ctrl K, ctrl D: format code

EEUouaJ3

----------------------------------------------------------

@await Html.PartialAsync("HiepTran\_Header")

@await Html.PartialAsync("HiepTran\_Footer")

@RenderBody()

@await Component.InvokeAsync("CauLacBoMenu")

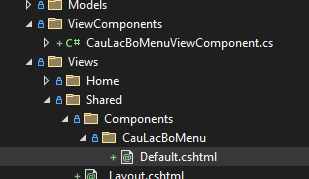
<**a** **asp-controller**="home" **asp-action**="themcauthu">Thêm mới một cầu thủ</**a**>

<**a** **asp-controller**="home" **asp-action**="suacauthu" asp-route-macauthu=”@item.ma”>Sua cầu thủ</**a**>

---------------------- data ----------------------

Scaffold-DbContext "Data Source=TRANHUYHIEP;Initial Catalog=OnlineShop;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFailover=False" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models

------------------------ menu ----------------------



using KiemTraLan2.Models;

using Microsoft.AspNetCore.Mvc;

namespace KiemTraLan2.ViewComponents

{

public class MenuViewComponent : ViewComponent

{

QlbongDaContext db = new QlbongDaContext();

public MenuViewComponent() { }

public IViewComponentResult Invoke()

{

var lst = db.Trandaus.Take(7).ToList();

return View(lst);

}

}

}

------------------------------------------------------------------------------------------

@addTagHelper \*, Microsoft.AspNetCore.Mvc.TagHelpers

@model List<Trandau>

@foreach (var item in Model)

{

var url = "https://localhost:44398/api/TranDauAPI/" + item.TranDauId;

<li>

<a onclick="getCauThuByCLB('@url')">@item.TranDauId</a>

</li>

}

<script>

const getCauThuByCLB = (url) => {

$.ajax({

type: "GET",

url: url,

dataType: "json",

success: (data) => {

let str = "";

$.each(data, (key, val) => {

str += `<div class="item">

<div class="block-4 text-center">

<figure class="block-4-image">

<img src="../images/${val.anh}" width="150" height="300" alt="Image placeholder" class="img-fluid">

</figure>

<div class="block-4-text p-4">

<h3><a href="#">${val.hoVaTen}</a></h3>

</div>

<a href="https://localhost:44398/Home/suaCauThu?cauThuId=${val.cauThuId}">Edit</a>

</div>

</div>`;

})

$("#showProduct").html(str);

}

})

}

</script>

---------------------- add ----------------------

<**a** **asp-controller**="home" **asp-action**="themcauthu">Thêm mới một cầu thủ</**a**>

-----

// them

[HttpGet]

public IActionResult ThemCauThu()

{

ViewBag.CauLacBoId = new SelectList(db.Caulacbos.ToList(), "CauLacBoId", "TenClb");

return View();

}

[HttpPost]

public IActionResult ThemCauThu(Cauthu cauthu)

{

if (ModelState.IsValid)

{

db.Cauthus.Add(cauthu);

db.SaveChanges();

return RedirectToAction("Index", "Home");

}

ViewBag.CauLacBoId = new SelectList(db.Caulacbos.ToList(), "CauLacBoId", "TenClb");

return View(cauthu);

}

------------------------ chi tiet ----------------------

// chi tiet

[HttpGet]

public IActionResult ChiTietCauThu(string cauThuId)

{

var temp = db.Cauthus.SingleOrDefault(x => x.CauThuId == cauThuId);

return View(temp);

}

---------------------- edit ----------------------

// sua

[HttpGet]

public IActionResult suaCauThu(string cauThuId)

{

var temp = db.Cauthus.SingleOrDefault(x => x.CauThuId == cauThuId);

ViewBag.CauLacBoId = new SelectList(db.Caulacbos.ToList(), "CauLacBoId", "TenClb");

return View(temp);

}

[HttpPost]

public IActionResult suaCauThu(Cauthu cauthu)

{

if (ModelState.IsValid)

{

db.Cauthus.Update(cauthu);

db.SaveChanges();

return RedirectToAction("Index", "Home");

}

ViewBag.CauLacBoId = new SelectList(db.Caulacbos.ToList(), "CauLacBoId", "TenClb");

return View(cauthu);

}

---------------------- other ----------------------

Layout = "~/Views/Shared/HiepTran\_Layout.cshtml";

RenderBody();

<a asp-controller="home" asp-action="ChiTietDoiTuong" asp-route-maSp="@Model.MaSp">@Model.TenSp</a>

---------------------- validate ----------------------

- Trung mat khau

[Compare("Password", ErrorMessage = "Passwords do not match")]

-email

[RegularExpression(@"^[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$", ErrorMessage = "email Không hợp lệ")]

- Chỉ được nhập chữ:

[RegularExpression("^[a-zA-Z ]\*$", ErrorMessage = "Chi duoc nhap chu")]

- Chỉ được nhập số:

[RegularExpression("^[0-9]\*$", ErrorMessage = "Chi duoc nhap số")]

- Validate email

[RegularExpression("^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$", ErrorMessage = "Email không đúng định dạng")]

- Đuôi phải là jpg

[RegularExpression(@"\w+\.jpg", ErrorMessage = "duoi phai la .jpg")]

- Độ dài nhỏ hơn 8:

[StringLength(8, ErrorMessage = "Name length can't be more than 8.")]

- Độ dài trong khoảng:

-[StringLength(8, ErrorMessage = "{0} length must be between {2} and {1}.", MinimumLength = 6)]