

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VĨNH LONG
KHOA SƯ PHẠM KỸ THUẬT & XÃ HỘI NHÂN VĂN



**BÀI THI KẾT THÚC MÔ ĐUN
CHUẨN BỊ DẠY HỌC**

Sinh viên: Nguyễn Tuấn Trọng MSSV: 21004235
Lớp: 1CTT21A3 Khóa: 46
GVHD: Trần Hoài Hạnh

Học kỳ 2, Năm học 2023 - 2024

NỘI DUNG		Trang
1	Lập kế hoạch dạy học	1
2	Xây dựng hồ sơ cho 1 bài giảng lý thuyết	8
2.1	Xác định vị trí bài giảng	8
2.2	Bài giảng	8
2.3	Kịch bản sư phạm	18
2.4	Giáo án lý thuyết	31
2.5	Bộ câu hỏi trắc nghiệm (8-10 câu)	38
3	Xây dựng hồ sơ cho 1 bài giảng thực hành	41
3.1	Xác định vị trí bài giảng	41
3.2	Bài giảng	41
3.3	Kịch bản sư phạm	51
3.4	Giáo án thực hành	53
3.5	Phiếu hướng dẫn thực hiện	59
3.6	Phiếu đánh giá quy trình	64
3.7	Phiếu đánh giá sản phẩm	66
4	Xây dựng hồ sơ cho 1 bài giảng tích hợp	68
4.1	Xác định vị trí bài giảng	68
4.2	Bài giảng	68
4.3	Kịch bản sư phạm	84
4.4	Giáo án tích hợp	86
4.5	Phiếu hướng dẫn thực hiện	93
4.6	Câu hỏi kiểm tra trắc nghiệm lựa chọn đa phương án	95
4.7	Phiếu đánh giá quy trình	98
4.8	Phiếu đánh giá sản phẩm	99

THÔNG TIN CÁ NHÂN

- Họ tên: Nguyễn Tuấn Trọng
- Ngày tháng, năm sinh: 21/09/2003
- Lớp: 1CTT21A3
- Ngành học: Công nghệ thông tin
- Khóa: 46
- Email: 21004235@st.vlute.edu.vn Điện thoại: 0352440418
- Quê quán : Xã Mỹ Thạnh Trung, Huyện Tam Bình, Tỉnh Vĩnh Long



1. Kế hoạch dạy học

BẢNG GHI TÓM TẮT NỘI DUNG

HỌ VÀ TÊN GIÁO VIÊN: Nguyễn Tuấn Trọng
MÔN HỌC/MÔ-ĐUN: LẬP TRÌNH WINDOWS 2

NGÀY LÊN LỚP	SỐ TIẾT			TÓM TẮT NỘI DUNG BÀI DẠY, KIỂM TRA	SỐ HỌC SINH/SINH VIÊN VẮNG MẶT	CHỦ KÝ GIÁO VIÊN
	Lý thuyết	Thực hành	Kiểm tra			
02/09/2024	2			Chương 1 : Tổng quan về ADO.Net 1.1 .Lịch sử phát triển ADO.net 1.2.Phần mềm cần thiết 1.3.Kiến trúc của ADO.net 1.4 .Các đối tượng ADO.Net trong.NET Framework		
09/09/2024	2			Chương 1 : Tổng quan về ADO.net (tiếp theo) 1.4 Các đối tượng ADO.net trong NET Framework (tiếp theo) 1.5. Biểu diễn dữ liệu trong bộ nhớ 1.6.Quản lý Recordset		
16/09/2024	2			Chương 2 : Tương tác với cơ sở dữ liệu 2.1 Đối tượng Connection		
23/09/2024	2			Chương 2 : Tương tác với cơ sở dữ liệu (tiếp theo) 2.2 Đối tượng Command		
30/09/2024	2			Chương 2 : Tương tác với cơ sở dữ liệu (tiếp theo)		

				2.3 Đối tượng DataReader		
07/10/2024	2			Chương 3 : Xử lý dữ liệu 3.1 Thành phần DataSet 3.2 Tương tác với DataSet		
14/10/2024	2			Chương 3 : Xử lý dữ liệu (tiếp theo) 3.2 Tương tác với DataSet (tiếp theo)		
21/10/2024	2			Chương 4 : Bộ điều hợp dữ liệu DataAdapter 4.1 Vai trò của DataAdapter 4.2 Nạp dữ liệu từ cơ sở dữ liệu vào DataSet 4.3 Cập nhật cơ sở dữ liệu		
28/10/2024	1		1	Kiểm tra trắc nghiệm lần 1 (C1,C2,C3) Chương 4 : Bộ điều hợp dữ liệu DataAdapter (tiếp theo) 4.3 Cập nhật cơ sở dữ liệu(tiếp theo)		
4/11/2024	2			Chương 4 : Bộ điều hợp dữ liệu DataAdapter (tiếp theo) 4.3 Cập nhật cơ sở dữ liệu (tiếp theo) Chương 5 : Sử dụng các ràng buộc dữ liệu 5.1 Tạo ràng buộc		
11/11/2024	2			Chương 5 : Sử dụng các		

				ràng buộc dữ liệu (tiếp theo) 5.2 Lọc dữ liệu trong hộp danh sách 5.3 Buộc dữ liệu vào hộp văn bản		
18/11/2024	2			Chương 5 : Sử dụng các ràng buộc dữ liệu (tiếp theo) 5.4 Sửa đổi và cập nhật dữ liệu từ hộp văn bản 5.5 Thêm mới và xóa bản ghi dùng các điều khiển ràng buộc dữ		
25/11/2024	2			Chương 5 : Sử dụng các ràng buộc dữ liệu (tiếp theo) 5.6 Quản lý lỗi khi tương tác dữ liệu Chương 6 : Tạo báo cáo với Crystal Report 6.1 Tạo báo cáo dùng Report Expert		
02/12/2024	1		1	Kiểm tra trắc nghiệm lần 2 (C4,C5) Chương 6 : Tạo báo cáo với Crystal Report (tiếp theo) 6.2 Hiển thị các báo cáo đã tạo		
06/12/2024	2			Chương 6 : Tạo báo cáo với		

				Crystal Report (tiếp theo) 6.3 Thêm cột tính toán 6.4 Chọn bản ghi hiển thị		
09/12/2024	2			Chương 7 : ADO.net và XML 7.1 Tạo dữ liệu XML 7.2 Nạp dữ liệu XML		
13/12/2024	2			Chương 7 : ADO.net và XML (tiếp theo) 7.3 Giải đố XML 7.4 Kiểm tra sự hợp lệ của dữ liệu		
16/12/2024	1		1	Kiểm tra lần trắc nghiệm lần 3 (C6,C7) Ôn tập tổng kết		

Trường :	HỆ THỐNG CÁC BÀI HỌC	Số giờ học trong học kỳ:
Khoa/Bộ môn : Công nghệ thông tin	THỰC HÀNH	50h
Năm học : 2024-2025	Môn học : Lập trình windows 2	Số giờ thực tập :48h
Học kỳ : 2	Cán bộ giảng dạy : Nguyễn Tuấn Trọng	Số tuần lẽ :10
	Lớp :	

TT	TÊN ĐỀ MỤC TRONG CHƯƠNG TRÌNH	SỐ GIỜ QUY ĐỊNH CỦA ĐỀ MỤC	THỨ TỰ BÀI HỌC ĐỀ THỰC HIỆN	NỘI DUNG VÀ YÊU CẦU CỦA BÀI HỌC	THỜI GIAN THỰC HIỆN	RÚT KINH NGHIỆM VIỆC THỰC HIỆN
1	Bài 1 : Làm quen với ADO .NET và môi trường phát triển	1	1	Hướng dẫn cài đặt các phần mềm cần thiết như Visual Studio, SQL Server. Thực hành cài đặt và thiết lập môi trường phát triển, kiểm tra cài đặt bằng cách tạo một dự án đơn giản kết nối đến cơ sở dữ liệu	Tuần 38 16/09/2024	
	Bài 2 : Tương tác với CSDL (Đối tượng Connection)			-Quy trình tạo và quản lý kết nối đến cơ sở dữ liệu trong các dự án -Thực hành tạo đối tượng Connection, mở và đóng kết nối.		
2	Bài 2 : Tương tác với CSDL (Đối tượng Command)	2	2	- Cách sử dụng đối tượng Command để thực hiện các lệnh SQL (Select, Insert, Update, Delete) - Thực hành viết các đoạn mã để thực hiện các	Tuần 39 23/09/2024	

				lệnh SQL trên cơ sở dữ liệu.		
	Bài 2 : Tương tác với CSDL (Đối tượng DataReader)	3		- Cách sử dụng DataReader để đọc dữ liệu một cách hiệu quả - Thực hành viết các đoạn mã để đọc và hiển thị dữ liệu từ cơ sở dữ liệu bằng DataReader.	Tuần 40 30/09/2024	
3	Bài 3: Sử dụng DataSet và DataTable	5	3	-Trình tự lưu trữ và hiển thị dữ liệu từ cơ sở dữ liệu bằng DataSet và DataTable - Thực hành tạo và sử dụng các đối tượng này.	Tuần 41 07/10/2024	
4	Bài 4 : Xử lý dữ liệu với DataSet	5	4	- Cách thêm, sửa, xóa và cập nhật dữ liệu trong DataSet - Thực hiện viết các đoạn mã để thực hiện các thao tác này và hiển thị kết quả.	Tuần 42 14/10/2024	
5	Bài 5 : Bộ điều hợp dữ liệu DataAdepter (nạp dữ liệu vào DataSet)	5	5	-Cách sử dụng DataAdepter để kết nối DataSet với cơ sở dữ liệu - Thực hành sử dụng DataAdapter để nạp dữ liệu từ cơ sở dữ liệu vào DataSet.	Tuần 43 21/10/2024	
6	Bài 5 : Bộ điều hợp dữ liệu DataAdepter (cập nhật vào DataSet)	5	6	-Quy trình cập nhật dữ liệu từ DataSet vào cơ sở dữ liệu - Cách xử lý các xung đột dữ liệu -Thực hành viết các đoạn mã để cập nhật dữ liệu và xử lý lỗi.	Tuần 44 28/10/2024	

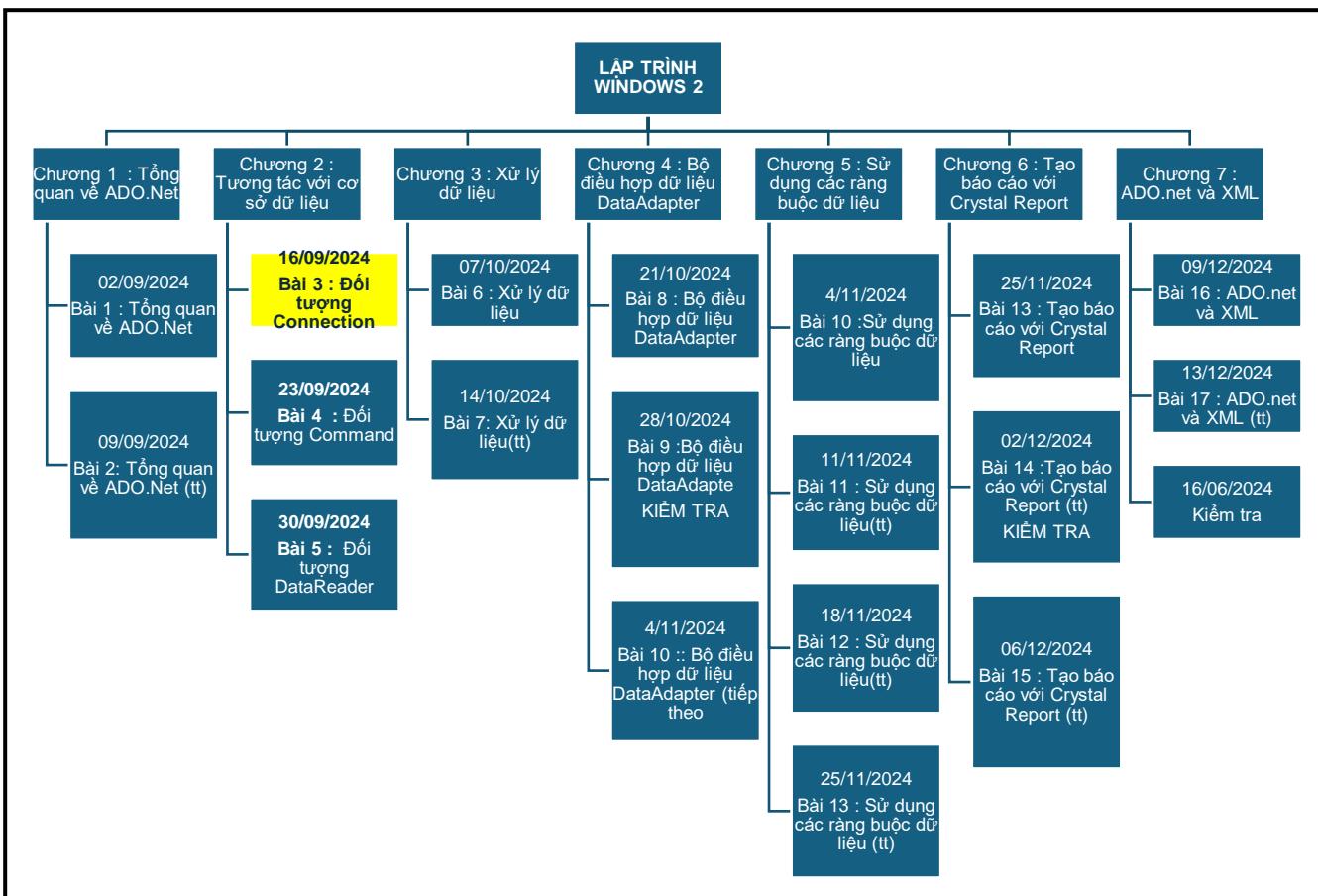
7	Bài 6 : Tạo báo cáo với Crystal Report	5	7	-Cách tạo buộc dữ liệu vào hộp văn bản, cập nhật dữ liệu từ hộp văn bản vào cơ sở dữ liệu - Cách tạo báo cáo dùng Crystal Report, tạo báo cáo dùng Report Expert - Thực hành các bài tập liên quan đến hộp văn bản ràng buộc dữ liệu. - Thực hành tạo báo cáo và hiển thị báo cáo đã tạo	Tuần 45 04/11/2024	
8	Bài 7 : Sử dụng ADO.Net với XML (Kiểm tra hợp lệ dữ liệu XML)	5	8	-Cách tạo tài liệu XML, nạp dữ liệu XML. - Thủ thuật kiểm tra sự hợp lệ của dữ liệu XML, sử dụng giản đồ XML để xác thực dữ liệu - Thực hành tạo và xử lý dữ liệu XML với ADO.Net. - Thực hành kiểm tra và xác thực dữ liệu XML.	Tuần 46 11/11/2024	
9	Bài 7 : Sử dụng ADO.Net với XML (Cập nhật dữ liệu và xử lý lỗi)	5	9	-Cách quản lý và xử lý lỗi khi tương tác với dữ liệu - Cách thêm mới và xóa bản ghi dùng các điều khiển ràng buộc dữ liệu - Thực hành các bài tập liên quan.	Tuần 47 18/11/2024	
10	Kiểm tra	2	10	-Kiểm tra thực hành	Tuần 48 25/11/2024	
	Ôn tập tổng kết	3		-Ôn tập các nội dung chuẩn bị thi kết thúc học phần		

--	--	--	--	--	--	--	--

2. Hồ sơ bài giảng lý thuyết

Tên bài : Đổi tượng Connection

2.1 Vị trí bài giảng



2.2 Bài giảng

Tên nghề : Quản trị mạng máy tính

Trình độ : Cao đẳng

Tên môn : Lập trình Windows 2

Tên bài : ĐỔI TƯỢNG CONNECTION

Thời gian : 90 phút

BÀI 3 : ĐỔI TƯỢNG CONNECTION

A. MỤC TIÊU

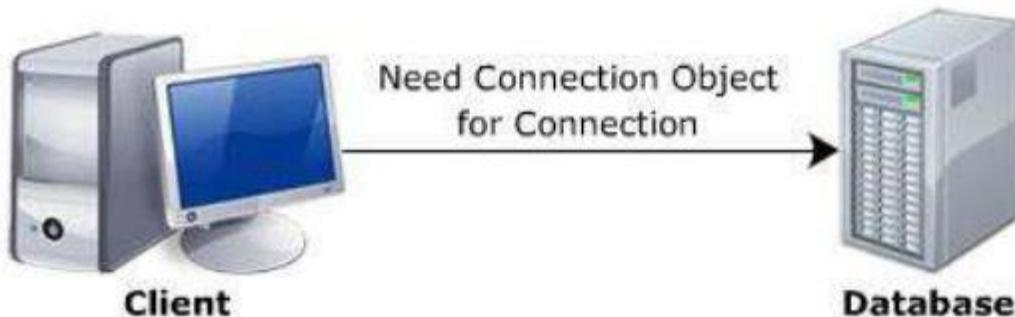
- Trình bày được vai trò và chức năng của đổi tượng Connection trong ADO.NET.

- Phân biệt các loại đối tượng Connection như SqlConnection và OleDbConnection.
- Mô tả cấu trúc và cách tạo một chuỗi kết nối (connection string) phù hợp.
- Viết được chương trình mở và đóng kết nối cơ sở dữ liệu bằng SqlConnection và OleDbConnection.
- Xử lý các lỗi kết nối cơ sở dữ liệu một cách hiệu quả và cung cấp được thông tin lỗi chi tiết
- Hình thành được tính cẩn thận và tỉ mỉ khi làm việc với cơ sở dữ liệu và quản lý tài nguyên

B. NỘI DUNG

1. Giới thiệu về đối tượng Connection

1.1 Vai trò của đối tượng Connection



Đối tượng Connection cho phép tạo kết nối giữa ứng dụng với CSDL (nguồn dữ liệu). Nó lưu trữ các thông tin được yêu cầu cho việc thiết lập kết nối. Bằng cách thiết lập kết nối, ứng dụng có thể truy xuất và thao tác trực tiếp với dữ liệu trong CSDL. Chúng ta sử dụng đối tượng Connection nếu muốn ứng dụng truy xuất CSDL nhiều lần. Kết nối giữa ứng dụng và nguồn dữ liệu chỉ được được thiết lập cho một lần cụ thể cho việc nhận hoặc cập nhật dữ liệu.

Đây là đối tượng đầu tiên mà các ứng dụng cần sử dụng để tương tác với cơ sở dữ liệu. Một số vai trò cụ thể của đối tượng Connection bao gồm :

- **Thiết lập kết nối:** Đối tượng Connection giúp thiết lập một liên kết giữa ứng dụng và cơ sở dữ liệu. Điều này bao gồm việc xác định nguồn dữ liệu (như tên máy chủ hoặc địa chỉ IP), tên cơ sở dữ liệu, thông tin xác thực (như tên người dùng và mật khẩu), và các thuộc tính khác cần thiết để mở kết nối.
- **Quản lý trạng thái kết nối:** chịu trách nhiệm quản lý trạng thái của kết nối, bao gồm việc mở kết nối (`Open()`), duy trì kết nối trong suốt quá trình tương tác với cơ sở dữ liệu, và đóng kết nối (`Close()`) khi không còn cần thiết. Điều này đảm bảo rằng kết nối không bị giữ mở quá lâu, gây ra lãng phí tài nguyên.
- **Bảo mật kết nối:** Đối tượng Connection cung cấp các tùy chọn để bảo mật kết nối đến cơ sở dữ liệu, chẳng hạn như sử dụng giao thức SSL/TLS, xác thực bằng

các phương thức bảo mật khác nhau, và quản lý chuỗi kết nối để đảm bảo rằng thông tin nhạy cảm không bị lộ ra ngoài.

- **Quản lý lỗi kết nối:** Connection cũng chịu trách nhiệm xử lý các lỗi kết nối. Khi có sự cố xảy ra, chẳng hạn như cơ sở dữ liệu không khả dụng hoặc thông tin xác thực không hợp lệ, đối tượng Connection sẽ ném ra các ngoại lệ (SqlException, OleDbException) để ứng dụng có thể bắt và xử lý các tình huống này một cách thích hợp.
- **Cấu hình và tối ưu hóa:** Đối tượng Connection cho phép cấu hình các thuộc tính như thời gian chờ kết nối (connection timeout), thời gian chờ lệnh (command timeout), và các tùy chọn khác để tối ưu hóa hiệu suất và đáp ứng yêu cầu của ứng dụng.
- **Quản lý giao dịch:** Connection cung cấp khả năng quản lý giao dịch (transactions) để đảm bảo tính toàn vẹn của dữ liệu. Điều này cho phép nhóm nhiều thao tác cơ sở dữ liệu lại với nhau thành một đơn vị công việc, đảm bảo rằng tất cả các thao tác đều thành công hoặc tất cả đều bị hủy bỏ trong trường hợp có lỗi.

