using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

/\*Lungile Shongwe

\* Purpose:. Its purpose is to facilitate secure access to the application's features and content by verifying the identity

\*/

namespace App1

{

// This class represents the login user interface

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class LoginUI : ContentPage

{

public LoginUI()

{

InitializeComponent();

}

// This method is called when the login button is clicked

private async void Button\_Clicked(object sender, EventArgs e)

{

// Check if email or password fields are empty

if (string.IsNullOrWhiteSpace(txtEmail.Text) || string.IsNullOrWhiteSpace(txtPassword.Text))

{

// Display an error alert if either field is empty

await DisplayAlert("Error", "Please enter email and password.", "OK");

return;

}

// Check if the entered email is in a valid format

if (!IsValidEmail(txtEmail.Text))

{

// Display an error alert if the email format is invalid

await DisplayAlert("Error", "Invalid Email format.", "OK");

return;

}

// Authenticate the user with entered credentials

bool isAuthenticated = AuthenticateUser(txtEmail.Text, txtPassword.Text);

if (isAuthenticated)

{

// Navigate to the main page upon successful login

await Navigation.PushAsync(new MainPage());

}

else

{

// Display an error alert if the entered credentials are incorrect

await DisplayAlert("Oops...", "Username or Password is incorrect!", "Ok");

}

}

// This method performs user authentication based on predefined email and password

private bool AuthenticateUser(string email, string password)

{

// Define predefined email and password

string predefinedEmail = "user@example.com";

string predefinedPassword = "password";

// Check if the entered email and password match the predefined values

return email == predefinedEmail && password == predefinedPassword;

}

// This method validates the format of an email address using a regular expression

private bool IsValidEmail(string email)

{

// Regular expression pattern for email validation

string emailPattern = @"^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$";

// Check if the email matches the pattern

return Regex.IsMatch(email, emailPattern);

}

// This method is called when the register label is tapped

private async void OnRegisterLabelTapped(object sender, EventArgs e)

{

// Navigate to the registration page

await Navigation.PushAsync(new RegistrationPage());

}

// This method is a placeholder for a gesture recognizer event, currently not used

private void TapGestureRecognizer\_Tapped(object sender, EventArgs e)

{

// This method does nothing for now

}

}

}

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App1.LoginUI" BackgroundColor="#112BA7">

<ContentPage.Content>

<StackLayout Padding="40" VerticalOptions="Center" HorizontalOptions="FillAndExpand" >

<Image Source="Screenshot.png" WidthRequest="180" HeightRequest="180"></Image>

<Entry x:Name="txtEmail" Placeholder="Email" TextColor="White" PlaceholderColor="Gray" ></Entry>

<Entry x:Name="txtPassword" Placeholder="Password" IsPassword="True" TextColor="White" PlaceholderColor="Gray" ></Entry>

<Button

VerticalOptions="Center"

Text="Login"

TextColor="White"

BackgroundColor="Transparent"

BorderColor="ForestGreen"

BorderWidth="1.5"

CornerRadius="50"

Clicked="Button\_Clicked"

></Button>

<Label HorizontalOptions="Center">

<Label.FormattedText>

<FormattedString>

<Span Text="Don't have an account?" TextColor="Gray"/>

<Span Text="Register" TextColor="Gray" FontAttributes="Bold" TextDecorations="Underline" />

</FormattedString>

</Label.FormattedText>

<Label.GestureRecognizers>

<TapGestureRecognizer Tapped="OnRegisterLabelTapped"/>

</Label.GestureRecognizers>

</Label>

</StackLayout>

</ContentPage.Content>

</ContentPage>

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App1.MainPage">

<StackLayout VerticalOptions="CenterAndExpand" Padding="20">

<Label x:Name="welcomeLabel" Text="Hello, [UserFirstName]!" FontSize="Subtitle" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Label Text="Welcome!" FontSize="Title" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Label Text="Hello, [UserFirstName]!" FontSize="Subtitle" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Button Text="View Account Balance" Clicked="OnViewAccountBalanceClicked" HorizontalOptions="FillAndExpand" Margin="0,10,0,0"/>

