



C# Web – MVC

PRO/PRW - C# Web 1

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Doel

- ASP.Net Core toepassing
 - Identity Framework

ASP.NET CORE

MVC Web Application
RazorVoertuigIdentity

Identity Framework

- ASP.Net Core – MVC – RazorVoertuigIdentity
- Authenticatie aanduiden

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Web

Framework ⓘ

.NET 6.0 (Long-term support)

Authentication type ⓘ

Individual Accounts

☒ Configure for HTTPS ⓘ

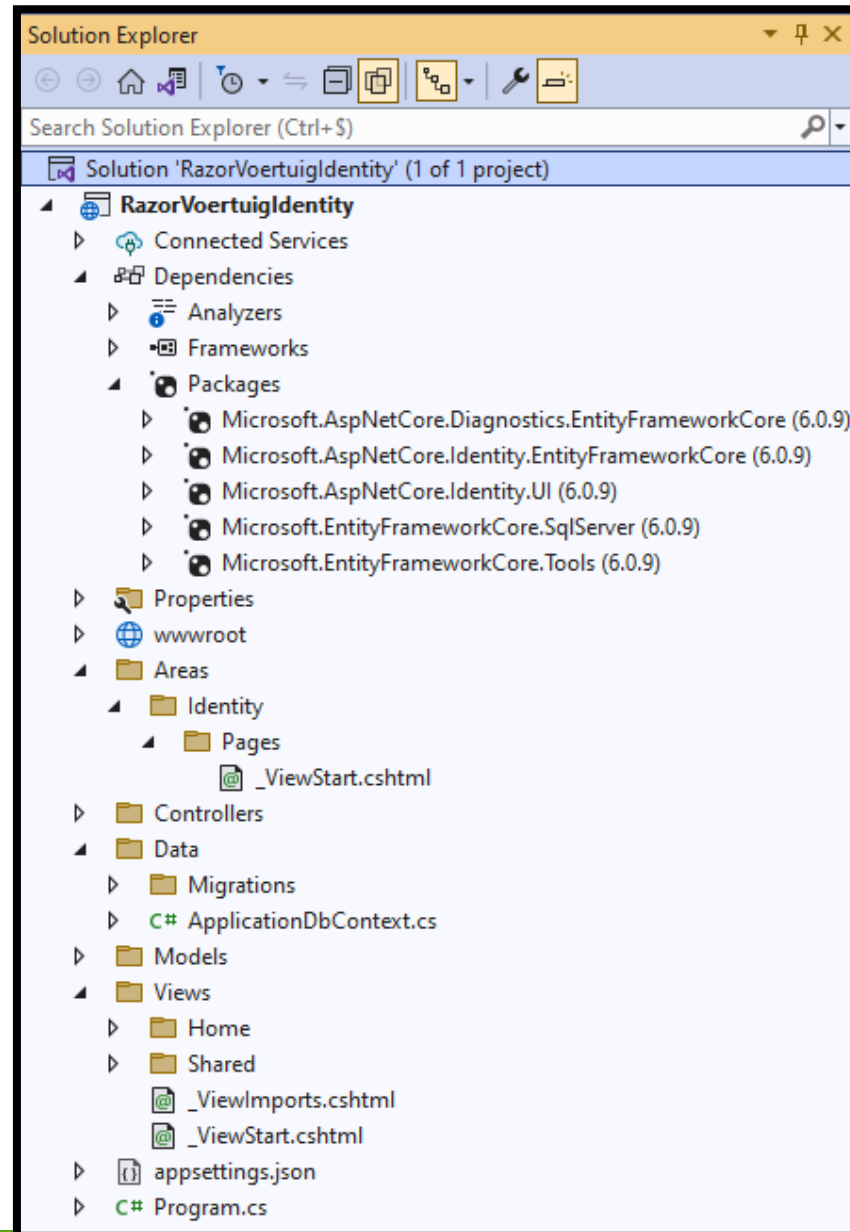
☐ Enable Docker ⓘ

Docker OS ⓘ

Linux

☐ Do not use top-level statements ⓘ

Identity Framework



Identity Framework

- Data - ApplicationDbContext

```
public class ApplicationDbContext : IdentityDbContext
{
    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
        : base(options)
    {
    }
}
```

