**Dot Net Phase 4 – Custom Support Logger**

**Phase-End Project Problem Statement**

Github: <https://github.com/11812142/EndProject>

**SourceCode**

**1.View**

**Cart**

**Index.cshtml**

@model CartViewModel

@{

ViewData["Title"] = "Cart Overview";

}

@if (Model.CartItems.Count > 0)

{

<div class="cartWrapper">

<div class="cartbg d-none">

<h3 class="text-center">Redirecting you to paypal</h3>

</div>

<table class="table">

<tr>

<th>Product</th>

<th>Quantity</th>

<th></th>

<th>Price</th>

<th>Total</th>

</tr>

@foreach (var item in Model.CartItems)

{

<tr>

<td>@item.ProductName</td>

<td>@item.Quantity</td>

<td>

<**a** class="btn btn-primary btn-sm" **asp-action**="Add" **asp-route-id**="@item.ProductId">+</**a**>

<**a** class="btn btn-info btn-sm" **asp-action**="Decrease" **asp-route-id**="@item.ProductId">-</**a**>

<**a** class="btn btn-danger btn-sm" **asp-action**="Remove" **asp-route-id**="@item.ProductId">Remove</**a**>

</td>

<td>@item.Price.ToString("C2")</td>

<td>@Model.CartItems.Where(x => x.ProductId == item.ProductId).Sum(x => x.Quantity \* x.Price).ToString("C2")</td>

</tr>

}

<tr>

<td class="text-right" colspan="4">Grand Total: @Model.GrandTotal.ToString("C2")</td>

</tr>

<tr>

<td class="text-right" colspan="4">

<**a** class="btn btn-danger" **asp-action**="Clear">Clear Cart</**a**>

<a class="btn btn-primary checkout" href="#">Checkout</a>

</td>

</tr>

</table>

</div>

}

else

{

<h3 class="display-4 text-center">Your cart is empty.</h3>

}

<**partial** **name**="~/Views/Cart/\_PaypalPartial.cshtml" **for**="CartItems" />

@section Scripts {

<script>

$(function () {

$("a.checkout").click(function (e) {

e.preventDefault();

$("div.cartbg").removeClass("d-none");

$.get("/cart/clear", {}, function () {

$("form.paypalform").submit();

});

});

});

</script>

}

**Home**

**Index.cshtml**

@{

ViewData["Title"] = "Home Page";

}

<div style="background-image: url('D:\velocity\velocity\project\Phase-4-Project-Testing-and-Deployment-main\Phase 4 project- Testing and deployment\PizzaHub\PizzaHub\wwwroot\media\products\Domino's.jpg')" class="text-center">

<h2 class="display-4">Domino's Pizza </h2>

<**img** src="~/media/products/image.jpg" width:"292px" height="173px" ;"/>

<**img** src="~/media/products/Domino's.jpg" width:"292px" heigh"t="173px ;" />

<**img** src="~/media/products/PizzaType.jpg" width:" 292px" height="173px" ;" />

</div>

**Products**

**Index.cshtml**

@model IEnumerable<Product>

@{

ViewData["Title"] = "Products";

}

<h1 background-color="pink">Pizzas</h1>

<div class="row">

@foreach (var item in Model)

{

<div class="col-4">

<img src="/media/products/@item.Image" class="img-fluid" alt="" />

<h4>@item.Name</h4>

<div>

@Html.Raw(item.Description)

</div>

<p>

@item.Price.ToString("C2")

</p>

<p>

<**a** class="btn btn-primary" **asp-controller**="Cart" **asp-action**="Add" **asp-route-id**="@item.Id">Add to cart</**a**>

</p>

</div>

}

@if (ViewBag.TotalPages > 1)

{

<div class="d-flex w-100 justify-content-center">

<**pagination** **page-count**="@ViewBag.TotalPages"

**page-target**="/products/@ViewBag.CategorySlug"

**page-number**="@ViewBag.PageNumber"

**page-range**="@ViewBag.PageRange">

</**pagination**>

</div>

}

</div>

**2.Controller**

**CartConroller.cs**

using Microsoft.AspNetCore.Mvc;

using PizzaHub.Infrastructure;

using PizzaHub.Models;

using PizzaHub.Models.ViewModels;

namespace PizzaHub.Controllers

{

public class CartController : Controller

{

private readonly DataContext \_context;

public CartController(DataContext context)

{

\_context = context;

}

public IActionResult Index()

{

List<CartItem> cart = HttpContext.Session.GetJson<List<CartItem>>("Cart") ?? new List<CartItem>();

CartViewModel cartVM = new()

{

CartItems = cart,

GrandTotal = cart.Sum(x => x.Quantity \* x.Price)

};

return View(cartVM);

}

public async Task<IActionResult> Add(long id)

{

Product product = await \_context.Products.FindAsync(id);

