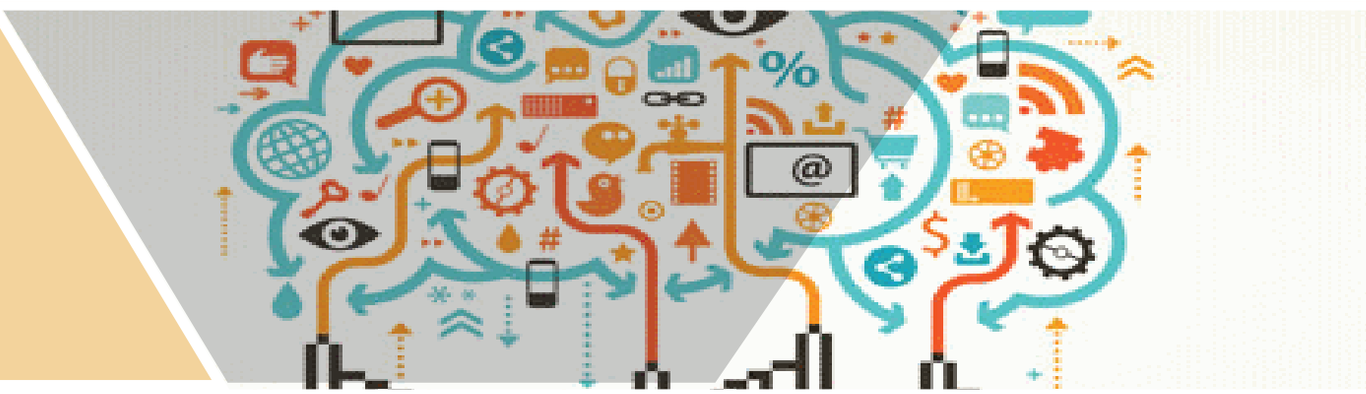
ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle and Graded as Category–I University by MHRD-UGC]

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION

M.Sc. [Computer Science]
341 24



LAB – . NET PROGRAMMING

II - Semester

ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle
and Graded as Category–I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

KARAIKUDI – 630 003

Directorate of Distance Education

M.Sc. [Computer Science]

II - Semester

341 24

LAB – .NET PROGRAMMING

Authors

Dr Syed Mohsin Saif Andrabi, *Assistant Professor, Islamic University of Science & Technology, Awantipora, Jammu and Kashmir*

Dr Mudasir M Kirmani, *Assistant Professor-cum-Junior Scientist, Sher-e-Kashmir University of Sciences and Technology of Kashmir*

Vikas® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.
E-28, Sector-8, Noida - 201301 (UP)
Phone: 0120-4078900 • Fax: 0120-4078999
Regd. Office: 7361, Ravindra Mansion, Ram Nagar, New Delhi 110 055
• Website: www.vikaspublishing.com • Email: helpline@vikaspublishing.com

Work Order No. AU/DDE/DE1-291/Preparation and Printing of Course Materials/2018 Dated 19.11.2018 Copies - 500

LAB – .NET PROGRAMMING

SYLLABI

BLOCK 1: .NET FRAMEWORK

1. Programs using variables, constants and data types
2. Programs using arrays and dynamic arrays
3. Program using control flow statement

BLOCK 2 : VISUAL BASIC.NET

4. Programs using functions and procedures, MDI forms, events
5. Programs using msgbox, inputbox, dialog boxes, working with multiple forms
6. Anchoring and docking controls, event handling, RichTextBoxes

BLOCK 3 : WINDOWS CONTROLS

7. Programs using windows common controls
8. Programs using menus, built in dialog boxes, Image lis, tree and list views
9. Programs using toolbars, statusbars, progressbars, tab controls, graphics and file handling

BLOCK 4 : ASP.NET

10. Writing ASP programs using HttpRequest and HttpResponse
11. Develop an application for ASP web controls, list controls, validation and rich controls, Data controls
12. Develop an application for HTML server controls, custom controls, logging and error handling

BLOCK 5 : ADO.NET

13. Database applications using ADO.NET
14. Accessing a database using SQL commands, Data binding controls, DataList

INTRODUCTION

NOTES

A programming language is an artificial language specifically designed to communicate instructions to a machine, particularly a computer. Technically, the programming languages are used to create programs that control the behaviour of a machine and/or to express algorithms precisely. The earliest programming languages predate the invention of the computer and were used to direct the behaviour of machines, such as Jacquard looms. Since then thousands of different programming languages have been created, mostly in the computer field, with many more being created every year. The majority programming languages illustrate computation in an imperative style, i.e., as a sequence of commands, while some languages, such as those that support functional programming or logic programming use alternative forms of description. The description of a programming language is usually split into the two components of syntax (form) and semantics (meaning). Visual Basic or VB is considered as a third generation or high level event driven programming language with Integrated Development Environment (IDE) specifically developed by Microsoft for its COM (Component Object Model) programming model and was first released in 1991. Visual Basic is uniquely designed to be relatively easy to learn and use. The term Visual Basic was derived from BASIC (Beginners' All-purpose Symbolic Instruction Code) and enables the Rapid Application Development (RAD) of Graphical User Interface (GUI) applications, access to databases using Data Access Objects, Remote Data Objects or ActiveX Data Objects and creation of ActiveX controls and objects. The final release was version 6 in 1998. Microsoft's extended support ended in March 2008 and the designated successor was Visual Basic .NET or VB.NET. Characteristically, VB.NET is an object-oriented computer programming language and is considered as an evolution of the classic Visual Basic (VB), implemented on the .NET Framework.

This lab manual, *.Net Programming*, contains several programs based on VB, ASP to provide the concept of programming. In addition, it will help students in coding and debugging their programs. The manual provides programs having step by step explanation that can help to write own programs very easily. These exercises shall be taken as the base reference during lab activities for students.

Software Requirements:

1. Microsoft Visual Studio(2012)
2. DotNet Framework above 4.0
3. MS SQL Server(2008)

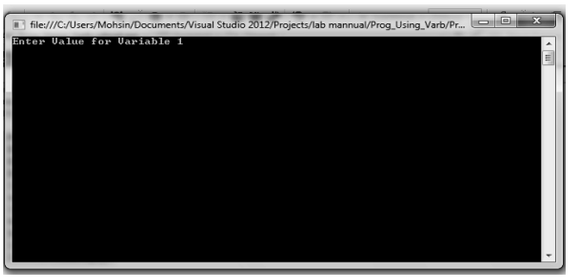
NOTES

1. Write a program to demonstrate the implementation of variables in VB.

Step 1: Compile `'ProgVariables.vb`

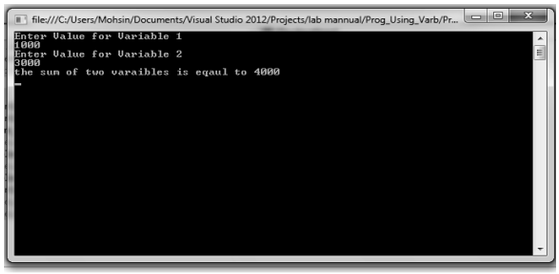
```
'ProgVariables.vb
'Programe to demenstrate the usage of Varibles using VB in DOT NET
Module ProgVariables
     Sub Main()
   Dim variable1 As Integer ' decleration of varaibles to handle user data
   Dim Variable2 As Integer
   Dim Sum_Variable As Integer
   Console.WriteLine("Enter Value for Variable 1") ' reading data for variable from user
     variable1 = Convert.ToInt32(Console.ReadLine())
     Console.WriteLine("Enter Value for Variable 2")
     Variable2 = Convert.ToInt32(Console.ReadLine())
     Sum_Variable = variable1 + Variable2 ' performing add operation
     Console.WriteLine("the sum of two varaibles is eqaul to {0}", +Sum_Variable)
     Console.ReadKey()
   End Sub
End Module
```

Step 2: Output: Build the program and run it, the following console will be displayed.



Step 3: Enter the value for variables as specified through console and the same is shown in figure below:

NOTES



2. Write a program to manipulate data types in VB.

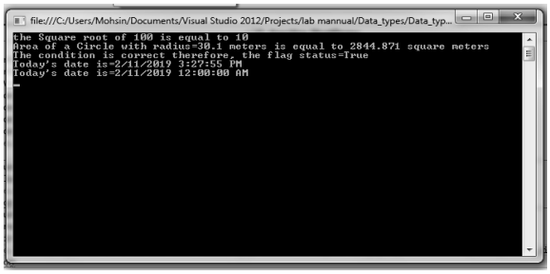
```
'Compile DataTypeDemo.vb
'Program to Demonestrate the use of datatypes in VB
'DataTypeDemo.vb
Module Module1
   Sub Main()
      Dim Var1 As Integer = 100 'Declaration of Interger type
      Dim Var2 As Single = 30.1 'Declaration of Single (float) type
      Const Var3 = 3.14 'Declaration of Constant float
      function_IntegerDemo(Var1) 'function call to implement integer variables
      function_FloatDemo(Var2, Var3) 'function call to implement float variables
      function_BoolDemo(Var1, Var2)
      function_DateDemo()
      Console.ReadKey()
   End Sub
   Function function_IntegerDemo(ByVal Var1) 'Definition of function_IntegerDemo(Var1)
      Dim value As Integer = Math.Sqrt(Var1)
      Console.WriteLine("the Square root of {0} is equal to {1}", Var1, value)
   End Function
   Function function_FloatDemo(ByVal Var2, ByVal Var3)
      Dim Aera_cirlce As Single
      Aera_cirlce = Var3 * (Var2 * Var2)
      Console.WriteLine("the Area of the circle with radius={0} meters is equal to {1}

         square meters", Var2, Aera_cirlce)
```

NOTES

```
   End Function
   Function function_BoolDemo(ByVal Var1, ByVal Var2)
      Dim flag As Boolean
      If (Var1 > Var2) Then
         flag = True
         Console.WriteLine("The condition is correct
therefore, the flag status={0}", flag)
      Else
         flag = False
         Console.WriteLine("The condition is correct
therefore, the flag status={0}", flag)
      End If
   End Function
   Function function_DateDemo()
      Dim Date1 As Date = Date.Now
      Console.WriteLine("Today's date is={0}", Date1)
      Console.WriteLine("Today's date is={0}", Date.
Today)
   End Function
End Module
```

Output: Build and run `DataTypeDemo.vb`

After starting this program the following will be displayed



3. Write a program illustrating constant variable declaration.

'Compile the Const_Decl.vb program in VB

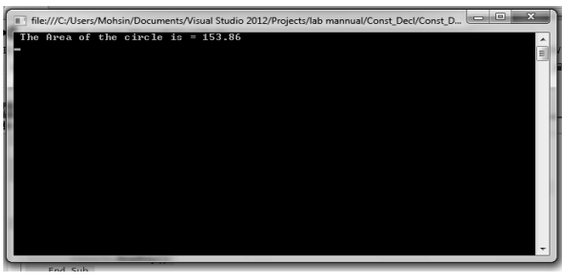
```
'program using constant type to store the value of PI to
calculate area of circle
Module Module1
   Sub Main()
      Const PI = 3.14 ' declaring PI as Constant
```

NOTES

```
    'Even the constant declaration can be done using "Dim
PI As Single = 3.14" statement
      Dim radius, area As Single
      radius = 7
      area = PI * radius * radius ' calculating area of
circle
      Console.WriteLine(" The Area of the circle is =
{0}", area)
      Console.ReadKey()
   End Sub
End Module
```

Output: Build and Start the Application Const_Decl.vb

After Starting Const_Decl.vb the following output console window will be displayed.



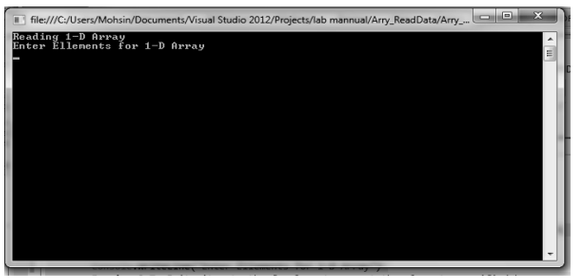
4. Write a program to read one dimensional array.

'Compile `Arry_ReadData.vb` as shown below

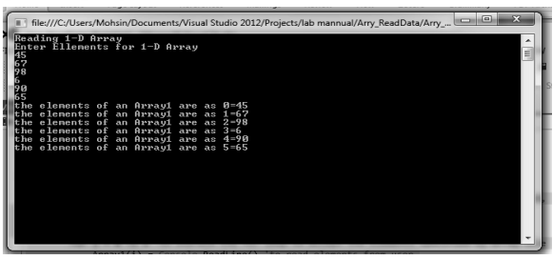
```
'Arry_ReadData.vb
'Array Operation
' Reading and Displaying array elements
Module Module1
   Sub Main()
      Dim Array1(5) As Integer 'Decleration of an Array
with size=6 of type Integer
      ' Array decleration can also be done as Dim Array1()
As Integer = {12, 16, 20, 24, 28, 32}
      Dim i As Integer 'Decleration of Array Index
variable or looping varible
      Console.WriteLine("Reading 1-D Array")
      Console.WriteLine("Enter Ellements for 1-D Array")
      For i = 0 To 5 'to iterate the for loop to accept
the elements specified by array size
```

```
        Array1(i) = Console.ReadLine() 'to read elements
from user
      Next i
      For i = 0 To 5 'to display the array elements
        Console.WriteLine("the elements of an Array1 are
as {0}={1}", i, Array1(i))
      Next i
      Console.ReadKey()
   End Sub
End Module
```

Output: After Build and Start the edited `Arry_ReadData.vb` the following oupt will get displayed.



Enter the Elements; once you are done with feeding array elements the following will get displayed.



5. Write a program to perform operations on one dimensional array.

'Edit `OneDArrayOP.vb` as shown below

```
'One Dimensional Array Operation
'1 Addition
'2 Subtraction
'3 Multiplication
Module Module1
   Sub Main()
      Dim Array1() As Integer = {12, 24, 36, 48, 60}
' Declaring Elements for Array one
```

NOTES

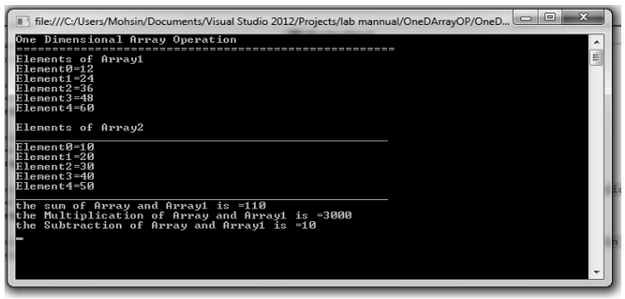
NOTES

```
    Dim Array2() As Integer = {10, 20, 30, 40, 50}
' Declaring Elements for Second  Array
    Dim Array_Add, i, Array_MUL, Array_Sub As Integer
'Declare the varaible to hold the sum of two arrays
    Console.WriteLine("One Dimensional Array Operation")
    Console.WriteLine("=======================")
    Console.WriteLine("Elements of Array1") 'Displaying
Elements of Array1
    For i = 0 To 4
      Console.WriteLine("Element{0}={1}", i, Array1(i))
    Next i
    Console.WriteLine("")
    Console.WriteLine("Elements of Array2")
    Console.WriteLine("_____________________")
    For i = 0 To 4 'Displaying Elements of Array2
      Console.WriteLine("Element{0}={1}", i, Array2(i))
    Next i
    Console.WriteLine("_________________________")
    For i = 0 To 4
      Array_Add = Array1(i) + Array2(i) 'Performing
Addition of Array1 and Array2
    Next i
    Console.WriteLine("the sum of Array and Array1 is
={0}", Array_Add) ' displaying the sum of arrays
    For i = 0 To 4
      Array_MUL = Array1(i) * Array2(i) 'Performing
Multiplication of Array1 and Array2
    Next i
    Console.WriteLine("the Multiplication of Array
and Array1 is ={0}", Array_MUL) ' displaying the
Multiplication of arrays
    For i = 0 To 4
      Array_Sub = Array1(i) - Array2(i) 'Performing
Subtraction of Array1 and Array2
    Next i
    Console.WriteLine("the Subtraction of Array and
Array1 is ={0}", Array_Sub) ' displaying the Subtraction
of arrays
    Console.ReadKey()
```

```
   End Sub
End Module
```

Output: After editing `OneDArrayOP.vb`, Build and start the program. The following console will get displayed.

NOTES



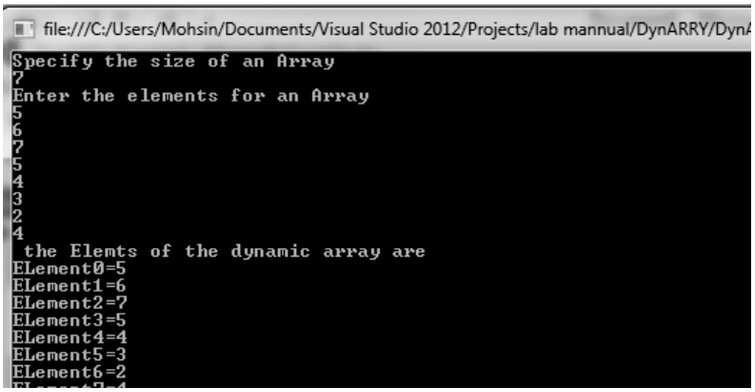
6. Write a program for implementation of dynamic arrays.

'Edit 'DynARRY.vb as shown below:

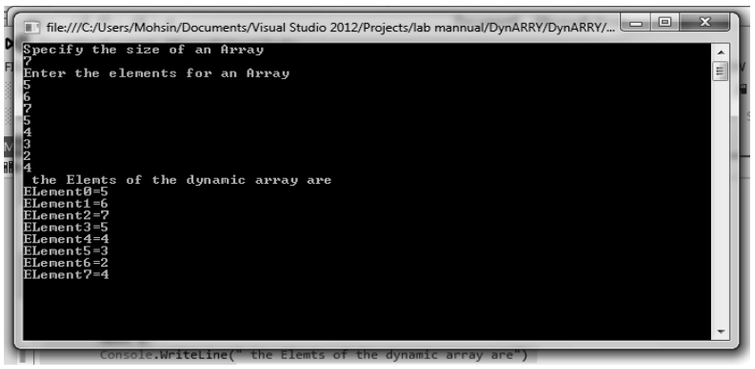
```
'DynARRY.vb
'Implementation of Dynamic Array
Module Module1
   Sub Main()
      Dim DynArr() As Integer ' Declaring an array DynArr
      Dim size As Integer
      Console.WriteLine("Specify the size of an Array")
      size = Console.ReadLine()
      ReDim DynArr(size) 'Dynamic Size allocation by using
ReDim
      Dim i As Integer
      Console.WriteLine("Enter the elements for an Array")
      For i = 0 To size
         DynArr(i) = Console.ReadLine()
      Next i
      Console.WriteLine(" the Elemts of the dynamic array
are")
      For i = 0 To size
         Console.WriteLine("ELement{0}={1}", i, DynArr(i))
' Displaying the elments of Dynamic array
      Next i
      Console.ReadKey()
   End Sub
End Module
```

NOTES

Output: After Editing `'DynARRY.vb,` Build solution and start `'DynARRY.vb` and the following results will be displayed:



After defining the size enter the elements for an array as shown below:



7. Write a program for reading and displaying two dimensional arrays.

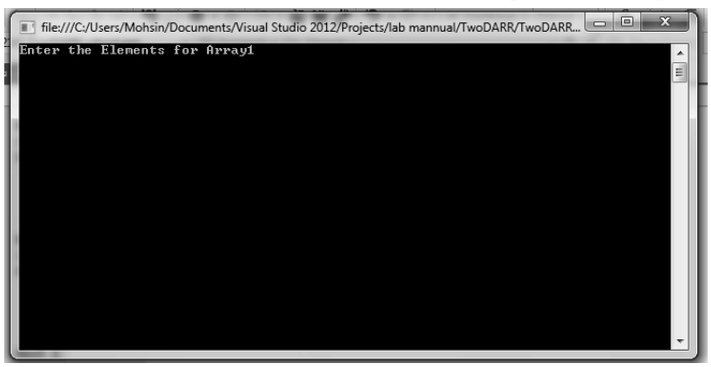
'Edit `TwoDARR.vb` as shown below

```
'Program Reading and displaying two Dimensional Arrays
Module Module1
   Sub Main()
      Dim Array1(2, 3) As Integer 'Declaring Array1 as
two dimensional array
      Dim Array2(2, 3) As Integer 'Declaring Array2 as
two dimensional array
      Dim i, j As Integer
      Console.WriteLine("Enter the Elements for Array1")
      For i = 0 To 2
         For j = 0 To 3
            Array1(i, j) = Console.ReadLine()
```

```
      Next j
   Next i
   Console.WriteLine("Enter the Elements for Array2")
   For i = 0 To 2
      For j = 0 To 3
         Array2(i, j) = Console.ReadLine()
      Next j
   Next i
   Console.WriteLine(" the Elements of Array1 are")
   For i = 0 To 2
      For j = 0 To 3
         Console.Write(Array1(i, j))
      Next j
      Console.WriteLine()
   Next i
   Console.WriteLine(" the Elements of Array2 are")
   For i = 0 To 2
      For j = 0 To 3
         Console.Write(Array2(i, j))
      Next j
      Console.WriteLine()
   Next i
   Console.ReadKey()
End Sub
End Module
```

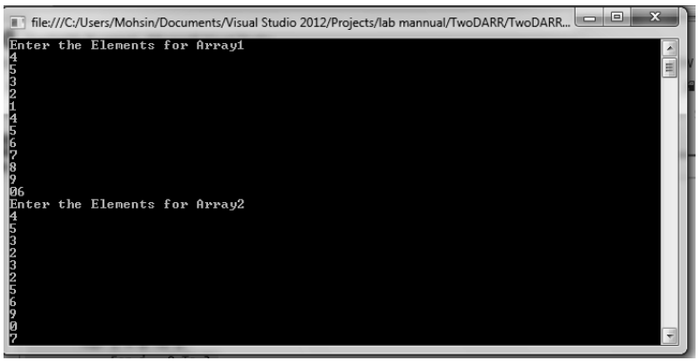
NOTES

Output: Build and start `TwoDARR.vb` the following result will get displayed:

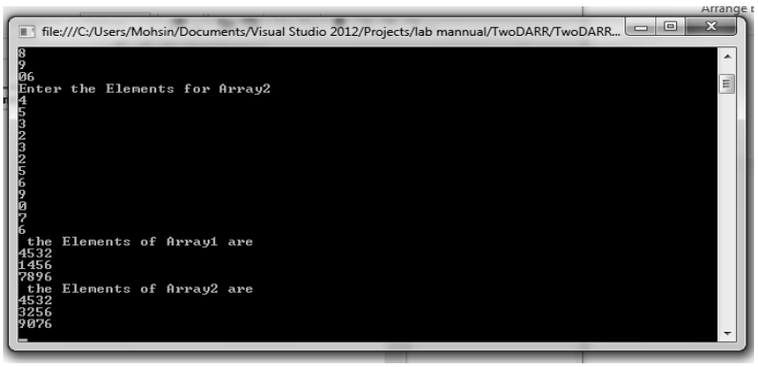


NOTES

Enter the elements for array1 and array2 as shown below:



After finishing entering elements the entered arrays are displayed as shown below:



8. Write a program to perform addition and subtraction on two-dimensional arrays.

'Edit `AddSub2DARR.vb` as shown below

```
' AddSub2DARR.vb
' Program to perform Addition and Subtration operation
on 2-D Arrays
Module Module1
   Sub Main()
      Dim Array1(2, 3) As Integer
      Dim Array2(2, 3) As Integer
      Dim Array3(2, 3) As Integer
      Dim Array4(2, 3) As Integer
      Dim i, j As Integer
      Console.WriteLine("Enter the Elements for Array1")
      For i = 0 To 2
```

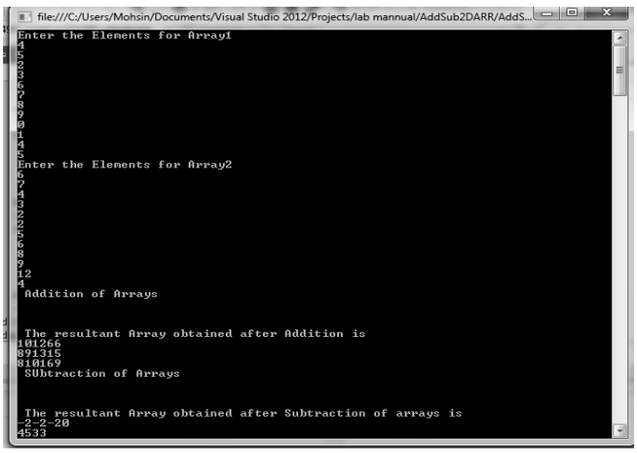
```
      For j = 0 To 3
         Array1(i, j) = Console.ReadLine()
      Next j
   Next i
   Console.WriteLine("Enter the Elements for Array2")
   For i = 0 To 2
      For j = 0 To 3
         Array2(i, j) = Console.ReadLine()
      Next j
   Next i
   Console.WriteLine(" Addition of Arrays")
   For i = 0 To 2
      For j = 0 To 3
         Array3(i, j) = Array1(i, j) + Array2(i, j)
      Next j
      Console.WriteLine()
   Next i
   Console.WriteLine(" The resultant Array obtained
after Addition is")
   For i = 0 To 2
      For j = 0 To 3
         Console.Write(Array3(i, j))
      Next j
      Console.WriteLine()
   Next i
   Console.WriteLine(" SUbtraction of Arrays")
   For i = 0 To 2
      For j = 0 To 3
         Array4(i, j) = Array1(i, j) - Array2(i, j)
      Next j
      Console.WriteLine()
   Next i
   Console.WriteLine(" The resultant Array obtained
after Subtraction of arrays is")
   For i = 0 To 2
```

NOTES

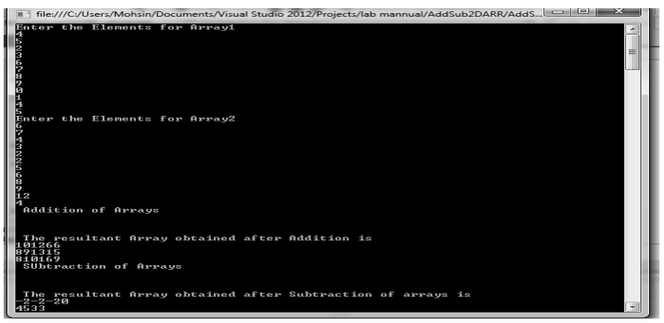
NOTES

```
            For j = 0 To 3
               Console.Write(Array4(i, j))
            Next j
            Console.WriteLine()
        Next i
        Console.ReadKey()
    End Sub
End Module
```

Output: After Editing `AddSub2DARR.vb`, Build the solution and start the program to get the results as shown below:



The Result of Addition and Subtraction is shown below:



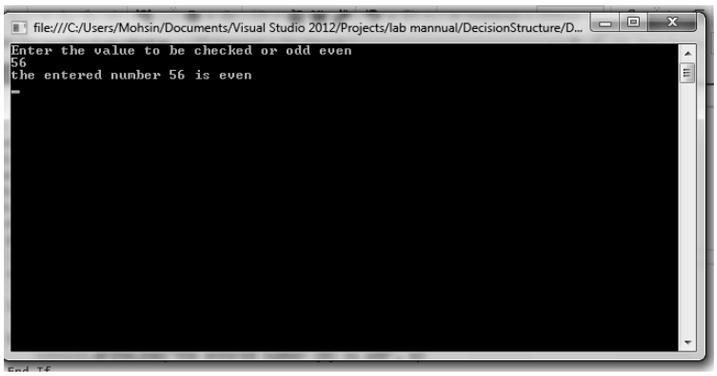
9. Write a program for implementation of decision controls if, if-else.

```
'Edit DecisionStructure.vb as shown below
'Program to Implement Decision Controls if, if-else,
'program will check whether the number is odd or even
using if statement
'If condition Then
```

NOTES

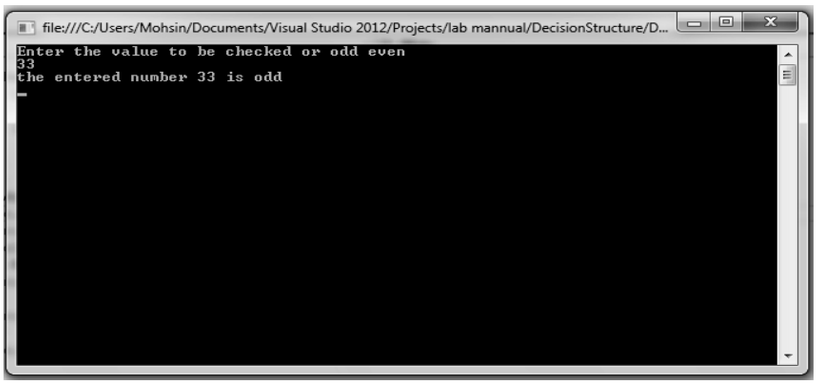
```
'[Statement(s)]
'End If
'If (a < 20) Then
' if condition is true then print the following
'  Else
' if condition is false then print the following
'  End If
Module Module1
   Sub Main()
      Dim x As Integer
      Console.WriteLine("enter the value to be checked
or odd even")
      x = Console.ReadLine()
      ' Console.WriteLine("the entered number {0} is
even", x)
      If ((x Mod 2) = 0) Then
         Console.WriteLine("the entered number {0} is
even", x)
      Else
         Console.WriteLine("the entered number {0} is
odd", x)
      End If
      Console.ReadKey()
   End Sub
End Module
```

Build and Start `DecisionStructure.vb`, the following results will be displayed:



NOTES

Or



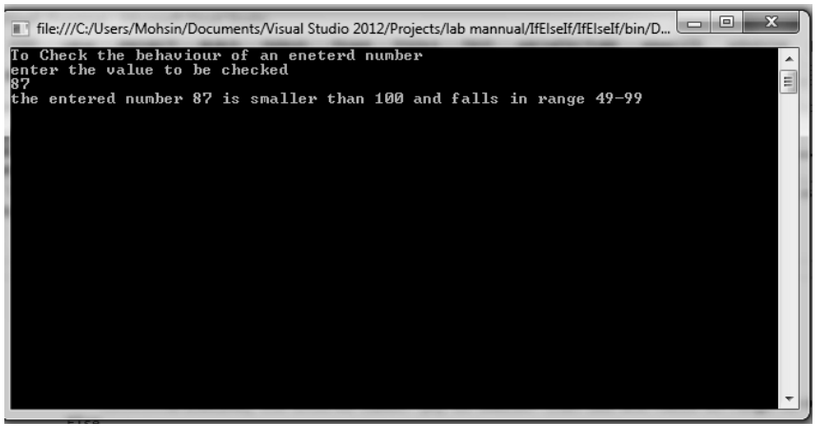
10. Write a program to implement If-elseif-if control statement.

