Project Report - BrightBudget

A blue background with white text

AI-generated content may be incorrect.

Mobile and windows store development

**By Group-1**

SAhinda MAharjan – 4560719

swikrit Khatiwada – 4560139

2025

Table of Contents

[Overview: 3](#_Toc202733793)

[UI Screenshots: 4](#_Toc202733794)

[1. Signup 4](#_Toc202733795)

[2. Login 5](#_Toc202733796)

[3. Home 6](#_Toc202733797)

[4. Menu (Hamburger) 7](#_Toc202733798)

[5. Add Income & Expenses 8](#_Toc202733799)

[6. About Us 9](#_Toc202733800)

[7. Contact Us 11](#_Toc202733801)

[8. Plan Selection 13](#_Toc202733802)

[9. Payment 14](#_Toc202733803)

[10. Successful Payment 15](#_Toc202733804)

[XAML Code: 16](#_Toc202733805)

[1. Signup 16](#_Toc202733806)

[2. Login 16](#_Toc202733807)

[3. Home 16](#_Toc202733808)

[4. Add Income & Expenses 16](#_Toc202733809)

[5. About Us 16](#_Toc202733810)

[6. Contact Us 16](#_Toc202733811)

[7. Plan Selection 16](#_Toc202733812)

[8. Payment 16](#_Toc202733813)

[9. Successful Payment 16](#_Toc202733814)

[C# Code 16](#_Toc202733815)

[1. Signup 16](#_Toc202733816)

[2. Login 16](#_Toc202733817)

[3. Home 16](#_Toc202733818)

[4. Menu (Hamburger) 16](#_Toc202733819)

[5. Add Income & Expenses 16](#_Toc202733820)

[6. About Us 16](#_Toc202733821)

[7. Contact Us 16](#_Toc202733822)

[8. Plan Selection 16](#_Toc202733823)

[9. Payment 16](#_Toc202733824)

[10. Successful Payment 16](#_Toc202733825)

[Challenges Faced: 17](#_Toc202733826)

[Future Work: 18](#_Toc202733827)

[Live Demo 19](#_Toc202733828)

[References 20](#_Toc202733829)

# Overview:

Budget Tracker App is a mobile application designed with the help of Xamarin. Forms. It enables users to take control of their finance, keeping track of their incomes and expenses, choosing free or paid plans, and making payments within an app. The main aspects are an easy-to-understand login and registration feature, a dashboard reflecting the budget overview, and including the payment gateway to manage subscriptions. It is an application that should make personal finance management easy to use with a clean and familiar interface that can be used on Android devices.

# UI Screenshots:

## Signup

A screenshot of a phone

AI-generated content may be incorrect.

## Login

A screenshot of a login screen

AI-generated content may be incorrect.

## Home

A screenshot of a cell phone

AI-generated content may be incorrect.

## Menu (Hamburger)

A screenshot of a phone

AI-generated content may be incorrect.

## Add Income & Expenses

A screenshot of a calculator

AI-generated content may be incorrect.

## About Us

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone

AI-generated content may be incorrect.

## Contact Us

A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a phone application

AI-generated content may be incorrect.

## Plan Selection

A screenshot of a phone

AI-generated content may be incorrect.

## Payment

A screenshot of a credit card

AI-generated content may be incorrect.

## Successful Payment

A screenshot of a subscription box

AI-generated content may be incorrect.

# XAML Code:

## Signup

<?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="Project.Signup">

<ContentPage.Content>

<ScrollView>

<StackLayout Padding="30" Spacing="20" VerticalOptions="CenterAndExpand">

<Image Source="logo.png"

HeightRequest="80"

WidthRequest="160"

HorizontalOptions="Center"/>

<Label Text="Create your Account"

FontSize="20"

FontAttributes="Bold"

TextColor="Black"

HorizontalOptions="Center"/>

<!-- First Name -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

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

FontSize="16"

TextColor="Black"/>

</Frame>

<!-- Last Name -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

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

FontSize="16"

TextColor="Black"/>

</Frame>

<!-- Email -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

<Entry x:Name="emailerror" Placeholder="Email"

FontSize="16"

Keyboard="Email"

TextColor="Black"/>

</Frame>

<!-- Password -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

<Entry x:Name="pwerror" Placeholder="Password"

IsPassword="True"

FontSize="16"

TextColor="Black"/>

</Frame>

<!-- Confirm Password -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

<Entry x:Name="Confirmerror" Placeholder="Confirm password"

IsPassword="True"

FontSize="16"

TextColor="Black"/>

</Frame>

<!-- Signup Button -->

<Button Clicked="Btnclick\_Clicked" Text="Sign up"

