NAC Database uses .NetCore Identity for Authentication and Authorization

It uses Roles based Authorization rather than Claims.

Controllers\

AccountController

ApplicationRoleController

UsersController

Controllers\Datatables

DatatablesController

Views:

Views\Accounts

AccessDenied.cshtml

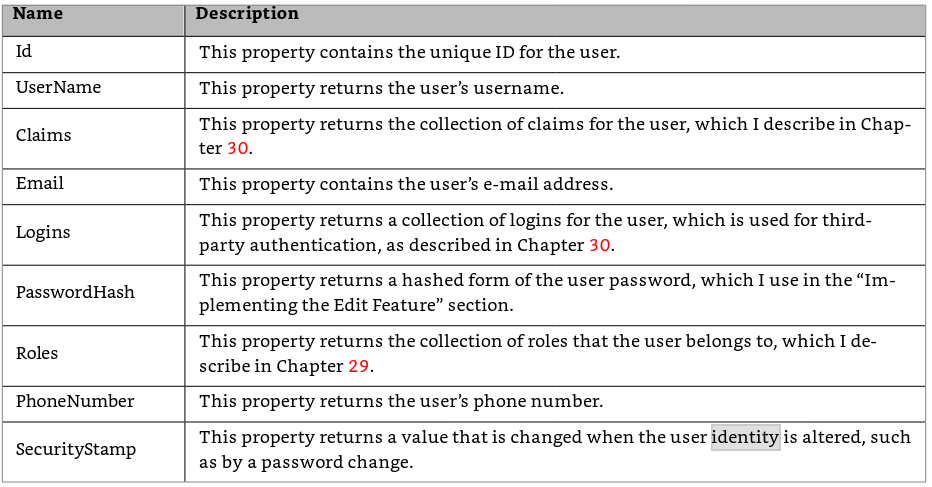
Login.cshtml

Register.cshtml

Models

**ApplicationUser.cs** derived from **IdentityUser** Models\ApplicationUser.cs

**IdentityUser** class has following defined properties:



Additional properties added into **ApplicationUser** class**:**

**LastName**

**FirstName**

**UserName** (generated from first character of FirstName and LastName)

**FullName**

**ApplicationUserRole.cs**

Models/AccountViewModels

ExternalLoginViewModel.cs

ForgotPasswordViewModel.cs

LoginViewModel.cs

RegisterViewModel.cs

ResetPasswordViewModel.cs

Identity classes

UserManager<TUser>

RoleManager<TRole> Provides the API for managing roles in Asp.Net Core Identity

SignInManager<TUser> Provides API for user sign in

Identity adds following tables to the **NWVSData** database:

dbo.AspNetRoleClaims

dbo.AspNetRoles

dbo.AspNetUserClaims

dbo.AspNetUserLogins

dbo.AspNetUserRoles

dbo.AspNetUsers

dbo.AspNetUserTokens

Identity uses the following DbContextclass: \Data\ApplicationDbContext.cs

class ApplicationDbContext : IdentityDbContext<ApplicationUser, ApplicationRole, string>

The **Startup.cs** class is defined as a partial class composed of **Startup.Pdf.cs** and **Startup.Roles.cs**

In **Startup.cs** **ConfigureServices** method Add the Identy services:

services.AddIdentity<ApplicationUser, ApplicationRole>()

.AddEntityFrameworkStores<ApplicationDbContext>()

.AddDefaultTokenProviders();

In **Configure** method add

app.UseAuthentication();

before the app.UseMVC method

Users are assigned to roles:

|  |  |
| --- | --- |
| **Role** | **Description** |
| Read Role | User can read any information in database. By default added to each newly created user |
| Create Role | User with this role can add new items to database |
| Update Role | User with this role can edit existing items in database |
| Delete Role | User with this role can remove items from database. Use with caution! |
| Admin Role | User with this role can assign roles and edit roles for other users. They can also change any information in database in any way they want |

These roles are setup in the **Startup.cs** Configure method

await CreateRoles(app);

await CreateAnonymousRole(app);

await CreateReportingRole(app);

These call methods on the partial class Startup.Roles

After Authenication users Authorization to the Action methods of controllers is determined by attributes on those controllers and/or methods.

[Authorize]

[Authorize(Roles = "Read Role, Admin Role")]