

Lập trình web

# LẬP TRÌNH ASP.NET MVC P3

---

GV: Nguyễn Huy Cường

Email: [nh.cuong@hutech.edu.vn](mailto:nh.cuong@hutech.edu.vn)

# Nội dung

## 1. Tương tác với CSDL SQL Server

 Tổng quan

 Entity Framework Core, Entity Framework Core Tools

 Truy vấn, thêm, xóa, sửa

## 2. Kiểm lỗi trên ASP.NET MVC

 Tổng quan

 Kiểm lỗi ở server

 Kiểm lỗi ở client

# TƯƠNG TÁC VỚI SQL SERVER

---

 Code First

 DB First

# Tổng quan

- OLE DB (1990 ~2000)
- **ADO.NET** (2002)
- NHibernate (2004), Dapper (2011)
- LINQ TO SQL (2007 )
- ADO.NET Entity Framework (2008)
- **Entity Framework Core**

## Entity Framework Core 8.0

```
using (var context = new SchoolContext())
{
    // Retrieve a list of students from the db
    var students = context.Students.ToList();
    // //
}
```

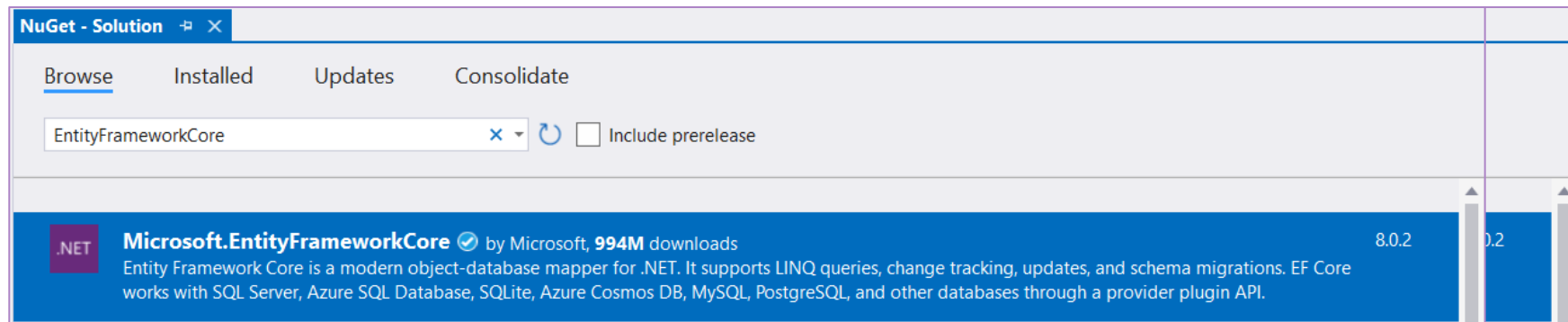
## ADO.NET

```
string connectionString = "Data Source=myServerAddress;Initial Catalog=myDataBase;User
Id=myUsername;Password=myPassword;";
// Create a SqlConnection object to connect to the database
using (SqlConnection connection = new SqlConnection(connectionString))
{
    try
    {
        // Open the database connection
        connection.Open();
        // Create a SqlCommand object to execute a SQL command
        string sql = "SELECT StudentID, Name, ClassName, AverageScore FROM Students";
        SqlCommand command = new SqlCommand(sql, connection);

        // Execute the SQL command and read the results
        SqlDataReader reader = command.ExecuteReader();
        List<Student> students = new List<Student>();
        while (reader.Read())
        {
            int studentId = reader.GetInt32(0);
            string name = reader.GetString(1);
            string className = reader.GetString(2);
            double averageScore = reader.GetDouble(3);
            Student student = new Student(studentId, name, className, averageScore);
            students.Add(student);
        }
        reader.Close();
        // Print the list of students
    }
    catch (Exception ex)
    {
        //Console.WriteLine("Error: " + ex.Message);
    }
}
```

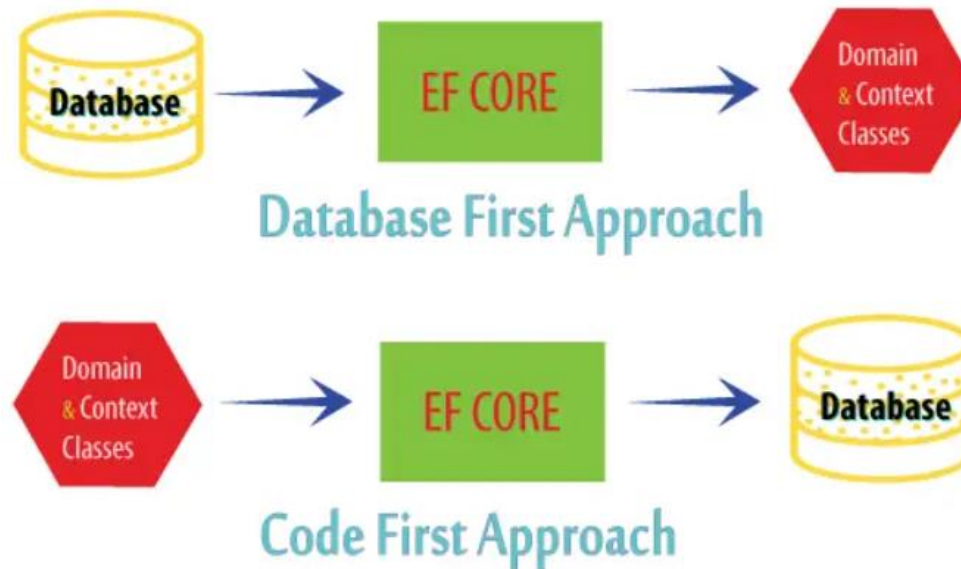
# Entity Framework Core

- EF Core là thư viện cung cấp khả năng **ánh xạ** đối tượng lập trình với các đối tượng CSDL (ORM): tạo DB, đọc, ghi, xóa dữ liệu, sử dụng Linq để thực hiện các truy vấn.
- EF sử dụng “**Context**”. **Context** là một lớp hoạt động như cầu nối giữa CSDL và ứng dụng. Nó cung cấp một bộ API để thực hiện các thao tác CRUD trên các đối tượng, quản lý transaction và truy vấn CSDL.
- Cài đặt EF từ Nuget hoặc từ Package Console



## Entity Framework Core hỗ trợ các CSDL nào?

- **Entity framework core** hỗ trợ cho phép làm việc với nhiều loại database và ta phải install các **database provider** qua *NuGet packages*.
  - ❑ **SQL Server** ([Microsoft.EntityFrameworkCore.SqlServer](#))
  - ❑ **MySQL** ([MySql.Data.EntityFrameworkCore](#))
  - ❑ **PostgreSQL, Oracle**
  - ❑ **SQLite (mobile), DB2, Firebird, Jet (Microsoft Access), Azure Cosmos DB**
  - ❑ **In-memory (for testing)** ([Microsoft.EntityFrameworkCore.InMemory](#))
- Các hướng tiếp cận



