

# Model ASP.NET MVC Core 3.1 Web App - Deel 4

## 1 Start project

In deel 3 zijn we gekomen tot deze code: <https://github.com/CSharpSyntraWest/PittigRestoMVC>

We starten vanaf deze code

## 2 Aanpassen van MenuItem Controller en CRUD views om images te kunnen uploaden en bewaren op de webserver

Om images te kunnen uploaden en te bewaren onder de wwwroot/images folder vanaf de Create MenuItem razor view, moeten we de Create action methoden aanpassen van de **MenuItemController**:

### 1. Open MenuItemController onder de folder Areas/Admin/Controllers en wijzig de Edit action methoden:

```
//GET - EDIT
public async Task<IActionResult> Edit(int? id)
{
    if(id==null)
    {
        return NotFound();
    }

    MenuItemVM.MenuItem = await _db.MenuItem.Include(m => m.Category).Include(m
=> m.SubCategory).SingleOrDefaultAsync(m => m.Id == id);
    MenuItemVM.SubCategory = await _db.SubCategory.Where(s => s.CategoryId ==
MenuItemVM.MenuItem.CategoryId).ToListAsync();

    if(MenuItemVM.MenuItem ==null)
    {
        return NotFound();
    }
    return View(MenuItemVM);
}

[HttpPost, ActionName("Edit")]
[ValidateAntiForgeryToken]
public async Task<IActionResult> EditPOST(int? id)
{
    if(id==null)
    {
        return NotFound();
    }
    MenuItemVM.MenuItem.SubCategoryId =
Convert.ToInt32(Request.Form["SubCategoryId"].ToString());
```

```

        if (!ModelState.IsValid)
        {
            MenuItemVM.SubCategory = await _db.SubCategory.Where(s => s.CategoryId ==
MenuItemVM.MenuItem.CategoryId).ToListAsync();
            return View(MenuItemVM);
        }

        //Work on the image saving section

        string webRootPath = _hostingEnvironment.WebRootPath;
        var files = HttpContext.Request.Form.Files;

        var menuItemFromDb = await _db.MenuItem.FindAsync(MenuItemVM.MenuItem.Id);

        if (files.Count > 0)
        {
            //New Image has been uploaded
            var uploads = Path.Combine(webRootPath, "images");
            var extension_new = Path.GetExtension(files[0].FileName);

            //Delete the original file
            var imagePath = Path.Combine(webRootPath,
menuItemFromDb.Image.TrimStart('\\'));

            if(System.IO.File.Exists(imagePath))
            {
                System.IO.File.Delete(imagePath);
            }

            //we will upload the new file
            using (var filesStream = new FileStream(Path.Combine(uploads,
MenuItemVM.MenuItem.Id + extension_new), FileMode.Create))
            {
                files[0].CopyTo(filesStream);
            }
            menuItemFromDb.Image = @"\"images\" + MenuItemVM.MenuItem.Id +
extension_new;
        }

        menuItemFromDb.Name = MenuItemVM.MenuItem.Name;
        menuItemFromDb.Description = MenuItemVM.MenuItem.Description;
        menuItemFromDb.Price = MenuItemVM.MenuItem.Price;
        menuItemFromDb.Spicyiness = MenuItemVM.MenuItem.Spicyiness;
        menuItemFromDb.CategoryId = MenuItemVM.MenuItem.CategoryId;
        menuItemFromDb.SubCategoryId = MenuItemVM.MenuItem.SubCategoryId;

        await _db.SaveChangesAsync();

        return RedirectToAction(nameof(Index));
    }

```

## 2. En passen eveneens de Edit.cshtml aan onder Areas/Views/MenuItem:

```

@model PittigRestoMVC.Models.ViewModels.MenuItemViewModel
@using PittigRestoMVC.Extensions

