C# Project

ConvertXCalc

Arithmetic page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace CodeWeb
{   internal class Arithmetic
  {
    public string NumberToString(int number)
    {
      return number.ToString();
    }

    public int StringToNumber(string strNumber)
    {
      if (int.TryParse(strNumber, out int result))
      {
        return result;
      }

      Console.WriteLine(" Please enter a valid Number.");
      return 0;
    }

    public string Add(string num1, string num2)
    {
      int intNum1 = StringToNumber(num1);
      int intNum2 = StringToNumber(num2);

      return NumberToString(intNum1 + intNum2);
    }

    public string Subtract(string num1, string num2)
    {
      int intNum1 = StringToNumber(num1);
```

```
    int intNum2 = StringToNumber(num2);

    return NumberToString(intNum1 - intNum2);
}

public string Multiply(string num1, string num2)
{
    int intNum1 = StringToNumber(num1);
    int intNum2 = StringToNumber(num2);

    return NumberToString(intNum1 * intNum2);
}

public string Divide(string num1, string num2)
{
    int intNum1 = StringToNumber(num1);
    int intNum2 = StringToNumber(num2);

    if (intNum2 == 0)
    {
        Console.WriteLine("Cannot divide by zero.");
        return "Error";
    }

    return NumberToString(intNum1 / intNum2);
}

public string ConvertToRupees(string result)
{
    if (int.TryParse(result, out int numericResult))
    {
        return ConvertToRupees(numericResult);
    }

    return " Invalid result for conversion.";
}

public string ConvertToRupees(int numericResult)
{
```

```
if (numericResult == 0)
{
    return "Zero";
}

if (numericResult < 0)
{
    return "Negative ";
}

string rupeesInWords = "";

// Convert Crores
if (numericResult >= 10000000)
{
    int crores = numericResult / 10000000;
    rupeesInWords += $"{ConvertToWords(crores)} Crore ";
    numericResult %= 10000000;
}

// Convert Lakhs
if (numericResult >= 100000)
{
    int lakhs = numericResult / 100000;
    rupeesInWords += $"{ConvertToWords(lakhs)} Lakh ";
    numericResult %= 100000;
}

// Convert Thousands
if (numericResult >= 1000)
{
    int thousands = numericResult / 1000;
    rupeesInWords += $"{ConvertToWords(thousands)} Thousand ";
    numericResult %= 1000;
}

// Convert Ones
if (numericResult > 0)
{
```

```
            rupeesInWords += ConvertToWords(numericResult);
        }

        return $"{rupeesInWords}";
    }
    private string ConvertToWords(int number)
    {
        string[] ones = { "", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine" };
        string[] teens = { "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };
        string[] tens = { "", "Ten", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };

        string result = "";

        // Convert hundreds
        if (number >= 100)
        {
            result += $"{ones[number / 100]} Hundred ";
            number %= 100;
        }

        // Convert tens and ones
        if (number > 0)
        {
            if (number >= 11 && number <= 19)
            {
                result += $"{teens[number - 11]} ";
            }
            else
            {
                result += $"{tens[number / 10]} {ones[number % 10]} ";
            }
        }

        return result;
    }
  }
}
```

Binary Converter page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class BinaryConverter
    {
        private string binaryNumber;

        // Constructor
        public BinaryConverter(string binaryNumber)
        {
            this.binaryNumber = binaryNumber;
        }

        // Convert to Decimal
        public int ToDecimal()
        {
            return Convert.ToInt32(binaryNumber, 2);
        }

        // Convert to Octal
        public string ToOctal()
        {
            int decimalValue = ToDecimal();
            return Convert.ToString(decimalValue, 8);
        }

        // Convert to Hexadecimal
        public string ToHexadecimal()
        {
            int decimalValue = ToDecimal();
            return Convert.ToString(decimalValue, 16).ToUpper();
        }

        // Convert to ASCII
        public string ToAscii()
        {
            string asciiResult = "";
            for (int i = 0; i < binaryNumber.Length; i += 8)
            {
                string binaryByte = binaryNumber.Substring(i, Math.Min(8, binaryNumber.Length - i));
                int decimalValue = Convert.ToInt32(binaryByte, 2);
                char asciiChar = (char)decimalValue;
                asciiResult += asciiChar;
            }
            return asciiResult;
        }

        // Convert Decimal to Binary
        public string FromDecimalToBinary(int decimalInput)
        {
            return Convert.ToString(decimalInput, 2);
        }

        // Convert Decimal to Octal
        public string FromDecimalToOctal(int decimalInput)
        {
            return Convert.ToString(decimalInput, 8);
        }

        // Convert Decimal to Hexadecimal
        public string FromDecimalToHexadecimal(int decimalInput)
        {
            return Convert.ToString(decimalInput, 16).ToUpper();
        }

        // Convert Decimal to ASCII
        public string FromDecimalToAscii(int decimalInput)
        {
            return ((char)decimalInput).ToString();
        }
    }
}
```

