using malshetwi\_CapstoneProject\_SDA.LMS.Data;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Static;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels;

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Identity;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Controllers

{

public class AccountController : Controller

{

private readonly UserManager<ApplicationUser> \_userManager;

private readonly SignInManager<ApplicationUser> \_signInManager;

private readonly AppDbContext \_context;

public AccountController(UserManager<ApplicationUser> userManager, SignInManager<ApplicationUser> signInManager, AppDbContext context)

{

\_userManager = userManager;

\_signInManager = signInManager;

\_context = context;

}// V.85

[Authorize(Roles = UserRoles.Admin)] // V.97

public async Task<IActionResult> Users()

{

var users = await \_context.Users.ToListAsync();

return View(users);

}

public IActionResult Login()

{

return View(new LoginVM());

}// V.86

[HttpPost]

public async Task<IActionResult> Login(LoginVM loginVM)

{

if (!ModelState.IsValid) return View(loginVM);

var user = await \_userManager.FindByEmailAsync(loginVM.EmailAddress);

if (user != null)

{

var PassCheck = await \_userManager.CheckPasswordAsync(user, loginVM.Password);

if (PassCheck)

{

var result = await \_signInManager.PasswordSignInAsync(user, loginVM.Password, false, false);

if (result.Succeeded)

{

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

}

}

TempData["Error"] = "Wrong credential, please try again!";

return View(loginVM);

}

TempData["Error"] = "Wrong credential, please try again!";

return View(loginVM);

}// V.87

public IActionResult Register()

{

return View(new RegisterVM());

}// V.88

[HttpPost]

public async Task<IActionResult> Register(RegisterVM registerVM)

{

if (!ModelState.IsValid) return View(registerVM);

var User = await \_userManager.FindByEmailAsync(registerVM.EmailAddress);

if (User != null)

{

TempData["Error"] = "Email address is already in use";

return View(registerVM);

}

var NewUser = new ApplicationUser()

{

FullName = registerVM.FullName,

Email = registerVM.EmailAddress,

UserName = registerVM.EmailAddress

};

var NewUserResponse = await \_userManager.CreateAsync(NewUser, registerVM.Password);

if (NewUserResponse.Succeeded)

{

await \_userManager.AddToRoleAsync(NewUser, UserRoles.User);

return View("RegisterCompleted");

}

TempData["Error"] = "Password must contain: At least 6 characters, OneUpperCase, OneLowerCase, OneNumber, Symbol as [$#@]"; // [TroubleShooting] when seed pass shoud provide asp.netPass\_requiremen [PassValidation] MSH

return View(registerVM);

}// V.89

[HttpPost]

public async Task<IActionResult> Logout()

{

await \_signInManager.SignOutAsync();

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

}// V.90

public IActionResult AccessDenied(string ReturnUrl)

{

return View();

} //v.97

//----MSH-----

public IActionResult DemoCredentials()

{

return View();

}

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Controllers

{

public class HomeController : Controller

{

private readonly ILogger<HomeController> \_logger;

public HomeController(ILogger<HomeController> logger)

{

\_logger = logger;

}

public IActionResult About()

{

return View();

}

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

public IActionResult Error()

{

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

}

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Services;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Static;

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Controllers

{

[Authorize(Roles = UserRoles.Admin)] // V.97

public class MedicinesController : Controller

{

private readonly IMedicinesService \_Service;

public MedicinesController(IMedicinesService Service)

{

\_Service = Service;

}

// <summary>

[AllowAnonymous] // V.96

public async Task<IActionResult> Index()

{

var Data = await \_Service.GetAll();

return View(Data);

}// End of GetAll </summary>

[AllowAnonymous] // V.96

public async Task<IActionResult> Filter(string searchString)

{

var Allmedicines = await \_Service.GetAll();

if (!string.IsNullOrEmpty(searchString))

{

var FilterResult = Allmedicines.Where(n => n.Name.Contains(searchString) || n.MedicineCategory.ToString().Contains(searchString)).ToList(); // Search by name, and by MedicineCategory to sort. MSH

if (FilterResult.Count != 0)

{

return View("Index", FilterResult);

}

TempData["Error"] = "Hmm no result, check letter case OR Sort by use category name"; // Optmize Code [+if&TempData] MSH

}

return View("Index",Allmedicines);

} // End of Filter V.61

// <summary> Get Medicine for CreateView

public IActionResult Create()

{

return View();

}

// Post CreateMedicine

[HttpPost]

public async Task<IActionResult> Create([Bind("ImageCode,Name,Description,Price,MedicineCategory")] Medicine medicine)

{

if (!ModelState.IsValid)

{

return View(medicine);

}

await \_Service.Add(medicine);

return RedirectToAction(nameof(Index));

}// End of Create Get&Post </summary>

// <summary> Get Medicine for GetEditView + [used to get for delete also]

public async Task<IActionResult> Edit(int id)

{

// <QA> lins insure data still in Db not just in UI \_ Case1:

var MedicineDetails = await \_Service.GetByID(id);

if(MedicineDetails == null ) return View ("NotFounded"); // </QA>

return View(MedicineDetails);

}

// Post EditMedicine

[HttpPost]

public async Task<IActionResult> Edit(int id,[Bind("ID,ImageCode,Name,Description,Price,MedicineCategory")] Medicine medicine)

{

/\*// <QA> lins insure data still in Db not just in UI \_ Case2:

var MedicineDetails = await \_Service.GetByID(id); // Error ocurred with QA lins Case2 | MSG\_[Model tracked by other instance]

if (MedicineDetails == null) return View("NotFounded"); // </QA> \*/

if (!ModelState.IsValid)

{

return View(medicine);

}

//Edit or Update

await \_Service.Update(id,medicine);

return RedirectToAction(nameof(Index));

}// End of Edit Get&Post </summary>

// <summary> Get Medicine to DeletetView Using EditView

// Post DeleteMedicine

public async Task<IActionResult> Delete(int id)

{

// <QA> lins insure data still in Db not just in UI \_ Case3:

var MedicineDetails = await \_Service.GetByID(id);

if (MedicineDetails == null) return View ("NotFounded"); // </QA>

await \_Service.Delete(id);

return RedirectToAction(nameof(Index));

}// End of Delete Post </summary>

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Cart;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Services;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels;

using Microsoft.AspNetCore.Authorization;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Security.Claims;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Controllers

{

[Authorize] // Vid.96

public class OrdersController : Controller

{

private readonly IMedicinesService \_medicinesService;

private readonly IOrdersService \_ordersService;

private readonly ShoppingCart \_ShoppingCart;

public OrdersController(IMedicinesService medicinesService, IOrdersService ordersService, ShoppingCart shoppingCart)

{

\_medicinesService = medicinesService;

\_ShoppingCart = shoppingCart;

\_ordersService = ordersService;

}

// Get Orders

public async Task<IActionResult> Index()

{

string userId = User.FindFirstValue(ClaimTypes.NameIdentifier);

string userRole = User.FindFirstValue(ClaimTypes.Role);

var Orders = await \_ordersService.GetOrdersByUserIdAndRole(userId, userRole);

return View(Orders);

} // Vid. 75, 94 Implement Identity

[AllowAnonymous] // Vid.96

public IActionResult ShoppingCart() // < as index of shoppingCart

{

var Items = \_ShoppingCart.GetShoppingCartItems();

\_ShoppingCart.ShoppingCartItems = Items;

var response = new ShoppingCartVM()

{

ShoppingCart = \_ShoppingCart,

ShoppingCartTotal = \_ShoppingCart.GetShoppingCartTotal()

};

return View(response); // Vid 67

}

public async Task<IActionResult> AddItemToShoppingCart(int id)

{

var item = await \_medicinesService.GetByID(id);

if (item != null)

{

\_ShoppingCart.AddItemToCart(item);

}

return RedirectToAction(nameof(ShoppingCart)); // Vid 70

}

public async Task<IActionResult> RemoveItemFromShoppingCart(int id)

{

var item = await \_medicinesService.GetByID(id);

if (item != null)

{

\_ShoppingCart.DeleteItemFromCart(item);

}

return RedirectToAction(nameof(ShoppingCart)); // Vid 71

}

public async Task<IActionResult> CompleteOrder()

{

var items = \_ShoppingCart.GetShoppingCartItems();

if (items.Count!=0)

