**DATINGAPP:**

DATABASE CONNECTIONS:

namespace test

{

public class Databaseconnection

{

public class dbconnection

{

// HUSK AT ÆNDR

private static readonly string \_conJ = @"Data Source=SKAB1-PC-07\JOAKIMFKK;Initial Catalog=DatingDB;Integrated Security=True";

private static readonly string \_conS= @"Data Source=SKAB1-PC-11\SIMONE;Initial Catalog=DatingDB;Integrated Security=True";

public static string ConnectionDB()

{

return \_conS;

}

}

}

}

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<configSections>

</configSections>

<connectionStrings>

<add name="ConnectionDB"

connectionString="Data Source=SKAB1-PC-11\SIMONE;Initial Catalog=DatingDB;Integrated Security=True"

providerName="System.Data.SqlClient" />

</connectionStrings>

<startup>

<supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />

</startup>

</configuration>

<!--Ctrl + k + c ctrl k + u-->

<!--connectiostring ="Data Source=SKAB1-PC-07\JOAKIMFKK;Initial Catalog=DatingDB;Integrated Security=True"-->

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Windows;

using static test.Databaseconnection;

namespace test

{

/// <summary>

/// Interaction logic for ChatLog.xaml

/// </summary>

public partial class ChatLog : Window

{

// Constructor

public ChatLog(int CurrentUserID, int CurrentChatterID, string YourUserName, string TheirUserName)

{

currentUserID = CurrentUserID;

currentChatterID = CurrentChatterID;

yourUserName = YourUserName;

theirUserName = TheirUserName;

InitializeComponent();

BindMessagesToTextBox();

chatBox.ScrollToEnd();

IsReadUpdate();

}

// Fields

private int currentUserID = 0;

private int currentChatterID = 0;

private string yourUserName = "";

private string theirUserName = "";

// Load chat-beskeder

private void BindMessagesToTextBox()

{

chatBox.Document.Blocks.Clear();

string searchQuery = $"SELECT senderID, messagetext FROM directMessages WHERE receiverID={currentUserID} "

+ $"AND senderID={currentChatterID} OR senderID={currentUserID} AND receiverID={currentChatterID}";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

for (int i = 0; i < dt.Rows.Count; i++)

{

string senderName = "";

if (dt.Rows[i]["senderID"].ToString() == currentUserID.ToString())

{

senderName = yourUserName;

}

else

{

senderName = theirUserName;

}

string sender = dt.Rows[i]["senderID"].ToString();

string textMessage = dt.Rows[i]["messagetext"].ToString();

chatBox.AppendText(senderName + ": ");

chatBox.AppendText(textMessage + "\n");

}

}

chatBox.ScrollToEnd();

}

public void SendMessage(object sender, RoutedEventArgs e)

{

if (txtMessage.Text.Trim() != "")

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

string message = txtMessage.Text.Trim();

string sqlQuery = $"INSERT INTO directMessages (senderID, receiverID,messageText) VALUES ({currentUserID}, {currentChatterID}, '{message}')";

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

cmd.ExecuteNonQuery();

sqlCon.Close();

}

txtMessage.Text = "";

MessageBox.Show("Beskeden er sendt.");

BindMessagesToTextBox();

}

else

{

MessageBox.Show("ingen text.");

}

}

private void IsReadUpdate()

{

string updateQuery = $"UPDATE directMessages SET isRead=1 WHERE senderID={currentChatterID} AND receiverID={currentUserID} ";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

//SqlDataAdapter sda = new SqlDataAdapter(updateQuery, sqlCon);

SqlCommand cmd = new SqlCommand(updateQuery, sqlCon);

sqlCon.Open();

cmd.ExecuteNonQuery();

sqlCon.Close();

}

}

}

}

--------

XAML:

Window x:Class="test.ChatLog"

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:test"

mc:Ignorable="d"

Title="ChatLog" Height="550" Width="450">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="0" />

<RowDefinition Height="150" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="11\*" />

<ColumnDefinition Width="115\*"/>

<ColumnDefinition Width="48\*"/>

<ColumnDefinition Width="auto" />

</Grid.ColumnDefinitions>

<TextBox x:Name="txtMessage" Grid.Row="2" MaxHeight="100" Height="100"

Margin="15,15,0,15" Grid.ColumnSpan="3"

TextWrapping="Wrap"

>

</TextBox>

<Button Grid.Row="2" Grid.Column="3" Height="75" Width="75"

Margin="10,39.8,10.2,40.4" FontSize="20"

Click="SendMessage">

Send

</Button>

<RichTextBox Grid.Row="0" Grid.ColumnSpan="4"

IsReadOnly="True" Margin="15,15,14.2,14.2"

x:Name="chatBox"

ScrollViewer.VerticalScrollBarVisibility="Auto"

MaxWidth="410"

>

</RichTextBox>

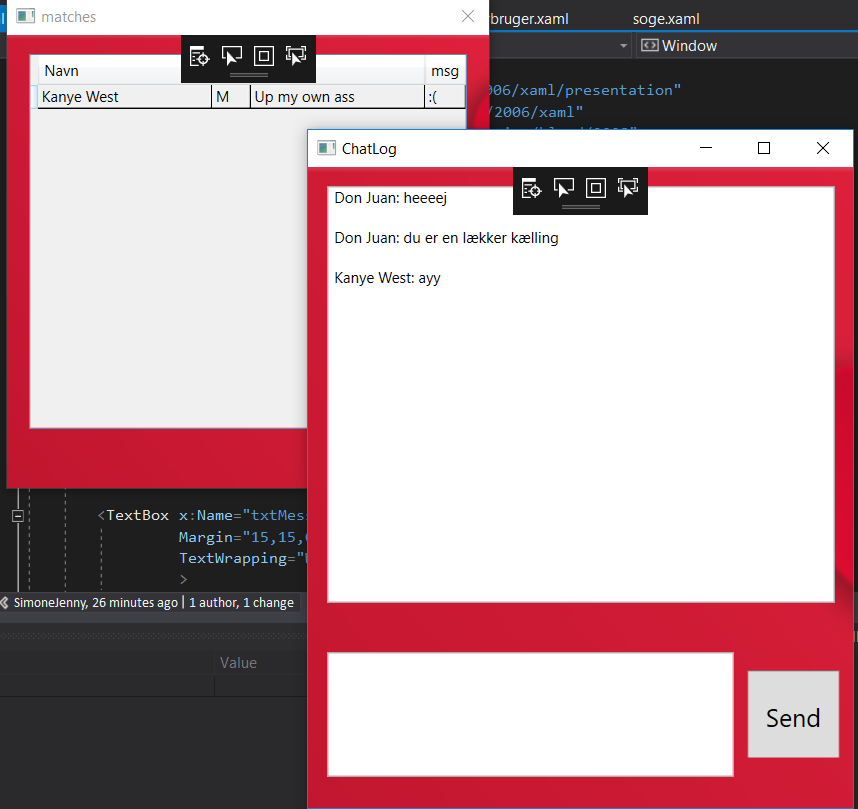
<Grid.Background>

<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill" />

</Grid.Background>

</Grid>

</Window>



---------

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Documents;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using static test.Databaseconnection;

namespace test

{

/// <summary>

/// Interaction logic for loginuser.xaml

/// </summary>

public partial class loginuser : Window

{

public loginuser(string receivedUserID)

{

currentUserID = Convert.ToInt32(receivedUserID);

InitializeComponent();

BindUserInfo();

}

private string yourFullName = "";

private int currentUserID = 0;

private string avatarName = "";

private void BindUserInfo()

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

string searchQueryAccount = $"SELECT \* FROM userAccount WHERE userID={currentUserID.ToString()}";

string searchQueryProfile = $"SELECT \* FROM userProfile WHERE userID={currentUserID}";

DataTable dtAccount = new DataTable();

SqlDataAdapter sdaAccount = new SqlDataAdapter(searchQueryAccount, sqlCon);

sdaAccount.Fill(dtAccount);

DataTable dtProfile = new DataTable();

SqlDataAdapter sdaProfile = new SqlDataAdapter(searchQueryProfile, sqlCon);

sdaProfile.Fill(dtProfile);

txtUsername.Content = dtAccount.Rows[0]["username"].ToString();

txtFullName.Content = dtProfile.Rows[0]["fullName"].ToString();

yourFullName = dtProfile.Rows[0]["fullName"].ToString();

FlowDocument fDoc = new FlowDocument();

fDoc.Blocks.Add(new Paragraph(new Run(dtProfile.Rows[0]["aboutMe"].ToString())));

avatarName = dtProfile.Rows[0]["avatar"].ToString();

txtAboutMe.Document = fDoc;

string fileName = dtProfile.Rows[0]["avatar"].ToString().Trim();

try

{

imgAvatar.Source = new BitmapImage(new Uri(GetFilePath()));

}

catch

{

imgAvatar.Source = new BitmapImage(new Uri(@"billede/ProfilBilleder/defaultCowboy.png", UriKind.Relative));

}

}

}

private string GetFilePath()

{

string filepath = System.Reflection.Assembly.GetEntryAssembly().Location;

string filepathtoimages = @"billede\ProfilBilleder\";

int index = filepath.LastIndexOf(@"\bin\") + 1;

return filepath.Substring(0, index) + filepathtoimages + avatarName;

}

public void opdater(object sender, RoutedEventArgs e)

{

Opdaterbruger up = new Opdaterbruger(currentUserID);

up.Show();

this.Close();

}

public void soge(object sender, RoutedEventArgs e)

{

soge up = new soge(currentUserID);

up.Show();

this.Close();

}

private void slet(object sender, RoutedEventArgs e)

{

sletbruger slet = new sletbruger();

slet.Show();

this.Close();

}

public void matches(object sender, RoutedEventArgs e)

{

matches mat = new matches(currentUserID, yourFullName);

mat.Show();

this.Close();

}

}

}

----

<Window x:Class="test.loginuser"

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:test"

mc:Ignorable="d"

Title="loginuser" Height="450" Width="800"

ResizeMode="NoResize"

>

<Grid>

<Button Margin="24,39,620.6,338" Click="opdater">Opdater dine oplysninger</Button>

<Button Margin="24,121,620.6,256" Click="slet">Slet bruger/konto</Button>

<Button Margin="24,209,620.6,168" Click="soge">Søg på medlemmer</Button>

<Button Margin="24,297,620.6,80" Click="matches" RenderTransformOrigin="0.501,0.5">Se dine Matchs</Button>

<Label x:Name="txtUsername" Content="{Binding username}" HorizontalAlignment="Left" Margin="300,39,0,0" VerticalAlignment="Top" Width="125"/>

<Label x:Name="txtFullName" Content="{Binding fullName}" HorizontalAlignment="Left" Margin="300,82,0,0" VerticalAlignment="Top"/>

<RichTextBox x:Name="txtAboutMe" IsReadOnly="True" HorizontalAlignment="Left" Height="209" Margin="300,131,0,0" VerticalAlignment="Top" Width="217">

</RichTextBox>

<Image x:Name="imgAvatar" HorizontalAlignment="Left" Height="175" Width="175" Margin="600,10,0,0" VerticalAlignment="Top"/>

<!--<TextBlock HorizontalAlignment="Left" Margin="550,270,0,0" TextWrapping="Wrap"

x:Name="testName"

Text="{Binding username}" VerticalAlignment="Top"

/>-->

<Grid.Background>

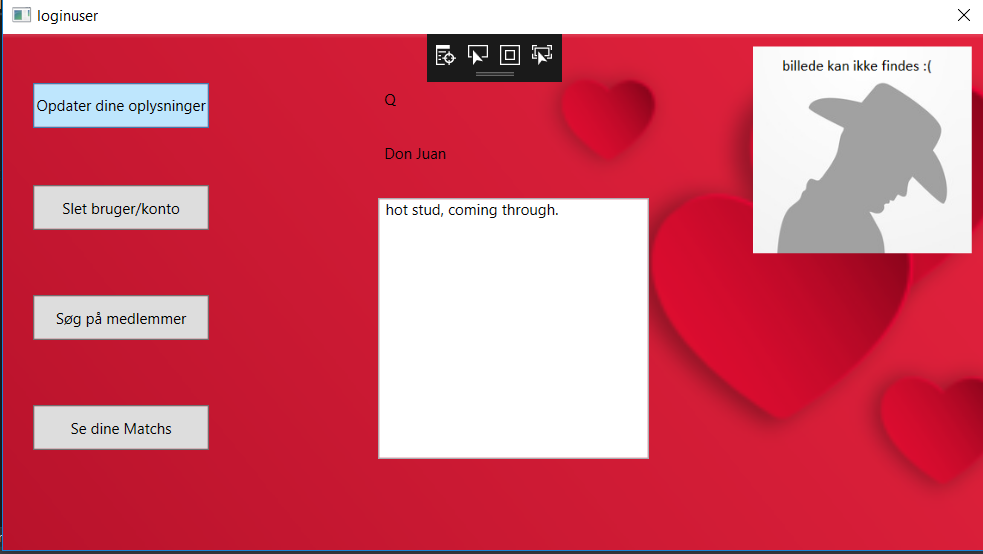
<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</Grid.Background>

</Grid>

</Window>



-----

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using test.Primary\_funkions;

namespace test

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void opretkontoo(object sender, RoutedEventArgs e)

{

opretkonto opretk = new test.opretkonto();

opretk.Show();

this.Close();

}

private void Loginuser(object sender, RoutedEventArgs e)

{

vmOpret.Loginuser(); //den kan finde loginuser via bindings. den kan findes i DOOPRET

}

}

}

---

<Window x:Class="test.MainWindow"

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:test"

xmlns:pfunc="clr-namespace:test.Primary\_funkions"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="600"

ResizeMode="NoResize"

>

<Window.DataContext>

<pfunc:DOOPRET x:Name="vmOpret" />

</Window.DataContext>

<Grid>

<Grid.Background>

<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</Grid.Background>

<Button Padding="5" Margin="35,303,334.6,43" BorderThickness="5" Click="opretkontoo">Opret konto</Button>

<Button Padding="5" Margin="329,303,53.6,43" BorderThickness="5" Click="Loginuser" RenderTransformOrigin="0.5,0.503">Log ind</Button>

<TextBlock FontSize="50" Margin="150,54,148.6,298" TextWrapping="WrapWithOverflow" Foreground="White">Dateme.dk</TextBlock>

<TextBox x:Name="txtBrugernavn" Text="{Binding Login,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

HorizontalAlignment="Left" Height="23" Margin="207,177,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="238"/>

<TextBox x:Name="txtPassword" Text="{Binding Adgangskode,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

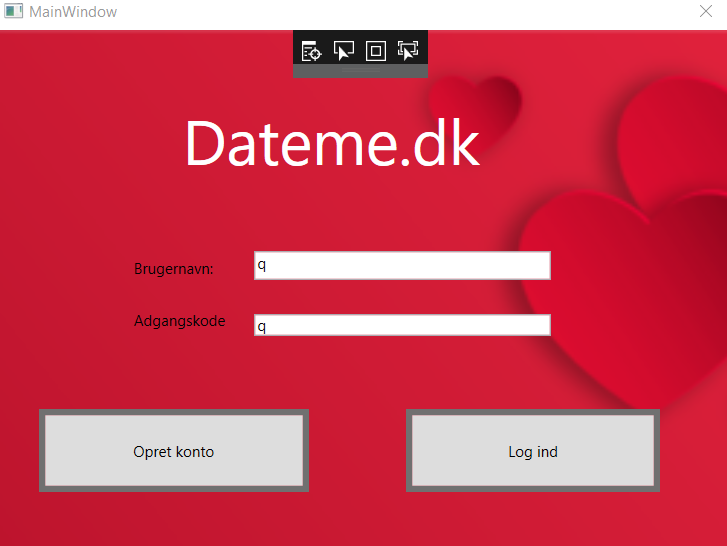
HorizontalAlignment="Left" Margin="207,227,0,0" VerticalAlignment="Top" Width="238" Height="18"/>

<Label Content="Brugernavn:" HorizontalAlignment="Left" Margin="106,177,0,0" VerticalAlignment="Top"/>

<Label Content="Adgangskode" HorizontalAlignment="Left" Margin="106,219,0,0" VerticalAlignment="Top"/>

</Grid>

</Window>



------

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using static test.Databaseconnection;

using System.Configuration;

using test.Primary\_funkions; //pga. mappestruktur

namespace test

{

/// <summary>

/// Interaction logic for matches.xaml

/// </summary>

public partial class matches : Window

{

public matches(int currentUserID, string YourUserName)

{

this.currentUserID = currentUserID;

yourUserName = YourUserName;

InitializeComponent();

BindDataGrid();

}

// HUSK AT ÆNDR

// receiverI

private int receiverID = 0;

private int currentUserID = 0;

private string yourUserName = "";

private string theirUserName = "";

//system .configuration

DOOPRET op = new DOOPRET();

public void BindDataGrid()

{

try

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

string searchQuery = $"SELECT fullName, sex, city, userID FROM userProfile, match where userProfile.userID=senderID AND isMatch=1 AND receiverID={currentUserID.ToString()}";

