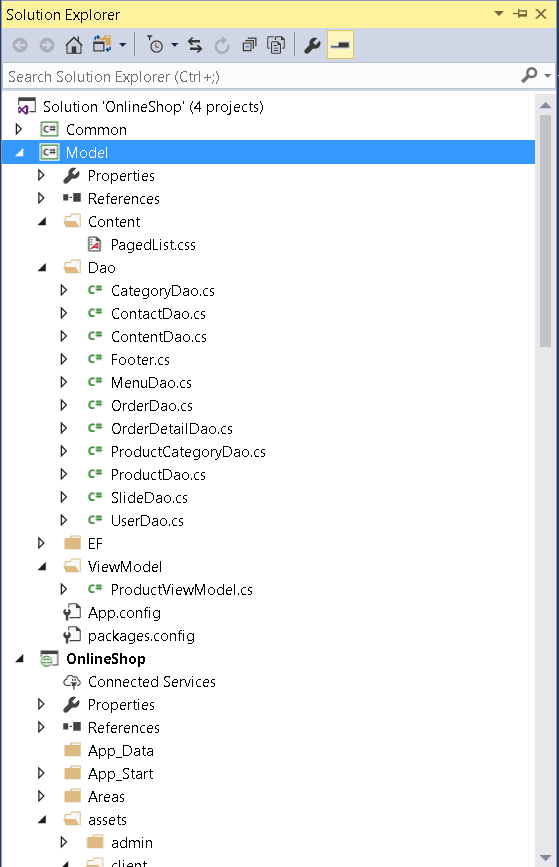
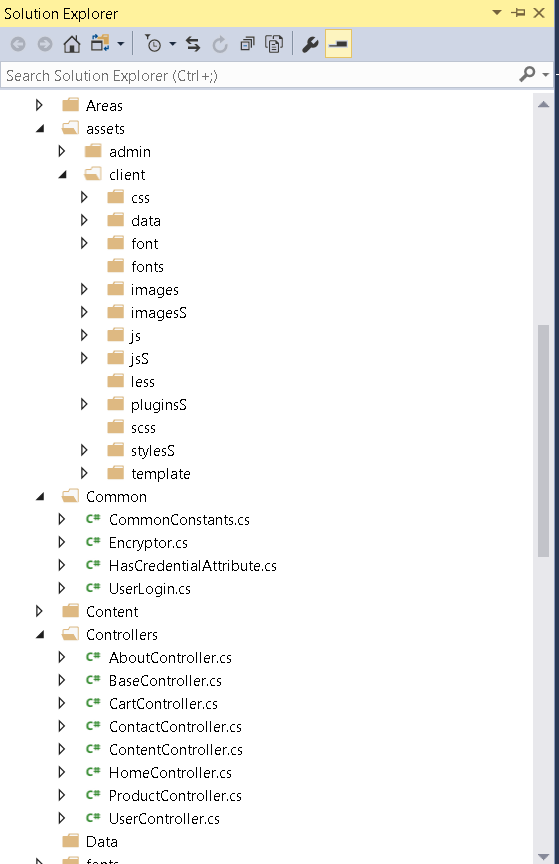
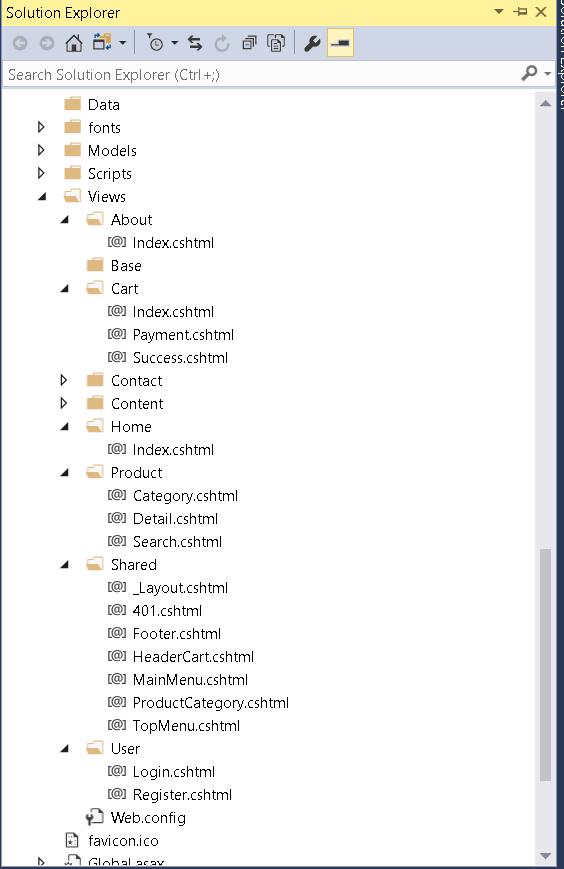
BÁO CÁO CODE THƯƠNG MẠI ĐIỆN TỬ

