

CHUYÊN ĐỀ ASP.NET

Gv: Đoàn Phước Miền
Email: phuocmien@tvu.edu.vn



Nội dung

- Giới thiệu về ASP.net
- Ngôn ngữ C Sharp
- Control cơ bản trong ASP.net
- Control nâng cao trong ASP.net
- ASP.net và CSDL
- Project demo

Giới thiệu về ASP.NET

- ASP.NET là một framework phát triển ứng dụng web mạnh mẽ và đa dạng, được phát triển bởi Microsoft.
- Tên gọi "ASP.NET" viết tắt của "Active Server Pages.NET", là một phiên bản tiếp theo và phát triển từ công nghệ ASP (Active Server Pages). ASP.NET giúp nhà phát triển xây dựng các ứng dụng web chất lượng cao, an toàn và hiệu suất tốt.

Giới thiệu về ASP.NET

▪ Giới thiệu

- Trong thế giới ngày nay, ứng dụng web đã trở thành một phần quan trọng của cuộc sống và kinh doanh.
- ASP.NET ra đời để giải quyết các thách thức trong việc xây dựng ứng dụng web một cách hiệu quả và linh hoạt hơn.
- ASP.NET là một framework phát triển ứng dụng web mạnh mẽ được phát triển bởi Microsoft.
- Với sự tích hợp sâu sẽ các dịch vụ của Microsoft và khả năng tùy chỉnh mạnh mẽ, ASP.NET là lựa chọn lý tưởng cho việc phát triển ứng dụng web.

Giới thiệu về ASP.NET

▪ Lợi ích của ASP.NET

- Cung cấp môi trường phát triển mạnh mẽ và dễ dàng sử dụng.
- Hỗ trợ xây dựng ứng dụng web an toàn và bảo mật.
- Cung cấp các công cụ tích hợp để triển khai ứng dụng lên môi trường đám mây.

▪ Các thành phần chính của ASP.NET

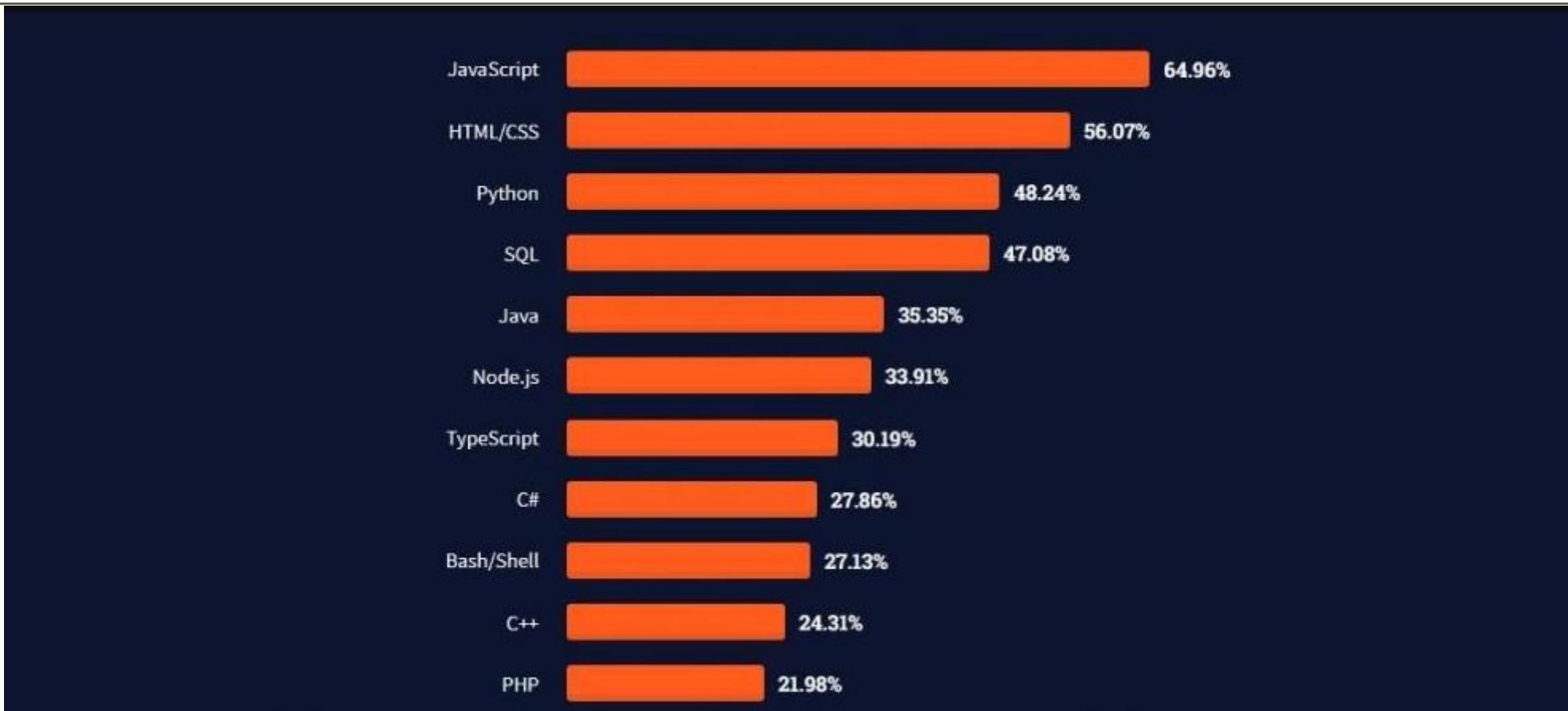
- ASP.NET gồm nhiều thành phần giúp chúng ta xây dựng ứng dụng web.
- Web Forms: Cho phép chúng ta tạo giao diện dựa trên quy trình phát triển theo kiểu hình mẫu.
- MVC (Model-View-Controller): Cung cấp kiến trúc phân tách mô hình, giao diện và điều khiển.
- Razor Pages: Đơn giản hóa quy trình xây dựng giao diện bằng cách tập trung vào từng trang riêng lẻ.

Giới thiệu về ASP.NET

■ Kiến thức cần có trước khi học ASP.NET

- Để hiểu rõ hơn về ASP.NET, chúng ta cần có kiến thức cơ bản về:
 - Lập trình C# hoặc một ngôn ngữ lập trình hướng đối tượng khác.
 - Cơ bản về HTML, CSS và JavaScript.
- Tuy nhiên, không cần phải thành thạo mọi thứ ngay từ đầu. Chúng ta sẽ đi từng bước và dần dần nắm vững.

Giới thiệu về ASP.NET



Giới thiệu về ASP.NET

- JavaScript nổi bật với khả năng thích ứng với nhiều trình duyệt web và cú pháp linh hoạt của nó.
- Mặc dù JavaScript là ngôn ngữ lập trình dành riêng cho giao diện người dùng nhưng nó vẫn có thể sử dụng được cho phần phụ trợ thông qua Node.js.
- JavaScript cũng là ngôn ngữ lập trình giúp website chạy mượt mà, ổn định và mang lại trải nghiệm tốt nhất cho người dùng.
- JavaScript - ngôn ngữ lập trình phổ biến nhất hiện nay

