

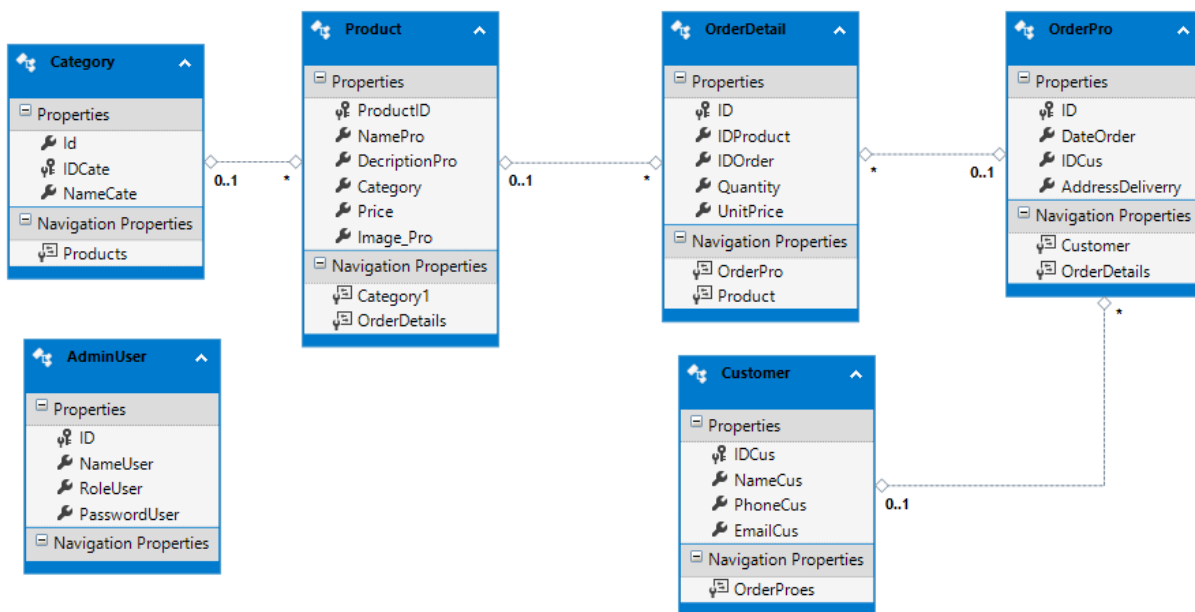
# THỰC HÀNH LẬP TRÌNH WEB NÂNG CAO

## BÀI THỰC HÀNH 04: Cart

### Kết nối CSDL bằng Database First

- Cơ sở dữ liệu được sử dụng trong bài thực hành: **DBSportStore**

[Link tải database](#)



### I. Tạo lớp xử lý giỏ hàng Cart.cs

#### Hướng dẫn thực hành

Models=>Add => Class => Cart.cs

```

namespace DemoConectDB.Models
{
    //Tạo lớp CartItem chứa các dòng sản phẩm trong giỏ hàng
    0 references
    public class CartItem
    {
        0 references
        public Product _product { get; set; }
        0 references
        public int _quantity { get; set; }
    }
    0 references
    public class Cart
    {
    }
}

```

Trong lớp Cart:

-Khai báo thuộc tính dòng sản phẩm items trong lớp Cart

```

0 references
public class Cart
{
    List<CartItem> items = new List<CartItem>();
    0 references
    public IEnumerable<CartItem> Items
    {
        get { return items; }
    }
}

```

-Viết hàm lấy sản phẩm bỏ vào giỏ hàng

```

public void Add_Product_Cart(Product _pro,int _quan = 1)
{
    var item = Items.FirstOrDefault(s => s._product.ProductID == _pro.ProductID);
    if (item == null) //nếu giỏ hàng rỗng thì thêm dòng hàng mới vào giỏ
        items.Add(new CartItem
        {
            _product = _pro,
            _quantity = _quan
        });
    else
        item._quantity += _quan; //tổng số lượng trong giỏ hàng được cộng dồn
}