BackgroundColor="Teal"

TextColor="White"

FontSize="16"

CornerRadius="25"

HeightRequest="50"

HorizontalOptions="Center"

WidthRequest="250"/>

<Label x:Name="errorLabel"

TextColor="Red"

FontSize="14"

IsVisible="False"

HorizontalOptions="Center" />

<Label Text="– Or sign up with –"

HorizontalOptions="Center"

FontSize="14"

TextColor="Gray"/>

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

Spacing="20">

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="google.png" HeightRequest="24" WidthRequest="24"/>

</Frame>

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="fblogo.png" HeightRequest="24" WidthRequest="24" />

</Frame>

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="x.png" HeightRequest="34" WidthRequest="24"/>

</Frame>

</StackLayout>

</StackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

## Login

<?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="Project.Login">

<ContentPage.Content>

<ScrollView>

<StackLayout Padding="30" Spacing="20" VerticalOptions="CenterAndExpand">

<Image Source="Logo.jpg"

HeightRequest="80"

WidthRequest="160"

HorizontalOptions="Center"/>

<Label Text="Welcome Back"

FontSize="20"

FontAttributes="Bold"

TextColor="Black"

HorizontalOptions="Center"/>

<!-- Email -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

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

FontSize="16"

Keyboard="Email"

TextColor="Black"/>

</Frame>

<!-- Password -->

<Frame HasShadow="True" CornerRadius="10" Padding="10" BackgroundColor="White">

<Entry x:Name="passwordEntry" Placeholder="Password"

IsPassword="True"

FontSize="16"

TextColor="Black"/>

</Frame>

<!-- Login Button -->

<Button Text="Log in"

Clicked="LoginButton\_Clicked"

BackgroundColor="Teal"

TextColor="White"

FontSize="16"

CornerRadius="25"

HeightRequest="50"

HorizontalOptions="Center"

WidthRequest="250"/>

<!-- Error Label -->

<Label x:Name="errorLabel"

TextColor="Red"

FontSize="14"

IsVisible="False"

HorizontalOptions="Center" />

<!-- Social Login Text -->

<Label Text="– Or log in with –"

HorizontalOptions="Center"

FontSize="14"

TextColor="Gray"/>

<!-- Social Icons -->

<StackLayout Orientation="Horizontal"

HorizontalOptions="Center"

Spacing="20">

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="google.png" HeightRequest="24" WidthRequest="24"/>

</Frame>

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="fblogo.png" HeightRequest="24" WidthRequest="24"/>

</Frame>

<Frame Padding="10" CornerRadius="10" HasShadow="True" BackgroundColor="White">

<Image Source="x.png" HeightRequest="34" WidthRequest="24"/>

</Frame>

</StackLayout>

<!-- Sign up link -->

<StackLayout Orientation="Horizontal" HorizontalOptions="Center" Margin="0,20,0,0">

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

<StackLayout>

<Label Text="Sign up" TextColor="Blue" FontAttributes="Bold"/>

<StackLayout.GestureRecognizers>

<TapGestureRecognizer Tapped="Signup\_Tapped" />

</StackLayout.GestureRecognizers>

</StackLayout>

</StackLayout>

</StackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

## Home

<?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="Project.Home"

Title="Dashboard"

BackgroundColor="White">

<ScrollView>

<StackLayout Padding="20" Spacing="25">

<Frame BackgroundColor="#2196F3" CornerRadius="15" Padding="20" HasShadow="True">

<StackLayout>

<Label Text="Total Budget" FontSize="18" TextColor="White" />

<Label Text="$5,000" FontSize="36" FontAttributes="Bold" TextColor="White" />

</StackLayout>

</Frame>

<Frame BackgroundColor="#4CAF50" CornerRadius="15" Padding="20" HasShadow="True">

<StackLayout>

<Label Text="Expenses This Month" FontSize="18" TextColor="White" />

<Label Text="$2,100" FontSize="36" FontAttributes="Bold" TextColor="White" />

</StackLayout>

</Frame>

<Frame BackgroundColor="#FF9800" CornerRadius="15" Padding="20" HasShadow="True">

<StackLayout>

<Label Text="Remaining Budget" FontSize="18" TextColor="White" />

<Label Text="$2,900" FontSize="36" FontAttributes="Bold" TextColor="White" />

</StackLayout>

</Frame>

<Button Text="Add Record"

BackgroundColor="Teal"

TextColor="White"

CornerRadius="25"

HeightRequest="50"

HorizontalOptions="FillAndExpand"

