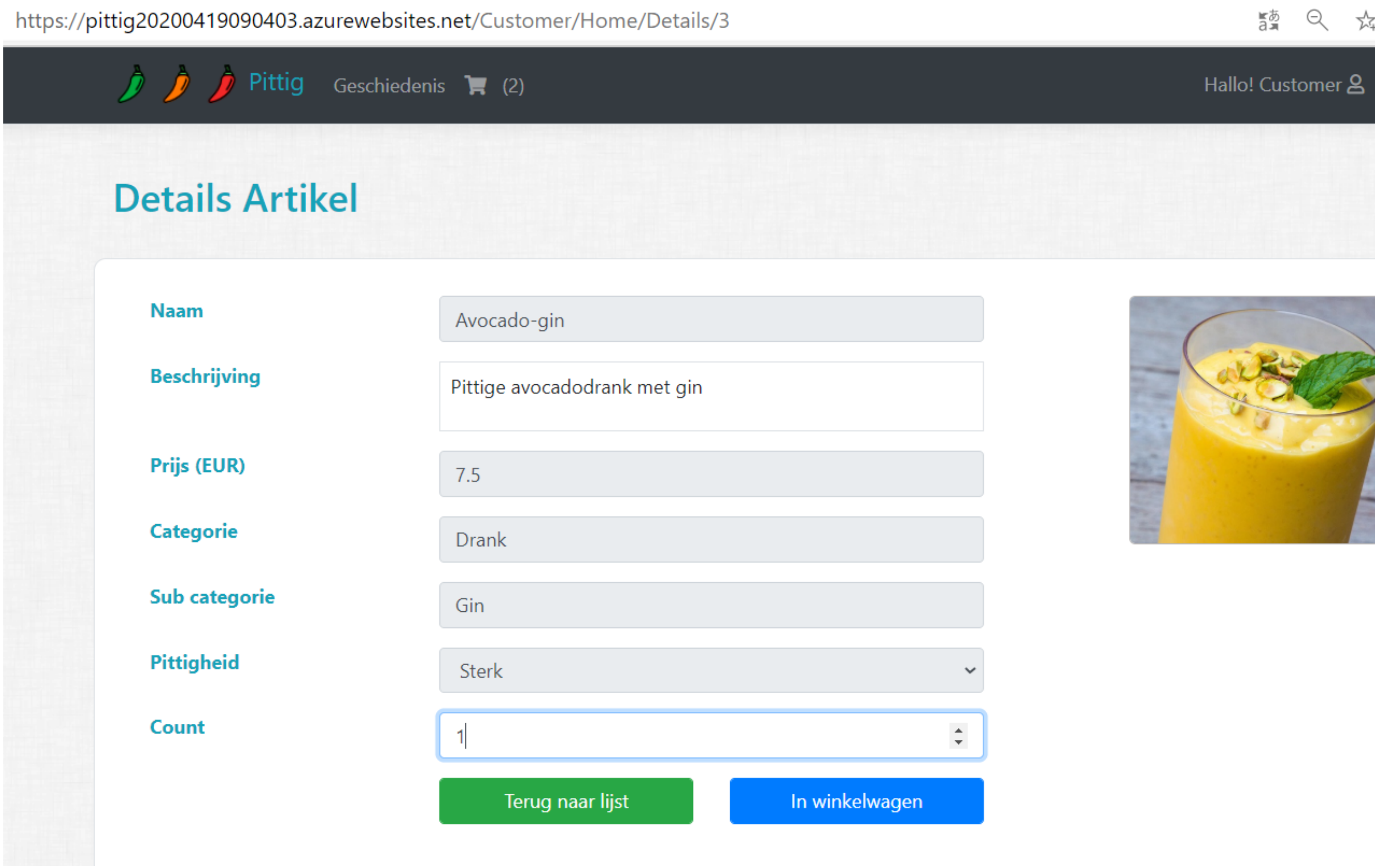
Model ASP.NET MVC Core 3.1 Web App - Deel 8

1 Details van menu-item vanaf Home pagina

Wanneer er op de homepagina wordt geklikt op een “Details” button naast een menu-item, moet er een Details-pagina worden

getoond:



https://pittig20200419090403.azurewebsites.net/Customer/Home/Details/3

Details Artikel

Naam	Avocado-gin
Beschrijving	Pittige avocadodrank met gin
Prijs (EUR)	7.5
Categorie	Drank
Sub categorie	Gin
Pittigheid	Sterk
Count	1

Hiervoor gaan we een Details action methode en razor page toevoegen:

1.1 Toevoegen van Details Action methode aan HomeController

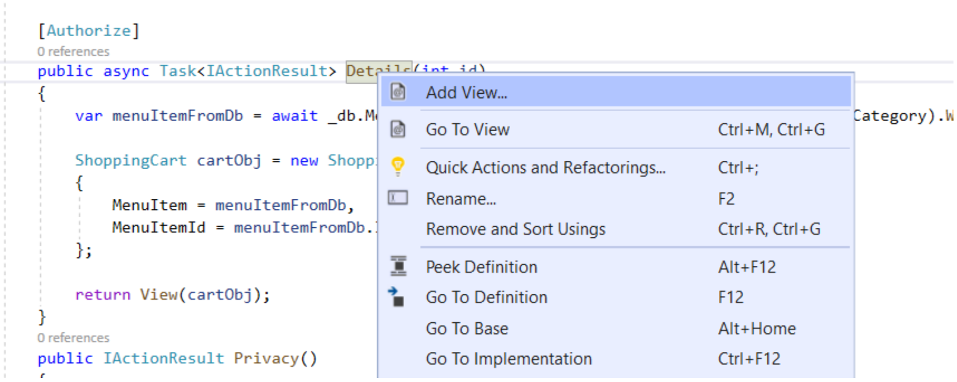
Open Areas/Customer/Home/HomeController.cs en voeg de action method Details (HttpGet) toe (enkel voor ingelogde gebruikers)

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Logging;
using PittigRestoMVC.Data;
using PittigRestoMVC.Models;
using PittigRestoMVC.Models.ViewModels;

namespace PittigRestoMVC.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);
        }

        [Authorize]
        public async Task<IActionResult> Details(int id)
        {
            var menuItemFromDb = await _db.MenuItem.Include(m =>
m.Category).Include(m => m.SubCategory).Where(m => m.Id ==
id).FirstOrDefaultAsync();

            ShoppingCart cartObj = new ShoppingCart()
            {
                MenuItem = menuItemFromDb,
                MenuItemId = menuItemFromDb.Id
            };

            return View(cartObj);
        }
        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 });
        }
    }
}
```

1.2 Toevoegen van Details Razor View

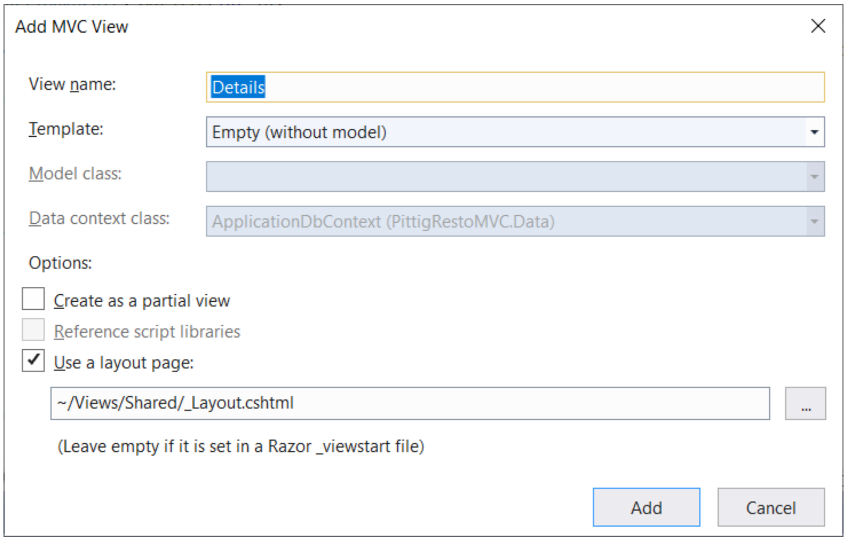
We gaan de Razor View toevoegen vanaf de action methode:

Rechtsklik met de muis op de methode Details en kies “Add View”:



```
[Authorize]
0 references
public async Task<IActionResult> Details(int id)
{
    var menuItemFromDb = await _db.M                                    Category).W

    ShoppingCart cartObj = new Shopp
    {
        MenuItem = menuItemFromDb,
        MenuItemId = menuItemFromDb.I
    };

    return View(cartObj);
}
0 references
public IActionResult Privacy()
```

Add View...
Go To View Ctrl+M, Ctrl+G
Quick Actions and Refactorings... Ctrl+;
Rename... F2
Remove and Sort Usings Ctrl+R, Ctrl+G
Peek Definition Alt+F12
Go To Definition F12
Go To Base Alt+Home
Go To Implementation Ctrl+F12



Add MVC View

View name: Details

Template: Empty (without model)

Model class:

Data context class: ApplicationDbContext (PittigRestoMVC.Data)

Options:

☐ Create as a partial view

☐ Reference script libraries

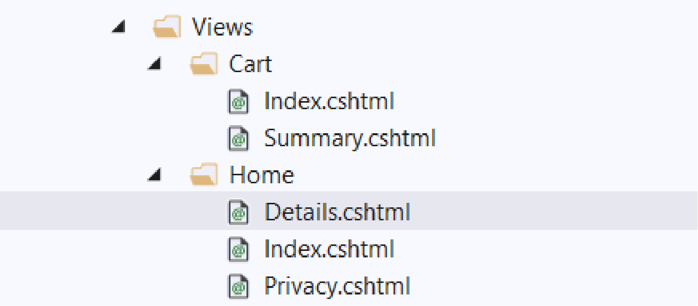
☑ Use a layout page:

~/Views/Shared/_Layout.cshtml

(Leave empty if it is set in a Razor _viewstart file)

Klik op de button “Add”

De Details view wordt aangemaakt:



```
◢ Views
    ◢ Cart
        Index.cshtml
        Summary.cshtml
    ◢ Home
        Details.cshtml
        Index.cshtml
        Privacy.cshtml
```

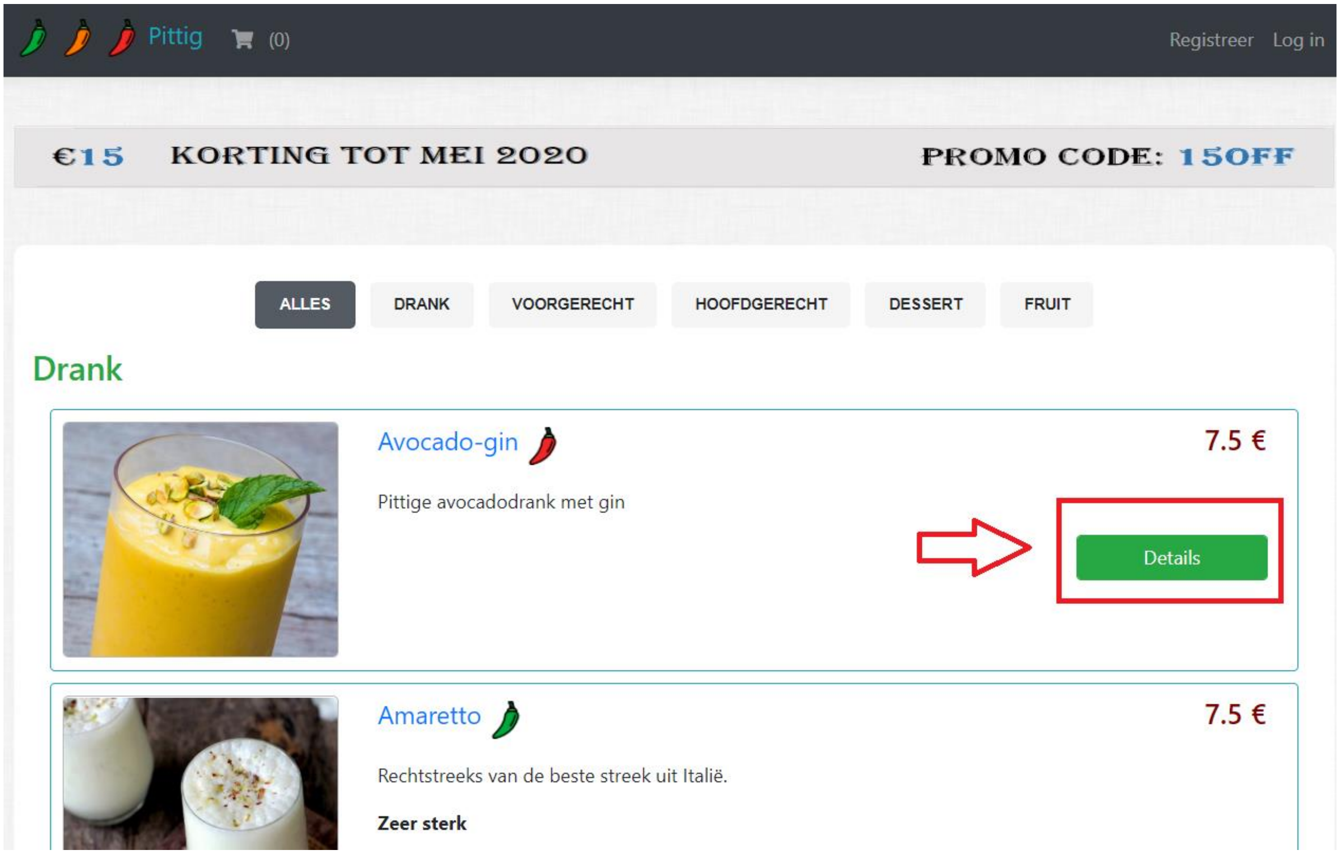
Pas de inhoud aan van de Details.cshtml Razor View:

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

<br />
<h2 class="text-info">Details Artikel</h2>
<br />

<form method="post">
    <div class="border backgroundWhite row">

        <div asp-validation-summary="ModelOnly" class="text-danger"></div>
        <div class="col-12 pb-4 d-md-none">
            <img src="@Model.MenuItem.Image" width="100%" style="border-radius:5px;border: 1px
solid #bbb9b9;" />
        </div>
        <div class="col-md-8">
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.Name" class="text-info font-weight-bold"></label>
                </div>
                <div class="col-md-8">
                    <input asp-for="MenuItem.Name" readonly class="form-control" />
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.Description" class="text-info font-weight-
bold"></label>
                </div>
                <div class="col-md-8">
                    <div class="border p-2">
                        @Html.Raw(Model.MenuItem.Description)
                    </div>
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.Price" class="text-info font-weight-bold"></label>
                </div>
                <div class="col-md-8">
                    <input asp-for="MenuItem.Price" readonly class="form-control" />
                </div>

            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.CategoryId" class="text-info font-weight-
bold"></label>
```

```
                </div>
                <div class="col-md-8">
                    <input asp-for="MenuItem.Category.Name" readonly class="form-control" />
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.SubCategoryId" class="text-info font-weight-
bold"></label>
                </div>
                <div class="col-md-8">
                    <input asp-for="MenuItem.SubCategory.Name" readonly class="form-control" />
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="MenuItem.Spicyness" class="text-info font-weight-
bold"></label>
                </div>
                <div class="col-md-8">
                    <select asp-for="MenuItem.Spicyness" disabled asp-
items="Html.GetEnumSelectList<MenuItem.ESpicy>()" class="form-control"></select>
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                    <label asp-for="Count" class="text-info font-weight-bold"></label>
                </div>
                <div class="col-md-8">
                    <input asp-for="Count"  class="form-control" />
                    <span asp-validation-for="Count" class="text-danger"></span>
                </div>
            </div>
            <div class="form-group row">
                <div class="col-md-4">
                </div>
                <div class="col-md-4">
                    <a asp-action="Index" class="btn btn-success form-control">Terug naar
lijst</a>
                </div>
                <div class="col-md-4">
                    <button type="submit" value="Add to Cart" class="btn btn-primary form-
control">In winkelwagen</button>
                </div>
            </div>
        </div>
        <div class="col-md-3 offset-1 d-none d-md-block">
            <img src="@Model.MenuItem.Image" width="100%" style="border-radius:5px;border: 1px
solid #bbb9b9;" />
        </div>



    </div>
    <input asp-for="MenuItemId" hidden />
</form>

@section Scripts{


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

1.3 Test de Button navigatie op de button details op de home pagina:

Klik op een Details button naast een menu-item op de homepagina



€15 KORTING TOT MEI 2020 PROMO CODE: 15OFF

Drank

Avocado-gin 7.5 €

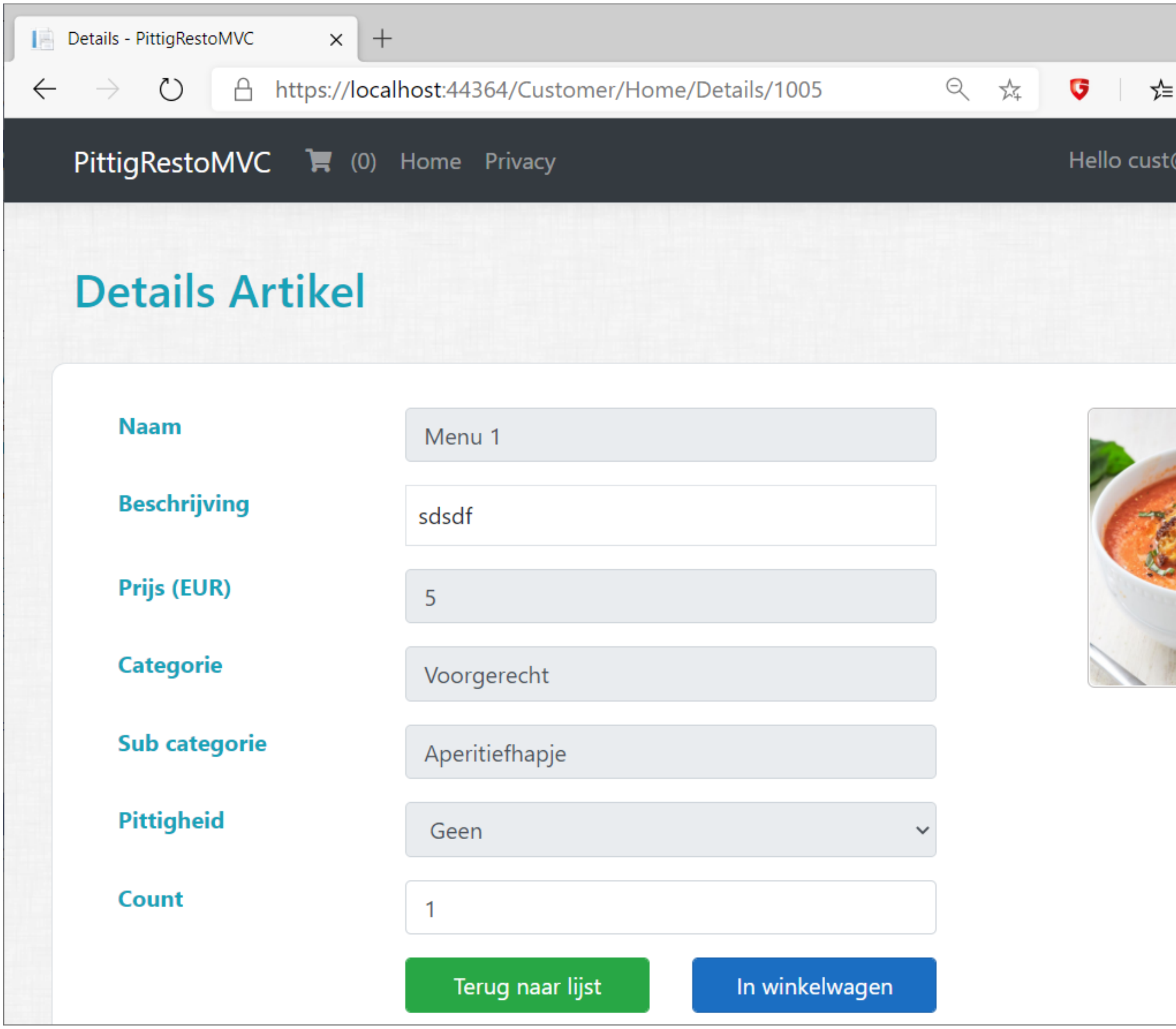
Pittige avocadodrank met gin

Amaretto 7.5 €

Rechtstreeks van de beste streek uit Italië.

Zeer sterk

A. Wanneer een gebruiker is ingelogd, wordt de details Razor View getoond:

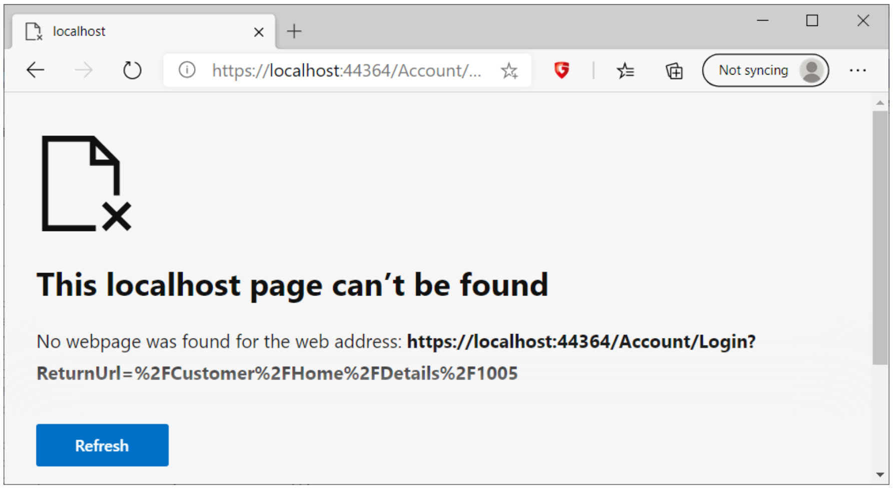


Details Artikel

Naam	Menu 1
Beschrijving	sdsdf
Prijs (EUR)	5
Categorie	Voorgerecht
Sub categorie	Aperitiefhapje
Pittigheid	Geen
Count	1

B. Wanneer een gebruiker nog niet is ingelogd, moet deze automatisch worden doorgestuurd naar de login-pagina

Maar we krijgen deze foutmelding:



This localhost page can’t be found

No webpage was found for the web address: **https://localhost:44364/Account/Login?ReturnUrl=%2FCustomer%2FHome%2FDetails%2F1005**

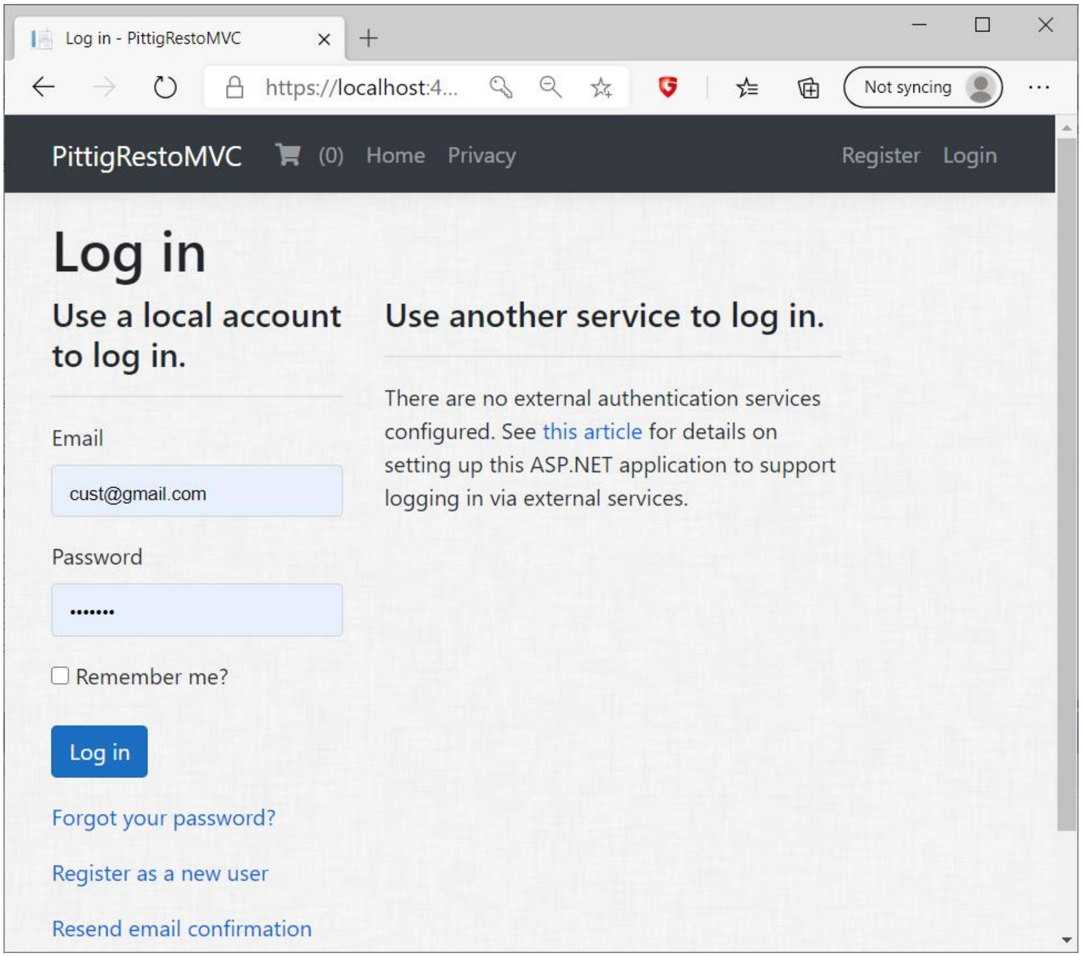
Refresh

Oplossing: Open startup.cs en voeg de volgende lijn toe in de methode ConfigureServices:

```
       public void ConfigureServices(IServiceCollection services)
        {
            services.AddDbContext<ApplicationDbContext>(options =>
                options.UseSqlServer(
                    Configuration.GetConnectionString("DefaultConnection")));
            services.AddIdentity<IdentityUser,IdentityRole>()
                .AddDefaultUI()
                .AddDefaultTokenProviders()
                .AddEntityFrameworkStores<ApplicationDbContext>();// options =>
options.SignIn.RequireConfirmedAccount = true

            services.AddSingleton<IEmailSender, EmailSender>();
            services.Configure<EmailOptions>(Configuration);
            services.AddScoped<IDbInitializer, DbInitializer>();
            services.AddControllersWithViews();
            services.AddRazorPages().AddRazorRuntimeCompilation();

        }
```

Test opnieuw door op de details button te klikken bij een niet-ingelogde gebruiker

De login pagina wordt nu getoond:



Log in

Use a local account to log in.

Email

cust@gmail.com

Password

•••••••

☐ Remember me?

Forgot your password?

Register as a new user

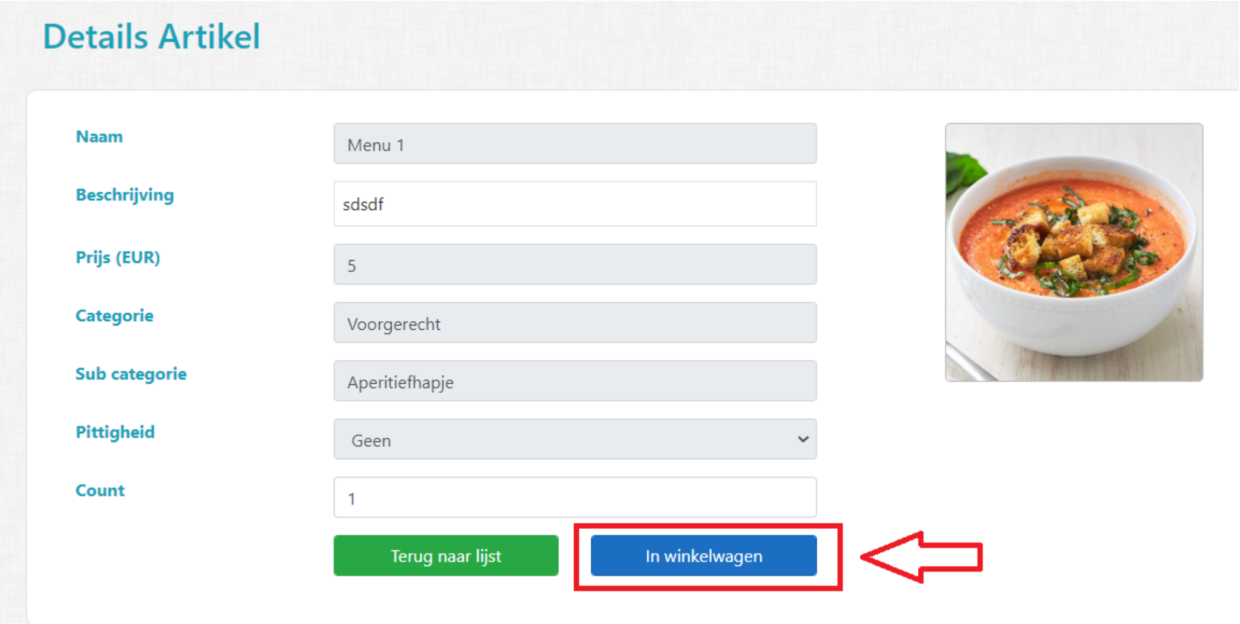
Resend email confirmation

Use another service to log in.

There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

2 Toevoegen van Menu-item aan winkelwagen

Wanneer een ingelogde gebruiker klikt op de button “In Winkelwagen” op de Details Razor view, gaan we een menu-item toevoegen aan de winkelwagen van de gebruiker:



Details Artikel

Naam: Menu 1

Beschrijving: sdsdf

Prijs (EUR): 5

Categorie: Voorgerecht

Sub categorie: Aperitiefhapje

Pittigheid: Geen

Count: 1

2.1 Toevoegen van HttpPost action method voor toevoegen van Menu-Item aan winkelwagen

Wanneer op de knop “in winkelwagen” wordt geklikt, wordt er een httpPost request gestuurd om het menu-item toe te voegen aan de winkelwagen.

Het aantal items in de winkelwagen wordt steeds getoond in de navigatie bar in elke webpagina.

Aangezien we niet bij elke pagina-navigatie opnieuw het aantal items in de winkelwagen uit de database willen opvragen, gaan we dit bijhouden in een session-variabele. (Deze session variable zal het aantal items in de winkelwagen tijdelijk bijgehouden totdat de browser-sessie of tabblad in browser wordt gesloten)

Open Areas/Customer/Controllers/HomeController.cs en voeg een HttpPost Action Method Details toe:

```
    [Authorize]
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> Details(ShoppingCart CartObject)
    {
        CartObject.Id = 0;
        if(ModelState.IsValid)
        {
            var claimsIdentity = (ClaimsIdentity)this.User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
            CartObject.ApplicationUserId = claim.Value;

            ShoppingCart cartFromDb = await _db.ShoppingCart.Where(c =>
c.ApplicationUserId == CartObject.ApplicationUserId
```

```
                                                && c.MenuItemId ==
CartObject.MenuItemId).FirstOrDefaultAsync();

                if(cartFromDb==null)
                {
                    await _db.ShoppingCart.AddAsync(CartObject);
                }
                else
                {
                    cartFromDb.Count = cartFromDb.Count + CartObject.Count;
                }
                await _db.SaveChangesAsync();

                var count = _db.ShoppingCart.Where(c => c.ApplicationUserId ==
CartObject.ApplicationUserId).ToList().Count();
                HttpContext.Session.SetInt32(SD.ssShoppingCartCount, count);

                return RedirectToAction("Index");
            }
            else
            {

                var menuItemFromDb = await _db.MenuItem.Include(m =>
m.Category).Include(m => m.SubCategory).Where(m => m.Id ==
CartObject.MenuItemId).FirstOrDefaultAsync();

                ShoppingCart cartObj = new ShoppingCart()
                {
                    MenuItem = menuItemFromDb,
                    MenuItemId = menuItemFromDb.Id
                };

                return View(cartObj);
            }
        }
```

2.2 Configuratie van Session variabelen in Startup.cs

Session variabelen zijn niet standaard ter beschikking in ASP.NET Core. Deze moeten worden geconfigureerd in startup.cs:

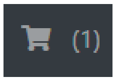
1. **Open Startup.cs en voeg de volgende lijnen code toe aan de methode ConfigureServices:**

```
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddDbContext<ApplicationDbContext>(options =>
                options.UseSqlServer(
                    Configuration.GetConnectionString("DefaultConnection")));
            services.AddIdentity<IdentityUser,IdentityRole>()
                .AddDefaultUI()
                .AddDefaultTokenProviders()
                .AddEntityFrameworkStores<ApplicationDbContext>();// options =>
options.SignIn.RequireConfirmedAccount = true

            services.AddSingleton<IEmailSender, EmailSender>();
            services.Configure<EmailOptions>(Configuration);
            services.AddScoped<IDbInitializer, DbInitializer>();
            services.AddControllersWithViews();
            services.AddRazorPages().AddRazorRuntimeCompilation();


            services.AddSession(options =>
            {
                options.Cookie.IsEssential = true;
                options.IdleTimeout = TimeSpan.FromMinutes(30);
                options.Cookie.HttpOnly = true;
            });

        }
```

2. **Open de volgende lijnen code toe aan de methode Configure:**

```
app.UseCookiePolicy();

app.UseSession();
```

2.3 Aanpassen van navigatie-bar tonen van aantal items in winkelwagen


(1)

Open de Razor view Views/Shared/_Layout.cshtml en voeg de volgende code toe:

Bovenaan:

```
@using PittigRestoMVC.Utility
@using Microsoft.AspNetCore.Http
@inject IHttpContextAccessor HttpContextAccessor
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - PittigRestoMVC</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU"
crossorigin="anonymous">
    <link rel="stylesheet" href="~/css/site.css" />
</head>
<body class="search-background">
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark
border-bottom box-shadow mb-3">
            <div class="container">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">PittigRestoMVC</a>
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-
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 flex-sm-row-
reverse">
                    <partial name="_LoginPartial" />
                    <ul class="navbar-nav flex-grow-1">
                        @if (User.IsInRole(SD.ManagerUser))
                        {
                            <li class="nav-item dropdown text-white-50">
                                <a class="nav-link dropdown-toggle" href="#"
id="navbarDropDownMenuLink" role="button"
                                   data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                                    Inhoud beheren
                                </a>
                                <div class="dropdown-menu" aria-
labelledby="navbarDropDownMenuLink">
                                    <a class="dropdown-item" asp-action="Index" asp-
controller="Category" asp-area="Admin">categorie</a>
                                    <a class="dropdown-item" asp-action="Index" asp-
controller="SubCategory" asp-area="Admin">subcategorie</a>
                                    <a class="dropdown-item" asp-action="Index" asp-
controller="MenuItem" asp-area="Admin">Menu</a>
                                    <a class="dropdown-item" asp-action="Index" asp-
controller="Coupon" asp-area="Admin">Kortingen</a>
                                    <a class="dropdown-item" asp-action="Index" asp-
controller="User" asp-area="Admin">Personeel</a>

                                </div>
                            </li>
                        }

                        @if
(HttpContextAccessor.HttpContext.Session.GetInt32(@SD.ssShoppingCartCount) != null)
                        {
                            <li style="color:white">
                                <a asp-area="Customer" asp-controller="Cart" asp-
action="Index" class="nav-link">
                                    @{
                                        var count =
HttpContextAccessor.HttpContext.Session.GetInt32(@SD.ssShoppingCartCount);
                                    }
                                    <i class="fas fa-shopping-cart"></i> &nbsp; (@count)
                                </a>
                            </li>
                        }
                        else
                        {
                            <li style="color:white">
                                <a href="#" class="nav-link">
                                    <i class="fas fa-shopping-cart"></i> &nbsp; (0)
                                </a>
                            </li>
                        }


                        @*<li class="nav-item">
                            <a class="nav-link" asp-area="" asp-controller="Home" asp-
action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" 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 pl-3 footer text-white-50" style="background-
color:#343a40">
        <div class="container">
            &copy; 2020 - PittigRestoMVC - <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>
    @RenderSection("Scripts", required: false)
</body>
</html>
```

2.4 Ophalen van aantal items in winkelwagen bij terug inloggen

Wanneer de gebruiker items toevoegt aan zijn winkelwagen, wordt dit bijgehouden in de database.

Wanneer de gebruiker terug inlogt, halen we het aantal items op uit de winkelwagen terug op en plaatsen dit in een session variabele

Open Areas/Identity/Pages/Account/ Login.cshtml.cs en voeg de volgende code toe:

```
namespace PittigRestoMVC.Areas.Identity.Pages.Account
{
    [AllowAnonymous]
    public class LoginModel : PageModel
    {
        private readonly UserManager<IdentityUser> _userManager;
        private readonly SignInManager<IdentityUser> _signInManager;
        private readonly ILogger<LoginModel> _logger;
        private readonly ApplicationDbContext _db;
        public LoginModel(SignInManager<IdentityUser> signInManager,
ILogger<LoginModel> logger, ApplicationDbContext db)
        {
            _signInManager = signInManager;
            _logger = logger;
            _db = db;
        }

        [BindProperty]
        public InputModel Input { get; set; }
        public IList<AuthenticationScheme> ExternalLogins { get; set; }

        public string ReturnUrl { get; set; }

        [TempData]
        public string ErrorMessage { get; set; }

        public class InputModel
        {
            [Required]
            [EmailAddress]
            public string Email { get; set; }

            [Required]
            [DataType(DataType.Password)]
            public string Password { get; set; }

            [Display(Name = "Remember me?")]
            public bool RememberMe { get; set; }
        }

        public async Task OnGetAsync(string returnUrl = null)
        {
            if (!string.IsNullOrEmpty(ErrorMessage))
            {
                ModelState.AddModelError(string.Empty, ErrorMessage);
            }

            returnUrl = returnUrl ?? Url.Content("~/");

            // Clear the existing external cookie to ensure a clean login process
            await HttpContext.SignOutAsync(IdentityConstants.ExternalScheme);

            ExternalLogins = (await 
_signInManager.GetExternalAuthenticationSchemesAsync()).ToList();

