using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace pz9.serv

{

internal class captcha

{

private static readonly Random random = new Random();

private const string Characters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

/// <summary>

/// Генерирует случайный текст заданной длины для капчи.

/// </summary>

/// <param name="length">Длина текста капчи.</param>

/// <returns>Случайный текст капчи.</returns>

public static string GenerateCaptchaText(int length)

{

if (length <= 0)

throw new ArgumentException("Длина текста капчи должна быть больше нуля.");

StringBuilder captchaText = new StringBuilder(length);

for (int i = 0; i < length; i++)

{

int index = random.Next(Characters.Length);

captchaText.Append(Characters[index]);

}

return captchaText.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace pz9.serv

{

internal class greeting

{

public static string Greetingg()

{

DateTime time = DateTime.Now;

int hour = time.Hour;

hour = 18;

if (hour >= 10 && hour < 12)

{

return $"Доброе утро {avtoriz.surname} {avtoriz.name}!";

}

else if (hour >= 12 && hour < 17)

{

return $"Добрый день {avtoriz.surname} {avtoriz.name}!";

}

else if (hour >= 17 && hour < 19)

{

return $"Добрый вечер {avtoriz.surname} {avtoriz.name}!";

}

else

{

return "";

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Net;

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 System.Windows.Threading;

using pz9.model;

using pz9.serv;

namespace pz9

{

/// <summary>

/// Логика взаимодействия для avtoriz.xaml

/// </summary>

public partial class avtoriz : Page

{

private static Entities \_context;

public static string name { get; set; }

public static string surname { get; set; }

private DispatcherTimer timer;

int click;

private int timeLeft;

string ccode;

public avtoriz()

{

InitializeComponent();

click = 0;

GenerateCapctcha();

timer = new DispatcherTimer();

timer.Interval = TimeSpan.FromSeconds(1);

timer.Tick += Timer\_Tick;

tbCaptcha.Visibility = Visibility.Hidden;

tblCaptcha.Visibility = Visibility.Hidden;

capBtn.Visibility = Visibility.Hidden;

\_context = new Entities();

}

private void GenerateCapctcha()

{

string capctchaText = captcha.GenerateCaptchaText(6);

tblCaptcha.Text = capctchaText;

tblCaptcha.TextDecorations = TextDecorations.Strikethrough;

}

private void NavigationServise\_Navigated(object sender, NavigationEventArgs e)

{

}

private void btnEnterGuest\_Click(object sender, RoutedEventArgs e)

{

// NavigationService.Navigate(new Client(null, null));

}

private void btnEnter\_Click(object sender, RoutedEventArgs e)

{

click += 1;

string login = tbLogin.Text.Trim();

string password = hash.HashPassword(tbPassword.Text.Trim());

// Находим пользователя

var user = \_context.emp

.FirstOrDefault(x => x.login == login && x.password == password);

if (user == null)

{

throw new InvalidOperationException("Пользователь не найден.");

}

if (user.postID == 0)

{

throw new InvalidOperationException("У пользователя не указан postID.");

}

// Находим пост по postID

var postTitle = \_context.post

.Where(p => p.postID == user.postID)

.Select(p => p.title)

.FirstOrDefault();

if (postTitle == null)

{

throw new InvalidOperationException("Пост не найден.");

}

// Используем postTitle

Console.WriteLine($"Название поста: {postTitle}");

if (click <= 3)

{

tbLogin.IsEnabled = true;

tbPassword.IsEnabled = true;

tbCaptcha.IsEnabled = true;

btnEnter.IsEnabled = true;

btnEnterGuest.IsEnabled = true;

tbTimer.Visibility = Visibility.Hidden;

tbTimer.Text = string.Empty;

if (click == 1)

{

if (user != null)

{

MessageBox.Show("Вы вошли под: " + postTitle.ToString());

LoadPage(postTitle.ToString());

tbPassword.Clear();

tbCaptcha.Clear();

tbLogin.Clear();

tbCaptcha.Visibility = Visibility.Hidden;

tblCaptcha.Visibility = Visibility.Hidden;

//if (ccode == tbCode.Text)

//{

// tbCode.Clear();

//}

//else

//{

// MessageBox.Show("Получите и введите код с почты ");

