```vb

'1.] The interest rate percentage (%) is not hardcoded, thus if needed to be chancged it can be done so by re-writting the text of the textbox (before running the App).
'2.] Form1 is alocated as the "Split screen" of the App // form2 is the "input/welcome/main" page // form3 is the "result/restart" page



Public Class Form2

    'Declaring variables globaly so that all subs can access it

    Dim decAnnualSalary As Decimal    'This is the annual salary
    Dim decLoanAmount As Decimal      'This is the Loan amount
    Dim intMaxTermYrs As Integer      'This is the maximum term of the loan in years
    Dim intMaxTermMonths As Integer   'This is the maximum term of the loan in months
    Dim decInterestrate As Decimal    'This is the interest rate of the loan during the term period
    Dim intThreshold As Integer       'This is the repayment Threshold of the loan
    Dim decRepPercent As Decimal      'This is the Rpayment percentage the user will have to pay
    Dim decRefund As Decimal          'The refund the user will get (if needed)
    '=========================================================================================================================================================
    '# All the actions taking place after clicking the "Calculate" button
    '=========================================================================================================================================================


    Private Sub Calculatebtn_Click(sender As Object, e As EventArgs) Handles Calculatebtn.Click


        '==============================================================================
        '# Validating all the input text Boxes to make sure we have the wanted inputs
        '==============================================================================

        '=> Ceate individual Lock booleans for the validations to ensure Form3 wont appear and the error messages will show , if there is an input error
        '-----------------------------------------------------------------------------------------------------------------------------------------

        Dim Lock1 As Boolean
        Dim Lock2 As Boolean
        Dim Lock3 As Boolean
        Dim Lock4 As Boolean
        Dim Lock5 As Boolean
        Dim Lock6 As Boolean


        'Validating term for numeric values and non-negative values and non-blanck boxes
        If (IsNumeric(txtboxMaxTerms.Text) AndAlso (txtboxMaxTerms.Text > 0) AndAlso IsNothing(txtboxMaxTerms.Text) = False) Then
            'Assign the Term value if it is numeric and grater than 0
            intMaxTermYrs = Val(txtboxMaxTerms.Text)
            intMaxTermMonths = intMaxTermYrs * 12

            Lock1 = True
        Else
            'Show a message box to the user with the appropriate message
            MsgBox("The *Term* value is invalid", MsgBoxStyle.Exclamation, "Error")

            Lock1 = False

        End If


        'Validating loan amount for numeric value and non-negative values and non-blank boxes
        If (IsNumeric(txtboxLoanAmount.Text) AndAlso (txtboxLoanAmount.Text > 0) AndAlso IsNothing(txtboxLoanAmount.Text) = False) Then
            'Assign the Loan value if it is numeric and grater than 0
            decLoanAmount = Val(txtboxLoanAmount.Text)

            Lock2 = True

        Else
            'Show a message box to the user with the appropriate message
            MsgBox("The *Loan Amount* value is invalid", MsgBoxStyle.Exclamation, "Error")

            Lock2 = False

        End If


        'Validating Annual salary amount for numeric value and non-negative values and non-blank boxes'
        If (IsNumeric(txtboxSalary.Text) AndAlso (txtboxSalary.Text > 0) AndAlso IsNothing(txtboxSalary.Text) = False) Then
            'Assign the salary value if it is numeric and grater than 0 '
            decAnnualSalary = Val(txtboxSalary.Text)

            Lock3 = True

        Else
            'Show a message box to the user with the appropriate message'
            MsgBox("The *Salary* value is invalid", MsgBoxStyle.Exclamation, "Error")

            Lock3 = False

        End If


        'Validating interest rate for numeric values and non-negative values and non-blanck boxes'
        If (IsNumeric(txtboxInterestRate.Text) AndAlso (txtboxInterestRate.Text > 0) AndAlso IsNothing(txtboxInterestRate.Text) = False) Then
            'Assign the Term value if it is numeric and grater than 0 '
            decInterestrate = Val(txtboxInterestRate.Text)

            Lock4 = True

        Else
            'Show a message box to the user with the appropriate message'
            MsgBox("The *Interest rate* value is invalid", MsgBoxStyle.Exclamation, "Error")

            Lock4 = False

        End If

        'Validating Repayment Threshold for numeric values and non-negative values and non-blanck boxes'
        If (IsNumeric(txtboxThreshold.Text) AndAlso (txtboxThreshold.Text > 0) AndAlso IsNothing(txtboxThreshold.Text) = False) Then
            'Assign the Term value if it is numeric and grater than 0 '
