

WPF Login en Navigatie



```
<Page x:Class="WpfAdmin.Pages.LoginPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="LoginPage">

    <Grid Background="White">
        <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
            <TextBlock Text="Gebruikersnaam" Margin="0,5"/>
            <TextBox x:Name="txtGebruikersnaam" Width="200" Margin="0,5"/>

            <TextBlock Text="Wachtwoord" Margin="0,5"/>
            <PasswordBox x:Name="txtWachtwoord" Width="200" Margin="0,5"/>
            <Button x:Name="btnLogin" Content="Inloggen" Width="100" Margin="0,10"
Click="btnLogin_Click"/>
        </StackPanel>
    </Grid>
</Page>

<Window x:Class="WpfAdmin.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="WPF Admin" Height="600" Width="900">

    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="200" />
            <ColumnDefinition Width="*" />
        </Grid.ColumnDefinitions>

        <!-- Sidebar -->
        <StackPanel x:Name="NavPanel" Background="#f0f0f0" Grid.Column="0" Visibility="Collapsed">
            <Button Name="btnPersonen" Content="👤 Personen" Margin="10"
Click="BtnPersonen_Click" />
            <Button Name="btnUitloggen" Content="🚪 Uitloggen" Margin="10"
Click="BtnUitloggen_Click" />
        </StackPanel>

        <!-- Main content -->
        <Frame Name="MainFrame" Grid.Column="1" NavigationUIVisibility="Hidden" />
    </Grid>
</Window>

using System.Windows;
using WpfAdmin.Pages;

namespace WpfAdmin
```

```
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();

            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }

        public void LoginGeslaagd()
        {
            NavPanel.Visibility = Visibility.Visible;

            // Na login navigeren we eerst naar een lege pagina of welkom pagina
            MainFrame.Content = null; // Of maak een welkomspagina als je wil
        }

        private void BtnPersonen_Click(object sender, RoutedEventArgs e)
        {
            MainFrame.Navigate(new PersoonBewerken());
        }

        private void BtnUitloggen_Click(object sender, RoutedEventArgs e)
        {
            NavPanel.Visibility = Visibility.Collapsed;
            MainFrame.Navigate(new LoginPage(this));
        }
    }
}

<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Persoon Bewerken" Margin="10">

    <StackPanel Width="300" HorizontalAlignment="Center" VerticalAlignment="Top">
        <Label Content="Voornaam:"/>
        <TextBox Name="VoornaamBox" Margin="0,0,0,10"/>

        <Label Content="Achternaam:"/>
        <TextBox Name="AchternaamBox" Margin="0,0,0,10"/>

        <Label Content="Email:"/>
        <TextBox Name="EmailBox" Margin="0,0,0,10"/>

        <Label Content="Wachtwoord:"/>
        <PasswordBox Name="WachtwoordBox" Margin="0,0,0,10"/>

        <Label Content="Rol:"/>
        <ComboBox Name="RolBox" Margin="0,0,0,10">
```

```

        <ComboBoxItem Content="Admin"/>
        <ComboBoxItem Content="User"/>
        <ComboBoxItem Content="Manager"/>
    </ComboBox>

    <Button Content="Opslaan" Click="BtnOpslaan_Click" Width="100"
HorizontalAlignment="Center"/>
</StackPanel>
</Page>
using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
        }

        private void BtnOpslaan_Click(object sender, RoutedEventArgs e)
        {
            Person persoon = new Person
            {
                Voornaam = VoornaamBox.Text,
                Achternaam = AchternaamBox.Text,
                Email = EmailBox.Text,
                Wachtwoord = WachtwoordBox.Password,
                Rol = (RolBox.SelectedItem as ComboBoxItem)?.Content.ToString()
            };

            // Nu tijdelijk MessageBox
            MessageBox.Show("Persoon opgeslagen:\n" +
                $"{persoon.Voornaam} {persoon.Achternaam} ({persoon.Rol})");

            // Terug naar overzichtspagina (hier: PersonenOverzichtPage of leeg)
            // this.NavigationService.Navigate(new PersonenOverzichtPage());
            this.NavigationService.GoBack(); // Of iets anders
        }
    }

    // Dummy Person class als voorbeeld, vervang met jouw echte model
    public class Person
    {
        public string Voornaam { get; set; }
        public string Achternaam { get; set; }
        public string Email { get; set; }
        public string Wachtwoord { get; set; }
        public string Rol { get; set; }
    }
}