## Tương tác với SQL Server

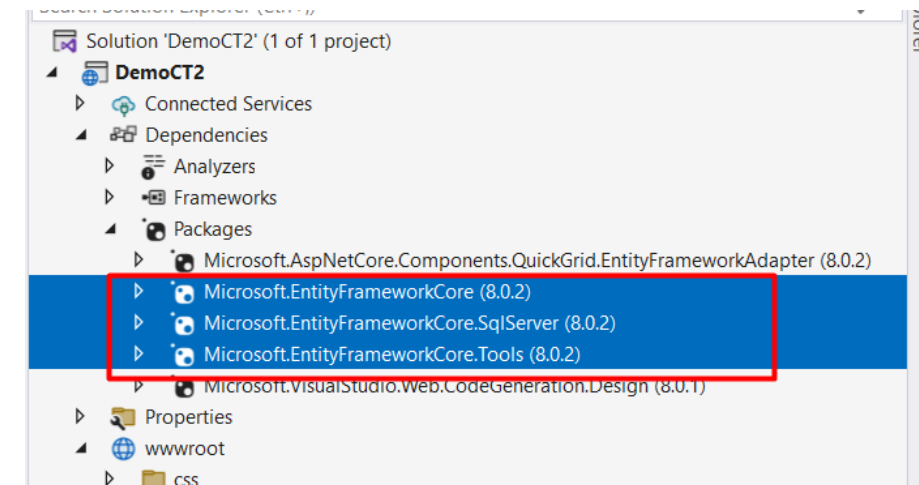
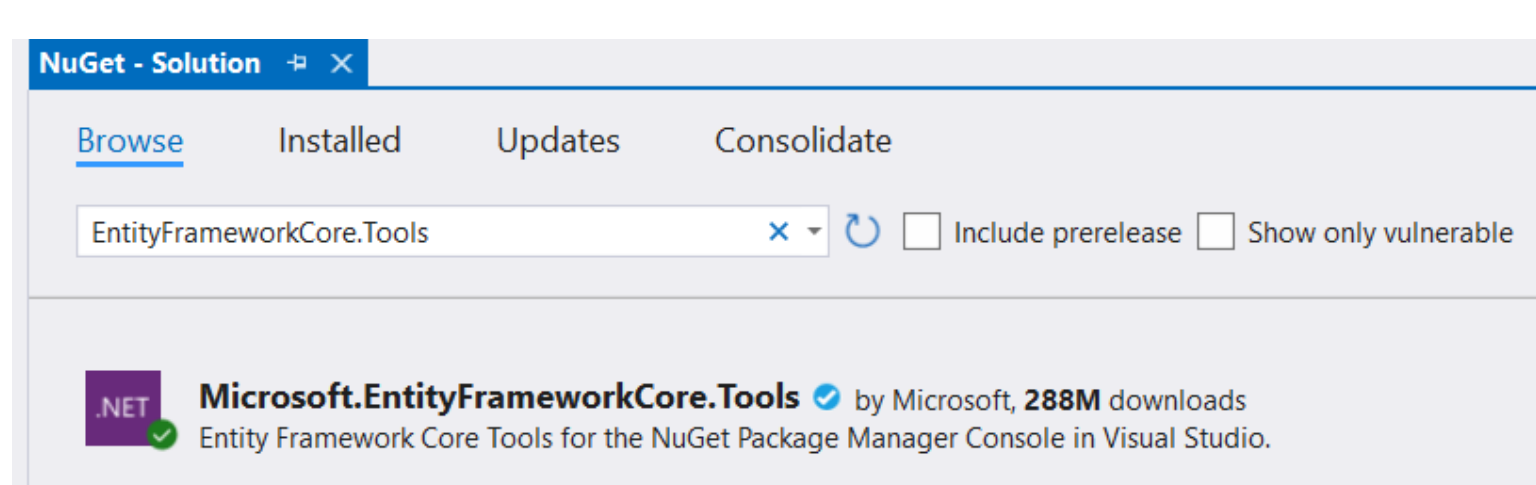
- Cài đặt EF Core, EF Core Tools, và EF Core.SqlServer

- ☐ **EF Core**

- ☐ **EF Core DB provider** (Microsoft.EntityFrameworkCore.SqlServer)

- ☐ **EF Core tools** (<https://microsoft.entityframeworkcore.tools/>) để sử dụng trong PMC (nếu không thì dùng .NET CLI)

EF Core Tools hỗ trợ 1 số câu lệnh: Add-Migration, Get-Migration, Remove-Migration, Update-Database, Scaffold-DbContext...





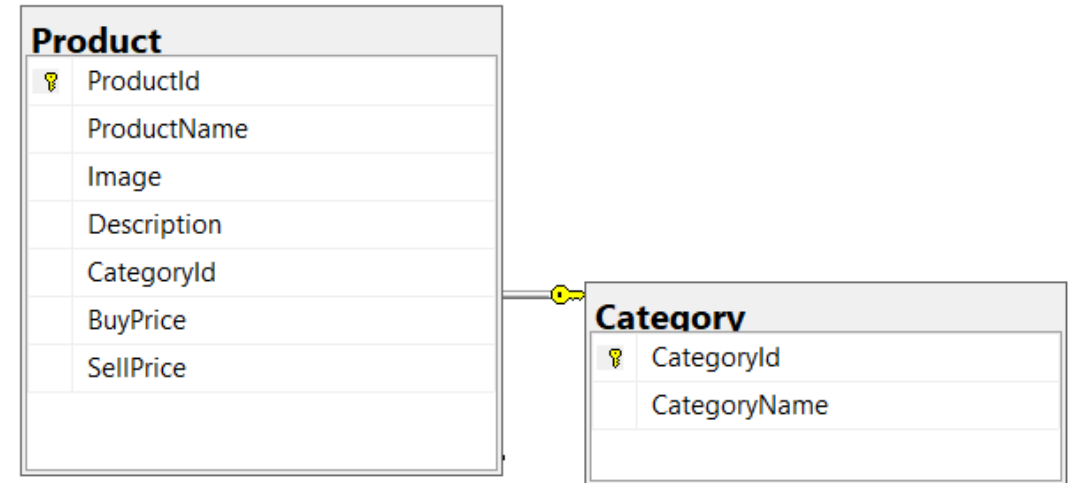
# 1. Sử dụng Db-First



(o) Tạo trước **CSDL** Product gồm 2 bảng **Product**, **Category**

| Column Name | Data Type      | Allow Nulls                         |
|-------------|----------------|-------------------------------------|
| ProductId   | nvarchar(10)   | <input type="checkbox"/>            |
| ProductName | nvarchar(50)   | <input type="checkbox"/>            |
| Image       | nvarchar(100)  | <input checked="" type="checkbox"/> |
| Description | nvarchar(255)  | <input checked="" type="checkbox"/> |
| CategoryId  | int            | <input type="checkbox"/>            |
| BuyPrice    | decimal(18, 0) | <input type="checkbox"/>            |
| SellPrice   | decimal(18, 0) | <input type="checkbox"/>            |

| Column Name  | Data Type     | Allow Nulls              |
|--------------|---------------|--------------------------|
| CategoryId   | int           | <input type="checkbox"/> |
| CategoryName | nvarchar(250) | <input type="checkbox"/> |

