Ingredients Implementation

GROUP 4

Mustafa

Eric

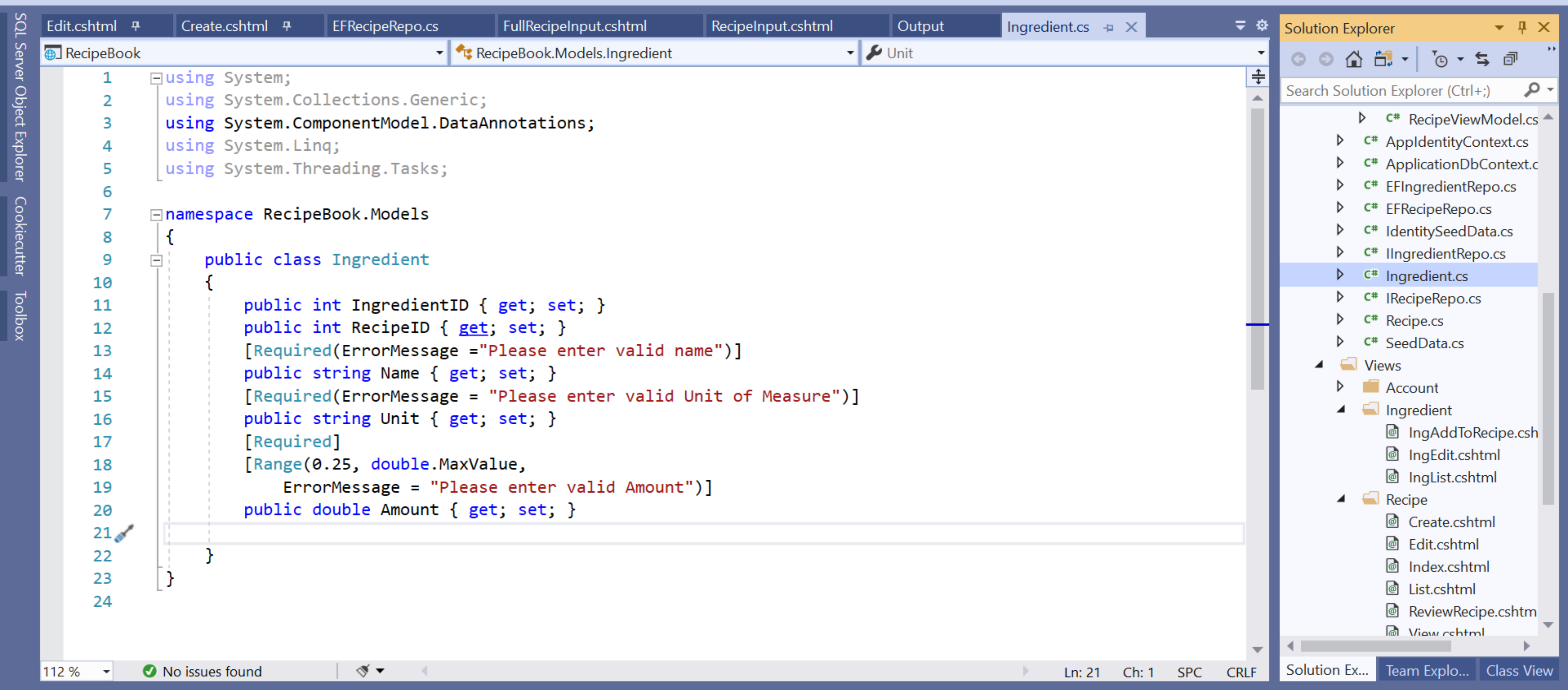
Cao

MODELS:

1. **Ingredient**
2. **IIngredientRepo**
3. **EFIngredientRepo**
4. **RecipeViewModel**
5. **Updated Recipe**
6. **Updated ApplicationDbContext**
7. **Updated SeedData**

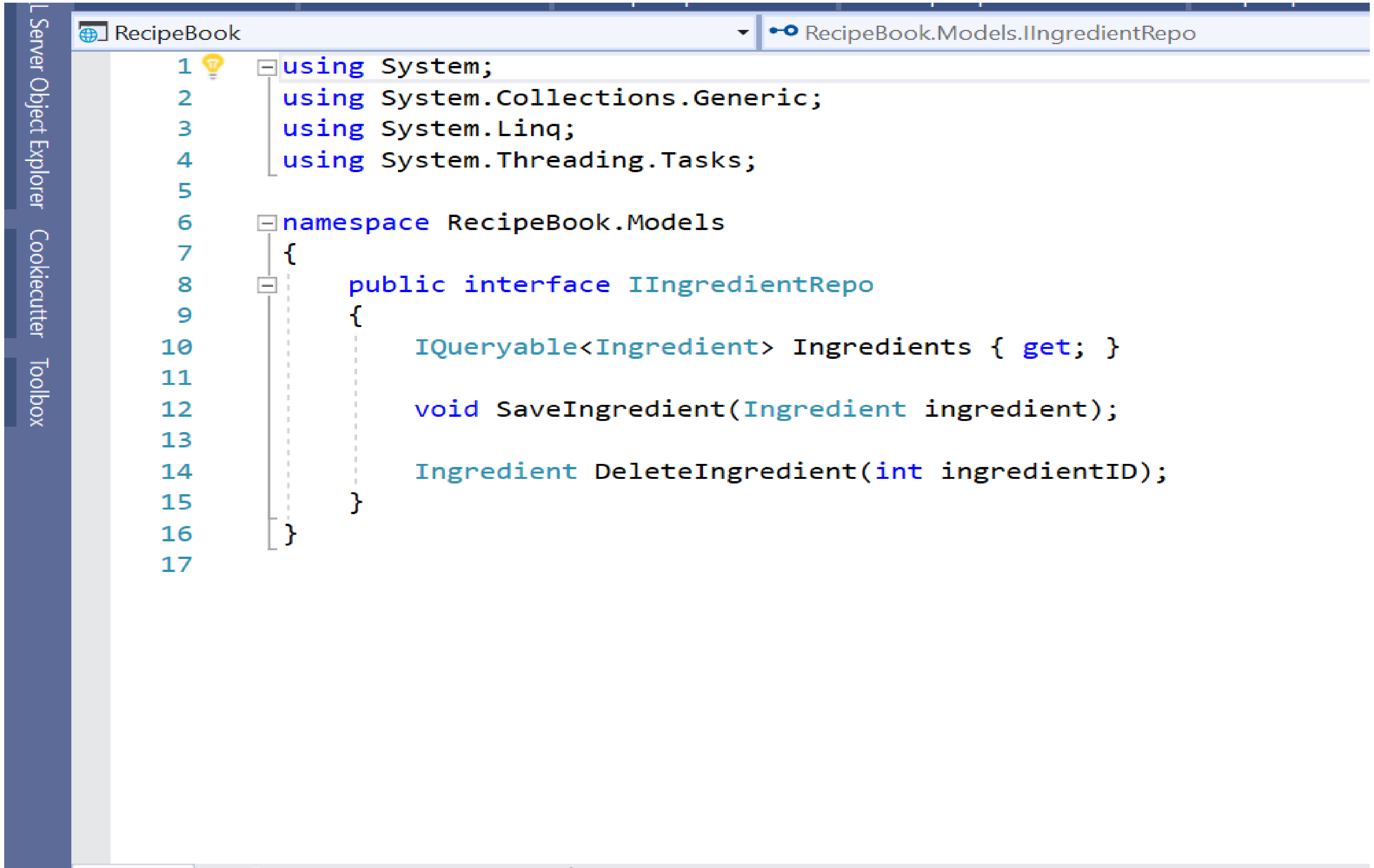
Ingredient:

Included RecipeID to associate with Recipe



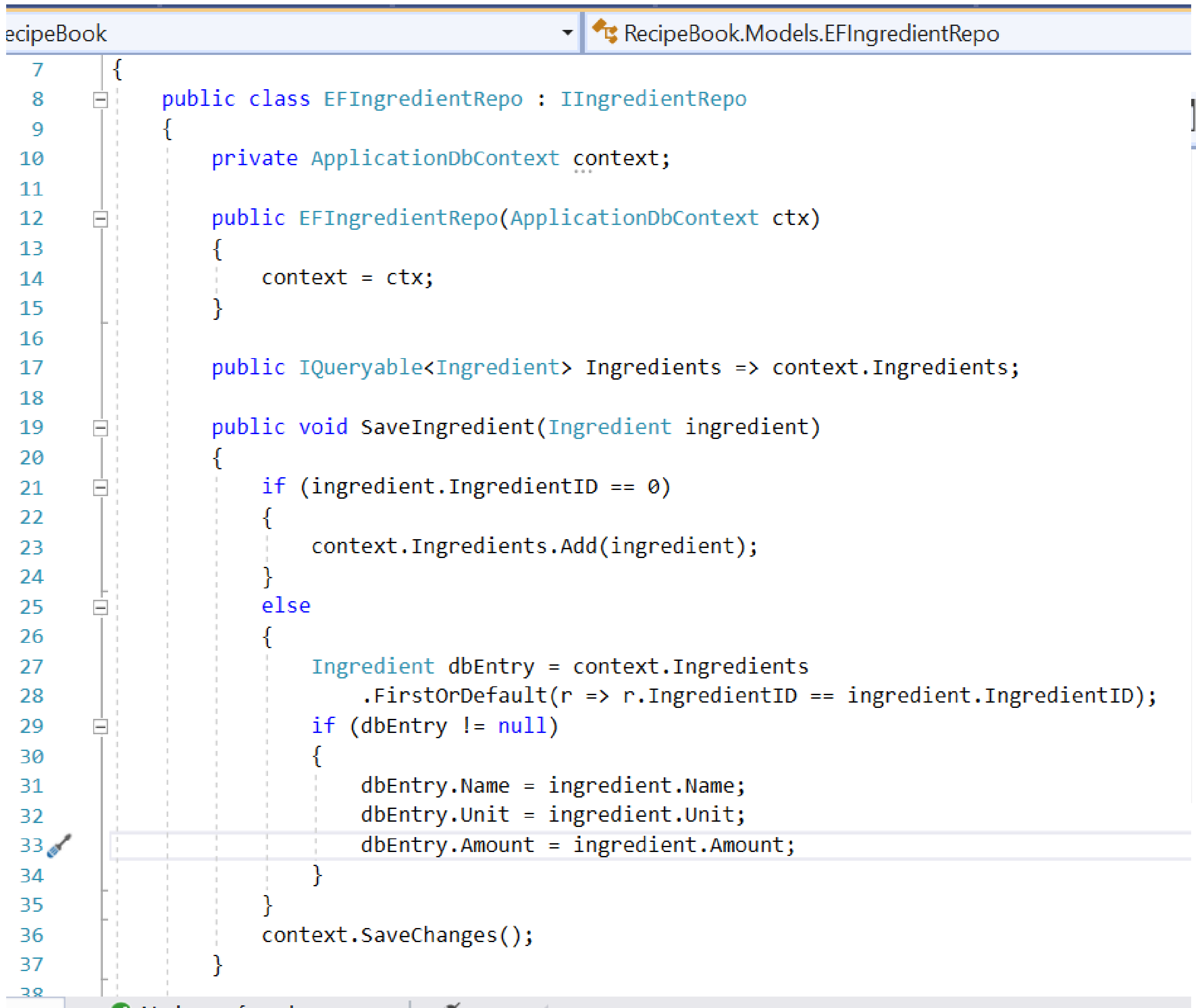
```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Threading.Tasks;

namespace RecipeBook.Models
{
    public class Ingredient
    {
        public int IngredientID { get; set; }
        public int RecipeID { get; set; }
        [Required(ErrorMessage ="Please enter valid name")]
        public string Name { get; set; }
        [Required(ErrorMessage = "Please enter valid Unit of Measure")]
        public string Unit { get; set; }
        [Required]
        [Range(0.25, double.MaxValue,
            ErrorMessage = "Please enter valid Amount")]
        public double Amount { get; set; }

    }
}
```

IIngredientRepo:



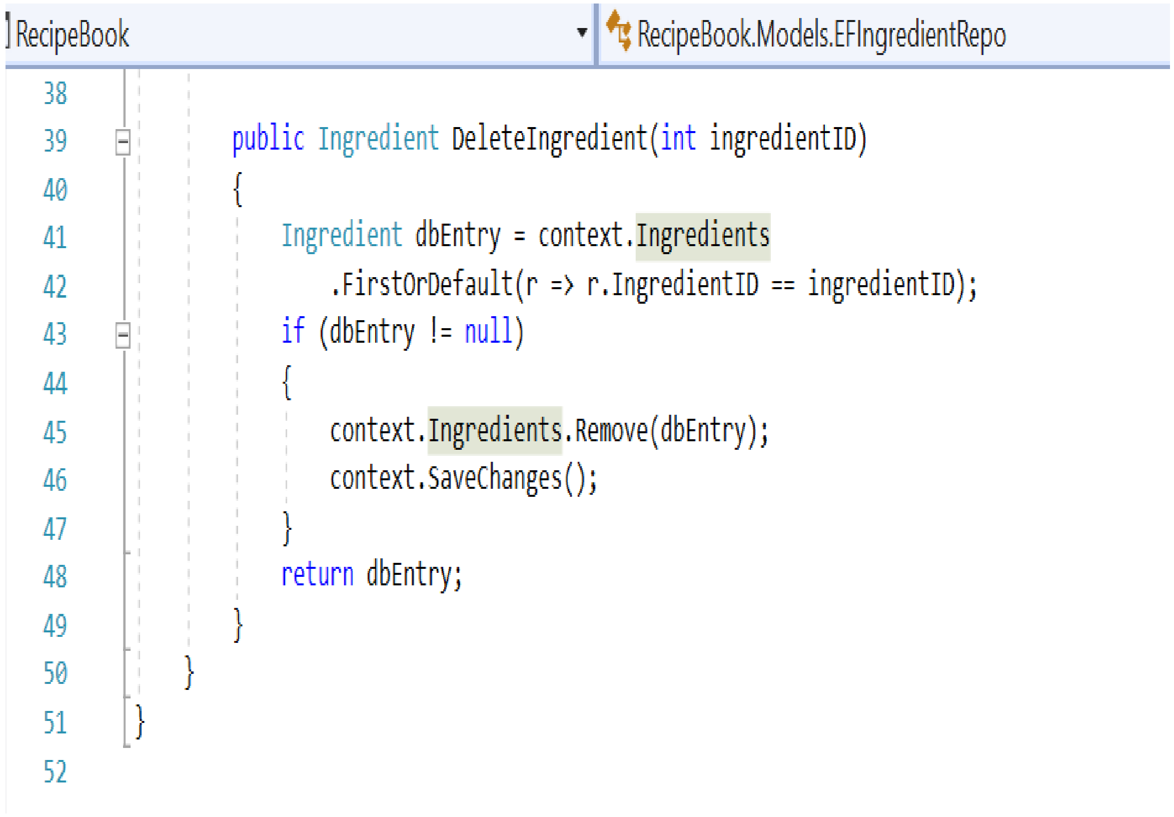
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace RecipeBook.Models
{
    public interface IIngredientRepo
    {
        IQueryable<Ingredient> Ingredients { get; }

        void SaveIngredient(Ingredient ingredient);

