WPF Note:

Các hàm liên quan:

.ShowDialog(): Mở cửa sổ mới từ cửa sổ gốc thì ko cho click vào cửa sổ gốc đến khi nào hoàn thành nhiệm vụ trên cửa sổ con

.SaveChanges: Lưu thay đổi => Nhớ dùng sau khi display, update ... trong DB

Chơi với DropDown: Combo Box trong WPF

.ItemsSource : Lấy nguồn item từ đâu

.DisplayMemberPath: Hiển thị cột gì

.SelectedValuePath: Lấy dữ lieu từ cột gì

Disbled 1 ô nhập: .isEnable = false;

Kĩ thuật truyền dữ lieu giữa hai màn hình

public Aircodi editedaircon{get;set;} = null;

DataGrid có 1 cái property ngầm là .SelectedItem nhưng sẽ trả về Object. Cần ép kiểu

MessageBox.Show() : Tìm hiểu ở link https://learn.microsoft.com/en-us/dotnet/api/system.windows.messagebox.show?view=windowsdesktop-8.0

--- Trọn bộ Fill Combo Box:

cboCategory.ItemsSource = \_cateRepo.GetCategories();

cboCategory.DisplayMemberPath = "CategoryName";

cboCategory.SelectedValue = 1;

cboCategory.SelectedValuePath = "CategoryId";

--- Lấy giá trị từ combo box trong update:

CategoryId = int.Parse(cboCategory.SelectedValue.ToString())

--- ShutDown App (Thường trong nút close button)

Application.Current.Shutdown();

--- Lấy dữ liệu đang chọn từ data grid:

Project p = dgData.SelectedItem as Project;

--- FillDataGrid

private void FillDataGrid()

{

var products = repo.GetAllProject();

dgData.ItemsSource = null;

dgData.ItemsSource = products;

}

--- Fix lỗi không hiển thị các thuộc tính trên WPF

Step1: Thêm vào thằng nào thiếu : VerticalAlignment="Top"

Step2: Di chuyển nhẹ 1 tí để App nhận ra là đã có sự thay đổi

--- 1 dạng đề không có bảng Type nhưng bắt hiển thị Type ở combox

Step 1: \_context.Projects.Select(p => p.Type).Distinct().ToList(); để lấy list các type có trong bảng project

Step 2: Gán cho .ItemSource (Vì chuỗi string nên là không dùng MemberDisplayPath, SelectedValuePath)

--- Template 01:

<DataGrid AutoGenerateColumns="False" Margin="25,75,194,36">

<DataGrid.Columns>

<DataGridTextColumn Header="CourseId" Width="60"/>

<DataGridTextColumn Header="Title" Width="100"/>

<DataGridTextColumn Header="Description" Width="200"/>

<DataGridTextColumn Header="IntrucstorId" Width="60"/>

<DataGridTextColumn Header="IntrucstorName" Width="100"/>

</DataGrid.Columns>

</DataGrid>

<ComboBox HorizontalAlignment="Left" Margin="25,36,0,0" VerticalAlignment="Top" Width="581"/>

<Label Content="Id" HorizontalAlignment="Left" Margin="634,120,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.401,0.554"/>

<Label Content="Title" HorizontalAlignment="Left" Margin="628,147,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.621,1.021"/>

<Label Content="Description" HorizontalAlignment="Left" Margin="609,168,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.621,1.021"/>

<Label Content="Instructor" HorizontalAlignment="Left" Margin="611,191,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.621,1.021"/>

<TextBox HorizontalAlignment="Left" Margin="680,149,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="94" Height="21"/>

<TextBox HorizontalAlignment="Left" Margin="680,170,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="94" Height="21"/>

<ComboBox HorizontalAlignment="Left" Margin="680,191,0,0" VerticalAlignment="Top" Width="94" Height="21"/>

<Button Content="Edit" HorizontalAlignment="Left" Margin="644,227,0,0" VerticalAlignment="Top" Width="90"/>

---

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

{

Course c = dgData.SelectedItem as Course;

if(c!= null)

{

lbId.Content = c.CourseId; // Label

txtTitle.Text = c.Title; // TextBox

txtDescription.Text = c.Description; // TextBox

cboInstructor.SelectedValue = c.InstructorId; // ComboBox

}

}

---

public void FillComboBoxIntruc()

{

cboInstructor.ItemsSource = repo.GetAllInstructor();

cboInstructor.DisplayMemberPath = "Name";

cboInstructor.SelectedValue = 1;

cboInstructor.SelectedValuePath = "InstructorId";

}

---

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

{

try

{

Course c = new Course()