DigitToWord page

```
using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class DigitToWord
    {
        String[] nums = { "Zero", "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven",
"Twelve", "Thirteen", "FourTeen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };

        String[] Tens = { "", "", "Twenty", "Thirty", "Forty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };

        internal String ConvertNum(double number)
        {
            long numint = (long)number;
            long numdec = (long)Math.Round((number - (double)numint) * 100);
            if (numdec == 0)
            {
                return Convert(numint);
            }
            else
            {
                return Convert(numint) + " Point " + ConvertDecimal(numdec);
            }
        }

        internal String Convert(long i)
        {
            if (i < 10)
            {
                return nums[i];
            }

            if (i < 100)
            {
                return Tens[i / 10] + ((i % 10 > 0) ? " " + Convert(i % 10) : "");
            }

            if (i < 1000)
            {
                return nums[i / 100] + " Hundred" + ((i % 100 > 0) ? " " + Convert(i % 100) : "");
            }
            if (i < 100000)
            {
                return Convert(i / 1000) + " Thousand" + ((i % 1000 > 0) ? " " + Convert(i % 1000) : "");
            }
            if (i < 10000000)
            {
                return Convert(i / 100000) + " Lakh" + ((i % 100000 > 0) ? " " + Convert(i % 100000) : "");
            }
            return Convert(i / 10000000) + " Crore" + ((i % 10000000 > 0) ? " " + Convert(i % 10000000) : "");

        }
        internal String ConvertDecimal(long i)
        {
            if (i < 10)
            {
                return Tens[i / 10];
            }
            else
            {
                return nums[i / 10] + ((i % 10 > 0) ? " " + Convert(i % 10) : " ");
            }
        }
    }
}
```

Footer page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class footer
    {
        internal void footerPage()
        {
            Console.SetCursorPosition(0, 60);

Console.Write("**********************************************************************************************
**********************************************************************************************
******");

            Console.SetCursorPosition(16, 62);
```

```
            Console.Write("   Some Information ");
            Console.SetCursorPosition(10, 63);
            Console.Write("-----------------------------------");
            Console.SetCursorPosition(11, 64);
            Console.Write("DevSMITH is a next-generation");
            Console.SetCursorPosition(11, 65);
            Console.Write("IT-Company that helps enterprises");
            Console.SetCursorPosition(11, 66);
            Console.Write("reimagine their businesses for");
            Console.SetCursorPosition(11, 67);
            Console.Write("the digital age. DevSmith is ");
            Console.SetCursorPosition(11, 68);
            Console.Write("started in 2023.");



            Console.SetCursorPosition(92, 62);
            Console.Write("   Links   ");
            Console.SetCursorPosition(80, 63);
            Console.Write("-----------------------------------");
            Console.SetCursorPosition(95, 64); Console.Write("HOME");
            Console.SetCursorPosition(95, 65); Console.Write("About Us");
            Console.SetCursorPosition(95, 66); Console.Write("Services");
            Console.SetCursorPosition(95, 67); Console.Write("Our Team");
            Console.SetCursorPosition(160, 62);
            Console.Write("   Contact Us  ");
            Console.SetCursorPosition(150, 63);
            Console.Write("-----------------------------------");
            Console.SetCursorPosition(152, 64); Console.Write("DevSmith Software Solutions PVT,LTD");
            Console.SetCursorPosition(152, 65); Console.Write("Office:- Sitapur Chitrakoot (UP)");
            Console.SetCursorPosition(152, 66); Console.Write("Email:- devsmith123@gmail.com");
            Console.SetCursorPosition(152, 67); Console.Write("Contact:- +91 9918278412");
            Console.SetCursorPosition(0, 69);