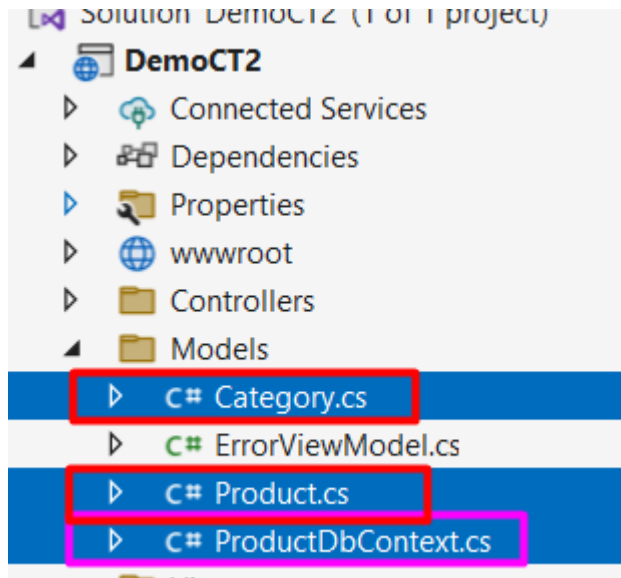


(1) Tạo kết nối **Scaffold-DbContext** [-Connection] [-Provider] [-OutputDir]

**Scaffold-DbContext** "Data Source= **HUYCUONG\SQLEXPRESS**;Initial Catalog= **ProductDB**;User ID=**sa**;Password=**123456**;Encrypt=False;TrustServerCertificate=False;" **Microsoft.EntityFrameworkCore.SqlServer** -OutputDir **Models**



# Kết quả sử dụng DB-First



(2) Điều chỉnh `OnConfiguring` ở `ProductDbContext` đưa vào `Program.cs`

```
builder.Services.AddDbContext<ProductDbContext>(options =>options.UseSqlServer("Data  
Source= HUYCUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User  
ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"));
```

(3) [Optional] Tạo câu kết nối ở `Appsetting.json` thay vì hard-code trong `program.cs`

# Tạo Productscontroller ( sử dụng template EF)

Program.cs\*

```

7  using System.Configuration;
8
9  var builder = WebApplication.CreateBuilder(args);
10
11  // Add services to the container.
12  builder.Services.AddControllersWithViews();
13  builder.Services.AddDbContext<ProductDbContext>(options =>options.UseSqlServer(builder.Configuration.GetConnectionString("ProductDatabase")));
14

```

appsettings.json

Schema: https://json.schemastore.org/appsettings.json

```

1  {
2    "ConnectionStrings": {
3      "ProductDatabase": "Data Source= HUYSUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"
4    },

```

## (4) Thêm Products Controller từ template



Add MVC Controller with views, using Entity Framework

Model class: Product (DemoCT2.Models)

DbContext class: ProductDbContext (DemoCT2.Models)

Database provider: Configured from the selected DbContext

Views

- ☒ Generate views
- ☐ Reference script libraries
- ☒ Use a layout page

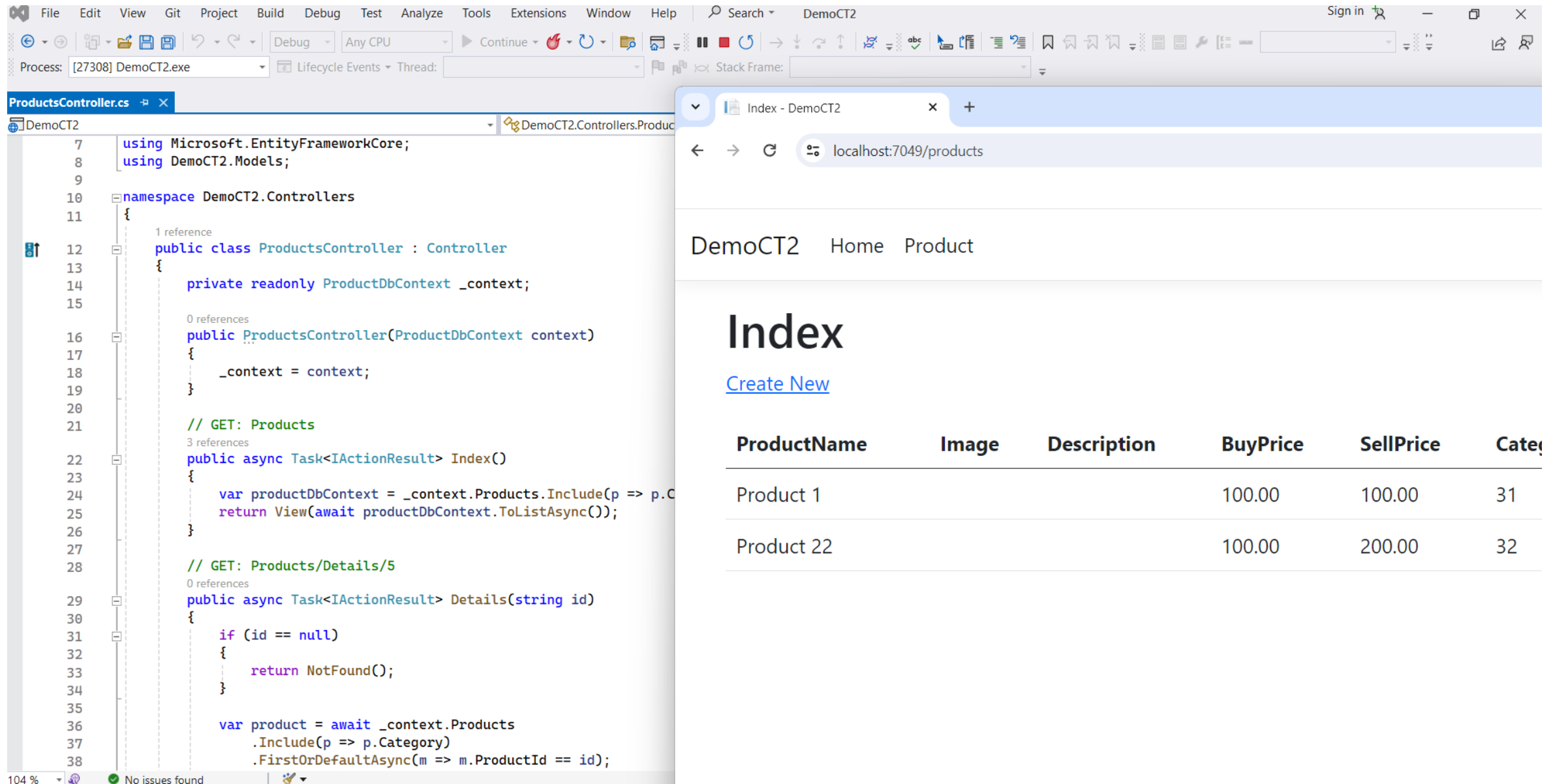
~/Views/Shared/\_Layout.cshtml

(Leave empty if it is set in a Razor \_viewstart file)