1.2 Các loại đối tượng Connection trong ADO.NET

Trong ADO.NET, có nhiều loại đối tượng Connection khác nhau để hỗ trợ kết nối với các nguồn dữ liệu đa dạng. Mỗi loại đối tượng Connection phục vụ một mục đích cụ thể và hỗ trợ các loại cơ sở dữ liệu hoặc nguồn dữ liệu khác nhau. Dưới đây là chi tiết về một số loại đối tượng Connection phổ biến:

SqlConnection :

- Mục đích: Sử dụng để kết nối với Microsoft SQL Server.
- Đặc điểm :
 - + Tối ưu hóa cho SQL Server, cung cấp các tính năng đặc biệt của SQL Server như hỗ trợ cho các kiểu dữ liệu đặc biệt (e.g., XML, JSON).
 - + Hỗ trợ các phương thức bảo mật như Integrated Security (sử dụng thông tin đăng nhập của Windows).
 - + Cung cấp các tùy chọn kết nối như Connection Pooling để tối ưu hóa hiệu suất và quản lý tài nguyên.

Ví dụ 1.2.1 :

```

string connectionString = "Data Source=SERVER_NAME;
    Initial Catalog = DATABASE_NAME;
    User ID = USER_NAME;
    Password = PASSWORD; ";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    try
    {
        connection.Open();
        Console.WriteLine("Kết nối thành công đến SQL Server.");

        // Thực hiện các thao tác với cơ sở dữ liệu
    }
    catch (SqlException ex)
    {
        Console.WriteLine("Lỗi khi kết nối đến SQL Server: " +ex.Message);
    }
}

```

Giải thích:

- **SqlConnection** được sử dụng để kết nối với Microsoft SQL Server.
- **connectionString**: Chứa thông tin chi tiết cần thiết để kết nối đến SQLServer, bao gồm tên máy chủ (SERVER_NAME), tên cơ sở dữ liệu (DATABASE_NAME), và thông tin đăng nhập (USER_NAME, PASSWORD).
- **using**: Đảm bảo rằng đối tượng SqlConnection được giải phóng tài nguyên sau khi hoàn tất công việc.
- **connection.Open()**: Mở kết nối đến cơ sở dữ liệu.
- **catch (SqlException ex)**: xử lý các ngoại lệ liên quan đến SQLServer.

OleDbConnection:

- Mục đích : Sử dụng để kết nối với cơ sở dữ liệu Oracle.
- Đặc điểm :
 - + Có thể kết nối với nhiều loại cơ sở dữ liệu khác nhau, bao gồm SQL Server, MySQL, Oracle, và nhiều cơ sở dữ liệu khác.
 - + Cần cấu hình DSN (Data Source Name) hoặc sử dụng DSN-less connection string.

Ví dụ 1.2.2 :

```
string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;
                           Data Source = DatabaseFilePath.accdb;
                           Persist Security Info = False; ";
using (OleDbConnection connection = new OleDbConnection(connectionString))
{
    try
    {
        connection.Open();
        Console.WriteLine("Kết nối thành công đến Microsoft Access.");

        // Thực hiện các thao tác với cơ sở dữ liệu
    }
    catch (OleDbException ex)
    {
        Console.WriteLine("Lỗi khi kết nối đến Access: " + ex.Message);
    }
}
```

Giải thích :

- **OleDbConnection** được sử dụng để kết nối với các nguồn dữ liệu hỗ trợ giao thức OLE DB như Microsoft Access.
- **connectionString**: Chứa thông tin cần thiết để kết nối đến cơ sở dữ liệu Access, bao gồm nhà cung cấp OLE DB (Microsoft.ACE.OLEDB.12.0) và đường dẫn đến file cơ sở dữ liệu (DatabaseFilePath.accdb).
- **connection.Open()**: Mở kết nối đến cơ sở dữ liệu Access.
- **catch (OleDbException ex)**: Bắt và xử lý các ngoại lệ liên quan đến OLE DB.

2. Chuỗi kết nối (Connection String)

2.1 Định nghĩa chuỗi kết nối

Trong lập trình, đặc biệt là khi làm việc với cơ sở dữ liệu, chuỗi kết nối là một khái niệm quan trọng. Nó là một chuỗi ký tự cung cấp thông tin cần thiết để ứng dụng có thể thiết lập và quản lý kết nối với cơ sở dữ liệu. Chuỗi kết nối chứa các thông tin như tên máy chủ, tên cơ sở dữ liệu, thông tin xác thực và các tham số khác liên quan đến việc kết nối.

2.2 Cấu trúc của chuỗi kết nối

Cấu trúc của một chuỗi kết nối thường bao gồm các thành phần chính sau:

- **Data Source**: Thường là tên máy chủ hoặc địa chỉ IP của máy chủ cơ sở dữ liệu. Đây là nơi mà cơ sở dữ liệu đang lưu trữ.
- **Initial Catalog**: Là tên của cơ sở dữ liệu mà bạn muốn kết nối đến. Đây là cơ sở dữ liệu cụ thể mà bạn muốn làm việc.

- **User ID và Password:** Thông tin xác thực cần thiết để truy cập vào cơ sở dữ liệu. Đối với một số loại cơ sở dữ liệu, thông tin xác thực này bao gồm tên người dùng và mật khẩu.
- **Provider:** Thường chỉ định loại nhà cung cấp dữ liệu (provider) khi sử dụng OleDbConnection. Đây là thành phần tùy chọn và không được sử dụng trong tất cả các loại chuỗi kết nối.
- **Các tham số khác:** Ngoài các thành phần cơ bản, có thể có các tham số khác như cài đặt bảo mật, cấu hình hiệu suất, và các tùy chọn kết nối khác.

Các thành phần trong chuỗi kết nối phụ thuộc vào loại cơ sở dữ liệu và nhà cung cấp dữ liệu đang sử dụng. Việc đặt các giá trị đúng cho mỗi thành phần là rất quan trọng để đảm bảo kết nối thành công và an toàn đến cơ sở dữ liệu.

Ví dụ 2.1.1 :

```
string connectionString = "Data Source=SERVER_NAME;
                           Initial Catalog = DATABASE_NAME;
                           User ID = USER_NAME;
                           Password = PASSWORD; ";
```

- **myServerAddress** là địa chỉ hoặc tên máy chủ SQL Server.
- **myDataBase** là tên của cơ sở dữ liệu.
- **myUsername** và **myPassword** là thông tin xác thực của người dùng để truy cập cơ sở dữ liệu.

Ví dụ 2.1.2 :

```
string connectionString = "Provider=Microsoft.ACE.OLEDB.12.0;
                           Data Source = DatabaseFilePath.accdb;
                           Persist Security Info = False; ";
```

- **Microsoft.ACE.OLEDB.12.0** là nhà cung cấp dữ liệu OLE DB được sử dụng để kết nối với cơ sở dữ liệu.
- **DatabaseFilePath.accdb** là tên tệp ACCDB chứa cơ sở dữ liệu mà bạn muốn truy cập.
- **Persist Security Info:** Là một cờ boolean (true hoặc false) chỉ định liệu thông tin xác thực sẽ được lưu trữ trong chuỗi kết nối sau khi kết nối đã được thiết lập hay không. Trong trường hợp này, **False** cho biết rằng thông tin xác thực sẽ

không được lưu trữ trong chuỗi kết nối, điều này giúp bảo mật thông tin xác thực.

3. Tạo và quản lý đối tượng Connection

3.1 Cách khởi tạo đối tượng Connection

Để thiết lập kết nối đến cơ sở dữ liệu trong ADO.NET, chúng ta sử dụng đối tượng Connection. Cách thông thường để khởi tạo đối tượng Connection là sử dụng một chuỗi kết nối (connection string) chứa các thông tin cần thiết để kết nối đến cơ sở dữ liệu.

Ví dụ 3.1:

```
string connectionString = "Data Source=SERVER_NAME;  
Initial Catalog = DATABASE_NAME;  
User ID = USER_NAME;  
Password = PASSWORD; ";  
SqlConnection connection = new SqlConnection(connectionString);
```

3.2 Phương thức Open() và Close()

Sau khi đã khởi tạo đối tượng Connection, chúng ta có thể mở kết nối bằng cách gọi phương thức `Open()`, và đóng kết nối bằng cách gọi phương thức `Close()`.

Ví dụ 3.2 :

```
connection.Open();  
  
// Thực hiện các thao tác với cơ sở dữ liệu  
connection.Close();
```

3.3 Sử dụng khối using để quản lý tài nguyên hiệu quả

Để đảm bảo rằng tài nguyên được giải phóng đúng cách ngay cả khi có lỗi xảy ra, chúng ta có thể sử dụng khối `using` trong C#. Khối `using` sẽ tự động giải phóng tài nguyên sau khi kết thúc khối lệnh.

Ví dụ 3.3 :

```
using (SqlConnection connection = new SqlConnection(connectionString))  
{  
    connection.Open();  
    // Thực hiện các thao tác với cơ sở dữ liệu  
} // Kết nối sẽ tự động được đóng khi ra khỏi khối using
```

Trong ví dụ này, khi khối `using` kết thúc, đối tượng `Connection` sẽ tự động được đóng mà không cần phải gọi phương thức `Close()` một cách rõ ràng. Điều này giúp quản lý tài nguyên hiệu quả và giảm nguy cơ xảy ra lỗi do quên đóng kết nối.

4. Xử lý ngoại lệ khi làm việc với đối tượng Connection

4.1 Các ngoại lệ thường gặp khi làm việc với đối tượng Connection

Khi làm việc với đối tượng Connection trong ADO.NET, có thể xảy ra một số ngoại lệ khác nhau do các lỗi liên quan đến kết nối cơ sở dữ liệu. Dưới đây là một số ngoại lệ thường gặp và cách xử lý chúng :

SqlException:

- Nguyên nhân : Đây là loại ngoại lệ chính thường gặp khi có lỗi xảy ra trong quá trình thực thi các lệnh SQL trên cơ sở dữ liệu. Các lỗi có thể bao gồm lỗi kết nối, lỗi cú pháp SQL, hoặc các lỗi cơ sở dữ liệu khác.
- Cách khắc phục :
 - + Bắt và xử lý SqlException bằng cách sử dụng khối try-catch.
 - + Xác định nguyên nhân cụ thể của lỗi từ các thuộc tính của SqlException như Number và Message
 - + Thực hiện các hành động phù hợp như hiển thị thông báo lỗi cho người dùng, ghi log để theo dõi, hoặc thực hiện các thao tác khắc phục.

InvalidOperationException:

- Nguyên nhân : Thường xảy ra khi có thao tác không hợp lệ được thực hiện trên đối tượng Connection, chẳng hạn như mở kết nối khi nó đã được mở, hoặc đóng kết nối khi nó đã được đóng.
- Cách khắc phục :
 - + Xác định nguyên nhân cụ thể của lỗi bằng cách kiểm tra hoạt động được thực hiện trước khi ngoại lệ xảy ra.
 - + Sửa đổi mã nguồn để đảm bảo các thao tác trên đối tượng Connection được thực hiện đúng cách.

IOException:

- Nguyên nhân : Có thể xảy ra khi có lỗi đọc hoặc ghi dữ liệu từ cơ sở dữ liệu, chẳng hạn khi mất kết nối mạng hoặc có sự cố với ổ đĩa.
- Cách khắc phục :
 - + Đảm bảo rằng mạng và các tài nguyên hệ thống khác hoạt động đúng cách.
 - + Xử lý các lỗi kết nối mạng hoặc ổ đĩa trước khi thực hiện các thao tác với cơ sở dữ liệu.

NotSupportedException:

- Nguyên nhân : Xảy ra khi một phương thức hoặc tính năng không được hỗ trợ trong môi trường hiện tại hoặc cho đối tượng Connection cụ thể.

- Cách khắc phục :

- + Kiểm tra xem tính năng hoặc phương thức đang được sử dụng có được hỗ trợ trong môi trường hiện tại không.
- + Nếu không được hỗ trợ, xem xét sử dụng cách thức hoặc phương thức thay thế.

4.2 Cách bắt và xử lý ngoại lệ

Để xử lý các ngoại lệ khi làm việc với đối tượng Connection, chúng ta có thể sử dụng các khối **try-catch** để bắt và xử lý ngoại lệ một cách an toàn. Dưới đây là một ví dụ minh họa:

```
try
{
    // Thực hiện các thao tác với đối tượng Connection
    connection.Open();
    // Thực thi các lệnh SQL, đọc hoặc ghi dữ liệu
}
catch (SqlException ex)
{
    // Xử lý ngoại lệ SqlException
    Console.WriteLine("Đã xảy ra lỗi SQL: " + ex.Message);
}
catch (Exception ex)
{
    // Xử lý các ngoại lệ khác
    Console.WriteLine("Đã xảy ra lỗi: " + ex.Message);
}
finally
{
    // Đảm bảo rằng kết nối được đóng sau khi hoàn thành công việc
    if (connection.State == ConnectionState.Open)
    {
        connection.Close();
    }
}
```

Trong ví dụ trên:

- Các thao tác với đối tượng Connection được thực hiện trong khối try.
- Các ngoại lệ cụ thể được bắt và xử lý trong các khối catch tương ứng.
- Trong khối finally, chúng ta đảm bảo rằng kết nối được đóng sau khi hoàn thành công việc, ngay cả khi có lỗi xảy ra. Điều này giúp đảm bảo việc giải phóng tài nguyên đúng cách và tránh các vấn đề liên quan đến kết nối còn mở dẫn đến lãng phí tài nguyên hoặc lỗi không mong muốn.

Xử lý ngoại lệ khi làm việc với đối tượng Connection là một phần quan trọng của việc phát triển ứng dụng. Bằng cách bắt và xử lý các ngoại lệ một cách chính xác và hiệu

qua, chúng ta có thể tăng tính ổn định và đáng tin cậy của ứng dụng, đồng thời cung cấp trải nghiệm người dùng tốt nhất có thể.

TÀI LIỆU KHAM KHẢO

- [1]. **Trần Đình Quê**, *Lập trình cơ sở dữ liệu với .NET*.
- [2]. **Hướng dẫn lập trình ADO.NET - Viblo.asia** (Truy cập lúc 13 giờ ngày 14/5/2024).
- [3]. **Khóa học Lập trình C# cơ bản và nâng cao - Kyna.vn** (Truy cập lúc 13 giờ ngày 14/5/2024).
- [4]. **Diễn đàn Đạo Nhau Học** (Truy cập lúc 13 giờ ngày 14/5/2024).
- [5]. **Awesome Vietnam C# Projects - GitHub** (Truy cập lúc 13 giờ ngày 14/5/2023)

3.Kịch bản sư phạm

TT	NỘI DUNG	LỜI GIẢNG	HĐ DẠY	HĐ HỌC	THỜI GIAN
1	Ôn định lớp	<p>-Thầy chào cả lớp, mời cả lớp ngồi</p> <p>-Thầy bắt đầu điểm danh, các bạn trật tự nghe điểm danh, khi thầy, hãy trả lời "có". Đồng thời, các em cũng nên quan sát xem liệu có bất kỳ sai sót nào trong việc điểm danh hay không. Nếu có, hãy thông báo cho thầy để có thể sửa chữa ngay lập tức.</p> <p>-Thầy nhận thấy tình hình tham gia lớp của lớp mình đầy đủ, tác phong trang phục đúng với quy định nhà trường. Thầy có lời khen và mong các bạn tiếp tục phát huy tinh thần này trong các tiết học sau</p> <p>-Giờ thì mời các bạn lấy tài liệu, giáo trình ra để chuẩn bị học bài mới</p>	<ul style="list-style-type: none"> - Chào cả lớp -Trình chiếu danh sách lớp và điểm danh - Nhận xét lớp -Mời lớp lấy tài liệu, giáo trình ra 	<ul style="list-style-type: none"> -Cả lớp chào thầy - Quan sát và lắng nghe điểm danh -Lắng nghe nhận xét -Nghiêm túc lấy tài liệu, giáo trình 	5 phút
2	Dẫn nhập	<p>- Ở chương 1 thầy đã hướng dẫn sơ qua cho các bạn kiến trúc của ADO .NET, giờ thì bạn nào có thể dựa vào sơ đồ trên và nhắc lại cho thầy và các bạn nghe, kiến trúc của ADO .NET có mấy thành phần và nó gì những gì ?</p>	<ul style="list-style-type: none"> - Trình chiếu lại sơ đồ kiến trúc của ADO .NET và đặt câu hỏi 	<ul style="list-style-type: none"> - Quan sát sơ đồ và suy nghĩ câu hỏi 	5 phút

		<p>- Thầy mời bạn</p> <p>- Cảm ơn bạn, mời bạn ngồi xuống, bạn đã trả lời chính xác, kiến trúc của ADO .NET gồm các thành phần cơ bản như trên, mỗi thành phần đều đóng vai trò vô cùng quan trọng trong chương trình, ứng dụng. Để hiểu rõ hơn tường thành phần, các hoạt động, khai báo và sử dụng của từng thành phần, Hôm nay thầy sẽ hướng dẫn các bạn thành phần đầu tiên trong kiến trúc đó là đối tượng Connection</p> <p>-Sau khi học xong bài học này , các bạn sẽ hoàn thành mục tiêu sau :</p> <p>Kiến thức :</p> <ul style="list-style-type: none"> + Trình bày được vai trò và chức năng của đối tượng Connection trong ADO.NET. +Phân biệt các loại đối tượng Connection như SqlConnection và OleDbConnection. +Mô tả cấu trúc và cách tạo một chuỗi kết nối (connection string) phù hợp. <p>Kỹ năng:</p>	<p>-Mời ngẫu nhiên một bạn trả lời</p> <p>-Nhận xét câu trả lời và dẫn tới nội dung chi tiết</p>	<p>- Đứng dậy và trả lời câu hỏi</p> <p>- Chú ý lắng nghe và chuanr bị tâm thế vào bài học mới</p>	
--	--	---	--	--	--

		<p>+Viết được chương trình mở và đóng kết nối cơ sở dữ liệu bằng SqlConnection và OleDbConnection.</p> <p>+Xử lý các lỗi kết nối cơ sở dữ liệu một cách hiệu quả và cung cấp được thông tin lỗi chi tiết</p> <p>Thái độ :</p> <p>+Hình thành được tính cẩn thận và tỉ mỉ khi làm việc với cơ sở dữ liệu và quản lý tài nguyên</p> <p>- Nội dung bài học gồm có 4 phần chính :</p> <ul style="list-style-type: none"> +1. Giới thiệu về đối tượng Connection +2. Chuỗi kết nối +3. Cách tạo và quản lý đối tượng Connection +4. Xử lý ngoại lệ khi làm việc với đối tượng Connection 			
3	Giảng bài 1. Giới thiệu về đối tượng Connection 1.1 Vai trò của đối tượng Connection	<p>-Các bạn đã bao giờ tự hỏi làm thế nào một ứng dụng hay chương trình có thể giao tiếp được với cơ sở dữ liệu chưa?</p> <p>- Các bạn hãy tưởng tượng đối tượng Connection giống như cầu nối giữa chương trình, ứng dụng với cơ sở dữ</p>	<ul style="list-style-type: none"> - Trình chiếu slide và ghi tên bài đề mục -Đặt câu hỏi mang tính gợi tính tò mò - Giải thích về câu hỏi đặt ra 	<ul style="list-style-type: none"> -Quan sát, ghi chép -Suy nghĩ về câu hỏi và tò mò về nội dung học - Chú ý lắng nghe và suy nghĩ 	6 phút

	<p>liệu, nó đại diện cho một kết nối vật lý giữa ứng dụng của bạn và cơ sở dữ liệu, cho phép bạn thực hiện các thao tác như đọc dữ liệu từ cơ sở dữ liệu hoặc ghi dữ liệu vào đó</p> <p>- Vậy đối tượng Connection thực sự làm những gì? Các bạn hãy nghiên cứu tài liệu và trả lời cho thầy biết câu hỏi trên</p> <p>-Thày mời bạn phát biểu</p> <p>- Cảm ơn bạn đã trả lời rất chính xác. Bây giờ, chúng ta hãy cùng nhau ngồi xuống và thảo luận kỹ hơn về đối tượng Connection trong ADO.NET. Đây là đối tượng đầu tiên mà bất kỳ ứng dụng nào cũng cần sử dụng để tương tác với cơ sở dữ liệu. Đối tượng Connection đóng một vai trò vô cùng quan trọng và có nhiều chức năng cụ thể. Hãy cùng đi vào chi tiết các chức năng này :Đầu tiên, nó giúp thiết lập và quản lý trạng thái kết nối giữa ứng dụng và cơ sở dữ liệu, bao gồm mở và đóng kết nối. Nó cung cấp các tùy chọn bảo</p>	<p>-Đặt câu hỏi cho người học tìm hiểu tài liệu</p> <p>-Mời ngẫu nhiên một bạn trả lời</p> <p>- Nhận xét câu trả lời và giảng vào từng nội dung</p>	<p>-Nghiêm túc tìm hiểu tài liệu và xung phong giơ tay trả lời</p> <p>-Đứng dậy trả lời</p> <p>-Lắng nghe và chăm chú ghi chép</p>	
--	---	---	--	--

