

Model ASP.NET MVC Core 3.1 Web App - Deel 9

1 Toevoegen van samenvatting prijs winkelwagen

1.1 Datepicker

Download de laatste versie van jquery-ui om de volgende datepicker te kunnen gebruiken:

<https://jqueryui.com/datepicker/>



jquery-ui.js



jquery-ui.css

1.2 TimePicker

Refereer de timepicker van deze site:

<https://cdnjs.com/libraries/jquery-timepicker>

1.3 Views/Shared/_Layout.cshtml aanpassen

Voeg de volgende lijnen toe aan de _Layout.cshtml om de stylesheet en js libraries te refereren:

```
@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="~/css/jquery-ui.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-timepicker/1.10.0/jquery.timepicker.css" />
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css"
    integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3B1XeVIU"
    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">
                                    @if
                                    {
                                        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/jquery-ui.js"></script>
  <script src="~/js/site.js" asp-append-version="true"></script>
  @RenderSection("Scripts", required: false)
</body>
</html>

```

1.4 Areas/Customter/Views/Cart/Summary.cshtml aanpassen

Voeg voor het gebruik van de datepicker de volgende code toe onderaan de Summary Razor View:

```

@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;
      }
    })
  </script>

```

```

    $('#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>
}

```

2 Betaling via Kredietkaart

Vorige les hebben we in de CartController een HttpGet Summary gemaakt

Deze les geven we de mogelijkheid om een bestelling te plaatsen vanuit de Summary-Cart pagina en te betalen via kredietkaart

Hiervoor gebruiken we de plug-in Stripe (zie vorige 2 lessen).

2.1 SummaryPost action methode toevoegen aan CartController

```

[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();
        //het bedrag moet in euro en in cent (dus vermenigvuldiging met 100)
        var options = new ChargeCreateOptions
        {
            Amount = Convert.ToInt32(detailCart.OrderHeader.OrderTotal * 100),
            Currency = "eur",
            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");
}

```

3 Authenticatie via Sociale media (bv google en facebook)

3.1 Installeer NuGet Packages

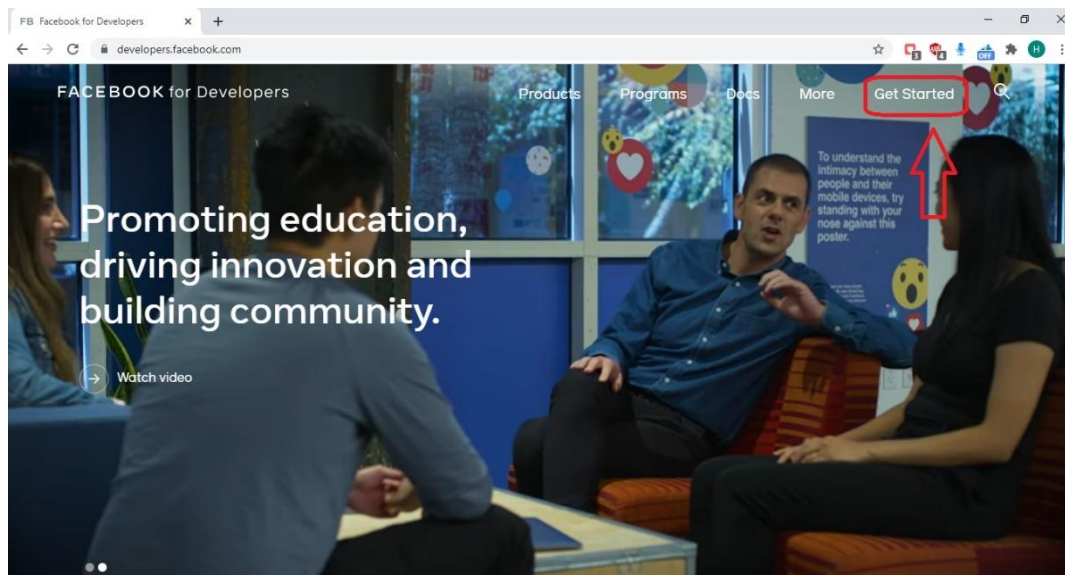
Microsoft.AspNetCore.Authentication.Facebook

Microsoft.AspNetCore.Authentication.Google

3.2 Creatie van Facebook App ID

Ga naar <https://developers.facebook.com/products/facebook-login/>

En klik op de link onder 'Get started':



Log in op Facebook en creëer een nieuw App ID:

Create a New App ID

Get started integrating Facebook into your app or website

Display Name

Pittig

Contact Email

hcoppieters@hotmail.com

This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

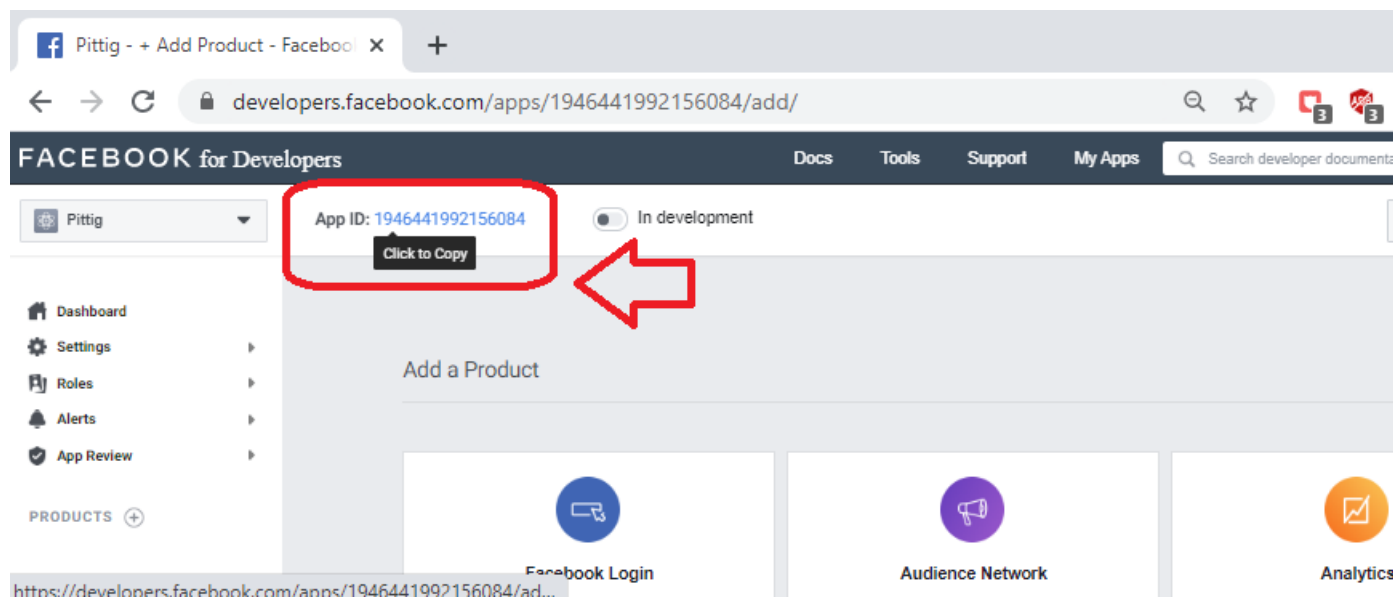
By proceeding, you agree to the [Facebook Platform Policies](#)

Cancel

Create App ID

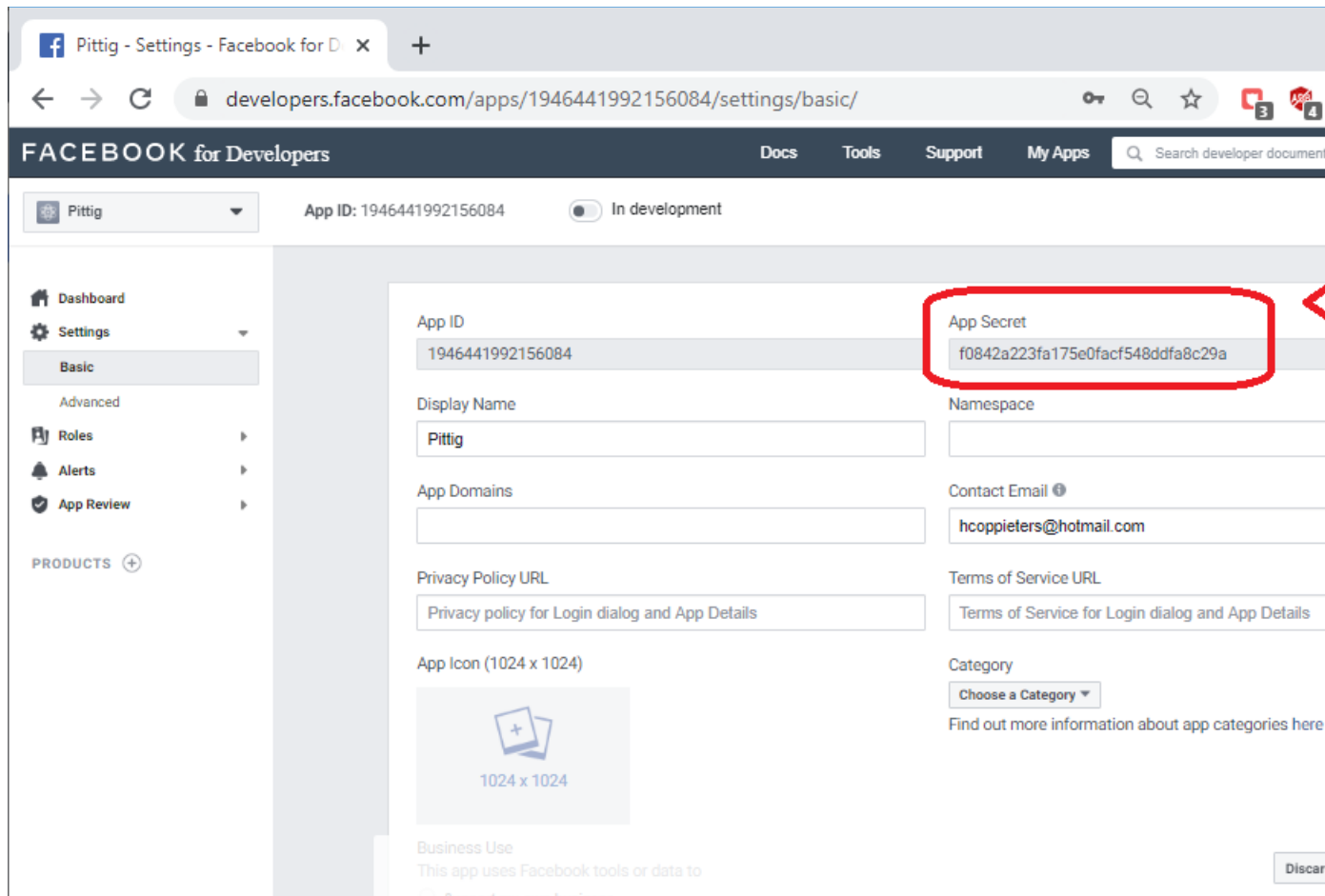
3.3 Toevoegen van Facebook Authentication service aan startup.cs (ConfigureServices methode):