```

```
}
}
```

Je hebt al een goede basis! Ik zie dat je `LoginPage` en `MainWindow` hebt en ook `PersoonBewerken`. Wat er nog ontbreekt is de volledige `LoginPage` code-behind met de logincontrole en het doorgeven van succes aan `MainWindow`. Ook is er een kleine inconsistentie bij het navigeren naar `PersoonBewerken` vanuit het menu: dat moet waarschijnlijk een overzichtspagina zijn (`PersonenOverzichtPage`) en niet direct de bewerkpagina.

Hieronder een volledige, werkende en samenhangende versie met:

- `LoginPage.xaml` + `LoginPage.xaml.cs` met logincontrole.
- `MainWindow.xaml` + `MainWindow.xaml.cs` met navigatie en zichtbaarheid sidebar.
- `PersoonBewerken.xaml` + `PersoonBewerken.xaml.cs` met opslaan.
- Toevoeging van een simpele `PersonenOverzichtPage` als overzichtspagina.
- Correcte navigatie vanuit sidebar.

1. LoginPage.xaml (jij had al)

xml

```
<Page x:Class="WpfAdmin.Pages.LoginPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title=""
```



ik wil dat de gegevens van mijn sql companies vertoond word bij mijn persoon bewerken xaml--
Yasser El Aouati

```
-- Create Database EP3_q1720602;
```

```
CREATE DATABASE EP3_q1720602;
USE EP3_q1720602;
```

```
--Aanmaken van tabel ActiviteitTypes
```

```
CREATE TABLE ActiviteitTypes (
    id INT PRIMARY KEY IDENTITY,
    type VARCHAR(255) NOT NULL UNIQUE
);
```

```
--Aanmaken van tabel Personen
```

```
CREATE TABLE Personen (
    id INT PRIMARY KEY IDENTITY,
    voornaam VARCHAR(255) NULL,
    familienaam VARCHAR(255) NOT NULL,
    islid BIT NOT NULL DEFAULT 0
```

```
);
--Aanmaken van tabel Activiteiten

CREATE TABLE Activiteiten (
  id INT PRIMARY KEY IDENTITY,
  naam VARCHAR(255) NOT NULL,
  idtype INT NOT NULL,
  prijs DECIMAL(10, 2) NULL CHECK (prijs > 0),
  prijslid DECIMAL(10, 2) NULL CHECK (prijslid < prijs OR prijslid IS NULL),
  FOREIGN KEY (idtype) REFERENCES ActiviteitTypes(id),
  datum DATE NOT NULL,
  maxdeelnemers INT NULL,
  CHECK (datum > GETDATE()),
  CHECK (maxdeelnemers IS NULL OR maxdeelnemers >= 10)

```

```
);
--Aanmaken van tabel Inschrijvingen

CREATE TABLE Inschrijvingen (
  id INT PRIMARY KEY IDENTITY,
  idpersoon INT NOT NULL,
  idactiviteit INT NOT NULL,
  deelnemers INT NOT NULL CHECK (deelnemers > 0),
  datum DATE NOT NULL DEFAULT GETDATE(),
  status CHAR(1) NOT NULL CHECK (status IN ('I', 'W')),
  FOREIGN KEY (idpersoon) REFERENCES Personen(id),
  FOREIGN KEY (idactiviteit) REFERENCES Activiteiten(id)
);

```

-- Mockdata plaatsen in tabel personen

```
INSERT INTO Personen (voornaam, familienaam, islid)
VALUES ('Yasser', 'El Aouati', 1),
       ('Adam', 'Amri', 0),
       ('Matheo', 'De Bakker', 0);

```

-- Mockdata plaatsen in tabel Activiteittypes

```
INSERT INTO ActiviteitTypes (type)
VALUES ('Film'),
       ('Wandeling'),
       ('Vergadering'),
       ('Cursus'),
       ('Fietstocht');
GO

```

-- aanmaken van de functie udfTeBetalen

```
CREATE FUNCTION udfTeBetalen (@inschrijvingID INT)
RETURNS DECIMAL(10, 2)

