* Tao form Submit rồi và khi nhấn Submit bind (liên kết ) dữ liệu đến

Property Của Object trong tham số truyền vào funtion IActionResult

Bind (liên kết ) đến Property Của Object dựa trên [asp-for ] Name] trong input , [asp-for ] Name] trong input trùng tên Property Của Object

* Form gửi dữ liệu về Controller . ASP.NET sẽ **tự động tạo một instance mới** của Class (kiểu dữ liệu tham chiếu ) và **bind** dữ liệu vào đó (gọi setter).

ViewData["Categories"] = new SelectList(\_context.Categories, "Id", "Name"); // truyền SelectList sang Vew :: Property “Id” của \_context.Categories đặt làm value còn property “Name” đặt làm content khi hiểu thị trong thẻ select

<select >

    <option value="1">Bút</option>

    <option value="2">Vở</option>

    <option value="3">Thước</option>

</select>

<select asp-for =”Property của Model” asp-items = “ViewBag.Key” > </select>

Khi chọn select value = “1” or value =”2” or value =”3” sẽ được gán vào Property của Model

**1. Tạo Model**

File: Models/Person.cs

namespace MyApp.Models

{

public class Person

{

public string Name { get; set; }

}

}

👉 Đây là class để lưu dữ liệu người dùng nhập vào.

**2. Tạo Controller**

File: Controllers/HomeController.cs

using Microsoft.AspNetCore.Mvc;

using MyApp.Models;

namespace MyApp.Controllers

{

public class HomeController : Controller

{

// GET: hiển thị form nhập

[HttpGet]

public IActionResult Index()

{

// Trả về view kèm 1 model trống

return View(new Person());

}

TH1

// POST: nhận dữ liệu khi submit

[HttpPost]

public IActionResult Submit(Person person)

{

// Lúc này person.Name chứa dữ liệu người dùng nhập

// Trả lại view Result để hiển thị

return View("Result", person);

}

}

}

TH2

[HttpPost] // Chỉ nhận POST requests

public async Task<IActionResult> Create( Item item)

{

if (ModelState.IsValid) // Kiểm tra validation // kiểu tra dữ liệu Item item có hợp lệ không

{

\_context.Items.Add(item); // Thêm vào DbContext

await \_context.SaveChangesAsync(); // Lưu vào database

return RedirectToAction("Index", “Item”); // Chuyển hướng sang Controller/Action

}

return View(item); // Nếu invalid, trả về view với data

}

👉 Ở đây có 2 action:

* **Index (GET)** → Hiển thị form.
* **Submit (POST)** → Nhận dữ liệu từ form, bind vào object Person, rồi chuyển sang view kết quả.

**3. Tạo View**

**Views/Home/Index.cshtml (Form nhập)**

@model MyApp.Models.Person

<h2>Nhập thông tin</h2>

<form asp-action="Submit" method="post" **asp-controller**="Item" novalidate>

<div>

<label for="Name">Tên:</label>

<input asp-for="Name" />

</div>

<button type="submit">Submit</button>

</form>

👉 Ở đây:

* @model MyApp.Models.Person khai báo View dùng model Person.
* <input asp-for="Name" /> → Razor sinh ra:
* <input type="text" id="Name" name="Name" value="" />

Lúc submit → giá trị sẽ bind vào property Name.

**Views/Home/Result.cshtml (Kết quả sau khi submit)**

@model MyApp.Models.Person

<h2>Kết quả</h2>

<p>Bạn đã nhập tên: <strong>@Model.Name</strong></p>

<a asp-action="Index">Quay lại</a>

👉 View này nhận Person từ action Submit và hiển thị dữ liệu.

 **asp-for="Name"** ánh xạ trực tiếp input tới property Name của model.

 Khi submit, dữ liệu trong ô input được bind vào property Name.