Thư mục quản lý:



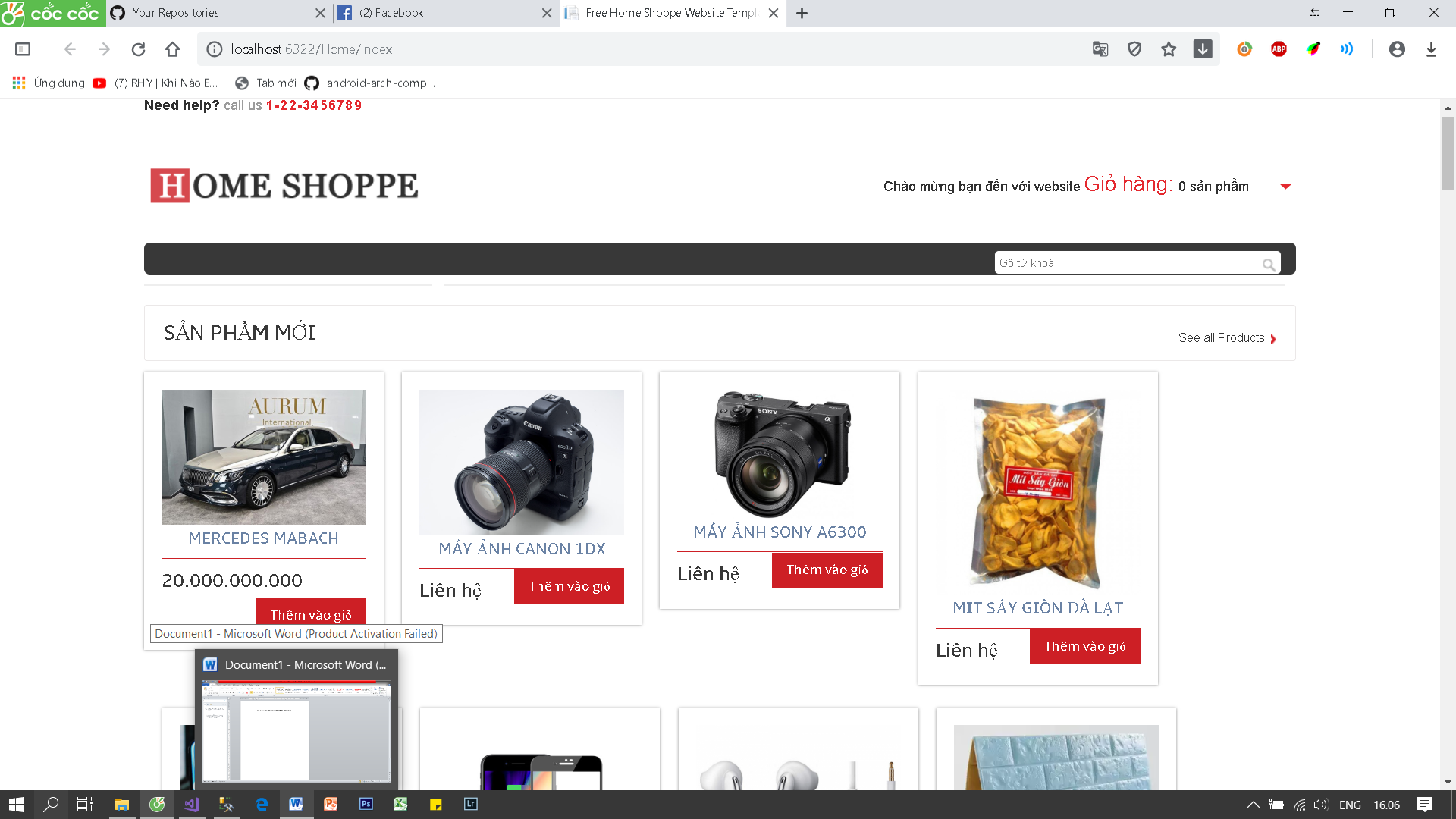
Kết nối CSDL

<connectionStrings>

<add name="OnlineShopDbContext" connectionString="data source=DESKTOP-7181COM\SQLEXPRESS;initial catalog=OnlineShop;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" />

</connectionStrings>

Trang Index



ProductDao:

using Model.EF;

using Model.ViewModel;

using PagedList;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Model.Dao

{

public class ProductDao

{

OnlineShopDbContext db = null;

public ProductDao()

{

db = new OnlineShopDbContext();

}

public List<Product> ListNewProduct(int top)

{

return db.Products.OrderByDescending(x => x.CreateDate).Take(top).ToList();

}

public List<string> ListName(string keyword)

{

return db.Products.Where(x => x.Name.Contains(keyword)).Select(x => x.Name).ToList();

}

public IEnumerable<Product> ListAllPaging(string searchString, int page = 5, int pageSize = 10)

{

IQueryable<Product> model = db.Products;

if (!string.IsNullOrEmpty(searchString))

{

model = model.Where(x => x.Name.Contains(searchString) || x.Name.Contains(searchString));

}

return model.OrderByDescending(x => x.CreateDate).ToPagedList(page, pageSize);

}

/// <summary>

/// Get list product by category

/// </summary>

/// <param name="categoryID"></param>

/// <returns></returns>

public List<ProductViewModel> ListByCategoryId(long categoryID, ref int totalRecord, int pageIndex = 10, int pageSize = 20)

{

totalRecord = db.Products.Where(x => x.CategoryID == categoryID).Count();

var model = (from a in db.Products

join b in db.ProductCategories

on a.CategoryID equals b.id

where a.CategoryID == categoryID

select new

{

CateMetaTitle = b.MetaTitle,

CateName = b.Name,

CreatedDate = a.CreateDate,

ID = a.Id,

Images = a.Image,

Name = a.Name,

MetaTitle = a.MetaTitle,

Price = a.Price

}).AsEnumerable().Select(x => new ProductViewModel()

{

CateMetaTitle = x.MetaTitle,

CateName = x.Name,

CreatedDate = x.CreatedDate,

ID = x.ID,

Images = x.Images,

Name = x.Name,

MetaTitle = x.MetaTitle,

Price = x.Price

});

model.OrderByDescending(x => x.CreatedDate).Skip((pageIndex - 1) \* pageSize).Take(pageSize);

return model.ToList();

}

public List<ProductViewModel> Search(string keyword, ref int totalRecord, int pageIndex = 10, int pageSize = 20)

{

totalRecord = db.Products.Where(x => x.Name == keyword).Count();

var model = (from a in db.Products

join b in db.ProductCategories

on a.CategoryID equals b.id

where a.Name.Contains(keyword)

select new

{

CateMetaTitle = b.MetaTitle,

CateName = b.Name,

CreatedDate = a.CreateDate,

ID = a.Id,

Images = a.Image,

Name = a.Name,

MetaTitle = a.MetaTitle,

Price = a.Price

}).AsEnumerable().Select(x => new ProductViewModel()

{

CateMetaTitle = x.MetaTitle,

CateName = x.Name,

CreatedDate = x.CreatedDate,

ID = x.ID,

Images = x.Images,

Name = x.Name,

MetaTitle = x.MetaTitle,

Price = x.Price

});

model.OrderByDescending(x => x.CreatedDate).Skip((pageIndex - 1) \* pageSize).Take(pageSize);

return model.ToList();

}

/// <summary>

/// List feature product

/// </summary>

/// <param name="top"></param>

/// <returns></returns>

public List<Product> ListFeatureProduct(int top)

{

return db.Products.Where(x => x.TopHot != null && x.TopHot > DateTime.Now).OrderByDescending(x => x.CreateDate).Take(top).ToList();