		<p>mật, xử lý lỗi kết nối và cho phép cấu hình tối ưu hóa thời gian chờ. Cuối cùng, Connection hỗ trợ quản lý giao dịch để đảm bảo tính toàn vẹn và nhất quán của dữ liệu.</p> <p>1.2 Các loại đối tượng Connection trong ADO.NET</p> <p>- Sau khi nghe thầy trình bày về vai trò của đối tượng Connection các bạn có câu hỏi nào muốn đặt cho thầy không</p> <p>- Trong ADO.NET, có nhiều loại đối tượng Connection khác nhau để hỗ trợ kết nối với các nguồn dữ liệu đa dạng. Mỗi loại đối tượng Connection phục vụ một mục đích cụ thể và hỗ trợ các loại cơ sở dữ liệu hoặc nguồn dữ liệu khác nhau. Trên đây là chi tiết về một số loại đối tượng Connection phổ biến :</p> <ul style="list-style-type: none"> + SqlConnection : được sử dụng để kết nối với Microsoft SQL Server, tối ưu hóa hiệu suất và hỗ trợ các tính năng đặc biệt như XML và JSON. Nó cung cấp các phương thức bảo mật như Integrated Security và hỗ trợ Connection Pooling để quản lý tài nguyên hiệu quả. + OleDbConnection kết nối 	<p>- Trả lời câu hỏi</p> <p>-Trình chiếu slide đến nội dung, ghi tiêu mục và trình bày</p> <p>-Trình bày và giải thích nội dung</p>	<p>-Đặt câu hỏi</p> <p>- Tâm trung quan sát và lắng nghe</p> <p>-Lắng nghe, quan sát và ghi chép</p>	14 phút
--	--	--	---	--	----------------

	<p>với các nguồn dữ liệu hỗ trợ giao thức OLE DB như Microsoft Access. Điều đặc biệt là nó có thể kết nối với nhiều loại cơ sở dữ liệu khác nhau. Sau khi kết nối, bạn có thể thao tác với cơ sở dữ liệu và đảm bảo đóng kết nối để tránh lãng phí tài nguyên.</p> <p>- Đây là ví dụ minh họa code để kết nối cơ sở dữ liệu theo SqlConnection và OleDbConnection. Các bạn không cần phải quá lo lắng về các chi tiết bên trong, vì chúng ta sẽ thảo luận chi tiết về chúng trong các phần tiếp theo.</p> <p>- Các bạn có thắc mắc không</p> <p>- Tiếp theo, chúng ta cùng tìm hiểu về chuỗi kết nối.</p> <p>- Các bạn đã từng nghe về khái niệm chuỗi kết nối chưa?</p> <p>- Thầy mời bạn trả lời</p>	<p>-Đưa ra ví dụ minh họa</p> <p>- Giải đáp thắc mắc</p> <p>-Chuyển slide, ghi tiêu nội dung</p> <p>-Đặt câu hỏi</p> <p>-Mời phát biểu</p>	<p>-Quan sát và suy nghĩ</p> <p>-Đặt ra câu hỏi</p> <p>-Quan sát chép</p> <p>-Suy nghĩ về câu hỏi và xung phong phát biểu</p> <p>-Trả lời câu hỏi</p>	
	<h2>2. Chuỗi kết nối (ConnectionString)</h2> <h3>2.1 Định nghĩa chuỗi kết nối</h3>			3 phút

		<p>-Cảm ơn bạn, mời bạn ngồi xuống, bạn trả lời khá là chính xác. Chuỗi kết nối là một chuỗi ký tự cung cấp thông tin cần thiết để ứng dụng có thể thiết lập và quản lý kết nối với cơ sở dữ liệu. Nó chứa các thông tin quan trọng như tên máy chủ, tên cơ sở dữ liệu, thông tin xác thực và các tham số khác liên quan đến việc kết nối.</p>	<p>-Nhận xét câu trả lời và trình bày nội dung</p>	<p>-Chăm chú lắng nghe và ghi chép</p>	
	2.2 Cấu trúc của chuỗi kết nối	<p>- Các bạn đã hiểu được định nghĩa của chuỗi kết nối, tiếp theo thầy sẽ cùng các bạn nghiên cứu chi tiết các thành phần của chuỗi kết nối</p> <p>-Một chuỗi kết nối thường bao gồm các thành phần chính sau : Data Source là tên máy chủ hoặc địa chỉ IP của cơ sở dữ liệu. Initial Catalog là tên cụ thể của cơ sở dữ liệu bạn muốn truy cập. User ID và Password cung cấp thông tin xác thực cho quyền truy cập. Provider chỉ định nhà cung cấp dữ liệu khi sử dụng OleDbConnection. Các tham số khác có thể bao gồm cài đặt bảo mật và hiệu suất.</p> <p>- Để minh họa cách sử dụng chuỗi kết nối, chúng ta có thể xem xét các ví dụ sau:</p>	<p>-Ghi tiêu mục, chuyển slide, trình bày nội dung</p>	<p>-Lắng nghe, quan sát và ghi chép</p> <p>-Giải thích các thành phần của chuỗi kết nối</p>	<p>15 phút</p>

		<p>+ Trong ví dụ 2.1.1, chúng ta thấy một chuỗi kết nối được sử dụng để kết nối với SQL Server, với thông tin như tên máy chủ, tên cơ sở dữ liệu và thông tin xác thực của người dùng.</p> <p>+Trong ví dụ 2.1.2, chúng ta thấy một chuỗi kết nối được sử dụng để kết nối với một tệp cơ sở dữ liệu Access, với các thông tin về nhà cung cấp dữ liệu OLE DB và đường dẫn đến tệp ACCDB.</p> <p>- Sau khi nghe thầy trình bày về cấu trúc của chuỗi kết nối và thông qua hai ví dụ minh họa. Các bạn có câu hỏi nào muốn đặt cho thầy không</p>	ví dụ		
	3. Tạo và quản lý đối tượng Connection 3.1 Cách khởi tạo đối tượng Connection	<p>- Nếu như các bạn không còn câu hỏi nào nữa thì chúng ta sẽ sang phần tiếp theo, cách tạo và quản lý đối tượng Connection</p> <p>- Cách thông thường để khởi tạo đối tượng Connection là sử dụng một chuỗi kết nối (connection string) chúng ta đã tìm hiểu kỹ ở phần trước, chứa các thông tin cần thiết để kết nối đến cơ sở dữ liệu</p>	-Ghi đề mục, chuyển slide -Trình bày nội dung	- Tập trung quan sát và ghi chép - Lắng nghe và ghi chép	7 Phút

		<ul style="list-style-type: none"> - Các bạn quan sát ví dụ trong giáo trình. Bạn nào có thể giải thích lại các thành phần trong dòng chuỗi kết nối không ? - Xin mời bạn phát biểu <p>-Cảm ơn câu trả lời của bạn, bạn hiểu bài tốt , nắm vững các kiến thức về chuỗi ở phần trước. Đối với việc khởi tạo đối tượng Connection , đầu tiên các bạn cần khởi tạo một chuỗi kết nối với các thành phần như bạn vừa phát biểu, Tiếp theo, một đối tượng SqlConnection được tạo để thiết lập kết nối đến cơ sở dữ liệu sử dụng chuỗi kết nối này.</p>	<ul style="list-style-type: none"> - Đặt câu hỏi - Mời phát biểu 	<ul style="list-style-type: none"> - Đọc tài liệu và xung phong trả lời câu hỏi -Trả lời phát biểu <p>-Nhận xét câu trả lời và giải thích nội dung</p>	
	3.2 Phương thức Open() và Close()	<ul style="list-style-type: none"> - Tiếp theo chúng ta cùng nhau nghiên cứu hai phương thức cơ bản của đối tượng Connection là Open() và Close() -Khi đã tạo đối tượng Connection, chúng ta sử dụng phương thức Open() để mở kết nối với cơ sở dữ liệu và bắt đầu thực hiện các thao tác. Sau khi hoàn thành công việc, để đóng kết nối và giải phóng tài nguyên, chúng ta sử dụng phương thức Close(). Điều này giúp quản lý tài 	<ul style="list-style-type: none"> -Chuyển slide và ghi tiêu mục -Trình bày nội dung chức năng Open(), Close() 	<ul style="list-style-type: none"> - Theo dõi và ghi bài - Chú ý lắng nghe và ghi chú 	4 Phút

		nguyên hiệu quả và đảm bảo tính toàn vẹn của dữ liệu			
	3.3 Sử dụng khối using để quản lý tài nguyên hiệu quả	<p>- Sang phân tiếp theo , thầy sẽ hướng dẫn các bạn tìm hiểu cách sử dụng khối using để quản lý hiệu quả tài nguyên</p> <p>- Khối using thường được sử dụng cho các đối tượng cần được giải phóng ngay sau khi không còn cần thiết nữa, như các đối tượng kết nối đến cơ sở dữ liệu. Ví dụ, khi chúng ta kết nối đến cơ sở dữ liệu bằng đối tượng SqlConnection, chúng ta thường muốn đảm bảo rằng kết nối được đóng sau khi đã hoàn thành việc thực hiện các thao tác.</p> <p>- Đến đây các bạn có nhận ra một điều, close() và using đều được sử dụng để giải phóng tài nguyên không lẽ chúng là một sao ?, các bạn hãy thảo luận và đưa ra những điểm khác biệt giữa chúng</p> <p>- Thầy mời bạn</p> <p>- Cảm ơn bạn, bạn trả lời khá chi tiết. Thầy sẽ giải thích lại cho các bạn dễ hiểu là điểm khác biệt chính giữa hai</p>	<p>- Trình chiếu slide đến nội dung và ghi tiêu mục</p> <p>-Trình bày cách sử dụng và chức năng của using</p> <p>-Đặt câu hỏi thảo luận</p> <p>-Mời phát biểu</p> <p>- Nhận xét câu trả lời và giải thích thêm</p>	<p>- Quan sát và ghi chú</p> <p>- Chú ý theo dõi và ghi chép</p> <p>- Thực hiện thảo luận tìm câu trả lời</p> <p>-Trả lời câu hỏi thảo luận được</p> <p>- Chú ý lắng nghe và ghi chép</p>	7 phút

	<p>phương pháp này là cách chúng được sử dụng và quản lý tài nguyên. Trong khi <code>Close()</code> là một phương thức cụ thể được gọi trên đối tượng cụ thể để giải phóng tài nguyên và không thể khi có ngoại lệ, <code>using</code> là một cấu trúc ngôn ngữ tự động giải phóng tài nguyên ngay sau khi không còn cần thiết nữa ngay cả khi có ngoại lệ xảy ra., mang lại sự tiện lợi và tự động hóa trong quản lý tài nguyên.</p> <p>- Sau khi nghe thầy giải thích xong có bạn nào có thắc mắc gì không ?</p>			
	<p>4. Xử lý ngoại lệ khi làm việc với đối tượng Connection</p> <p>4.1 Các ngoại lệ thường gặp khi làm việc với đối tượng Connection</p>	<p>- Trong quá trình làm việc với đối tượng <code>Connection</code> trong ADO.NET, chúng ta thường phải đối mặt với các ngoại lệ có thể xảy ra. Các ngoại lệ này thường làm gián đoạn luồng, ở phần này thầy sẽ cho các bạn làm quen với các ngoại lệ thường gặp và cách khắc phục</p> <p>-SqlException: Xảy ra khi có lỗi trong quá trình thực thi lệnh SQL. Cần bắt và xử lý ngoại lệ trong khôi <code>try-catch</code>, xác định nguyên nhân và hiển</p>	<p>-Trình chiếu slide đến nội dung và ghi đè mục</p> <p>-Trình bày cách nhận biết ngoại lệ và cách khắc phục</p>	<p>- Chú ý theo dõi và ghi chú</p> <p>4 phút</p> <p>-Tập chung lắng nghe và ghi chép</p>

		<p>thị thông báo lỗi.</p> <p>InvalidOperationException: Xảy ra khi thao tác không hợp lệ trên đối tượng Connection. Khắc phục bằng cách xác định nguyên nhân và sửa đổi mã nguồn để thao tác đúng cách.</p> <p>OException: Xảy ra khi có lỗi đọc hoặc ghi dữ liệu. Đảm bảo mạng và tài nguyên hệ thống hoạt động đúng, xử lý lỗi kết nối trước khi thao tác với cơ sở dữ liệu.</p> <p>NotSupportedException: Xảy ra khi phương thức hoặc tính năng không được hỗ trợ. Kiểm tra tính năng hỗ trợ và sử dụng phương thức thay thế nếu cần.</p>			
	4.2 Cách bắt và xử lý ngoại lệ	<ul style="list-style-type: none"> - Tiếp theo Chúng ta sẽ nói về cách xử lý ngoại lệ khi làm việc với đối tượng Connection trong ADO.NET. Điều này rất quan trọng để đảm bảo rằng ứng dụng của chúng ta hoạt động ổn định và tài nguyên được quản lý đúng cách - Các bạn quan sát ví dụ trong tài liệu, try – catch các bạn đã quá quen với các môn học trước rồi nên thầy không nhắc lại nữa, thầy sẽ giải thích ví dụ này cho các em dễ hiểu : Trong ví dụ trên, các 	<ul style="list-style-type: none"> -Chuyển slide đến nội dung và ghi tiêu mục 	<ul style="list-style-type: none"> - Theo dõi và chăm chú ghi chép 	7 phút

		<p>thao tác với đối tượng Connection được thực hiện trong khối try. Nếu có lỗi xảy ra, chúng ta sẽ bắt và xử lý các ngoại lệ trong các khối catch tương ứng. Ví dụ, nếu xảy ra lỗi SQL, khối catch (SqlException ex) sẽ bắt và xử lý ngoại lệ này, hiển thị thông báo lỗi cụ thể Ngoài ra, chúng ta có khối finally. Đây là phần mã được thực thi bất kể có xảy ra ngoại lệ hay không. Trong khối finally, chúng ta đảm bảo rằng kết nối được đóng sau khi hoàn thành công việc. Điều này giúp giải phóng tài nguyên đúng cách và tránh các vấn đề liên quan đến kết nối còn mở, dẫn đến lãng phí tài nguyên hoặc lỗi không mong muốn.</p> <p>- Nếu các bạn có khó khăn gì hãy đặt câu hỏi cho thầy</p>			
4	Củng cố kiến thức	<p>- Như vậy thầy đã hướng dẫn xong cho các bạn phân nội dung kiến thức của đối tượng Connection, và để xem mức độ hiểu bài của các bạn thầy có một số câu hỏi củng cố dành cho các bạn</p> <p>- Chiếu slide câu hỏi và mời trả lời</p>	<ul style="list-style-type: none"> - Chuyển ý sang củng cố bài học - Trình chiếu slide câu hỏi 	<ul style="list-style-type: none"> - Lắng nghe - Tập trung theo dõi và suy nghĩ tìm đáp án 	10 phút

			- Nhận xét câu trả lời	- Lắng nghe và ghi nhận nhận xét	
5	Hướng dẫn tự học	<ul style="list-style-type: none"> - Các bạn có thể tự tìm tài liệu trên mạng - Về nhà các bạn xem lại nội dung bài hôm nay, đồng thời xem trước nội dung bài tiếp theo - Nếu không còn gì thắc mắc các bạn dọn dẹp vệ sinh tại chỗ ngồi các bạn và ra về 	<ul style="list-style-type: none"> - Giới thiệu tài liệu tham khảo - Nhắc nhở xem bài cũ và chuẩn bị bài mới - Nhắc nhở dọn vệ sinh - Lau bảng, chào lớp, ra về 	<ul style="list-style-type: none"> - Theo dõi, lắng nghe - Lắng nghe và ghi nhận 	3 phút

Giáo án số: 03

Thời gian thực hiện: 2 tiết

Tên BH : Đối tượng Connection

Thực hiện : 16/09/2024

TÊN BÀI: ĐỐI TƯỢNG CONNECTION

Mục tiêu của bài:

Sau khi học xong bài này NH có khả năng:

- Trình bày được vai trò và chức năng của đối tượng Connection trong ADO.NET.
- Phân biệt các loại đối tượng Connection như SqlConnection và OleDbConnection.
- Mô tả cấu trúc và cách tạo một chuỗi kết nối (connection string) phù hợp.
- Viết được chương trình mở và đóng kết nối cơ sở dữ liệu bằng SqlConnection và OleDbConnection.
- Xử lý các lỗi kết nối cơ sở dữ liệu một cách hiệu quả và cung cấp được thông tin lỗi chi tiết
- Hình thành được tính cẩn thận và tỉ mỉ khi làm việc với cơ sở dữ liệu và quản lý tài nguyên

I. Ôn định lớp học:

Thời gian: 5 phút

- Chào cả lớp
- Trình chiếu danh sách lớp và điểm danh
- Nhận xét tình hình tham gia lớp, tác phong trang phục và nhắc nhở những bạn thường xuyên vắng học
- Mời lớp lấy tài liệu và giáo trình chuẩn bị cho bài học mới

II. Thực hiện BH.

TT	Nội dung	HĐ DH		Thời Gian
		HĐ của GV	HĐ của học sinh	
1	Dẫn nhập - Tên bài : Đối tượng Connection -Mục tiêu bài học	- Trình chiếu lại sơ đồ kiến trúc của ADO .NET	- Quan sát - Quan sát sơ đồ và	5 phút

	<p>+ Kiến thức + Kỹ năng +Thái độ</p> <p>- Nội dung bài học mới</p> <ol style="list-style-type: none"> 1. Giới thiệu về đối tượng Connection 2. Chuỗi kết nối 3. Cách tạo và quản lý đối tượng Connection 4. Xử lý ngoại lệ khi làm việc với đối tượng Connection 	<ul style="list-style-type: none"> - Đặt câu hỏi về các thành phần sơ đồ - Mời ngẫu nhiên một bạn trả lời - Nhận xét câu hỏi đã nêu và dẫn vào nội dung chi tiết -Trình bày mục tiêu bài học - Khái quát nội dung bài học 	<p>suy nghĩ câu trả lời</p> <p>- Trả lời câu hỏi</p> <p>-Lắng nghe nhận xét câu hỏi và chuẩn bị tâm thế vào bài học mới</p> <p>-Chú ý lắng nghe</p> <p>-Chăm chú theo dõi</p>	
2	Giảng bài mới			
	<p>1. Giới thiệu về đối tượng Connection</p> <p>1.1 Vai trò của đối tượng Connection</p>	<ul style="list-style-type: none"> - Trình chiếu slide và ghi tên bài để mục -Đặt câu hỏi mang tính gợi tính tò mò - Giải thích về câu hỏi đặt ra -Đặt câu hỏi cho người học tìm hiểu tài liệu -Mời ngẫu nhiên một bạn trả lời - Nhận xét câu trả lời và giảng vào từng nội dung 	<p>-Quan sát, ghi chép</p> <p>-Suy nghĩ về câu hỏi và tò mò về nội dung học</p> <p>- Chú ý lắng nghe và suy nghĩ</p> <p>-Nghiêm túc tìm hiểu tài liệu và xung phong giờ tay trả lời</p> <p>-Đứng dậy trả lời</p> <p>-Lắng nghe và chăm chú ghi chép</p>	4 phút

		<ul style="list-style-type: none"> -Trả lời câu hỏi 	<ul style="list-style-type: none"> -Đặt câu hỏi 	
	1.2 Các loại đối tượng Connection trong ADO.NET	<p>Trình chiếu slide đến nội dung, ghi tiêu mục và trình bày</p> <p>-Trình bày và giải thích các loại đối tượng trong Connection</p> <p>-Đưa ra ví dụ minh họa</p> <p>- Giải đáp thắc mắc</p>	<p>Lắng nghe, quan sát và ghi chép</p> <p>-Lắng nghe, suy nghĩ và ghi chép cẩn thận</p> <p>-Tập trung quan sát và suy nghĩ</p> <p>- Đặt câu hỏi</p>	16 phút
	2. Chuỗi kết nối (Connection String) 2.1 Định nghĩa chuỗi kết nối	<p>-Chuyển slide, ghi tiêu nội dung</p> <p>-Đặt câu hỏi</p> <p>-Mời phát biểu</p> <p>-Nhận xét câu trả lời và trình bày nội dung</p>	<p>Quan sát, ghi chép</p> <p>-Suy nghĩ về câu hỏi và xung phong phát biểu</p> <p>-Trả lời câu hỏi</p> <p>-Chăm chú lắng nghe và ghi chép</p>	3 phút
	2.2 Cấu trúc của chuỗi kết nối	<p>-Ghi tiêu mục, chuyển slide, trình bày nội dung</p> <p>- Giải thích các thành phần của chuỗi kết nối</p>	<p>- Tâm trung quan sát và lắng nghe</p> <p>-Lắng nghe, quan sát và ghi chép</p>	15 phút

