

```
1 using System.Linq;
2 using System.Windows;
3 using System.Windows.Input;
4 using CommunityToolkit.Mvvm.ComponentModel;
5 using CommunityToolkit.Mvvm.Input;
6 using ShapeDrawer.Models;
7 using ShapeDrawer.Views;
8
9 namespace ShapeDrawer.ViewModels
10 {
11     internal class LoginViewModel : ObservableObject
12     {
13         public ICommand LoginCommand { get; }
14         public ICommand RegisterCommand { get; }
15
16         private string _username;
17         public string Username
18         {
19             get => _username;
20             set => SetProperty(ref _username, value);
21         }
22
23         private string _password;
24         public string Password
25         {
26             get => _password;
27             set => SetProperty(ref _password, value);
28         }
29
30         public bool LoginWasSuccessful { get; private set; } // Flag to track login success
31
32         public LoginViewModel()
33         {
34             LoginCommand = new RelayCommand(Login);
35             RegisterCommand = new RelayCommand(NavigateToRegister);
36         }
37
38         private void Login()
39         {
40             if (ValidateUser(Username, Password))
41             {
42                 LoginWasSuccessful = true; // Set the flag to true
43                 var mainWindow = new MainWindow();
44                 mainWindow.Show();
45
46                 // Close the login window
47                 Application.Current.Windows.OfType<Login>().FirstOrDefault()?.Close();
48             }
49         }
50     }
51 }
```

```
48         }
49         else
50         {
51             MessageBox.Show("Invalid username or password.");
52         }
53     }
54
55     private void NavigateToRegister()
56     {
57         // Open RegisterView
58         var registerView = new Register();
59         registerView.Show(); // Show the Register window first
60
61         // Close the login window after the Register window is shown
62         Application.Current.Windows.OfType<Login>().FirstOrDefault()?.Close();
63     }
64
65     private bool ValidateUser(string username, string password)
66     {
67         using (var context = new ShapeDrawerDbContext())
68         {
69             // Find the user by username
70             var user = context.Users.FirstOrDefault(u => u.Username == username);
71
72             if (user != null)
73             {
74                 // Verify the hashed password
75                 return BCrypt.Net.BCrypt.Verify(password, user.Password);
76             }
77
78             return false; // User not found
79         }
80     }
81 }
82 }
```