```

```

@{
    ViewData["Title"] = "Wijzig";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br />
<h2 class="text-info">Wijzig Menu</h2>
<br />

<form method="post" asp-action="Create" enctype="multipart/form-data">
    <div class="border backgroundWhite row">
        <input type="hidden" asp-for="MenuItem.Id" />

        <div asp-validation-summary="ModelOnly" class="text-danger"></div>

        <div class="col-8">
            <div class="form-group row">
                <div class="col-4">
                    <label asp-for="MenuItem.Name" class="col-form-label"></label>
                </div>
                <div class="col-8">
                    <input asp-for="MenuItem.Name" class="form-control" />
                </div>
                <span asp-validation-for="MenuItem.Name" class="text-danger"></span>
            </div>
            <div class="form-group row">
                <div class="col-4">
                    <label asp-for="MenuItem.Description" class="col-form-label"></label>
                </div>
                <div class="col-8">
                    @Html.TextAreaFor(m => m.MenuItem.Description)
                </div>
                <span asp-validation-for="MenuItem.Description" class="text-
danger"></span>
            </div>
            <div class="form-group row">
                <div class="col-4">
                    <label asp-for="MenuItem.Price" class="col-form-label"></label>
                </div>
                <div class="col-8">
                    <input asp-for="MenuItem.Price" class="form-control" />
                </div>
                <span asp-validation-for="MenuItem.Price" class="text-danger"></span>
            </div>
            <div class="form-group row">
                <div class="col-4">
                    <label asp-for="MenuItem.Image" class="col-form-label"></label>
                </div>
                <div class="col-8">
                    <input type="file" name="files" multiple class="form-control" />
                </div>
            </div>
            <div class="form-group row">
                <div class="col-4">
                    <label asp-for="MenuItem.CategoryId" class="col-form-label"></label>
                </div>
                <div class="col-8">

```

```

                <select asp-for="@Model.MenuItem.CategoryId" id="CategoryId" asp-
items="Model.Category.ToSelectListItem(Model.MenuItem.CategoryId)" class="form-
control"></select>
            </div>
        </div>
        <div class="form-group row">
            <div class="col-4">
                <label asp-for="MenuItem.SubCategoryId" class="col-form-
label"></label>
            </div>
            <div class="col-8">
                <select asp-for="@Model.MenuItem.SubCategoryId" name="SubCategoryId"
id="SubCategoryId" asp-items="@((new SelectList(string.Empty, "Id", "Name")))" class="form-
control"></select>
            </div>
        </div>
        <div class="form-group row">
            <div class="col-4">
                <label asp-for="MenuItem.Spicyness" class="col-form-label"></label>
            </div>
            <div class="col-8">
                <select asp-for="MenuItem.Spicyness" asp-
items="Html.GetEnumSelectList<MenuItem.ESpicy>()" class="form-control"></select>
            </div>
        </div>
        </div>
        <div class="col-3 offset-1">
            
        </div>

        <div class="col-8">
            <div class="col-8 offset-4">
                <partial name="_EditAndBackToListButton" model="Model.MenuItem.Id" />
            </div>
        </div>
    </div>
</form>

```

```

@section Scripts{
    <script type="text/javascript">
        $(document).ready(function () {
            getSubCategory();
            $('#CategoryId').change(function () {
                getSubCategory();
            });
        });

        function getSubCategory() {
            var url = '@Url.Content("~/")' + "Admin/SubCategory/GetSubCategory";
            var ddlsorce = '#CategoryId';
            $.getJSON(url, { id: $(ddlsorce).val() }, function (data) {
                var items = '';
                $('#SubCategoryId').empty();
            });
        }
    </script>

```

```

        $.each(data, function (i, subcategory) {
            items += "<option value='" + subcategory.value + "'>" +
subcategory.text + "</option>";
        });

        $('#SubCategoryId').html(items);
    }
</script>

@{ await Html.RenderPartialAsync("_ValidationScriptsPartial"); }
}

```

Deze Edit Razor view maakt gebruik van een Partial View \_EditAndBackToListButton.cshtml

3. Maak een Partial View \_EditAndBackToListButton.cshtml deze aan onder Views/Shared:

```

@model int

<div class="row">
    <div class="col-6">
        <input type="submit" class="btn btn-info form-control" asp-route-id="@Model"
value="Update" />
    </div>
    <div class="col-6">
        <a asp-action="Index" class="btn btn-success form-control">Terug naar
lijst</a>
    </div>
</div>

