**Đăng ký, đăng Nhập, hash password**

Đăng ký tài khoản mới

Tạo hàm Hash Password

Tại foler Helper, tạo Folder mới: PasswordHelper.cs

using System.Security.Cryptography;

using System.Text;

namespace ShopManager.Helper

{

public static class PasswordHelper

{

public static string GenerateRandomKey(int length = 5)

{

var pattern = @"fksjifjeijkgjioewmmvnTUYOMBGYHCGJKLHJHJK\*&^%%&%^%&!";

var stringBuilder = new StringBuilder();

var rd = new Random();

for (int i = 0; i < length; i++)

{

stringBuilder.Append(pattern[rd.Next(0, pattern.Length)]);

}

return stringBuilder.ToString();

}

#region [Hashing Extension]

public static string ToSHA256Hash(this string password, string? saltKey)

{

var sha256 = SHA256.Create();

byte[] encryptedSHA256 = sha256.ComputeHash(Encoding.UTF8.GetBytes(string.Concat(password, saltKey)));

sha256.Clear();

return Convert.ToBase64String(encryptedSHA256);

}

public static string ToSHA512Hash(this string password, string? saltKey)

{

SHA512Managed sha512 = new SHA512Managed();

byte[] encryptedSHA512 = sha512.ComputeHash(Encoding.UTF8.GetBytes(string.Concat(password, saltKey)));

sha512.Clear();

return Convert.ToBase64String(encryptedSHA512);

}

public static string ToMd5Hash(this string password, string? saltKey)

{

using (var md5 = MD5.Create())

{

byte[] data = md5.ComputeHash(Encoding.UTF8.GetBytes(string.Concat(password, saltKey)));

StringBuilder sBuilder = new StringBuilder();

for (int i = 0; i < data.Length; i++)

{

sBuilder.Append(data[i].ToString("x2"));

}

return sBuilder.ToString();

}

}

#endregion

}

}

Khai báo Authentication bằng Cookie

[Overview of ASP.NET Core Authentication | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/security/authentication/?view=aspnetcore-8.0)

Program.cs

using Microsoft.AspNetCore.Authentication.Cookies;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

builder.Services.AddAuthentication(CookieAuthenticationDefaults.AuthenticationScheme)

.AddCookie(option =>

{

option.LoginPath = "/Customer/SignIn";

option.AccessDeniedPath = "/AccessDenied";

});

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllerRoute(

name: "Admin",

pattern: "{area:exists}/{controller=Category}/{action=Index}/{id?}");

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

Controllers/HomeController.cs

using Microsoft.AspNetCore.Mvc;

using ShopManager.Models;

using System.Diagnostics;

namespace ShopManager.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult Index()

{

return View();

}

public IActionResult Privacy()

{

return View();

}

[Route("/404")]

public IActionResult PageNotfound()

{

return View();

}

[Route("/AccessDenied")]

public IActionResult AccessDenied()

{

return View();

}

[ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

public IActionResult Error()

{

return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

}

}

}

Thêm Razor View cho function AccessDenied

Views/Home/AccessDenied.cshtml

@{

ViewData["Title"] = "AccessDenied";

}

<!-- Single Page Header start -->

<div class="container-fluid page-header py-5">

<h1 class="text-center text-white display-6">Access Denied</h1>

<ol class="breadcrumb justify-content-center mb-0">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item"><a href="#">Pages</a></li>

<li class="breadcrumb-item active text-white">Access Denied</li>

</ol>

</div>

<!-- Single Page Header End -->

<!-- 404 Start -->

<div class="container-fluid py-5">

<div class="container py-5 text-center">

<div class="row justify-content-center">

<div class="col-lg-6">

<i class="bi bi-exclamation-triangle display-1 text-secondary"></i>

<h1 class="display-1">Thông báo</h1>

<h1 class="mb-4">Không truy cập được</h1>

<p class="mb-4">Bạn không được cấp quyền để truy cập địa chỉ này</p>

<a class="btn border-secondary rounded-pill py-3 px-5" href="/">Trở về trang chủ</a>

</div>

</div>

</div>

</div>

<!-- 404 End -->

Controllers/CustomerController.cs

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Helper;

using ShopManager.Models;

using System.Data;

using System.Security.Claims;

namespace ShopManager.Controllers

{

public class CustomerController : Controller

{

CustomerDAL customerDAL = new CustomerDAL();

public IActionResult Index()

{

return View();

}

// Profile

[Authorize]

public IActionResult Profile()