            intThreshold = Val(txtboxThreshold.Text)

            Lock5 = True

        Else
            'Show a message box to the user with the appropriate message'
            MsgBox("The *Repayment Threshold* value is invalid", MsgBoxStyle.Exclamation, "Error")

            Lock5 = False

        End If

        'Validating Repayment Threshold for numeric values and non-negative values and non-blanck boxes'
        If (IsNumeric(txtboxThreshold.Text) AndAlso (txtboxThreshold.Text > 0) AndAlso IsNothing(txtboxThreshold.Text) = False) Then
            'Assign the Term value if it is numeric and grater than 0 '
            decRepPercent = Val(txtboxRepaymentPrcent.Text)
            Lock6 = True

        Else
            'Show a message box to the user with the appropriate message'
            MsgBox("The *Repayment Percentage* value is invalid",
              MsgBoxStyle.Exclamation, "Error")

            Lock6 = False

        End If

        If (Lock1 = True And Lock2 = True And Lock3 = True And Lock4 = True And Lock5 = True And Lock6 = True) Then

            Form3.Show()
            Me.Hide()

        Else
            'We erase all the previous results (if this was the second+ attempt) in order to not effect further calculations
            Form3.txtboxLoanConfirm.Clear()
            Form3.txtboxAnnualRepayment.Clear()
            Form3.txtboxMonthlyRepayment.Clear()
            Form3.txtboxMonths.Clear()
            Form3.txtboxYears.Clear()
            Form3.txtboxTotalinterestPayed.Clear()
            Form3.txtboxTotalPayment.Clear()
            Form3.lstBoxInfo.Items.Clear()

            'Program skips ahead to the location of "exitpoint", which is at the "End Sub", thus ending the "Calculation Button" actions.
            GoTo exitpoint1

        End If


'================================================================================
        '# Calculation Section

'================================================================================

        'Local variable decleration for our calculations

        Dim decBalanceNow As Decimal
        Dim decClosingBalance As Decimal
        Dim decAnnualRepayment As Decimal
        Dim decMonthlyRepayment As Decimal
        Dim decAnnualInterest As Decimal
        Dim decMonthlyInterest As Decimal
        Dim decTotalInterest As Decimal

        '# IMPORTANT : decAnnualInterest/decMonthlyInterest => Refer to the ammount of money(£) of that interest ///  decInterestrate => refers to the percentage(%) of interest . NOT money.

        '---------------------------------------------------------------------
        'Find and dislay annual and monlthly repayment  ( step 1 )
```

```vbnet
        '--------------------------------------------------------------------

        If (decAnnualSalary   intThreshold) Then
            'MsgBox(" The *Annual Salary* can't be lower than the *Repayment Threshold* ")

            lblCalculationConfirm.Text = " The *Annual Salary* can't be lower than the *Repayment Threshold* "

            'We ensure that in the case of an insufficient Salary , form3 won't be shown (claculations will be skipped because of the "Goto" statement)
            Me.Show()
            Form3.Hide()

            GoTo exitpoint1

            'We clarify that if the above "if" statement is correct, the program will skip all the below calculations and go straight to "end sub".
        '------------------------------------------------------------------------------------------------------------------------------------
        Else

            lblCalculationConfirm.Text = " "

            decAnnualRepayment = (decAnnualSalary - intThreshold) * (decRepPercent / 100)
            decMonthlyRepayment = decAnnualRepayment / 12

            'Rounding up the results
            '--------------------------------
            decMonthlyRepayment = Math.Round(decMonthlyRepayment, 2)
            decAnnualRepayment = Math.Round(decAnnualRepayment, 2)


            'Displaying the results
            '-----------------------

            Form3.txtboxAnnualRepayment.Text = decAnnualRepayment
            Form3.txtboxMonthlyRepayment.Text = decMonthlyRepayment



        End If
        '-----------------------------------------------------------------------------------
        'Find and show the opening/closing  balance and interest for each month ( step 2 )
        '-----------------------------------------------------------------------------------

        'Make counter

        Dim intCounter As Integer
        intCounter = 0

        Dim BalanceRepaid As Boolean
        BalanceRepaid = False

        Dim decRepaymentMade As Decimal
        Dim decBalance As Decimal
        decBalance = decLoanAmount

        decTotalInterest = 0
        decRepaymentMade = 0


        'FORM 3 LIST BO
        '-------------------------------

        Form3.lstBoxInfo.Items.Clear()

        ' The "Try" statement protects the programe from errors. The statement catches the errors and improuves the user experience.