//}

}

else

{

MessageBox.Show("Вы ввели логин или пароль неверно!");

GenerateCapctcha();

tbPassword.Clear();

tbCaptcha.Clear();

tbLogin.Clear();

tbCaptcha.Visibility = Visibility.Visible;

tblCaptcha.Visibility = Visibility.Visible;

}

}

else if (click > 1)

{

tbCaptcha.Visibility = Visibility.Visible;

tblCaptcha.Visibility = Visibility.Visible;

if (user != null && tbCaptcha.Text == tblCaptcha.Text)

{

if (ccode == tbCode.Text)

{

MessageBox.Show($"Вы вошли под: " + postTitle.ToString());

LoadPage(postTitle.ToString());

tbPassword.Clear();

tbCaptcha.Clear();

tbLogin.Clear();

tbCaptcha.Visibility = Visibility.Hidden;

tblCaptcha.Visibility = Visibility.Hidden;

tbCode.Clear() ;

}

else

{

MessageBox.Show("Получите и введите код с почты ");

}

}

else if (tblCaptcha.Text != tbCaptcha.Text)

{

MessageBox.Show("вы ввели капчу неверно");

GenerateCapctcha();

tbCaptcha.Clear();

}

else

{

MessageBox.Show("Введите данные заново!");

tbPassword.Clear();

tbLogin.Clear();

}

}

}

else

{

click = 0;

tbLogin.IsEnabled = false;

tbPassword.IsEnabled = false;

tbCaptcha.IsEnabled = false;

btnEnter.IsEnabled = false;

btnEnterGuest.IsEnabled = false;

tbTimer.Visibility = Visibility.Visible;

tbTime.Foreground = Brushes.Red;

timeLeft = 10;

timer.Start();

tbTime.Text = $"Осталось времени: {timeLeft} секунд";

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

}

private void Timer\_Tick(object sender, EventArgs e)

{

timeLeft--;

if (timeLeft > 0)

{

tbTime.Text = $"Осталось времени: {timeLeft} секунд";

}

else

{

timer.Stop();

tbTime.Text = "";

tbLogin.IsEnabled = true;

tbPassword.IsEnabled = true;

tbCaptcha.IsEnabled = true;

btnEnter.IsEnabled = true;

btnEnterGuest.IsEnabled = true;

tbTimer.Visibility = Visibility.Hidden;

tbTimer.Text = string.Empty;

}

}

private void LoadPage(string role)

{

string greetingg = greeting.Greetingg();

click = 0;

switch (role)

{

//case "Клиент":

// NavigationService.Navigate(new Client());

// break;

//case "Инженер ":

// NavigationService.Navigate(new injener(role,greetingg));

// break;

case "Aдминистратор":

NavigationService.Navigate(new admin(role));

break;

default:

MessageBox.Show("Роль не распознана.");

break;

}

}

private void a(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new code());

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

string login = tbLogin.Text.Trim();

string password = hash.HashPassword(tbPassword.Text.Trim());

var user = \_context.emp.FirstOrDefault(x => x.login == login && x.password == password);

newPassw a = new newPassw();

ccode = newPassw.newPassword(user.email);

}

}

}

<Page x:Class="pz9.avtoriz"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:pz9"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="avtoriz" Background="White">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="260"/>

<ColumnDefinition/>

<ColumnDefinition Width="260"/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="1" Grid.Column="1" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBlock Text="Логин"/>

<TextBox x:Name="tbLogin"/>

<TextBlock Text="Пароль"/>

<TextBox x:Name="tbPassword"/>

<TextBlock x:Name="tblCaptcha" Text="Капча"/>

<TextBox x:Name="tbCaptcha"/>

<TextBlock x:Name="tbTimer"/>

<TextBlock x:Name="tbTime" TextWrapping="Wrap"/>

<StackPanel Orientation="Horizontal" Height="80" HorizontalAlignment="Center">

<Button x:Name="btnEnterGuest" Content="Войти как гость" Click="btnEnterGuest\_Click"/>

<Button x:Name="btnEnter" Content="Войти" Click="btnEnter\_Click"/>

</StackPanel>

</StackPanel>

<Frame x:Name="NavigationServise" Grid.Row="1" NavigationUIVisibility="Hidden" Navigated="NavigationServise\_Navigated" />