'Edit IfElseIf.vb' to perform the specified operation as shown below

```
'IfElseIf.vb'
'Program to implement If-elseif-if control statement to find the behaviour(range) of an entered by user
Module Module1
   Sub Main()
      Console.WriteLine("To Check the behaviour of an eneterd number") Dim x As Integer
      Console.WriteLine("enter the value to be checked ")
      x = Console.ReadLine()
      If (x > 100) Then
         Console.WriteLine("the entered number {0} is Greater than 100", x)
      ElseIf (x < 100) And (x > 50) Then
         Console.WriteLine("the entered number {0} is smaller than 100 and falls in range 49-99", x)
      Else
         Console.WriteLine("the entered number {0} is less than 49", x)
      End If
      Console.ReadKey()
   End Sub
End Module
```

Output: Build the solution and start `IfElseIf.vb`, the results obtained are displayed as below:

NOTES



11. Write a program illustrating case selection.

'Edit `Case_Select.vb` using following code:

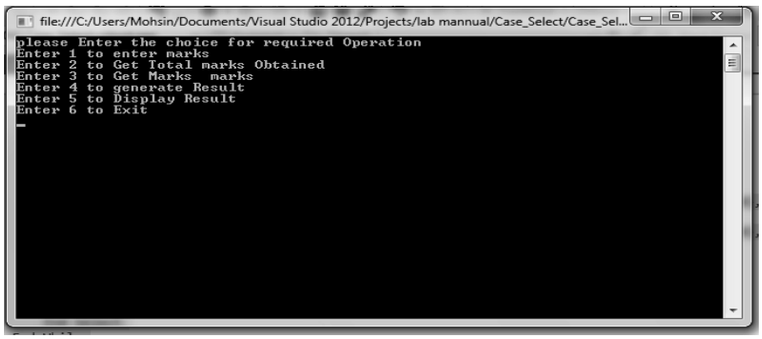
```
'Case_Select.vb
'Program to implement Case Select statment to pefrom spcified operations on student deta
Module Module1
   Sub Main()
      'local variable definition
      Dim Sub_marks1, Sub_Marks2, Sub_Marks3, Tot_Marks As Integer
      Dim percentage As Single
      Dim Result As Boolean
      Dim Choice As Char
      Console.WriteLine("please Enter the choice for required Operation")
      Console.WriteLine("Enter 1 to enter marks")
      Console.WriteLine("Enter 2 to Get Total marks Obtained")
      Console.WriteLine("Enter 3 to Get Marks  marks")
      Console.WriteLine("Enter 4 to generate Result")
      Console.WriteLine("Enter 5 to Display Result")
      Console.WriteLine("Enter 6 to Exit")
      While (1) ' if user enteres true value
        Choice = Console.ReadLine() 'reading variable to store choice
        Select Case Choice 'Checking the choice variable
```

NOTES

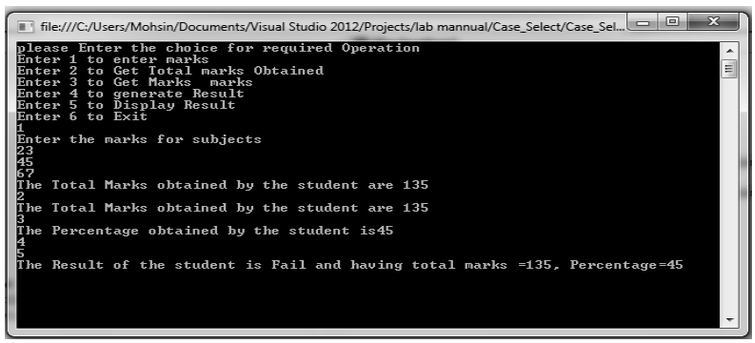
```
            Case "1" ' if choice variable matches with
case 1 then code of case 1 will be executed
            Console.WriteLine("Enter the marks for
subjects")
            Sub_marks1 = Console.ReadLine()
            Sub_Marks2 = Console.ReadLine()
            Sub_Marks3 = Console.ReadLine()
            Tot_Marks = Sub_marks1 + Sub_Marks2 + Sub_
Marks3
            Console.WriteLine("The Total Marks obtained
by the student are {0}", Tot_Marks)
            Case "2"
            Tot_Marks = Sub_marks1 + Sub_Marks2 + Sub_
Marks3
            Console.WriteLine("The Total Marks obtained
by the student are {0}", Tot_Marks)
            Case "3"
            percentage = (Tot_Marks / 300) * 100
            Console.WriteLine("The Percentage obtained by
the student is{0}", percentage)
            Case "4"
            If (percentage > 50.0) Then
                  Result = True
            Else
                  Result = False
            End If
            Case "5"
            If Result = True Then
                  Console.WriteLine("The Result of
the student is Pass and having total marks ={0},
Percentage={1}", Tot_Marks, percentage)
            Else
                  Console.WriteLine("The Result of
the student is Fail and having total marks ={0},
Percentage={1}", Tot_Marks, percentage)
            End If
            Case "6"
            Exit Sub
```

```
            Case Else
            Console.WriteLine("Enter Vaild OPtion")
         End Select
      End While
      Console.ReadKey()
   End Sub
End Module
```

Output: Build and Start `Case_Select.vb`



Select the choice to perform the specified operation as shown below:



12. Write a program to implement various loops in Vb.

'Edit `Loops.vb` to perfrom the implementation of various loops aspecifed in the code below

```
'Loops.vb
'Program to implement loops like While,do-while, For-Next,
Module Module1
   Sub Main()
      Dim a As Integer
      Dim i = 0, z As Integer
```

NOTES

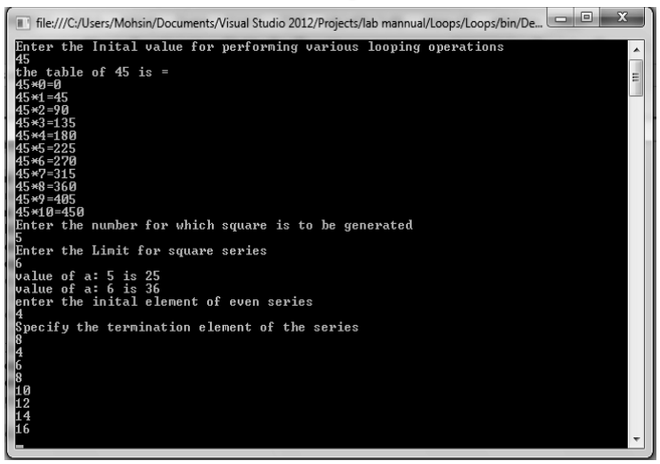
NOTES

```
      Console.WriteLine("Enter the Inital value for
performing various looping operations")
      a = Console.ReadLine()
      ' while loop execution '
      Console.WriteLine("the table of {0} is =", a)
      'using while loop to generate table of a number
      While i <= 10
         z = i * a
         Console.WriteLine("{0}*{1}={2}", a, i, z)
         i = i + 1
      End While
      Dim num As Integer = 10
      'do loop execution
      Console.WriteLine("Enter the number for which
square is to be generated")
      num = Console.ReadLine()
      Console.WriteLine("Enter the Limit for square
series")
      Dim limt = Console.ReadLine()
      'Implementation of do-while to generate square of
a number
      Do
         Dim sq = num * num
         Console.WriteLine("value of a: {0} is {1}", num,
sq)
         num = num + 1
      Loop While (num <= limt) 'conditional statement of
dowhile to restrict the behaviour of loop
      Dim Even, Term As Integer
      Dim loop_var As Integer
      Console.WriteLine("enter the inital element of even
series")
      Even = Console.ReadLine()
      Console.WriteLine("Specify the termination element
of the series")
      Term = Console.ReadLine()
      ' for loop execution
      For loop_var = 2 To Term Step 1 'For loop to generate
even series up to number
```

```
        Even = loop_var * 2
        Console.WriteLine("{0}", Even)
      Next
      Console.ReadLine()
   End Sub
End Module
```

NOTES

Output: Build and start `Loops.vb` to get results as shown below:



13. Write a program illustrating the concept of nested loops in VB.

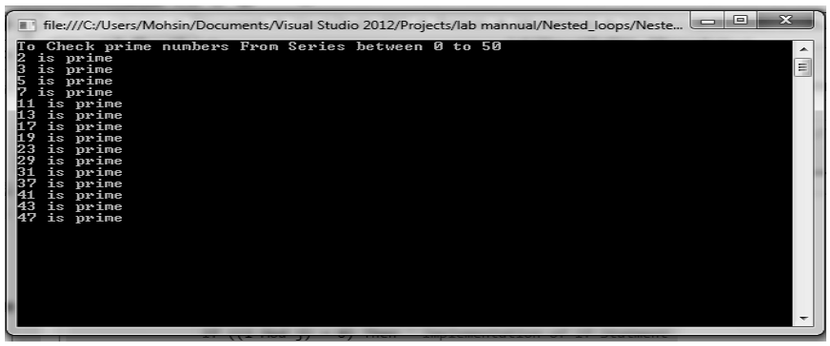
'Edit Nested_Loops.vb to demonestrate the implementation of Nested Loops as shown below:

```
'Nested_Loops.vb
'Program to demonestrate the nested Loops
' Program checks numbers between 2 to 100 to find the primes numbers
'and displays the same as a result
Module Module1
   Sub Main()
      ' local variable definition
      Console.WriteLine("To Check prime numbers From Series between 0 to 50")
      Dim i, j As Integer
      For i = 2 To 50 'Generates numbers between 2 to 100
         For j = 2 To i ' implementation of for loop
            ' if factor found, not prime
            If ((i Mod j) = 0) Then ' Implementation of If Statment
            Exit For
```

NOTES

```
            End If
         Next j
         If (j > (i \ j)) Then
            Console.WriteLine("{0} is prime", i)
         End If
      Next i
      Console.ReadLine()
   End Sub
End Module
```

Output: Build and start `Nested_Loops.vb` to display the result of implemention as shown below:



14. Write a program implementing function to perfrom various arithmetic operations.

'Edit `Simple_Function.vb` to perfrom the desired operation using function as shown below

```
'Simple_Function.vb
'Program to implement a simple function to perfrom
addition, Subtraction and division of two numbers
Module Module1
   Dim val1, val2 As Integer 'Decleration of varibales
accessible to all functions within the module
   Dim Sum, Prod, Div, Subt As Integer
   Sub Main()
      Sum_function() ' Fucntion Decleration or Function
call for various functions
      Sub_function()
      Mult_fucntion()
      Div_function()
      Console.ReadKey()
   End Sub
```

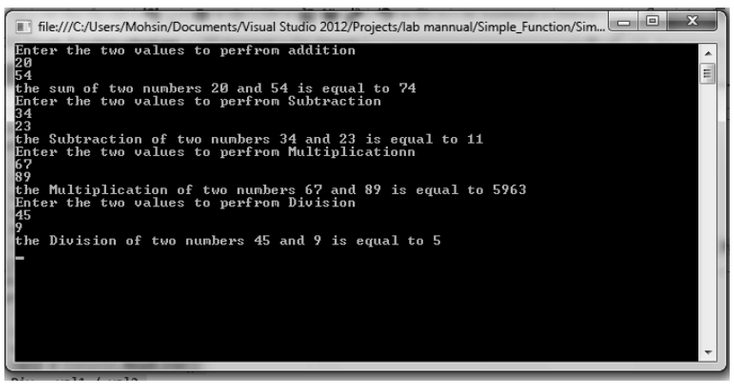
NOTES

```
   'Code to implement Sum() function
   Function Sum_function() ' Function defintion to perfrom
the desired operation
      ' Dim val1, val2 As Integer
      Console.WriteLine("Enter the two values to perfrom
addition")
      val1 = Console.ReadLine() ' Accepting value for
val1 and val2 variables from the user
      val2 = Console.ReadLine()
      Sum = val1 + val2
      Console.WriteLine("the sum of two numbers {0} and
{1} is equal to {2}", val1, val2, Sum)
   End Function
   'Code to implement Sub() function
   Function Sub_function()
      ' Dim val1, val2 As Integer
      Console.WriteLine("Enter the two values to perfrom
Subtraction")
      val1 = Console.ReadLine()
      val2 = Console.ReadLine()
      Subt = val1 - val2
      Console.WriteLine("the Subtraction of two numbers
{0} and {1} is equal to {2}", val1, val2, Subt)
   End Function
   'Code to implement Mult() function
   Function Mult_fucntion()
      ' Dim val1, val2 As Integer
      Console.WriteLine("Enter the two values to perfrom
Multiplicationn")
      val1 = Console.ReadLine()
      val2 = Console.ReadLine()
      Prod = val1 * val2
      Console.WriteLine("the Multiplication of two
numbers {0} and {1} is equal to {2}", val1, val2, Prod)
   End Function
   'Code to implement Div() function
   Function Div_function()
      ' Dim val1, val2 As Integer
```

NOTES

```
      Console.WriteLine("Enter the two values to perfrom
Division")
      val1 = Console.ReadLine()
      val2 = Console.ReadLine()
      Div = val1 / val2
      Console.WriteLine("the Division of two numbers {0}
and {1} is equal to {2}", val1, val2, Div)
   End Function
End Module
```

Output: Build solution and Start `Simple_Function.vb` to display the results of functions defined as shown below:



15. Write a program implementing procedure to perform addition and subtraction of two numbers.

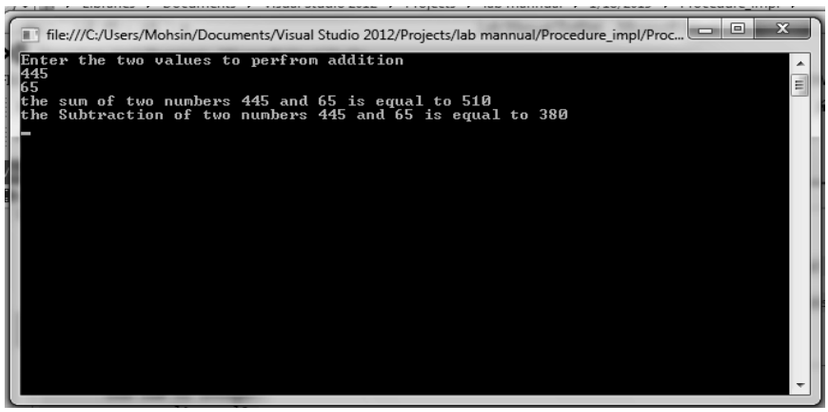
'Edit `Procedure_Impl.vb` to implement the usage of procedures in Vb by writing following code

```
'Procedure_Impl.vb
' Program to implement procedure to perfrom addtion and
subtraction of two numbers
Module Module1
   Sub Main()
      Dim val1, val2 As Integer
      Console.WriteLine("Enter the two values to perform
addition")
      val1 = Console.ReadLine() ' Accepting value for
val1 and val2 variables from the user
      val2 = Console.ReadLine()
      Sum_function(val1, val2) ' Fucntion Decleration or
Function call for various functions with two argumnets
      Sub_function(val1, val2)
      Console.ReadKey()
```

NOTES

```
   End Sub
   Sub Sum_function(ByVal val1 As Integer, ByVal val2 As
Integer)
      Dim sum As Integer
      sum = val1 + val2
      Console.WriteLine("the sum of two numbers {0} and
{1} is equal to {2}", val1, val2, sum)
   End Sub
   Sub Sub_function(ByVal val1 As Integer, ByVal val2 As
Integer)
      Dim subt As Integer
      subt = val1 - val2
      Console.WriteLine("the Subtraction of two numbers
{0} and {1} is equal to {2}", val1, val2, subt)
   End Sub
End Module
```

Output: Build the solution and start `Procedure_Impl.vb` to display the result as the one obtained shown below:



16. Write a program demonstrating function with arguments.

'Edit `FunctionArguments.vb` to implement function with arguments as shown in code below:

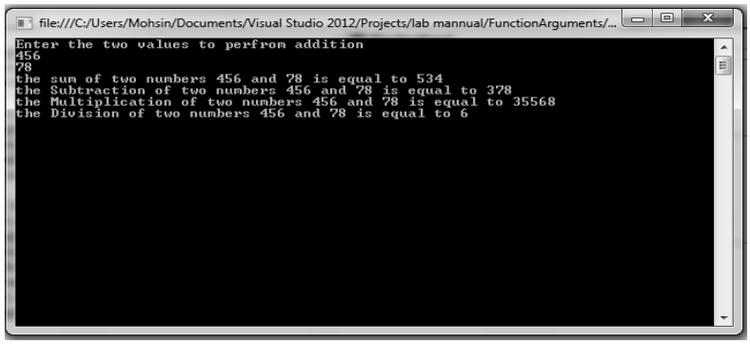
```
'FunctionArguments.vb
'Program to implement functions with arguments
Module Module1
   Dim val1, val2 As Integer 'Decleration of varibales
accessible to all functions within the module
   Dim Sum, Prod, Div, Subt As Integer
   Sub Main()
```

NOTES

```
      Console.WriteLine("Enter the two values to perfrom
addition")
      val1 = Console.ReadLine() ' Accepting value for
val1 and val2 variables from the user
      val2 = Console.ReadLine()
      Sum_function(val1, val2) ' Fucntion Decleration or
Function call for various functions with two argumnets
      Sub_function(val1, val2)
      Mult_fucntion(val1, val2)
      Div_function(val1, val2)
      Console.ReadKey()
   End Sub
   'Code to implement Sum() function
   Function Sum_function(ByVal val1, ByVal val2) '
Function defintion to perfrom the desired operation
      'Byval specifies the argument is passed as a value
not as a refrence
      Sum = val1 + val2
      Console.WriteLine("the sum of two numbers {0} and
{1} is equal to {2}", val1, val2, Sum)
   End Function
   'Code to implement Sub() function
   Function Sub_function(ByVal val1, ByVal val2)
      Subt = val1 - val2
      Console.WriteLine("the Subtraction of two numbers
{0} and {1} is equal to {2}", val1, val2, Subt)
   End Function
   'Code to implement Mult() function
   Function Mult_fucntion(ByVal val1, ByVal val2
      Prod = val1 * val2
      Console.WriteLine("the Multiplication of two
numbers {0} and {1} is equal to {2}", val1, val2, Prod)
   End Function
   'Code to implement Div() function
   Function Div_function(ByVal val1, ByVal val2)
      Div = val1 / val2
      Console.WriteLine("the Division of two numbers {0}
and {1} is equal to {2}", val1, val2, Div)
   End Function
```

```
End Module
```

Output: Build the solution and start the `FunctionArguments.vb` to display the results as shown below:



NOTES

17. Write a program to implement function returning values.

'Edit `FunctionReturingVal.vb` to implement function that returns values using the following code

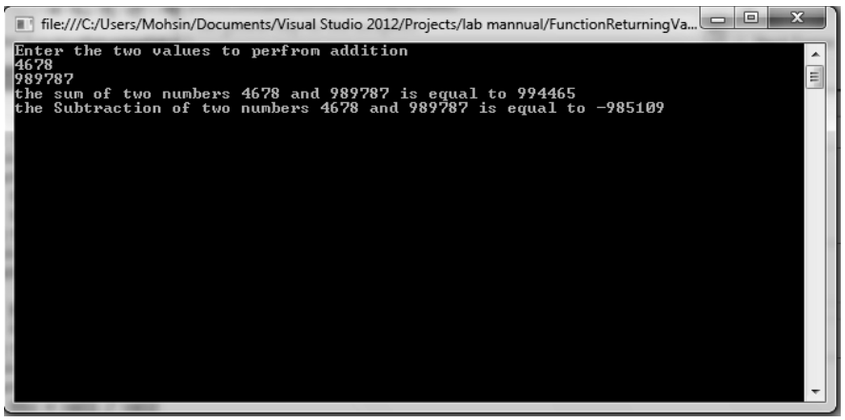
```
'FunctionReturingVal.vb
'Program to demonestrate the implementation of Function returing values
Module Module1
   Dim val1, val2 As Integer 'Decleration of varibales accessible to all functions within the module
   Dim Sum, Prod, Subt As Integer
   Dim Div As Single
   Sub Main()
      Console.WriteLine("Enter the two values to perfrom addition")
      val1 = Console.ReadLine() ' Accepting value for val1 and val2 variables from the user
      val2 = Console.ReadLine()
      Sum = Sum_function(val1, val2) ' Fucntion Decleration or Function call for various functions with two argumnets
      Console.WriteLine("the sum of two numbers {0} and {1} is equal to {2}", val1, val2, Sum)
      Subt = Sub_function(val1, val2)
      Console.WriteLine("the Subtraction of two numbers {0} and {1} is equal to {2}", val1, val2, Subt)
      Prod = Mult_fucntion(val1, val2)
      Console.WriteLine("the Multiplication of two numbers {0} and {1} is equal to {2}", val1, val2, Prod)
      Div = Div_function(val1, val2)
```

NOTES

```
        Console.WriteLine("the Division of two numbers {0} and {1} is equal to {2}", val1, val2, Div)
        Console.ReadKey()
    End Sub
    'Code to implement Sum() function
    Function Sum_function(ByVal val1, ByVal val2) As Integer  ' Function defintion to perfrom the desired operation
        'As Integer specifes the type of the variable to be returned by function
        'Byval specifies the argument is passed as a value not as a refrence
        Sum = val1 + val2
        Sum_function = Sum 'decleration to return value of sum variable generated
    End Function
    'Code to implement Sub() function
    Function Sub_function(ByVal val1, ByVal val2) As Integer
        Subt = val1 - val2
        Sub_function = Subt
    End Function
    'Code to implement Mult() function
    Function Mult_fucntion(ByVal val1, ByVal val2) As Integer
        ' Dim val1, val2 As Integer
        Prod = val1 * val2
        Mult_fucntion = Prod
    End Function
    'Code to implement Div() function
    Function Div_function(ByVal val1, ByVal val2) As Single
        Div = val1 / val2
        Div_function = Div
    End Function
End Module
```

Output: Build the solution and Start `FunctionReturingVal.vb` to display the results as shown below:

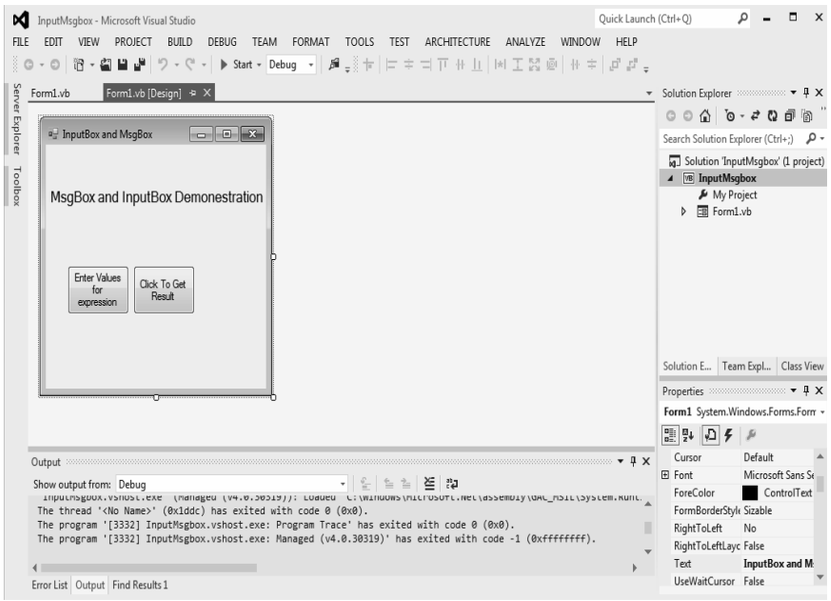
NOTES



18. Write a program to implement inputbox.

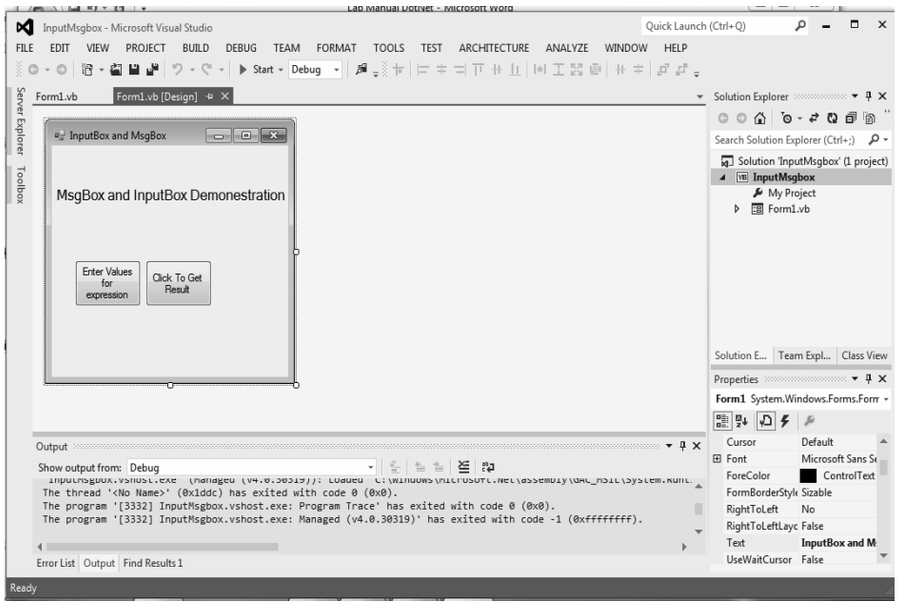
Step 1: Create Windows form based application (File→ New→project→Windows→Windows Forms Application.

Step 2: A Form will be displayed and you need to add two buttons from toolbox on your form (`Form1.vb[Design]` as shown below:



Step 3: You can customize the properties of form and each control on the form from the properties window displayed on right button of your Visual Studio as shown below:

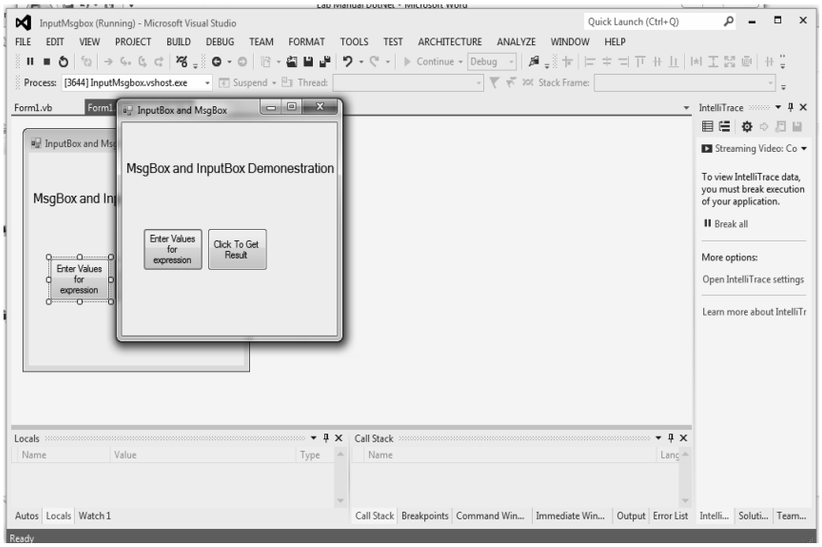
NOTES



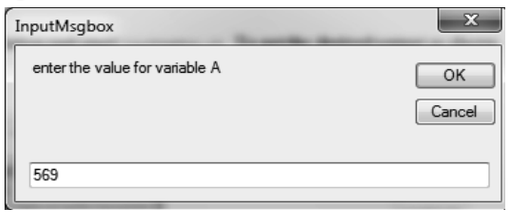
Step 4: Click over the button with text "Enter Values for expression" and "Click To Get Result" , Edit `Form1.vb` the way as shown below to customize the event associated to it as shown below:

```
'Form1.vb
'Program to demonestrate the implementation of Input Box and Msg Box in VB
Public Class Form1
   Dim a As Integer
   Dim B, result As Integer
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Enter_Values_for_expression_button.Click
      a = Val(InputBox("enter the value for variable A"))
' Use of inputbox to accept user value for varibale A
      B = Val(InputBox("enter the value for variable B"))
   End Sub
   Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Click_to_Get_Result_Button.Click
      result = a + B
      result = MsgBox(result, 3, "Addition Result")
' Use of MsgBox to display the result of addition of A+B
   End Sub
End Class
```

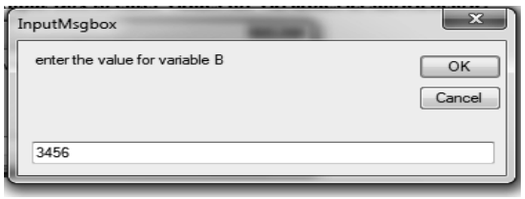
Step 5: Built the solution and start `InputMsgBox.vb` to get the desired output as shown below:



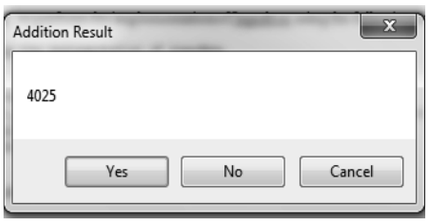
Step 6: After click on button "Enter Values for Expression" you will be prompted to by Input Box to enter values for variables as shown below:



Click Ok to enter value for variable B as shown below:



Step 7: After feeding values for A and B variable click on "Click to Get Result" Button to get results displayed through MsgBox as shown below:

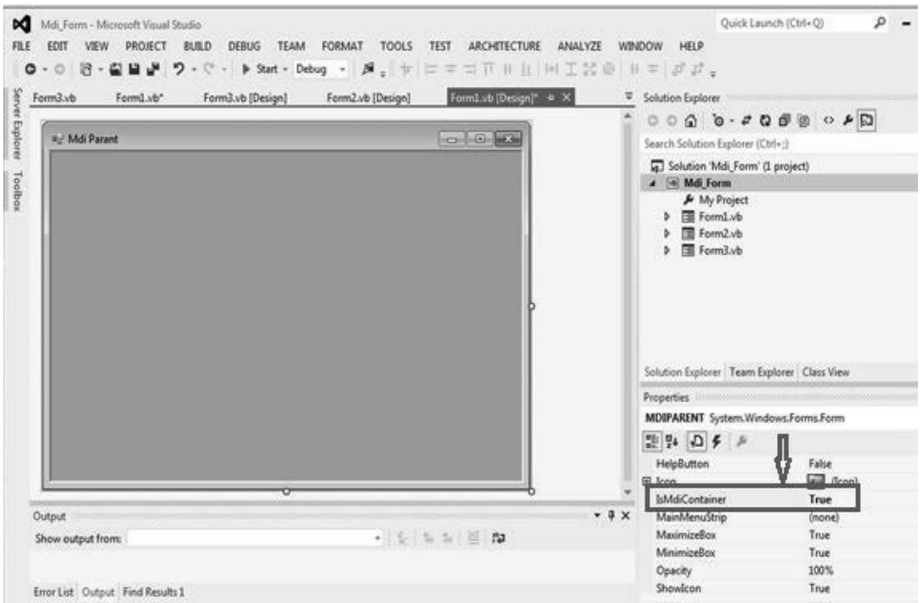


NOTES

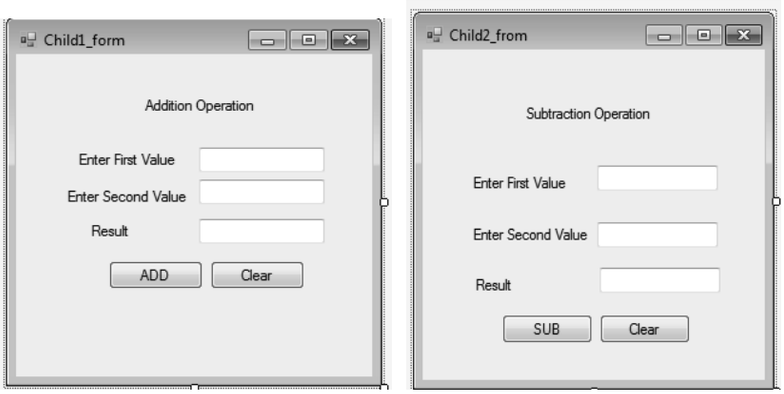
NOTES

19. Write a program to demonstrate the implementation of MDI forms.

Step 1: Create new Windows forms application and name it as "Mdi_Form". A default Windows form will be displayed. Click on the Form Header and visit its properties window. From Properties Window change property "IsMdiContainer to True"as shown below:



Step 2: Right click on Mdi_From in solution explorer and add two Forms by clicking over Add option therein and Name them Form2 and Form3 respectively. Customize these Form2(Child1_form) and Form3(Child2_form) as shown below to make them to perform two separate functions.



Step 3: Customise each control pasted on the Form2 and Form3 as per the functions specified on labeled as show above. After customizing these form2. vb Form3.vb the code behind them will be as shown below:

NOTES

Form2.vb

```
'From2.vb
'Program to demonestrate the use of MDI Form to perform
Addition operation
Public Class Child1_form
   Private Sub Label1_Click(sender As Object, e As
EventArgs) Handles Label1.Click
   End Sub
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      Dim sum As Integer
      sum = Convert.ToInt32(TextBox1.Text) + Convert.
ToInt32(TextBox2.Text)
      TextBox3.Text = sum
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      TextBox1.Text = " "
      TextBox2.Text = " "
      TextBox3.Text = " "
   End Sub
End Class
```

Form3.vb

```
'From3.vb
'Program to demonestrate the use of MDI Form to perfrom
Subtration Operation
Public Class Child_form2
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      Dim subt As Integer
      subt = Convert.ToInt32(TextBox2.Text) - Convert.
ToInt32(TextBox1.Text)
      TextBox3.Text = subt
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      TextBox1.Text = " "
```

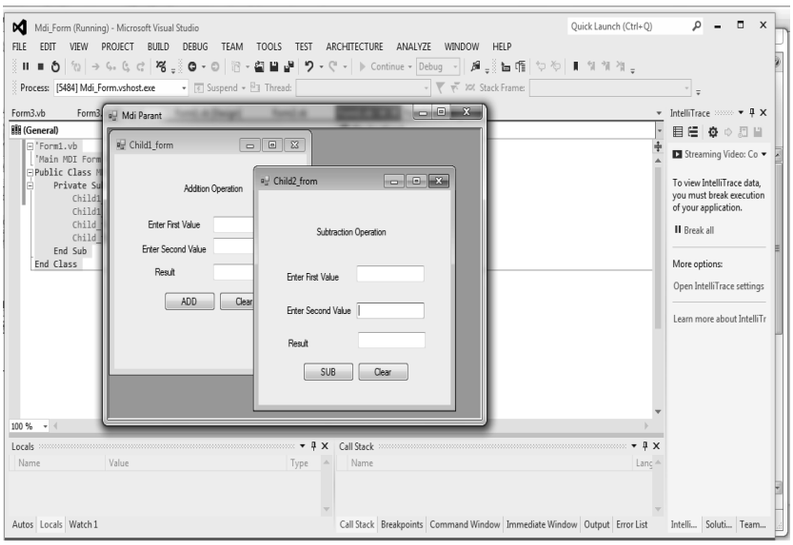
```
        TextBox2.Text = " "
        TextBox3.Text = " "
    End Sub
End Class
```

NOTES

Step 4: Click over the MDI Form and Write the code as shown below:

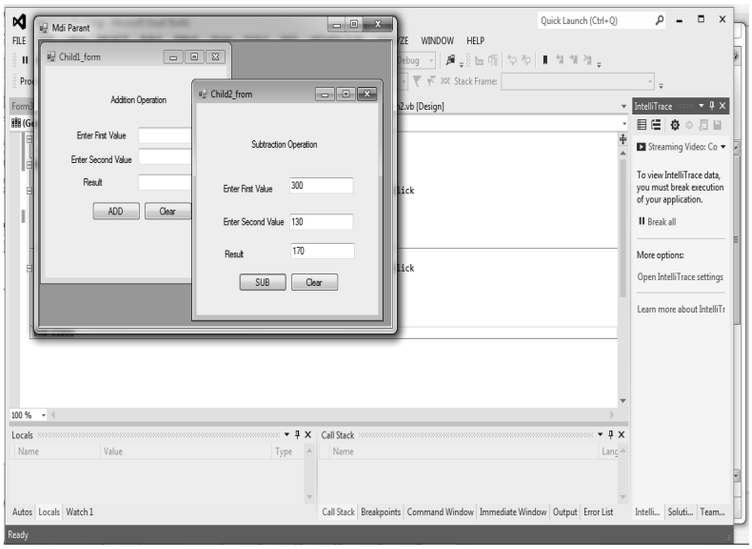
```
'Form1.vb
'Main MDI Form to link two Forms(Form2 and Form3) to perfrom two separate opertions on them
Public Class MDIPARENT
    Private Sub MDIPARENT_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Child1_form.MdiParent = Me
        Child1_form.Show()
        Child_form2.MdiParent = Me
        Child_form2.Show()
    End Sub
End Class
```

Step 5: Build and start Mdi_From to display the outcome as shown below:

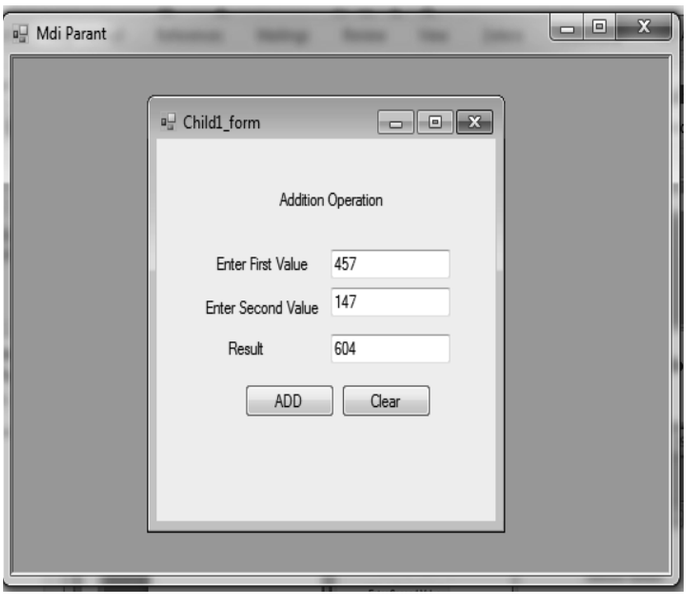


Click on these separate forms to perform the operations specified on them the way shown below:

NOTES



Close the current form and perform operation on another form as shown below:



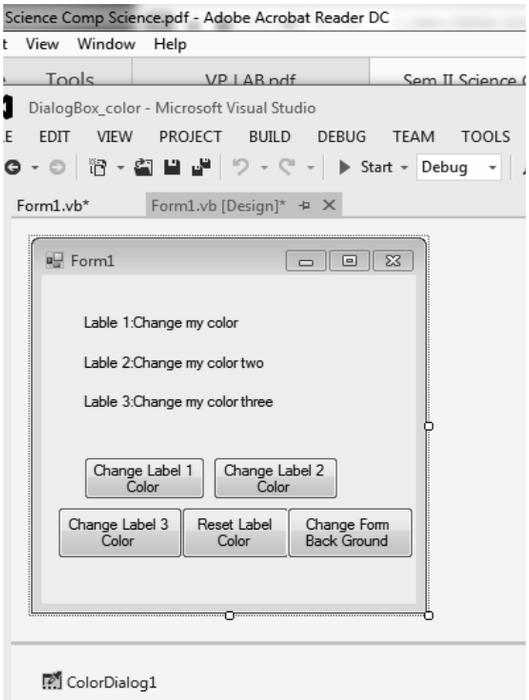
20. Write a program for creating color dialog box.