```

-Viết hàm tính tổng số lượng trong giỏ hàng

```

0 references
public int Total_quantity()
{
    return items.Sum(s => s._quantity);
}

```

-Viết hàm tính thành tiền cho mỗi dòng sản phẩm trong giỏ hàng

```
public decimal Total_money()
{
    var total = items.Sum(s => s._quantity * s._product.Price);
    return (decimal)total;
}
```

-Viết hàm cập nhật lại số lượng sản phẩm ở mỗi dòng sản phẩm khi khách hàng muốn đặt mua thêm

```
public void Update_quantity(int id, int _new_quan)
{
    var item = items.Find(s => s._product.ProductID == id);
    if (item != null)
        item._quantity = _new_quan;
}
```

-Viết hàm xóa sản phẩm trong giỏ hàng

```
0 references
public void Remove_CartItem(int id)
{
    items.RemoveAll(s => s._product.ProductID == id);
}
```

-Viết hàm xóa giỏ hàng sau khi Khách hàng thực hiện thanh toán

```
public void ClearCart()
{
    items.Clear();
}
```

## II. Tạo controller xử lý chức năng mua hàng

Controller => New => Controller ...=> Empty => ShoppingCartController.cs

1) Khởi tạo kết nối Database, thiết kế form Giỏ hàng

Trong **ShoppingCartController.cs**:

-Kết nối Database và viết action **ShowCart()**, sau đó render đến View để thiết kế form

```

DBSportStoreEntities database = new DBSportStoreEntities();
// GET: ShoppingCart
0 references
public ActionResult ShowCart()
{
    if (Session["Cart"] == null)
        return RedirectToAction("ShowCart", "ShoppingCart");
    Cart _cart = Session["Cart"] as Cart;
    return View(_cart);
}

```

-Nhấp phải chuột vào action **ShowCart()** -> **Add View**

Add View

View name: ShowCart

Template: Empty

Model class: Cart (DemoContextDB.Models)

Data context class:

Options:

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page:

~/Views/Shared/\_Layout.cshtml

(Leave empty if it is set in a Razor \_viewstart file)

Add Cancel

-Trong ShowCart.cshtml, thiết kế form giỏ hàng

```

1  @model DemoContextDB.Models.Cart
2
3  @{
4      ViewBag.Title = "ShowCart";
5      Layout = "~/Views/Shared/_Layout.cshtml";
6  }
7
8  <h2>Giỏ Hàng</h2>
9  <a href="@Url.Action("Index","Product")"> Tiếp tục mua hàng </a>
10 @if (Model != null)
11 {
12     <table class="table">
13         <thead>
14             <tr>
15                 <th>Tên sản phẩm </th>
16                 <th>Hình sản phẩm</th>
17                 <th>Giá</th>
18                 <th>Số lượng </th>
19                 <th>Thành tiền</th>
20                 <th></th>
21             </tr>
22         </thead>
23

```

```

24 <tbody>
25     @foreach (var item in Model.Items)
26     {
27         decimal total = item._quantity * (decimal)item._product.Price.Value;
28         <tr>
29             <td>@Html.DisplayFor(model=>item._product.NamePro)</td>
30             <td></td>
31             <td>@Html.DisplayFor(model => item._product.Price) </td>
32             <td>
33                 @using(Html.BeginForm("Update_Cart_Quantity", "ShoppingCart"))
34                 {
35                     <input type="hidden" value="@item._product.ProductID" name="idPro" />
36                     <input type="number" value="@item._quantity" name="cartQuantity" />
37                     <input type="submit" value="Update" />
38                 }
39             </td>
40             <td>@total</td>
41             <td>
42                 <a href="/ShoppingCart/RemoveCart/@item._product.ProductID">Xóa</a>
43             </td>
44         </tr>
45     }
46 </tbody>
47 <tfoot>
48     <tr>
49         <td colspan="6" style="text-align:right"> Tổng tiền: @Model.Total_money() </td>
50     </tr>
51 </tfoot>
52 </table>
53
54