            ReturnUrl = returnUrl;
        }

        public async Task<IActionResult> OnPostAsync(string returnUrl = null)
        {
            returnUrl = returnUrl ?? Url.Content("~/");

            if (ModelState.IsValid)
            {
                // This doesn't count login failures towards account lockout
                // To enable password failures to trigger account lockout, set 
lockoutOnFailure: true
                var result = await _signInManager.PasswordSignInAsync(Input.Email, 
Input.Password, Input.RememberMe, lockoutOnFailure: false);
                if (result.Succeeded)
                {
                    var user = await _db.Users.Where(u => u.Email == 
Input.Email).FirstOrDefaultAsync();
                    List<ShoppingCart> lstShoppingCart = await 
_db.ShoppingCart.Where(u => u.ApplicationUserId == user.Id).ToListAsync();
```

```
                    HttpContext.Session.SetInt32(SD.ssShoppingCartCount, 
lstShoppingCart.Count);
                    _logger.LogInformation("User logged in.");
                    return LocalRedirect(returnUrl);
                }

                if (result.RequiresTwoFactor)
                {
                    return RedirectToPage("./LoginWith2fa", new { ReturnUrl = 
returnUrl, RememberMe = Input.RememberMe });
                }
                if (result.IsLockedOut)
                {
                    _logger.LogWarning("User account locked out.");
                    return RedirectToPage("./Lockout");
                }
                else
                {
                    ModelState.AddModelError(string.Empty, "Invalid login 
attempt.");
                    return Page();
                }
            }

            // If we got this far, something failed, redisplay form
            return Page();
        }
    }
}
```

2.5 Session variabele (Aantal items in winkelwagen) op 0 zetten bij het uitloggen

Open Areas/Identity/Pages/Account/ Logout.cshtml.cs en voeg de volgende code toe aan de methode OnPost:

```
public async Task<IActionResult> OnPost(string returnUrl = null)
{
    await _signInManager.SignOutAsync();
    _logger.LogInformation("User logged out.");
    HttpContext.Session.SetInt32(SD.ssShoppingCartCount, 0);

    if (returnUrl != null)
    {
        return LocalRedirect(returnUrl);
    }
    else
    {
        return RedirectToPage();
    }
}
```

2.6 Ophalen van aantal Items in winkelwagen bij het tonen van de HomePagina

Wanneer de gebruiker via een cookie terugkomt, is hij automatisch ingelogd.

Bij het aankomen op de homepagina, vragen we het aantal items in de winkelwagen op.

Open Areas/Customer/Controllers/HomeController en voeg de volgende lijnen toe aan de methode Index:

```
        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()
            };

            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

            if (claim != null)
            {
                var cnt = _db.ShoppingCart.Where(u => u.ApplicationUserId ==
claim.Value).ToList().Count;
                HttpContext.Session.SetInt32(SD.ssShoppingCartCount, cnt);
            }
            return View(IndexVM);
        }
```

3 Lijst van items in Winkelwagen

3.1 Viewmodel en Razor View toevoegen voor het overzicht van de items in Winkelwagen

```
namespace Pittig.Models.ViewModels
{
    public class OrderDetailsCart
    {
        public List<ShoppingCart> listCart { get; set; }
        public OrderHeader OrderHeader { get; set; }
    }
}
```

3.2 CartController toevoegen

Voeg een nieuwe Controller met naam CartController toe onder de folder Areas/Customer/Controllers

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Identity.UI.Services;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using PittigRestoMVC.Data;
using PittigRestoMVC.Models;
using PittigRestoMVC.Models.ViewModels;
using PittigRestoMVC.Utility;
using Stripe;

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

        [BindProperty]
        public OrderDetailsCart detailCart { get; set; }

        public CartController(ApplicationDbContext db, IEmailSender emailSender)
        {
            _db = db;
            _emailSender = emailSender;
        }

        public async Task<IActionResult> Index()
        {

            detailCart = new OrderDetailsCart()
            {
                OrderHeader = new PittigRestoMVC.Models.OrderHeader()
            };

            detailCart.OrderHeader.OrderTotal = 0;

            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

            var cart = _db.ShoppingCart.Where(c => c.ApplicationUserId == claim.Value);
            if (cart != null)
            {
                detailCart.listCart = cart.ToList();
            }

            foreach (var list in detailCart.listCart)
            {
                list.MenuItem = await _db.MenuItem.FirstOrDefaultAsync(m => m.Id ==
list.MenuItemId);
                detailCart.OrderHeader.OrderTotal = detailCart.OrderHeader.OrderTotal +
(list.MenuItem.Price * list.Count);
                list.MenuItem.Description =
SD.ConvertToRawHtml(list.MenuItem.Description);
                if (list.MenuItem.Description.Length > 100)
                {
                    list.MenuItem.Description = list.MenuItem.Description.Substring(0,
99) + "...";
                }
            }
            detailCart.OrderHeader.OrderTotalOriginal =
detailCart.OrderHeader.OrderTotal;

            if (HttpContext.Session.GetString(SD.ssCouponCode) != null)
            {
                detailCart.OrderHeader.CouponCode =
HttpContext.Session.GetString(SD.ssCouponCode);
                var couponFromDb = await _db.Coupon.Where(c => c.Name.ToLower() ==
detailCart.OrderHeader.CouponCode.ToLower()).FirstOrDefaultAsync();
```

```
                detailCart.OrderHeader.OrderTotal = SD.DiscountedPrice(couponFromDb, 
detailCart.OrderHeader.OrderTotalOriginal);
            }


            return View(detailCart);

        }


        public async Task<IActionResult> Summary()
        {

            detailCart = new OrderDetailsCart()
            {
                OrderHeader = new PittigRestoMVC.Models.OrderHeader()
            };

            detailCart.OrderHeader.OrderTotal = 0;

            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);
            ApplicationUser applicationUser = await _db.ApplicationUser.Where(c => c.Id 
== claim.Value).FirstOrDefaultAsync();
            var cart = _db.ShoppingCart.Where(c => c.ApplicationUserId == claim.Value);
            if (cart != null)
            {
                detailCart.listCart = cart.ToList();
            }

            foreach (var list in detailCart.listCart)
            {
                list.MenuItem = await _db.MenuItem.FirstOrDefaultAsync(m => m.Id == 
list.MenuItemId);
                detailCart.OrderHeader.OrderTotal = detailCart.OrderHeader.OrderTotal + 
(list.MenuItem.Price * list.Count);

            }
            detailCart.OrderHeader.OrderTotalOriginal = 
detailCart.OrderHeader.OrderTotal;
            detailCart.OrderHeader.PickupName = applicationUser.Name;
            detailCart.OrderHeader.PhoneNumber = applicationUser.PhoneNumber;
            detailCart.OrderHeader.PickUpTime = DateTime.Now;


            if (HttpContext.Session.GetString(SD.ssCouponCode) != null)
            {
                detailCart.OrderHeader.CouponCode = 
HttpContext.Session.GetString(SD.ssCouponCode);
                var couponFromDb = await _db.Coupon.Where(c => c.Name.ToLower() == 
detailCart.OrderHeader.CouponCode.ToLower()).FirstOrDefaultAsync();
                detailCart.OrderHeader.OrderTotal = SD.DiscountedPrice(couponFromDb, 
detailCart.OrderHeader.OrderTotalOriginal);
            }


            return View(detailCart);
        }


        [HttpPost]
        [ValidateAntiForgeryToken]
        [ActionName("Summary")]
        public async Task<IActionResult> SummaryPost(string stripeToken)
        {
            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);


            detailCart.listCart = await _db.ShoppingCart.Where(c => c.ApplicationUserId
== claim.Value).ToListAsync();

            detailCart.OrderHeader.PaymentStatus = SD.PaymentStatusPending;
            detailCart.OrderHeader.OrderDate = DateTime.Now;
            detailCart.OrderHeader.UserId = claim.Value;
            detailCart.OrderHeader.Status = SD.PaymentStatusPending;
            detailCart.OrderHeader.PickUpTime =
Convert.ToDateTime(detailCart.OrderHeader.PickUpDate.ToShortDateString() + " " +
detailCart.OrderHeader.PickUpTime.ToShortTimeString());

            List<OrderDetails> orderDetailsList = new List<OrderDetails>();
            _db.OrderHeader.Add(detailCart.OrderHeader);
            await _db.SaveChangesAsync();

            detailCart.OrderHeader.OrderTotalOriginal = 0;


            foreach (var item in detailCart.listCart)
            {
                item.MenuItem = await _db.MenuItem.FirstOrDefaultAsync(m => m.Id ==
item.MenuItemId);
                OrderDetails orderDetails = new OrderDetails
                {
                    MenuItemId = item.MenuItemId,
                    OrderId = detailCart.OrderHeader.Id,
                    Description = item.MenuItem.Description,
                    Name = item.MenuItem.Name,
                    Price = item.MenuItem.Price,
                    Count = item.Count
                };
                detailCart.OrderHeader.OrderTotalOriginal += orderDetails.Count *
orderDetails.Price;
                _db.OrderDetails.Add(orderDetails);

            }

            if (HttpContext.Session.GetString(SD.ssCouponCode) != null)
            {
                detailCart.OrderHeader.CouponCode =
HttpContext.Session.GetString(SD.ssCouponCode);
                var couponFromDb = await _db.Coupon.Where(c => c.Name.ToLower() ==
detailCart.OrderHeader.CouponCode.ToLower()).FirstOrDefaultAsync();
                detailCart.OrderHeader.OrderTotal = SD.DiscountedPrice(couponFromDb,
detailCart.OrderHeader.OrderTotalOriginal);
            }
            else
            {
                detailCart.OrderHeader.OrderTotal =
detailCart.OrderHeader.OrderTotalOriginal;
            }
            detailCart.OrderHeader.CouponCodeDiscount =
detailCart.OrderHeader.OrderTotalOriginal - detailCart.OrderHeader.OrderTotal;

            _db.ShoppingCart.RemoveRange(detailCart.listCart);
            HttpContext.Session.SetInt32(SD.ssShoppingCartCount, 0);
            await _db.SaveChangesAsync();

            var options = new ChargeCreateOptions
            {
                Amount = Convert.ToInt32(detailCart.OrderHeader.OrderTotal * 100),
                Currency = "usd",
                Description = "Order ID : " + detailCart.OrderHeader.Id,
                Source = stripeToken

            };
            var service = new ChargeService();
            Charge charge = service.Create(options);

            if (charge.BalanceTransactionId == null)
            {
                detailCart.OrderHeader.PaymentStatus = SD.PaymentStatusRejected;
            }
            else
            {
                detailCart.OrderHeader.TransactionId = charge.BalanceTransactionId;
            }

            if (charge.Status.ToLower() == "succeeded")
            {
                detailCart.OrderHeader.PaymentStatus = SD.PaymentStatusApproved;
                detailCart.OrderHeader.Status = SD.StatusSubmitted;
            }
            else
            {
                detailCart.OrderHeader.PaymentStatus = SD.PaymentStatusRejected;
            }

            await _db.SaveChangesAsync();
            return RedirectToAction("Index", "Home");

        }


        public IActionResult AddCoupon()
        {
            if (detailCart.OrderHeader.CouponCode == null)
            {
                detailCart.OrderHeader.CouponCode = "";
            }
            HttpContext.Session.SetString(SD.ssCouponCode,
detailCart.OrderHeader.CouponCode);

            return RedirectToAction(nameof(Index));
        }

        public IActionResult RemoveCoupon()
        {

            HttpContext.Session.SetString(SD.ssCouponCode, string.Empty);

            return RedirectToAction(nameof(Index));
        }


        public async Task<IActionResult> Plus(int cartId)
        {
            var cart = await _db.ShoppingCart.FirstOrDefaultAsync(c => c.Id == cartId);
            cart.Count += 1;
            await _db.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }

        public async Task<IActionResult> Minus(int cartId)
        {
            var cart = await _db.ShoppingCart.FirstOrDefaultAsync(c => c.Id == cartId);
            if (cart.Count == 1)
            {
                _db.ShoppingCart.Remove(cart);
                await _db.SaveChangesAsync();

                var cnt = _db.ShoppingCart.Where(u => u.ApplicationUserId ==
cart.ApplicationUserId).ToList().Count;
                HttpContext.Session.SetInt32(SD.ssShoppingCartCount, cnt);
            }
            else
            {
                cart.Count -= 1;
                await _db.SaveChangesAsync();
            }

            return RedirectToAction(nameof(Index));
        }

        public async Task<IActionResult> Remove(int cartId)
        {
            var cart = await _db.ShoppingCart.FirstOrDefaultAsync(c => c.Id == cartId);

            _db.ShoppingCart.Remove(cart);
            await _db.SaveChangesAsync();

            var cnt = _db.ShoppingCart.Where(u => u.ApplicationUserId ==
cart.ApplicationUserId).ToList().Count;
            HttpContext.Session.SetInt32(SD.ssShoppingCartCount, cnt);


            return RedirectToAction(nameof(Index));
        }

    }
}
```

