Models:

public class Bouquet

{

public int id { get; set; }

public string name { get; set; }

public string price { get; set; }

public string image { get; set; }

}

public class Customer

{

public int id { get; set; }

public string f\_name { get; set; }

public string l\_name { get; set; }

public DateOnly dob { get; set; }

public char gender { get; set; }

public string p\_no { get; set; }

public String address { get; set; }

public string email { get; set; }

public String password { get; set; }

public string? role { get; set; }

}

public class Order

{

[Key]

public int OrderId { get; set; }

public int BouqId { get; set; }

public int CustId { get; set; }

public int Quantity { get; set; }

public string DiliveryAddress { get; set; }

public string Status { get; set; }

[ForeignKey("BouqId")]

public Bouquet Bouquet { get; set; }

[ForeignKey("CustId")]

public Customer Customer { get; set; }

}

DbContext:

public class myContext : DbContext

{

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

{

}

public DbSet<Bouquet> Bouquets { get; set; }

public DbSet<Customer> Customers { get; set; }

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

public DbSet<Message> Messages { get; set; }

}

Connection String:

"AllowedHosts": "\*",

"ConnectionStrings": {

"myconnection": "server=(localdb)\\local;database=db\_floral;user id=local;password=12345 ;Trusted\_Connection=True;TrustServerCertificate=True;Encrypt=False"

}

Program.cs:

builder.Services.AddDbContext<myContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("myconnection")));

builder.Services.AddMvc();

builder.Services.AddSession(options =>

{

options.IdleTimeout = TimeSpan.FromSeconds(100);

});

builder.Services.AddHttpContextAccessor();

app.UseSession();

app.UseStaticFiles();

app.Run();

Controller(Order):

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using Project.Data;

using Project.Migrations;

using Project.Models;

namespace Project.Controllers

{

public class OrdersController : Controller

{

private myContext \_context;

public static List<Dictionary<string, string>> mycart = BouquetController.mycart;

public OrdersController(myContext context)

{

\_context = context;

}

[HttpPost]

public IActionResult addorders(string deliveryaddress, string custId)

{

foreach (var order in mycart)

{

Order o = new Order();//int.Parse(HttpContext.Session.GetString("User\_id").ToString()), int.Parse(order["id"]), int.Parse(order["qty"]),deliveryaddress,"pending"

//Console.WriteLine(int.Parse(HttpContext.Session.GetString("User\_id").ToString()));

o.CustId = int.Parse(custId);

o.BouqId = int.Parse(order["id"]);

o.Quantity = int.Parse(order["qty"]);

o.DiliveryAddress = deliveryaddress;

o.Status = "pending";

\_context.Orders.Add(o);

\_context.SaveChanges();

}

return RedirectToAction("fetchOrder");

}

public IActionResult fetchOrder()

{

var orders = \_context.Orders.Include(c => c.Bouquet).Include(c => c.Customer).ToList();

return View(orders);

}

public IActionResult deleteOrder(int id)

{

var orders = \_context.Orders.Find(id);

\_context.Orders.Remove(orders);

\_context.SaveChanges();

return RedirectToAction("fetchOrder");

}

}

}

Controller(Boquet):

using Microsoft.AspNetCore.Mvc;

using Project.Data;

using Project.Models;

using System.Net;

namespace Project.Controllers

{

public class BouquetController : Controller

{

private myContext \_context;

private IWebHostEnvironment \_env;

public static List<Dictionary<string,string>> mycart=new List<Dictionary<string,string>>();

public BouquetController(myContext context, IWebHostEnvironment env)

{

\_context = context;

\_env = env;

}

public IActionResult Ins\_bouquet()

{

return View();

}

[HttpPost]

public IActionResult Ins\_bouquet(IFormFile image, Bouquet bqt)

{

string filename = Path.GetFileName(image.FileName);

string filepath = Path.Combine(\_env.WebRootPath, "images/"+filename);

FileStream fs = new FileStream(filepath, FileMode.Create);

image.CopyTo(fs);

bqt.image = filename;

\_context.Bouquets.Add(bqt);

\_context.SaveChanges();

return RedirectToAction("Showbqt");

}

public IActionResult Showbqt()

{

if (HttpContext.Session.GetString("User\_role") == "admin")

{

var students = \_context.Bouquets.ToList();

return View(students);

}

else { return RedirectToAction("Error", "Exception"); }

}

public IActionResult deletebouquets(int id)

{

var bouquets = \_context.Bouquets.Find(id);

\_context.Bouquets.Remove(bouquets);

\_context.SaveChanges();

return RedirectToAction("Showbqt");

}

public IActionResult editbouquets(int id)

{

var bouquets = \_context.Bouquets.Find(id);

return View(bouquets);

}

[HttpPost]

public IActionResult editbouquets(Bouquet std)

{

\_context.Bouquets.Update(std);

\_context.SaveChanges();

return RedirectToAction("Showbqt");

}

public IActionResult Detailpage(int id)

{

List<Bouquet> bqts = \_context.Bouquets.ToList();

Bouquet bqt = bqts.Find(b=> b.id== id);

//Bouquet bqt= (Bouquet)\_context.Bouquets.Where(b => b.id == id);

return View(bqt);

}

[HttpPost]

public IActionResult addtocart(int id, string name, string price,string image, int qty)

{

if (HttpContext.Session.GetString("User\_role") != null)

{

mycart.Add(new Dictionary<string, string>());

mycart[mycart.Count - 1].Add("id", id.ToString());

mycart[mycart.Count - 1].Add("name", name);

mycart[mycart.Count - 1].Add("price", price);

mycart[mycart.Count - 1].Add("image", image);

mycart[mycart.Count - 1].Add("qty", qty.ToString());

return RedirectToAction("cart","Cart");

}

else

{

return RedirectToAction("Login", "Customer");

}

}

}

}

View (Order):

@using Project.Models;

@model List<Order>

@using Microsoft.AspNetCore.Http

@inject IHttpContextAccessor Accessor;

@using Project.Data;

@{

if (Accessor.HttpContext.Session.GetString("User\_role") == "admin")

Layout = "../Shared/\_Layout.cshtml";

else

{

Layout = "../Shared/\_homelayout.cshtml";

}

}

@foreach (var item in Model)

{

if (Accessor.HttpContext.Session.GetString("User\_role") == "admin")

{

<tr>

<td>@item.OrderId</td>

<td>@item.Bouquet.name</td>

<td>@item.Bouquet.price</td>

<td><**img** src="~/images/@item.Bouquet.image" height="100px" width="100px" /></td>

<td>@item.Quantity</td>

<td>

@(item.Quantity \* int.Parse(item.Bouquet.price))

</td>

<td>@item.DiliveryAddress</td>

<td>@item.Customer.f\_name</td>

<td>@item.Status</td>

<td>

@\* add deleteOrder method in OrderController \*@

<a href="@Url.Action("deleteOrder","Orders",new {id=item.OrderId})">Delete</a>

</td>

</tr>

}

}

Output:



