Course Name: Visual Programming
Course Code: 201NMCA306

AND

Course Name: Visual Programming Lab
Course **Code:** 201NMCA334

Topic: Currency convertor

Name: Alwin Varghese

Class: S6 INTMCA

Roll no: 12

Problem Statement

This mini-project focuses on building a user-friendly Currency Converter using Visual Basic .NET. It introduces students to the fundamental concepts of graphical user interface (GUI) design, event-driven programming, and data processing techniques. The main objective of this project is to enable students to develop an interactive application that allows users to perform currency conversions based on predefined exchange rates. The program utilizes a dictionary to store exchange rates with USD as the base currency. Users can select their desired source and target currencies from a dropdown list, input the amount to be converted, and receive an instant conversion result. Through this project, students will gain proficiency in handling user inputs, managing data structures efficiently, implementing real-time calculations, and applying exception handling to enhance the reliability of the application. Additionally, this project helps students understand the significance of user experience (UX) design, as they will be required to create an intuitive interface that simplifies currency conversion for users. By the end of this miniproject, students will have a solid foundation in Visual Basic .NET programming, allowing them to build and expand upon more complex applications in the future.

Logic Explanation

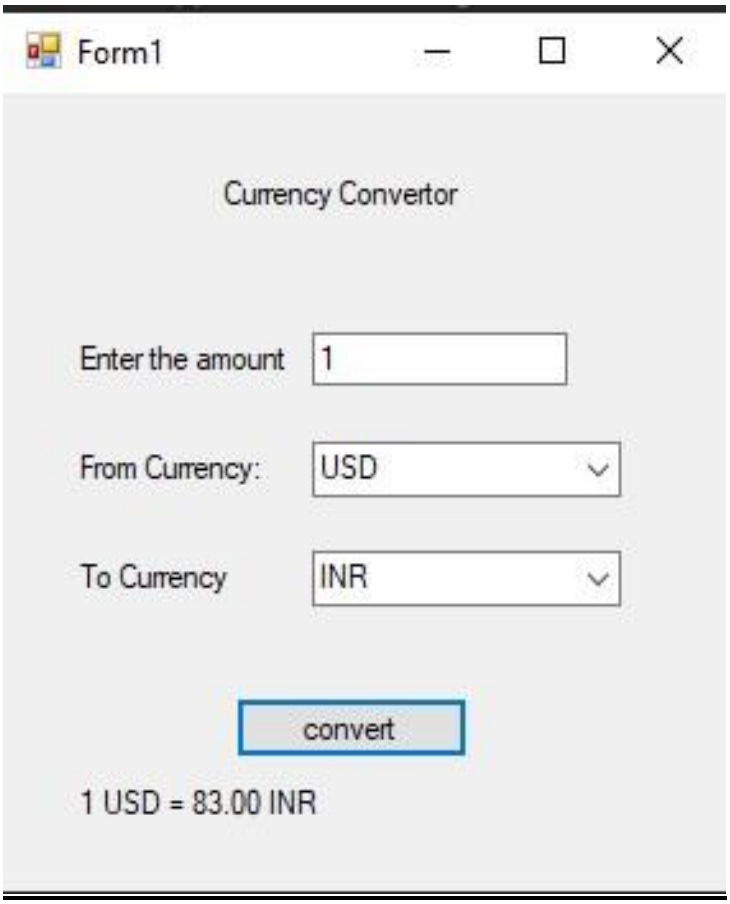
The Currency Converter application is designed using Visual Basic .NET, utilizing a dictionary to store exchange rates with USD as the base currency. Users select the source and target currencies from combo boxes and enter the amount to be converted. The application first converts the entered amount to USD and then to the target currency using predefined exchange rates. The result is displayed dynamically in a label, ensuring real-time updates. Error handling mechanisms, such as input validation and exception handling, are implemented to ensure users enter valid numerical values and prevent application crashes. The use of structured data handling techniques allows for efficient currency conversions, while an intuitive interface ensures a smooth user experience. By leveraging event-driven programming, the converter dynamically updates results based on user selections, reinforcing fundamental programming concepts.

Code Implementation

```
Public Class Form1
    ' Dictionary to store exchange rates (Base: USD)
    Dim exchangeRates As New Dictionary(Of String, Double) From {
        {"USD", 1.0},
        {"EUR", 0.92},
        {"GBP", 0.78},
        {"INR", 83.0},
        {"JPY", 150.25},
        {"AUD", 1.52},
        {"CAD", 1.35}
    }

    ' Form Load - Populate ComboBoxes
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        ' Add currency codes to both combo boxes          For Each currency As
String In exchangeRates.Keys          cmbFromCurrency.Items.Add(currency)
cmbToCurrency.Items.Add(currency)
        Next

        '  Set  default  selection                  cmbFromCurrency.SelectedIndex  =  0
cmbToCurrency.SelectedIndex = 1
    End Sub

    ' Conversion Logic
    Private Sub btnConvert_Click(sender As Object, e As EventArgs)
Handles btnConvert.Click
        Try
            ' Validate input
            Dim amount As Double
            If Not Double.TryParse(txtAmount.Text, amount) OrElse amount < 0 Then
                MessageBox.Show("Please enter a valid amount!",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
                Exit Sub
            End If

            ' Get selected currencies         Dim fromCurrency
As String = cmbFromCurrency.SelectedItem.ToString()
Dim          toCurrency          As          String          =
cmbToCurrency.SelectedItem.ToString()
```

```
            ' Convert amount to USD first, then to target currency
            Dim amountInUSD As Double = amount / exchangeRates(fromCurrency)
            Dim convertedAmount As Double = amountInUSD * exchangeRates(toCurrency)

            ' Display result
            lblResult.Text = $"{amount} {fromCurrency} =
{convertedAmount.ToString("F2")} {toCurrency}"
        Catch ex As Exception
            MessageBox.Show("An error occurred: " & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub


End Class
```

Output



Result: The question have been successfully executed and the output is verified