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

DataTable dt1 = new DataTable();

dt.Columns.Add(new DataColumn("isRead", typeof(string)));

// Checker hvis ulæst besked.

for (int i = 0; i < dt.Rows.Count; i++)

{

receiverID = (int)dt.Rows[0]["userID"];

string countQuery = $"SELECT COUNT(isRead) as isRead FROM directMessages WHERE senderID={dt.Rows[i]["userID"].ToString()} AND receiverID={currentUserID} AND isRead=0";

SqlDataAdapter sda1 = new SqlDataAdapter(countQuery, sqlCon);

sda1.Fill(dt1);

if ((int)dt1.Rows[i]["isRead"] >= 1)

{

dt.Rows[i]["isRead"] = "!";

}

else

{

dt.Rows[i]["isRead"] = ":(";

}

}

g1.ItemsSource = dt.DefaultView;

}

}

catch (SqlException ex)

{

MessageBox.Show("Fejl opstået" + ex);

}

}

private void GetSelectedRowCellValue(object sender, SelectionChangedEventArgs e)

{

DataRowView row = g1.SelectedItem as DataRowView;

if (row == null)

{

g1.SelectedItem = null;

receiverID = 0;

}

else

{

receiverID = Convert.ToInt32(row.Row.ItemArray[3].ToString());

MessageBox.Show(receiverID.ToString());

theirUserName = row.Row.ItemArray[0].ToString();

}

}

public void SelectButton\_Click(object sender, RoutedEventArgs e, matches ay) { }

public void SelectButton\_Click(object sender, RoutedEventArgs e) { }

private void DataGridRow\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

ChatLog chatLog = new ChatLog(currentUserID, receiverID, yourUserName, theirUserName);

chatLog.ShowDialog();

BindDataGrid();

}

}

}

---

<Window x:Class="test.matches"

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:test"

mc:Ignorable="d"

Title="matches" Height="400" Width="400"

ResizeMode="NoResize"

>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="\*"></RowDefinition>

</Grid.RowDefinitions>

<StackPanel Grid.Column="1">

<DataGrid x:Name="g1" Margin="5,15,5,0"

AutoGenerateColumns="False"

AlternatingRowBackground="Gray"

CanUserAddRows="False"

ScrollViewer.CanContentScroll="True"

ScrollViewer.VerticalScrollBarVisibility="Auto"

ScrollViewer.HorizontalScrollBarVisibility="Auto"

Height="300" Width="350"

SelectionChanged="GetSelectedRowCellValue"

SelectionMode="Single"

>

<DataGrid.Resources>

<Style TargetType="DataGridRow">

<EventSetter Event="MouseDoubleClick" Handler="DataGridRow\_MouseDoubleClick" />

</Style>

</DataGrid.Resources>

<DataGrid.Columns>

<DataGridTextColumn Binding="{Binding fullName}" Header="Navn"

Width="\*" IsReadOnly="true" />

<DataGridTextColumn Binding="{Binding sex}" Header="Køn"

Width="auto" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding city}" Header="By"

Width="\*" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding isRead}" Header="msg"

Width="auto" IsReadOnly="true"/>

</DataGrid.Columns>

</DataGrid>

</StackPanel>

<Grid.Background>

<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill" />

</Grid.Background>

</Grid>

</Window>

---

using System.Windows;

using System.Data.SqlClient;

using static test.Databaseconnection;

using System;

namespace test

{

/// <summary>

/// Interaction logic for Opdaterbruger.xaml

/// </summary>

public partial class Opdaterbruger : Window

{

public Opdaterbruger(int userID)

{

currentUserID = userID;

InitializeComponent();

MessageBox.Show(userID.ToString());

}

private int currentUserID = 0;

private void updateuser(object sender, RoutedEventArgs e)

{

string searchQuery = "UPDATE userProfile SET ";

string searchQueryEnding = $" WHERE userID={currentUserID}";

int numberOfChanges = CountChangesToAccount();

if (txtFullName.Text.Trim() != "")

{

searchQuery += $"fullName='{txtFullName.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtSex.Text.Trim() != "")

{

searchQuery += $"sex='{txtSex.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtAge.Text.Trim() != "")

{

searchQuery += $"age={txtAge.Text.Trim()}";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtCity.Text.Trim() != "")

{

searchQuery += $"city='{txtCity.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtAboutMe.Text.Trim() != "")

{

searchQuery += $"aboutMe='{txtAboutMe.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtlookingFor.Text.Trim() != "")

{

searchQuery += $"lookingFor='{txtlookingFor.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

if (txtAvatar.Text.Trim() != "")

{

searchQuery += $"avatar='{txtAvatar.Text.Trim()}'";

if (numberOfChanges != 1)

{

searchQuery += ", ";

numberOfChanges -= 1;

}

}

searchQuery += searchQueryEnding;

try

{

using (SqlConnection con = new SqlConnection(dbconnection.ConnectionDB()))

{

SqlCommand adapter = new SqlCommand(searchQuery, con);

adapter.Connection.Open();

adapter.ExecuteNonQuery();

adapter.Connection.Close();

MessageBox.Show("Success!");

}

}

catch (SqlException ex)

{

MessageBox.Show("fejl" + e);

}

}

private int CountChangesToAccount()

{

int countingChanges = 0;

if (!txtFullName.Text.Trim().Equals(""))

countingChanges++;

if (!txtAge.Text.Trim().Equals(""))

countingChanges++;

if (!txtSex.Text.Trim().Equals(""))

countingChanges++;

if (!txtCity.Text.Trim().Equals(""))

countingChanges++;

if (!txtAboutMe.Text.Trim().Equals(""))

countingChanges++;

if (!txtlookingFor.Text.Trim().Equals(""))

countingChanges++;

if (!txtAvatar.Text.Trim().Equals(""))

countingChanges++;

return countingChanges;

}

}

}

---

<Window x:Class="test.Opdaterbruger"

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:test"

mc:Ignorable="d"

Title="Opdaterbruger" Height="600" Width="800">

<StackPanel Margin="10">

<Label Target="{Binding ElementName=txtid}" Height="30" Margin="20,20,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Navn" />

</StackPanel>

</Label>

<TextBox x:Name ="txtFullName" Height="24" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtalder}" Height="27" Margin="20,0,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Alder"/>

</StackPanel>

</Label>

<TextBox x:Name ="txtAge" Height="23" Margin="20,0,670.6,0" />

<Label Margin="20,0">vælg køn</Label>

<ComboBox x:Name="txtSex" Margin="20,0,670.6,0">

<ComboBoxItem>

k

</ComboBoxItem>

<ComboBoxItem>

M

</ComboBoxItem>

<ComboBoxItem>

I

</ComboBoxItem>

</ComboBox>

<Label Target="{Binding ElementName=txtby}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_By" />

</StackPanel>

</Label>

<TextBox x:Name ="txtCity" Height="23" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtommig}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Om dig" />

</StackPanel>

</Label>

<TextBox x:Name ="txtAboutMe" Height="23" Margin="20,0,9.6,0"/>

<Label Target="{Binding ElementName=txtinteresser}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Interesser" />

</StackPanel>

</Label>

<TextBox x:Name ="txtlookingFor" Height="24" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtsoger}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Avatar" />

</StackPanel>

</Label>

<TextBox x:Name ="txtAvatar" Height="23" Margin="20,0,450.6,0"/>

<Button Margin="70,30,595.6,0" Height="32" Content="Opdater" Click="updateuser" RenderTransformOrigin="-0.241,0.426"/>

