Colors.xaml

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

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

<SolidColorBrush x:Key="BackgroundWindowColor" Color="#F2F2F2"/>

<SolidColorBrush x:Key="BackgroundMenuColor" Color="#8A8C99"/>

<SolidColorBrush x:Key="BackgroundHeaderColor" Color="#424C55"/>

<SolidColorBrush x:Key="MouseOverColor" Color="#D1CCDC"/>

<SolidColorBrush x:Key="ActiveStateColor" Color="#FFFFFF"/>

<SolidColorBrush x:Key="TitleWindowColor" Color="#F5EDF0"/>

<SolidColorBrush x:Key="CloseHoverColor" Color="#E81123"/>

<SolidColorBrush x:Key="TestButtonHoverColor" Color="#5f6e7b"/>

<SolidColorBrush x:Key="TestButtonPressColor" Color="#1a1e21"/>

</ResourceDictionary>

Fonts.xaml

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

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

<FontFamily x:Key="Mulish">pack://application:,,,/Fonts/#Mulish</FontFamily>

</ResourceDictionary>

Icons.xaml

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

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

<Path Fill="Black" x:Key="MenuCloseIcon" Data="M3.75 22.5V20H20V22.5H3.75ZM24.5 21.25L18.25 15L24.5 8.75L26.25 10.5L21.75 15L26.25 19.5L24.5 21.25ZM3.75 16.25V13.75H16.25V16.25H3.75ZM3.75 10V7.5H20V10H3.75Z"/>

<Path Fill="Black" x:Key="MenuIcon" Data="M3.75 22.5V20H26.25V22.5H3.75ZM3.75 16.25V13.75H26.25V16.25H3.75ZM3.75 10V7.5H26.25V10H3.75Z" />

<Path Fill="Black" x:Key="CopyIcon" Data="M360-240q-33 0-56.5-23.5T280-320v-480q0-33 23.5-56.5T360-880h360q33 0 56.5 23.5T800-800v480q0 33-23.5 56.5T720-240H360Zm0-80h360v-480H360v480ZM200-80q-33 0-56.5-23.5T120-160v-560h80v560h440v80H200Zm160-240v-480 480Z" />

<Path Fill="Black" x:Key="MinimizeIcon" Data="M7.5 26.25V23.75H22.5V26.25H7.5Z" />

<Path Fill="Black" x:Key="ContentCopyIcon" Data="M9 18C8.45 18 7.97917 17.8042 7.5875 17.4125C7.19583 17.0208 7 16.55 7 16V4C7 3.45 7.19583 2.97917 7.5875 2.5875C7.97917 2.19583 8.45 2 9 2H18C18.55 2 19.0208 2.19583 19.4125 2.5875C19.8042 2.97917 20 3.45 20 4V16C20 16.55 19.8042 17.0208 19.4125 17.4125C19.0208 17.8042 18.55 18 18 18H9ZM9 16H18V4H9V16ZM5 22C4.45 22 3.97917 21.8042 3.5875 21.4125C3.19583 21.0208 3 20.55 3 20V6H5V20H16V22H5Z" />

<Path Fill="Black" x:Key="FullScreenExitIcon" Data="M4.5 23.25V19.5H0.75V17H7V23.25H4.5ZM17 23.25V17H23.25V19.5H19.5V23.25H17ZM0.75 7V4.5H4.5V0.75H7V7H0.75ZM17 7V0.75H19.5V4.5H23.25V7H17Z"/>

<Path Fill="Black" x:Key="FullScreenIcon" Data="M3.75 26.25V20H6.25V23.75H10V26.25H3.75ZM20 26.25V23.75H23.75V20H26.25V26.25H20ZM3.75 10V3.75H10V6.25H6.25V10H3.75ZM23.75 10V6.25H20V3.75H26.25V10H23.75Z"/>

<Path Fill="Black" x:Key="CloseIcon" Data="M8 23.75L6.25 22L13.25 15L6.25 8L8 6.25L15 13.25L22 6.25L23.75 8L16.75 15L23.75 22L22 23.75L15 16.75L8 23.75Z"/>

<Path Fill="Black" x:Key="TestIcon" Data="M3.25 25.25C2.5625 25.25 1.97396 25.0052 1.48438 24.5156C0.994792 24.026 0.75 23.4375 0.75 22.75V5.25C0.75 4.5625 0.994792 3.97396 1.48438 3.48438C1.97396 2.99479 2.5625 2.75 3.25 2.75H8.5C8.77083 2 9.22396 1.39583 9.85938 0.9375C10.4948 0.479167 11.2083 0.25 12 0.25C12.7917 0.25 13.5052 0.479167 14.1406 0.9375C14.776 1.39583 15.2292 2 15.5 2.75H20.75C21.4375 2.75 22.026 2.99479 22.5156 3.48438C23.0052 3.97396 23.25 4.5625 23.25 5.25V22.75C23.25 23.4375 23.0052 24.026 22.5156 24.5156C22.026 25.0052 21.4375 25.25 20.75 25.25H3.25ZM3.25 22.75H20.75V5.25H3.25V22.75ZM5.75 20.25H14.5V17.75H5.75V20.25ZM5.75 15.25H18.25V12.75H5.75V15.25ZM5.75 10.25H18.25V7.75H5.75V10.25ZM12 4.3125C12.2708 4.3125 12.4948 4.22396 12.6719 4.04688C12.849 3.86979 12.9375 3.64583 12.9375 3.375C12.9375 3.10417 12.849 2.88021 12.6719 2.70312C12.4948 2.52604 12.2708 2.4375 12 2.4375C11.7292 2.4375 11.5052 2.52604 11.3281 2.70312C11.151 2.88021 11.0625 3.10417 11.0625 3.375C11.0625 3.64583 11.151 3.86979 11.3281 4.04688C11.5052 4.22396 11.7292 4.3125 12 4.3125Z"/>

<Path Fill="Black" x:Key="ArrowBackIcon" Data="M7.825 13L13.425 18.6L12 20L4 12L12 4L13.425 5.4L7.825 11H20V13H7.825Z" />

<Path Fill="Black" x:Key="QuestionMarkIcon" Data="M14.9375 22.5C15.375 22.5 15.7448 22.349 16.0469 22.0469C16.349 21.7448 16.5 21.375 16.5 20.9375C16.5 20.5 16.349 20.1302 16.0469 19.8281C15.7448 19.526 15.375 19.375 14.9375 19.375C14.5 19.375 14.1302 19.526 13.8281 19.8281C13.526 20.1302 13.375 20.5 13.375 20.9375C13.375 21.375 13.526 21.7448 13.8281 22.0469C14.1302 22.349 14.5 22.5 14.9375 22.5ZM13.8125 17.6875H16.125C16.125 17 16.2031 16.4583 16.3594 16.0625C16.5156 15.6667 16.9583 15.125 17.6875 14.4375C18.2292 13.8958 18.6562 13.3802 18.9688 12.8906C19.2812 12.401 19.4375 11.8125 19.4375 11.125C19.4375 9.95833 19.0104 9.0625 18.1562 8.4375C17.3021 7.8125 16.2917 7.5 15.125 7.5C13.9375 7.5 12.974 7.8125 12.2344 8.4375C11.4948 9.0625 10.9792 9.8125 10.6875 10.6875L12.75 11.5C12.8542 11.125 13.0885 10.7188 13.4531 10.2812C13.8177 9.84375 14.375 9.625 15.125 9.625C15.7917 9.625 16.2917 9.80729 16.625 10.1719C16.9583 10.5365 17.125 10.9375 17.125 11.375C17.125 11.7917 17 12.1823 16.75 12.5469C16.5 12.9115 16.1875 13.25 15.8125 13.5625C14.8958 14.375 14.3333 14.9896 14.125 15.4062C13.9167 15.8229 13.8125 16.5833 13.8125 17.6875ZM15 27.5C13.2708 27.5 11.6458 27.1719 10.125 26.5156C8.60417 25.8594 7.28125 24.9688 6.15625 23.8438C5.03125 22.7188 4.14062 21.3958 3.48438 19.875C2.82812 18.3542 2.5 16.7292 2.5 15C2.5 13.2708 2.82812 11.6458 3.48438 10.125C4.14062 8.60417 5.03125 7.28125 6.15625 6.15625C7.28125 5.03125 8.60417 4.14062 10.125 3.48438C11.6458 2.82812 13.2708 2.5 15 2.5C16.7292 2.5 18.3542 2.82812 19.875 3.48438C21.3958 4.14062 22.7188 5.03125 23.8438 6.15625C24.9688 7.28125 25.8594 8.60417 26.5156 10.125C27.1719 11.6458 27.5 13.2708 27.5 15C27.5 16.7292 27.1719 18.3542 26.5156 19.875C25.8594 21.3958 24.9688 22.7188 23.8438 23.8438C22.7188 24.9688 21.3958 25.8594 19.875 26.5156C18.3542 27.1719 16.7292 27.5 15 27.5ZM15 25C17.7917 25 20.1562 24.0312 22.0938 22.0938C24.0312 20.1562 25 17.7917 25 15C25 12.2083 24.0312 9.84375 22.0938 7.90625C20.1562 5.96875 17.7917 5 15 5C12.2083 5 9.84375 5.96875 7.90625 7.90625C5.96875 9.84375 5 12.2083 5 15C5 17.7917 5.96875 20.1562 7.90625 22.0938C9.84375 24.0312 12.2083 25 15 25Z"/>

