**PHÁT TRIỂN ỨNG DỤNG WEB**

*Mục tiêu: Tìm hiểu cách xây dựng 1 website động dựa trên .NET*

Hình thức đánh giá:

* 40%: Bài tập nhóm
* 60%: Thi cuối kỳ trên máy

Yêu cầu:

* HTML, CSS, Javascript
* SQL Server
* Internet

Thực hành:

* Lab 1, 2: Data Access
* Lab 3, 4: Front-end
* Lab 5, 6: Back-end
* Lab 7: Summary

Tài liệu tham khảo:

<https://www.tutorialspoint.com/asp.net/index.htm>

<http://noobtuts.com/>

# Các thành phần của Website

1. Header
2. Slider
3. Banner
4. Content area
5. Footer

# Các bước phát triển Website

1. Định hướng

* Ý tưởng tổng quan: giao diện (logo, màu đồng bộ), các thành phần,…
* Mục đích cần đạt được là gì? (có hỗ trợ nhiều ngôn ngữ hay không?)
* Đối tượng nhắm tới là những ai? (trẻ em, thanh thiếu niên, trung niên, người lớn tuối)
* Thông tin gì đã có trong tay và sử dụng chúng như thế nào?

1. Cơ sở dữ liệu

* Tạo CSDL cho website
* Tổ chức cơ sở dữ liệu
* Thiết kế bảng, thủ tục, hàm

1. Cấu trúc ADO (ActiveX Data Object)

* Xây dựng mô hình ADO cho website
* Xây dựng mô hình kết nối từ ADO tới CSDL
* Thử nghiệm kết nối
* Xây dựng tập tin cấu hình cho kết nối

1. Xây dựng website

* Xây dựng từng trang trong website
* Front-end và Back-end có thể song song
* Test thử hệ thống

1. Triển khai hệ thống

* Test thử nội bộ
* Publish website
* Mua tên miền, hosting
* Triển khai
* Bảo trì hệ thống

# 3-tier model

1. Presentation tier
2. Business tier
3. Data tier

# ASP.net

Cấu trúc thư mục dự án:

1. DB (chứa cơ sở dữ liệu)
2. Data (Resources): icon, png, Excel
3. Project
4. Documents

Cấu trúc thư mục website:

1. Images
2. Styles
3. Javascript

**BASIC CONTROLS IN ASP.NET**

1. Server side:

Server side là phần của trang web được xây dựng bởi các ngôn ngữ lập trình như C#, Visual Basic, Java,…

Các phương thức chính:

* HtmlEncode, HtmlDecode: Chuyển đổi một chuỗi bình thường thành chuỗi HTML hợp lệ và ngược lại.

Ví dụ: Server.HtmlDecode (“<a>Chào</a>”) thành “Chào”

* UrlDecode: Chuyển đổi một chuỗi địa chỉ thành chuỗi bình thường.
* MapPath: Trả về đường dẫn vật lý của một file trong website.

Ví dụ: Server.MapPath(“index.aspx”);

Các thuộc tính chính:

* Cookies
* FilePath
* Files
* Form
* QueryString

1. Client side:

* Client side scripts: onblur, onfocus, onclick,…
* Client side source code: tập tin nội dung và định dạng (.aspx), tập tin mã nguồn thực thi (.cs)
* HTML Server controls: Các control của client nhưng xử lý ở server, sử dụng thuộc tính runat=“server”

1. User control

Là những control tạo riêng, để nhúng vào một phần của trang web.

Có đuôi mở rộng là .ascx và .cs

Không chứa các thẻ <html>, <body>, <form> nhưng có thể chứa các thẻ khác.

# MỘT SỐ CONTROL NÂNG CAO TRONG ASP.NET

1. Các chỉ thị (Directives)

ASP.net không chỉ đơn thuần là 1 trang web mà còn bao gồm nhiều thành phần khác được nhúng vào.