<StackPanel.Background>

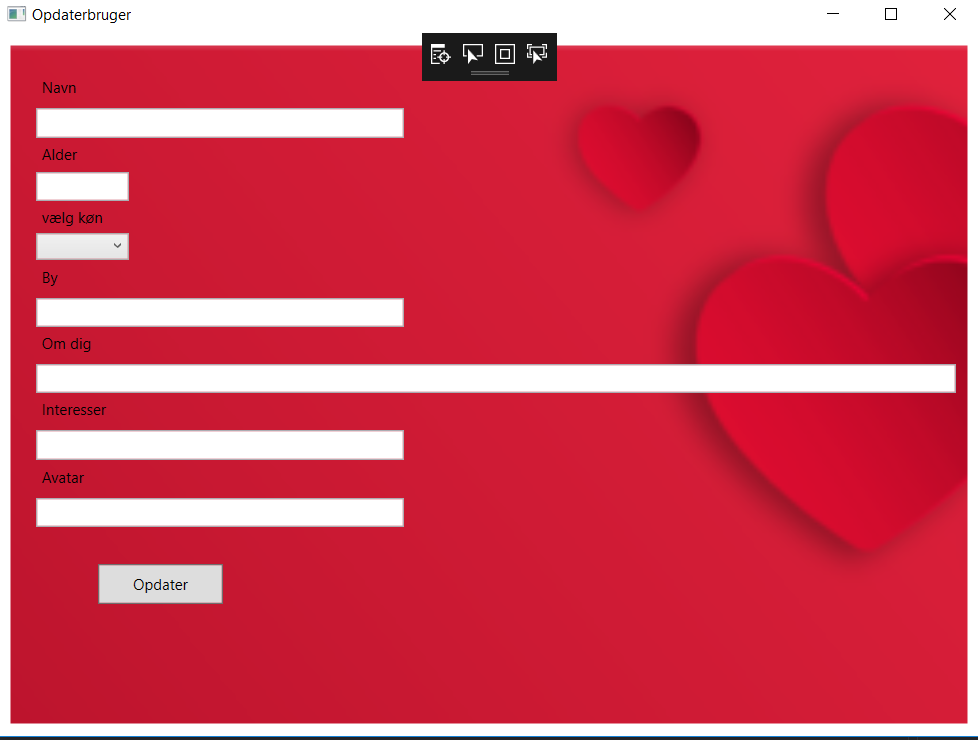
<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</StackPanel.Background>

</StackPanel>

</Window>



---

using System;

using System.Windows;

using System.Data.SqlClient;

using static test.Databaseconnection;

using Microsoft.Win32;

using System.Windows.Media.Imaging;

using System.IO;

using test.Primary\_funkions;

namespace test

{

/// <summary>

/// Interaction logic for opretbruger.xaml

/// </summary>

public partial class opretbruger : Window

{

public opretbruger(int userID)

{

currentUserID = userID;

InitializeComponent();

locationToAvatarDirectory = @"..\..\billede\ProfilBilleder\";

avatarFilename = currentUserID.ToString() + "\_Avatar.png";

MessageBox.Show(currentUserID.ToString());

}

private string locationToAvatarDirectory = "";

private string avatarFilename = "";

private int currentUserID = 0;

public void gembruger(object sender, RoutedEventArgs e)

{

if (txtsex.Text.Trim() != "" && txtnavn.Text.Trim() != "" &&

txtalder.Text.Trim() != "" && txtby.Text.Trim() != "" &&

txtommig.Text.Trim() != "" && txtinter.Text.Trim() != "")

{

try

{

//string searchQuery = "INSERT INTO UserProfile(sex,fullname, alder, [by], hvemerjeg, jegsoger, interesser) VALUES (@sex,@fullname, @alder, @by, @hvemerjeg, @jegsoger, @interesser)";

SaveImageToDB();

string searchQuery = "INSERT INTO userProfile " +

"(sex, fullName, age, city, aboutMe, lookingFor, avatar, userID) " +

"VALUES(@sex, @fullname, @age, @city, @aboutMe, @lookingFor, @avatar, @userID)";

using (SqlConnection con = new SqlConnection(dbconnection.ConnectionDB()))

{

SqlCommand command = new SqlCommand(searchQuery, con);

command.Connection.Open();

command.Parameters.AddWithValue("@sex", txtsex.Text.Trim());

command.Parameters.AddWithValue("@fullname", txtnavn.Text.Trim());

command.Parameters.AddWithValue("@age",txtalder.Text.Trim());

command.Parameters.AddWithValue("@city", txtby.Text.Trim());

command.Parameters.AddWithValue("@aboutMe", txtommig.Text.Trim());

command.Parameters.AddWithValue("@lookingFor", txtinter.Text.Trim());

// GET BACK

command.Parameters.AddWithValue("@avatar", avatarFilename);

command.Parameters.AddWithValue("@userID", currentUserID.ToString());

command.ExecuteNonQuery();

command.Connection.Close();

}

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

}

}

public void UploadLocallyStoredImage(object sender, RoutedEventArgs e)

{

// Should work, doesn't save.

OpenFileDialog dlg = new OpenFileDialog();

dlg.DefaultExt = ".png";

dlg.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg";

Nullable<bool> result = dlg.ShowDialog();

if (result == true) {

string filename = dlg.FileName;

ImgAvatar.Source = new BitmapImage(new Uri(filename));

}

}

public void LinkToImage()

{

if (txtAvatar.Text.Trim() != "") {

try {

ImgAvatar.Source = new BitmapImage(new Uri(txtAvatar.Text.Trim()));

}

catch (Exception) {

MessageBox.Show("invalid file");

}

}

else {

}

}

public void SaveImageToDB()

{

// TODO: GEM BILLEDE FRA SØGTE PROFIL HVIS MANGLER

// C:\Users\tec\source\repos\TestMedDataBinding\wpf\test\billede\ProfilBilleder

//string filePathOnly = @"..\..\billede\ProfilBilleder\";

//string nameForCurrentUserAvatar = filePathOnly + currentUserID.ToString() + "\_ProfilePic.png";

//currentUserAvatar = nameForCurrentUserAvatar;

// fungere også med andre filtyper end png

var encoder = new PngBitmapEncoder();

encoder.Frames.Add(BitmapFrame.Create((BitmapSource)ImgAvatar.Source));

// TODO TRY CATCH

using (FileStream fs = new FileStream(locationToAvatarDirectory + avatarFilename, FileMode.Create))

{

// ..\..\billede\ProfilBilleder\

encoder.Save(fs);

locationToAvatarDirectory = locationToAvatarDirectory.Substring(5) + avatarFilename;

txtAvatar.Text = locationToAvatarDirectory;

}

}

}

}

---

<Window x:Class="test.opretbruger"

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:test"

xmlns:opret="clr-namespace:test.Primary\_funkions"

mc:Ignorable="d"

Title="opretbruger" Height="800" Width="800"

ResizeMode="NoResize"

>

<!--<Window.DataContext>

<opret:DOOPRET x:Name="vmOpretter" />

</Window.DataContext>-->

<StackPanel Margin="10">

<Label Target="{Binding ElementName=txtid}" Height="30" Margin="20,20,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Name" />

</StackPanel>

</Label>

<TextBox x:Name ="txtnavn" Text="{Binding Ditfuldenavn,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

Height="24" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtalder}" Height="27" Margin="20,0,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Alder"/>

</StackPanel>

</Label>

<TextBox x:Name ="txtalder" Text="{Binding Alder,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

Height="23" Margin="20,0,670.6,0" />

<Label Margin="20,0">vælg køn</Label>

<ComboBox x:Name="txtsex" Margin="20,0,670.6,0">

<ComboBoxItem>

K

</ComboBoxItem>

<ComboBoxItem>

M

</ComboBoxItem>

<ComboBoxItem>

I

</ComboBoxItem>

</ComboBox>

<Label Target="{Binding ElementName=txtby}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_By" />

</StackPanel>

</Label>

<TextBox x:Name ="txtby" Text="{Binding By,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

Height="23" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtommig}"

Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Om dig" />

</StackPanel>

</Label>

<TextBox x:Name ="txtommig" Text="{Binding Hvemerjeg,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

Height="23" Margin="20,0,148.6,0"/>

<Label Target="{Binding ElementName=txtinteresser}" Height="30" Margin="20,0,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Interesser" />

</StackPanel>

</Label>

<TextBox x:Name ="txtinter" Text="{Binding Interesser,Mode=OneWayToSource,UpdateSourceTrigger=PropertyChanged}"

Height="24" Margin="20,0,450.6,0"/>

<Label Target="{Binding ElementName=txtsoger}" Height="30" Margin="20,0,622,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Avatar filepath" />

</StackPanel>

</Label>

<TextBox x:Name ="txtAvatar" IsReadOnly="True" Height="23" Margin="20,0,450.6,0"/>

<Image x:Name="ImgAvatar" Height="100" Width="Auto" Margin="50,10,567.6,0" RenderTransformOrigin="0.509,-1.437"/>

<Button Content="Upload Locally Stored Image" Click="UploadLocallyStoredImage" Margin="20,20,567.6,0" Height="40"/>

<Button Margin="50,20,659,0" Height="32" Content="Gem" Click="gembruger" RenderTransformOrigin="-0.241,0.426"/>