```

-Mở trang hiển thị danh sách sản phẩm **View/Product/Index.cshtml** thêm liên kết đến action **AddToCart** của **ShoppingCartController.cs**

+Thực hiện thêm link:

```

1  @model IEnumerable<DemoContextDB.Models.Product>
2
3  @{
4      ViewBag.Title = "Index";
5  }
6  <h2>DANH SÁCH SẢN PHẨM</h2>
7  <div class="container">
8      <div class="row">
9          @foreach (var pro in Model)
10          {
11              <div class="col-sm-12 col-sm-6 col-md-4 col-lg-3">
12                  <div class="card-pro">
13                      <p>@Html.DisplayFor(model => pro.ProductID)</p>
14                      
15                      <p>@Html.DisplayFor(model => pro.NamePro)</p>
16                      <p>@Html.DisplayFor(model => pro.Price)</p>
17                      <a href="/ShoppingCart/AddToCart/@pro.ProductID" class="btn btn-default">Buy</a>
18                  </div>
19              </div>
20          }
21      </div>
22  </div>
23
24

```

2) Trong **ShoppingCartController.cs**, viết hàm thêm sản phẩm vào giỏ hàng

```

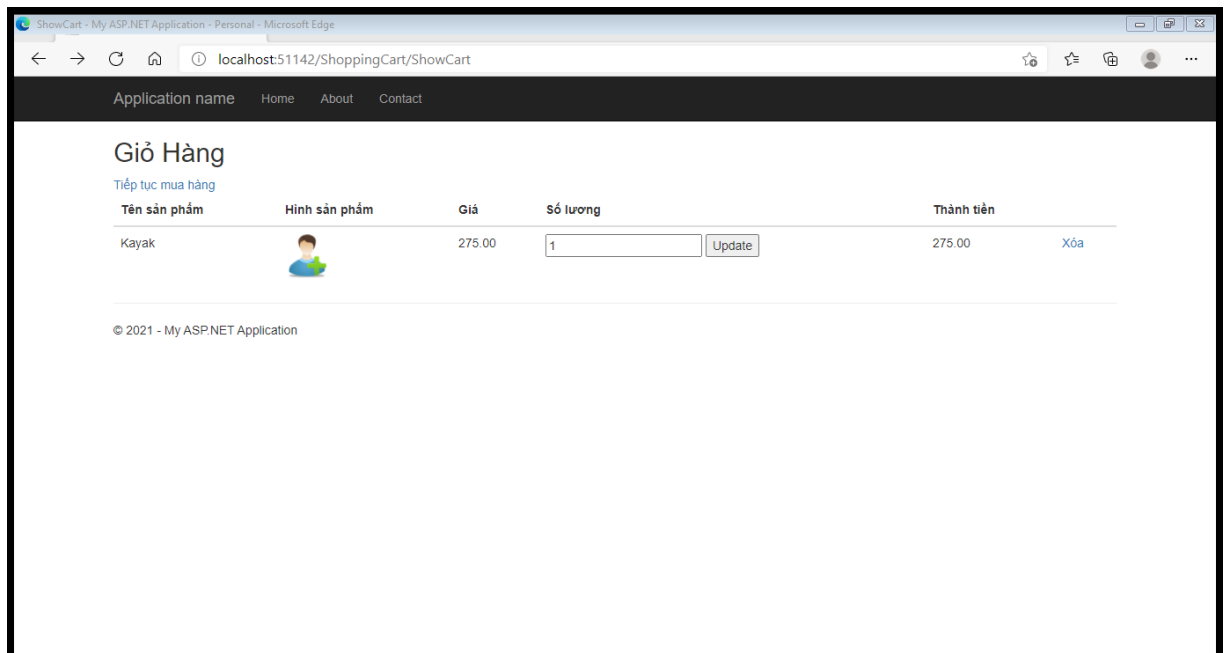
//Action Tạo mới giỏ hàng
1 reference
public Cart GetCart()
{
    Cart cart = Session["Cart"] as Cart;
    if (cart == null || Session["Cart"] == null)
    {
        cart = new Cart();
        Session["Cart"] = cart;
    }
    return cart;
}

//Action thêm product vào giỏ hàng
0 references
public ActionResult AddToCart(int id)
{
    var _pro = database.Products.SingleOrDefault(s => s.ProductID == id); //lấy pro theo ID
    if (_pro != null)
    {
        GetCart().Add_Product_Cart(_pro);
    }
    return RedirectToAction("ShowCart", "ShoppingCart");
}