Kopieer de App ID die is aangemaakt:



<https://developers.facebook.com/apps/1946441992156084/dashboard/>

Kies in het menu aan de linkerkant onder Settings/basic en kopieer de App Secret:



Open Startup.cs en voeg de volgende code toe (met de gekopieerde AppId en AppSecret) aan ConfigureServices:

```
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.AddDefaultIdentity<IdentityUser>()
    //.AddEntityFrameworkStores<ApplicationDbContext>();
}
```

```

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

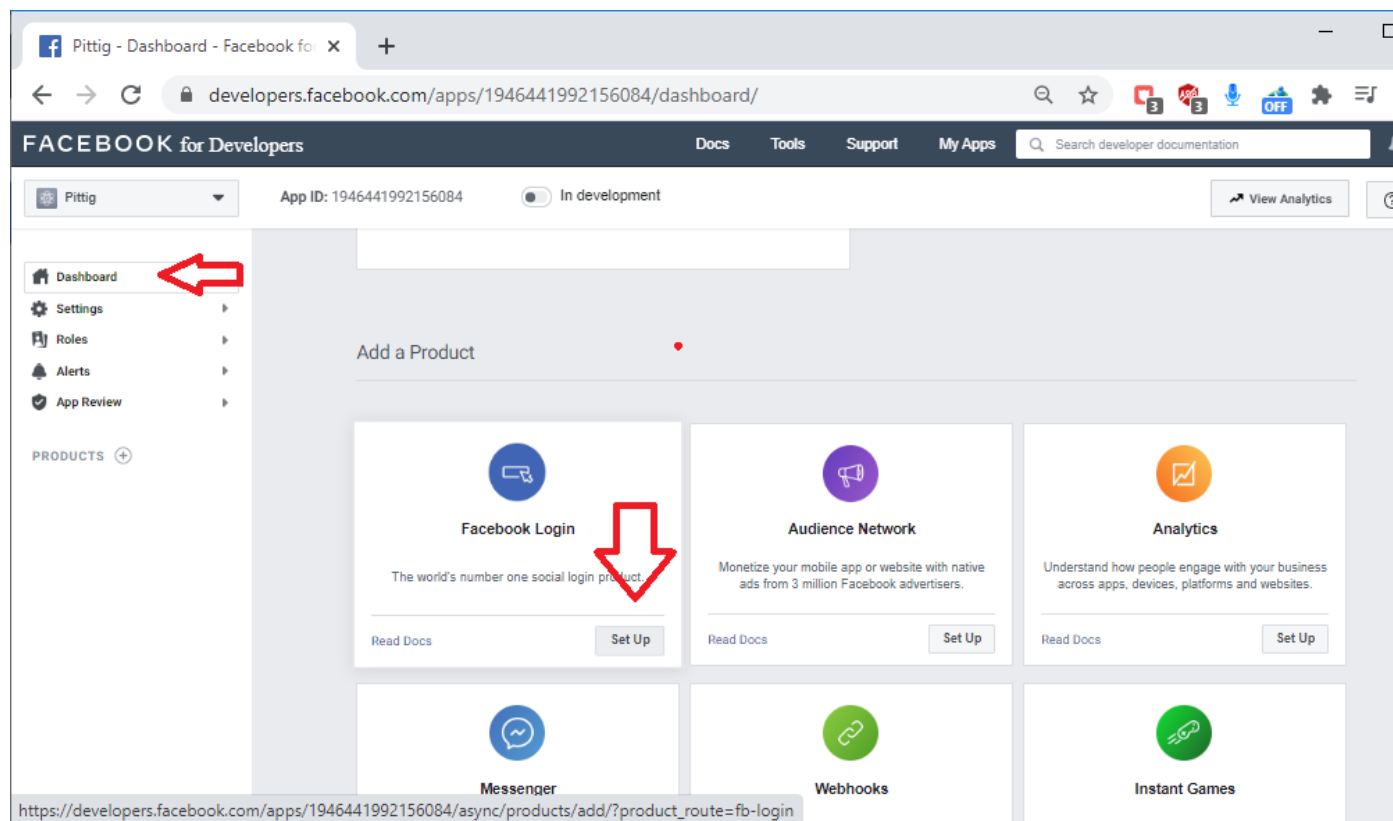
services.AddControllersWithViews();
services.AddRazorPages();

services.AddAuthentication().AddFacebook(facebookOptions =>
{
    facebookOptions.AppId = "1946441992156084";
    facebookOptions.AppSecret = "f0842a223fa175e0facf548ddfa8c29a";
});

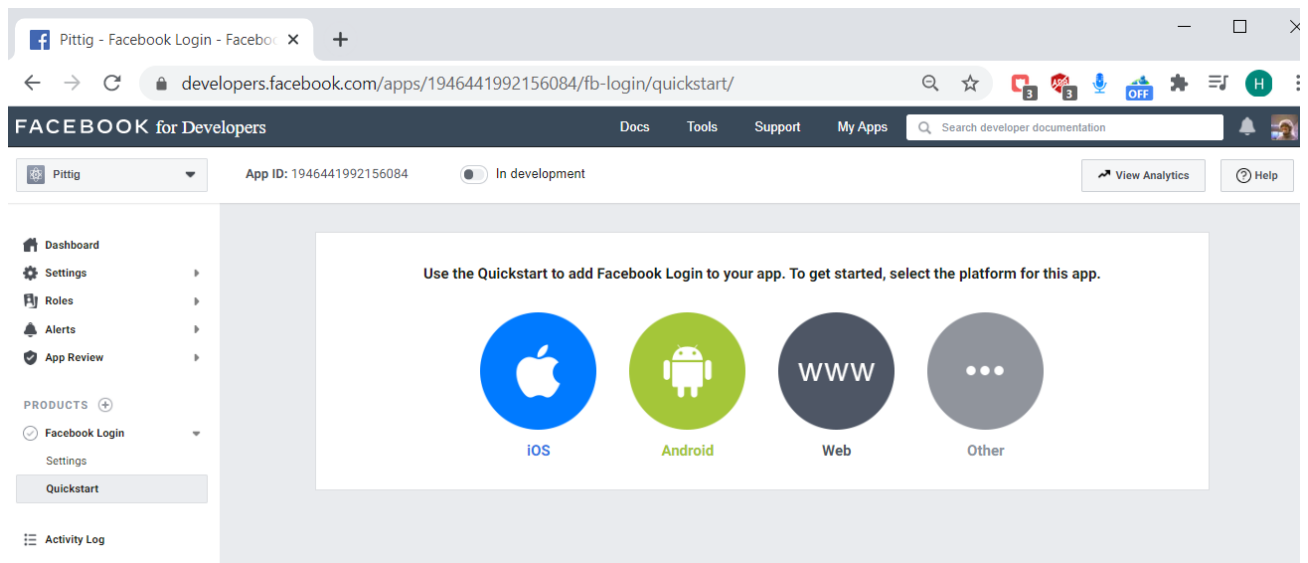
services.AddSession(options =>
{
    options.Cookie.IsEssential = true;
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
});
}