List<CartItem> cart = HttpContext.Session.GetJson<List<CartItem>>("Cart") ?? new List<CartItem>();

CartItem cartItem = cart.Where(c => c.ProductId == id).FirstOrDefault();

if (cartItem == null)

{

cart.Add(new CartItem(product));

}

else

{

cartItem.Quantity += 1;

}

HttpContext.Session.SetJson("Cart", cart);

TempData["Success"] = "The product has been added!";

return Redirect(Request.Headers["Referer"].ToString());

}

public async Task<IActionResult> Decrease(long id)

{

List<CartItem> cart = HttpContext.Session.GetJson<List<CartItem>>("Cart");

CartItem cartItem = cart.Where(c => c.ProductId == id).FirstOrDefault();

if (cartItem.Quantity > 1)

{

--cartItem.Quantity;

}

else

{

cart.RemoveAll(p => p.ProductId == id);

}

if (cart.Count == 0)

{

HttpContext.Session.Remove("Cart");

}

else

{

HttpContext.Session.SetJson("Cart", cart);

}

TempData["Success"] = "The product has been removed!";

return RedirectToAction("Index");

}

public async Task<IActionResult> Remove(long id)

{

List<CartItem> cart = HttpContext.Session.GetJson<List<CartItem>>("Cart");

cart.RemoveAll(p => p.ProductId == id);

if (cart.Count == 0)

{

HttpContext.Session.Remove("Cart");

}

else

{

HttpContext.Session.SetJson("Cart", cart);

}

TempData["Success"] = "The product has been removed!";

return RedirectToAction("Index");

}

public IActionResult Clear()

{

HttpContext.Session.Remove("Cart"); if (HttpContext.Request.Headers["X-Requested-With"] != "XMLHttpRequest")

return Redirect(Request.Headers["Referer"].ToString());

return Ok();

}

}

}

**ProductsConroller.cs**

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

using PizzaHub.Infrastructure;

using PizzaHub.Models;

namespace PizzaHub.Controllers

{

public class ProductsController : Controller

{

private readonly DataContext \_context;

public ProductsController(DataContext context)

{

\_context = context;

}

public async Task<IActionResult> Index(string categorySlug = "", int p = 1)

{

int pageSize = 3;

ViewBag.PageNumber = p;

ViewBag.PageRange = pageSize;

ViewBag.CategorySlug = categorySlug;

if (categorySlug == "")

{

ViewBag.TotalPages = (int)Math.Ceiling((decimal)\_context.Products.Count() / pageSize);

return View(await \_context.Products.OrderByDescending(p => p.Id).Skip((p - 1) \* pageSize).Take(pageSize).ToListAsync());

}

Category category = await \_context.Categories.Where(c => c.Slug == categorySlug).FirstOrDefaultAsync();

if (category == null) return RedirectToAction("Index");

var productsByCategory = \_context.Products.Where(p => p.CategoryId == category.Id);

ViewBag.TotalPages = (int)Math.Ceiling((decimal)productsByCategory.Count() / pageSize);

return View(await productsByCategory.OrderByDescending(p => p.Id).Skip((p - 1) \* pageSize).Take(pageSize).ToListAsync());

}

}

}

**3.Models**

**CartItems.cs**

namespace PizzaHub.Models

{

public class CartItem

{

public long ProductId { get; set; }

public string ProductName { get; set; }

public int Quantity { get; set; }

public decimal Price { get; set; }

public decimal Total

{

get { return Quantity \* Price; }

}

public string Image { get; set; }

public CartItem()

{

}

public CartItem(Product product)

{

ProductId = product.Id;

ProductName = product.Name;

Price = product.Price;

Quantity = 1;

Image = product.Image;

}

}

}

**Category.cs**

namespace PizzaHub.Models

{

public class Category

{

public long Id { get; set; }

public string Name { get; set; }

public string Slug { get; set; }

}

}

**Product.cs**

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace PizzaHub.Models

{

public class Product

{

public long Id { get; set; }

[Required(ErrorMessage = "Please enter a value")]

public string Name { get; set; }

public string Slug { get; set; }

[Required, MinLength(4, ErrorMessage = "Minimum length is 4")]

public string Description { get; set; }

[Required]

[Range(0.01, double.MaxValue, ErrorMessage = "Please enter a value")]

[Column(TypeName = "decimal(8, 2)")]

public decimal Price { get; set; }

public long CategoryId { get; set; }

public Category Category { get; set; }

public string Image { get; set; }

}

}

**View Models**

**CartViewModel.cs**

namespace PizzaHub.Models.ViewModels

{

public class CartViewModel

{

public List<CartItem> CartItems { get; set; }

public decimal GrandTotal { get; set; }

}

}

**SmallCartViewModel.cs**

namespace PizzaHub.Models.ViewModels

{

public class SmallCartViewModel

{

public int NumberOfItems { get; set; }

public decimal TotalAmount { get; set; }

}

}