{

string userId = User.FindFirstValue(ClaimTypes.NameIdentifier);

string userEmailAddress = User.FindFirstValue(ClaimTypes.Email);

await \_ordersService.StoreOrder(items, userId, userEmailAddress);

await \_ShoppingCart.ClearShoppingCart();

return View();

}

TempData["Error"] = "Trespassing detected, cart must contain item to complete order"; // <QA> code lines for security, +if & error msg MSH <QA/>

return RedirectToAction(nameof(ShoppingCart));

} // Vid 74, 94 Implement Identity

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Http;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.DependencyInjection;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Cart

{

public class ShoppingCart

{

public AppDbContext \_context { get; set; }

public string ShoppingCartID2 { get; set; } // Local ShoppingCartID

public List<ShoppingCartItem> ShoppingCartItems { get; set; }

public ShoppingCart(AppDbContext context)

{

\_context = context;

}

public static ShoppingCart GetShoppingCart(IServiceProvider services) // Initiate of Cart and Session

{ // ? if not null

ISession session = services.GetRequiredService<IHttpContextAccessor>()?.HttpContext.Session;

var context = services.GetService<AppDbContext>();

// ?? if null

string CartId = session.GetString("CartId") ?? Guid.NewGuid().ToString();

session.SetString("CartId", CartId);

return new ShoppingCart(context) { ShoppingCartID2 = CartId };

}

public void AddItemToCart(Medicine medicine)

{

var shoppingCartItem = \_context.shoppingCartItems.FirstOrDefault(n => n.Medicine.ID == medicine.ID

&& n.ShoppingCartID == ShoppingCartID2);

if(shoppingCartItem == null)

{

shoppingCartItem = new ShoppingCartItem()

{ ShoppingCartID = ShoppingCartID2,

Medicine = medicine,

Amount = 1};

\_context.shoppingCartItems.Add(shoppingCartItem);

}

else { shoppingCartItem.Amount++; }

\_context.SaveChanges(); // Vid 65

}

public void DeleteItemFromCart(Medicine medicine)

{

var shoppingCartItem = \_context.shoppingCartItems.FirstOrDefault(n => n.Medicine.ID == medicine.ID

&& n.ShoppingCartID == ShoppingCartID2);

if (shoppingCartItem != null)

{

if (shoppingCartItem.Amount > 1) { shoppingCartItem.Amount--; }

else { \_context.shoppingCartItems.Remove(shoppingCartItem);}

}

\_context.SaveChanges(); // Vid 66

}

public List<ShoppingCartItem> GetShoppingCartItems()

{

return ShoppingCartItems ?? (ShoppingCartItems = \_context.shoppingCartItems.Where(n => n.ShoppingCartID == ShoppingCartID2)

.Include(n => n.Medicine).ToList()); // Vid 64

}

public double GetShoppingCartTotal()

{

var Total = \_context.shoppingCartItems.Where(n => n.ShoppingCartID == ShoppingCartID2)

.Select(n => n.Medicine.Price \* n.Amount).Sum();

return Total; // Vid 64

}

public async Task ClearShoppingCart()

{

var items = await \_context.shoppingCartItems.Where(n => n.ShoppingCartID == ShoppingCartID2).ToListAsync();

\_context.shoppingCartItems.RemoveRange(items);

await \_context.SaveChangesAsync();

} // Vid 74

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data

{

public enum MedicineCategory

{

MumAndBaby,

BeautyCare,

PersonalCare,

MedicineAndTreatment,

Vitamins

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Services

{

public interface IMedicinesService

{

Task<IEnumerable<Medicine>> GetAll();

Task<Medicine> GetByID(int id);

Task Add(Medicine medicine);

Task<Medicine> Update(int id, Medicine newMedicine);

Task Delete(int id);

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Services

{

public interface IOrdersService

{

Task StoreOrder(List<ShoppingCartItem> items, string userId, string userEmailAddress);

Task<List<Order>> GetOrdersByUserIdAndRole(string userId, string userRole);

} //Vid. 73 & 94Implement Identity

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Services

{

public class MedicinesService : IMedicinesService

{

private readonly AppDbContext \_context;

public MedicinesService(AppDbContext context)

{

\_context = context;

}

public async Task Add(Medicine medicine)

{

await \_context.Medicines.AddAsync(medicine);

await \_context.SaveChangesAsync();

}

public async Task Delete(int id)

{

var result = await \_context.Medicines.FirstOrDefaultAsync(n => n.ID == id);

\_context.Medicines.Remove(result);

await \_context.SaveChangesAsync();

}

public async Task<IEnumerable<Medicine>> GetAll()

{

var result = await \_context.Medicines.ToListAsync();

return result;

}

public async Task<Medicine> GetByID(int id)

{

var result = await \_context.Medicines.FirstOrDefaultAsync(n => n.ID == id);

return result;

}

public async Task<Medicine> Update(int id, Medicine newMedicine)

{

\_context.Update(newMedicine);

await \_context.SaveChangesAsync();

return newMedicine;

}

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Services

{

public class OrdersService : IOrdersService

{

private readonly AppDbContext \_context;

public OrdersService(AppDbContext context)

{

\_context = context;

}

public async Task<List<Order>> GetOrdersByUserIdAndRole(string userId, string userRole)

{

var orders = await \_context.Orders.Include(n => n.OrderItems).ThenInclude(n => n.Medicine).Include(n => n.User).ToListAsync();

if (userRole != "Admin")

{

orders = orders.Where(n => n.UserID == userId).ToList();

}

return orders;

} // Vid. 73 & 94 Implement Identity

public async Task StoreOrder(List<ShoppingCartItem> items, string userId, string userEmailAddress)

{

var order = new Order()

{

UserID = userId,

Email = userEmailAddress,

};

await \_context.Orders.AddAsync(order);

await \_context.SaveChangesAsync();

foreach (var item in items)

{

var orderitem = new OrderItem()

{

Amount = item.Amount,

Price = item.Medicine.Price,

MedicineID = item.Medicine.ID,

OrderID= order.ID

};

await \_context.OrderItems.AddAsync(orderitem);

}

await \_context.SaveChangesAsync();

}// Vid 73

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.Static

{

public static class UserRoles

{

public const string Admin = "Admin";

public const string User = "User";

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Cart;

using Microsoft.AspNetCore.Mvc;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewComponents

{

public class ShoppingCartSummary : ViewComponent

{

private readonly ShoppingCart \_shoppingCart;

public ShoppingCartSummary(ShoppingCart shoppingCart)

{

\_shoppingCart = shoppingCart;

}

public IViewComponentResult Invoke()

{

var items = \_shoppingCart.GetShoppingCartItems();

return View (items.Count);

}

} // Vid.72

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels

{

public class LoginVM

{

[Display(Name = "Email")]

[Required(ErrorMessage ="Email is Required")]

public string EmailAddress { get; set; }

[Required]

[DataType(DataType.Password)]

public string Password { get; set; }

}// V.86

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels

{

public class RegisterVM

{

[Display(Name = "FullName")]

[Required(ErrorMessage = "FullName is Required")]

public string FullName { get; set; }

[Display(Name = "Email")]

[Required(ErrorMessage ="Email is Required")]

public string EmailAddress { get; set; }

[Required]

[DataType(DataType.Password)]

public string Password { get; set; }

[Display(Name = "Confirm Password")]

[Required(ErrorMessage = "Confirm Password is Required")]

[DataType(DataType.Password)]

[Compare("Password", ErrorMessage ="Passwords don't match")]

public string ConfirmPassword { get; set; }

}// V.88

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Cart;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels

{

public class ShoppingCartVM

{

public ShoppingCart ShoppingCart { get; set; }

public double ShoppingCartTotal { get; set; }

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data

{

public class AppDbContext:IdentityDbContext<ApplicationUser>

{

public AppDbContext(DbContextOptions<AppDbContext> options): base (options)

{

}

public DbSet<Medicine> Medicines { get; set; }

// Order Related tables

public DbSet<Order> Orders { get; set; }

public DbSet<OrderItem> OrderItems { get; set; }

public DbSet<ShoppingCartItem> shoppingCartItems { get; set; }

//~

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Static;

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Identity;

using Microsoft.AspNetCore.Identity.EntityFrameworkCore;

using Microsoft.Extensions.DependencyInjection;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Data

