Source Code Pizza Project:

- Home Controller

```
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.AspNetCore.Identity;
using System.Threading.Tasks;
using PizzaProj1.Data;
using PizzaProj1.Models;
using System.Diagnostics;

namespace PizzaProj1.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;
        private readonly ApplicationDbContext _dbContext;
        private readonly UserManager<IdentityUser> _userManager; // Corrected the
user model type

        public HomeController(ILogger<HomeController> logger, ApplicationDbContext
dbContext, UserManager<IdentityUser> userManager)
        {
            _logger = logger;
            _dbContext = dbContext;
            _userManager = userManager;
        }

        public IActionResult Index()
        {
            var pizzaItems = _dbContext.Pizzas.ToList();
            return View(pizzaItems);
        }

        [HttpPost]
        public async Task<IActionResult> AddToCart(int pizzaId, int quantity) // Added
async and Task for correct usage of await
        {
            IdentityUser user = await _userManager.GetUserAsync(User);
            if (user == null)
            {
                return BadRequest("User is not authenticated.");
```

```
                                    }

                                    string userId = user.Id;

                                    var cartItem = new CartItem
                                    {
                                                PizzaId = pizzaId,
                                                Quantity = quantity,
                                                UserId = userId
                                    };

                                    _dbContext.Carts.Add(cartItem);
                                    _dbContext.SaveChanges();

                                        //HttpContext.Session.SetString("UserId", userId);

                                    return Json(new { success = true, message = "Product added to cart
successfully!" });
                        }




        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore =
true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ??
HttpContext.TraceIdentifier });
        }
    }
}
```

- Checkout COntroller

```
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PizzaProj1.Data;

namespace PizzaProj1.Controllers
{
    public class CheckoutController : Controller
    {
        private readonly ApplicationDbContext _dbContext;
        private readonly UserManager<IdentityUser> _userManager; // Corrected
the user model type


        public CheckoutController(ApplicationDbContext dbContext,
UserManager<IdentityUser> userManager)
        {
            _dbContext = dbContext;
            _userManager = userManager;
        }

        public async Task<IActionResult> Checkout()
        {
            IdentityUser user = await _userManager.GetUserAsync(User);
            //string userId = HttpContext.Session.GetString("UserId");
            string userId = user.Id;
            if (userId != null)
            {
                var cartItems = _dbContext.Carts
                    .Include(cartItem => cartItem.Pizza)
                    .Where(cartItem => cartItem.UserId == userId)
                    .ToList();

                return View(cartItems);
            }

            // If user is not authenticated, you may want to handle this
accordingly
            return RedirectToAction("Index", "Home");

        }

        [HttpPost]
        public async Task<IActionResult> ClearCart()
        {
            IdentityUser user = await _userManager.GetUserAsync(User);
            string userId = user.Id;

            var cartItems = _dbContext.Carts
                .Where(cartItem => cartItem.UserId == userId);
            _dbContext.Carts.RemoveRange(cartItems);
            await _dbContext.SaveChangesAsync();

            return Content("");
            ;
        }

        [HttpPost]
        public async Task<IActionResult> RemoveCartItem(int cartId)
        {
            var cartItem = await _dbContext.Carts.FindAsync(cartId);

            if (cartItem == null)
            {
                return NotFound(); // Return a 404 Not Found if the cart
item is not found
            }

            _dbContext.Carts.Remove(cartItem);
            await _dbContext.SaveChangesAsync();

            return Ok();
        }
    }
}
```

- Confirmation Controller

```
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PizzaProj1.Controllers;
using PizzaProj1.Data;

namespace PizzaProj1.Controllers
{
    public class ConfirmationController : Controller
    {

            private readonly ApplicationDbContext _dbContext;
```

```
        private readonly UserManager<IdentityUser> _userManager; // Corrected the
user model type


        public ConfirmationController(ApplicationDbContext dbContext,
UserManager<IdentityUser> userManager)
        {
            _dbContext = dbContext;
            _userManager = userManager;
        }

        public async Task<IActionResult> ConfirmationPage()
          {
            IdentityUser user = await _userManager.GetUserAsync(User);
            //string userId = HttpContext.Session.GetString("UserId");
            string userId = user.Id;
            if (userId != null)
            {
                var cartItems = _dbContext.Carts
                    .Include(cartItem => cartItem.Pizza)
                    .Where(cartItem => cartItem.UserId == userId)
                    .ToList();

                return View(cartItems);
            }

            // If user is not authenticated, you may want to handle this accordingly
            return RedirectToAction("Index", "Home");

        }
    }
}
```