}

public List<Product> ListRelatedProducts(long productId)

{

var product = db.Products.Find(productId);

return db.Products.Where(x => x.Id != productId && x.CategoryID == product.CategoryID).ToList();

}

public void UpdateImages(long productId, string images)

{

var product = db.Products.Find(productId);

product.MoreImage = images;

db.SaveChanges();

}

public Product ViewDetail(long id)

{

return db.Products.Find(id);

}

}

}

HomeController:

using Model.Dao;

using OnlineShop.Common;

using OnlineShop.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.UI;

namespace OnlineShop.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

ViewBag.Slides = new SlideDao().ListAll();

var productDao = new ProductDao();

ViewBag.NewProducts = productDao.ListNewProduct(28);

ViewBag.ListFeatureProducts = productDao.ListFeatureProduct(28);

return View();

}

[ChildActionOnly]

[OutputCache(Duration = 3600 \* 24)]

public ActionResult MainMenu()

{

var model = new MenuDao().ListByGroupId(1);

return PartialView(model);

}

[ChildActionOnly]

[OutputCache(Duration = 3600 \* 24)]

public ActionResult TopMenu()

{

var model = new MenuDao().ListByGroupId(2);

return PartialView(model);

}

[ChildActionOnly]

public PartialViewResult HeaderCart()

{

var cart = Session[CommonConstants.CartSession];

var list = new List<CartItem>();

if (cart != null)

{

list = (List<CartItem>)cart;

}

return PartialView(list);

}

[ChildActionOnly]

[OutputCache(Duration = 3600 \* 24)]

public ActionResult Footer()

{

var model = new FooterDao().GetFooter();

return PartialView(model);

}

}

}

ProductController:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using Model.Dao;

namespace OnlineShop.Controllers

{

public class ProductController : Controller

{

// GET: Product

public ActionResult Index()

{

return View();

}

[ChildActionOnly]

public PartialViewResult ProductCategory()

{

var model = new ProductCategoryDao().ListAll();

return PartialView(model);

}

public JsonResult ListName(string q)

{

var data = new ProductDao().ListName(q);

return Json(new

{

data = data,

status = true

},JsonRequestBehavior.AllowGet);

}

public ActionResult Category(Int32 cateId, int page = 10, int pageSize = 10)

{

var category = new CategoryDao().ViewDetail(cateId);

ViewBag.Category = category;

int totalRecord = 0;

var model = new ProductDao().ListByCategoryId(cateId, ref totalRecord, page, pageSize);

ViewBag.Total = totalRecord;

ViewBag.Page = page;

int maxPage = 100;

int totalPage = 0;

totalPage = (int)Math.Ceiling((double)(totalRecord / pageSize));

ViewBag.TotalPage = totalPage;

ViewBag.MaxPage = maxPage;

ViewBag.First = 1;

ViewBag.Last = totalPage;

ViewBag.Next = page + 1;

ViewBag.Prev = page - 1;

return View(model);

}

public ActionResult Search(string keyword, int page = 10, int pageSize = 10)

{

int totalRecord = 0;

var model = new ProductDao().Search(keyword, ref totalRecord, page, pageSize);

ViewBag.Total = totalRecord;

ViewBag.Page = page;

ViewBag.Keyword = keyword;

int maxPage = 20;

int totalPage = 0;

totalPage = (int)Math.Ceiling((double)(totalRecord / pageSize));

ViewBag.TotalPage = totalPage;

ViewBag.MaxPage = maxPage;

ViewBag.First = 1;

ViewBag.Last = totalPage;

ViewBag.Next = page + 1;

ViewBag.Prev = page - 1;

return View(model);

}

[OutputCache(CacheProfile = "Cache1DayForProduct")]

public ActionResult Detail(long id)

{

var product = new ProductDao().ViewDetail(id);

ViewBag.Category = new ProductCategoryDao().ViewDetail(product.CategoryID.Value);

ViewBag.RelatedProducts = new ProductDao().ListRelatedProducts(id);

return View(product);

}

}

}

Hiển thị danh sách sản phẩm html:

<div class="content\_top">

<div class="heading">

<h3>Sản phẩm mới</h3>

</div>

<div class="see">

<p><a href="/san-pham">See all Products</a></p>

</div>

<div class="clear"></div>

</div>

<div class="section group">

@foreach (var item in listNewProducts)

{

var url = "/chi-tiet/" + item.MetaTitle + "-" + item.Id;

<div class="grid\_1\_of\_4 images\_1\_of\_4">

<a href="/san-pham"><img src="@item.Image" alt="@item.Name" /></a>

<h2>@item.Name</h2>

<div class="price-details">

<div class="price-number">

<p><span class="rupees">@(item.Price.HasValue ? item.Price.Value.ToString("N0") : "Liên hệ")</span></p>

</div>

<div class="add-cart">

<h4><a href="/them-gio-hang?productId=@item.Id&quantity=1">Thêm vào giỏ</a></h4>

</div>

<div class="clear"></div>

</div>

</div>

}

</div>

<div class="content">

<div class="content\_top">

<div class="heading">

<h3>Sản phẩm HOT</h3>

</div>

<div class="see">

<p><a href="/san-pham">See all Products</a></p>

</div>

<div class="clear"></div>

</div>

<div class="section group">

@foreach (var item in listNewProducts)

{

var url = "/chi-tiet/" + item.MetaTitle + "-" + item.Id;

<div class="grid\_1\_of\_4 images\_1\_of\_4">

<a href="/san-pham"><img src="@item.Image" alt="@item.Name" /></a>

<h2>@item.Name</h2>

<div class="price-details">

<div class="price-number">

<p><span class="rupees">@(item.Price.HasValue ? item.Price.Value.ToString("N0") : "Liên hệ")</span></p>

