using ClassLibrary1.Repository;

using Microsoft.AspNetCore.Mvc;

namespace mvcshop.Controllers

{

public class ProductController : Controller

{

private readonly IShopRepo repo;

public ProductController(IShopRepo repo)

{

this.repo = repo;

}

public IActionResult Index()

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index","Admin"); //session expired/fail

}

List<ClassLibrary1.Models.Product> lib\_plist= repo.GetAllProducts();

List<mvcshop.Models.Product> mvc\_plist = new List<Models.Product>();

foreach (ClassLibrary1.Models.Product libp in lib\_plist)

{

mvcshop.Models.Product mvcp= new Models.Product();

mvcp.ProductId= libp.ProductId;

mvcp.ProductName= libp.ProductName;

mvcp.Price= libp.Price;

mvc\_plist.Add(mvcp);

}

return View(mvc\_plist);

}

[HttpGet]

public IActionResult Create()

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

return View();

}

[HttpPost]

public IActionResult Create(mvcshop.Models.Product mvcp)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

if (ModelState.IsValid)

{

ClassLibrary1.Models.Product libp = new ClassLibrary1.Models.Product();

libp.ProductId = mvcp.ProductId;

libp.ProductName = mvcp.ProductName;

libp.Price = mvcp.Price;

bool res = repo.AddProduct(libp);

}

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

}

[HttpGet]

public IActionResult Edit(int id)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

ClassLibrary1.Models.Product libp= repo.GetProductById(id);

mvcshop.Models.Product mvcp = new Models.Product();

mvcp.ProductId = libp.ProductId;

mvcp.ProductName = libp.ProductName;

mvcp.Price = libp.Price;

return View(mvcp);

}

[HttpPost]

public IActionResult Edit(mvcshop.Models.Product mvcp)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

if (ModelState.IsValid)

{

ClassLibrary1.Models.Product libp = new ClassLibrary1.Models.Product();

libp.ProductId = mvcp.ProductId;

libp.ProductName = mvcp.ProductName;

libp.Price = mvcp.Price;

bool res = repo.UpdateProduct(libp);

}

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

}

[HttpGet]

public IActionResult Details(int id)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

ClassLibrary1.Models.Product libp = repo.GetProductById(id);

mvcshop.Models.Product mvcp = new Models.Product();

mvcp.ProductId = libp.ProductId;

mvcp.ProductName = libp.ProductName;

mvcp.Price = libp.Price;

return View(mvcp);

}

[HttpGet]

public IActionResult Delete(int id)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

ClassLibrary1.Models.Product libp = repo.GetProductById(id);

mvcshop.Models.Product mvcp = new Models.Product();

mvcp.ProductId = libp.ProductId;

mvcp.ProductName = libp.ProductName;

mvcp.Price = libp.Price;

return View(mvcp);

}

[HttpPost]

[ActionName(name:"Delete")]

public IActionResult DeleteConfirmed(int id)

{

if (HttpContext.Session.GetString("email") == null)

{

return RedirectToAction("Index", "Admin"); //session expired/fail

}

repo.DeleteProduct(id);

return RedirectToAction("Index");

}

}

}