Controller name: ProductsController

Add Cancel

# Kết quả



The screenshot displays the Visual Studio IDE on the left and a web browser on the right. The Visual Studio window shows the code for `ProductsController.cs` in the `DemoCT2` project. The code defines a `ProductsController` class that inherits from `Controller`. It includes a `private readonly ProductDbContext _context;` and a constructor that initializes the context. The `Index()` method is an asynchronous action that returns a view with the products from the database. The `Details(string id)` method is also an asynchronous action that returns a view with the details of a specific product.

The web browser window shows the `Index` page of the `DemoCT2` application. The page has a navigation bar with links to `DemoCT2`, `Home`, and `Product`. Below the navigation bar is a heading `Index` and a link `Create New`. The main content area contains a table with the following data:

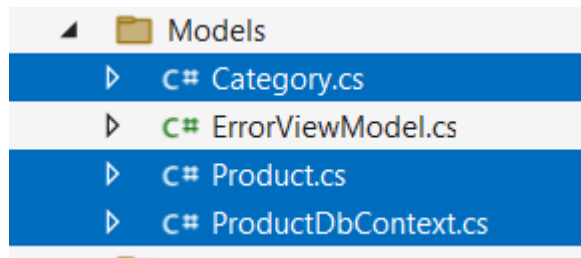
| ProductName | Image | Description | BuyPrice | SellPrice | Cate |
|-------------|-------|-------------|----------|-----------|------|
| Product 1   |       |             | 100.00   | 100.00    | 31   |
| Product 22  |       |             | 100.00   | 200.00    | 32   |



## 2. Sử dụng Code-first



### (1) Tạo **models** và lớp **DbContext**



```

public class Product
{
    public required string ProductId { get; set; }
    public required string ProductName { get; set; }
    public string? Image { get; set; }
    public string? Description { get; set; }
    public int CategoryId { get; set; }
    public decimal BuyPrice { get; set; }
    public decimal SellPrice { get; set; }
    public virtual required Category Category { get; set; }
}

```

```

public partial class Category
{
    public int CategoryId { get; set; }

    public string CategoryName { get; set; } = null!;

    public virtual ICollection<Product>? Products { get; set; }
}

```

```

public class ProductDbContext : DbContext
{
    public ProductDbContext(DbContextOptions<ProductDbContext> options)
        : base(options)
    {
    }

    public virtual DbSet<Category> Categories { get; set; }
    public virtual DbSet<Product> Products { get; set; }
}

```

# Cấu hình kết nối

## (2) Bổ sung chuỗi kết nối & sử dụng SQLserver

### Appsetting.json

```
"ConnectionStrings": {
  "ProductDatabase": "Data Source= HUVCUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"
},
```

### Program.cs

```
builder.Services.AddDbContext<ProductDbContext>(options
=>options.UseSqlServer(builder.Configuration.GetConnectionString("ProductDatabase")));
```

appsettings.json

Schema: <https://json.schemastore.org/appsettings.json>

```
1 {
2   "ConnectionStrings": {
3     "ProductDatabase": "Data Source= HUVCUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"
4   },
5   "Logging": {
6     "LogLevel": {
7       "Default": "Information",
8       "Microsoft.AspNetCore": "Warning"
9     }
10  },
11  "AllowedHosts": "*"
12 }
```

Program.cs

```
1
2 using DemoCT2.Models;
3 using Microsoft.Build.Evaluation;
4 using Microsoft.EntityFrameworkCore;
5 using Microsoft.Extensions.Configuration;
6 using Microsoft.Extensions.Options;
7 using System.Configuration;
8
9 var builder = WebApplication.CreateBuilder(args);
10
11 // Add services to the container.
12 builder.Services.AddControllersWithViews();
13 builder.Services.AddDbContext<ProductDbContext>(options =>options.UseSqlServer(builder.Configuration.GetConnectionString("ProductDatabase")));
```

# Thực hiện ánh xạ CSDL

## (2) Bổ sung chuỗi kết nối & sử dụng SQLserver

### Appsetting.json

```
"ConnectionStrings": {
  "ProductDatabase": "Data Source= HUVCUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"
},
```

### Program.cs

```
builder.Services.AddDbContext<ProductDbContext>(options
=>options.UseSqlServer(builder.Configuration.GetConnectionString("ProductDatabase")));
```

appsettings.json

Schema: <https://json.schemastore.org/appsettings.json>

```
1 {
2   "ConnectionStrings": {
3     "ProductDatabase": "Data Source= HUVCUONG\\SQLEXPRESS;Initial Catalog= ProductDB;User ID=sa;Password=123456;Encrypt=False;TrustServerCertificate=False;"
4   },
5   "Logging": {
6     "LogLevel": {
7       "Default": "Information",
8       "Microsoft.AspNetCore": "Warning"
9     }
10  },
11  "AllowedHosts": "*"
12 }
```

Program.cs

```
1
2 using DemoCT2.Models;
3 using Microsoft.Build.Evaluation;
4 using Microsoft.EntityFrameworkCore;
5 using Microsoft.Extensions.Configuration;
6 using Microsoft.Extensions.Options;
7 using System.Configuration;
8
9 var builder = WebApplication.CreateBuilder(args);
10
11 // Add services to the container.
12 builder.Services.AddControllersWithViews();
13 builder.Services.AddDbContext<ProductDbContext>(options =>options.UseSqlServer(builder.Configuration.GetConnectionString("ProductDatabase")));
```

