**Lab Assignment 8 – Sarveshwaran Sampathkumar**

**Payment.cshtml**

@model Ch24ShoppingCartMVC.Models.CartViewModel

@{

ViewBag.Title = "Payment";

}

@section css {

<link href="~/Content/Cart.css" rel="stylesheet" />

}

@using (@Html.BeginForm("ShippingConf", "Checkout", FormMethod.Post))

{

<h2>Payment Information</h2>

<br />

<div class="form-horizontal">

<h3>Payment Method</h3>

<table>

<tr>

<td><h3>@Html.ActionLink("PayPal", "/PayPal", "Checkout")</h3></td>

</tr>

</table>

</div>

<h3>Enter Card Details</h3>

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.CreditCardNumber, htmlAttributes: new { @class = "control-label col-md-3" })

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

@Html.EditorFor(model => model.CreditCardNumber, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CreditCardNumber, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.CardType, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.CardType, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CardType, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.CardholderName, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.CardholderName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CardholderName, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.CardProvider, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.CardProvider, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.CardProvider, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.ExpirationDate, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.ExpirationDate, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.ExpirationDate, "", new { @class = "text-danger" })

</div>

</div>

<br />

<br />

<input type="submit" id="btnAdd" value="Place Order" />

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**List.cshtml**

@model Ch24ShoppingCartMVC.Models.CartViewModel

@{

ViewBag.Title = "Checkout";

}

@section css {

<link href="~/Content/Cart.css" rel="stylesheet" />

}

@{ decimal totalPrice = 0.0m;}

@{ decimal taxPrice = 0.0m;}

<h2>Order Information:</h2>

@if (Model.Cart.Count == 0)

{

<p>No items to Checkout. Please go to order page and add items for checkout</p>

}

else

{

foreach (var item in @Model.Cart)

{

<table style="width:100%">

<tr align=left>

<th>Product Name</th>

<th>Individual Price</th>

<th>Quantity</th>

<th>Total Product Price</th>

</tr>

<tr align=left>

<td>@item.Name</td>

<td>$@item.UnitPrice.ToString("0.00")</td>

<td>@item.Quantity</td>

<td>@((@item.Quantity \* @item.UnitPrice).ToString("c"))</td>

</tr>

</table>

totalPrice += @item.Quantity \* @item.UnitPrice;

}

<br />

<h2>Order Subtotal:</h2>

<span>$@totalPrice.ToString("0.00")</span>

<br />

<br />

taxPrice = totalPrice \* 0.075m;

<h2>Tax Information:</h2>

<table style="width:100%">

<tr align=left>

<th>Tax Rate</th>

<th>Estimated Tax to be Collected</th>

</tr>

<tr align=left>

<td>7.5%</td>

<td>$<span>@taxPrice.ToString("0.00")</span></td>

</tr>

</table>

totalPrice += taxPrice;

<br />

<h2>Order Total:</h2>

<span>$@totalPrice.ToString("0.00")</span>

<br />

<br />

using (Html.BeginForm("SubmitCheckout", "Checkout", FormMethod.Post))

{

<h2>Shipping Address:</h2>

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.FirstName, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.FirstName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.FirstName, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.LastName, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.LastName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.LastName, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Address, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.Address, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Address, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.City, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.City, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.City, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.State, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.State, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.State, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.PostalCode, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.PostalCode, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.PostalCode, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Country, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.Country, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Country, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.PhoneNumber, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.PhoneNumber, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.PhoneNumber, "", new { @class = "text-danger" })

</div>

</div>

<br />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.Email, htmlAttributes: new { @class = "control-label col-md-2" })

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

@Html.EditorFor(model => model.Email, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.Email, "", new { @class = "text-danger" })

</div>

</div>

<br />

<br />

<input type="submit" id="btnAdd" value="Add Payment" />

}

}

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Confirmation.cshtml**

@model Ch24ShoppingCartMVC.Models.CartViewModel

@{

ViewBag.Title = "Confirmation";

}

<h2>Confirmation Page</h2>

<p>Thank you for shoppig with us!</p>

<br />

<p>Your Order is Successfully Placed and its currently being processed...!</p>

<br />

<p>You will receive an email containing your order information shortly.</p>

<br />

<p>Thank You</p>

<br />

<p>Have a lovely day!</p>

**CheckoutViewModel.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Ch24ShoppingCartMVC.Models.ViewModels

{

public class CheckoutViewModel

{

[RegularExpression(@"[0-9]+-", ErrorMessage = "The Credit Card number must be 16 digits")]

[Display(Name = "Credit Card Number")]

[Required]

[StringLength(16, MinimumLength = 16)]

public string CreditCardNumber { get; set; }

[Display(Name = "Cardholder Name")]

[Required]

[StringLength(60, MinimumLength = 2)]

public string CardholderName { get; set; }

[Display(Name = "Card Type")]

[Required]