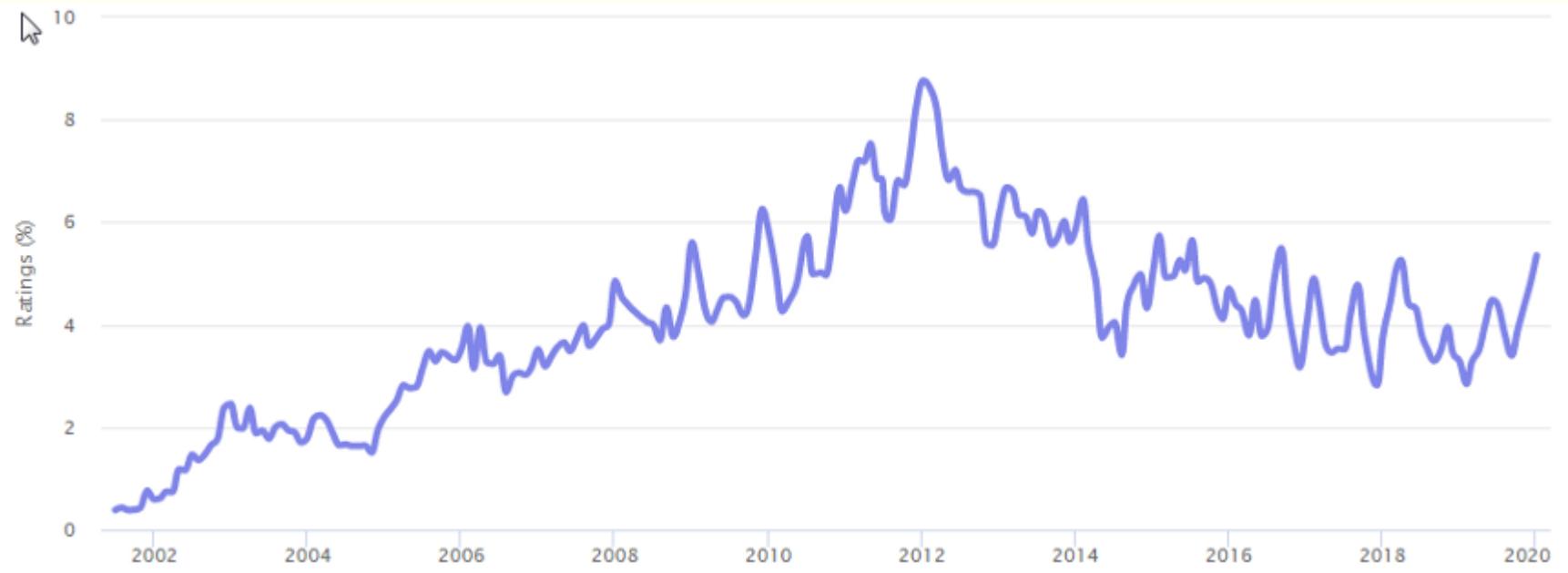
Giới thiệu về ASP.NET

- Trong tất cả các ngôn ngữ lập trình phổ biến hiện nay, Python được bình chọn trong top 2 ngôn ngữ lập trình thú vị nhất để học (phiếu khảo sát của PYPL).
- Python cực kỳ dễ tiếp cận đối với người mới bắt đầu nhờ cú pháp rõ ràng và trực quan của nó.
- Ngôn ngữ này nổi tiếng với ưu điểm: chặt chẽ - nhanh chóng - mạnh mẽ và ngày nay được áp dụng cho mọi hệ điều hành.
- Đối với những người muốn theo đuổi sự nghiệp lập trình phụ trợ, Python là lựa chọn hoàn hảo để bắt đầu.

Ngôn ngữ C sharp

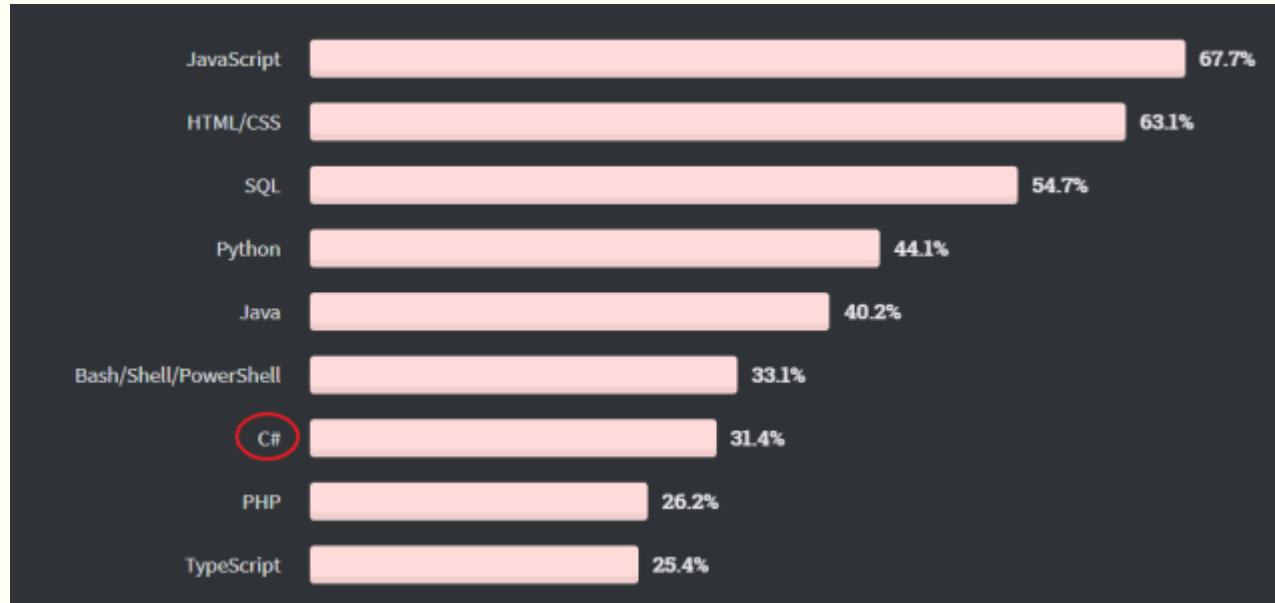
- Được phát triển và ứng dụng vào năm 2000 bởi “ông lớn” Microsoft, C-Sharp (hay C#) là xuất phát điểm trong kế hoạch .NET của hãng công nghệ hàng đầu này.
- C# được coi là một ngôn ngữ lập trình cực kỳ linh hoạt.
- Ngôn ngữ này không chỉ dễ tiếp cận và sử dụng cho người mới mà C# còn có tính bảo mật cực cao.
- Chính vì vậy, C# được mệnh danh là một trong những ngôn ngữ lập trình “khủng” nhất hiện nay.
- C# thường được sử dụng trong phát triển ứng dụng web và máy tính để bàn, cũng như phát triển trò chơi VR, 2D và 3D...

Ngôn ngữ C sharp



Trang web xếp hạng ngôn ngữ phổ biến TIOBE đã xếp thứ 5 với mức tăng đáng kể

Ngôn ngữ C sharp



Cuộc khảo sát dành cho nhà phát triển StackOverflow đã xếp C# là ngôn ngữ phổ biến thứ 4 (Công nghệ phổ biến thứ 7 trong năm 2020):