        Ingredient DeleteIngredient(int ingredientID);
    }
}
```

EFIngredientRepo:



ecipeBook | RecipeBook.Models.EFIngredientRepo

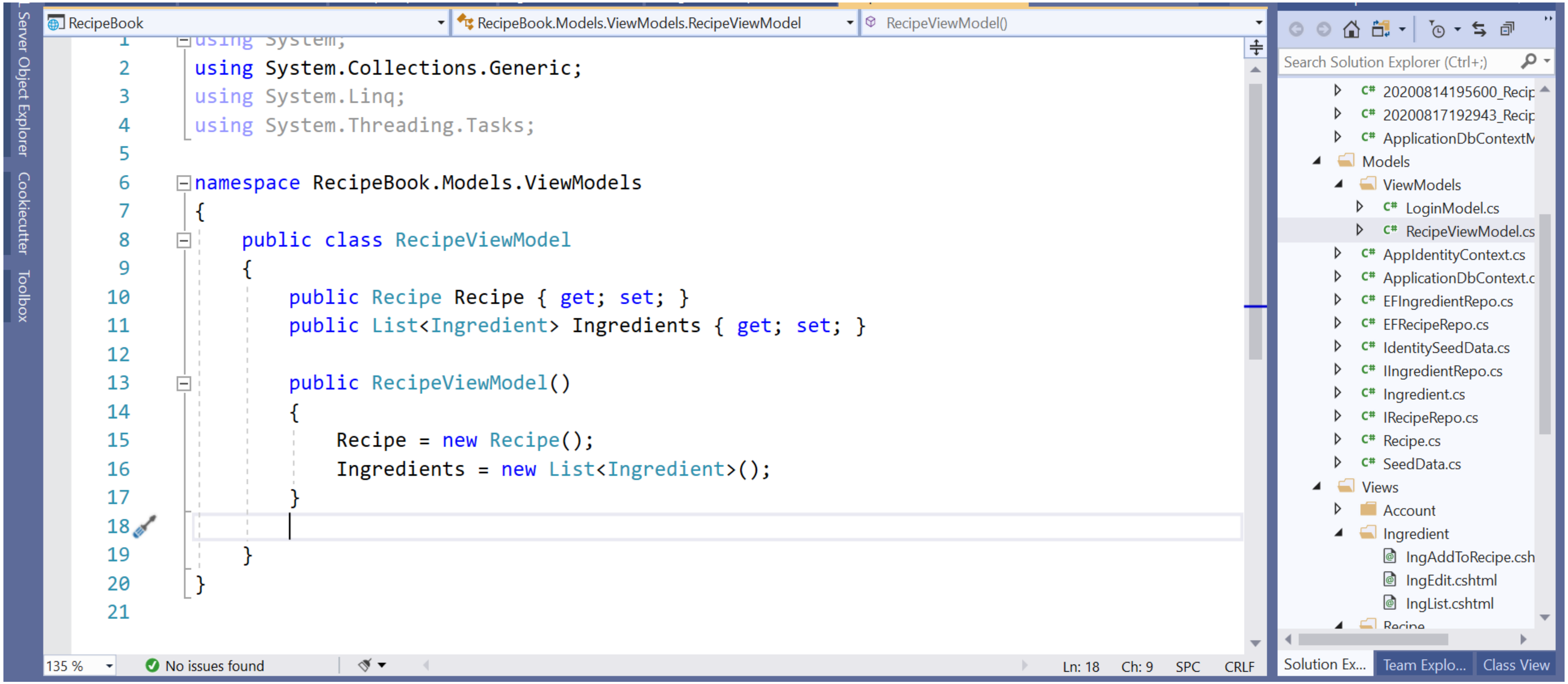
```
{
    public class EFIngredientRepo : IIngredientRepo
    {
        private ApplicationDbContext context;

        public EFIngredientRepo(ApplicationDbContext ctx)
        {
            context = ctx;
        }

        public IQueryable<Ingredient> Ingredients => context.Ingredients;

        public void SaveIngredient(Ingredient ingredient)
        {
            if (ingredient.IngredientID == 0)
            {
                context.Ingredients.Add(ingredient);
            }
            else
            {
                Ingredient dbEntry = context.Ingredients
                    .FirstOrDefault(r => r.IngredientID == ingredient.IngredientID);
                if (dbEntry != null)
                {
                    dbEntry.Name = ingredient.Name;
                    dbEntry.Unit = ingredient.Unit;
                    dbEntry.Amount = ingredient.Amount;
                }
            }
            context.SaveChanges();
        }
```



RecipeBook | RecipeBook.Models.EFIngredientRepo

```

        public Ingredient DeleteIngredient(int ingredientID)
        {
            Ingredient dbEntry = context.Ingredients
                .FirstOrDefault(r => r.IngredientID == ingredientID);
            if (dbEntry != null)
            {
                context.Ingredients.Remove(dbEntry);
                context.SaveChanges();
            }
            return dbEntry;
        }
    }
}
```

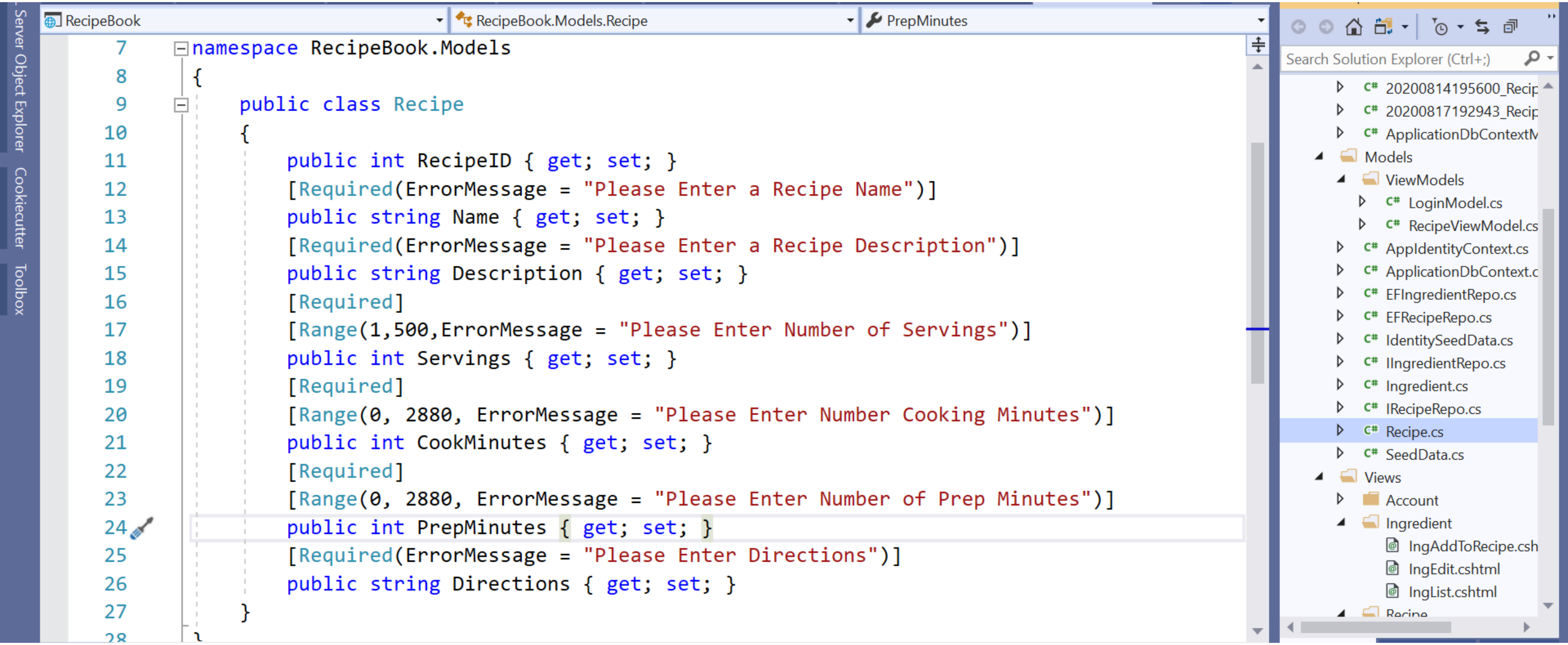
RecipeViewModel:



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace RecipeBook.Models.ViewModels
{
    public class RecipeViewModel
    {
        public Recipe Recipe { get; set; }
        public List<Ingredient> Ingredients { get; set; }

        public RecipeViewModel()
        {
            Recipe = new Recipe();
            Ingredients = new List<Ingredient>();
        }
        
    }
}
```

Recipe Update:

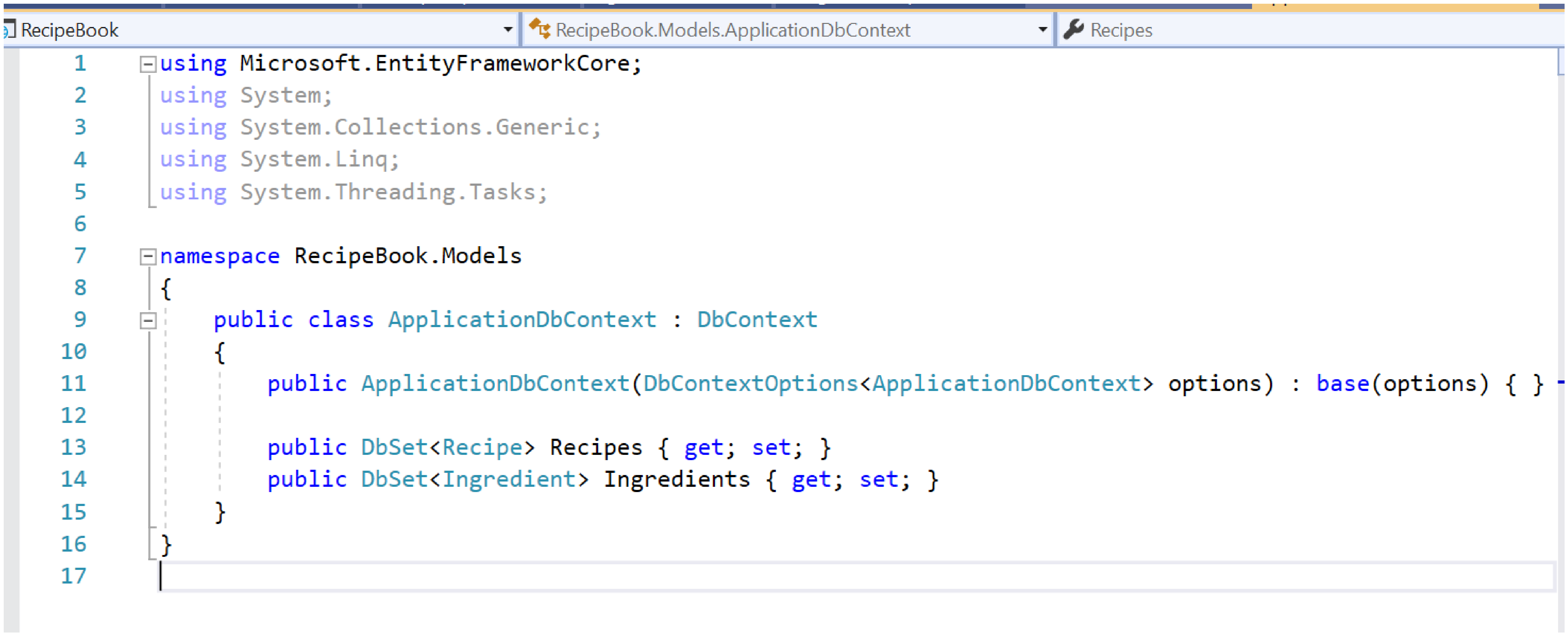
Removed string Ingredient Property



```
namespace RecipeBook.Models
{
    public class Recipe
    {
        public int RecipeID { get; set; }
        [Required(ErrorMessage = "Please Enter a Recipe Name")]
        public string Name { get; set; }
        [Required(ErrorMessage = "Please Enter a Recipe Description")]
        public string Description { get; set; }
        [Required]
        [Range(1,500,ErrorMessage = "Please Enter Number of Servings")]
        public int Servings { get; set; }
        [Required]
        [Range(0, 2880, ErrorMessage = "Please Enter Number Cooking Minutes")]
        public int CookMinutes { get; set; }
        [Required]
        [Range(0, 2880, ErrorMessage = "Please Enter Number of Prep Minutes")]
        public int PrepMinutes { get; set; }
        [Required(ErrorMessage = "Please Enter Directions")]
        public string Directions { get; set; }
    }
}
```

ApplicationDbContext Update:

Added DbSet for Ingredients

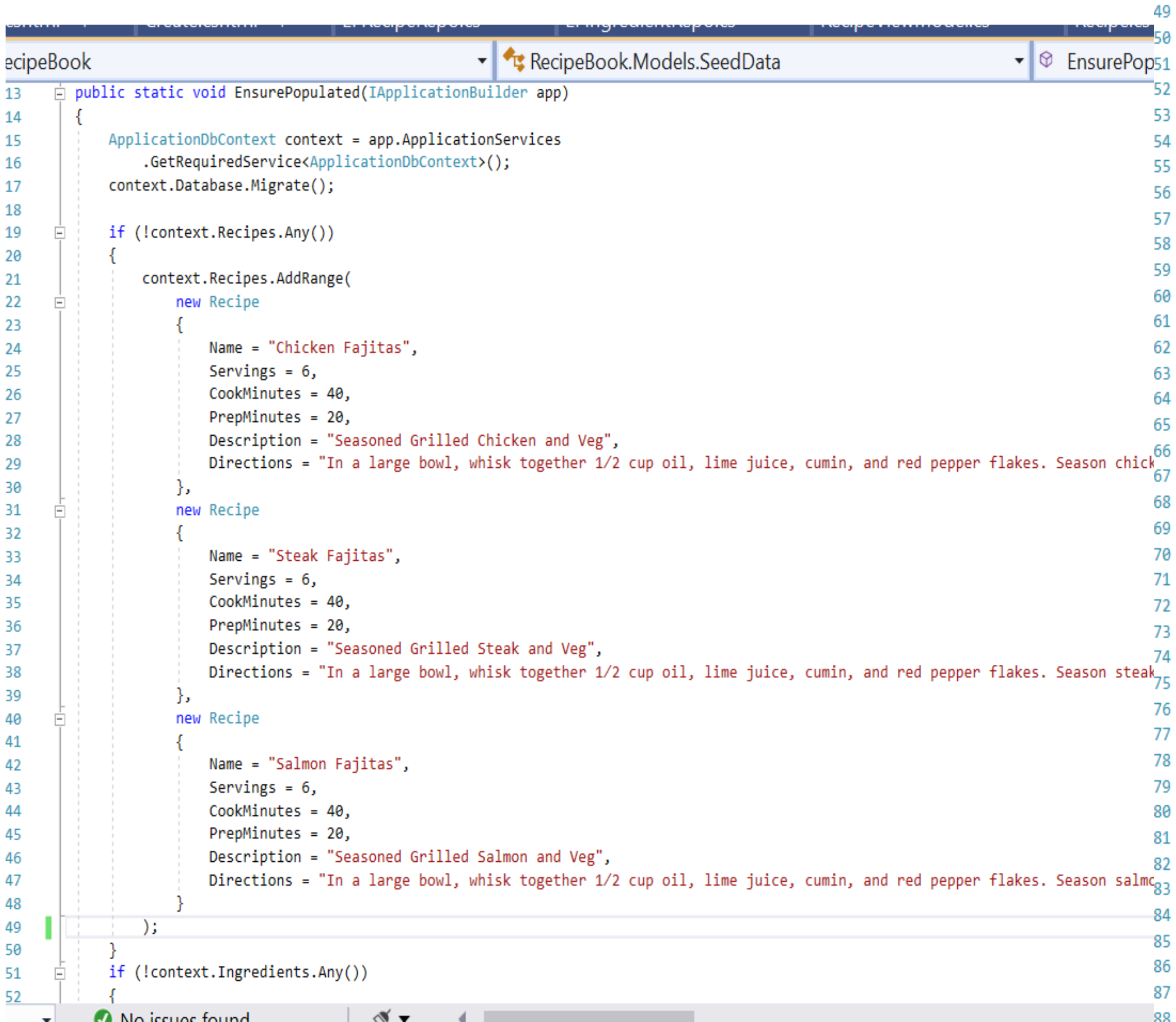


RecipeBook | RecipeBook.Models.ApplicationDbContext | Recipes

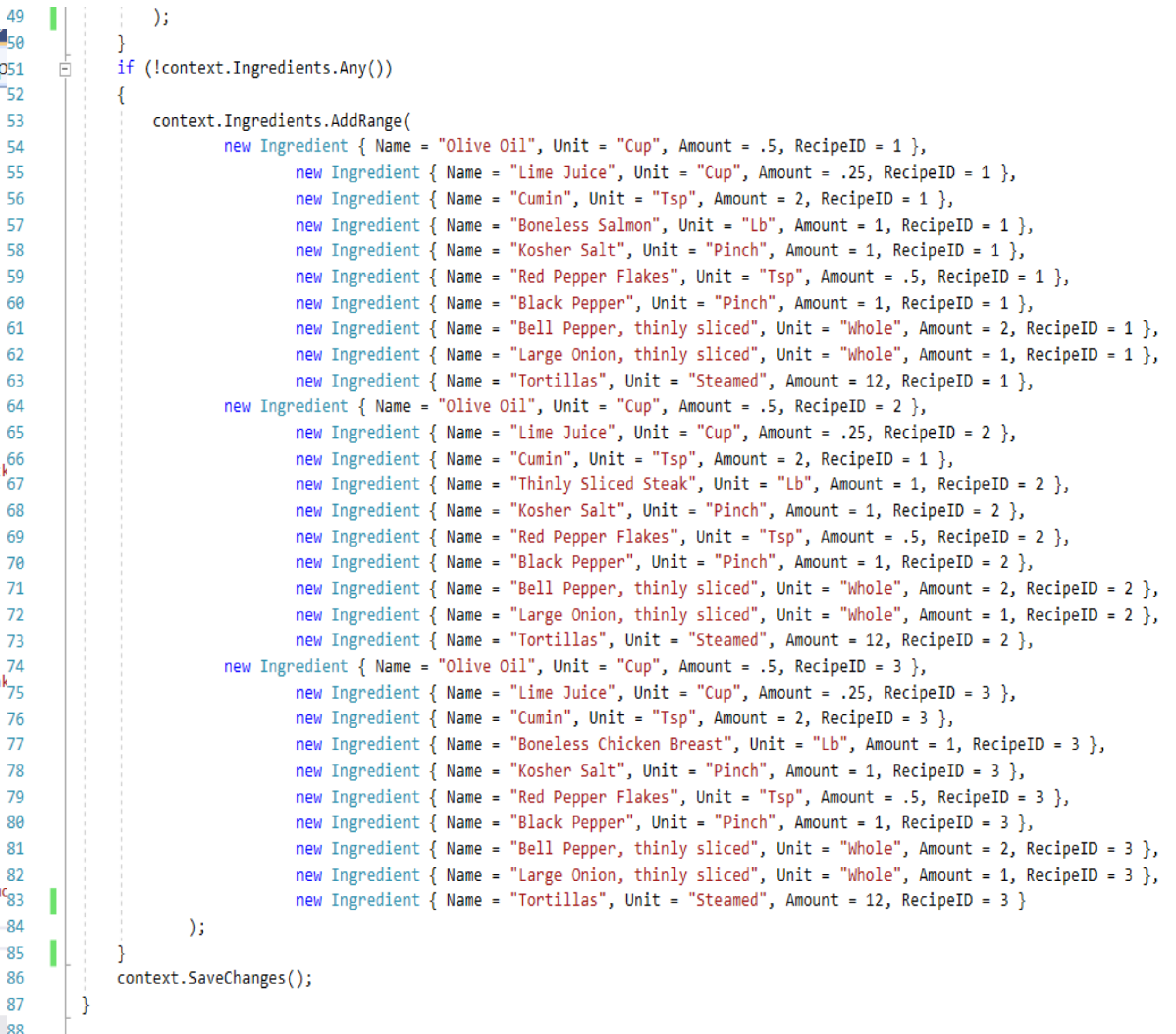
```
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace RecipeBook.Models
{
    public class ApplicationDbContext : DbContext
    {
        public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) : base(options) { }

        public DbSet<Recipe> Recipes { get; set; }
        public DbSet<Ingredient> Ingredients { get; set; }
    }
}

```

SeedData Update:

Added SeedData for Ingredient Context and removed Ingredient SeedData from Recipe Context



```
public static void EnsurePopulated(IApplicationBuilder app)
{
    ApplicationDbContext context = app.ApplicationServices
        .GetRequiredService<ApplicationDbContext>();
    context.Database.Migrate();

    if (!context.Recipes.Any())
    {
        context.Recipes.AddRange(
            new Recipe
            {
                Name = "Chicken Fajitas",
                Servings = 6,
                CookMinutes = 40,
                PrepMinutes = 20,
                Description = "Seasoned Grilled Chicken and Veg",
                Directions = "In a large bowl, whisk together 1/2 cup oil, lime juice, cumin, and red pepper flakes. Season chick
            },
            new Recipe
            {
                Name = "Steak Fajitas",
                Servings = 6,
                CookMinutes = 40,
                PrepMinutes = 20,
                Description = "Seasoned Grilled Steak and Veg",
                Directions = "In a large bowl, whisk together 1/2 cup oil, lime juice, cumin, and red pepper flakes. Season steak
            },
            new Recipe
            {
                Name = "Salmon Fajitas",
                Servings = 6,
                CookMinutes = 40,
                PrepMinutes = 20,
                Description = "Seasoned Grilled Salmon and Veg",
                Directions = "In a large bowl, whisk together 1/2 cup oil, lime juice, cumin, and red pepper flakes. Season salmo
            }
        );
    }
    if (!context.Ingredients.Any())
    {
```



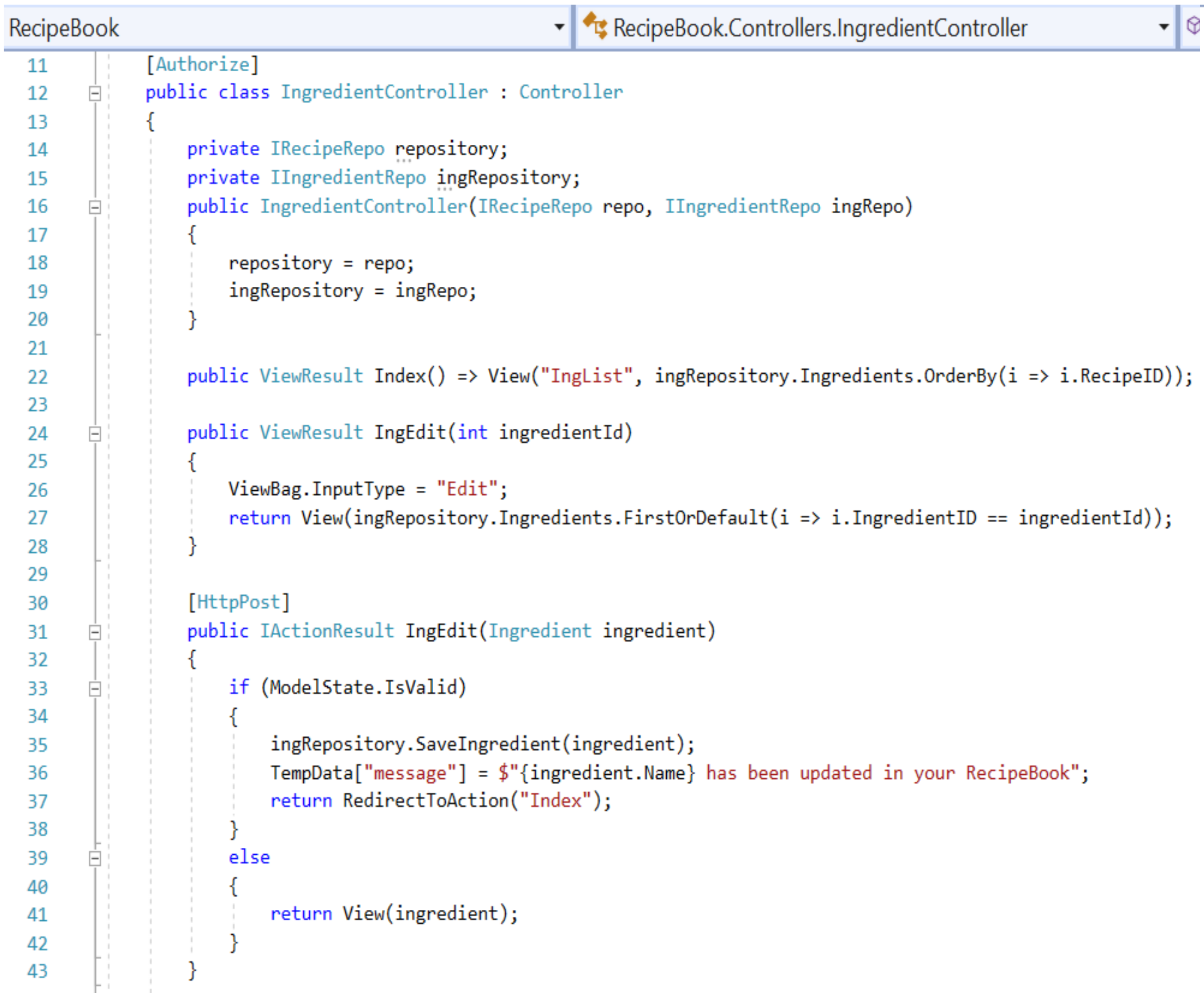
```
        );
    }
    if (!context.Ingredients.Any())
    {
        context.Ingredients.AddRange(
                new Ingredient { Name = "Olive Oil", Unit = "Cup", Amount = .5, RecipeID = 1 },
                        new Ingredient { Name = "Lime Juice", Unit = "Cup", Amount = .25, RecipeID = 1 },
                        new Ingredient { Name = "Cumin", Unit = "Tsp", Amount = 2, RecipeID = 1 },
                        new Ingredient { Name = "Boneless Salmon", Unit = "Lb", Amount = 1, RecipeID = 1 },
                        new Ingredient { Name = "Kosher Salt", Unit = "Pinch", Amount = 1, RecipeID = 1 },
                        new Ingredient { Name = "Red Pepper Flakes", Unit = "Tsp", Amount = .5, RecipeID = 1 },
                        new Ingredient { Name = "Black Pepper", Unit = "Pinch", Amount = 1, RecipeID = 1 },
                        new Ingredient { Name = "Bell Pepper, thinly sliced", Unit = "Whole", Amount = 2, RecipeID = 1 },
                        new Ingredient { Name = "Large Onion, thinly sliced", Unit = "Whole", Amount = 1, RecipeID = 1 },
                        new Ingredient { Name = "Tortillas", Unit = "Steamed", Amount = 12, RecipeID = 1 },
                new Ingredient { Name = "Olive Oil", Unit = "Cup", Amount = .5, RecipeID = 2 },
                        new Ingredient { Name = "Lime Juice", Unit = "Cup", Amount = .25, RecipeID = 2 },
                        new Ingredient { Name = "Cumin", Unit = "Tsp", Amount = 2, RecipeID = 1 },
                        new Ingredient { Name = "Thinly Sliced Steak", Unit = "Lb", Amount = 1, RecipeID = 2 },
                        new Ingredient { Name = "Kosher Salt", Unit = "Pinch", Amount = 1, RecipeID = 2 },
                        new Ingredient { Name = "Red Pepper Flakes", Unit = "Tsp", Amount = .5, RecipeID = 2 },
                        new Ingredient { Name = "Black Pepper", Unit = "Pinch", Amount = 1, RecipeID = 2 },
                        new Ingredient { Name = "Bell Pepper, thinly sliced", Unit = "Whole", Amount = 2, RecipeID = 2 },
                        new Ingredient { Name = "Large Onion, thinly sliced", Unit = "Whole", Amount = 1, RecipeID = 2 },
                        new Ingredient { Name = "Tortillas", Unit = "Steamed", Amount = 12, RecipeID = 2 },
                new Ingredient { Name = "Olive Oil", Unit = "Cup", Amount = .5, RecipeID = 3 },
                        new Ingredient { Name = "Lime Juice", Unit = "Cup", Amount = .25, RecipeID = 3 },
                        new Ingredient { Name = "Cumin", Unit = "Tsp", Amount = 2, RecipeID = 3 },
                        new Ingredient { Name = "Boneless Chicken Breast", Unit = "Lb", Amount = 1, RecipeID = 3 },
                        new Ingredient { Name = "Kosher Salt", Unit = "Pinch", Amount = 1, RecipeID = 3 },
                        new Ingredient { Name = "Red Pepper Flakes", Unit = "Tsp", Amount = .5, RecipeID = 3 },
                        new Ingredient { Name = "Black Pepper", Unit = "Pinch", Amount = 1, RecipeID = 3 },
                        new Ingredient { Name = "Bell Pepper, thinly sliced", Unit = "Whole", Amount = 2, RecipeID = 3 },
                        new Ingredient { Name = "Large Onion, thinly sliced", Unit = "Whole", Amount = 1, RecipeID = 3 },
                        new Ingredient { Name = "Tortillas", Unit = "Steamed", Amount = 12, RecipeID = 3 }
            );
    }
    context.SaveChanges();
}
```

