CartController:

using BookShop.Models;

using Model.DAO;

using Model.EF;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Script.Serialization;

namespace BookShop.Controllers

{

public class CartController : Controller

{

private const string CartSession = "CartSession";

// GET: Cart

public ActionResult Index()

{

//hiển thị giỏ

var cart = Session[CartSession];

var list = new List<CartItem>();

if (cart != null)

{

list =(List<CartItem>)cart;

}

return View(list);

}

//Thêm mới sản phẩm vào giỏ

public ActionResult AddItem(long BookID, int quantity)

{

var book = new BookDAO().ViewDetail(BookID);

var cart = Session[CartSession];

if (cart != null)//Nếu giỏ đã chứa sản phẩm

{

var list = (List<CartItem>)cart;// , lấy từ session ra và ép kiểu sang kiểu ép kiểu của Lis<CartItem>

if (list.Exists(x => x.Book.ID == BookID))//nếu giỏ đã chứa sản phẩm có ID = BookID

{

foreach(var item in list)

{

if(item.Book.ID == BookID)

{

item.Quantity += quantity;

}

}

}

else

{

//Tạo đối tượng mới

var item = new CartItem();

item.Book = book;

item.Quantity = quantity;

list.Add(item);

}

}

else//nếu giỏ hàng trống

{//tao moi doi tuong cartItem

var item = new CartItem();

item.Book = book;

item.Quantity = quantity;

var list = new List<CartItem>();

list.Add(item);

// gan vao session

Session[CartSession] = list;

}

return RedirectToAction("Index");

}

public JsonResult Edit(string cartModel)// cartModel đúng tên của biến trong Ajax

{

var JsonCart = new JavaScriptSerializer().Deserialize<List<CartItem>>(cartModel);// truyền CartModel vào, kiểu là kiểu List<CartItem>

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

foreach(var item in cartSec)

{

var jsonItem = JsonCart.SingleOrDefault(x => x.Book.ID == item.Book.ID);

if (jsonItem != null)

{

item.Quantity = jsonItem.Quantity;

}

}

Session[CartSession] = cartSec;

return Json(new

{

status = true

});

}

//Xóa toàn bộ giỏ hàng

public JsonResult DeleteAll()

{

Session[CartSession] = null;

return Json(new

{

status = true

});

}

//Xóa từng sản phẩm

public JsonResult Delete(long id)

{

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

cartSec.RemoveAll(x => x.Book.ID == id);

Session[CartSession] = cartSec;

return Json(new

{

status = true

});

}

[HttpGet]

public ActionResult Payment()

{

if(Session["userSession"] == null)

{

setAlert("Vui lòng đăng nhập vào tài khoản", "error");

return Redirect("/");

}

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();

if (Session["UserID"] != null)

{

order.CreatID = (long)Session["UserID"];

}

order.CreateDate = DateTime.Now;

order.ShipName = ShipName;

order.ShipMobile = mobile;

order.ShipAdress = address;

order.ShipEmail = email;

order.Status = 1;

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

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

var detailDao = new Order\_DetailDAO();

foreach(var item in cart)

{

var orderdetail = new Order\_Detail();

orderdetail.BookID = item.Book.ID;

orderdetail.Quantity = item.Quantity;

orderdetail.OrderID = id;

orderdetail.Price = item.Book.Price;

detailDao.Insert(orderdetail);

}

Session[CartSession] = null;

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

}

//Cập nhật lại lượt mua sách khi khách hàng thanh toán

public JsonResult Success(string id)

{

var IDPay = new JavaScriptSerializer().Deserialize<List<CartItem>>(id);

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

foreach(var item in cartSec)

{

var json = IDPay.SingleOrDefault(x => x.Book.ID == item.Book.ID);

item.Book.Buys++;

json.Book.Buys = item.Book.Buys;

new BookDAO().UpdateBuys(json.Book.ID);

}

return Json(new {

status = true

});

}

public ActionResult Payment\_success()

{

return View();

}

public void setAlert(string message, string type)

{

TempData["AlertMessage"] = message;

if (type == "success")

{

TempData["AlertType"] = "alert-success";

}

else if (type == "error")

{

TempData["AlertType"] = "alert-danger";

}

}

}

}

Index.cshtml

@model List<BookShop.Models.CartItem>

@{

ViewBag.Title = "Index";

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

var sum = new long();

}

@section cartSection{

<script src="/Assets/Client/js/Controller/CartController.js"></script>

}

<h2 class="text-center">Giỏ hàng của bạn</h2>

<div class="container">

@if (Model.Count > 0)

{

<table id="cart" class="table table-hover table-condensed">

<thead>

<tr>

<th style="width:50%">Tên sản phẩm</th>

<th style="width:10%">Giá</th>

<th style="width:8%">Số lượng</th>

<th style="width:22%" class="text-center">Thành tiền</th>

<th style="width:10%"> </th>

</tr>

</thead>

<tbody>

@foreach (var item in Model)

{

<tr>

<td data-th="Product">

<div class="row">

<div class="col-sm-2 hidden-xs">

<img src="@item.Book.Image" alt="@item.Book.Name" class="img-responsive" width="100">

</div>

<div class="col-sm-10">

<h4 class="nomargin">@item.Book.Name</h4>

<p>@item.Book.Author</p>

</div>

</div>

</td>

<td data-th="Price">@item.Book.Price.Value.ToString("N0") ₫</td>

<td data-th="Quantity">

<div class="form-control text-center">

<input class="txtQuantity" value="@item.Quantity" type="number" data-id="@item.Book.ID">

</div>

</td>

<td data-th="Subtotal" class="text-center">@((item.Book.Price.GetValueOrDefault(0)\*item.Quantity).ToString("N0")) ₫</td>

<td class="actions" data-th="">

<div class="btn-Edit" title="Sửa số lượng">

<a class="btn btn-info btn-sm">

<i class="fa fa-edit"></i>@\*button sửa sản phẩm\*@

</a>

</div>

<div class="btn-Delete" title="Xóa sản phẩm" data-id="@item.Book.ID">

<a class="btn btn-danger btn-sm">

<i class="fa fa-trash-o"></i>@\*button xóa sản phẩm\*@

</a>

</div>

</td>

</tr>

sum += (long)(item.Book.Price.GetValueOrDefault(0) \* item.Quantity);

}

</tbody>

<tfoot>

<tr>

<td>

<a class="btn btn-success" id="btnDeleteAll"><i class="fa fa-times"style="color:#ffd800; margin-right:2px;"></i>Xóa giỏ hàng</a>

</td>

<td>

<a href="/" class="btn btn-warning"><i class="fa fa-angle-left"></i> Tiếp tục mua hàng</a>

</td>

<td colspan="2" class="hidden-xs"></td>

<td class="hidden-xs text-center">

<strong>Tổng tiền: @sum.ToString("N0") ₫ </strong>

</td>

<td>

<a href="/Cart/Payment" class="btn btn-success btn-block">Thanh toán <i class="fa fa-angle-right"></i></a>

</td>

</tr>

</tfoot>

</table>

}

else

{

<tfoot style="margin-left:300px;">

<tr>

<td>

<a href="/" class="btn btn-warning"><i class="fa fa-angle-left"></i> Tiếp tục mua hàng</a>

</td>

</tr>

</tfoot>

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

}

</div>