{

public class AppDbInitializer

{

public static void Seed(IApplicationBuilder applicationBuilder)

{

using (var serviceScop = applicationBuilder.ApplicationServices.CreateScope())

{

var context = serviceScop.ServiceProvider.GetService<AppDbContext>();

context.Database.EnsureCreated();

// Medicines

if (!context.Medicines.Any())

{

context.Medicines.AddRange(new List<Medicine>() {

new Medicine()

{

ImageCode = 1,

Name = "Metanium Everyday Barrier Ointment 40gm",

Description="Brand: Metanium, PROTECT and PREVENT the baby from nappy rash, delicate of skin",

Price=45.77,

MedicineCategory=MedicineCategory.MumAndBaby},

new Medicine()

{

ImageCode = 2,

Name = "Biomagic Hair Color Cream",

Description="Brand: BioMagic, Keratin & Argan Oil, 8.72 Light Beige Blonde Color, Free From Ammonia - 1 Kit",

Price=74.75,

MedicineCategory=MedicineCategory.BeautyCare},

new Medicine()

{

ImageCode = 3,

Name = "Apricot Scrub",

Description="Brand: Nature'S Bounty, Apricot Scrub Gently Eliminates Dead Skin Cells And Roughness",

Price=33.50,

MedicineCategory=MedicineCategory.PersonalCare},

new Medicine()

{

ImageCode = 4,

Name = "Panadol Blue",

Description="Brand: Panadol, Panadol Blue Paracetamol 500 Mg - 24 Tabs",

Price=5.00,

MedicineCategory=MedicineCategory.MedicineAndTreatment},

new Medicine()

{

ImageCode = 5,

Name = "Paradox Omega 3",

Description="Brand: Paradox, Omega 3 Chews for School Kids, suitable for ages 5 - 18",

Price=93.00,

MedicineCategory=MedicineCategory.Vitamins},

new Medicine()

{

ImageCode = 6,

Name = "Nivea After Shave Balm Replenishing",

Description="Deeply moisturizer Lotion helps prevent dry and tight skin delivers a long-lasting",

Price=31.76,

MedicineCategory=MedicineCategory.PersonalCare},

new Medicine()

{

ImageCode = 7,

Name = "King.C.Gillette, Beard Trimmer",

Description="King.C.Gillette, Beard Trimmer 4 Accessories - 1 Kit",

Price=151.86,

MedicineCategory=MedicineCategory.PersonalCare},

new Medicine()

{

ImageCode = 8,

Name = "Betika, Ointment, Intensive Care",

Description="Improve Skin Appearance, For Wound Healing - 60 Gm",

Price=79.35,

MedicineCategory=MedicineCategory.MedicineAndTreatment},

new Medicine()

{

ImageCode = 9,

Name = "Biodal 50.000, Vitamin D3",

Description="Biodal 50.000 Iu 20 Tab, Vitamin D, Storage conditions: store below 30°c",

Price=64.1,

MedicineCategory=MedicineCategory.Vitamins}

});

context.SaveChanges();

}

}

}

public static async Task SeedUsersAndRoles(IApplicationBuilder applicationBuilder)

{

using (var serviceScop = applicationBuilder.ApplicationServices.CreateScope())

{

// Roles

var roleManager = serviceScop.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>(); // Used default IdentityRole V.83

if (!await roleManager.RoleExistsAsync(UserRoles.Admin))

await roleManager.CreateAsync(new IdentityRole(UserRoles.Admin));

if (!await roleManager.RoleExistsAsync(UserRoles.User))

await roleManager.CreateAsync(new IdentityRole(UserRoles.User));

//~

// Users | Admin

var userManager = serviceScop.ServiceProvider.GetRequiredService<UserManager<ApplicationUser>>(); // Created model customized inheret from IdentityUserRole V.83

string AdminUserEmail= "admin@capstone.com";

var adminUser = await userManager.FindByEmailAsync(AdminUserEmail);

if (adminUser == null)

{

var newAdminUser = new ApplicationUser

{

FullName = "Admin",

UserName = "admin-User",

Email = AdminUserEmail,

EmailConfirmed = true,

LockoutEnabled = false

};

await userManager.CreateAsync(newAdminUser, "!Admin1"); // [TroubleShooting] when seed pass shoud provide pass\_requiremen otherwice get FK\_Conflect at FK\_AspNetUserRoles MSH

await userManager.AddToRoleAsync(newAdminUser,UserRoles.Admin);

}//~

// Users | User

string XUserEmail = "xuser@capstone.com";

var XUser = await userManager.FindByEmailAsync(XUserEmail);

if (XUser == null)

{

var newXUser = new ApplicationUser

{

FullName = "User No.0",

UserName = "Xuser-User",

Email = XUserEmail,

EmailConfirmed = true,

LockoutEnabled = false

};

await userManager.CreateAsync(newXUser, "!User1");

await userManager.AddToRoleAsync(newXUser, UserRoles.User);

}//~

}

}

}

}

using Microsoft.EntityFrameworkCore.Migrations;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Migrations

{

public partial class initial : Migration

{

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "Medicines",

columns: table => new

{

ID = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

ImageCode = table.Column<int>(type: "int", nullable: false),

Name = table.Column<string>(type: "nvarchar(max)", nullable: true),

Description = table.Column<string>(type: "nvarchar(max)", nullable: true),

Price = table.Column<double>(type: "float", nullable: false),

MedicineCategory = table.Column<int>(type: "int", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_Medicines", x => x.ID);

});

}

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "Medicines");

}

}

}

using Microsoft.EntityFrameworkCore.Migrations;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Migrations

{

public partial class Order\_and\_OerderItem\_added : Migration

{

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.AlterColumn<string>(

name: "Name",

table: "Medicines",

type: "nvarchar(40)",

maxLength: 40,

nullable: false,

defaultValue: "",

oldClrType: typeof(string),

oldType: "nvarchar(max)",

oldNullable: true);

migrationBuilder.AlterColumn<string>(

name: "Description",

table: "Medicines",

type: "nvarchar(max)",

nullable: false,

defaultValue: "",

oldClrType: typeof(string),

oldType: "nvarchar(max)",

oldNullable: true);

migrationBuilder.CreateTable(

name: "Orders",

columns: table => new

{

ID = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

Email = table.Column<string>(type: "nvarchar(max)", nullable: true),

UserID = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_Orders", x => x.ID);

});

migrationBuilder.CreateTable(

name: "OrderItems",

columns: table => new

{

ID = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

Amount = table.Column<int>(type: "int", nullable: false),

Price = table.Column<double>(type: "float", nullable: false),

MedicineID = table.Column<int>(type: "int", nullable: false),

OrderID = table.Column<int>(type: "int", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_OrderItems", x => x.ID);

table.ForeignKey(

name: "FK\_OrderItems\_Medicines\_MedicineID",

column: x => x.MedicineID,

principalTable: "Medicines",

principalColumn: "ID",

onDelete: ReferentialAction.Cascade);

table.ForeignKey(

name: "FK\_OrderItems\_Orders\_OrderID",

column: x => x.OrderID,

principalTable: "Orders",

principalColumn: "ID",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateIndex(

name: "IX\_OrderItems\_MedicineID",

table: "OrderItems",

column: "MedicineID");

migrationBuilder.CreateIndex(

name: "IX\_OrderItems\_OrderID",

table: "OrderItems",

column: "OrderID");

}

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "OrderItems");

migrationBuilder.DropTable(

name: "Orders");

migrationBuilder.AlterColumn<string>(

name: "Name",

table: "Medicines",

type: "nvarchar(max)",

nullable: true,

oldClrType: typeof(string),

oldType: "nvarchar(40)",

oldMaxLength: 40);

migrationBuilder.AlterColumn<string>(

name: "Description",

table: "Medicines",

type: "nvarchar(max)",

nullable: true,

oldClrType: typeof(string),

oldType: "nvarchar(max)");

}

}

}

using Microsoft.EntityFrameworkCore.Migrations;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Migrations