</Grid>

</Page>

<Page x:Class="pz9.admin"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:pz9"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="admin" Background="White">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="38"/>

<RowDefinition Height="36"/>

<RowDefinition/>

<RowDefinition Height="60"/>

</Grid.RowDefinitions>

<FlowDocumentReader x:Name="flowDocumentReader" ViewingMode="Scroll" Margin="0,0,0,10" Grid.RowSpan="3">

<FlowDocument PageWidth="Auto">

<BlockUIContainer>

<ListView Name="LViewProduct" Grid.Column="0" Grid.ColumnSpan="3" Grid.Row="2" Margin="5,5,10,5" d:ItemsSource="{d:SampleData ItemCount=1}" SelectionChanged="lViewUsers\_SelectionChanged">

<ListView.ItemContainerStyle>

<Style TargetType="ListViewItem">

<Setter Property="HorizontalContentAlignment" Value="Stretch"/>

</Style>

</ListView.ItemContainerStyle>

<ListView.ItemTemplate>

<DataTemplate>

<Border BorderBrush="Gray" BorderThickness="1" Background="White" Margin="5">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="180"/>

</Grid.ColumnDefinitions>

<Rectangle Width="80" Height="80" Fill="LightGray" VerticalAlignment="Center" Margin="10"/>

<StackPanel Grid.Column="1" Margin="10">

<StackPanel Grid.Column="1" Orientation="Horizontal">

<TextBlock Text="{Binding surname}" FontWeight="Bold" FontSize="14"/>

<TextBlock Text="{Binding name}" FontWeight="Bold" FontSize="14"/>

<TextBlock Text="{Binding patronymic}" FontWeight="Bold" FontSize="14"/>

</StackPanel>

<TextBlock Text="{Binding post.title}" FontWeight="Bold" FontSize="14" />

<TextBlock Text="{Binding phoneNumber}" Foreground="Blue" Margin="0,5,0,0"/>

</StackPanel>

</Grid>

</Border>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</BlockUIContainer>

</FlowDocument>

</FlowDocumentReader>

<Button Content="подробнее" HorizontalAlignment="Left" Margin="10,0,0,0" Grid.Row="3" VerticalAlignment="Center" Click="Button\_Click"/>

<Button x:Name="excelBtn" Content="Выгрузить в Excel" HorizontalAlignment="Left" Margin="318,0,0,0" Grid.Row="3" VerticalAlignment="Center" Width="156" Click="excelBtn\_Click"/>

</Grid>

</Page>

public partial class admin : Page

{

public admin(string role)

{

InitializeComponent();

string a = greeting.Greetingg();

var emp = Entities.GetContext().emp.ToList();

LViewProduct.ItemsSource = emp;

}

private void btnNewEmp(object sender, RoutedEventArgs e)

{

NavigationService.Navigate(new newempl());

}

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

{

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

var selectedItem = LViewProduct.SelectedItem as emp;

if (selectedItem == null)

{

MessageBox.Show("Выберите элемент из списка!");

return;

}

// Переход на страницу detali с передачей выбранного элемента

// detali detailPage = new detali(selectedItem);

NavigationService.Navigate(new detali(selectedItem));

}

private void pectat(object sender, RoutedEventArgs e)

{

var items = LViewProduct.Items.Cast<dynamic>().ToList();

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "PDF файлы (\*.pdf)|\*.pdf";

saveFileDialog.FileName = "output.pdf";

saveFileDialog.DefaultExt = ".pdf";

if (saveFileDialog.ShowDialog() == true)

{

string filePath = saveFileDialog.FileName; // Получаем выбранный путь

try

{

string fontPath = System.IO.Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.Fonts), "arial.ttf");

BaseFont baseFont = BaseFont.CreateFont(fontPath, BaseFont.IDENTITY\_H, BaseFont.EMBEDDED);

Font font = new Font(baseFont, 12, Font.NORMAL);

Document document = new Document();

PdfWriter.GetInstance(document, new FileStream(filePath, FileMode.Create));

document.Open();

document.Add(new iTextSharp.text.Paragraph("Список сотрудников:", font));

foreach (var item in items)