{

…

}

// Update Detail Customer

[HttpPost]

public IActionResult UpdateDetailCustomer(Customer customerUpdate, IFormFile ImgUpload)

{

//Tự động lấy thời gian updateAt theo giờ hệ thống khi Customer Update thông tin

DateTime now = DateTime.Now;

customerUpdate.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerUpdate.Img = ImageName;

}

else if (customerUpdate.Img == null)

{

customerUpdate.Img = "";

}

//kiểm tra Address có null hay không

if (customerUpdate.Address == null)

{

customerUpdate.Address = "";

}

bool isSuccess = customerDAL.UpdateDetailCustomer(customerUpdate, customerUpdate.Id);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Cập nhật thông tin thành công";

return RedirectToAction("Profile");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("Profile");

}

}

// ------------------ SIGN UP --------------------

public IActionResult SignUp()

{

return View();

}

[HttpPost]

public IActionResult SignUp(Customer customerSignUp, IFormFile ImgUpload)

{

// RegisterAt va UpdateAt được lấy tự động theo giờ hệ thống

DateTime now = DateTime.Now;

customerSignUp.RegisterAt = now;

customerSignUp.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerSignUp.Img = ImageName;

}

else

{

//Sử dụng avatar mặc định của project

customerSignUp.Img = "avatar-default.jpg";

}

//kiểm tra Address có null hay không

if (customerSignUp.Address == null)

{

customerSignUp.Address = "";

}

//HashPassword

customerSignUp.RandomKey = PasswordHelper.GenerateRandomKey();

customerSignUp.Password = customerSignUp.Password.ToMd5Hash(customerSignUp.RandomKey);

bool isSuccess = customerDAL.SignUp(customerSignUp);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Đăng ký thành công";

return RedirectToAction("SignIn");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("SignUp");

}

}

}

}

Add Razor View Empty cho SignUp

Views/Customer/SignUp.cshtml

@model ShopManager.Models.Customer

@{

ViewData["Title"] = "SignUp";

}

<!-- Form Start -->

<div class="container-fluid py-5" style="margin-top:100px">

<div class="container py-5">

<h2 class="mb-4">Đăng ký tài khoản mới</h2>

<**form** method="post" **asp-action**="SignUp" enctype="multipart/form-data">

<div class="row g-5">

<div class="col-md-12 col-lg-6 col-xl-5">

<!-- Avatar Start -->

<!-- Avatar -->

<div class="w-50 justify-content-md-center"

style="position: relative; left: 50%; transform: translate(-50%)">

<**img** src="~/img/avatar-default.jpg"

class="img-fluid w-100"

id="image\_preview"

style="

aspect-ratio: 1;

object-fit: cover;

border-radius: 100%;

box-shadow: 0 5px 15px 0 rgba(0, 0, 0, 0.25);

" />

</div>

<!-- Avatar End -->

<!-- Menu Profile Start -->

<div class="d-flex mt-5 justify-content-center">

<a class="btn btn-outline-primary" style="margin-right: 20px; position:relative">

<label style="font-size: 16px">Chọn ảnh đại diện</label>

<input type="file"

name="ImgUpload"

id="image\_upload"

style="

position: absolute;

width: 100%;

height: 100%;

top: 0;

left: 0;

opacity: 0;

" />

</a>

</div>

<!-- Menu Profile End -->

</div>

<div class="col-md-12 col-lg-6 col-xl-7">

<div class="row">

<div class="col-md-12 col-lg-6">

<div class="form-item w-100">

<label class="form-label my-3">Họ<**span** **asp-validation-for**="LastName" class="text-danger"></**span**></label>

<**input** **type**="text" **asp-for**="LastName" class="form-control" />

</div>

</div>

<div class="col-md-12 col-lg-6">

<div class="form-item w-100">

<label class="form-label my-3">Tên<**span** **asp-validation-for**="FirstName" class="text-danger"></**span**></label>

<**input** **type**="text" **asp-for**="FirstName" class="form-control" />

</div>

</div>

</div>

<div class="form-item">

<label class="form-label my-3">Mật khẩu<**span** **asp-validation-for**="Password" class="text-danger"></**span**></label>

<**input** **type**="text" **asp-for**="Password" class="form-control" />

</div>

<div class="form-item">

<label class="form-label my-3">Số điện thoại<**span** **asp-validation-for**="Phone" class="text-danger"></**span**></label>

<**input** **type**="text" **asp-for**="Phone" class="form-control" />

</div>

<div class="form-item">

<label class="form-label my-3">Email <**span** **asp-validation-for**="Email" class="text-danger"></**span**></label>