```

-Chạy chức năng: Thêm sản phẩm vào giỏ hàng

Mở View/Product/Index.cshtml => Buy



- 3) Trong **ShoppingCartController.cs**, viết **hàm cập nhật số lượng sản phẩm** và tính lại tổng tiền:

```

public ActionResult Update_Cart_Quantity(FormCollection form)
{
    Cart cart = Session["Cart"] as Cart;
    int id_pro = int.Parse(form["idPro"]);
    int _quantity = int.Parse(form["cartQuantity"]);
    cart.Update_quantity(id_pro, _quantity);
    return RedirectToAction("ShowCart", "ShoppingCart");
}

```



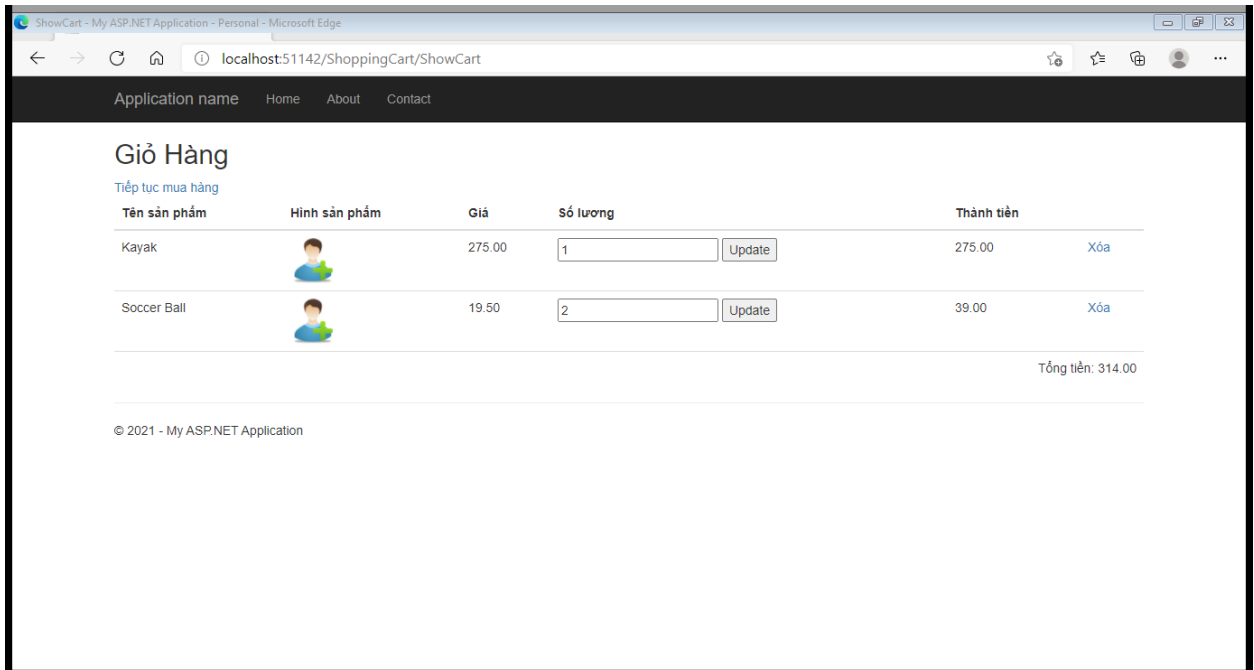
Chú ý: tên trong form được định nghĩa trong ShowCart.cshtml