{

string employeeInfo = $"{item.surname} {item.name} {item.patronymic}, {item.post.title}, {item.phoneNumber}";

document.Add(new iTextSharp.text.Paragraph(employeeInfo, font));

}

document.Close();

MessageBox.Show($"PDF успешно создан и сохранен: {filePath}");

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при создании PDF: {ex.Message}");

}

}

}

private void GetEmploymentContractButton\_Click(object sender, RoutedEventArgs e)

{

if (LViewProduct.SelectedItem is emp selectedEmployee)

{

// Создаем окно для заполнения недостающих данных договора

EmploymentContractWindow contractWindow = new EmploymentContractWindow(selectedEmployee);

// Открываем окно как диалог

bool? result = contractWindow.ShowDialog();

}

else

{

MessageBox.Show("Пожалуйста, выберите сотрудника из списка.", "Информация",

MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

private void excelBtn\_Click(object sender, RoutedEventArgs e)

{

var staffAccount = Entities.GetContext().emp.ToList();

var excelApp = new Excel.Application();

excelApp.Visible = true;

excelApp.Workbooks.Add();

Excel.\_Worksheet worksheet = excelApp.ActiveSheet;

worksheet.Cells[1, "A"] = "empID";

worksheet.Cells[1, "B"] = "фамилия";

worksheet.Cells[1, "C"] = "имя";

worksheet.Cells[1, "D"] = "отчество";

worksheet.Cells[1, "E"] = "дата рождения";

worksheet.Cells[1, "F"] = "номер телефона";

worksheet.Cells[1, "G"] = "департамент";

worksheet.Cells[1 , "H"] = "должность";

worksheet.Cells[1, "I"] = "email";

worksheet.Cells[1, "J"] = "адрес";

var row = 1;

foreach(var staff in staffAccount)

{

row++;

worksheet.Cells[row, "A"] = staff.empID;

worksheet.Cells[row, "B"] = staff.surname;

worksheet.Cells[row, "C"] = staff.name;

worksheet.Cells[row, "D"] = staff.patronymic;

worksheet.Cells[row, "E"] = staff.dateOfBirth;

worksheet.Cells[row, "F"] = staff.phoneNumber;

worksheet.Cells[row, "G"] = staff.departm.title;

worksheet.Cells[row, "H"] = staff.post.title;

worksheet.Cells[row, "I"] = staff.email;

worksheet.Cells[row, "J"] = staff.address;

}

worksheet.Columns[1].AutoFit();

worksheet.Columns[2].AutoFit();

worksheet.Columns[3].AutoFit();

worksheet.Columns[4].AutoFit();

worksheet.Columns[5].AutoFit();

worksheet.Columns[6].AutoFit();

worksheet.Columns[7].AutoFit();

worksheet.Columns[8].AutoFit();

worksheet.Columns[9].AutoFit();

worksheet.Columns[10].AutoFit();

}

}

<Page x:Class="pz9.detali"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:pz9"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="detali" Background="White" Height="661">

<Grid Margin="0,0,0,10">

<Grid Margin="20,20,20,0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<!-- Левая колонка с формой -->

<ScrollViewer Grid.Column="0" VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Disabled">

<FlowDocumentReader x:Name="flowDocumentReader" ViewingMode="Scroll" Margin="0,0,0,10" Grid.RowSpan="3">

<FlowDocument PageWidth="Auto">

<BlockUIContainer>

<StackPanel x:Name="SPanel" Margin="20,10,10,10" Orientation="Vertical" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">

<TextBlock Text="Фамилия" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="tb1" Text="{Binding surname}" Margin="0,0,0,15" Padding="5" HorizontalAlignment="Stretch"/>

<TextBlock Text="Имя" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="tb2" Text="{Binding name}" Margin="0,0,0,15" Padding="5" HorizontalAlignment="Stretch"/>

<TextBlock Text="Отчество" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="tb3" Text="{Binding patronymic}" Margin="0,0,0,15" Padding="5" HorizontalAlignment="Stretch"/>

<TextBlock Text="Дата рождения" FontWeight="Bold" Margin="0,0,0,5"/>

<DatePicker x:Name="dateOfBir"