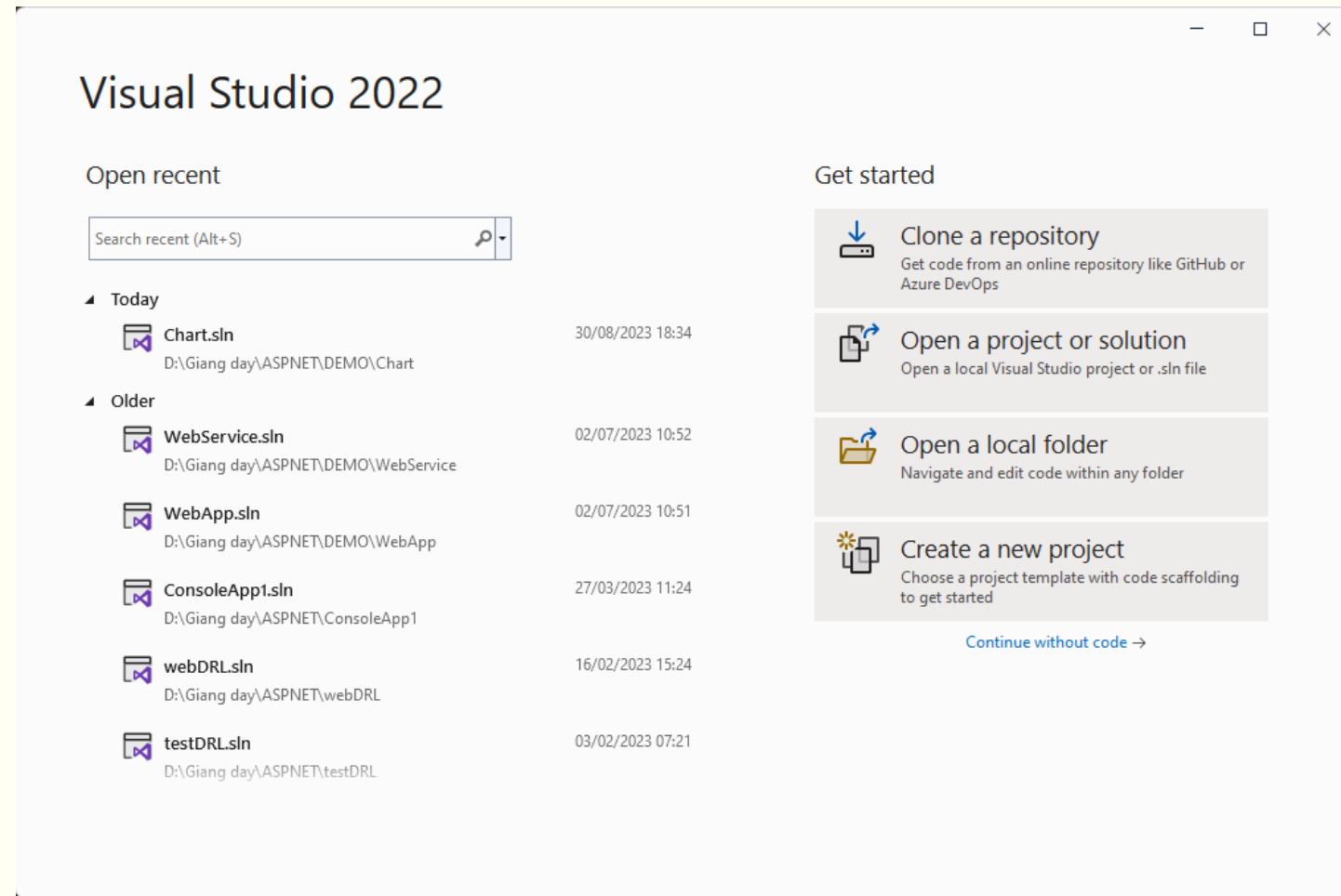
Ngôn ngữ C sharp

- **Ứng dụng**
 - Server-Side programming – Lập trình phía Server
 - App development – Phát triển ứng dụng
 - Web Application development – Phát triển ứng dụng Web
 - Game Development – Phát triển trò chơi
 - Software for Windows Platform – Phần mềm cho Nền tảng Windows
- **Ưu điểm:**
 - Gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java, Pascal.
 - Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của chúng.
 - Cải tiến các khuyết điểm của C/C++ như con trỏ, hiệu ứng phụ,....
 - Dễ dàng tiếp cận, dễ phát triển.
- **Nhược điểm:**
 - Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows.

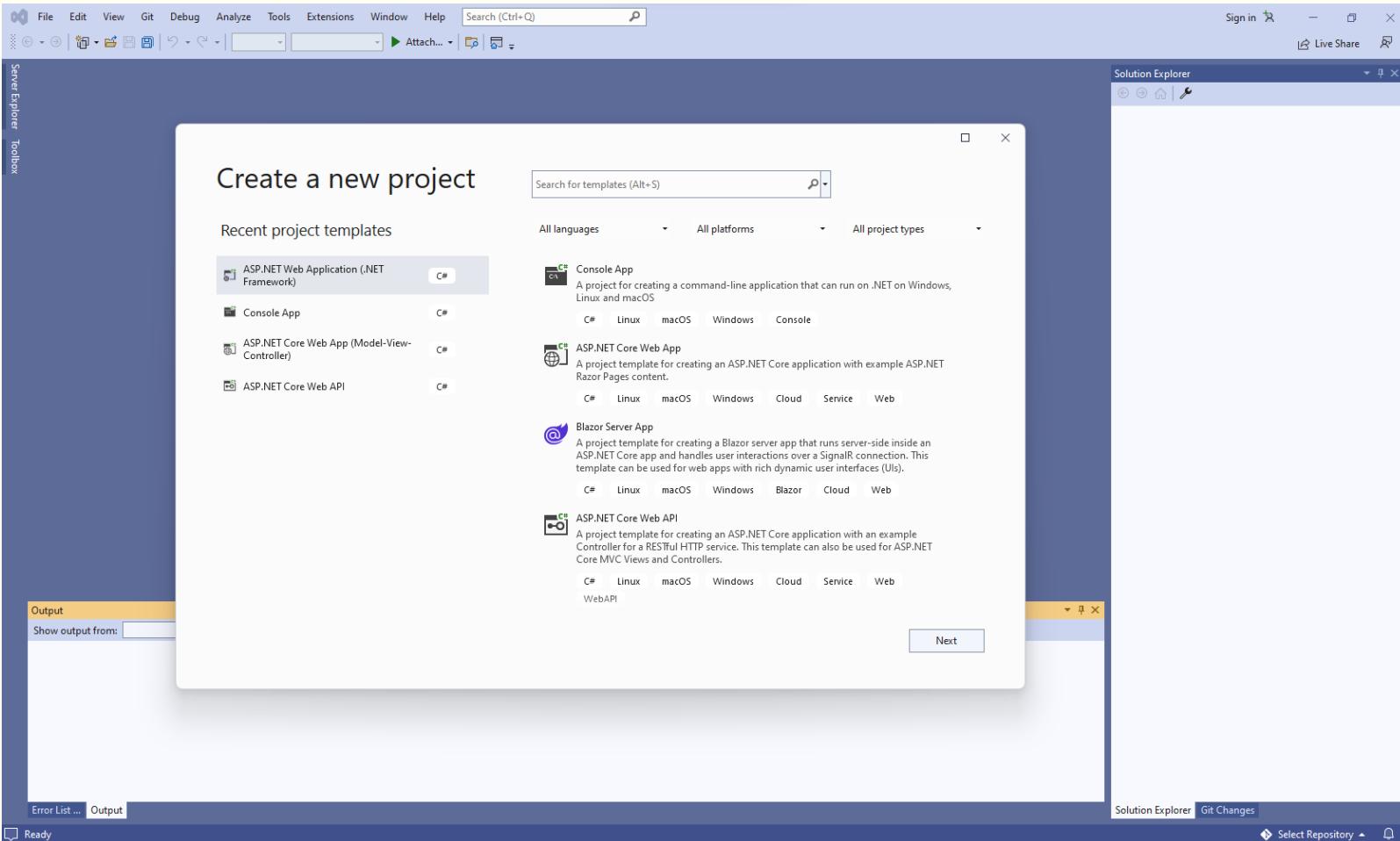
Ngôn ngữ C sharp

- **Ứng dụng**
 - Server-Side programming – Lập trình phía Server
 - App development – Phát triển ứng dụng
 - Web Application development – Phát triển ứng dụng Web
 - Game Development – Phát triển trò chơi
 - Software for Windows Platform – Phần mềm cho Nền tảng Windows
- **Ưu điểm:**
 - Gần gũi với các ngôn ngữ lập trình thông dụng như C++, Java, Pascal.
 - Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của chúng.
 - Cải tiến các khuyết điểm của C/C++ như con trỏ, hiệu ứng phụ,....
 - Dễ dàng tiếp cận, dễ phát triển.
- **Nhược điểm:**
 - Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework. Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác, hầu hết phải dựa vào windows.