</div>

<div class="add-cart">

<h4><a href="/them-gio-hang?productId=@item.Id&quantity=1">Thêm vào giỏ</a></h4>

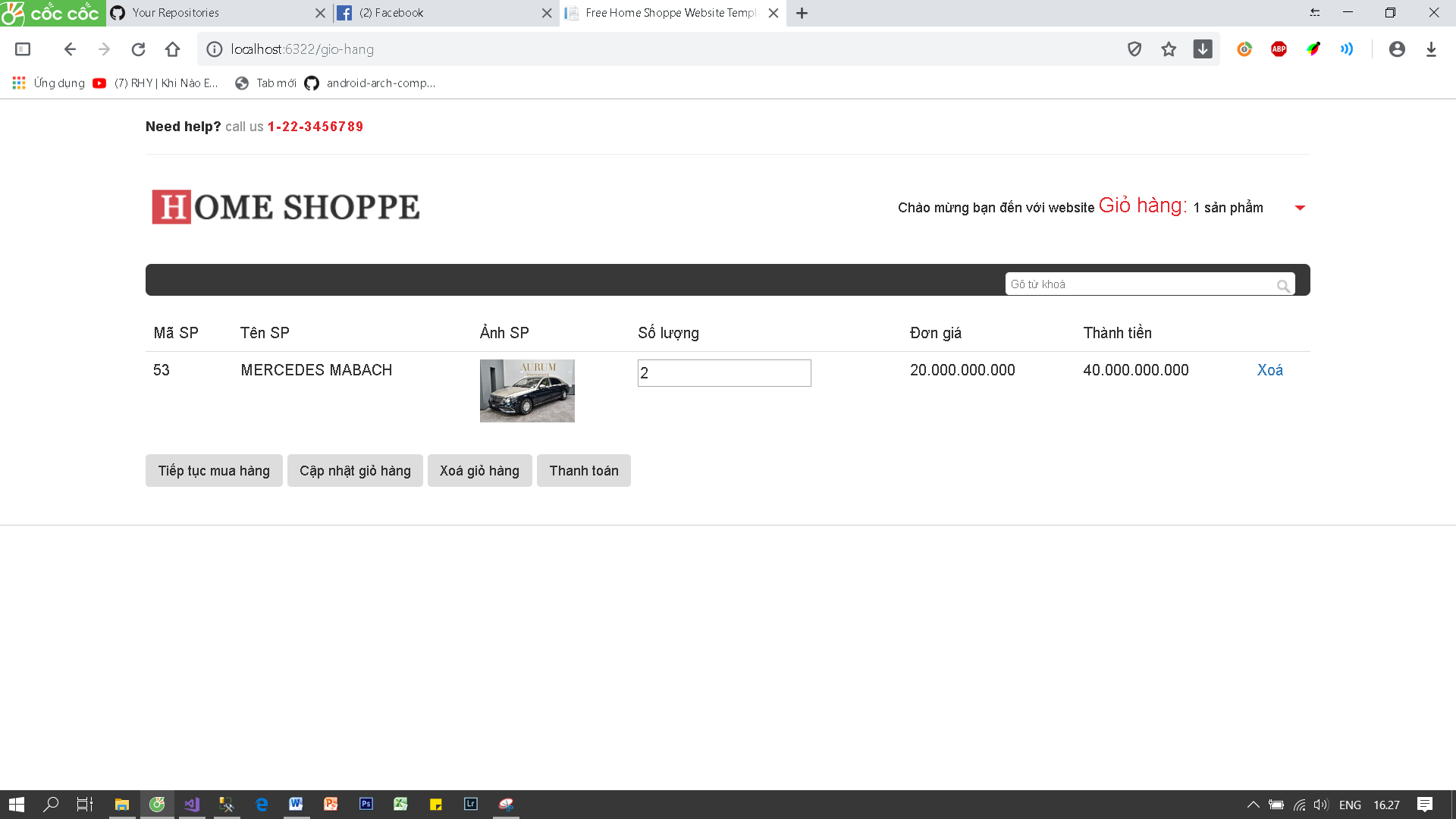
</div>

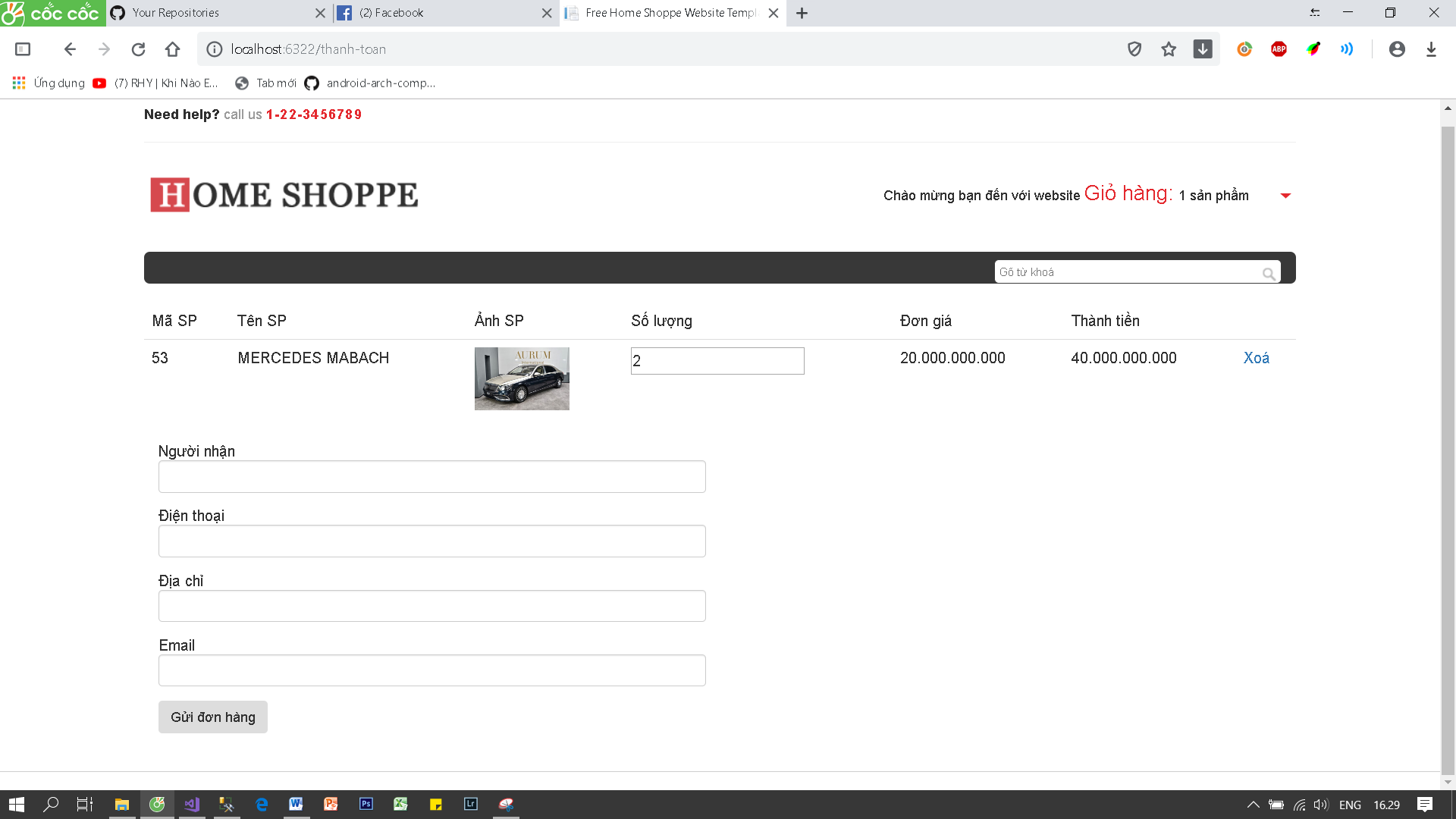
<div class="clear"></div>

</div>

</div>

Giỏ hàng và thanh toán :





CartController:

using Model.Dao;

using OnlineShop.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Script.Serialization;

using Model.EF;

using Common;

using System.Configuration;

using System.IO;

namespace OnlineShop.Controllers

{

public class CartController : Controller

{

private const string CartSession = "CartSession";

// GET: Cart

public ActionResult Index()

{

var cart = Session[CartSession];

var list = new List<CartItem>();

if (cart != null)

{

list = (List<CartItem>)cart;

}

return View(list);

}

public JsonResult DeleteAll()

{

Session[CartSession] = null;

return Json(new

{

status = true

});

}

public JsonResult Delete(long id)

{

var sessionCart = (List<CartItem>)Session[CartSession];

sessionCart.RemoveAll(x => x.Product.Id == id);

Session[CartSession] = sessionCart;

return Json(new

{

status = true

});

}

public JsonResult Update(string cartModel)

{

var jsonCart = new JavaScriptSerializer().Deserialize<List<CartItem>>(cartModel);

var sessionCart = (List<CartItem>)Session[CartSession];

foreach (var item in sessionCart)

{

var jsonItem = jsonCart.SingleOrDefault(x => x.Product.Id == item.Product.Id);

if (jsonItem != null)

{

item.Quantity = jsonItem.Quantity;

}

}

Session[CartSession] = sessionCart;