SelectedDate="{Binding dateOfBirth, Mode=TwoWay}"

Margin="0,0,0,15"

Padding="5"

HorizontalAlignment="Stretch"/>

<TextBlock Text="Телефон" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="tb4" Text="{Binding phoneNumber}" Margin="0,0,0,15" Padding="5" HorizontalAlignment="Stretch"/>

<TextBlock Text="Специальность" FontWeight="Bold" Margin="0,0,0,5"/>

<ComboBox x:Name="PositionComboBox"

SelectedValue="{Binding postID, Mode=TwoWay}"

SelectedValuePath="postID"

DisplayMemberPath="title"

Margin="0,5" HorizontalAlignment="Stretch"/>

<TextBlock Text="Департамент" FontWeight="Bold" Margin="0,0,0,5"/>

<ComboBox x:Name="PositionComboBoxx"

SelectedValue="{Binding departamentID, Mode=TwoWay}"

SelectedValuePath="departmID"

DisplayMemberPath="title"

Margin="0,5" HorizontalAlignment="Stretch"/>

<TextBlock Text="E-mail" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="tb5" Text="{Binding email}" Margin="0,0,0,15" Padding="5" HorizontalAlignment="Stretch"/>

</StackPanel>

</BlockUIContainer>

</FlowDocument>

</FlowDocumentReader>

</ScrollViewer>

<!-- Правая колонка с изображением и кнопками -->

<Grid Grid.Column="1" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<!-- Блок с изображением -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Margin="0,0,0,10" Padding="10">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Width="185">

<Image Source="C:\Users\log\source\repos\pz3\pz3\Resources\cf70a9180f90f2cf83982a696912e30e.jpg"

Width="100" Height="100" Margin="0,0,0,10"/>

<Button Content="Добавить фото" Width="134" Padding="5"/>

</StackPanel>

</Border>

<!-- Кнопки "Сохранить" и "Очистить" -->

<StackPanel Grid.Row="1" VerticalAlignment="Bottom" Margin="0,0,0,10">

<Button Content="Сохранить" Width="134" Padding="5" Margin="0,0,0,10" Click="SaveButton\_Click"/>

<Button Content="Очистить" Width="134" Padding="5" Click="ClearButton\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Grid>

</Page>

using iTextSharp.text.pdf;

using iTextSharp.text;

using pz9.model;

using System;

using System.Collections.Generic;

using System.IO;

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 Microsoft.Win32; // Для использования SaveFileDialog

namespace pz9

{

/// <summary>

/// Логика взаимодействия для detali.xaml

/// </summary>

public partial class detali : Page

{

private Entities \_context;

public detali(object selectedItem)

{

InitializeComponent();

\_context = new Entities();

this.DataContext = selectedItem;

LoadPositions();

LoadDepartments();

}

private void LoadPositions()

{

var positions = \_context.post.ToList();

PositionComboBox.ItemsSource = positions;

}

private void LoadDepartments()

{

// Загрузка данных из таблицы departm

var departments = \_context.departm.ToList();

// Установка данных в ComboBox

PositionComboBoxx.ItemsSource = departments;

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

try

{

// Получаем текущий объект пользователя

var user = this.DataContext as emp;

if (user == null)

{

MessageBox.Show("Ошибка: данные пользователя не загружены.");

return;

}

// Находим пользователя в базе данных

var userInDb = \_context.emp.FirstOrDefault(u => u.empID == user.empID);

if (userInDb == null)

{

MessageBox.Show("Ошибка: пользователь не найден в базе данных.");

return;

}

// Обновляем данные пользователя

userInDb.surname = user.surname;

userInDb.name = user.name;

userInDb.patronymic = user.patronymic;

userInDb.phoneNumber = user.phoneNumber;

userInDb.postID = user.postID;

userInDb.departamentID = user.departamentID;

userInDb.dateOfBirth = user.dateOfBirth;

userInDb.email = user.email;

// Сохраняем изменения в базе данных

\_context.SaveChanges();

MessageBox.Show("Изменения успешно сохранены.");

}

catch (System.Data.Entity.Validation.DbEntityValidationException ex)

{

foreach (var validationErrors in ex.EntityValidationErrors)