		<ul style="list-style-type: none"> - Đưa ra ví dụ minh họa và giải thích ví dụ - Trả lời câu hỏi 	<ul style="list-style-type: none"> - Quan sát và suy nghĩ - Đặt ra câu hỏi 	
	<p>3. Tạo và quản lý đối tượng Connection</p> <p>3.1 Cách khởi tạo đối tượng Connection</p>	<ul style="list-style-type: none"> Trình chiếu slide và ghi tên bài đề mục - Đặt câu hỏi mang tính gợi tính tò mò - Giải thích về câu hỏi đặt ra - Đặt câu hỏi cho người học tìm hiểu tài liệu - Mời ngẫu nhiên một bạn trả lời - Nhận xét câu trả lời và giảng vào từng nội dung - Trả lời câu hỏi 	<ul style="list-style-type: none"> - Quan sát, ghi chép - Suy nghĩ về câu hỏi và tò mò về nội dung học - Chú ý lắng nghe và suy nghĩ - Nghiêm túc tìm hiểu tài liệu và xung phong giơ tay trả lời - Đứng dậy trả lời - Lắng nghe và chăm chú ghi chép - Đặt câu hỏi 	7 phút
	3.2 Phương thức Open() và Close()	<ul style="list-style-type: none"> - Chuyển slide và ghi tiêu mục - Trình bày nội dung chức năng Open(), Close() 	<ul style="list-style-type: none"> - Theo dõi và ghi bài - Chú ý lắng nghe và ghi chú 	4 phút
	3.3 Sử dụng khôi using để quản lý tài nguyên hiệu quả	<ul style="list-style-type: none"> - Ghi đề mục, chuyển slide 	<ul style="list-style-type: none"> - Tập trung quan sát và ghi chép 	7 phút

		<ul style="list-style-type: none"> -Trình bày cách Sử dụng khói using để quản lý tài nguyên hiệu quả - Đặt câu hỏi - Mời phát biểu -Nhận xét câu trả lời và giải thích nội dung 	<ul style="list-style-type: none"> - Lắng nghe và ghi chép - Đọc tài liệu và xung phong trả lời câu hỏi -Trả lời phát biểu -Chú ý lắng nghe và ghi chép 	
	<p>4. Xử lý ngoại lệ khi làm việc với đối tượng Connection</p> <p>4.1 Các ngoại lệ thường gặp khi làm việc với đối tượng Connection</p>	<ul style="list-style-type: none"> -Trình chiếu slide đến nội dung và ghi đè mục -Trình bày cách nhận biết ngoại lệ và cách khắc phục 	<ul style="list-style-type: none"> - Chú ý theo dõi và ghi chú -Tập chung lắng nghe và ghi chép 	4 phút
	4.2 Cách bắt và xử lý ngoại lệ	<ul style="list-style-type: none"> -Chuyển slide đến nội dung và ghi tiêu mục -Trình bày cách nội dung và giải thích cụ thể nội dung -Trả lời câu hỏi 	<ul style="list-style-type: none"> - Theo dõi và chăm chú ghi chép -Tập trung lắng nghe, đọc tài liệu và ghi chép -Đặt câu hỏi 	7 phút
3	Củng cố kiến thức	<ul style="list-style-type: none"> - Chuyển ý sang củng cố bài học - Trình chiếu slide câu hỏi 	<ul style="list-style-type: none"> - Lắng nghe - Tập trung theo dõi và suy nghĩ tìm đáp 	10 phút

		<ul style="list-style-type: none"> - Chọn ngẫu nhiên trả lời - Nhận xét câu trả lời 	<ul style="list-style-type: none"> án - Trả lời câu hỏi - Lắng nghe và ghi nhận nhận xét 	
4	Hướng dẫn tự học	<ul style="list-style-type: none"> - Giới thiệu tài liệu tham khảo - Nhắc nhở xem bài cũ và chuẩn bị bài mới - Nhắc nhở dọn vệ sinh - Lau bảng, chào lớp, ra về 	<ul style="list-style-type: none"> - Theo dõi, lắng nghe - Lắng nghe và ghi nhận - Thực hiện dọn vệ sinh - Chào thầy và ra về 	3 phút

IV. Rút kinh nghiệm tổ chức thực hiện:

.....

.....

.....

.....

.....

.....

.....

Trưởng khoa/ trưởng tổ môn

Ngày tháng năm

Giáo viên

Trọng
Nguyễn Văn Trọng

BỘ CÂU HỎI TRẮC NGHIỆM

Câu 1: Đối tượng Connection trong ADO.NET có chức năng gì?

- a) Thiết lập kết nối với cơ sở dữ liệu
- b) Quản lý trạng thái kết nối
- c) Bảo mật kết nối
- d) Tất cả các câu trên

Câu 2: Đối tượng Connection nào được sử dụng để kết nối với Microsoft SQL Server?

- a) OleDbConnection
- b) MySqlConnection
- c) SqlConnection
- d) OdbcConnection

Câu 3 : Thành phần nào trong chuỗi kết nối chứa thông tin xác thực người dùng?

- a) Data Source
- b) Initial Catalog
- c) User ID và Password
- d) Provider

Câu 4 :Khi nào phương thức Open() của đối tượng Connection được gọi?

- a) Để thiết lập kết nối với cơ sở dữ liệu
- b) Để đóng kết nối với cơ sở dữ liệu
- c) Để đọc dữ liệu từ cơ sở dữ liệu
- d) Để ghi dữ liệu vào cơ sở dữ liệu

Câu 5 : Khối lệnh nào trong C# tự động giải phóng tài nguyên sau khi hoàn tất công việc?

- a) try-catch
- b) finally
- c) using
- d) if-else

Câu 6 : Ngoại lệ nào thường gặp khi có lỗi xảy ra trong quá trình thực thi các lệnh SQL trên cơ sở dữ liệu?

- a) InvalidOperationException
- b) IOException
- c) NotSupportedException
- d) SQLException

Câu 7 : Để đảm bảo rằng kết nối được đóng sau khi hoàn thành công việc, chúng ta nên sử dụng khối lệnh nào?

- a) try
- b) catch
- c) finally
- d) throw

Câu 8: Khi có lỗi xảy ra trong kết nối mạng hoặc ổ đĩa, ngoại lệ nào sẽ được ném ra?

- a) InvalidOperationException
- b) IOException
- c) SQLException
- d) NotSupportedException

Câu 9 : Thành phần nào trong chuỗi kết nối chỉ định tên của cơ sở dữ liệu muốn kết nối đến?

- a) Data Source
- b) Initial Catalog
- c) User ID
- d) Password

Câu 10 : Phương thức nào của đối tượng Connection được sử dụng để đóng kết nối?

- a) Open()
- b) Close()
- c) Dispose()
- d) Execute()

3.Hồ sơ bài giảng thực hành

Tên nghề : Quản trị mạng máy tính

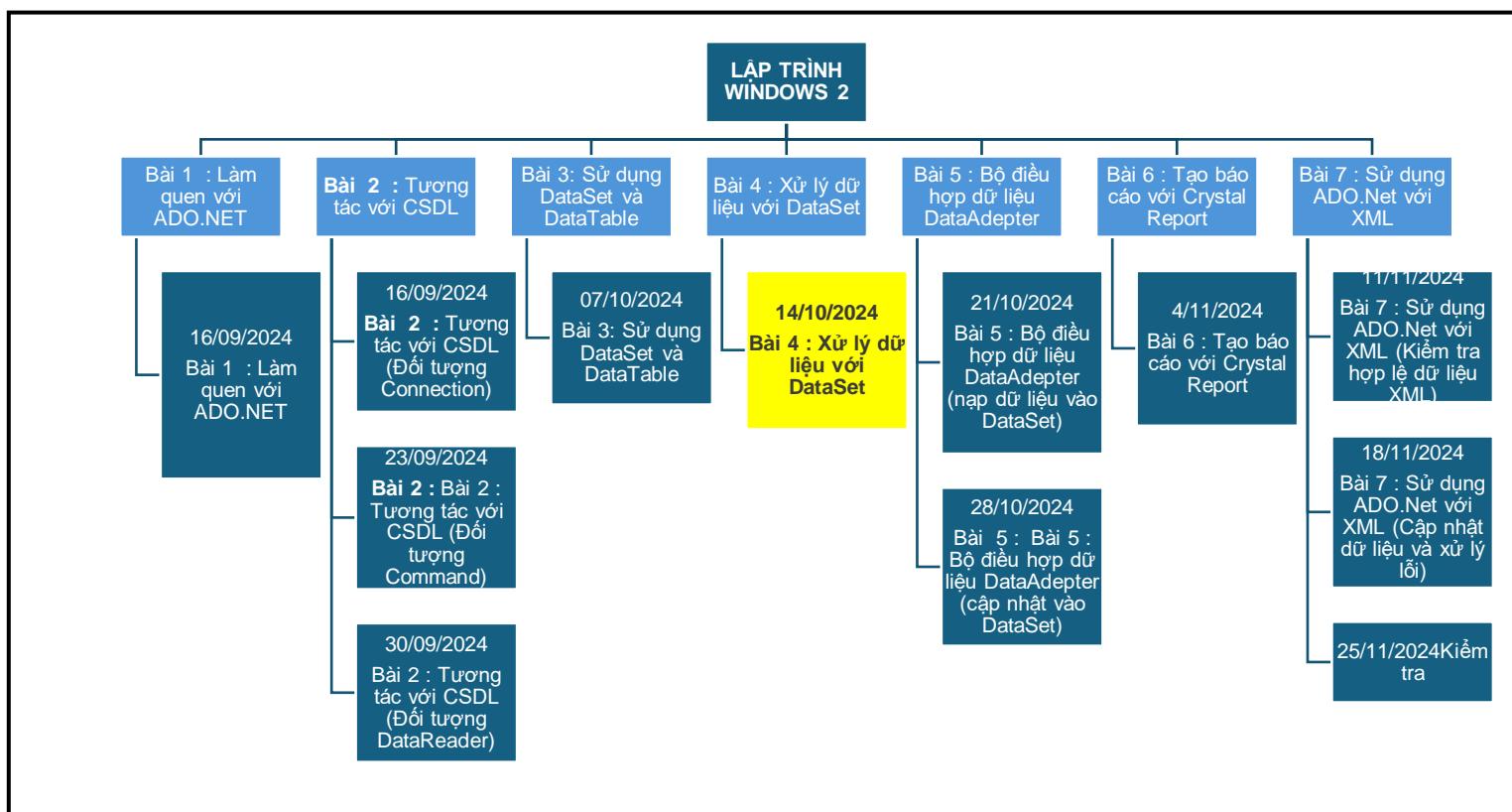
Trình độ : Cao đẳng

Tên môn : Lập trình Windows 2

Thời gian : 5 giờ

Tên bài : Xử lý dữ liệu với DataSet

3.1 Vị trí bài giảng



3.2 Bài giảng

A. MỤC TIÊU

- Trình bày được cách thêm, sửa, xoá và duyệt qua các bảng, cột và hàng trong DataSet.
- Sử dụng thành thạo các phương thức và thuộc tính của DataSet trong lập trình
- Thực hiện các thao tác cơ bản như thêm mới, cập nhật, và xoá dữ liệu trong DataSet
- Nhận biết và khắc phục các lỗi thường gặp khi làm việc với DataSet

- Hình thành tinh thần chủ động học hỏi, tìm hiểu sâu hơn về DataSet và các công cụ xử lý dữ liệu liên quan

B. NỘI DUNG

1. Giới thiệu về DataSet

DataSet là một đối tượng trong .NET Framework dùng để lưu trữ dữ liệu tạm thời trên bộ nhớ, cho phép thao tác với dữ liệu mà không cần kết nối trực tiếp và liên tục với cơ sở dữ liệu.

DataSet có khả năng chứa nhiều bảng dữ liệu (DataTable), mỗi bảng lại chứa các hàng (DataRow) và cột (DataColumn).

Vai trò của DataSet trong lập trình:

- **Lưu trữ dữ liệu tạm thời:** DataSet lưu trữ dữ liệu trên bộ nhớ, giúp các ứng dụng có thể thao tác với dữ liệu mà không cần truy xuất liên tục đến cơ sở dữ liệu, giảm tải cho máy chủ cơ sở dữ liệu và tăng hiệu suất làm việc.
- **Quản lý dữ liệu offline:** DataSet cho phép các ứng dụng làm việc với dữ liệu offline, nghĩa là có thể thao tác với dữ liệu ngay cả khi không kết nối mạng. Khi kết nối lại, có thể đồng bộ dữ liệu với cơ sở dữ liệu gốc.
- **Hỗ trợ dữ liệu quan hệ:** DataSet cho phép tạo và quản lý các mối quan hệ giữa các bảng dữ liệu, tương tự như trong cơ sở dữ liệu quan hệ (RDBMS). Điều này giúp thực hiện các thao tác phức tạp và truy vấn dữ liệu liên quan một cách dễ dàng.
- **Đa năng và dễ sử dụng:** DataSet có thể được sử dụng với nhiều nguồn dữ liệu khác nhau, chẳng hạn như SQL Server, Oracle, XML, v.v. Đồng thời, DataSet có nhiều phương thức và thuộc tính hỗ trợ cho việc thao tác với dữ liệu, giúp lập trình viên dễ dàng xử lý và quản lý dữ liệu.

2. Tạo và cấu hình DataSet

- B1 : Tạo một DataSet mới

Để bắt đầu, tạo một đối tượng DataSet. DataSet có thể được tạo mà không cần bất kỳ tham số nào hoặc có thể được đặt tên :

```
// Tạo một DataSet mới không có tên  
DataSet dataSet1 = new DataSet();  
  
// Tạo một DataSet mới có tên  
DataSet dataSet2 = new DataSet("MyDataSet");
```

-B2 : Thêm các DataTable vào DataSet

Tiếp theo, tạo các bảng dữ liệu (DataTable) và thêm chúng vào DataSet. Mỗi DataTable tương đương với một bảng trong cơ sở dữ liệu.

```

// Tạo một DataTable mới
DataTable table1 = new DataTable("Table1");
// Thêm DataTable vào DataSet
dataSet2.Tables.Add(table1);

```

-B3 : Định nghĩa các DataColumn cho DataTable

Mỗi DataTable cần có các cột (DataColumn) để xác định cấu trúc của dữ liệu. Chúng ta sẽ thêm các DataColumn vào DataTable bằng cách khởi tạo các đối tượng DataColumn và thêm chúng vào thuộc tính Columns của DataTable.

```

// Tạo các DataColumn
DataColumn column1 = new DataColumn("ID", typeof(int));
DataColumn column2 = new DataColumn("Name", typeof(string));

// Thêm các DataColumn vào DataTable
table1.Columns.Add(column1);
table1.Columns.Add(column2);

```

-B4 : Thêm dữ liệu vào DataTable bằng DataRow

Sau khi đã định nghĩa cấu trúc của DataTable, chúng ta có thể thêm dữ liệu vào bảng bằng cách tạo các hàng dữ liệu (DataRow) và thêm chúng vào thuộc tính Rows của DataTable.

```

// Tạo một DataRow mới
DataRow row1 = table1.NewRow();
row1["ID"] = 1;
row1["Name"] = "John Doe";

// Thêm DataRow vào DataTable
table1.Rows.Add(row1);

// Tạo một DataRow khác
DataRow row2 = table1.NewRow();
row2["ID"] = 2;
row2["Name"] = "Jane Smith";

// Thêm DataRow vào DataTable
table1.Rows.Add(row2);

```

Ví dụ quy trình hoàn chỉnh :

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace chuanbidayhoc
{

```

```

internal class Program
{
    static void Main(string[] args)
    {
        // Bước 1: Tạo một DataSet mới có tên
        DataSet dataSet = new DataSet("MyDataSet");

        // Bước 2: Tạo một DataTable mới và thêm vào DataSet
        DataTable table = new DataTable("MyTable");
        dataSet.Tables.Add(table);

        // Bước 3: Tạo các DataColumn và thêm vào DataTable
        DataColumn column1 = new DataColumn("ID", typeof(int));
        DataColumn column2 = new DataColumn("Name", typeof(string));
        table.Columns.Add(column1);
        table.Columns.Add(column2);

        // Bước 4: Tạo các DataRow và thêm vào DataTable
        DataRow row1 = table.NewRow();
        row1["ID"] = 1;
        row1["Name"] = "John Doe";
        table.Rows.Add(row1);

        DataRow row2 = table.NewRow();
        row2["ID"] = 2;
        row2["Name"] = "Jane Smith";
        table.Rows.Add(row2);

        // Hiển thị dữ liệu trong DataSet
        foreach (DataTable dt in dataSet.Tables)
        {
            Console.WriteLine("Table: " + dt.TableName);
            foreach (DataRow dr in dt.Rows)
            {
                foreach (DataColumn dc in dt.Columns)
                {
                    Console.Write(dr[dc] + "\t");
                }
                Console.WriteLine();
            }
        }
    }
}

```

3. Thao tác cơ bản với DataSet

3.1 Thêm mới dữ liệu vào DataSet

-B1 : Tạo một DataRow mới

Tùy khi thêm dữ liệu mới vào DataSet, chúng ta cần tạo một đối tượng DataRow mới để chứa dữ liệu.

```

// Tao một DataRow mới
DataRow newRow = dataSet.Tables["TableName"].NewRow();

```

-B2 : Gán giá trị cho các cột

Sau khi đã tạo DataRow mới, chúng ta gán giá trị cho các cột của DataRow này thông qua tên hoặc chỉ số của cột.

```
// Gán giá trị cho các cột của DataRow mới  
newRow["Column1"] = value1;  
newRow["Column2"] = value2;
```

-B3 : Thêm DataRow vào DataTable

Sau khi đã gán giá trị cho các cột của DataRow, chúng ta thêm DataRow mới này vào DataTable bằng cách sử dụng phương thức `Rows.Add()` của DataTable.

```
// Thêm DataRow vào DataTable  
dataSet.Tables["TableName"].Rows.Add(newRow);
```

3.2 Cập nhật dữ liệu trong DataSet

-B1 : Truy cập DataRow cần cập nhật

Để cập nhật dữ liệu trong DataSet, chúng ta cần truy cập vào DataRow cần cập nhật trong DataTable sử dụng chỉ số hoặc tên của DataRow

```
// Truy cập DataRow cần cập nhật  
DataRow rowToUpdate = dataSet.Tables["TableName"].Rows[index];
```

-B2 : Thay đổi giá trị của các cột

Sau khi đã truy cập vào DataRow cần cập nhật, chúng ta có thể thay đổi giá trị của các cột trong DataRow này

```
// Thay đổi giá trị của các cột trong DataRow cần cập nhật  
rowToUpdate["Column1"] = newValue1;  
rowToUpdate["Column2"] = newValue2;
```

-B3 : Xác nhận cập nhật

Sau khi đã thay đổi dữ liệu của DataRow, chúng ta cần gọi phương thức `AcceptChanges()` của DataRow để xác nhận các thay đổi.

```
// Xác nhận cập nhật  
rowToUpdate.AcceptChanges();
```

3.3 Xóa dữ liệu khỏi DataSet

-B1 : Truy cập DataRow cần xóa

Để xóa dữ liệu khỏi DataSet, chúng ta cần truy cập vào DataRow cần xóa trong DataTable sử dụng chỉ số hoặc tên của DataRow.

```
// Truy cập DataRow cần xóa
```

```
DataRow rowToDelete = dataSet.Tables["TableName"].Rows[index];
```

-B2 : Xóa DataRow

Sau khi đã truy cập vào DataRow cần xóa, chúng ta sử dụng phương thức `Delete()` của DataRow để xóa DataRow này khỏi DataTable.

```
// Xóa DataRow  
rowToDelete.Delete();
```

3.4 Duyệt qua các bảng, cột và hàng trong DataSet

-B1 : Duyệt qua các bảng, cột và hàng trong DataSet

Để duyệt qua các bảng trong DataSet, chúng ta sử dụng vòng lặp `foreach` để lặp qua từng DataTable trong DataSet.

```
// Duyệt qua các bảng trong DataSet  
foreach (DataTable table in dataSet.Tables)  
{  
    Console.WriteLine("Table: " + table.TableName);  
}
```

-B2 : Duyệt qua các cột trong DataTable

Để duyệt qua các cột trong DataTable, chúng ta sử dụng vòng lặp `foreach` để lặp qua từng DataColumn trong DataColumnCollection của DataTable.