Step 1: Create New Windows Form Application "DialogBox_Color"

Step 2: Right Click on "DialogBox_Color" in solution explorer to add Windows form as Form1.vb

Step 3: Add the Labels, Buttons, and ColorDialog from toolbox on Form1. Design as shown below:

NOTES



Step 4: Customize each control provided on the form1.Design in its From1.vb file as shown below:

```
'Form1.vb
'Program to demonestrate the implementation of Color dialog box and the role of different'
'color options associated with color dialog box in vb
Public Class Form1
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      If ColorDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label1.ForeColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click
   End Sub
   Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
      If ColorDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label2.ForeColor = ColorDialog1.Color
```

```
    End If
  End Sub
  Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
    If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
      Label3.ForeColor = ColorDialog1.Color
    End If
  End Sub
  Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
    Label1.ForeColor = Color.Black
    Label2.ForeColor = Color.Black
    Label3.ForeColor = Color.Black
  End Sub
  Private Sub Button5_Click(sender As Object, e As
EventArgs) Handles Button5.Click
    If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
      Me.BackColor = ColorDialog1.Color
    End If
  End Sub
  Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
  End Sub
End Class
```

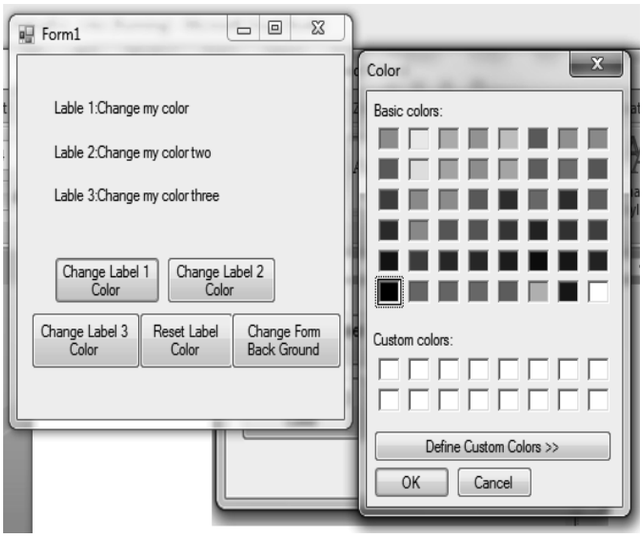
NOTES

Step 5: Build and start the project to run the effects of color dialog box as shown below:

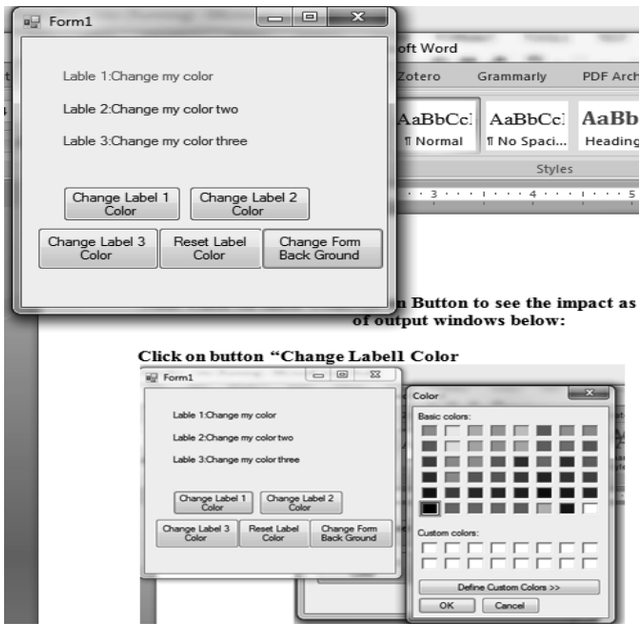
NOTES

Next click on these controls on button to see the impact as shown the series of output windows below:

Click on button “Change Label1 Color

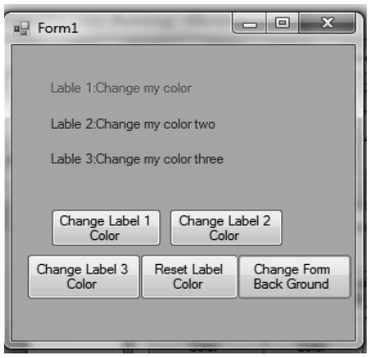


Select color from color grid and click Ok.

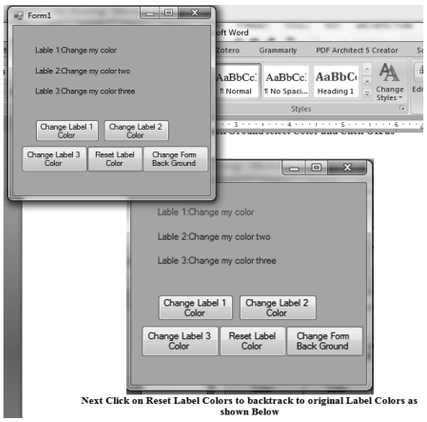


Next click on ‘Change Form Back Ground’ select Color and Click OK as shown below:

NOTES



Next click on 'Reset Label Colors' to backtrack to original Label Colors as shown below:

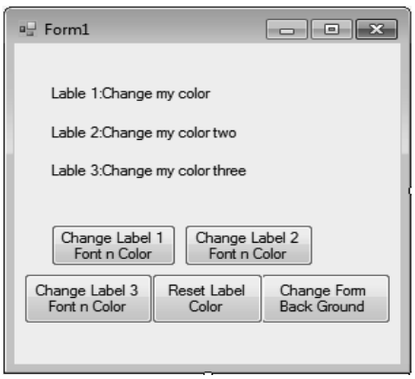


21. Write a program to demonstrate the use of Font Dialog Box.

Step 1: Create New Windows Form Application "FontDialog".

Step 2: Right click on "FontDialog" in solution explorer to add Windows form as Font_ColorDialog.vb.

Step 3: Add the Labels, Buttons, FontDialog and ColorDialog from toolbox on Font_ColorDialog..Design as shown below:



Step 4: Customize each control provided on the Font_ColorDialog.Design in its Font_ColorDialog.vb file as shown below:

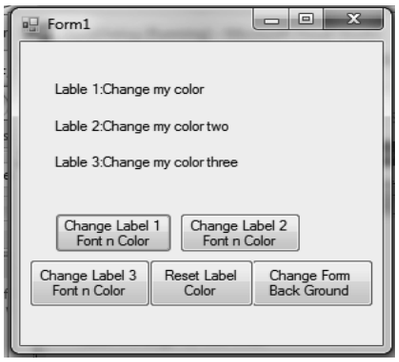
Font_ColorDialog.vb

NOTES

```
'Font_ColorDialog.vb
'Program to demonestrate the implementation the functionality ofFont and Color dialog box
Public Class Font_ColorDialog
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      If FontDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label1.ForeColor = FontDialog1.Color
         Label1.Font = FontDialog1.Font
      End If
      If ColorDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label1.ForeColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
      If FontDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label2.ForeColor = FontDialog1.Color
         Label2.Font = FontDialog1.Font
      End If
      If ColorDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label2.ForeColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
      If FontDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         Label3.ForeColor = FontDialog1.Color
```

```
        Label3.Font = FontDialog1.Font
      End If
      If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
        Label3.ForeColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
      Me.ColorDialog1.Reset()
   End Sub
   Private Sub Button5_Click(sender As Object, e As
EventArgs) Handles Button5.Click
      If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
        Me.BackColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Font_ColorDialog_Load(sender As Object,
e As EventArgs) Handles MyBase.Load
   End Sub
End Class
```

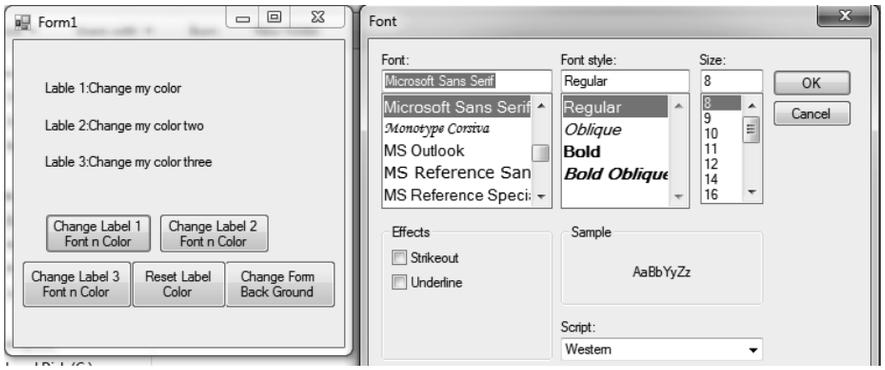
Step 5: Build and start the project to run the effects of Font and Color Dialog Box as shown below:



Next click on these controls on button to see the impact as shown the series of output Windows below:

NOTES

Click on button "Change Label1Font n Color" from the dialog box shown below:



Next select color as shown below:

Simultaneously perform the operations specified in each button to change font and color the overall results are shown as under:

Next click on 'Reset Label Color' and 'Change Form Back Groud' to revert form to its original status as shown below:

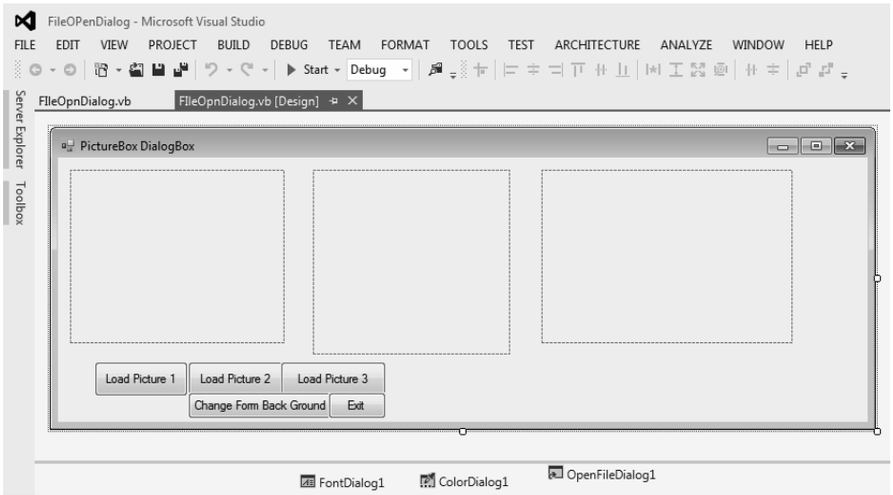
NOTES

22. Write a program to demonstrate the use of File Open Dialog Box.

Step 1: Create New Windows Form Application "FileOPenDialog".

Step 2: Right click on "FileOPenDialog" in solution explorer to add Windows form as FileOpenDialog.vb.

Step 3: Add the Labels, Buttons, PictureBox, FileOpenDialog, FontDialog and ColorDialog from toolbox on FIleOpnDialog.Design as shown below:



Step 4: Customize each control provided on the form1.Design in its From1. vb file as shown below:

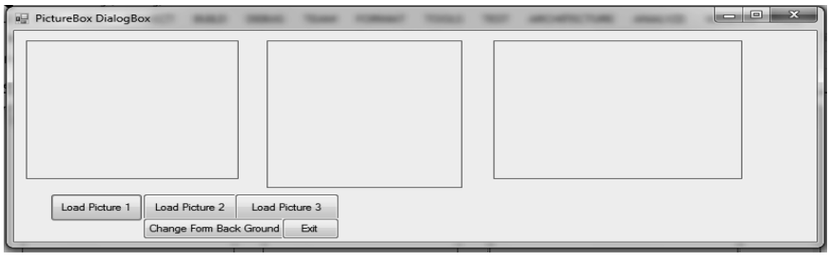
FileOpnDialog.vb

```
'FileOpnDialog.vb
'Program to demonestrate the implementation of
FileOpenDialog, FOntDialog and Color dialog box.
Public Class FIleOpnDialog
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
```

NOTES

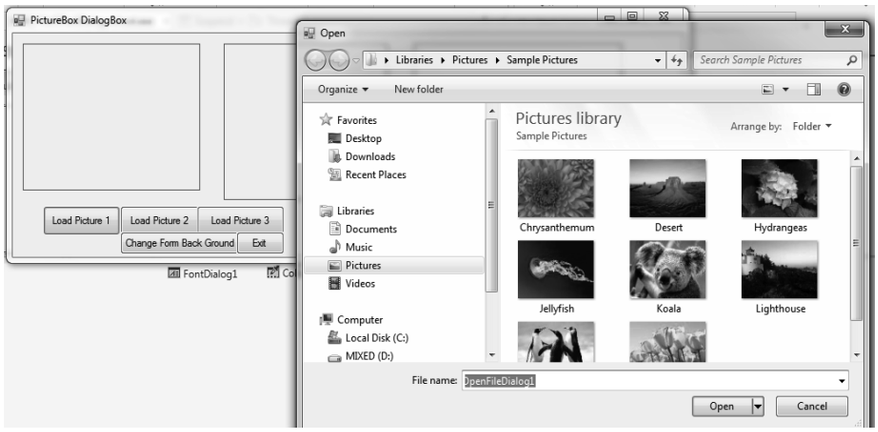
```
      If OpenFileDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then 'Overriding the functionnality
of FileOpenDialog box
      PictureBox1.Image = Image.FromFile(OpenFileDialog1.
FileName) ' Specifying the file here in this case an image
file
         PictureBox1.SizeMode = PictureBoxSizeMode.Zoom
      End If
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      If OpenFileDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
         PictureBox2.Image = Image.
FromFile(OpenFileDialog1.FileName)
         PictureBox2.SizeMode = PictureBoxSizeMode.Zoom
      End If
   End Sub
   Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
      If OpenFileDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
         PictureBox3.Image = Image.
FromFile(OpenFileDialog1.FileName)
         PictureBox3.SizeMode = PictureBoxSizeMode.Zoom
      End If
   End Sub
   Private Sub Button5_Click(sender As Object, e As
EventArgs) Handles Button5.Click
      If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
         Me.BackColor = ColorDialog1.Color
      End If
   End Sub
   Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
      Me.Close()
   End Sub
End Class
```

Step 5: Build and start the project to run the implementation FileOpenDialog as shown below:

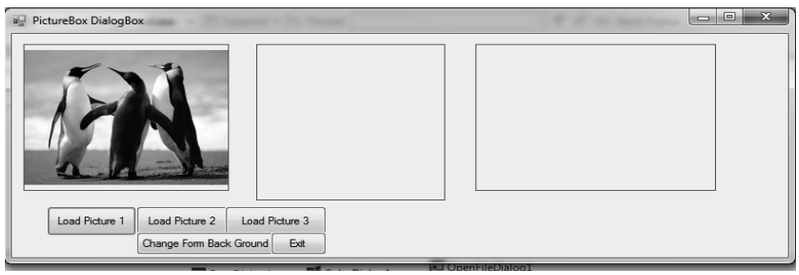


NOTES

Step 6: Click on the button controls with Caption Load Picture 1, After clicking over the said button FileOpenDialog will be prompted to select the particular image file to loaded into PictureBox1 the same is shown in figure below:



Next select any Image and Click Open and the selected image you will see loaded into PictureBox1 as shown below:



Repeat the above operation for other two buttons and the final outcome will look like as shown below:

NOTES

To Change Form Background click on the desired button and form background will look like

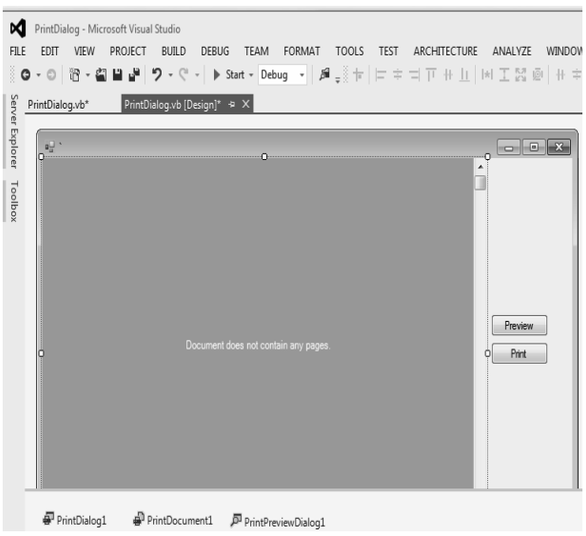
To make exit from this project click over the Exit button.

23. Write a program to demonstrate the implementation of PrintDialog box.

Step 1: Create New Windows Form Application "PrintDialog".

Step 2: Right Click on "PrintDialog" in Solution Explorer to Add Windows Form as PrintDialog.vb.

Step 3: Add the Labels, Buttons, PrintDialog, PrintDocument and PrintPreviewDialog from toolbox on PrintDialog.Design as shown below:



Step 4: Customize Each Control provided on the form1.Design in its From1.vb file as shown below:

PrintDialog.vb

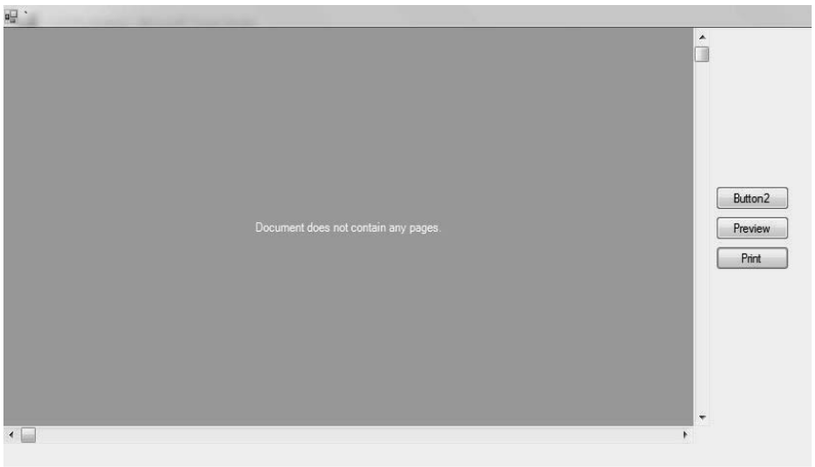
NOTES

```
'PrintDialog.vb
'Program to demonestrate the implementation of PrintDialog, PrintDocument and PrintPreviewDialog
Public Class PrintDialog
   Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
   End Sub
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Print.Click
      If PrintDialog1.ShowDialog = DialogResult.OK Then
         PrintDocument1.PrinterSettings = PrintDialog1.PrinterSettings
         PrintDocument1.Print()
      End If
   End Sub
   Private Sub PrintDocument1_PrintPage(sender As Object, e As Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage
      Dim DocumentFont As Font = New Drawing.Font("Time New Roman", 14)
      e.Graphics.DrawString("TITLE OF THE DOCUMENT", DocumentFont, Brushes.Red, 100, 100)
      e.Graphics.DrawString("Body OF THE DOCUMENT", DocumentFont, Brushes.Black, 100, 130)
   End Sub
   Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Preview.Click
      PrintPreviewControl1.Document = PrintDocument1
   End Sub
   Private Sub PrintPreviewControl1_Click(sender As Object, e As EventArgs) Handles PrintPreviewControl1.Click
   End Sub
   Private Sub Button2_Click_1(sender As Object, e As EventArgs) Handles Button2.Click
```

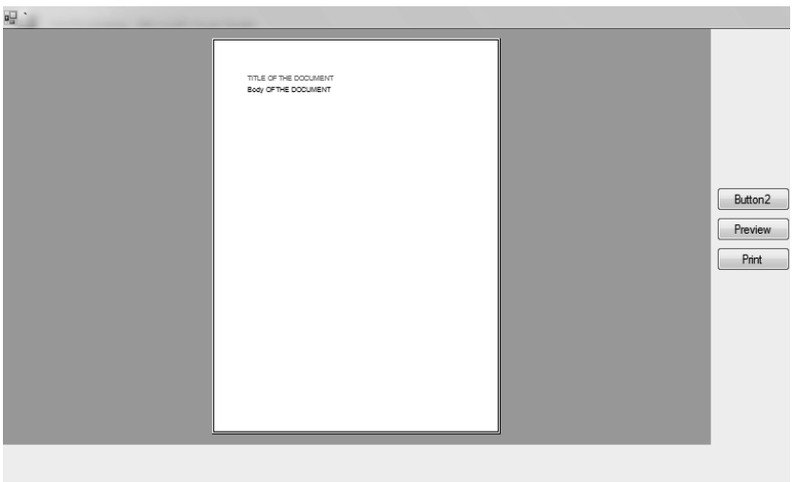
```
        PrintDocument1.DocumentName = Convert.
ToString((OpenFileDialog1.FileName))
   End Sub
End Class
```

NOTES

Step 5: Build and start the project to run the implementation PrintDialog as shown below:

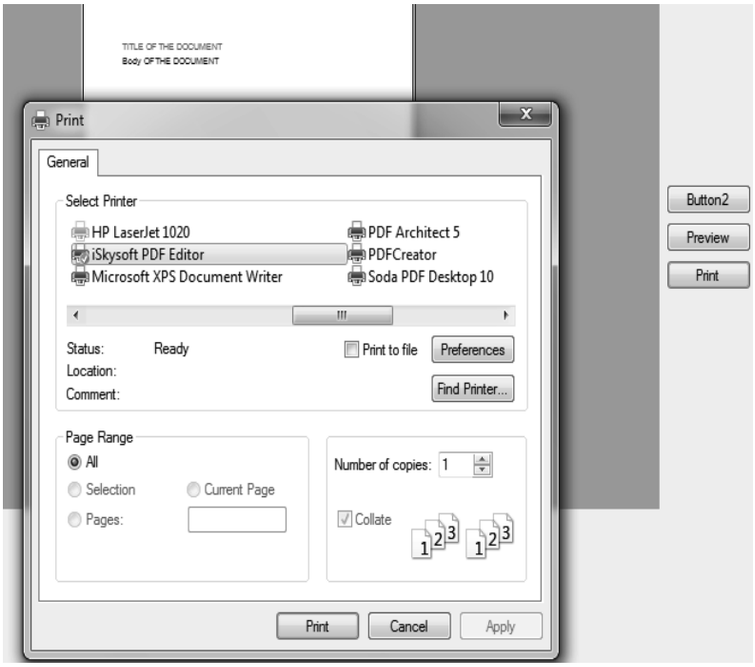


Next click on Preview button to see the preview of the document as shown below:



Next click on Print button to specify the print to take print out the document as shown below:

NOTES



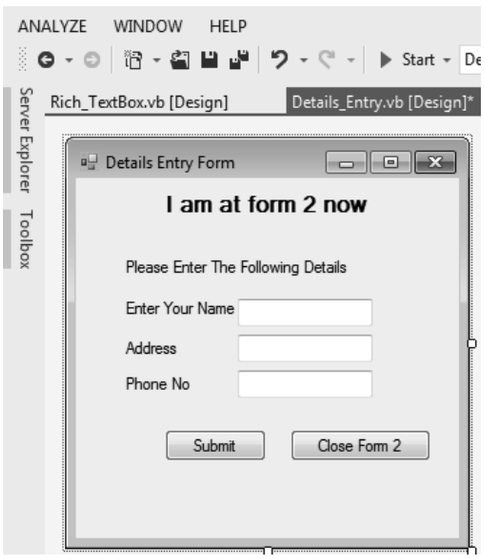
After selecting the printer click on Print to execute the print operation.

24. Write a program to demonstrate the implementation of Rich TextBoxes in VB.

Step 1: Create New Windows Form Application "RichTextBox".

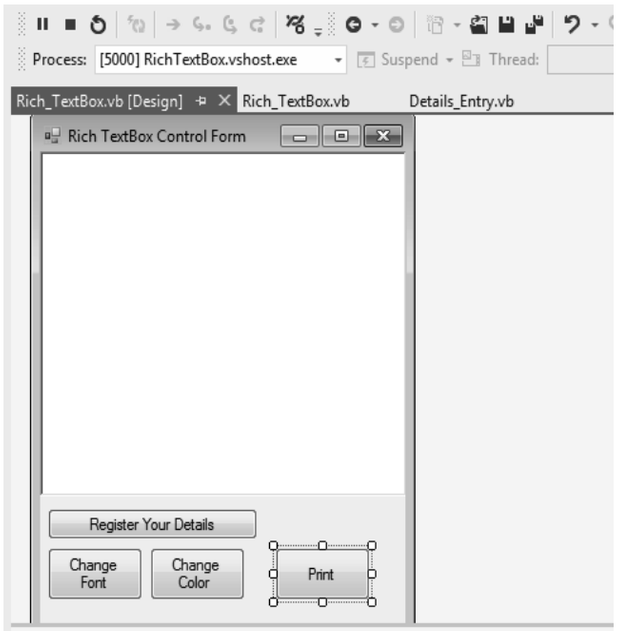
Step 2: Right click on "RichTextBox" in solution explorer to add two Windows form that is Details_Entry.vb and Rich_TextBox.vb.

Step 3: On Details_Entry.vb[Design] form add controls as shown from toolbox, the same is shown in figure below:



Step 4: On Rich_TextBox.vb[Design] form add RichTextBox control and other shown controls(Labels, Buttons) on form from toolbox, the same is shown in figure below:

NOTES



Step 5: Customize the controls specified on Rich_TextBox.vb as shown below:

```
'Rich_TextBox.vb
'Program to demonestrate the implementation of Multiple Forms and using FontDialog and ColorDialog
'box to manupliate the text contained within the richtextbox
Public Class Rich_TextBox
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      If FontDialog1.ShowDialog <> Windows.Forms.DialogResult.Cancel Then
         'RichTextBox1.Font = FontDialog1.Font 'used to change the font of all the text contained within the richtextbox
         RichTextBox1.SelectionFont = FontDialog1.Font 'Used to change font of the selected text from within the Rich Textbox
      End If
   End Sub
   Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
```

NOTES

```
     If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
     RichTextBox1.SelectionColor = ColorDialog1.Color
'Used to change Color of the selected text from within
the Rich Textbox
     End If
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
     PrintDocument1.Print()
   End Sub
   Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
     Details_Entry.Show()
   End Sub
   Private Sub RichTextBox1_TextChanged(sender As Object,
e As EventArgs) Handles RichTextBox1.TextChanged
   End Sub
End Class
```

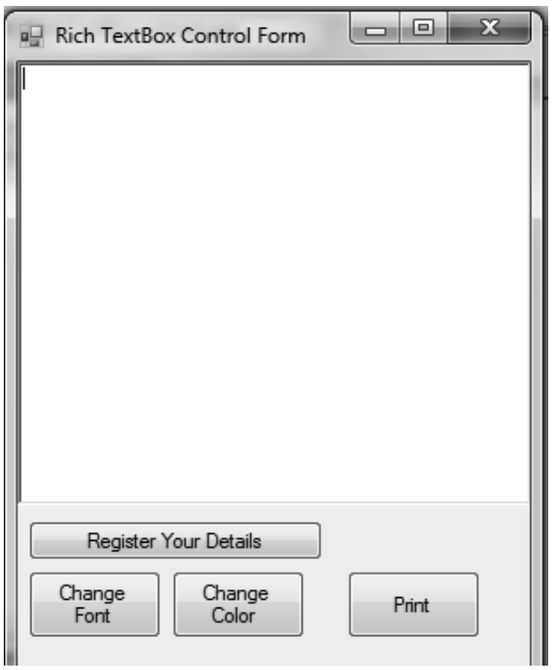
Step 6: Customize the controls specified on Details_Entry.vb as shown below:

```
'Rich_TextBox.vb
'Program to demonestrate the implementation of Multiple
Forms and using FontDialog and ColorDialog
'box to manupliate the text contained within the
richtextbox
Public Class Rich_TextBox
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles ChangeFont.Click
     If FontDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
        'RichTextBox1.Font = FontDialog1.Font 'used to
change the font of all the text contained within the
richtextbox
       RichTextBox1.SelectionFont = FontDialog1.Font
'Used to change font of the selected text from within
the Rich Textbox
     End If
   End Sub
   Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
```

NOTES

```
        If ColorDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
            ' RichTextBox1.ForeColor = ColorDialog1.Color
'used to change the color of all the text contained
within the richtextbox
            RichTextBox1.SelectionColor = ColorDialog1.Color
'Used to change Color of the selected text from within
the Rich Textbox
        End If
    End Sub
    Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
        PrintDocument1.Print()
    End Sub
    Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
        Details_Entry.Show()
    End Sub
    Private Sub RichTextBox1_TextChanged(sender As Object,
e As EventArgs) Handles RichTextBox1.TextChanged
    End Sub
End Class
```

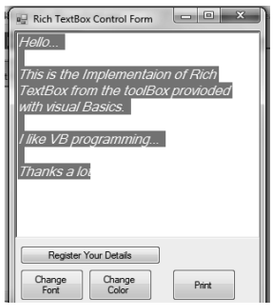
Step 7: Build and start the project RichTextBox to observe the implementation of RichTextBox, The same is shown in figure below:

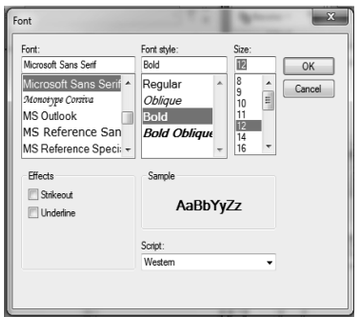


Step 8: Enter any text in the RichTextBox as shown below:

NOTES

Next you can change the Color and Font, the text edited in RichTextBox is shown below:





Select the portion or all the text from the RichTextBox then select the Font and click Ok to see the effects on the text as shown below:

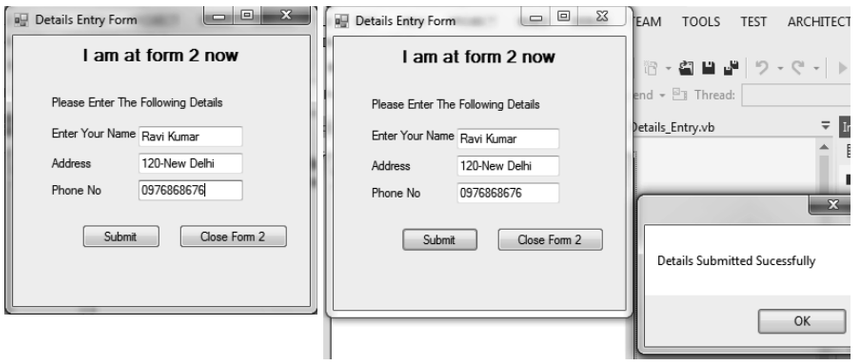
NOTES

To change the color, select the text from within the RichTextBox then select the color you would like then click Ok to see the effect as shown below:

Next click "Change Color" button, select the color as shown below:

Step 9: Click on "Register Your Details" button to enter the details as shown below:

Next click 'Submit' after filling the required details as shown below:



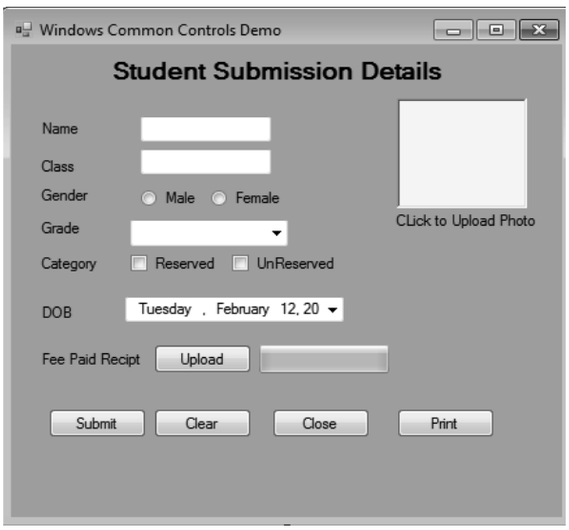
Next click Ok and close current form by clicking on button "Close Form 2" as shown in above figure.

25. Write a program to illustrate the concept of Windows common controls.

Step 1: Create a new Project (Windows Forms Application) "CommonContol1".

Step 2: Right click on "CommonControls1" in Solution Explorer and add form Common_Controls.vb.

Step 3: Design Common_Controls.vb by selecting various Windows controls form toolbox as shown below:



Step 4: Customize Common_Controls.vb by writing code against each control to functionalize the form as shown below:

```
'Common_Controls.vb
'Program to demonastrate the implementation of various
windows controls in vb
```

NOTES

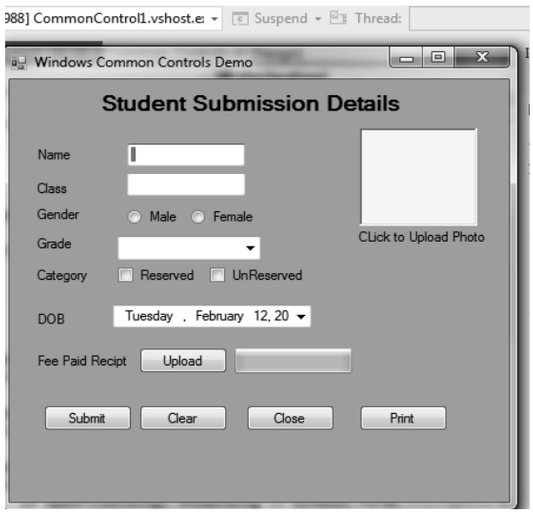
NOTES

```
Public Class Common_Controls
   Private Sub ListBox1_SelectedIndexChanged(sender As
Object, e As EventArgs)
   End Sub
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      If OpenFileDialog1.ShowDialog() = Windows.Forms.
DialogResult.Cancel Then
         OpenFileDialog1.ShowDialog()
      End If
      ProgressBar1.Minimum = 0
      ProgressBar1.MarqueeAnimationSpeed = 5
      ProgressBar1.Maximum = 100
      ProgressBar1.Value = 100
      MessageBox.Show("Document Uploaded Sucessfully")
   End Sub
   Private Sub ProgressBar1_Click(sender As Object, e As
EventArgs) Handles ProgressBar1.Click
   End Sub
   Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
      PrintForm1.Print()
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      If ((TextBox1.Text = " ") Or (TextBox2.Text = " "))
Then
         MessageBox.Show("Please enter the details for
the fields provided")
      Else
         MessageBox.Show("Form Details Submitted
Sucessfully")
      End If
   End Sub
   Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
      Me.Controls.Clear()
   End Sub
```

```
   Private Sub Button5_Click(sender As Object, e As
EventArgs) Handles Button5.Click
      Me.Close()
   End Sub
   Private Sub PictureBox1_Click(sender As Object, e As
EventArgs) Handles PictureBox1.Click
      If OpenFileDialog1.ShowDialog <> Windows.Forms.
DialogResult.Cancel Then
         PictureBox1.Image = Image.
FromFile(OpenFileDialog1.FileName)
      End If
   End Sub
   Private Sub Common_Controls_Load(sender As Object, e
As EventArgs) Handles MyBase.Load
   End Sub
End Class
```

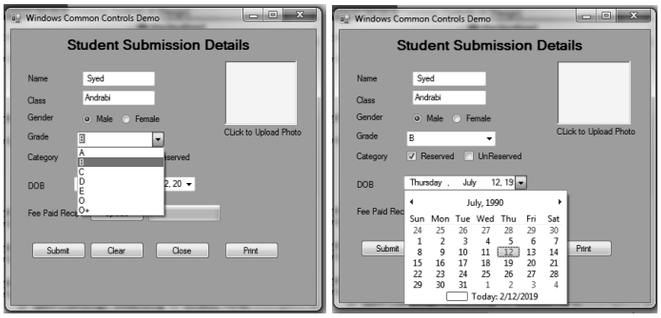
NOTES

Step 5: Build and Start the Project CommonContorl1 to observe the role of these controls as shown below:

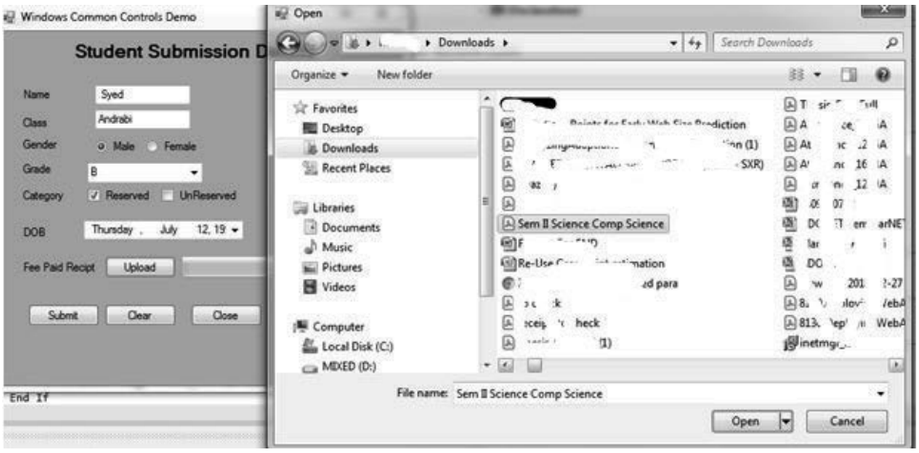


Next enter the details against the controls shown in the form Window:

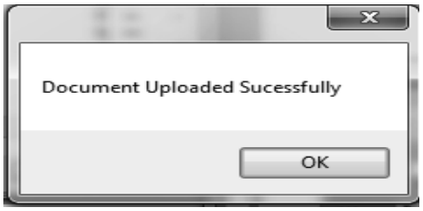
NOTES



Next upload the document as shown below:



Next click on Open to upload.