```

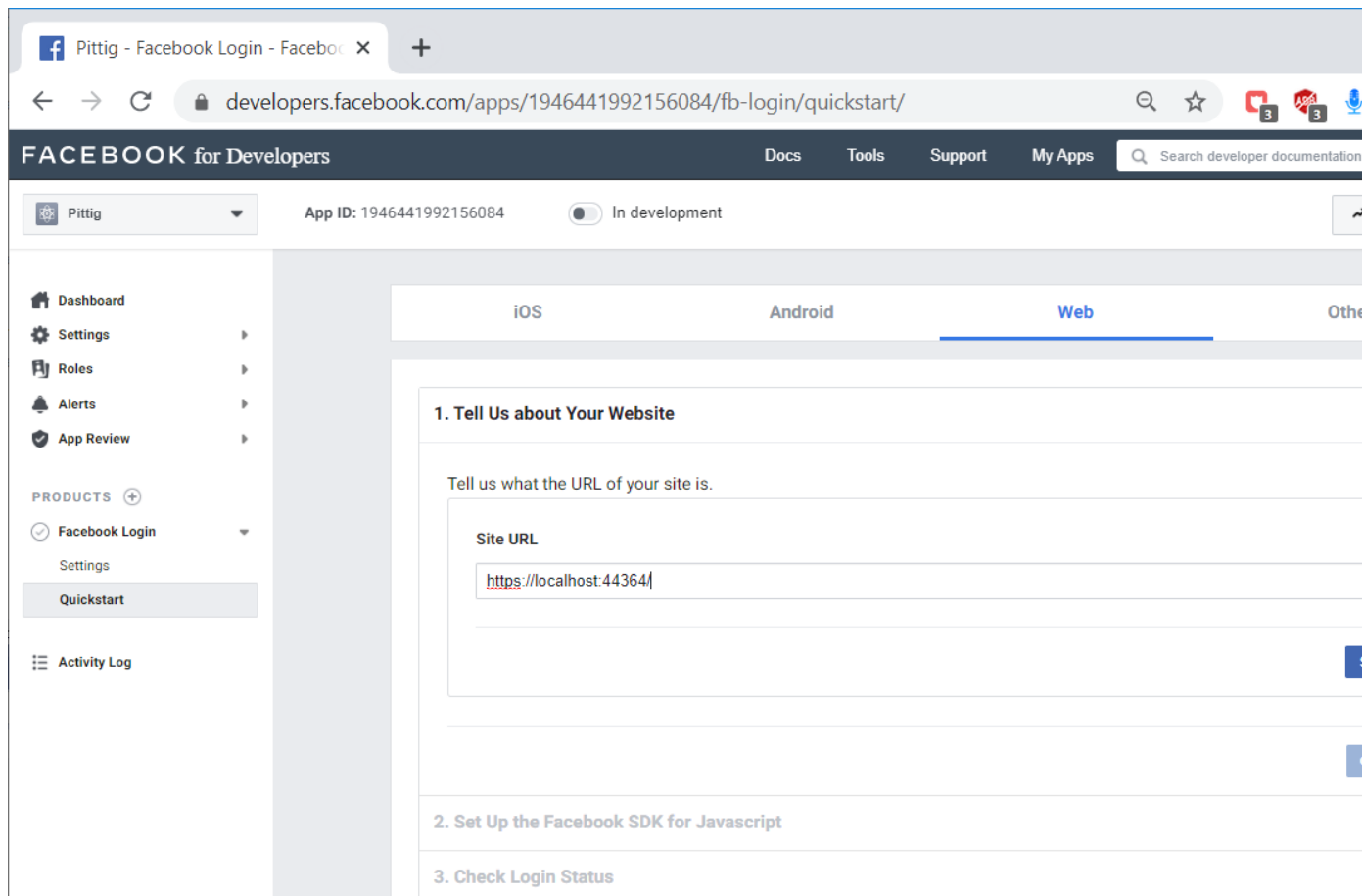
Op de facebook developer dashboard pagina, klik op de 'Set up' button bij Facebook Login:



Klik op WWW (web):

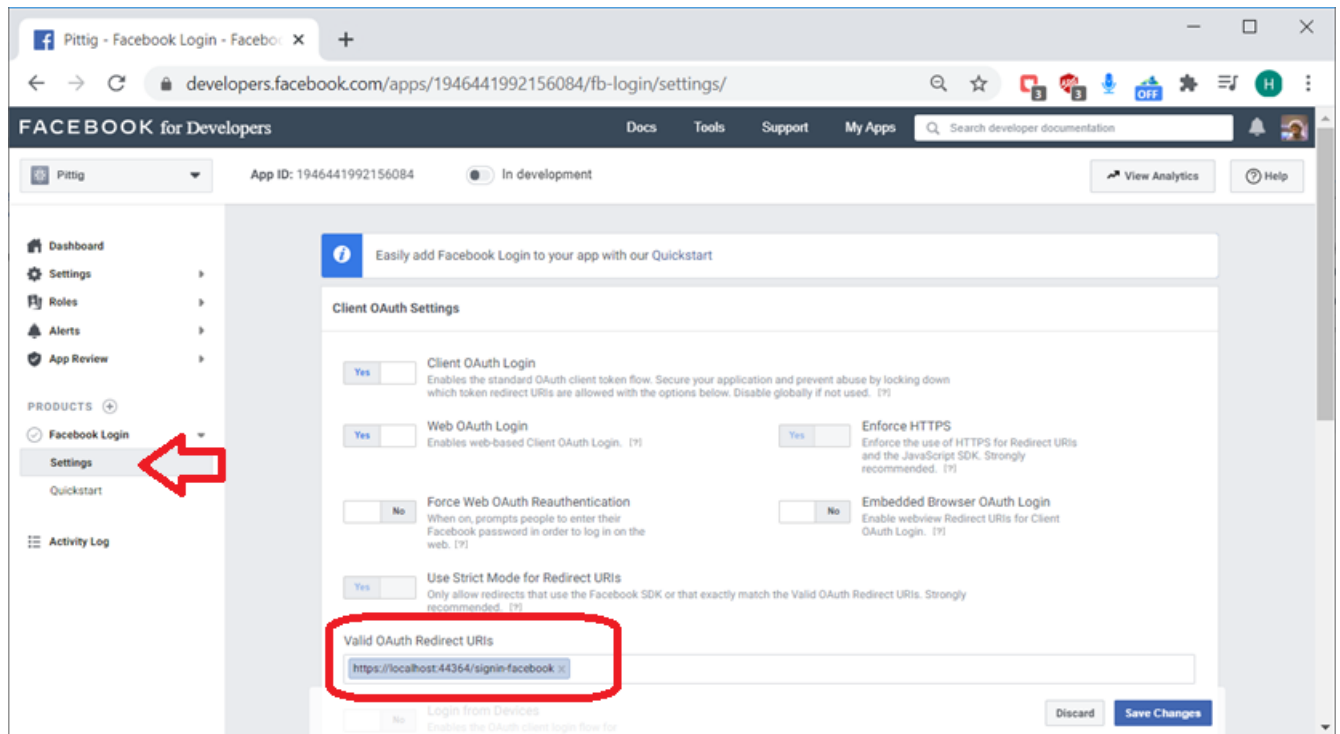


Zet de URL van je website en klik op Save:



Ga naar Settingspagina onder Facebook login en zet de return URL op je website nadat er is ingelogd op facebook:

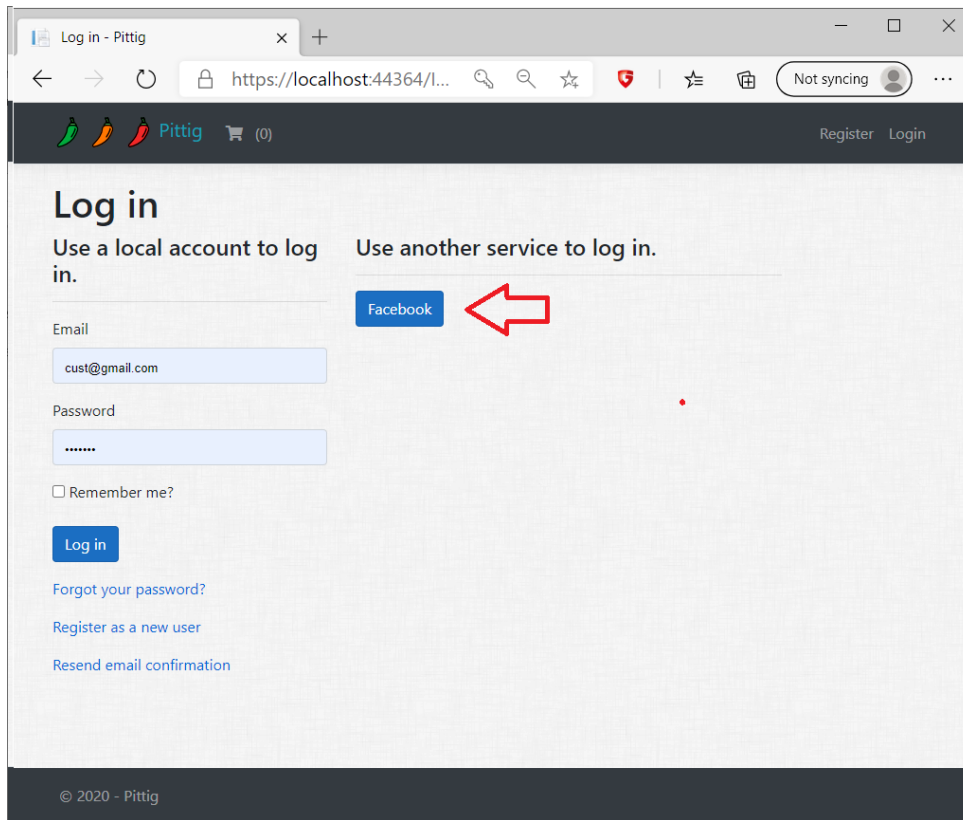
<https://localhost:44347/signin-facebook>



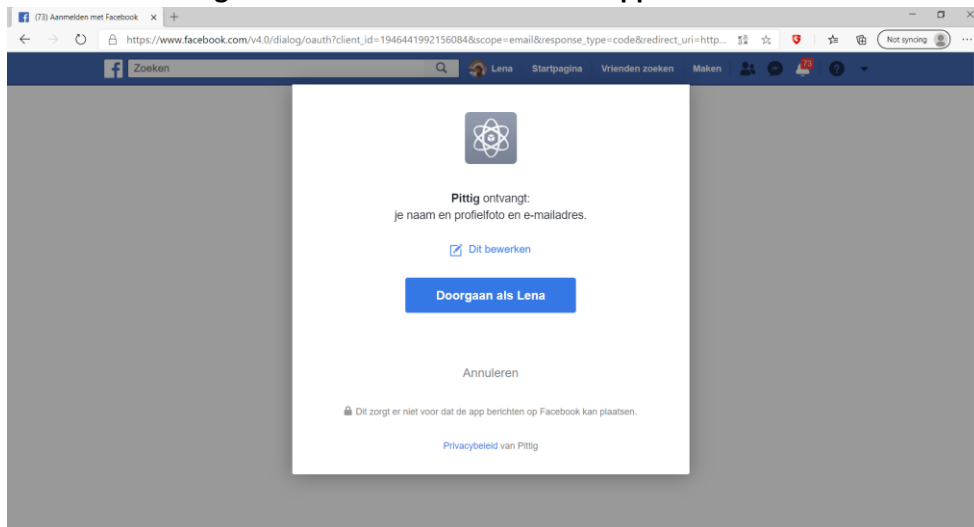
Klik op de button 'Save changes'