<Button Text="Transfer Between Accounts" Clicked="OnTransferBetweenAccountsClicked" HorizontalOptions="FillAndExpand" Margin="0,10,0,0"/>

<Button Text="Logout" Clicked="OnLogoutClicked" HorizontalOptions="FillAndExpand" Margin="0,10,0,0"/>

</StackLayout>

</ContentPage>

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

namespace App1

{

public partial class MainPage : ContentPage

{

public MainPage()

{

InitializeComponent();

string userFirstName = GetCurrentUserFirstName();

welcomeLabel.Text = $"Welcome, {userFirstName}!";

}

private async void OnViewAccountBalanceClicked(object sender, EventArgs e)

{

// Navigating to the View Account Balance page

await Navigation.PushAsync(new ViewAccountBalancePage());

}

private async void OnTransferBetweenAccountsClicked(object sender, EventArgs e)

{

// Navigating to the Transfer Between Accounts page

await Navigation.PushAsync(new TransferBetweenAccountsPage());

}

private async void OnLogoutClicked(object sender, EventArgs e)

{

// logout action

await Navigation.PopToRootAsync();

}

private string GetCurrentUserFirstName()

{

return "Lungile";

}

}

}

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App1.RegistrationPage">

<StackLayout VerticalOptions="CenterAndExpand" Padding="20">

<Label Text="Registration" FontSize="Title" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Entry Placeholder="First Name" x:Name="firstNameEntry"/>

<Entry Placeholder="Last Name" x:Name="lastNameEntry"/>

<Entry Placeholder="Email" x:Name="emailEntry"/>

<Entry Placeholder="Password" IsPassword="True" x:Name="passwordEntry"/>

<Entry Placeholder="Mobile Number" Keyboard="Telephone" x:Name="mobileEntry"/>

<Picker Title="Gender" x:Name="genderPicker">

<Picker.Items>

<x:String>Male</x:String>

<x:String>Female</x:String>

<x:String>Other</x:String>

</Picker.Items>

</Picker>

<Button Text="Register" Clicked="OnRegisterButtonClicked" HorizontalOptions="FillAndExpand" Margin="0,20,0,0"/>

<Button Text="Login" Clicked="OnLoginButtonClicked" HorizontalOptions="FillAndExpand" Margin="0,10,0,0"/>

</StackLayout>

</ContentPage>

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

/\*Lungile Shongwe

\* Purpose: To facilitate the creation of new user accounts and provide information such as their email address, desired username, password.\*/

namespace App1

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class RegistrationPage : ContentPage

{

public RegistrationPage()

{

InitializeComponent();

}

private async void OnRegisterButtonClicked(object sender, EventArgs e)

{

// validatio

if (string.IsNullOrWhiteSpace(firstNameEntry.Text) ||

string.IsNullOrWhiteSpace(lastNameEntry.Text) ||

string.IsNullOrWhiteSpace(emailEntry.Text) ||

string.IsNullOrWhiteSpace(passwordEntry.Text) ||

string.IsNullOrWhiteSpace(mobileEntry.Text) ||

genderPicker.SelectedIndex == -1)

{

await DisplayAlert("Error", "Please fill in all fields.", "OK");

return;

}

// Performing further validation such as email format

if (!IsValidEmail(emailEntry.Text))

{

await DisplayAlert("Error", "Please enter a valid email address.", "OK");

return;

}

// Performoing additional validation as per requirements (e.g., password length)

if (passwordEntry.Text.Length < 5)

{

await DisplayAlert("Error", "Password must be at least five characters long.", "OK");

return;

}

/\* var newUser = new DatabaseHelper.User

{

FirstName = firstNameEntry.Text,

LastName = lastNameEntry.Text,

Email = emailEntry.Text,

Password = passwordEntry.Text,

mobileNumber = mobileEntry.Text,

gender = genderPicker.SelectedItem.ToString()