{

public partial class ShoppingCartItem\_added : Migration

{

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "shoppingCartItems",

columns: table => new

{

ID = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

MedicineID = table.Column<int>(type: "int", nullable: true),

Amount = table.Column<int>(type: "int", nullable: false),

ShoppingCartID = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_shoppingCartItems", x => x.ID);

table.ForeignKey(

name: "FK\_shoppingCartItems\_Medicines\_MedicineID",

column: x => x.MedicineID,

principalTable: "Medicines",

principalColumn: "ID",

onDelete: ReferentialAction.Restrict);

});

migrationBuilder.CreateIndex(

name: "IX\_shoppingCartItems\_MedicineID",

table: "shoppingCartItems",

column: "MedicineID");

}

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "shoppingCartItems");

}

}

}

using System;

using Microsoft.EntityFrameworkCore.Migrations;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Migrations

{

public partial class Add\_AppUser : Migration

{

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "AspNetRoles",

columns: table => new

{

Id = table.Column<string>(type: "nvarchar(450)", nullable: false),

Name = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

NormalizedName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetRoles", x => x.Id);

});

migrationBuilder.CreateTable(

name: "AspNetUsers",

columns: table => new

{

Id = table.Column<string>(type: "nvarchar(450)", nullable: false),

FullName = table.Column<string>(type: "nvarchar(max)", nullable: true),

UserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

NormalizedUserName = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

Email = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

NormalizedEmail = table.Column<string>(type: "nvarchar(256)", maxLength: 256, nullable: true),

EmailConfirmed = table.Column<bool>(type: "bit", nullable: false),

PasswordHash = table.Column<string>(type: "nvarchar(max)", nullable: true),

SecurityStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),

ConcurrencyStamp = table.Column<string>(type: "nvarchar(max)", nullable: true),

PhoneNumber = table.Column<string>(type: "nvarchar(max)", nullable: true),

PhoneNumberConfirmed = table.Column<bool>(type: "bit", nullable: false),

TwoFactorEnabled = table.Column<bool>(type: "bit", nullable: false),

LockoutEnd = table.Column<DateTimeOffset>(type: "datetimeoffset", nullable: true),

LockoutEnabled = table.Column<bool>(type: "bit", nullable: false),

AccessFailedCount = table.Column<int>(type: "int", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUsers", x => x.Id);

});

migrationBuilder.CreateTable(

name: "AspNetRoleClaims",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

RoleId = table.Column<string>(type: "nvarchar(450)", nullable: false),

ClaimType = table.Column<string>(type: "nvarchar(max)", nullable: true),

ClaimValue = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetRoleClaims", x => x.Id);

table.ForeignKey(

name: "FK\_AspNetRoleClaims\_AspNetRoles\_RoleId",

column: x => x.RoleId,

principalTable: "AspNetRoles",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

name: "AspNetUserClaims",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

UserId = table.Column<string>(type: "nvarchar(450)", nullable: false),

ClaimType = table.Column<string>(type: "nvarchar(max)", nullable: true),

ClaimValue = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserClaims", x => x.Id);

table.ForeignKey(

name: "FK\_AspNetUserClaims\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

name: "AspNetUserLogins",

columns: table => new

{

LoginProvider = table.Column<string>(type: "nvarchar(450)", nullable: false),

ProviderKey = table.Column<string>(type: "nvarchar(450)", nullable: false),

ProviderDisplayName = table.Column<string>(type: "nvarchar(max)", nullable: true),

UserId = table.Column<string>(type: "nvarchar(450)", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserLogins", x => new { x.LoginProvider, x.ProviderKey });

table.ForeignKey(

name: "FK\_AspNetUserLogins\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

name: "AspNetUserRoles",

columns: table => new

{

UserId = table.Column<string>(type: "nvarchar(450)", nullable: false),

RoleId = table.Column<string>(type: "nvarchar(450)", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserRoles", x => new { x.UserId, x.RoleId });

table.ForeignKey(

name: "FK\_AspNetUserRoles\_AspNetRoles\_RoleId",

column: x => x.RoleId,

principalTable: "AspNetRoles",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

table.ForeignKey(

name: "FK\_AspNetUserRoles\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateTable(

name: "AspNetUserTokens",

columns: table => new

{

UserId = table.Column<string>(type: "nvarchar(450)", nullable: false),

LoginProvider = table.Column<string>(type: "nvarchar(450)", nullable: false),

Name = table.Column<string>(type: "nvarchar(450)", nullable: false),

Value = table.Column<string>(type: "nvarchar(max)", nullable: true)

},

constraints: table =>

{

table.PrimaryKey("PK\_AspNetUserTokens", x => new { x.UserId, x.LoginProvider, x.Name });

table.ForeignKey(

name: "FK\_AspNetUserTokens\_AspNetUsers\_UserId",

column: x => x.UserId,

principalTable: "AspNetUsers",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateIndex(

name: "IX\_AspNetRoleClaims\_RoleId",

table: "AspNetRoleClaims",

column: "RoleId");

migrationBuilder.CreateIndex(

name: "RoleNameIndex",

table: "AspNetRoles",

column: "NormalizedName",

unique: true,

filter: "[NormalizedName] IS NOT NULL");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserClaims\_UserId",

table: "AspNetUserClaims",

column: "UserId");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserLogins\_UserId",

table: "AspNetUserLogins",

column: "UserId");

migrationBuilder.CreateIndex(

name: "IX\_AspNetUserRoles\_RoleId",

table: "AspNetUserRoles",

column: "RoleId");

migrationBuilder.CreateIndex(

name: "EmailIndex",

table: "AspNetUsers",

column: "NormalizedEmail");

migrationBuilder.CreateIndex(

name: "UserNameIndex",

table: "AspNetUsers",

column: "NormalizedUserName",

unique: true,

filter: "[NormalizedUserName] IS NOT NULL");

}

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "AspNetRoleClaims");

migrationBuilder.DropTable(

name: "AspNetUserClaims");

migrationBuilder.DropTable(

name: "AspNetUserLogins");

migrationBuilder.DropTable(

name: "AspNetUserRoles");

migrationBuilder.DropTable(

name: "AspNetUserTokens");

migrationBuilder.DropTable(

name: "AspNetRoles");

migrationBuilder.DropTable(

name: "AspNetUsers");

}

}

}

// <auto-generated />

using System;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Infrastructure;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

using malshetwi\_CapstoneProject\_SDA.LMS.Data;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Migrations

{

[DbContext(typeof(AppDbContext))]

partial class AppDbContextModelSnapshot : ModelSnapshot

{

protected override void BuildModel(ModelBuilder modelBuilder)

{

#pragma warning disable 612, 618

modelBuilder

.HasAnnotation("Relational:MaxIdentifierLength", 128)

.HasAnnotation("ProductVersion", "5.0.10")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityRole", b =>

{

b.Property<string>("Id")

.HasColumnType("nvarchar(450)");

b.Property<string>("ConcurrencyStamp")

.IsConcurrencyToken()

.HasColumnType("nvarchar(max)");

b.Property<string>("Name")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.Property<string>("NormalizedName")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.HasKey("Id");

b.HasIndex("NormalizedName")

.IsUnique()

.HasDatabaseName("RoleNameIndex")

.HasFilter("[NormalizedName] IS NOT NULL");

b.ToTable("AspNetRoles");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityRoleClaim<string>", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<string>("ClaimType")

.HasColumnType("nvarchar(max)");

b.Property<string>("ClaimValue")

.HasColumnType("nvarchar(max)");

b.Property<string>("RoleId")

.IsRequired()

.HasColumnType("nvarchar(450)");

b.HasKey("Id");

b.HasIndex("RoleId");

b.ToTable("AspNetRoleClaims");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserClaim<string>", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<string>("ClaimType")

.HasColumnType("nvarchar(max)");

b.Property<string>("ClaimValue")

.HasColumnType("nvarchar(max)");

b.Property<string>("UserId")

.IsRequired()

.HasColumnType("nvarchar(450)");

b.HasKey("Id");

b.HasIndex("UserId");

b.ToTable("AspNetUserClaims");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserLogin<string>", b =>

{

b.Property<string>("LoginProvider")

.HasColumnType("nvarchar(450)");

b.Property<string>("ProviderKey")

.HasColumnType("nvarchar(450)");

b.Property<string>("ProviderDisplayName")

.HasColumnType("nvarchar(max)");

b.Property<string>("UserId")

.IsRequired()

