

Model

Model?

- Model ជាបណ្តុំនៃ objects ជាច្រើនដែលវាមានតួនាទីផ្ទុកនូវទិន្នន័យរបស់កម្មវិធីហើយវាអាចមាននូវ business logic ផងដែរ។
- គេបែងចែក Model ជាបីប្រភេទ៖
 - Domain Model
 - View Model
 - និង Edit Model

Domain Model

- Domain model ជា object តំណាងអោយទិន្នន័យដែលមានក្នុង database ។
- Domain model តែងតែមានទំនាក់ទំនងពីតារាងមួយទៅតារាងមួយទៀត (one to one relationship) ក្នុង database ។
- Domain model វាជាប់ទាក់ទងជាមួយ data access layer របស់ application យើង (The Domain Model is related to the data access layer of our application) ដែលទាញទិន្នន័យពីឬបញ្ជូនទិន្នន័យទៅ database ។
- Domain model គេអាចនិយាយម្យ៉ាងទៀតថាជា entity model ឬ data model (The domain models are also referred to as the entity model or data model) ។

ឧទាហរណ៍៖

Table in database

dbo.Product			
	Column Name	Data Type	Allow Nulls
🔑	ProductID	int	<input type="checkbox"/>
	Name	nvarchar(50)	<input type="checkbox"/>
	BrandID	int	<input type="checkbox"/>
	SupplierID	bigint	<input type="checkbox"/>
	Qty	decimal(10, 2)	<input type="checkbox"/>
	Price	decimal(10, 2)	<input type="checkbox"/>
▶	Rating	int	<input type="checkbox"/>
			<input type="checkbox"/>

Domain model

```
public class Product
{
    public int ProductId { get; set; }
    public string Name { get; set; }

    public Decimal Price { get; set; }
    public int Rating { get; set; }

    public Brand Brand { get; set; }
    public Supplier Supplier { get; set; }
}
```

View Model

- View model សំដៅអំពី object ដែលផ្ទុកទិន្នន័យដែលត្រូវបង្ហាញអោយអ្នកប្រើប្រាស់មើលឃើញ (The View Model refers to the objects which hold the data that needs to be shown to the user) ។
- View model គឺជាប់ទាក់ទងនឹងផ្នែកមើលឃើញនៃកម្មវិធី (The View Model is related to the presentation layer of our application) ។

View Model

```
public class ProductViewModel
{
    public int ProductId { get; set; }
    public string Name { get; set; }

    public Decimal Price { get; set; }
    public int Rating { get; set; }

    public string BrandName { get; set; }
    public string SupplierName { get; set; }
```

Edit Model

- Edit ឬ input model គំណាងទិន្នន័យដែលចាំបាច់ត្រូវបង្ហាញអោយអ្នកប្រើប្រាស់ដើម្បីកែប្រែឬបញ្ចូល(The Edit Model or Input Model represents the data that needs to be presented to the user for modification/inserting)។

Edit Model

```
public class ProductEditModel
{
    public int ProductId { get; set; }

    [Required(ErrorMessage = "Product Name is Required")]
    [Display(Name = "Product Name")]
    public string Name { get; set; }

    public Decimal Price { get; set; }
    public int Rating { get; set; }

    public List<Brand> Brands { get; set; }
    public List<Supplier> Suppliers { get; set; }

    public int BrandID { get; set; }
    public int SupplierID { get; set; }
}
```


ការបញ្ជូនទិន្នន័យពី controller to view

- គេមានវិធីសាស្ត្រមួយចំនួនដើម្បីបញ្ជូនទិន្នន័យពី controller ទៅ view
 - ViewBag
 - ViewData
 - Model property
 - View() helper method

ការបញ្ជូនទិន្នន័យដោយប្រើ View() helper method

ឧទាហរណ៍

```
public class Product
{
    [Required]
    14 references
    public int ProductId { get; set; }
    [Required]
    17 references
    public string ProductName { get; set; }
    13 references
    public int? CategoryId { get; set; }
    0 references
    public int Quantity { get; set; }
    0 references
    public decimal? UnitPrice { get; set; }
    0 references
    public decimal? SalePrice { get; set; }
    0 references
    public string Description { get; set; }
    0 references
    public string Image { get; set; }
    0 references
    public Category Category { get; set; }
}
```

ឧទាហរណ៍

```
@model IEnumerable<Product>
@{
    ViewData["Title"] = "Index";
}
<h3>Product information</h3>
@if (Model != null)
{
    <table border="1">
        <tr>
            <td>@Html.DisplayNameFor(p => p.ProductId)</td>
            <td>@Html.DisplayNameFor(p => p.CategoryId)</td>
            <td>@Html.DisplayNameFor(p => p.ProductName)</td>
        </tr>
        @foreach (var p in Model)
        {
            <tr>
                <td>@p.ProductId</td>
                <td>@p.CategoryId</td>
                <td>@p.ProductName</td>
            </tr>
        }
    </table>
}

public class ProductController : Controller
{
    public IActionResult Index()
    {
        var products = new List<Product>
        {
            new Product{ ProductId=1,CategoryId=1,ProductName="Product-1" },
            new Product{ ProductId=2,CategoryId=1,ProductName="Product-2"},
            new Product{ ProductId=3,CategoryId=2,ProductName="Product-3"},
            new Product{ ProductId=4,CategoryId=2,ProductName="Product-4"},
        };
        return View(products);
    }
}
```



លទ្ធផល

Product information

ProductId	CategoryId	ProductName
1	1	Product-1
2	1	Product-2
3	2	Product-3
4	2	Product-4

