```
'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
4
5
6
7
  Public Class Form2
8
9
      'Declaring variables globaly so that all subs can access it
10
     Dim decAnnualSalary As Decimal
11
                                'This is the annual salary
     Dim decLoanAmount As Decimal
12
                                'This is the Loan amount
                                'This is the maximum term of the loan in >
     Dim intMaxTermYrs As Integer
13
        years
                                'This is the maximum term of the loan in ₹
14
     Dim intMaxTermMonths As Integer
        months
15
     Dim decInterestrate As Decimal
                                'This is the interest rate of the loan >
       during the term period
                                'This is the repayment Threshold of the >
     Dim intThreshold As Integer
16
       1<sub>oan</sub>
17
     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
18
       needed)
19
   ______
20
      '# All the actions taking place after clicking the "Calculate" button
21
   ______
22
23
      Private Sub Calculatebtn_Click(sender As Object, e As EventArgs) Handles
24
       Calculatebtn.Click
25
26
   '-------
27
         '# Validating all the input text Boxes to make sure we have the wanted ➤
           inputs
28
   29
30
         '=> Ceate individual Lock booleans for the validations to ensure Form3 →
           wont appear and the error messages will show , if there is an input ₹
           error
31
            _____
32
33
         Dim Lock1 As Boolean
```

```
C:\Users\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb
 34
             Dim Lock2 As Boolean
             Dim Lock3 As Boolean
 35
             Dim Lock4 As Boolean
 36
 37
             Dim Lock5 As Boolean
             Dim Lock6 As Boolean
 38
 39
 40
             'Validating term for numeric values and non-negative values and non-
 41
               blanck boxes
 42
             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
 43
 44
                 intMaxTermYrs = Val(txtboxMaxTerms.Text)
                 intMaxTermMonths = intMaxTermYrs * 12
 45
 46
                 Lock1 = True
 47
 48
             Else
 49
                  'Show a message box to the user with the appropriate message
 50
                 MsgBox("The *Term* value is invalid", MsgBoxStyle.Exclamation,
                   "Error")
 51
                 Lock1 = False
 52
 53
 54
             End If
 55
 56
 57
             'Validating loan amount for numeric value and non-negative values and >
               non-blank boxes
 58
             If (IsNumeric(txtboxLoanAmount.Text) AndAlso (txtboxLoanAmount.Text > >
               0) AndAlso IsNothing(txtboxLoanAmount.Text) = False) Then
 59
                  'Assign the Loan value if it is numeric and grater than 0
 60
                 decLoanAmount = Val(txtboxLoanAmount.Text)
 61
 62
                 Lock2 = True
 63
             Else
 64
                 \ensuremath{^{\mathsf{'}}} \mathsf{Show} a message box to the user with the appropriate message
 65
 66
                 MsgBox("The *Loan Amount* value is invalid",
                   MsgBoxStyle.Exclamation, "Error")
 67
                 Lock2 = False
 68
 69
             End If
 70
 71
 72
             'Validating Annual salary amount for numeric value and non-negative
 73
               values and non-blank boxes'
             If (IsNumeric(txtboxSalary.Text) AndAlso (txtboxSalary.Text > 0)
 74
               AndAlso IsNothing(txtboxSalary.Text) = False) Then
 75
                  'Assign the salary value if it is numeric and grater than 0 '
                 decAnnualSalary = Val(txtboxSalary.Text)
 76
 77
 78
                 Lock3 = True
 79
             Else
 80
 81
                  'Show a message box to the user with the appropriate message'
```

```
C:\Users\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb
 82
                 MsgBox("The *Salary* value is invalid", MsgBoxStyle.Exclamation,
                   "Error")
 83
 84
                 Lock3 = False
 85
 86
             End If
 87
 88
 89
             'Validating interest rate for numeric values and non-negative values
               and non-blanck boxes'
             If (IsNumeric(txtboxInterestRate.Text) AndAlso
 90
                                                                                      P
               (txtboxInterestRate.Text > 0) AndAlso IsNothing
                                                                                      P
               (txtboxInterestRate.Text) = False) Then
                 'Assign the Term value if it is numeric and grater than 0 '
 91
 92
                 decInterestrate = Val(txtboxInterestRate.Text)
 93
 94
                 Lock4 = True
 95
 96
             Else
 97
                 'Show a message box to the user with the appropriate message'
                 MsgBox("The *Interest rate* value is invalid",
 98
                   MsgBoxStyle.Exclamation, "Error")
 99
100
                 Lock4 = False
101
             End If
102
103
             'Validating Repayment Threshold for numeric values and non-negative
104
               values and non-blanck boxes'
             If (IsNumeric(txtboxThreshold.Text) AndAlso (txtboxThreshold.Text > 0) →
105
                AndAlso IsNothing(txtboxThreshold.Text) = False) Then
106
                 'Assign the Term value if it is numeric and grater than 0 '
107
                 intThreshold = Val(txtboxThreshold.Text)
108
                 Lock5 = True
109
110
             Else
111
                 'Show a message box to the user with the appropriate message'
112
113
                 MsgBox("The *Repayment Threshold* value is invalid",
                   MsgBoxStyle.Exclamation, "Error")
114
115
                 Lock5 = False
116
             End If
117
118
             'Validating Repayment Threshold for numeric values and non-negative
119
               values and non-blanck boxes'
120
             If (IsNumeric(txtboxThreshold.Text) AndAlso (txtboxThreshold.Text > 0) →
                AndAlso IsNothing(txtboxThreshold.Text) = False) Then
121
                 'Assign the Term value if it is numeric and grater than 0 '
122
                 decRepPercent = Val(txtboxRepaymentPrcent.Text)
                 Lock6 = True
123
124
125
             Else
126
                 'Show a message box to the user with the appropriate message'
127
                 MsgBox("The *Repayment Percentage* value is invalid",
```

```
C:\Users\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb
                 MsgBoxStyle.Exclamation, "Error")
128
129
               Lock6 = False
130
           End If
131
132
133
           If (Lock1 = True And Lock2 = True And Lock3 = True And Lock4 = True
             And Lock5 = True And Lock6 = True) Then
134
135
               Form3.Show()
136
               Me.Hide()
137
138
           Else
               'We erase all the previous results (if this was the second+
139
                 attempt) in order to not effect further calculations
140
               Form3.txtboxLoanConfirm.Clear()
141
               Form3.txtboxAnnualRepayment.Clear()
142
               Form3.txtboxMonthlyRepayment.Clear()
143
               Form3.txtboxMonths.Clear()
144
               Form3.txtboxYears.Clear()
145
               Form3.txtboxTotalinterestPayed.Clear()
               Form3.txtboxTotalPayment.Clear()
146
147
               Form3.lstBoxInfo.Items.Clear()
148
149
               'Program skips ahead to the location of "exitpoint", which is at
                the "End Sub", thus ending the "Calculation Button" actions.
150
               GoTo exitpoint1
151
152
           End If
153
154
155
    '------
           '# Calculation Section
156
157
    '-------
158
           'Local variable decleration for our calculations
159
160
161
           Dim decBalanceNow As Decimal
162
           Dim decClosingBalance As Decimal
163
           Dim decAnnualRepayment As Decimal
           Dim decMonthlyRepayment As Decimal
164
           Dim decAnnualInterest As Decimal
165
166
           Dim decMonthlyInterest As Decimal
167
           Dim decTotalInterest As Decimal
168
           '# IMPORTANT : decAnnualInterest/decMonthlyInterest => Refer to the
169
             ammount of money(£) of that interest /// decInterestrate ⇒ refers ➤
             to the percentage(%) of interest . NOT money.
170
171
            '_____
172
           'Find and dislay annual and monlthly repayment ( step 1 )
```

C:\U	sers\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb	5
173	1	_
174		
175	If (decAnnualSalary intThreshold) Then	
176		P
_, _	Threshold* ")	
177	/// C5/1024 /	
178	<pre>lblCalculationConfirm.Text = " The *Annual Salary* can't be lower</pre>	P
170	than the *Repayment Threshold* "	
179	chair the Repayment Thi eshota	
180	'We ensure that in the case of an insufficient Salary , form3	P
100	won't be shown (claculations will be skipped because of the	P
	"Goto" statement)	
181	Me.Show()	
182	Form3.Hide()	
183	Totalis.httde()	
184	GoTo exitpoint1	
185	do to exicpoinci	
	'We clarify that if the above "if" statement is connect the	_
186	'We clarify that if the above "if" statement is correct, the program will skip all the below calculations and go straight to	7
	"end sub".	4
107	end Sub .	
187	1	2
		. 4
100	Else	
188	EISE	
189	lblCalculationConfirm Toyt - " "	
190	<pre>lblCalculationConfirm.Text = " "</pre>	
191	dA	_
192	<pre>decAnnualRepayment = (decAnnualSalary - intThreshold) *</pre>	4
400	(decRepPercent / 100)	
193	<pre>decMonthlyRepayment = decAnnualRepayment / 12</pre>	
194		
195	'Rounding up the results	
196	'	
197	<pre>decMonthlyRepayment = Math.Round(decMonthlyRepayment, 2)</pre>	
198	<pre>decAnnualRepayment = Math.Round(decAnnualRepayment, 2)</pre>	
199		
200		
201	'Displaying the results	
202	'	
203		
204	Form3.txtboxAnnualRepayment.Text = decAnnualRepayment	
205	Form3.txtboxMonthlyRepayment.Text = decMonthlyRepayment	
206		
207		
208		
209	End If	
210		P
	'	. P
244		
211	'Find and show the opening/closing balance and interest for each	7
24.0	month (step 2)	
212		P
	'	٠ ٦
242		
213	I Make counton	
214	'Make counter	

```
C:\Users\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb
215
216
             Dim intCounter As Integer
             intCounter = 0
217
218
             Dim BalanceRepaid As Boolean
219
220
             BalanceRepaid = False
221
             Dim decRepaymentMade As Decimal
222
             Dim decBalance As Decimal
223
224
             decBalance = decLoanAmount
225
             decTotalInterest = 0
226
227
             decRepaymentMade = 0
228
229
             'FORM 3 LIST BO
230
231
232
233
             Form3.lstBoxInfo.Items.Clear()
234
235
             ' The "Try" statement protects the programe from errors. The statement →
                catches the errors and improuves the user experience.