3.4 Testen van Facebook Authenticatie

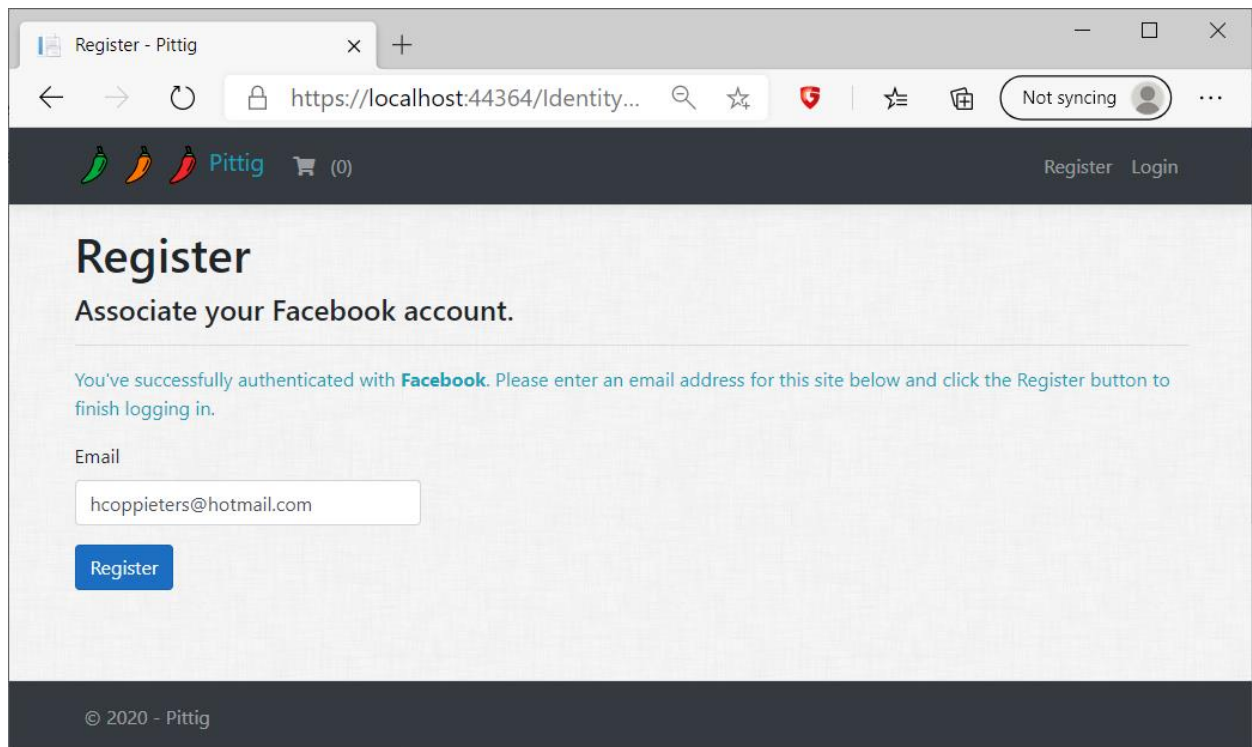
1. Run je app en ga naar de login pagina
Klik op de button 'Facebook'



2. Je wordt nu doorgestuurd naar de facebook web app.



3. Nadat je geklikt hebt op 'Doorgaan als..' word je teruggestuurd naar je website (de Return URL dat je hebt gegeven op facebook):



3.5 Toevoegen van registratie input-velden voor externe login registratie:

We willen meer velden voor de registratie van de externe login:

1. Open `Areas/Identity/Account/ExternalLogin.cshtml` en voeg de extra Velden toe aan de Razor Page:

```
@page
@model ExternalLoginModel
@{
    ViewData["Title"] = "Registreer";
}

<h1>@ViewData["Title"]</h1>
<h4>Associateer uw @Model.LoginProvider account.</h4>
<hr />

<p class="text-info">
    U bent succesvol geauthenticeerd bij <strong>@Model.LoginProvider</strong>.
    Geef uw email adres in deze site en klik op Registreer om aan te melden
</p>

<div class="row">
    <div class="col-md-12">
        <form asp-page-handler="Confirmation" asp-route-returnUrl="@Model.ReturnUrl"
method="post">
            <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="Input.Name" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.Name" class="form-control" />
        </div>
        <span asp-validation-for="Input.Name" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.Email" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.Email" class="form-control" />
        </div>
        <span asp-validation-for="Input.Email" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.PhoneNumber" class="col-form-
label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.PhoneNumber" class="form-control" />
        </div>
        <span asp-validation-for="Input.PhoneNumber" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.StreetAddress" class="col-form-
label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.StreetAddress" class="form-control" />
        </div>
        <span asp-validation-for="Input.StreetAddress" class="text-
danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.City" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.City" class="form-control" />
        </div>
        <span asp-validation-for="Input.City" class="text-danger"></span>
    </div>
    <div class="form-group row">
        <div class="col-2">
            <label asp-for="Input.State" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.State" class="form-control" />
        </div>
        <span asp-validation-for="Input.State" class="text-danger"></span>
    </div>
    <div class="form-group row">

```

```

        <div class="col-2">
            <label asp-for="Input.PostalCode" class="col-form-label"></label>
        </div>
        <div class="col-5">
            <input asp-for="Input.PostalCode" class="form-control" />
        </div>
        <span asp-validation-for="Input.PostalCode" class="text-
danger"></span>
    </div>

    <div class="form-group row">
        <div class="col-5 offset-2">
            <button type="submit" class="btn btn-primary form-
control">Registreer</button>
        </div>
    </div>
</div>

</form>
</div>
</div>
@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}

```

2. Open code behind Areas/Identity/Account/ExternalLogin.cshtml.cs en voeg code toe om de ingelogde externe gebruiker automatisch de 'Customer' rol krijgt:

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.Extensions.Logging;
using PittigRestoMVC.Models;
using PittigRestoMVC.Utility;

namespace PittigRestoMVC.Areas.Identity.Pages.Account
{
    [AllowAnonymous]
    public class ExternalLoginModel : PageModel
    {
        private readonly SignInManager<IdentityUser> _signInManager;
        private readonly UserManager<IdentityUser> _userManager;
        private readonly ILogger<ExternalLoginModel> _logger;
        private readonly RoleManager<IdentityRole> _roleManager;

        public ExternalLoginModel(
            SignInManager<IdentityUser> signInManager,

```



```

        UserManager<IdentityUser> userManager,
        ILogger<ExternalLoginModel> logger,
        RoleManager<IdentityRole> roleManager
    )
    {
        _signInManager = signInManager;
        _userManager = userManager;
        _logger = logger;
        _roleManager = roleManager;
    }

    [BindProperty]
    public InputModel Input { get; set; }

    public string LoginProvider { get; set; }

    public string returnUrl { get; set; }

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

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

        [Required]
        public string Name { get; set; }

        public string StreetAddress { get; set; }
        public string PhoneNumber { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string PostalCode { get; set; }
    }

    public IActionResult OnGetAsync()
    {
        return RedirectToPage("./Login");
    }

    public IActionResult OnPost(string provider, string returnUrl = null)
    {
        // Request a redirect to the external login provider.
        var redirectUrl = Url.Page("./ExternalLogin", pageHandler: "Callback",
values: new { returnUrl });
        var properties =
_signInManager.ConfigureExternalAuthenticationProperties(provider, redirectUrl);
        return new ChallengeResult(provider, properties);
    }

    public async Task<IActionResult> OnGetCallbackAsync(string returnUrl = null,
string remoteError = null)
    {
        returnUrl = returnUrl ?? Url.Content("~/");
        if (remoteError != null)
        {

```

```

        ErrorMessage = $"Error from external provider: {remoteError}";
        return RedirectToPage("./Login", new { returnUrl = returnUrl });
    }
    var info = await _signInManager.GetExternalLoginInfoAsync();
    if (info == null)
    {
        ErrorMessage = "Error loading external login information.";
        return RedirectToPage("./Login", new { returnUrl = returnUrl });
    }

    // Sign in the user with this external login provider if the user already
has a login.
    var result = await
        _signInManager.ExternalLoginSignInAsync(info.LoginProvider, info.ProviderKey,
        isPersistent: false, bypassTwoFactor: true);
    if (result.Succeeded)
    {
        _logger.LogInformation("{Name} logged in with {LoginProvider}
provider.", info.Principal.Identity.Name, info.LoginProvider);
        return LocalRedirect(returnUrl);
    }
    if (result.IsLockedOut)
    {
        return RedirectToPage("./Lockout");
    }
    else
    {
        // If the user does not have an account, then ask the user to create
an account.

        returnUrl = returnUrl;
        LoginProvider = info.LoginProvider;
        if (info.Principal.HasClaim(c => c.Type == ClaimTypes.Email))
        {
            Input = new InputModel
            {
                Email = info.Principal.FindFirstValue(ClaimTypes.Email),
                Name = info.Principal.FindFirstValue(ClaimTypes.Name)
            };
        }
        return Page();
    }
}

public async Task<IActionResult> OnPostConfirmationAsync(string returnUrl =
null)
{
    returnUrl = returnUrl ?? Url.Content("~/");
    // Get the information about the user from the external login provider
    var info = await _signInManager.GetExternalLoginInfoAsync();
    if (info == null)
    {
        ErrorMessage = "Error loading external login information during
confirmation.";
        return RedirectToPage("./Login", new { returnUrl = returnUrl });
    }

    if (ModelState.IsValid)
    {

```

```

var user = new ApplicationUser
{
    UserName = Input.Email,
    Email = Input.Email,
    Name = Input.Name,
    City = Input.City,
    StreetAddress = Input.StreetAddress,
    State = Input.State,
    PostalCode = Input.PostalCode,
    PhoneNumber = Input.PhoneNumber
};

var result = await _userManager.CreateAsync(user);
if (result.Succeeded)
{
    await _userManager.AddToRoleAsync(user, SD.CustomerEndUser);
    result = await _userManager.AddLoginAsync(user, info);
    if (result.Succeeded)
    {
        await _signInManager.SignInAsync(user, isPersistent: false);
        _logger.LogInformation("User created an account using {Name} provider.", info.LoginProvider);
        return LocalRedirect(returnUrl);
    }
}
foreach (var error in result.Errors)
{
    ModelState.AddModelError(string.Empty, error.Description);
}