ឧទាហរណ៍

```
public class Product
{
    [Required]
    14 references
    public int ProductId { get; set; }
    [Required]
    17 references
    public string ProductName { get; set; }
    13 references
    public int? CategoryId { get; set; }
    0 references
    public int Quantity { get; set; }
    0 references
    public decimal? UnitPrice { get; set; }
    0 references
    public decimal? SalePrice { get; set; }
    0 references
    public string Description { get; set; }
    0 references
    public string Image { get; set; }
    0 references
    public Category Category { get; set; }
}
```

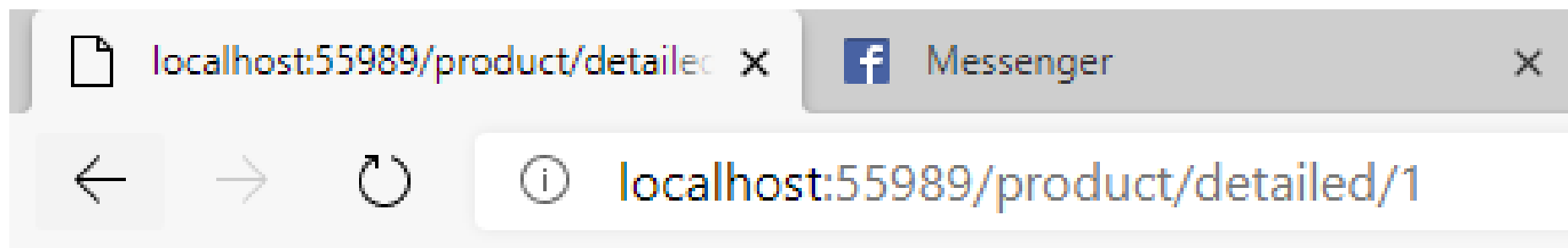
ឧទាហរណ៍-controller

```
public class ProductController : Controller
{
    0 references
    public IActionResult detailed(int id)
    {
        var products = new List<Product>
        {
            new Product{ ProductId=1,CategoryId=1,ProductName="Product-1" },
            new Product{ ProductId=2,CategoryId=1,ProductName="Product-2"},
            new Product{ ProductId=3,CategoryId=2,ProductName="Product-3"},
            new Product{ ProductId=4,CategoryId=2,ProductName="Product-4"},
        };
        var product = products.Where(p => p.ProductId == id).FirstOrDefault();
        return View(product);
    }
}
```

ឧទាហរណ៍-detailed view

```
Product.cs    detailed.cshtml  X Index.cshtml  ProductController.cs
1  @model Product
2  @{
3      Layout = null;
4  }
5  @if (Model != null)
6  {
7      <p>Id:<strong>@Model.ProductId</strong></p>
8      <p>Category:<strong>@Model.CategoryId</strong></p>
9      <p>Name:<strong>@Model.ProductName</strong></p>
10 }
```


លទ្ធផល



Id:1

Category:1

Name:Product-1

ការបញ្ជូនទិន្នន័យដោយប្រើ Model property

ឧទាហរណ៍

```
public class Product
{
    [Required]
    14 references
    public int ProductId { get; set; }
    [Required]
    17 references
    public string ProductName { get; set; }
    13 references
    public int? CategoryId { get; set; }
    0 references
    public int Quantity { get; set; }
    0 references
    public decimal? UnitPrice { get; set; }
    0 references
    public decimal? SalePrice { get; set; }
    0 references
    public string Description { get; set; }
    0 references
    public string Image { get; set; }
    0 references
    public Category Category { get; set; }
}
```

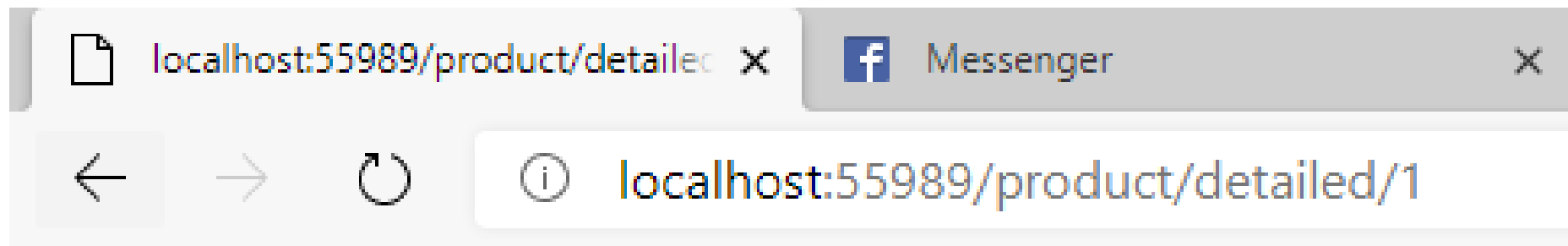
ឧទាហរណ៍-controller

```
public class ProductController : Controller
{
    References
    public IActionResult detailed(int id)
    {
        var products = new List<Product>
        {
            new Product{ ProductId=1,CategoryId=1,ProductName="Product-1" },
            new Product{ ProductId=2,CategoryId=1,ProductName="Product-2"},
            new Product{ ProductId=3,CategoryId=2,ProductName="Product-3"},
            new Product{ ProductId=4,CategoryId=2,ProductName="Product-4"},
        };
        var product = products.Where(p => p.ProductId == id).FirstOrDefault();
        ViewData.Model = product;
        return View();
    }
}
```

ឧទាហរណ៍-detailed view

```
Product.cs    detailed.cshtml  Index.cshtml  ProductController.cs
1  @model Product
2  @{
3      Layout = null;
4  }
5  @if (Model != null)
6  {
7      <p>Id:<strong>@Model.ProductId</strong></p>
8      <p>Category:<strong>@Model.CategoryId</strong></p>
9      <p>Name:<strong>@Model.ProductName</strong></p>
10 }
```

លទ្ធផល



Id:1

Category:1

Name:Product-1