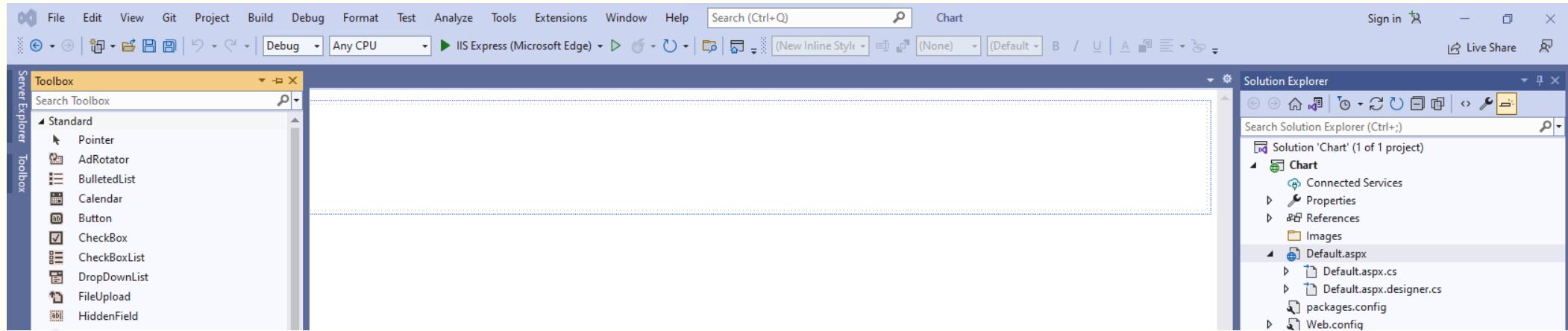
GIAO DIỆN



GIAO DIỆN



GIAO DIỆN



Control cơ bản trong ASP.NET

ASP.NET cung cấp nhiều loại điều khiển (controls) cơ bản để xây dựng giao diện người dùng trong ứng dụng web. Dưới đây là một số điều khiển quan trọng và cơ bản trong ASP.NET:

- **TextBox (TextBoxControl):** Điều khiển cho phép người dùng nhập và chỉnh sửa văn bản. TextBox được sử dụng rộng rãi để lấy thông tin từ người dùng như địa chỉ email, tên người dùng, v.v.
- **Button (ButtonControl):** Điều khiển nút (button) thường được sử dụng để khởi động các hành động hoặc sự kiện khi người dùng nhấn vào nút đó.
- **Label (LabelControl):** Điều khiển label dùng để hiển thị văn bản tĩnh (không thể chỉnh sửa) trên giao diện. Label thường được sử dụng để hiển thị thông tin, hướng dẫn hoặc nhãn cho các phần khác.

Control cơ bản trong ASP.NET

- **DropDownList (DropDownListControl)**: Điều khiển thả xuống (dropdown) là một danh sách các tùy chọn mà người dùng có thể chọn một trong số đó. Nó thường được sử dụng để chọn giá trị từ một danh sách.
- **CheckBox (CheckBoxControl)**: Điều khiển checkbox cho phép người dùng chọn hoặc bỏ chọn một hoặc nhiều tùy chọn.
- **RadioButton (RadioButtonControl)**: Điều khiển nút radio được sử dụng để cho phép người dùng chọn một lựa chọn duy nhất từ một tập hợp các tùy chọn.
- **HyperLink (HyperLinkControl)**: Điều khiển siêu liên kết (hyperlink) cho phép người dùng nhấn vào một liên kết để chuyển đến một trang web khác.

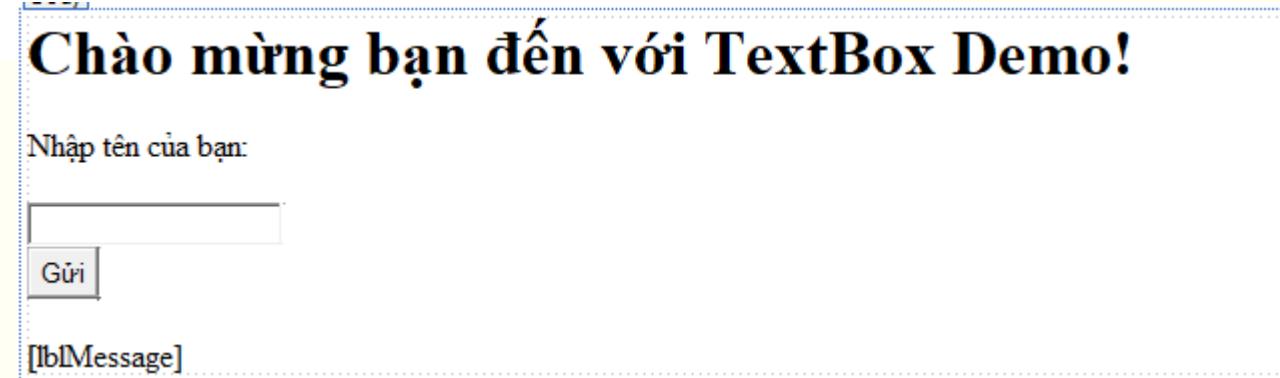
Control cơ bản trong ASP.NET

- **Image (ImageControl):** Điều khiển hình ảnh cho phép hiển thị hình ảnh trên giao diện. Nó thường được sử dụng để hiển thị biểu tượng, logo hoặc hình ảnh khác.
- **Literal (LiteralControl):** Điều khiển literal cho phép hiển thị văn bản tĩnh hoặc mã HTML trực tiếp trên giao diện. Điều này rất hữu ích khi bạn muốn hiển thị nội dung tĩnh không cần xử lý động.
- **Panel (PanelControl):** Điều khiển panel là một "khung" để chứa các điều khiển khác. Nó thường được sử dụng để nhóm và quản lý các điều khiển liên quan tới nhau.

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển TextBox (TextBoxControl) trong ASP.NET. Ví dụ này sẽ tạo một trang web có một TextBox cho phép người dùng nhập tên của họ và sau đó hiển thị thông tin chào mừng.
- Default.aspx (Trang chính của ứng dụng):**

```
<form id="form1" runat="server">
    <div>
        <h1>Chào mừng bạn đến với TextBox Demo!</h1>
        <p>Nhập tên của bạn:</p>
        <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
        <br />
        <asp:Button ID="btnSubmit" runat="server" Text="Gửi" OnClick="btnSubmit_Click" />
        <br />
        <p><asp:Label ID="lblMessage" runat="server" Text=""></asp:Label></p>
    </div>
</form>
```



Control cơ bản trong ASP.NET

- Default.aspx.cs (Code-behind của trang chính):

```
protected void Page_Load(object sender, EventArgs e)
{
}

protected void btnSubmit_Click(object sender, EventArgs e)
{
    // Lấy dữ liệu từ TextBox và hiển thị thông báo chào mừng
    string name = txtName.Text;
    lblMessage.Text = "Chào mừng bạn, " + name + "!";
}
```

Trong ví dụ này, chúng ta có một trang web với một TextBox (txtName) để người dùng nhập tên và một nút (btnSubmit) để gửi dữ liệu. Khi người dùng nhấn nút "Gửi", sự kiện btnSubmit_Click được kích hoạt và mã C# xử lý dữ liệu từ TextBox và hiển thị thông báo chào mừng trên nhãn (lblMessage).

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Button (ButtonControl)** và **Label (LabelControl)** trong ASP.NET. Ví dụ này sẽ tạo một trang web có một nút và khi người dùng nhấn vào nút, một thông báo sẽ xuất hiện thông báo "Xin chào!"
- Default.aspx (Trang chính của ứng dụng):**

```
<form id="form1" runat="server">
    <div>
        <h1>Chào mừng bạn đến với Button Demo!</h1>
        <p>Nhấn nút để xem thông báo:</p>
        <asp:Button ID="btnShowMessage" runat="server" Text="Hiển thị thông báo" OnClick="btnShowMessage_Click" />
        <br />
        <p><asp:Label ID="lblMessage" runat="server" Text=""></asp:Label></p>
    </div>
</form>
```

Chào mừng bạn đến với Button Demo!