<Path Fill="Black" x:Key="InfoIcon" Data="M11 17H13V11H11V17ZM12 9C12.2833 9 12.5208 8.90417 12.7125 8.7125C12.9042 8.52083 13 8.28333 13 8C13 7.71667 12.9042 7.47917 12.7125 7.2875C12.5208 7.09583 12.2833 7 12 7C11.7167 7 11.4792 7.09583 11.2875 7.2875C11.0958 7.47917 11 7.71667 11 8C11 8.28333 11.0958 8.52083 11.2875 8.7125C11.4792 8.90417 11.7167 9 12 9ZM12 22C10.6167 22 9.31667 21.7375 8.1 21.2125C6.88333 20.6875 5.825 19.975 4.925 19.075C4.025 18.175 3.3125 17.1167 2.7875 15.9C2.2625 14.6833 2 13.3833 2 12C2 10.6167 2.2625 9.31667 2.7875 8.1C3.3125 6.88333 4.025 5.825 4.925 4.925C5.825 4.025 6.88333 3.3125 8.1 2.7875C9.31667 2.2625 10.6167 2 12 2C13.3833 2 14.6833 2.2625 15.9 2.7875C17.1167 3.3125 18.175 4.025 19.075 4.925C19.975 5.825 20.6875 6.88333 21.2125 8.1C21.7375 9.31667 22 10.6167 22 12C22 13.3833 21.7375 14.6833 21.2125 15.9C20.6875 17.1167 19.975 18.175 19.075 19.075C18.175 19.975 17.1167 20.6875 15.9 21.2125C14.6833 21.7375 13.3833 22 12 22ZM12 20C14.2333 20 16.125 19.225 17.675 17.675C19.225 16.125 20 14.2333 20 12C20 9.76667 19.225 7.875 17.675 6.325C16.125 4.775 14.2333 4 12 4C9.76667 4 7.875 4.775 6.325 6.325C4.775 7.875 4 9.76667 4 12C4 14.2333 4.775 16.125 6.325 17.675C7.875 19.225 9.76667 20 12 20Z" />

</ResourceDictionary>

Styles.xaml

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

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

xmlns:gc="clr-namespace:ConnectingDBToApp.GlobalClasses">

<!--Global styles-->

<Style TargetType="TextBlock">

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

</Style>

<Style TargetType="TabItem">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="Foreground" Value="{DynamicResource MouseOverColor}"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Padding" Value="20 12"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TabItem">

<Border Background="{TemplateBinding Background}"

Padding="{TemplateBinding Padding}"

CornerRadius="5"

Margin="0 0 5 0">

<ContentPresenter

HorizontalAlignment="Center"

VerticalAlignment="Center"

ContentSource="Header"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="{DynamicResource MouseOverColor}"/>

<Setter Property="Foreground" Value="Black"/>

</Trigger>

<Trigger Property="IsSelected" Value="True">

<Setter Property="Background" Value="{DynamicResource BackgroundHeaderColor}"/>

<Setter Property="Foreground" Value="{DynamicResource TitleWindowColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="TabControl">

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TabControl">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0"

IsItemsHost="True"

Margin="0 0 0 20"/>

<Border Grid.Row="1">

<ContentPresenter ContentSource="SelectedContent"/>

</Border>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="ScrollBarThumbStyle" TargetType="Thumb">

<Setter Property="Background" Value="{DynamicResource MouseOverColor}"/>

<Setter Property="Focusable" Value="False"/>

<Setter Property="IsTabStop" Value="False"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Thumb">

<Border CornerRadius="5"

Background="{TemplateBinding Background}"

Margin="3 5"/>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Background" Value="{DynamicResource BackgroundMenuColor}"/>

</Trigger>

<Trigger Property="IsDragging" Value="True">

<Setter Property="Background" Value="{DynamicResource BackgroundHeaderColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="ScrollBar">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ScrollBar">

<Track x:Name="PART\_Track" IsDirectionReversed="True" Grid.Row="1">

<Track.Thumb>

<Thumb Style="{StaticResource ScrollBarThumbStyle}"/>

</Track.Thumb>

</Track>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="PrimaryButton" TargetType="Button">

<Setter Property="Background" Value="{DynamicResource BackgroundHeaderColor}"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="FontWeight" Value="Black"/>

<Setter Property="Padding" Value="30 15"/>

<Setter Property="HorizontalAlignment" Value="Center"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

CornerRadius="5"

Background="{TemplateBinding Background}"

Padding="{TemplateBinding Padding}">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background"

Value="{DynamicResource TestButtonHoverColor}"/>

<Setter Property="Cursor" Value="Hand"/>

</Trigger>

<Trigger Property="IsPressed" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background"

Value="{DynamicResource TestButtonPressColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="SecondaryButton" TargetType="Button">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="Foreground" Value="{StaticResource BackgroundHeaderColor}"/>

<Setter Property="Padding" Value="30 15"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

BorderThickness="2"

CornerRadius="5"

Background="{TemplateBinding Background}"

BorderBrush="{TemplateBinding Foreground}"

Padding="{TemplateBinding Padding}">

<ContentPresenter HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsEnabled" Value="False">

<Setter Property="BorderBrush" Value="{StaticResource MouseOverColor}" TargetName="BackgroundElement"/>

<Setter Property="Foreground" Value="{StaticResource MouseOverColor}"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Cursor" Value="Hand"/>

<Setter TargetName="BackgroundElement"

Property="BorderBrush" Value="{StaticResource BackgroundMenuColor}"/>

<Setter Property="Foreground" Value="{StaticResource BackgroundMenuColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!--End: Global styles-->

<!--Header style-->

<Style x:Key="TitleWindow" TargetType="TextBlock">

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Foreground" Value="{DynamicResource TitleWindowColor}"/>

<Setter Property="Text" Value="Технология подключения базы данных к настольному приложению"/>

<Setter Property="Margin" Value="10 0 0 0"/>

</Style>

<Style x:Key="MenuButton" TargetType="ToggleButton">

<Setter Property="Width" Value="40"/>

<Setter Property="Height" Value="40"/>

<Setter Property="ToolTip" Value="Скрыть меню"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ToggleButton">

<Border x:Name="BackgroundElement">

<Rectangle x:Name="Icon"

Width="22.5" Height="15"

Fill="{DynamicResource TitleWindowColor}">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource MenuCloseIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource BackgroundMenuColor}"/>

</Trigger>

<Trigger Property="IsChecked" Value="True">

<Setter TargetName="Icon"

Property="OpacityMask">

<Setter.Value>

<VisualBrush Stretch="Uniform" Visual="{StaticResource MenuIcon}"/>

</Setter.Value>

</Setter>

<Setter Property="ToolTip" Value="Показать меню"/>

</Trigger>

<Trigger Property="IsEnabled" Value="False">

<Setter TargetName="Icon"

Property="Fill" Value="{DynamicResource BackgroundHeaderColor}"/>

<Setter TargetName="Icon"