[StringLength(6, MinimumLength = 5)]

public string CardType { get; set; }

[Display(Name = "Card Provider")]

[Required]

[StringLength(30, MinimumLength = 4)]

public string CardProvider { get; set; }

[Display(Name = "Expiration Date")]

[Required]

[DataType(DataType.Date)]

[DisplayFormat(DataFormatString = "{MM}")]

public DateTime ExpirationDate { get; set; }

[Display(Name = "First Name")]

[Required]

[StringLength(100, MinimumLength = 4, ErrorMessage = "The Name must be at least 4 - 100 "

+ "characters long.")]

public string FirstName { get; set; }

[Display(Name = "Last Name")]

[Required]

[StringLength(100, MinimumLength = 4, ErrorMessage = "The Last Name must be at least 4 - 100 "

+ "characters long.")]

public string LastName { get; set; }

[Display(Name = "Address")]

[Required]

[StringLength(100, MinimumLength = 2)]

public string Address { get; set; }

[Display(Name = "City")]

[Required]

[StringLength(100, MinimumLength = 2)]

public string City { get; set; }

[Display(Name = "State")]

[Required]

[StringLength(5, MinimumLength = 2)]

public string State { get; set; }

[Display(Name = "Postal Code")]

[Required]

[StringLength(11, MinimumLength = 5)]

public string PostalCode { get; set; }

[Display(Name = "Country")]

[Required]

[StringLength(60, MinimumLength = 2)]

public string Country { get; set; }

[RegularExpression(@"[0-9]+-", ErrorMessage = "The Phone number must be 10 digits")]

[Display(Name = "Phone Number")]

[Required]

[StringLength(14, MinimumLength = 12)]

public string PhoneNumber { get; set; }

[Display(Name = "Email Address")]

[Required]

[StringLength(100, MinimumLength = 6)]

public string Email { get; set; }

}

}

**CartController.cs**

using Ch24ShoppingCartMVC.Models;

namespace Ch24ShoppingCartMVC.Controllers {

public class CartController : Controller

{

private CartModel cart = new CartModel();

[HttpGet]

public RedirectToRouteResult Index()

{

return RedirectToAction("List/");

}

[HttpGet]

public ViewResult List() {

CartViewModel model = (CartViewModel)TempData["cart"];

//if the model is null, then call the method GetCart

if (model == null)

model = cart.GetCart();

//Passing model to View

return View(model);

}

[HttpPost]

public RedirectToRouteResult List(OrderViewModel order)

{

CartViewModel model = cart.GetCart(order.SelectedProduct.ProductID);

//Assign the quantity of the selected product to the quantity of the added product

model.AddedProduct.Quantity = order.SelectedProduct.Quantity;

//Call the method AddtoCart

cart.AddToCart(model);

//Assign model to the TempData

TempData["cart"] = model;

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

}

}

}

**CheckoutController.cs**

using Ch24ShoppingCartMVC.Models;

using Ch24ShoppingCartMVC.Models.ViewModels;

namespace Ch24ShoppingCartMVC.Controllers

{

public class CheckoutController : Controller

{

CartModel cart = new CartModel();

public RedirectToRouteResult Index()

{

return RedirectToAction("List/");

}

[HttpGet]

public ActionResult List()

{

CartViewModel model = (CartViewModel)TempData["cart"];

if (model == null)

model = cart.GetCart();

return View(model);

}

[HttpPost]

public RedirectToRouteResult List(OrderViewModel order)

{

CartViewModel model = cart.GetCart(order.SelectedProduct.ProductID);

model.AddedProduct.Quantity = order.SelectedProduct.Quantity;

cart.AddToCart(model);

TempData["cart"] = model;

return RedirectToAction("Index", "Checkout");

}

[HttpPost]

public RedirectToRouteResult SubmitCheckout(CartViewModel model)

{

try

{

HalloweenEntities1 halloweenEntities1 = new HalloweenEntities1();

halloweenEntities1.Customers.Add(new Customer { Email = model.Email, FirstName=model.FirstName,LastName = model.LastName, Address = model.Address, City = model.City, State = model.State, ZipCode = model.PostalCode, PhoneNumber = model.PhoneNumber });

halloweenEntities1.SaveChanges();

//return RedirectToAction("PaymentPage", "Checkout");

}

catch(System.Data.Entity.Validation.DbEntityValidationException dbx)

{

}

return RedirectToAction("PaymentPage", "Checkout");

}

[HttpGet]

public ActionResult Confirmation()

{

return View();

}

public ActionResult PaymentPage()

{

return RedirectToAction("Payment", "Checkout");

}

public ActionResult Payment(CartViewModel model)

{

return View();

}

public ActionResult PayPal()

{

return Content("<script>window.location = 'https://www.paypal.com/signin?country.x=US&locale.x=en\_US';</script>");

}

public ActionResult ShippingConf(CartViewModel model)

{

return RedirectToAction("Confirmation", "Checkout");

}

}

}