.HasColumnType("nvarchar(450)");

b.HasKey("LoginProvider", "ProviderKey");

b.HasIndex("UserId");

b.ToTable("AspNetUserLogins");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserRole<string>", b =>

{

b.Property<string>("UserId")

.HasColumnType("nvarchar(450)");

b.Property<string>("RoleId")

.HasColumnType("nvarchar(450)");

b.HasKey("UserId", "RoleId");

b.HasIndex("RoleId");

b.ToTable("AspNetUserRoles");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserToken<string>", b =>

{

b.Property<string>("UserId")

.HasColumnType("nvarchar(450)");

b.Property<string>("LoginProvider")

.HasColumnType("nvarchar(450)");

b.Property<string>("Name")

.HasColumnType("nvarchar(450)");

b.Property<string>("Value")

.HasColumnType("nvarchar(max)");

b.HasKey("UserId", "LoginProvider", "Name");

b.ToTable("AspNetUserTokens");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.ApplicationUser", b =>

{

b.Property<string>("Id")

.HasColumnType("nvarchar(450)");

b.Property<int>("AccessFailedCount")

.HasColumnType("int");

b.Property<string>("ConcurrencyStamp")

.IsConcurrencyToken()

.HasColumnType("nvarchar(max)");

b.Property<string>("Email")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.Property<bool>("EmailConfirmed")

.HasColumnType("bit");

b.Property<string>("FullName")

.HasColumnType("nvarchar(max)");

b.Property<bool>("LockoutEnabled")

.HasColumnType("bit");

b.Property<DateTimeOffset?>("LockoutEnd")

.HasColumnType("datetimeoffset");

b.Property<string>("NormalizedEmail")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.Property<string>("NormalizedUserName")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.Property<string>("PasswordHash")

.HasColumnType("nvarchar(max)");

b.Property<string>("PhoneNumber")

.HasColumnType("nvarchar(max)");

b.Property<bool>("PhoneNumberConfirmed")

.HasColumnType("bit");

b.Property<string>("SecurityStamp")

.HasColumnType("nvarchar(max)");

b.Property<bool>("TwoFactorEnabled")

.HasColumnType("bit");

b.Property<string>("UserName")

.HasMaxLength(256)

.HasColumnType("nvarchar(256)");

b.HasKey("Id");

b.HasIndex("NormalizedEmail")

.HasDatabaseName("EmailIndex");

b.HasIndex("NormalizedUserName")

.IsUnique()

.HasDatabaseName("UserNameIndex")

.HasFilter("[NormalizedUserName] IS NOT NULL");

b.ToTable("AspNetUsers");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.Medicine", b =>

{

b.Property<int>("ID")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<string>("Description")

.IsRequired()

.HasColumnType("nvarchar(max)");

b.Property<int>("ImageCode")

.HasColumnType("int");

b.Property<int>("MedicineCategory")

.HasColumnType("int");

b.Property<string>("Name")

.IsRequired()

.HasMaxLength(40)

.HasColumnType("nvarchar(40)");

b.Property<double>("Price")

.HasColumnType("float");

b.HasKey("ID");

b.ToTable("Medicines");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.Order", b =>

{

b.Property<int>("ID")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<string>("Email")

.HasColumnType("nvarchar(max)");

b.Property<string>("UserID")

.HasColumnType("nvarchar(max)");

b.HasKey("ID");

b.ToTable("Orders");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.OrderItem", b =>

{

b.Property<int>("ID")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<int>("Amount")

.HasColumnType("int");

b.Property<int>("MedicineID")

.HasColumnType("int");

b.Property<int>("OrderID")

.HasColumnType("int");

b.Property<double>("Price")

.HasColumnType("float");

b.HasKey("ID");

b.HasIndex("MedicineID");

b.HasIndex("OrderID");

b.ToTable("OrderItems");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.ShoppingCartItem", b =>

{

b.Property<int>("ID")

.ValueGeneratedOnAdd()

.HasColumnType("int")

.HasAnnotation("SqlServer:ValueGenerationStrategy", SqlServerValueGenerationStrategy.IdentityColumn);

b.Property<int>("Amount")

.HasColumnType("int");

b.Property<int?>("MedicineID")

.HasColumnType("int");

b.Property<string>("ShoppingCartID")

.HasColumnType("nvarchar(max)");

b.HasKey("ID");

b.HasIndex("MedicineID");

b.ToTable("shoppingCartItems");

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityRoleClaim<string>", b =>

{

b.HasOne("Microsoft.AspNetCore.Identity.IdentityRole", null)

.WithMany()

.HasForeignKey("RoleId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserClaim<string>", b =>

{

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.ApplicationUser", null)

.WithMany()

.HasForeignKey("UserId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserLogin<string>", b =>

{

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.ApplicationUser", null)

.WithMany()

.HasForeignKey("UserId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserRole<string>", b =>

{

b.HasOne("Microsoft.AspNetCore.Identity.IdentityRole", null)

.WithMany()

.HasForeignKey("RoleId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.ApplicationUser", null)

.WithMany()

.HasForeignKey("UserId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

});

modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserToken<string>", b =>

{

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.ApplicationUser", null)

.WithMany()

.HasForeignKey("UserId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.OrderItem", b =>

{

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.Medicine", "Medicine")

.WithMany()

.HasForeignKey("MedicineID")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.Order", "Order")

.WithMany("OrderItems")

.HasForeignKey("OrderID")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.Navigation("Medicine");

b.Navigation("Order");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.ShoppingCartItem", b =>

{

b.HasOne("malshetwi\_CapstoneProject\_SDA.LMS.Models.Medicine", "Medicine")

.WithMany()

.HasForeignKey("MedicineID");

b.Navigation("Medicine");

});

modelBuilder.Entity("malshetwi\_CapstoneProject\_SDA.LMS.Models.Order", b =>

{

b.Navigation("OrderItems");

});

#pragma warning restore 612, 618

}

}

}

using Microsoft.AspNetCore.Identity;

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class ApplicationUser:IdentityUser

{

[Display(Name = "Full name")]

public string FullName { get; set; }

}

}

using System;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class ErrorViewModel

{

public string RequestId { get; set; }

public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data;

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class Medicine

{

[Key]

public int ID { get; set; }

[Display(Name= "Medicine image name")]

[Required (ErrorMessage ="Name of image is required")]

public int ImageCode { get; set; }

[Display(Name = "Medicine/Item name")]

[Required(ErrorMessage = "Name of Medicine is required")]

[StringLength(40, MinimumLength = 3, ErrorMessage = "Name must be between 3 and 40 chars")] // V:32

public string Name { get; set; }

[Required(ErrorMessage = "Description is required")]

public string Description { get; set; }

[Required(ErrorMessage = "Price is required")]

public double Price { get; set; }

[Required(ErrorMessage = "Choose category")]

public MedicineCategory MedicineCategory { get; set; }

}

}

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class Order

{

[Key]

public int ID { get; set; }

public string Email { get; set; }

public string UserID { get; set; }

[ForeignKey(nameof(UserID))]

public ApplicationUser User { get; set; }

// ArrayList items = new ArrayList(); // Test\_Algorithem from me

public List<OrderItem> OrderItems { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations.Schema;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class OrderItem

{

public int ID { get; set; }

public int Amount { get; set; }

public double Price { get; set; }

//

public int MedicineID { get; set; }

[ForeignKey("MedicineID")]

public Medicine Medicine { get; set; }

//

public int OrderID { get; set; }

[ForeignKey("OrderID")]

public Order Order { get; set; }

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS.Models

{

public class ShoppingCartItem

{

[Key]

public int ID { get; set; }

public Medicine Medicine { get; set; }

public int Amount { get; set; }

public string ShoppingCartID { get; set; }

}

}

@{

ViewData["Title"] = "403 | Access Denied";

}

<div class="col-md-12 text-md-center">

<**img** src="~/Images/UX\_UI\_Design/403.png" style="height:100%;width:38%" />

<h2><span style="color:orangered"><b>403 ACCESS DENIED !!</b></span></h2>

<h1 style="color:darkred">The page you're trying to access has restricted access</h1>

<h2 style="color:cornflowerblue"><i class="bi bi-exclamation-triangle-fill"></i> Please refer to your system administrator</h2><br />

<**a** class="btn btn-outline-info" **asp-controller**="Medicines" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back, Home page</i></**a**>