Property="OpacityMask">

<Setter.Value>

<VisualBrush Stretch="Uniform" Visual="{StaticResource MenuIcon}"/>

</Setter.Value>

</Setter>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="MinimizeButton" TargetType="Button">

<Setter Property="Width" Value="40"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

Background="Transparent"

Padding="0 20 0 0">

<Rectangle x:Name="Icon"

Width="15" Height="5"

Fill="{DynamicResource TitleWindowColor}">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform"

Visual="{StaticResource MinimizeIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource BackgroundMenuColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="FullscreenButton" TargetType="Button">

<Setter Property="Width" Value="40"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement">

<Rectangle x:Name="Icon"

Width="22.5" Height="22.5"

Fill="{DynamicResource TitleWindowColor}">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource FullScreenIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource BackgroundMenuColor}"/>

</Trigger>

<DataTrigger Binding="{Binding WindowState, RelativeSource={RelativeSource AncestorType=Window}}"

Value="Normal">

<Setter TargetName="Icon"

Property="OpacityMask">

<Setter.Value>

<VisualBrush Stretch="Uniform" Visual="{StaticResource FullScreenIcon}"/>

</Setter.Value>

</Setter>

<Setter Property="ToolTip" Value="Развернуть"/>

</DataTrigger>

<DataTrigger Binding="{Binding WindowState, RelativeSource={RelativeSource AncestorType=Window}}"

Value="Maximized">

<Setter TargetName="Icon"

Property="OpacityMask">

<Setter.Value>

<VisualBrush Stretch="Uniform" Visual="{StaticResource FullScreenExitIcon}"/>

</Setter.Value>

</Setter>

<Setter Property="ToolTip" Value="Свернуть в окно"/>

</DataTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="CloseButton" TargetType="Button">

<Setter Property="Width" Value="40"/>

<Setter Property="Height" Value="40"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement">

<Rectangle x:Name="Icon"

Width="17.5" Height="17.5"

Fill="{DynamicResource TitleWindowColor}">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource CloseIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource CloseHoverColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!--End: Header style-->

<!--Sidebar style-->

<Style x:Key="SideBarItem" TargetType="ListBoxItem">

<Setter Property="FontSize" Value="18"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ListBoxItem">

<Border x:Name="BackgroundElement"

CornerRadius="5"

Padding="15 15 0 15"

Margin="0 0 0 10"

Background="Transparent">

<StackPanel Orientation="Horizontal">

<Border x:Name="Icon"

Width="25" Height="25"

Margin="0 0 10 0"

Background="{TemplateBinding Background}">

</Border>

<ContentPresenter/>

</StackPanel>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource MouseOverColor}"/>

</Trigger>

<Trigger Property="IsSelected" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource ActiveStateColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="AboutBoxStyle" TargetType="Button">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="FontSize" Value="16"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border Background="{TemplateBinding Background}">

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center">

<Rectangle x:Name="Icon"

Fill="Black"

Width="20" Height="20"

VerticalAlignment="Center"

Margin="0 0 10 0">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource InfoIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

<TextBlock Text="О программе" VerticalAlignment="Center"/>

</StackPanel>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="Icon"

Property="Fill" Value="{StaticResource MouseOverColor}"/>

<Setter Property="Foreground" Value="{StaticResource MouseOverColor}"/>

<Setter Property="Cursor" Value="Hand"/>

</Trigger>

<Trigger Property="IsPressed" Value="True">

<Setter TargetName="Icon"

Property="Fill" Value="{StaticResource ActiveStateColor}"/>

<Setter Property="Foreground" Value="{StaticResource ActiveStateColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="AboutBoxButton" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ListBoxItem">

<Button Style="{StaticResource AboutBoxStyle}" Command="{Binding DataContext.OpenAboutBoxCommand, RelativeSource={RelativeSource AncestorType=Window}}"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="SideBar" TargetType="ListBox">

<Setter Property="Width" Value="230"/>

<Setter Property="Background" Value="{DynamicResource BackgroundMenuColor}"/>

<Setter Property="Padding" Value="10 30 10 0"/>

<Setter Property="ItemsPanel">

<Setter.Value>

<ItemsPanelTemplate>

<DockPanel LastChildFill="False"/>

</ItemsPanelTemplate>

</Setter.Value>

</Setter>

</Style>

<!--End: Sidebar style-->

<!--Styles of items for style selector-->

<Style x:Key="H1" TargetType="TextBlock">

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

<Setter Property="FontSize" Value="36"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Margin" Value="0 10 0 30"/>

<Setter Property="TextWrapping" Value="Wrap"/>

</Style>

<Style x:Key="H2" TargetType="TextBlock">

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

<Setter Property="FontSize" Value="24"/>

<Setter Property="FontWeight" Value="Bold"/>

<Setter Property="Margin" Value="0 15 0 15"/>

<Setter Property="TextWrapping" Value="Wrap"/>

</Style>

<Style x:Key="PlainText" TargetType="TextBlock">

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="FontWeight" Value="Regular"/>

<Setter Property="Margin" Value="0 10"/>

<Setter Property="TextWrapping" Value="Wrap"/>

</Style>

<Style x:Key="H1Item" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<TextBlock x:Name="TextBlockContent"

Text="{Binding Text}"

Style="{StaticResource H1}"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="H2Item" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<TextBlock x:Name="TextBlockContent"

Text="{Binding Text}"

Style="{StaticResource H2}"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="PlainTextItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<Border>

<TextBlock x:Name="TextBlockContent"

Text="{Binding Text}"

Style="{StaticResource PlainText}"/>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="ImageItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<Image Source="{Binding Text}"

Margin="0 5"

MaxWidth="600"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="DoubleImageItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<UniformGrid Columns="2"

Margin="0 5">

<Image Source="{Binding Text}"

Margin="0 0 5 0"/>

<Image Source="{Binding AdditionalText}"/>

</UniformGrid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="CopyButtonStyle" TargetType="Button">

<Style.Resources>

<Style x:Key="BackgroundElement" TargetType="Border">

<Setter Property="Background" Value="Transparent"/>

<Setter Property="CornerRadius" Value="5"/>

</Style>

</Style.Resources>

<Setter Property="Width" Value="30"/>

<Setter Property="Height" Value="30"/>

<Setter Property="ToolTip" Value="Скопировать текст"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

Style="{StaticResource BackgroundElement}">

<Rectangle x:Name="Icon"

Width="22.5" Height="22.5"

Fill="Black">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource ContentCopyIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background"

Value="{DynamicResource MouseOverColor}"/>

<Setter Property="Cursor" Value="Hand"/>

</Trigger>

<Trigger Property="IsPressed" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background"

Value="{DynamicResource BackgroundHeaderColor}"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="CopyButtonForImageStyle" TargetType="Button">

<Setter Property="Width" Value="30"/>

<Setter Property="Height" Value="30"/>

<Setter Property="ToolTip" Value="Скопировать код"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

Background="Transparent"

CornerRadius="5">

<Rectangle x:Name="Icon"

Width="22.5" Height="22.5"

Fill="White">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource ContentCopyIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="White"/>

<Setter TargetName="Icon"

Property="Fill" Value="Black"/>

<Setter Property="Cursor" Value="Hand"/>

</Trigger>

<Trigger Property="IsPressed" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="Black"/>

<Setter TargetName="Icon"

Property="Fill" Value="White"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="CopyButtonItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button Grid.Column="0"

Style="{StaticResource CopyButtonStyle}"

Content="Скопировать"

Command="{Binding DataContext.CopyTextCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding Code}"

Margin="10 0 5 0"/>

<TextBlock Grid.Column="1"

Style="{StaticResource PlainText}"

Text="{Binding Text}"

VerticalAlignment="Center"/>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="CodeImageItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<Border Margin="0 5"

MaxWidth="600">

<Grid>

<Image Source="{Binding Text}"/>

<Button Content="Скопировать код"

Command="{Binding DataContext.CopyTextCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding Code}"

Style="{StaticResource CopyButtonForImageStyle}"

VerticalAlignment="Top"

HorizontalAlignment="Right"

Margin="0 30 30 0"/>

</Grid>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="HyperLinkItem" TargetType="ListBoxItem">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate>