CONTROLLERS:

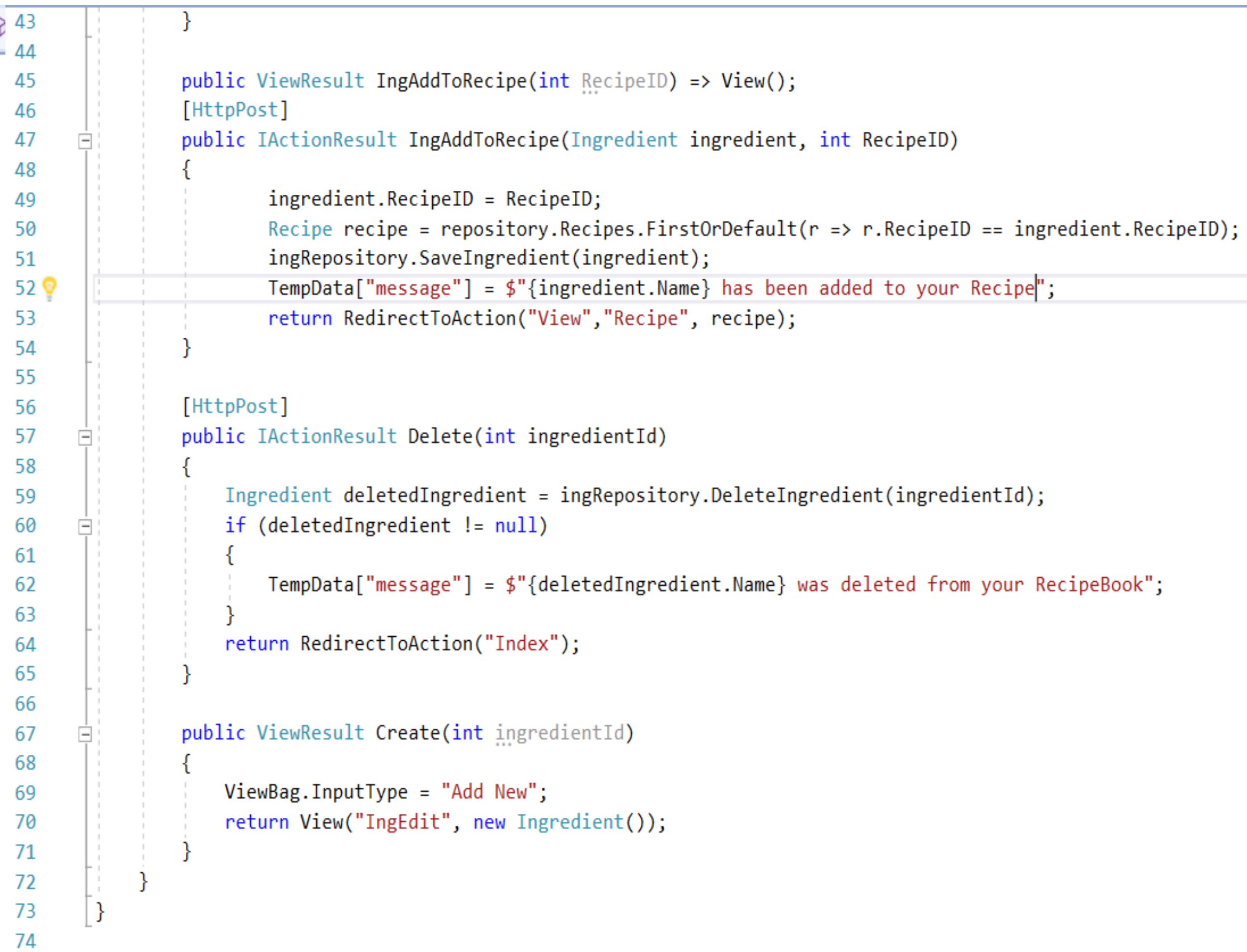
1. **IngredientController**
2. **RecipeController Update**

IngredientController:

Created CRUD controller for viewing Ingredient list, Deleting single Ingredient, Editing Ingredient and Adding Single Ingredient to a Recipe

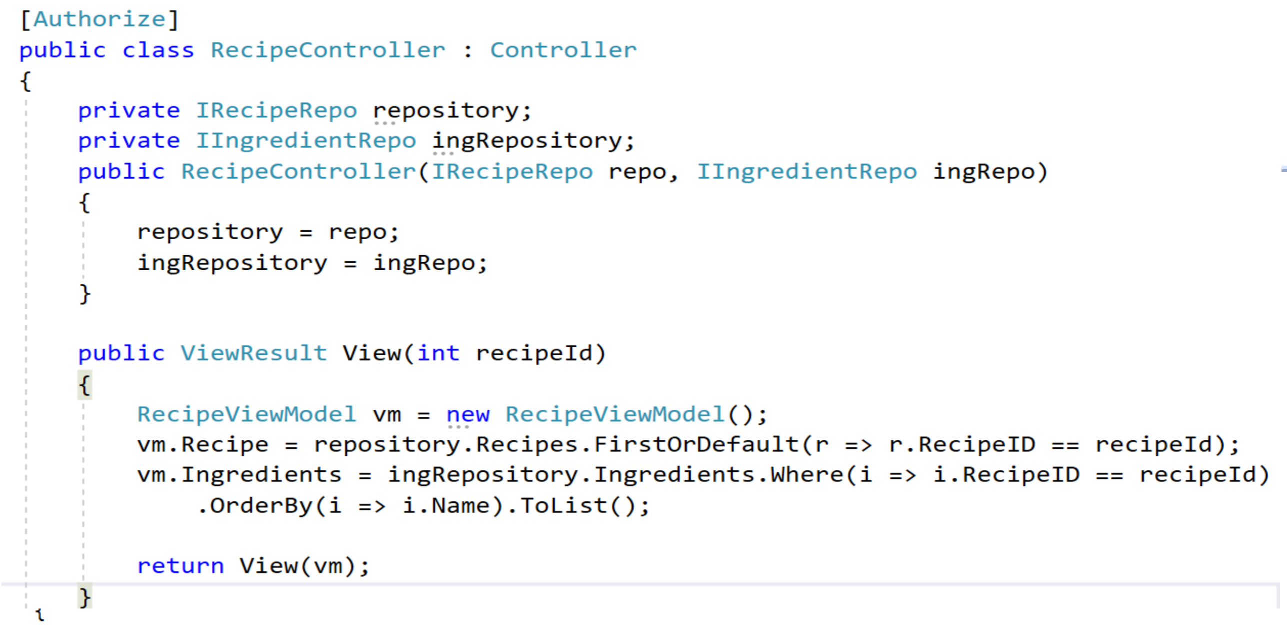


```
    [Authorize]
    public class IngredientController : Controller
    {
        private IRecipeRepo repository;
        private IIngredientRepo ingRepository;
        public IngredientController(IRecipeRepo repo, IIngredientRepo ingRepo)
        {
            repository = repo;
            ingRepository = ingRepo;
        }

        public ViewResult Index() => View("IngList", ingRepository.Ingredients.OrderBy(i => i.RecipeID));

        public ViewResult IngEdit(int ingredientId)
        {
            ViewBag.InputType = "Edit";
            return View(ingRepository.Ingredients.FirstOrDefault(i => i.IngredientID == ingredientId));
        }

        [HttpPost]
        public IActionResult IngEdit(Ingredient ingredient)
        {
            if (ModelState.IsValid)
            {
                ingRepository.SaveIngredient(ingredient);
                TempData["message"] = $"{ingredient.Name} has been updated in your RecipeBook";
                return RedirectToAction("Index");
            }
            else
            {
                return View(ingredient);
            }
        }
```

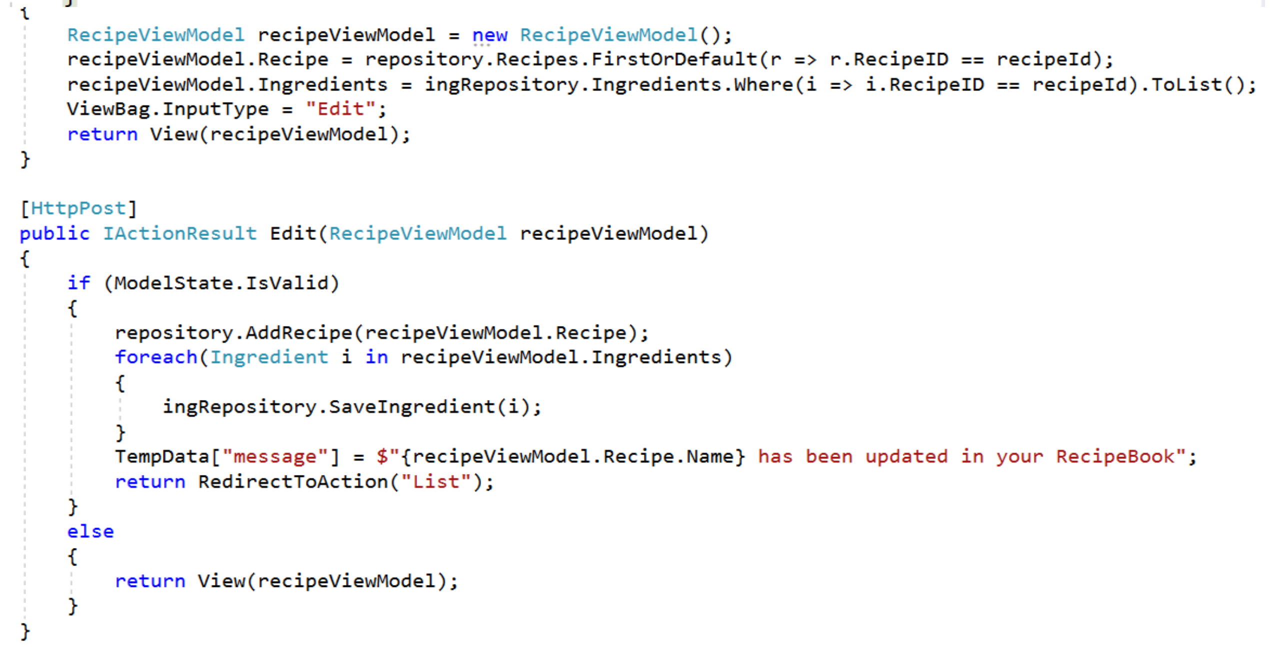


```
        }

        public ViewResult IngAddToRecipe(int RecipeID) => View();
        [HttpPost]
        public IActionResult IngAddToRecipe(Ingredient ingredient, int RecipeID)
        {
                ingredient.RecipeID = RecipeID;
                Recipe recipe = repository.Recipes.FirstOrDefault(r => r.RecipeID == ingredient.RecipeID);
                ingRepository.SaveIngredient(ingredient);
                TempData["message"] = $"{ingredient.Name} has been added to your Recipe";
                return RedirectToAction("View","Recipe", recipe);
        }

        [HttpPost]
        public IActionResult Delete(int ingredientId)
        {
            Ingredient deletedIngredient = ingRepository.DeleteIngredient(ingredientId);
            if (deletedIngredient != null)
            {
                TempData["message"] = $"{deletedIngredient.Name} was deleted from your RecipeBook";
            }
            return RedirectToAction("Index");
        }

        public ViewResult Create(int ingredientId)
        {
            ViewBag.InputType = "Add New";
            return View("IngEdit", new Ingredient());
        }
    }
}
```

RecipeController Update:

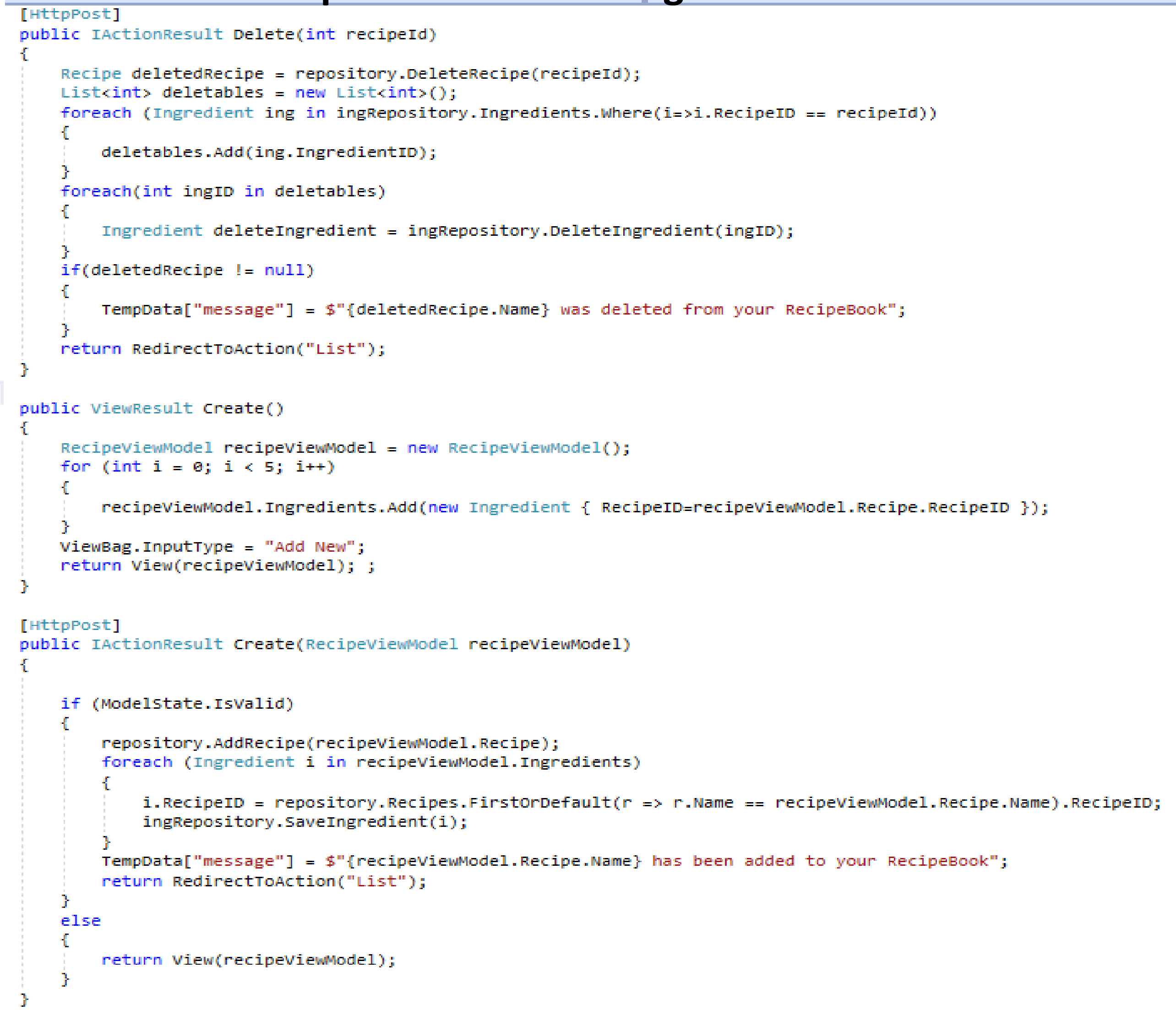


```
[Authorize]
public class RecipeController : Controller
{
    private IRecipeRepo repository;
    private IIngredientRepo ingRepository;
    public RecipeController(IRecipeRepo repo, IIngredientRepo ingRepo)
    {
        repository = repo;
        ingRepository = ingRepo;
    }

    public ViewResult View(int recipeId)
    {
        RecipeViewModel vm = new RecipeViewModel();
        vm.Recipe = repository.Recipes.FirstOrDefault(r => r.RecipeID == recipeId);
        vm.Ingredients = ingRepository.Ingredients.Where(i => i.RecipeID == recipeId)
            .OrderBy(i => i.Name).ToList();

        return View(vm);
    }
```



```
{
    RecipeViewModel recipeViewModel = new RecipeViewModel();
    recipeViewModel.Recipe = repository.Recipes.FirstOrDefault(r => r.RecipeID == recipeId);
    recipeViewModel.Ingredients = ingRepository.Ingredients.Where(i => i.RecipeID == recipeId).ToList();
    ViewBag.InputType = "Edit";
    return View(recipeViewModel);
}

[HttpPost]
public IActionResult Edit(RecipeViewModel recipeViewModel)
{
    if (ModelState.IsValid)
    {
        repository.AddRecipe(recipeViewModel.Recipe);
        foreach(Ingredient i in recipeViewModel.Ingredients)
        {
            ingRepository.SaveIngredient(i);
        }
        TempData["message"] = $"{recipeViewModel.Recipe.Name} has been updated in your RecipeBook";
        return RedirectToAction("List");
    }
    else
    {
        return View(recipeViewModel);
    }
}
```