```
// Duyệt qua các cột trong DataTable  
foreach (DataColumn column in table.Columns)  
{  
    Console.WriteLine("Column: " + column.ColumnName);  
}
```

-B3 : Duyệt qua các hàng trong DataTable

Để duyệt qua các hàng trong DataTable, chúng ta sử dụng vòng lặp `foreach` để lặp qua từng DataRow trong DataRowCollection của DataTable

```
// Duyệt qua các hàng trong DataTable  
foreach (DataRow row in table.Rows)  
{  
    foreach (DataColumn column in table.Columns)  
    {  
        Console.Write(row[column] + "\t");  
    }  
    Console.WriteLine();  
}
```

4.Quản lý lỗi

4.1 Lỗi thiếu dữ liệu hoặc null

Lỗi thiếu dữ liệu hoặc null có thể xảy ra khi bạn truy cập dữ liệu trong DataSet mà dữ liệu không tồn tại hoặc là null. Điều này có thể dẫn đến các lỗi runtime, NullReferenceException, khi bạn cố gắng truy cập thuộc tính hoặc phương thức của một đối tượng null.

Khi bạn cố gắng truy cập dữ liệu trong DataSet mà không kiểm tra trước, có thể xảy ra trường hợp một hoặc nhiều giá trị trong DataSet không tồn tại hoặc bằng null. Ví dụ, nếu bạn cố gắng truy cập một hàng hoặc một cột mà không tồn tại trong DataSet, hoặc nếu một cột trong DataSet có giá trị là null và bạn không kiểm tra điều kiện này trước khi truy cập, lỗi sẽ xảy ra.

Khắc phục:

- **Kiểm tra dữ liệu trước khi truy cập:** Trước khi truy cập dữ liệu trong DataSet, hãy kiểm tra xem dữ liệu đó có tồn tại hay không bằng cách sử dụng các điều kiện kiểm tra. Điều này giúp đảm bảo rằng chỉ truy cập dữ liệu khi nó tồn tại và không null.
- **Sử dụng điều kiện IF hoặc kiểm tra null:** Sử dụng câu lệnh điều kiện IF hoặc kiểm tra null trước khi truy cập dữ liệu. Nếu dữ liệu không tồn tại hoặc bằng null, hãy thực hiện các hành động xử lý phù hợp, chẳng hạn như báo lỗi hoặc thực hiện các xử lý thay thế.
- **Xử lý ngoại lệ (Exception handling):** Bắt và xử lý các ngoại lệ một cách an toàn nếu lỗi xảy ra. Điều này giúp ứng dụng của bạn trở nên ổn định hơn và tránh việc bị đóng bất ngờ.

Ví dụ :

```
// Kiểm tra xem DataRow có tồn tại và không null trước khi truy cập
if (dataSet.Tables["TableName"].Rows.Count > 0)
{
    DataRow row = dataSet.Tables["TableName"].Rows[0];
    // Thực hiện xử lý khi dữ liệu tồn tại
}
else
{
    // Thông báo lỗi hoặc thực hiện các xử lý khác khi dữ liệu không tồn tại
}
```

4.2 Lỗi định dạng dữ liệu

Lỗi định dạng dữ liệu xảy ra khi dữ liệu trong DataSet không khớp với định dạng mà bạn mong đợi. Điều này có thể dẫn đến các lỗi khi bạn cố gắng chuyển đổi dữ liệu hoặc thực hiện các phép toán trên dữ liệu. Ví dụ, nếu một cột trong DataSet dự kiến chứa số nhưng lại chứa chuỗi, việc chuyển đổi chuỗi này sang số sẽ gây ra lỗi.

Khắc phục :

- **Kiểm tra định dạng dữ liệu trước khi xử lý:** Trước khi thực hiện bất kỳ thao tác nào trên dữ liệu, hãy kiểm tra xem dữ liệu có đúng định dạng không. Có thể sử dụng các phương thức kiểm tra kiểu dữ liệu như `int.TryParse`, `double.TryParse`, `DateTime.TryParse`, v.v.
- **Sử dụng điều kiện kiểm tra định dạng:** Sử dụng các câu lệnh điều kiện để kiểm tra định dạng dữ liệu trước khi xử lý. Nếu dữ liệu không đúng định dạng, hãy thông báo lỗi hoặc thực hiện các xử lý thay thế.
- **Xử lý ngoại lệ khi chuyển đổi dữ liệu:** Bắt và xử lý các ngoại lệ khi chuyển đổi dữ liệu. Điều này giúp ứng dụng trở nên ổn định hơn và tránh bị đóng bất ngờ.

Ví dụ :

```
// Kiểm tra và chuyển đổi dữ liệu trước khi xử lý
if (dataSet.Tables["TableName"].Rows.Count > 0)
{
    DataRow row = dataSet.Tables["TableName"].Rows[0];
    string value = row["ColumnName"].ToString();

    if (int.TryParse(value, out int number))
    {
        // Thực hiện xử lý khi dữ liệu đúng định dạng
    }
    else
    {
        // Thông báo lỗi hoặc thực hiện các xử lý khác khi dữ liệu
    }
}
else
{
    // Thông báo lỗi hoặc thực hiện các xử lý khác khi dữ liệu
}
```

4.3 Lỗi khi truy vấn dữ liệu

Lỗi truy vấn dữ liệu có thể xảy ra khi cú pháp truy vấn SQL không đúng, hoặc khi truy vấn không tìm thấy dữ liệu mong muốn. Điều này có thể dẫn đến việc không lấy được dữ liệu cần thiết hoặc nhận được dữ liệu không chính xác.

Khắc phục :

- **Kiểm tra cú pháp truy vấn SQL:** Trước khi thực hiện truy vấn, hãy kiểm tra cú pháp của câu lệnh SQL để đảm bảo nó đúng. Sử dụng các công cụ hỗ trợ hoặc trình quản lý cơ sở dữ liệu để kiểm tra.

- **Kiểm tra kết quả truy vấn:** Sau khi thực hiện truy vấn, hãy kiểm tra kết quả trả về để đảm bảo rằng nó chứa dữ liệu mong muốn. Nếu không có dữ liệu hoặc dữ liệu không đúng, hãy thực hiện các xử lý phù hợp.
- **Xử lý ngoại lệ khi thực hiện truy vấn:** Bắt và xử lý các ngoại lệ có thể xảy ra khi thực hiện truy vấn, chẳng hạn như `SqlException`. Điều này giúp ứng dụng trở nên ổn định hơn và tránh bị đóng bắt ngay.

Ví dụ :

```

try
{
    string query = "SELECT * FROM TableName WHERE ColumnName = @value";
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@value", value);

    SqlDataAdapter adapter = new SqlDataAdapter(command);
    DataSet dataSet = new DataSet();
    adapter.Fill(dataSet);