Next click on photo frame to upload a photo as shown in screenshot below:

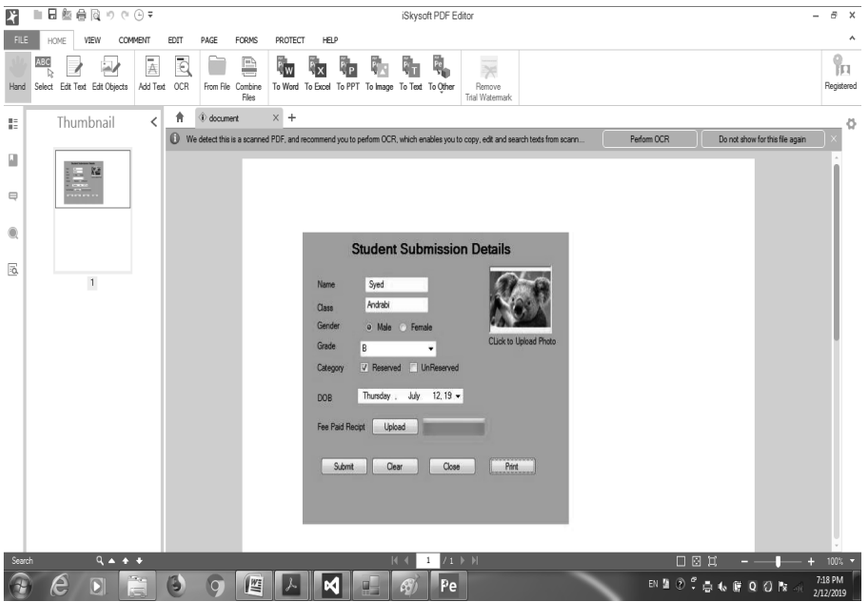
Next select an image and click 'open' the form will look like as shown below:

Next click on 'submit' button to virtually submit the details as shown below:

Next click on 'Print' to take the printout or save as PDF of the above form with filled-in details as shown below:

NOTES

NOTES



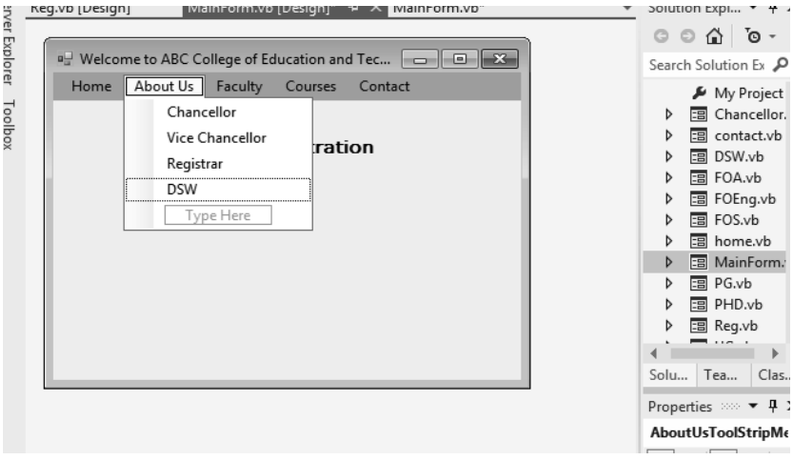
26. Write a program to demonstrate the implementation of Menu Driven Form using VB Menu control.

Step 1: Create new Windows Forms Application "Using_Menu".

Step 2: Add MainForm.vb Windows Form after right click on Using_Menu in solution explorer.

Step 3: Add MenuStrip from toolbox on MainForm.vb[Design] as shown in figure below.

MainForm.vb[Design]



Step 4: Each added Menu and SubMenu is added on the MainForm.vb as shown in above figure (PartialView).

Step 5: Add Windows Forms required to link all the menu and submenu items on the main form. The list of the added forms is shown in solution explorer in the figure above.

Step 6: Double click on every menu item to edit its code behind event. The code behind associated with all the menu items can be edited as MainForm. vb shown below:

NOTES

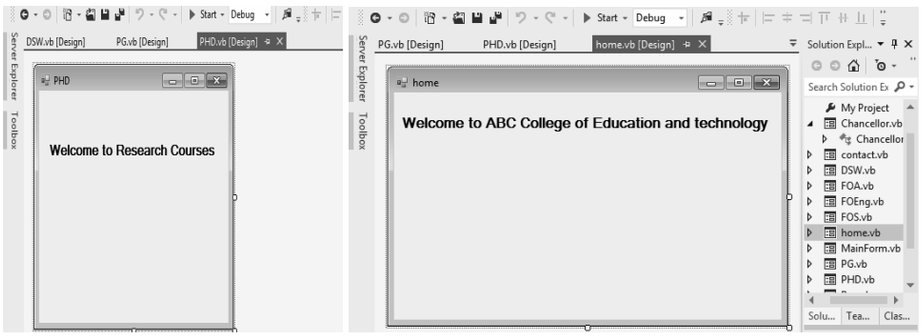
```
'MainForm.vb
'To demonstrate the Implementation of Menu Driven Form
using VB Menu Control
' This is the Main Form with Different Menu Items embedded
.
' Each Menu item is linked to a specific Form
Public Class MainForm
   Private Sub ChancellorToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles ChancellorToolStripMenuItem.
Click
      Chancellor.Show() '  Click takes control to open
Chancellor Form
   End Sub
   Private Sub ViceChancelloeToolStripMenuItem_
Click(sender As Object, e As EventArgs) Handles
ViceChancelloeToolStripMenuItem.Click
      VC.Show()
   End Sub
   Private Sub RToolStripMenuItem_Click(sender As Object,
e As EventArgs) Handles RToolStripMenuItem.Click
      Reg.Show()
   End Sub
   Private Sub DSWToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles DSWToolStripMenuItem.
Click
      DSW.Show()
   End Sub
   Private Sub FacultyOfSciencesToolStripMenuItem_
Click(sender As Object, e As EventArgs) Handles
FacultyOfSciencesToolStripMenuItem.Click
      FOS.Show()
   End Sub
   Private Sub FacultyOfArtsToolStripMenuItem_
Click(sender As Object, e As EventArgs) Handles
FacultyOfArtsToolStripMenuItem.Click
      FOA.Show()
   End Sub
   Private Sub FacultyOfComputerSciencesToolStripMenuItem_
Click(sender As Object, e As EventArgs) Handles
FacultyOfComputerSciencesToolStripMenuItem.Click
```

NOTES

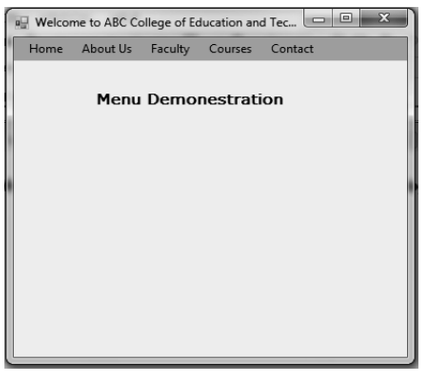
```
        FOEng.Show()
    End Sub
    Private Sub UGToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles UGToolStripMenuItem.
Click
        UG.Show()
    End Sub
    Private Sub PGToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles PGToolStripMenuItem.
Click
        PG.Show()
    End Sub
    Private Sub PHDToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles PHDToolStripMenuItem.
Click
        PHD.Show()
    End Sub
    Private Sub ContactToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles ContactToolStripMenuItem.
Click
        contact.Show()
    End Sub
    Private Sub HomeToolStripMenuItem_Click(sender As
Object, e As EventArgs) Handles HomeToolStripMenuItem.
Click
        home.Show()
    End Sub
    Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
    End Sub
    Private Sub Form1_DoubleClick(sender As Object, e As
EventArgs) Handles MyBase.DoubleClick
        ColorDialog1.ShowDialog()
        Me.BackColor = ColorDialog1.Color
    End Sub
End Class
```

Step 7: All the added forms can be customized by putting controls on them. In this case only a Label Control was added to indicate the status of the linked form with particular menu item as shown below (only few forms):

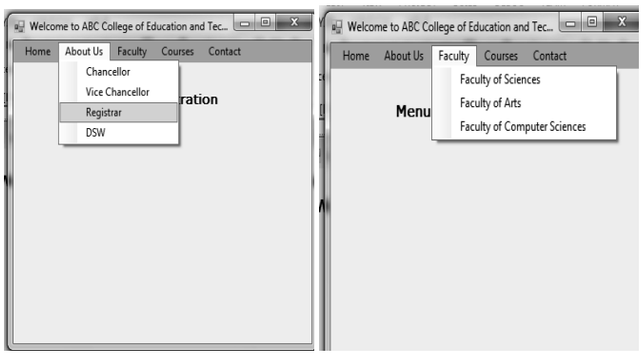
NOTES



Step 8: Build and start the project Using_Menu to see the actual working behaviour of MenuStrip the same is shown in figure below:



Next click on any of the menu items and check for sub menu if any as shown below:



NOTES

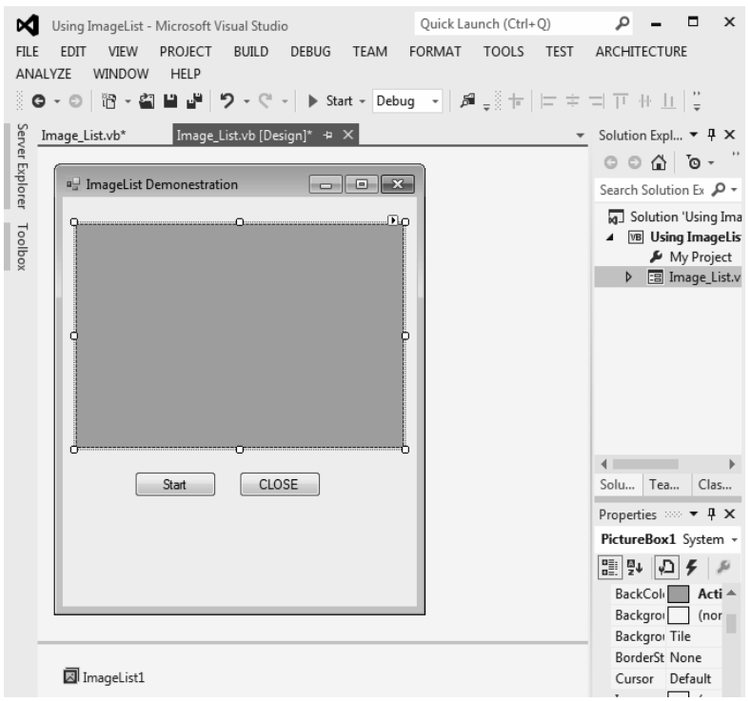
Next click on any menu to check its navigation as shown in figure below: About Us→Registrar.

27. Write a program to demonstrate the implementation of Image List.

Step 1: Create a new Project "Using_ImageList"

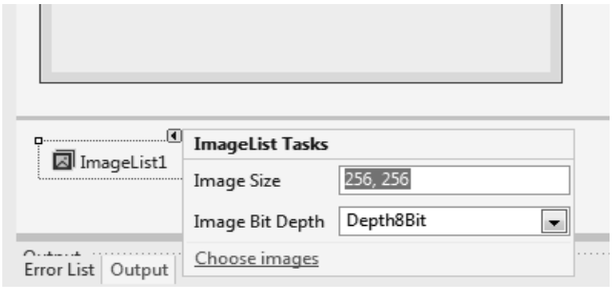
Step 2: Add Image_List.vb to current Project

Step 3: Add controls like Buttons, ImageList and PictureBox from toolbox on Image_List.vb[Design] as shown below:

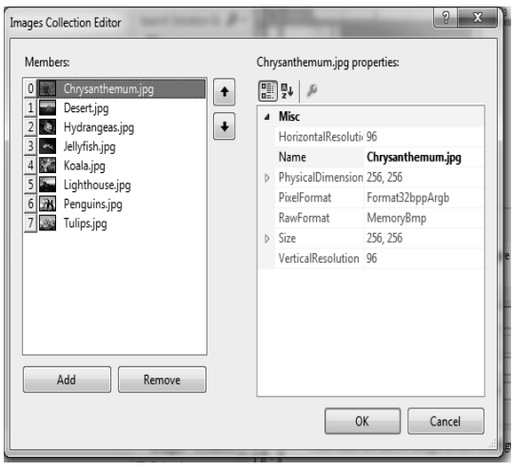


Step 4: Click on ImageList as shown in figure above to link array of images with ImageList1 as shown in figure below:

NOTES



Next click on Choose Images as shown in figure above, the image collection editor will open and prompt you to choose the images from within your computer (by Clicking on Add button) as shown below:



Next after choosing images click ok.

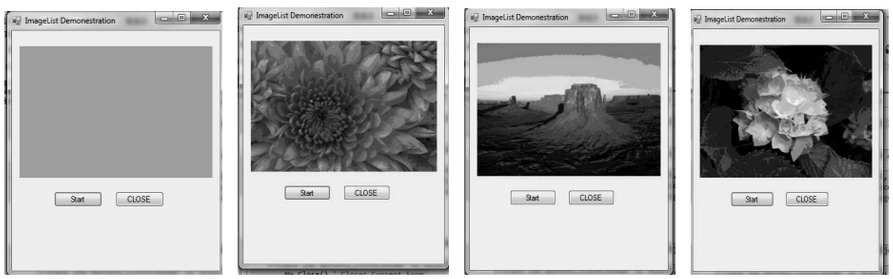
Step 5: Edit Image_List.vb start button by writing following code in its click event.

```
'Image_List.vb
'Program to demonestrate the implemnetation of ImageList
Public Class Image_List
   Dim count As Integer = 0
   Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      PictureBox1.Image = ImageList1.Images(count)
'Assigns Differnt images to image box
      count += 1 ' Count indicates total images in list
      If count = 6 Then
         count = 0
```

NOTES

```
        End If
    End Sub
    Private Sub PictureBox1_Click(sender As Object, e As
EventArgs) Handles PictureBox1.Click
    End Sub
    Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        Me.Close() ' Closes Current Form
    End Sub
End Class
```

Step 6: Build and start the project to see the operational behaviour of Image list by clicking on Start button as shown below:



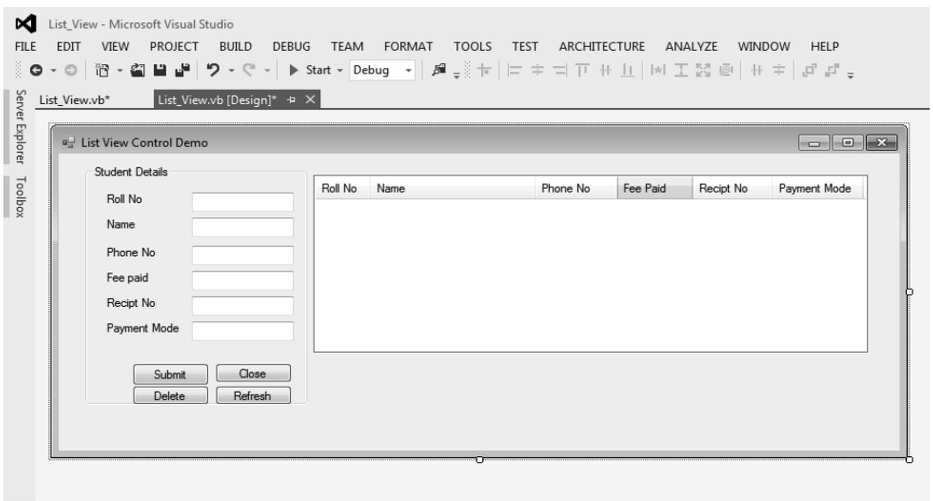
Click Close to exit.

28. Write a program to demonstrate the implementation of list view control in VB.

Step 1: Create a new Project "List_View"

Step 2: Add List_View.vb to current Project

Step 3: Add controls like GroupBox, Textboxes, Buttons, ListView from toolbox on List_View.vb[Design] as shown below:



NOTES

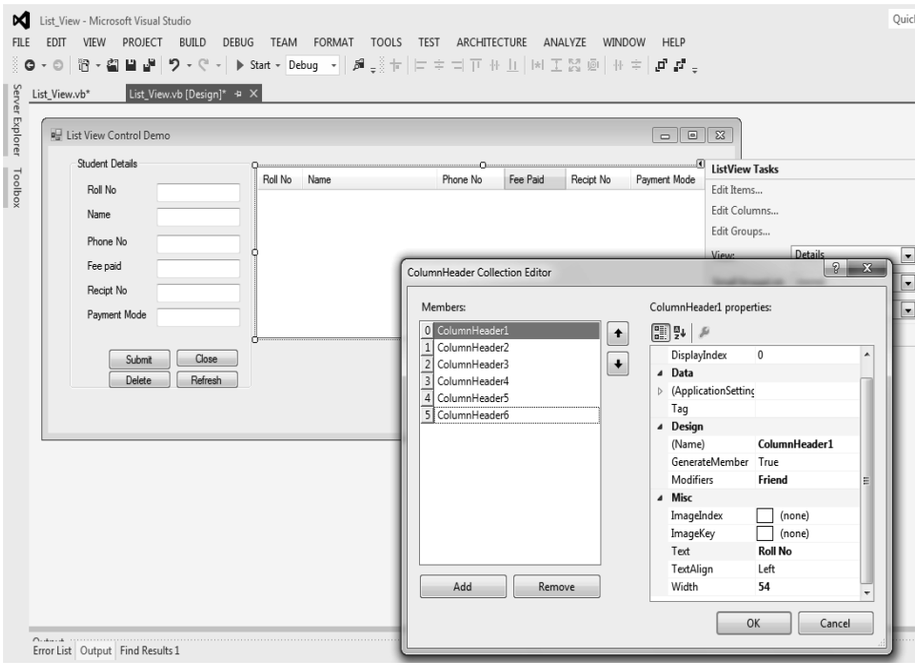
Step 4: Click on ListView1 control as shown above.

Next click on arrow point that appears on ListView1.

Next click on Edit Columns, the ColumnHeader Collection Editor will appear to add Members (Column Headers) as per the requirements (in this case 6)

Next select each ColumnHeader to change its properties like text, name etc from ColumnHeader Properties as shown below:

Click Ok to save changes made to ListView1.



Step 5: Edit the code specific to the controls provided in the List_View. vb[Design] to handle the events associated with them. The same is shown below:

List_View.vb

```
'List_View.vb
'Program to demonestrate the implementation of ListView Control to Display the Student Details entered by a student
Public Class List_View
   Dim currentitem As Integer = 0
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      ListView1.Items.Add(TextBox1.Text) ' Adds the first items to the list view created
```

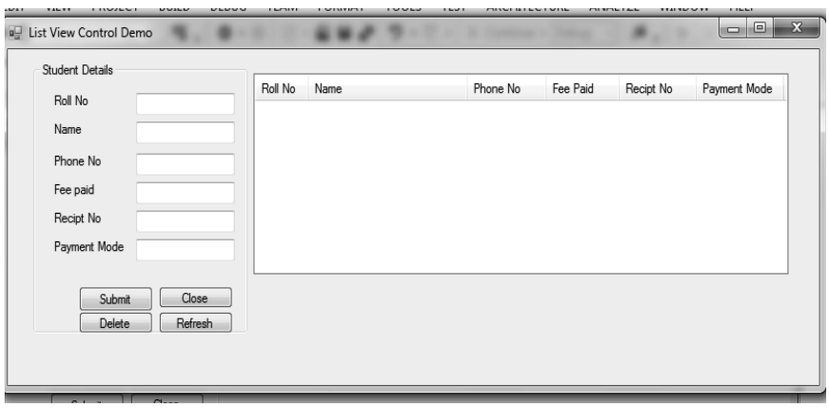
NOTES

```
        ListView1.Items(currentitem).SubItems.
Add(TextBox2.Text) ' adds first subitem to current list
        ListView1.Items(currentitem).SubItems.
Add(TextBox3.Text)
        ListView1.Items(currentitem).SubItems.
Add(TextBox4.Text)
        ListView1.Items(currentitem).SubItems.
Add(TextBox5.Text)
        ListView1.Items(currentitem).SubItems.
Add(TextBox6.Text)
        currentitem += 1
    End Sub
    Private Sub Label1_Click(sender As Object, e As
EventArgs) Handles Label1.Click
    End Sub
    Private Sub TextBox1_TextChanged(sender As Object, e
As EventArgs) Handles TextBox1.TextChanged
    End Sub
    Private Sub Label2_Click(sender As Object, e As
EventArgs) Handles Label2.Click
    End Sub
    Private Sub TextBox2_TextChanged(sender As Object, e
As EventArgs) Handles TextBox2.TextChanged
    End Sub
    Private Sub Label3_Click(sender As Object, e As
EventArgs) Handles Label3.Click
    End Sub
    Private Sub TextBox3_TextChanged(sender As Object, e
As EventArgs) Handles TextBox3.TextChanged
    End Sub
    Private Sub Label4_Click(sender As Object, e As
EventArgs) Handles Label4.Click
    End Sub
    Private Sub TextBox4_TextChanged(sender As Object, e
As EventArgs) Handles TextBox4.TextChanged
    End Sub
    Private Sub Label5_Click(sender As Object, e As
EventArgs) Handles Label5.Click
    End Sub
    Private Sub TextBox5_TextChanged(sender As Object, e
As EventArgs) Handles TextBox5.TextChanged
    End Sub
```

NOTES

```
   Private Sub Label6_Click(sender As Object, e As
EventArgs) Handles Label6.Click
   End Sub
   Private Sub TextBox6_TextChanged(sender As Object, e
As EventArgs) Handles TextBox6.TextChanged
   End Sub
   Private Sub GroupBox1_Enter(sender As Object, e As
EventArgs) Handles GroupBox1.Enter
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      Me.Close() 'Closes the current form window
   End Sub
   Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click
      ListView1.Items.RemoveAt(ListView1.
SelectedIndices(0))
      'RemovesAt() Removes the selected list from the
list view
   End Sub
   Private Sub Button4_Click(sender As Object, e As
EventArgs) Handles Button4.Click
      Me.Refresh() 'Refreshes the listview
   End Sub
   Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
   End Sub
End Class
```

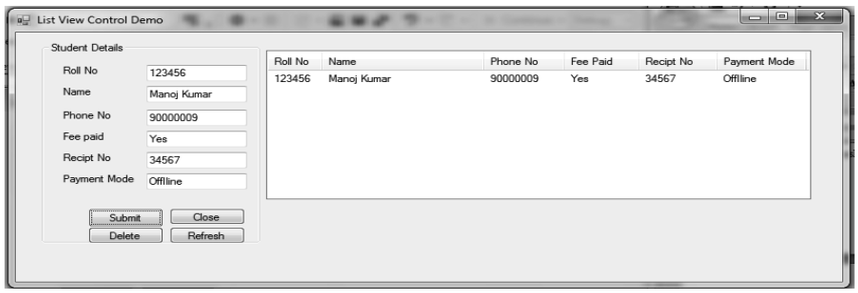
Step 6: Build and Start the Project "List_View" to see the Output. The same is shown in figures provided as under:



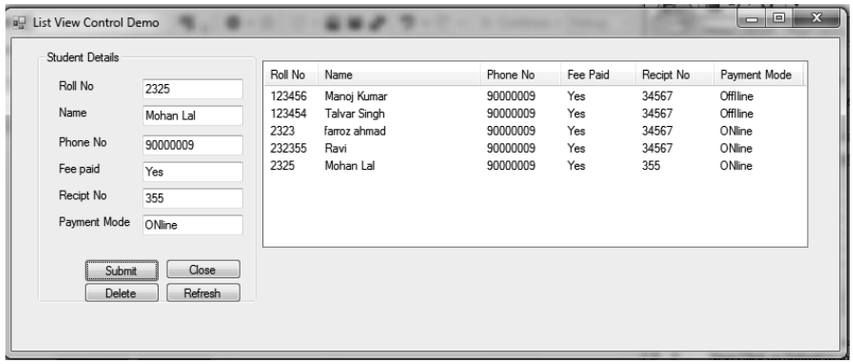
NOTES

Next enter the details as asked in the form window shown above:

Next click on Submit to the entered details into ListView control as shown below:

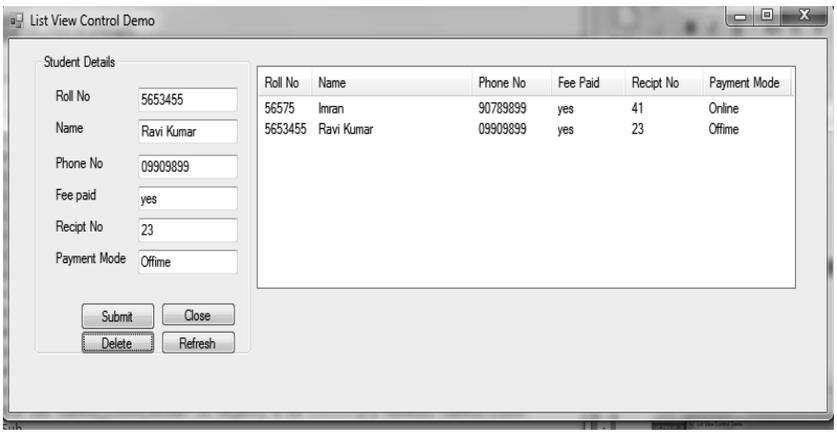


Repeat the entry into fields and submit to fill more ListView items as shown below:



Next you can delete any item in the list view by first selecting a particular Roll No then click button “Delete” as shown below:

NOTES



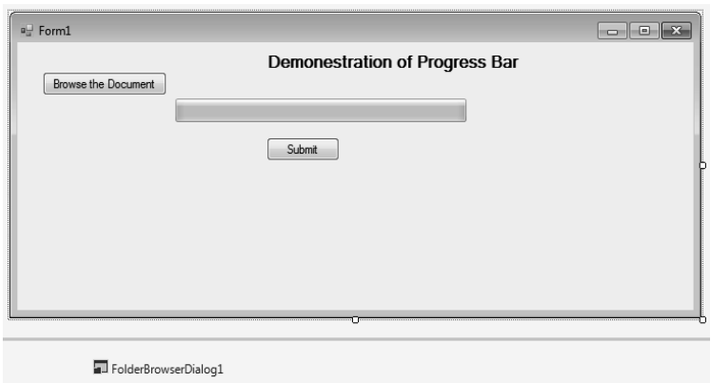
Click Refresh Button to refresh the contents therein ListView and Close button to close the form.

29. Write a program to demonstrate the implementation of Progress Bar in VB.

Step 1: Create a new Project "StatusBarProgbar"

Step 2: Add `Progress_Bar.vb` to current Project

Step 3: Add controls like Button, Label, ProgressBar and FolderBrowserDialog from toolbox on Progress_Bar.vb [Design] as shown below:



Step 4: Edit Progress_Bar.vb to handle the events of the controls provided on the forms window. The same is shown below.

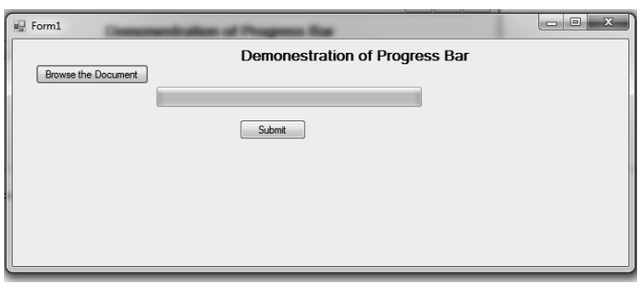
Progress_Bar.vb

```
'Progress_Bar.vb
'Program to demonstrate the implemention of Progress bar
in VB
Public Class Progress_Bar
```

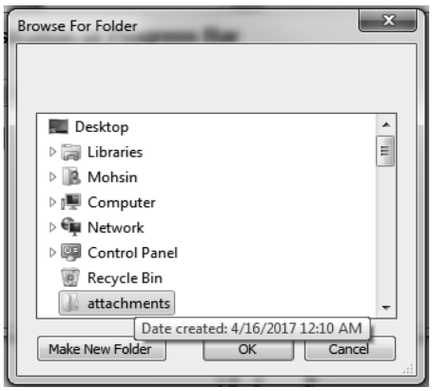
NOTES

```
   Private Sub ProgressBar1_Click(sender As Object, e As
EventArgs) Handles ProgressBar1.Click
   End Sub
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      If (FolderBrowserDialog1.ShowDialog() = DialogResult.
OK) Then ' Open folderBrowserDialog o select file to be
uploaded
         Label1.Text = FolderBrowserDialog1.SelectedPath
' Assigns the name of the file selected to label
      End If
   End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
      ProgressBar1.Minimum = 0 ' Assigns minimum value
to progress bar
      ProgressBar1.Maximum = 200 'Maximum limit of
progress bar
      ProgressBar1.Increment(15) 'Step Increment
      ProgressBar1.Value = 200 ' Bar completion value
      If ProgressBar1.Value = ProgressBar1.Maximum Then
         MsgBox("submited sucessfully")
      End If
   End Sub
   Private Sub Label3_Click(sender As Object, e As
EventArgs)
   End Sub
   Private Sub Form1_Load(sender As Object, e As
EventArgs) Handles MyBase.Load
   End Sub
End Class
```

Step 5: Build the solution and Start the Project. The output window that gets generated is shown below:



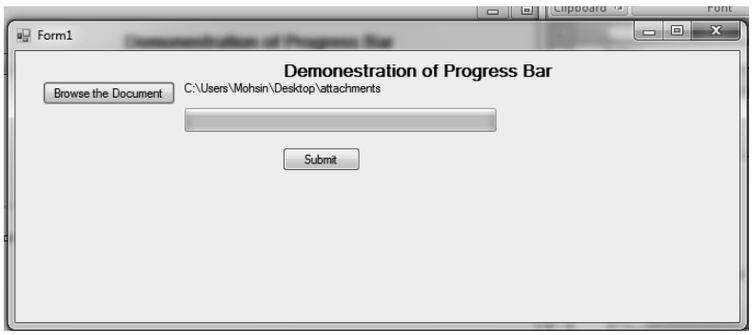
Next Click on button "Browse the Document" to select the document to be uploaded as shown below:



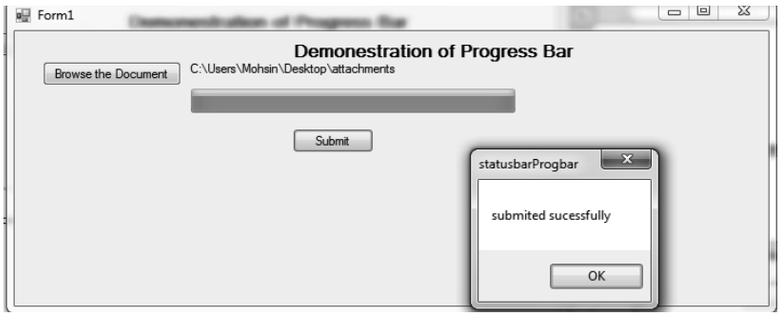
NOTES

After selecting the folder or file, click Ok.

The Location of the selected Folder/file gets reflected on the windows form1 as shown below:



Next click "Submit" button to check the progress of the upload by watching the status of progress bar. Once the successful upload is done the message gets prompted to acknowledge the successful upload the same is shown in figure below:



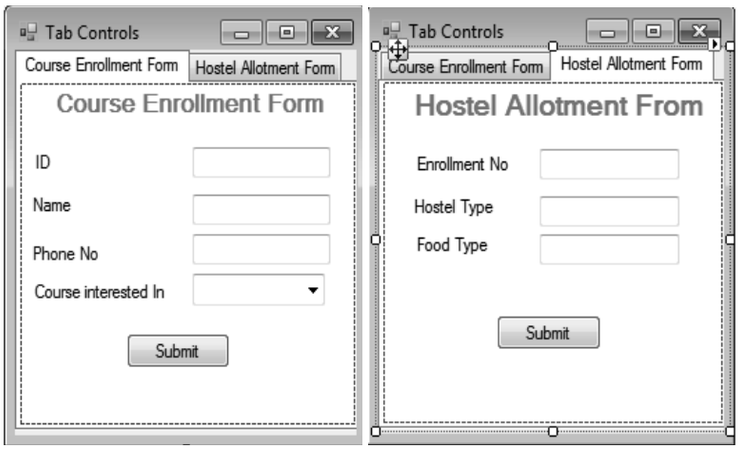
30. Write a program to demonstrate the implementation of Tab Control in VB.

NOTES

Step 1: Create a new Project "Tabcontrols"

Step 2: Add `Tab_Control.vb` to current Project

Step 3: Add controls like TabControl, Button, Labels, TextBoxes and ComboBoxes from toolbox on `Tab_Control.vb`[Design]. Click over the TabControl1 then on the small arrow sign that appers there. Click on Add Tab. The Tab gets added on Tab_Control.vb[Design]. Click over the tab page area to customize its properties in properties window.In this example two tabs were added with text as"Course Enrollmnet Form" and Hostel Allotment Form". The deisgned view of Tab_Control.vb is shown as under:



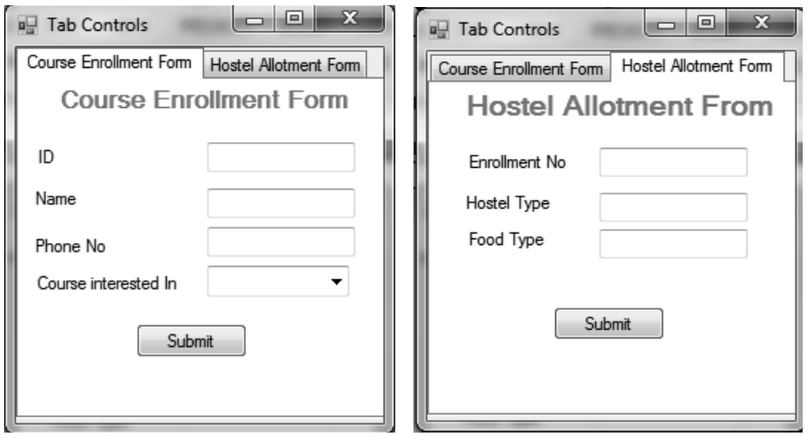
Step 4: Edit Tab_Control.vb as shown below:

```
'Tab_Control.vb
'Program to demonstrate the implementation of Tab Control
in VB
Public Class Tab_Control
   Private Sub TabPage1_Click(sender As Object, e As
EventArgs) Handles TabPage1.Click
   End Sub
   Private Sub Label8_Click(sender As Object, e As
EventArgs) Handles Label8.Click
   End Sub
    'Button1_clicK Handles Event of tab1 that is Course
Enrollment Form Tab
   Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      If TextBox1.Text = "" Then
         MsgBox("Please enter your ID")
      End If
      MsgBox("You have sucessfully Enrolled for Course")
   End Sub
```

```
'Button2_Click Handles Event of tab1 that is Hostel Allotment Form tab
   Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
      If TextBox1.Text = "" Then
         MsgBox("Please Enter your Enrollment number")
      End If
      MsgBox("You have sucessfully Applied for Hostel")
   End Sub
   Private Sub TabPage2_Click(sender As Object, e As EventArgs) Handles TabPage2.Click
   End Sub
End Class
```

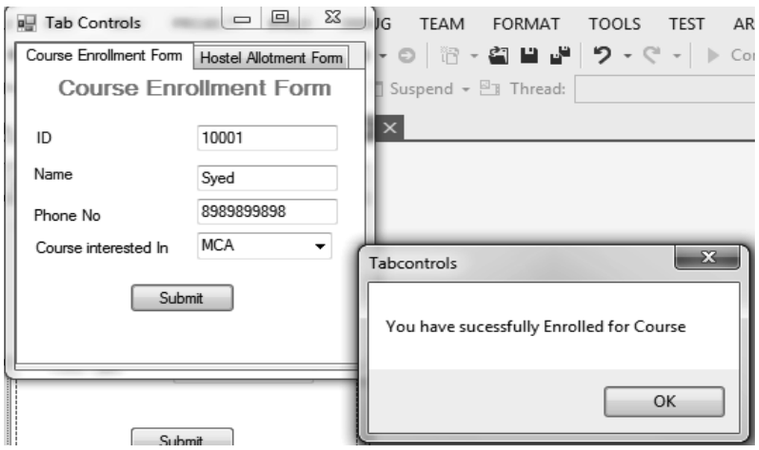
NOTES

Step 5: Build and start the project to see the outcome of Tab_Control.vb. The same is shown below:



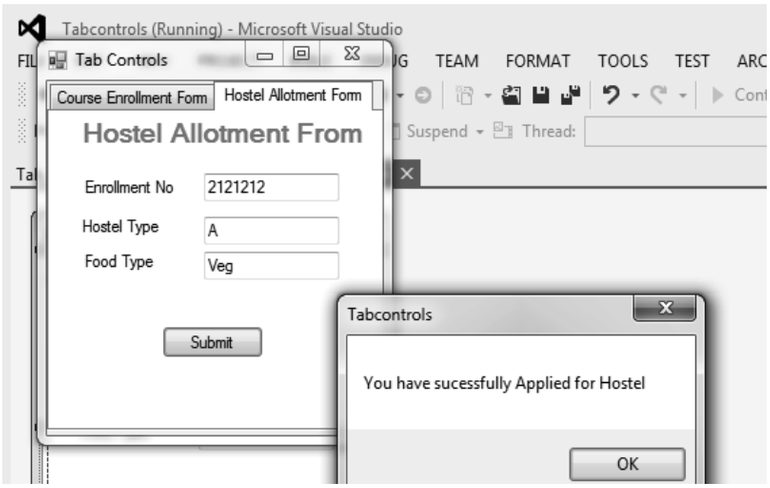
Next enter the details as asked in each tab as shown below:

a. Course Enrollment Form Tab.