Identity Framework

- Data - connection string

```
{  
  "ConnectionStrings": {  
    "DefaultConnection":  
    "Server=(localdb)\\mssqllocaldb;  
Database=aspnet-RazorVoertuigIdentity-9781FC8A-3611-49C1-86D8-214723AFBEB0;  
Trusted_Connection=True;  
MultipleActiveResultSets=true"  
  }  
}
```

Identity Framework

- Program.cs - Services

```
var connectionString =  
builder.Configuration.GetConnectionString("DefaultConnection");  
builder.Services.AddDbContext<ApplicationDbContext>(options =>  
    options.UseSqlServer(connectionString));  
  
builder.Services.AddDefaultIdentity<IdentityUser>(options =>  
    options.SignIn.RequireConfirmedAccount = true)  
    .AddEntityFrameworkStores<ApplicationDbContext>();
```


Identity Framework

- Program.cs - Middleware

```
app.UseAuthentication();  
app.UseAuthorization();  
  
app.MapControllerRoute(  
    name: "default",  
    pattern: "{controller=Home}/{action=Index}/{id?}");  
app.MapRazorPages();
```

Identity Framework

- Views/Shared
 - _Layout.cshtml

```
<header>  
  <nav ...>  
    <partial name="_LoginPartial" />  
  </nav>  
</header>
```

```
<li class="nav-item">  
  <a class="nav-link text-dark" asp-area="Identity"  
    asp-page="/Account/Register">Register</a>  
</li>  
<li class="nav-item">  
  <a class="nav-link text-dark" asp-area="Identity"  
    asp-page="/Account/Login">Login</a>  
</li>
```

Identity Framework

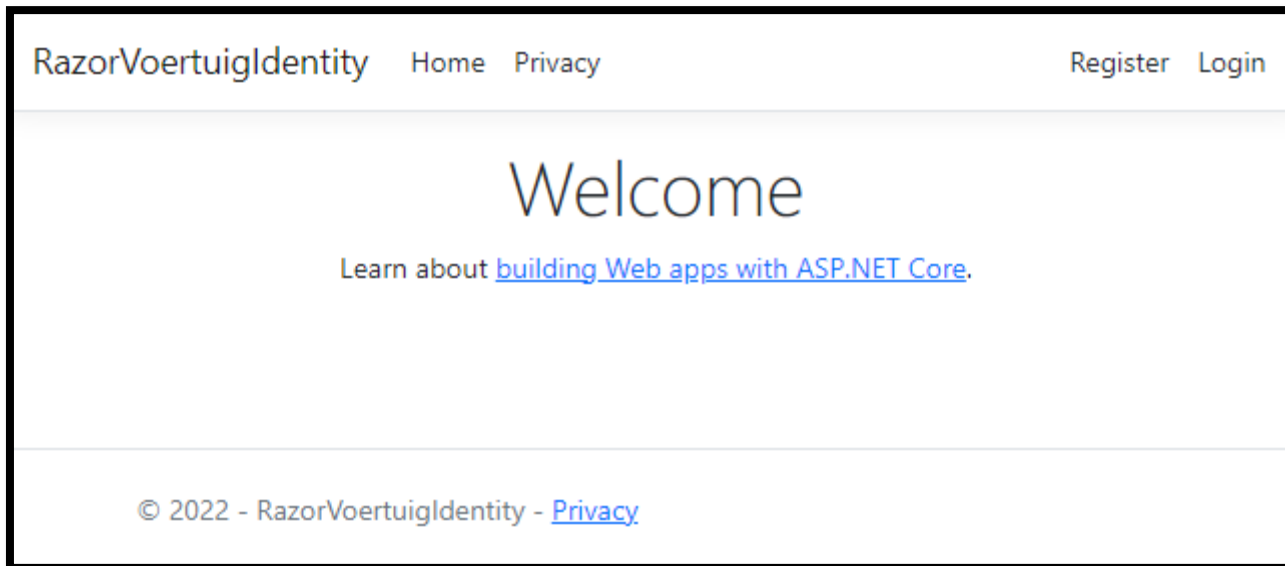
- Views/Shared
 - _LoginPartial.cshtml

@inject -> Dependency Injection in Views

```
@using Microsoft.AspNetCore.Identity
@inject SignInManager<IdentityUser> SignInManager
@inject UserManager<IdentityUser> UserManager

<ul class="navbar-nav">
@if (SignInManager.IsSignedIn(User))
```

Identity Framework



Identity Framework - Register

A database operation failed while processing the request.

