

# Inlogpagina en Adminpanel



```
using System.Windows;

namespace WpfAdmin
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            MainFrame.Navigate(new Pages.LoginPage(this));
        }

        public void ShowAdminPanel()
        {
            MainFrame.Navigate(new Pages.AdminPage());
        }
    }
}

<Window x:Class="WpfAdmin.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="WpfAdmin" Height="450" Width="800">
    <Grid>
        <Frame x:Name="MainFrame" NavigationUIVisibility="Hidden" />
    </Grid>
</Window>using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class LoginPage : Page
    {
        private MainWindow _mainWindow;

        public LoginPage(MainWindow mainWindow)
        {
            InitializeComponent();
            _mainWindow = mainWindow;
        }
    }
}

<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" Height="300" Width="400">
  <StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">
    <TextBox x:Name="txtGebruikersnaam" Width="200" Margin="0,10"/>
    <PasswordBox x:Name="txtWachtwoord" Width="200" Margin="0,10"/>
    <Button Content="Inloggen" Width="100" Click="Login_Click"/>
  </StackPanel>
</Grid>
</Page>
```

ik wil dat als ik het programma run eerst de inlog page tevoorschijn komt dan als de gebruiker inlogt met de admin gegevens dan kan hij data creeren updaten en deleted en lezen

Oké, je wilt dus een WPF-applicatie met een inlogpagina waar een gebruiker zich kan aanmelden. Als de gebruiker correcte admin-gegevens invoert, moet hij doorgaan naar een admin-paneel waar data kan worden aangemaakt, bijgewerkt, verwijderd en gelezen.

Ik help je stap voor stap met de basis van zo'n flow:

## 1. Huidige setup is bijna goed:

- `MainWindow` toont bij opstart de `LoginPage`.
- `LoginPage` heeft de velden voor gebruikersnaam en wachtwoord, en een knop `Inloggen`.

## 2. Voeg in `LoginPage` de login-logica toe:

In `LoginPage.xaml.cs` voeg je een event handler toe voor de button click:

```
csharp

using System.Windows;
using System.Windows.Controls;

namespace WpfAdmin.Pages
{
    public partial class LoginPage : Page
    {
        private MainWindow _mainWindow;

        public LoginPage(MainWindow mainWindow)
        {
            InitializeComponent();
            _mainWindow = mainWindow;
        }

        private void Login_Click(object sender, RoutedEventArgs e)
        {
            string username = txtGebruikersnaam.Text;
            string password = txtWachtwoord.Password;

            // Eenvoudige admin credentials check (hardcoded)
            if (username == "admin" && password == "admin123")
            {
                // Login succesvol -> admin paneel tonen
                _mainWindow.ShowAdminPanel();
            }
            else
            {
                MessageBox.Show("Ongeldige gebruikersnaam of wachtwoord.", "Fout", MessageBoxButtonOk);
            }
        }
    }
}
```

```
}
}
```

### 3. Maak de `AdminPage`:

Voeg een eenvoudige `AdminPage` toe waar je straks CRUD-functionaliteit kunt toevoegen.

#### AdminPage.xaml:

```
xml

<Page x:Class="WpfAdmin.Pages.AdminPage"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      Title="AdminPage">
    <Grid Background="LightGray">
        <TextBlock Text="Welkom in het admin paneel"
                  VerticalAlignment="Center"
                  HorizontalAlignment="Center"
                  FontSize="24" FontWeight="Bold"/>
        <!-- Hier kan je straks je CRUD UI toevoegen -->
    </Grid>
</Page>
```

#### AdminPage.xaml.cs:

```
csharp

using System.Windows.Controls;

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

### 4. Resultaat:

- Bij het starten van de app zie je `LoginPage`.
- Als je `admin` / `admin123` invult en op inloggen drukt, navigeert het scherm naar `AdminPage`.
- Onjuiste gegevens geven een foutmelding.

### Wat kan je nu doen?

- AdminPage uitbreiden met een lijst (bv. een `DataGrid`) en knoppen om data te creëren, updaten, verwijderen en lezen.
- Data opslaan in een database, bestand of in-memory lijst.
- Login verbeteren met echte authenticatie.

Als je wil, kan ik je helpen met het opzetten van die CRUD-functies in `AdminPage`. Wil je dat?