Nhấn nút để xem thông báo:

Hiển thị thông báo

[lblMessage]

Control cơ bản trong ASP.NET

- Default.aspx.cs (Code-behind của trang chính):

```
0 references
protected void Page_Load(object sender, EventArgs e)
{
}

0 references
protected void btnShowMessage_Click(object sender, EventArgs e)
{
    lblMessage.Text = "Xin chào!";
}
```

Trong ví dụ này, chúng ta có một trang web với một nút (btnShowMessage) và một nhãn (lblMessage). Khi người dùng nhấp vào nút "Hiển thị thông báo", sự kiện btnShowMessage_Click được kích hoạt và mã C# xử lý để hiển thị thông báo "Xin chào!" lên nhãn.

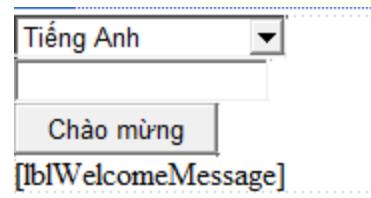
Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **DropDownList** (**DropDownListControl**) trong ASP.NET. Ví dụ này sẽ tạo một trang web có một TextBox cho phép người dùng nhập tên của họ và sau đó hiển thị thông tin chào mừng.
- Default.aspx (Trang chính của ứng dụng):**

```
<div>

    <asp:DropDownList ID="ddlLanguages" runat="server">
        <asp:ListItem Text="Tiếng Anh" Value="en" />
        <asp:ListItem Text="Tiếng Việt" Value="vi" />
        <asp:ListItem Text="Tiếng Tây Ban Nha" Value="es" />
    </asp:DropDownList>
    <br />
    <asp:TextBox ID="txtName" runat="server" placeholder="Nhập tên của bạn"></asp:TextBox>
    <br />
    <asp:Button ID="btnSubmit" runat="server" Text="Chào mừng" OnClick="btnSubmit_Click" />
    <br />
    <asp:Label ID="lblWelcomeMessage" runat="server"></asp:Label>

</div>
```



Control cơ bản trong ASP.NET

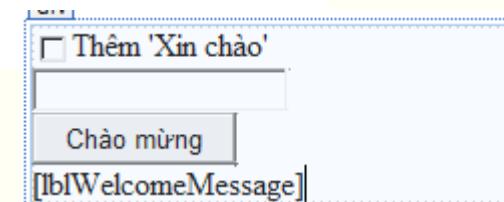
▪ Default.aspx.cs (Code-behind của trang chính):

```
0 references
protected void btnSubmit_Click(object sender, EventArgs e)
{
    string selectedLanguage = ddlLanguages.SelectedValue;
    string name = txtName.Text;
    string welcomeMessage = GetWelcomeMessage(selectedLanguage, name);
    lblWelcomeMessage.Text = welcomeMessage;
}
1 reference
private string GetWelcomeMessage(string language, string name)
{
    switch (language)
    {
        case "en":
            return $"Welcome, {name}!";
        case "vi":
            return $"Chào mừng, {name}!";
        case "es":
            return $"¡Bienvenido/a, {name}!";
        default:
            return $"Welcome, {name}!";
    }
}
```

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **CheckBox** (**CheckBoxControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:CheckBox ID="chkAddHello" runat="server" Text="Thêm 'Xin chào'" />
<br />
<asp:TextBox ID="txtName" runat="server" placeholder="Nhập tên của bạn"></asp:TextBox>
<br />
<asp:Button ID="btnSubmit" runat="server" Text="Chào mừng" OnClick="btnSubmit_Click" />
<br />
<asp:Label ID="lblWelcomeMessage" runat="server"></asp:Label>
```



Control cơ bản trong ASP.NET

- Default.aspx.cs (Code-behind của trang chính):

```
0 references
protected void btnSubmit_Click(object sender, EventArgs e)
{
    string name = txtName.Text;
    bool addHello = chkAddHello.Checked;
    string welcomeMessage = GetWelcomeMessage(name, addHello);
    lblWelcomeMessage.Text = welcomeMessage;
}
1 reference
private string GetWelcomeMessage(string name, bool addHello)
{
    if (addHello)
    {
        return $"Xin chào, {name}!";
    }
    else
    {
        return $"Chào mừng, {name}!";
    }
}
```

Trong ví dụ này, chúng ta sử dụng điều khiển CheckBox để cho phép người dùng quyết định xem liệu họ muốn thêm từ "Xin chào" vào thông báo chào mừng hay không. Khi họ nhấn nút "Chào mừng", chúng ta sử dụng một hàm để tạo thông báo chào mừng tương ứng với tùy chọn "Thêm 'Xin chào'" và tên mà họ đã nhập. Sau đó, chúng ta hiển thị thông báo chào mừng trên một nhãn (Label).

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **RadioButton** (**RadioButtonControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:RadioButton ID="radMale" runat="server" GroupName="gender" Text="Nam" Checked="True" />
<asp:RadioButton ID="radFemale" runat="server" GroupName="gender" Text="Nữ" />
<br />
<asp:TextBox ID="txtName" runat="server" placeholder="Nhập tên của bạn"></asp:TextBox>
<br />
<asp:Button ID="btnSubmit" runat="server" Text="Chào mừng" OnClick="btnSubmit_Click" />
<br />
<asp:Label ID="lblWelcomeMessage" runat="server"></asp:Label>
```



Control cơ bản trong ASP.NET

▪ Default.aspx.cs (Code-behind của trang chính):

```
0 references
protected void btnSubmit_Click(object sender, EventArgs e)
{
    string gender = radMale.Checked ? "Nam" : "Nữ";
    string name = txtName.Text;
    string welcomeMessage = GetWelcomeMessage(gender, name);
    lblWelcomeMessage.Text = welcomeMessage;
}

1 reference
private string GetWelcomeMessage(string gender, string name)
{
    return $"Chào mừng bạn, {name}! Bạn là {gender}.";
}
```

Người dùng có thể chọn giới tính bằng cách chọn một trong hai nút radMale hoặc radFemale, nhập tên của họ vào ô txtName, và nhấn nút btnSubmit để xem thông báo chào mừng dựa trên giới tính và tên đã chọn.

Trong ví dụ trên, chúng ta sử dụng điều khiển RadioButton để cho phép người dùng chọn giới tính. Khi họ nhấn nút "Chào mừng", chúng ta sử dụng một hàm để tạo thông báo chào mừng tương ứng với giới tính và tên mà họ đã nhập. Sau đó, chúng ta hiển thị thông báo chào mừng trên một nhãn (Label).

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **HyperLink** (**HyperLinkControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:HyperLink ID="hyperlinkGoogle" runat="server" Text="Truy cập Google" NavigateUrl="https://www.google.com" />
```



Control cơ bản trong ASP.NET

- Trong ví dụ trên, chúng ta sử dụng điều khiển HyperLink để tạo một liên kết trang web từ trang web hiện tại đến trang web khác (trong trường hợp này là <https://www.google.com>).
- Khi người dùng nhấp vào liên kết, trình duyệt sẽ mở trang web đích trong một cửa sổ mới hoặc cửa sổ trình duyệt hiện tại, tùy thuộc vào cài đặt.

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Image (ImageControl)** trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:Image ID="imgExample" runat="server" ImageUrl("~/Images/example.jpg" AlternateText="Hình ảnh ví dụ" />
```



Control cơ bản trong ASP.NET

- Trong ví dụ trên, chúng ta sử dụng điều khiển Image để hiển thị một hình ảnh trên trang web. Thuộc tính ImageUrl được sử dụng để chỉ định đường dẫn của hình ảnh trong thư mục Images.
- Thuộc tính AlternateText là một văn bản mô tả thay thế cho hình ảnh, sẽ hiển thị khi hình ảnh không thể hiển thị hoặc khi người dùng rê chuột qua hình ảnh.