Clicked="Button\_Clicked"/>

</StackLayout>

</ScrollView>

</ContentPage>

## Add Income & Expenses

<?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="Project.AddIncomeExpenses"

Title="Add Record">

<StackLayout Padding="10" Spacing="10">

<!-- Cancel and Save buttons -->

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<Button Text="✕ CANCEL" TextColor="Red" BackgroundColor="Transparent"

Clicked="OnCancelClicked" HorizontalOptions="Start" />

<Button Text="✓ SAVE" TextColor="Blue" BackgroundColor="Transparent"

Clicked="OnSaveClicked" HorizontalOptions="End" Grid.Column="1" />

</Grid>

<!-- Transaction type toggle -->

<StackLayout Orientation="Horizontal" HorizontalOptions="Center" Spacing="15">

<RadioButton x:Name="incomeButton" TextColor="Gray" Padding="10, 5" CornerRadius="20" BackgroundColor="White" Content="Income" GroupName="TransactionType" Value="Income"

CheckedChanged="OnTransactionTypeChanged" />

<RadioButton x:Name="expenseButton" Content="Expense" TextColor="Gray" Padding="10, 5" CornerRadius="20" BackgroundColor="White" GroupName="TransactionType" Value="Expense"

IsChecked="True" CheckedChanged="OnTransactionTypeChanged" />

</StackLayout>

<!-- Notes input -->

<Editor Placeholder="Add notes" PlaceholderColor="Gray" BackgroundColor="LightGray" TextColor="Black" HeightRequest="80" />

<!-- Amount display -->

<Label x:Name="AmountDisplay" Text="0" FontSize="36"

TextColor="Gray" HorizontalOptions="End" />

<!-- Calculator keypad -->

<Grid ColumnSpacing="5" RowSpacing="5">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!-- Row 1 -->

<Button Text="+" Clicked="OnOperatorClicked" Grid.Row="0" Grid.Column="0" />

<Button Text="7" Clicked="OnNumberClicked" Grid.Row="0" Grid.Column="1" />

<Button Text="8" Clicked="OnNumberClicked" Grid.Row="0" Grid.Column="2" />

<Button Text="9" Clicked="OnNumberClicked" Grid.Row="0" Grid.Column="3" />

<!-- Row 2 -->

<Button Text="-" Clicked="OnOperatorClicked" Grid.Row="1" Grid.Column="0" />

<Button Text="4" Clicked="OnNumberClicked" Grid.Row="1" Grid.Column="1" />

<Button Text="5" Clicked="OnNumberClicked" Grid.Row="1" Grid.Column="2" />

<Button Text="6" Clicked="OnNumberClicked" Grid.Row="1" Grid.Column="3" />

<!-- Row 3 -->

<Button Text="×" Clicked="OnOperatorClicked" Grid.Row="2" Grid.Column="0" />

<Button Text="1" Clicked="OnNumberClicked" Grid.Row="2" Grid.Column="1" />

<Button Text="2" Clicked="OnNumberClicked" Grid.Row="2" Grid.Column="2" />

<Button Text="3" Clicked="OnNumberClicked" Grid.Row="2" Grid.Column="3" />

<!-- Row 4 -->

<Button Text="÷" Clicked="OnOperatorClicked" Grid.Row="3" Grid.Column="0" />

<Button Text="0" Clicked="OnNumberClicked" Grid.Row="3" Grid.Column="1" />

<Button Text="." Clicked="OnNumberClicked" Grid.Row="3" Grid.Column="2" />

<Button Text="=" Clicked="OnEqualsClicked" Grid.Row="3" Grid.Column="3" />

</Grid>

<Button Text="Clear" BackgroundColor="Teal" TextColor="White" BorderWidth="10" Clicked="OnClearClicked"/>

<!-- Date and Time pickers -->

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

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<DatePicker x:Name="DateInput" BackgroundColor="Teal" TextColor="White" Grid.Column="0" />

<TimePicker x:Name="TimeInput" BackgroundColor="Teal" TextColor="White" Grid.Column="1" />

</Grid>

</StackLayout>

</ContentPage>

## About Us

<?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="Project.AboutUs"

Title="About Us"

BackgroundColor="White">

<ScrollView>

<Grid Padding="25" RowSpacing="40" ColumnSpacing="25">

<!-- Rows -->

<Grid.RowDefinitions>

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

<RowDefinition Height="Auto" />

</Grid.RowDefinitions>

<!-- Columns -->

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<!-- Intro -->

<StackLayout Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" HorizontalOptions="Center" Spacing="15">

<Image Source="Logo.jpg" HeightRequest="100" WidthRequest="100" />

<Label Text="Welcome to Budget Tracker! We help you manage your personal finances effortlessly. Our team is dedicated to making budgeting easy and fun."