Console.Write("**********************************************************************************************
**********************************************************************************************************
******");



        }
    }
}
```

Guidelines Page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class GuideLine
    {
        internal void Guide()
        {
            // Set the encoding of the console to UTF-8
            Console.OutputEncoding = Encoding.UTF8;
            // Console.WindowWidth = Console.LargestWindowWidth;

            // Set the title of the console window
            Console.Title = "Project Guidelines";

            // Set the foreground color to white and the background color to dark blue
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.BackgroundColor = ConsoleColor.Black;

            // Clear the console
            Console.Clear();

            // Write the header of the page
            Console.WriteLine();
            Console.WriteLine("
************************************************************");
            Console.WriteLine("                                                *                  PROJECT GUIDELINES
*");
            Console.WriteLine("
************************************************************");
            Console.WriteLine();

            Console.WriteLine("
==============================================================================================================
=");
            Console.WriteLine();
            Console.WriteLine();
```

```
            Console.WriteLine("            \u2660 In English, a number to word converter in C# is a program that converts a
number into its English word representation. It helps\r\n \t       in reading and understanding numeric values by
representing them in words.");
            Console.WriteLine("            \u2660 Example: Enter Number:45 ; Number in Words: Forty-five.");
            Console.WriteLine();
            Console.WriteLine("            \u2665 A Currency Converterin C#is a program that converts one currency into
another.It helps in understanding the value of a currency\r\n \t       in a different currency. This is particularly useful for
international transactions or when you want to understand the value of\r\n \t       foreign Currency.");
            Console.WriteLine("            \u2665 Example: Currency in $: 10 ; Currency in Rupees: 820.");
            Console.WriteLine();
            Console.WriteLine("            \u2666 A Binary Converter in C# is a program that converts decimal numbers into
binary numbers. It allows us to represent a number in \r\n \t       it binary form. This is particularly useful when we want to
read, understand, or manipulate binary numbers.");
            Console.WriteLine("            \u2666 Example: Enter the number: 99 ; Binary Number of Entered number: 1100011 ");
            Console.WriteLine();
            Console.WriteLine("            \u2663 A Roman to number and number to roman converter is a prgram that converts the
given number into Roman numbers and vice-versa.It\r\n\t       helps to read and write dates, names, titles, and other
expressions that use roman numerals ");
            Console.WriteLine("            \u2663 Example: Enter number: 25 ; Roman form : XXV ");
            Console.WriteLine("            \u2663 Example: Enter Roman Number: L ; Number form: 50;");
            Console.WriteLine();
            Console.WriteLine("            * An arithmetic converter is a tool that can help you perform arithmetic operations
such as addition, subtraction, multiplication,\r\n\t       and division on different types of numbers, such as decimals,
fractions, percentages, and more. ");
            Console.WriteLine("            * Example: Choose the operation: * ; Enter the numbers: 5,7 ; Result: 35");
            Console.WriteLine();
            Console.WriteLine();


            Console.WriteLine("
==========================================================================================================================================
=");
            // Write a footer of the page
            Console.WriteLine();
            Console.WriteLine("
************************************************************");
            Console.WriteLine("                                                *                END OF PROJECT GUIDELINES
*");
            Console.WriteLine("
************************************************************");

            // Wait for the user to press any key to exit
            Console.WriteLine();
            Console.WriteLine("            Press any key to exit...");
        }
    }
}
```