```

```

AS
BEGIN
    DECLARE @prijs DECIMAL(10, 2);
    DECLARE @prijslid DECIMAL(10, 2);
    DECLARE @deelnemers INT;
    DECLARE @islid BIT;

    SELECT @prijs = A.prijs, @prijslid = A.prijslid, @deelnemers = I.deelnemers, @islid = P.islid
    FROM Inschrijvingen I
    JOIN Activiteiten A ON I.idactiviteit = A.id
    JOIN Personen P ON I.idpersoon = P.id
    WHERE I.id = @inschrijvingID;

    IF @prijs IS NULL
        RETURN 0;
    IF @inschrijvingID IS NULL
        RETURN 1;
    IF @islid = 1
        RETURN @prijslid * @deelnemers;
    ELSE
        RETURN @prijs * @deelnemers;
END;
GO
-- aanmaken van de functie udfInschrijvingAantal

CREATE FUNCTION udfInschrijvingAantal (@activiteitID INT, @type INT)
RETURNS INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Activiteiten WHERE id = @activiteitID)
        RETURN -1;

    IF @type = 0
        RETURN (SELECT SUM(deelnemers) FROM Inschrijvingen WHERE idactiviteit = @activiteitID AND
status = 'I');
    ELSE IF @type = 1
        RETURN (SELECT maxdeelnemers - SUM(deelnemers)
        FROM Activiteiten A
        JOIN Inschrijvingen I ON A.id = I.idactiviteit
        WHERE A.id = @activiteitID AND I.status = 'I');
    RETURN NULL;
END;
GO

-- Aanmaken van view vewTeBetalen

CREATE VIEW vewTeBetalen AS
SELECT I.id, N.naam, F.familienaam, dbo.udfTeBetalen(I.id) AS teBetalen
FROM Inschrijvingen I
JOIN Activiteiten A ON I.idactiviteit = A.id
JOIN Personen P ON I.idpersoon = P.id;

```

GO

-- Aanmaken van procedure sprActiviteitToevoegen

CREATE PROCEDURE sprActiviteitToevoegen

@naam VARCHAR(255),
@idtype INT,
@datum DATE,
@maxdeelnemers INT,
@prijs DECIMAL(10, 2),
@prijslid DECIMAL(10, 2)

AS

BEGIN

IF @naam IS NULL OR LEN(@naam) = 0 RETURN 1;
IF NOT EXISTS (SELECT 1 FROM ActiviteitTypes WHERE id = @idtype) RETURN 2;
IF @datum <= GETDATE() RETURN 3;
IF @maxdeelnemers IS NOT NULL AND @maxdeelnemers < 10 RETURN 4;
IF @prijs IS NOT NULL AND @prijs < 1 RETURN 5;
IF @prijslid IS NOT NULL AND @prijslid >= @prijs RETURN 6;

INSERT INTO Activiteiten (naam, idtype, datum, maxdeelnemers, prijs, prijslid)
VALUES (@naam, @idtype, @datum, @maxdeelnemers, @prijs, @prijslid);

END;

GO

-- Aanmaken van procedure sprInschrijvingToevoegen

CREATE PROCEDURE sprInschrijvingToevoegen

@idpersoon INT,
@idactiviteit INT,
@deelnemers INT

AS

BEGIN

DECLARE @maxdeelnemers INT;
DECLARE @inschrijvingen INT;

SELECT @maxdeelnemers = maxdeelnemers FROM Activiteiten WHERE id = @idactiviteit;
SELECT @inschrijvingen = dbo.udfInschrijvingAantal(@idactiviteit, 1);

IF @maxdeelnemers IS NULL OR @inschrijvingen IS NULL OR @inschrijvingen >= @deelnemers
BEGIN

INSERT INTO Inschrijvingen (idpersoon, idactiviteit, deelnemers, status)
VALUES (@idpersoon, @idactiviteit, @deelnemers, 'I');
RETURN 0;

END

ELSE

BEGIN

INSERT INTO Inschrijvingen (idpersoon, idactiviteit, deelnemers, status)
VALUES (@idpersoon, @idactiviteit, @deelnemers, 'W');
RETURN 1;