<TextBlock>

<Hyperlink Command="{Binding DataContext.NavigateLinkCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding Code}">

<TextBlock Text="{Binding Text}"/>

</Hyperlink>

</TextBlock>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!--End: Styles of items for style selector-->

<!--Data page style-->

<Style x:Key="DataListBox" TargetType="ListBox">

<Setter Property="ScrollViewer.HorizontalScrollBarVisibility" Value="Disabled"/>

<Setter Property="ScrollViewer.VerticalScrollBarVisibility" Value="Disabled"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="Margin" Value="30 0"/>

<Setter Property="MaxWidth" Value="850"/>

<Setter Property="ItemContainerStyleSelector">

<Setter.Value>

<gc:CustomStyleSelector

H1Item="{StaticResource H1Item}"

H2Item="{StaticResource H2Item}"

ImageItem="{StaticResource ImageItem}"

DoubleImageItem="{StaticResource DoubleImageItem}"

PlainTextItem="{StaticResource PlainTextItem}"

CopyButtonItem="{StaticResource CopyButtonItem}"

CodeImageItem="{StaticResource CodeImageItem}"

HyperLinkItem="{StaticResource HyperLinkItem}"/>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="NextPartHyperlink" TargetType="Hyperlink">

<Setter Property="TextDecorations" Value="None"/>

<Setter Property="Foreground" Value="{StaticResource MouseOverColor}"/>

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Foreground" Value="{StaticResource BackgroundHeaderColor}"/>

</Trigger>

</Style.Triggers>

</Style>

<!--End: Data page style-->

<!--Styles of the test section-->

<Style x:Key="VariantRadioButton" TargetType="RadioButton">

<Setter Property="FontSize" Value="20"/>

<Setter Property="Margin" Value="0 0 0 20"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="RadioButton">

<BulletDecorator>

<BulletDecorator.Bullet>

<Grid>

<Ellipse x:Name="Border"

Width="20" Height="20"

Fill="{StaticResource BackgroundHeaderColor}"/>

<Ellipse x:Name="CheckMark"

Width="14" Height="14"

StrokeThickness="2"

Stroke="White"

Fill="White"/>

</Grid>

</BulletDecorator.Bullet>

<ContentPresenter Margin="15 0 0 0"/>

</BulletDecorator>

<ControlTemplate.Triggers>

<Trigger Property="IsChecked" Value="True">

<Setter Property="Fill" Value="Transparent"

TargetName="CheckMark"/>

</Trigger>

<Trigger Property="IsMouseOver" Value="True">

<Setter Property="Cursor" Value="Hand"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="UsernameTextBox" TargetType="TextBox">

<Setter Property="Width" Value="200"/>

<Setter Property="Padding" Value="5 3 0 3"/>

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

<Setter Property="FontSize" Value="18"/>

<Setter Property="FontFamily" Value="{StaticResource Mulish}"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="TextBoxBase">

<Border x:Name="Border"

BorderThickness="1"

BorderBrush="{DynamicResource BackgroundMenuColor}"

CornerRadius="5"

Padding="{TemplateBinding Padding}">

<Grid>

<TextBlock x:Name="Placeholder"

Text="Username"

Foreground="{StaticResource BackgroundMenuColor}"

VerticalAlignment="Center"

Margin="5 0 0 0"

Visibility="Collapsed"/>

<ScrollViewer x:Name="PART\_ContentHost"

VerticalAlignment="Center"/>

</Grid>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsFocused" Value="True">

<Setter Property="BorderBrush" Value="{StaticResource BackgroundHeaderColor}"

TargetName="Border"/>

<Setter Property="Visibility" Value="Collapsed"

TargetName="Placeholder"/>

</Trigger>

<DataTrigger Binding="{Binding Text}" Value="">

<Setter Property="Visibility" Value="Visible"

TargetName="Placeholder"/>

</DataTrigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="ArrowBackButton" TargetType="Button">

<Setter Property="Width" Value="40"/>

<Setter Property="Height" Value="40"/>

<Setter Property="BorderThickness" Value="0"/>

<Setter Property="BorderBrush" Value="Black"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="Button">

<Border x:Name="BackgroundElement"

Background="White"

CornerRadius="200"

BorderThickness="{TemplateBinding BorderThickness}"

BorderBrush="{TemplateBinding BorderBrush}">

<Rectangle x:Name="Icon"

Width="20" Height="20"

Fill="Black">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform"

Visual="{StaticResource ArrowBackIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource MouseOverColor}"/>

<Setter TargetName="Icon"

Property="Fill" Value="Black"/>

</Trigger>

<Trigger Property="IsPressed" Value="True">

<Setter TargetName="BackgroundElement"

Property="Background" Value="{DynamicResource BackgroundHeaderColor}"/>

<Setter TargetName="Icon"

Property="Fill" Value="White"/>

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="ResultListViewItem" TargetType="ListViewItem">

<Setter Property="Background" Value="{DynamicResource MouseOverColor}"/>

<Setter Property="Foreground" Value="Black"/>

<Setter Property="FontSize" Value="18"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ListViewItem">

<Border Background="{TemplateBinding Background}"

Padding="15 10"

CornerRadius="5">

<GridViewRowPresenter VerticalAlignment="Center"

Columns="{TemplateBinding GridView.ColumnCollection}"

Content="{TemplateBinding Content}"/>

</Border>

<ControlTemplate.Triggers>

<Trigger Property="ItemsControl.AlternationIndex" Value="1">

<Setter Property="Background" Value="{DynamicResource BackgroundMenuColor}" />

</Trigger>

</ControlTemplate.Triggers>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<Style x:Key="ResultListView" TargetType="ListView">

<Setter Property="ItemContainerStyle" Value="{StaticResource ResultListViewItem}"/>

<Style.Triggers>

<Trigger Property="HasItems" Value="False">

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="ListView">

<TextBlock Text="Нет результатов"

Style="{StaticResource PlainText}"

HorizontalAlignment="Center"

VerticalAlignment="Center"/>

</ControlTemplate>

</Setter.Value>

</Setter>

</Trigger>

</Style.Triggers>

</Style>

<Style x:Key="HeaderListView" TargetType="GridViewColumnHeader">

<Setter Property="FontSize" Value="18"/>

<Setter Property="Background" Value="{DynamicResource BackgroundHeaderColor}"/>

<Setter Property="Foreground" Value="White"/>

<Setter Property="Template">

<Setter.Value>

<ControlTemplate TargetType="GridViewColumnHeader">

<Grid>

<Border Background="{TemplateBinding Background}"

BorderThickness="0 0 0 1"

BorderBrush="White"

Padding="15 10"

CornerRadius="5"

Margin="0 0 1 0">

<ContentPresenter HorizontalAlignment="Center" VerticalAlignment="Center"/>

</Border>

</Grid>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>

<!--End: Styles of the test section-->

</ResourceDictionary>

Chapters.cs

namespace ConnectingDBToApp.Enums

{

public enum Chapters

{

SSMS,

SQLite,

Test

}

}

CustomStyleSelector.cs

using System.Windows;

using System.Windows.Controls;

using ConnectingDBToApp.Models;

namespace ConnectingDBToApp.GlobalClasses

{

public class CustomStyleSelector : StyleSelector

{

public Style PlainTextItem { get; set; } = null!;

public Style H1Item { get; set; } = null!;

public Style H2Item { get; set; } = null!;

public Style ImageItem { get; set; } = null!;

public Style DoubleImageItem { get; set; } = null!;

public Style CopyButtonItem { get; set; } = null!;

public Style CodeImageItem { get; set; } = null!;

public Style HyperLinkItem { get; set; } = null!;

public override Style SelectStyle(object item, DependencyObject container)

{

var elementItem = (ElementItem)item;

switch (elementItem.ObjectType)

{

case "H1":

return H1Item;

case "H2":

return H2Item;

case "Image":

return ImageItem;

case "DoubleImage":

return DoubleImageItem;

case "CopyButton":

return CopyButtonItem;

case "CodeImage":

return CodeImageItem;

case "HyperLink":

return HyperLinkItem;

default:

return PlainTextItem;

