Source Code

Pizza.cshtml

```
@page
@model PizzaModel
@{
    ViewData["Title"] = "Pizza Selection";
}

<h1>@ViewData["Title"]</h1>

<form method="post" asp-page-handler="AddToCart">
    <label for="pizzaType">Select Pizza Type:</label>
    <select asp-for="SelectedPizza" asp-items="Model.PizzaTypes">
        <option value="">-- Select Pizza --</option>
    </select>

    <label for="quantity">Quantity:</label>
    <input type="number" asp-for="Quantity" min="1" required />

    <button type="submit">Add to Cart</button>
</form>
```

Pizza.cshtml.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.AspNetCore.Mvc.Rendering;
using System.Collections.Generic;

public class PizzaModel : PageModel
{
    [BindProperty]
    public string SelectedPizza { get; set; }

    [BindProperty]
    public int Quantity { get; set; }

    public List<SelectListItem> PizzaTypes { get; } = new List<SelectListItem>
    {
        new SelectListItem("Margherita", "Margherita"),
        new SelectListItem("Pepperoni", "Pepperoni"),
        new SelectListItem("Vegetarian", "Vegetarian"),
        // Add more pizza types as needed
    };
    public IActionResult OnPostAddToCart()
    {
        // Implement the logic to add the selected pizza to the shopping cart
        // You can store the selected pizza and quantity in a session variable or
a database

        // Redirect to the Order Checkout page
        return RedirectToPage("OrderCheckout", new { pizza = SelectedPizza,
quantity = Quantity });
    }
}
```

OrderCheckout.cshtml

```
@page
@model OrderCheckoutModel
@{
    ViewData["Title"] = "Order Checkout";
}

<h1>@ViewData["Title"]</h1>

<p>Pizza: @Model.SelectedPizza</p>
<p>Quantity: @Model.Quantity</p>
<p>Total Price: $@Model.TotalPrice</p>

<form method="post" asp-page-handler="PlaceOrder">
    <button type="submit">Checkout</button>
</form>
```

OrderCheckout.cshtml.cs

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;

public class OrderCheckoutModel : PageModel
{
    [BindProperty(SupportsGet = true)]
    public string SelectedPizza { get; set; }

    [BindProperty(SupportsGet = true)]
    public int Quantity { get; set; }

    public double TotalPrice => CalculateTotalPrice();

    public IActionResult OnPostPlaceOrder()
    {
        // Implement the logic to place the order
        // You can generate an order ID, store the order details in a database, 
etc.

        // Redirect to the Order Confirmation page
        return RedirectToPage("OrderConfirmation", new { orderId = 123, amount = 
TotalPrice, pizza = SelectedPizza });
    }

    private double CalculateTotalPrice()
    {
        // Implement the logic to calculate the total price based on the selected 
pizza and quantity
        // This could involve querying a database for pizza prices or using a 
predefined price list

        // For simplicity, let's assume a fixed price per pizza
        double pizzaPrice = 10.0;
        return Quantity * pizzaPrice;
    }
}
```

OrderConfirmation.cshtml

```
@page
@model OrderConfirmationModel
@{
    ViewData["Title"] = "Order Confirmation";
}

<h1>@ViewData["Title"]</h1>

<p>Thank you for your order!</p>
<p>Order ID: @Model.OrderId</p>
<p>Amount: $@Model.Amount</p>
<p>Selected Pizza: @Model.SelectedPizza</p>
```

OrderConfirmation.cshtml.cs

```
using Microsoft.AspNetCore.Mvc.RazorPages;

public class OrderConfirmationModel : PageModel
{
    public int OrderId { get; set; }
    public double Amount { get; set; }

    public string SelectedPizza { get; set; }

    public void OnGet(int orderId, double amount, string pizza)
    {
        OrderId = orderId;
        Amount = amount;
        SelectedPizza = pizza;
    }
}
```

UnitTest1.cs

```
using CourseEndPizza;
using Microsoft.AspNetCore.Mvc;

namespace Testing
{
    [TestFixture]
    public class PizzaOrderTests
    {
        [Test]
        public void PizzaSelectionPage_ShouldReturnPageResult()
        {
            // Arrange
            var pizzaPageModel = new PizzaModel();

            // Act
            var result = pizzaPageModel.OnPostAddToCart();

            // Assert
            Assert.IsInstanceOf<RedirectToPageResult>(result);
            var redirectResult = (RedirectToPageResult)result;
            Assert.AreEqual("OrderCheckout", redirectResult.PageName);
        }

        [Test]
        public void OrderCheckoutPage_ShouldReturnPageResult()
        {
            // Arrange
            var orderCheckoutModel = new OrderCheckoutModel
            {
                SelectedPizza = "Pepperoni",
                Quantity = 2
            };

            // Act
            var result = orderCheckoutModel.OnPostPlaceOrder();

            // Assert
            Assert.IsInstanceOf<RedirectToPageResult>(result);
            var redirectResult = (RedirectToPageResult)result;
            Assert.AreEqual("OrderConfirmation", redirectResult.PageName);
        }

        // Add more test methods as needed to cover different scenarios
    }
}
```