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển Literal (**LiteralControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:Literal ID="litExample" runat="server" Text="Đây là một ví dụ về sử dụng điều khiển Literal trong ASP.NET." />
```

Đây là một ví dụ về sử dụng điều khiển Literal trong ASP.NET.

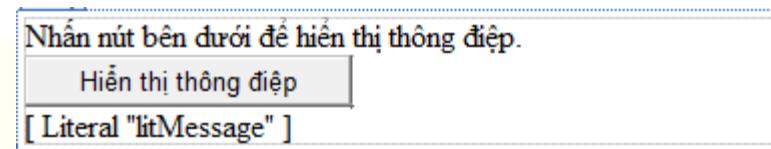
Control cơ bản trong ASP.NET

- Trong ví dụ trên, chúng ta sử dụng điều khiển Literal để hiển thị một chuỗi văn bản trực tiếp trên trang web.
- Thuộc tính Text của điều khiển Literal được sử dụng để chỉ định nội dung hiển thị. Bạn cũng có thể sử dụng Literal để hiển thị mã HTML, và nội dung trong Text sẽ được hiển thị mà không bị mã hóa HTML.

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển Panel (PanelControl) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:Panel ID=" pnlExample" runat="server">
    <asp:Label ID="lblMessage" runat="server" Text="Nhấn nút bên dưới để hiển thị thông điệp." />
    <br />
    <asp:Button ID="btnShowMessage" runat="server" Text="Hiển thị thông điệp" OnClick="btnShowMessage_Click" />
    <br />
    <asp:Literal ID="litMessage" runat="server" Visible="false" />
</asp:Panel>
```



Control cơ bản trong ASP.NET

- Default.aspx.cs (Code-behind của trang chính):

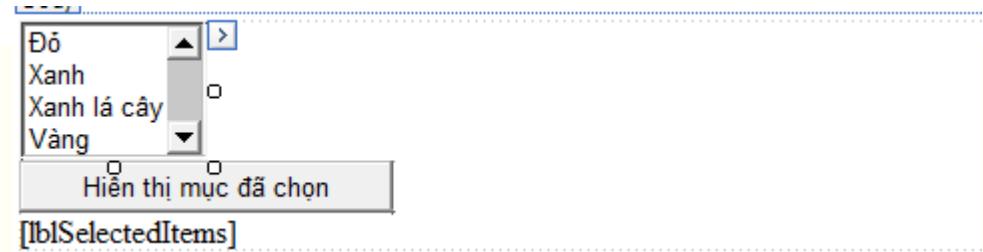
```
0 references
protected void btnShowMessage_Click(object sender, EventArgs e)
{
    litMessage.Text = "Đây là thông điệp được hiển thị bởi Panel.";
    litMessage.Visible = true;
}
```

Trong ví dụ trên, chúng ta sử dụng điều khiển Panel để nhóm các điều khiển lại với nhau. Trong Panel, chúng ta thêm một Label và một Button để hiển thị thông điệp khi nút "Hiển thị thông điệp" được nhấn. Sự kiện click của nút được xử lý để hiển thị thông điệp bằng cách đặt nội dung vào Literal và đặt Visible của Literal thành true.

Control cơ bản trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển ListBox.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:ListBox ID="lstColors" runat="server" SelectionMode="Multiple">
    <asp:ListItem Text="Đỏ" Value="red" />
    <asp:ListItem Text="Xanh" Value="blue" />
    <asp:ListItem Text="Xanh lá cây" Value="green" />
    <asp:ListItem Text="Vàng" Value="yellow" />
    <asp:ListItem Text="Cam" Value="orange" />
</asp:ListBox>
<br />
<asp:Button ID="btnShowSelection" runat="server" Text="Hiển thị mục đã chọn" OnClick="btnShowSelection_Click" />
<br />
<asp:Label ID="lblSelectedItems" runat="server"></asp:Label>
```



Control cơ bản trong ASP.NET

▪ Default.aspx.cs (Code-behind của trang chính):

```
0 references
protected void btnShowSelection_Click(object sender, EventArgs e)
{
    string selectedItems = string.Empty;
    foreach (ListItem item in lstColors.Items)
    {
        if (item.Selected)
        {
            selectedItems += item.Text + ", ";
        }
    }
    if (!string.IsNullOrEmpty(selectedItems))
    {
        selectedItems = selectedItems.TrimEnd(',', ' ');
        lblSelectedItems.Text = "Các mục đã chọn: " + selectedItems;
    }
    else
    {
        lblSelectedItems.Text = "Không có mục nào được chọn.";
    }
}
```

Trong ví dụ trên, chúng ta sử dụng điều khiển ListBox để hiển thị danh sách các mục có thể chọn. Thuộc tính SelectionMode của ListBox được đặt thành Multiple để cho phép chọn nhiều mục cùng một lúc. Sự kiện click của nút "Hiển thị mục đã chọn" được xử lý để lặp qua danh sách các mục và hiển thị các mục đã chọn trên một nhãn (Label).

Control nâng cao trong ASP.NET

ASP.NET cung cấp nhiều loại điều khiển nâng cao hơn để giúp bạn xây dựng giao diện và tương tác phong phú hơn trong ứng dụng web.

- **GridView (GridViewControl):** Điều khiển này cho phép bạn hiển thị dữ liệu từ một nguồn dữ liệu (ví dụ: cơ sở dữ liệu) trong một bảng lưới (grid). GridView cho phép bạn hiển thị, chỉnh sửa và xóa dữ liệu dễ dàng.
- **DataList (DataListControl):** Điều khiển này cho phép bạn hiển thị dữ liệu trong một danh sách linh hoạt. Bạn có thể tùy chỉnh cách dữ liệu được hiển thị bằng cách sử dụng các mẫu mô tả.
- **Repeater (RepeaterControl):** Tương tự như DataList, điều khiển Repeater cho phép bạn hiển thị dữ liệu trong một danh sách. Tuy nhiên, Repeater cung cấp sự linh hoạt cao hơn trong việc tạo giao diện tùy chỉnh.

Control nâng cao trong ASP.NET

- **ListView (ListViewControl)**: Điều khiển này là một phiên bản nâng cao của GridView, cho phép bạn hiển thị dữ liệu trong một danh sách linh hoạt với các tùy chọn kiểm soát giao diện chi tiết.
- **FormView (FormViewControl)**: Điều khiển này giúp bạn hiển thị và chỉnh sửa dữ liệu theo một mẫu chi tiết. Nó thường được sử dụng cho việc xem và chỉnh sửa chi tiết một bản ghi trong cơ sở dữ liệu.
- **FileUpload (FileUploadControl)**: Điều khiển cho phép người dùng tải lên tệp từ máy tính của họ lên máy chủ. Điều khiển này rất hữu ích khi bạn cần cho phép người dùng tải lên hình ảnh, tệp tin, v.v.
- **Calendar (CalendarControl)**: Điều khiển lịch cho phép người dùng chọn ngày từ một lịch. Điều khiển này thường được sử dụng trong các ứng dụng yêu cầu chọn ngày.

Control nâng cao trong ASP.NET

- **Wizard (WizardControl):** Điều khiển này giúp bạn tạo giao diện dẫn dắt người dùng qua một loạt các bước. Điều khiển này thường được sử dụng trong quy trình đăng ký, đặt hàng, v.v.
- **Login (LoginControl):** Điều khiển đăng nhập cung cấp một giao diện tiêu chuẩn cho việc đăng nhập người dùng vào hệ thống. Nó bao gồm nhập tên đăng nhập, mật khẩu và các tùy chọn khác.
- **Navigation Controls:** Các điều khiển điều hướng như Menu, TreeView và SiteMapPath cho phép bạn tạo menu, cây thư mục và đường dẫn địa chỉ trên giao diện người dùng.

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **GridView** (**GridViewControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

```
<asp:GridView ID="GridView1" runat="server" DataKeyNames="username" OnRowDeleting="GridView1_RowDeleting" OnSelectedIndexChanged="GridView1_SelectedIndexChanged" >
  <Columns>
    <asp:CommandField ShowSelectButton="True" />
    <asp:CommandField ShowDeleteButton="True" />
  </Columns>
</asp:GridView>
```

		Column0	Column1	Column2
Select	Delete	abc	abc	abc
Select	Delete	abc	abc	abc
Select	Delete	abc	abc	abc
Select	Delete	abc	abc	abc
Select	Delete	abc	abc	abc

Control nâng cao trong ASP.NET

▪ Default.aspx.cs (Code-behind của trang chính):