}

}

}

}

DbContext.cs

using ConnectingDBToApp.MyDbContext;

namespace ConnectingDBToApp.GlobalClasses

{

public static class DbContext

{

public static DataContext Tables { get; set; } = new DataContext();

}

}

GlobalObjs.cs

using System.Windows.Controls;

using System.Windows.Controls.Primitives;

using ConnectingDBToApp.Models;

namespace ConnectingDBToApp.GlobalClasses

{

public static class GlobalObjs

{

public static Frame MainFrame { get; set; } = null!;

public static ListBox SideBar { get; set; } = null!;

public static ScrollViewer MainScrollViewer { get; set; } = null!;

public static Button SideBarButton { get; set; } = null!;

public static ToggleButton MenuButton { get; set; } = null!;

public static TabControl SMMSTabControl { get; set; } = null!;

public static TabControl SMMSSubTabControl { get; set; } = null!;

public static TabControl SQLiteTabControl { get; set; } = null!;

}

}

MainWindowMethods.cs

using System;

namespace ConnectingDBToApp.GlobalClasses

{

public static class MainWindowMethods

{

public static Action? Minimize { get; set; }

public static Action? Close { get; set; }

public static Action? FullScreen { get; set; }

}

}

GetTestResultMessage.cs

using ConnectingDBToApp.Models;

using CommunityToolkit.Mvvm.Messaging.Messages;

namespace ConnectingDBToApp.Messages

{

public class GetTestResultMessage : ValueChangedMessage<TestResult>

{

public GetTestResultMessage(TestResult value) : base(value)

{

}

}

}

GetUsernameMessage.cs

using CommunityToolkit.Mvvm.Messaging.Messages;

namespace ConnectingDBToApp.Messages

{

public class GetUsernameMessage : ValueChangedMessage<string>

{

public GetUsernameMessage(string value) : base(value)

{

}

}

}

ElementItem.cs

namespace ConnectingDBToApp.Models

{

public interface ElementItem

{

public int Id { get; set; }

public string ObjectType { get; set; }

public string Chapter { get; set; }

public string Text { get; set; }

public string? AdditionalText { get; set; }

public string? Code { get; set; }

}

}

SQLiteElement.cs

namespace ConnectingDBToApp.Models;

public partial class SQLiteElement : ElementItem

{

public int Id { get; set; }

public string ObjectType { get; set; } = null!;

public string Chapter { get; set; } = null!;

public string Text { get; set; } = null!;

public string? AdditionalText { get; set; }

public string? Code { get; set; }

}

SSMSElement.cs

namespace ConnectingDBToApp.Models;

public partial class SSMSElement : ElementItem

{

public int Id { get; set; }

public string ObjectType { get; set; } = null!;

public string Chapter { get; set; } = null!;

public string Text { get; set; } = null!;

public string? AdditionalText { get; set; }

public string? Code { get; set; }

}

TestQuestion.cs

namespace ConnectingDBToApp.Models;

public partial class TestQuestion

{

public int Id { get; set; }

public string Question { get; set; } = null!;

public string Variant1 { get; set; } = null!;

public string Variant2 { get; set; } = null!;

public string Variant3 { get; set; } = null!;

public string Variant4 { get; set; } = null!;

public string RightAnswer { get; set; } = null!;

}

TestResult.cs

namespace ConnectingDBToApp.Models;

public partial class TestResult

{

public int Id { get; set; }

public string Username { get; set; } = null!;

public int CountRightAnswer { get; set; }

public int CountQuestions { get; set; }

public double Percentages { get; set; }

}

DataContext.cs

using System.Configuration;

using Microsoft.EntityFrameworkCore;

using ConnectingDBToApp.Models;

namespace ConnectingDBToApp.MyDbContext;

public partial class DataContext : DbContext

{

public DataContext() { }

public DataContext(DbContextOptions<DataContext> options) : base(options) { }

public virtual DbSet<SQLiteElement> SQLiteElements { get; set; }

public virtual DbSet<SSMSElement> SSMSElements { get; set; }

public virtual DbSet<TestQuestion> TestQuestions { get; set; }

public virtual DbSet<TestResult> TestResults { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

=> optionsBuilder.UseSqlite(ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString);

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<SQLiteElement>(entity =>

{

entity.ToTable("SQLiteElements");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.ObjectType).HasDefaultValueSql("'PlainText'");

});

modelBuilder.Entity<SSMSElement>(entity =>

{

entity.ToTable("SSMSElements");

entity.Property(e => e.Id).HasColumnName("ID");

entity.Property(e => e.ObjectType).HasDefaultValueSql("'PlainText'");

});

modelBuilder.Entity<TestQuestion>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

});

modelBuilder.Entity<TestResult>(entity =>

{

entity.Property(e => e.Id).HasColumnName("ID");

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

AllResultsViewModel.cs

using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Models;

using ConnectingDBToApp.GlobalClasses;

namespace ConnectingDBToApp.ViewModels

{

public partial class AllResultsViewModel : ObservableObject

{

public AllResultsViewModel()

{

Results = new ObservableCollection<TestResult>(DbContext.Tables.TestResults);

}

[ObservableProperty]

[NotifyCanExecuteChangedFor(nameof(ClearResultsCommand))]

private ObservableCollection<TestResult> \_results;

[RelayCommand]

private void BackPage()

{

GlobalObjs.MainFrame.GoBack();

}

[RelayCommand(CanExecute = nameof(CanClearResults))]

private void ClearResults()

{

DbContext.Tables.TestResults.RemoveRange(Results);

DbContext.Tables.SaveChanges();

Results.Clear();

}

private bool CanClearResults()

{

return Results.Count > 0;

}

}

}

MainViewModel.cs

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Animation;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Enums;

using ConnectingDBToApp.Views.Pages;

using ConnectingDBToApp.Views.Windows;

using ConnectingDBToApp.GlobalClasses;

namespace ConnectingDBToApp.ViewModels

{

public partial class MainViewModel : ObservableObject

{

[RelayCommand]

private void SelectedChanged(ListBox listBox)

{

GlobalObjs.MainScrollViewer.ScrollToHome();

switch (listBox.SelectedIndex)

{

case (int)Chapters.SSMS:

GlobalObjs.MainFrame.Navigate(new SSMSPage()); break;

case (int)Chapters.SQLite:

GlobalObjs.MainFrame.Navigate(new SQLitePage()); break;

case (int)Chapters.Test:

GlobalObjs.MainFrame.Navigate(new TestPage()); break;

}

}

[RelayCommand]

private void MinimizeWindow() => MainWindowMethods.Minimize!();

[RelayCommand]

private void FullScreenWindow() => MainWindowMethods.FullScreen!();

[RelayCommand]

private void CloseWindow() => MainWindowMethods.Close!();

[RelayCommand]

private void ShowSideBar(ListBox sideBar)

{

var animation = new DoubleAnimation();

animation.EasingFunction = new QuadraticEase() { EasingMode = EasingMode.EaseOut };

animation.Duration = TimeSpan.FromMilliseconds(200);

switch (sideBar.Width)

{

case 230:

animation.To = 0; break;

case 0:

animation.To = 230; break;

}

sideBar.BeginAnimation(FrameworkElement.WidthProperty, animation);

}

[RelayCommand]

private void OpenAboutBox() => new AboutBoxWindow().ShowDialog();

}

}

QuestionViewModel.cs

using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Animation;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.Messaging;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Models;

using ConnectingDBToApp.Messages;

using ConnectingDBToApp.GlobalClasses;

using ConnectingDBToApp.Views.Pages;

namespace ConnectingDBToApp.ViewModels

{

public partial class QuestionViewModel : ObservableObject, IRecipient<GetUsernameMessage>

{

private int \_countQuestions;

private int \_countRightAnswers;

private string \_username;

public Queue<TestQuestion> TestQuestions { get; set; }

public QuestionViewModel()

{

\_countRightAnswers = 0;

\_username = string.Empty;

WeakReferenceMessenger.Default.Register(this);

TestQuestions = new Queue<TestQuestion>(DbContext.Tables.TestQuestions);

\_countQuestions = TestQuestions.Count;

CurrentQuestion = TestQuestions.Dequeue();

}

public void Receive(GetUsernameMessage message)

{

\_username = message.Value;