return Json(new

{

status = true

});

}

public ActionResult AddItem(long productId, int quantity)

{

var product = new ProductDao().ViewDetail(productId);

var cart = Session[CartSession];

if (cart != null)

{

var list = (List<CartItem>)cart;

if (list.Exists(x => x.Product.Id == productId))

{

foreach (var item in list)

{

if (item.Product.Id == productId)

{

item.Quantity += quantity;

}

}

}

else

{

//tạo mới đối tượng cart item

var item = new CartItem();

item.Product = product;

item.Quantity = quantity;

list.Add(item);

}

//Gán vào session

Session[CartSession] = list;

}

else

{

//tạo mới đối tượng cart item

var item = new CartItem();

item.Product = product;

item.Quantity = quantity;

var list = new List<CartItem>();

list.Add(item);

//Gán vào session

Session[CartSession] = list;

}

return RedirectToAction("Index");

}

[HttpGet]

public ActionResult Payment()

{

var cart = Session[CartSession];

var list = new List<CartItem>();

if (cart != null)

{

list = (List<CartItem>)cart;

}

return View(list);

}

[HttpPost]

public ActionResult Payment(string shipName,string mobile,string address,string email)

{

var order = new Order();

order.CreatedDate = DateTime.Now;

order.ShipAddress = address;

order.ShipMobile = mobile;

order.ShipName = shipName;

order.ShipEmail = email;

try

{

var id = new OrderDao().Insert(order);

var cart = (List<CartItem>)Session[CartSession];

var detailDao = new Model.Dao.OrderDetailDao();

decimal total = 0;

foreach (var item in cart)