LoginProvider = info.LoginProvider;
ReturnUrl = returnUrl;
return Page();
}
}
}

```

3.6 Google login

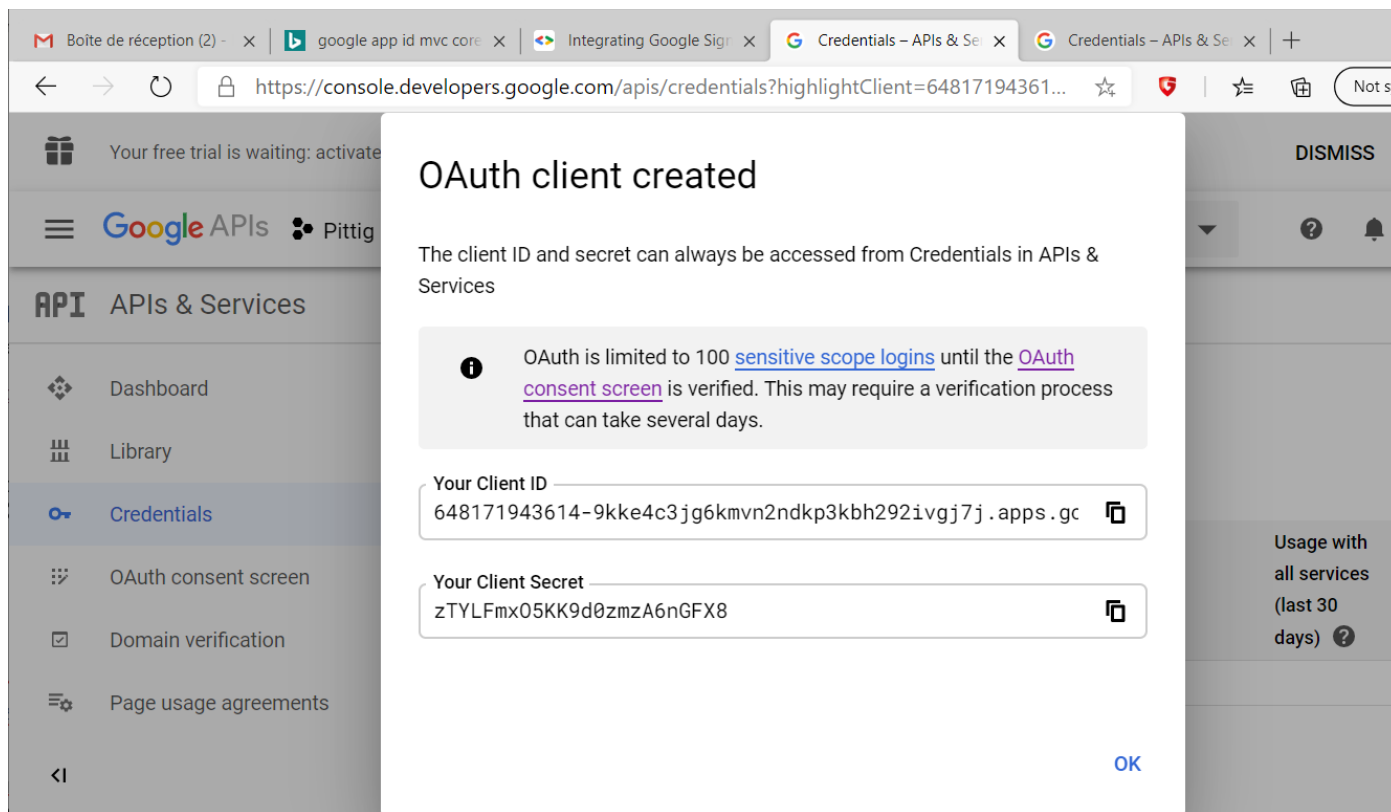
De methode is gelijkaardig als facebook

Start vanaf deze pagina:

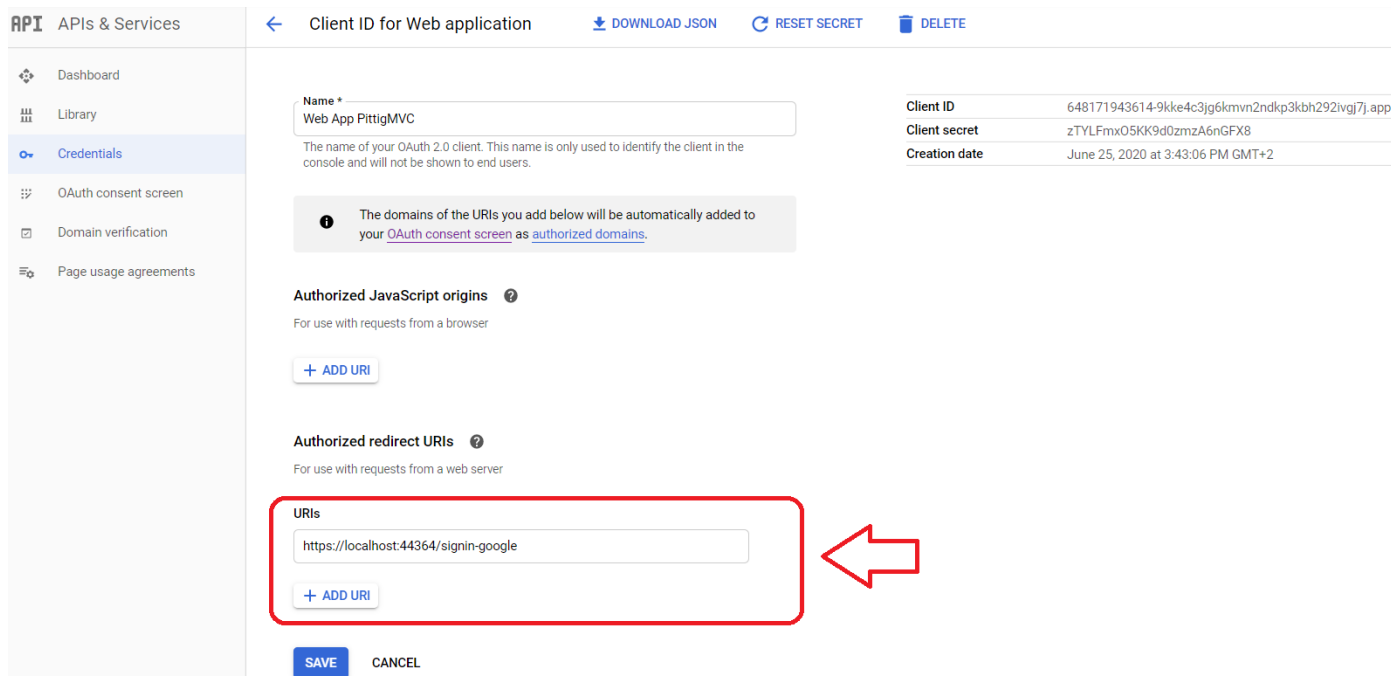
<https://developers.google.com/identity/sign-in/web/sign-in>

En volg deze stappen:

1. Go to the [Credentials page](#).
2. Click **Create credentials > OAuth client ID**.
3. Select the **Web application** application type.
4. Name your OAuth 2.0 client and click **Create**
(Er wordt dan gevraagd om eerst een OAuth consent screen aan te maken)



De Return URL na login bij google:



Open Startup.cs en voeg de volgende lijnen toe aan ConfigureServices:

```
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.AddDefaultIdentity<IdentityUser>()
//.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.AddAuthentication().AddFacebook(facebookOptions =>
{
    facebookOptions.AppId = "1946441992156084";
    facebookOptions.AppSecret = "f0842a223fa175e0facf548ddfa8c29a";
});

services.AddAuthentication().AddGoogle(googleOptions =>
{
    googleOptions.ClientId = "648171943614-
9kke4c3jg6kmvn2ndkp3kbh292ivgj7j.apps.googleusercontent.com";
    googleOptions.ClientSecret = "zTYLFmx05KK9d0zmzA6nGFX8";
});

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

```

4 Bestellingen

4.1 Aanpassen van Views/Shared/_Layout.cshtml