```
public void hienthi()
{
    string sql = "select * from tbluser";
    SqlDataAdapter da = new SqlDataAdapter
    DataSet ds = new DataSet();
    da.Fill(ds); //ds dang giu csdl
    GridView1.DataSource = ds;
    GridView1.DataBind();
}

protected void GridView1_SelectedIndexChangedCha
{
    GridViewRow row = GridView1.SelectedF
    this.txtuser.Text = row.Cells[2].Text
    this.txtfull.Text = row.Cells[4].Text
    this.txtemail.Text = row.Cells[5].Tex
    this.DropDownList1.SelectedValue = ro
}

public void them(string sql)
{
    con.Open();
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.ExecuteNonQuery();
    con.Close();
}

public void Sua(string sql)
{
    con.Open();
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.ExecuteNonQuery();
    con.Close();
}

public void xoa(string sql)
{
    con.Open();
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.ExecuteNonQuery();
    con.Close();
}

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    string a = GridView1.DataKeys[e.RowIndex].Values["username"].ToString();
    SqlCommand cmd = new SqlCommand();
    Connection = kn.con;
    CommandText = "delete from tbluser where username='" + a + "'";
    CommandType = CommandType.Text;

    on.Open(); // mo ket noi
    ExecuteNonQuery(); // thuc thi
    on.Close(); // dong ket noi

    thi();
}

cmd.Connection = kn.con;
cmd.CommandText = "update tbluser set fullname='"+b+"',email='"+c+"', gender='"+d+"' where username='"+ a + "'";
cmd.CommandType = CommandType.Text;

kn.con.Open(); // mo ket noi
cmd.ExecuteNonQuery(); // thuc thi
kn.con.Close(); // dong ket noi
```

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **DataList** (**DataListControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Repeater** (**RepeaterControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **ListView** (**ListViewControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **FormView** (**FormViewControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **FileUpload** (**FileUploadControl**) trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

▪ Default.aspx.cs (Code-behind của trang chính):

```
bool CheckFileType(string fileName)
{
    string ext = Path.GetExtension(fileName);
    switch (ext.ToLower())
    {
        case ".gif":
            return true;
        case ".png":
            return true;
        case ".jpg":
            return true;
        case ".jpeg":
            return true;
        default:
            return false;
    }
}

protected void btnUpload_Click(object sender, EventArgs e)
{
    if (Page.IsValid && FileUpload1.HasFile && CheckFileType(FileUpload1.FileName))
    {
        string fileName = "images/" + DateTime.Now.ToString("ddMMyyyy_hhmmss_tt_") + FileUpload1.FileName;
        string filePath = MapPath(fileName);
        FileUpload1.SaveAs(filePath);
        //Image1.ImageUrl = fileName;
        SqlCommand cmd = new SqlCommand("insert into tblcauhoi(noidung,goiy,hinhanh) values(N'" + txtnoidung.Text + "',N'" + txtgoiy.Text + "','" + fileName + "')", kn.con);
        kn.con.Open();
        cmd.ExecuteNonQuery();
        kn.con.Close();
        Response.Write("<script>alert('Thanh cong')</script>");
    }
}
```

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Wizard (WizardControl)** trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Login (LoginControl)** trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Dưới đây là một ví dụ đơn giản về cách sử dụng điều khiển **Navigation Controls** trong ASP.NET.
- **Default.aspx (Trang chính của ứng dụng):**

Control nâng cao trong ASP.NET

- **Default.aspx.cs (Code-behind của trang chính):**

Trong ví dụ này,

Control nâng cao trong ASP.NET

- Các điều khiển kiểm tra hợp lệ (Validator Controls) trong ASP.NET cho phép bạn thực hiện kiểm tra và xác thực dữ liệu nhập vào từ người dùng trước khi nó được gửi đến máy chủ hoặc lưu vào cơ sở dữ liệu.

Control nâng cao trong ASP.NET

- **CompareValidator:** Điều khiển này cho phép bạn so sánh giá trị của một điều khiển với giá trị của một điều khiển khác hoặc giá trị cố định. Ví dụ, bạn có thể sử dụng nó để so sánh hai mật khẩu nhập vào để đảm bảo rằng chúng khớp nhau.

```
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
<asp:TextBox ID="txtConfirmPassword" runat="server" TextMode="Password"></asp:TextBox>
<asp:CompareValidator ID="cmpPassword" runat="server"
    ControlToValidate="txtConfirmPassword"
    ControlToCompare="txtPassword"
    Operator="Equal" ErrorMessage="Mật khẩu không khớp"></asp:CompareValidator>
```

Control nâng cao trong ASP.NET

- **CustomValidator:** Điều khiển này cho phép người viết mã kiểm tra hợp lệ tùy chỉnh bằng mã C# hoặc mã JavaScript. Chúng ta có thể thực hiện kiểm tra phức tạp hơn mà không nằm trong phạm vi các kiểm tra tiêu chuẩn.

```
<asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
<asp:CustomValidator ID="custAge" runat="server"
    ControlToValidate="txtAge"
    OnServerValidate="ValidateAge"
    ErrorMessage="Vui lòng nhập tuổi hợp lệ"></asp:CustomValidator>
```

```
0 references
protected void ValidateAge(object source, ServerValidateEventArgs args)
{
    int age;
    if (int.TryParse(args.Value, out age))
    {
        args.IsValid = age >= 18; // Kiểm tra tuổi có lớn hơn hoặc bằng 18 không
    }
    else
    {
        args.IsValid = false;
    }
}
```

Control nâng cao trong ASP.NET

- **RangeValidator:** Điều khiển này kiểm tra xem giá trị nhập vào có nằm trong khoảng giá trị xác định hay không.

```
<asp:TextBox ID="txtNumber" runat="server"></asp:TextBox>
<asp:RangeValidator ID="rngNumber" runat="server"
    ControlToValidate="txtNumber"
    Type="Integer"
    MinimumValue="1" MaximumValue="10"
    ErrorMessage="Vui lòng nhập số từ 1 đến 10"></asp:RangeValidator>
```

Control nâng cao trong ASP.NET

- **RegularExpressionValidator:** Điều khiển này sử dụng biểu thức chính quy để kiểm tra xem giá trị nhập vào có phù hợp với một mẫu cụ thể hay không.

```
<asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
<asp:RegularExpressionValidator ID="regEmail" runat="server"
    ControlToValidate="txtEmail"
    ValidationExpression="^([a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$"
    ErrorMessage="Vui lòng nhập địa chỉ email hợp lệ"></asp:RegularExpressionValidator>
```

Control nâng cao trong ASP.NET

- **RequiredFieldValidator:** Điều khiển này đảm bảo rằng một trường đã được nhập dữ liệu và không để trống.

```
<asp:TextBox ID="txtName" runat="server"></asp:TextBox>
<asp:RequiredFieldValidator ID="reqName" runat="server"
    ControlToValidate="txtName"
    ErrorMessage="Vui lòng nhập tên"></asp:RequiredFieldValidator>
```

Control nâng cao trong ASP.NET

- **ValidationSummary:** Điều khiển này tổng hợp tất cả các thông báo lỗi từ các điều khiển kiểm tra hợp lệ và hiển thị chúng dưới dạng danh sách.