SQLException: Cannot open database "aspnet-RazorVoertuigIdentity-9781FC8A-3611-49C1-86D8-214723AFBE0" requested by the login. The login failed. Login failed for user 'PXLPERs\20007064'.

Applying existing migrations may resolve this issue

There are migrations that have not been applied to the following database(s):

ApplicationDbContext

- 0000000000000000_CreateIdentitySchema

Apply Migrations

In Visual Studio, you can use the Package Manager Console to apply pending migrations to the database:

PM> Update-Database

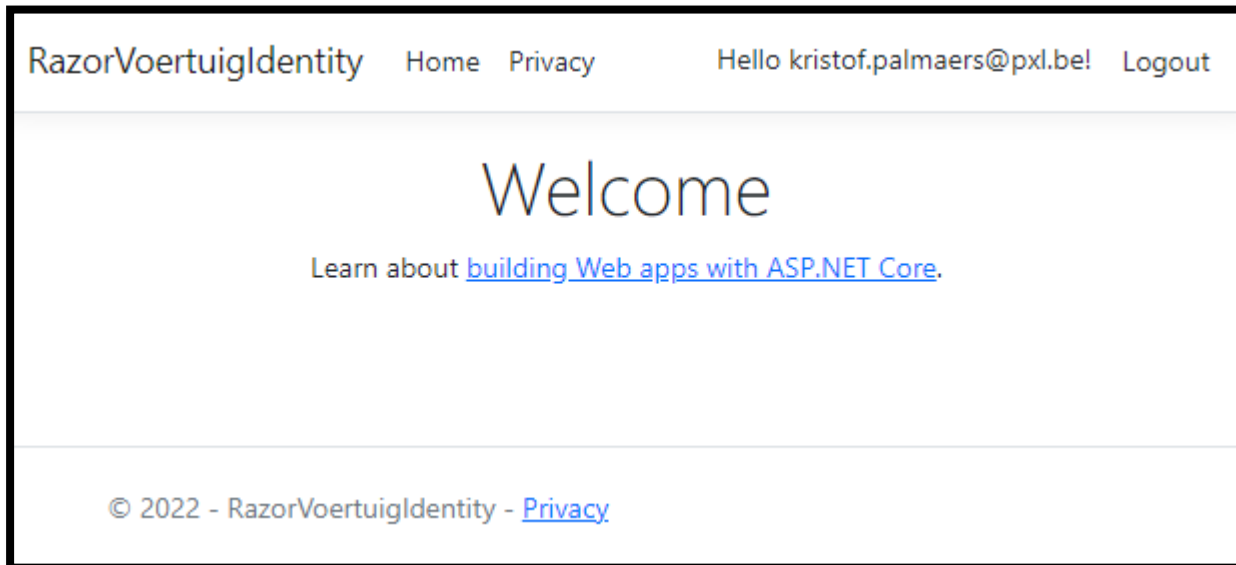
Alternatively, you can apply pending migrations from a command prompt at your project directory:

```
> dotnet ef database update
```