NOTES

b. Hostel Allotment Form Tab.

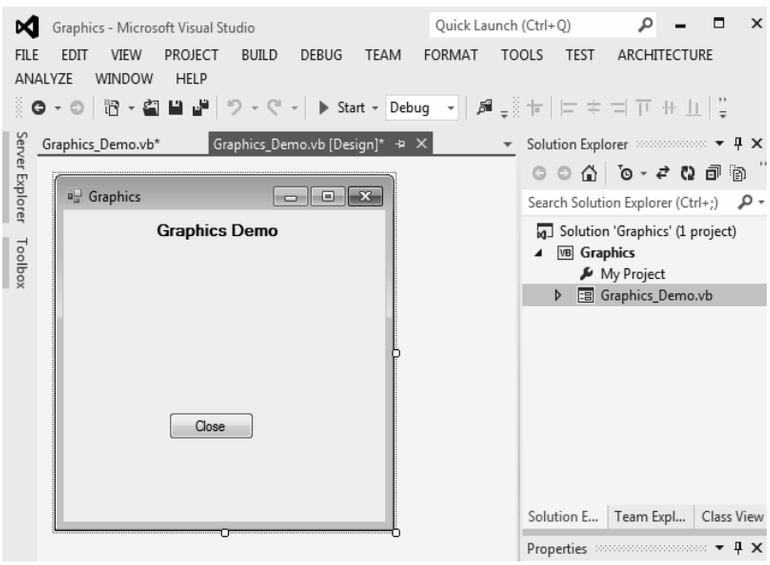


31. Write a program to demonstrate the implementation of Graphics in VB.

Step 1: Create a new Project "Graphics"

Step 2: Add `Graphics_Demo.vb` to current Project

Step 3: Add button control and Label from toolbox on `Graphics_Demo.vb`[Design] as Shown below.



Step 4: Click on form to edit code behind form of Graphics_Demo.vb to implement graphics as shown below:

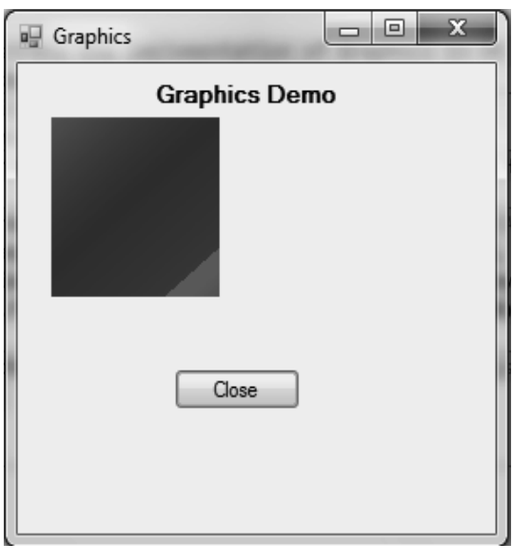
Graphics_Demo.vb

```
'Graphics_Demo.vb
```

NOTES

```
' Program to demonstrate the implementation of Graphics in VB
Imports System.Windows.Forms.PaintEventArgs 'Imports PaintEventArgs to extend to Graphics functionaltiy
Public Class Graphics_Demo
   Dim e As System.Windows.Forms.PaintEventArgs
   Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
   End Sub
   Private Sub Form1_Paint(sender As Object, e As PaintEventArgs) Handles MyBase.Paint
      Dim myRectangle As New Drawing.Rectangle(10, 10, 100, 100) ' drawing Rectangle at coordinates specifed
      Dim myGradient As New Drawing2D.LinearGradientBrush(myRectangle, Color.Red, Color.Blue, 50) ' Filles the gradientfill to designed rectangle
      e.Graphics.FillRectangle(myGradient, 20, 30, 100, 100)
   End Sub
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
      Me.Close()
   End Sub
End Class
```

Step 5: Build the solution and start the project "Graphics" to see the output as shown below:



Next click Close button to close the Graphics_Demo.vb.

32. Write a program to demonstrate the concept of File handling in VB using File Stream and Stream Reader Objects.

NOTES

Step 1: Create a new Project "Graphics"

Step 2: Add `FileHandling.vbvb` to current Project

Step 3: Add button control and label from toolbox on `FileHandling.vb` [Design] as shown below.

Step 4: Click on form to edit code behind form associated with each button control of FileHandling.vb to implement file handling operation as shown below:

FileHandling.vb

```
'FileHandling.vb
'Program to demonestrate the concept of File handling in VB using FileStream and StreamReader Objects
Imports System.IO
Public Class FileHandling
   Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
   End Sub
   Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click 'FileStreamDemo
      Dim f1 As FileStream = New FileStream("sample.txt", FileMode.OpenOrCreate, FileAccess.ReadWrite)
      Dim i As Integer
      For i = 0 To 20
         f1.WriteByte(CByte(i))
```

NOTES

```
      Next i
      f1.Position = 0
      For i = 0 To 20
        MessageBox.Show(+(f1.ReadByte()), "STREAM
READER")
      Next i
      f1.Close()
        End Sub
   Private Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click 'ReadFrmFile
      Try
        ' Create an instance of StreamReader to read from
a file.
        ' The using statement also closes the StreamReader.
        Using sr As StreamReader = New StreamReader("D:\
Dr yoousf paper\table referencing.txt")
          Dim line As String
          ' Read and display lines from the file until
the end of the file is reached.
          line = sr.ReadLine()
          While (line <> Nothing)
          MsgBox((line))
          line = sr.ReadLine()
          End While
        End Using
      Catch
        ' Let the user know what went wrong.
        Console.WriteLine("The file could not be read:")
        Console.WriteLine("err")
      End Try
   End Sub
   Private Sub Button3_Click(sender As Object, e As
EventArgs) Handles Button3.Click 'WriteToFile
      Dim names As String() = New String() {"Alias", _
      "Nuhman Ali", "Virat Kholi", "John Abrahim", "Ajit
Singh"}
      Dim s As String
      Using sw As StreamWriter = New StreamWriter("names.
txt")
```

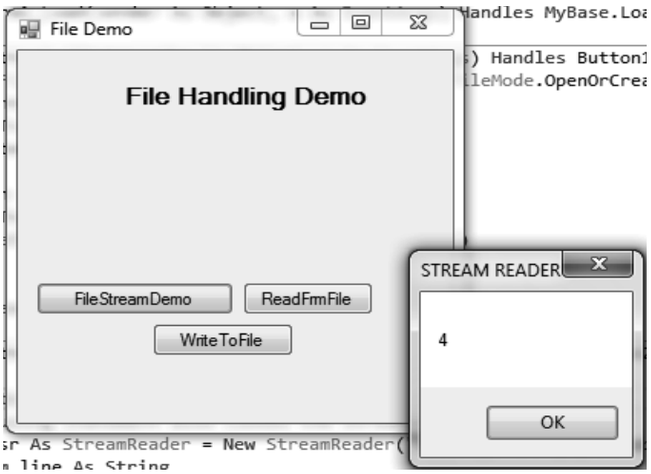
NOTES

```
            For Each s In names
               sw.WriteLine(s)
            Next s
         End Using
         ' Read and show each line from the file.
         Dim line As String
         Using sr As StreamReader = New StreamReader("names.
txt")
            line = sr.ReadLine()
            While (line <> Nothing)
               MessageBox.Show(line)
               line = sr.ReadLine()
            End While
         End Using
      End Sub
End Class
```

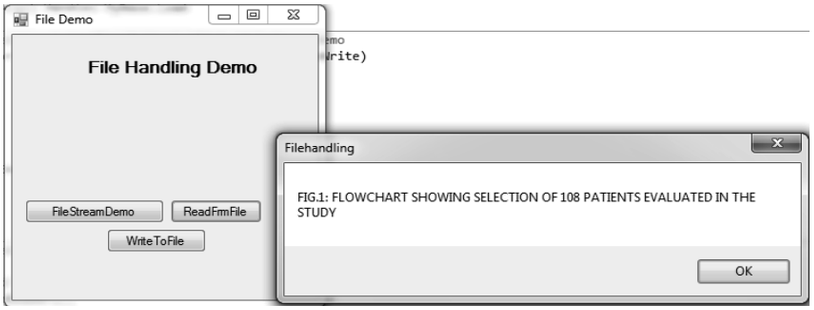
Step 5: build the solution and start the project "Filehandling" to see the output as shown below:

Next click on button "FileStreamDemo" to display File Stream as shown below to show a stream of numbers from 0 to 20:

NOTES



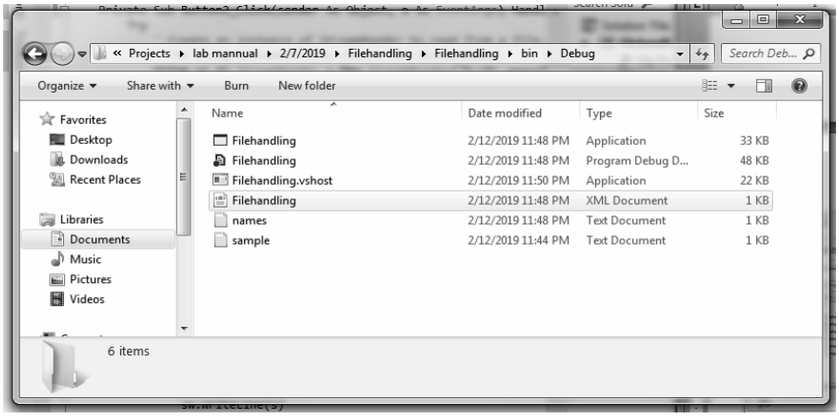
Next click on button "ReadFrmFile" to read contents of a file specified in SteamReader object as shown below:



Next click on button "Write ToFile" to write contents of a into file by StreamWriter object as shown below:

Step 6: Visit Filehandling→bin→Debug to see the files created during this program as shown below:

NOTES



33. Write a program to demonstrate the implementation of various data controls in ASP.NET using VB.

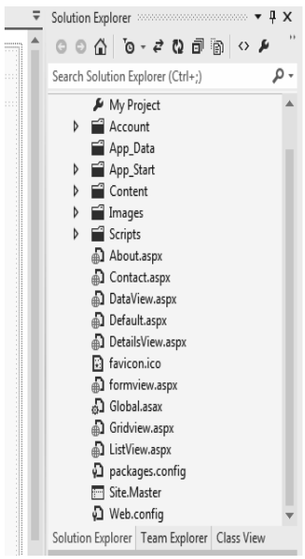
Step 1: Create a ASP.NET based web project using following steps:

File→New→Project→ASP→NET Empty Web Application→DataControls→Ok.

Step 2: Right click on data controls in Solution Explorer, Add Windows Forms, name the form as:

- DataView.aspx
- formView.aspx
- GridView.aspx
- ListView.aspx

As shown in Solution Explorer below:



Step 3: Open each data control form to implement their functionality as shown below:

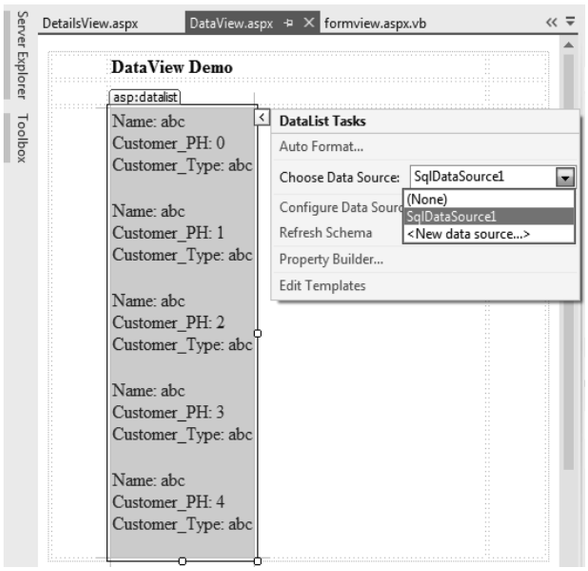
a. DataView.aspx

NOTES

Step I: From Menu, Click on Table and then click Insert Table as per the requirements:

Step II: From ToolBox select and insert into any of the table cell or anywhere on the form the DataView Control.

Step III: Click onr arrow sign appears after you click over DataView Control in DataView.aspx and click on choose data source to link with this DataView Control as shown below:



Step IV: Select datasource from the database you have created using (say) SQLServer. Follow the steps as and when prompted to fullfil connection with desired data source in database.

Step V: Here in this case the database selected contains five rows with three columns as shown in figure above:

Step VI: Before you build the project you need to specify the server to host the project. In order to does that click on project from Menu bar, click DataControl properties option.

Step VII: Click on Web from the options displayed on the left side of the form displayed. Go to Start Action tab, choose specific page to start your project to run. From Servers tab select "Use Visiual Studio Development Server", check "Auto-Assign Port".

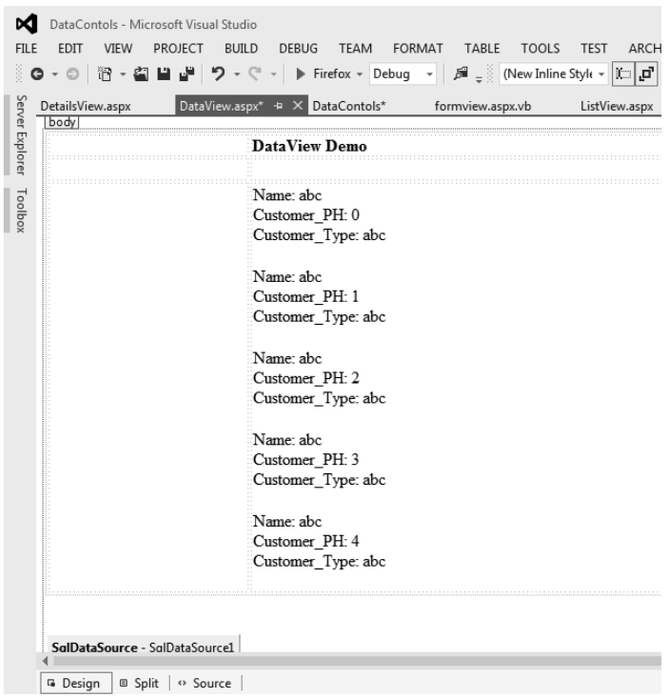
Step VIII: Build the project.

Step IX: Specify the browser to display the outcome of the project.

Step X: If project builds without errors the resultant display will be loaded into a browser specified by the programmer say FireFox in this case.

The Design of DataView.aspx will look like the figure shown below:

NOTES



DataView.aspx source code will look like as given below:

```
<style type="text/css">
   .auto-style1 {
      width: 100%;
      height: 378px;
   }
   .auto-style2 {
      width: 251px;
   }
   .auto-style3 {
      width: 86px;
   }
   .auto-style4 {
      width: 67px;
   }
</style>
<p>
```

NOTES

```
   <table class="auto-style1">
      <tr>
         <td class="auto-style4"> </td>
         <td class="auto-style2"><strong>DataView Demo</
strong></td>
         <td class="auto-style3"> </td>
      </tr>
      <tr>
         <td class="auto-style4"> </td>
         <td class="auto-style2"> </td>
         <td class="auto-style3"> </td>
      </tr>
      <tr>
         <td class="auto-style4"> </td>
         <td class="auto-style2"><asp:datalist
runat="server" DataSourceID="SqlDataSource1">
            <ItemTemplate>
            Name:
            <asp:Label ID="NameLabel" runat="server"
Text='<%# Eval("Name") %>' />
            <br />
            Customer_PH:
            <asp:Label ID="Customer_PHLabel" runat="server"
Text='<%# Eval("Customer_PH") %>' />
            <br />
            Customer_Type:
            <asp:Label ID="Customer_TypeLabel"
runat="server" Text='<%# Eval("Customer_Type") %>' />
            <br />
<br />
            </ItemTemplate>
            </asp:datalist>
         </td>
         <td class="auto-style3"> </td>
      </tr>
   </table>
   <br />
</p>
```

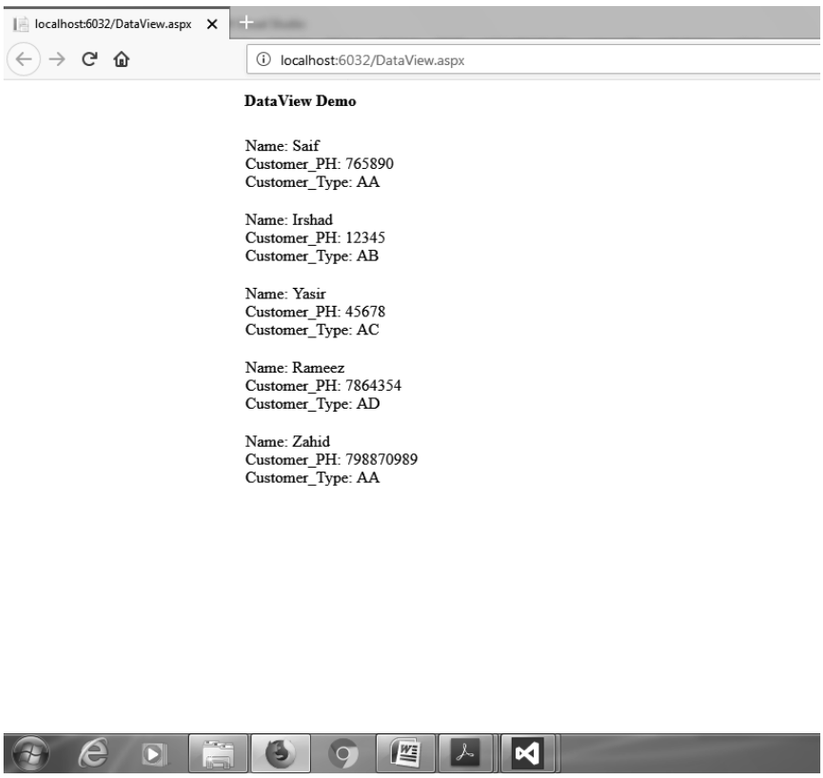
NOTES

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:CustomerDetailConnectionString %>" SelectCommand="SELECT * FROM [Cust_Det]"></asp:SqlDataSource>
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="DataView.aspx.vb" Inherits="DataContols.DataView" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <div>

   </div>
   </form>
</body>
</html>
```

DataView.aspx.vb

```
'Program to implement DataView Control in ASP.NET
Public Class DataView
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
   End Sub
End Class
```

Step 4: After successful build and start the output window obtained is shown in figure below:

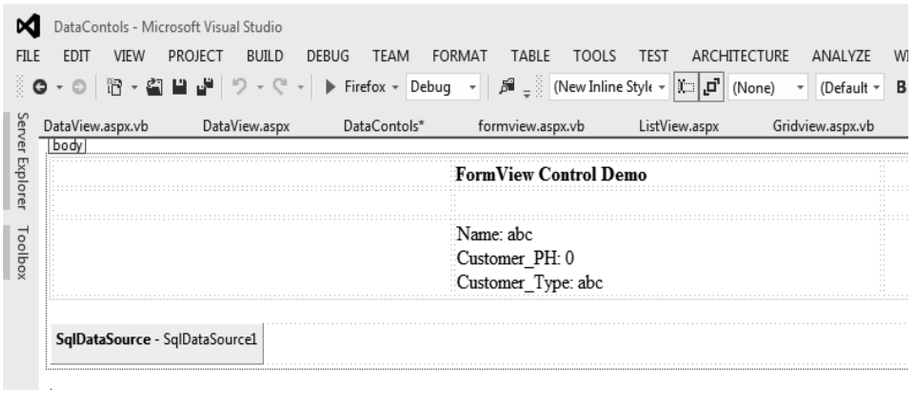


Similarly the other DataControls can be used to implement their functionality in your ASP.NET web project.

34. Write a program to use FormView data control.

Step 1: Follow similar steps as discussed above for DataView Control. However, instead of DataView Control you need to Use FormView Control from toolbox.

Step 2: Choose data source for FormView Control and design the windows from "formview.aspx" as shown below:



Source code of formview.aspx is given below:

NOTES

formview.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="formview.aspx.vb" Inherits="DataContols.
formview" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
   <title></title>
<style type="text/css">
   .auto-style1 {
      width: 100%;
   }
   .auto-style2 {
      width: 368px;
   }
</style>
   </head>
<body>
   <form id="form1" runat="server">
      <div >
<table class="auto-style1">
   <tr>
      <td> </td>
      <td class="auto-style2"><strong>FormView Control
Demo</strong></td>
      <td> </td>
   </tr>
   <tr>
      <td> </td>
      <td class="auto-style2"> </td>
      <td> </td>
   </tr>
   <tr>
      <td> </td>
      <td class="auto-style2">
         <asp:FormView ID="FormView1" runat="server"
DataSourceID="SqlDataSource1" Height="78px">
            <EditItemTemplate>
```

```
            Name:
            <asp:TextBox ID="NameTextBox" runat="server"
Text='<%# Bind("Name") %>' />
            <br />
            Customer_PH:
            <asp:TextBox ID="Customer_PHTextBox"
runat="server" Text='<%# Bind("Customer_PH") %>' />
            <br />
            Customer_Type:
            <asp:TextBox ID="Customer_TypeTextBox"
runat="server" Text='<%# Bind("Customer_Type") %>' />
            <br />
            <asp:LinkButton ID="UpdateButton"
runat="server" CausesValidation="True"
CommandName="Update" Text="Update" />
             <asp:LinkButton ID="UpdateCancelButton"
runat="server" CausesValidation="False"
CommandName="Cancel" Text="Cancel" />
            </EditItemTemplate>
            <InsertItemTemplate>
            Name:
            <asp:TextBox ID="NameTextBox" runat="server"
Text='<%# Bind("Name") %>' />
            <br />
            Customer_PH:
            <asp:TextBox ID="Customer_PHTextBox"
runat="server" Text='<%# Bind("Customer_PH") %>' />
            <br />
            Customer_Type:
            <asp:TextBox ID="Customer_TypeTextBox"
runat="server" Text='<%# Bind("Customer_Type") %>' />
            <br />
            <asp:LinkButton ID="InsertButton"
runat="server" CausesValidation="True"
CommandName="Insert" Text="Insert" />
             <asp:LinkButton ID="InsertCancelButton"
runat="server" CausesValidation="False"
CommandName="Cancel" Text="Cancel" />
            </InsertItemTemplate>
            <ItemTemplate>
```

NOTES

NOTES

```
            Name:
            <asp:Label ID="NameLabel" runat="server"
Text='<%# Bind("Name") %>' />
            <br />
            Customer_PH:
            <asp:Label ID="Customer_PHLabel" runat="server"
Text='<%# Bind("Customer_PH") %>' />
            <br />
            Customer_Type:
            <asp:Label ID="Customer_TypeLabel"
runat="server" Text='<%# Bind("Customer_Type") %>' />
            <br />
            </ItemTemplate>
        </asp:FormView>
        <asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:CustomerDetailConnectionString
%>" SelectCommand="SELECT * FROM [Cust_Det]"></
asp:SqlDataSource>
     </td>
     <td> </td>
   </tr>
</table>
</div>
     </form>
   </body>
</html>
```

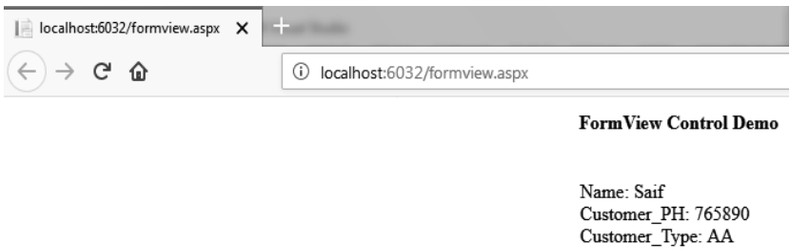
Code behind formview.aspx that is formview.aspx.vb is given below:

```
'Implementation of FormView DataControl in asp.net using
VB
Public Class formview
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   Protected Sub SqlDataSource1_Selecting(sender As
Object, e As SqlDataSourceSelectingEventArgs) Handles
SqlDataSource1.Selecting
```

```
    End Sub
End Class
```

Step 3: Build and run the project. The output generated is shown in figure below:

NOTES

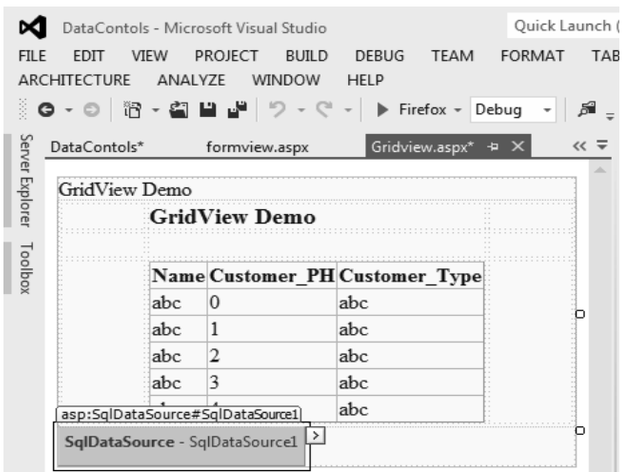


Note: FormView data control displays only a single row retrieved from the linked data source into the browser window as shown above.

35. Write a program to use GridView data control.

Step 1: Follow similar steps as discussed above for DataView Control. However, instead of DataView Control you need to Use GridViewControl from toolbox.

Step 2: Choose data source for to be linked with GridView Control and design the windows from "Gridview.aspx" as shown below:



NOTES

Source code of Gridview.aspx is given below:

```
'Gridview.aspx
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="Gridview.aspx.vb" Inherits="DataContols.
Gridview" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style1 {
         width: 100%;
      }
      .auto-style2 {
         width: 206px;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div >

      <asp:Label ID="Label1" runat="server" Text="GridView
Demo" ></asp:Label>

      <br />
      <table class="auto-style1">
         <tr>
            <td> </td>
            <td class="auto-style2"><strong ><h3>GridView
Demo</h3>
            </strong></td>
            <td> </td>
         </tr>
         <tr>
            <td> </td>
            <td class="auto-style2"> </td>
```

NOTES

```
            <td> </td>
        </tr>
        <tr>
            <td> </td>
            <td class="auto-style2">
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False" DataSourceID="SqlDataSource1">
                <Columns>
                    <asp:BoundField DataField="Name" HeaderText="Name" SortExpression="Name" />
                    <asp:BoundField DataField="Customer_PH" HeaderText="Customer_PH" SortExpression="Customer_PH" />
                    <asp:BoundField DataField="Customer_Type" HeaderText="Customer_Type" SortExpression="Customer_Type" />
                </Columns>
            </asp:GridView>
            </td>
            <td> </td>
        </tr>
    </table>
    <asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%$ ConnectionStrings:CustomerDetailConnectionString %>" SelectCommand="SELECT * FROM [Cust_Det]"></asp:SqlDataSource>

  </div>
  </form>
</body>
</html>
```

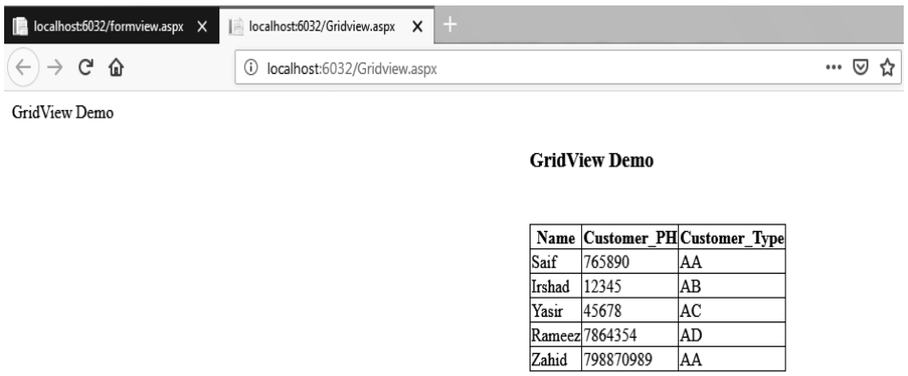
Code behind Gridview.aspx that is Gridview.aspx.vb is given below:

```
'Gridview.aspx.vb
'Implementation of GridView DataControl in asp.net using VB
Public Class Gridview
  Inherits System.Web.UI.Page
```

NOTES

```
    Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load

    End Sub

    Protected Sub SqlDataSource1_Selecting(sender As
Object, e As SqlDataSourceSelectingEventArgs) Handles
SqlDataSource1.Selecting

    End Sub

    Protected Sub GridView1_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles GridView1.
SelectedIndexChanged

    End Sub

End Class
```

Step 3: Build and run the project. The output generated is shown in figure given below:

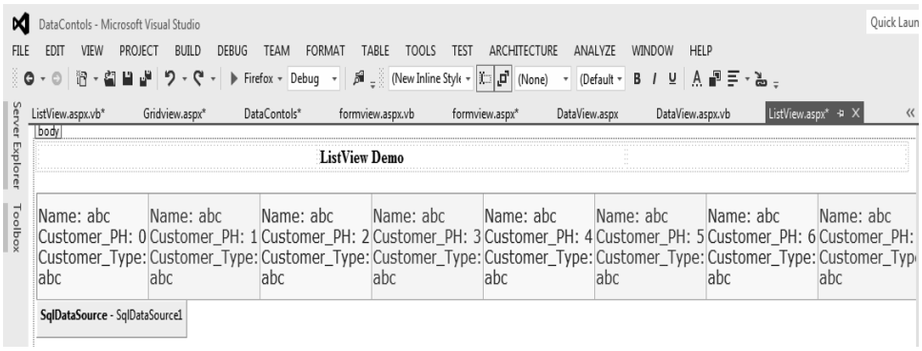


36. Write a program to use ListView data control.

Step 1: Follow similar steps as discussed above for DataView Control. However, instead of DataView Control you need to Use ListViewControl from toolbox.

Step 2: Choose data source for to be linked with ListView Control and design the windows from "ListView.aspx" as shown below:

NOTES



Source code of ListView.aspx is given below:

```
'ListView.aspx
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="ListView.aspx.vb" Inherits="DataContols.
ListView" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
   .auto-style1 {
      width: 100%;
   }
   .auto-style2 {
      width: 374px;
   }
</style>
<p>
   <table class="auto-style1">
      <tr>
         <td> </td>
         <td class="auto-style2"><strong>ListView Demo</
strong></td>
         <td> </td>
      </tr>
   </table>
</p>
<asp:listview runat="server" DataSourceID="SqlDataSource1"
OnSelectedIndexChanged="Unnamed1_SelectedIndexChanged">
   <AlternatingItemTemplate>
```

NOTES

```
      <td runat="server" style="background-color:
#FAFAD2;color: #284775;">Name:
        <asp:Label ID="NameLabel" runat="server"
Text='<%# Eval("Name") %>' />
        <br />
        Customer_PH:
        <asp:Label ID="Customer_PHLabel" runat="server"
Text='<%# Eval("Customer_PH") %>' />
        <br />
        Customer_Type:
        <asp:Label ID="Customer_TypeLabel" runat="server"
Text='<%# Eval("Customer_Type") %>' />
        <br />
      </td>
    </AlternatingItemTemplate>
    <EditItemTemplate>
      <td runat="server" style="background-color:
#FFCC66;color: #000080;">Name:
        <asp:TextBox ID="NameTextBox" runat="server"
Text='<%# Bind("Name") %>' />
        <br />
        Customer_PH:
        <asp:TextBox ID="Customer_PHTextBox"
runat="server" Text='<%# Bind("Customer_PH") %>' />
        <br />
        Customer_Type:
        <asp:TextBox ID="Customer_TypeTextBox"
runat="server" Text='<%# Bind("Customer_Type") %>' />
        <br />
        <asp:Button ID="UpdateButton" runat="server"
CommandName="Update" Text="Update" />
        <asp:Button ID="CancelButton" runat="server"
CommandName="Cancel" Text="Cancel" />
      </td>
    </EditItemTemplate>
    <EmptyDataTemplate>
      <table style="background-color: #FFFFFF;border-
collapse: collapse;border-color: #999999;border-
style:none;border-width:1px;">
```

```
        <tr>
           <td>No data was returned.</td>
        </tr>
     </table>
  </EmptyDataTemplate>
  <InsertItemTemplate>
     <td runat="server" style="">Name:
        <asp:TextBox ID="NameTextBox" runat="server"
Text='<%# Bind("Name") %>' />
        <br />Customer_PH:
        <asp:TextBox ID="Customer_PHTextBox"
runat="server" Text='<%# Bind("Customer_PH") %>' />
        <br />Customer_Type:
        <asp:TextBox ID="Customer_TypeTextBox"
runat="server" Text='<%# Bind("Customer_Type") %>' />
        <br />
        <asp:Button ID="InsertButton" runat="server"
CommandName="Insert" Text="Insert" />
        <asp:Button ID="CancelButton" runat="server"
CommandName="Cancel" Text="Clear" />
     </td>
  </InsertItemTemplate>
  <ItemTemplate>
     <td runat="server" style="background-color:
#FFFBD6;color: #333333;">Name:
        <asp:Label ID="NameLabel" runat="server"
Text='<%# Eval("Name") %>' />
        <br />
        Customer_PH:
        <asp:Label ID="Customer_PHLabel" runat="server"
Text='<%# Eval("Customer_PH") %>' />
        <br />
        Customer_Type:
        <asp:Label ID="Customer_TypeLabel" runat="server"
Text='<%# Eval("Customer_Type") %>' />
        <br />
     </td>
  </ItemTemplate>
  <LayoutTemplate>
```

NOTES

NOTES

```
      <table runat="server" border="1" style="background-
color: #FFFFFF;border-collapse: collapse;border-color:
#999999;border-style:none;border-width:1px;font-family:
Verdana, Arial, Helvetica, sans-serif;">
        <tr id="itemPlaceholderContainer" runat="server">
          <td id="itemPlaceholder" runat="server"></td>
        </tr>
      </table>
      <div style="text-align: center;background-color:
#FFCC66;font-family: Verdana, Arial, Helvetica, sans-
serif;color: #333333;">
      </div>
   </LayoutTemplate>
   <SelectedItemTemplate>
      <td runat="server" style="background-color:
#FFCC66;font-weight: bold;color: #000080;">Name:
        <asp:Label ID="NameLabel" runat="server"
Text='<%# Eval("Name") %>' />
        <br />
        Customer_PH:
        <asp:Label ID="Customer_PHLabel" runat="server"
Text='<%# Eval("Customer_PH") %>' />
        <br />
        Customer_Type:
        <asp:Label ID="Customer_TypeLabel" runat="server"
Text='<%# Eval("Customer_Type") %>' />
        <br />
      </td>
   </SelectedItemTemplate>
</asp:listview>
<asp:SqlDataSource ID="SqlDataSource1"
runat="server" ConnectionString="<%$
ConnectionStrings:CustomerDetailConnectionString
%>" SelectCommand="SELECT * FROM [Cust_Det]"></
asp:SqlDataSource>
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
```

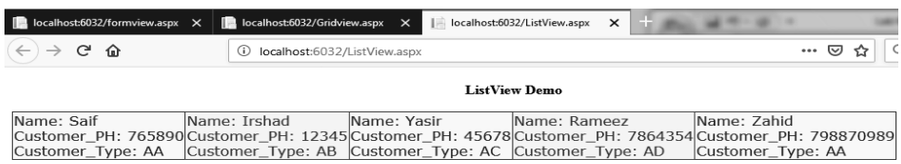
```
   <div>


   </div>
   </form>
</body>
</html>
```

Code behind ListView.aspx that is ListView.aspx.vb is given below:

ListView.aspx.vb

```
'Program to demonestrate the use of ListView DataContol
in asp.net
Public Class ListView
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   Protected Sub Unnamed1_SelectedIndexChanged(sender As
Object, e As EventArgs)
   End Sub
End Class
```

Step 3: Build and run the project. The output generated is shown in figure given below:



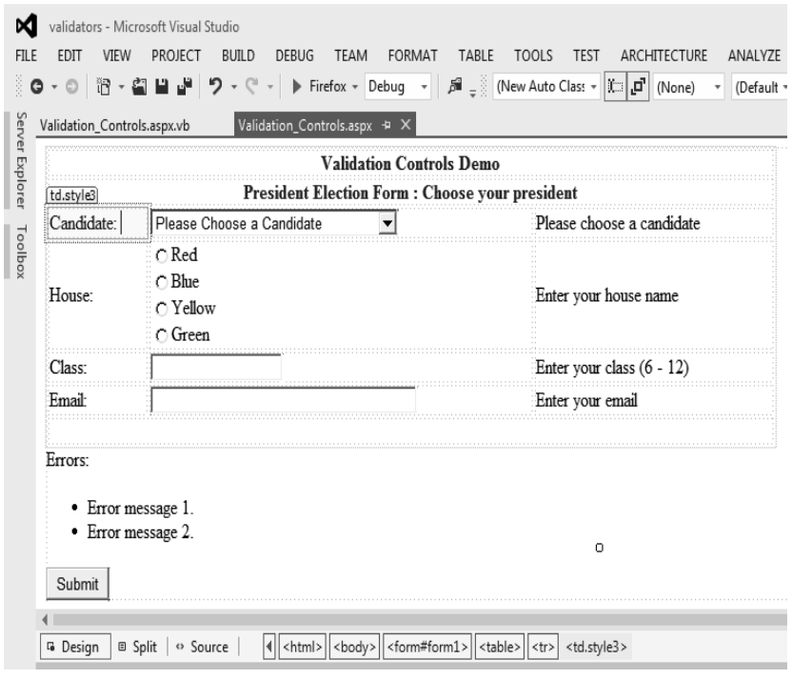
37. Write a program to implement validation controls in asp.net using VB.

