LEARNING ASP.NET CORE 6 RAZOR PAGE

Create 1 :

Private readonly ApplicationDbContext \_db;

Public Category Category {get;set;}

Public CreateModel(ApplicationDbContext db)

{

\_db = db;

}

Public async Task<IActionResult> OnPost(Category category)

{

Await \_db.Category.AddAsync(category);

Await \_db.SaveChangeAsync();

Return RedicrectToPage(‘index’);

}

Create 2 :

Private readonly ApplicationDbContext \_db;

[BindProperty] //có thể để ở ngoài namespace để bind all

Public Category Category {get;set;}

Public CreateModel(ApplicationDbContext db)

{

\_db = db;

}

Public async Task<IActionResult> OnPost()

{

Await \_db.Category.AddAsync(Category);

Await \_db.SaveChangeAsync();

Return RedicrectToPage(‘index’);

}

Validation:

ModelState.isValid : trường nhập vào hợp lệ

Return Page(); : trả về page hiện tại

<span asp-validation-for=”Category.Name” class=”text-danger”><span> : thông báo lỗi của trường cate name

ModelState.AddMoreError(“Tên của error muốn đặt, có thể đặt theo tên của trường dữ liệu”,”Thông báo lỗi”); : thêm thông báo lỗi

<div asp-validation-summary=”All” ></div> : thông báo lỗi tất cả

[Display(Name=”Real name”)] : đặt ở model //xem tất cả tại docs microsoft data annotation

[Range(1,100,ErrorMessage = “must be in range of 1-100”)] : giới hạn độ lớn int

@section Scripts{

@{

<partial name “\_ValidationScriptsPartial”/>

}

} : trang sẽ không load lại nếu lặp lại một lỗi

Edit

\_db.Category.Find(id);

\_db.Category.Update(cate);

\_db.SaveChanges();

Delete

\_db.Category.Remove(cate);

Đặt tên lại cho action : [HttpPost,ActionName(“new name”)]

Temp data :

TempData[“ten”] = “Thong bao”;

In view : @if(TempData[“ten”] != null) { <h1>@TempData[“ten”]</h1> }

Partial view :

<partial name=”ten view” />

Toastr Alert - Thông báo đẹp, ở góc : asp.net core 6 freecodecamp 2h50p

@page “url” : địa chỉ routing trỏ đến