FontSize="16"

HorizontalTextAlignment="Center" />

</StackLayout>

<!-- Team Title -->

<Label Text="Our Team"

FontSize="30"

FontAttributes="Bold"

HorizontalOptions="Center"

Grid.Row="1"

Grid.Column="0"

Grid.ColumnSpan="2" />

<!-- Sahinda: Image Left, Text Right -->

<Frame Grid.Row="2" Grid.Column="0" Grid.ColumnSpan="2" Padding="15" Margin="0" HasShadow="True" CornerRadius="10" BackgroundColor="#E0F2F1">

<Grid ColumnSpacing="20">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto" />

<ColumnDefinition Width="\*" />

</Grid.ColumnDefinitions>

<StackLayout Grid.Column="0" HorizontalOptions="Center" VerticalOptions="Center">

<Image Source="sahinda.png"

Aspect="AspectFit"

HeightRequest="85"

WidthRequest="85" />

<Label Text="Sahinda Maharjan" FontSize="18" FontAttributes="Bold" TextColor="#004d40" />

</StackLayout>

<StackLayout Grid.Column="1" VerticalOptions="Center" Spacing="7">

<Label Text="• Computer programming student at Niagara College Toronto" FontSize="16" TextColor="#00695C" />

<Label Text="• Lives in Scarborough, Toronto" FontSize="16" TextColor="#00695C" />

<Label Text="• Loves coding and sketching" FontSize="16" TextColor="#00695C" />

<Label Text="sahinda@gmail.com"

FontSize="12"

TextColor="#00796B"

FontAttributes="Italic" />

</StackLayout>

</Grid>

</Frame>

<!-- Swikrit: Text Left, Image Right -->

<Frame Grid.Row="3" Grid.Column="0" Grid.ColumnSpan="2" Padding="15" Margin="0" HasShadow="True" CornerRadius="10" BackgroundColor="#E0F2F1">

<Grid ColumnSpacing="20">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*" />

<ColumnDefinition Width="Auto" />

</Grid.ColumnDefinitions>

<StackLayout Grid.Column="0" VerticalOptions="Center" Spacing="7">

<Label Text="• Computer programming student at Niagara College Toronto" FontSize="16" TextColor="#00695C" />

<Label Text="• Lives in Mississauga" FontSize="16" TextColor="#00695C" />

<Label Text="• Loves coding and watching anime" FontSize="16" TextColor="#00695C" />

<Label Text="swikritkhati010@gmail.com"

FontSize="12"

TextColor="#00796B"

FontAttributes="Italic" />

</StackLayout>

<StackLayout Grid.Column="1" HorizontalOptions="Center" VerticalOptions="Center">

<Image Source="swikrit.png"

Aspect="AspectFit"

HeightRequest="85"

WidthRequest="85"/>

<Label Text="Swikrit Khatiwada" FontSize="18" FontAttributes="Bold" TextColor="#004d40" />

</StackLayout>

</Grid>

</Frame>

</Grid>

</ScrollView>

</ContentPage>

## Contact Us

<?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="Project.Contactpage"

BackgroundColor="White"

Title="Contact Us">

<ContentPage.Content>

<ScrollView>

<StackLayout Padding="20" Spacing="25">

<!-- Contact Info Card -->

<Frame CornerRadius="15" BackgroundColor= "Teal" Padding="20" HasShadow="True">

<StackLayout Spacing="15" HorizontalOptions="Center">

<Label Text="Contact Information" FontSize="20" TextColor="White" FontAttributes="Bold" HorizontalOptions="Center" />

<Label Text="Say something to start a live chat!" FontSize="14" TextColor="LightGray" HorizontalOptions="Center" />

<Image Source="phone.png" HeightRequest="30" WidthRequest="30" HorizontalOptions="Center" />

<Label Text="+647 3456 789" FontSize="14" TextColor="White" HorizontalOptions="Center" />

<Image Source="mail.png" HeightRequest="30" WidthRequest="30" HorizontalOptions="Center" />

<Label Text="Brightbudgetcsr@gmail.com" FontSize="14" TextColor="White" HorizontalOptions="Center" />

<Image Source="location.png" HeightRequest="30" WidthRequest="30" HorizontalOptions="Center" />

<Label Text="132 Dundas Street Toronto, Ontario A4B 2S9 Canada"

FontSize="14" TextColor="White" HorizontalOptions="Center" HorizontalTextAlignment="Center" />