WeakReferenceMessenger.Default.UnregisterAll(this);

}

[ObservableProperty]

private TestQuestion \_currentQuestion;

[ObservableProperty]

[NotifyCanExecuteChangedFor(nameof(NextQuestionCommand))]

private RadioButton \_selectedRadioButton = null!;

[ObservableProperty]

private string \_buttonContent = "Далее";

[RelayCommand]

private void CheckedRadioButton(RadioButton radioButton)

{

SelectedRadioButton = radioButton;

}

[RelayCommand(CanExecute = nameof(CanNextQuestion))]

private void NextQuestion()

{

string selectedAnswer = SelectedRadioButton.Content.ToString()!;

if (selectedAnswer == CurrentQuestion.RightAnswer)

{

\_countRightAnswers++;

}

if (TestQuestions.Count >= 1)

{

if (TestQuestions.Count == 1)

ButtonContent = "Завершить";

CurrentQuestion = TestQuestions.Dequeue();

SelectedRadioButton.IsChecked = false;

SelectedRadioButton = null!;

}

else

{

var result = new TestResult()

{

Username = \_username,

CountRightAnswer = \_countRightAnswers,

CountQuestions = \_countQuestions,

Percentages = \_countRightAnswers \* 1.0 / \_countQuestions

};

DbContext.Tables.Add(result);

DbContext.Tables.SaveChanges();

GlobalObjs.MainFrame.Navigate(new ResultPage());

WeakReferenceMessenger.Default.Send(new GetTestResultMessage(result));

DoubleAnimation animation = new DoubleAnimation()

{

To = 230,

Duration = TimeSpan.FromMilliseconds(200)

};

animation.EasingFunction = new QuadraticEase() { EasingMode = EasingMode.EaseOut };

GlobalObjs.SideBar.BeginAnimation(FrameworkElement.WidthProperty, animation);

GlobalObjs.MenuButton.IsEnabled = true;

}

}

private bool CanNextQuestion()

{

return SelectedRadioButton != null;

}

}

}

ResultViewModel.cs

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.Messaging;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Models;

using ConnectingDBToApp.Messages;

using ConnectingDBToApp.GlobalClasses;

using ConnectingDBToApp.Views.Pages;

namespace ConnectingDBToApp.ViewModels

{

public partial class ResultViewModel : ObservableObject, IRecipient<GetTestResultMessage>

{

public ResultViewModel()

{

WeakReferenceMessenger.Default.Register(this);

}

[ObservableProperty]

private TestResult \_result = null!;

public void Receive(GetTestResultMessage message)

{

Result = message.Value;

WeakReferenceMessenger.Default.UnregisterAll(this);

}

[RelayCommand]

private void NavigateTestPage()

{

GlobalObjs.MainFrame.Navigate(new TestPage());

}

}

}

SQLiteViewModel.cs

using System;

using System.Linq;

using System.Windows;

using System.Diagnostics;

using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Models;

using ConnectingDBToApp.GlobalClasses;

using ConnectingDBToApp.Views.Windows;

namespace ConnectingDBToApp.ViewModels

{

public partial class SQLiteViewModel : ObservableObject

{

[ObservableProperty]

private ObservableCollection<SQLiteElement> \_creatingDbItems;

[ObservableProperty]

private ObservableCollection<SQLiteElement> \_EFCoreConnectingDbItems;

public SQLiteViewModel()

{

CreatingDbItems = new ObservableCollection<SQLiteElement>(DbContext.Tables.SQLiteElements.Where(item => item.Chapter == "CreatingDB"));

EFCoreConnectingDbItems = new ObservableCollection<SQLiteElement>(DbContext.Tables.SQLiteElements.Where(item => item.Chapter == "EFCoreConnectingDB"));

}

[RelayCommand]

private void CopyText(string code)

{

Clipboard.SetText(code);

new MessageWindow("Скопировано!").ShowDialog();

}

[RelayCommand]

private void NavigateLink(string link)

{

Process.Start(new ProcessStartInfo(link) { UseShellExecute = true });

}

[RelayCommand]

private void NavigatePart(object i)

{

var index = Convert.ToInt32(i);

GlobalObjs.SQLiteTabControl.SelectedIndex = index;

GlobalObjs.MainScrollViewer.ScrollToHome();

}

}

}

SSMSViewModel.cs

using System;

using System.Linq;

using System.Windows;

using System.Collections.ObjectModel;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Models;

using ConnectingDBToApp.GlobalClasses;

using ConnectingDBToApp.Views.Windows;

namespace ConnectingDBToApp.ViewModels

{

public partial class SSMSViewModel : ObservableObject

{

[ObservableProperty]

private ObservableCollection<SSMSElement> \_creatingDbItems;

[ObservableProperty]

private ObservableCollection<SSMSElement> \_connectionDBItems;

[ObservableProperty]

private ObservableCollection<SSMSElement> \_SQLClientDownloadItems;

[ObservableProperty]

private ObservableCollection<SSMSElement> \_SQLClientConnectionDBItems;

[ObservableProperty]

private ObservableCollection<SSMSElement> \_SQLClientQueryExecutionItems;

[ObservableProperty]

private ObservableCollection<SSMSElement> \_EFCore7Items;

public SSMSViewModel()

{

CreatingDbItems = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "CreatingDB"));

ConnectionDBItems = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "ConnectionDB"));

SQLClientDownloadItems = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "SQLClientDownload"));

SQLClientConnectionDBItems = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "SQLClientConnectionDB"));

SQLClientQueryExecutionItems = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "SQLClientQueryExecution"));

EFCore7Items = new ObservableCollection<SSMSElement>(DbContext.Tables.SSMSElements.Where(item => item.Chapter == "EFCore7"));

}

[RelayCommand]

private void CopyText(string code)

{

Clipboard.SetText(code);

new MessageWindow("Cкопировано!").ShowDialog();

}

[RelayCommand]

private void NavigatePart(object i)

{

var index = Convert.ToInt32(i);

GlobalObjs.SMMSTabControl.SelectedIndex = index;

GlobalObjs.MainScrollViewer.ScrollToHome();

}

[RelayCommand]

private void NavigateSubPart(object i)

{

var index = Convert.ToInt32(i);

GlobalObjs.SMMSSubTabControl.SelectedIndex = index;

GlobalObjs.MainScrollViewer.ScrollToHome();

}

}

}

TestViewModel.cs

using System;

using System.Windows;

using System.Windows.Media.Animation;

using CommunityToolkit.Mvvm.Input;

using CommunityToolkit.Mvvm.Messaging;

using CommunityToolkit.Mvvm.ComponentModel;

using ConnectingDBToApp.Messages;

using ConnectingDBToApp.GlobalClasses;

using ConnectingDBToApp.Views.Pages;

using ConnectingDBToApp.Views.Windows;

namespace ConnectingDBToApp.ViewModels

{

public partial class TestViewModel : ObservableObject

{

[ObservableProperty]

private string \_username = string.Empty;

[RelayCommand]

private void OpenTest()

{

if (string.IsNullOrEmpty(Username))

{

new MessageWindow("Пустое поле ввода.\nВведите имя!").ShowDialog();

return;

}

else if (Username.Length > 16)

{

new MessageWindow("Слишком длинное имя!").ShowDialog();

return;

}

var animation = new DoubleAnimation()

{

To = 0,

Duration = TimeSpan.FromMilliseconds(200)

};

animation.EasingFunction = new QuadraticEase() { EasingMode = EasingMode.EaseOut };

GlobalObjs.MenuButton.IsEnabled = false;

GlobalObjs.SideBar.BeginAnimation(FrameworkElement.WidthProperty, animation);

GlobalObjs.MainFrame.Navigate(new QuestionPage());

WeakReferenceMessenger.Default.Send(new GetUsernameMessage(Username));

}

[RelayCommand]

private void OpenAllResults()

{

GlobalObjs.MainFrame.Navigate(new AllResultsPage());

}

}

}

AllResultsPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.AllResultsPage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Title="AllResultsPage" Background="White">

<Page.DataContext>

<vm:AllResultsViewModel/>

</Page.DataContext>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Button Grid.Row="0"

Style="{StaticResource ArrowBackButton}"

Command="{Binding BackPageCommand}"