236
             Try
237
238
                 Do
                     ' Inserting counter
239
240
                     intCounter = intCounter + 1
241
242
                     'Calculations part which will be desplayed in the listbox.
                     decAnnualInterest = decBalance * (decInterestrate / 100)
243
244
                     decMonthlyInterest = decAnnualInterest / 12
245
                     decMonthlyInterest = Math.Round(decMonthlyInterest, 2)
246
247
                     decBalanceNow = decBalance + decMonthlyInterest
                     decClosingBalance = decBalanceNow - decMonthlyRepayment
248
249
250
                     decBalance = decClosingBalance
251
                     decRepaymentMade = decRepaymentMade + decMonthlyRepayment
252
                     decTotalInterest = decTotalInterest + decMonthlyInterest
253
                     'Ensuring there are no extra charges at the end of the
254
                       repayment period, and the closing balance does not reach
                       negative value.
255
                     If (decClosingBalance
                                             0) Then
256
                         decRepaymentMade = decRepaymentMade - (decMonthlyRepayment >
257
                          decClosingBalance)
258
                         decClosingBalance = 0
259
                         decRefund = decMonthlyRepayment - decClosingBalance
260
                         '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.
261
                         Form3.lblRefund.Text = "You got a refund of : £"
262
                         Form3.lblRefundPrice.Text = decRefund
```

End If

263264

```
C:\Users\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb
265
266
                    'LIST BO results from FORM 3
267
268
                    Form3.lstBoxInfo.Items.Add("Repayment for month: "
269
                      intCounter)
                    Form3.lstBoxInfo.Items.Add("Closing balance is : f"
270
                      decClosingBalance)
                    Form3.lstBoxInfo.Items.Add("Interest payed is : f"
271
                      decMonthlyInterest)
272
                    Form3.lstBoxInfo.Items.Add("") ' Leaving an empty space
                     between months to make it easier for the user to read.
273
274
275
276
                                                                                P
    ============
277
                    'Showing the core results of the application (Step 3)
278
     '------
    ============
279
280
                   If (intCounter = intMaxTermMonths Or decClosingBalance = 0) >
                     Then
281
282
                       BalanceRepaid = True
283
284
                        'TE T BO ES FROM FORM 3 // Total Repayment and Interest
                       payed through the period
285
286
                       Form3.txtboxTotalPayment.Text = decRepaymentMade
287
288
                       Form3.txtboxTotalinterestPayed.Text = decTotalInterest
289
290
                    End If
291
292
                Loop Until intCounter = intMaxTermMonths Or BalanceRepaid = True
293
294
            Catch ex As Exception
295
296
            End Try
297
            Dim decCounterYrs As Decimal 'creating a variable to show the total
298
              years taken to repay the loan
            Form3.txtboxLoanConfirm.Text = decLoanAmount
299
300
            Form3.txtboxMonths.Text = intCounter
301
            decCounterYrs = intCounter / 12
            decCounterYrs = Math.Round(decCounterYrs, 1) 'round-up the yearly
302
              counter up to 1 decimal number since we reffer to years so anything >
              more than that is insignifficant
303
            Form3.txtboxYears.Text = decCounterYrs
304
305 exitpoint1: End Sub
```

C:\L	Jsers\takis\Desktop\AppDev_Draft2\AppDev_Draft2\Form2.vb	8
306		
307		₽
	'======================================	P
	=======================================	
308	'Actions after clicking the "Exit" and "=>" (Arrow) buttons	
309		P
	'	P
	=======================================	
310		
311		
312	'This is the " uit" button.When clicked , form 2 closes and there should	P
	be no program forms shown.	
313		P
5_5		7
		•
314		
315	Private Sub uitbtn_Click(sender As Object, e As EventArgs) Handles	7
313	uitbtn.Click	
216	ultbtii.tlttk	
316	Dim Degranes As DislasDegrals	
317	Dim Response As DialogResult	
318	Response = MessageBox.Show("Would you like to exit ", "",	
319	MessageBoxButtons.YesNo, MessageBoxIcon. uestion)	
320		
321	<pre>If Response = DialogResult.Yes Then</pre>	
322		
323	Application.Exit()	
324		
325	End If	
326		
327	End Sub	
328		₽
	'======================================	₽
	=======================================	
329	'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	
330		P
	'======================================	P
	=======================================	
331	Private Sub Button3_Click(sender As Object, e As EventArgs) Handles	P
	Button3.Click	
332		
333	Hide()	
334	Form3.Show()	
335	1 3. m3 + 3 non ()	
336	End Sub	
337	EIN JUU	
	End Class	
338	End Class	