```

## 3 Kortingen toevoegen

### 3.1 Aanmaken model voor Coupon

Onder de folder models, voeg een nieuwe class toe voor Coupon

```

public class Coupon
{
    [Key]
    public int Id { get; set; }

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

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

    public enum ECouponType { Percent = 0, Euro = 1 }

    [Required]
    [Display(Name = "Korting")]
}

```

```

    public double Discount { get; set; }

    [Required]
    [Display(Name = "Minimum bedrag")]
    public double MinimumAmount { get; set; }
    [Display(Name = "Afbeelding")]
    public byte[] Picture { get; set; }

    [Display(Name = "Is Actief")]
    public bool IsActive { get; set; }
}

```

### 3.2 Aanpassen ApplicationDbContext

Voeg aan de class ApplicationDbContext DbSet public property toe voor Coupon

```

public DbSet<Coupon> Coupon { get; set; }

```

### 3.3 Database migraties toevoegen voor de nieuwe Coupon tabel en database updaten

Indien je applicatie goed werkt, kan je nu de tabellen in de database creëren.

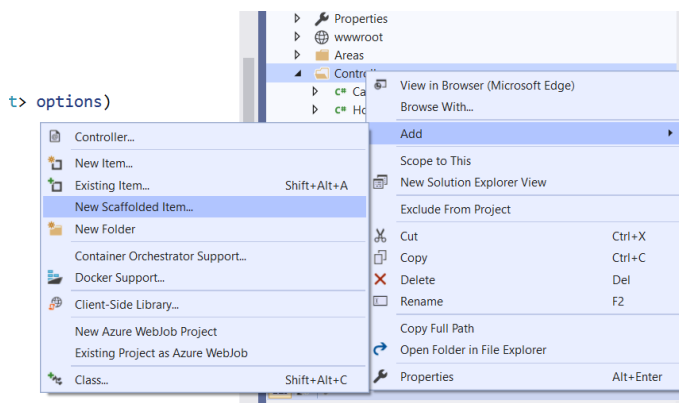
We doen dit via de **NuGet Package Manager Console** door eerst data-migraties op te zetten en daarna update-database commando

```
PM> add-migration AddedCouponToDb
```

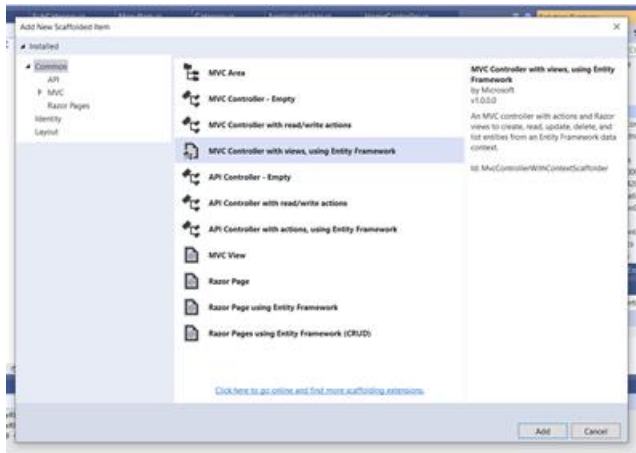
```
PM> update-database
```

### 3.4 Aanmaken Controller en Views voor CRUD operaties voor Coupon

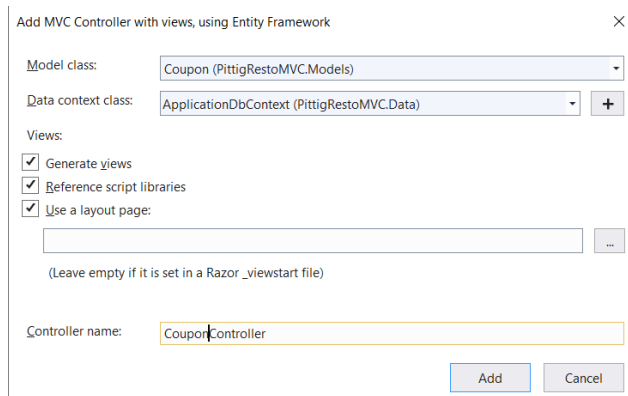
Rechtsklik op de folder Controller en kies Add/New Scaffolded Item...