</div>

@{

ViewData["Title"] = "Demo credentials";

}

<div class="row">

<div class="col-md-8 offset-2" style="z-index:1">

<h1><b>@ViewData["Title"]</b></h1>

<br />

<h2>Manual of credentials, for Demo use</h2>

<h3 style="color:cornflowerblue"><i class="bi bi-info-circle-fill"></i> Default credentials seeded and created automatically</h3>

<hr>

<br />

<br />

<h5><b>Admin user credential</b></h5>

<div class="form-group">

<label class="control-label">Email:</label>

<label class="control-label">admin@capstone.com</label>

<br />

<label class="control-label">Password:</label>

<label class="control-label">!Admin1</label>

</div>

<hr>

<h5><b>Normal/Standard user credential</b></h5>

<div class="form-group">

<label class="control-label">Email:</label>

<label class="control-label">xuser@capstone.com</label>

<br />

<label class="control-label">Password:</label>

<label class="control-label">!User1</label>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/GitHub\_malshetwi.png" style="height: 46%; width: 26%; position: fixed; right: 220px; bottom: 27.9px; z-index: 0;" /></div>

@using malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels;

@model LoginVM

@{

ViewData["Title"] = "Log in";

}

<div class="row">

<div class="col-md-6 offset-3" >

<p>

<h4>Login in to your account</h4>

</p>

@if (TempData["Error"] != null)

{

<div class="col-md-12 alert alert-danger"><span><b>Sorry!</b> - @TempData["Error"]</span></div>

}

<div class="row">

<div class="col-md-8 offset-2">

<**form** **asp-action**="Login"> @\*// forms by default perform request post\*@

<**div** **asp-validation-summary**="ModelOnly"></**div**>

<div class="form-group">

<**label** **asp-for**="EmailAddress" class="control-label"></**label**><br />

<**input** **asp-for**="EmailAddress" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Password" class="control-label"></**label**><br />

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

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

</div>

<div class="form-group">

<input type="submit" value="Login" class="btn btn-outline-success float-right" />

<**a** class="btn btn-outline-info" **asp-controller**="Medicines" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Cancel</i></**a**>

<**a** class="text-info" **asp-controller**="Account" **asp-action**="DemoCredentials">Demo credentials</**a**>

</div>

</**form**>

</div>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Login.png" style="height:100%;width: 40%" /></div>

@using malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels;

@model RegisterVM

@{

ViewData["Title"] = "Sign up";

}

<div class="row">

<div class="col-md-6 offset-3" >

<p>

<h4>Sign up to your account</h4>

</p>

@if (TempData["Error"] != null)

{

<div class="col-md-12 alert alert-danger"><span><b>Sorry!</b> - @TempData["Error"]</span></div>

}

<div class="row">

<div class="col-md-8 offset-2" style="z-index:1">

@\*// forms by default perform request post\*@

<**form** **asp-action**="Register">

<**div** **asp-validation-summary**="ModelOnly"></**div**>

<div class="form-group">

<**label** **asp-for**="FullName" class="control-label"></**label**><br />

<**input** **asp-for**="FullName" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="EmailAddress" class="control-label"></**label**><br />

<**input** **asp-for**="EmailAddress" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Password" class="control-label"></**label**><br />

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

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

</div>

<div class="form-group">

<**label** **asp-for**="ConfirmPassword" class="control-label"></**label**><br />

<**input** **asp-for**="ConfirmPassword" class="form-control" />

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

</div>

<div class="form-group">

<input type="submit" value="Sign up" class="btn btn-success float-right" />

<**a** class="btn btn-outline-info" **asp-controller**="Medicines" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Cancel</i></**a**>

</div>

</**form**>

</div>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Register.png" style="height: 46%; width: 26%; position: fixed; right: 220px; bottom: 27.9px; z-index: 0;"/></div>

@{

ViewData["Title"] = "Registration Completed";

}

<br />

<div class="row">

<div class="col-md-6 offset-3 alert alert-success">

<h1 class="text-center"><i class="bi bi-person-badge"></i></h1>

<h2><i class="bi bi-check-square-fill"></i> Registration Completed Successfully!</h2>

<p class="h4">You can log in now using your credentials</p><br />

<hr />

<p class="h4 font-italic">Welcome !</p>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/RegisterCompleted.png" style="height:100%;width: 40%" /></div>

@model IEnumerable<ApplicationUser>

@{

ViewData["Title"] = "Application Users";

}

<div class="row">

<div class="col-md-8 offset-2">

<p><h4>List of all Application Users</h4></p>

<table class="table">

<thead>

<tr>

<th>User Name</th>

<th>Full Name</th>

<th>Role</th>

</tr>

</thead>

<tbody>

@foreach (var user in Model)

{

<tr>

<td class="align-middle">@user.UserName</td>

<td>@user.FullName</td>

@if (user.FullName == ("Admin"))

{

<td>Admin</td>

}

else

{

<td>User</td>

}

</tr>

}

</tbody>

<tfoot>

<tr>

<td colspan="4"></td>

</tr>

</tfoot>

</table>

<**a** class="btn btn-outline-info" **asp-controller**="Medicines" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back Home Page</i></**a**>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Users.png" style="height:100%;width: 40%" /></div>

</div>

@{

ViewData["Title"] = "About";

}

<div class="row">

<div class="col-md-8 offset-2" style="z-index:1">

<h1>@ViewData["Title"]</h1>

<div>

<br />

<h4>ABC-Healthcare</h4>

<br>

<h5 sstyle="color:brown">

ABC Healthcare is a Mumbai-based pharma company which sells medicines<br /> throughout India as per the

demand it receives from multiple hospitals and<br /> clinics.It has been operating in India for the last 15 years.

It emerges as the<br />one of leading Health care across the Globe.It also bags the

2nd best<br /> HealthCare in India.The Mumbai peoples have Best services over the <br />years.

</h5>

<br>

<br>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/logo.png" style="height: 36%; width: 20%; position: fixed; right: 220px; bottom: 27.9px; z-index: 0;"/></div>

@model Medicine;

@using malshetwi\_CapstoneProject\_SDA.LMS.Data;

@{

ViewData["Title"] = "Add new Medicine item";

}

<div class="row text">

<div class="col-md-8 offset-2" style="z-index:1">

<p>

<h1>Add a New Medicine item</h1>

</p>

<br />

<div class="row">

<div class="col-md-8 offset-2">

<**form** **asp-action**="Create">

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

<div class="form-group">

<div class="alert alert-warning col-md-12"><i class="bi bi-exclamation-triangle-fill"></i> Write image name (as Integer), save it at ~/Images/UX\_UI\_Design/ file with png extension</div>

<**label** **asp-for**="ImageCode" class="control-label"></**label**><br />

<**input** **asp-for**="ImageCode" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Name" class="control-label"></**label**><br />

<**input** **asp-for**="Name" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Description" class="control-label"></**label**><br />

<**input** **asp-for**="Description" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="MedicineCategory" class="control-label"></**label**><br />

<**select** **asp-for**="MedicineCategory" class="form-control" **asp-items**="Html.GetEnumSelectList<MedicineCategory>()"></**select**>

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

</div>

<div class="form-group">

<**label** **asp-for**="Price" class="control-label"></**label**><br />

<**input** **asp-for**="Price" class="form-control" />

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

</div>

<div class="form-group">

<input type="submit" value="ADD Item" class="btn btn-outline-success float-right" />

<**a** class="btn btn-outline-info" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back, All Items</i></**a**>

</div>

</**form**>

</div>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/New\_Item.png" style="height: 49%; width: 28%; position: fixed; right: 30px; bottom: 15px; z-index: 0;" /></div>

@model Medicine;

@using malshetwi\_CapstoneProject\_SDA.LMS.Data;

@{

ViewData["Title"] = "Edit Medicine item";

}

<div class="row text">

<div class="col-md-8 offset-2" style="z-index:1">

<p>

<h1>Edit or Delete Medicine item</h1>

</p>

<br />

<div class="row">

<div class="col-md-8 offset-2">

<**form** **asp-action**="Edit">

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

<div class="form-group">

<div class="alert alert-warning col-md-12"><i class="bi bi-exclamation-triangle-fill"></i> Write image name (as Integer), save it at ~/Images/UX\_UI\_Design/ file with png extension</div>