};\*/

// App.Database.InsertUser(newUser);

await DisplayAlert("Success", "Registration successful!", "OK");

// Navigate to the login page after successful registration

await Navigation.PushAsync(new LoginUI());

}

private async void OnLoginButtonClicked(object sender, EventArgs e)

{

// Navigating to the login page

await Navigation.PushAsync(new LoginUI());

}

private bool IsValidEmail(string email)

{

//email validation

return !string.IsNullOrWhiteSpace(email) && email.Contains("@") && email.Contains(".");

}

}

}

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App1.TransferBetweenAccountsPage">

<StackLayout VerticalOptions="CenterAndExpand" Padding="20">

<Label Text="Transfer Between Accounts" FontSize="Title" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Entry Placeholder="Amount to Transfer" Keyboard="Numeric" x:Name="amountEntry"/>

<Picker Title="Select Transfer Direction" x:Name="directionPicker">

<Picker.ItemsSource>

<x:Array Type="{x:Type x:String}">

<x:String>From Current to Savings</x:String>

<x:String>From Savings to Current</x:String>

</x:Array>

</Picker.ItemsSource>

</Picker>

<Button Text="Transfer" Clicked="OnTransferClicked" HorizontalOptions="FillAndExpand" Margin="0,20,0,0"/>

</StackLayout>

</ContentPage>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

/\*Lungile Shongwe

\* Purpose: Transfers Between Accounts being made successfully

\*/

namespace App1

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class TransferBetweenAccountsPage : ContentPage

{

public TransferBetweenAccountsPage ()

{

InitializeComponent ();

}

private async void OnTransferClicked(object sender, EventArgs e)

{

// Get the amount to transfer from the entry field

if (!double.TryParse(amountEntry.Text, out double amount))

{

await DisplayAlert("Error", "Please enter a valid amount.", "OK");

return;

}

// Determining the transfer direction

string transferDirection = directionPicker.SelectedItem as string;

if (string.IsNullOrEmpty(transferDirection))

{

await DisplayAlert("Error", "Please select a transfer direction.", "OK");

return;

}

// Perform the transfer based on the selected direction

if (transferDirection == "From Current to Savings")

{

// current account to savings account

await DisplayAlert("Success", $"Transferred ${amount} from Current to Savings.", "OK");

}

else if (transferDirection == "From Savings to Current")

{

// from savings account to current account

await DisplayAlert("Success", $"Transferred ${amount} from Savings to Current.", "OK");

}

// Clearing entry field after transfer

amountEntry.Text = string.Empty;

}

}

}

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

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

x:Class="App1.ViewAccountBalancePage">

<StackLayout VerticalOptions="CenterAndExpand" Padding="20">

<Label Text="Account Balances" FontSize="Title" HorizontalOptions="Center" Margin="0,0,0,20"/>

<Label Text="Current Account Balance: $5000" HorizontalOptions="Center"/>

<Label Text="Savings Account Balance: $10000" HorizontalOptions="Center" Margin="0,10,0,0"/>

<Button Text="Back to Main Page" Clicked="OnBackToMainPageClicked" HorizontalOptions="FillAndExpand" Margin="0,20,0,0"/>

</StackLayout>

</ContentPage>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

/\* Lungile Shongwe

\* Purpose: On this Page the user can view his or her account balance.

\*/

namespace App1

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class ViewAccountBalancePage : ContentPage

{

public ViewAccountBalancePage ()

{

InitializeComponent ();

BindingContext = this;

// Setting the initial account balance values

CurrentAccountBalance = 5000;

SavingsAccountBalance = 10000;

}

public double CurrentAccountBalance { get; set; }

public double SavingsAccountBalance { get; set; }

private async void OnBackToMainPageClicked(object sender, EventArgs e)

{

// Navigate back to the main page

await Navigation.PopAsync();

}

}

}