{

CourseId = int.Parse(lbId.Content.ToString()),

Title = txtTitle.Text,

Description = txtDescription.Text,

InstructorId = int.Parse(cboInstructor.SelectedValue.ToString()),

};

repo.EditCourse(c);

LoadDataGrid();// Caapj nhat lai du lieu moi nhat

MessageBox.Show("Update success!", "Updated", MessageBoxButton.OK, MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show("Update failed!","Error",MessageBoxButton.OK, MessageBoxImage.Error);

}

}

---

GUI

login

<Grid>

<Label Content="LOGIN" FontSize="30" HorizontalAlignment="Left" Margin="290,94,0,0" FontWeight="Bold" VerticalAlignment="Top" Height="54" Width="124"/>

<Label Content="Email" FontWeight="Bold" HorizontalAlignment="Left" Margin="198,153,0,0" VerticalAlignment="Top" Height="23" Width="70"/>

<TextBox x:Name="txtEmail" HorizontalAlignment="Left" Margin="282,153,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="206" Height="23"/>

<PasswordBox x:Name="txtPassword" HorizontalAlignment="Left" Margin="282,193,0,0" VerticalAlignment="Top" Width="206" Height="22"/>

<Label Content="Password" FontWeight="Bold" HorizontalAlignment="Left" Margin="198,192,0,0" VerticalAlignment="Top" Height="23" Width="79"/>

<Button x:Name="btnLogin" Content="LOGIN" HorizontalAlignment="Left" Margin="290,282,0,0" VerticalAlignment="Top" Height="29" Width="122" Click="btnLogin\_Click"/>

</Grid>

xName: btnLogin

textbox: txtEmail, txtPassword

Datagrid + 4 btn CRUD (Không edit thẳng trong màn này được):

<Grid >

<DataGrid x:Name="dgData" AutoGenerateColumns="False" Margin="25,75,194,36">

<DataGrid.Columns>

<DataGridTextColumn Header="Col1" Binding="{Binding Property1}" Width="60"/>

<DataGridTextColumn Header="Col2" Binding="{Binding Property1}" Width="100"/>

<DataGridTextColumn Header="Col3" Binding="{Binding Property1}" Width="200"/>

<DataGridTextColumn Header="Col4" Binding="{Binding Property1}" Width="60" />

</DataGrid.Columns>

</DataGrid>

<Button x:Name="btnAdd" Content="Add Item" HorizontalAlignment="Left" Margin="638,75,0,0" VerticalAlignment="Top" Height="41" Width="115"/>

<Button x:Name="btnEdit" Content="Edit Item" HorizontalAlignment="Left" Margin="638,135,0,0" VerticalAlignment="Top" Height="41" Width="115" />

<Button x:Name="btnDelete" Content="Delete Item" HorizontalAlignment="Left" Margin="638,201,0,0" VerticalAlignment="Top" Height="41" Width="115"/>

<TextBox x:Name="txtArchi" HorizontalAlignment="Left" Margin="100,26,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="152" Height="26"/>

<TextBlock HorizontalAlignment="Left" Margin="16,31,0,0" TextWrapping="Wrap" Text="Label1" VerticalAlignment="Top"/>

<TextBox x:Name="txtPlayerName" HorizontalAlignment="Left" Margin="391,26,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="152" Height="26"/>

<TextBlock HorizontalAlignment="Left" Margin="308,34,0,0" TextWrapping="Wrap" Text="Label2" VerticalAlignment="Top"/>

<Button x:Name="btnSearch" Content="Search" HorizontalAlignment="Left" Margin="638,27,0,0" VerticalAlignment="Top" Height="25" Width="115" />

<Button x:Name="btnClose" Content="Close" HorizontalAlignment="Left" Margin="696,445,0,0" VerticalAlignment="Top" Height="29" Width="94" />

</Grid>

Datagrid + 4 btn CRUD (Edit thẳng trong màn này được):

<Grid>

<DataGrid x:Name="dgData" AutoGenerateColumns="False" Margin="10,36,347,39" SelectionChanged="dgData\_SelectionChanged">

<DataGrid.Columns>

<DataGridTextColumn Header="Col1" Binding="{Binding Property}" Width="30"/>

<DataGridTextColumn Header="Col1" Binding="{Binding Property}" Width="60"/>

<DataGridTextColumn Header="Col1" Binding="{Binding Property}" Width="150"/>

<DataGridTextColumn Header="Col1" Binding="{Binding StartDate, StringFormat=\{0:dd/MM/yyyy\}}" Width="100"/>