END

END;

```
GO
-- aanmaken van de trigger udfTeBetalen

CREATE TRIGGER trgUpdateWachtrij
ON Inschrijvingen
AFTER DELETE
AS
BEGIN
    DECLARE @activiteitID INT, @vrijPlaatsen INT, @idWachtrij INT, @deelnemers INT;

    SELECT @activiteitID = deleted.idactiviteit FROM deleted;
    IF (SELECT datum FROM Activiteiten WHERE id = @activiteitID) <= GETDATE()
        RETURN;

    SELECT @vrijPlaatsen = dbo.udfInschrijvingAantal(@activiteitID, 1);

    DECLARE cWachtrij CURSOR FOR
        SELECT id, deelnemers FROM Inschrijvingen WHERE idactiviteit = @activiteitID AND status = 'W'
    ORDER BY datum;

    END;
GO
SELECT TOP (1000) [id]
    ,[name]
    ,[contact]
    ,[address]
    ,[zip]
    ,[city]
    ,[country]
    ,[phone]
    ,[email]
    ,[btw]
    ,[login]
    ,[password]
    ,[regdate]
    ,[acceptdate]
    ,[lastmodified]
    ,[status]
    ,[language]
    ,[logo]
    ,[nacecode_code]
FROM [BenchmarkDB].[dbo].[Companies]
```

Om gegevens uit jouw **Companies**-tabel uit SQL Server te tonen in je `PersoonBewerken.xaml` WPF-pagina, moet je eerst:

1. Een verbinding leggen met je `BenchmarkDB`.
2. De relevante gegevens ophalen uit de `Companies`-tabel.

3. De gegevens weergeven in bijvoorbeeld een `ComboBox`, `ListBox` of `DataGrid` op je `PersoonBewerken.xaml` pagina.

Stap-voor-stap aanpak

1. Voeg een nieuwe control toe aan je XAML

Bijvoorbeeld een `ComboBox` om bedrijven te tonen:

```
xml

<Label Content="Bedrijf:"/>
<ComboBox Name="CompaniesComboBox" Margin="0,0,0,10" DisplayMemberPath="name"/>
```

2. Maak een `Company`-modelklasse in C#

```
csharp

public class Company
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

3. Voeg een method toe om bedrijven op te halen in `PersoonBewerken.xaml.cs`

Zorg dat je `System.Data.SqlClient` of `Microsoft.Data.SqlClient` gebruikt.

```
csharp

using System.Collections.Generic;
using System.Data.SqlClient;

private void LoadCompanies()
{
    string connectionString = "Server=localhost;Database=BenchmarkDB;Trusted_Connection=True;";
    List<Company> companies = new List<Company>();

    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        conn.Open();
        string query = "SELECT id, name FROM Companies";

        using (SqlCommand cmd = new SqlCommand(query, conn))
        using (SqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                companies.Add(new Company
                {
                    Id = reader.GetInt32(0),
                    Name = reader.GetString(1)
                });
            }
        }
    }
}
```

```
CompaniesComboBox.ItemsSource = companies;
}
```

Opmerking: Pas `connectionString` aan indien jouw setup anders is (gebruikersnaam/wachtwoord of poort).

4. Roep `LoadCompanies()` op in de constructor van `PersoonBewerken`

```
csharp

public PersoonBewerken()
{
    InitializeComponent();
    LoadCompanies();
}
```

✓ Resultaat

In je `PersoonBewerken.xaml` zie je nu een lijst van bedrijven in een `ComboBox`, op basis van de gegevens uit jouw `BenchmarkDB.dbo.Companies`-tabel.

+ Uitbreiding (optioneel)

Wil je ook **bedrijf selecteren en opslaan bij een persoon**, dan moet je:

- Een `CompanyId` of `SelectedCompany` property toevoegen aan je `Person`-object.
- In `BtnOpslaan_Click` ook het geselecteerde bedrijf verwerken:

```
csharp

var selectedCompany = CompaniesComboBox.SelectedItem as Company;
if (selectedCompany != null)
{
    persoon.CompanyId = selectedCompany.Id;
}
```

Laat weten als je ook de database wilt aanpassen zodat `Personen` een `CompanyId` kunnen bevatten, en ik help je daarmee ook.



geef me de hele codes van alles zodat het werkt

Om de gegevens uit de **Companies**-tabel van je SQL Server-database weer te geven in de `PersoonBewerken.xaml`-pagina van je WPF-toepassing, volgen hier de volledige stappen en codevoorbeelden.

1. Voeg een `Company`-modelklasse toe

Maak een nieuwe klasse aan om de bedrijfsgegevens te representeren:

```
csharp

public class Company
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

2. Pas `PersoonBewerken.xaml` aan

Voeg een `ComboBox` toe aan je XAML om de bedrijvenlijst weer te geven:

```
xml

<Label Content="Bedrijf:"/>
<ComboBox Name="CompaniesComboBox" Margin="0,0,0,10"
    DisplayMemberPath="Name"
    SelectedValuePath="Id"/>
```

3. Pas `PersoonBewerken.xaml.cs` aan

Voeg de benodigde namespaces toe bovenaan het bestand:

```
csharp

using System.Collections.Generic;
using System.Data.SqlClient;
```

Voeg vervolgens de methode toe om de bedrijven op te halen en de `ComboBox` te vullen:

```
csharp

private void LoadCompanies()
{
    string connectionString = "Server=localhost;Database=BenchmarkDB;Trusted_Connection=True;";
    List<Company> companies = new List<Company>();

    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        conn.Open();
        string query = "SELECT id, name FROM Companies";

        using (SqlCommand cmd = new SqlCommand(query, conn))
        using (SqlDataReader reader = cmd.ExecuteReader())
        {
            while (reader.Read())
            {
                companies.Add(new Company
                {
                    Id = reader.GetInt32(0),
                    Name = reader.GetString(1)
                });
            }
        }

        CompaniesComboBox.ItemsSource = companies;
    }
}
```

Roep deze methode aan in de constructor van de `



geef me alles in aparte files zodat ik het direct kan kopiëren en geeft het ook de juiste benaming

Hier zijn alle bestanden die je nodig hebt om de gegevens uit de **Companies**-tabel correct te tonen op de `PersoonBewerken.xaml`-pagina in je WPF Admin-project. Je kunt de code direct kopiëren en in Visual Studio plakken. De bestanden zijn opgesplitst zoals gevraagd.

✓ Bestand 1: Company.cs

Locatie: `Models/Company.cs`

```
csharp

namespace WpfAdmin.Models
{
    public class Company
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

✓ Bestand 2: PersoonBewerken.xaml

Locatie: `Pages/PersoonBewerken.xaml`

```
xml

<Page x:Class="WpfAdmin.Pages.PersoonBewerken"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:local="clr-namespace:WpfAdmin.Pages"
      mc:Ignorable="d"
      Title="PersoonBewerken">

    <Grid Margin="20">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
        </Grid.RowDefinitions>

        <Label Content="Bedrijf:"/>
        <ComboBox x:Name="CompaniesComboBox"
                  Grid.Row="1"
                  Margin="0,5,0,0"
                  DisplayMemberPath="Name"
                  SelectedValuePath="Id"/>

    </Grid>
</Page>
```

✓ Bestand 3: PersoonBewerken.xaml.cs

Locatie: `Pages/PersoonBewerken.xaml.cs`

```
csharp

using System.Collections.Generic;
using System.Data.SqlClient;
using System.Windows.Controls;
using WpfAdmin.Models;

namespace WpfAdmin.Pages
{
    public partial class PersoonBewerken : Page
    {
        public PersoonBewerken()
        {
            InitializeComponent();
            LoadCompanies();
        }

        private void LoadCompanies()
        {
            string connectionString = "Server=localhost;Database=BenchmarkDB;Trusted_Connection=True;";
            List<Company> companies = new List<Company>();

            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                conn.Open();
                string query = "SELECT id, name FROM Companies";

                using (SqlCommand cmd = new SqlCommand(query, conn))
                using (SqlDataReader reader = cmd.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        companies.Add(new Company
                        {
                            Id = reader.GetInt32(0),
                            Name = reader.GetString(1)
                        });
                    }
                }

                CompaniesComboBox.ItemsSource = companies;
            }
        }
    }
}
```

Structuur in je project:

```
markdown

WpfAdmin
├── Models
│   └── Company.cs
└── Pages
    ├── PersoonBewerken.xaml
    └── PersoonBewerken.xaml.cs
```

Laat me gerust weten als je ook de selectie van een bedrijf aan een persoon wil koppelen, of als je dit in combinatie met een `Personen`-database wil opslaan.