Added IIngredientRepo access. Implemented RecipeViewModel in all Recipe CRUDs except list. Deleting Recipe will delete its Ingredients too.

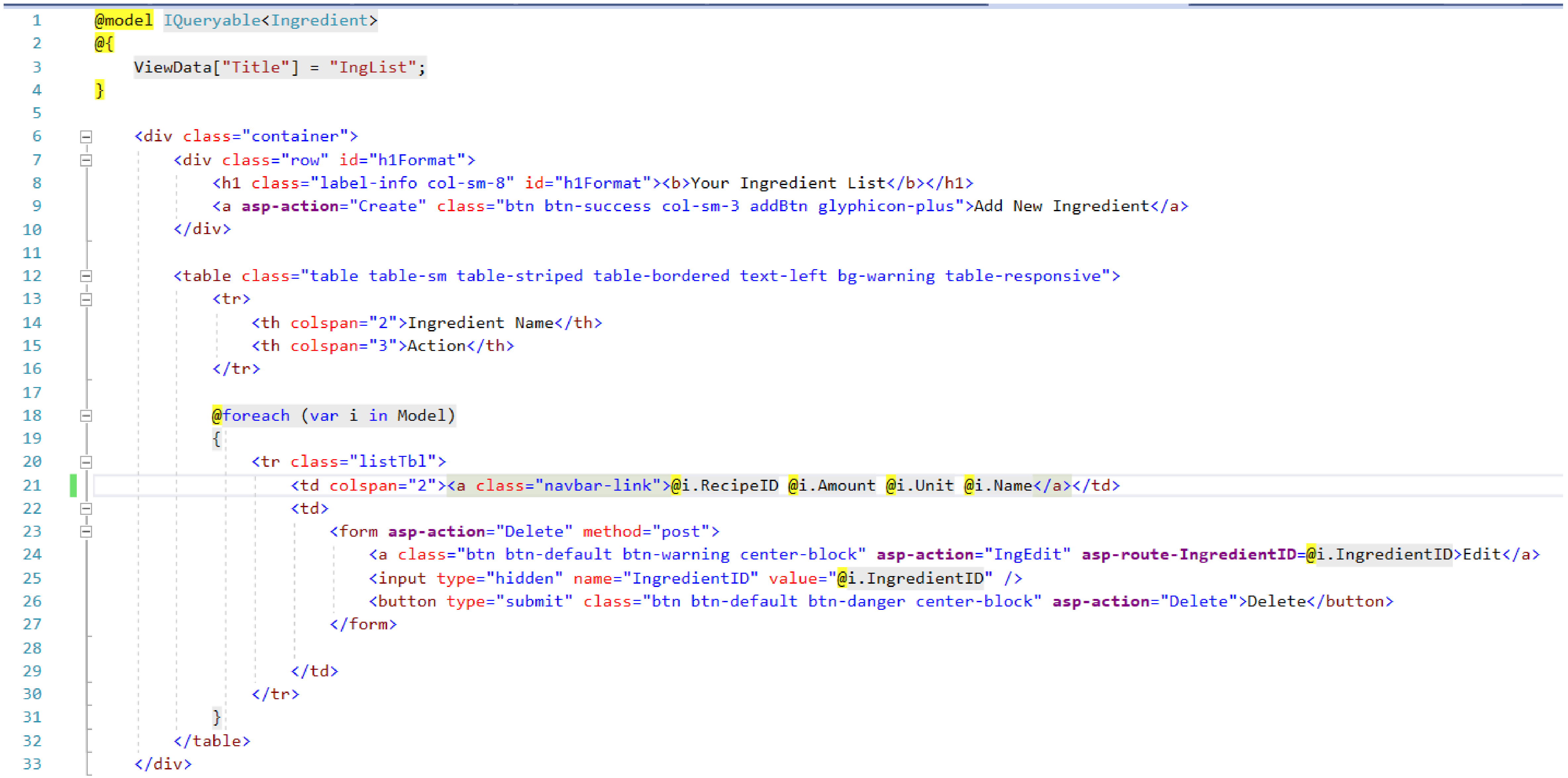


```
[HttpPost]
public IActionResult Delete(int recipeId)
{
    Recipe deletedRecipe = repository.DeleteRecipe(recipeId);
    List<int> deletables = new List<int>();
    foreach (Ingredient ing in ingRepository.Ingredients.Where(i=>i.RecipeID == recipeId))
    {
        deletables.Add(ing.IngredientID);
    }
    foreach(int ingID in deletables)
    {
        Ingredient deleteIngredient = ingRepository.DeleteIngredient(ingID);
    }
    if(deletedRecipe != null)
    {
        TempData["message"] = $"{deletedRecipe.Name} was deleted from your RecipeBook";
    }
    return RedirectToAction("List");
}

public ViewResult Create()
{
    RecipeViewModel recipeViewModel = new RecipeViewModel();
    for (int i = 0; i < 5; i++)
    {
        recipeViewModel.Ingredients.Add(new Ingredient { RecipeID=recipeViewModel.Recipe.RecipeID });
    }
    ViewBag.InputType = "Add New";
    return View(recipeViewModel); ;
}

[HttpPost]
public IActionResult Create(RecipeViewModel recipeViewModel)
{

    if (ModelState.IsValid)
    {
        repository.AddRecipe(recipeViewModel.Recipe);
        foreach (Ingredient i in recipeViewModel.Ingredients)
        {
            i.RecipeID = repository.Recipes.FirstOrDefault(r => r.Name == recipeViewModel.Recipe.Name).RecipeID;
            ingRepository.SaveIngredient(i);
        }
        TempData["message"] = $"{recipeViewModel.Recipe.Name} has been added to your RecipeBook";
        return RedirectToAction("List");
    }
    else
    {
        return View(recipeViewModel);
    }
}
```

VIEWS:

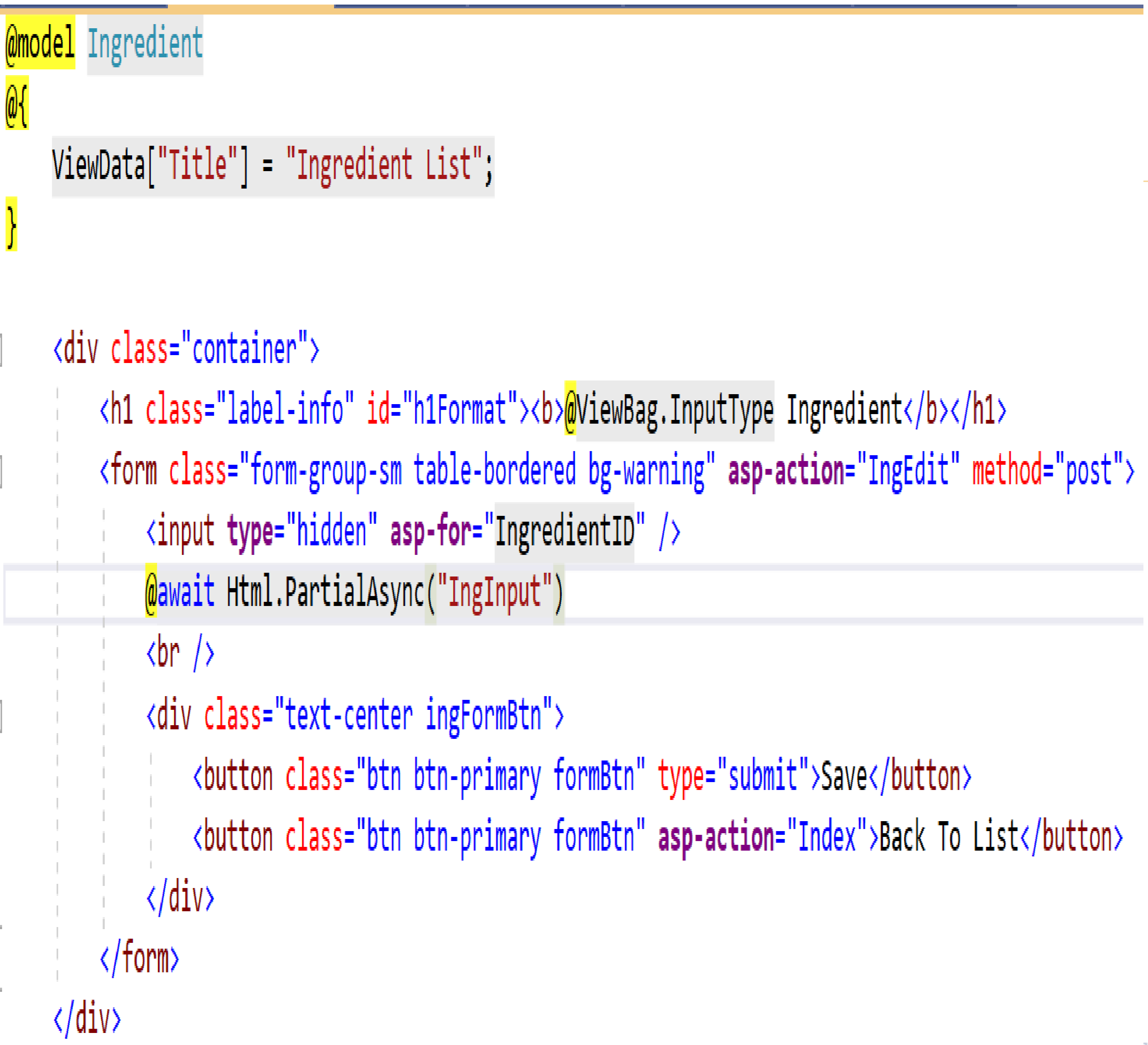
1. **Ingredient/**
    1. **ListIngredient**
    2. **EditIngredient**
    3. **AddIngredientToRecipe**
2. **Recipe/**
    1. **Updated CreateRecipe**
    2. **Updated EditRecipe**
    3. **Updated ViewRecipe**
3. **Shared/**
    1. **FullRecipeInput (Partial)**

Ingredient/List:



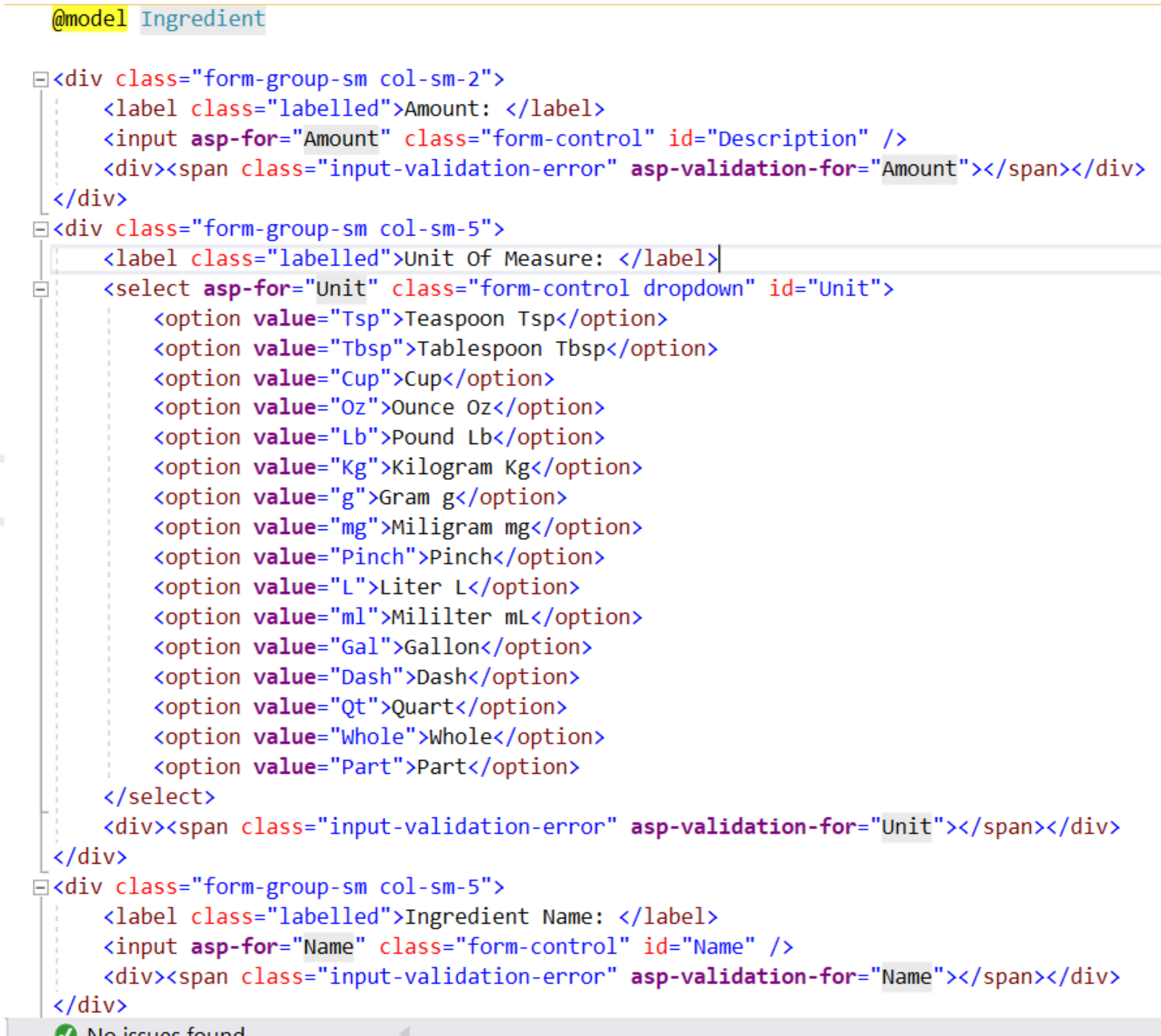
```
@model IQueryable<Ingredient>
@{
    ViewData["Title"] = "IngList";
}

    <div class="container">
        <div class="row" id="h1Format">
            <h1 class="label-info col-sm-8" id="h1Format"><b>Your Ingredient List</b></h1>
            <a asp-action="Create" class="btn btn-success col-sm-3 addBtn glyphicon-plus">Add New Ingredient</a>
        </div>

        <table class="table table-sm table-striped table-bordered text-left bg-warning table-responsive">
            <tr>
                <th colspan="2">Ingredient Name</th>
                <th colspan="3">Action</th>
            </tr>

            @foreach (var i in Model)
            {
                <tr class="listTbl">
                    <td colspan="2"><a class="navbar-link">@i.RecipeID @i.Amount @i.Unit @i.Name</a></td>
                    <td>
                        <form asp-action="Delete" method="post">
                            <a class="btn btn-default btn-warning center-block" asp-action="IngEdit" asp-route-IngredientID=@i.IngredientID>Edit</a>
                            <input type="hidden" name="IngredientID" value="@i.IngredientID" />
                            <button type="submit" class="btn btn-default btn-danger center-block" asp-action="Delete">Delete</button>
                        </form>

                    </td>
                </tr>
            }
        </table>
    </div>
```

Ingredient/Edit:



```
@model Ingredient
@{
    ViewData["Title"] = "Ingredient List";
}

<div class="container">
    <h1 class="label-info" id="h1Format"><b>@ViewBag.InputType Ingredient</b></h1>
    <form class="form-group-sm table-bordered bg-warning" asp-action="IngEdit" method="post">
        <input type="hidden" asp-for="IngredientID" />
        @await Html.PartialAsync("IngInput")
        <br />
        <div class="text-center ingFormBtn">
            <button class="btn btn-primary formBtn" type="submit">Save</button>
            <button class="btn btn-primary formBtn" asp-action="Index">Back To List</button>
        </div>
    </form>
</div>
```

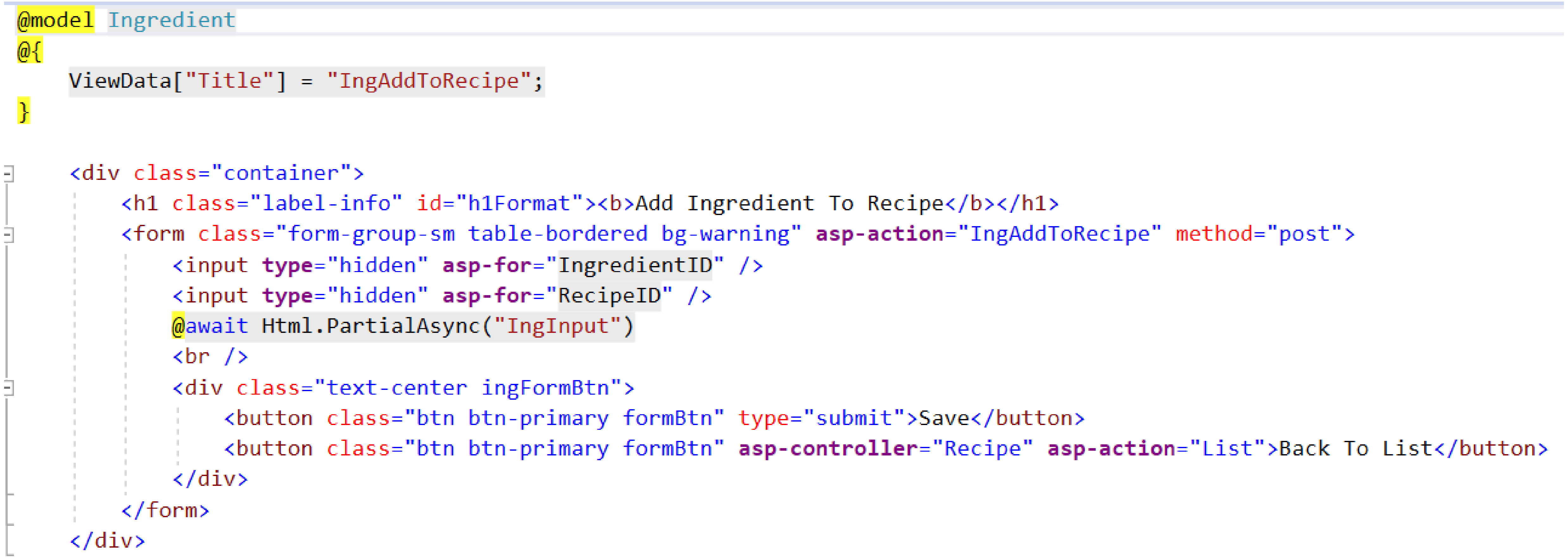
Edit calls partial View for Single Ingredient input



```
@model Ingredient

<div class="form-group-sm col-sm-2">
    <label class="labelled">Amount: </label>
    <input asp-for="Amount" class="form-control" id="Description" />
    <div><span class="input-validation-error" asp-validation-for="Amount"></span></div>
</div>
<div class="form-group-sm col-sm-5">
    <label class="labelled">Unit Of Measure: </label>
    <select asp-for="Unit" class="form-control dropdown" id="Unit">
        <option value="Tsp">Teaspoon Tsp</option>
        <option value="Tbsp">Tablespoon Tbsp</option>
        <option value="Cup">Cup</option>
        <option value="Oz">Ounce Oz</option>
        <option value="Lb">Pound Lb</option>
        <option value="Kg">Kilogram Kg</option>
        <option value="g">Gram g</option>
        <option value="mg">Miligram mg</option>
        <option value="Pinch">Pinch</option>
        <option value="L">Liter L</option>
        <option value="ml">Mililter mL</option>
        <option value="Gal">Gallon</option>
        <option value="Dash">Dash</option>
        <option value="Qt">Quart</option>
        <option value="Whole">Whole</option>
        <option value="Part">Part</option>
    </select>
    <div><span class="input-validation-error" asp-validation-for="Unit"></span></div>
</div>
<div class="form-group-sm col-sm-5">
    <label class="labelled">Ingredient Name: </label>
    <input asp-for="Name" class="form-control" id="Name" />
    <div><span class="input-validation-error" asp-validation-for="Name"></span></div>
</div>
```

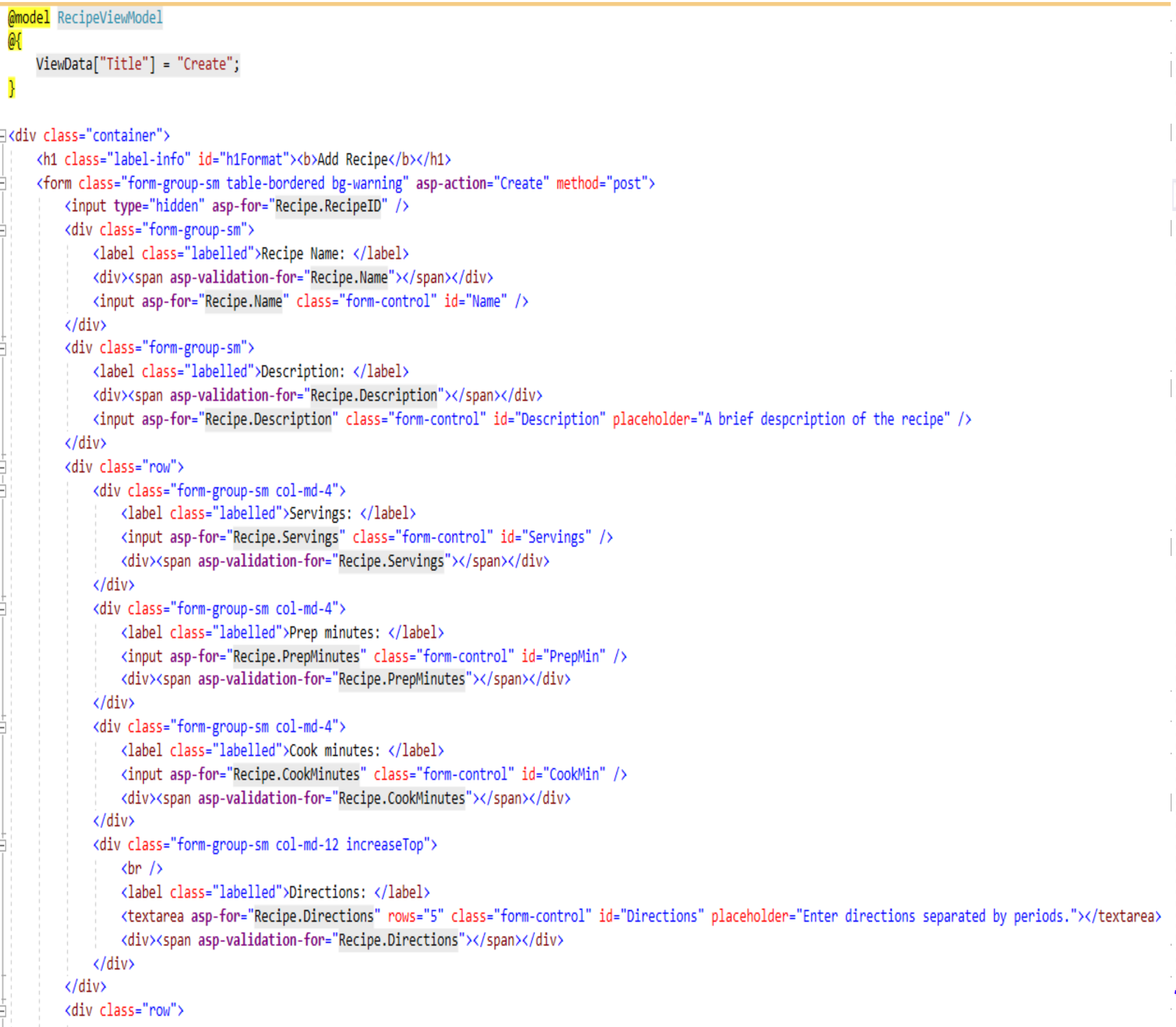
Ingredient/AddIngredientToRecipe:

AddIngredientToRecipe calls partial View for Single Ingredient input

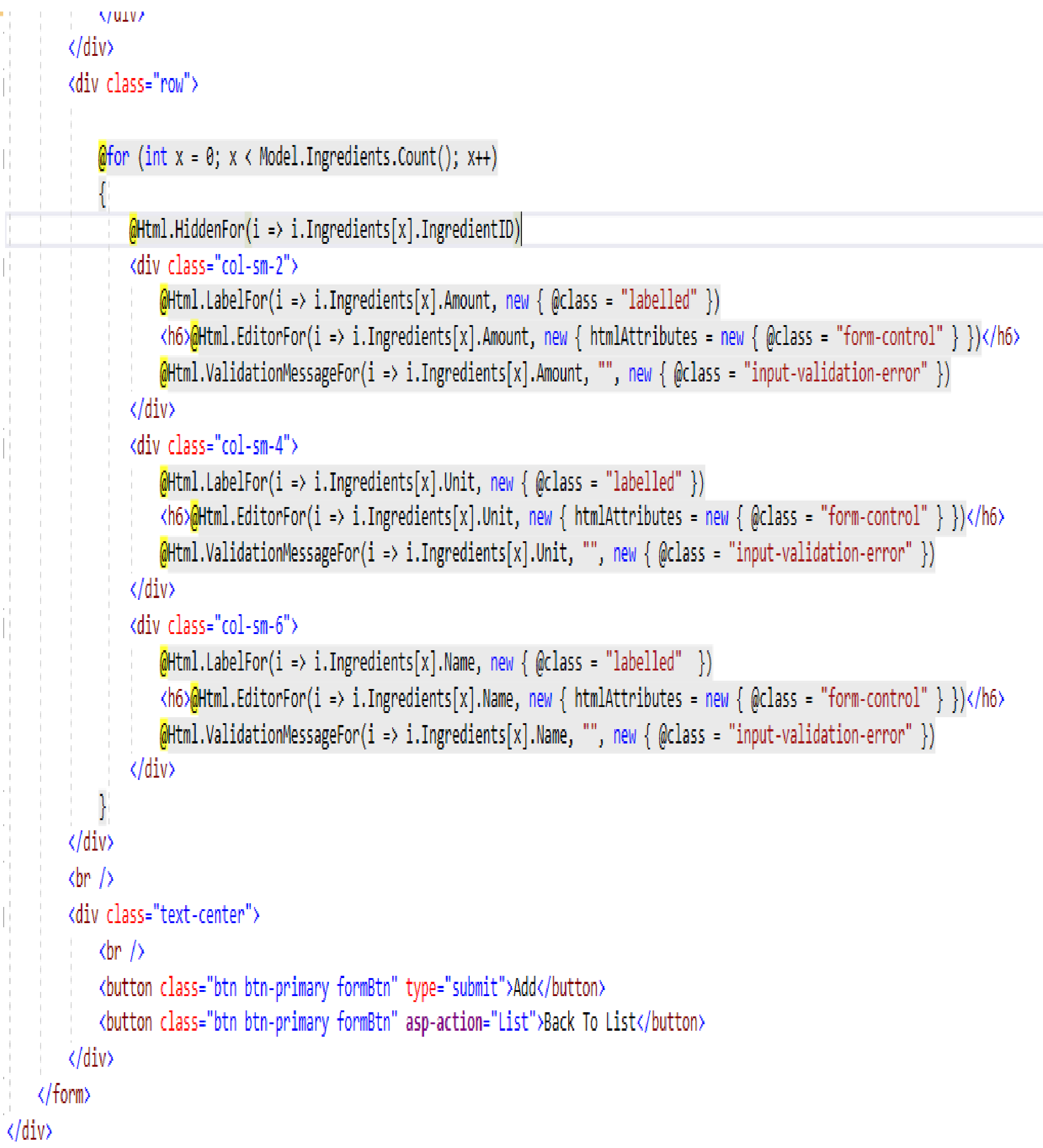


```
@model Ingredient
@{
    ViewData["Title"] = "IngAddToRecipe";
}

    <div class="container">
        <h1 class="label-info" id="h1Format"><b>Add Ingredient To Recipe</b></h1>
        <form class="form-group-sm table-bordered bg-warning" asp-action="IngAddToRecipe" method="post">
            <input type="hidden" asp-for="IngredientID" />
            <input type="hidden" asp-for="RecipeID" />
            @await Html.PartialAsync("IngInput")
            <br />
            <div class="text-center ingFormBtn">
                <button class="btn btn-primary formBtn" type="submit">Save</button>
                <button class="btn btn-primary formBtn" asp-controller="Recipe" asp-action="List">Back To List</button>
            </div>
        </form>
    </div>
```