<StackLayout Orientation="Horizontal" HorizontalOptions="Center" Spacing="20">

<Image Source="x.png" HeightRequest="30" WidthRequest="30" />

<Image Source="instagram.png" HeightRequest="30" WidthRequest="30" />

<Image Source="discord.png" HeightRequest="30" WidthRequest="30" />

</StackLayout>

</StackLayout>

</Frame>

<StackLayout Spacing="15">

<Entry Placeholder="First Name" />

<Entry Placeholder="Last Name" />

<Entry Placeholder="Email" Keyboard="Email" />

<Entry Placeholder="Phone Number" Keyboard="Telephone" />

<Label Text="Select Subject?" TextColor="DarkGray" FontSize="20" FontAttributes="Bold"/>

<Grid ColumnSpacing="10" RowSpacing="10">

<RadioButton GroupName="subject" Content="General enquiry" IsChecked="True" Grid.Row="0" Grid.Column="0"/>

<RadioButton GroupName="subject" Content="Support" Grid.Row="0" Grid.Column="1"/>

<RadioButton GroupName="subject" Content="Feedback" Grid.Row="1" Grid.Column="0"/>

<RadioButton GroupName="subject" Content="Other" Grid.Row="1" Grid.Column="1"/>

</Grid>

<Editor Placeholder="Write your message.." HeightRequest="100" />

<Button Text="Send Message" BackgroundColor= "Teal" TextColor="White" CornerRadius="8" />

</StackLayout>

</StackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

## Plan Selection

<?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="Project.PlanSelection"

Title="PlanSelection">

<ContentPage.Content>

<ScrollView>

<StackLayout Padding="20" Spacing="25">

<!--Starter Plan-->

<Frame BorderColor="Gray" CornerRadius="10" Padding="15">

<StackLayout>

<Label Text="Starter" FontSize="18" FontAttributes="Bold"/>

<Label Text="Free" TextColor="Green"/>

<Label Text="✅ Local data only with ads"/>

<Label Text="✅ Track expenses and income"/>

<Label Text="✅ Basic balance overview"/>

</StackLayout>

</Frame>

<!--Premium Plan-->

<Frame BackgroundColor="Teal" CornerRadius="10" Padding="15">

<StackLayout>

<Label Text="Premium" TextColor="White" FontSize="18" FontAttributes="Bold"/>

<Label Text="$10/month" TextColor="AntiqueWhite"/>

<Label Text="✅ All features from Starter plan without ads" TextColor="AntiqueWhite"/>

<Label Text="✅ Charts and analytics" TextColor="AntiqueWhite"/>

<Label Text="✅ Reminders and themes" TextColor="AntiqueWhite"/>

</StackLayout>

</Frame>

<!--Button-->

<Button Text="Proceed to Payment"

Clicked="Button\_Clicked"

BackgroundColor="Teal"

TextColor="White"

CornerRadius="10"/>

</StackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

## Payment

<?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="Project.Payment"

Title="PlanSelection">

<ContentPage.Content>

<ScrollView>

<StackLayout Padding="20" Spacing="15">

<Label Text="Payment details" FontSize="24" FontAttributes="Bold" HorizontalOptions="StartAndExpand" />

<Frame BorderColor="LightGray" CornerRadius="10" Padding="10">

<StackLayout>

<Label Text="Credit Card" FontSize="18" FontAttributes="Bold" TextColor="Black"/>

<Label Text="Accepted cards:" FontSize="12" TextColor="Gray"/>

<StackLayout Padding="10" Orientation="Horizontal" Spacing="20">

<Image Source="visa.png" HeightRequest="40" WidthRequest="40"/>

<Image Source="amex.png" HeightRequest="40" WidthRequest="40"/>

<Image Source="mastercard.png" HeightRequest="40" WidthRequest="40"/>

</StackLayout>

<Label Text="Name on card" FontSize="14" FontAttributes="Bold" TextColor="Black"/>

<Entry Placeholder="Enter your name" TextColor="Gray" FontAttributes="Italic" />

<Label Text="Card number" FontSize="14" FontAttributes="Bold" TextColor="Black"/>

<Entry x:Name="CardNum" Placeholder="XXXX-XXXX-XXXX-XXXX" MaxLength="16" Keyboard="Numeric" TextColor="Gray" FontAttributes="Italic"/>

<StackLayout Orientation="Horizontal" Spacing="20">

<StackLayout HorizontalOptions="FillAndExpand" Spacing="5">

<Label Text="Expiry date" FontSize="14" FontAttributes="Bold" TextColor="Black"/>