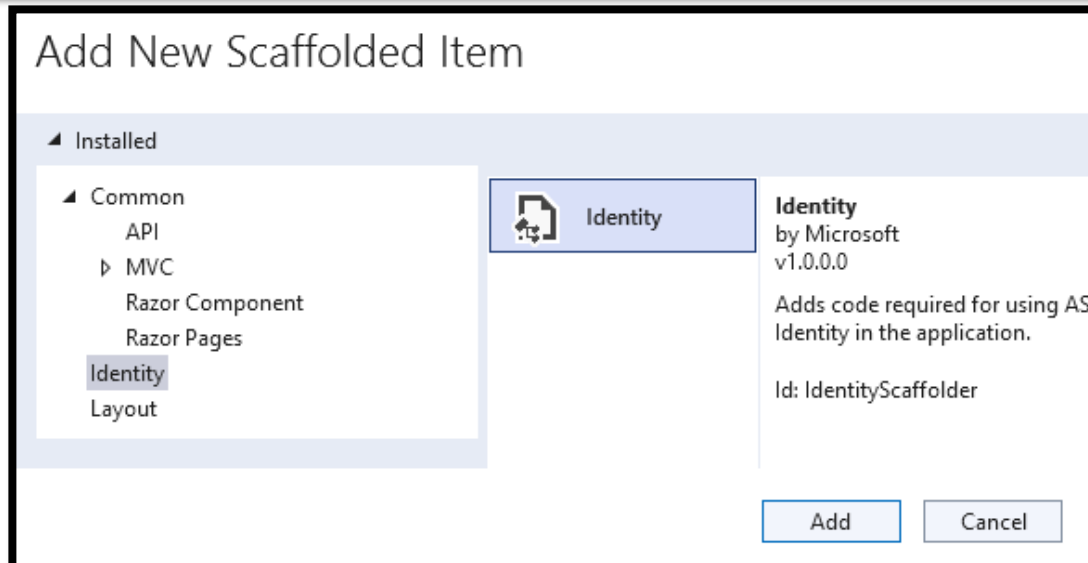
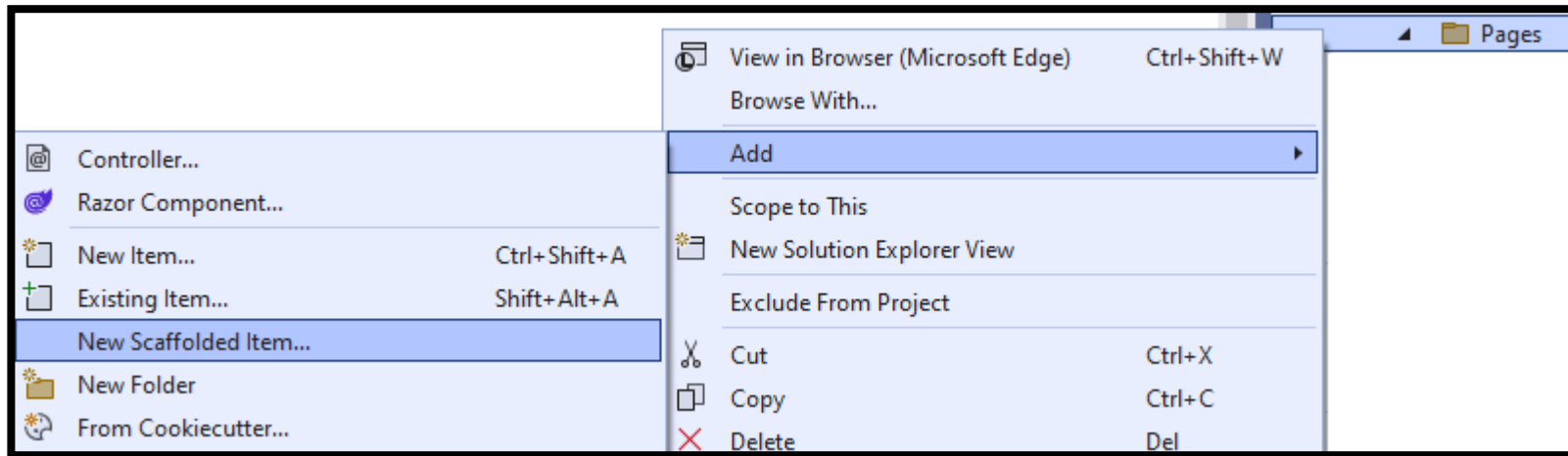
Identity Framework

- Register
- Login



Identity Framework

- Scaffold Identity Pages



Identity Framework

- Scaffold Identity Pages

×

Add Identity

Select an existing layout page, or specify a new one:
/Areas/Identity/Pages/Account/Manage/_Layout.cshtml
(Leave empty if it is set in a Razor _viewstart file)

...