```

@using PittigRestoMVC.Utility
@using Microsoft.AspNetCore.Http

@Inject IHttpContextAccessor HttpContextAccessor

```

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - Pittig</title>

    <environment include="Development">
        <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
        <link rel="stylesheet" href="~/css/jquery-ui.css" />
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-
timepicker/1.10.0/jquery.timepicker.css" />
        <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3B1XeVIU"
crossorigin="anonymous">
    </environment>
    <environment exclude="Development">
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/jquery-
timepicker/1.10.0/jquery.timepicker.css" />
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/css/bootstrap.min.css"
asp-fallback-href="~/lib/bootstrap/dist/css/bootstrap.min.css"
asp-fallback-test-class="sr-only" asp-fallback-test-property="position"
asp-fallback-test-value="absolute"
crossorigin="anonymous"
integrity="sha256-eSi1q2PG6J7g7ib17yAaWMcrr5GrtohyChqibrV7PBE=" />
        <link rel="stylesheet" href="~/css/jquery-ui.css" />
        <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-
B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3B1XeVIU"
crossorigin="anonymous">
    </environment>
    <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="Customer" asp-controller="Home" asp-
action="Index">
                    
                    
                    
                    <label class="text-info" style="cursor:pointer">Pittig</label>
                </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 (User.IsInRole(SD.ManagerUser) ||
User.IsInRole(SD.CustomerEndUser) || User.IsInRole(SD.FrontDeskUser) ||
User.IsInRole(SD.KitchenUser) )
        {
            <li class="nav-item">
                <a asp-area="Customer" asp-controller="Order" asp-
action="OrderHistory" class="nav-link">
                    Geschiedenis
                </a>
            </li>
        }

        @if (User.IsInRole(SD.FrontDeskUser) ||
User.IsInRole(SD.KitchenUser) || User.IsInRole(SD.ManagerUser))
        {
            <li class="nav-item dropdown text-white-50">
                <a class="nav-link dropdown-toggle" href="#"
id="navbarDropdown" role="button" data-toggle="dropdown" aria-haspopup="true" aria-
expanded="false">
                    Bestellingen
                </a>
                <div class="dropdown-menu" aria-
labelledby="navbarDropDownMenuLink">
                    @if (User.IsInRole(SD.KitchenUser) ||
User.IsInRole(SD.ManagerUser))
                    {
                        <a class="dropdown-item" asp-action="ManageOrder"
asp-controller="Order" asp-area="Customer">Beheer Bestelling</a>
                    }
                    @if (User.IsInRole(SD.FrontDeskUser) ||
User.IsInRole(SD.ManagerUser))
                    {

```

```

        <a class="dropdown-item" asp-action="OrderPickup"
asp-controller="Order" asp-area="Customer">Ophalen Bestelling</a>
    }
</div>

```

}

```

@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>
}
</ul>
</div>
</div>
</nav>
</header>
<div class="container">
    <partial name="_CookieConsentPartial" />
    <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 - Pittig
    </div>
</footer>

<environment include="Development">
    <script
src="https://cloud.tinymce.com/stable/tinymce.min.js?apiKey=n94ifuzvl80pchikopiwgz2esrw8r
28dwcywpejlrregfp"></script>
    <script src="~/lib/jquery/dist/jquery.js"></script>
    <script src="~/lib/bootstrap/dist/js/bootstrap.bundle.js"></script>
    <script src="~/js/jquery-ui.js"></script>
</environment>

```



```

    <environment exclude="Development">
      <script
src="https://cloud.tinymce.com/stable/tinymce.min.js?apiKey=n94ifuzvl80pchikopiwgz2esrw8n
28dwcywvpeljqrregfp"></script>
      <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"
asp-fallback-src="~/lib/jquery/dist/jquery.min.js"
asp-fallback-test="window.jQuery"
crossorigin="anonymous"
integrity="sha256-FgpcB/KJQLNfOu91ta32o/NMZxltwRo8QtmkMRdAu8=">
      </script>
      <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/4.1.3/js/bootstrap.bundle.min.js"
asp-fallback-src="~/lib/bootstrap/dist/js/bootstrap.bundle.min.js"
asp-fallback-test="window.jQuery && window.jQuery.fn &&
window.jQuery.fn.modal"
crossorigin="anonymous"
integrity="sha256-E/V4cWE4qvAe05M0hjtGtqDzPndR01LBk81J/PR7CA4=">
      </script>
      <script src="~/js/jquery-ui.js"></script>
    </environment>
    <script src="~/js/site.js" asp-append-version="true"></script>

    @RenderSection("Scripts", required: false)
  </body>
</html>

```

4.2 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; }
    }
}

```

4.3 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; }
}

```

4.4 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; }
    }
}

```

4.5 Partial Views onder Views/Shared/ toevoegen:



PartialViews.zip.zip

4.6 Order Controller toevoegen

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

*(Opmerking: De action method GetOrderDetails in Order Controller zal een **Partial View** teruggeven)*

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Claims;
using System.Text;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Authorization;
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;

namespace PittigRestoMVC.Areas.Customer.Controllers
{
    [Area("Customer")]
    public class OrderController : Controller
    {
        private readonly IEmailSender _emailSender;
        private readonly ApplicationDbContext _db;
        private int PageSize = 2;
        public OrderController(ApplicationDbContext db, IEmailSender emailSender)
        {
            _db = db;
            _emailSender = emailSender;
        }

        [Authorize]
        public async Task<IActionResult> Confirm(int id)
        {
            var claimsIdentity = (ClaimsIdentity)User.Identity;
            var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

```

```

        OrderDetailsViewModel orderDetailsViewModel = new OrderDetailsViewModel()
        {
            OrderHeader = await _db.OrderHeader.Include(o =>
o.ApplicationUser).FirstOrDefaultAsync(o => o.Id == id && o.UserId == claim.Value),
            OrderDetails = await _db.OrderDetails.Where(o => o.OrderId ==
id).ToListAsync()
        };

        return View(orderDetailsViewModel);
    }

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

    public IActionResult GetOrderStatus(int Id)
    {
        return PartialView("_OrderStatus", _db.OrderHeader.Where(m => m.Id ==
Id).FirstOrDefault().Status);
    }

    [Authorize]
    public async Task<IActionResult> OrderHistory(int productPage=1)
    {
        var claimsIdentity = (ClaimsIdentity)User.Identity;
        var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

        OrderListViewModel orderListVM = new OrderListViewModel()
        {
            Orders = new List<OrderDetailsViewModel>()
        };

        List<OrderHeader> OrderHeaderList = await _db.OrderHeader.Include(o =>
o.ApplicationUser).Where(u => u.UserId == claim.Value).ToListAsync();

        foreach (OrderHeader item in OrderHeaderList)
        {
            OrderDetailsViewModel individual = new OrderDetailsViewModel
            {
                OrderHeader = item,
                OrderDetails = await _db.OrderDetails.Where(o => o.OrderId ==
item.Id).ToListAsync()
            };
            orderListVM.Orders.Add(individual);
        }

        var count = orderListVM.Orders.Count;
        orderListVM.Orders = orderListVM.Orders.OrderByDescending(p =>
p.OrderHeader.Id)
            .Skip((productPage - 1) * PageSize)
            .Take(PageSize).ToList();

        orderListVM.PagingInfo = new PagingInfo
        {
            CurrentPage = productPage,

```

```

        ItemsPerPage = PageSize,
        TotalItem = count,
        urlParam = "/Customer/Order/OrderHistory?productPage="
    };

    return View(orderListVM);
}

[Authorize(Roles = SD.KitchenUser + "," + SD.ManagerUser)]
public async Task<IActionResult> ManageOrder(int productPage = 1)
{
    List<OrderDetailsViewModel> orderDetailsVM = new
List<OrderDetailsViewModel>();

    List<OrderHeader> OrderHeaderList = await
_db.OrderHeader.Where(o=>o.Status==SD.StatusSubmitted ||
o.Status==SD.StatusInProgress).OrderByDescending(u=>u.PickUpTime).ToListAsync();

    foreach (OrderHeader item in OrderHeaderList)
    {
        OrderDetailsViewModel individual = new OrderDetailsViewModel
        {
            OrderHeader = item,
            OrderDetails = await _db.OrderDetails.Where(o => o.OrderId ==
item.Id).ToListAsync()
        };
        orderDetailsVM.Add(individual);
    }

    return View(orderDetailsVM.OrderBy(o=>o.OrderHeader.PickUpTime).ToList());
}

public async Task<IActionResult> GetOrderDetails(int Id)
{
    OrderDetailsViewModel orderDetailsViewModel = new OrderDetailsViewModel()
    {
        OrderHeader = await _db.OrderHeader.Include(e1 =>
e1.ApplicationUser).FirstOrDefaultAsync(m => m.Id == Id),
        OrderDetails = await _db.OrderDetails.Where(m => m.OrderId ==
Id).ToListAsync()
    };
    //orderDetailsViewModel.OrderHeader.ApplicationUser = await
_db.ApplicationUser.FirstOrDefaultAsync(u => u.Id ==
orderDetailsViewModel.OrderHeader.UserId);

    return PartialView("_IndividualOrderDetails", orderDetailsViewModel); //Deze
action method geeft een Partial View terug
}

[Authorize(Roles =SD.KitchenUser + ","+ SD.ManagerUser)]
public async Task<IActionResult> OrderPrepare(int OrderId)
{
    OrderHeader orderHeader = await _db.OrderHeader.FindAsync(OrderId);

```

```

        orderHeader.Status = SD.StatusInProgress;
        await _db.SaveChangesAsync();
        return RedirectToAction("ManageOrder", "Order");
    }

    [Authorize(Roles = SD.KitchenUser + "," + SD.ManagerUser)]
    public async Task<IActionResult> OrderReady(int OrderId)
    {
        OrderHeader orderHeader = await _db.OrderHeader.FindAsync(OrderId);
        orderHeader.Status = SD.StatusReady;
        await _db.SaveChangesAsync();

        //Email logic to notify user that order is ready for pickup
        await _emailSender.SendEmailAsync(_db.Users.Where(u => u.Id ==
orderHeader.UserId).FirstOrDefault().Email, "Pittig - Bestelling klaar voor ophaling " +
orderHeader.Id.ToString(), "Bestelling is klaar voor ophaling.");

        return RedirectToAction("ManageOrder", "Order");
    }

    [Authorize(Roles = SD.KitchenUser + "," + SD.ManagerUser)]
    public async Task<IActionResult> OrderCancel(int OrderId)
    {
        OrderHeader orderHeader = await _db.OrderHeader.FindAsync(OrderId);
        orderHeader.Status = SD.StatusCancelled;
        await _db.SaveChangesAsync();
        await _emailSender.SendEmailAsync(_db.Users.Where(u => u.Id ==
orderHeader.UserId).FirstOrDefault().Email, "Pittig - Order Cancelled " +
orderHeader.Id.ToString(), "Order has been cancelled successfully.");

        return RedirectToAction("ManageOrder", "Order");
    }

    [Authorize]
    public async Task<IActionResult> OrderPickup(int productPage = 1, string
searchEmail=null, string searchPhone = null, string searchName = null)
    {
        //var claimsIdentity = (ClaimsIdentity)User.Identity;
        //var claim = claimsIdentity.FindFirst(ClaimTypes.NameIdentifier);

        OrderListViewModel orderListVM = new OrderListViewModel()
        {
            Orders = new List<OrderDetailsViewModel>()
        };

        StringBuilder param = new StringBuilder();
        param.Append("/Customer/Order/OrderPickup?productPage=");
        param.Append("&searchName=");
        if (searchName != null)
        {
            param.Append(searchName);
        }
        param.Append("&searchEmail=");
        if (searchEmail != null)
        {
            param.Append(searchEmail);
        }
    }