<Entry x:Name="ExpiryDate" Placeholder="MM / YY" MaxLength="5" HorizontalOptions="FillAndExpand" FontAttributes="Italic"/>

</StackLayout>

<StackLayout HorizontalOptions="FillAndExpand" Spacing="5">

<Label Text="Security Code" FontSize="14" FontAttributes="Bold" TextColor="Black"/>

<Entry x:Name="CVV" Placeholder="CVV" Keyboard="Numeric" MaxLength="3" HorizontalOptions="FillAndExpand" FontAttributes="Italic"/>

</StackLayout>

</StackLayout>

</StackLayout>

</Frame>

<Button Text="Pay $10"

BackgroundColor="Teal"

TextColor="White"

CornerRadius="10"

Clicked="ConfirmedClicked" />

</StackLayout>

</ScrollView>

</ContentPage.Content>

</ContentPage>

## Successful Payment

<?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="Project.SuccessfulPayment"

Title="PlanSelection">

<ContentPage.Content>

<StackLayout Padding="30" Spacing="25" VerticalOptions="StartAndExpand">

<Image Source="success.png" HeightRequest="100" WidthRequest="100"/>

<Label Text="Subscription confirmed!" FontSize="24" FontAttributes="Bold" HorizontalOptions="Center"/>

<Label Text="Thank you for subscribing to the premium plan. You'll receive a confirmation email shortly." FontSize="14" HorizontalOptions="Center" HorizontalTextAlignment="Center"/>

<Button Text="Go to Home" BackgroundColor="Teal" TextColor="White" CornerRadius="10" Clicked="Button\_Clicked"/>

</StackLayout>

</ContentPage.Content>

</ContentPage>

# C# Code

## Signup

using System;

using System.Linq;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Signup : ContentPage

{

public Signup()

{

InitializeComponent();

NavigationPage.SetHasNavigationBar(this, false);

}

private void ShowError(string message)

{

errorLabel.Text = message;

errorLabel.IsVisible = true;

}

private void Btnclick\_Clicked(object sender, EventArgs e)

{

string firstName = firstNameEntry.Text;

string lastName = lastNameEntry.Text;

string email = emailerror.Text;

string password = pwerror.Text;

string confirmpassword = Confirmerror.Text;

errorLabel.IsVisible = false;

if (string.IsNullOrWhiteSpace(firstName))

{

ShowError("First name is required.");

return;

}

if (string.IsNullOrWhiteSpace(lastName))

{

ShowError("Last name is required.");

return;

}

if (string.IsNullOrEmpty(email) || !email.Contains("@"))

{

ShowError("Enter a valid email.");

return;

}

if (string.IsNullOrEmpty(password) || password.Length < 10 || !password.Any(char.IsDigit))

{

ShowError("Password must be at least 10 characters and include a number.");

return;

}

if (confirmpassword != password)

{

ShowError("Passwords do not match.");

return;

}

// Clear error if everything is valid

errorLabel.IsVisible = false;

// Proceed with registration logic

DisplayAlert("Success", "Account created successfully!", "OK");

}

}

}

## Login

using System;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Login : ContentPage

{

public Login()

{

InitializeComponent();

NavigationPage.SetHasNavigationBar(this, false);

}

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

{

string email = emailEntry.Text;

string password = passwordEntry.Text;

errorLabel.IsVisible = false;

if (string.IsNullOrEmpty(email) || !email.Contains("@"))

{

ShowError("Enter a valid email.");

return;

}

if (string.IsNullOrEmpty(password))

{

ShowError("Password is required.");

return;

}

Navigation.InsertPageBefore(new Home(), this);

await Navigation.PopAsync();

}

private void ShowError(string message)

{

errorLabel.Text = message;

errorLabel.IsVisible = true;

}

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

{

// Navigate to signup page

Navigation.PushAsync(new Signup());

}

}

}

## Home

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Home : ContentPage

{

public Home()

{

InitializeComponent();

}

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

{

Navigation.InsertPageBefore(new AddIncomeExpenses(), this);

await Navigation.PopAsync();

}

}

}

## Menu (Hamburger)

using Xamarin.Forms;

namespace Project

{

public class Mainpage : FlyoutPage

{

public Mainpage()

{

var menuPage = new ContentPage

{

Title = "Menu",

BackgroundColor = Color.LightGray,

Padding = new Thickness(10),

Content = new StackLayout

{

Children =

{

new Label

{

Text = "Menu",

FontSize = 24,

FontAttributes = FontAttributes.Bold,

Margin = new Thickness(0, 20, 0, 20)