3.3 Razor View toevoegen voor het overzicht van de items in Winkelwagen

Voeg een Razor View “Index.chtml” toe onder de folder Voeg een nieuwe Controller met naam CartController toe onder de folder Areas/Customer/Views/Cart

```
@model PittigRestoMVC.Models.ViewModels.OrderDetailsCart

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

<form method="post">
    @if (Model.listCart.Count > 0)
    {
        <br />
        <input id="userId" asp-for="@Model.listCart[0].ApplicationUserId" hidden />
        <div class="backgroundWhiteBorder">
            <div class="container">
                <div class="card">
                    <div class="card-header bg-dark text-light ml-0 row container">
                        <div class="col-6">
                            <i class="fa fa-shopping-cart"></i> &nbsp;
                            Winkelwagen
                        </div>
                        <div class="col-6 text-right">
                            <a asp-area="Customer" asp-controller="Home" asp-
action="Index" class="btn btn-outline-info btn-sm">Verder winkelen</a>
                        </div>
                    </div>
                    <div class="card-body">
                        @foreach (var item in Model.listCart)
                        {
                            <div class="row">
                                <div class="d-none d-lg-block col-lg-2 text-center py-2">
                                    <img src="@item.MenuItem.Image" class="rounded"
width="120" height="80" />
                                </div>
                                <div class="col-12 text-sm-center col-lg-5 text-lg-left">
                                    <h4><strong>@item.MenuItem.Name</strong></h4>
                                    <h4><small>@item.MenuItem.Description</small></h4>
                                </div>
                                <div class="col-12 text-sm-center col-lg-5 text-lg-right
row">
                                    <div class="col-4 text-md-right" style="padding-
top:5px;">
                                        <h6><strong>€@item.MenuItem.Price <span
class="text-muted">x</span> @item.Count </strong></h6>
                                    </div>
                                    <div class="col-6 col-sm-4 col-lg-6">
                                        <div class="float-right mx-1">
                                            <button type="submit" class="btn btn-primary"
asp-action="plus" asp-route-cartId="@item.Id">
                                                <i class="fas fa-plus"></i>
                                            </button>
                                        </div>
                                        <div class="float-right mx-1">
                                            <button type="submit" class="btn btn-danger"
asp-action="minus" asp-route-cartId="@item.Id">
                                                <i class="fas fa-minus"></i>
                                            </button>
                                        </div>
                                    </div>
                                    <div class="col-2 col-sm-4 col-lg-2 text-right">
                                        <button type="submit" class="btn btn-outline-
danger" asp-action="remove" asp-route-cartId="@item.Id">
                                            <i class="fas fa-trash"></i>
                                        </button>
                                    </div>
                                </div>
                            </div>
                            <hr />
                        }

                        <div class="row">
                            <div class="col-12 col-md-5">
                                <div class="row">
                                    <div class="col-7">
                                        <input asp-for="@Model.OrderHeader.CouponCode"
id="txtCouponCode" class="form-control" placeholder="kortingscode..." />
                                    </div>
                                    <div class="col-5" style="margin-top:2px;">
                                        <button type="submit" class="btn btn-sm form-
control btn-outline-success" id="btnCoupon" asp-action="AddCoupon">
                                            Toevoegen
                                        </button>
                                        <button type="submit" class="btn btn-sm form-
control btn-outline-danger" style="display:none" id="btnRemoveCoupon" asp-
action="RemoveCoupon">
                                            Verwijderen
                                        </button>
                                    </div>
                                </div>
                            </div>
                            <div class="col-12 col-md-6 offset-md-1 col-lg-4 offset-lg-3
pr-4">
                                <ul class="list-group">
                                    <li class="list-group-item d-flex justify-content-
between bg-light">
                                        <span class="text-info"> Totaal (EUR)</span>
                                        <strong class="text-info">€ <span
id="txtOrderTotal">@Model.OrderHeader.OrderTotal</span></strong>
                                    </li>
                                </ul>
                            </div>
                        </div>


                    </div>
                    <div class="card-footer">
                        <div class="col-12 col-lg-4 offset-lg-8 col-md-6 offset-md-6">
                            <a asp-action="Summary" asp-area="Customer" asp-
controller="Cart" class="btn btn-success form-control">Overzicht</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    }
    else
    {
        <div class="backgroundWhiteBorder"> Winkelwagen is leeg...</div>
    }
</form>

@section Scripts{
    <script>
        $(function () {
            var couponCode = document.getElementById("txtCouponCode").value;

            if (couponCode.length > 0) {
                document.getElementById('btnCoupon').style.display = 'none';
                document.getElementById('btnRemoveCoupon').style.display = '';
            }
            else {
                document.getElementById('btnCoupon').style.display = '';
                document.getElementById('btnRemoveCoupon').style.display = 'none';
            }
        });
    </script>
}
```

4 Toevoegen van samenvatting prijs winkelwagen

4.1 Toevoegen van Utility class voor Stripe (plugin voor betaling met Kredietkaart)

Toevoegen van nieuwe class StripeSettings onder Utility folder

```
namespace PittigRestoMVC.Utility
{
    public class StripeSettings
    {
        public string SecretKey { get; set; }
        public string PublishableKey { get; set; }
    }
}
```

4.2 Toevoegen van Stripe settings in appsettings.json (zie vorige les)

```
{
  "ConnectionStrings": {
    "DefaultConnection":
"Server=.;Database=Spice;Trusted_Connection=True;MultipleActiveResultSets=true"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*",
  "Stripe": {
    "SecretKey": "sk_test_kGzUGDqcDrmVHhktqZZUFyI7",
    "PublishableKey": "pk_test_DS6jsf1BPuhpojL8FqWXbBpT"
  },
  "SendGridKey": "SG.IJp0uP22S8-yO9AlgP_C7g.c40-
DFVksleYE42JyEm3RLT1Dd_fmIQdzRixMn8FSzE"
}
```

4.3 Registreren van Stripe als service in Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.Configure<CookiePolicyOptions>(options =>
    {
        // This lambda determines whether user consent for non-essential cookies
is needed for a given request.
        options.CheckConsentNeeded = context => true;
        options.MinimumSameSitePolicy = SameSiteMode.None;
    });

    services.AddDbContext<ApplicationDbContext>(options =>
        options.UseSqlServer(
            Configuration.GetConnectionString("DefaultConnection")));
            services.AddIdentity<IdentityUser, IdentityRole>()
                .AddDefaultTokenProviders()
                .AddEntityFrameworkStores<ApplicationDbContext>();

            services.AddScoped<IDbInitializer, DbInitializer>();
            services.Configure<StripeSettings>(Configuration.GetSection("Stripe"));
            services.AddSingleton<IEmailSender, EmailSender>();
            services.Configure<EmailOptions>(Configuration);

            services.AddControllersWithViews();
            services.AddRazorPages();

            services.AddSession(options =>
            {
                options.Cookie.IsEssential = true;
                options.IdleTimeout = TimeSpan.FromMinutes(30);
                options.Cookie.HttpOnly = true;
            });
        }