- 4) Trong **ShoppingCartController.cs**, viết action xóa dòng sản phẩm trong Giỏ hàng

```

public ActionResult RemoveCart(int id)
{
    Cart cart = Session["Cart"] as Cart;
    cart.Remove_CartItem(id);
    return RedirectToAction("ShowCart", "ShoppingCart");
}

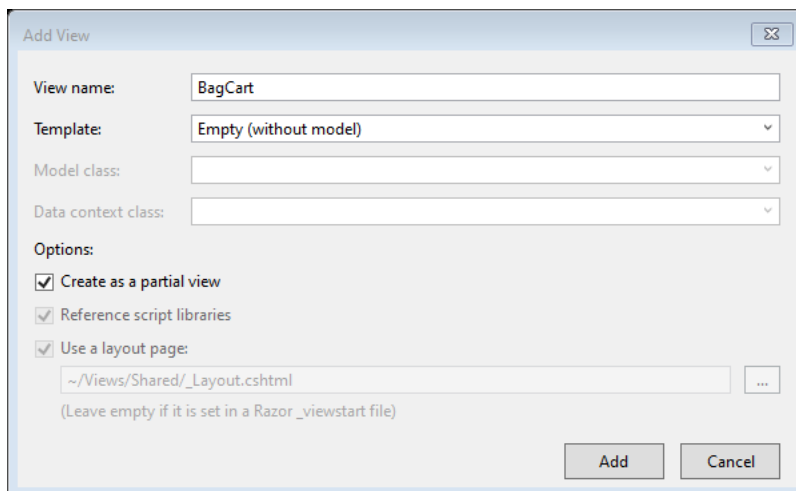
```



##### 5) Icon giỏ hàng trên navbar-right hiển thị tổng số lượng sản phẩm có trong giỏ

-Trong ShoppingCartController.cs, viết hàm lấy tổng số lượng sản phẩm có trong Giỏ hàng, sau đó Render đến PartialView

```
public PartialViewResult BagCart()
{
    int total_quantity_item = 0;
    Cart cart = Session["Cart"] as Cart;
    if (cart != null)
    {
        total_quantity_item = cart.Total_quantity();
        ViewBag.QuantityCart = total_quantity_item;
    }
    return PartialView("BagCart");
}
```





-Nội dung trong BagCart.cshtml

```
BagCart.cshtml* X ShowCart.cshtml ShoppingCartController.cs Cart.cs Create.cshtml Index.cshtml
1 <span style="color:white; margin-left:-8px" class="badge">@ViewBag.QuantityCart</span>
2
```