☐ Override all files

Choose files to override

<input type="checkbox"/> Account\StatusMessage	<input type="checkbox"/> Account\AccessDenied	<input type="checkbox"/> Account\ConfirmEmail
<input type="checkbox"/> Account\ConfirmEmailChange	<input type="checkbox"/> Account\ExternalLogin	<input type="checkbox"/> Account\ForgotPassword
<input type="checkbox"/> Account\ForgotPasswordConfirmation	<input type="checkbox"/> Account\Lockout	<input checked="" type="checkbox"/> Account>Login
<input type="checkbox"/> Account>LoginWith2fa	<input type="checkbox"/> Account>LoginWithRecoveryCode	<input type="checkbox"/> Account\Logout
<input type="checkbox"/> Account\Manage\Layout	<input type="checkbox"/> Account\Manage\ManageNav	<input type="checkbox"/> Account\Manage\StatusMessage
<input type="checkbox"/> Account\Manage\ChangePassword	<input type="checkbox"/> Account\Manage\DeletePersonalData	<input type="checkbox"/> Account\Manage\Disable2fa
<input type="checkbox"/> Account\Manage\DownloadPersonalData	<input type="checkbox"/> Account\Manage\Email	<input type="checkbox"/> Account\Manage\EnableAuthenticator
<input type="checkbox"/> Account\Manage\ExternalLogins	<input type="checkbox"/> Account\Manage\GenerateRecoveryCodes	<input type="checkbox"/> Account\Manage\Index
<input type="checkbox"/> Account\Manage\PersonalData	<input type="checkbox"/> Account\Manage\ResetAuthenticator	<input type="checkbox"/> Account\Manage\SetPassword
<input type="checkbox"/> Account\Manage\ShowRecoveryCodes	<input type="checkbox"/> Account\Manage\TwoFactorAuthentication	<input checked="" type="checkbox"/> Account\Register
<input type="checkbox"/> Account\RegisterConfirmation	<input type="checkbox"/> Account\ResendEmailConfirmation	<input type="checkbox"/> Account\ResetPassword
<input type="checkbox"/> Account\ResetPasswordConfirmation		

Data context class +

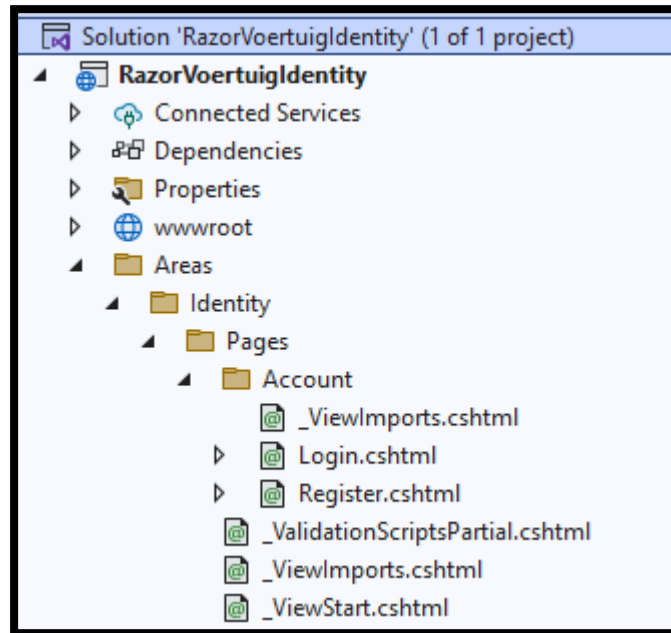
☐ Use SQLite instead of SQL Server

User class

AddCancel

Identity Framework

- Scaffold Identity Pages



Identity Framework

- Register action

```
var result = await _userManager.CreateAsync(  
    user, Input.Password);
```

- Signin action

```
await _signInManager.SignInAsync(user, isPersistent: false);
```

- Errors – Identity result

```
foreach (var error in result.Errors)  
{  
    ModelState.AddModelError(string.Empty, error.Description);  
}
```

Identity Framework

- login action

```
var result = await _signInManager.PasswordSignInAsync(  
    Input.Email,  
    Input.Password,  
    Input.RememberMe,  
    lockoutOnFailure: false);
```

- Errors – Identity result

```
foreach (var error in result.Errors)  
{  
    ModelState.AddModelError(string.Empty, error.Description);  
}
```

ASP.NET CORE

MVC Web Application
MVCVoertuig

MVCVoertuig - Identity

- Appsettings.json
 - MVCVoertuigb
 - Package manager console
 - Update database (nieuwe db)
- Nuget Package Manager - Installed
 - EntityFrameworkCore.SqlServer
 - EntityFrameworkCore.Tools

- Nuget Package Manager - Browse
 - Identity.EntityFrameworkCore



Microsoft.AspNetCore.Identity.EntityFrameworkCore ✓ by Microsoft, 66M downloads
ASP.NET Core Identity provider that uses Entity Framework Core.