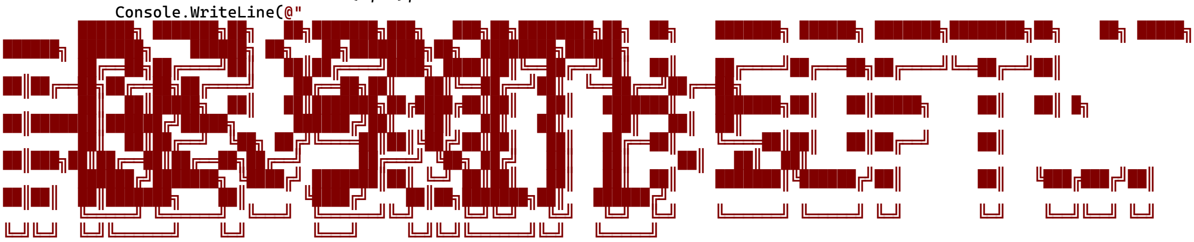
Guidelines page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class GuideLine
    {
        internal void Guide()
        {
            // Set the encoding of the console to UTF-8
            Console.OutputEncoding = Encoding.UTF8;
            // Console.WindowWidth = Console.LargestWindowWidth;

            // Set the title of the console window
            Console.Title = "Project Guidelines";

            // Set the foreground color to white and the background color to dark blue
            Console.ForegroundColor = ConsoleColor.DarkYellow;
            Console.BackgroundColor = ConsoleColor.Black;

            // Clear the console
            Console.Clear();

            // Write the header of the page
            Console.WriteLine();
            Console.WriteLine("                                                ************************************************************");
            Console.WriteLine("                                                *                    PROJECT GUIDELINES                    *");
            Console.WriteLine("                                                ************************************************************");
            Console.WriteLine();

            Console.WriteLine("
==========================================================================================================================================");
            Console.WriteLine();
            Console.WriteLine();
            Console.WriteLine("            \u2660 In English, a number to word converter in C# is a program that converts a number into its English word
representation. It helps\r\n \t       in reading and understanding numeric values by representing them in words.");
            Console.WriteLine("            \u2660 Example: Enter Number:45 ; Number in Words: Forty-five.");
            Console.WriteLine();
            Console.WriteLine("            \u2665 A Currency Converterin C#is a program that converts one currency into another.It helps in understanding
the value of a currency\r\n \t       in a different currency. This is particularly useful for international transactions or when you want to understand
the value of\r\n \t       foreign Currency.");
            Console.WriteLine("            \u2665 Example: Currency in $: 10 ; Currency in Rupees: 820.");
            Console.WriteLine();
            Console.WriteLine("            \u2666 A Binary Converter in C# is a program that converts decimal numbers into binary numbers. It allows us
to represent a number in \r\n \t       it binary form. This is particularly useful when we want to read, understand, or manipulate binary numbers.");
            Console.WriteLine("            \u2666 Example: Enter the number: 99 ; Binary Number of Entered number: 1100011 ");
            Console.WriteLine();
            Console.WriteLine("            \u2663 A Roman to number and number to roman converter is a prgram that converts the given number into Roman
numbers and vice-versa.It\r\n\t       helps to read and write dates, names, titles, and other expressions that use roman numerals ");
            Console.WriteLine("            \u2663 Example: Enter number: 25 ; Roman form : XXV ");
            Console.WriteLine("            \u2663 Example: Enter Roman Number: L ; Number form: 50;");
            Console.WriteLine();
```

```
            Console.WriteLine("            * An arithmetic converter is a tool that can help you perform arithmetic operations such as addition, subtraction, multiplication,\r\n\t       and division on different types of numbers, such as decimals, fractions, percentages, and more. ");
            Console.WriteLine("            * Example: Choose the operation: * ; Enter the numbers: 5,7 ; Result: 35");
            Console.WriteLine();
            Console.WriteLine();


            Console.WriteLine("
=================================================================================================================================");
            // Write a footer of the page
            Console.WriteLine();
            Console.WriteLine("                                                                ************************************************************");
            Console.WriteLine("                                                                *                   END OF PROJECT GUIDELINES              *");
            Console.WriteLine("                                                                ************************************************************");