```

```

    }
    param.Append("&searchPhone=");
    if (searchPhone != null)
    {
        param.Append(searchPhone);
    }

    List<OrderHeader> OrderHeaderList = new List<OrderHeader>();
    if (searchName != null || searchEmail != null || searchPhone != null)
    {
        var user = new ApplicationUser();

        if(searchName!=null)
        {
            OrderHeaderList = await _db.OrderHeader.Include(o =>
o.ApplicationUser)
                .Where(u =>
u.PickupName.ToLower().Contains(searchName.ToLower()))
                .OrderByDescending(o =>
o.OrderDate).ToListAsync();
        }
        else
        {
            if (searchEmail != null)
            {
                user = await _db.ApplicationUser.Where(u =>
u.Email.ToLower().Contains(searchEmail.ToLower())).FirstOrDefaultAsync();
                OrderHeaderList = await _db.OrderHeader.Include(o =>
o.ApplicationUser)
                    .Where(o=>o.UserId==user.Id)
                    .OrderByDescending(o =>
o.OrderDate).ToListAsync();
            }
            else
            {
                if (searchPhone != null)
                {
                    OrderHeaderList = await _db.OrderHeader.Include(o =>
o.ApplicationUser)
                        .Where(u =>
u.PhoneNumber.Contains(searchPhone))
                        .OrderByDescending(o =>
o.OrderDate).ToListAsync();
                }
            }
        }
        else
        {
            OrderHeaderList = await _db.OrderHeader.Include(o =>
o.ApplicationUser).Where(u => u.Status == SD.StatusReady).ToListAsync();
        }

        foreach (OrderHeader item in OrderHeaderList)
        {
            OrderDetailsViewModel individual = new OrderDetailsViewModel
            {
                OrderHeader = item,
                OrderDetails = await _db.OrderDetails.Where(o => o.OrderId ==
item.Id).ToListAsync()
            };
        }
    }
}

```

```

        orderListVM.Orders.Add(individual);
    }

    var count = orderListVM.Orders.Count;
    orderListVM.Orders = orderListVM.Orders.OrderByDescending(p =>
p.OrderHeader.Id)
        .Skip((productPage - 1) * PageSize)
        .Take(PageSize).ToList();

    orderListVM.PagingInfo = new PagingInfo
    {
        CurrentPage = productPage,
        ItemsPerPage = PageSize,
        TotalItem = count,
        urlParam = param.ToString()
    };

    return View(orderListVM);
}

[Authorize(Roles = SD.FrontDeskUser + "," + SD.ManagerUser)]
[HttpPost]
[ActionName("OrderPickup")]
public async Task<IActionResult> OrderPickupPost(int orderId)
{
    OrderHeader orderHeader = await _db.OrderHeader.FindAsync(orderId);
    orderHeader.Status = SD.StatusCompleted;
    await _db.SaveChangesAsync();
    await _emailSender.SendEmailAsync(_db.Users.Where(u => u.Id ==
orderHeader.UserId).FirstOrDefault().Email, "Pittig - Order Completed " +
orderHeader.Id.ToString(), "Order has been completed successfully.");

    return RedirectToAction("OrderPickup", "Order");
}
}
}

```

4.7 Bevestigingspagina Bestelling

Confirm.cshtml Razor View toevoegen:

```

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

<br />
<div class="backgroundWhite">
    <div class="row px-3">
        <div class="col-8">
            <h2 class="text-info">Bestelling bevestigd</h2>
        </div>
        <div class="col-4 text-right">
            <a asp-action="Index" asp-controller="Home" class="btn btn-success form-
control">Nieuwe bestelling</a>

```

```

        </div>
    </div>

    @if (Model.OrderHeader != null)
    {
        <div style="color:maroon" class="p-3">
            Uw bestelling is bevestigd. Hieronder vindt u het overzicht van uw
            bestelling.
            <br />

        </div>
        <partial name="_IndividualOrderDetails" model="Model" />
    }
    else
    {
        <div class="text-danger">Fout bij verwerken bestelling... Gelieve opnieuw te
            proberen.</div>
    }
</div>

```

4.8 Beheer van Bestelling

Toevoegen van Razor View ManageOrder.cshtml

```

@model List<PittigRestoMVC.Models.ViewModels.OrderDetailsViewModel>
@using PittigRestoMVC.Utility
@{
    ViewData["Title"] = "Beheer bestelling";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2 class="text-info">Beheer bestellingen</h2>
<form method="post">
    <br />
    @if (Model.Count > 0)
    {
        <div class="container row bg-white p-2 border mb-3 rounded">
            @foreach (var item in Model)
            {
                <div class="col-12 pb-3 pt-3">
                    <div class="backgroundWhiteBorder10Padding">
                        <div class="row">
                            <div class="col-lg-4 col-12">
                                BestelNr : @item.OrderHeader.Id
                                <ul class="text-info">
                                    @foreach (var details in item.OrderDetails)
                                    {
                                        <li>@details.Name x @details.Count</li>
                                    }
                                </ul>
                            </div>
                            <div class="col-lg-4 col-12 pb-2">
                                <div class="input-group pb-2">
                                    <div class="input-group-append">
                                        <span class="input-group-text">Tijd</span>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            }
        </div>
    }

```



```

        <input type="text" class="border rounded form-
control" readonly value="@item.OrderHeader.PickUpTime" />
    </div>
    @{
        var comments = item.OrderHeader.Comments;
        if (comments == null || comments.Trim().Length == 0)
        {
            item.OrderHeader.Comments = "No Instructions";
        }
    }
    <textarea asp-for="@item.OrderHeader.Comments" readonly
class="rounded border form-control" style="width:100%;height:75px;"></textarea>
</div>
<div class="col-lg-4 col-12 d-flex align-content-center">
    @if (item.OrderHeader.Status.Equals(SD.StatusSubmitted))
    {
        <div class="col-12">
            <button type="submit" asp-action="OrderPrepare"
asp-route-OrderId="@item.OrderHeader.Id" class="btn btn-info form-control mb-3">
                <i class="fas fa-check"></i> &nbsp; Start met
Koken
            </button>
            <button type="submit" asp-action="OrderCancel"
asp-route-OrderId="@item.OrderHeader.Id" class="btn btn-danger form-control mb-3">
                <i class="fas fa-times"></i> &nbsp; Annuleer
bestelling
            </button>
        </div>
    }
    else
    {
        @if
(item.OrderHeader.Status.Equals(SD.StatusInProgress))
        {
            <div class="col-12">
                <button type="submit" asp-action="OrderReady"
asp-route-OrderId="@item.OrderHeader.Id" class="btn btn-success form-control mb-3">
                    <i class="fas fa-flag-checkered"></i>
&nbsp; Bestelling klaar
                </button>
                <button type="submit" asp-
action="OrderCancel" asp-route-OrderId="@item.OrderHeader.Id" class="btn btn-danger form-
control mb-3">
                    <i class="fas fa-times"></i> &nbsp;
Annuleer bestelling
                </button>
            </div>
        }
    }
</div>
</div>
</div>
</div>
}
</div>
}
</form>

```

4.9 Geschiedenis van Bestelling

Toevoegen van Razor View OrderHistory:

```
@model PittigRestoMVC.Models.ViewModels.OrderListViewModel
@using PittigRestoMVC.Utility
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<br />
<br />
<div class="border backgroundWhite">
    <div class="row">
        <div class="col-7">
            <h2 class="text-info"> Overzicht bestellingen</h2>
        </div>
        <div class="col-7 text-right">
            <partial name="_CreateButtonPartial" />
        </div>
    </div>
    <br />
    <div>
        @if (Model.Orders.Count() > 0)
        {
            <table class="table table-striped border">
                <tr class="table-secondary">
                    <th>
                        @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.Id)
                    </th>
                    <th>
                        @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.PickupName)
                    </th>
                    <th>
                        @Html.DisplayNameFor(m =>
m.Orders[0].OrderHeader.ApplicationUser.Email)
                    </th>
                    <th>
                        @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.PickUpTime)
                    </th>
                    <th>
                        @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.OrderTotal)
                    </th>
                    <th>
                        Total Items
                    </th>
                </tr>
                @foreach (var item in Model.Orders)
                {
                    <tr>
                        <td>
                            @Html.DisplayFor(m => item.OrderHeader.Id)
                        </td>

