



using eStoreViewModels;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Web;

using System.Web.Mvc;

namespace eStoreWebsite.Helpers

{

public static class ViewCartHelper

{

public static HtmlString ViewCart(this HtmlHelper helper, string id)

{

//Create tag builder

var builder = new TagBuilder("table class='row text-center'style='margin-left: 0px; border-spacing: 15px; width: 100%; text-align: center;'");

StringBuilder innerHtml = new StringBuilder();

//Create valid id

builder.GenerateId(id);

//Render tag

if (HttpContext.Current.Session["Cart"] != null) // havn't been to the db yet

{

//Populate Data Members and Create Math Variables

CartItem[] cart = (CartItem[])HttpContext.Current.Session["Cart"];

double itemTotal = 0;

double cartTotal = 0;

double tax = 0;

double orderTotal = 0;

#region Header

innerHtml.Append("<tr style='text-align: center;'>");

innerHtml.Append("<th style='text-align: center;'><strong>Product Code</strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong>Product Name</strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong>MSRP</strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong>QTY</strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong>Extended</strong></th>");

innerHtml.Append("</tr>");

#endregion

foreach (CartItem item in cart)

{

if (item.Qty > 0)

{

#region Body

itemTotal = Convert.ToDouble(item.Msrp \* item.Qty);

innerHtml.Append("<tr style='padding:5px;'>");

innerHtml.Append("<td>" + item.ProdCd + "</td>");

innerHtml.Append("<td>" + item.ProdName + "</td>");

innerHtml.Append("<td>$" + Round(item.Msrp) + "</td>");

innerHtml.Append("<td>" + item.Qty + "</td>");

innerHtml.Append("<td>$" + Round(itemTotal) + "</td>");

innerHtml.Append("</tr>");

#endregion

cartTotal += itemTotal;

}

tax = cartTotal \* 0.13;

orderTotal += tax;

}

#region Footer

innerHtml.Append("<tr style='text-align: center;'>");

innerHtml.Append("<th style='text-align: center;'><strong></strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong></strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong></strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong></strong></th>");

innerHtml.Append("<th style='text-align: center;'><strong>-------</strong></th>");

innerHtml.Append("</tr>");

innerHtml.Append("<tr style='text-align: center;'>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: right;'>Total:</th>");

innerHtml.Append("<th style='text-align: right'>"+cartTotal.ToString("C2")+"</th>");

innerHtml.Append("</tr>");

innerHtml.Append("<tr style='text-align: center;'>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: right;'>Tax:</th>");

innerHtml.Append("<th style='text-align: right;'>" + tax.ToString("C2") + "</th>");

innerHtml.Append("</tr>");

innerHtml.Append("<tr style='text-align: center;'>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: center;'></th>");

innerHtml.Append("<th style='text-align: right;'>Order Total:</th>");

innerHtml.Append("<th style='text-align: right;'>" + orderTotal.ToString("C2") + "</th>");

innerHtml.Append("</tr>");

#endregion

}

else

{

innerHtml.Append("<h2 style='text-align: center;'>Sorry, Your Cart is Empty!</h2>");

}

builder.InnerHtml = innerHtml.ToString();

return new HtmlString(builder.ToString());

}

private static string Round(decimal itemTotal)

{

double roundedValue = (Convert.ToDouble(itemTotal) + 0.005) \* 100;

roundedValue = (int)roundedValue;

return Convert.ToString(roundedValue / 100.00);

}

private static string Round(double itemTotal)

{

double roundedValue = (itemTotal + 0.005) \* 100;

roundedValue = (int)roundedValue;

return Convert.ToString(roundedValue / 100.00);

}

}

}