Recipe/Create Update:



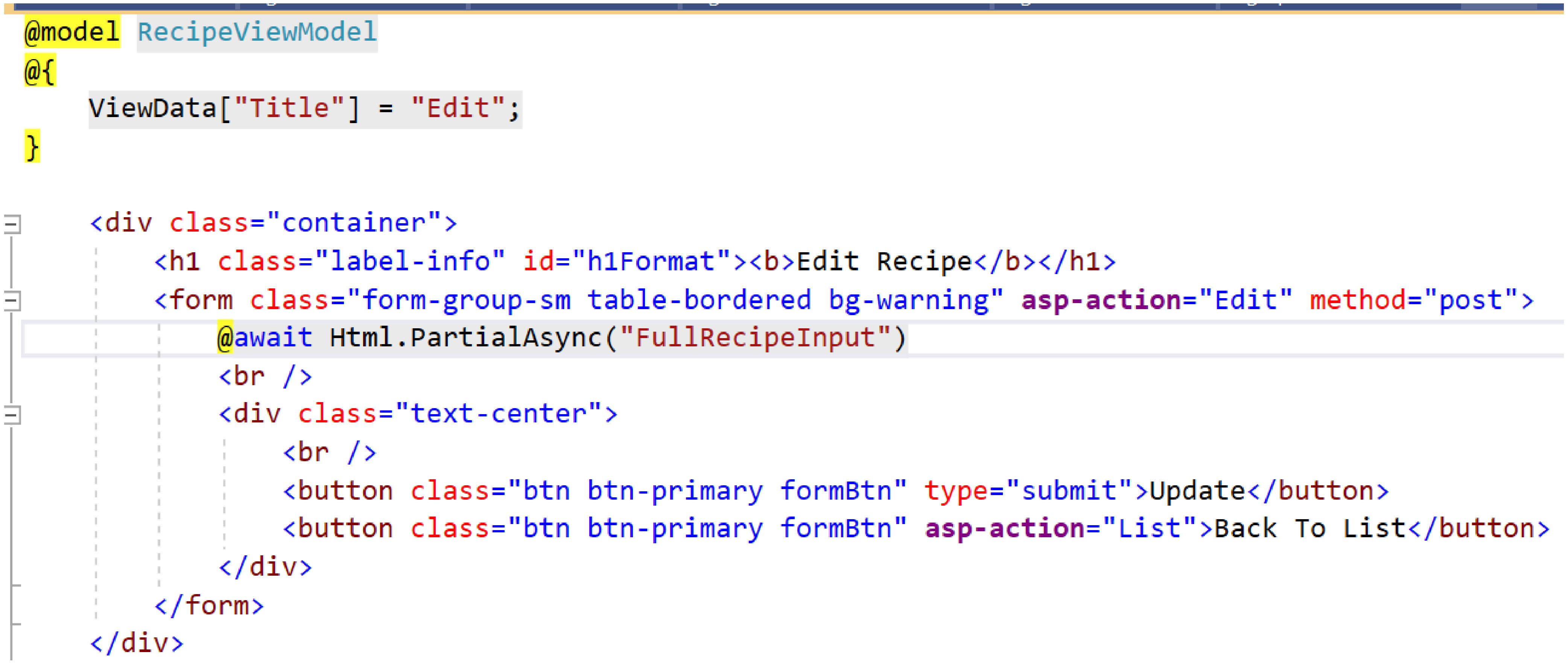
```
@model RecipeViewModel
@{
    ViewData["Title"] = "Create";
}

<div class="container">
    <h1 class="label-info" id="h1Format"><b>Add Recipe</b></h1>
    <form class="form-group-sm table-bordered bg-warning" asp-action="Create" method="post">
        <input type="hidden" asp-for="Recipe.RecipeID" />
        <div class="form-group-sm">
            <label class="labelled">Recipe Name: </label>
            <div><span asp-validation-for="Recipe.Name"></span></div>
            <input asp-for="Recipe.Name" class="form-control" id="Name" />
        </div>
        <div class="form-group-sm">
            <label class="labelled">Description: </label>
            <div><span asp-validation-for="Recipe.Description"></span></div>
            <input asp-for="Recipe.Description" class="form-control" id="Description" placeholder="A brief despcription of the recipe" />
        </div>
        <div class="row">
            <div class="form-group-sm col-md-4">
                <label class="labelled">Servings: </label>
                <input asp-for="Recipe.Servings" class="form-control" id="Servings" />
                <div><span asp-validation-for="Recipe.Servings"></span></div>
            </div>
            <div class="form-group-sm col-md-4">
                <label class="labelled">Prep minutes: </label>
                <input asp-for="Recipe.PrepMinutes" class="form-control" id="PrepMin" />
                <div><span asp-validation-for="Recipe.PrepMinutes"></span></div>
            </div>
            <div class="form-group-sm col-md-4">
                <label class="labelled">Cook minutes: </label>
                <input asp-for="Recipe.CookMinutes" class="form-control" id="CookMin" />
                <div><span asp-validation-for="Recipe.CookMinutes"></span></div>
            </div>
            <div class="form-group-sm col-md-12 increaseTop">
                <br />
                <label class="labelled">Directions: </label>
                <textarea asp-for="Recipe.Directions" rows="5" class="form-control" id="Directions" placeholder="Enter directions separated by periods."></textarea>
                <div><span asp-validation-for="Recipe.Directions"></span></div>
            </div>
        </div>
        <div class="row">
```



```
            </div>
        </div>
        <div class="row">

            @for (int x = 0; x < Model.Ingredients.Count(); x++)
            {
                @Html.HiddenFor(i => i.Ingredients[x].IngredientID)
                <div class="col-sm-2">
                    @Html.LabelFor(i => i.Ingredients[x].Amount, new { @class = "labelled" })
                    <h6>@Html.EditorFor(i => i.Ingredients[x].Amount, new { htmlAttributes = new { @class = "form-control" } })</h6>
                    @Html.ValidationMessageFor(i => i.Ingredients[x].Amount, "", new { @class = "input-validation-error" })
                </div>
                <div class="col-sm-4">
                    @Html.LabelFor(i => i.Ingredients[x].Unit, new { @class = "labelled" })
                    <h6>@Html.EditorFor(i => i.Ingredients[x].Unit, new { htmlAttributes = new { @class = "form-control" } })</h6>
                    @Html.ValidationMessageFor(i => i.Ingredients[x].Unit, "", new { @class = "input-validation-error" })
                </div>
                <div class="col-sm-6">
                    @Html.LabelFor(i => i.Ingredients[x].Name, new { @class = "labelled"  })
                    <h6>@Html.EditorFor(i => i.Ingredients[x].Name, new { htmlAttributes = new { @class = "form-control" } })</h6>
                    @Html.ValidationMessageFor(i => i.Ingredients[x].Name, "", new { @class = "input-validation-error" })
                </div>
            }
        </div>
        <br />
        <div class="text-center">
            <br />
            <button class="btn btn-primary formBtn" type="submit">Add</button>
            <button class="btn btn-primary formBtn" asp-action="List">Back To List</button>
        </div>
    </form>
</div>
```

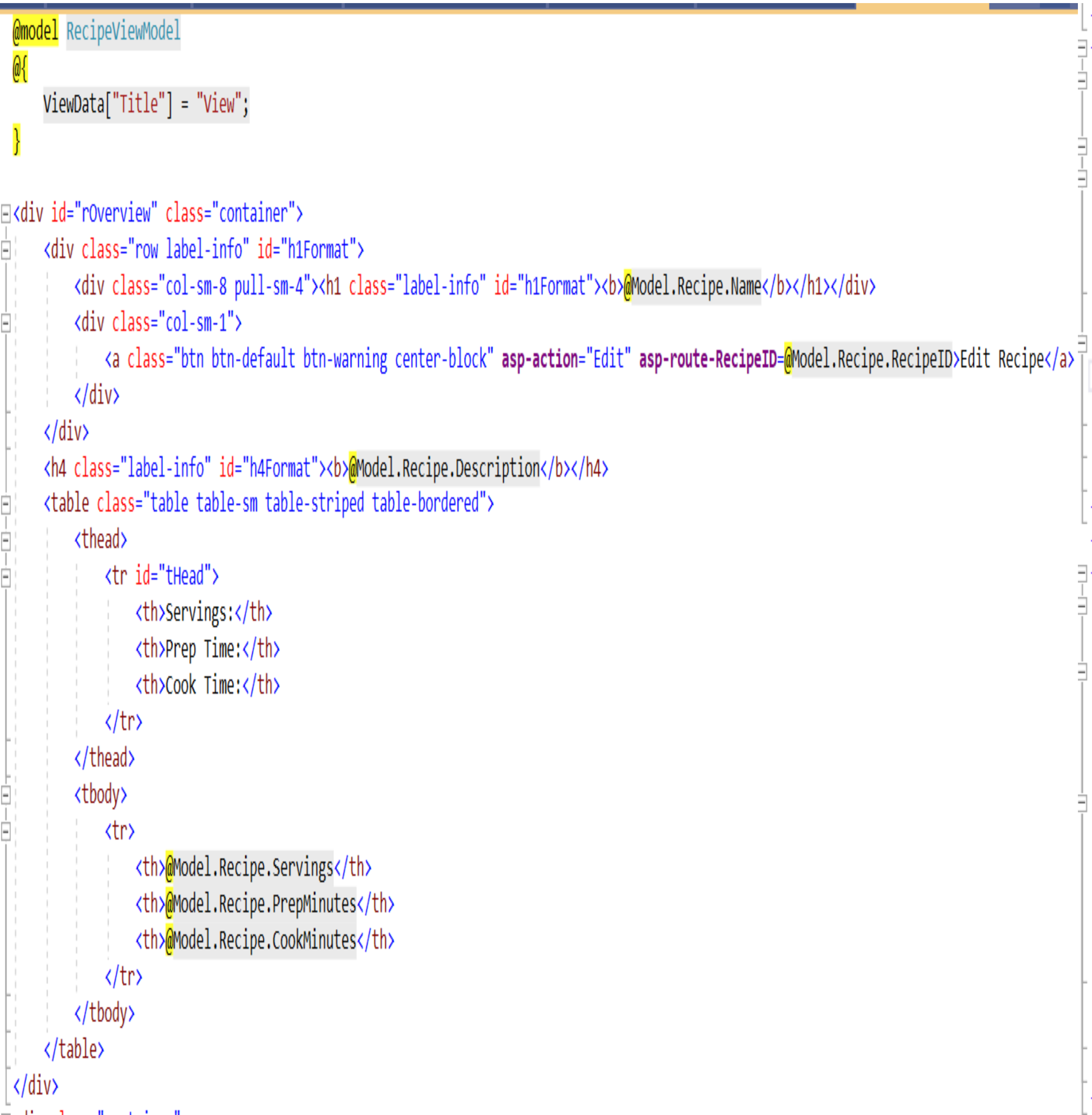
Recipe/Edit Update:

Edit calls FullRecipeInput partial view for recipe input

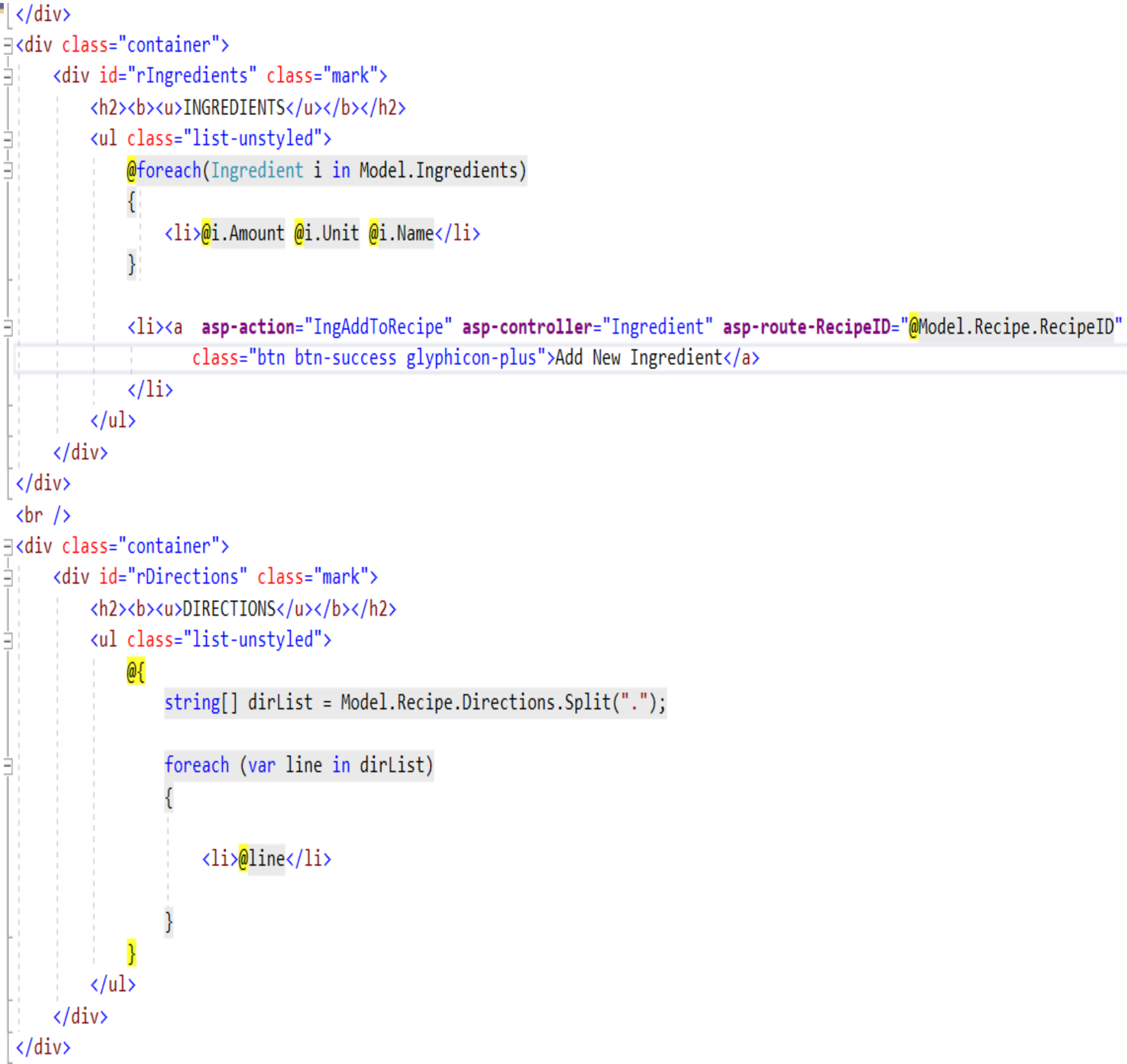


```
@model RecipeViewModel
@{
    ViewData["Title"] = "Edit";
}

    <div class="container">
        <h1 class="label-info" id="h1Format"><b>Edit Recipe</b></h1>
        <form class="form-group-sm table-bordered bg-warning" asp-action="Edit" method="post">
            @await Html.PartialAsync("FullRecipeInput")
            <br />
            <div class="text-center">
                <br />
                <button class="btn btn-primary formBtn" type="submit">Update</button>
                <button class="btn btn-primary formBtn" asp-action="List">Back To List</button>
            </div>
        </form>
    </div>
```