},

new Button

{

Text = "Add Record",

Command = new Command(() => NavigateTo(new AddIncomeExpenses()))

},

new Button

{

Text = "Go Premium",

Command = new Command(() => NavigateTo(new PlanSelection()))

},

new Button

{

Text = "About us",

Command = new Command(() => NavigateTo(new AboutUs()))

},

new Button

{

Text = "Contact us",

Command = new Command(() => NavigateTo(new Contactpage()))

},

new Button

{

Text = "Settings",

Command = new Command(() => DisplayAlert("Menu", "Settings clicked", "OK"))

},

new Button

{

Text = "Logout",

TextColor = Color.Red,

Command = new Command(async () =>

{

// Ask for confirmation before logging out

bool confirm = await Application.Current.MainPage.DisplayAlert(

"Logout",

"Are you sure you want to logout?",

"Yes",

"No"

);

if (confirm)

{

Application.Current.MainPage = new NavigationPage(new Login());

}

})

}

}

}

};

var dashboardPage = new Home();

Flyout = menuPage;

Detail = new NavigationPage(dashboardPage);

}

private void NavigateTo(Page page)

{

IsPresented = false; // Close the flyout

if (Detail is NavigationPage navPage)

{

navPage.Navigation.PushAsync(page);

}

else

{

Detail = new NavigationPage(page);

}

}

}

}

## Add Income & Expenses

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class AddIncomeExpenses : ContentPage

{

string currentEntry = "";

string operatorSymbol = "";

double firstNumber = 0;

bool isNewEntry = true;

private string transactionType;

public AddIncomeExpenses()

{

InitializeComponent();

}

void OnNumberClicked(object sender, EventArgs e)

{

var button = (Button)sender;

string digit = button.Text;

if (isNewEntry)

{

currentEntry = digit;

isNewEntry = false;

}

else

{

currentEntry += digit;

}

AmountDisplay.Text = currentEntry;

}

void OnOperatorClicked(object sender, EventArgs e)

{

var button = (Button)sender;

operatorSymbol = button.Text;

if (double.TryParse(currentEntry, out firstNumber))

{

isNewEntry = true;

}

}

void OnEqualsClicked(object sender, EventArgs e)

{

if (!double.TryParse(currentEntry, out double secondNumber)) return;

double result = 0;

switch (operatorSymbol)

{

case "+": result = firstNumber + secondNumber; break;

case "-": result = firstNumber - secondNumber; break;

case "×": result = firstNumber \* secondNumber; break;

case "÷":

result = secondNumber != 0 ? firstNumber / secondNumber : 0;

break;

}

AmountDisplay.Text = result.ToString("0.##");

currentEntry = result.ToString();

isNewEntry = true;

}

async void OnCancelClicked(object sender, EventArgs e)

{

Navigation.InsertPageBefore(new Home(), this);

await Navigation.PopAsync();

}

void OnClearClicked(object sender, EventArgs e)

{

currentEntry = "";

firstNumber = 0;

operatorSymbol = "";

isNewEntry = true;

AmountDisplay.Text = "0";

}

void OnSaveClicked(object sender, EventArgs e)

{

string amount = AmountDisplay.Text;

DateTime selectedDateTime = DateInput.Date + TimeInput.Time;

DisplayAlert("Saved", $"Amount: {amount}\nDate: {selectedDateTime}", "OK");

}

void OnTransactionTypeChanged(object sender, CheckedChangedEventArgs e)

{

var radio = (RadioButton)sender;

if (e.Value)

{

transactionType = radio.Value.ToString();

if (transactionType == "Income")

{

incomeButton.TextColor = Color.White;

incomeButton.BackgroundColor = Color.Teal;

expenseButton.TextColor = Color.Gray;

expenseButton.BackgroundColor = Color.White;

}

if (transactionType == "Expense")

{

incomeButton.TextColor = Color.Gray;

incomeButton.BackgroundColor = Color.White;

expenseButton.TextColor = Color.White;

expenseButton.BackgroundColor = Color.Teal;

}

}

}

}

}

## About Us

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

using Xamarin.Essentials;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class AboutUs : ContentPage

{

public AboutUs()

{

InitializeComponent();

}

}

}

## Contact Us

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Contactpage : ContentPage

{

public Contactpage()

{

InitializeComponent();

}

}

}

## Plan Selection

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class PlanSelection : ContentPage

{

public PlanSelection()

{

InitializeComponent();

}

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

{

await Navigation.PushAsync(new Payment());

}

}

}

## Payment

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class Payment : ContentPage

{

public Payment()

{

InitializeComponent();