```
<asp:ValidationSummary ID="valSummary" runat="server"  
HeaderText="Có lỗi sau đây:" />
```

Có lỗi sau đây:

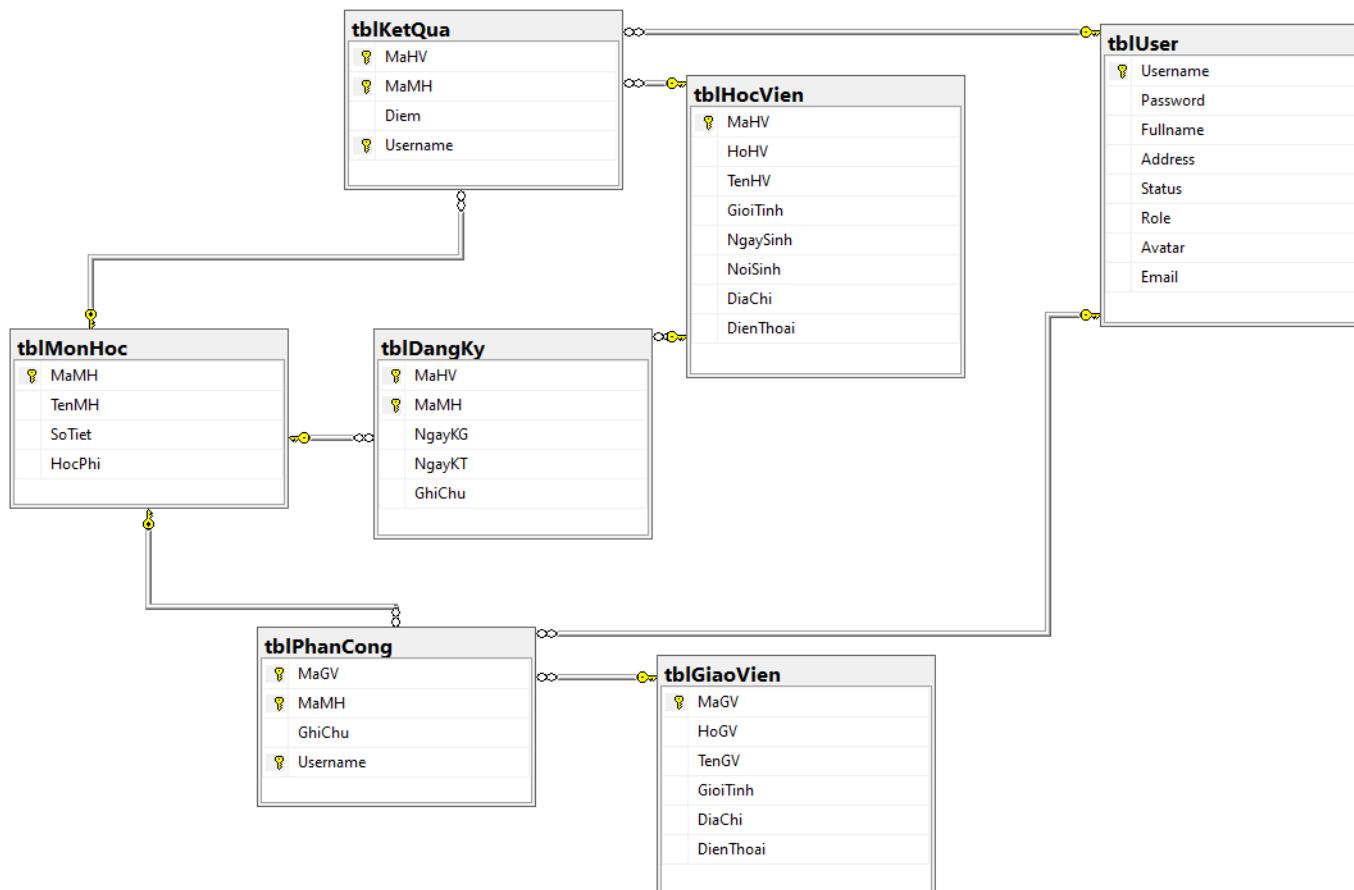
- Vui lòng nhập tên
- Vui lòng nhập địa chỉ email hợp lệ
- Vui lòng nhập số từ 1 đến 10

Control nâng cao trong ASP.NET

- **Lưu ý:** Khi gặp lỗi trong quá trình sử dụng các control validator, nên bổ sung đoạn code vào web.config để khắc phục lỗi.

```
<appSettings>  
    <add key="ValidationSettings:UnobtrusiveValidationMode" value="None" />  
</appSettings>
```

ASP.NET VÀ CSDL



ASP.NET VÀ CSDL

ANTONIO86DOAN\S...- dbo.tblHocVien ➔ X ANTONIO86DOAN\S...dbo.t

Column Name	Data Type	Allow Nulls
MaHV	varchar(50)	<input type="checkbox"/>
HoHV	nvarchar(50)	<input checked="" type="checkbox"/>
TenHV	nvarchar(50)	<input checked="" type="checkbox"/>
GioiTinh	nvarchar(50)	<input checked="" type="checkbox"/>
NgaySinh	date	<input checked="" type="checkbox"/>
NoiSinh	nvarchar(50)	<input checked="" type="checkbox"/>
DiaChi	nvarchar(50)	<input checked="" type="checkbox"/>
DienThoai	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...HV - dbo.tblUser ➔ X ANTONIO86DOAN\S...db

Column Name	Data Type	Allow Nulls
Username	varchar(50)	<input type="checkbox"/>
Password	varchar(50)	<input checked="" type="checkbox"/>
Fullname	nvarchar(50)	<input checked="" type="checkbox"/>
Address	nvarchar(50)	<input checked="" type="checkbox"/>
Status	int	<input checked="" type="checkbox"/>
Role	int	<input checked="" type="checkbox"/>
Avatar	varchar(50)	<input checked="" type="checkbox"/>
Email	varchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...- dbo.tblMonHoc ➔ X ANTONIO86DOAN\S...- dbo

Column Name	Data Type	Allow Nulls
MaMH	varchar(50)	<input type="checkbox"/>
TenMH	nvarchar(50)	<input checked="" type="checkbox"/>
SoTiet	int	<input checked="" type="checkbox"/>
HocPhi	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...- dbo.tblDangKy ➔ X

Column Name	Data Type	Allow Nulls
MaHV	varchar(50)	<input type="checkbox"/>
MaMH	varchar(50)	<input type="checkbox"/>
NgayKG	date	<input checked="" type="checkbox"/>
NgayKT	date	<input checked="" type="checkbox"/>
GhiChu	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...dbo.tblGiaoVien ➔ X ANTONIO86DOAN\S...dbo.tb

Column Name	Data Type	Allow Nulls
MaGV	varchar(50)	<input type="checkbox"/>
HoGV	nvarchar(50)	<input checked="" type="checkbox"/>
TenGV	nvarchar(50)	<input checked="" type="checkbox"/>
GioiTinh	nvarchar(50)	<input checked="" type="checkbox"/>
DiaChi	nvarchar(50)	<input checked="" type="checkbox"/>
DienThoai	char(10)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...dbo.tblPhanCong ➔ X ANTONIO86DOAN\S...

Column Name	Data Type	Allow Nulls
MaGV	varchar(50)	<input type="checkbox"/>
MaMH	varchar(50)	<input type="checkbox"/>
GhiChu	nvarchar(50)	<input checked="" type="checkbox"/>
Username	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

ANTONIO86DOAN\S...- dbo.tbKetQua ➔ X ANTONIO86DOAN\S...-

Column Name	Data Type	Allow Nulls
MaHV	varchar(50)	<input type="checkbox"/>
MaMH	varchar(50)	<input type="checkbox"/>
Diem	int	<input checked="" type="checkbox"/>
Username	varchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Project demo

- Kết nối cơ sở dữ liệu

The screenshot shows a Microsoft Visual Studio code editor window. The code is written in C# and defines a class named `Ketnoi` with a static method `mahoam`. The code uses several namespaces from the `System` and `System.Data.SqlClient` families. It includes a connection string to a local SQL Server instance named `sqlexpress`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using System.Web.Security;

namespace aspnet
{
    public class Ketnoi
    {
        public SqlConnection con = new SqlConnection("Data Source=.\sqlexpress;Initial Catalog=QLHV;Integrated Security=True");

        [Obsolete]
        public string mahoam(string str)
        {
            return FormsAuthentication.HashPasswordForStoringInConfigFile(str.Trim(), "SHA1");
        }
    }
}
```