{

foreach (var validationError in validationErrors.ValidationErrors)

{

MessageBox.Show($"Error: {validationError.ErrorMessage}");

}

}

}

catch (Exception ex)

{

MessageBox.Show($"Ошибка при сохранении данных: {ex.Message}");

}

}

private void ClearButton\_Click(object sender, RoutedEventArgs e)

{

var user = this.DataContext as emp;

//if (user != null)

//{

// user.surname = string.Empty;

// user.name = string.Empty;

// user.patronymic = string.Empty;

// user.phoneNumber = string.Empty;

// user.postID = 0;

// user.departamentID = 0;

//}

tb1.Text = "";

tb2.Text = "";

tb3.Text = "";

tb4.Text = "";

PositionComboBox.SelectedIndex = -1;

PositionComboBoxx.SelectedIndex = -1;

this.DataContext = user;

}

}

}

<Window x:Class="pz9.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:pz9"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="65"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid Grid.Row="0" Background="#FF76E383">

<Image Width="45" Height="45" HorizontalAlignment="Left" Margin="5"/>

<Button x:Name="btnBack" HorizontalAlignment="Right" Content="Назад" Click="btnBack\_Click"/>

<Image HorizontalAlignment="Left" Height="47" VerticalAlignment="Top" Width="50" Margin="10,10,0,0"/>

</Grid>

<Frame x:Name="FrmMain" Grid.Row="1" NavigationUIVisibility="Hidden" ContentRendered="FrmMain\_ContentRendered" Navigated="FrmMain\_Navigated"/>

</Grid>

</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;

namespace pz9

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

FrmMain.Navigate(new avtoriz());

}

private void FrmMain\_Navigated(object sender, NavigationEventArgs e)

{

}

private void FrmMain\_ContentRendered(object sender, EventArgs e)

{

if (FrmMain.CanGoBack)

btnBack.Visibility = Visibility.Visible;

else

btnBack.Visibility = Visibility.Hidden;

}

private void btnBack\_Click(object sender, RoutedEventArgs e)

{

FrmMain.GoBack();

}

}

}

<Page x:Class="pz9.newempl"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:pz9"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="newempl" Background="White" Height="718">

<Grid Margin="0,0,0,10">

<Grid Margin="20">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="2\*"/>

<ColumnDefinition Width="1\*"/>

</Grid.ColumnDefinitions>

<!-- Левая колонка с формой -->

<ScrollViewer Grid.Column="0" VerticalScrollBarVisibility="Auto" HorizontalScrollBarVisibility="Disabled">

<StackPanel x:Name="SPanel1" Orientation="Vertical" Margin="20,10,10,10" VerticalAlignment="Stretch" HorizontalAlignment="Stretch">

<TextBlock Text="Фамилия" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="surname" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Имя" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="Name" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Отчество" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="patronymic" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Дата рождения" FontWeight="Bold" Margin="0,0,0,5"/>

<DatePicker x:Name="DPbd" Margin="0,0,0,15" Padding="5" Height="51"/>

<TextBlock Text="Телефон" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="phoneNumber" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Специальность" FontWeight="Bold" Margin="0,0,0,5"/>

<ComboBox x:Name="PositionComboBox" Margin="0,5">

<ComboBoxItem Content="Менеджер" />

<ComboBoxItem Content="Инженер" />

<ComboBoxItem Content="Aдминистратор" />

</ComboBox>

<TextBlock Text="Департамент" FontWeight="Bold" Margin="0,0,0,5"/>

<ComboBox x:Name="PositionComboBoxx" Margin="0,5">

<ComboBoxItem Content="Горно-добывающий департамент" />

<ComboBoxItem Content="Департамент обогащения и переработки" />

<ComboBoxItem Content="Технико-экологический департамент" />

</ComboBox>

<TextBlock Text="E-mail" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="email" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Логин" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="login" Margin="0,0,0,15" Padding="5"/>

<TextBlock Text="Пароль" FontWeight="Bold" Margin="0,0,0,5"/>

<TextBox x:Name="password" Margin="0,0,0,15" Padding="5"/>

</StackPanel>

</ScrollViewer>

<!-- Правая колонка с изображением и кнопками -->

<Grid Grid.Column="1" Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<!-- Блок с изображением -->