MVCVoertuig - Identity

Program.cs

```
builder.Services.AddControllersWithViews();  
var connectionString =  
builder.Configuration.GetConnectionString("VoertuigConnection");  
builder.Services.AddDbContext<VoertuigDbContext>(  
    options => options.UseSqlServer(connectionString));  
builder.Services.AddIdentity<IdentityUser, IdentityRole>()  
    .AddEntityFrameworkStores<VoertuigDbContext>();
```

MVCVoertuig - Identity

Program.cs

...

```
app.UseAuthentication();  
app.UseAuthorization();
```

...

MVCVoertuig - Identity

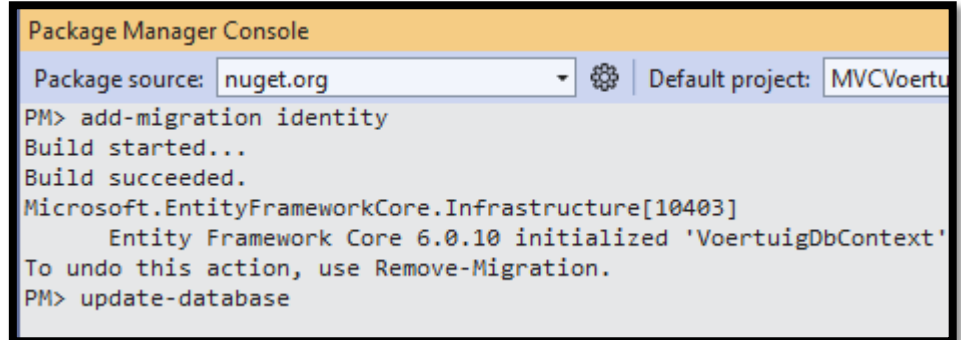
Data/VoertuigDbContext

```
public class VoertuigDbContext : IdentityDbContext
{
    ...
}
```


MVCVoertuig - Identity

Nuget Package Manager Console

Add-migration identity
Update-database

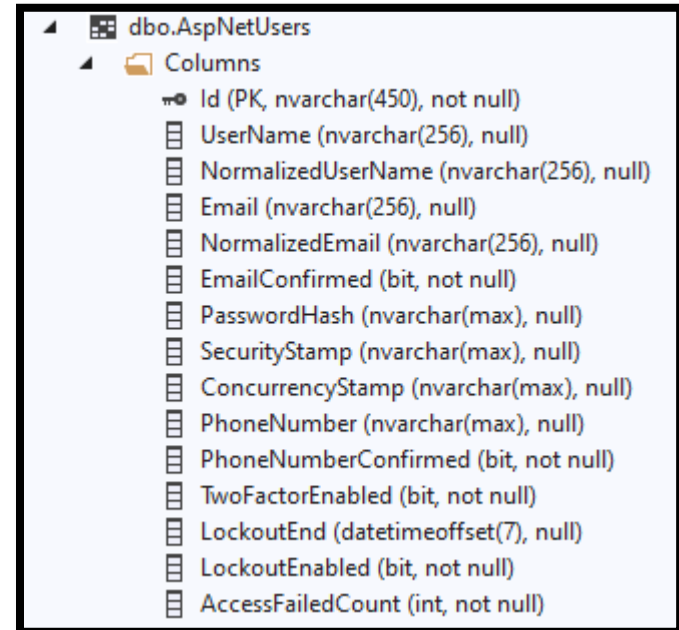
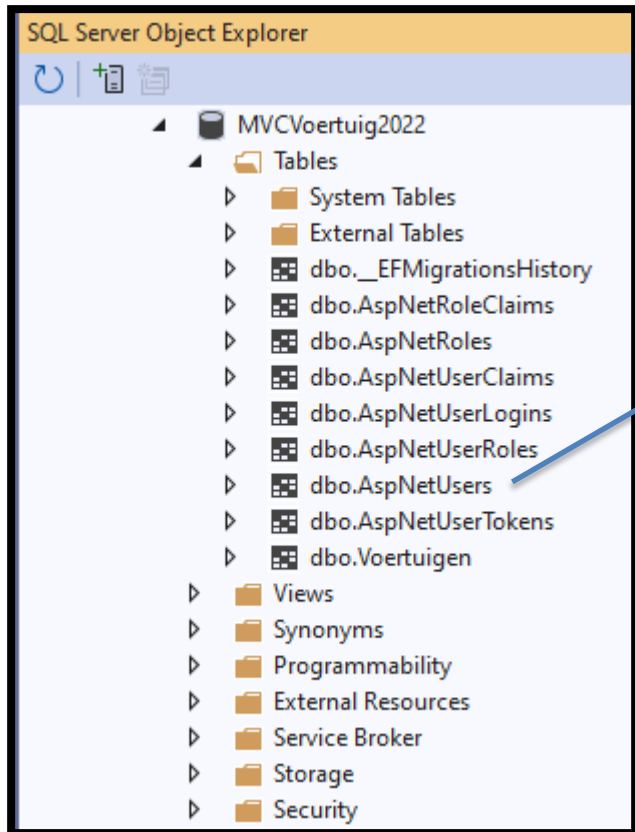


```
Package Manager Console
Package source: nuget.org
Default project: MVCVoertuig
PM> add-migration identity
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
    Entity Framework Core 6.0.10 initialized 'VoertuigDbContext'
To undo this action, use Remove-Migration.
PM> update-database
```