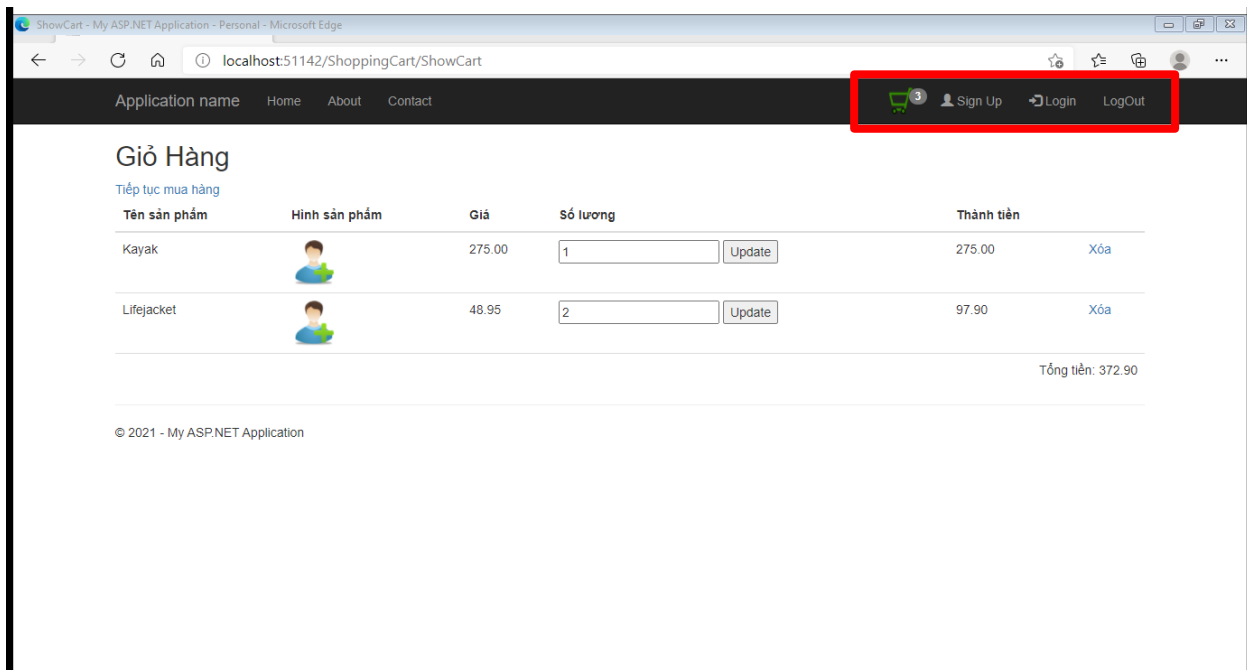
-Vào trong **View/Shared/\_Layout.cshtml** để gắn **BagCart** vào **navbar-right**

-Trong navbar-right viết các liên kết đến các chức năng Register, Login, Logout

```
14 <div class="navbar navbar-inverse navbar-fixed-top">
15 <div class="container">
16 <div class="navbar-header">
17 <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
18 <span class="icon-bar"></span>
19 <span class="icon-bar"></span>
20 <span class="icon-bar"></span>
21 </button>
22 @Html.ActionLink("Application name", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
23 </div>
24 <div class="navbar-collapse collapse">
25 <ul class="nav navbar-nav">
26 <li>@Html.ActionLink("Home", "Index", "Home")</li>
27 <li>@Html.ActionLink("About", "About", "Home")</li>
28 <li>@Html.ActionLink("Contact", "Contact", "Home")</li>
29 @*<li><a href="@Html.Action("LogOutUser", "LoginUser")">LogOut</a></li>*@
30 </ul>
31 <ul class="nav navbar-nav navbar-right">
32 <li>
33 
34 @Html.Action("BagCart", "ShoppingCart")
35 </li>
36 <li><a href="/LoginUser/RegisterUser"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
37 <li><a href="/LoginUser/Index"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
38 <li><a href="@Url.Action("LogOutUser", "LoginUser")">LogOut</a></li>
39 </ul>
40 </div>
41 </div>
42 </div>
43 </div>
```

-Trong LoginUserController.cs, viết hàm LogOutUser()

```
LoginUserController.cs X _Layout.cshtml BagCart.cshtml ShowCart.cshtml
DemoContextDB
5 using System.Web.Mvc;
6 using DemoContextDB.Models;
7
8 namespace DemoContextDB.Controllers
9 {
10     public class LoginUserController : Controller
11     {
12         DBSportStoreEntities database = new DBSportStoreEntities();
13         public ActionResult RegisterUser()...
14     }
15     [HttpPost]
16     public ActionResult RegisterUser(AdminUser _user)...
17
18     // GET: LoginUser : Form Login
19     public ActionResult Index()...
20     [HttpPost]
21
22     public ActionResult LoginAccount(AdminUser _user)...
23
24     public ActionResult LogOutUser()
25     {
26         Session.Abandon();
27         return RedirectToAction("Index", "LoginUser");
28     }
29 }
30
```