        // This method gets called by the runtime. Use this method to configure the HTTP 
request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env, 
IDbInitializer dbInitializer)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
                app.UseDatabaseErrorPage();
            }
            else
            {
                app.UseExceptionHandler("/Home/Error");
                // The default HSTS value is 30 days. You may want to change this for 
production scenarios, see https://aka.ms/aspnetcore-hsts.
                app.UseHsts();
            }
            app.UseRouting();
            StripeConfiguration.ApiKey = Configuration.GetSection("Stripe")["SecretKey"];
            dbInitializer.Initialize();
            app.UseHttpsRedirection();
            app.UseStaticFiles();
            app.UseCookiePolicy();
            app.UseSession();
            app.UseAuthentication();
            app.UseAuthorization();
            app.UseEndpoints(endpoints =>
            {
                endpoints.MapControllerRoute(
                    name: "areas",
                    pattern: "{area=Customer}/{controller=Home}/{action=Index}/{id?}");
                endpoints.MapRazorPages();
            });
        }
    }
```

4.4 Toevoegen van samenvatting winkelwagen Razor View

Maak een nieuwe Razor View Summary.cshtml aan onder de folder Areas/Customer/Views/Cart

```
@model PittigRestoMVC.Models.ViewModels.OrderDetailsCart
@using Microsoft.Extensions.Options

@inject IOptions<pittigrestomvc.utility.stripesettings>
    Stripe

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

    <form method="post">
        @if (Model.listCart.Count > 0)
        {
            <br />
            <input id="userId" asp-for="@Model.listCart[0].ApplicationUserId" hidden />
            <div class="backgroundWhiteBorder">
                <div class="container">
                    <div class="card">
                        <div class="card-header bg-dark text-light ml-0 row container">
                            <div class="col-6">
                                <i class="fa fa-shopping-cart"></i> &nbsp;
                                Overzicht bestelling
                            </div>
                            <div class="col-6 text-right">
                                <a asp-area="Customer" asp-controller="Cart" asp-
action="Index" class="btn btn-outline-info btn-sm">Terug naar winkelwagen</a>
                            </div>
                        </div>
                        <div class="card-body">
                            <div class="container rounded p-2">
                                <div class="row">
                                    <div class="col-12 col-lg-6 pb-4">
                                        <div class="row">
                                            <h4 class="d-flex justify-content-between
align-items-center mb-3">
                                                <span class="text-info">Ophaling:</span>
                                            </h4>
                                        </div>
                                        <div class="row my-1">
                                            <div class="col-3">
                                                <label>Naam</label>
                                            </div>
                                            <div class="col-9">
                                                <input asp-
for="@Model.OrderHeader.PickupName" id="txtName" type="text" class="form-control" />
                                            </div>
                                        </div>
                                        <div class="row my-1">
                                            <div class="col-3">
                                                <label>Telefoon</label>
                                            </div>
                                            <div class="col-9">
```

```
                                                <input asp-
for="@Model.OrderHeader.PhoneNumber" id="txtPhone" type="text" class="form-control" />
                                            </div>
                                        </div>
                                        <div class="row my-1">
                                            <div class="col-3">
                                                <label>Datum</label>
                                            </div>
                                            <div class="col-9">
                                                <input asp-
for="@Model.OrderHeader.PickUpDate" id="datepicker" value="" type="text" class="form-
control" />
                                            </div>
                                        </div>
                                        <div class="row my-1">
                                            <div class="col-3">
                                                <label>Tijdstip</label>
                                            </div>
                                            <div class="col-9">
                                                <input asp-
for="@Model.OrderHeader.PickUpTime" id="timepicker" value="" type="text" class="form-
control" />
                                            </div>
                                        </div>
                                        <div class="row my-1">
                                            <div class="col-3">
                                                <label>Commentaar: </label>
                                            </div>
                                            <div class="col-9">
                                                <textarea asp-
for="@Model.OrderHeader.Comments" style="height:100px;" class="form-control"></textarea>
                                            </div>
                                        </div>
                                    </div>

                                    <div class="col-12 col-lg-5 offset-lg-1">
                                        <h4 class="d-flex justify-content-between align-
items-center mb-3">

                                            <span class="text-info">Overzicht
bestelling:</span>
                                        </h4>
                                        <ul class="list-group mb-3">
                                            @{ var total = 0.0;}

                                            @foreach (var details in Model.listCart)
                                            {
                                                <li class="list-group-item d-flex
justify-content-between">
                                                    <div>
                                                        <h6 class="my-
0">@details.MenuItem.Name</h6>
                                                        <small class="text-
muted">Hoeveelheid: @details.Count</small>
                                                    </div>
                                                    <span class="text-
muted">€@(details.MenuItem.Price * details.Count)</span>
                                                    @{ total += (details.MenuItem.Price *
details.Count); }
```

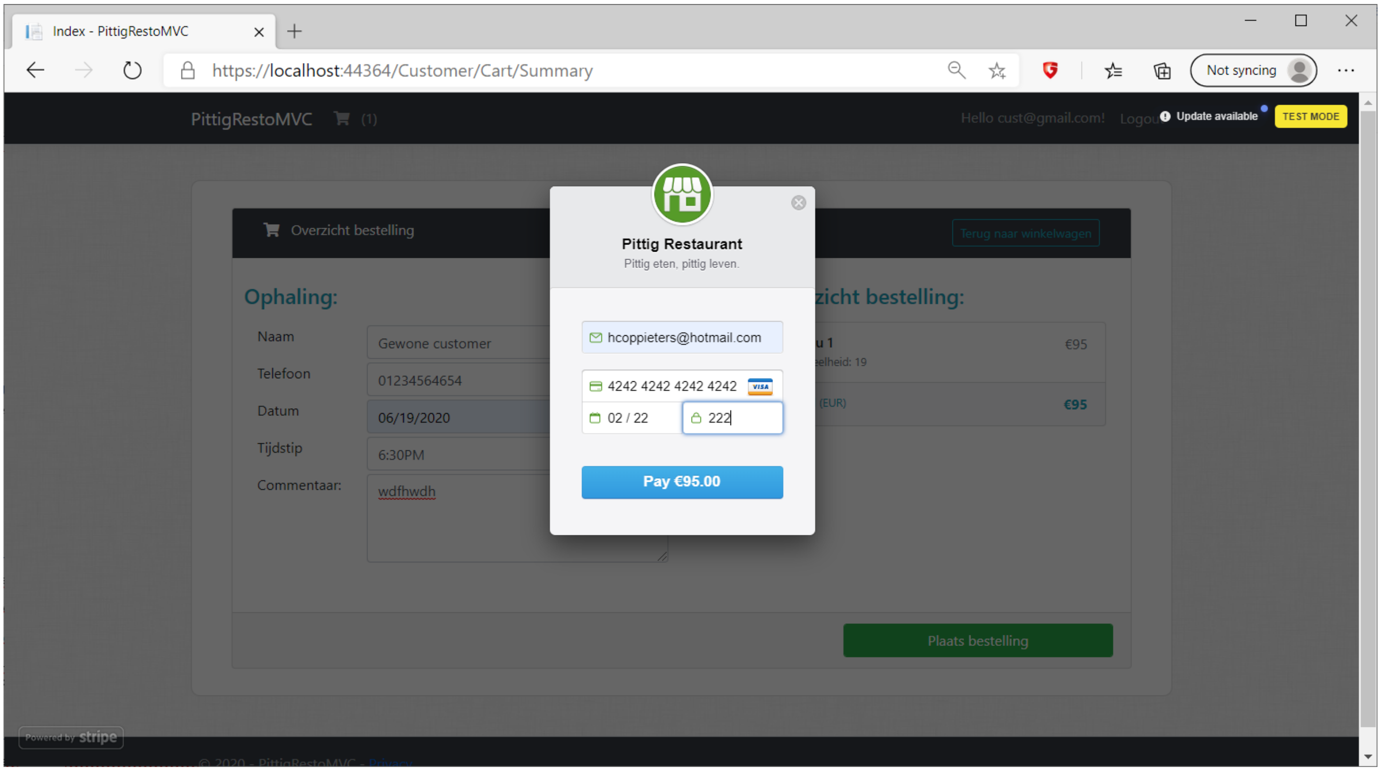
```
                                                </li>
                                            }
                                            @if (Model.OrderHeader.CouponCode != null)
                                            {
                                                <li class="list-group-item d-flex
justify-content-between">
                                                    <div class="text-success">
                                                        <h6 class="my-0">Promotie
Code</h6>
                                                        <small class="text-
success">@Model.OrderHeader.CouponCode</small>
                                                    </div>
                                                    <span class="text-success">-€@(total
- Model.OrderHeader.OrderTotal)</span>
                                                </li>
                                            }
                                            <li class="list-group-item d-flex justify-
content-between bg-light">
                                                <small class="text-info">Totaal
(EUR)</small>
                                                <strong class="text-
info">€@(Model.OrderHeader.OrderTotal)</strong>
                                            </li>
                                        </ul>
                                    </div>
                                </div>
                            </div>
                        </div>
                        <div class="card-footer">
                            <div class="row">
                                <div class="col-12 col-md-4 offset-md-8">