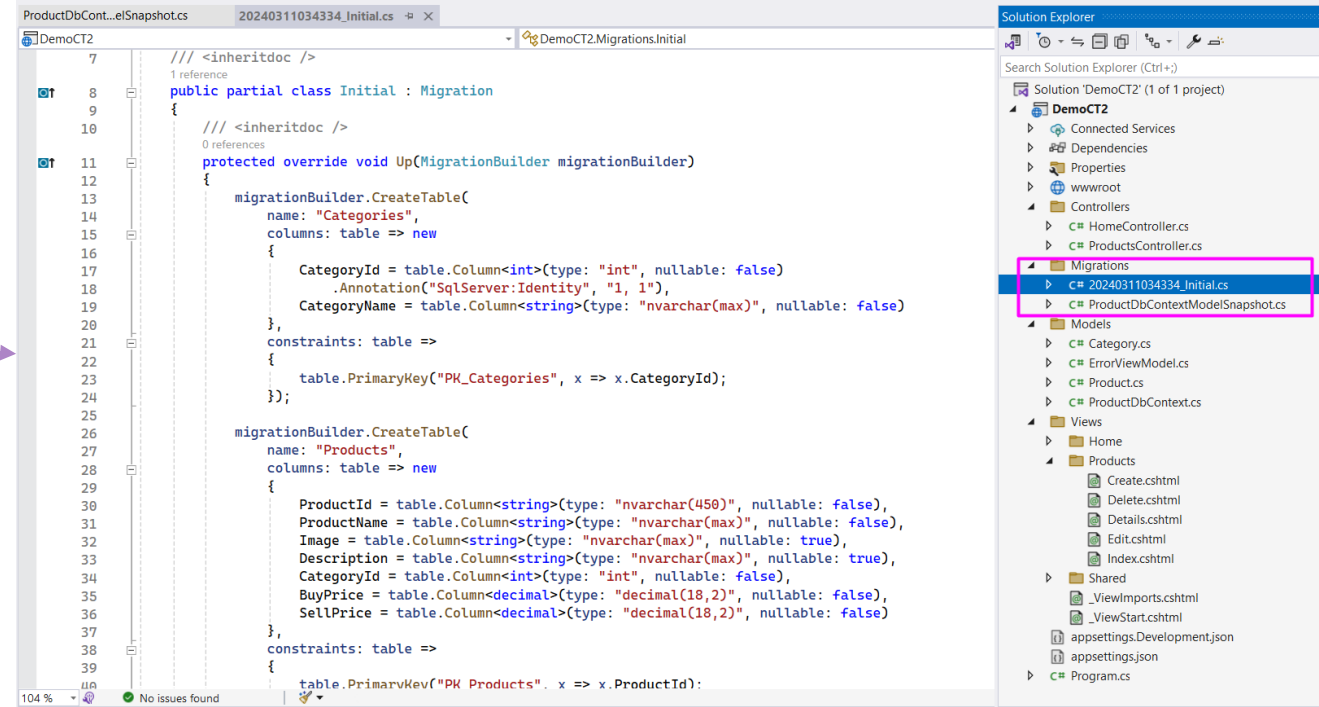
# Khởi tạo và cập nhật CSDL (Add-Migration, Update-Database)

## (3) Chạy PCM (package console manager)

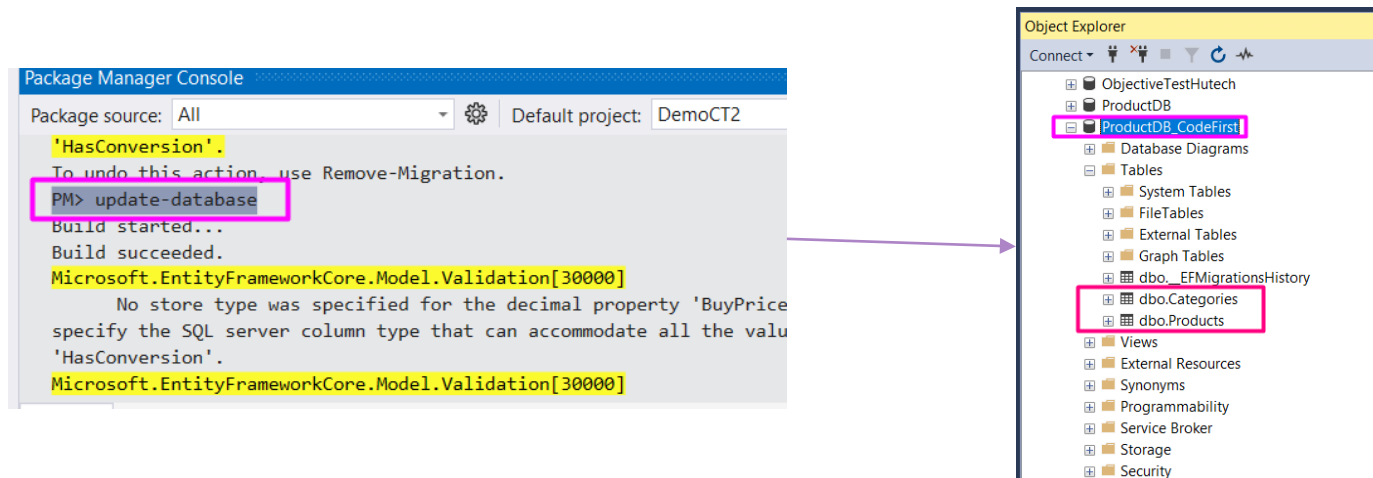
### Add-Migration Initial

```
Package Manager Console
Package source: All Default project: DemoCT2

To protect potentially sensitive information in your connection string,
configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For
PM> Add-Migration Initial
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Model.Validation[30000]
No store type was specified for the decimal property 'BuyPrice' or
```