<**input** **type**="email" **asp-for**="Email" class="form-control" placeholder="" />

</div>

<div class="form-item">

<label class="form-label my-3">Ngày sinh<sup>\*</sup></label>

<**input** **type**="date" **asp-for**="DateOfBirth" class="form-control w-50" />

</div>

<div class="form-item">

<label class="form-label my-3">Địa chỉ<sup></sup></label>

<**input** **type**="text" **asp-for**="Address" class="form-control" />

</div>

<div class="form-item my-3">

<button type="submit" class="btn btn-primary">

<span class="mx-3" style="color: white"> Đăng ký </span>

</button>

</div>

</div>

</div>

</**form**>

</div>

</div>

<!-- Form Page End -->

<!-- Display Image Upload Start -->

<script>

// --------- Image Upload ---------------

const imageUpload = document.getElementById("image\_upload");

const imagePreview = document.getElementById("image\_preview");

imageUpload.addEventListener("change", (event) => {

const files = event.target.files;

const [file] = event.target.files;

console.log("files", file);

imagePreview.src = URL.createObjectURL(event.target.files[0]);

imagePreview.onload = function () {

URL.revokeObjectURL(output.src); // free memory

};

});

</script>

<!-- Display Image Upload End -->

Đăng nhập với tài khoản đã đăng ký

Tạo Model mới: CustomerSignIn.cs

using System.ComponentModel.DataAnnotations;

namespace ShopManager.Models

{

public class CustomerSignIn

{

[Display(Name = "Email")]

[Required(ErrorMessage = "\*")]

[MaxLength(50, ErrorMessage = "Tối đa 50 kí tự")]

public string Email { get; set; }

[Display(Name = "Mật khẩu")]

[Required(ErrorMessage = "\*")]

[MaxLength(200, ErrorMessage = "Tối đa 200 kí tự")]

public string Password { get; set; }

}

}

Controllers/CustomerController.cs

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Helper;

using ShopManager.Models;

using System.Data;

using System.Security.Claims;

namespace ShopManager.Controllers

{

public class CustomerController : Controller

{

CustomerDAL customerDAL = new CustomerDAL();

public IActionResult Index()

{

return View();

}

// Profile

[Authorize]

public IActionResult Profile()

{

string idTam = HttpContext.User.FindFirstValue("CustomerId");

if (idTam == null)

{

return RedirectToAction("SignIn");

}

int idCustomer = Convert.ToInt32(idTam!);

Customer? customer = new Customer();

customer = customerDAL.GetCustomerById(idCustomer);

// Nếu không tìm thấy User thì trả về trang 404 - Not Found

if (customer == null)

{

return Redirect("/404");

}

return View(customer);

}

// Update Detail Customer

[HttpPost]

public IActionResult UpdateDetailCustomer(Customer customerUpdate, IFormFile ImgUpload)

{

//Tự động lấy thời gian updateAt theo giờ hệ thống khi Customer Update thông tin

DateTime now = DateTime.Now;

customerUpdate.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerUpdate.Img = ImageName;

}

else if (customerUpdate.Img == null)

{

customerUpdate.Img = "";

}

//kiểm tra Address có null hay không

if (customerUpdate.Address == null)

{

customerUpdate.Address = "";

}

bool isSuccess = customerDAL.UpdateDetailCustomer(customerUpdate, customerUpdate.Id);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Cập nhật thông tin thành công";

return RedirectToAction("Profile");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("Profile");

}

}

// ------------------ SIGN IN --------------------

public IActionResult SignIn(string? ReturnUrl)

{

ViewBag.ReturnUrl = ReturnUrl;

return View();

}

[HttpPost]

public async Task<IActionResult> SignInAsync(CustomerSignIn customerSignIn, string? ReturnUrl)

{

if (ModelState.IsValid)

{

ViewBag.ReturnUrl = ReturnUrl;

Customer? customer = new Customer();

customer = customerDAL.GetCustomerByEmail(customerSignIn.Email);

if (customer == null)

{

ModelState.AddModelError("Loi", "Không có khách hàng này");

}

else

{

if (customer.IsActive == 0)

{

ModelState.AddModelError("Thông báo", "Tài khoản của bạn đã bị vô hiệu hóa. Vui lòng liên hệ Admin");

}

else

{

string hashPassword = customerSignIn.Password.ToMd5Hash(customer.RandomKey);

if (customer.Password != hashPassword)

{

ModelState.AddModelError("Thông báo", "Sai mật khẩu");