- Cart Item Model

```
using System.ComponentModel.DataAnnotations;
```

```
namespace PizzaProj1.Models
{
    public class CartItem
    {
            [Key]  // Specify CartId as the primary key

            public int CartId { get; set; }
            public int PizzaId { get; set; }
            public int Quantity { get; set; }
            public string UserId { get; set; }

            public virtual PizzaItem Pizza { get; set; }
    }
}
```

- Pizza Item Model

```
using System.ComponentModel.DataAnnotations;

namespace PizzaProj1.Models
{
    public class PizzaItem
    {
        [Key]
        public int PizzaId { get; set; }
        public string Name { get; set; }
        public decimal Price { get; set; }
        public string Description { get; set; }
        public string ImageURL { get; set;}
    }
}
```

    - index.View

```
@{
    ViewData["Title"] = "Home Page";
}

@{
```

```
            ViewBag.Title = "Index";
}

@model List<PizzaProj1.Models.PizzaItem>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">

<style>

    .product-item {
        border: 1px solid #ccc;
        border-radius: 12px;
        box-shadow: 0 4px 8px rgba(255, 255, 255, 0.1);
        transition: box-shadow 0.3s ease;
        background-color: rgba(255, 255, 255, 0.9); /* Adjust transparency */
        padding: 15px;
        transform: translateY(10px);
        transition: 0.5s;
    }


    .product-item:hover {
            box-shadow: 0 8px 16px rgba(255, 255, 255, 0.1);
            transform: translateY(-10px);
            transition: 0.5s;
    }

    .container {
        padding: 25px 0;
    }



    a {
        color: inherit;
        text-decoration: none;
    }

    a:hover {
        color: inherit;
        text-decoration: none;
    }

    .row {
```

```
/*          display: flex;
        flex-wrap: wrap; */
         margin-bottom:50px;
    }

    .col-md-4 {
        flex: 1;
        margin-bottom: 15px;
    }

    .add-to-cart-button {
        position: absolute;
        bottom: 15px;
        right: 15px;
    }

    .product-quantity .input {
        width: 10px;
    }

    .banner img {
        width: 100%;
        height: auto;

    }

    h1 {
        margin: 0;
    }

</style>


<div class="banner">
            @* <img
src="https://d1csarkz8obe9u.cloudfront.net/posterpreviews/pizza-banner-design-template-5623a47ee70d2ca4f3a4eca9c19a8039_screen.jpg?ts=1572691129"
alt="Banner Image" style="width: 100%;display:block;" /> *@
    <img src="@Url.Content("~/Image/pizza-banner.png")" alt="Banner Image" />
</div>

<div class="container">
            <div class="row">
                        @foreach (var product in Model)
        {
    <div class="col-12 col-md-6 col-lg-4" style="margin-bottom:50px">
                <div class="card product-item h-100">
                        <img src="@product.ImageURL"
alt="@product.Name" class="card-img-top img-fluid" style="height: 200px;">
                        <div class="card-body">
                            <h5
class="card-title">@product.Name</h5>
                            <p
class="card-text">@product.Description</p>
                            <p
class="card-text">$@product.Price</p>
                            <div class="form-group
product-quantity d-flex align-items-center">
                                <label
for="quantity-@product.PizzaId">Qty:</label>
                                <input type="number"
class="form-control" id="quantity-@product.PizzaId" value="1" min="1" max="10"
style="width: 60px; margin-left:5px;">
                                <button class="btn
btn-primary add-to-cart-button ml-auto" id="@product.PizzaId"
onclick="addToCart(@product.PizzaId)" data-pizza-id="@product.PizzaId">Add
to Cart</button>
                            </div>

                        </div>
                </div>
            </div>
        }
    </div>
</div>




<script>
  function addToCart(pizzaId) {

    var quantityInputId = "quantity-" + pizzaId;
    var quantity = parseInt(document.getElementById(quantityInputId).value,
10);

    // Ensure the quantity is at least 1
    if (quantity < 1) {
      alert("Please enter a valid quantity.");
```

```
            return;
        }

        // Send data to the server
        $.ajax({
            type: "POST",
            url: "/Home/AddToCart/"+ pizzaId,

            data: {
                pizzaId: pizzaId,
                quantity: quantity
            },
            success: function (result) {
                // alert("Item added to cart successfully.");
                var button = document.querySelector('[data-pizza-id="' + pizzaId + '"]')
                button.innerHTML = "Added to Cart";
                button.classList.add("btn-success");
                button.disabled = true;

            console.log("Item added to cart successfully.");
            },
            error: function (error) {
                console.error("Error adding item to cart:", error);
            }
        });
    }
</script>
```