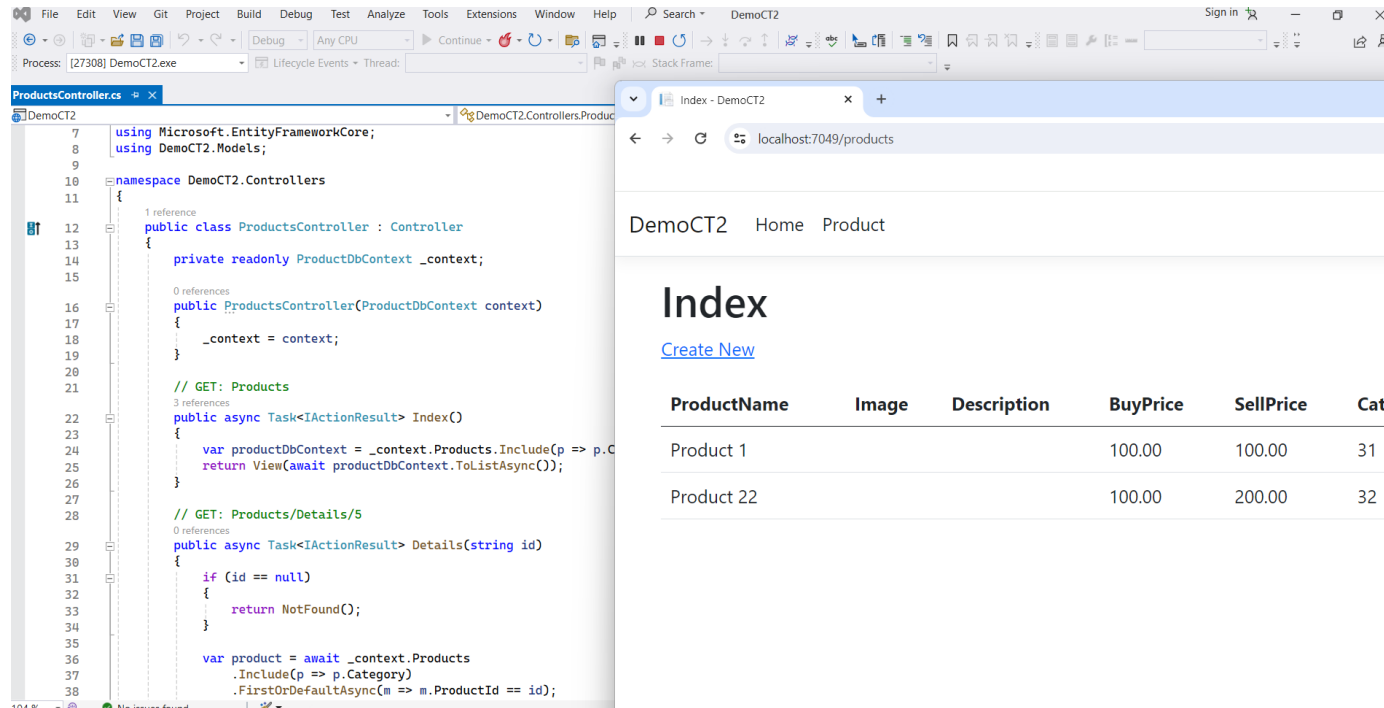
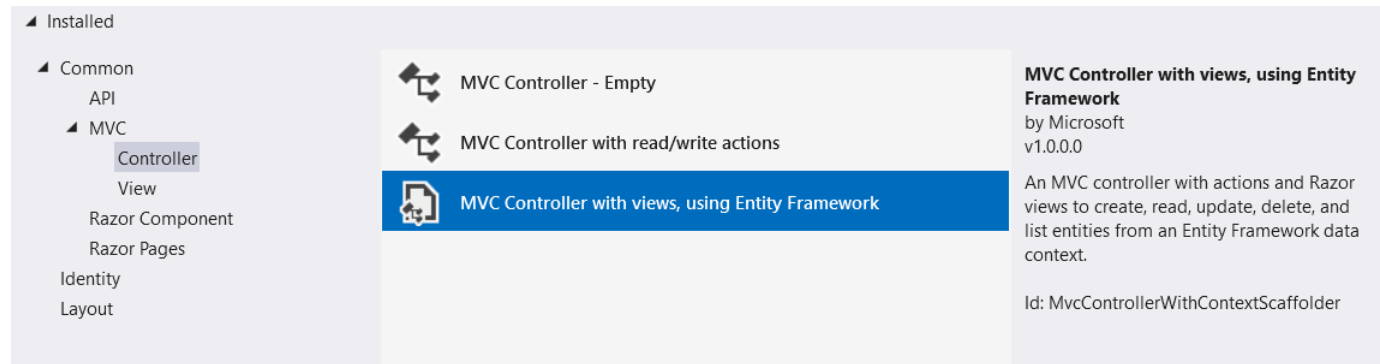
### Update-Database





# Tạo Productscontroller ( sử dụng template EF) & KQ

## (4) Thêm Products Controller từ template



Add MVC Controller with views, using Entity Framework

Model class: Product (DemoCT2.Models)

DbContext class: ProductDbContext (DemoCT2.Models) +

Database provider: Configured from the selected DbContext

Views

☒ Generate views

☐ Reference script libraries

☒ Use a layout page

~/Views/Shared/\_Layout.cshtml

(Leave empty if it is set in a Razor \_viewstart file)

Controller name: ProductsController

Add Cancel



## LINQ to Entities

- Thực hiện truy vấn dữ liệu sử dụng LINQ to Entities như: Chọn (**select**), Lọc (**where**), tìm kiếm (find, First, Last), sắp xếp (**orderby**, **descending**), gom nhóm (**group by**)...
- Sử dụng đối tượng Context

```
private readonly ProductDbContext _context;

public ProductsController(ProductDbContext context) {
    _context = context;
}
```

- Có 2 cách viết LINQ: Cú pháp phương thức / cú pháp truy vấn

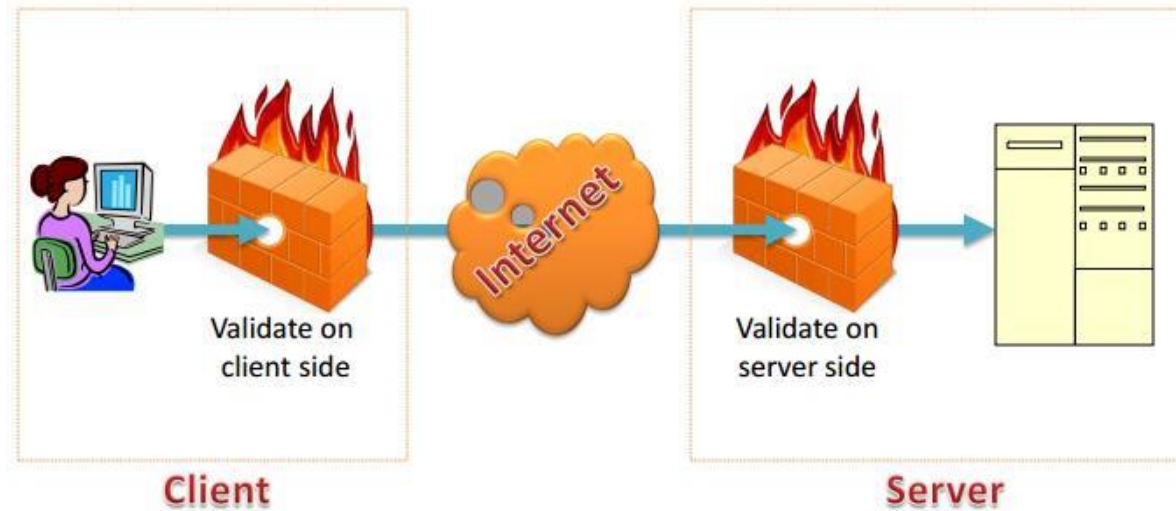
| Ví dụ   | Cú pháp phương thức   | Cú pháp truy vấn   |
|---|---|--|
| Lấy danh sách sản phẩm (không lấy thông tin category) | <pre>var list = _context.Products.ToList();</pre>                           | <pre>var list = (from p in _context.Products             select p).ToList();</pre>   |
| Lấy danh sách sản phẩm (có thông tin category)        | <pre>var list = _context.Products.Include(p=&gt;p.Category).ToList();</pre> | <pre>var list = (from p in _context.Products             join c in _context.Categories on             p.CategoryId equals c.CategoryId             select new { Product = p, Category = c             }).ToList();</pre> |