    if (dataSet.Tables["TableName"].Rows.Count > 0)
    {
        DataRow row = dataSet.Tables["TableName"].Rows[0];
        // Thực hiện xử lý khi có dữ liệu trả về
    }
    else
    {
        // Thông báo lỗi hoặc thực hiện các xử lý khác
    }
}
catch (SqlException ex)
{
    // Xử lý lỗi khi thực hiện truy vấn SQL
    Console.WriteLine("Lỗi truy vấn SQL: " + ex.Message);
}

```

5. Bài tập rèn luyện

Bài 1 : Tạo một ứng dụng console bằng C# để kết nối với cơ sở dữ liệu SQL Server và lấy dữ liệu vào `DataSet`. Sau đó, hiển thị danh sách nhân viên từ `DataSet` trên console, bao gồm các cột ID, Name và Salary.

Bài 2 : Tạo một `DataSet` chứa một bảng dữ liệu "Products". Thêm một sản phẩm mới vào `DataSet` với các thông tin như ID, Name và Price, sau đó sửa đổi giá của một sản phẩm hiện có và xóa một sản phẩm từ `DataSet`. Cuối cùng, hiển thị danh sách sản phẩm từ `DataSet` sau mỗi thao tác để kiểm tra kết quả.

Bài 3 : Tạo một DataSet chứa dữ liệu mới, ví dụ như thông tin của các đơn hàng. Thêm dữ liệu này vào cơ sở dữ liệu SQL Server sử dụng các thao tác INSERT SQL. Đảm bảo kiểm tra tính hợp lệ của dữ liệu trước khi thêm vào cơ sở dữ liệu.

Bài 4 : Thực hiện việc sửa đổi dữ liệu trong DataSet, ví dụ như sửa giá sản phẩm hoặc thay đổi thông tin khách hàng. Cập nhật các thay đổi này vào cơ sở dữ liệu sử dụng các thao tác UPDATE SQL. Đảm bảo đồng bộ hóa dữ liệu giữa DataSet và cơ sở dữ liệu.

Bài 5 : Thực hiện việc xóa dữ liệu từ DataSet, ví dụ như xóa một đơn hàng hoặc sản phẩm. Xóa các dữ liệu này khỏi cơ sở dữ liệu sử dụng các thao tác DELETE SQL. Đảm bảo đồng bộ hóa dữ liệu giữa DataSet và cơ sở dữ liệu

Bài 6 : Xây dựng các kịch bản kiểm thử để xử lý các trường hợp đặc biệt khi thêm, sửa, xóa dữ liệu từ DataSet và cơ sở dữ liệu, bao gồm:

- Xác nhận thêm mới: Đảm bảo dữ liệu được thêm mới đúng và không gây ra lỗi.
- Xử lý khi sửa đổi dữ liệu không hợp lệ: Kiểm tra và xử lý khi dữ liệu sửa đổi không đáp ứng các ràng buộc hoặc không hợp lệ.
- Xác nhận xóa: Đảm bảo dữ liệu được xóa một cách an toàn và không gây mất mát dữ liệu.

TÀI LIỆU KHAM KHẢO

- [1] Nguyễn Ngọc Hùng (2018), "Giáo Trình Lập Trình .NET", NXB Khoa Học và Kỹ Thuật.
- [2] Trần Hữu Nghị (2020), "Lập Trình Ứng Dụng Với C#", NXB Lao Động Xã Hội
- [3] Joydip Kanjilal (2016), "Mastering C# and .NET Framework", Packt Publishing.
- [4]. Diễn đàn Dạy Nhau Học (Truy cập lúc 16 giờ ngày 20/5/2024).
- [5]. Awesome Vietnam C# Projects - GitHub (Truy cập lúc 16 giờ ngày 20/5/2023)

3.2 Kịch bản sư phạm

TT	NỘI DUNG	LỜI GIẢNG	THỜI GIAN
1	Ôn định lớp	<ul style="list-style-type: none"> - Chào cả lớp - Các bạn trật tự nghe thầy điểm danh - Nhận xét tình hình tham gia lớp và tác phong trang phục - Mời cả lớp tập trung vào bài học mới 	5 phút
2	Dẫn nhập	<ul style="list-style-type: none"> - Đặt câu hỏi: “Các em hãy tưởng tượng mình đang làm việc với một cơ sở dữ liệu rất lớn và phải truy xuất liên tục đến cơ sở dữ liệu này. Các em nghĩ việc này sẽ ảnh hưởng như thế nào đến hiệu suất của ứng dụng?” - Mời ngẫu nhiên một bạn trả lời - Nhận xét câu trả lời và trình chiếu slide bài học, dẫn vào nội dung - Trình bày mục tiêu bài học - Giới thiệu sơ lược về nội dung bài học 	5 phút
3	Hướng dẫn ban đầu	<ul style="list-style-type: none"> - Trình chiếu slide đến nội dung - Giới thiệu lại sơ lược về DataSet - Đặt câu hỏi nhắc lại kiến thức : “Làm thế nào DataSet giúp chúng ta thao tác với dữ liệu mà không cần kết nối liên tục với cơ sở dữ liệu?” - Mời ngẫu nhiên một bạn trả lời - Nhận xét câu trả lời - Dẫn đến nội dung vai trò của DataSet và trình bày nội dung - Chuyển slide đến nội dung cách tạo và cấu hình DataSet - Giải thích từng bước - Thực hiện ví dụ minh họa - Giải thích ví dụ minh họa - Đặt câu hỏi : “Các bạn đã nắm rõ phần này chưa, có gì thắc mắc không ?” - Giải đáp thắc mắc - Trình chiếu slide đến nội dung các thao tác 	35 phút

		<p>cơ bản với DataSet</p> <ul style="list-style-type: none"> - Trình bày các thao tác cơ bản - Hướng dẫn thực hiện các thao tác cơ bản - Thực hiện làm ví dụ minh họa và mời phát biểu trình bày lại cách thực hiện ở mỗi thao tác - Giải thích lại ví dụ - Cố tình thao tác phát sinh lỗi - Đặt câu hỏi : “Trong quá trình thao tác ta gặp phải các lỗi như trên ta phải làm sao ?” - Mời phát biểu - Nhận xét câu trả lời và dẫn đến nội dung quản lý lỗi - Trình bày các lỗi thường gặp - Hướng dẫn cách khắc phục 	
4	Hướng dẫn thường xuyên	<ul style="list-style-type: none"> - Bố trí máy tính cho từng người học - Giao bài tập thực hành - Cho người học thực hành - Quan sát quá trình thực hành của từng người học - Hướng dẫn, chỉnh sửa trong quá trình thực hành - Nhắc nhở chung nếu có quá nhiều người học mắc chung một lỗi 	235 phút
5	Hướng dẫn kết thúc	<ul style="list-style-type: none"> - Nhận xét quá trình thực hành của lớp - Mời ngẫu nhiên các bạn thực hiện lại một thao tác thực hành trong buổi học - Nhận xét và nêu sai sót chung 	15 phút
6	Hướng dẫn kết thúc	<ul style="list-style-type: none"> - Gợi ý tài liệu , bài tập rèn luyện thêm - Yêu cầu xem trước nội dung cho buổi thực hành sau - Yêu cầu người học cắm lại điện, mạng và dọn vệ sinh máy tính nơi mình ngồi 	5 phút

3.4. GIÁO ÁN THỰC HÀNH

GIÁO ÁN SỐ: 4

Thời gian thực hiện: 5 giờ

Tên bài: Xử lý dữ liệu với DataSet

Thực hiện ngày 14 tháng 10 năm 2024

TÊN BÀI: XỬ LÝ DỮ LIỆU VỚI DATASET

MỤC TIÊU CỦA BÀI:

Sau khi học xong bài học này, học viên có khả năng :

- Trình bày được cách thêm, sửa, xoá và duyệt qua các bảng, cột và hàng trong DataSet.
- Sử dụng thành thạo các phương thức và thuộc tính của DataSet trong lập trình
- Thực hiện các thao tác cơ bản như thêm mới, cập nhật, và xoá dữ liệu trong DataSet
- Nhận biết và khắc phục các lỗi thường gặp khi làm việc với DataSet
- Hình thành tinh thần chủ động học hỏi, tìm hiểu sâu hơn về DataSet và các công cụ xử lý dữ liệu liên quan

ĐỒ DÙNG VÀ PHƯƠNG TIỆN DẠY HỌC

- Đồ dùng, dụng cụ : Tài liệu , giáo trình, bảng , bút lông,
- Thiết bị : Tivi, Máy tính, phần mềm soạn thảo code, slide bài giảng

HÌNH THỨC TỔ CHỨC DẠY HỌC:

- Hướng dẫn ban đầu: Cả lớp
- Hướng dẫn thường xuyên : Cá nhân, cả lớp
- Hướng dẫn kết thúc : Cả lớp

I. ÔN ĐỊNH LỚP HỌC

Thời gian : **5 phút**

- Điểm danh
- Nhận xét lớp
- Nhắc nhở

II. THỰC HIỆN BÀI HỌC

TT	NỘI DUNG	HOẠT ĐỘNG DẠY HỌC		THỜI GIAN
		HĐ DẠY	HĐ HỌC	
1	Dẫn nhập			5 phút

	<ul style="list-style-type: none"> - Đặt câu hỏi dẫn đến nội dung bài học - Tên bài : Xử lý dữ liệu với DataSet -Mục tiêu : <ul style="list-style-type: none"> + Kiến thức + Kỹ năng + Thái độ - Nội dung bài học: <ul style="list-style-type: none"> + Giới thiệu về DataSet + Tạo và cấu hình DataSet + Thao tác cơ bản với DataSet + Quản lý lỗi 	<ul style="list-style-type: none"> - Đặt câu hỏi: “Các em hãy tưởng tượng mình đang làm việc với một cơ sở dữ liệu rất lớn và phải truy xuất liên tục đến cơ sở dữ liệu này. Các em nghĩ việc này sẽ ảnh hưởng như thế nào đến hiệu suất của ứng dụng?” - Nhận xét câu trả lời - Trình chiếu slide và dẫn vào nội dung bài học - Trình bày mục tiêu bài học - Chuyển slide giới thiệu sơ lược về nội dung bài học 	<ul style="list-style-type: none"> - Suy nghĩ về câu hỏi, xung phong trả lời - Lắng nghe nhận xét -Chăm chú quan sát , lắng nghe chuẩn bị tâm thế vào bài học - Lắng nghe - Tập trung quan sát , lắng nghe 	
2	<p>Hướng dẫn ban đầu</p> <p>-Giới thiệu sơ lược về DataSet và nhắc lại kiến thức trên lớp lý thuyết</p>	<ul style="list-style-type: none"> - Trình chiếu slide giới thiệu sơ lược về DataSet - Đặt câu hỏi nhắc lại kiến thức lý thuyết : “Làm thế nào DataSet giúp chúng 	<ul style="list-style-type: none"> -Quan sát ,lắng nghe -Đọc tài liệu, suy nghĩ câu trả lời 	35 phút

	<p><i>ta thao tác với dữ liệu mà không cần kết nối liên tục với cơ sở dữ liệu?"</i></p> <ul style="list-style-type: none"> -Mời ngẫu nhiên một bạn trả lời -Nhận xét câu trả lời -Dẫn đến nội dung vai trò của DataSet, trình chiếu slide -Trình bày vai trò của DataSet 	<ul style="list-style-type: none"> -Đứng dậy trả lời câu hỏi -Lắng nghe nhận xét -Nghiêm túc lắng nghe và quan sát -Tập chung lắng nghe 	
	<ul style="list-style-type: none"> -Các bước tạo và cấu hình DataSet: +Tạo DataSet mới +Thêm DataTable vào DataSet + Định nghĩa các DataColumn + Thêm dữ liệu vào DataTable 	<ul style="list-style-type: none"> -Trình chiếu slide nội dung tạo và cấu hình DataSet - Trình bày các bước tạo và cấu hình DataSet -Giải thích chi tiết từng bước thực hiện -Thực hiện làm ví dụ mẫu và giải thích các bước thực hiện ví dụ - Đặt câu hỏi : "<i>Các bạn đã hiểu hết phần này chưa, có thắc mắc gì cứ hỏi thầy ?</i>" 	<ul style="list-style-type: none"> - Theo dõi -Chăm chú quan sát và lắng nghe và ghi chép - Tập trung lắng nghe và ghi chép - Chú ý quan sát và lắng nghe -Đặt câu hỏi cho giáo viên

		<ul style="list-style-type: none"> - Trả lời câu hỏi -Chuyển slide đến nội dung thao tác cơ bản với DataSet -Các thao tác cơ bản: +Thêm +Sửa +Xóa -Hướng dẫn thực hiện từng thao tác với DataSet -Thực hiện làm ví dụ mẫu và mời từng bạn nhắc lại các cách thực hiện thao tác -Cố tình tạo một thao tác phát sinh lỗi và đặt câu hỏi : “Trong quá trình thao tác với DataSet, nếu như gặp phải lỗi phát sinh như trên thì ta phải làm gì” -Quản lý lỗi : +Lỗi thiếu dữ liệu hoặc null +Lỗi định dạng +Lỗi truy vấn - Mời ngẫu nhiên một bạn trả lời -Nhận xét câu hỏi và dẫn đến nội dung quản lý lỗi -Hướng dẫn cách khắc phục lỗi phát sinh 	<ul style="list-style-type: none"> -Lắng nghe và ghi nhớ câu trả lời -Chăm chú theo dõi -Nghiêm túc theo dõi và ghi chép -Ghi chép và lắng nghe -Quan sát, suy nghĩ và xung phong trả lời câu hỏi -Quan sát lỗi phát sinh, thảo luận tìm câu trả lời -Trả lời câu hỏi -Lắng nghe câu hỏi và chuẩn bị tâm thế sang nội dung tiếp theo -Tập trung quan sát lắng nghe và ghi chép 	
3	Hướng dẫn thường xuyên -Tổ chức thực hành ,	- Bố trí máy tính cho từng	-Ngồi theo sự bố trí của	235 phút

	mỗi các nhân một máy tính, rèn luyện các bài tập được giao	người học - Giao bài tập thực hành - Cho người học thực hành - Quan sát quá trình thực hành của từng người học - Hướng dẫn, chỉnh sửa trong quá trình thực hành - Nhắc nhở chung nếu có quá nhiều người học mắc chung một lỗi	giáo viên -Ghi nhận bài tập cần thực hiện -Tiến hành thực hành -Nghiêm túc thực hành -Chăm chú lắng nghe và chỉnh sửa theo góp ý của giáo viên -Tập chung lắng nghe và ghi nhận	
4	Hướng dẫn kết thúc Tổng kết lại nội dung buổi học thực hành	- Nhận xét quá trình thực hành của lớp - Mời ngẫu nhiên các bạn thực hiện lại một thao tác nhiều bạn mắc lỗi trong buổi học - Nhận xét và nêu sai sót chung	-Chú ý lắng nghe -Thực hiện thao tác theo yêu cầu của giáo viên -Tập trung lắng nghe và ghi chép	15 phút
5	Hướng dẫn tự rèn luyện	- Gợi ý tài liệu , bài tập rèn luyện thêm - Yêu cầu xem trước nội dung cho buổi thực hành sau - Yêu cầu người học cắm lại điện, mạng và dọn vệ sinh máy tính nơi mình ngồi	-Ghi nhận lại các tài liệu, bài tập gợi ý -Nghiêm túc lắng nghe -Thực hiện theo yêu cầu của giáo viên	5 phút

IV. Rút kinh nghiệm tổ chức thực hiện:

.....
.....
.....
.....
.....

Trưởng khoa/ trưởng tổ môn

Ngày tháng năm

Giáo viên

Trọng
Nguyễn Túan Trọng

3.5 Phiếu hướng dẫn thực hiện

3.5.1 Phiếu hướng dẫn thực hiện Tạo và cấu hình DataSet

Mô đun : Xử lý dữ liệu với DataSet					
Kỹ năng : Tạo và cấu hình DataSet					
TT	Bước	Tiêu chuẩn	Phương pháp	Phương tiện	Lưu ý / An toàn
1	Tạo DataSet mới	<ul style="list-style-type: none"> - Khởi tạo đối tượng DataSet với tên duy nhất - Chọn một tên phản ánh rõ mục đích và nội dung của DataSet. 	<ul style="list-style-type: none"> - Mở Visual Studio - Tạo một dự án mới bằng cách chọn File > New > Project. - Trong cửa sổ Solution Explorer, nhấp chuột phải vào dự án của bạn và chọn Add > New Item - Chọn DataSet từ danh sách và đặt tên cho DataSet của bạn 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code 	<ul style="list-style-type: none"> Đảm bảo đặt tên DataSet một cách duy nhất và rõ ràng
2	Thêm DataTable mới vào DataSet	<ul style="list-style-type: none"> - Khởi tạo được một đối tượng DataTable mới. - Gán tên duy nhất cho DataTable. - Thêm được DataTable vào DataSet. 	<ul style="list-style-type: none"> - Khởi tạo một DataSet bằng phương thức New DataSet() - Tạo một DataTable mới bằng phương thức new DataTable("NewTable") - Thêm các cột DataTable mới 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code 	<ul style="list-style-type: none"> Chắc chắn tên DataTable được chọn là hợp lệ và mô tả chính xác
3	Định nghĩa các cột cho DataTable	<ul style="list-style-type: none"> - Khởi tạo đúng các đối tượng DataColumn - Gán tên và kiểu dữ liệu cho mỗi cột 	<ul style="list-style-type: none"> - Lấy biến datatable bạn vừa tạo thực hiện tạo cột bằng phương thức Datatable.Columns.Add(1,2) Với (1) là tên của cột (2) là kiểu dữ liệu của cột 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code 	<ul style="list-style-type: none"> Đảm bảo các kiểu dữ liệu được chọn phù hợp với nhu cầu

		<ul style="list-style-type: none"> - Thêm được các cột vào DataTable 			
4	Thêm dữ liệu vào DataTable	<ul style="list-style-type: none"> - Khởi tạo đúng một đối tượng DataRow mới <ul style="list-style-type: none"> - Gán giá trị cho từng cột - Tạo được một DataRow mới. - Thêm được DataRow vào DataTable 	<ul style="list-style-type: none"> - Sau khi đã thêm các cột cho datatble bạn tiếp tục thêm các row dữ liệu bằng phương thức dataTable.NewRow() 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code 	<p>Kiểm tra tính hợp lệ của dữ liệu trước khi thêm vào</p>

3.5.2 Phiếu hướng dẫn thực hiện Thao tác cơ bản với DataSet

Mô đun : Xử lý dữ liệu với DataSet					
Kỹ năng : Thao tác cơ bản với DataSet					
TT	Bước	Tiêu chuẩn	Phương pháp	Phương tiện	Lưu ý / An toàn
1	Thêm mới dữ liệu vào DataSet	<ul style="list-style-type: none"> - Tạo một DataRow mới để chứa dữ liệu cần thêm. - Gán giá trị cho các cột của DataRow - Thêm được DataRow mới vào DataTable 	<ul style="list-style-type: none"> - Đầu tiên, bạn khởi tạo một DataSet để chứa các bảng dữ liệu. new DataSet(); - Thực hiện tạo Datable các bước tương tự nhu trong phiếu hướng dẫn “tạo vào câu hinhd DataSet” với nhiều 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code 	<p>Kiểm tra tính hợp lệ của dữ liệu trước khi thêm vào</p>

		bảng dữ liệu		
2	Cập nhật dữ liệu trong DataSet	<ul style="list-style-type: none"> - Thay đổi được giá trị của các cột trong DataRow. 	<ul style="list-style-type: none"> - Truy cập vào row cần cập nhật dữ liệu <code>dataset.Tables["1"].rows["2"]</code> +Với “1” là tên table +“2” là index của dòng dữ liệu - Cập nhật : <code>row["name"] = "update"</code> 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code <p>Đảm bảo chỉnh sửa đúng dữ liệu và xác nhận cập nhật hợp lệ và mô tả chính xác</p>
3	Xóa dữ liệu khỏi DataSet	<ul style="list-style-type: none"> - Xóa được dữ liệu DataRow cần xóa trong DataTable. 	<ul style="list-style-type: none"> - Truy cập vào datarow cần xóa <code>dataSet.Tables["MyTable"].Rows.Find(id)</code> - Thực hiện xóa : <code>dataSet.Tables["MyTable"].Rows.Remove(rowToDelete);</code> 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code <p>Xác nhận trước khi xóa và kiểm tra lại dữ liệu sau xóa</p>
4	Duyệt qua các bảng, cột và hàng trong DataSet	<ul style="list-style-type: none"> - Duyệt qua từng bảng trong - Duyệt qua từng cột trong -Duyệt qua từng hàng trong <code>DataRowCollection</code> của DataTable 	<ul style="list-style-type: none"> - Sử dụng vòng lặp <code>foreach</code> để duyệt qua từng cột trong mỗi DataTable - Sử dụng thuộc tính <code>Columns</code> của DataTable để lấy danh sách các cột - Sử dụng vòng lặp <code>foreach</code> để duyệt qua từng hàng (<code>DataRow</code>) trong mỗi DataTable. 	<ul style="list-style-type: none"> - Máy tính - Phần mềm soạn thảo code <p>Kiểm tra mỗi bước duyệt qua để đảm bảo tính chính xác</p>

			- Sử dụng thuộc tính Rows của DataTable để lấy danh sách các hàng		
--	--	--	---	--	--

3.5.3 Phiếu hướng dẫn thực hiện Quản lý lỗi

Mô đun : Xử lý dữ liệu với DataSet

Kỹ năng : Quản lý lỗi

TT	Bước	Tiêu chuẩn	Phương pháp	Phương tiện	Lưu ý / An toàn
1	Xử lý lỗi khi thiếu dữ liệu hoặc null	Báo lỗi hoặc thực hiện các xử lý thay thế nếu dữ liệu không tồn tại hoặc là null	Sử dụng câu lệnh điều kiện IF hoặc kiểm tra null	- Máy tính - Phần mềm soạn thảo code	Kiểm tra dữ liệu trước khi truy cập để tránh lỗi
2	Xử lý lỗi khi định dạng dữ liệu không đúng	- Thông báo lỗi hoặc thực hiện các xử lý khác nếu dữ liệu không đúng định dạng	Sử dụng câu lệnh điều kiện IF hoặc kiểm tra định dạng dữ liệu - Sử dụng các phương thức int.TryParse, double.TryParse, DateTime.TryParse để kiểm tra định dạng dữ liệu.	- Máy tính - Phần mềm soạn thảo code	Kiểm tra định dạng dữ liệu trước khi xử lý
3	Xử lý lỗi khi thực hiện truy vấn dữ liệu	- Bắt và xử lý các ngoại lệ có thể xảy ra khi thực hiện	Sử dụng câu lệnh try-catch	- Máy tính - Phần mềm soạn thảo code	Bắt và xử lý các ngoại lệ một cách an toàn và đúng đắn

		truy vấn, chẳng hạn như SqlException.			
--	--	--	--	--	--

3.6 Phiếu đánh giá quy trình

3.6.1 Phiếu đánh giá quy trình tạo và cấu hình DataSet

Khóa học	Lập trình Window 2 2024-2025							
Kỹ năng	Tạo và cấu hình DataSet							
Học viên								
Hướng dẫn: <i>Đánh dấu ✓ vào những bước mà học viên đó thực hiện VÀ đảm bảo tiêu chuẩn</i>								
TT	Bước	Tiêu chuẩn	Lưu ý an toàn lao động	✓				
1	Tạo DataSet mới	Khởi tạo đối tượng DataSet với tên duy nhất						
		Chọn một tên phản ánh rõ mục đích và nội dung của DataSet						
2	Thêm DataTable mới vào DataSet	Khởi tạo một đối tượng DataTable mới						
		Gán tên cho DataTable						
		Thêm được DataTable vào DataSet						
3	Định nghĩa các cột cho DataTable	Gán tên và kiểu dữ liệu cho mỗi cột						
4	Thêm dữ liệu vào DataTable	Khởi tạo được một đối tượng DataRow mới						
		Gán đúng giá trị cho từng cột của DataRow dựa trên cấu trúc đã định nghĩa						

	Tạo chính xác một DataRow mới		
	Thêm dữ liệu vào thành công		

3.6.2 Phiếu đánh giá quy trình thao tác với DataSet

Khóa học	Lập trình Window 2 2024-2025					
Kỹ năng	Thao tác với DataSet					
Học viên						
Hướng dẫn: <i>Đánh dấu ✓ vào những bước mà học viên đó thực hiện VÀ đảm bảo tiêu chuẩn</i>						
TT	Bước	Tiêu chuẩn	Lưu ý an toàn lao động			
1	Thêm mới dữ liệu vào DataSet	Tạo một DataRow mới	.			
		Gán giá trị cho các cột của DataRow				
		Thêm được DataRow mới vào DataTable trong DataSet				
2	Cập nhật dữ liệu trong DataSet	Thay đổi đúng giá trị của các cột cần cập nhật trong DataRow				
		Xác nhận các thay đổi				
3	Xóa dữ liệu khỏi DataSet	Xóa được dữ liệu chỉ định				
		Xác nhận trước khi xóa và kiểm tra lại dữ liệu sau xóa.				
4	Duyệt qua các bảng, cột và hàng trong DataSet	Duyệt đúng qua từng cột trong DataColumnCollection của DataTable.				
		Duyệt đúng qua từng hàng				

		trong DataRowCollection của DataTable		
--	--	---------------------------------------	--	--

3.6.2 Phiếu đánh giá quy trình quản lý lỗi

Khóa học	Lập trình Window 2 2024-2025							
Kỹ năng	Quản lý lỗi							
Học viên								
Hướng dẫn: <i>Đánh dấu ✓ vào những bước mà học viên đó thực hiện VÀ đảm bảo tiêu chuẩn</i>								
TT	Bước	Tiêu chuẩn	Lưu ý an toàn lao động	✓				
1	Xử lý lỗi khi thiếu dữ liệu hoặc null	Sử dụng câu lệnh điều kiện IF hoặc kiểm tra null trước khi truy cập dữ liệu	.					
		Báo lỗi hoặc thực hiện các xử lý thay thế nếu dữ liệu không tồn tại hoặc là null						
2	Xử lý lỗi khi định dạng dữ liệu không đúng	Thông báo lỗi hoặc thực hiện các xử lý khác nếu dữ liệu không đúng định dạng						
		Sử dụng các phương thức kiểm tra kiểu dữ liệu						
3	Xử lý lỗi khi thực hiện truy vấn dữ liệu	Có thao tác kiểm tra cú pháp truy vấn SQL trước khi thực hiện truy vấn						
		Thông báo kết lỗi khi xảy ra lỗi không mong muốn						
		Bắt và xử lý các ngoại lệ một cách an toàn và đúng đắn						

3.7 Phiếu đánh giá sản phẩm

PHIẾU ĐÁNH GIÁ SẢN PHẨM

Tên sản phẩm: DataSet sau khi thao tác

Mã số:

Tên học sinh: Ngày:

TT	Tiêu chuẩn	Bằng chứng	Đánh giá	
			Đạt	Không đạt
1	Kỹ thuật	- Tiêu chuẩn 1: Tạo và cấu hình thành công DataSet		
		- Tiêu chuẩn 2: Thao tác thêm , sửa , xóa chính xác với DataSet		
		- Tiêu chuẩn 3: Quản lý lỗi hiệu quả trong khi làm việc với DataSet		
2	Thẩm mỹ	- Tiêu chuẩn 1: Soạn thảo code theo đúng format, dễ nhìn		
		- Tiêu chuẩn 2: Soạn thảo code theo đúng format, dễ nhìn		
		- Tiêu chuẩn 3: Soạn thảo code theo đúng format, dễ nhìn		
3	An toàn	- Không có		

4	Thời gian	- Tiêu chuẩn 1 : Hoàn thành thao tác không quá 7 phút			
		- Tiêu chuẩn 2 : Đảm bảo thời gian thực hiện không quá 30 phút			

4. Hồ sơ bài giảng tích hợp

Tên nghề: Tin học viễn thông ứng dụng

Trình độ: Cao đẳng

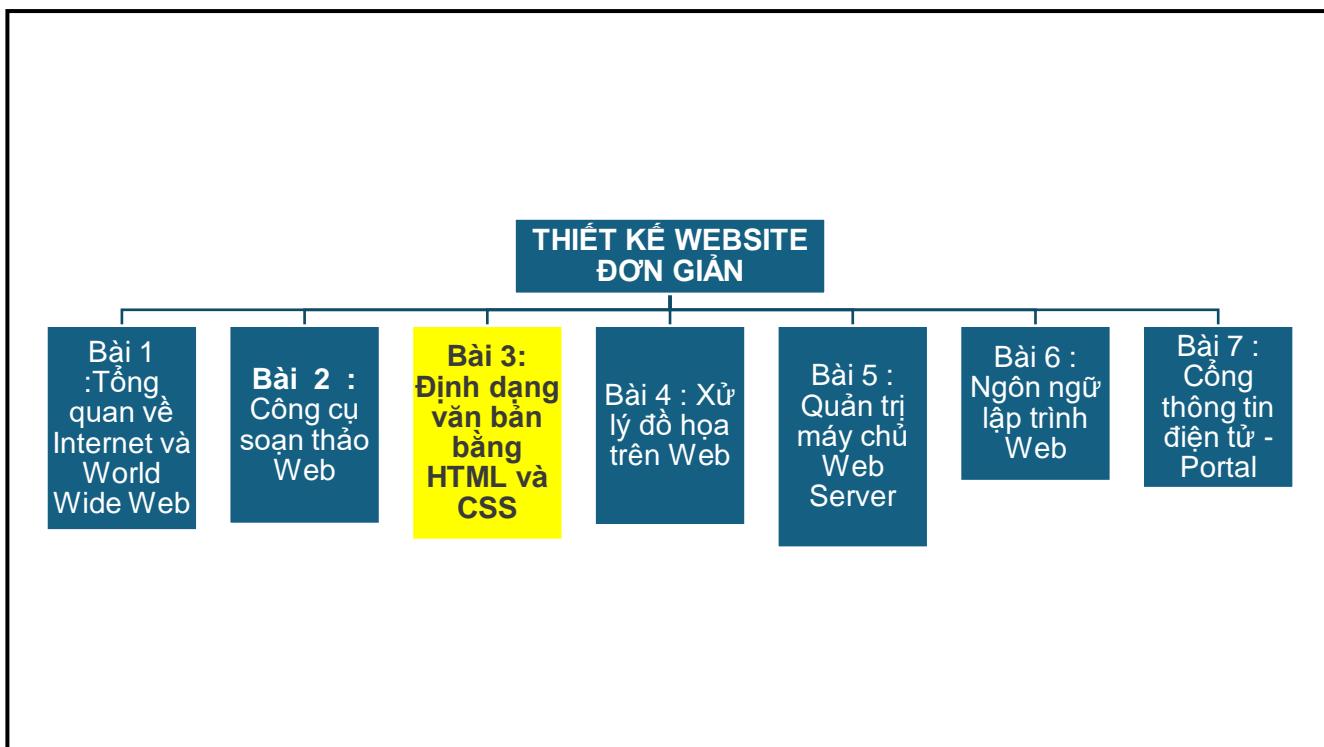
Tên môn học: Thiết kế và quản trị Web

Tên chương/Bài: Định dạng văn bản bằng HTML và CSS

Thời gian: 2.5 giờ

Tên bài : KẾT HỢP HTML VÀ CSS TRONG XÂY DỰNG GIAO DIỆN WEB

4.1 Vị trí bài giảng



4.2 Bài giảng

A. MỤC TIÊU

- Trình bày được vai trò của HTML và CSS trong việc xây dựng bà thiêt kế giao diện web
- Phân biệt được các tính năng và cách sử dụng của các thẻ HTML cơ bản
- Nêu được các thuộc tính cơ bản của CSS
- Áp dụng các thuộc tính CSS cơ bản để định dạng giao diện web

- Sử dụng HTML và CSS để tạo cấu trúc và định dạng một trang web đơn giản
- Sử dụng các công cụ phát triển như trình duyệt web hoặc Developer Tools để kiểm tra, thử nghiệm và điều chỉnh giao diện web
- Tạo được bố cục giao diện web : bố cục một cột, hai cột, hoặc ba cột, sử dụng các kỹ thuật layout như flexbox hoặc grid.
- Khuyến khích tư duy sáng tạo trong việc thiết kế giao diện web
- Phát triển tính kiên nhẫn và tỉ mỉ trong việc viết mã và kiểm tra lỗi

B. NỘI DUNG

1. Giới thiệu về HTML và CSS

HTML (HyperText Markup Language) : là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web, nghĩa là các mẫu thông tin được trình bày trên World Wide Web. Ban đầu, vào năm 1991, HTML được Tim Berners-Lee tạo ra với một số thẻ cơ bản để định nghĩa cấu trúc của một trang web. Từ đó, HTML đã trải qua nhiều phiên bản và cải tiến, từ HTML 1.0 cho đến HTML5, với mỗi phiên bản đều mang lại các tính năng mới và cải tiến, giúp nâng cao khả năng đa dạng và tương tác của các trang web.

CSS (Cascading Style Sheets) : định nghĩa về cách hiển thị của một tài liệu HTML. CSS đặc biệt hữu ích trong việc thiết kế Web. Được giới thiệu vào năm 1996 và nhanh chóng trở thành một phần không thể thiếu trong việc phát triển trang web. CSS cho phép các nhà phát triển web áp dụng các kiểu dáng như màu sắc, phông chữ, bố cục và hiệu ứng, giúp tạo ra các trang web có giao diện đẹp mắt và chuyên nghiệp. Nó giúp cho người thiết kế dễ dàng áp đặt các phong cách đã được thiết kế lên bất kì page nào của website một cách nhanh chóng, đồng bộ.

Tính nhất quán giữa HTML và CSS là yếu tố quan trọng nhất trong việc xây dựng trang web. Bằng cách kết hợp chúng một cách hiệu quả, nhà phát triển web có thể tạo ra các trang web phong phú về nội dung, thú vị về giao diện và dễ sử dụng cho người dùng.

2. Làm quen với HTML

2.1 Cấu trúc cơ bản của HTML

Có thể dùng notepad hoặc text editor để soạn thảo 1 tài liệu HTML. Ta cùng làm 1 ví dụ nhỏ về HTML nhé:

- + B1: Mở notepad hoặc text editor
- + B2: Nhập đoạn HTML sau:

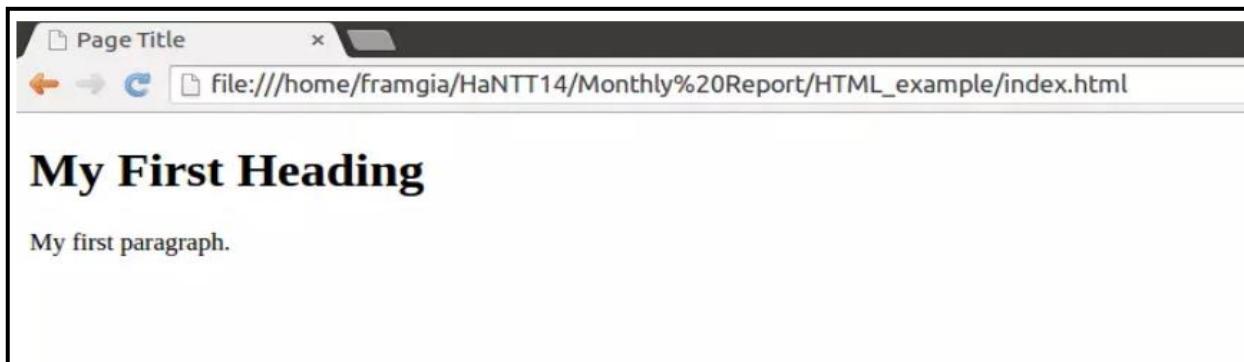
```
File Edit View

<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
</head>
<body>
    <h1>My First Heading</h1>
    <p>My first paragraph</p>
</body>
</html>
```

+ B3 : Save file với định dạng .html hoặc .htm

+ B4 : Mở file bằng 1 trình duyệt bất kỳ

Kết quả :



Cấu trúc cơ bản của HTML:

- Một tài liệu HTML luôn được gói trong cặp tag <html> và </html>
- Cặp tag <body> và </body> sẽ là nơi mô tả những gì có thể nhìn thấy của trang.
Như vậy, một trang web viết bằng html sẽ có cấu trúc cơ bản như sau:

```
<html>
  <body>
    Phần trình bày nội dung
  </body>
</html>
```

Ngoài phần body còn có phần head, được viết bởi cặp tag `<head></head>`. Nếu bạn sử dụng cặp tag này, bạn bắt buộc phải viết thêm một cặp tag nữa, đó là `<title></title>`. Giữa `<title>` và `</title>` là tên của trang web được hiển thị phía trên cùng của menubar. Như vậy một trang web với lúc này sẽ có cấu trúc như sau:

```
<head>
  <title>Tiêu đề của trang web</title>
</head>
<body>
  Phần trình bày nội dung
</body>
</html>
```

2.2 Các thẻ cơ bản trong HTML

Một tài liệu HTML được tạo nên từ các cặp thẻ html

- Thẻ được bắt đầu bằng dấu `<` và kết thúc bằng dấu `>`
- Tên thẻ nằm giữa cặp dấu `<>`

Ví dụ 1: `<a>` tức là đang khai báo thẻ a

- Cặp thẻ được tạo nên từ thẻ mở và thẻ đóng

Ví dụ 2: `<a>` trong đó `<a>` là thẻ mở, `` là thẻ đóng

- Nội dung của thẻ sẽ được nằm giữa thẻ đóng và thẻ mở
- Cũng có một số thẻ chỉ có thẻ mở mà không có thẻ đóng.

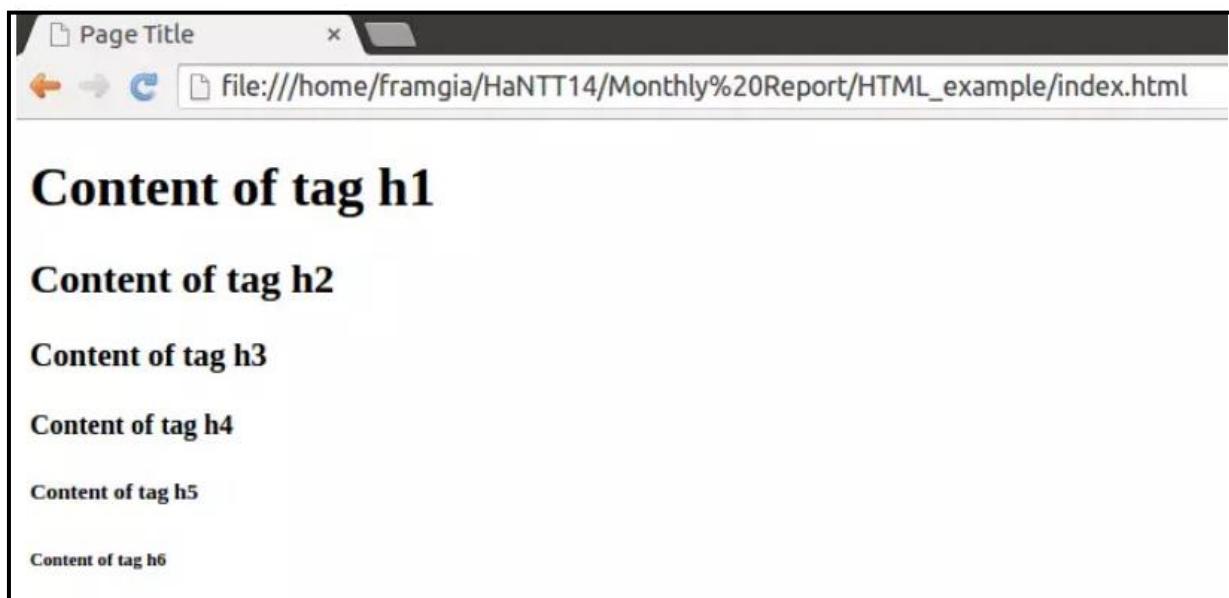
2.2.1 Các thẻ tiêu đề (HTML Headings):

- Thường được sử dụng để thể hiện cho tiêu đề của bài viết, bản tin hay các mục nhấn mạnh
- Bao gồm các thẻ từ `<h1>` đến `<h6>`
- Font chữ của nội dung trong các thẻ giảm từ `<h1>` đến `<h6>`

Ví dụ: nhập đoạn lệnh bên dưới vào trong cặp thẻ `<body></body>`

```
<h1>Content of tag h1</h1>
<h2>Content of tag h2</h2>
<h3>Content of tag h3</h3>
<h4>Content of tag h4</h4>
<h5>Content of tag h5</h5>
<h6>Content of tag h6</h6>
```

Và kết quả là:



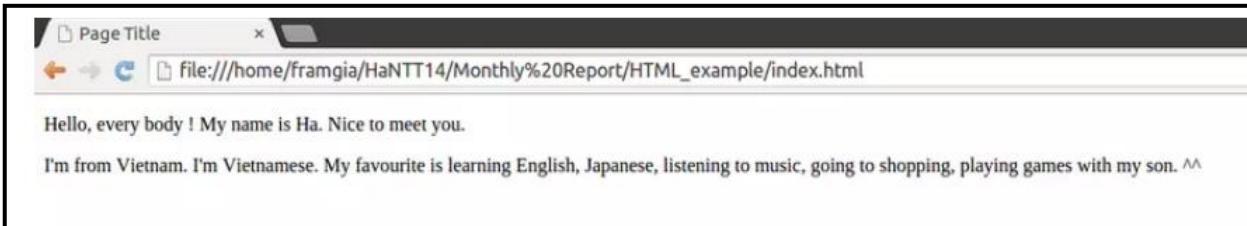
2.2.2 Đoạn văn bản trong html (HTML Paragraphs)

Nội dung văn bản được thể hiện trong cặp thẻ `<p></p>`

Ví dụ : nhập đoạn lệnh bên dưới vào cặp thẻ <body></body>

```
<p> Hello, every body !</p>
<p> My name is Ha. Nice to meet you.</p>
```

Và kết quả là:



2.2.3 Liên kết (HTML Links):

- a dùng cặp thẻ <a> để làm công việc liên kết các trang web với nhau
- Thuộc tính của thẻ <a> gồm:
 - href: qui định địa chỉ mà url trỏ tới
 - target: qui định liên kết sẽ được mở ra ở đâu
 - _blank: cửa sổ mới
 - _self: trang hiện tại

Ví dụ: nhập đoạn lệnh bên dưới vào cặp thẻ <body></body>

```
<a href="https://google.com.vn" target="_blank">Go to google page</a>
```

Và kết quả là : trang google được mở ra trên 1 cửa sổ mới

2.2.4 Thẻ tạo danh sách

Danh sách là một tập hợp các phần tử (ví dụ danh sách các học sinh, các món ăn, các môn học ...). Có hai loại danh sách, danh sách có thứ tự và không có thứ tự.

- Danh sách có thứ tự (ordered list) được tạo ra bằng thẻ , mỗi phần tử khi xuất hiện có chỉ số ở đầu (1,2, 3 ... A, B, C ...)

- Danh sách không có thứ tự (unordered list) được tạo ra bằng thẻ ``, mỗi phần tử khi xuất hiện có thể có ký hiệu như dấu chấm, gạch ngang ... các phần tử.

Đây là một ví dụ về danh sách có thứ tự:

```
<ol>
    <li>HTML</li>
    <li>CSS</li>
    <li>Javascript</li>
    <li>C#</li>
</ol>
```

Kết quả :

```
1. HTML
2. CSS
3. Javascript
4. C#
```

Đây là một ví dụ về danh sách không có thứ tự:

```
<ul type="circle">
  <li>HTML</li>
  <li>CSS</li>
  <li>Javascript</li>
  <li>C#</li>
</ul>
```

Kết quả:

- HTML
- CSS
- Javascript
- C#

2.2.5 Các thẻ khác

- (bold): Chữ In đậm
- <u> (Underline): Chữ gạch chân
- <i> (italic): Chữ in nghiêng
- <big> (Big): Chữ lớn hơn
- <sub> (Subscript) Chỉ số dưới, ví dụ: H₂O
- <sup> (Superscript): Chỉ số trên, ví dụ: x²y
- In đậm (nhấn mạnh)
- (emphasized): Chữ in nghiêng, Nhấn mạnh hơn <i>

3. Làm quen với CSS

3.1.1 Cú pháp cơ bản CSS

CSS được viết dưới dạng các quy tắc (rules) gồm một bộ chọn (selector) và một tập hợp các thuộc tính và giá trị.

Ví dụ :

```
selector {  
    property1: value1;  
    property2: value2;  
}
```

3.1.2 Cách sử dụng CSS trong tài liệu HTML

□ CSS có thể được sử dụng bằng cách nhúng (inline), nội tuyến (internal), hoặc bên ngoài (external).

□ Sử dụng inline CSS: <p style="color: red;">Ví dụ về văn bản màu đỏ</p>

Ví dụ 1 : Sử dụng internal CSS

```
<head>  
    <style>  
        p {  
            color: blue;  
        }  
    </style>  
</head>
```

Ví dụ 2 : Sử dụng external CSS

```
<link rel="stylesheet" href="styles.css">
```

3.1.3 Các thuộc tính cơ bản

- Màu sắc (color):

- Định nghĩa màu chữ cho phần tử.
- Ví dụ: color: red;, color: #00ff00;, color: rgb(255, 0, 0);

- Phông chữ (font-family, font-size, font-weight):

- font-family: Định nghĩa font chữ cho phần tử.
- font-size: Định nghĩa kích thước của font chữ.
- font-weight: Định nghĩa độ đậm nhè của font chữ.
- Ví dụ: font-family: Arial, sans-serif;, font-size: 16px;, font-weight: bold;

- Kích thước (width, height):

- width: Định nghĩa chiều rộng của phần tử.
- height: Định nghĩa chiều cao của phần tử.
- Ví dụ: width: 300px;, height: 200px;

- Định dạng (margin, padding):

- margin: Khoảng cách giữa phần tử và các phần tử khác bên ngoài.

- padding: Khoảng cách giữa nội dung của phần tử và biên của nó.
- Ví dụ: margin: 10px; padding: 20px;

- Hiển thị và vị trí (display, position):

- display: Xác định cách phần tử được hiển thị trong layout.
- position: Xác định vị trí của phần tử trong trang.
- Ví dụ: display: block; position: relative;

3.1.4 Lựa chọn phần tử trong CSS

- Theo tên, class hoặc id của phần tử HTML:

- Tên: Lựa chọn phần tử dựa trên tên của nó (ví dụ: p, div, header).
- Class: Lựa chọn phần tử dựa trên class mà nó thuộc vào (ví dụ: .container, .button).
- ID: Lựa chọn phần tử dựa trên id duy nhất của nó (ví dụ: #header, #content).

Sử dụng các chọn lọc như element, .class, #id:

- element: Lựa chọn tất cả các phần tử cùng loại (ví dụ: div, span).
- .class: Lựa chọn tất cả các phần tử có class được chỉ định (ví dụ: .container, .button).
- #id: Lựa chọn phần tử có id duy nhất được chỉ định (ví dụ: #header, #content).

4. Kết hợp HTML và CSS để tạo trang web đơn giản

4.1 Xác định cấu trúc cơ bản HTML

- Mở trình soạn thảo code : có thể sử dụng bất kỳ trình soạn thảo văn bản nào để viết mã HTML, như Notepad, Sublime Text, hoặc VS Code. Trình soạn thảo này giúp bạn viết mã HTML một cách dễ dàng và thuận tiện.

- Tạo tệp HTML với cấu trúc :

+ **Thẻ <!DOCTYPE html>:** Đây là thẻ bắt buộc và đầu tiên trong mọi tài liệu HTML. Nó khai báo loại tài liệu HTML bạn đang sử dụng.

Ví dụ: <!DOCTYPE html>

+ **Thẻ <html>:** Thẻ này bao quanh toàn bộ nội dung trang web của bạn và xác định rằng nội dung là mã HTML. Mọi thứ trong trang web của bạn đều được bao quanh bởi thẻ này.

Ví dụ:

```
<!DOCTYPE html>
<html lang="en">
    <!-- Nội dung trang web ở đây -->
</html>
```

+ **Thẻ <head>**: Thẻ <head> chứa các siêu dữ liệu và tiêu đề trang, cũng như các liên kết với các tệp CSS và JavaScript. Trong thẻ này, bạn có thể đặt các thẻ như <title>, <meta>, và <link> để mô tả trang web và liên kết nó với các tài nguyên bên ngoài.

Ví dụ:

```
<head>
    <title>Tiêu đề trang web của bạn</title>
    <meta charset="UTF-8">
    <meta name="description" content="Mô tả trang web của bạn">
    <link rel="stylesheet" href="style.css">
</head>
```

+ **Thẻ <body>**: Thẻ <body> chứa toàn bộ nội dung hiển thị của trang web. Tất cả các phần tử HTML như văn bản, hình ảnh, liên kết, và các phần tử khác sẽ được đặt trong thẻ này

Ví dụ :

```
<body>
    <!-- Nội dung hiển thị của trang web ở đây -->
</body>
```

4.2. Viết mã HTML

Tạo các phần tử HTML cho tiêu đề, menu, nội dung, footer, vv. Sử dụng các thẻ như <div>, <header>, <nav>, <section>, <article>, <footer> để phân chia nội dung.\

Ví dụ :

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Trang Web Đơn Giản</title>
        <link rel="stylesheet" href="style.css">
    </head>
    <body>
        <header>
            <h1>Tiêu đề trang web</h1>
            <nav>
                <ul>
                    <li><a href="#">Trang chủ</a></li>
                    <li><a href="#">Giới thiệu</a></li>
                    <li><a href="#">Dịch vụ</a></li>
                    <li><a href="#">Liên hệ</a></li>
```

```

        </ul>
    </nav>
</header>

<section id="content">
    <article>
        <h2>Bài viết 1</h2>
        <p>Nội dung bài viết 1...</p>
    </article>
    <article>
        <h2>Bài viết 2</h2>
        <p>Nội dung bài viết 2...</p>
    </article>
</section>

<footer>
    <p>© 2024 Trang web đơn giản. Tất cả các quyền được bảo lưu.</p>
</footer>
</body>
</html>

```

4.3 Tạo và liên kết tệp CSS

- Tạo tệp CSS mới
 - + Mở trình soạn thảo văn bản và tạo một tệp mới.
 - + Lưu tệp với phần mở rộng .css,
ví dụ: style.css.
- Liên kết với tệp CSS trong phần <head> của tài liệu HTML:
Ví dụ:

```

<head>
    <!-- Các thẻ meta và tiêu đề khác -->
    <link rel="stylesheet" href="style.css">
</head>

```

4.4 Viết mã CSS

Xác định các phần tử HTML bạn muốn định dạng, chẳng hạn như tiêu đề, menu, phần nội dung, footer, vv.. Sau đó Sử dụng các thuộc tính CSS như color, font-size, padding, margin, background, vv để định dạng các phần tử theo ý muốn của bạn

Ví dụ :

```
/* Định dạng tiêu đề */
h1 {
    color: blue;
    font-size: 36px;
    margin-bottom: 20px;
}

/* Định dạng menu */
nav ul {
    list-style-type: none;
    padding: 0;
}

nav li {
    display: inline;
    margin-right: 20px;
}

nav a {
    text-decoration: none;
    color: black;
}

/* Định dạng phần nội dung */
#content {
    width: 80%;
    margin: 0 auto;
}

article {
    margin-bottom: 20px;
}

/* Định dạng chân trang */
footer {
    background-color: #f2f2f2;
    padding: 10px 0;
    text-align: center;
}
```

4.5. Kiểm tra trang web

- Mở trình duyệt:

- Mở trình duyệt web mà thường xuyên sử dụng, ví dụ như Google Chrome, Mozilla Firefox, hoặc Microsoft Edge.

- Kiểm tra kết quả:

- Sau khi trình duyệt được mở, nhập địa chỉ tệp index.html vào thanh địa chỉ và nhấn Enter. Trang web sẽ được hiển thị trên trình duyệt.
- Xác nhận rằng trang web hiển thị đúng và áp dụng các quy tắc CSS đã thêm vào trong tệp style.css. Kiểm tra mọi phần tử HTML đã được định dạng đúng cách và màu sắc, font chữ, và hiệu ứng CSS đã được áp dụng theo mong muốn.

5. Bài tập rèn luyện

Bài 1 : Tạo trang web có nội dung là Bài tập sử dụng các tag cơ bản



Bài 2 : Tạo trang web có nội dung được định dạng theo nhiều hình thức

Định dạng văn bản

Hóa học: Nước: H_2O ; Axít Sunfuric: H_2SO_4

Tóan học: Phương trình bậc 2: $\text{ax}^2 + \text{bx} + \text{c} = 0$

Tin học: Vòng lặp For.

```
For i = 0 To 10
    Tong = Tong + 1
Next
```

Bài 3 : Tạo nội dung là bài viết về Thượng Hải, được định dạng theo nhiều hình thức

**THƯỢNG HẢI
TƯƠNG LAI CỦA TRUNG HOA HIỆN ĐẠI**

Không thể so sánh với Bắc Kinh, Tây An về chiều sâu văn hóa hay bề dày lịch sử, Thượng Hải hấp dẫn du khách bởi một khía cạnh khác: điểm nóng nhất của Trung Hoa hiện đại

Bến Thượng Hải

Là sự kết hợp hài hòa một cách kỳ lạ giữa **Liverpool** và **Mahattan** những năm 1920, con phố ăn uống nhất ở Thượng Hải là Trung Sơn lộ, người phương Tây quen gọi là **Bund** còn người Việt Nam gọi là **Bến Thượng Hải**. Xưa kia, đây chính là trung tâm náo nhiệt nhất của Thượng Hải, là khu tô giới bị tranh chấp dữ dội nhất của ba nước Anh, Nhật, Mỹ. Một bên là sông Hoàng Phố, bên kia là những công trình kiến trúc theo lối Tây phương - cũng chính là trung tâm kinh tế đầu não của thành phố này

Bài 4 : Tạo trang web có nội dung là thông tin về công ty Hoa tươi Nhân Ái. Được định nghĩa theo nhiều hình thức

Công ty Hoa tươi Nhân Ái



1. **Giới thiệu**

- Thành lập ngày 25/05/2007
- Chuyên cung cấp các loại hoa tươi
- Có trên 20 cửa hàng bán lẻ tại TP.HCM
- Nhận kết giỏ hoa theo yêu cầu của khách hàng

2. **Liên hệ**

- Điện thoại: 84-08-8351056
- Địa chỉ: 227 Nguyễn Văn Cừ, Quận 5, TP.HCM

© Copyright: [Trung Tâm Tin Học - Đại Học Khoa Học Tự Nhiên TP.HCM](#)

Bài 5 : Tạo trang web có nội dung là danh sách các món ăn ngon Hà Nội



Các món ngon Hà Nội

STT	Tên món ăn
1.	Bánh cuốn Thanh Trì
2.	Xôi Hà Nội
3.	Chả cá Lã Vọng
4.	Cháo Cóc
5.	Bánh Gai Ninh Giang
6.	Bún Thang Hà Nội
7.	Canh Lǎng
8.	Vịt quay lạng Sơn

Bài 6 : Tạo trang web với nội dung là chương trình học lập trình

Chương trình học - ngành lập trình

Nhóm học phần	Môn học	Số tiết	
		Lý thuyết	Thực hành
I Tin học văn phòng	Hệ điều hành Windows	3	5
	Sử dụng Internet	3	5
	Soạn thảo văn bản với MS Word	24	40
	Tạo bài trình diễn với MS PowerPoint	9	15
	Xử lý số liệu với MS Excel	24	40
	Tạo trang Web với MS FrontPage	9	15
II Kỹ thuật lập trình (VB.NET)	Nhập môn lập trình	36	60
	Kỹ thuật lập trình	36	60
.....			