HorizontalAlignment="Left"

VerticalAlignment="Center"

FontSize="20"

Margin="0 0 0 30"/>

<TextBlock Style="{StaticResource H2}"

Text="Результаты пользователей"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Margin="0 0 0 30"/>

<ListView Grid.Row="1"

Style="{StaticResource ResultListView}"

ItemsSource="{Binding Results}"

AlternationCount="2"

SelectionMode="Single"

BorderThickness="0">

<ListView.View>

<GridView ColumnHeaderContainerStyle="{StaticResource HeaderListView}">

<GridViewColumn Header="Имя пользователя" DisplayMemberBinding="{Binding Username}"/>

<GridViewColumn Header="Количество правильных ответов" DisplayMemberBinding="{Binding CountRightAnswer}"/>

<GridViewColumn Header="Результат в процентах" DisplayMemberBinding="{Binding StringFormat={}{0:P1}, Path=Percentages}"/>

</GridView>

</ListView.View>

</ListView>

<StackPanel Orientation="Vertical" Grid.Row="2">

<Button Style="{StaticResource SecondaryButton}"

Content="Удалить все результаты"

Command="{Binding ClearResultsCommand}"

HorizontalAlignment="Center"

Margin="0 30 0 0"/>

</StackPanel>

</Grid>

</Page>

QuestionPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.QuestionPage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Background="White"

Title="QuestionPage">

<Page.DataContext>

<vm:QuestionViewModel/>

</Page.DataContext>

<StackPanel Orientation="Vertical"

DataContext="{Binding CurrentQuestion}"

MaxWidth="850"

VerticalAlignment="Center">

<TextBlock Text="{Binding StringFormat=Вопрос {0}, Path=Id}"

Style="{StaticResource H1}"/>

<TextBlock Text="{Binding Question}"

Style="{StaticResource H2}"

TextWrapping="Wrap"

Margin="0 0 0 30"/>

<UniformGrid Columns="1" Margin="30 0 0 0">

<RadioButton Content="{Binding Variant1}"

Style="{StaticResource VariantRadioButton}"

Command="{Binding DataContext.CheckedRadioButtonCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=Self}}"/>

<RadioButton Content="{Binding Variant2}"

Style="{StaticResource VariantRadioButton}"

Command="{Binding DataContext.CheckedRadioButtonCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=Self}}"/>

<RadioButton Content="{Binding Variant3}"

Style="{StaticResource VariantRadioButton}"

Command="{Binding DataContext.CheckedRadioButtonCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=Self}}"/>

<RadioButton Content="{Binding Variant4}"

Style="{StaticResource VariantRadioButton}"

Command="{Binding DataContext.CheckedRadioButtonCommand, RelativeSource={RelativeSource AncestorType=Page}}"

CommandParameter="{Binding RelativeSource={RelativeSource Mode=Self}}"/>

</UniformGrid>

<Button Content="{Binding DataContext.ButtonContent, RelativeSource={RelativeSource AncestorType=Page}}"

Style="{StaticResource SecondaryButton}"

HorizontalAlignment="Center"

Margin="0 20 0 0"

Command="{Binding DataContext.NextQuestionCommand, RelativeSource={RelativeSource AncestorType=Page}}"/>

</StackPanel>

</Page>

ResultPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.ResultPage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Title="ResultPage">

<Page.DataContext>

<vm:ResultViewModel/>

</Page.DataContext>

<UniformGrid Columns="1"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Style="{StaticResource H2}">

<TextBlock.Text>

<MultiBinding StringFormat="{}{0} набрал(а) {1} из {2} баллов">

<Binding Path="Result.Username"/>

<Binding Path="Result.CountRightAnswer"/>

<Binding Path="Result.CountQuestions"/>

</MultiBinding>

</TextBlock.Text>

</TextBlock>

<TextBlock Style="{StaticResource PlainText}"

HorizontalAlignment="Center">

<Run Text="Процентное соотношение: "/>

<Run Text="{Binding StringFormat={}{0:P1}, Path=Result.Percentages}"

FontWeight="Bold"/>

</TextBlock>

<Button Content="Перейти на начальную страницу"

Command="{Binding NavigateTestPageCommand}"

Style="{StaticResource PrimaryButton}"/>

</UniformGrid>

</Page>

SQLitePage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.SQLitePage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Title="SQLitePage">

<Page.DataContext>

<vm:SQLiteViewModel/>

</Page.DataContext>

<StackPanel>

<TextBlock Text="SQLite"

Style="{StaticResource H1}"/>

<TabControl x:Name="SQLiteTabControl">

<TabItem Header="Создание БД">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding CreatingDbItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="1"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Подключение к БД →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

<TabItem Header="Подключение БД">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding EFCoreConnectingDbItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 50 0">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="0"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Создание БД"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

</TabControl>

</StackPanel>

</Page>

SSMSPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.SSMSPage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Title="SSMSPage">

<Page.DataContext>

<vm:SSMSViewModel/>

</Page.DataContext>

<StackPanel Orientation="Vertical">

<TextBlock Text="SQL Server Management Studio"

Style="{StaticResource H1}"/>

<TabControl x:Name="SMMSTabControl">

<TabItem Header="Создание БД"

IsSelected="True">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding CreatingDbItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="1"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Cоединение БД в VS →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

<TabItem Header="Cоединение БД в VS">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding ConnectionDBItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 0 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="0"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Создание БД"/>

</Hyperlink>

</TextBlock>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="2"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Работа с SqlClient →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

<TabItem Header="Работа с SqlClient">

<TabControl x:Name="SMMSSubTabControl">

<TabItem Header="Установка SqlClient">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding SQLClientDownloadItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 0 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="1"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Соединение БД в VS"/>

</Hyperlink>

</TextBlock>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigateSubPartCommand}"

CommandParameter="1"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Подключение БД →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

<TabItem Header="Подключение БД">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding SQLClientConnectionDBItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 0 10">

<Hyperlink Command="{Binding NavigateSubPartCommand}"

CommandParameter="0"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Установка SqlClient"/>

</Hyperlink>

</TextBlock>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigateSubPartCommand}"

CommandParameter="2"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Выполнение запросов →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

<TabItem Header="Выполнение запросов">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding SQLClientQueryExecutionItems}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 0 10">

<Hyperlink Command="{Binding NavigateSubPartCommand}"

CommandParameter="1"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Подключение БД"/>

</Hyperlink>

</TextBlock>

<TextBlock Grid.Row="1"

HorizontalAlignment="Right"

Margin="0 50 50 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="3"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="Работа с EF Core 7 →"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

</TabControl>

</TabItem>

<TabItem Header="Работа EF Core 7">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<ListBox Grid.Row="0"

ItemsSource="{Binding EFCore7Items}"

Style="{StaticResource DataListBox}"/>

<TextBlock Grid.Row="1"

HorizontalAlignment="Left"

Margin="20 50 0 10">

<Hyperlink Command="{Binding NavigatePartCommand}"

CommandParameter="2"

Style="{StaticResource NextPartHyperlink}">

<TextBlock Text="← Работа с SqlClient"/>

</Hyperlink>

</TextBlock>

</Grid>

</TabItem>

</TabControl>

</StackPanel>

</Page>

StartPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.StartPage"

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"

mc:Ignorable="d"

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

Title="StartPage">

<StackPanel Orientation="Vertical"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<Rectangle Fill="{StaticResource MouseOverColor}"

Width="50" Height="50">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource ArrowBackIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

<TextBlock Text="Выберите один из разделов"

Style="{StaticResource H2}"

Foreground="{StaticResource MouseOverColor}"/>

</StackPanel>

</Page>

TestPage.xaml

<Page x:Class="ConnectingDBToApp.Views.Pages.TestPage"

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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

mc:Ignorable="d"

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

Background="White"

Title="TestPage">

<Page.DataContext>

<vm:TestViewModel/>

</Page.DataContext>

<Grid>

<StackPanel Orientation="Vertical"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Text="Проверка знаний"

Style="{StaticResource H1}"/>

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Margin="0 0 0 20">

<TextBlock Text="Ваше имя:"

FontSize="18"

Margin="0 0 10 0"

VerticalAlignment="Center"/>

<TextBox Text="{Binding Username}"