Step 1: Create a New ASP.NET Empty Web form by following the steps as discussed earlier.

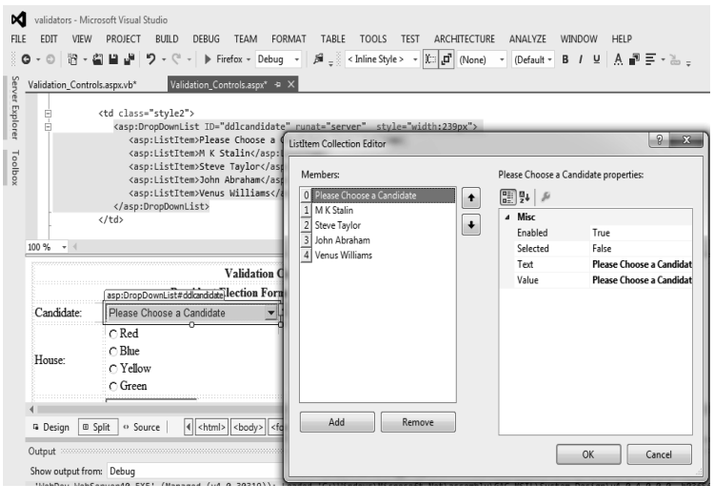
Step 2: Add the windows form and name it as "Validation_Controls.aspx".

Step 3: Design "Validation_Controls.aspx" by using controls from toolbox as shown in design look of Validation_Controls.aspx below:

NOTES

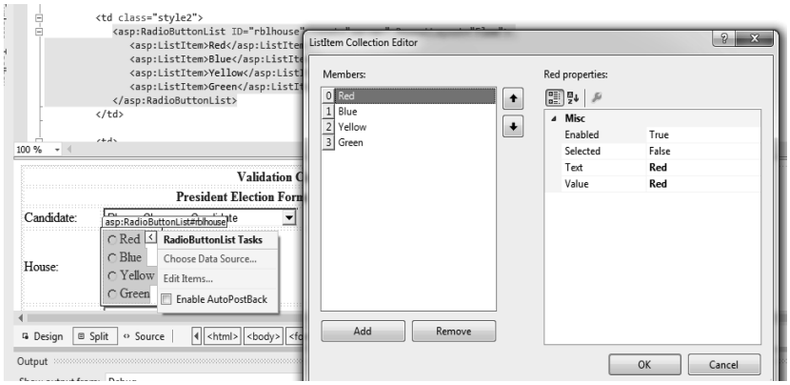


Customize the items of DropDownList by clicking on small arrow symbol that appears and choose either database or Edit Items (used in this example) as shown below:

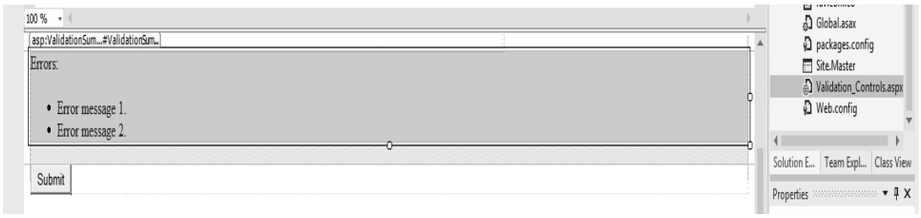


Similarly click on RadioButtonList to add option as shown below:

NOTES



Add validation summary from tool box to take handle the validation errors if raised by project as shown below:



Change the property i.e. DisplayMode= BulletList.

Source code of Validation_Controls.aspx is given below:

Validation_Controls.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="Validation_Controls.aspx.vb"
Inherits="validators.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
</head>
<body>
   <form id="form1" runat="server">
   <table style="width: 66%;">
   <tr>
      <td class="style1" colspan="3" align="center">
      <asp:Label ID="Label1"
         Text="Validation Controls Demo"
         runat="server"  Font-Bold="True"
ForeColor="#000099" />
```

NOTES

```
        </td>
    </tr>
    <tr>
        <td class="style1" colspan="3" align="center">

            <asp:Label ID="lblmsg"
            Text="President Election Form : Choose your
president"
            runat="server" Font-Bold="True"
ForeColor="#000099" />
        </td>
    </tr>
    <tr>
        <td class="style3">
            Candidate:
        </td>
        <td class="style2">
            <asp:DropDownList ID="ddlcandidate" runat="server"
style="width:239px">
            <asp:ListItem>Please Choose a Candidate</
asp:ListItem>
            <asp:ListItem>M K Stalin</asp:ListItem>
            <asp:ListItem>Steve Taylor</asp:ListItem>
            <asp:ListItem>John Abraham</asp:ListItem>
            <asp:ListItem>Venus Williams</asp:ListItem>
            </asp:DropDownList>
        </td>
        <td>
            <asp:RequiredFieldValidator ID="rfvcandidate"
            runat="server" ControlToValidate ="ddlcandidate"
            ErrorMessage="Please choose a candidate"
            InitialValue="Please choose a candidate">
            </asp:RequiredFieldValidator>
        </td>
    </tr>
    <tr>
        <td class="style3">
            House:
```

NOTES

```
      </td>
      <td class="style2">
        <asp:RadioButtonList ID="rblhouse" runat="server"
RepeatLayout="Flow">
        <asp:ListItem>Red</asp:ListItem>
        <asp:ListItem>Blue</asp:ListItem>
        <asp:ListItem>Yellow</asp:ListItem>
        <asp:ListItem>Green</asp:ListItem>
        </asp:RadioButtonList>
      </td>
      <td>
        <asp:RequiredFieldValidator ID="rfvhouse"
runat="server"
        ControlToValidate="rblhouse" ErrorMessage="Enter
your house name" >
        </asp:RequiredFieldValidator>
        <br />
      </td>
   </tr>
   <tr>
      <td class="style3">
        Class:
      </td>
      <td class="style2">
        <asp:TextBox ID="txtclass" runat="server"></
asp:TextBox>
      </td>
      <td>
        <asp:RangeValidator ID="rvclass"
        runat="server" ControlToValidate="txtclass"
        ErrorMessage="Enter your class (6 - 12)"
MaximumValue="12"
        MinimumValue="6" Type="Integer">
        </asp:RangeValidator>
      </td>
   </tr>
   <tr>
      <td class="style3">
```

NOTES

```
            Email:
         </td>
         <td class="style2">
            <asp:TextBox ID="txtemail" runat="server"
style="width:250px">
            </asp:TextBox>
         </td>
         <td>
            <asp:RegularExpressionValidator ID="remail"
runat="server"
            ControlToValidate="txtemail" ErrorMessage="Enter
your email"
            ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\
w+)*\.\w+([-.]\w+)*">
            </asp:RegularExpressionValidator>
         </td>
      </tr>
      <tr>
         <td class="style3" align="center" colspan="3">
             </td>
      </tr>
      </table>
        <asp:ValidationSummary ID="ValidationSummary1"
runat="server"
      DisplayMode ="BulletList" ShowSummary ="true"
HeaderText="Errors:" />
        <asp:Button ID="btnsubmit" runat="server"
Text="Submit" />
</form>
</body>
</html>
```

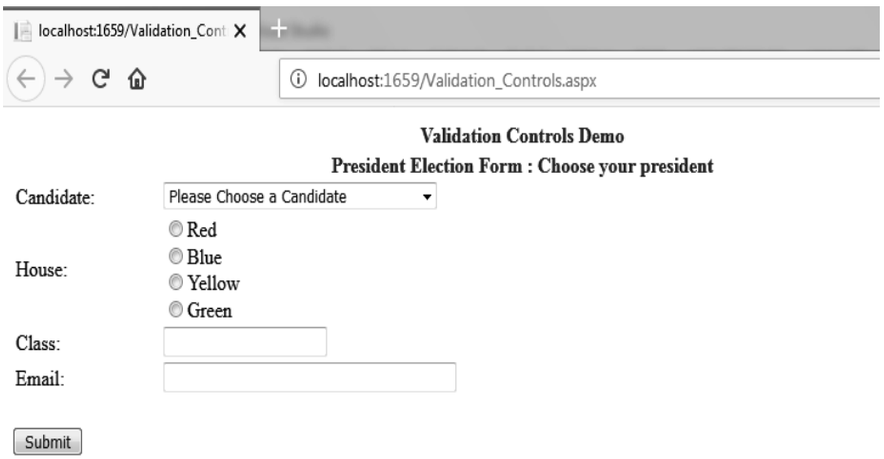
Code behind Validation_Controls.aspx design is given below:

```
'Program to demonestrate the use of Validation in ASP.
NET Program
Public Class WebForm1
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
```

```
   Protected Sub btnsubmit_Click(sender As Object, e As
EventArgs) Handles btnsubmit.Click
      If (Page.IsValid) Then
         lblmsg.Text = "Thank You"
      Else
      End If
      lblmsg.Text = "Fill up all the fields"
   End Sub
   Protected Sub ddlcandidate_SelectedIndexChanged(sender
As Object, e As EventArgs) Handles ddlcandidate.
SelectedIndexChanged
   End Sub
End Class
```

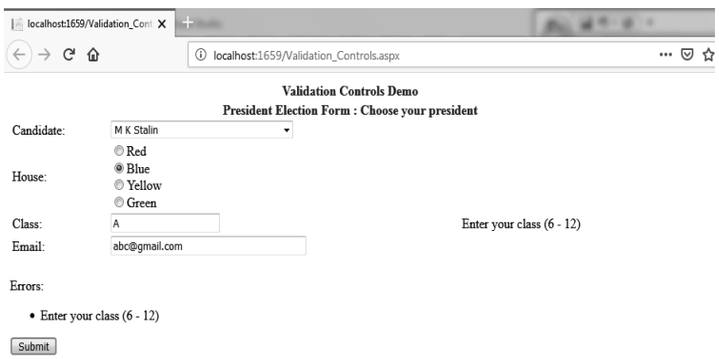
NOTES

Step 4: Build and start "Validators" project to observe the impact of validators used, the same is shown in subsequent figures below:



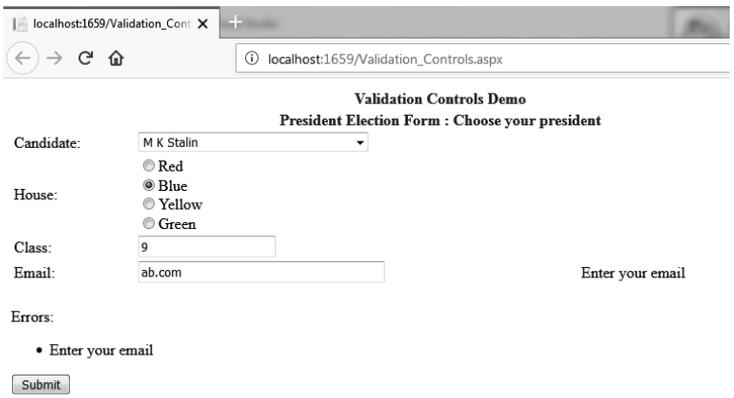
Next enter the details as required

a. Validation on Class



NOTES

b. Validation on Email



38. Write a program to implement various rich controls in Asp.net using VB.

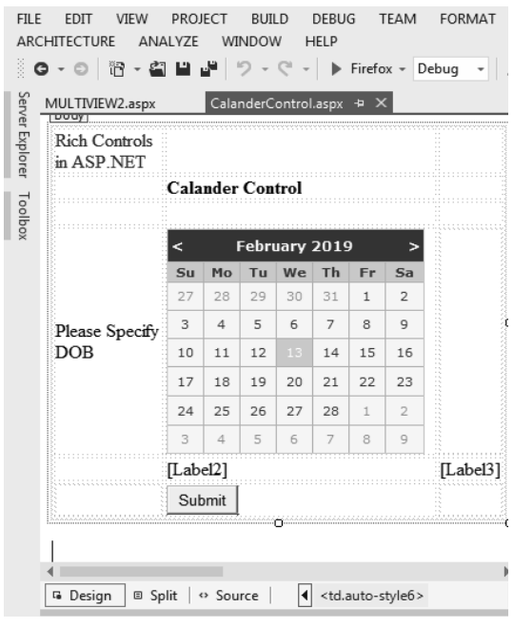
I. Calendar Control

Step 1: Create a new ASP.NET EMPTY WEB FORM and name it as "richcontrol".

Step 2: Right click on rich control project in Solution Explorer to add new Windows Form and name the form as "CalanderControl.aspx".

Step 3: Design "CalanderControl.aspx" as shown below by using various controls from toolbox like Calender control, Labels and button as shown in figure below.

CalanderControl.aspx[Design]



Source code of CalanderControl.aspx[Design] is given below:

NOTES

```
'CalanderControl.aspx
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="CalanderControl.aspx.vb"
Inherits="richcontrol1.CalanderControl" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style3 {
         width: 238px;
      }
      .auto-style2 {
         width: 279px;
      }
      .auto-style1 {
         width: 100%;
      }
      .auto-style4 {
         width: 238px;
         height: 26px;
      }
      .auto-style5 {
         width: 279px;
         height: 26px;
      }
      .auto-style6 {
         height: 26px;
      }
      .auto-style7 {
         width: 238px;
         height: 30px;
      }
      .auto-style8 {
         width: 279px;
         height: 30px;
```

NOTES

```
        }
        .auto-style9 {
           height: 30px;
        }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div>

           <table class="auto-style1">
        <tr>
           <td class="auto-style3">

     <asp:Label ID="Label1"  runat="server" Text="Rich
Controls in ASP.NET" style="text-align: center"
ForeColor="#0000CC"></asp:Label>

           </td>
           <td class="auto-style2"> </td>
           <td> </td>
        </tr>
        <tr>
           <td class="auto-style3"> </td>
           <td  class="auto-style2"><strong>Calander
Control</strong></td>
           <td> </td>
        </tr>
        <tr>
           <td class="auto-style3"> </td>
           <td class="auto-style2"> </td>
           <td> </td>
        </tr>
        <tr>
```

NOTES

```
            <td class="auto-style4">Please Specify DOB</td>
            <td class="auto-style5">
            <asp:Calendar ID="Calendar1" runat="server"
BackColor="#FFFFCC" BorderColor="#FFCC66"
BorderWidth="1px" DayNameFormat="Shortest" Font-
Names="Verdana" Font-Size="8pt" ForeColor="#663399"
Height="200px" ShowGridLines="True" Width="220px">
                <DayHeaderStyle BackColor="#FFCC66"
Font-Bold="True" Height="1px" />
                <NextPrevStyle Font-Size="9pt"
ForeColor="#FFFFCC" />
                <OtherMonthDayStyle ForeColor="#CC9966"
/>
                <SelectedDayStyle BackColor="#CCCCFF"
Font-Bold="True" />
                <SelectorStyle BackColor="#FFCC66" />
                <TitleStyle BackColor="#990000" Font-
Bold="True" Font-Size="9pt" ForeColor="#FFFFCC" />
                <TodayDayStyle BackColor="#FFCC66"
ForeColor="White" />
            </asp:Calendar>

            <td class="auto-style6"></td>
        </tr>
        <tr>
            <td class="auto-style3"> </td>
            <td class="auto-style2">
            <asp:Label ID="Label2" runat="server"></
asp:Label>
            </td>
            <td>
            <asp:Label ID="Label3" runat="server"></
asp:Label>
            </td>
        </tr>
        <tr>
            <td class="auto-style7"></td>
```

NOTES

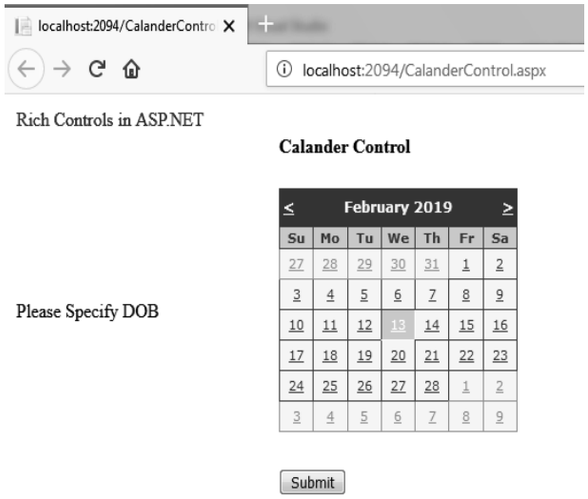
```
            <td class="auto-style8">
            <asp:Button ID="Button1" runat="server"
Text="Submit" />
            </td>
            <td class="auto-style9"></td>
        </tr>
      </table>

   </div>
   </form>
</body>
</html>
```

Code behind CalanderControl.aspx is given below:

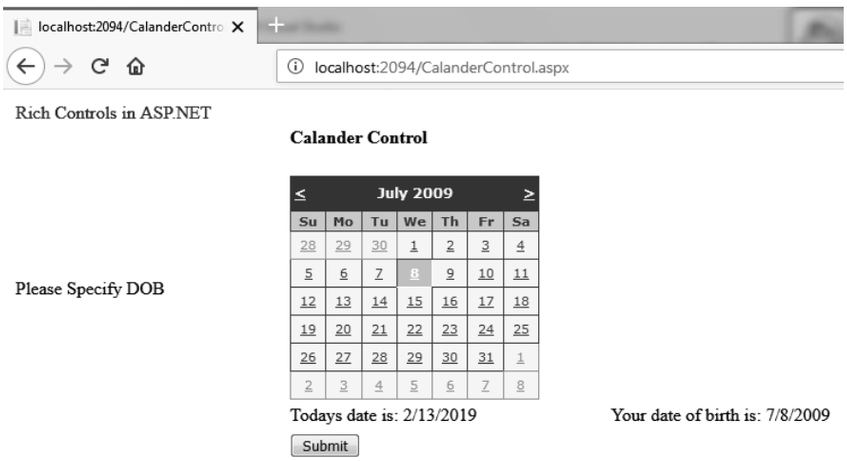
CalanderControl.aspx.vb

```
Public Class CalanderControl
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   Protected Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      Label2.Text = "Todays date is: " + Calendar1.
TodaysDate.ToShortDateString()
      Label3.Text = "Your date of birth is: " + Calendar1.
SelectedDate.ToShortDateString()
   End Sub
Step
   Protected Sub Calendar1_SelectionChanged(sender As
Object, e As EventArgs) Handles Calendar1.SelectionChanged


   End Sub
End Class
```

Step 4: Build and start CalanderControl.aspx to display the outcome of the implementation as shown below:



Next select date (Day, Month, year) represents your DOB.



The DOB selected and current date is displayed.

II. File Upload Control

Step 1: Create a new ASP.NET EMPTY WEB FORM and name it as "richcontrol".

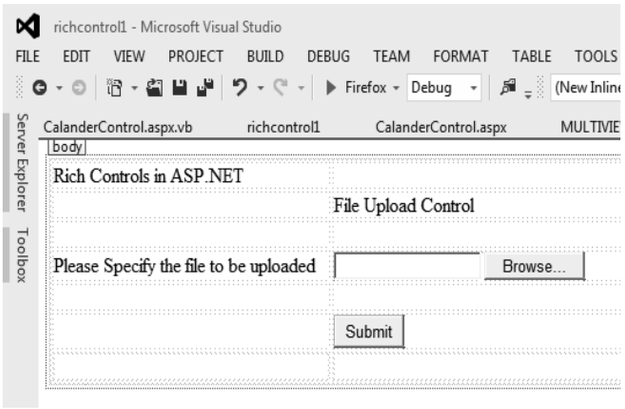
Step 2: Right click on rich control project in Solution Explorer to add new Windows Form and name the form as "FileUpladControl.aspx".

Step 3: Design "FileUploadControl.aspx" as shown below by using various controls from toolbox like FileUpload control, Labels and button as shown in figure below.

FileUploadControl.aspx[Design]

NOTES

NOTES



Source code of FileUploadControl.aspx[Design] is given below:

FileUploadControl.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="FileUpLoadControl.aspx.vb"
Inherits="richcontrol1.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style1 {
         width: 100%;
      }
      .auto-style2 {
         width: 279px;
      }
      .auto-style3 {
         width: 238px;
      }
      .auto-style4 {
         width: 238px;
         height: 23px;
      }
      .auto-style5 {
         width: 279px;
         height: 23px;
      }
```

NOTES

```
      .auto-style6 {
        height: 23px;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div>
          <table class="auto-style1">
        <tr>
          <td class="auto-style3">

     <asp:Label ID="Label1"  runat="server" Text="Rich
Controls in ASP.NET" style="text-align: center"></
asp:Label>

          </td>
          <td class="auto-style2"> </td>
          <td> </td>
        </tr>
        <tr>
          <td class="auto-style3"> </td>
          <td class="auto-style2">File Upload Control</td>
          <td> </td>
        </tr>
        <tr>
          <td class="auto-style3"> </td>
          <td class="auto-style2"> </td>
          <td> </td>
        </tr>
        <tr>
          <td class="auto-style3">Please Specify the
file to be uploaded</td>
          <td class="auto-style2">
          <asp:FileUpload ID="FileUpload1" runat="server" />
```

NOTES

```
            <td> </td>
         </tr>
         <tr>
            <td class="auto-style3"> </td>
            <td class="auto-style2"> </td>
            <td> </td>
         </tr>
         <tr>
            <td class="auto-style3"> </td>
            <td class="auto-style2">
            <asp:Button ID="Button1" runat="server"
Text="Submit" />
            </td>
            <td> </td>
         </tr>
         <tr>
            <td class="auto-style4"></td>
            <td class="auto-style5"></td>
            <td class="auto-style6"></td>
         </tr>
      </table>
            </div>
   </form>
</body>
</html>
```

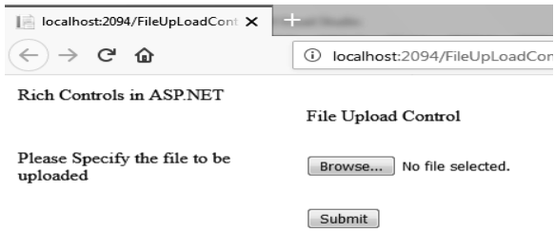
Code behind FileUploadControl.aspx is given below:

```
'FileUploadControl.aspx.vb
Public Class WebForm1
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   Protected Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      If FileUpload1.HasFile Then
         MsgBox("FIle Uploaded Sucessfully")
      Else
```

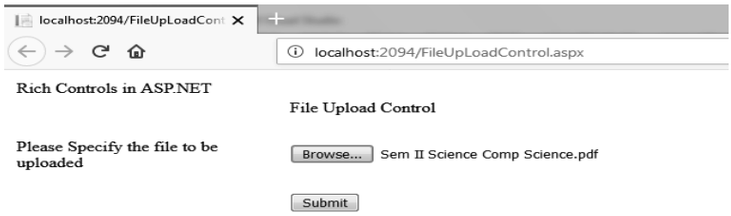
```
        MsgBox("Please upload the File Content")
      End If
   End Sub
End Class
```

NOTES

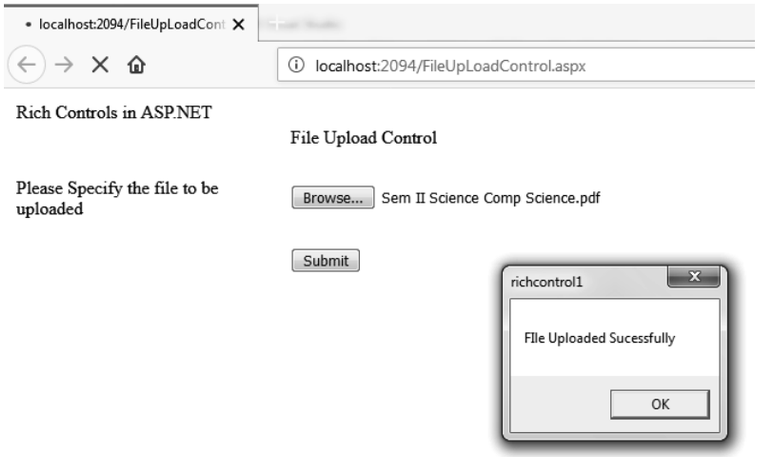
Step 4: Build and start FileUploadControl.aspx to display the outcome of the implementation as shown below:



Next select the file to upload by click on Browse button.



Acknowledgment on successful upload.



III. File Upload Control

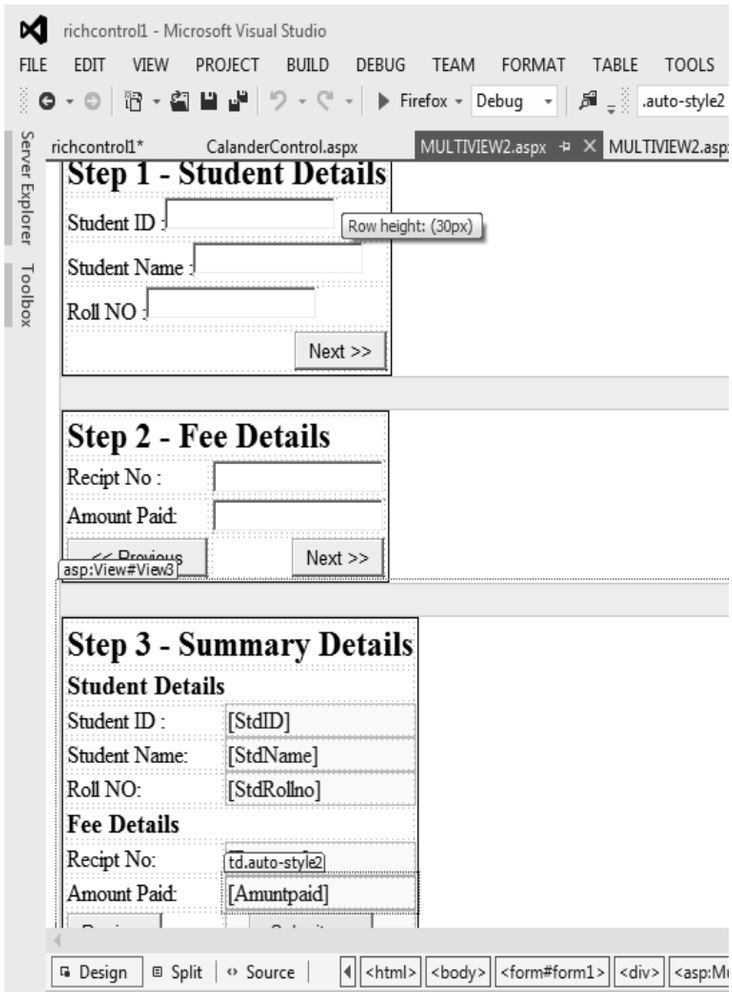
Step 1: Create a new ASP.NET EMPTY WEB FORM and name it as "richcontrol".

NOTES

Step 2: Right click on richcontrol project in Solution Explorer to add new Windows Form and name the form as "MULTIVIEW2.aspx".

Step 3: Design "MULTIVIEW2.aspx" as shown below by using various controls from toolbox like MultiView control, Labels and button as shown in figure below.

MULTIVIEW2 .aspx[Design]



Source code of MULTIVIEW2.aspx[Design] is given below:

```
'MULTIVIEW2.aspx
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="MULTIVIEW2.aspx.vb" Inherits="richcontrol1.
MULTIVIEW2" %>
```

NOTES

```
<!DOCTYPE html>
<html xmlns=”http://www.w3.org/1999/xhtml”>
<head runat=”server”>
   <title></title>
   <style type=”text/css”>
      .auto-style3 {
         height: 26px;
         width: 239px;
      }
      .auto-style1 {
         height: 30px;
      }
      .auto-style2 {
         width: 109px;
      }
      .auto-style4 {
         width: 239px;
      }
   </style>
</head>
<body>
   <form id=”form1” runat=”server”>
   <div>
      <asp:MultiView ID=”MultiView1” runat=”server”>
         <asp:View ID=”View1” runat=”server”>
            <table style=”border: 1px solid black”>
            <tr>
                  <td class=”auto-style4”>
                        <h2>Step 1 - Student Details</h2>
                  </td>
            </tr>
            <tr>
                  <td  class=”auto-style3”>Student  ID
:<asp:TextBox ID=”TextBox1” runat=”server”></asp:TextBox>
                  </td>
            </tr>
```

NOTES

```
            <tr>
                  <td class="auto-style4">Student Name
:<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
                  </td>
            </tr>
            <tr>
                  <td class="auto-style4">Roll NO
:<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
                  </td>
            </tr>
            <tr>
                  <td style="text-align: right"
class="auto-style4">
                        <asp:Button ID="btnStep2"
runat="server" onclick="btnStep2_Click" Text="Next
&gt;&gt;" />
                  </td>
            </tr>
            </table>
        </asp:View>
        <asp:View ID="View2" runat="server">
           <table style="border: 1px solid black">
           <tr>
                  <td colspan="2">
                        <h2>Step 2 - Fee Details</h2>
                  </td>
           </tr>
           <tr>
                  <td>Recipt No :</td>
                  <td>
        <asp:TextBox ID="txtOrderID" runat="server"></
asp:TextBox>
                  </td>
           </tr>
           <tr>
                  <td>Amount Paid:</td>
                  <td>
```

```
        <asp:TextBox ID="txtQuantity" runat="server"></
asp:TextBox>
                </td>
           </tr>
           <tr>
                <td>
     <asp:Button  ID="btnBackToStep1"  runat="server"
onclick="btnBackToStep1_Click" Text="&lt;&lt; Previous"
/>
                </td>
                <td style="text-align: right">
        <asp:Button  ID="btnStep3"  runat="server"
onclick="btnGoToStep3_Click" Text="Next &gt;&gt;" />
                </td>
           </tr>
           </table>
        </asp:View>
        <asp:View ID="View3" runat="server">
           <table style="border: 1px solid black">
           <tr>
                <td class="auto-style1" colspan="2">
                     <h2>Step 3 - Summary Details</h2>
                </td>
           </tr>
           <tr>
                <td colspan="2">
                     <h3>Student Details</h3>
                </td>
           </tr>
           <tr>
                <td>Student ID :</td>
                <td class="auto-style2">
        <asp:Label ID="StdID" runat="server"></asp:Label>
                </td>
           </tr>
           <tr>
```

NOTES

NOTES

```
                    <td>Student Name:</td>
                    <td class="auto-style2">
               <asp:Label  ID="StdName"  runat="server"></
asp:Label>
                    </td>
               </tr>
               <tr>
                    <td>Roll NO:</td>
                    <td class="auto-style2">
               <asp:Label  ID="StdRollno"  runat="server"></
asp:Label>
                    </td>
               </tr>
               <tr>
                    <td colspan="2">
                         <h3>Fee Details</h3>
                    </td>
               </tr>
               <tr>
                    <td>Recipt No:</td>
                    <td class="auto-style2">
               <asp:Label  ID="Recptno"  runat="server"></
asp:Label>
                    </td>
               </tr>
               <tr>
                    <td>Amount Paid:</td>
                    <td class="auto-style2">
                         <asp:Label  ID="Amuntpaid"
runat="server"></asp:Label>
                    </td>
               </tr>
               <tr>
                    <td>
               <asp:Button  ID="Button1"  runat="server"
Text="Previous" />
                    </td>
               <td  class="auto-style2"  style="text-align:
right">
```

```
          <asp:Button ID="btnSubmit" runat="server"
OnClick="btnSubmit_Click" style="height: 26px"
Text="Submit &gt;&gt;" />
                </td>
          </tr>
          </table>
        </asp:View>
      </asp:MultiView>

   </div>
   </form>
</body>
</html>
```

Code behind FileUploadControl.aspx is given below:

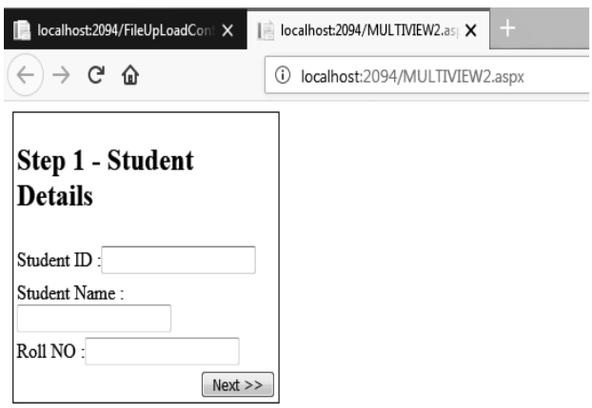
```
'MULTIVIEW2.aspx.vb
Public Class MULTIVIEW2
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
      If IsPostBack = False Then
        MultiView1.ActiveViewIndex = 0
      End If
   End Sub
   Protected Sub MultiView1_ActiveViewChanged(sender
As Object, e As EventArgs) Handles MultiView1.
ActiveViewChanged
   End Sub
   Protected Sub btnStep2_Click(sender As Object, e As
EventArgs) Handles btnStep2.Click
      MultiView1.ActiveViewIndex = 1
   End Sub
   Protected Sub btnBackToStep1_Click(sender As Object,
e As EventArgs) Handles btnBackToStep1.Click
      MultiView1.ActiveViewIndex = 0
   End Sub
   Protected Sub btnSubmit_Click(sender As Object, e As
EventArgs) Handles btnSubmit.Click
      MsgBox("Data Saved Sucessfully")
   End Sub
```

NOTES

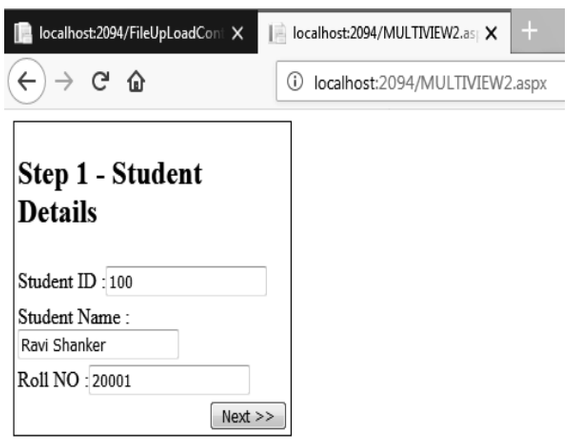
NOTES

```
    Protected Sub btnGoToStep3_Click(sender As Object, e
As EventArgs) Handles btnStep3.Click
        MultiView1.ActiveViewIndex = 2
        StdID.Text = TextBox1.Text
        StdName.Text = TextBox2.Text
        StdRollno.Text = TextBox3.Text
        Recptno.Text = txtOrderID.Text
        Amuntpaid.Text = txtQuantity.Text
    End Sub
    Protected Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
        MultiView1.ActiveViewIndex = 1
    End Sub
End Class
```

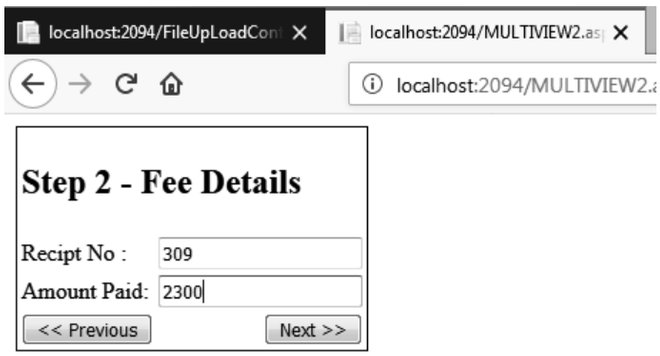
Step 4: Build and start MULTIVIEW2.aspx to display the outcome of the implementation in subsequent figures as shown below:



Next fill the data.