{

var orderDetail = new OrderDetail();

orderDetail.ProductID = item.Product.Id;

orderDetail.OrderID = id;

orderDetail.Price = item.Product.Price;

orderDetail.Quantity = item.Quantity;

detailDao.Insert(orderDetail);

total += (item.Product.Price.GetValueOrDefault(0) \* item.Quantity);

}

string content = System.IO.File.ReadAllText(Server.MapPath("~/assets/client/template/neworder.html"));

content = content.Replace("{{CustomerName}}", shipName);

content = content.Replace("{{Phone}}", mobile);

content = content.Replace("{{Email}}", email);

content = content.Replace("{{Address}}", address);

content = content.Replace("{{Total}}", total.ToString("N0"));

var toEmail = ConfigurationManager.AppSettings["ToEmailAddress"].ToString();

new MailHelper().SendMail(email, "Đơn hàng mới từ OnlineShop", content);

new MailHelper().SendMail(toEmail, "Đơn hàng mới từ OnlineShop", content);

}

catch (Exception ex)

{

//ghi log

//return Redirect("/loi-thanh-toan");

return Redirect("/hoan-thanh");

}

return Redirect("/hoan-thanh");

}

public ActionResult Success()

{

return View();

}

}

}

Html Giỏ hàng:

@if (Model.Count > 0)

{

<table class="table">

<thead>

<tr>

<td>Mã SP</td>

<td>Tên SP</td>

<td>Ảnh SP</td>

<td>Số lượng</td>

<td>Đơn giá</td>

<td>Thành tiền</td>

<td></td>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td>@item.Product.Id</td>

<td>@item.Product.Name</td>

<td><img src="@item.Product.Image" width="100" /></td>

<td><input type="text" class="txtQuantity" data-id="@item.Product.Id" value="@item.Quantity" /></td>

<td>@item.Product.Price.GetValueOrDefault(0).ToString("N0")</td>

<td>@((item.Product.Price.GetValueOrDefault(0) \* item.Quantity).ToString("N0"))</td>

<td><a href="#" data-id="@item.Product.Id" class="btn-delete">Xoá</a></td>

</tr>

}

</tbody>

</table>

<button id="btnContinue" class="btn">Tiếp tục mua hàng</button>

<button id="btnUpdate" class="btn">Cập nhật giỏ hàng</button>

<button id="btnDeleteAll" class="btn">Xoá giỏ hàng</button>

<button id="btnPayment" class="btn">Thanh toán</button>

}

else

{

<span class="alert alert-danger">Chưa có sản phẩm nào trong giỏ hàng</span>

}

Html Thanh Toán:

@if (Model.Count > 0)

{

<table class="table">

<thead>

<tr>

<td>Mã SP</td>

<td>Tên SP</td>

<td>Ảnh SP</td>

<td>Số lượng</td>

<td>Đơn giá</td>

<td>Thành tiền</td>

<td></td>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td>@item.Product.Id</td>

<td>@item.Product.Name</td>

<td><img src="@item.Product.Image" width="100" /></td>

<td><input type="text" class="txtQuantity" data-id="@item.Product.Id" value="@item.Quantity" /></td>

<td>@item.Product.Price.GetValueOrDefault(0).ToString("N0")</td>

<td>@((item.Product.Price.GetValueOrDefault(0) \* item.Quantity).ToString("N0"))</td>

<td><a href="#" data-id="@item.Product.Id" class="btn-delete">Xoá</a></td>

</tr>

}

</tbody>

</table>

}

@using (Html.BeginForm("Payment", "Cart", FormMethod.Post))

{

<div class="form-group">

<label>Người nhận</label>

<input name="shipName" class="form-control" />

</div>

<div class="form-group">

<label>Điện thoại</label>

<input name="mobile" class="form-control" />

</div>

<div class="form-group">

<label>Địa chỉ</label>

<input name="address" class="form-control" />

</div>

<div class="form-group">

<label>Email</label>

<input name="email" class="form-control" />

</div>

<button type="submit" class="btn">Gửi đơn hàng</button>

}