En daarna Mvc Controller with views, using Entity Framework



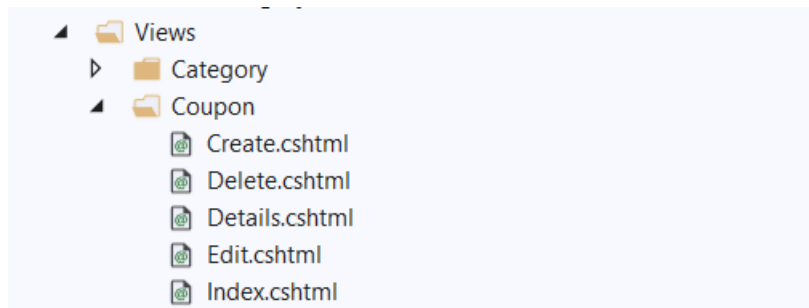
Kies in het volgende popup-venster bv als **model class Coupon**



**Data class ApplicationDbContext**

**Controller name CouponController**

Zorg dat de **Generate views aangevinkt** staat, deze zal dan onder de Views/Coupon folder Razor views genereren voor een Coupon (Create.cshtml, Delete.cshtml, Edit.cshtml en Index.cshtml, Details.cshtml) die een standaard UI voorzien waarin een gebruiker CRUD operaties kan verrichten voor Coupons.



Onder de folder controller is er eveneens een CouponController.cs toegevoegd

### 3.5 Aanpassen van Create en Edit action methods in de CouponController

**Om een bestand in binair formaat te kunnen opslaan in de database (de coupon-image), moeten de Create en Edit action methods worden aangepast.**

**Open de CouponController en pas de methoden Create en Edit aan:**

```
public IActionResult Create()
{
    return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create(Coupon coupons)
{
    if (ModelState.IsValid)
```



```

        {
            var files = HttpContext.Request.Form.Files;
            if (files.Count > 0)
            {
                byte[] p1 = null;
                using (var fs1 = files[0].OpenReadStream())
                {
                    using (var ms1 = new MemoryStream())
                    {
                        fs1.CopyTo(ms1);
                        p1 = ms1.ToArray();
                    }
                }
                coupons.Picture = p1;
            }
            _context.Coupon.Add(coupons);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        return View(coupons);
    }
}

//GET Edit Coupon
public async Task<IActionResult> Edit(int? id)
{
    if (id == null)
    {
        return NotFound();
    }
    var coupon = await _context.Coupon.SingleOrDefaultAsync(m => m.Id == id);
    if (coupon == null)
    {
        return NotFound();
    }
    return View(coupon);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Edit(Coupon coupons)
{
    if (coupons.Id == 0)
    {
        return NotFound();
    }

    var couponFromDb = await _context.Coupon.Where(c => c.Id == coupons.Id).FirstOrDefaultAsync();

    if (ModelState.IsValid)
    {
        var files = HttpContext.Request.Form.Files;
        if (files.Count > 0)
        {
            byte[] p1 = null;
            using (var fs1 = files[0].OpenReadStream())
            {

```

```

        using (var ms1 = new MemoryStream())
        {
            fs1.CopyTo(ms1);
            p1 = ms1.ToArray();
        }
        couponFromDb.Picture = p1;
    }
    couponFromDb.MinimumAmount = coupons.MinimumAmount;
    couponFromDb.Name = coupons.Name;
    couponFromDb.Discount = coupons.Discount;
    couponFromDb.CouponType = coupons.CouponType;
    couponFromDb.IsActive = coupons.IsActive;

    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
}
return View(coupons);
}

```

### 3.6 Aanpassen van Coupon Razor views Create en Edit

Om een image te kunnen uploaden in de Razor views, moeten er eveneens aanpassingen in deze views worden verricht:

1. Open Areas/Admin/Views/Coupon/Create.cshtml en pas de Razor code aan:

```

@model PittigRestoMVC.Models.Coupon
@{
    ViewData["Title"] = "Nieuw";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br />
<h2 class="text-info">Aanmaken Korting</h2>
<br />

<form method="post" enctype="multipart/form-data">
    <div class="border backgroundWhite">
        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Name" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input asp-for="Name" class="form-control" />
            </div>
            <span asp-validation-for="Name" class="text-danger"></span>
        </div>
        <div class="form-group row">
            <div class="col-2">
                <label asp-for="Picture" class="col-form-label"></label>
            </div>
            <div class="col-5">
                <input type="file" id="projectImage" name="files" multiple
class="form-control" />
            </div>
        </div>
        <div class="form-group row">

```

```

        <div class="col-2">
            <label asp-for="CouponType" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <select asp-for="CouponType" asp-
items="Html.GetEnumSelectList<Coupon.ECouponType>()" class="form-
control"></select>
        </div>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Discount" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Discount" class="form-control" />
        </div>
        <span asp-validation-for="Discount" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="MinimumAmount" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="MinimumAmount" class="form-control" />
        </div>
        <span asp-validation-for="MinimumAmount" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label class="form-check-label" asp-for="IsActive"></label>
        </div>
        <div class="col-5">
            <input class="form-check-input mx-1" type="checkbox" asp-
for="IsActive">
        </div>
    </div>
    <div class="form-group row">
        <div class="col-5 offset-2">
            <partial name="_CreateAndBackToListButton" />
        </div>
    </div>
</div>
</form>

@section Scripts{
    @{ await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}

```

## 2. Open Areas/Admin/Views/Coupon/Edit. cshtml en pas de Razor code aan:

```

@model PittigRestoMVC.Models.Coupon
@{
    ViewData["Title"] = "Wijzig";
}

<br />
<h2 class="text-info">Wijzig korting</h2>
<br />

```

```

<form asp-action="Edit" method="post" enctype="multipart/form-data">
  <div class="row borderWidth">

    <br />
    <div class="col-12 border">
      @{
        var base64 = Convert.ToBase64String(Model.Picture);
        var imgsrc = string.Format("data:image/jpg;base64,{0}", base64);
      }
      
    </div>
    <div class="col-12 pt-4">
      <input hidden asp-for="Id" />
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group row">
        <div class="col-2">
          <label asp-for="Name" class="custom-label"></label>
        </div>
        <div class="col-5">
          <input asp-for="Name" class="form-control" />
        </div>
        <span asp-validation-for="Name" class="text-danger"></span>
      </div>
      <div class="form-group row">
        <div class="col-2">
          <label asp-for="Picture" class="custom-label"></label>
        </div>
        <div class="col-5">
          <input type="file" id="projectImage" name="files" multiple
class="form-control" />
        </div>
      </div>
      <div class="form-group row">
        <div class="col-2">
          <label asp-for="CouponType" class="custom-label"></label>
        </div>
        <div class="col-5">
          <select asp-for="CouponType" asp-
items="Html.GetEnumSelectList<Coupon.ECouponType>()" class="form-
control"></select>
        </div>
      </div>
      <div class="form-group row">
        <div class="col-2">
          <label asp-for="Discount" class="custom-label"></label>
        </div>
        <div class="col-5">
          <input asp-for="Discount" class="form-control" />
        </div>
        <span asp-validation-for="Discount" class="text-danger"></span>
      </div>
      <div class="form-group row">
        <div class="col-2">
          <label asp-for="MinimumAmount" class="custom-label"></label>
        </div>
        <div class="col-5">
          <input asp-for="MinimumAmount" class="form-control" />
        </div>
      </div>
    </div>
  </div>

```

```

        <span asp-validation-for="MinimumAmount" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="IsActive" class="custom-label"></label>
        </div>
        <div class="col-5">
            <input type="checkbox" asp-for="IsActive" class="form-control"
/>
        </div>
    </div>
    <br />
    <div class="form-group row">
        <div class="col-5 offset-2">
            <partial name="_EditAndBackToListButton" model="Model.Id" />
        </div>
    </div>
</div>
</form>