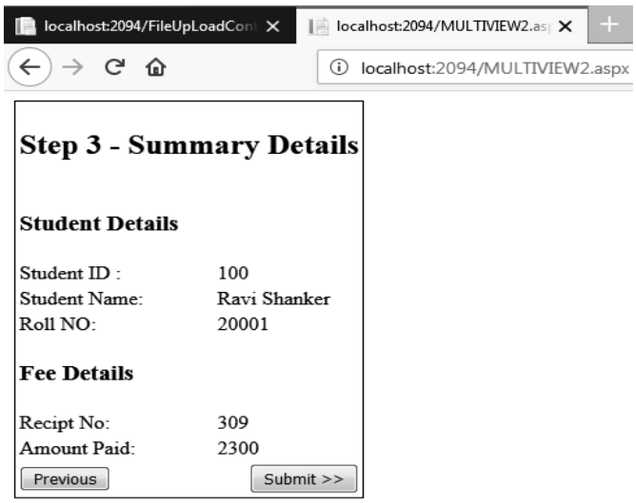


Click next.



NOTES

Click Next



Click submit to acknowledge the submission.

Or click Previous to modify the entries made.

39. Write a program to demonstrate the implementation of HTML server controls used in asp.net.

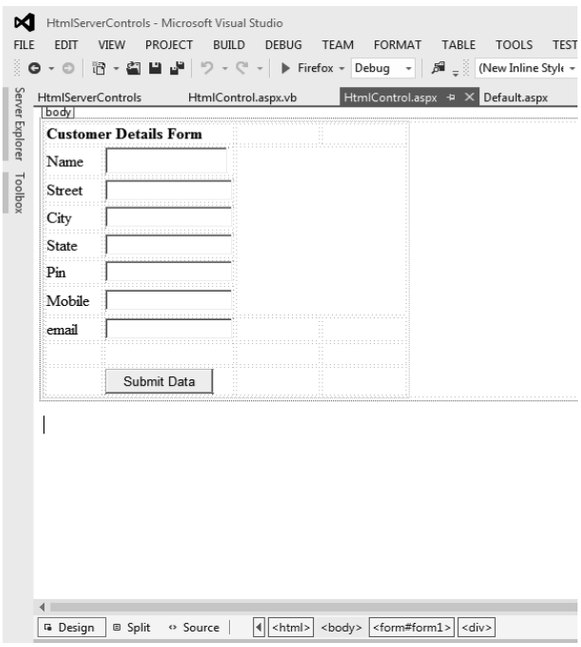
NOTES

Step 1: Create a new ASP.NET EMPTY WEB FORM and Name it as "HtmlServerControls".

Step 2: Right click on HtmlServerControlsl project in Solution Explorer to add new Windows Form and name the form as "HtmlControl.aspx".

Step 3: Design "HtmlControl.aspx" as shown below by using various controls from toolbox like TextBoxes, Labels and button as shown in figure below.

HtmlControl .aspx[Design]



Source code of HtmlControl.aspx[Design] is given below:

HtmlControl..aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="HtmlControl.aspx.vb"
Inherits="HtmlServerControls.HtmlControl" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style1 {
         height: 289px;
```

```
   width: 366px;
}
.auto-style3 {
   width: 101px;
   height: 26px;
}
.auto-style4 {
   height: 30px;
   width: 101px;
}
.auto-style7 {
   height: 19px;
}
.auto-style8 {
   width: 101px;
   height: 19px;
}
.auto-style12 {
   width: 101px;
   height: 22px;
}
.auto-style18 {
}
.auto-style19 {
   width: 101px;
   height: 25px;
}
.auto-style20 {
   width: 55px;
   height: 25px;
}
.auto-style21 {
   width: 55px;
   height: 26px;
}
.auto-style22 {
   width: 55px;
   height: 22px;
```

NOTES

NOTES

```
        }
        .auto-style23 {
           height: 30px;
           width: 55px;
        }
        .auto-style24 {
           width: 55px;
           height: 19px;
        }
        .auto-style26 {
           height: 26px;
        }
        .auto-style27 {
           height: 22px;
        }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div>
   <table class="auto-style1">
        <tr>
           < t d    c l a s s = " a u t o - s t y l e 7 "
colspan="2"><strong>Customer Details Form</strong></td>
           <td class="auto-style7"> </td>
           <td class="auto-style7"> </td>
        </tr>
        <tr>
           <td class="auto-style20">Name</td>
           <td class="auto-style19">
           <asp:TextBox ID="txtName" runat="server"
Height="22px" style="margin-right: 0px" Width="115px"></
asp:TextBox>
           </td>
           <td class="auto-style18" colspan="2"
rowspan="6" id="display_print" runat="server" >  </
td>
        </tr>
        <tr>
```

```
    <td class="auto-style20">Street</td>
    <td class="auto-style19">
    <asp:TextBox ID="txtStreet" runat="server"></
    asp:TextBox>
    </td>
</tr>
<tr>
    <td class="auto-style21">City</td>
    <td class="auto-style3">
    <asp:TextBox ID="txtCity" runat="server"></
    asp:TextBox>
    </td>
</tr>
<tr>
    <td class="auto-style21">State</td>
    <td class="auto-style3">
    <asp:TextBox ID="txtState" runat="server"></
    asp:TextBox>
    </td>
</tr>
<tr>
    <td class="auto-style22">Pin</td>
    <td class="auto-style12">
    <asp:TextBox ID="txtPin" runat="server"></
    asp:TextBox>
    </td>
</tr>
<tr>
    <td class="auto-style23">Mobile</td>
    <td class="auto-style4">
    <asp:TextBox ID="txtMob" runat="server"></
    asp:TextBox>
    </td>
</tr>
<tr>
    <td class="auto-style22">email</td>
    <td class="auto-style12">
    <asp:TextBox ID="txtEmail" runat="server"
    type="email"></asp:TextBox>
```

NOTES

NOTES

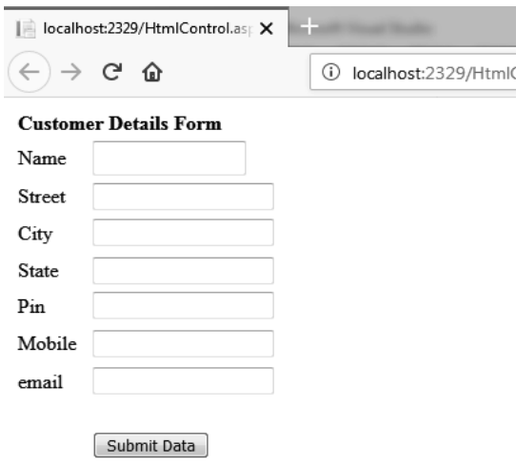
```
            </td>
            <td class="auto-style27"> </td>
            <td class="auto-style27"> </td>
         </tr>
         <tr>
            <td class="auto-style24"> </td>
            <td class="auto-style8"> </td>
            <td class="auto-style7"> </td>
            <td class="auto-style7"> </td>
         </tr>
         <tr>
            <td class="auto-style21"> </td>
            <td class="auto-style3">
            <asp:Button ID="Button1" runat="server"
            Text="Submit Data" />
            </td>
            <td class="auto-style26"> </td>
            <td class="auto-style26"> </td>
         </tr>
      </table>
      </div>
   </form>
</body>
</html>
```

Code behind HtmlControl.aspx is given below:

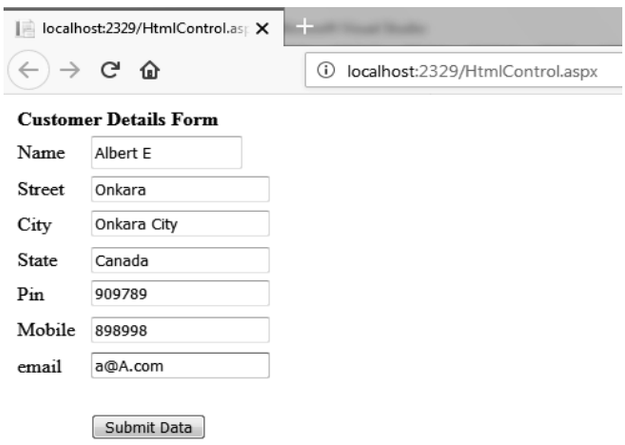
```
'HtmlControl.aspx.vb
'Program To Demonstrate the implementation of HTMLServer
Controls used in asp.net
Public Class HtmlControl
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   Protected Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
      Dim User_inf As String = ""
      User_inf += "Name:  " + txtName.Text
      User_inf += vbNewLine + "Street:  " + txtStreet.
Text + vbNewLine
```

```
      User_inf += vbNewLine + "City:" + txtCity.Text +
vbNewLine
      User_inf += vbNewLine + "State:  " + txtState.Text
+ vbNewLine
      User_inf += vbNewLine + "Pin:    " + txtPin.Text +
vbNewLine
      User_inf += vbNewLine + "Mobile:   " + txtMob.Text
+ vbNewLine
      User_inf += vbNewLine + "Email:   " + txtEmail.Text
+ vbNewLine
      display_print.InnerHtml = User_inf
   End Sub
End Class
```

Step 4: Build and start HtmlControls.aspx to display the outcome of the implementation in subsequent figures as shown below:

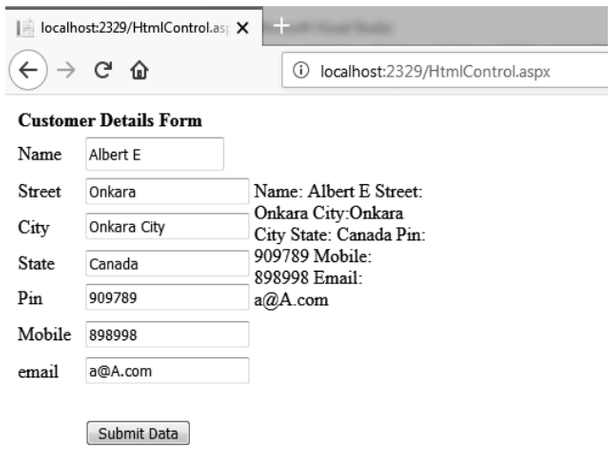


Next fill the details.



NOTES

Click Submit Data.



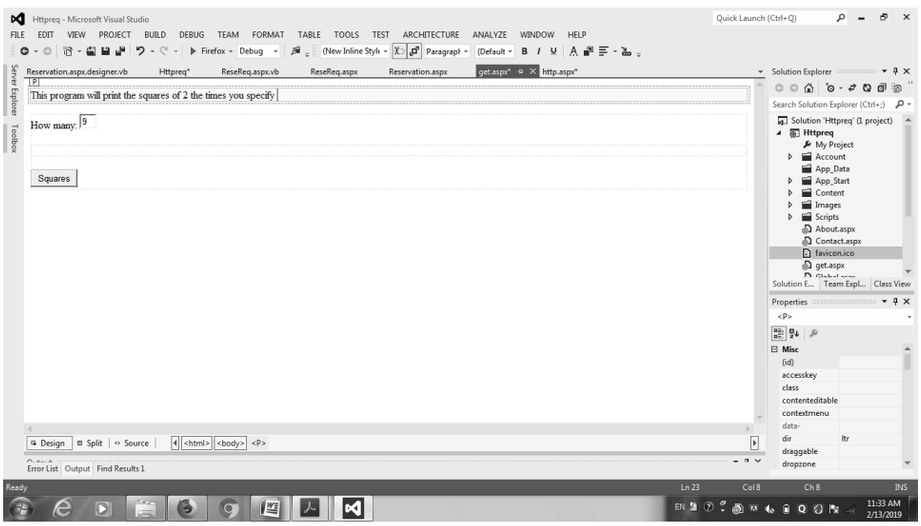
40. Write a program to demonestrate the project using HttpRequest and HttpResponse.

Step 1: Create a new ASP.NET EMPTY WEB FORM and Name it as "Httpreq".

Step 2: Right click on "Httpreq" project in solution Explorer to add two new Windows Form and name themas "httpl.aspx" and "get.aspx"

Step 3: Design "get.aspx" as shown below by using various controls from toolbox like Input box, Labels and Table as shown in figure below.

get .aspx[Design]



Step 4: Click on the get.aspx to open source code of get.aspx[Design] which is given below:

```
<%@ Page Trace = "true" %>
```

NOTES

```
<html>
<head>
</head>
<body>
<P>This program will print the squares of 2 the times
you specify </P>
<form method=”get” action =http.aspx>
How many:
<INPUT type=text size=3 value=9 name=txtCount>
<P></P>
<INPUT type=submit value=Squares name=cmdSquares>
</form>
</body>
</html>
```

Next open http.aspx and edit its source that is http.aspx the way provided as under

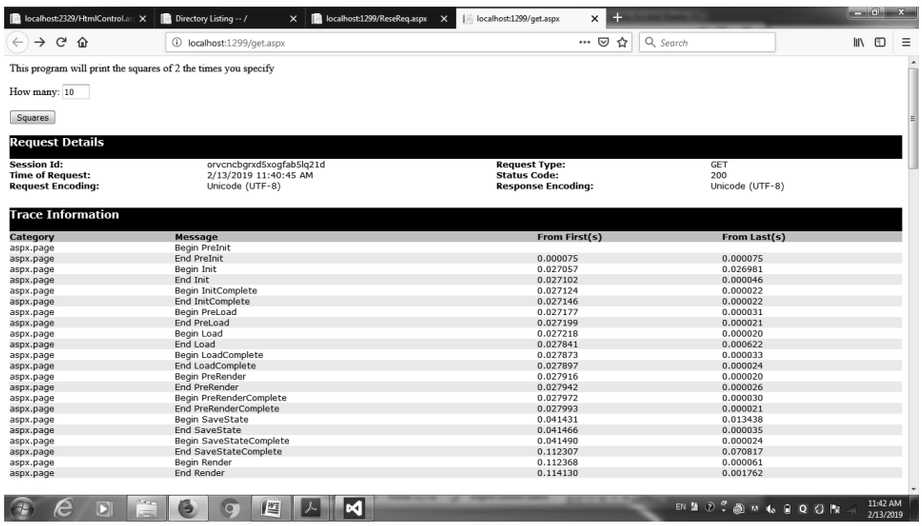
```
‘http.aspx
<%@ Page Language=”VB” Trace=”true”%>
<script runat=”server”>
   Sub Page_Init(sender As Object, e As EventArgs) ‘
httpRsponsse page
      Dim strQuery As String = _
         Request.ServerVariables(“QUERY_STRING”)
      Response.Write(“QUERY_STRING = “ & strQuery &
“<br>”)
      Dim strAgent As String = _
         Request.ServerVariables(“HTTP_USER_AGENT”)
      Response.Write(“HTTP_USER_AGENT = “ & strAgent &
“<br>”)
      Dim length As Integer = Request.ContentLength
      Response.Write(“ContentLength = “ & length & “<br>”)
      Dim strCount As String = Request.Params(“txtCount”)
      Dim count As Integer = Convert.ToInt32(strCount)
      Dim i As Integer
      For i = 1 To count
         Response.Write(i * i)
         Response.Write(“<br>”)
      Next
```

```
    End Sub
</script>
```

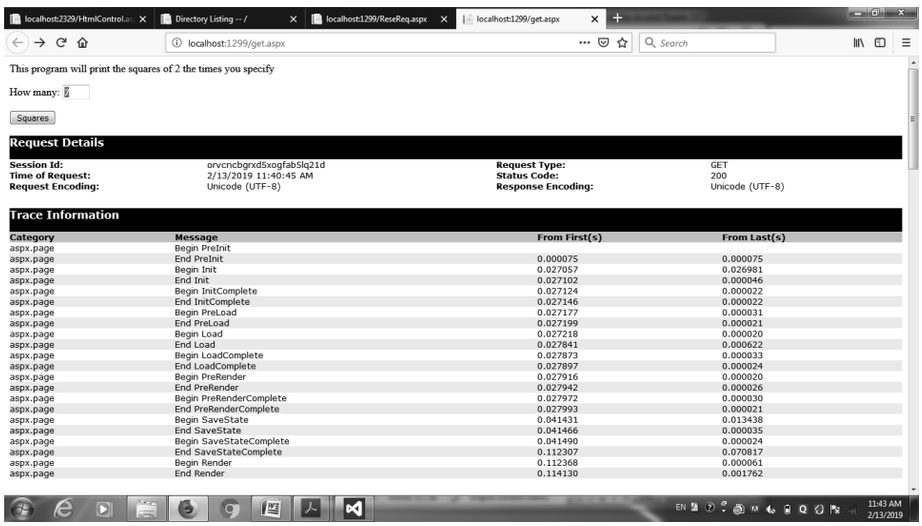
Step 5: Custmoise the server setting, build and start the project by assigning get.aspx as start page in start Action tab.

NOTES

The output generated by HttpResponse page is shown below:

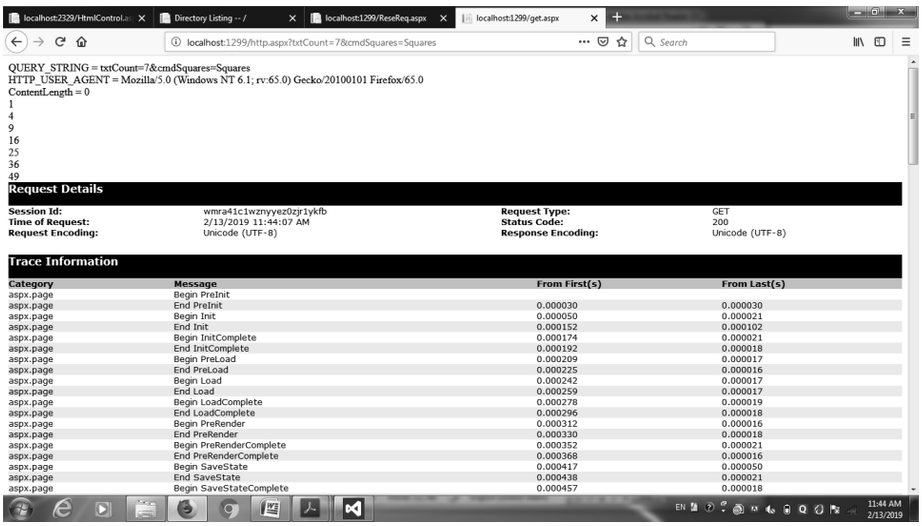


Next you can change the count value from 10 to desired one as shown below:

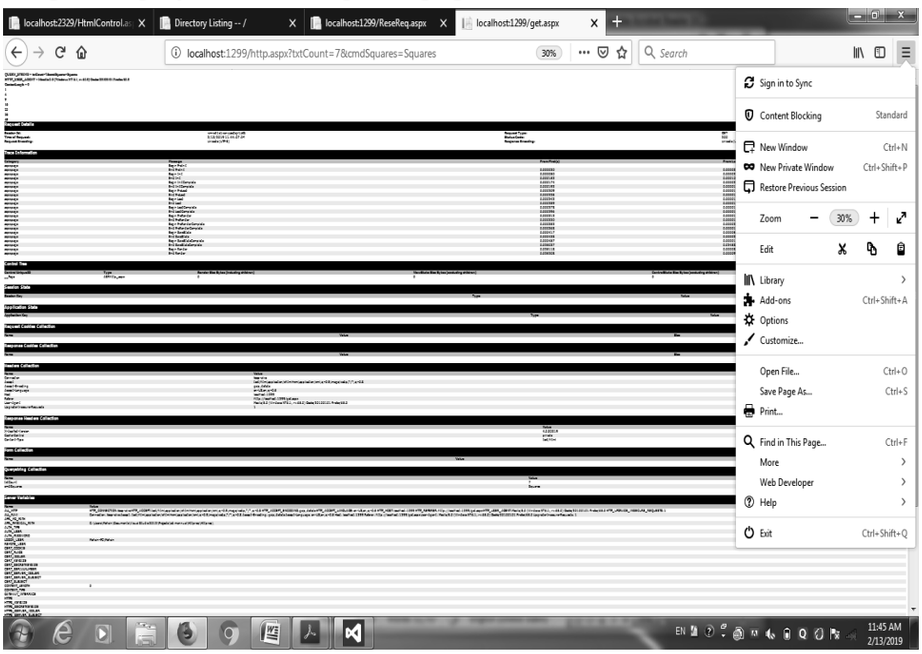


Next click on square button to moniter the results as shown below:

NOTES



The whole information like Query_String. Request Details, Trace Information etc gets displayed on the requestd page as shown below:



41. Write a program to demonstrate error handling in ASP.NET.

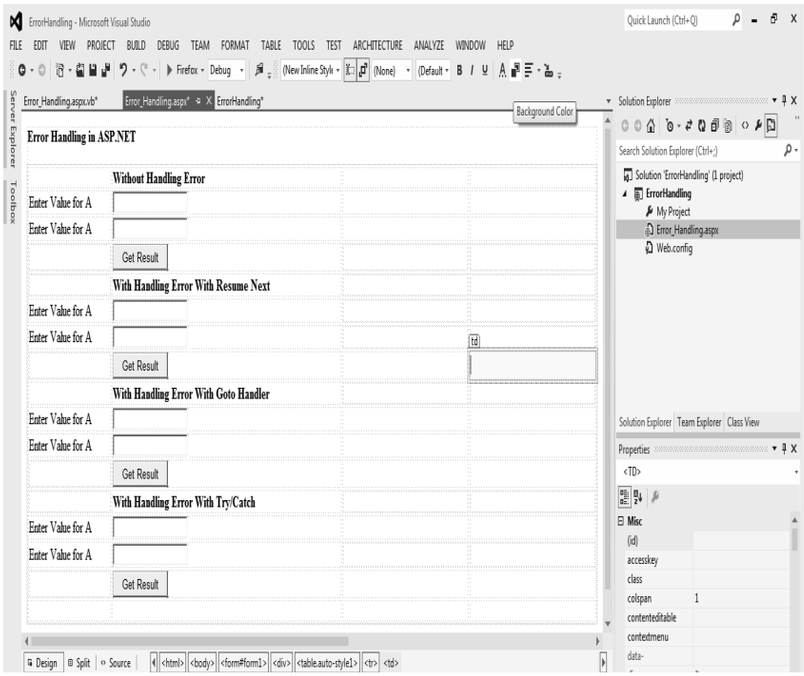
Step 1: Create a new project by choosing ASP.NET EMPTY WEB FORM and Name it as "ErrorHandling".

Step 2: Right Click on "ErrorHandling" project in solution Explorer to add new Windows Form and name" `Error_Handling.aspx`"

Step 3: Design "`Error_Handling.aspx`" as shown below by using various controls from toolbox like Table, Text box, Labels and buttons as shown in figure below.

NOTES

Error_Handling.aspx [Design]



Step 4: Click on the get.aspx to open source code of Error_Handling. aspx[Design] which is shown below:

```
'Error_Handling.aspx[Design]
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="Error_Handling.aspx.vb"
Inherits="ErrorHandling.Error_GoToand_Resume" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style1 {
         width: 100%;
      }
      .auto-style2 {
         width: 139px;
      }
      .auto-style3 {
         width: 139px;
         height: 23px;
```

NOTES

```
      }
      .auto-style4 {
         height: 23px;
      }
      .auto-style5 {
         width: 389px;
      }
      .auto-style6 {
         height: 23px;
         width: 389px;
      }
      .auto-style7 {
         width: 139px;
         height: 26px;
      }
      .auto-style8 {
         width: 389px;
         height: 26px;
      }
      .auto-style9 {
         height: 26px;
      }
   </style>
</head>
<body>
   <form id="form1" runat="server">
   <div>

      <strong style="text-align: center">Error Handling
in ASP.NET</strong><br />
      <table class="auto-style1">
         <tr>
            <td class="auto-style2"> </td>
            <td  class="auto-style5"><strong>Without
Handling Error</strong></td>
            <td> </td>
            <td> </td>
```

NOTES

```
        </tr>
        <tr>
           <td class="auto-style2">Enter Value for A</
           td>
           <td class="auto-style5">
           <asp:TextBox ID="TextBox1" runat="server"></
           asp:TextBox>
           </td>
           <td> </td>
           <td> </td>
        </tr>
        <tr>
           <td class="auto-style2">Enter Value for A</
           td>
           <td class="auto-style5">
           <asp:TextBox ID="TextBox2" runat="server"></
           asp:TextBox>
           </td>
           <td> </td>
           <td> </td>
        </tr>
        <tr>
           <td class="auto-style3"></td>
           <td class="auto-style6">
           <asp:Button ID="Show_Result" runat="server"
           Text="Get Result" />
           </td>
           <td class="auto-style4"></td>
           <td class="auto-style4"></td>
        </tr>
        <tr>
           <td class="auto-style3"></td>
           <td class="auto-style6"><strong>With Handling
Error With Resume Next</strong></td>
           <td class="auto-style4"></td>
           <td class="auto-style4"></td>
        </tr>
        <tr>
```

NOTES

```
    <td class="auto-style2">Enter Value for A</
    td>
    <td class="auto-style5">
    <asp:TextBox ID="TextBox3" runat="server"></
    asp:TextBox>
    </td>
    <td class="auto-style4"></td>
    <td class="auto-style4"></td>
</tr>
<tr>
    <td class="auto-style3">Enter Value for A</
    td>
    <td class="auto-style6">
    <asp:TextBox ID="TextBox4" runat="server"></
    asp:TextBox>
    </td>
    <td class="auto-style4"></td>
    <td class="auto-style4"></td>
</tr>
<tr>
    <td class="auto-style3"></td>
    <td class="auto-style6">
    <asp:Button ID="Show_Result0" runat="server"
    Text="Get Result" />
    </td>
    <td> </td>
    <td> </td>
</tr>
<tr>
    <td class="auto-style2"> </td>
    <td class="auto-style5"><strong>With Handling
    Error With Goto Handler</strong></td>
    <td class="auto-style4"></td>
    <td class="auto-style4"></td>
</tr>
<tr>
    <td class="auto-style2">Enter Value for A</
    td>
```

NOTES

```
    <td class="auto-style5">
    <asp:TextBox ID="TextBox5" runat="server"></
    asp:TextBox>
    </td>
    <td> </td>
    <td> </td>
</tr>
<tr>
    <td class="auto-style2">Enter Value for A</
    td>
    <td class="auto-style5">
    <asp:TextBox ID="TextBox6" runat="server"></
    asp:TextBox>
    </td>
    <td> </td>
    <td> </td>
</tr>
<tr>
    <td class="auto-style3"></td>
    <td class="auto-style6">
    <asp:Button ID="Show_Result1" runat="server"
    Text="Get Result" />
    </td>
    <td> </td>
    <td> </td>
</tr>
<tr>
    <td class="auto-style3"></td>
    <td class="auto-style6"><strong>With Handling
    Error With Try/Catch</strong></td>
    <td class="auto-style4"></td>
    <td class="auto-style4"></td>
</tr>
<tr>
    <td class="auto-style2">Enter Value for A</
    td>
    <td class="auto-style5">
```

NOTES

```
            <asp:TextBox ID="TextBox7" runat="server"></
            asp:TextBox>
            </td>
            <td> </td>
            <td> </td>
         </tr>
         <tr>
            <td class="auto-style7">Enter Value for A</
            td>
            <td class="auto-style8">
            <asp:TextBox ID="TextBox8" runat="server"></
            asp:TextBox>
            </td>
            <td class="auto-style9"></td>
            <td class="auto-style9"></td>
         </tr>
         <tr>
            <td class="auto-style3"></td>
            <td class="auto-style6">
            <asp:Button ID="Show_Result2" runat="server"
            Text="Get Result" />
            </td>
            <td> </td>
            <td> </td>
         </tr>
         <tr>
            <td class="auto-style2"> </td>
            <td class="auto-style5"> </td>
            <td> </td>
            <td> </td>
         </tr>
      </table>
   </div>
   </form>
</body>
</html>
```

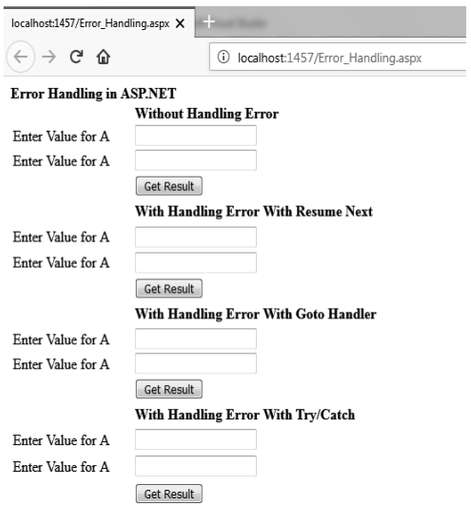
NOTES

Next open http.aspx and edit its source as shown below:

```
'Error_Handling.aspx
'Program to demostrate various error handling mechannisms
in asp.net
Imports System.Exception
Public Class Error_GoToand_Resume
   Inherits System.Web.UI.Page
   Protected Sub Page_Load(ByVal sender As Object, ByVal
e As System.EventArgs) Handles Me.Load
   End Sub
   'Without any Error Handling
   Protected Sub Show_Result_Click(sender As Object, e
As EventArgs) Handles Show_Result.Click
      Dim A As Integer = Convert.ToInt32(TextBox1.Text)
      Dim B As Integer = Convert.ToInt32(TextBox2.Text)
      Dim Result As Integer
      Result = A / B
      MsgBox(Result)
   End Sub
   'Handling Error using "on Error Resume Next"
   Protected Sub Show_Result0_Click(sender As Object, e
As EventArgs) Handles Show_Result0.Click
      On Error Resume Next
      Dim A As Integer = Convert.ToInt32(TextBox1.Text)
      Dim B As Integer = Convert.ToInt32(TextBox2.Text)
      Dim Result As Integer
      Result = A / B
      MsgBox(Result)
   End Sub
   'Handling Error using" on Error GoTo Errorhandler"
   Protected Sub Show_Result1_Click(sender As Object, e
As EventArgs) Handles Show_Result1.Click
      On Error GoTo Errorhandler
      Dim A As Integer = Convert.ToInt32(TextBox1.Text)
      Dim B As Integer = Convert.ToInt32(TextBox2.Text)
      Dim Result As Integer
      Result = A / B
      MsgBox(Result)
```