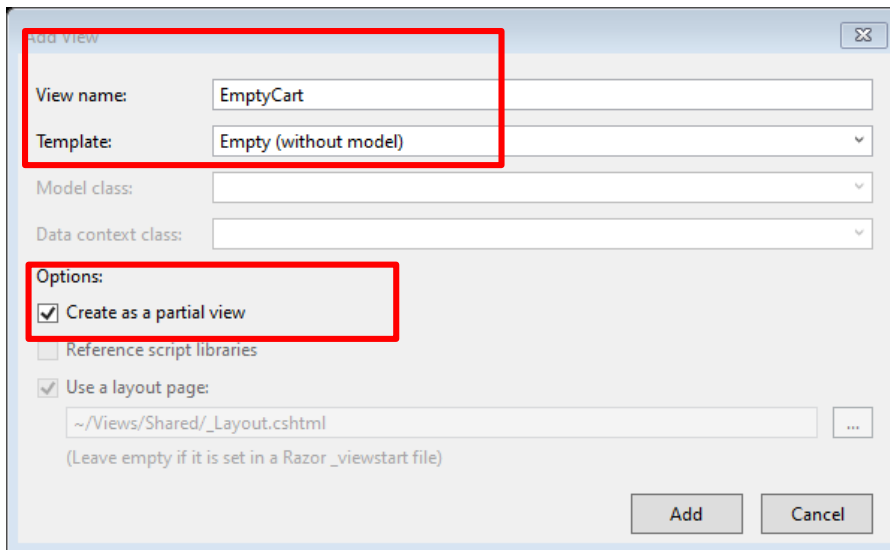
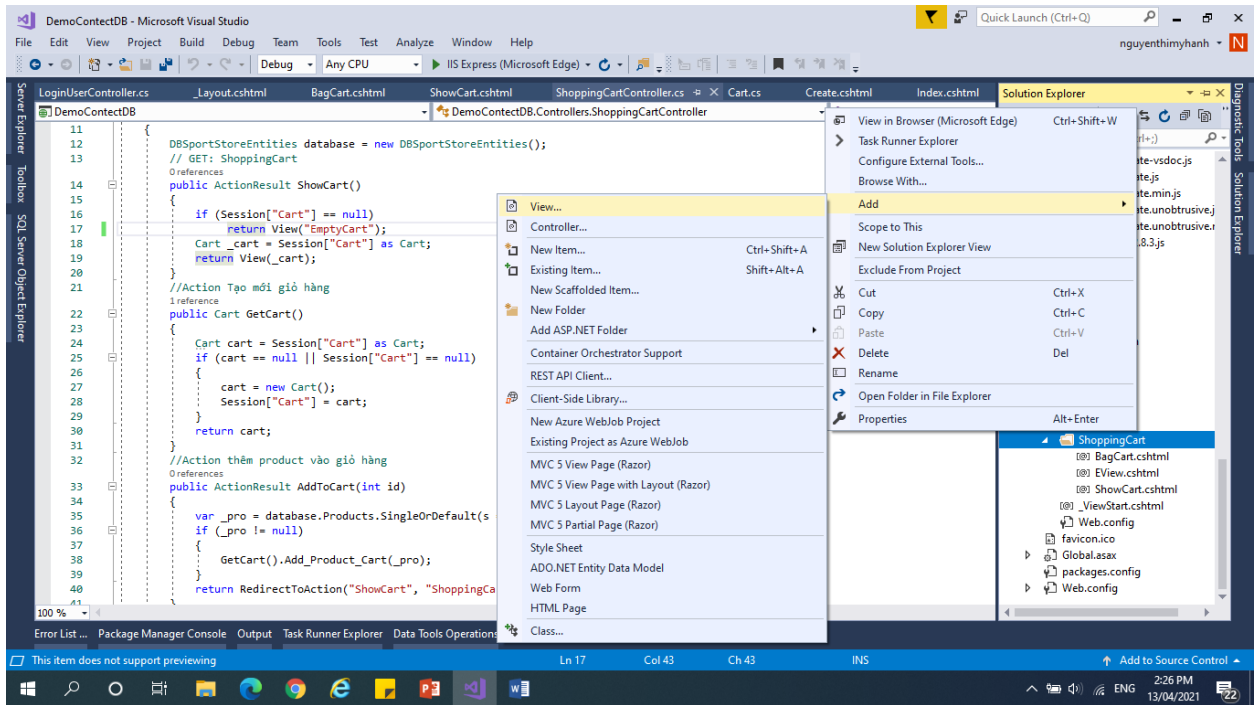


