Sue's Salon Invoice Program Guide

THIS USER GUIDE WILL DEMONSTRATE SOME KEY FEATURES

Chhun Chhim | COP2170 | 12/08/2016

Table of Contents

Introduction

Thank you for taking the time to familiarize yourself with our invoicing program. This guide is intended to help you navigate your way around the program and teach you about the key features.

This invoicing program was built to be user friendly and straight forward. The options and features was designed for people with little to no knowledge of how an invoicing program works. We guarantee that you will be able to start using this program when you are finished with this guide.

Thank you for purchasing this invoicing program and hope you find it to be a valuable tool for your business. Let's get started!
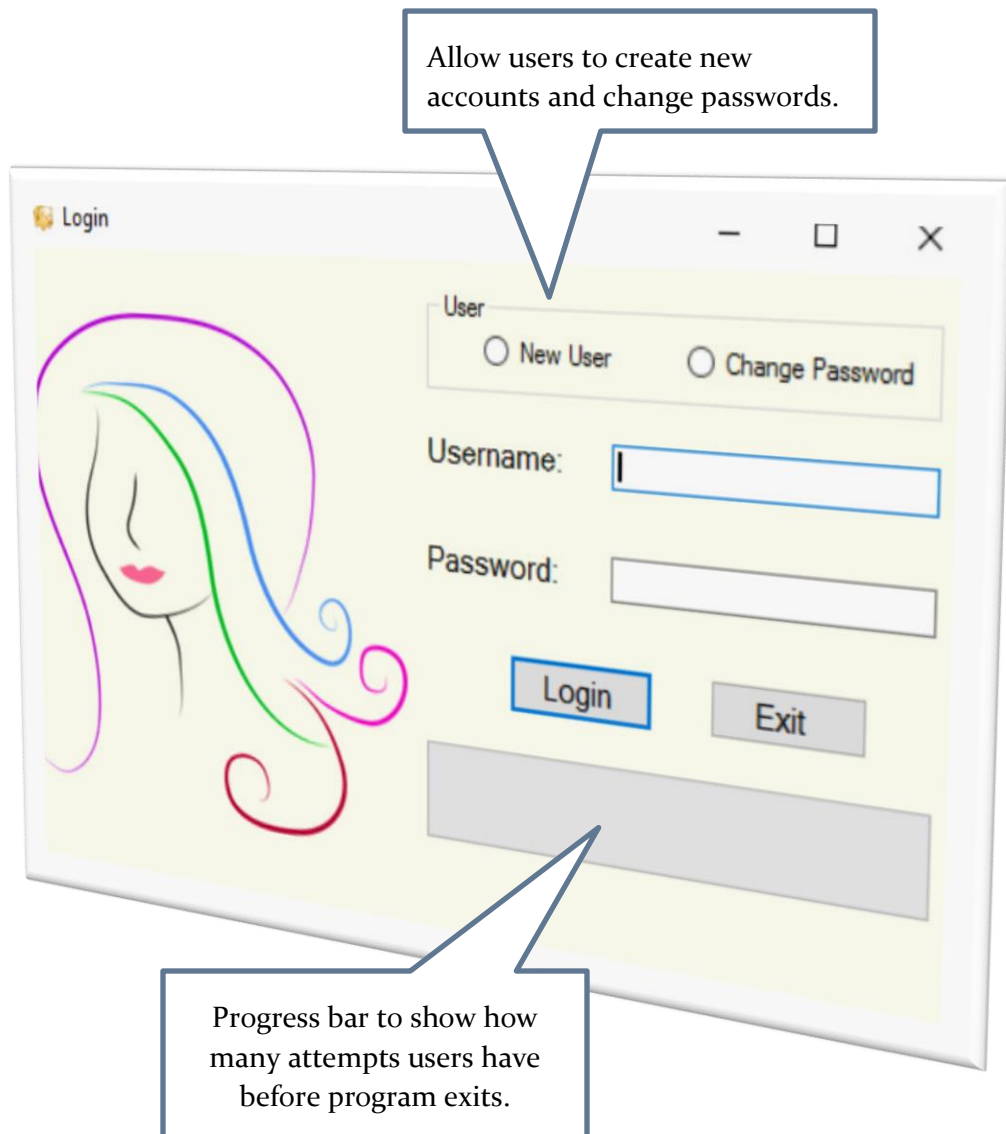
The Splash Screen

The first thing you will notice when you open this program is the splash screen. This screen gives you value detail about the program and your business. It is design to popup and orient in the middle of your screen for a few second while your program is loading.