```
Errorhandler:
      MsgBox(Err.Description)
   End Sub
   'Error Handling using Try-Catch-Finally
   Protected Sub Show_Result2_Click(sender As Object, e
As EventArgs) Handles Show_Result2.Click
      Try
         Dim A As Integer = Convert.ToInt32(TextBox1.Text)
         Dim B As Integer = Convert.ToInt32(TextBox2.Text)
         Dim Result As Integer
         Result = A / B
         MsgBox(Result)
      Catch ExceptionGenerated As Exception
         MsgBox(ExceptionGenerated.ToString, MsgBoxStyle.
DefaultButton3, "ExceptionGenerated is")
      Finally
         MsgBox("Close")
      End Try
   End Sub
End Class
```

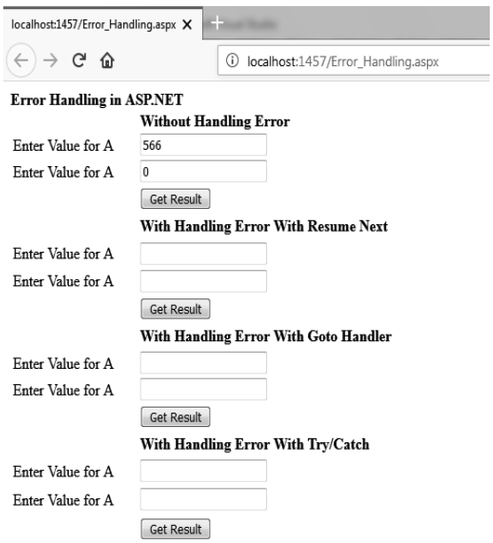
Step 5: Configure Start Action and Server settings, then build the project and start it to display the outcome as shown in figure below:



Next try to fill data into textboxes and check all the error handling types as shown in subsequent figures below:

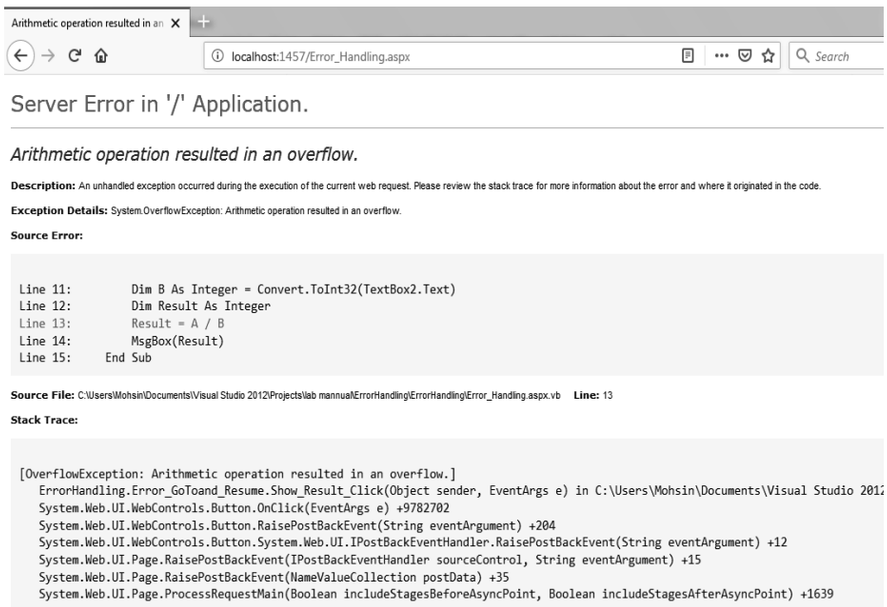
NOTES

a.

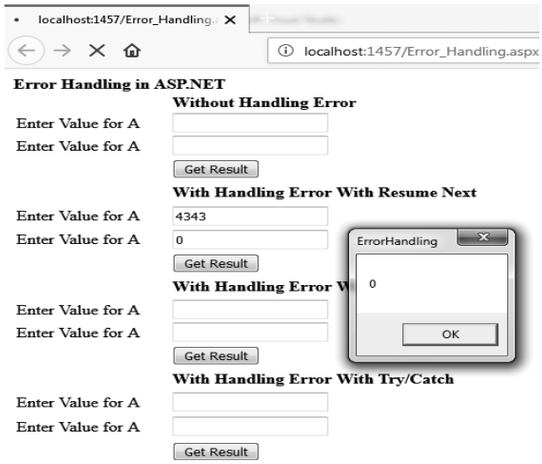


Next click on Get Result.

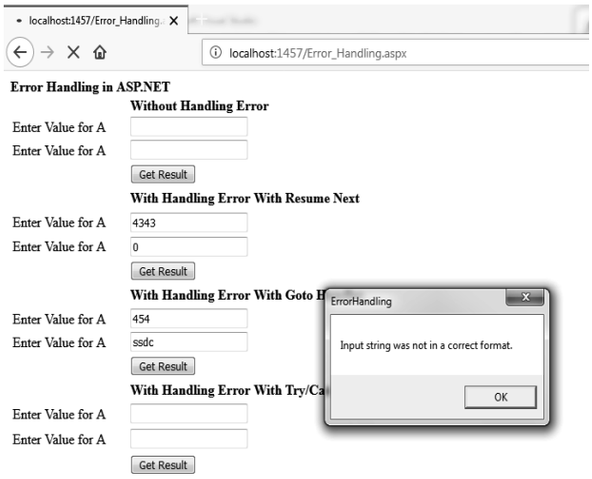
The data entered has generated unhandled error as shown below:



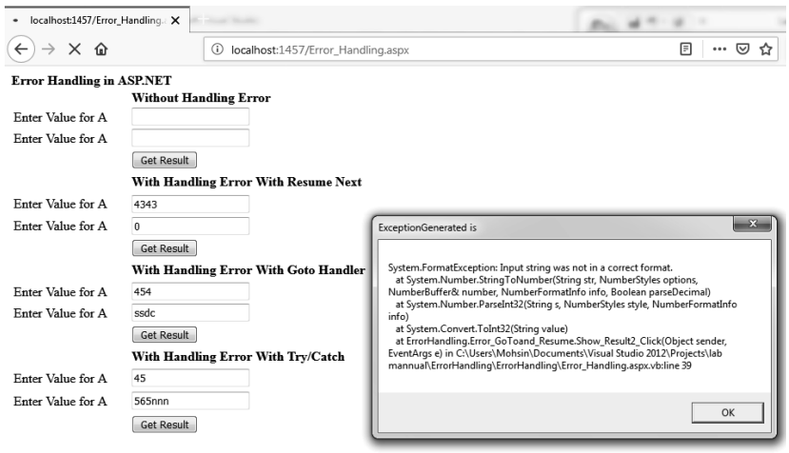
b. With Resume Next



c. With GoTo Handler



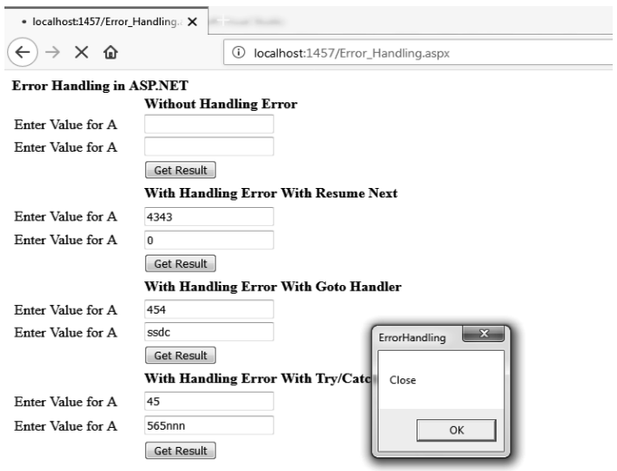
d. Next with Try-Catch



NOTES

Finally

NOTES



PRILIMNARY REQUIREMENTS FOR ADO.NET PROJECTS:

- MS SQLServer installed on your Computer
- Database created to perfrom ADO operations in VB

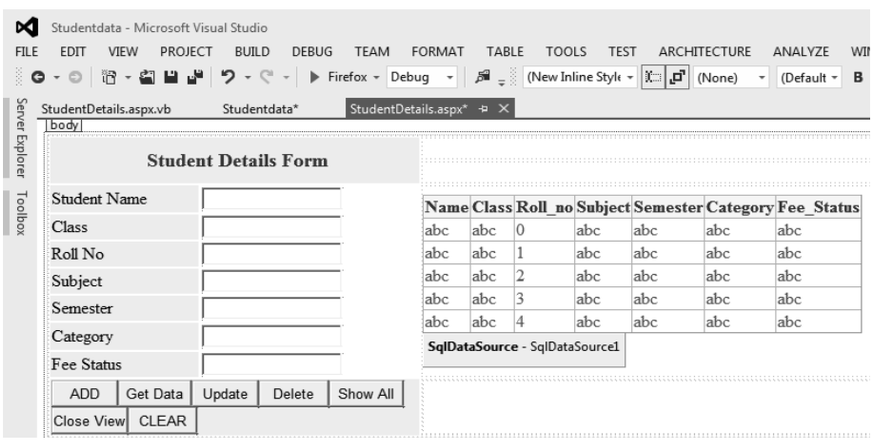
42. Write a program on database applications using ADO.NET to perfrom various database access operations using SQL commands, data binding controls and DataList.

Step 1: Create a new Project using ASP.NET EMPTY WEB FORM and name it as "Studentdata".

Step 2: Right click on "Studentdata" project in Solution Explorer to add a Windows Form as "StudentDetails.aspx".

Step 3: Design "StudentDetails.aspx" as shown below by using various controls from toolbox like Textbox, Labels, GridView, buttons and table as shown in figure below.

StudentDetails.aspx[Design]



Step 4: Next create a database (in this case MS SQLServer) table to store data and manuplate data that is expected to be generated from this web application by performing below step:

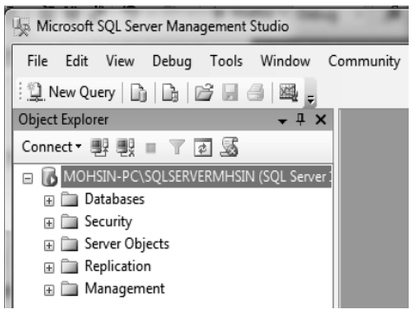
NOTES

a. Install MS SQL Server (SQL Server 2008).

b. Follow the steps as and when asked to complete installation.

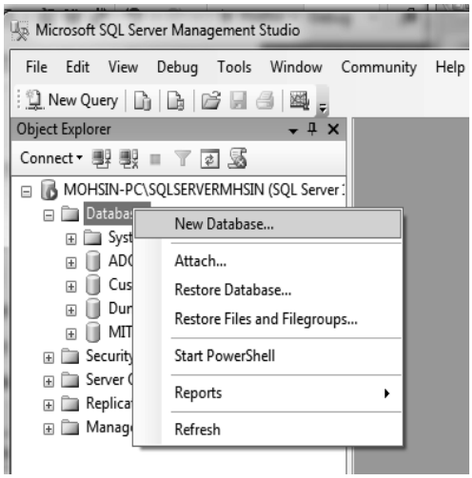
c. Open SQL Server installed as shown below:

d. Click on Connect to enter into SQL server management as shown below:

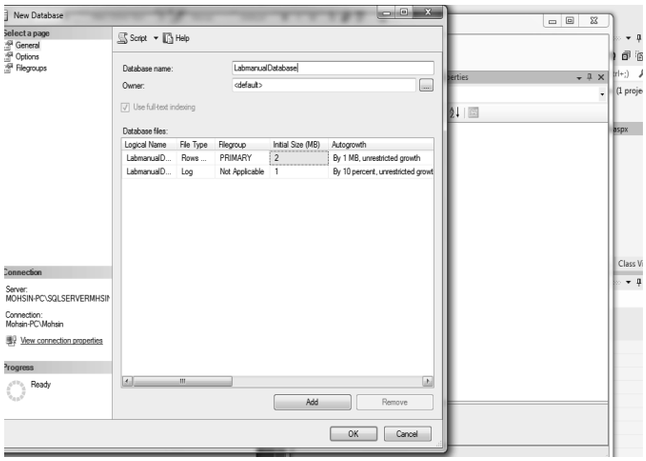


e. Right click on Databases in Object Explorer (Left side) then Click on Add New Database as shown below:

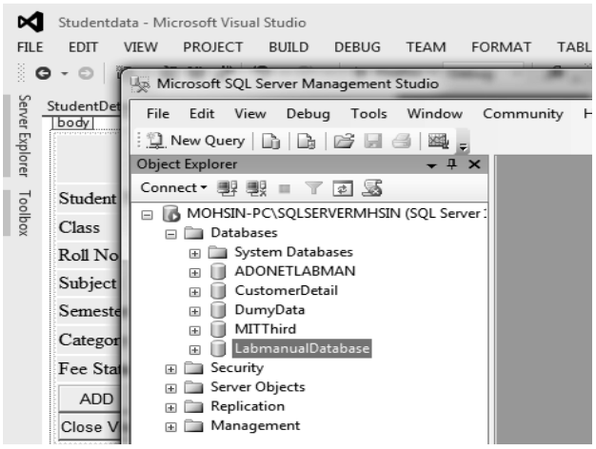
NOTES



f. Next enter database name and other details (if required) Click Ok to get it created as shown below:

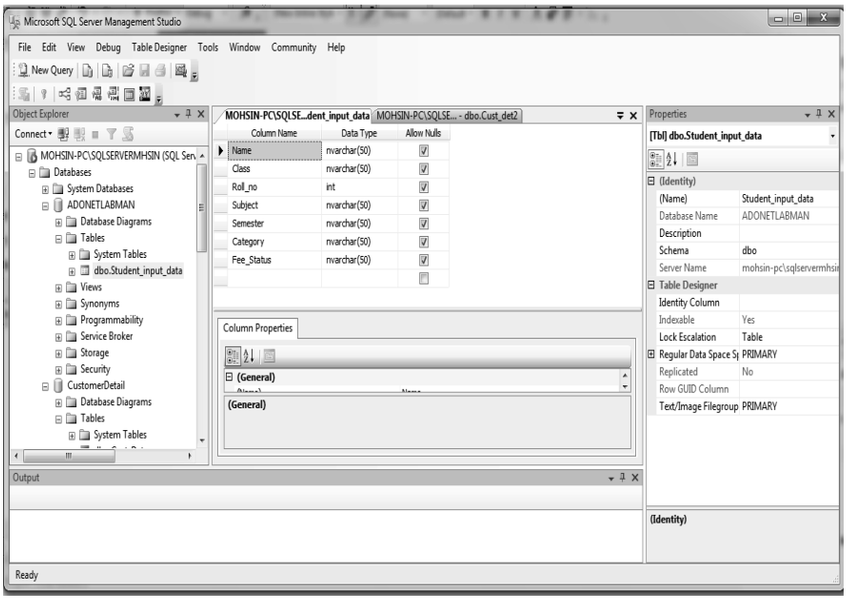


Next you will see the database "labmanualDatabase" created in object explorer.



NOTES

g. To create table within the database click on the particular database, a list of folders gets populated, right click on tables then add new table. Create the table as per the data expected to be manuplated from form. In case of this web project "Studentdata" the table created looks like as shown below:



h. Your are now Done with Database and again go to StudentDetails.aspx

Step 5: Connect the DataGrid with the created database the same way we did earlier with datacontrols.

Step 6: The source code of StudentDetails.aspx[Design] is given below:

StudentDetails.aspx

```
<%@ Page Language="vb" AutoEventWireup="false"
CodeBehind="StudentDetails.aspx.vb"
Inherits="Studentdata.WebForm1" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title></title>
   <style type="text/css">
      .auto-style1 {
         width: 100%;
      }
      .auto-style7 {
         width: 133px;
         height: 26px;
```

NOTES

```
    background-color: #FFFF99;
}
.auto-style8 {
    width: 197px;
    height: 26px;
    background-color: #FFFFFF;
}
.auto-style9 {
}
.auto-style6 {
    height: 23px;
    width: 133px;
    background-color: #FFFF99;
}
.auto-style4 {
    width: 197px;
    height: 23px;
    background-color: #FFFFFF;
}
.auto-style10 {
    color: #0000FF;
    text-align: center;
}
.auto-style11 {
    background-color: #FFFF99;
}
.auto-style12 {
    color: #800000;
}
.auto-style13 {
    width: 197px;
    background-color: #FFFFFF;
}
.auto-style14 {
    height: 30px;
```

NOTES

```
        background-color: #FFFF99;
      }
      .auto-style15 {
        height: 30px;
      }
      .auto-style16 {
        width: 133px;
        background-color: #FFFF99;
      }
      </style>
</head>
<body>
   <form id="form2" runat="server">
      <div>
        <table class="auto-style1" draggable="false">
          <tr>
          <td class="auto-style11" colspan="2"
rowspan="2"><h3 class="auto-style10"> <span
class="auto-style11">Student Details Form</span></h3>
</td>
          <td> </td>
          </tr>
          <tr>
          <td> </td>
          </tr>
          <tr>
          <td class="auto-style7">Student Name</td>
          <td class="auto-style8">
               <asp:TextBox ID="txtName" runat="server"
               CssClass="auto-style12"></asp:TextBox>
          </td>
          <td class="auto-style9" rowspan="7">
               <asp:GridView ID="GridView1"
runat="server" AutoGenerateColumns="False"
DataSourceID="SqlDataSource1" Visible="False"
ForeColor="#CC3300">
        <Columns>
```

NOTES

```
<asp:BoundField DataField="Name" HeaderText="Name"
SortExpression="Name" />
<asp:BoundField DataField="Class"
HeaderText="Class" SortExpression="Class" />
<asp:BoundField DataField="Roll_no"
HeaderText="Roll_no" SortExpression="Roll_no"
/>
<asp:BoundField DataField="Subject"
HeaderText="Subject" SortExpression="Subject"
/>
<asp:BoundField DataField="Semester"
HeaderText="Semester"
SortExpression="Semester" />
<asp:BoundField DataField="Category"
HeaderText="Category"
SortExpression="Category" />
<asp:BoundField DataField="Fee_Status"
HeaderText="Fee_Status" SortExpression="Fee_
Status" />
</Columns>
<HeaderStyle BackColor="#FFFFCC"
BorderStyle="Solid" ForeColor="Blue" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" run-
at="server" ConnectionString="<%$ Connec-
tionStrings:ADONETLABMANConnectionString %>"
SelectCommand="SELECT * FROM [Student_in-
put_data]"></asp:SqlDataSource>
</td>
</tr>
<tr>
<td class="auto-style16">Class</td>
<td class="auto-style13">
<asp:TextBox ID="txtClass" runat="server"
CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style16">Roll No</td>
<td class="auto-style13">
```

```
      <asp:TextBox ID="txtRoll" runat="server"
      CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style16">Subject</td>
<td class="auto-style13">
      <asp:TextBox ID="txtSub" runat="server"
      CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style6">Semester</td>
<td class="auto-style4">
      <asp:TextBox ID="txtSem" runat="server"
      CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style7">Category</td>
<td class="auto-style8">
      <asp:TextBox ID="txtCat" runat="server"
      CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style6">Fee Status</td>
<td class="auto-style4">
      <asp:TextBox ID="txtFeeSt" runat="server"
      CssClass="auto-style12"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style14" colspan="2">
      <asp:Button  ID="Button_ADD"
      runat="server" Text="ADD" Width="61px"
      />
```

NOTES

NOTES

```
            <asp:Button ID="Button_GetData" runat="server"
            Text="Get Data" Width="67px" />
            <asp:Button  ID="Button2"  runat="server"
            Text="Update" />
            <asp:Button ID="Button_Delete" runat="server"
            Text="Delete" Width="64px" />
            <asp:Button ID="Button_ShowAll" runat="server"
            Text="Show All" Width="68px" />
            <asp:Button ID="Button_Close" runat="server"
            Text="Close View" Visible="False" Width="70px"
            Height="26px" />
            <asp:Button  ID="Button1"  runat="server"
            Text="CLEAR" />
            </td>
            <td class="auto-style15"></td>
            </tr>
         </table>
      </div>
   </form>
   <p>
       </p>
</body>
</html>
```

Step 6: Customize the functionality specified against each button with particular rule in codebehind window of StudentDetails.aspx as shown below.

```
'Database applications using ADO.NET
'Accessing a database using SQL commands, Data binding
controls, DataList
Imports System.Data.SqlClient ' Decleration or importing
necessary namespaces to inheret the functions required
'to perfrom database related operations
Imports System.Data.SqlClient.SqlDataReader
Imports System.Web
Imports System.Data
Imports System
Public Class WebForm1
```

NOTES

```
Inherits System.Web.UI.Page
'Creating Connection String to establish the communnication with the database created for storing data
Dim DBConn As SqlConnection = New SqlConnection("Data Source=MOHSIN-PC\SQLSERVERMHSIN;Initial Catalog=ADONETLABMAN;Integrated Security=True")
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
End Sub
Protected Sub Button_ADD_Click(sender As Object, e As EventArgs) Handles Button_ADD.Click
    Dim DBConn As SqlConnection ' creating Sql Connection object
    DBConn = New SqlConnection("Data Source=MOHSIN-PC\SQLSERVERMHSIN;Initial Catalog=ADONETLABMAN;Integrated Security=True")
    Dim Name As String = txtName.Text ' Declearing text boxes as per the requiremet to capture user data
    Dim Clas As String = txtClass.Text
    Dim RollNo As String = (txtRoll.Text)
    Dim Subject As String = txtSub.Text
    Dim semester As String = txtSem.Text
    Dim category As String = txtCat.Text
    Dim feestatus As String = txtFeeSt.Text
    ' Creating a specific sql query a
    Dim Query As String = "insert into Student_input_data values ('" & Name & "','" & Clas & "'," & RollNo & ",'" & Subject & "','" & semester & "','" & category & "','" & feestatus & "')"
    Dim DBInsert As SqlCommand = New SqlCommand(Query, DBConn) ' Creating Sql Command and feeding Sql query and Connection string as arguments
    DBInsert.CommandType = CommandType.Text
    DBConn.Open() ' opening the connection with database specifed
    If (DBInsert.ExecuteNonQuery().Equals(1)) Then ' executing the sqlnonquery string
        MsgBox("Information stored in database")
```

NOTES

```
        Else
           MsgBox("Not stored in database")
        End If
    End Sub
    Protected Sub Button_GetData_Click(sender As Object,
e As EventArgs) Handles Button_GetData.Click
        Dim Query2 As String = "SELECT * FROM Student_input_
data where Name = '" & txtName.Text & "'"
        DBConn.Open()
        Dim DBCmd As SqlCommand = New SqlCommand(Query2,
DBConn)
        Dim datareader As SqlDataReader ' declearing sql
datareader to handle the data to be retrived by select
query from the database
        datareader = DBCmd.ExecuteReader()
        If datareader.Read = True Then ' checking wheather
the query retrived any valid data if yes then handling
the same
           txtName.Text = Convert.ToString(datareader(0))
           txtClass.Text = Convert.ToString(datareader(1))
           txtRoll.Text = Convert.ToString(datareader(2))
           txtSub.Text = Convert.ToString(datareader(3))
           txtSem.Text = Convert.ToString(datareader(4))
           txtCat.Text = Convert.ToString(datareader(5))
           txtFeeSt.Text = Convert.ToString(datareader(6))
        Else
           MsgBox("No data Retrieved, Please enter Vaild
Search String")
        End If
        DBConn.Close()
    End Sub
    Protected Sub Button_Delete_Click(sender As Object,
e As EventArgs) Handles Button_Delete.Click
        ' performing sql delete operation on the stored
data in database
        Dim Query3 As String = "Delete FROM Student_input_
data where Name = '" & txtName.Text & "'"
        DBConn.Open()
```

NOTES

```
    Dim DBCmd As SqlCommand = New SqlCommand(Query3,
DBConn)
    DBCmd.ExecuteNonQuery()
    MsgBox("Information Deleted sucessfully from
database")
    DBConn.Close()
  End Sub
  Protected Sub Button_Close_Click(sender As Object, e
As EventArgs) Handles Button_Close.Click
    ' creating a databinding control that is gridview
to display all the data stored in database
    GridView1.Visible = False
    Button_ShowAll.Visible = True
    Button_Close.Visible = False
  End Sub
  Protected Sub Button_ShowAll_Click(sender As Object,
e As EventArgs) Handles Button_ShowAll.Click
    GridView1.Visible = True
    Button_ShowAll.Visible = False
    Button_Close.Visible = True
    GridView1.DataBind()
  End Sub
  Protected Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
    'implementing the Clear button
    txtCat.Text = ""
    txtClass.Text = ""
    txtName.Text = ""
    txtRoll.Text = ""
    txtSem.Text = ""
    txtSub.Text = ""
    txtFeeSt.Text = " "
  End Sub
  Protected Sub Button2_Click(sender As Object, e As
EventArgs) Handles Button2.Click
    'perfroming update operation if required on any of
the data variables after retriving data using get data
```

NOTES

```
      Dim Query4 As String = "Update Student_input_data
set Name = '" & txtName.Text & "', Class='" & txtClass.
Text & "', Roll_no= '" & txtRoll.Text & "',  Subject='"
& txtSub.Text & "', Semester='" & txtSem.Text & "',
Category='" & txtCat.Text & "', Fee_status='" & txtFeeSt.
Text & "' where Name = '" & txtName.Text & "'"
      DBConn.Open()
      Dim DBCmd As SqlCommand = New SqlCommand(Query4,
DBConn)
      DBCmd.ExecuteNonQuery()
      MsgBox("Information Updated sucessfully from
database")
      DBConn.Close()
   End Sub
End Class
```

Step 7: In order to connect the database with ADO.NET application, one needs to do so by finding the ConnectionString and use the same in SqlConnection object to fullfil successful connectivity.

To obtain ConnectionString apply the following steps.

a. Click on ServerExplorer on left top or from within the view menu.

b. Right click on Data Connections.

c. Click on Add New Connection

d. The dialogbox will be prompted like shown below:

NOTES

e. Enter server name.

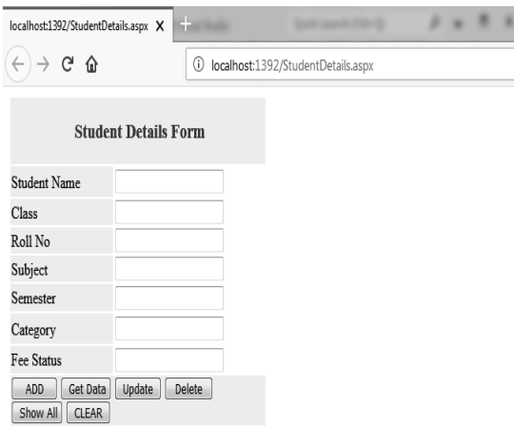
f. Click Ok and follow the subsequent Steps to make it done.

g. Once you are successfully connected with database, the same will now gets displayed in Server Explorer as shown in figure above.

h. Click on this added connection, go to properties→Connection→ConnectionString→Copy it.

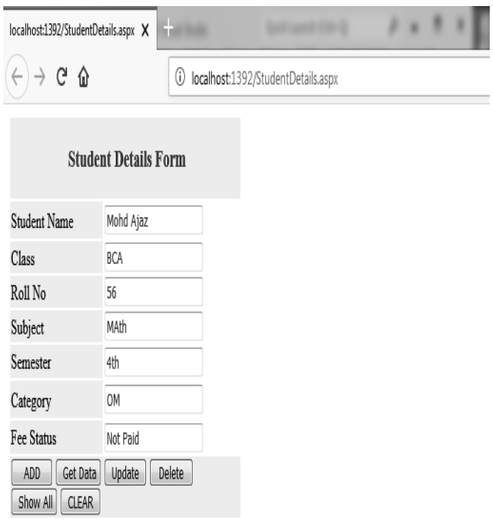
i. Use this copied ConnectionString as used in code behind of StudentDetails.aspx shown above.

Step 8: Cutomize Active page, Server settings and then build the project and start to get its funcutional results displayed in browser as shown below:

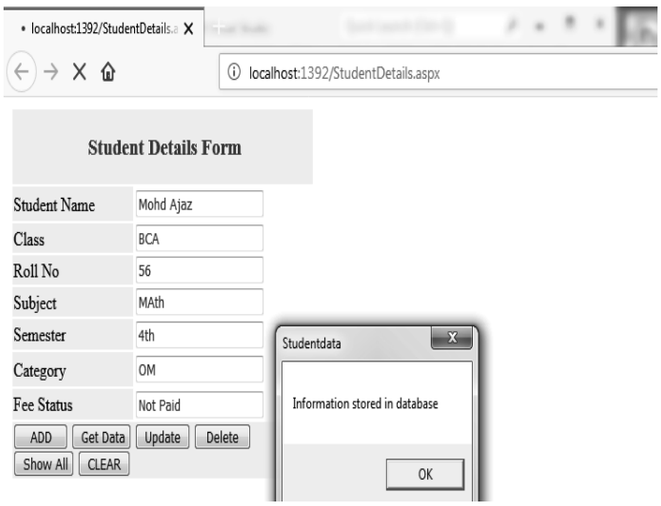


Next enter the details to be saved into database as shown below:

NOTES



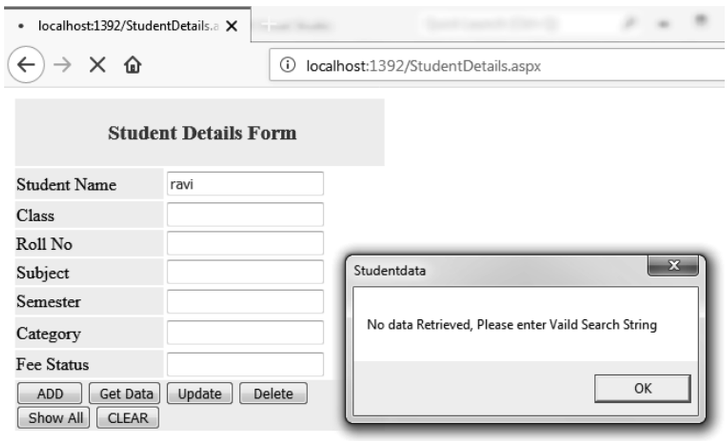
Next click on ADD button to invoke SQL command "Insert into Table" to store the filled in data into database, the successful insert will be acknowledged as:



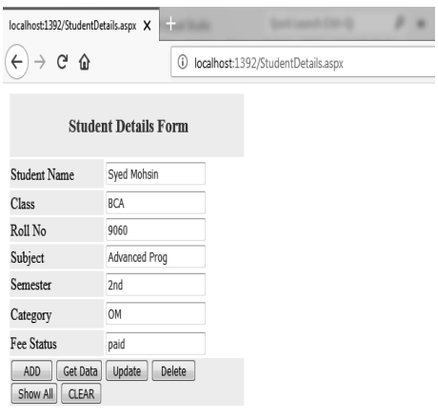
Next you can perfrom any of the operation specified through buttons like Get Data, Update, Delete, Show All, CLEAR by clicking on them separatively.

In order to get data from the database you need to specify the search string in Student Name textbox. If this matches with database entry the required results gets displayed as shown below:

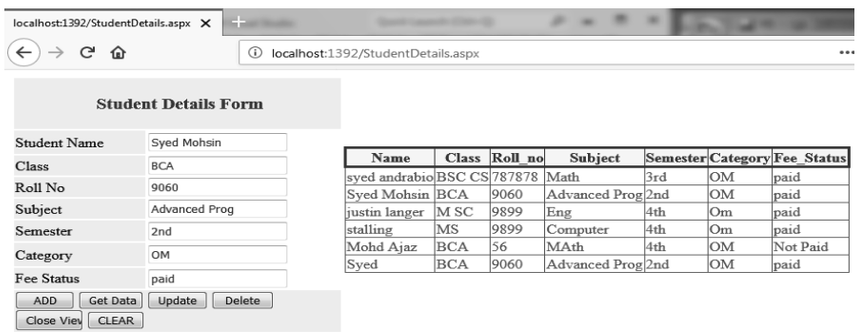
NOTES



Next enter valid stored name the details shown below are retrived.



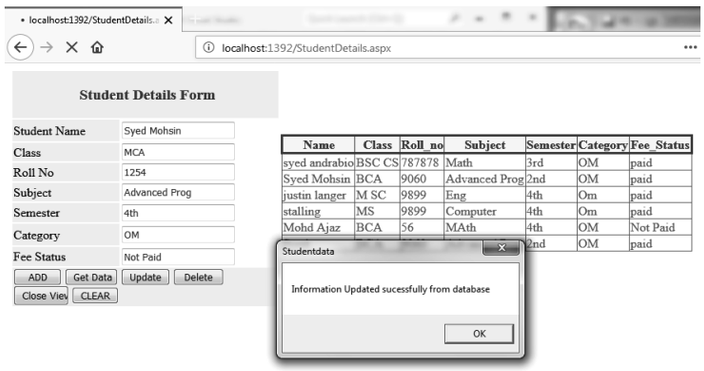
Next click on Show All to retrieve all data from the database to get displayed through datagrid as shown below:



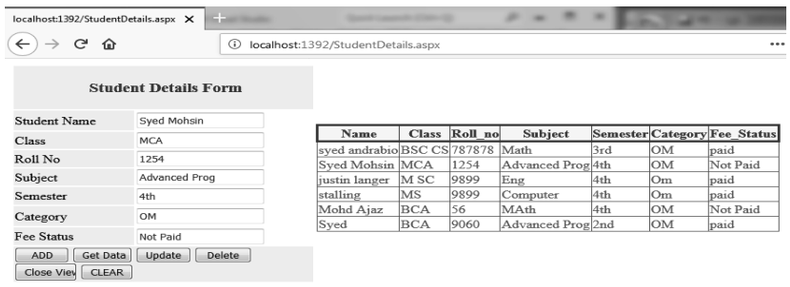
NOTES

You can see the Show All button disapers and Close View appears.

Next to update any stored details against any name, enter the Same name in Student Name textbox , click on Get Data button to retrieve data then update the information required to be changed, After making changes click on "Update" button to save this back into database as shown below:



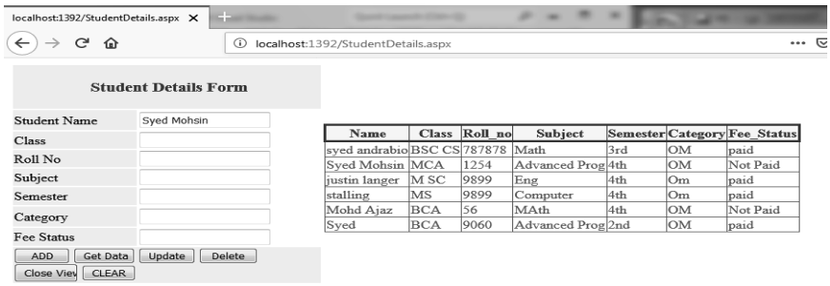
Next click Ok then "Close View", the "Show All" Button appears again click on it to see the updated information against Syed Mohsin record as shown below:



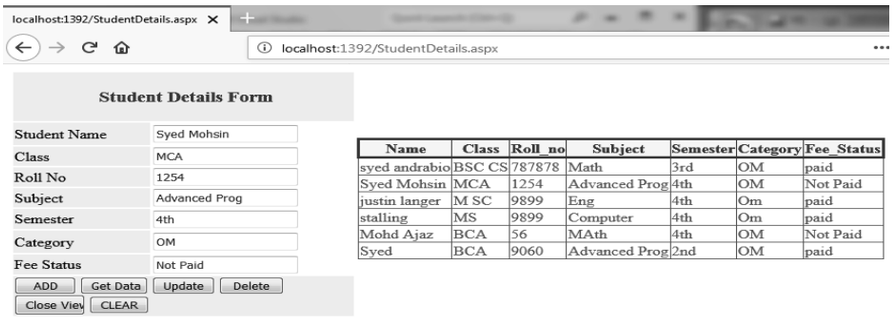
You can compare the above two DataGrids to see the update.

Next to delete any record from the database, you first need to specify the data field in Student Name text box then click get data then delete. After the delete is successful refresh the GridView by Close View and then Show All to see whether the data field has been deleted as shown below.

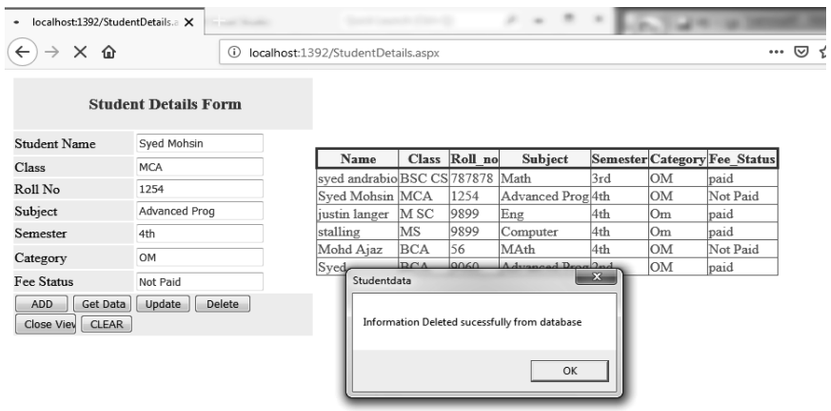
Lets delete Syed Mohsin record from database.



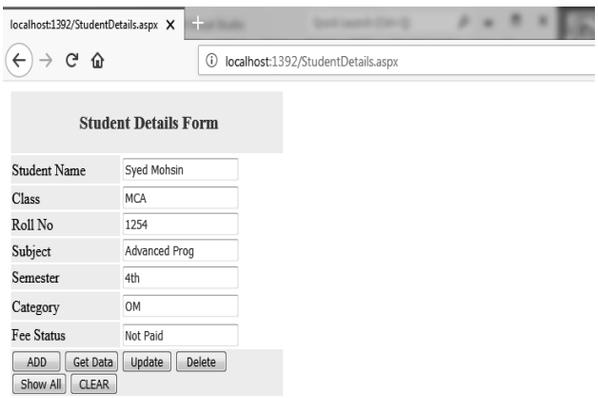
Next click “Get Data”.



Next click on delete.



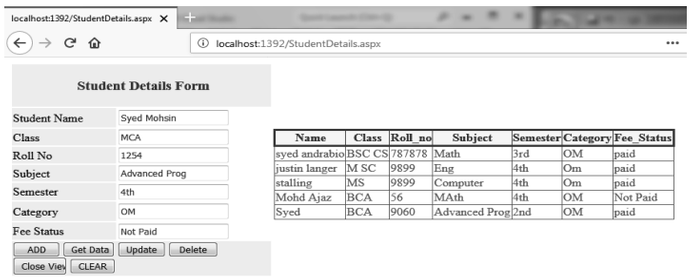
Next Close GridView.



NOTES

NOTES

Next click Show All.



You will see in GridView Syed Mohsin record is deleted from the database.

M.Sc. [Computer Science]
341 24
LAB – . NET PROGRAMMING

II - Semester

ALAGAPPA UNIVERSITY

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle and Graded as Category–I University by MHRD-UGC]

KARAIKUDI – 630 003

DIRECTORATE OF DISTANCE EDUCATION

ISBN 978-93-5338-516-3