* Application Directive:

<%@ Application Language=“C#” %>

* Assembly Directive:

<%@ Assembly Name=“name” %>

* Implement :
* Import
* Master

1. Quản lý trạng thái

* ViewState: (Phải được bật trước khi sử dụng)
* SessionState: Quản lý phiên làm việc của một website, lưu trữ thông tin đăng nhập hoặc đếm số lượt xem.

string text = “abcdef”;

Session[“text”] = text;

string newtext = Session[“text”];

* ApplicationState

\* Có thể viết đoạn mã C# ngay trong file .aspx:

<%

//…

%>

Hoặc:

<script runat=“Server”>

//Phương thức tại đây

</script>

1. Validators

Kiểm tra tính chính xác của dữ liệu nhập vào tại client, thay vì xử lí tại Server.

* RequiredFieldValidator
* RangeValidator
* CompareValidator
* RegularExpressionValidator
* CustomValidator
* ValidationSummary

1. File Uploading

ASP.net cung cấp hai control cho phép tải tập tin từ người dùng lên Server. Bao gồm: HtmlInputFile và FileUpload

Chọn đường dẫn của file 🡪 Đổi tên file (lưu tên cũ + tên mới trong cơ sở dữ liệu) 🡪 Lưu vào hệ thống

1. Ad rotator

Cho phép lựa chọn các banner từ một danh sách lưu trữ trong tập tin XML.

Bên trong file XML:

<Advertisement>

<Ad>

…

</Ad>

</Advertisement>

1. Calendar

Cung cấp khả năng điều khiển lịch dương.

1. Panel Controls

# DATABASE ACCESS AND CONNECTION

1. Database Access

Asp.Net cho phép truy xuất các nguồn dữ liệu từ database, XML, Business Objects, Flat files,…

Sử dụng kiến trúc đa tầng (n-tier).

1. ADO.net
2. Data Provider classes

* SqlConnection
* SqlCommand
* SqlDataReader

1. Singleton pattern
2. N-tier model

# DATA CONTROLS BINDING

1. Data Sources

Một data source chính là một nguồn dữ liệu được kết xuất từ 1 hoặc nhiều bảng trong Database.

Data Souce có thể được lấy từ Database, XML hoặc tự tạo bằng class (Oriented-Object Database Management System – OODMS)

Một data source có thể có các tính chất của một cơ sở dữ liệu, có thể thêm, xoá, sửa, sắp xếp,…

1. Data Binding

Là quá trình chuyển dữ liệu từ Data Sources vào một công cụ hiển thị dữ liệu để cho người dùng thấy kết quả.

Công cụ nhận dữ liệu có thể là Repeater, DataList, GridView,… hoặc ComboBox.

Quá trình chuyển dữ liệu được thực hiện bởi phương thức DataBind().

1. Data Controls

* Repeater : Lặp lại đúng số dòng trong Data Source.
* DataList: Hiển thị dữ liệu trong CSDL theo danh sách. Mặc định dùng table để hiển thị dữ liệu.
  + Cập nhật và xoá dữ liệu thông qua Data List.
* GridView:

# PAGING

* Quá trình thực hiện phân trang trong ASP.NET khi có nhiều dữ liệu.
* Có thể phân trang tại CSDL hoặc xử lý trên RAM.

Tại CSDL:

* + Thêm cột STT cho mỗi record
  + Lấy tổng số record
  + Lấy record có STT nằm trong phạm vi cho phép

# CONTROL BINDING ADVANCED

1. AJAX Control

* AJAX là viết tắt của Asynchronous JavaScript and XML.
* **The ScriptManager Control:** Là một Control quan trọng nhất, phải được đặt lên trang cho các control khác hoạt động
* **The UpdatePanel Control:** là một Control dùng để chứa các Control khác muốn đồng bộ hóa
* **The UpdateProgress Control:** cung cấp một cơ chế chờ đợi khi đang tải dữ liệu từ phía Server bởi một Control khác
* **The Timer Control**: khởi tạo việc lấy dữ liệu một cách tự động. Có 2 cách để thực hiện.

1. Personalization:

* Cho phép một website ghi nhớ các thông tin người dùng và hiển thị khi cần thiết.

1. Error Handling:
2. Security

* MD5 Verify
* Authentication images
* SSL - HTTPS

1. Deployment
2. Applications