<Border BorderBrush="Gray" BorderThickness="1" CornerRadius="5" Margin="0,0,0,10" Padding="10">

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Center" Width="185">

<Image Source="C:\Users\log\source\repos\pz3\pz3\Resources\cf70a9180f90f2cf83982a696912e30e.jpg"

Width="100" Height="100" Margin="0,0,0,10"/>

<Button Content="Добавить фото" Width="134" Padding="5"/>

</StackPanel>

</Border>

<!-- Кнопки "Сохранить" и "Очистить" -->

<StackPanel Grid.Row="1" VerticalAlignment="Bottom" Margin="0,0,0,10">

<Button Content="Сохранить" Width="134" Padding="5" Margin="0,0,0,10" Click="SaveButton\_Click"/>

<Button Content="Очистить" Width="134" Padding="5" Click="ClearButton\_Click"/>

</StackPanel>

</Grid>

</Grid>

</Grid>

</Page>

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Runtime.Remoting.Contexts;

using System.Security.Policy;

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 pz9.model;

using pz9.Properties;

using pz9.serv;

namespace pz9

{

/// <summary>

/// Логика взаимодействия для newempl.xaml

/// </summary>

public partial class newempl : Page

{

private static Entities \_context;

public newempl()

{

InitializeComponent();

\_context = new Entities();

}

private void SaveButton\_Click(object sender, RoutedEventArgs e)

{

var newEmployee = new emp

{

surname = surname.Text,

name = Name.Text,

patronymic = patronymic.Text,

dateOfBirth = DPbd.SelectedDate ?? DateTime.MinValue,

phoneNumber = phoneNumber.Text,

departamentID = PositionComboBoxx.SelectedIndex + 1,

postID = PositionComboBox.SelectedIndex + 1,

email = email.Text,

login = login.Text,

password = hash.HashPassword(password.Text),

dateOfEmployment = DateTime.Now

};

var context = new ValidationContext(newEmployee);

var results = new List<System.ComponentModel.DataAnnotations.ValidationResult>();

if (PositionComboBox.SelectedIndex == -1)

{

results.Add(new System.ComponentModel.DataAnnotations.ValidationResult("Выберите специальность.", new[] { nameof(newEmployee.postID) }));

}

if (PositionComboBoxx.SelectedIndex == -1)

{

results.Add(new System.ComponentModel.DataAnnotations.ValidationResult("Выберите департамент.", new[] { nameof(newEmployee.departamentID) }));

}

if (!Validator.TryValidateObject(newEmployee, context, results, true))

{

StringBuilder errorMessage = new StringBuilder("Не удалось сохранить данные:\n");

foreach (var error in results)

{

errorMessage.AppendLine(error.ErrorMessage);

}

MessageBox.Show(errorMessage.ToString(), "Ошибка валидации", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

using (var db = new Entities())

{

int maxEmpID = db.emp.Any() ? db.emp.Max(emp => emp.empID) : 0;

newEmployee.empID = maxEmpID + 1;

db.emp.Add(newEmployee);

db.SaveChanges();

}

MessageBox.Show("Данные успешно сохранены!", "Успех", MessageBoxButton.OK, MessageBoxImage.Information);

NavigationService.GoBack();

}

//try

//{

//}

//catch (Exception ex)

//{

// string errorMessage = $"Произошла ошибка: {ex.Message}";

// if (ex.InnerException != null)

// {

// errorMessage += $"\nВнутреннее исключение: {ex.InnerException.Message}";

// }

// MessageBox.Show(errorMessage, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

//}

}

private void ClearButton\_Click(object sender, RoutedEventArgs e)

{

var user = this.DataContext as emp;

//if (user != null)

//{

// user.surname = string.Empty;

// user.name = string.Empty;

// user.patronymic = string.Empty;

// user.phoneNumber = string.Empty;

// user.postID = 0;

// user.departamentID = 0;

//}

surname.Text = "";

Name.Text = "";

patronymic.Text = "";

phoneNumber.Text = "";

PositionComboBox.SelectedIndex = -1;

PositionComboBoxx.SelectedIndex = -1;

login.Text = "";

password.Text ="";

this.DataContext = user;

}

}

}