ClaimApp (WPF) — Ready-to-run project files

This canvas contains a complete Visual Studio WPF project you can copy into a folder, open in Visual Studio, build and run. It implements the Lecturer / Coordinator claim system and uses the dark purple-blue gradient visual style you provided in the screenshot.

How to use

1. Create a new folder `ClaimApp` on your machine.
2. Inside that folder, create the file tree and copy each file content below into the matching path & filename.
3. Open `ClaimApp.sln` in Visual Studio *or* open the folder as a project in VS Code with the C# extension. (If you don't have the `.sln`, open `ClaimApp.csproj` directly by opening the folder.)
4. In Visual Studio, restore NuGet packages (Tools > NuGet > Restore) then Build and Run.

NuGet packages required (these are included as PackageReference in the .csproj):

- Microsoft.EntityFrameworkCore
- Microsoft.EntityFrameworkCore.Sqlite

If you get any missing package warnings, run `dotnet restore` from the project folder.

Project file tree

```
ClaimApp/
├─ ClaimApp.sln
├─ ClaimApp/
│  ├─ ClaimApp.csproj
│  ├─ App.xaml
│  ├─ App.xaml.cs
│  ├─ MainWindow.xaml
│  ├─ MainWindow.xaml.cs
│  ├─ Resources/Theme.xaml
│  ├─ Views/
│  │  ├─ LecturerView.xaml
│  │  └─ CoordinatorView.xaml
│  ├─ ViewModels/
│  │  ├─ MainViewModel.cs
│  │  ├─ LecturerViewModel.cs
│  │  └─ CoordinatorViewModel.cs
│  ├─ Models/
│  │  ├─ Claim.cs
```

```
│   │   └── ClaimStatus.cs
│   ├── Services/
│   │   ├── IClaimService.cs
│   │   └── ClaimService.cs
│   ├── Data/
│   │   └── ApplicationDbContext.cs
│   └── Helpers/
│       └── RelayCommand.cs
├── .gitignore
└── README.txt
```

ClaimApp.sln

> Create a new solution with Visual Studio named `ClaimApp` and add the `ClaimApp` WPF project. If you prefer manual, skip the `.sln` and open the single `ClaimApp.csproj` project file.

ClaimApp/ClaimApp.csproj

```
<Project Sdk="Microsoft.NET.Sdk.WindowsDesktop">
  <PropertyGroup>
    <OutputType>WinExe</OutputType>
    <TargetFramework>net7.0-windows</TargetFramework>
    <UseWPF>true</UseWPF>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Microsoft.EntityFrameworkCore" Version="7.0.0" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite"
Version="7.0.0" />
  </ItemGroup>
</Project>
```

ClaimApp/App.xaml

```
<Application x:Class="ClaimApp.App"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
```

```
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             StartupUri="MainWindow.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary Source="Resources/Theme.xaml" />
            </ResourceDictionary.MergedDictionaries>
        </ResourceDictionary>
    </Application.Resources>
</Application>
```

ClaimApp/App.xaml.cs

```
using System.Windows;

namespace ClaimApp
{
    public partial class App : Application
    {
    }
}
```

ClaimApp/Resources/Theme.xaml

This file contains the gradient, styles, and brushes used for the UI to match your screenshot (rounded window corners aren't a native WPF feature without extra OS-level work; we emulate the look inside the window and set a rounded border in the main window content).

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

  <!-- Colors / Brushes -->
  <LinearGradientBrush x:Key="WindowBackground" StartPoint="0,0" EndPoint="1,1">
    <GradientStop Color="#6B2A9E" Offset="0" />
    <GradientStop Color="#142B5C" Offset="1" />
  </LinearGradientBrush>

  <SolidColorBrush x:Key="CardBackground" Color="#0F1530" Opacity="0.45" />
  <SolidColorBrush x:Key="ControlBackground" Color="#14203B" Opacity="0.35" />
  <SolidColorBrush x:Key="AccentBrush" Color="#3E50B4" />
```

```
  <SolidColorBrush x:Key="PrimaryButton" Color="#1E88E5" />
  <SolidColorBrush x:Key="White50" Color="#FFFFFF" Opacity="0.9" />

  <!-- Text styles -->
  <Style TargetType="TextBlock">
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontFamily" Value="Segoe UI" />
  </Style>

  <Style x:Key="TitleLarge" TargetType="TextBlock">
    <Setter Property="FontSize" Value="36" />
    <Setter Property="FontWeight" Value="Bold" />
  </Style>

  <Style x:Key="LabelStyle" TargetType="TextBlock">
    <Setter Property="FontSize" Value="16" />
    <Setter Property="Foreground" Value="#E6E6F0" />
    <Setter Property="VerticalAlignment" Value="Center" />
  </Style>

  <Style TargetType="Button">
    <Setter Property="Padding" Value="10,6" />
    <Setter Property="FontSize" Value="16" />
    <Setter Property="FontFamily" Value="Segoe UI" />
  </Style>

  <Style x:Key="PrimaryAction" TargetType="Button" BasedOn="{StaticResource
{x:Type Button}}">
    <Setter Property="Background" Value="#3549A5" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="BorderThickness" Value="0" />
    <Setter Property="MinWidth" Value="200" />
    <Setter Property="Height" Value="48" />
    <Setter Property="FontWeight" Value="SemiBold" />
    <Setter Property="Template">
      <Setter.Value>
        <ControlTemplate TargetType="Button">
          <Border Background="{TemplateBinding Background}"
                  CornerRadius="12"
                  Padding="{TemplateBinding Padding}">
            <ContentPresenter HorizontalAlignment="Center"
VerticalAlignment="Center"/>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  </Style>
```

```
  <Style TargetType="TextBox">
    <Setter Property="Height" Value="40" />
    <Setter Property="Margin" Value="0" />
    <Setter Property="Padding" Value="10,6" />
    <Setter Property="Background" Value="#10213B" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="BorderBrush" Value="#2C3A6B" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="VerticalContentAlignment" Value="Center" />
    <Setter Property="Template">
      <Setter.Value>
        <ControlTemplate TargetType="TextBox">
          <Border Background="{TemplateBinding Background}" CornerRadius="8"
BorderBrush="{TemplateBinding BorderBrush}" BorderThickness="{TemplateBinding
BorderThickness}">
            <ScrollViewer Margin="0" x:Name="PART_ContentHost"/>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  </Style>

  <Style TargetType="DataGrid">
    <Setter Property="Background" Value="#071224" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="RowBackground" Value="#0E1A33" />
    <Setter Property="AlternatingRowBackground" Value="#081426" />
    <Setter Property="BorderThickness" Value="0" />
  </Style>

</ResourceDictionary>
```

ClaimApp/MainWindow.xaml

MainWindow hosts a rounded card-like area where the UI is placed to emulate the screenshot's rounded corners and gradient.

```
<Window x:Class="ClaimApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:vm="clr-namespace:ClaimApp.ViewModels"
        Title="Contract Monthly Claim System (CMCS)" Height="720" Width="1200"
WindowStartupLocation="CenterScreen" Background="Black">
```

```
    <Window.DataContext>
        <vm:MainViewModel />
    </Window.DataContext>

    <Grid>
        <!-- Background gradient fills the whole window -->
        <Rectangle Fill="{StaticResource WindowBackground}" />

        <!-- Center card (rounded) emulating the app screen -->
        <Border CornerRadius="28" Margin="34" Background="#0F1930"
Opacity="0.98">
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="Auto" />
                    <RowDefinition Height="*" />
                </Grid.RowDefinitions>

                <StackPanel Margin="28">
                    <TextBlock Text="Contract Monthly Claim System (CMCS)"
FontSize="20" Foreground="#EDE9FF" />
                    <TextBlock Text="Lecturer View" Style="{StaticResource
TitleLarge}" Margin="0,6,0,20" />

                    <Grid ColumnSpacing="16">
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="300" />
                            <ColumnDefinition Width="*" />
                        </Grid.ColumnDefinitions>

                        <StackPanel Grid.Column="0" VerticalAlignment="Top">
                            <TextBlock Text="Hours Worked:"
Style="{StaticResource LabelStyle}" />
                            <TextBox Text="{Binding LecturerVM.HoursWorked,
UpdateSourceTrigger=PropertyChanged}" />

                            <TextBlock Text="Hourly Rate:"
Style="{StaticResource LabelStyle}" Margin="0,10,0,0" />
                            <TextBox Text="{Binding LecturerVM.HourlyRate,
UpdateSourceTrigger=PropertyChanged}" />

                            <TextBlock Text="Notes (optional):"
Style="{StaticResource LabelStyle}" Margin="0,10,0,0" />
                            <TextBox Text="{Binding LecturerVM.Notes,
UpdateSourceTrigger=PropertyChanged}" Height="86" AcceptsReturn="True"
TextWrapping="Wrap" />

                            <StackPanel Orientation="Horizontal"
Margin="0,12,0,0" HorizontalAlignment="Left" Spacing="12">
```

```
                                <Button Content="Upload Supporting Document"
Command="{Binding LecturerVM.UploadCommand}" Style="{StaticResource
PrimaryAction}"/>
                                <Button Content="Submit Claim"
Command="{Binding LecturerVM.SubmitCommand}" Background="#233A7A"
Foreground="White" Style="{StaticResource PrimaryAction}"/>
                            </StackPanel>

                            <TextBlock Text="{Binding
LecturerVM.UploadedFileName}" Margin="0,8,0,0" Foreground="#DDE6FF" />
                        </StackPanel>

                        <StackPanel Grid.Column="1">
                            <TextBlock Text="Your recent claims" FontSize="18"
FontWeight="SemiBold" Margin="0,0,0,8" />
                            <DataGrid ItemsSource="{Binding LecturerVM.Claims}"
AutoGenerateColumns="False" CanUserAddRows="False" Height="360">
                                <DataGrid.Columns>
                                    <DataGridTextColumn Header="Claim Date"
Binding="{Binding SubmittedAt, StringFormat=yyyy-MM-dd}" Width="140" />
                                    <DataGridTextColumn Header="Hours Worked"
Bind
```