}

else

{

var claims = new List<Claim>

{

new Claim("CustomerEmail", customer.Email),

new Claim(ClaimTypes.Name, customer.FirstName),

new Claim("CustomerFirstName", customer.FirstName),

new Claim("CustomerLastName", customer.LastName),

new Claim("CustomerId", customer.Id.ToString()),

//claim -role động

new Claim(ClaimTypes.Role, customer.Role == 1 ? "Administrator" : "Customer"),

};

var claimIdentity = new ClaimsIdentity(claims, "login");

var claimPricipal = new ClaimsPrincipal(claimIdentity);

await HttpContext.SignInAsync(claimPricipal);

if (Url.IsLocalUrl(ReturnUrl))

{

return Redirect(ReturnUrl);

}

else

{

return Redirect("/");

}

}

}

}

return View();

}

else return View();

}

// ------------------ SIGN UP --------------------

public IActionResult SignUp()

{

return View();

}

[HttpPost]

public IActionResult SignUp(Customer customerSignUp, IFormFile ImgUpload)

{

// RegisterAt va UpdateAt được lấy tự động theo giờ hệ thống

DateTime now = DateTime.Now;

customerSignUp.RegisterAt = now;

customerSignUp.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerSignUp.Img = ImageName;

}

else

{

//Sử dụng avatar mặc định của project

customerSignUp.Img = "avatar-default.jpg";

}

//kiểm tra Address có null hay không

if (customerSignUp.Address == null)

{

customerSignUp.Address = "";

}

//HashPassword

customerSignUp.RandomKey = PasswordHelper.GenerateRandomKey();

customerSignUp.Password = customerSignUp.Password.ToMd5Hash(customerSignUp.RandomKey);

bool isSuccess = customerDAL.SignUp(customerSignUp);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Đăng ký thành công";

return RedirectToAction("SignIn");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("SignUp");

}

}

}

}

Tạo Razor View Empty cho SignIn

Views/Customer/SignIn.cshtml

@model ShopManager.Models.CustomerSignIn

<!-- Form Start -->

<div div class="container-fluid py-5" style="margin-top: 130px">

<div class="container">

<div class="row justify-content-md-center">

<!-- Alert Success Start -->

@if (!string.IsNullOrWhiteSpace(TempData["ProfileSuccessMessage"]?.ToString()))

{

<div class="w-100 d-flex justify-content-end">

<div class="alert alert-success alert-dismissible fade show w-25 r-0 mr-5" role="alert" id="alert\_error">

<strong>Success!</strong> @TempData["ProfileSuccessMessage"]?.ToString()

<button type="button" class="btn-close" data-dismiss="alert" aria-label="Close">

</button>

</div>

</div>

}

<!-- Alert Success End -->

<div class="col-md-12 col-lg-5 col-xl-4">

<h2>Đăng nhập</h2>

<**form** method="post" **asp-action**="SignIn"

**asp-route-returnUrl**="@ViewBag.ReturnUrl"

>

<**div** class="text-danger" **asp-validation-summary**="All"></**div**>

<!-- Email input -->

<div data-mdb-input-init class="form-outline mb-2">

<label class="form-label" for="form2Example1">Email <**span** **asp-validation-for**="Email" class="text-danger"></**span**></label>

<**input** **type**="email" **asp-for**="Email" class="form-control" />

</div>

<!-- Password input -->

<div data-mdb-input-init class="form-outline mb-4">

<label class="form-label">Mật khẩu <**span** **asp-validation-for**="Password" class="text-danger"></**span**></label>

<**input** **type**="password" **asp-for**="Password" class="form-control" />

</div>

<!-- 2 column grid layout for inline styling -->

<div class="row mb-4">

<div class="col d-flex justify-content-center">

<!-- Checkbox -->

<div class="form-check">

<input class="form-check-input"

type="checkbox"

value=""

id="form2Example31"

checked />

<label class="form-check-label" for="form2Example31"> Remember me </label>

</div>

</div>

<div class="col">

<!-- Simple link -->

<**a** **asp-controller**="Customer" **asp-action**="ForgotPassword">Quên mật khẩu?</**a**>

</div>

</div>

<!-- Submit button -->

<div class="">

<button type="submit"

data-mdb-button-init

data-mdb-ripple-init

class="btn btn-primary btn-block mb-4 w-100">

<span class="px-2" style="color: white"> Sign in </span>

</button>

</div>

<!-- Register buttons -->

<div class="text-center">

<p>

Chưa có tài khoản?