Error: Voertuigen table exists already in database

Drop-database
Update-database

MVCVoertuig - Identity



Identity Framework - Authentication

- Controllers - VoertuigenController

```
namespace MVCVoertuig.Controllers
{
    [Authorize]
    public class VoertuigController : Controller
```

Identity Framework - Authentication

```
namespace MVCVoertuig.Controllers
{
    [Authorize]
    public class VoertuigenController : Controller
```

An unhandled exception occurred while processing the request.

InvalidOperationException: No authenticationScheme was specified, and there was no DefaultChallengeScheme found. The default schemes can be set using either `AddAuthentication(string defaultScheme)` or `AddAuthentication(Action<AuthenticationOptions> configureOptions)`.

Microsoft.AspNetCore.Authentication.AuthenticationService.ChallengeAsync(HttpContext context, string scheme, AuthenticationProperties properties)

Stack Query Cookies Headers Routing

InvalidOperationException: No authenticationScheme was specified, and there was no DefaultChallengeScheme found. The default schemes can be set using either `AddAuthentication(string defaultScheme)` or `AddAuthentication(Action<AuthenticationOptions> configureOptions)`.

MVCVoertuig - Identity

- Add folder ViewModels in Models Folder
 - Add class LoginViewModel

```
using System.ComponentModel.DataAnnotations;

namespace MVCVoertuig.Models.ViewModels
{
    public class LoginViewModel
    {
        public int Id { get; set; }
        [Required]
        public string? Email { get; set; }
        [Required]
        public string? Password { get; set; }
    }
}
```