```

3. Open als laatste Areas/Admin/Coupons/Index.cshtml en pas eveneens de Razor view met lijst van kortingen aan:

```

@model IEnumerable<Coupon>

@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br />
<br />
<div class="border backgroundWhite">
    <div class="row">
        <div class="col-6">
            <h2 class="text-info"> Lijst van kortingen</h2>
        </div>
        <div class="col-6 text-right">
            <p>
                <a asp-action="Create" class="btn btn-info"><i class="fas fa-
plus"></i> &nbsp; Nieuw </a>
            </p>
        </div>
    </div>
    <br />
    <div>
        @if (Model.Count() > 0)
        {
            <table class="table table-striped border">
                <tr class="table-secondary">
                    <th>
                        @Html.DisplayNameFor(m => m.Name)
                    </th>
                    <th>
                        @Html.DisplayNameFor(m => m.Discount)

```

```

        </th>
        <th>
            @Html.DisplayNameFor(m => m.MinimumAmount)
        </th>
        <th>
            @Html.DisplayNameFor(m => m.IsActive)
        </th>
        <th></th>
        <th></th>
    </tr>
    @foreach (var item in Model)
    {
    <tr>
        <td>
            @Html.DisplayFor(m => item.Name)
        </td>
        <td>
            @Html.DisplayFor(m => item.Discount)
        </td>
        <td>
            @Html.DisplayFor(m => item.MinimumAmount)
        </td>
        <td>
            @Html.DisplayFor(m => item.IsActive)
        </td>
        <td>
            <a asp-action="Edit" asp-route-id="@item.Id">Edit</a> |
            <a asp-action="Details" asp-route-id="@item.Id">Details</a> |
            <a asp-action="Delete" asp-route-id="@item.Id">Delete</a>
        </td>
    </tr>
    }
    </table>
    }
    else
    {
        <p>Er bestaan nog geen kortingen...</p>
    }
    </div>
</div>

```

Start de applicatie en test nu de applicatie met deze url:

<https://localhost:xxxxx/Admin/Coupon/index>

## 4 Tonen van Categorieën,coupons en menuitems op de homepagina

We moeten op de homepagina gegevens uit verschillende entiteiten en tonen en dus uit tabellen van de database halen.

Een goede manier om dit te doen is om een ViewModel class te maken die de gegevens uit verschillende tabellen zal bijhouden. In de HomeController vullen we de properties van een object van deze ViewModel in en halen de gegevens op vanuit. Ten slotte geeft de Controller dit ViewModel object door aan de View.

We maken een nieuwe folder Models/ViewModels om onze ViewModel class te plaatsen.

**Maak een folder ViewModels onder de folder Models**

**Maak onder ViewModels folder een nieuwe class IndexViewModel**

```
public class IndexViewModel
{
    public IEnumerable<MenuItem> MenuItem { get; set; }
    public IEnumerable<Category> Category { get; set; }
    public IEnumerable<Coupon> Coupon { get; set; }
}
```

**Open de Controller Areas\Customer\Controllers\HomeController.cs**

**Verwijder eerst deze code:**

```
private readonly ILogger<HomeController> _logger;

public HomeController(ILogger<HomeController> logger)
{
    _logger = logger;
}
```

We hebben **gegevens nodig uit de ApplicationDbContext**. Deze is reeds in de ConfigureServices in startup.cs geregistreerd als service in de DI Container van ASP.Net Core

We kunnen deze service **nu injecteren in onze HomeController** en we gaan dit doen via **Constructor-injection**:

**Maak hiervoor een private field aan waarin we een ApplicationDbContext object kunnen bijhouden.**

```
private readonly ApplicationDbContext _db;
```

**En een constructor die deze als parameter aanneemt aan de private field initialiseert:**

```
public HomeController(ApplicationDbContext db)
{
    _db = db;
}
```