# LINQ to Entities

| Ví dụ   | Cú pháp phương thức  | Cú pháp truy vấn   |
|---|--|--|
| Lấy các sản phẩm có loại 1 và được sắp xếp tăng dần theo Price  | <code>var listP = _context.Products.Where(p =&gt; p.CategoryId == 1).OrderBy(p =&gt; p.SellPrice).ToList();</code> | <code>var listP = from c in _context.Products where c.CategoryId == 1 orderby c.SellPrice select c;</code>               |
| Tìm sản phẩm đầu tiên có mã <b>id</b>                           | <code>var product = await _context.Products.FirstOrDefault(m =&gt; m.ProductId == <b>id</b>);</code>               | <code>var product = await (from m in _context.Products where m.ProductId == <b>id</b> select m).FirstOrDefault();</code> |
| Sắp xếp các sản phẩm theo giá giảm dần                          | ?  | ?  |
| Lấy top 10 sản phẩm có giá bán lớn nhất                         | ?  | ?  |
| Lấy mỗi sản phẩm thuộc 1 nhóm(category) mà có giá bán thấp nhất | ?  | ?  |

# KIỂM LỖI TRÊN ASP.NET CORE MVC

---



# Tổng quan

- Dữ liệu nhập không hợp lệ từ client sẽ gây các lỗi khó lường. Vì vậy cần kiểm soát dữ liệu  
Các lỗi thường gặp:
  - ✓ Để trống ô nhập...
  - ✓ Không đúng định dạng email, creditcard, url...
  - ✓ Sai kiểu số nguyên, số thực, ngày giờ...
  - ✓ Không hợp lệ - phải có giá trị tối thiểu, tối đa, trong phạm vi...
- Kiểm lỗi ở Core MVC sẽ được thực hiện: ở cả 2 phía **client** và **server**  
**Với MVC chỉ cần viết 1 lần nhưng kiểm tra cả 2 phía** là client và server . Nếu vì một lý do nào đó mà client không thực hiện được thì đã có server thay thế.

# Kiểm lỗi trong Core MVC

❑ **Model**: Thêm các **Validation attributes** kiểm lỗi ở **Property** như:

[Required], [StringLength], [Range][EmailAddress], [Phone], [RegularExpression], [DataType]...

<https://learn.microsoft.com/en-us/aspnet/core/mvc/models/validation?view=aspnetcore-8.0>

❑ **View**: Sử dụng css hoặc js

```
<div asp-validation-summary="All" class="text-danger"></div>
```

```
<span asp-validation-for="ProductName" class="text-danger"></span>
```

...

❑ **Controller** : theo trạng thái model (**ModelState**)

ModelState.IsValid

ModelState.AddModelError

```
using Microsoft.AspNetCore.Mvc;

public async Task<IActionResult> Create(Product product)
{
    if (ModelState.IsValid)
    {
        _context.Add(product);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.Categories, "CategoryId", "CategoryId", product.CategoryId);
    return View(product);
}
```

## Ví dụ: kiểm lỗi trên product

Y/C để kiểm lỗi cho việc thêm mới 1 **sản phẩm**:

- ✓ Mã sản phẩm: có đúng 6 kí tự, không được để trống
- ✓ Tên sản phẩm: không được để trống, không quá 50 kí tự
- ✓ Giá của sản phẩm: có thể để trống, hoặc có giá trị từ [1-999999]

**Thực hiện ở MVC ?**

ProductId

Mã sản phẩm không được để trống

ProductName

Tên sản phẩm không được để trống

Image

Description

CategoryId

BuyPrice

SellPrice

Create

ProductId

Phải có đúng 6 kí tự.

ProductName

Image

Description

CategoryId

BuyPrice

Giá trị từ 1-999999

SellPrice

Giá trị từ 1-999999

Create

## Thực hiện Validation Model

- Thêm các thẻ attribute tương ứng cần kiểm lỗi:

```
public class Product
{
    [Required(ErrorMessage = "Mã sản phẩm không được để trống")]
    [StringLength(6, MinimumLength = 6, ErrorMessage = "Phải có đúng 6 kí tự.")]
    public required string ProductId { get; set; }
    [Required(ErrorMessage = "Tên sản phẩm không được để trống")]
    [StringLength(50, ErrorMessage = "Tên sản phẩm không được quá 50 kí tự")]
    public required string ProductName { get; set; }
    public string? Image { get; set; }
    public string? Description { get; set; }
    public int CategoryId { get; set; }
    [Range(1, 999999, ErrorMessage = "Giá trị từ 1-999999")]
    public decimal? BuyPrice { get; set; }
    [Range(1, 999999, ErrorMessage = "Giá trị từ 1-999999")]
    public decimal? SellPrice { get; set; }
    public virtual required Category Category { get; set; }
}
```

## Thực hiện Validation Controller

- Kiểm tra ở server khi thực hiện submit: Sử dụng `ModelState.IsValid` để kiểm tra trạng thái model

```
[HttpPost]
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Create(Product product)
{
    if (ModelState.IsValid)
    {
        _context.Add(product);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["CategoryId"] = new SelectList(_context.Categories, "CategoryId", "CategoryId", product.CategoryId);
    return View(product);
}
```



## Thực hiện Validation

## View

- Thực hiện kiểm tra ở client

```

<form asp-action="Create">
  <div asp-validation-summary="All" class="text-danger"></div>
  <div class="form-group">
    <label asp-for="ProductId" class="control-label"></label>
    <input asp-for="ProductId" class="form-control" />
    <span asp-validation-for="ProductId" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="ProductName" class="control-label"></label>
    <input asp-for="ProductName" class="form-control" />
    <span asp-validation-for="ProductName" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Image" class="control-label"></label>
    <input asp-for="Image" class="form-control" />
    <span asp-validation-for="Image" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="Description" class="control-label"></label>
    <input asp-for="Description" class="form-control" />
    <span asp-validation-for="Description" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="CategoryId" class="control-label"></label>
    <select asp-for="CategoryId" class="form-control" asp-items="ViewBag.CategoryId"></select>
  </div>
  <div class="form-group">
    <label asp-for="BuyPrice" class="control-label"></label>
    <input asp-for="BuyPrice" class="form-control" />
    <span asp-validation-for="BuyPrice" class="text-danger"></span>
  </div>
  <div class="form-group">
    <label asp-for="SellPrice" class="control-label"></label>
    <input asp-for="SellPrice" class="form-control" />
    <span asp-validation-for="SellPrice" class="text-danger"></span>
  </div>
  <div class="form-group">

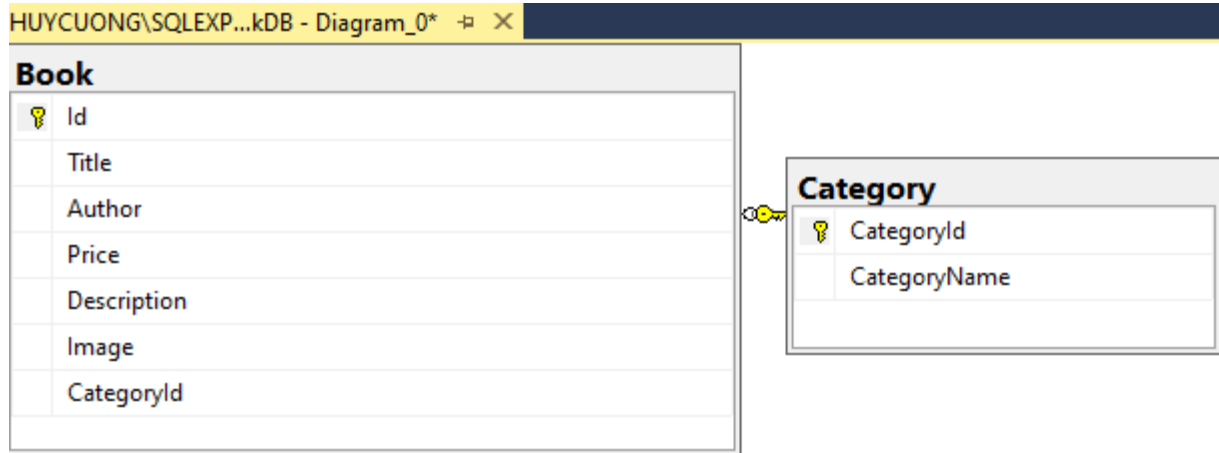
```