MVCVoertuig - Identity

Identity Framework – AccountController

- Controllers – Add empty MVC Controller
 - AccountController

Identity Framework – AccountController

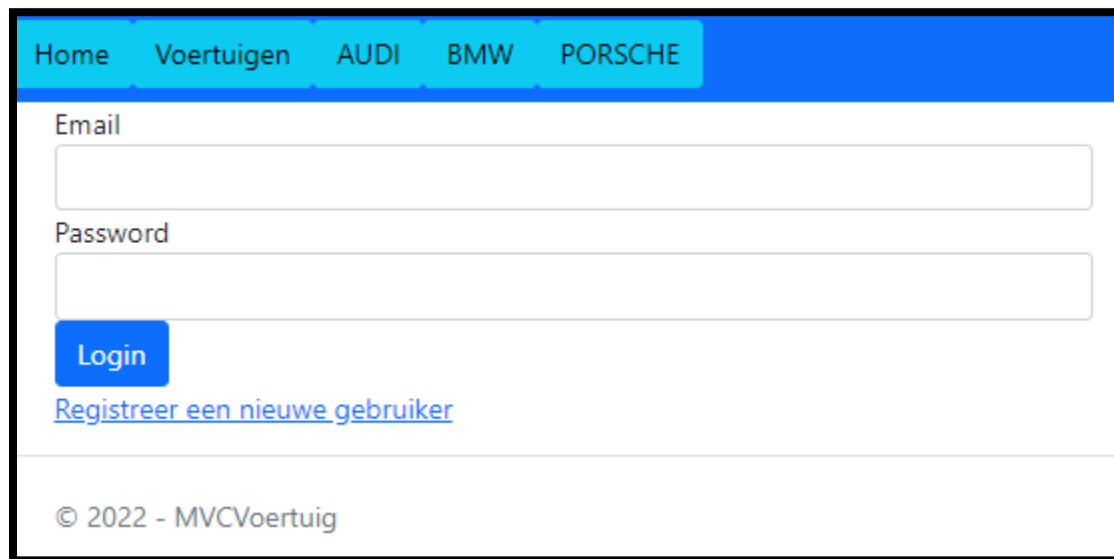
- AccountController - Login

```
namespace MVCVoertuig.Controllers
{
    public class AccountController : Controller
    {
        #region login
        [HttpGet]
        public IActionResult Login()
        {
            return View();
        }
        [HttpPost]
        public IActionResult Login(LoginViewModel user)
        {
            return View();
        }
        #endregion
    }
}
```

Identity Framework – Views/Account/Login

```
@model MVCVoertuig.Models.ViewModels.LoginViewModel
<div class="row">
    <div class="col-md-4">
        <form asp-action="Login">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Email" class="control-label"></label>
                <input asp-for="Email" class="form-control" />
                <span asp-validation-for="Email" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Password" class="control-label"></label>
                <input asp-for="Password" class="form-control" />
                <span asp-validation-for="Password" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Login" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
<div>
    <a asp-action="Register">Registreer een nieuwe gebruiker</a>
</div>
```


Identity Framework – Views/Account/Login



The screenshot shows a web application interface for login. At the top, there is a blue navigation bar with five items: 'Home', 'Voertuigen', 'AUDI', 'BMW', and 'PORSCHÉ'. Below the navigation bar, the main content area has a white background. It contains two input fields: 'Email' and 'Password'. Below the 'Password' field is a blue 'Login' button. Under the 'Login' button is a blue hyperlink that reads 'Registreer een nieuwe gebruiker'. At the bottom of the page, there is a footer with the text '© 2022 - MVCVoertuig'.

Home Voertuigen AUDI BMW PORSCHÉ

Email

Password

Login

[Registreer een nieuwe gebruiker](#)

© 2022 - MVCVoertuig

Identity Framework – AccountController

- AccountController - Register

```
public class AccountController : Controller
{
    #region login
    ...
    #endregion
    #region register
    [HttpGet]
    public IActionResult Register()
    {
        return View();
    }
    [HttpPost]
    public IActionResult Register(LoginViewModel user)
    {
        return View();
    }
    #endregion
}
```

Identity Framework – Views/Account/Register

```
@model MVCVoertuig.Models.ViewModels.LoginViewModel
<div class="row">
  <div class="col-md-4">
    <form asp-action="Register">
      <div asp-validation-summary="ModelOnly" class="text-danger"></div>
      <div class="form-group">
        <label asp-for="Email" class="control-label"></label>
        <input asp-for="Email" class="form-control" />
        <span asp-validation-for="Email" class="text-danger"></span>
      </div>
      <div class="form-group">
        <label asp-for="Password" class="control-label"></label>
        <input asp-for="Password" class="form-control" />
        <span asp-validation-for="Password" class="text-danger"></span>
      </div>
      <div class="form-group">
        <input type="submit" value="Register" class="btn btn-primary" />
      </div>
    </form>
  </div>
</div>
```

Identity Framework - Overview

Solution explorer

- Program.cs
 - AddIdentity
 - IdentityUser
 - IdentityRole
- Controllers
 - AccountController
- Models/ViewModels
 - LoginViewModel
- Views
 - Views/Account
 - Login.cshtml
 - Register.cshtml