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="SignUp">Đăng ký</**a**>

</p>

<p>or sign up with:</p>

<button type="button"

data-mdb-button-init

data-mdb-ripple-init

class="btn btn-link btn-floating mx-1">

<i class="fab fa-facebook-f"></i>

</button>

<button type="button"

data-mdb-button-init

data-mdb-ripple-init

class="btn btn-link btn-floating mx-1">

<i class="fab fa-google"></i>

</button>

<button type="button"

data-mdb-button-init

data-mdb-ripple-init

class="btn btn-link btn-floating mx-1">

<i class="fab fa-twitter"></i>

</button>

<button type="button"

data-mdb-button-init

data-mdb-ripple-init

class="btn btn-link btn-floating mx-1">

<i class="fab fa-github"></i>

</button>

</div>

</**form**>

</div>

</div>

</div>

</div>

<!-- Form End -->

Đăng xuất

Controllers/CustomerController.cs

using Microsoft.AspNetCore.Authentication;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Helper;

using ShopManager.Models;

using System.Data;

using System.Security.Claims;

namespace ShopManager.Controllers

{

public class CustomerController : Controller

{

CustomerDAL customerDAL = new CustomerDAL();

public IActionResult Index()

{

return View();

}

// Profile

[Authorize]

public IActionResult Profile()

{

string idTam = HttpContext.User.FindFirstValue("CustomerId");

if (idTam == null)

{

return RedirectToAction("SignIn");

}

int idCustomer = Convert.ToInt32(idTam!);

Customer? customer = new Customer();

customer = customerDAL.GetCustomerById(idCustomer);

// Nếu không tìm thấy User thì trả về trang 404 - Not Found

if (customer == null)

{

return Redirect("/404");

}

return View(customer);

}

// Update Detail Customer

[HttpPost]

public IActionResult UpdateDetailCustomer(Customer customerUpdate, IFormFile ImgUpload)

{

//Tự động lấy thời gian updateAt theo giờ hệ thống khi Customer Update thông tin

DateTime now = DateTime.Now;

customerUpdate.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerUpdate.Img = ImageName;

}

else if (customerUpdate.Img == null)

{

customerUpdate.Img = "";

}

//kiểm tra Address có null hay không

if (customerUpdate.Address == null)

{

customerUpdate.Address = "";

}

bool isSuccess = customerDAL.UpdateDetailCustomer(customerUpdate, customerUpdate.Id);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Cập nhật thông tin thành công";

return RedirectToAction("Profile");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("Profile");

}

}

// ------------------ SIGN IN --------------------

public IActionResult SignIn(string? ReturnUrl)

{

ViewBag.ReturnUrl = ReturnUrl;

return View();

}

[HttpPost]

public async Task<IActionResult> SignInAsync(CustomerSignIn customerSignIn, string? ReturnUrl)

{

if (ModelState.IsValid)

{

ViewBag.ReturnUrl = ReturnUrl;

Customer? customer = new Customer();

customer = customerDAL.GetCustomerByEmail(customerSignIn.Email);

if (customer == null)

{

ModelState.AddModelError("Loi", "Không có khách hàng này");

}

else

{

if (customer.IsActive == 0)

{

ModelState.AddModelError("Thông báo", "Tài khoản của bạn đã bị vô hiệu hóa. Vui lòng liên hệ Admin");

}

else

{

string hashPassword = customerSignIn.Password.ToMd5Hash(customer.RandomKey);

if (customer.Password != hashPassword)

{

ModelState.AddModelError("Thông báo", "Sai mật khẩu");

}

else

{

var claims = new List<Claim>

{

new Claim("CustomerEmail", customer.Email),

new Claim(ClaimTypes.Name, customer.FirstName),

new Claim("CustomerFirstName", customer.FirstName),

new Claim("CustomerLastName", customer.LastName),

new Claim("CustomerId", customer.Id.ToString()),

//claim -role động (Cấp quyền)

new Claim(ClaimTypes.Role, customer.Role == 1 ? "Administrator" : "Customer"),

};

var claimIdentity = new ClaimsIdentity(claims, "login");

var claimPricipal = new ClaimsPrincipal(claimIdentity);

await HttpContext.SignInAsync(claimPricipal);

if (Url.IsLocalUrl(ReturnUrl))

{

return Redirect(ReturnUrl);

}

else

{

return Redirect("/");

}

}

}

}

return View();

}

else return View();

}

//------------ SIGN OUT -------------

[Authorize]

public async Task<IActionResult> SignOutAsync()

{

await HttpContext.SignOutAsync();

return Redirect("/");

