

```

using System.ComponentModel.DataAnnotations;

namespace Phase3JenkinsProj.Models
{
    public class Pizza
    {
        public int Id { get; set; }

        [Required]
        [Display(Name = "Pizza Name")]
        public string Name { get; set; }

        [Required]
        [Display(Name = "Description")]
        public string Description { get; set; }

        [Display(Name = "Price")]
        public float Price { get; set; }
    }
}

using Microsoft.AspNetCore.Mvc;
using Phase3JenkinsProj.Models;
using System.Collections.Generic;
using System.Linq;

namespace Phase3JenkinsProj.Controllers
{
    public class PizzasController : Controller
    {
        // Existing list of pizzas
        static List<Pizza> pizzas = new List<Pizza>
        {
            new Pizza(){ Id = 1, Name="Margherita", Description="A cheesy Pizza",
Price=150 },
            new Pizza(){ Id = 2, Name="Farmhouse", Description="A Pizza with cheese,
tomatoes, and onions", Price=200 },
            new Pizza(){ Id = 3, Name="Pepperoni", Description="A Pizza with cheese
and pepperoni", Price=250 },
            new Pizza(){ Id = 4, Name="Spaghetti", Description="White sauce pasta",
Price=160 },
            new Pizza(){ Id = 5, Name="Chicken Tikka Pizza", Description="A cheesy
Pizza with chicken tikka cubes", Price=300 }
        };

        public IActionResult Index()
        {
            return View(pizzas);
        }

        [HttpGet]
        public IActionResult Create()
        {
            return View(new Pizza());
        }

        [HttpPost]
        public IActionResult Create(Pizza model)
        {

```

```

        if (ModelState.IsValid)
        {
            pizzas.Add(model);
            return RedirectToAction("Index");
        }
        else
        {
            return View(model);
        }
    }

    [HttpGet]
    public IActionResult Details(int id)
    {
        // Find the pizza by ID
        var pizza = pizzas.FirstOrDefault(p => p.Id == id);

        if (pizza == null)
        {
            return NotFound(); // Return a 404 Not Found response if not found
        }

        return View(pizza);
    }

    [HttpGet]
    public IActionResult Edit(int id)
    {
        // Find the pizza by ID
        var pizza = pizzas.FirstOrDefault(p => p.Id == id);

        if (pizza == null)
        {
            return NotFound(); // Return a 404 Not Found response if not found
        }

        return View(pizza);
    }

    [HttpPost]
    public IActionResult Edit(Pizza model)
    {
        if (ModelState.IsValid)
        {
            // Find the pizza in the list and update its properties
            var pizzaToUpdate = pizzas.FirstOrDefault(p => p.Id == model.Id);

            if (pizzaToUpdate != null)
            {
                pizzaToUpdate.Name = model.Name;
                pizzaToUpdate.Description = model.Description;
                pizzaToUpdate.Price = model.Price;
            }

            return RedirectToAction("Index");
        }
        else
        {

```

```

        return View(model);
    }
}

[HttpGet]
public IActionResult Delete(int id)
{
    // Find the pizza by ID
    var pizza = pizzas.FirstOrDefault(p => p.Id == id);

    if (pizza == null)
    {
        return NotFound(); // Return a 404 Not Found response if not found
    }

    return View(pizza);
}

[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public IActionResult DeleteConfirmed(int id)
{
    // Find and remove the pizza by ID
    var pizzaToDelete = pizzas.FirstOrDefault(p => p.Id == id);

    if (pizzaToDelete != null)
    {
        pizzas.Remove(pizzaToDelete);
    }

    return RedirectToAction("Index");
}
}
}

@model IEnumerable<Phase3JenkinsProj.Models.Pizza>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Name)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Description)
            </th>
            <th>

```

```

                @Html.DisplayNameFor(model => model.Price)
            </th>
        </th></th>
    </tr>
</thead>
<tbody>
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Id)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Description)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", "Pizzas", new { id = item.Id },
null) |
            @Html.ActionLink("Details", "Details", "Pizzas", new { id =
item.Id }, null) |
            @Html.ActionLink("Delete", "Delete", "Pizzas", new { id =
item.Id }, null)
        </td>
    </tr>
}
</tbody>
</table>
@model Phase3JenkinsProj.Models.Pizza

@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>Pizza</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Id" class="control-label"></label>
                <input asp-for="Id" class="form-control" />
                <span asp-validation-for="Id" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group">

```

```

        <label asp-for="Description" class="control-label"></label>
        <input asp-for="Description" class="form-control" />
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="Price" class="control-label"></label>
        <input asp-for="Price" class="form-control" />
        <span asp-validation-for="Price" class="text-danger"></span>
    </div>
    <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
    </div>
</form>
</div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
@model Phase3JenkinsProj.Models.Pizza

@{
    ViewData["Title"] = "Details";
}

<h1>Details</h1>

<div>
    <h4>Pizza</h4>
    <hr />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Description)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Description)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Price)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Price)
        </dd>
    </dl>
</div>

```

```

        </dd>
    </dl>
</div>
<div>
    @Html.ActionLink("Edit", "Edit", new { /* id = Model.PrimaryKey */ }) |
    <a asp-action="Index">Back to List</a>
</div>
@model Phase3JenkinsProj.Models.Pizza

@{
    ViewData["Title"] = "Edit";
}

<h1>Edit</h1>

<h4>Pizza</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Id" class="control-label"></label>
                <input asp-for="Id" class="form-control" />
                <span asp-validation-for="Id" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Description" class="control-label"></label>
                <input asp-for="Description" class="form-control" />
                <span asp-validation-for="Description" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Price" class="control-label"></label>
                <input asp-for="Price" class="form-control" />
                <span asp-validation-for="Price" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
@model Phase3JenkinsProj.Models.Pizza

@{

```

```

    ViewData["Title"] = "Delete";
}

<h1>Delete</h1>

<h3>Are you sure you want to delete this?</h3>
<div>
    <h4>Pizza</h4>
    <hr />
    <dl class="row">
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Id)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Id)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Name)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Name)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Description)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Description)
        </dd>
        <dt class="col-sm-2">
            @Html.DisplayNameFor(model => model.Price)
        </dt>
        <dd class="col-sm-10">
            @Html.DisplayFor(model => model.Price)
        </dd>
    </dl>

    <form asp-action="Delete">
        <input type="submit" value="Delete" class="btn btn-danger" /> |
        <a asp-action="Index">Back to List</a>
    </form>
</div>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - Phase3JenkinsProj</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="~/Phase3JenkinsProj.styles.css" asp-append-
version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-
white border-bottom box-shadow mb-3">
            <div class="container-fluid">

```

```

        <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">Phase3JenkinsProj</a>
        <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" aria-
controls="navbarSupportedContent"
            aria-expanded="false" aria-label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="navbar-collapse collapse d-sm-inline-flex justify-
content-between">
            <ul class="navbar-nav flex-grow-1">
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Index">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-
controller="Pizzas" asp-action="Index">Pizzas</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link text-dark" asp-area="" asp-
controller="Home" asp-action="Privacy">Privacy</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2023 - Phase3JenkinsProj - <a asp-area="" asp-controller="Home"
asp-action="Privacy">Privacy</a>
    </div>
</footer>
<script src="~/lib/jquery/dist/jquery.min.js"></script>
<script src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
<script src="~/js/site.js" asp-append-version="true"></script>
@await RenderSectionAsync("Scripts", required: false)
</body>
</html>

```