- Checkout View

```
@*
    For more information on enabling MVC for empty projects, visit https://go.microsoft.com/fwlink/?LinkID=397860
*@
@{
            ViewData["Title"] = "Checkout";
}

@model List<PizzaProj1.Models.CartItem>
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/bootstrap-icons.css
">

<div class="title">
        <img src="@Url.Content("~/Image/pizzaCart.png")" alt="pizza-cart"
class="pizza-cart"/>
</div>



<style>
        .title {
                text-align: center;
                margin: auto;
                display: flex;
                flex-direction: column;
                align-items: center;
        }

        .pizza-cart {
                width: 250px;
        }

        .td{
                vertical-align:middle;
        }

        .tablecontainer{
                border-radius: 12px;
                max-width:850px;
                margin-top:20px;
        }



</style>

<script>
```

```
    function redirectToConfirmation() {
        window.location.href = '@Url.Action("ConfirmationPage",
"Confirmation")';
    }




    function clearCart() {
        if (confirm("Are you sure you want to clear your cart?")) {
            $.ajax({
                type: "POST",
                url: '@Url.Action("ClearCart", "Checkout")',
                data: {},
                success: function () {
                    // Reload the current page
                    location.reload();
                },
                error: function () {
                    alert("An error occurred while clearing the
cart.");
                }
            });
        }
    }

    //remove cart item as per id
    function removeCartItem(cartId) {
        $.ajax({
            type: "POST",
            url: "/Checkout/RemoveCartItem?cartId=" + cartId,


            success: function () {
                // Reload the page to reflect the updated cart
                window.location.reload();
            },
            error: function (error) {
                console.error("Error removing item from cart:",
error);
            }
        });
    }
```

```
</script>


<div class="container tablecontainer" style="background-color:white">
    <button style="margin:10px 0;" class="btn btn-danger float-right" onclick="clearCart()">Clear Cart</button>
    <table class="table" >
        <thead>
        <tr>
            <th>Pizza</th>
            <th>Name</th>
            <th>Qty</th>
            <th>Price</th>
            <th>Total</th>
            <th></th>
        </tr>
        </thead>
        <tbody>
        @foreach (var item in Model)
        {
            <tr>
                <td style="width: 250px;"><img src="@item.Pizza.ImageURL" alt="@item.Pizza.Name" class="img-fluid"></td>
                <td id="pizzaName">@item.Pizza.Name</td>
                <td>@item.Quantity</td>
                <td>$@item.Pizza.Price</td>
                <td>$@(item.Quantity * item.Pizza.Price)</td>
                <td><button class="btn btn-danger" onclick="removeCartItem(@item.CartId)">X</button></td>
            </tr>
        }

        <tr>
            <td colspan="4" class="text-right"><strong>Total Cost:</strong></td>
            <td id="totalPrice">$@Model.Sum(item => item.Quantity * item.Pizza.Price)</td>
        </tr>
        </tbody>
    </table>

    <button style="padding-right:10px," class="btn btn-primary float-right" onclick="redirectToConfirmation()">Confirm</button>
```

```
</div>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
></script>
```