}

// ------------------ SIGN UP --------------------

public IActionResult SignUp()

{

return View();

}

[HttpPost]

public IActionResult SignUp(Customer customerSignUp, IFormFile ImgUpload)

{

// RegisterAt va UpdateAt được lấy tự động theo giờ hệ thống

DateTime now = DateTime.Now;

customerSignUp.RegisterAt = now;

customerSignUp.UpdateAt = now;

//Nếu có hình ảnh được Upload

if (ImgUpload != null)

{

//Upload Hinh

var ImageName = ImageHelper.UpLoadImage(ImgUpload, "KhachHang");

customerSignUp.Img = ImageName;

}

else

{

//Sử dụng avatar mặc định của project

customerSignUp.Img = "avatar-default.jpg";

}

//kiểm tra Address có null hay không

if (customerSignUp.Address == null)

{

customerSignUp.Address = "";

}

//HashPassword

customerSignUp.RandomKey = PasswordHelper.GenerateRandomKey();

customerSignUp.Password = customerSignUp.Password.ToMd5Hash(customerSignUp.RandomKey);

bool isSuccess = customerDAL.SignUp(customerSignUp);

// Kiểm tra truy vấn SQL thành công hay không?

if (isSuccess)

{

// Truy vấn Thành công

Console.WriteLine("Update Customer Success");

TempData["ProfileSuccessMessage"] = "Đăng ký thành công";

return RedirectToAction("SignIn");

}

else

{

// Truy vấn Thất bại

Console.WriteLine("Update Customer Fail");

TempData["ProfileErrorMessage"] = "Lỗi hệ thống";

return RedirectToAction("SignUp");

}

}

}

}

Thêm sự kiện đăng xuất trong View Profile

Views/Customer/Profile.cshtml

@model ShopManager.Models.Customer

@{

ViewData["Title"] = "Profile";

}

<!-- Single Page Header start -->

<div class="container-fluid page-header py-5">

<h1 class="text-center text-white display-6">Profile</h1>

<ol class="breadcrumb justify-content-center mb-0">

<li class="breadcrumb-item"><a href="#">Home</a></li>

<li class="breadcrumb-item"><a href="#">Pages</a></li>

<li class="breadcrumb-item active text-white">Profile</li>

</ol>

</div>

<!-- Single Page Header End -->

<!-- Profile Page Start -->

<div class="container-fluid py-5">

<div class="container ">

<!-- Alert Success Start -->

@if (!string.IsNullOrWhiteSpace(TempData["ProfileSuccessMessage"]?.ToString()))

{

<div class="w-100 d-flex justify-content-end">

<div class="alert alert-success alert-dismissible fade show w-50 r-0" role="alert" id="alert\_error">

<strong>Success!</strong> @TempData["ProfileSuccessMessage"]?.ToString()

<button type="button" class="btn-close" data-dismiss="alert" aria-label="Close">

</button>

</div>

</div>

}

<!-- Alert Success End -->

<h1>@Context.User.Identity.Name</h1>

<!-- Alert Error Start -->

@if (!string.IsNullOrWhiteSpace(TempData["ProfileErrorMessage"]?.ToString()))

{

<div class="w-100 d-flex justify-content-end">

<div class="alert alert-danger alert-dismissible fade show w-50 r-0" role="alert" id="alert\_error">

<strong>Error!</strong> @TempData["ProfileErrorMessage"]?.ToString()

<button type="button" class="btn-close" data-dismiss="alert" aria-label="Close">

</button>

</div>

</div>

}

<!-- Alert Error End -->

<h1 class="mb-4">Trang cá nhân</h1>

<form method="post" asp-action="UpdateDetailCustomer" asp-controller="Customer" enctype="multipart/form-data">

<div class="row g-5">

<div class="col-md-12 col-lg-6 col-xl-5">

<!-- Avatar -->

<div class="w-50 justify-content-md-center"

style="position: relative; left: 50%; transform: translate(-50%)">

<img src="~/images/KhachHang/@Model.Img"

class="img-fluid w-100"

id="image\_preview"

style="

aspect-ratio: 1;

object-fit: cover;

border-radius: 100%;

box-shadow: 0 5px 15px 0 rgba(0, 0, 0, 0.25);

" />

<div style="

position: absolute;

bottom: 20px;

right: 0px;

background-color: white;

padding: 10px 15px;

border-radius: 100%;

box-shadow: 0 5px 15px 0 rgba(0, 0, 0, 0.25);

