ưu ứng dụng, dễ dàng thêm mới và chỉnh sửa code hoặc giao diện
* **Model**: Đại diện cho hình dạng của dữ liệ. Một Class trong C# được sử dụng để mô tả 1 model. Các đối tượng model lưu trữ dữ liệu được truy xuất từ CSDL
* **View:** là phần giao diện dành cho user. View là phương tiện hiển thị cácđối tượng trong 1 ứng dụng
* **Controller:** là bộ phận xử lý các yêu cầu từ user đưa đến thông qua view

1. **Cơ chế hoạt động MVC ?**

* User gửi 1 yêu cầu tới server bằng cách truyền vào 1 URL trong browser.
* Yêu cầu đso được gửi tới controller đầu tiên, controller sẽ xử lý yêu cầu, Nếu yêu cầu truy xuất dữ liệu thì controller sẽ chuyển qua tầng model.
* Tại tầng model, dữ liệu được truy xuất từ database và sau đó truyền qua view thông qua controller.
* Controller sử giúp dữ liệu được chuyền từ model qua view.
* View là tần cuối cùng giao tiếp với user, mọi dữ liệu sẽ được hiển thị cho User.

1. **Ưu điểm ASP.NET MVC 5 ?**

* Các tầng trong mô hình lập trình Web tách biệt nên tối ưu ứng dụng và dễ dàng trong việc viết code, giao diện
* ASP.Net MVC sử dụng công nghệ thiết kế web HTML, CSS nên việc thết kế dễ dàng và giúp designer linh hoạt trong việc thiết kế
* ASP.Net MVC không sử dụng view state vì vậy trang web không bị tăng kích thước do đó hiệu năng không bị giảm

1. **So sánh ASP.NET vs ASP.NET MVC ?**

|  |  |  |
| --- | --- | --- |
| **Tính năng** | **ASP .NET 2.0** | **ASP .NET MVC** |
| Kiến trúc chương trình | Kiến trúc mô hình WebForm->Business->Databases | Sử dụng việc phân chia chương trình thành Controller, View Model |
| Cú pháp chương trình | Sử dụng cú pháp của webform, tất cả các sự kiện và controller do server quản lý | Các sự kiện được điều khiển bởi controllers, các controls không do server quản lý |
| Truy cập dữ liệu | Sử dụng hầu hết các công nghệ truy cập dữ liệu trong ứng dụng | Phần lớn sử dụng LINQ to SQL class để tạo mô hình truy cập đối tượng |
| Debug | Chương trình phải thực hiện tất cả bao gồm các lớp truy cập dữ liệu, sự hiển thị, điều khiển các controls | Có thể sử dụng các unit test kiểm tra các phương thức trong controller |
| Tốc độ phân tải | Tốc độ phân tải chậm trong trang có quá nhiều các controls vì ViewState quá lớn | Phân tải nhanh hơn do không phải quản lý ViewState để quản lý các controls |
| Tương tác với javascript | Tương tác với javascript khó khan vì các controls dc điều khiển bởi server | Tương tác với javscript dễ dàng vì các đối tượng không do server quản lý |
| URL Address | Cấu trúc địa chỉ URL có dạng  <filename>.aspx?&<params> | Cấu trúc địa chỉ rõ ràng, rành mạch theo dạng  Controllers/ActionId |

1. **Cần chuẩn bị gì để tạo ứng dụng ASP.NET MVC 5 ?**

* Hệ điều hành từ Window 7 trở lên .
* Visual Studio 2013 trở lên.
* .NET Framework 4.0 trở lên.
* SQL Server 2012 trở lên.

1. **Trình bày các bước tạo project ASP.NET MVC 5 ?**

* Chọn File -> New Project .
* Chọn ASP.NET web application(.NET Framework).
* Chọn MVC và No Authentication.
* Chọn Ok.

**Buổi 1-2**

1. **Routing MVC là gì ?**

* Là định tuyến xác định mẫu URL và thông tin xử lý
* Tất cả các tuyến được cấu hình của một ứng dụng được lưu trữ trong RouteTable và được sử dụng bởi công cụ định tuyến để xác định controller thích hợp

1. **Controller MVC là gì ?**

* Là nơi xử lý các yêu cầu của user đưa đến thông qua view, từ đó đưa ra các dữ liệu phù hợp với user

1. **Action Method là gì ?**

* Tất cả các phương thức của lớp controller
* Action method không thể được overloaded
* Action method không phải
* Có 3 Action method:
  + NonAction
  + ActionResult

1. **Action Result là gì ?**

* Là kiểu dữ liệu trả về của 1 method trong Class Controller

1. **Action selector là gì ?**

* Các Attribute gắn với các phương thức Action
* Giúp Route Engine lựa chọn chính xác các method Action tương ứng khi đáp ứng các yêu cầu HTTP Request

**BUỔI 3 - 4**

**LINQ là gì ?**

+ là một tập hợp cá thành phần mở rộng cho phép viết các câu truy vấn dữ liệu ngay trong một ngôn ngữ lập trình như C# hoặc VB.Net

+ khi tạo một đối tượng LINQ thì Visual Studio sẽ tự động sinh ra các lớp có thành phần tương ứng với CSDL của chúng ta

+ Điểm mạnh: viết truy vấn cho rất nhiều đối tượng dữ liệu từ CSDL, XML, Data Object … thậm chí là viết truy vấn cho một biến mẳng đã tạo ra trước đó. Vì vậy mới co các khái niệm LinQ to SQL, LinQ to XML, LinQ to Object