<**label** **asp-for**="ImageCode" class="control-label"></**label**><br />

<**input** **asp-for**="ImageCode" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Name" class="control-label"></**label**><br />

<**input** **asp-for**="Name" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="Description" class="control-label"></**label**><br />

<**input** **asp-for**="Description" class="form-control" />

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

</div>

<div class="form-group">

<**label** **asp-for**="MedicineCategory" class="control-label"></**label**><br />

<**select** **asp-for**="MedicineCategory" class="form-control" **asp-items**="Html.GetEnumSelectList<MedicineCategory>()"></**select**>

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

</div>

<div class="form-group">

<**label** **asp-for**="Price" class="control-label"></**label**><br />

<**input** **asp-for**="Price" class="form-control" />

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

</div>

<div class="form-group">

<input type="submit" value="Update Item" class="btn btn-outline-success float-right" /><span class="float-right">&nbsp;&nbsp;</span>

</div>

</**form**>

<**a** class="btn btn-outline-danger float-right" **asp-action**="Delete" **asp-route-ID**="@Model.ID"><i class="bi bi-trash-fill">Delete</i></**a**>

<**a** class="btn btn-outline-info" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back, All Items</i></**a**>

</div>

</div>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Edit\_Item.png" style="height: 46%; width: 26%; position: fixed; right: 30px; bottom: 48px; z-index: 0;" /></div>

@model IEnumerable<Medicine>;

@{

ViewData["Title"] = "Medicines";

}

<div class="row">

@if (TempData["Error"] != null)

{

<div class="alert alert-warning col-md-12 text-center h5"><i class="bi bi-exclamation-triangle-fill"></i> @TempData["Error"]</div>

}

@foreach (var item in Model)

{

<div class="col-md-4 col-xs-6 border-primary mb-3">

<div class="card mb-3" style="max-width: 540px;">

<div class="row g-0">

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

<div class="card-header text-white bg-info">

<p class="card-text">

<h5 class="card-title">

@item.Name

@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))

{

<**a** class="text-white float-right" **asp-action**="Edit" **asp-route-ID**="@item.ID"><i class="bi bi-pencil-square" title="Edit & Delete"></i></**a**>

}

</h5>

</p>

</div>

</div>

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

<**img** src="~/Images/Items/@(item.ImageCode).png" style="max-width:100%" alt="@item.Name">

</div>

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

<div class="card-body">

<b>Description:</b>

<p class="card-text">@item.Description</p>

<b>Category:</b>

<span class="badge badge-pill badge-warning">@item.MedicineCategory</span>

</div>

</div>

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

<div class="card-footer">

<p class="card-text">

@if (User.Identity.IsAuthenticated)

{

<**a** class="btn btn-success text-white" **asp-controller**="Orders" **asp-action**="AddItemToShoppingCart" **asp-route-ID**="@item.ID"><i class="bi bi-cart-plus"></i> Add to cart | <b>Price: @item.Price.ToString("c")</b></**a**>

}

else

{

<**a** class="btn btn-success text-white" **asp-controller**="Account" **asp-action**="Login"><i class="bi bi-cart-plus"></i> Login for shopping | <b>Price: @item.Price.ToString("c")</b></**a**>

}@\*// My idea for AddItemToShoppingCart\_btn if Authenticated or not\*@

</p>

</div>

</div>

</div>

</div>

</div>

}

</div>

@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))

{

<div style="position: fixed; right:25px; bottom: 25px; " class="text-white">

<**a** **asp-controller**="Medicines" **asp-action**="Create" class="btn btn-success">

<i class="bi bi-plus-circle"> Add New Item</i>

</**a**>

</div>

}

@{

ViewData["Title"] = "Order Completed";

}

<br />

<div class="row">

<div class="col-md-6 offset-3 alert alert-success">

<h1 class="text-center"><i class="bi bi-receipt-cutoff"></i></h1>

<h2><i class="bi bi-check-square-fill"></i> Order Complete Successfully!</h2>

<p class="h4">You can check your orders from your profile</p><br />

<hr />

<p class="h4 font-italic">Thank you !</p>

</div>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Order\_Completed.png" style="height:100%;width: 40%" /></div>

@model List<Order>

@{

ViewData["Title"] = "Orders";

}

<div class="row">

<div class="col-md-8 offset-2">

<p><h4>List of all orders</h4></p>

<table class="table">

<thead>

<tr>

<th>Order ID</th>

<th>

<div class="text-center">

Items Detalils

<div class="badge badge-info text-black-50"><span class="badge bg-success">Quantity</span> - [price] - Item Name</div>

</div>

</th>

<th>Total</th>

@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))

{

<th>User</th>

}

</tr>

</thead>

<tbody>

@foreach (var order in Model)

{

<tr>

<td class="align-middle">@order.ID</td>

<td class="align-middle">

<ul style="list-style-type:none">

@foreach (var item in order.OrderItems)

{

<li>

<div class="alert alert-info" role="alert">

<span class="badge bg-success">@item.Amount</span> - [@item.Price.ToString("c")] - @item.Medicine.Name

</div>

</li>

}

</ul>

</td>

<td>@order.OrderItems.Select(t => t.Price \* t.Amount).Sum().ToString("c")</td>

@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))

{

<td>@order.User.FullName</td>

}

</tr>

}

</tbody>

<tfoot>

<tr>

<td colspan="4"></td>

</tr>

</tfoot>

</table>

<**a** class="btn btn-outline-info" **asp-controller**="Medicines" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back Home Page</i></**a**>

</div>

<div class="col-md-8 offset-2"><**img** class="float-right" src="~/Images/UX\_UI\_Design/Orders\_Cart.png" style="height:100%;width: 40%"/></div>

</div>

@using malshetwi\_CapstoneProject\_SDA.LMS.Data.ViewModels;

@model ShoppingCartVM

@{

ViewData["Title"] = "Shopping Cart";

var CartItems = Model.ShoppingCart.GetShoppingCartItems();

var count = CartItems.Count;

string visibility = "visible";

@if (count == 0) { visibility = "hidden"; }

}

<div class="row">

@\*<Summary>\*@

@if (TempData["Error"] != null)

{

<div class="col-md-12 alert alert-danger text-center h4"><i class="bi bi-shield-lock-fill"></i> @TempData["Error"]</div>

}

@if (count == 0 && TempData["Error"] == null)

{<div class="col-md-12 alert alert-warning text-center h4"><i class="bi bi-cart-x-fill"></i> Cart is empty !!</div>} @\* Code lines QA & Secutiry MSH <Summary/>\*@

<div class="col-md-8 offset-2">

<table class="table" style="visibility:@visibility">

<thead>

<tr>

<th></th>

<th>Medicine Name</th>

<th>Price</th>

<th>Selected Amount</th>

<th>Sub total</th>

<th></th>

</tr>

</thead>

<tbody>

@foreach (var Item in Model.ShoppingCart.ShoppingCartItems)

{

<tr>

<td class="align-middle"><img class="img-rounded" src="~/Images/Items/@(Item.Medicine.ImageCode).png" alt="@Item.Medicine.Name" style="max-width:60px" /></td>

<td class="align-middle">@Item.Medicine.Name</td>

<td class="align-middle">@Item.Medicine.Price.ToString("c")</td>

<td class="align-middle">@Item.Amount</td>

<td class="align-middle">@((Item.Medicine.Price \* Item.Amount).ToString("c"))</td> <!--SubTotal-->

<td>

<a class="btn alert-danger" asp-controller="Orders" asp-action="RemoveItemFromShoppingCart" asp-route-id="@Item.Medicine.ID">

<i class="bi bi-cart-dash"></i>

</a>

<!---->

<a class="btn alert-success" asp-controller="Orders" asp-action="AddItemToShoppingCart" asp-route-id="@Item.Medicine.ID">

<i class="bi bi-cart-plus"></i>

</a>

</td>

</tr>

}

</tbody>

<tfoot>

<tr>

<td colspan="4"></td>

<td class="badge badge-warning"><b>Total : </b>@Model.ShoppingCartTotal.ToString("c")</td>

</tr>

</tfoot>

</table>

<br />

<div class="float-right col-md-4 col-xs-6">

@if (count != 0)