**Pas nu de Index() methode aan. Deze wordt uitgevoerd bij een HttpGet Request naar de Home-pagina**

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PittigRestoMVC.Data;
using PittigRestoMVC.Models;
using PittigRestoMVC.Models.ViewModels;

namespace Pittig.Controllers
{
    [Area("Customer")]
    public class HomeController : Controller
    {
        private readonly ApplicationDbContext _db;

        public HomeController(ApplicationDbContext db)
        {
            _db = db;
        }

        public async Task<IActionResult> Index()
        {
            IndexViewModel IndexVM = new IndexViewModel()
            {
                MenuItem = await _db.MenuItem.Include(m =>
                    m.Category).Include(m => m.SubCategory).ToListAsync(),
                Category = await _db.Category.ToListAsync(),
                Coupon = await _db.Coupon.Where(c => c.IsActive == true).ToListAsync()
            };

            return View(IndexVM);
        }

        public IActionResult Privacy()
        {
            return View();
        }

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



**Opmerking:** Aangezien we nu asynchrone calls maken in de Index methode, moet deze ook asynchroon worden

```
public async Task<IActionResult> Index()
```

Hier wordt een object van IndexViewModel aangemaakt en de Menus, Category en Coupons worden door asynchrone aanroepen uit de database gehaald en ingevuld in de public properties van het object.

**Ten slotte wordt het object meegegeven aan de View (standaard heeft de view dezelfde naam als de action method, dus in dit geval zal dit dus de view Index.cshtml zijn onder de folder Area/Customr/Home/**

**Opmerking:** De .Include() in LINQ to Entities bij bv\_db.MenuItem, Category en SubCategory

Is nodig omdat **EF Core Lazy Loading** gebruikt en de gegevens uit gerefereerde tabellen niet automatisch zal opladen. Dus indien je eveneens gegevens wil ophalen van de tabel die gelinkt is met een FK relatie, moet je dit expliciet vermelden in de LINQ Query. Indien je dit niet doet, kan je null waarden terugkrijgen voor de gelinkte data.

## 5 Pas de View van de homepagina aan

1. We maken eerst een partial Razor view “\_ThumbnailAreaPartial” onder de folder Views/Shared  
Deze krijgt een Model van type IEnumerable<MenuItem> binnen en deze wordt afgebeeld

```
@model IEnumerable<MenuItem>
@*
    For more information on enabling MVC for empty projects, visit
    http://go.microsoft.com/fwlink/?LinkID=397860
*@
@if (Model.Count() > 0)
{
    <div class="col-12 post @Model.FirstOrDefault().Category.Name.Replace("
",string.Empty) menu-restaurant">
        <div class="row">
            <h3 class="text-success"> @Model.FirstOrDefault().Category.Name </h3>
        </div>

        @foreach(var item in Model)
        {
            <div class="border border-info rounded col-12" style="margin-
bottom:10px; margin-top:10px; padding:10px">
                <div class="row">
                    <div class="col-md-3 col-sm-12">
                        
                    </div>
                    <div class="col-md-9 col-sm-12">
                        <div class="row pr-3">
                            <div class="col-8">
```

```

<label class="text-primary" style="font-
size:21px;color:maroon">@item.Name</label>
@if (item.Spicyness == "1")
{
    <img srcset="/images/mild.png" title="Mild" />
}
@if (item.Spicyness == "2")
{
    <img srcset="/images/spicy.png" title="Pittig"
/>
}
@if (item.Spicyness == "3")
{
    <img srcset="/images/veryspicy.png"
title="Zeer Pittig" />
}
</div>
<div class="col-4 text-right" style="color:maroon">
    <h4>@item.Price €</h4>
</div>
</div>

<div class="row col-12 text-justify d-none d-md-block">
    <p>@Html.Raw(item.Description)</p>
</div>
<div class="col-md-3 col-sm-12 offset-md-9 text-center">
    <a asp-action="Details" class="btn btn-success form-
control" asp-route-id="@item.Id">Details</a>
</div>
</div>
</div>
</div>
}
<div class="p-4"></div>
</div>
}

```

2. Open de Index.cshtml onder Area/Customer/Home/ Deze zal een 'carousel' tonen van de images van actieve kortingen, een lijst van categorieën tonen en we refereren de partial view die we in de vorige stap hebben aangemaakt die voor een categorie de lijst van menuitems toont.