            // Wait for the user to press any key to exit
            Console.WriteLine();
            Console.WriteLine("           Press any key to exit...");
        }
    }
}
```

Header page

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class header
    {
        internal void headerPage()
        {

            Console.ForegroundColor = ConsoleColor.Green;
            Console.SetCursorPosition(0, 3);
            Console.WriteLine(@"
```



```
");
            Console.SetCursorPosition(0, 12);

Console.Write("*************************************************************************************************************************************************************************************************************************");
            Console.SetCursorPosition(20, 13);
            Console.Write("1.HOME\t\t");
            Console.SetCursorPosition(35, 13);
            Console.Write("2.String\t\t");
            Console.SetCursorPosition(50, 13);
            Console.Write("3.Roman\t\t ");
            Console.SetCursorPosition(68, 13);
            Console.Write("4.Arithmetic\t");
            Console.SetCursorPosition(85, 13);
            Console.Write("    5.Binary");
            Console.SetCursorPosition(100, 13);
            Console.Write("    6.Currency");
            Console.SetCursorPosition(115, 13);
            Console.Write("    7.Guideline");
            //Console.SetBufferSize(500, 500);
            Console.SetCursorPosition(130, 13);
            Console.Write("    8.Our Team");
            Console.SetCursorPosition(150, 13);
            Console.Write("    9.About Us");
            Console.SetCursorPosition(0, 15);

Console.Write("*************************************************************************************************************************************************************************************************************************");


        }
    }
}
```

Number page

using System;

using System.Collections.Generic;

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class NumberConverter
    {
        private static readonly  Dictionary<char, int> RomanMap = new Dictionary <char, int>
    {
        {'I', 1},
        {'V', 5},
        {'X', 10},
        {'L', 50},
        {'C', 100},
        {'D', 500},
        {'M', 1000},

    };

        private static readonly string[] Symbol = { "I", "IV", "V", "IX", "X", "XL", "L", "XC", "C", "CD", "D", "CM", "M", "MMMM" };
        private static readonly int[] num = { 1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000, 4000 };

        public string IntegerToRoman(int n)
        {
            if (n < 1 || n > 10000000)
            {
                Console.WriteLine("Number must be between 1 and 10,000,000 for Roman numeral conversion.");
            }

            string result = "";
            int index = 13;

            while (n > 0)
            {
                int div = n / num[index];
                n %= num[index];
```

```
            for (int i = 0; i < div; i++)
            {
                result += Symbol[index];
            }

            index--;
        }

        return result;
    }

    public int RomanToInteger(string roman)
    {
        if (string.IsNullOrEmpty(roman))
        {
            Console.WriteLine("Roman numeral cannot be null or empty.", nameof(roman));
        }

        int result = 0;

        for (int i = 0; i < roman.Length; i++)
        {
            int currentSymValue = RomanMap[roman[i]];

            if (i + 1 < roman.Length && RomanMap[roman[i + 1]] > currentSymValue)
            {
                result -= currentSymValue;
            }
            else
            {
                result += currentSymValue;
            }
        }

        return result;
    }
  }
} using System;
```

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CodeWeb
{
    internal class NumberConverter
    {
        private static readonly  Dictionary<char, int> RomanMap = new Dictionary <char, int>
    {
        {'I', 1},
        {'V', 5},
        {'X', 10},
        {'L', 50},
        {'C', 100},
        {'D', 500},
        {'M', 1000},

    };

        private static readonly string[] Symbol = { "I", "IV", "V", "IX", "X", "XL", "L", "XC", "C", "CD", "D", "CM", "M", "MMMM" };
        private static readonly int[] num = { 1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500, 900, 1000, 4000 };

        public string IntegerToRoman(int n)
        {
            if (n < 1 || n > 10000000)
            {
                Console.WriteLine("Number must be between 1 and 10,000,000 for Roman numeral conversion.");
            }

            string result = "";
            int index = 13;

            while (n > 0)
            {
                int div = n / num[index];
                n %= num[index];
```

```
            for (int i = 0; i < div; i++)
            {
                result += Symbol[index];
            }

            index--;
        }

        return result;
    }

    public int RomanToInteger(string roman)
    {
        if (string.IsNullOrEmpty(roman))
        {
            Console.WriteLine("Roman numeral cannot be null or empty.", nameof(roman));
        }

        int result = 0;

        for (int i = 0; i < roman.Length; i++)
        {
            int currentSymValue = RomanMap[roman[i]];

            if (i + 1 < roman.Length && RomanMap[roman[i + 1]] > currentSymValue)
            {
                result -= currentSymValue;
            }
            else
            {
                result += currentSymValue;
            }
        }

        return result;
    }
}
```

}