<StackPanel.Background>

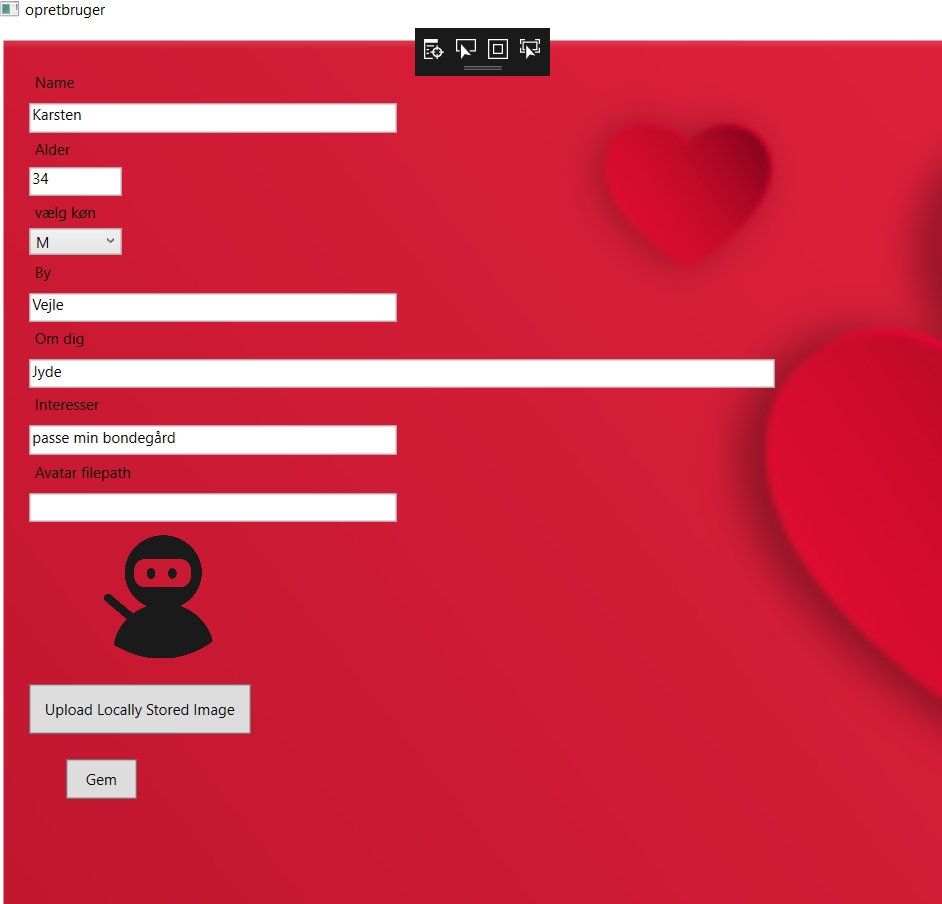
<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</StackPanel.Background>

</StackPanel>

</Window>



---

using System;

using System.Windows;

using System.Data;

using System.Data.SqlClient;

using static test.Databaseconnection;

namespace test

{

/// <summary>

/// Interaction logic for opretkonto.xaml

/// </summary>

public partial class opretkonto : Window

{

public opretkonto()

{

userAccountUserId = GetIdInTotal() + 1;

MessageBox.Show(userAccountUserId.ToString());

InitializeComponent();

}

private int userAccountUserId = 0;

private int GetIdInTotal()

{

using (SqlConnection con = new SqlConnection(dbconnection.ConnectionDB()))

{

string searchQuery = "SELECT COUNT(\*) FROM userAccount";

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, dbconnection.ConnectionDB());

sda.Fill(dt);

return Convert.ToInt32(dt.Rows[0][0].ToString());

}

}

public void CreateUserAccount()

{

try

{

string insertQuery = "INSERT INTO userAccount (username, [password], email) VALUES (@Login, @Adgangskode, @Email)";

using (SqlConnection con = new SqlConnection(dbconnection.ConnectionDB()))

{

// TODO Tjek at username er unikt.

// TODO Double tjek for password?

// TODO Tjek at email er unikt?

SqlCommand command = new SqlCommand(insertQuery, con);

command.Connection.Open();

command.Parameters.AddWithValue("@Login", txtName.Text);

command.Parameters.AddWithValue("@Email", txtPassword.Text);

command.Parameters.AddWithValue("@Adgangskode", txtEmail.Text);

command.ExecuteScalar();

command.Connection.Close();

}

}

catch (SqlException ex)

{

Console.WriteLine(ex.Message);

}

}

public void opretbruger(object sender, RoutedEventArgs e)

{

if (txtName.Text.Trim() != "" && txtPassword.Text.Trim() != "" && txtEmail.Text.Trim() != "")

{

CreateUserAccount();

// +1 for at få den nye bruger.

opretbruger opretbruger = new opretbruger(userAccountUserId);

opretbruger.Show();

this.Close();

} else // TODO Gør error message mere specifikt.

{

MessageBox.Show("Missing info");

}

}

}

}

---

<Window x:Class="test.opretkonto"

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:test"

mc:Ignorable="d"

Title="opretkonto" Height="600" Width="800"

ResizeMode="NoResize"

>

<StackPanel Margin="10">

<Label x:Name="lblUsername" Height="47" Margin="20,20,670.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="Username" />

</StackPanel>

</Label>

<TextBox x:Name="txtName" Height="32" Margin="20,0,293.6,0"/>

<Label x:Name="lblPassword" Height="39" Margin="20,10,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Password"/>

</StackPanel>

</Label>

<TextBox x:Name="txtEmail" Height="32" Margin="20,0,293.6,0" />

<Label x:Name="lblEmail" Height="39" Margin="20,10,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Email"/>

</StackPanel>

</Label>

<TextBox x:Name="txtPassword" Height="32" Margin="20,0,293.6,0" />

<Button Height="62" Margin="180,20,447.6,0" Click="opretbruger" Content="opret bruger"/>

<StackPanel.Background>

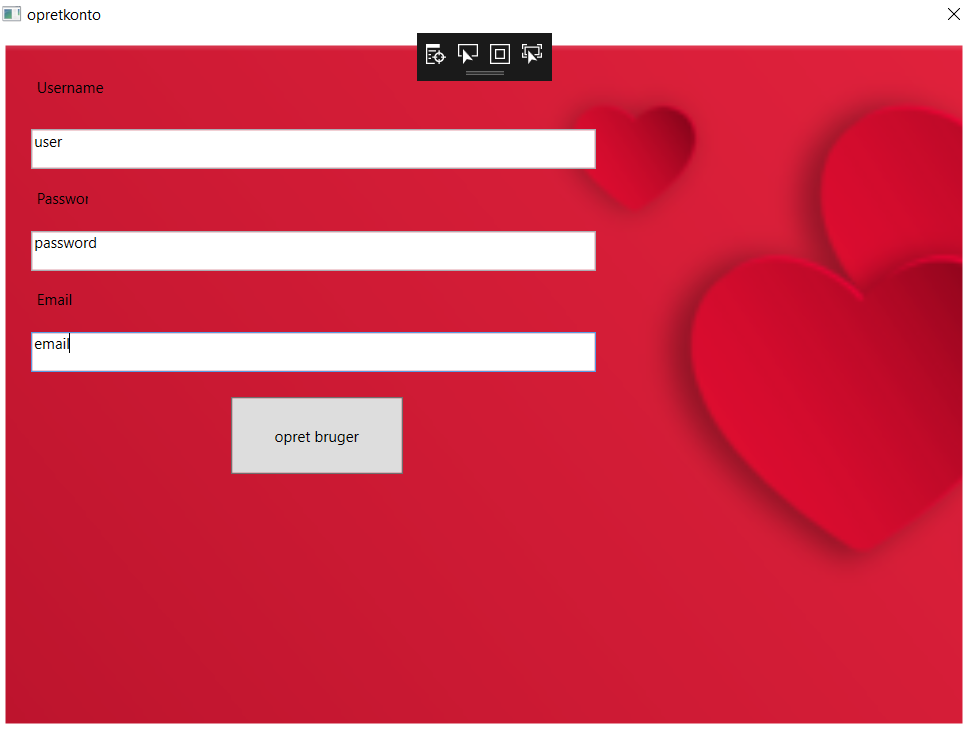
<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</StackPanel.Background>

</StackPanel>

</Window>



---

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data;

using System.Data.SqlClient;

using static test.Databaseconnection;

namespace test

{

/// <summary>

/// Interaction logic for sletbruger.xaml

/// </summary>

public partial class sletbruger : Window

{

public sletbruger()

{

InitializeComponent();

}

private void slet(object sender, RoutedEventArgs e)

{

try

{

DataTable table = new DataTable();

using (SqlConnection con = new SqlConnection(dbconnection.ConnectionDB()))

{

con.Open();

string DB = "UPDATE userAccount SET deleted='1' WHERE password=@password AND Email=@email";