Style="{StaticResource UsernameTextBox}"/>

</StackPanel>

<Button Content="ПРОЙТИ"

Style="{StaticResource PrimaryButton}"

Command="{Binding OpenTestCommand}"/>

</StackPanel>

<Button Content="Результаты пользователей"

Style="{StaticResource SecondaryButton}"

Command="{Binding OpenAllResultsCommand}"

HorizontalAlignment="Center"

VerticalAlignment="Bottom"

Margin="0 0 0 30"/>

</Grid>

</Page>

AboutBoxWindow.xaml

<Window x:Class="ConnectingDBToApp.Views.Windows.AboutBoxWindow"

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"

mc:Ignorable="d"

Title="AboutBoxWindow"

Height="300" Width="500"

WindowStyle="None"

WindowStartupLocation="CenterScreen"

ResizeMode="NoResize"

MouseDown="Window\_MouseDown"

BorderThickness="1"

BorderBrush="{StaticResource BackgroundHeaderColor}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="40"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Background="{StaticResource BackgroundHeaderColor}"

Orientation="Horizontal">

<Rectangle Fill="White"

Width="25" Height="25"

Margin="12.5 0 0 0">

<Rectangle.OpacityMask>

<VisualBrush Stretch="Uniform" Visual="{StaticResource InfoIcon}"/>

</Rectangle.OpacityMask>

</Rectangle>

<TextBlock Text="О программе"

FontSize="16"

FontWeight="Bold"

VerticalAlignment="Center"

Margin="12.5 0 0 0"

Foreground="White"/>

</StackPanel>

<Button Grid.Row="0"

Style="{StaticResource CloseButton}"

HorizontalAlignment="Right"

Click="CloseWindow"/>

<StackPanel Grid.Row="1" Margin="50 40 50 0">

<TextBlock Style="{StaticResource PlainText}"

FontSize="16"

Margin="0 0 0 20">

<Bold>Название программы:</Bold> “Технология подключения базы данных к настольному приложению”

</TextBlock>

<TextBlock Style="{StaticResource PlainText}"

FontSize="16"

Margin="0 0 0 20">

<Bold>Разработчик:</Bold> Казанцев Роман Андреевич, студент группы 403 ИСП

</TextBlock>

<TextBlock Style="{StaticResource PlainText}"

FontSize="16">

<Bold>Версия:</Bold> 1.0

</TextBlock>

</StackPanel>

</Grid>

</Window>

MainWindow.xaml

<Window x:Class="ConnectingDBToApp.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:vm="clr-namespace:ConnectingDBToApp.ViewModels"

xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

mc:Ignorable="d"

Title="MainWindow"

WindowStartupLocation="CenterScreen"

WindowStyle="None"

MinWidth="800" MinHeight="500"

x:Name="MainWindowApp"

MaxHeight="{x:Static SystemParameters.MaximizedPrimaryScreenHeight}">

<Window.DataContext>

<vm:MainViewModel/>

</Window.DataContext>

<DockPanel LastChildFill="True">

<DockPanel LastChildFill="True"

DockPanel.Dock="Top"

Height="40"

Background="{DynamicResource BackgroundHeaderColor}"

MouseDown="HeaderWindow\_MouseDown">

<ToggleButton DockPanel.Dock="Left"

x:Name="MenuButton"

Style="{StaticResource MenuButton}"

Command="{Binding ShowSideBarCommand}"

CommandParameter="{Binding ElementName=SideBar}"/>

<Button DockPanel.Dock="Right"

Style="{StaticResource CloseButton}"

Command="{Binding CloseWindowCommand}"

ToolTip="Закрыть"/>

<Button DockPanel.Dock="Right"

Style="{StaticResource FullscreenButton}"

Command="{Binding FullScreenWindowCommand}"/>

<Button DockPanel.Dock="Right"

Style="{StaticResource MinimizeButton}"

Command="{Binding MinimizeWindowCommand}"

ToolTip="Свернуть"/>

<StackPanel Orientation="Horizontal"

VerticalAlignment="Center"

Margin="10 0 0 0">

<TextBlock Style="{StaticResource TitleWindow}"

Margin="0 0 10 0"/>

</StackPanel>

</DockPanel>

<ListBox x:Name="SideBar"

DockPanel.Dock="Left"

Style="{StaticResource SideBar}"

ItemContainerStyle="{StaticResource SideBarItem}">

<ListBoxItem DockPanel.Dock="Top">

<ListBoxItem.Background>

<ImageBrush ImageSource="\Images\Icons\SSMS.png"/>

</ListBoxItem.Background>

<ListBoxItem.Content>

<TextBlock Text="SSMS"/>

</ListBoxItem.Content>

</ListBoxItem>

<ListBoxItem DockPanel.Dock="Top">

<ListBoxItem.Background>

<ImageBrush ImageSource="\Images\Icons\SQLite.png"/>

</ListBoxItem.Background>

<ListBoxItem.Content>

<TextBlock Text="SQLite"/>

</ListBoxItem.Content>

</ListBoxItem>

<ListBoxItem DockPanel.Dock="Top">

<ListBoxItem.Background>

<ImageBrush ImageSource="\Images\Icons\Test.png"/>

</ListBoxItem.Background>

<ListBoxItem.Content>

<TextBlock Text="Тест"/>

</ListBoxItem.Content>

</ListBoxItem>

<ListBoxItem DockPanel.Dock="Bottom"

Style="{StaticResource AboutBoxButton}"

Margin="0 0 0 20"/>

<i:Interaction.Triggers>

<i:EventTrigger EventName="SelectionChanged">

<i:InvokeCommandAction Command="{Binding SelectedChangedCommand}"

CommandParameter="{Binding RelativeSource={RelativeSource AncestorType=ListBox}}"/>

</i:EventTrigger>

</i:Interaction.Triggers>

</ListBox>

<ScrollViewer x:Name="MainScrollViewer"

HorizontalScrollBarVisibility="Disabled">

<Frame x:Name="MainFrameApp"

Source="../Pages/StartPage.xaml"

NavigationUIVisibility="Hidden"

Padding="30 20 0 30"/>

</ScrollViewer>

</DockPanel>

</Window>

MessageWindow.xaml

<Window x:Class="ConnectingDBToApp.Views.Windows.MessageWindow"

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"

mc:Ignorable="d"

Title="MessageWindow"

Height="150" Width="250"

WindowStyle="None"

WindowStartupLocation="CenterScreen"

ResizeMode="NoResize"

MouseDown="Window\_MouseDown"

BorderThickness="1"

BorderBrush="{StaticResource BackgroundHeaderColor}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="30"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0"

Background="{StaticResource BackgroundHeaderColor}"

Orientation="Horizontal">

<TextBlock Text="Сообщение"

FontSize="16"

FontWeight="Bold"

VerticalAlignment="Center"

Margin="20 0 0 0"

Foreground="White"/>

</StackPanel>

<Button Grid.Row="0"

Style="{StaticResource CloseButton}"

Width="30" Height="30"

HorizontalAlignment="Right"

Click="CloseWindow"/>

<StackPanel Grid.Row="1"

HorizontalAlignment="Center"

VerticalAlignment="Center">

<TextBlock Text="{Binding}"

Style="{StaticResource PlainText}"

FontSize="16"

Margin="0 15"/>

<Button Content="OK"

Style="{StaticResource SecondaryButton}"

Click="CloseWindow"

FontSize="14"

Padding="10 5"/>

</StackPanel>

</Grid>

</Window>

App.config

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

<configuration>

<connectionStrings>

<add name="DefaultConnection"

connectionString="Data Source=Database/data.db"

providerName="Microsoft.EntityFrameworkCore.Sqlite"/>

</connectionStrings>

</configuration>

App.xaml

<Application x:Class="ConnectingDBToApp.App"

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

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

xmlns:local="clr-namespace:ConnectingDBToApp"

StartupUri="Views/Windows/MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary Source="\Dictionaries\Fonts.xaml"/>

<ResourceDictionary Source="\Dictionaries\Colors.xaml"/>

<ResourceDictionary Source="\Dictionaries\Icons.xaml"/>

<ResourceDictionary Source="\Dictionaries\Styles.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>