                                    @{
            var OrderTotalForStrip = Model.OrderHeader.OrderTotal * 100;
                                    }
                                    <script src="https://checkout.stripe.com/checkout.js"
class="stripe-button"
                                            data-key="@Stripe.Value.PublishableKey"
                                            data-amount="@OrderTotalForStrip"
                                            data-name="Pittig Restaurant"
                                            data-label="Plaats bestelling"
                                            data-description="Pittig eten, pittig leven."
                                            data-locale="auto"
                                            data-allow-remember-me="false"
                                            data-currency="eur"
                                            data-
image="https://stripe.com/img/documentation/checkout/marketplace.png"></script>
                                    <script>document.getElementsByClassName("stripe-
button-el")[0].style.display = 'none';</script>
                                    <button type="submit" id="btnPlaceOrder"
onclick="return validateInput()" value="Place Order" class="btn btn-success form-
control">Plaats bestelling</button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
```

```
        }
        else
        {
            <div class="backgroundWhiteBorder">Winkelwagen is leeg...</div>
        }
    </form>

    @section Scripts{

        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
timepicker/1.10.0/jquery.timepicker.js"></script>

        <script>
$(function () {

            var minDt = 0;
            if (getTimeFromCurrentDate() >= "20.00") {
                minDt = 1;
            }


            $("#datepicker").datepicker({ minDate: minDt, maxDate: "+1W" });
            $("#timepicker").timepicker({'minTime': '11:00 AM', 'maxTime': '9:00 PM',
step: '30'})
        });




        function setPickUpTime() {
            var dateSelected = document.getElementById("datepicker").value;
            var dt = new Date(dateSelected);
            var today = new Date();

            dt.setHours(0, 0, 0, 0);
            if (dt > today) {
                selectedTime = '11:00am';
            }
            else {
                selectedTime = getTimeFromCurrentDate();
            }

            $('#timepicker').timepicker('option', 'minTime', selectedTime);

        }


        function getTimeFromCurrentDate() {
            var curDt = new Date();

            var minTime = "";
            if (curDt.getHours() < 11) {
                minTime = '12:00pm';
            }
            else {
                if (curDt.getMinutes() < 30) {
                    minTime = (curDt.getHours() + 1).toString() + ':30pm'
                }
                else {
                    minTime = (curDt.getHours() + 2).toString() + ':00pm'
                }
            }
            return minTime;
        }

        $('#datepicker').change(function () {
            setPickUpTime();
        });


        function validateInput() {
            var time = document.getElementById("timepicker").value;
            var dt = document.getElementById("datepicker").value;
            var name = document.getElementById("txtName").value;
            var phone = document.getElementById("txtPhone").value;

            if (dt.toString().length == 0) {
                alert("Please select pickup date")
                return false;
            }
            else if (time.toString().length == 0) {
                alert("Please select pickup time")
                return false;
            }
            else if (name.toString().length == 0) {
                alert("Please enter pickup name")
                return false;
            }
            else if (phone.toString().length == 0) {
                alert("Please enter a phone number")
                return false;
            }
            else {
                return true;
            }
        }</script>
    }
```

4.5 Test de Stripe plugin via een Test- Credit kaartnummer

Hier kan je voorbeelden vinden van Stripe test-krediekaartnummers:

https://stripe.com/docs/testing



Ophaling:

Naam: Gewone customer
Telefoon: 01234564654
Datum: 06/19/2020
Tijdstip: 6:30PM
Commentaar: wdfhwdh

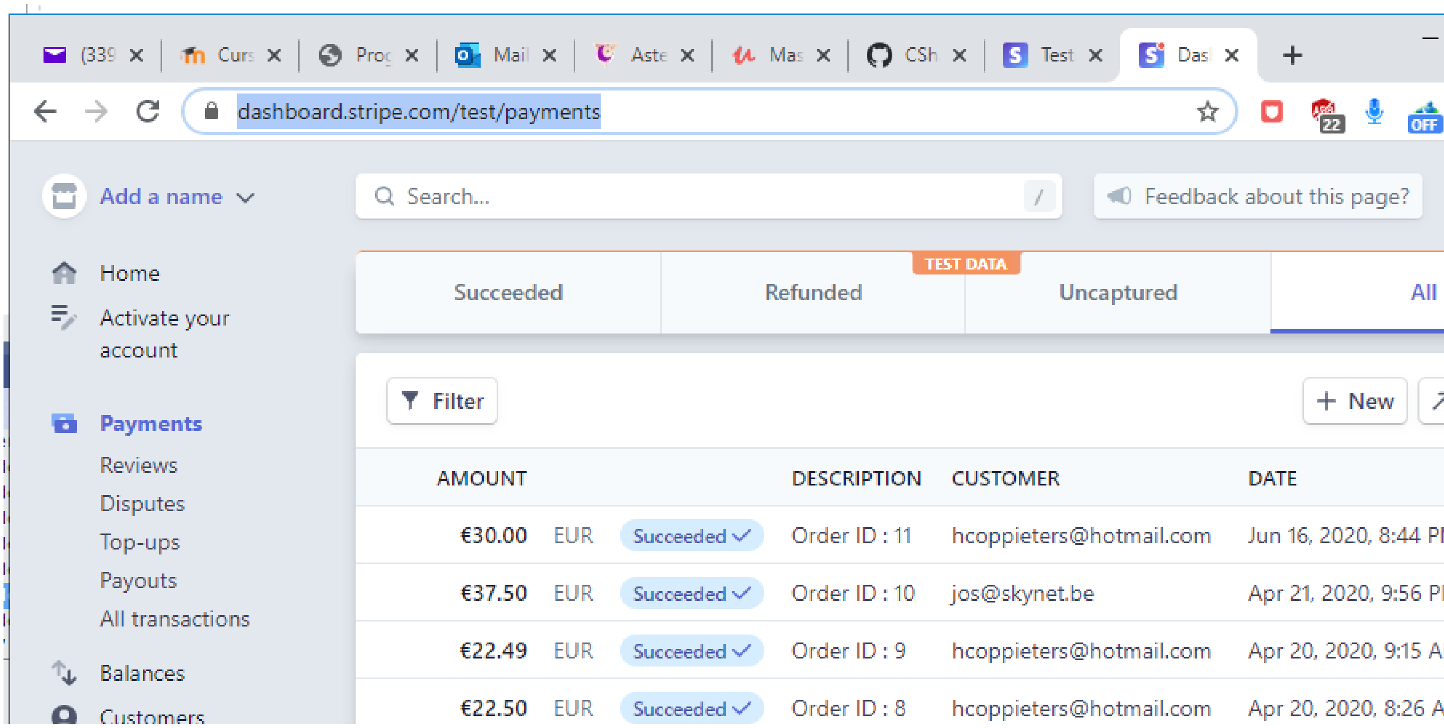
hcoppieters@hotmail.com
4242 4242 4242 4242
02 / 22
222

zicht bestelling:

u 1 — €95
elheid: 19
(EUR) — €95

Op de Stripe online dashboard kan je een overzicht zien van je test-betalingen:

https://dashboard.stripe.com/test/payments



dashboard.stripe.com/test/payments

AMOUNT			DESCRIPTION	CUSTOMER	DATE
€30.00	EUR	Succeeded ✓	Order ID : 11	hcoppieters@hotmail.com	Jun 16, 2020, 8:44 PI
€37.50	EUR	Succeeded ✓	Order ID : 10	jos@skynet.be	Apr 21, 2020, 9:56 PI
€22.49	EUR	Succeeded ✓	Order ID : 9	hcoppieters@hotmail.com	Apr 20, 2020, 9:15 A
€22.50	EUR	Succeeded ✓	Order ID : 8	hcoppieters@hotmail.com	Apr 20, 2020, 8:26 A

5 Bestellingen

5.1 Viewmodel class aanmaken voor OrderDetails

Maak in een folder Models/Viewmodels de class OrderDetailsViewModel aan:

```
namespace PittigRestoMVC.Models.ViewModels
{
    public class OrderDetailsViewModel
    {
        public OrderHeader OrderHeader { get; set; }
        public List<OrderDetails> OrderDetails { get; set; }
    }
}
```

5.2 Viewmodel class aanmaken voor OrderDetails

Maak in een folder Models/Viewmodels de class OrderDetailsViewModel aan:

```
namespace PittigRestoMVC.Models.ViewModels
{
    public class OrderDetailsViewModel
    {
        public OrderHeader OrderHeader { get; set; }
        public List<OrderDetails> OrderDetails { get; set; }
    }
}
```

5.3 Paging Info model aanmaken onder de folder Models (om te gebruiken in orderdetails)

```
public class PagingInfo
```

```
    {
        public int TotalItem { get; set; }

        public int ItemsPerPage { get; set; }

        public int CurrentPage { get; set; }

        public int totalPage => (int)Math.Ceiling((decimal)TotalItem / ItemsPerPage);

        public string urlParam { get; set; }
     }
```

5.4 Viewmodel class aanmaken voor `OrderListViewModel`

Maak in een folder Models/Viewmodels de class `OrderListViewModel` aan:

```
namespace PittigRestoMVC.Models.ViewModels
{
    public class OrderListViewModel
    {
        public IList<OrderDetailsViewModel> Orders { get; set; }
        public PagingInfo PagingInfo { get; set; }
    }
}
```

5.5 Order Controller toevoegen

Maak onder de folder Areas/Customer/Controllers een Lege MVC Controller toe met naam 'OrderController'

OrderController.cs

6 Referenties

https://docs.microsoft.com/en-us/aspnet/core/security/authentication/accconfirm?view=aspnetcore-3.1&tabs=visual-studio

https://fontawesome.com/6?next=%2Fv5.5.0%2Ficons%2Flock-open

http://bekenty.com/send-e-mail-in-asp-net-core-with-sendgrid/

https://hostlaunch.io/docs/how-to-get-a-sendgrid-api-key/

https://www.youtube.com/watch?v=ddSymc0hE0Ahttps://app.sendgrid.com/guide/integrate/langs/csharp

https://www.codeproject.com/Articles/1276970/NET-Core-Razor-Page-Email-Form-using-SendGrid-and

https://sendgrid.com/docs/API_Reference/index.html