De Index View ontvangt een object van Type IndexViewModel van de HomeController's Index() action method. We specificëren bovenaan deze pagina het type van het object dat binnenkomt dat ons Model zal zijn in deze view:

```
@model PittigRestoMVC.Models.ViewModels.IndexViewModel
```

```
<br />
```

```

@if (Model.Coupon.ToList().Count > 0)
{
    <div class="border">
        <div class="carousel" data-ride="carousel" data-interval="2500">
            @for (int i = 0; i < Model.Coupon.Count(); i++)

```

```

        {
            if (i == 0)
            {
                <div class="carousel-item active">
                    @{
                        var base64 =
Convert.ToBase64String(Model.Coupon.ToList()[i].Picture);
                        var imgsrc =
string.Format("data:image/jpg;base64,{0}", base64);
                    }

                    
                </div>
            }
            else
            {
                <div class="carousel-item">
                    @{
                        var base64 =
Convert.ToBase64String(Model.Coupon.ToList()[i].Picture);
                        var imgsrc =
string.Format("data:image/jpg;base64,{0}", base64);
                    }

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

<br />
<br />

<div class="backgroundWhite container">

    <ul id="menu-filters" class="menu-filter-list list-inline text-center">
        <li class="active btn btn-secondary ml-1 mr-1" data-filter=".menu-
restaurant">Alles</li>

        @foreach (var item in Model.Category)
        {
            <li class="ml-1 mr-1" data-filter=".@item.Name.Replace("
",string.Empty)">@item.Name</li>
        }
    </ul>

    @foreach (var category in Model.Category)
    {
        <div class="row" id="menu-wrapper">
            <partial name="_ThumbnailAreaPartial"
model="@Model.MenuItem.Where(u=>u.Category.Name.Equals(category.Name))" />
        </div>
    }
}

```

```

    }
</div>

@section Scripts{
    <script src="https://code.jquery.com/jquery-3.3.1.js"
        integrity="sha256-2Kok7Mb0yxpqUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60="
        crossorigin="anonymous"></script>

    <script>var posts = $('.post');

    (function ($) {

        $("#menu-filters li").click(function () {
            $("#menu-filters li").removeClass('active btn btn-secondary');
            $(this).addClass('active btn btn-secondary');

            var selectedFilter = $(this).data("filter");

            $(".menu-restaurant").fadeOut();

            setTimeout(function () {
                $(selectedFilter).slideDown();
            }, 300);
        });

    })(jQuery);</script>
}

```

## 6 GDPR: wetgeving i.v.m. de bescherming van persoonsgegevens

Om de toestemming te vragen aan de gebruiker om cookies te gebruiken (GDPR):

Voeg een Partial Razor view “\_CookieConsentPartial” toe onder de folder Views/Shared

```

@using Microsoft.AspNetCore.Http.Features

@{
    var consentFeature = Context.Features.Get<ITrackingConsentFeature>();
    var showBanner = !consentFeature?.CanTrack ?? false;
    var cookieString = consentFeature?.CreateConsentCookie();
}

@if (showBanner)
{
    <div id="cookieConsent" class="alert alert-info alert-dismissible fade show"
        role="alert">
        Use this space to summarize your privacy and cookie use policy. <a asp-area=""
        asp-controller="Home" asp-action="Privacy">Learn More</a>.
        <button type="button" class="accept-policy close" data-dismiss="alert" aria-
        label="Close" data-cookie-string="@cookieString">
            <span aria-hidden="true">Accept</span>
        </button>
    </div>
}

```

```

<script>
    (function () {
        var button = document.querySelector("#cookieConsent button[data-cookie-
string]");
        button.addEventListener("click", function (event) {
            document.cookie = button.dataset.cookieString;
        }, false);
    })();
</script>
}

```

Als laatste layout wijzigingen:

- aanpassen van css styles in wwwroot/css/site.css



site.css

- aanpassen van Views/Shared/\_Layout.cshtml



\_Layout.cshtml