        Try

            Do
                ' Inserting counter
                intCounter = intCounter + 1

                'Calculations part which will be desplayed in the listbox.
                decAnnualInterest = decBalance * (decInterestrate / 100)
                decMonthlyInterest = decAnnualInterest / 12
                decMonthlyInterest = Math.Round(decMonthlyInterest, 2)

                decBalanceNow = decBalance + decMonthlyInterest
                decClosingBalance = decBalanceNow - decMonthlyRepayment

                decBalance = decClosingBalance
                decRepaymentMade = decRepaymentMade + decMonthlyRepayment
                decTotalInterest = decTotalInterest + decMonthlyInterest

                'Ensuring there are no extra charges at the end of the repayment period, and the closing balance does not reach negative value.
                If (decClosingBalance   0) Then

                    decRepaymentMade = decRepaymentMade - (decMonthlyRepayment - decClosingBalance)
                    decClosingBalance = 0
                    decRefund = decMonthlyRepayment - decClosingBalance
                    'Displaying to the user the amount of refund he got / the refund will be the amount of money we removed from the total Payment made as shown above.
                    Form3.lblRefund.Text = "You got a refund of : £"
                    Form3.lblRefundPrice.Text = decRefund
                End If

```

```vb

                'LIST BO  results from FORM 3
                '--------------------------------------------------------

                Form3.lstBoxInfo.Items.Add("Repayment for month: " intCounter)
                Form3.lstBoxInfo.Items.Add("Closing balance is : £" decClosingBalance)
                Form3.lstBoxInfo.Items.Add("Interest payed is : £" decMonthlyInterest)
                Form3.lstBoxInfo.Items.Add("")  ' Leaving an empty space between months to make it easier for the user to read.



                '================================================================================================
                'Showing the core results of the application  (Step 3)
                '================================================================================================

                If (intCounter = intMaxTermMonths Or decClosingBalance  = 0) Then

                    BalanceRepaid = True

                    'TE T BO ES FROM FORM 3 // Total Repayment and Interest payed through the period
                    '---------------------------------------------------------------------------------

                    Form3.txtboxTotalPayment.Text = decRepaymentMade
                    Form3.txtboxTotalinterestPayed.Text = decTotalInterest

                End If

            Loop Until intCounter = intMaxTermMonths Or BalanceRepaid = True

        Catch ex As Exception

        End Try

        Dim decCounterYrs As Decimal 'creating a variable to show the total years taken to repay the loan
        Form3.txtboxLoanConfirm.Text = decLoanAmount
        Form3.txtboxMonths.Text = intCounter
        decCounterYrs = intCounter / 12
        decCounterYrs = Math.Round(decCounterYrs, 1) 'round-up the yearly counter up to 1 decimal number since we reffer to years so anything more than that is insignifficant
        Form3.txtboxYears.Text = decCounterYrs

exitpoint1: End Sub
```

```vb

'=======================================================================================================================================
    'Actions after clicking the "Exit" and "=>" ( Arrow ) buttons
'=======================================================================================================================================


    'This is the " uit"  button.When clicked , form 2 closes and there should be no program forms shown.
'---------------------------------------------------------------------------------------------------

    Private Sub  uitbtn_Click(sender As Object, e As EventArgs) Handles uitbtn.Click

        Dim Response As DialogResult
        Response = MessageBox.Show("Would you like to exit  ", "",
        MessageBoxButtons.YesNo, MessageBoxIcon. uestion)

        If Response = DialogResult.Yes Then

            Application.Exit()

        End If

    End Sub
'===================================================================================================================================
    'The use of "Arrow" button is to give freedome to the user to view both the inputs and the outputs after a successful calculation process
'===================================================================================================================================
    Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click

        Hide()
        Form3.Show()

    End Sub

End Class
```