">

<input type="file"

name="ImgUpload"

id="image\_upload"

accept="image/\*"

style=" position: absolute; width: 100%; top: 0; left: 0; height: 100%; opacity: 0; " />

<input type="text" asp-for="Img" style="display:none" />

<i class="bi bi-camera" style="color: grey"></i>

</div>

</div>

<!-- Menu Profile -->

<div class="d-flex mt-5 justify-content-center">

<a href="#" class="btn btn-outline-primary" style="margin-right: 20px">

<label style="font-size: 16px">Đổi mật khẩu</label>

</a>

<a asp-controller="Customer" asp-action="SignOut" class="btn btn-danger"><label>Đăng xuất</label></a>

</div>

</div>

<div class="col-md-12 col-lg-6 col-xl-7">

<div class="row">

<input type="text" asp-for="Id" style="display:none" />

<div class="col-md-12 col-lg-6">

<div class="form-item w-100">

<label class="form-label my-3">Họ<span asp-validation-for="LastName" class="text-danger"></span></label>

<input asp-for="LastName" type="text" class="form-control" />

</div>

</div>

<div class="col-md-12 col-lg-6">

<div class="form-item w-100">

<label class="form-label my-3">Tên<span asp-validation-for="FirstName" class="text-danger"></span></label>

<input asp-for="FirstName" type="text" class="form-control" />

</div>

</div>

</div>

<div class="form-item">

<label class="form-label my-3">Số điện thoại<span asp-validation-for="Phone" class="text-danger"></span></label>

<input asp-for="Phone" type="text" class="form-control" />

</div>

<div class="form-item">

<label class="form-label my-3">Email <span asp-validation-for="Email" class="text-danger"></span></label>

<input asp-for="Email" type="email" class="form-control" placeholder="" />

</div>

<div class="form-item">

<label class="form-label my-3">Ngày sinh</label>

<input asp-for="DateOfBirth" type="date" value="@Model.DateOfBirth.ToString("yyyy-MM-dd")" class="form-control w-50" />

</div>

<div class="form-item">

<label class="form-label my-3">Địa chỉ </label>

<input asp-for="Address" type="text" class="form-control" placeholder="" />

</div>

<div class="form-item my-3">

<button class="btn btn-primary " type="submit">

<label class="mx-3" style="color: white"> Lưu </label>

</button>

</div>

</div>

</div>

</form>

</div>

</div>

<!-- Profile Page End -->

<!-- Display Image Upload Start -->

<script>

// --------- Image Upload ---------------

const imageUpload = document.getElementById("image\_upload");

const imagePreview = document.getElementById("image\_preview");

imageUpload.addEventListener("change", (event) => {

const files = event.target.files;

const [file] = event.target.files;

console.log("files", file);

imagePreview.src = URL.createObjectURL(event.target.files[0]);

imagePreview.onload = function () {

URL.revokeObjectURL(output.src); // free memory

};

});

// ------------- Auto Close Alert ----------------

const alert\_error = document.getElementById("alert\_error")

window.setTimeout(function () {

$(".alert").fadeTo(500, 0).slideUp(500, function () {

$(this).remove();

});

}, 2000);

</script>

<!-- Display Image Upload End -->

Thêm quyền truy cập trang quản lý cho admin (khách hàng không được cấp quyền sẽ không được truy cập)

Areas/Admin/Controllers/ProductAdminController.cs

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using ShopManager.Areas.Admin.DAL;

using ShopManager.Areas.Admin.Models;

using ShopManager.Helper;

namespace ShopManager.Areas.Admin.Controllers

{

[Area("Admin")]

[Authorize(Roles = "Administrator")]

public class ProductAdminController : Controller

{

...

}

}

Tương tự với các Controller Admin khác

Ẩn menu truy cập trang quản lý, chỉ hiển thị mới tài khoản có Role là 1 (Administrator)

ViewComponents/MenuDynamicViewComponent.cs

using Microsoft.AspNetCore.Mvc;

using ShopManager.DAL;

using ShopManager.Models;

using System.Security.Claims;

namespace ShopManager.ViewComponents

{

public class MenuDynamicViewComponent : ViewComponent

{

MenuDAL menuDAL = new MenuDAL();

public IViewComponentResult Invoke()

{

//Truy cập Role của Customer đang đăng nhâp

string RoleCustomer = HttpContext.User.FindFirstValue(ClaimTypes.Role);

List<MenuItem> listMenu = new List<MenuItem>();

List<NavbarItem> navBar = new List<NavbarItem>();

listMenu = menuDAL.GetAllMenu();

foreach (var item in listMenu)