//VI updater så brugere altid kan komme tilbage - se under deleted = true i DB

SqlCommand command = new SqlCommand(DB, con);

command.Parameters.AddWithValue("@password", txtpassword.Text);

command.Parameters.AddWithValue("@email", txtemail.Text);

command.ExecuteNonQuery();

}

}

catch (SqlException ex)

{

MessageBox.Show("fejl opstået: " + ex);

}

}

} }

---

<Window x:Class="test.sletbruger"

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:test"

mc:Ignorable="d"

Title="sletbruger" Height="450" Width="800"

ResizeMode="NoResize"

>

<StackPanel Margin="10">

<Label Height="39" Margin="20,10,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Password"/>

</StackPanel>

</Label>

<TextBox x:Name ="txtemail" Height="32" Margin="20,0,162.6,0" />

<Label Height="39" Margin="20,10,694.6,0">

<StackPanel Orientation="Horizontal">

<AccessText Text="\_Email"/>

</StackPanel>

</Label>

<TextBox x:Name ="txtpassword" Height="32" Margin="20,0,162.6,0" />

<Button Margin="216,50,369.6,50" Height="61" Content="slet" Click="slet"/>

<StackPanel.Background>

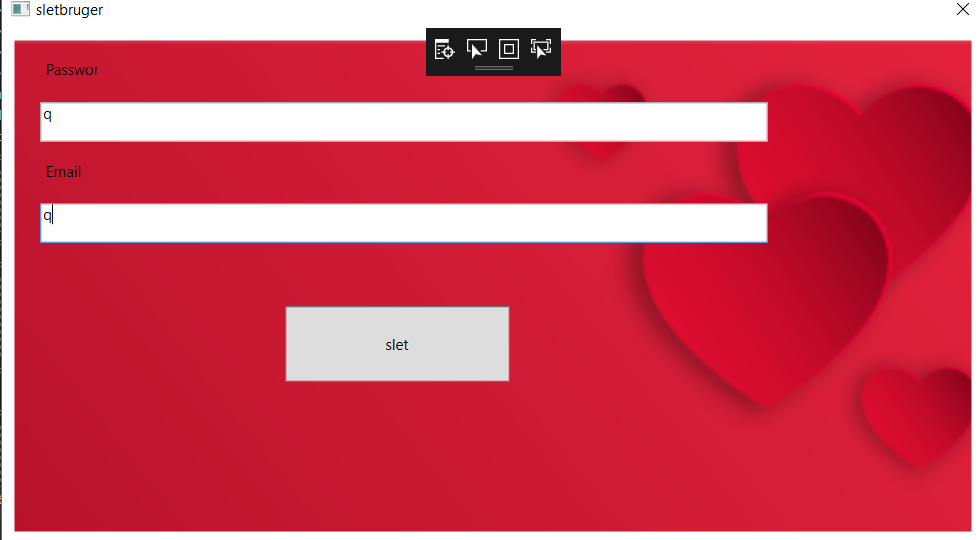
<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill">

</ImageBrush>

</StackPanel.Background>

</StackPanel>

</Window>



---

using System.Windows;

using System.Data;

using System.Data.SqlClient;

using static test.Databaseconnection;

using System.Configuration;

using System.Text.RegularExpressions;

using System.Windows.Input;

using System;

using System.Windows.Controls;

using System.Windows.Media.Imaging;

using test.Primary\_funkions;

namespace test

{

/// <summary>

/// Interaction logic for soge.xaml

/// </summary>

public partial class soge : Window

{

public soge(int receivedUserID)

{

InitializeComponent();

currentUserID = receivedUserID;

BindDataGrid();

}

private int maxAndMinAgeDifferenceInSearch = 5;

private int currentUserID = 0;

private int likedPersonUserID = 0;

DOOPRET op = new DOOPRET();

public void BindDataGrid()

{

try

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

//string searchQuery = "SELECT fullName, sex, city, age, aboutMe, lookingFor, userID, avatar FROM userProfile";

string searchQuery = $"SELECT p.fullName,p.sex,p.city,p.aboutMe,lookingFor,p.userID FROM userAccount as a, userProfile as p WHERE a.userID = p.userID and deleted IS NULL AND NOT p.userID ={ currentUserID.ToString()}";

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

g1.ItemsSource = dt.DefaultView;

}

}

catch (SqlException ex)

{

MessageBox.Show("Fejl opstået" + ex);

}

}

private void likebutton(object sender, RoutedEventArgs e) {

if (likedPersonUserID > 0)

{

string sqlQuery = $"INSERT INTO match (senderID,receiverID) VALUES ({currentUserID}, {likedPersonUserID})";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

SqlCommand cmd = new SqlCommand(sqlQuery, sqlCon);

cmd.ExecuteNonQuery();

sqlCon.Close();

}

if (CheckIfMatch())

{

ChangeMatchStatus();

}

else

{

MessageBox.Show("Du har liket en bruger som ikke kan lide dig du er en taber du bliver aldrig til noget :)");

}

MessageBox.Show(currentUserID.ToString() + " har liket " + likedPersonUserID.ToString());

}

else

{

//MessageBox.Show("nope.");

}

}

private void ChangeMatchStatus()

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

string changeCurrentUserStatus = $"UPDATE match SET isMatch=1 where senderID={currentUserID} AND receiverID={likedPersonUserID}";

string changeLikedUserStatus = $"UPDATE match SET isMatch=1 where senderID={likedPersonUserID} AND receiverID={currentUserID}";

SqlCommand cmdCurrentUser = new SqlCommand(changeCurrentUserStatus,sqlCon);

SqlCommand cmdLikedUser = new SqlCommand(changeLikedUserStatus,sqlCon);

cmdCurrentUser.ExecuteNonQuery();

cmdLikedUser.ExecuteNonQuery();

sqlCon.Close();

}

}

private bool CheckIfMatch()

{

string searchQuery = $"SELECT \* FROM match where senderID={likedPersonUserID} AND receiverID={currentUserID}";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB())) {

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

if (dt.Rows.Count >= 1)

{

return true;

}

return false;

}

}

public void sogemedlemmer(object sender, RoutedEventArgs e)

{

try

{

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

sqlCon.Open();

//string searchQuery = $"SELECT fullname, sex, city, age, aboutMe, lookingFor FROM userProfile WHERE sex='{txtSex.Text}'";

//string searchQuery = $"SELECT \* FROM userProfile WHERE sex='{txtSex.Text}'";

int minAge = 0, maxAge = 100, userSearchAge = 0;

if (txtAge.Text.Trim() != "")

{

userSearchAge = Convert.ToInt32(txtAge.Text.Trim());

minAge = userSearchAge - maxAndMinAgeDifferenceInSearch; maxAge = userSearchAge + maxAndMinAgeDifferenceInSearch;

} else

{

minAge = 0; maxAge = 100;

}

// TODO DET ER GRIMT AT KIGGE PÅ TODO

string searchQuery = $"SELECT fullName, sex, city, age, aboutMe, lookingFor, p.userID, avatar From userProfile as p, userAccount as a WHERE age BETWEEN {minAge} AND {maxAge}"

+ $"AND sex LIKE '%{txtSex.Text.Trim()}%' AND NOT p.userID={currentUserID.ToString()} AND a.userID=p.userID AND deleted IS NULL";

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

g1.ItemsSource = dt.DefaultView;

}

}

catch (SqlException ex)

{

MessageBox.Show("Fejl opstået" + ex);

}

}

public void SelectButton\_Click(object sender, RoutedEventArgs e)

{

DataRowView um = (DataRowView)g1.SelectedItem;

var selectrow = new DOOPRET()

{

Id = um[0].ToString(),

Ditfuldenavn = um[1].ToString(),

Alder = (int)um[2],

By = um[3].ToString(),

Hvemerjeg = um[4].ToString(),

Interesser = um[5].ToString(),

Soger = um[6].ToString()

};

}

// Få info omkring valgt bruger

private void GetSelectedRowCellValue()

{

DataRowView row = g1.SelectedItem as DataRowView;

if (row == null)

{

g1.SelectedItem = null;

likedPersonUserID = 0;

} else

{

likedPersonUserID = Convert.ToInt32(row.Row.ItemArray[6].ToString());

GetSelectedUserProfile(row);

}

}

// Fejlsikret fra kaldt metode TODO FIKS

private void GetSelectedUserProfile(DataRowView row)

{

string filepath = System.Reflection.Assembly.GetEntryAssembly().Location;

string filepathtoimages = @"billede\ProfilBilleder\";

int index = filepath.LastIndexOf(@"\bin\") + 1;

filepath = filepath.Substring(0, index) + filepathtoimages + row.Row.ItemArray[7].ToString();

imgAvatar.Source = new BitmapImage(new Uri(filepath));