{<a class="btn btn-outline-success" asp-action="CompleteOrder">Complete Order <i class="bi bi-arrow-right-square-fill"></i></a>}

</div>

<a class="btn btn-outline-info" asp-controller="Medicines" asp-action="Index"><i class="bi bi-arrow-left-square-fill"> Add more items</i></a>

</div>

</div>

@if (TempData["Error"] != null)

{

<div class="col-md-8 offset-2"><img class="float-right" src="~/Images/UX\_UI\_Design/shopping\_cart2.jpg" style="height:100%;width:40%" /></div>

}else

{

<div class="col-md-8 offset-2"><img class="float-right" src="~/Images/UX\_UI\_Design/shopping\_cart.png" style=" height:100%;width: 40%" /></div>

}

@\*V.91 Partial View Created, move these sections from \_Layout to here\*@

@inject UserManager<ApplicationUser> UserManager;

@if (!User.Identity.IsAuthenticated)

{

@\*Login & Register\*@

<**a** class="btn btn-outline-info" **asp-controller**="Account" **asp-action**="Login">

<i class="bi bi-box-arrow-in-right"></i> Log in

</**a**>

<span style="padding:3px"></span>

<**a** class="btn btn-outline-success" **asp-controller**="Account" **asp-action**="Register">

<i class="bi bi-person-plus"></i> Register

</**a**>

}

else

{

@\*User profile Section\*@

<div class="dropdown">

<button class="btn btn-outline-info dropdown-toggle" type="button" id="profile-btn" data-toggle="dropdown" aria-expanded="false">

<i class="bi bi-person-badge"></i> Hello @UserManager.GetUserName(User)

</button>

<ul class="dropdown-menu" aria-labelledby="profile-btn">

@if (User.Identity.IsAuthenticated && User.IsInRole("Admin"))

{

<li>

<**a** class="dropdown-item" **asp-controller**="Orders" **asp-action**="Index">

<i class="bi bi-list"></i> All Orders

</**a**>

</li>

<li>

<**a** class="dropdown-item" **asp-controller**="Account" **asp-action**="Users">

<i class="bi bi-people"></i> App Users

</**a**>

</li>

}

else

{

<li>

<**a** class="dropdown-item" **asp-controller**="Orders" **asp-action**="Index">

<i class="bi bi-list"></i> My Orders

</**a**>

</li>

}

</ul>

</div>@\*// profile Section ends, Vid: 76 \*@

<span style="padding:3px"></span>

@\*Log out\*@

<**form** **asp-controller**="Account" **asp-action**="Logout">

<button class="btn btn-outline-danger my-2 my-sm-0">

<i class="bi bi-box-arrow-in-right"></i> Log out

</button>

</**form**>

}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>@ViewData["Title"]</title>

<**link** rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />

<**link** rel="stylesheet" href="~/css/site.css" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.1/font/bootstrap-icons.css">

<**link** href="~/Images/UX\_UI\_Design/pharmacy-logo.jpg" rel="shortcut icon" />

</head>

<body>

<header>

<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">

<div class="container-fluid">

<**img** src="~/Images/UX\_UI\_Design/Health.jpg" style="height:10%;width:10%" />

<h2 class="p-3"><**a** class="badge badge-info" **asp-area**="" **asp-controller**="Medicines" **asp-action**="Index">ABC Healthcare pharmacy</**a**></h2>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target=".navbar-collapse" aria-controls="navbarSupportedContent"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">

<ul class="navbar-nav flex-grow-1">

<!-- <li class="nav-item"><a class="nav-link text-dark" asp-area="" asp-controller="Medicines" asp-action="Index">malshetwi\_CapstoneProject\_SDA.LMS</a></li>-->

<li class="nav-item"><**a** class="nav-link text-info" **asp-area**="" **asp-controller**="Medicines" **asp-action**="Index"><b>Medicines</b></**a**></li>

<li class="nav-item"><**a** class="nav-link text-info" **asp-area**="" **asp-controller**="Home" **asp-action**="About"><b>About</b></**a**></li>

<li class="nav-item"><**a** class="text text-info" **asp-controller**="Orders" **asp-action**="ShoppingCart"><h3><i class="bi bi-cart3" title="Shopping Cart"></i>@await Component.InvokeAsync("ShoppingCartSummary")</h3></**a**></li>

</ul>

<**form** class="form-inline my-1 my-lg-0" **asp-controller**="Medicines" **asp-action**="Filter" style="padding-right: 15px">

<input name="searchString" type="text" class="form-control" placeholder="Search for a Medicine ..." aria-label="Recipient's username" aria-describedby="button-addon2">

<button class="btn btn-outline-info my-2 my-sm-0" type="submit"><i class="bi bi-search"></i></button>

</**form**>

@\*await Html.RenderPartialAsync() // one of ways of rendering partial view\*@

<**partial** **name**="\_Identity.cshtml" />

</div>

</div>

</nav>

</header>

<div class="container-fluid">

<main role="main" class="pb-3">

@RenderBody()

</main>

</div>

<footer class="footer text-muted">

<div class="container text-info">

&copy; 2022 - Mohan\_CapstoneProject

</div>

</footer>

<**script** src="~/lib/jquery/dist/jquery.min.js"></**script**>

<**script** src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></**script**>

<**script** **src**="~/js/site.js" **asp-append-version**="true"></**script**>

@await RenderSectionAsync("Scripts", required: false)

</body>

</html>

@{

ViewData["Title"] = "404 | Not Founded";

}

<div class="col-md-12 text-md-center">

<br />

<**img** src="~/Images/UX\_UI\_Design/404.png" style="height:100%;width:30%" />

<h1>404 Data or Page your action on <span style="color:brown">NOT FOUND !!</span></h1>

<h2 style="color:cornflowerblue"><i class="bi bi-info-circle-fill"></i> Data may have been changed by another page or user</h2><br />

<**a** class="btn btn-outline-info" **asp-action**="Index"><i class="bi bi-arrow-left-square-fill"> Back, All Items</i></**a**>

</div>

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft": "Warning",

"Microsoft.Hosting.Lifetime": "Information"

}

},

"ConnectionStrings": {

"DefaultConnection": "Server=(localdb)\\MSSQLLocalDB;Database=E\_HealthCare;Trusted\_Connection=True;MultipleActiveResultSets=true"

},

"AllowedHosts": "\*"

}

using Microsoft.AspNetCore.Hosting;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.Hosting;

using Microsoft.Extensions.Logging;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS

{

public class Program

{

public static void Main(string[] args)

{

CreateHostBuilder(args).Build().Run();

}

public static IHostBuilder CreateHostBuilder(string[] args) =>

Host.CreateDefaultBuilder(args)

.ConfigureWebHostDefaults(webBuilder =>

{

webBuilder.UseStartup<Startup>();

});

}

}

using malshetwi\_CapstoneProject\_SDA.LMS.Data;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Cart;

using malshetwi\_CapstoneProject\_SDA.LMS.Data.Services;

using malshetwi\_CapstoneProject\_SDA.LMS.Models;

using Microsoft.AspNetCore.Authentication.Cookies;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.AspNetCore.Identity;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace malshetwi\_CapstoneProject\_SDA.LMS

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

// ~ DbContext configuration, define Db context(transelator) file and Data Storage(V14)

services.AddDbContext<AppDbContext>(options => options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

//~ Services Written by me

services.AddScoped<IMedicinesService, MedicinesService>();

services.AddScoped<IOrdersService, OrdersService>();

services.AddSingleton<IHttpContextAccessor, HttpContextAccessor>();

services.AddScoped(sc=> ShoppingCart.GetShoppingCart(sc));

//~ authentication and authorization

services.AddIdentity<ApplicationUser, IdentityRole>().AddEntityFrameworkStores<AppDbContext>();

services.AddMemoryCache();

services.AddSession();// Was Written before auth&authoriz

services.AddAuthentication(options => options.DefaultScheme = CookieAuthenticationDefaults.AuthenticationScheme);

//~ end of Serv's Written by me

services.AddControllersWithViews();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

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.UseSession();

//~ Use authentication and authorization

app.UseAuthentication();

app.UseAuthorization();

//app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

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

});

// Seed default data

AppDbInitializer.Seed(app);

//~ Seed default Roles&Users

AppDbInitializer.SeedUsersAndRoles(app).Wait();

}

}

}