{

//Nếu Không được phân quyền để truy cập trang Admin

// Và đường dẫn Url của Menu chứa Area Admin

if (RoleCustomer != "Administrator" && item.MenuUrl.Contains("Admin")){

//Bỏ qua Menu này

}

else

{

// Is Nav Bar Item

if (item.ParentId == null)

{

navBar.Add(

new NavbarItem()

{

Id = item.Id,

ParentId = item.ParentId,

Title = item.Title,

MenuUrl = item.MenuUrl,

MenuIndex = item.MenuIndex,

isVisible = item.isVisible,

subItems = new List<NavbarItem>(),

}

);

}

//Is Dropdown Item

else

{

//Find Item Parent

var navbarParent = navBar.Find(p => p.Id == item.ParentId);

// if HasValue

if (navbarParent != null)

{

// Add to List Dropdown Item

navbarParent.subItems!.Add(

new NavbarItem()

{

Id = item.Id,

ParentId = item.ParentId,

Title = item.Title,

MenuUrl = item.MenuUrl,

MenuIndex = item.MenuIndex,

isVisible = item.isVisible,

subItems = null

}

);

}

}

}

}

Console.WriteLine(navBar);

return View("MenuDynamic", navBar);

}

}

}

Views/Shared/Components/MenuDynamic/MenuDynamic.cshtml

@model IEnumerable<ShopManager.Models.NavbarItem>

<div class="container-fluid fixed-top">

<div class="container topbar bg-primary d-none d-lg-block">

<div class="d-flex justify-content-between">

<div class="top-info ps-2">

<small class="me-3">

<i class="fas fa-map-marker-alt me-2 text-secondary"></i>

<a href="#" class="text-white">123 Street, New York</a>

</small>

<small class="me-3">

<i class="fas fa-envelope me-2 text-secondary"></i><a href="#" class="text-white">Email@Example.com</a>

</small>

</div>

<div class="top-link pe-2">

@if (@Context.User.Identity.IsAuthenticated)

{

<label style="font-weight:400; font-size:12px; color:white; margin-top:2px ">Xin chào</label>

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="Profile" class="text-white ">

<small class="text-white mx-2 " style="font-weight:bold">

@Context.User.Identity.Name

</small>/

</**a**>

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="SignOut" class="text-white font-weight-bold">

<small class="text-white mx-2" style="font-weight:bold">Đăng xuất</small>

</**a**>

}

else

{

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="SignIn" class="text-white"><small class="text-white mx-2">Đăng nhập</small>/</**a**>

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="SignUp" class="text-white"><small class="text-white mx-2">Đăng ký</small></**a**>

}

</div>

</div>

</div>

<div class="container px-0">

<nav class="navbar navbar-light bg-white navbar-expand-xl">

<a href="index.html" class="navbar-brand"><h1 class="text-primary display-6">Fruitables</h1></a>

<button class="navbar-toggler py-2 px-3"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarCollapse">

<span class="fa fa-bars text-primary"></span>

</button>

<div class="collapse navbar-collapse bg-white" id="navbarCollapse">

<div class="navbar-nav mx-auto">

@foreach (var item in Model)

{

if (item.subItems?.Count() == 0)

{

<a href="@item.MenuUrl"

class="nav-item nav-link

@(item!.MenuUrl == Context.Request.Path ? "active" : "")">

@item.Title

</a>

}

else

{

<div class="nav-item dropdown">

<a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">@item.Title</a>

<div class="dropdown-menu m-0 bg-secondary rounded-0">

@foreach (var subItem in item!.subItems!)

{

<a href="@subItem.MenuUrl" class="dropdown-item">@subItem.Title</a>

}

</div>

</div>

}

}

</div>

<div class="d-flex m-3 me-0">

<button class="btn-search btn border border-secondary btn-md-square rounded-circle bg-white me-4"

data-bs-toggle="modal"

data-bs-target="#searchModal">

<i class="fas fa-search text-primary"></i>

</button>

<a href="#" class="position-relative me-4 my-auto">

<i class="fa fa-shopping-bag fa-2x"></i>

<span class="position-absolute bg-secondary rounded-circle d-flex align-items-center justify-content-center text-dark px-1"

style="top: -5px; left: 15px; height: 20px; min-width: 20px">3</span>

</a>

<**a** **asp-area**="" **asp-controller**="Customer" **asp-action**="Profile" class="my-auto">

<i class="fas fa-user fa-2x"></i>

</**a**>

</div>

</div>

</nav>

</div>

</div>