```

```

        <td>
            @Html.DisplayFor(m => item.OrderHeader.PickupName)
        </td>
        <td>
            @Html.DisplayFor(m => item.OrderHeader.ApplicationUser.Email)
        </td>
        <td>
            @Html.DisplayFor(m => item.OrderHeader.PickUpTime)
        </td>
        <td>
            € @Html.DisplayFor(m => item.OrderHeader.OrderTotal)
        </td>
        <td>
            @Html.DisplayFor(m => item.OrderDetails.Count)
        </td>
        <td>
            <button type="submit" class="btn btn-success anchorDetail"
data-id="@item.OrderHeader.Id" data-toggle="modal">
                <i class="far fa-list-alt"></i> Details
            </button>
            @if (item.OrderHeader.Status == SD.StatusSubmitted ||
item.OrderHeader.Status == SD.StatusInProgress || item.OrderHeader.Status ==
SD.StatusCompleted || item.OrderHeader.Status == SD.StatusReady)
            {
                <button type="button" class="btn btn-info btnOrderStatus"
data-id="@item.OrderHeader.Id" data-toggle="modal">
                    <i class="far fa-clock"></i> Status
                </button>
            }
        </td>
    </tr>
}
</table>
<div page-model="@Model.PagingInfo" page-classes-enabled="true" page-
class="btn border"
        page-class-normal="btn btn-light" page-class-selected="btn btn-info
active"
        class="btn-group float-right">
    </div>
<br />
}
else
{
    <p>Er zijn nog geen bestellingen...</p>
}
</div>
</div>

<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-hidden="true">
    <div class="modal-dialog-centered modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header bg-success text-light justify-content-center">
                <h5 class="modal-title">Details bestelling</h5>
            </div>
            <div class="modal-body justify-content-center" id="myModalContent">

            </div>
            <div class="modal-footer">

```

```

        <button type="button" class="btn btn-secondary" style="width:20%" data-
dismiss="modal">Sluit</button>
    </div>
</div>
</div>
</div>
<div class="modal fade" id="myOrderStatus" tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header bg-info text-light justify-content-center">
                <h5 class="modal-title">Status </h5>
            </div>
            <div class="modal-body justify-content-center" id='myOrderStatusContent'>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" style="width:30%" data-
dismiss="modal">Sluit</button>
            </div>
        </div>
    </div>
</div>
</div>

```

@section scripts

```

{
    <script>var PostBackURL = '/Customer/Order/GetOrderDetails';
    var OrderStatusURL = '/Customer/Order/GetOrderStatus';

    $(function () {
        $(".anchorDetail").click(function () {

            var $buttonClicked = $(this);
            var id = $buttonClicked.attr('data-id');

            $.ajax({
                type: "GET",
                url: PostBackURL,
                contentType: "application/json; charset=utf-8",
                data: { "Id": id },
                cache: false,
                dataType: "html",
                success: function (data) {
                    $('#myModalContent').html(data);
                    $('#myModal').modal('show');
                },
                error: function () {
                    alert("Dynamic content load failed");
                }
            });

        });

        $(".btnOrderStatus").click(function () {
            var $buttonClicked = $(this);
            var id = $buttonClicked.attr('data-id');
            $.ajax({
                type: "GET",

```

```

        url: OrderStatusURL,
        contentType: "text/html; charset=utf-8",
        data: { "Id": id },
        cache: false,
        dataType: "html",
        success: function (data) {
            $('#myOrderStatusContent').html(data);
            $('#myOrderStatus').modal('show');
        },
        error: function () {
            alert("Dynamic content load failed.");
        }
    });
});
});</script>
}

```

4.10 Ophalen van Bestelling

Toevoegen van Razor View OrderPickup:

```

@model PittigRestoMVC.Models.ViewModels.OrderListViewModel
@using PittigRestoMVC.Utility
@{
    ViewData["Title"] = "Index";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

```

```

<form method="get">

    <br /><br />
    <h2 class="text-info"> Bestellingen klaar voor ophaling : </h2>
    <div class="border backgroundWhite">
        <div class="container border border-secondary" style="height:60px;">
            <div class="row container">
                <div class="col-11">
                    <div class="row" style="padding-top:10px">
                        <div class="col-4">
                            @Html.Editor("searchName", new { htmlAttributes = new {
@class = "form-control", @placeholder = "Name..." } })
                        </div>
                        <div class="col-4">
                            @Html.Editor("searchPhone", new { htmlAttributes = new {
@class = "form-control", @placeholder = "Phone..." } })
                        </div>
                        <div class="col-4">
                            @Html.Editor("searchEmail", new { htmlAttributes = new {
@class = "form-control", @placeholder = "Email..." } })
                        </div>
                    </div>
                </div>
                <div class="col-1">
                    <div class="row" style="padding-top:10px;">
                        <button type="submit" name="submit" value="Search" class="btn
btn-info form-control">

```

```

        <i class="fas fa-search"></i>
    </button>
</div>
</div>
</div>
</div>
<br />
<div>
    @if (Model.Orders.Count() > 0)
    {
        <table class="table table-striped border">
            <tr class="table-secondary">
                <th>
                    @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.Id)
                </th>
                <th>
                    @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.PickupName)
                </th>
                <th>
                    @Html.DisplayNameFor(m =>
m.Orders[0].OrderHeader.ApplicationUser.Email)
                </th>
                <th>
                    @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.PickUpTime)
                </th>
                <th>
                    @Html.DisplayNameFor(m => m.Orders[0].OrderHeader.OrderTotal)
                </th>
                <th>
                    Total Items
                </th>
            </tr>
            @foreach (var item in Model.Orders)
            {
                <tr>
                    <td>
                        @Html.DisplayFor(m => item.OrderHeader.Id)
                    </td>
                    <td>
                        @Html.DisplayFor(m => item.OrderHeader.PickupName)
                    </td>
                    <td>
                        @Html.DisplayFor(m =>
item.OrderHeader.ApplicationUser.Email)
                    </td>
                    <td>
                        @Html.DisplayFor(m => item.OrderHeader.PickUpTime)
                    </td>
                    <td>
                        @Html.DisplayFor(m => item.OrderHeader.OrderTotal)
                    </td>
                    <td>
                        @Html.DisplayFor(m => item.OrderDetails.Count)
                    </td>
                </tr>
            }
        </table>
    }

```

```

        <button type="button" class="btn btn-success
anchorDetail" data-id="@item.OrderHeader.Id" data-toggle="modal">
        <i class="far fa-list-alt"></i> Details
        </button>

    </td>
</tr>
}
</table>
<div page-model="@Model.PagingInfo" page-classes-enabled="true" page-
class="btn border"
        page-class-normal="btn btn-light" page-class-selected="btn btn-info
active"
        class="btn-group float-right">
    </div>
    <br />
}
else
{
    <p>Lijst is leeg...</p>
}
</div>
</div>
</form>

<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-hidden="true">
    <div class="modal-dialog-centered modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header bg-success text-light ">
                <div class="col-10 offset-1">
                    <center><h5 class="modal-title">Details bestelling</h5></center>
                </div>
                <div class="col-1">
                    <button type="button" class="float-right btn-outline-secondary close"
data-dismiss="modal" aria-label="Close">
                        <span aria-hidden="true">&times;</span>
                    </button>
                </div>
            </div>
            <div class="modal-body justify-content-center" id="myModalContent">

                </div>

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

```

```

@section scripts
{
    <script>var PostBackURL = '/Customer/Order/GetOrderDetails';

    $(function () {
        $(".anchorDetail").click(function () {

            var $buttonClicked = $(this);
            var id = $buttonClicked.attr('data-id');

```

```

        $.ajax({
            type: "GET",
            url: PostBackURL,
            contentType: "application/json; charset=utf-8",
            data: { "Id": id },
            cache: false,
            datatype: "json",
            success: function (data) {
                $('#myModalContent').html(data);
                $('#myModal').modal('show');
            },
            error: function () {
                alert("Dynamic content load failed");
            }
        });
    });
});</script>
}

```

4.11 Custom TagHelper

Tot nu toe hebben we enkel met de built-in Razor taghelpers gewerkt. Het is mogelijk om zelf Taghelpers te definiëren

Hiervoor maken we een nieuwe afgeleide class van TagHelper:

We maken een nieuwe class 'PageLinkTagHelper' in een nieuwe folder TagHelpers:

```

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.AspNetCore.Mvc.Routing;
using Microsoft.AspNetCore.Mvc.ViewFeatures;
using Microsoft.AspNetCore.Razor.TagHelpers;
using PittigRestoMVC.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace PittigRestoMVC.TagHelpers
{
    [HtmlTargetElement("div", Attributes = "page-model")]
    public class PageLinkTagHelper : TagHelper
    {
        private IUrlHelperFactory urlHelperFactory;

        public PageLinkTagHelper(IUrlHelperFactory helperFactory)
        {
            urlHelperFactory = helperFactory;
        }

        [ViewContext]

```



```

[HtmlAttributeNotBound]
public ViewContext ViewContext { get; set; }

public PagingInfo PageModel { get; set; }
public string PageAction { get; set; }
public bool PageClassesEnabled { get; set; }
public string PageClass { get; set; }
public string PageClassNormal { get; set; }
public string PageClassSelected { get; set; }

public override void Process(TagHelperContext context, TagHelperOutput output)
{
    IUrlHelper urlHelper = urlHelperFactory.GetUrlHelper(ViewContext);
    TagBuilder result = new TagBuilder("div");

    for (int i = 1; i <= PageModel.totalPage; i++)
    {
        TagBuilder tag = new TagBuilder("a");
        string url = PageModel.urlParam.Replace(":", i.ToString());
        tag.Attributes["href"] = url;
        if (PageClassesEnabled)
        {
            tag.AddCssClass(PageClass);
            tag.AddCssClass(i == PageModel.CurrentPage ? PageClassSelected :
PageClassNormal);
        }
        tag.InnerHtml.Append(i.ToString());
        result.InnerHtml.AppendHtml(tag);
    }

    output.Content.AppendHtml(result.InnerHtml);
}
}

```

5 Referenties

<https://jqueryui.com/datepicker/>

<https://cdnjs.com/libraries/jquery-timepicker>

<https://stripe.com/docs>

<https://blog.elmah.io/accepting-payments-with-stripe-and-aspnet-core/>

<https://www.youtube.com/watch?v=KLrJDI8rc9M> (stripe tutorial video-reeks)

<https://stripe.com/docs/legacy-checkout/aspnet>

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/social/facebook-logins?view=aspnetcore-3.1>

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

<https://docs.microsoft.com/en-us/aspnet/core/security/authentication/social/google-logins?view=aspnetcore-3.1>

<https://developers.google.com/identity/sign-in/web/sign-in>