css cho lỗi tổng quát

css cho lỗi cụ thể

# ASM5: Viết chương trình CRUD cho BookDB

Cho cơ sở dữ liệu **BookDB** gồm 2 bảng (Book và Category)



HUYCUONG\SQLXP...okDB - dbo.Book

|    | Column Name | Data Type      | Allow Nulls                         |
|----|-------------|----------------|-------------------------------------|
| PK | Id          | int            | <input type="checkbox"/>            |
|    | Title       | nvarchar(150)  | <input type="checkbox"/>            |
|    | Author      | nvarchar(150)  | <input checked="" type="checkbox"/> |
|    | Price       | decimal(18, 0) | <input checked="" type="checkbox"/> |
|    | Description | nvarchar(MAX)  | <input checked="" type="checkbox"/> |
|    | Image       | nvarchar(100)  | <input checked="" type="checkbox"/> |
|    | CategoryId  | int            | <input checked="" type="checkbox"/> |
|    |             |                | <input type="checkbox"/>            |

HUYCUONG\SQLXP...B - dbo.Category

|   | CategoryId | CategoryName |
|---|------------|--------------|
| ▶ | 1          | Cuộc sống    |
|   | 2          | Lập trình    |
|   | 3          | Sức Khỏe     |
| * | NULL       | NULL         |

# Yêu cầu: ASM5

Trang Chủ
Đăng Ký
Đăng nhập
Giỏ hàng

**Chủ đề**


- Cuộc sống
- Lập trình
- Sức Khỏe

**Thêm mới**

**Cho tôi xin một vé đi tuổi thơ**
Edit

Tác giả: Nguyễn Nhật Ánh

Thể loại: Cuộc sống




Chi tiết
Xóa

**Lập trình C**
Edit

Tác giả: TS. Lê Xuân Việt

Thể loại: Lập trình




Chi tiết
Xóa

**Core Java: Fundamentals, Volume 1 (Oracle Press Java)**
Edit

Tác giả: Cay Horstmann

Thể loại: Lập trình




Chi tiết
Xóa

**Cuộc Sống Rất Giống Cuộc Đời**
Edit

Tác giả: Hải Dór

Thể loại: Cuộc sống



Chi tiết
Xóa

© 2023 - Lập trình Web

1. Chỉnh sửa layout Page
2. Chức năng Thêm, xóa, sửa, chi tiết sách  
Hình ảnh sách ở Content/ImageBooks  
**Phải kiểm lỗi tất cả các dữ liệu nhập/ sửa ( ở cả 2 phía client và server)**
3. Lấy danh sách chủ đề từ CSDL
4. Trong danh sách chủ đề, thể hiện từng số lượng sách có tương ứng VD: Cuộc sống (2), Lập trình (2), Sức khỏe (0)...

# Yêu cầu: ASM5 – chi tiết cuốn sách

[Trang Chủ](#)[Đăng Ký](#)[Đăng nhập](#)[Giỏ hàng](#)

## Chủ đề

- Cuộc sống
- Lập trình
- Sức Khỏe

## THÔNG TIN CHI TIẾT SÁCH



**Tên sách: Cuộc Sống Rất Giống Cuộc Đời**

**Tác giả: Hải Dó**

**Mô tả:** Đàn ông tuổi 15 mơ ước thành đàn ông tuổi 20, đàn ông tuổi 20 mơ ước thành đàn ông tuổi 30, đàn ông tuổi 30 mơ ước được trở thành đàn ông tuổi 40 và đàn ông tuổi 40 lại mơ ước đặt chân lên cỗ máy thời gian để quay lại tuổi 30 với toàn bộ tài sản của mình! Vậy đấy!

**Giá bán: 61000 VNĐ**

[Đặt mua](#)

# Yêu cầu: ASM5 – Thống kê số lượng sách theo chủ đề

[Trang Chủ](#)[Đăng Ký](#)[Đăng nhập](#)[Giỏ hàng](#)

## Chủ đề

- Cuộc sống (2)
- Lập trình (2)
- Sức Khỏe (0)


## Thêm mới

### Cho tôi xin một vé đi tuổi thơ

[Edit](#)

Tác giả: Nguyễn Nhật Ánh

Thể loại: Cuộc sống




Chi tiết Xóa

### Cuộc Sống Rất Giống Cuộc Đời

[Edit](#)

Tác giả: Hải Dór

Thể loại: Cuộc sống



Chi tiết Xóa

© 2023 - Lập trình Web

# Yêu cầu: chỉnh sửa giao diện chi tiết như mẫu

Hãy chỉnh sửa giao diện ở phần chi tiết (details) sau tham khảo mẫu ở:

<https://www.fahasa.com/cho-toi-xin-mot-ve-di-tuoi-tho-bia-mem-tai-ban-2018.html>



## Cho Tôi Xin Một Vé Đi Tuổi Thơ (Bìa Mềm) (Tái Bản 2018)

Nhà cung cấp: **NXB Trẻ**

Tác giả: **Nguyễn Nhật Ánh**

Nhà xuất bản: **NXB Trẻ**

★★★★★ (79 đánh giá)

**61.600 đ** ~~80.000 đ~~ **-23%**

Thời gian giao hàng


Giao hàng đến **Thay đổi**

Chính sách đổi trả

Đổi trả sản phẩm trong 30 ngày **Xem thêm**

Số lượng:

— 1 +

 **Thêm vào giỏ hàng**

**Mua ngay**