<DataGridTextColumn Header="Col1" Binding="{Binding Property}" Width="60"/>

</DataGrid.Columns>

</DataGrid>

<GroupBox Header="Current Project" Margin="481,29,17,39"/>

<Button x:Name="Refresh\_button" Content="Refresh" HorizontalAlignment="Left" Margin="379,404,0,0" VerticalAlignment="Top" Width="74" Click="Refresh\_button\_Click"/>

<Button x:Name="Add\_button" Content="Add" HorizontalAlignment="Left" Margin="509,300,0,0" VerticalAlignment="Top" Width="107" Click="Add\_button\_Click"/>

<ComboBox x:Name="cboType" HorizontalAlignment="Left" Margin="570,224,0,0" Width="192" VerticalAlignment="Top"

/>

<Button x:Name="Edit\_button" Content="Edit" HorizontalAlignment="Left" Margin="655,300,0,0" VerticalAlignment="Top" Width="107" Click="Edit\_button\_Click" RenderTransformOrigin="0.579,-0.274"/>

<TextBox x:Name="txtId" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="570,67,0,0" TextWrapping="Wrap" Width="124"/>

<Label VerticalAlignment="Top" Content="ID" HorizontalAlignment="Left" Margin="486,59,0,0" Width="46" RenderTransformOrigin="0.742,1.055"/>

<Label VerticalAlignment="Top" Content="Name" HorizontalAlignment="Left" Margin="486,88,0,0" Width="44" RenderTransformOrigin="0.742,1.055"/>

<Label VerticalAlignment="Top" Content="Description" HorizontalAlignment="Left" Margin="486,114,0,0" Width="91" RenderTransformOrigin="0.742,1.055"/>

<Label VerticalAlignment="Top" Content="StartDate" HorizontalAlignment="Left" Margin="486,189,0,0" Width="91" RenderTransformOrigin="0.742,1.055"/>

<Label VerticalAlignment="Top" Content="Type" HorizontalAlignment="Left" Margin="486,222,0,0" Width="92" RenderTransformOrigin="0.742,1.055"/>

<TextBox x:Name="txtName" HorizontalAlignment="Left" Margin="570,95,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="192"/>

<TextBox x:Name="txtDescription" HorizontalAlignment="Left" Margin="570,122,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="192" Height="63"/>

<DatePicker x:Name="selectDT" HorizontalAlignment="Left" Margin="570,190,0,0" VerticalAlignment="Top" Width="192" Height="25" RenderTransformOrigin="0.51,0.276"/>

<Button x:Name="Remove\_button" Content="Remove" HorizontalAlignment="Left" Margin="509,342,0,0" VerticalAlignment="Top" Width="107" RenderTransformOrigin="0.213,0.204" />

<Button x:Name="Close\_button" Content="Close" HorizontalAlignment="Left" Margin="718,409,0,0" VerticalAlignment="Top" Width="72" RenderTransformOrigin="1.027,0.549" />

</Grid>

### REPO LOGIN

public ClassTable? GetLogin(string email, string password)

{

context = new();

var res = context.ClassTable.FirstOrDefault(u => u.AccountPassword.Equals(password) && u.AccountEmail.Equals(email));

return res;

}

### CRUD Item

public List<ClassTable> GetAllFootballPlayer()

{

context = new();

return context.ClassTable.Include(u => u.FootballTeam).ToList();

}

public void AddPlayer(FootballPlayer f)

{

context = new();

context.FootballPlayers.Add(f);

context.SaveChanges();

}

public List<FootballPlayer> SearchFootballPlayer(string archivement, string name)

{

context = new();

List<FootballPlayer> res = GetAllFootballPlayer();

if (archivement != null && !archivement.IsNullOrEmpty())

{

res = res.FindAll(u => u.Achievements.Contains(archivement));

}

if (name != null && !name.IsNullOrEmpty())

{

res = res.FindAll(u => u.PlayerName.Contains(name));

}

return res;

}

public void UpdatePlayer(FootballPlayer football)

{

context = new();

context.FootballPlayers.Update(football);

context.SaveChanges();

}

public void RemoveItem(FootballPlayer item)

{

context = new();

context.FootballPlayers.Remove(item);

context.SaveChanges();

}

MessageBox:

Error:

MessageBox.Show("Login error", "Error", MessageBoxButton.OK, MessageBoxImage.Error);

Success:

MessageBox.Show("Login success", "Success", MessageBoxButton.OK, MessageBoxImage.Information);

Warning:

MessageBox.Show("Login Warning", "Warning", MessageBoxButton.OK, MessageBoxImage.Warning);

YesNo:

MessageBox.Show("Do you want remove this item", "Remove?", MessageBoxButton.YesNo, MessageBoxImage.Information);