}

// KUN TAL I TEKSTBOX

private void NumberValidationTextBox(object sender, TextCompositionEventArgs e)

{

Regex regex = new Regex("[^0-9]+");

e.Handled = regex.IsMatch(e.Text);

}

private void G1\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

GetSelectedRowCellValue();

}

}

}

---

<Window x:Class="test.soge"

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:test"

mc:Ignorable="d"

Title="soge" Height="732.4" Width="800"

ResizeMode="CanResize"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*" />

<RowDefinition Height="\*" />

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<StackPanel Margin="10" Grid.Row="0" Grid.Column="0">

<Label Margin="20,10,0,0">\_vælg køn du vil søge på:</Label>

<ComboBox x:Name="txtSex" Margin="20,10,281.8,0">

<ComboBoxItem>

F

</ComboBoxItem>

<ComboBoxItem>

M

</ComboBoxItem>

<ComboBoxItem>

I

</ComboBoxItem>

</ComboBox>

<Label Margin="20,10,0,0">\_vælg alder du vil søge på:</Label>

<TextBox x:Name="txtAge" PreviewTextInput="NumberValidationTextBox" Margin="20,10,277.8,0" RenderTransformOrigin="-0.97,-0.697" Height="25">

<TextBox.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform Angle="-0.363"/>

<TranslateTransform/>

</TransformGroup>

</TextBox.RenderTransform>

</TextBox>

<Button Height="20" Margin="20,20,281.8,0" RenderTransformOrigin="0.501,0.5" Click="sogemedlemmer">søg</Button>

<Button x:Name="myBtn\_DetailsTab\_Save" FlowDirection="LeftToRight"

HorizontalAlignment="Left" Margin="20,10,0,10" VerticalAlignment="Top"

Width="50" Height="50" BorderBrush="#FF0F6287"

HorizontalContentAlignment="Center" VerticalContentAlignment="Center"

FontFamily="B Titr" FontSize="15" FontWeight="Bold"

BorderThickness="2" Click="likebutton">

<StackPanel>

<Image HorizontalAlignment="Left" Source="billede/like.png" Height="36" Width="124" />

</StackPanel>

</Button>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1">

<Image x:Name="imgAvatar" Height="100" Width="100" Margin="0,0,0,0"/>

</StackPanel>

<StackPanel Grid.ColumnSpan="2" Grid.Row="1"

Orientation="Horizontal">

<DataGrid x:Name="g1" Margin="70,10,50,10.2"

AutoGenerateColumns="False"

AlternatingRowBackground="Gray"

CanUserAddRows="False"

ScrollViewer.CanContentScroll="True"

ScrollViewer.VerticalScrollBarVisibility="Auto"

ScrollViewer.HorizontalScrollBarVisibility="Auto"

SelectionChanged="G1\_SelectionChanged"

SelectionMode="Single"

>

<DataGrid.Columns>

<!--<DataGridTextColumn Binding="{Binding userProfileId}" Header="UserID"

Width="100" IsReadOnly="true"/>-->

<DataGridTextColumn Binding="{Binding fullName}" Header="Navn"

Width="100" IsReadOnly="true" />

<DataGridTextColumn Binding="{Binding sex}" Header="Køn"

Width="100" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding age}" Header="Alder"

Width="100" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding city}" Header="By"

Width="100" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding aboutMe}" Header="Om mig"

Width="100" IsReadOnly="true"/>

<DataGridTextColumn Binding="{Binding lookingFor}" Header="Søger"

Width="100" IsReadOnly="true"/>

<!--<DataGridTextColumn Binding="{Binding jegsoger}" Header="Søger"

Width="100" IsReadOnly="true"/>-->

<!--TODO-->

</DataGrid.Columns>

</DataGrid>

</StackPanel>

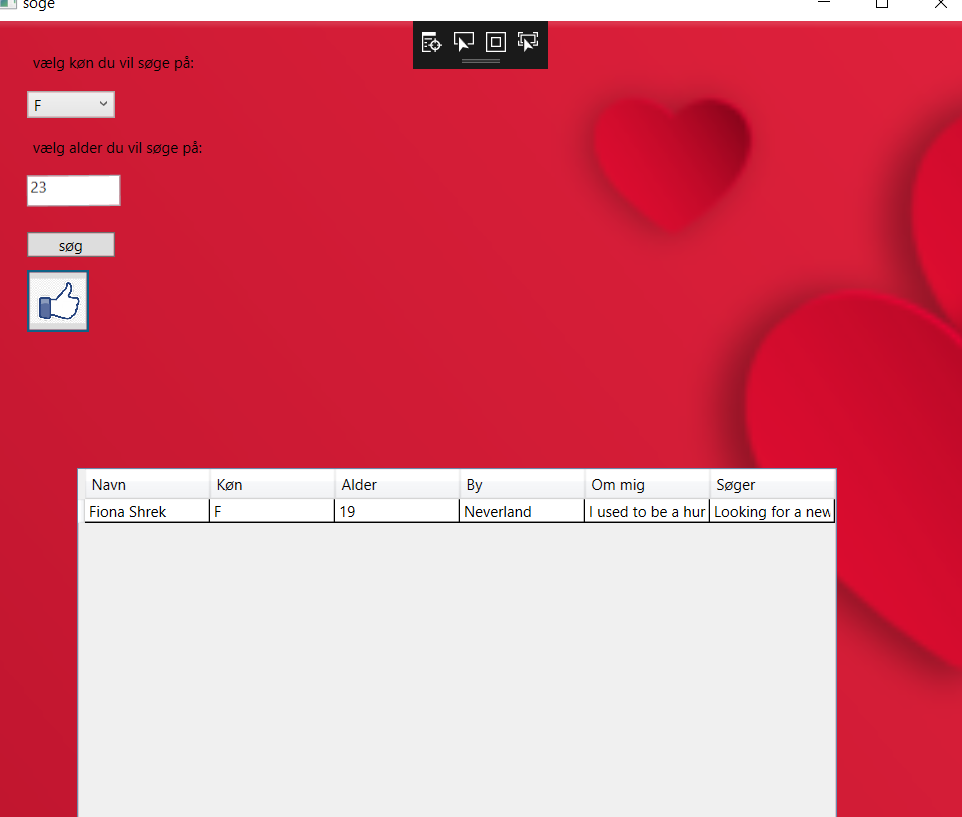
<Grid.Background>

<ImageBrush ImageSource="billede\frontpage.png" Stretch="UniformToFill" />

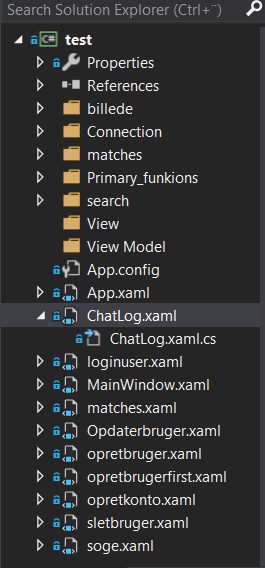
</Grid.Background>

</Grid>

</Window>



SOLUTION:



DOOPRET: (BINDINGS)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

using System.ComponentModel;

using static test.Databaseconnection;

using System.Windows;

namespace test.Primary\_funkions //pga. mappestruktur

{

public class DOOPRET : INotifyPropertyChanged //primært den der arbejder

{

private string \_adgangskode;

public string Adgangskode {

get { return \_adgangskode; }

set { \_adgangskode = value; OnPropertyChanged("adgangskode"); }

}

public string email

{

get { return email; }

set { email = value; OnPropertyChanged("email"); }

}

private string \_id;

public string Id {

get { return \_id; }

set { \_id = value; OnPropertyChanged("id"); }

}

private string \_hvemerjeg;

public string Hvemerjeg {

get { return \_hvemerjeg; }

set { \_hvemerjeg = value;OnPropertyChanged("hvemerjeg"); }

}

private string \_ditfuldenavn;

public string Ditfuldenavn {

get { return \_ditfuldenavn; }

set { \_ditfuldenavn = value; OnPropertyChanged("ditfuldenavn"); }

}

private int \_alder;

public int Alder {

get { return \_alder; }

set { \_alder = value; OnPropertyChanged("alder"); }

}

private string \_by;

public string By {

get { return \_by; }

set { \_by = value; OnPropertyChanged("by"); }

}

private string \_soger;

public string Soger {

get { return \_soger; }

set { \_soger = value; OnPropertyChanged("soge"); }

}

private string \_interesser;

public string Interesser {

get { return \_interesser; }

set { \_interesser = value; OnPropertyChanged("interesser"); }

}

private string \_login;

public string Login {

get { return \_login; }

set { \_login = value; OnPropertyChanged("login"); }

}

private string \_sex;

public string Sex {

get { return \_sex; }

set { \_sex = value; OnPropertyChanged("sex"); }

}

public event PropertyChangedEventHandler PropertyChanged;