Recipe/View Update:

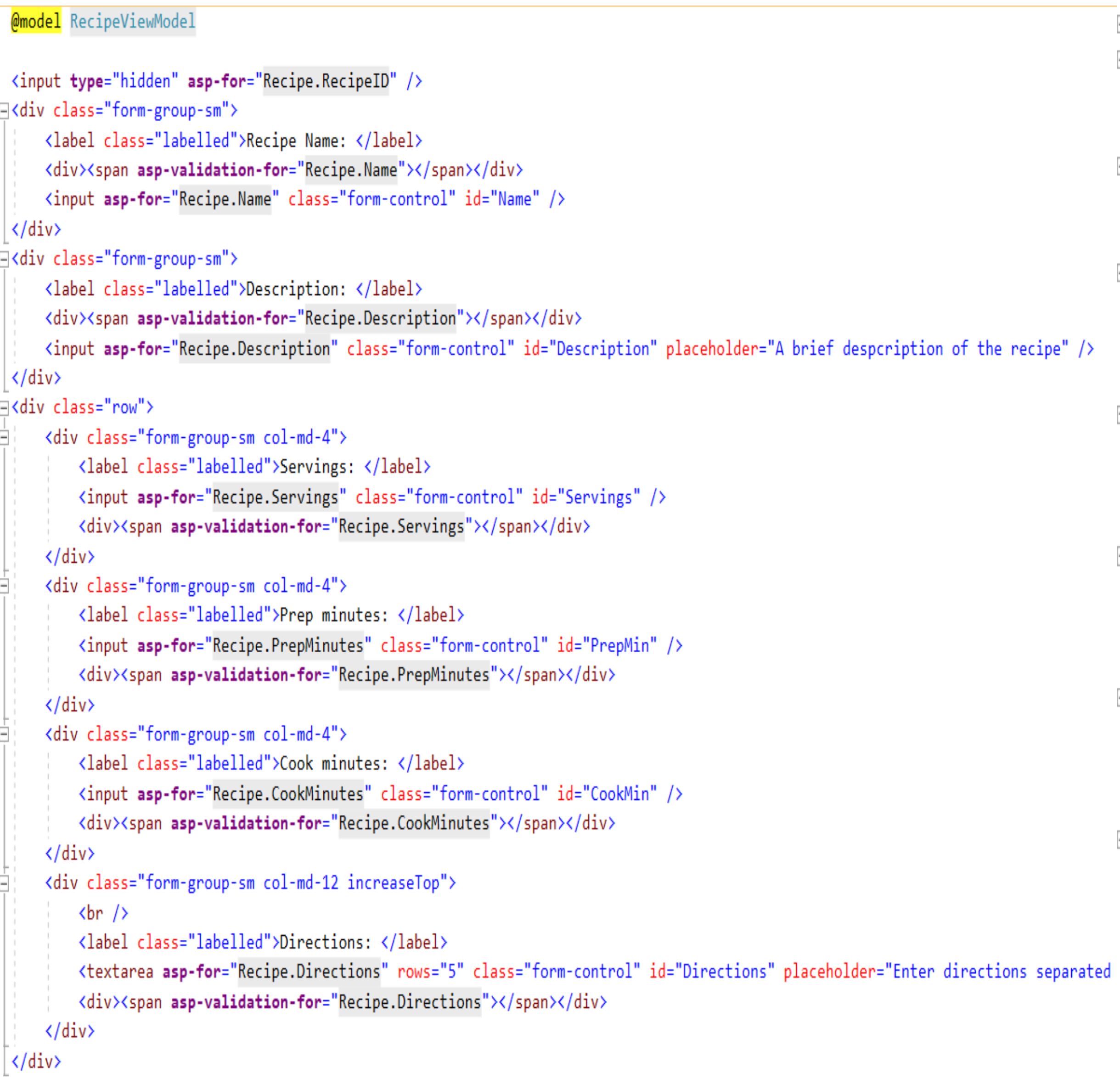


```
@model RecipeViewModel
@{
    ViewData["Title"] = "View";
}

<div id="rOverview" class="container">
    <div class="row label-info" id="h1Format">
        <div class="col-sm-8 pull-sm-4"><h1 class="label-info" id="h1Format"><b>@Model.Recipe.Name</b></h1></div>
        <div class="col-sm-1">
            <a class="btn btn-default btn-warning center-block" asp-action="Edit" asp-route-RecipeID=@Model.Recipe.RecipeID>Edit Recipe</a>
        </div>
    </div>
    <h4 class="label-info" id="h4Format"><b>@Model.Recipe.Description</b></h4>
    <table class="table table-sm table-striped table-bordered">
        <thead>
            <tr id="tHead">
                <th>Servings:</th>
                <th>Prep Time:</th>
                <th>Cook Time:</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <th>@Model.Recipe.Servings</th>
                <th>@Model.Recipe.PrepMinutes</th>
                <th>@Model.Recipe.CookMinutes</th>
            </tr>
        </tbody>
    </table>
</div>
```

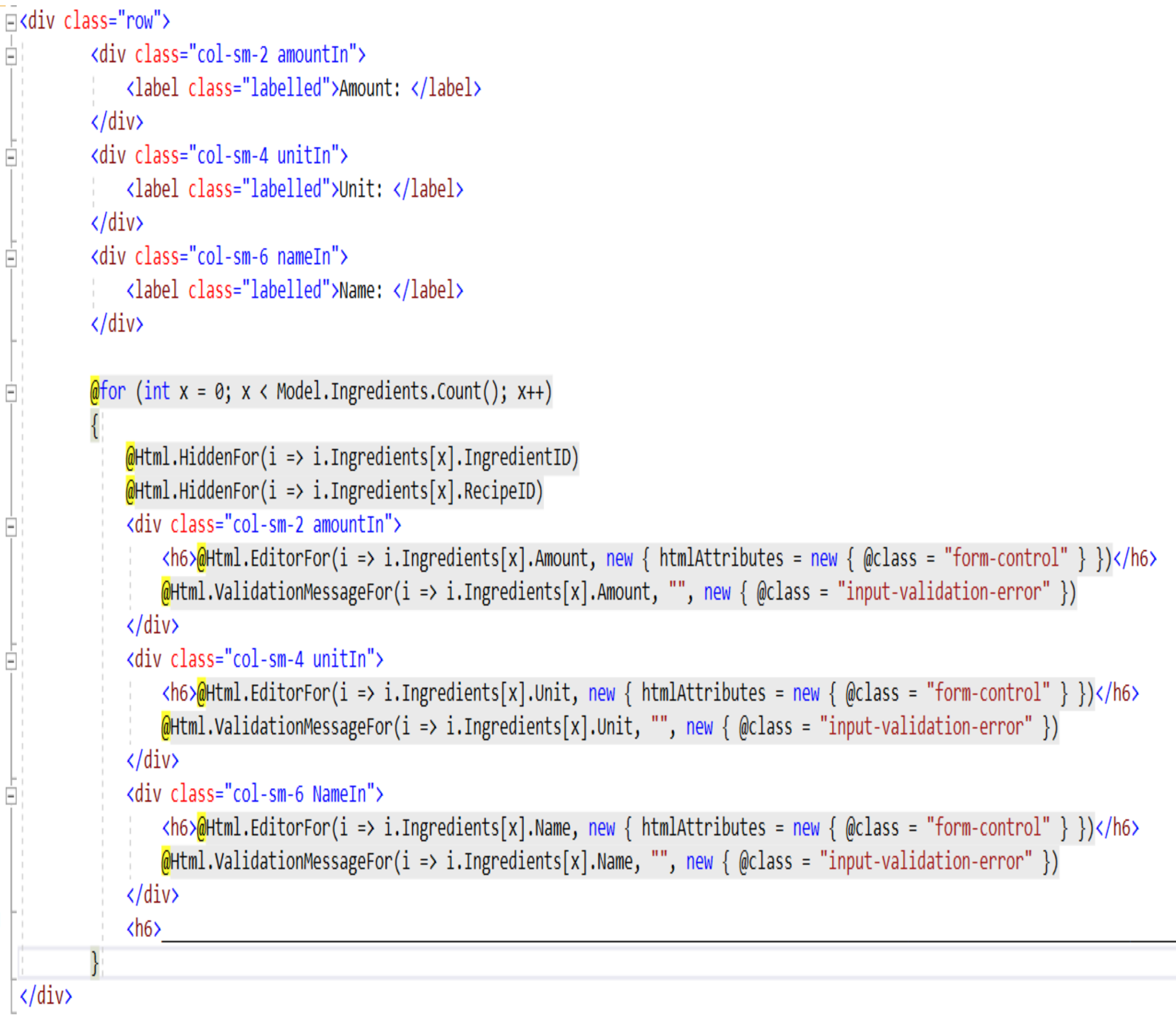


```
</div>
<div class="container">
    <div id="rIngredients" class="mark">
        <h2><b><u>INGREDIENTS</u></b></h2>
        <ul class="list-unstyled">
            @foreach(Ingredient i in Model.Ingredients)
            {
                <li>@i.Amount @i.Unit @i.Name</li>
            }

            <li><a  asp-action="IngAddToRecipe" asp-controller="Ingredient" asp-route-RecipeID="@Model.Recipe.RecipeID"
                   class="btn btn-success glyphicon-plus">Add New Ingredient</a>
            </li>
        </ul>
    </div>
</div>
<br />
<div class="container">
    <div id="rDirections" class="mark">
        <h2><b><u>DIRECTIONS</u></b></h2>
        <ul class="list-unstyled">
            @{
                string[] dirList = Model.Recipe.Directions.Split(".");

                foreach (var line in dirList)
                {

                    <li>@line</li>

                }
            }
        </ul>
    </div>
</div>
```

Shared/FullRecipeInput:



```
@model RecipeViewModel

<input type="hidden" asp-for="Recipe.RecipeID" />
<div class="form-group-sm">
    <label class="labelled">Recipe Name: </label>
    <div><span asp-validation-for="Recipe.Name"></span></div>
    <input asp-for="Recipe.Name" class="form-control" id="Name" />
</div>
<div class="form-group-sm">
    <label class="labelled">Description: </label>
    <div><span asp-validation-for="Recipe.Description"></span></div>
    <input asp-for="Recipe.Description" class="form-control" id="Description" placeholder="A brief despcription of the recipe" />
</div>
<div class="row">
    <div class="form-group-sm col-md-4">
        <label class="labelled">Servings: </label>
        <input asp-for="Recipe.Servings" class="form-control" id="Servings" />
        <div><span asp-validation-for="Recipe.Servings"></span></div>
    </div>
    <div class="form-group-sm col-md-4">
        <label class="labelled">Prep minutes: </label>
        <input asp-for="Recipe.PrepMinutes" class="form-control" id="PrepMin" />
        <div><span asp-validation-for="Recipe.PrepMinutes"></span></div>
    </div>
    <div class="form-group-sm col-md-4">
        <label class="labelled">Cook minutes: </label>
        <input asp-for="Recipe.CookMinutes" class="form-control" id="CookMin" />
        <div><span asp-validation-for="Recipe.CookMinutes"></span></div>
    </div>
    <div class="form-group-sm col-md-12 increaseTop">
        <br />
        <label class="labelled">Directions: </label>
        <textarea asp-for="Recipe.Directions" rows="5" class="form-control" id="Directions" placeholder="Enter directions separated
        <div><span asp-validation-for="Recipe.Directions"></span></div>
    </div>
</div>
```



```
<div class="row">
        <div class="col-sm-2 amountIn">
            <label class="labelled">Amount: </label>
        </div>
        <div class="col-sm-4 unitIn">
            <label class="labelled">Unit: </label>
        </div>
        <div class="col-sm-6 nameIn">
            <label class="labelled">Name: </label>
        </div>

        @for (int x = 0; x < Model.Ingredients.Count(); x++)
        {
            @Html.HiddenFor(i => i.Ingredients[x].IngredientID)
            @Html.HiddenFor(i => i.Ingredients[x].RecipeID)
            <div class="col-sm-2 amountIn">
                <h6>@Html.EditorFor(i => i.Ingredients[x].Amount, new { htmlAttributes = new { @class = "form-control" } })</h6>
                @Html.ValidationMessageFor(i => i.Ingredients[x].Amount, "", new { @class = "input-validation-error" })
            </div>
            <div class="col-sm-4 unitIn">
                <h6>@Html.EditorFor(i => i.Ingredients[x].Unit, new { htmlAttributes = new { @class = "form-control" } })</h6>
                @Html.ValidationMessageFor(i => i.Ingredients[x].Unit, "", new { @class = "input-validation-error" })
            </div>
            <div class="col-sm-6 NameIn">
                <h6>@Html.EditorFor(i => i.Ingredients[x].Name, new { htmlAttributes = new { @class = "form-control" } })</h6>
                @Html.ValidationMessageFor(i => i.Ingredients[x].Name, "", new { @class = "input-validation-error" })
            </div>
            <h6>
        }
</div>
```