+ Điểm yếu: Chậm và thiếu nhất quán

**Entity Framework là gì ?**

+ là một khung ORM ( Object Retional Mapper) mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ

+ là 1 phàn của NET Framework

+ là 1 mã nguồn mở ORM Framework

**Tại sao phải dùng Entity Framework**

+ Nếu thực hiện truy vấn data , ta có thể dụng Linq to Entities để thao tác với objects được sinh ra từ Entity Framework, nghĩa là không phải viết code sql

+ Việc update các classes , commands dễ dàng mỗi khi CSDL có sự thay đổi

+ tiết kiệm thời gian thực hiện các xử lý

+ Tự động tạo ra các classes , commands tương ứng cho việc select, update, delete dữ liệu từ CSDL quan hệ

**Tính năng của EF**

+ Hỗ trợ stored procedure

+ làm việc với bất kỳ CSDL quan hệ nào có Entity Framework hợp lệ

+ tạo truy vấn SQL từ LINQ to Entities

+ được phát triển như một sản phẩm mã nguồn mở

+ cho phép tạo những câu lệnh thêm xóa, sửa, cập nhật

+ theo dõi những thay đổi của các đối tượng trong bộ nhớ

**Khi nào sử dụng Entity Framework**

**HTML Helper là gì ?**

**- Razor là gì ? Mục đích của Razor là gì ?**

+ là 1 cú pháp đánh dấu cho phép nhúng server-base code và các trng web sử dụng C# và VB.Net

+ Razor không phải là 1 ngôn ngữ lập trình, nó là một ngôn ngữ đánh đấu phía máy chủ (server side markup language)

**- Mục đích chính:**

+ Sự phối hợp giữa ngôn ngữ lập trình C# hoặc VB.Net với mã HTML 1 cách dễ dàng

+ dễ học và dễ viết

+ Không phụ thuộc vào ASP.NET runtime

- Những điều cần nhớ trong Razor:

+ sử dụng @ để viết phía máy chủ

+ khối mã phía máy chủ bắt đầu @(\*code\*) sử dụng @ để hiển thị văn bản tử khối mã.

+ câu lệnh điều kiện bắt đầu bằng @if ()

+ Vòng lặp bắt đầu bằng @for

+ @model cho phép bạn sử dụng đối tượng model ở bất cứ đâu trong View

**@model: dùng để nhận data từ controller**

**@Model: dùng để lấy data nhận được hiển thị ra**

**Chia sẻ data là gì ?, có mấy cách truyền data ?**

+ là data được tạo ra từ thành phần này mà thành phần khác được sử dụng.

Có 2 cách truyền data:

+ controller sang view và từ view sang controller

**Session là gì ?**

+ là phiên làm việc được tính từ lúc bắt đầu truy cập website cho đến khi đóng của số trình duyệt hoặc hến hạn sử dụng

**Ứng dụng Session ?**

+ sử dụng làm giỏ hàng vì giỏ hàng cần duy trì trong suốt quá trình tìm kiếm và chọn mua hàng

+ Duy trì tài khoản đăng nhập vì tải khoản đăng nhập cần được duy trì trong suốt phiên làm việc để khi cần có thể sử dụng

**Application là gì ? ứng dụng ?**

+ là phạm vi chia sẻ dữ liệu trên toàn ứng dụng (tất cả user có thể tạo và sử dụng)

+ Cache dữ liệu người truy cập (đếm số người truy cập, Queue mails, Queue chat messages)

**Data Validation là gì ? có mấy cách dùng validation ?**

**- Data validation là kiểm tra dữ liệu đầu vào để xem dữ data đó thỏa các điều kiện sau:**

+ đúng định dạng

+ không chứa mã động

+ nằm trong phạm vi cho phép

* **Có 3 cách để validation**

+ dùng Model.State để kiểm tra bên controller

+ dùng Data Annotation để bên Model

+ Code thuần như check IsNullEmpty hoặc IsNullEmptyOrWhiteSpace

**ModelState** là một thuộc tính của controller được sử dụng để các xác thực biểu mẫu ở phía máy chủ và nó là tập các cặp name và value được gửi đến máy chủ trong quá trình post

**Cookie** là một file text nhỏ được lưu trong máy của người dùng. Nó lưu trữ 1 số thông tin về người dùng như: thông tin đăng nhập, các lựa chọn cá nhân hóa …

**Cookie** được sinh ra khi user dùng 1 trình duyệt để vào 1 trang web mà sử dụng cookie để lưu lại thông tin , trạng thái hiện tại của user

**Ưu điểm** :

* Truy cập nhanh chóng (do lưu trữ trên máy client)
* Dễ dang duy trì

**Nhược điểm:**

* Độ bảo mật không cao
* Người dùng có quyền thao tác với file cookie

**Khia nào nên sử dụng cookie:**

* Chỉ nên sử dụng khi muốn lưu những thông tin không yêu cầu bảo mật cao và được sử dụng trong suốt phiên làm việc

**Cách tạo Cookie trong ASP.NET:**

* Chúng ta thường sử dụng đối tượng Response để tạo
* Response.Cookies[“TTSV”][“hoten”] = “Nguyễn Tuấn Vũ”;
* StudenCookies.Expires = DateTime.Now.AddHours();

**Binding Attribute**

trong ASP.NET MVC framework cũng cho phép ta chỉ định thuộc tính nào của model class mà ta muốn liên kết. Thuộc tính [Bind] sẽ cho phép ta chỉ định các thuộc tính chính xác của một model nên bao bồm hoặc loại từ trong Bind

Có 3 cách Bind:

+ Bind nguyên thủy

+ Bind theo class model

+ Bind theo FormCollection