//constructor callled to set designtime data if necesary

private void OnPropertyChanged(string propertyName)

{

if (PropertyChanged != null)

PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

public bool InitialLogin(string bNavn, string pWord)

{

string searchQuery = $"SELECT \* FROM userAccount WHERE username='{bNavn}' AND [password]='{pWord}'";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchQuery, sqlCon);

sda.Fill(dt);

if (dt.Rows.Count == 1)

{

return true;

} else

{

return false;

}

}

}

public string GetCurrentUserId(string bNavn)

{

string searchForIdWithPasswordQuery = $"SELECT userID FROM userAccount WHERE username='{bNavn}'";

using (SqlConnection sqlCon = new SqlConnection(dbconnection.ConnectionDB()))

{

DataTable dt = new DataTable();

SqlDataAdapter sda = new SqlDataAdapter(searchForIdWithPasswordQuery, sqlCon);

sda.Fill(dt);

return dt.Rows[0]["userID"].ToString();

}

}

public void Loginuser() //her er vores loginuser

{

string bNavn = Login.Trim(), pWord = Adgangskode;

if (bNavn != "" && pWord != "")

{

DOOPRET bOpret = new DOOPRET();

if (bOpret.InitialLogin(bNavn, pWord))

{

loginuser login = new loginuser(bOpret.GetCurrentUserId(bNavn));

login.Show();

App.Current.Windows[0].Close();

//this.Close();

}

else

{

MessageBox.Show("fail :(");

}

}

else

{

// TODO

MessageBox.Show("PLACEHOLDER: FUCKED");

}

//loginuser log = new loginuser();

//log.Show();

//this.Close();

}

}

}

SQL:

-- !!!!! CHANGELOG 22-05 !!!!!

-- ## Ændret seekingSex til LookingFor

-- !!! OPSÆTNING AF INITIAL DATABASE !!! --

USE MASTER

GO

DROP DATABASE DatingDB

GO

CREATE DATABASE DatingDB

GO

USE DatingDB

-- !!! /OPSÆTNING AF INITIAL DATABASE !!! --

-- !!! OPSÆTNING AF TABLES !!! --

CREATE TABLE userAccount(

userID INT IDENTITY(1,1) PRIMARY KEY,

username NVARCHAR(50) NOT NULL UNIQUE,

[password] NVARCHAR(50) NOT NULL,

email NVARCHAR(50) NOT NULL,

[deleted] BIT NULL

)

CREATE TABLE userProfile(

userProfileID INT IDENTITY(1,1) PRIMARY KEY,

sex CHAR,

fullName NVARCHAR(100),

age INT,

city NVARCHAR(75),

aboutMe NVARCHAR(140),

lookingFor NVARCHAR(140),

avatar NVARCHAR(200) DEFAULT 'defaultCowboy.png',

userID INT FOREIGN KEY REFERENCES userAccount(userID)

)

CREATE TABLE userSearch(

searchID INT IDENTITY(1,1) PRIMARY KEY, -- ID for search Query

sex CHAR NULL,

interest NVARCHAR(140) NULL,

age INT NULL,

userID INT FOREIGN KEY REFERENCES userAccount(userID) -- The Searcher

)

CREATE TABLE directMessages(

messageID INT IDENTITY(1,1) PRIMARY KEY,

senderID INT FOREIGN KEY REFERENCES userProfile(userProfileID) NOT NULL,

receiverID INT FOREIGN KEY REFERENCES userProfile(userProfileID) NOT NULL,

messageText NVARCHAR(500),

isRead BIT DEFAULT 0 NOT NULL -- FALSE/TRUE INITIALLY FALSE

)

--ALTER TABLE directMessages

--ADD CONSTRAINT PK\_Message PRIMARY KEY (messageID, senderID, receiverID)

CREATE TABLE match(

messageID INT IDENTITY(1,1) PRIMARY KEY,

senderID INT FOREIGN KEY REFERENCES userProfile(userProfileID),

receiverID INT FOREIGN KEY REFERENCES userProfile(userProfileID),

isMatch BIT DEFAULT 0 NOT NULL -- INITALLY ALWAYS NOT MATCH

)

select \* from userAccount

select \* from userProfile

INSERT INTO userAccount(username, [password], email)

VALUES

('Q','Q','Q'),

('A','A','A'),

('B','B','B'),

('C','C','C'),

('D','D','D'),

('E','E','E'),

('F','F','F')

INSERT INTO userProfile(sex,fullName,age,city,aboutMe,lookingFor,userID)

VALUES

('M','Don Juan',32,'Venecia','hot stud, coming through.','Looking for you, bby',1),

('M','Kanye West',43,'Up my own ass','I like to rap :)','Looking for me, bby',2),

('F','Taylor Slow',28,'California','1989 is my only good album.','Looking for love, bby',3),

('F','Dizzy Miss Lizzy',75,'Somewhere','I don''t even like this band.','Looking for something, bby',4),

('F','Fiona Shrek',19,'Neverland','I used to be a human, now I''m married.','Looking for a new husband.',5),

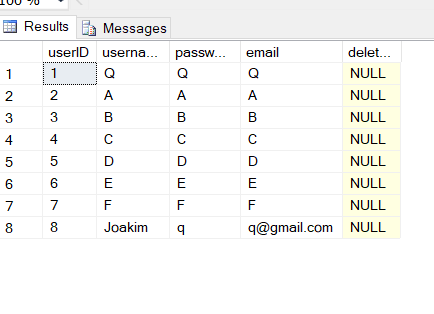
('M','The Rock Johnson',42,'Action movies','Not to be confused with Dwayne.','Looking for more movies to star in',6),

('I','Some Spooky Guy',132,'Tumblr','BEHOLD COWARDS.','LOOKING FOR MORE GENDERS',7)

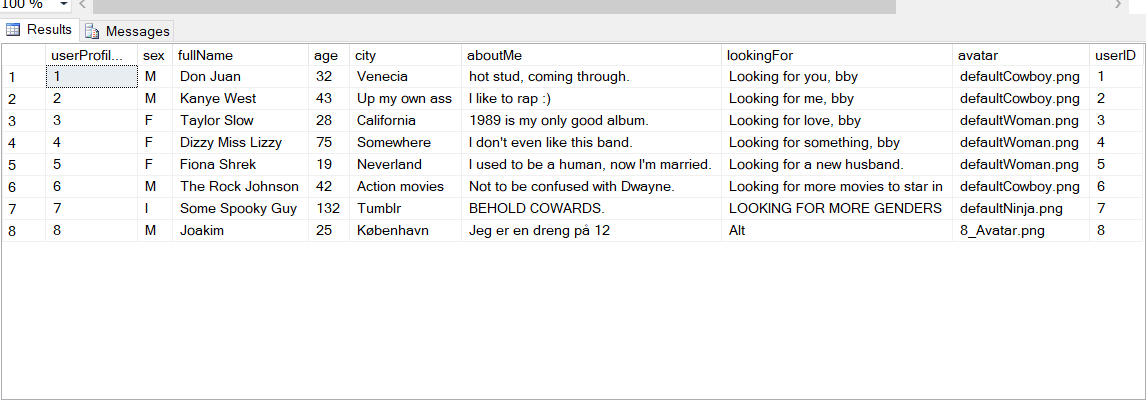
SELECT \* FROM userAccount as ua, userProfile as up WHERE ua.userID=up.userID

update userProfile set avatar='defaultWoman.png' where sex='F'

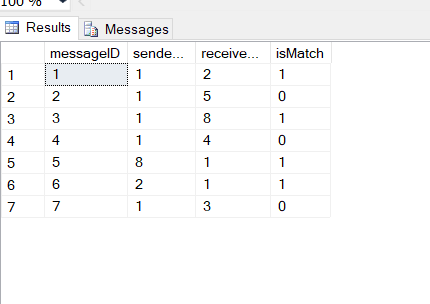
Useracccount



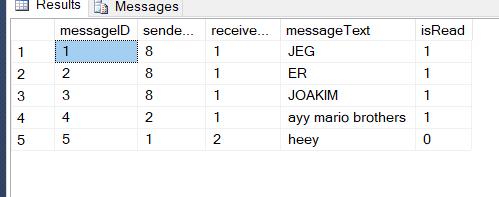
userprofile



Match



Besked



Isdeleted