}

private async void ConfirmedClicked(object sender, EventArgs e)

{

//validation for card number

string cardNum = CardNum.Text;

if (cardNum == null ||cardNum.Length < 16)

{

await DisplayAlert("Invalid", "Card number should be 16 numbers long.", "OK");

return;

}

//validation for cvv

string cvv = CVV.Text;

if (cvv == null || cvv.Length < 3)

{

await DisplayAlert("Invalid", "CVV should be 3 numbers long.", "OK");

return;

}

//validation for expiry date

string expiryDate = ExpiryDate.Text;

if(expiryDate == null || expiryDate[2] != '/')

{

await DisplayAlert("Invalid", "Use MM/YY format", "OK");

return;

}

var part = expiryDate.Split('/');

if (int.TryParse(part[0], out int mm) && int.TryParse(part[1], out int yy))

{

if (mm < 1 || mm > 12)

{

await DisplayAlert("Invalid", "Month should be between 01 and 12.", "OK");

return;

}

var fullDate = new DateTime(2000 + yy, mm, 1).AddMonths(1).AddDays(-1);

if (fullDate < DateTime.Now)

{

await DisplayAlert("Invalid", "Card has expired.", "OK");

return;

}

Navigation.InsertPageBefore(new SuccessfulPayment(), this);

await Navigation.PopAsync();

}

else

{

await DisplayAlert("Invalid", "Use numbers only in MM/YY format.", "OK");

}

}

}

}

## Successful Payment

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Xamarin.Forms;

using Xamarin.Forms.Xaml;

namespace Project

{

[XamlCompilation(XamlCompilationOptions.Compile)]

public partial class SuccessfulPayment : ContentPage

{

public SuccessfulPayment()

{

InitializeComponent();

}

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

{

Navigation.InsertPageBefore(new Mainpage(), this);

await Navigation.PopAsync();

}

}

}

# Challenges Faced:

The team had to face several technical and design issues during development:

1. **Navigation Problem**:   
   Proper page to page navigation was necessary particularly after signing in, in the payment flow, and there was a careful control over the stack, navigation maintaining.
2. **Input Validation**:   
   The correct input validation should be employed with respect to email address, password complexity, and billing details (such as a card number and expiry date) to avoid human mistakes.
3. **Hamburger Menu Integration:**The Flyout Page (hamburger menu) was implemented without any change in the general layout of the main content elements, which meant that the page structure had to be adjusted.
4. **Social Login Placeholders:**   
   Despite visual solutions to social login (Google, Facebook, Twitter) being added, making the actual authentication backends functional continues to be a task that will be done in the future.
5. **Code Sharing and Errors:**We used Git for version control; however, we faced issues when both of us made huge changes at the same time. We temporarily fixed this issue by sharing the project folder. But we will be solving the Git error as it will greatly enhance our productivity and efficiency.
6. **Emulator Errors:**

Both of us faced issue regarding the emulator view, most probably because of lower hardware functionality. However, we solved this problem by connecting our android device to the project and using it as an emulator.

# Future Work:

To work on the enhancement and development of the Budget Tracker App, the following future indication is outlined:

1. **Backend:**Integrate the app with a real-time database (e.g., Firebase or Azure) on which you can securely store data on the user.
2. **Authentication System**:   
   Provide complete functionality with social logins in addition to safe email/pass authentication.
3. **Analytics Dashboard:**   
   Provide additional inquiry by offering interactive graphics and information visualizations that provide the customers with better information about their spending habits.
4. **Push Notifications:**Add notification reminders of payments that a user should complete, overspend, or when planning to reach a new budget.
5. **Bug Fixes and Testing:**Conduct thorough usability tests, as well as bug fixes to promote comfortable user experience across devices.
6. **Smoother Page Flow:**   
   Refactoring the navigation structure to assure consistent and seamless transitions as well as to make the project lightweight. This includes optimizing back-stack behavior and reducing navigation-related confusion in the interface.

# Live Demo

This is a live demo video of our project BrightBudget.



**Note:** Kindly double click it to play the video.

# References

1. *CleanPng*. (n.d.). <https://www.cleanpng.com/>
2. AshProgHelp - Programming Help. (2021, June 15). *Flyout Page/Master Details page in Xamarin Forms / .NET MAUI - EP:7* [Video]. YouTube. <https://www.youtube.com/watch?v=f21I341_ASI>
3. AshProgHelp - Programming Help. (2021b, June 21). *MVVM & Data binding in Xamarin Forms / .NET MAUI - EP:9* [Video]. YouTube. <https://www.youtube.com/watch?v=VzfxVYsaYXk>