[HttpGet]

public IActionResult Create()

{

ViewBag.Categories = new SelectList(\_dataContext.Categories, "Id", "Name");

ViewBag.Brands = new SelectList(\_dataContext.Brands, "Id", "Name");

return View();

}

[HttpPost] // Chỉ nhận POST requests

public async Task <IActionResult> Create(ProductModel product)

{

ViewBag.Categories = new SelectList(\_dataContext.Categories, "Id", "Name" , product.CategoryId);

ViewBag.Brands = new SelectList(\_dataContext.Brands, "Id", "Name" , product.BrandId);

return View(product);

}

}

phần biệt 2 action trùng tên trong 1 Class

[HttpGet] // dùng để di chuyển đến Controller /Action bằng đường dẫn

[HttpPost] // gửi dữ liệu từ form đến Controller /Action khi Submit trong thẻ form

* Khi nào dùng 2 action cùng tên trong Class : khi có thẻ form

A screenshot of a computer program

AI-generated content may be incorrect.

* Tóm lại muốn thể hiện lỗi Property của đối tượng nếu .Tất cả dữ liệu khi truyền vào Property hợp lệ thì di chuyển đến Controller/Action khác không thì quay lại View chứa thẻ form

[HttpPost] // Chỉ nhận POST requests

* public IActionResult Create( Student item) // chỉ có Property của Student item trong function này sẽ ánh xạ đến

<**span** **asp-validation-for**="Name" class="text-danger"> </**span**>

* ModelState.IsValid **chỉ dựa vào dữ liệu từ form ban đầu**, không tự động thay đổi khi bạn gán object mới cho biến.

[HttpPost] // Chỉ nhận POST requests

public IActionResult Create( Student item) {

if (ModelState.IsValid)

{

return View("Index" , "Item");

}

return View (item);

}

View Create

@model Check.Models.Student

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

<**input** **asp-for**="Name" />

<**span** **asp-validation-for**="Name" class="text-danger"> </**span**>

<button type="submit">Lưu</button>

</**form**>

@section Scripts {

@{

await Html.PartialAsync("\_ValidationScriptsPartial");

}

}