TÀI LIỆU KHAM KHẢO

- [1]. TS. Hồ Văn Lâm (Chủ biên), ThS. Nguyễn Ngọc Dũng, ThS. Nguyễn Thị Kim Phương (2023), *Giáo trình lập trình ứng dụng web*, NXB xây dựng.
- [2]. Jennifer Robbins(2018), *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*.
- [3]. Jon Duckett (2014), *HTML and CSS: Design and Build Websites*.
- [4]. [W3Schools Online Web Tutorials](#) (Truy cập lúc 16h ngày 1/6/2024).
- [5]. VIBLO Partner tìm hiểu về HTML và CSS (Truy cập lúc 10h ngày 1/6/2024)

4.3 Kịch bản sư phạm

TT	NỘI DUNG	LỜI GIẢNG
1	Ôn định lớp	<ul style="list-style-type: none"> - Chào cả lớp - Các bạn trật tự nghe thầy điểm danh - Nhận xét tình hình tham gia lớp và tác phong trang phục - Mời cả lớp tập trung vào bài học mới
2	Dẫn nhập	<ul style="list-style-type: none"> - Mở một trang web bất kì trên internet - Đặt câu hỏi gợi sự tò mò : “Các em có bao giờ thắc mắc làm thế nào các trang web mà các em truy cập hàng ngày, chẵn hạn như trang web này được tạo ra như thế nào không?” - Mời phát biểu - Nhận xét câu trả lời và dẫn vào nội dung chi tiết - Chiếu slide bài học và trình bày mục tiêu bài học - Giới thiệu sơ lược về nội dung bài học
3	Giải quyết vấn đề lý thuyết	<ul style="list-style-type: none"> - Chuyển slide đến phần giới thiệu HTML và CSS , ghi đè mục - Đặt câu hỏi : “Các bạn hãy đọc giáo trình và cho thầy biết HTML được viết tắt của từ gì?” -Mời phát biểu -Nhận xét câu trả lời - Trình bày sơ lược về HTML -Chuyển slide và trình bày nội dung giới thiệu về CSS - Trình chiếu slide sang phần làm quen với HTML và ghi đè mục - Giới thiệu cấu trúc cơ bản của HTML - Cho người học soạn thảo thử cấu trúc HTML bằng notepad và chạy - Giải thích cấu trúc vừa soạn thảo - Chuyển slide sang phần các thẻ trong HTML và ghi đè mục - Giới thiệu các thẻ thông dụng trong HTML - Thực hiện demo ví dụ từng thẻ và giải thích

		<ul style="list-style-type: none"> - Trình chiếu slide nối tiếp sang phần làm quen với CSS và ghi đề mục - Chiếu hình ảnh cấu trúc cơ bản của CSS - Giải thích cấu trúc - Giới thiệu các cách sử dụng CSS - Trình bày các thuộc tính có bản của CSS - Thực hiện demo mẫu và giải thích - Chuyển slide sang nội dung tiếp theo vào ghi đề mục - Cho ví dụ minh họa mối quan hệ giữa HTML và CSS thông qua việc “xây nhà” - Trình bày các bước để xây dựng một trang web đơn giản bằng HTML và CSS - Hướng dẫn thực hiện từng bước
4	Hướng dẫn thường xuyên	<ul style="list-style-type: none"> - Cho người học thực hành - Quan sát quá trình thực hành của từng người học - Hướng dẫn, chỉnh sửa trong quá trình thực hành - Nhắc nhở chung nếu có quá nhiều người học mắc chung một lỗi
5	Thao tác mẫu	<ul style="list-style-type: none"> - Bố trí máy tính cho từng người học
6	Sai phạm thường gặp, nguyên nhân và cách phòng tránh	<ul style="list-style-type: none"> - Lỗi cú pháp - Đặt thẻ link không đúng chỗ hoặc sử dụng sai đường dẫn khiến cho HTML không nhận CSS - Thực hiện trình chiếu, hướng dẫn cách đặt thẻ link và cách lấy đường dẫn
7	Thực hành	<ul style="list-style-type: none"> - Giao bài tập thực hành
8	Kết thúc vấn đề	<ul style="list-style-type: none"> - Nhận xét quá trình thực hành của lớp - Mời ngẫu nhiên các bạn thực hiện lại một thao tác nhiều bạn mắc lỗi trong buổi học - Nhận xét và nêu sai sót chung - Khuyến nghị học sinh hoàn thiện trong các buổi học sau.

9	Hướng dẫn tự học	<ul style="list-style-type: none"> - Yêu cầu người học thực hiện các ví dụ, bài tập mở rộng trong tài liệu học tập. - Yêu cầu người học hoàn thành các bài tập về nhà và nộp trước bài học tiếp theo.
---	------------------	---

4.4. GIÁO ÁN TÍCH HỢP

GIÁO ÁN SỐ:3

Thời gian thực hiện: 2.5 giờ

Tên bài học trước: Kết hợp HTML và CSS trong xây dựng giao diện web

Thực hiện : 25 tháng 11 năm 2024

TÊN BÀI: KẾT HỢP HTML VÀ CSS TRONG XÂY DỰNG GIAO DIỆN WEB

MỤC TIÊU CỦA BÀI:

Sau khi học xong bài này người học có khả năng:

- Trình bày được vai trò của HTML và CSS trong việc xây dựng và thiết kế giao diện web
- Phân biệt được các tính năng và cách sử dụng của các thẻ HTML cơ bản
- Nhận được các thuộc tính cơ bản của CSS
- Áp dụng các thuộc tính CSS cơ bản để định dạng giao diện web
- Sử dụng HTML và CSS để tạo cấu trúc và định dạng một trang web đơn giản
- Sử dụng các công cụ phát triển như trình duyệt web hoặc Developer Tools để kiểm tra, thử nghiệm và điều chỉnh giao diện web
- Tạo được bố cục giao diện web : bố cục một cột, hai cột, hoặc ba cột, sử dụng các kỹ thuật layout như flexbox hoặc grid.
- Khuyến khích tư duy sáng tạo trong việc thiết kế giao diện web
- Phát triển tính kiên nhẫn và tỉ mỉ trong việc viết mã và kiểm tra lỗi

ĐỒ DÙNG VÀ TRANG THIẾT BỊ DẠY HỌC

- Đồ dùng, dụng cụ : Tài liệu , giáo trình, bảng , bút lông,
- Thiết bị : Tivi, Máy tính, phần mềm soạn thảo code, slide bài giảng

HÌNH THỨC TỔ CHỨC DẠY HỌC

- Giới thiệu chủ đề: Cả lớp
- Giải quyết vấn đề: Cá nhân, Cả lớp
- Kết thúc vấn đề: Cả lớp

I. ÔN ĐỊNH LỚP HỌC:

Thời gian: **5 phút**

- Điểm danh
- Nhận xét lớp
- Nhắc nhở

II. THỰC HIỆN BÀI HỌC.

TT	NỘI DUNG	HOẠT ĐỘNG DẠY HỌC		THỜI GIAN
		HOẠT ĐỘNG CỦA GIÁO VIÊN	HOẠT ĐỘNG CỦA HỌC SINH	
1	<p>Dẫn nhập</p> <ul style="list-style-type: none"> - Mở một trang web bất kỳ trên internet và đặt câu hỏi để tạo trang web đơn giản 	<ul style="list-style-type: none"> -Mở một trang web trên Internet -Đặt câu hỏi gợi sự tò mò: “Các bạn đã bao giờ tự hỏi những trang web như trên được xây dựng như thế nào không ?” - Mời ngẫu nhiên phát biểu -Nhận xét câu trả lời và dẫn vào nội dung bài học 	<ul style="list-style-type: none"> - Quan sát trang web - Nhìn trang web và suy nghĩ câu trả lời - Phát biểu câu trả lời -Lắng nghe nhận xét của giáo viên 	3 phút
2	<p>Giới thiệu chủ đề</p> <ul style="list-style-type: none"> - Tên bài : Xử lý dữ liệu với DataSet 	<ul style="list-style-type: none"> -Trình chiếu slide bài học và trình bày mục tiêu bài 	<ul style="list-style-type: none"> -Quan sát và chuẩn bị tâm thế 	2 phút

	<p>-Mục tiêu :</p> <ul style="list-style-type: none"> + Kiến thức + Kỹ năng + Thái độ <p>- Nội dung :</p> <ul style="list-style-type: none"> +Giới thiệu HTML và CSS +Làm quen với HTML +Làm quen với CSS +Kết hợp HTML và CSS 	<p>học</p> <p>-Giới thiệu sơ lược về nội dung bài học</p>	<p>vào bài học</p> <p>- Chú ý theo dõi</p>	
3	<p><u>Giải quyết vấn đề</u></p> <p>1 .Lý thuyết liên quan</p> <ul style="list-style-type: none"> - Ngôn ngữ HTML và CSS - Cấu trúc và các thẻ thông dụng của HTML - Cú pháp và thuộc tính cơ bản của CSS 	<ul style="list-style-type: none"> - Trình chiếu nội dung và ghi đề mục - Giới thiệu tổng quan về ngôn ngữ HTML - Đặt câu hỏi : “Các bạn hãy đọc tài liệu và cho thày biết HTML được viết tắt của từ gì?” - Mời một bạn phát biểu - Nhận xét câu trả lời và tiếp tục trình bày các phần sau - Tiếp tục chuyển slide và giới thiệu sơ về CSS - Trình chiếu Slide đến phần làm quen với HTML và ghi đề mục 	<ul style="list-style-type: none"> - Quan sát trình chiếu và ghi chép - Tập chung lắng nghe và ghi chép -Đọc tài liệu và tìm kiếm câu trả lời -Trả lời câu hỏi - Lắng nghe và ghi nhận - Theo dõi và ghi chép cẩn thận - Quan sát và ghi chép - Chăm chú lắng 	120 phút 20 phút

	<ul style="list-style-type: none"> - Giới thiệu cấu trúc cơ bản của HTML và cho ví dụ minh họa - Trình bày một số thẻ HTML thông dụng - Chuyển Slide đến phần làm quen với CSS - Trình bày các cú pháp và thuộc tính cơ bản của CSS <p>2. Quy trình thực hiện</p> <p>Bước 1 : Tạo và xác định cấu trúc HTML</p> <p>Bước 2 : Viết mã HTML</p> <p>Bước 3 : Tạo và liên kết tệp CSS</p> <p>Bước 4 : Viết mã CSS</p> <p>Bước 5 : Kiểm tra trang web</p> <p>3. Thao tác mỗ</p>	<ul style="list-style-type: none"> nghe và quan sát - Ghi chép cẩn thận và chú ý lắng nghe - Ghi chú cẩn thận - Nghiêm túc theo dõi và ghi chép đầy đủ <ul style="list-style-type: none"> - Chăm chú theo dõi - Chú ý theo dõi và ghi chép - Nghiêm túc lắng nghe và ghi chép 	5 phút
	<ul style="list-style-type: none"> - Trình chiếu màn hình thực hiện thao tác - Mở phần mềm soạn thảo code và hướng dẫn tạo file có đuôi .html - Hướng dẫn viết mã 	<ul style="list-style-type: none"> - Tập trung theo dõi - Chăm chú quan sát và ghi chép 	10 phút

		<p>HTML vào file vừa tạo</p> <ul style="list-style-type: none"> - Hướng dẫn tạo file có đuôi .css - Hướng dẫn thực hiện link file .css vào file .html - Hướng dẫn viết mã CSS - Hướng dẫn cách kiểm tra trang web vừa tạo - Đặt câu hỏi : “<i>Các bạn đã nắm được phần này chưa, có gì thắc mắc cứ hỏi thầy</i>” - Trả lời câu hỏi 	<ul style="list-style-type: none"> - Quan sát, lắng nghe - Tập trung quan sát, lắng nghe - Theo dõi và chăm chú lắng nghe - Quan sát, lắng nghe - Quan sát, lắng nghe. - Đặt câu hỏi - Lắng nghe và ghi nhận 	
	<p>4. Lỗi thường gặp, nguyên nhân, cách sửa lỗi</p> <ul style="list-style-type: none"> - Các sai phạm thường gặp - Lỗi cú pháp - Đặt thẻ link không đúng chỗ hoặc sử dụng sai đường dẫn khiến cho HTML không nhận CSS 	<ul style="list-style-type: none"> - Nêu các lỗi thường gặp của người học trong quá trình thực hiện. - Thực hiện trình chiếu, hướng dẫn cách đặt thẻ link và cách lấy đường dẫn 	<ul style="list-style-type: none"> - Ghi chú, theo dõi, ghi nhớ. - Theo dõi và ghi nhận 	5 phút
	<p>5. Thao tác thử</p>	<ul style="list-style-type: none"> - Yêu cầu người học thực hiện lại các bước - Mời ngẫu nhiên 5 bạn 	<ul style="list-style-type: none"> - Ghi nhớ lại quy trình và xung phong 	8 phút

	<p>6. Phân công luyện tập</p> <p>7. Thực hành luyện tập</p>	<ul style="list-style-type: none"> - Nhận xét cách người học thực hiện - Phân công mỗi cá nhân thực hiện luyện tập trên một máy tính - Giao bài tập thực hành cho người học 	<ul style="list-style-type: none"> - Tiến hành các thao tác trong quy trình - Chú ý lắng nghe, ghi chú nhận xét. - Lắng nghe, thực hành theo yêu cầu. - Tiến hành làm bài tập. 	1 phút 75 phút
4	<p>Kết thúc vấn đề</p> <ul style="list-style-type: none"> - Củng cố kỹ năng rèn luyện - Nhận xét, đánh giá kết quả học tập + Kết quả học tập + Lỗi sai phạm 	<ul style="list-style-type: none"> - Nhận xét quá trình thực hành của lớp - Mời ngẫu nhiên các bạn thực hiện làm câu hỏi trắc nghiệm - Nhận xét và nêu sai sót chung - Khuyến nghị học sinh hoàn thiện trong các buổi học sau. 	<ul style="list-style-type: none"> -Chú ý lắng nghe -Thực làm trắc nghiệm yêu cầu của giáo viên -Tập trung lắng nghe và ghi chép -Lắng nghe 	15 phút
5	<p>Hướng dẫn tư học</p> <ul style="list-style-type: none"> - Bài tập về nhà - Yêu cầu chuẩn bị cho bài học sau 	<ul style="list-style-type: none"> - Yêu cầu người học thực hiện các ví dụ, bài tập mở rộng trong tài liệu học tập. - Yêu cầu người học hoàn 	<ul style="list-style-type: none"> - Lắng nghe, ghi nhận - Ghi nhận 	5 phút

		<p>thành các bài tập về nhà và nộp trước bài học tiếp theo</p> <ul style="list-style-type: none"> - Yêu cầu người học đọc trước về ngôn ngữ JavaScrip 	<p>- Lắng nghe</p>	
--	--	--	--------------------	--

IV. Rút kinh nghiệm tổ chức thực hiện:

.....

Trưởng khoa/ trưởng tổ môn

Ngày tháng năm
Giáo viên

Trọng
Nguyễn Xuân Trọng

4.5 Phiếu hướng dẫn thực hiện

Mô đun :	Thiết kế và quản trị web				
Kỹ năng :	Kết hợp HTML và CSS trong giao diện Web				
TT	Bước	Tiêu chuẩn	Phương pháp	Phương tiện	Lưu ý / An toàn
1	Xác định cấu trúc cơ bản của HTML	<ul style="list-style-type: none"> - Tạo được file có đuôi .html - Đảm bảo tài liệu HTML bắt đầu với thẻ <!DOCTYPE html> và kết thúc trong thẻ </html> 	<ul style="list-style-type: none"> -Chọn “file” -> “New File” để tạo tệp mới và lưu file có đuôi html -Sử dụng trình soạn thảo code tạo tệp HTML mới và viết mã HTML theo cấu trúc đề ra. 	<ul style="list-style-type: none"> - Máy tính -Trình soạn thảo code 	<ul style="list-style-type: none"> -Kiểm tra cẩn thận xem tất cả các thẻ HTML được mở đóng đúng cách.
2	Viết Mã HTML	Mã HTML phải bao gồm tiêu đề, menu, nội dung, footer và được phân chia bằng các thẻ phù hợp	<ul style="list-style-type: none"> Sử dụng các thẻ HTML để tạo cấu trúc trang web - Gõ trực tiếp thẻ <html>, <head>, <title>, <body> vào tệp. 	<ul style="list-style-type: none"> - Máy tính -Trình soạn thảo code 	<ul style="list-style-type: none"> Kiểm tra lại xem mọi phần tử HTML đã được thêm vào và định dạng đúng
3	Tạo và Liên Kết Tệp CSS	Tạo được một tệp CSS mới và liên kết với tài liệu HTML	<ul style="list-style-type: none"> Chọn “file” -> “New File” để 	<ul style="list-style-type: none"> - Máy tính -Trình soạn thảo code 	<ul style="list-style-type: none"> Kiểm tra xem tệp CSS đã

			tạo tệp mới và lưu file có đuôi css - Tạo thẻ <link> trong phần head dẫn đến file css		được liên kết chính xác và các quy tắc định dạng đã được áp dụng đúng cách.
4	Viết Mã CSS	Mã CSS phải định dạng các phần tử HTML như tiêu đề, menu, nội dung, footer theo yêu cầu.	Sử dụng các thuộc tính CSS như color, font-size, padding, margin, background, vv. để định dạng các phần tử theo yêu cầu.	- Máy tính -Trình soạn thảo code	Kiểm tra lại mã CSS để đảm bảo tất cả các phần tử được định dạng đúng cách.
5	Kiểm Tra Trang Web	Trang web phải hiển thị đúng trên trình duyệt và áp dụng các quy tắc CSS đã định nghĩa.	Mở trình duyệt và nhập địa chỉ tệp HTML vào thanh địa chỉ. Kiểm tra xem trang web hiển thị đúng và các quy tắc CSS đã được áp dụng đúng cách.	- Máy tính -Trình soạn thảo code -Trình duyệt	Đảm bảo kiểm tra kỹ lưỡng trước khi hoàn thành.

4.6 Bộ câu hỏi trắc nghiệm đa phương án

Câu 1:

HTML là gì?

- a) Một ngôn ngữ lập trình.
- b) Một ngôn ngữ đánh dấu.
- c) Một kiểu dữ liệu.
- d) Một loại phần mềm thiết kế web.

Câu 2:

Ai là người tạo ra HTML?

- a) Brendan Eich
- b) Tim Berners-Lee
- c) Bill Gates
- d) Mark Zuckerberg

Câu 3:

Thẻ nào sau đây không phải là thẻ HTML tiêu đề?

- a) <h1>
- b) <h3>
- c) <h5>

d) <h7>

Câu 4:

Để tạo một liên kết trong HTML, bạn sử dụng thẻ nào?

a) <link>

b) <a>

c) <href>

d) <connect>

Câu 5:

CSS là viết tắt của từ gì?

a) Computer Style Sheets

b) Creative Style Sheets

c) Cascading Style Sheets

d) Colorful Style Sheets

Câu 6:

Cách nào sau đây là cách đúng để chèn CSS nội tuyến vào một thẻ HTML?

a) <p css="color: red;">

b) <p class="color: red;">

c) <p style="color: red;">

d) <p color="red;">

Câu 7:

Thuộc tính nào sau đây không phải là thuộc tính CSS?

a) color

b) font-size

- c) margin
- d) hyperlink

Câu 8:

Để định dạng chữ in đậm trong HTML, bạn sử dụng thẻ nào?

- a) <i>
- b) <u>
- c)
- d)

Câu 9:

Để tạo danh sách không có thứ tự trong HTML, bạn sử dụng thẻ nào?

- a)
- b)
- c)
- d) <dl>

Câu 10:

Thẻ nào dùng để chứa toàn bộ nội dung hiển thị của trang web trong HTML?

- a) <html>
- b) <head>
- c) <title>
- d) <body>

4.7 Phiếu đánh giá quy trình

Khóa học	Thiết kế website đơn giản 2024-2025			
Kỹ năng	Kết hợp HTML và CSS trong giao diện Web			
Học viên:	Ngày.....tháng..... năm....			
Hướng dẫn: <i>Dánh dấu ✓ vào những bước mà học viên đó thực hiện VÀ đảm bảo tiêu chuẩn</i>				
TT	Bước	Tiêu chuẩn	Lưu ý an toàn lao động	✓
1	Xác định cấu trúc cơ bản của HTML	Tạo được file có đuôi .html - Đảm bảo tài liệu HTML bắt đầu với thẻ <!DOCTYPE html> và kết thúc trong thẻ </html>	Kiểm tra cẩn thận xem tất cả các thẻ HTML được mở đóng đúng cách.	
2	Viết Mã HTML	Mã HTML phải bao gồm tiêu đề, menu, nội dung, footer và được phân chia bằng các thẻ phù hợp	Kiểm tra lại xem mọi phần tử HTML đã được thêm vào và định dạng đúng	
3	Tạo và Liên Kết Tệp CSS	Tạo một tệp CSS mới và liên kết với tài liệu HTML bằng thẻ <link> trong phần <head>	Kiểm tra xem tệp CSS đã được liên kết chính xác và các quy tắc định dạng đã được áp dụng đúng cách.	

4	Viết Mã CSS	-Mã CSS phải định dạng các phần tử HTML như tiêu đề, menu, nội dung, footer theo yêu cầu	-Kiểm tra lại mã CSS để đảm bảo tất cả các phần tử được định dạng đúng cách.	
5	Kiểm Tra Trang Web	Trang web phải hiển thị đúng trên trình duyệt và áp dụng các quy tắc CSS đã định nghĩa	Đảm bảo kiểm tra kỹ lưỡng trước khi hoàn thành.	

4.8 Phiếu đánh giá sản phẩm

PHIẾU ĐÁNH GIÁ SẢN PHẨM

Tên sản phẩm: Web đã định dạng

Mã số:

Tên học sinh: Ngày:

TT	Tiêu chuẩn	Bằng chứng	Đánh giá	
			Đạt	Không đạt
1	Kỹ thuật	- Tiêu chuẩn 1: : Tạo file HTML,CSS đúng định dạng.		
		- Tiêu chuẩn 2 Cấu trúc HTML,CSS hợp lý và đầy đủ.		
2	Thẩm mỹ	- Tiêu chuẩn 1: Cấu trúc HTML hợp lý và		

		đầy đủ.			
		- Tiêu chuẩn 2 : Trang web có bố cục hợp lý			
3	An toàn	Không có			
4	Thời gian	- Tiêu chuẩn : thực hiện trong khoảng thời gian quy định			