## 6) Fix lỗi khi giỏ hàng trống

**-Bước 1:** Mở ShoppingCartController.cs/ShowCart()

```
public ActionResult ShowCart()
{
    if (Session["Cart"] == null)
    {
        return View("EmptyCart");
    }
    Cart _cart = Session["Cart"] as Cart;
    return View(_cart);
}
```

**-Bước 2:** View/ShoppingCart => nhấp phải chuột => Add => View => EmptyCart



**-Bước 3:** Nội dung bên trong của `EmptyCart.cshtml`: là định dạng khung của `ShowCart`:

```

EView.cshtml EmptyCart.cshtml>LoginUserController.cs _Layout.cshtml
1
2
3 <h2>Giỏ Hàng</h2>
4 <a href="@Url.Action("Index","Product")"> Tiếp tục mua hàng </a>
5 <table class="table">
6 <thead>
7 <tr>
8 <th>Tên sản phẩm </th>
9 <th>Hình sản phẩm</th>
10 <th>Giá</th>
11 <th>Số lượng </th>
12 <th>Thành tiền</th>
13 <th></th>
14 </tr>
15
16 </thead>
17 <tbody>
18 <tr>
19 <td colspan="6">
20 Chưa có sản phẩm nào ...
21 </td>
22 </tr>
23 </tbody>
24 </table>

```

**-Bước 4:** Mở ShowCart.cshtml thêm vào lệnh else... của if

```

1 @model DemoContextDB.Models.Cart
2
3 @{
4     ViewBag.Title = "ShowCart";
5     Layout = "~/Views/Shared/_Layout.cshtml";
6 }
7
8 <h2>Giỏ Hàng</h2>
9 <a href="@Url.Action("Index","Product")"> Tiếp tục mua hàng </a>
10 @if (Model != null)
11 {
12 <table class="table">...</table>
13 }
14
15 else
16 {
17     @Html.Partial("EmptyCart")
18 }
19
20

```

**-Bước 5:**

**Ý tưởng:** Người dùng chưa chọn sản phẩm nào, nên giỏ hàng sẽ rỗng, người dùng muốn xem giỏ hàng rỗng bằng cách nhấp chuột vào giỏ hàng để mở Giỏ hàng.

**Xử lý:**

- Vào thư mục Shared/\_Layout.cshtml sửa lại nội dung chỗ BagCart Icon trên navbar-right:

```
<ul class="nav navbar-nav navbar-right">
  <li>
    <span>
      <a href="/ShoppingCart/ShowCart">
        
      </a>
    </span>
    @Html.Action("BagCart", "ShoppingCart")
  </li>
  <li><a href="/LoginUser/RegisterUser"><span class="glyphicon glyphicon-user"></span> Sign Up</a></li>
  <li><a href="/LoginUser/Index"><span class="glyphicon glyphicon-log-in"></span> Login</a></li>
  <li><a href="@Url.Action("LogOutUser", "LoginUser")">LogOut</a></li>
</ul>
```