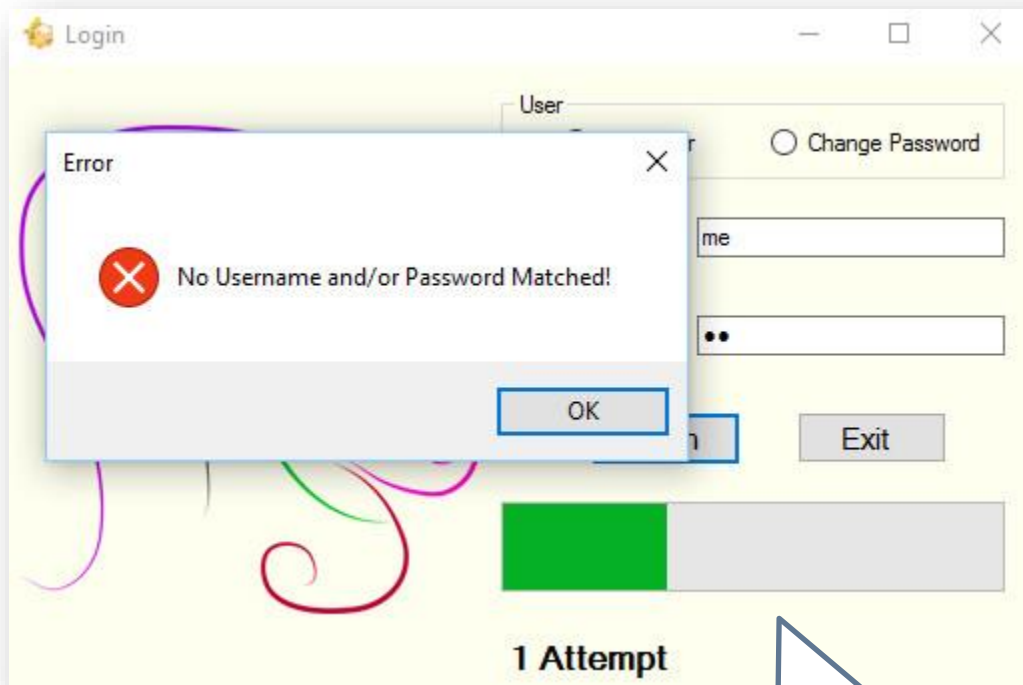Current version is 1.0 as of this guide.

Login Screen

The login screen (shown below) is the second screen that will be prompted when you run the program. This screen allows users to create a new account or change your password in case you forgot. It is important to note that the user will have 3 logins attempts to get the password right. If user fails to provide the correct password, the program will exit. There is a progress bar to show the user their remaining attempts.

Allow users to create new accounts and change passwords.

Progress bar to show how many attempts users have before program exits.

Login Screen Error

There will be an error message that appears every time you enter the wrong password or user name. The progress bar will display the missed attempts. If you fail all 3 attempts, the program will close. See example below.

The progress bar will display your attempts

The Main Entry

The main entry of the program is designed to be user friendly and easy to use. All of calculation and inputs are displayed on the main page of the program. User can select from description, number of styling, price, and the program will calculate the subtotal before taxes. Users can also add any discount they wish to apply by checking the discount box.

The menu is easy to understand

File   Edit   Format   Tools   Help

Main   Stuff   Web Browser

Sue's Salon and Spa

Invoice Information

Description

Number of Styling:

Price of Styling:

SubTotal:

These areas need to be inputted for calculation of total and tax

Tax:

Total Cost:

Purchase Date: 12/06/16

Accumlate Number of Styling:

Accumlate Grand Total:

☐ Check for Discount

Clear   Save   Exit

There is also a client information section for collecting client's information for future sales or promotional distribution. This section allows users to input client's name, address, and whether they are new or existing clients.

Client Information

Full Name: _____ ○ Existing Client

Address: _____ ○ New Client

City: _____

Zip Code: _____ **Sort**

State: [AK ▾] **Print**

This area is where your data table will display when you enter your price, quantity, and description.

Display

Once users have inputted all the required information, the program will calculate and generate the cost of each styling in the invoice box. If you click on the display button, the program will generate a table. Below is an example of how it should look.

Report #1

This is the main report that generates. All that you see here is how it should look like when you input the information according to each box. The only thing you can't input is the tax. The tax box will generate by itself in accordance to the tax laws of each state.

Report #2

This is another example of a report that will generate when you click on the display button in the stuff tab. This data shows how many cars are available and what their horsepower will be, so that it can give you a better understanding of what

| File | Edit | Format | Tools | Help |

| Main | Stuff | Web Browser |

Make	Model	Horse Power
Honda	Prelude	210
Toyota	Supra	250
Ford	Cobra	320
Acura	NSX	350
Dodge	Charger	410

Display
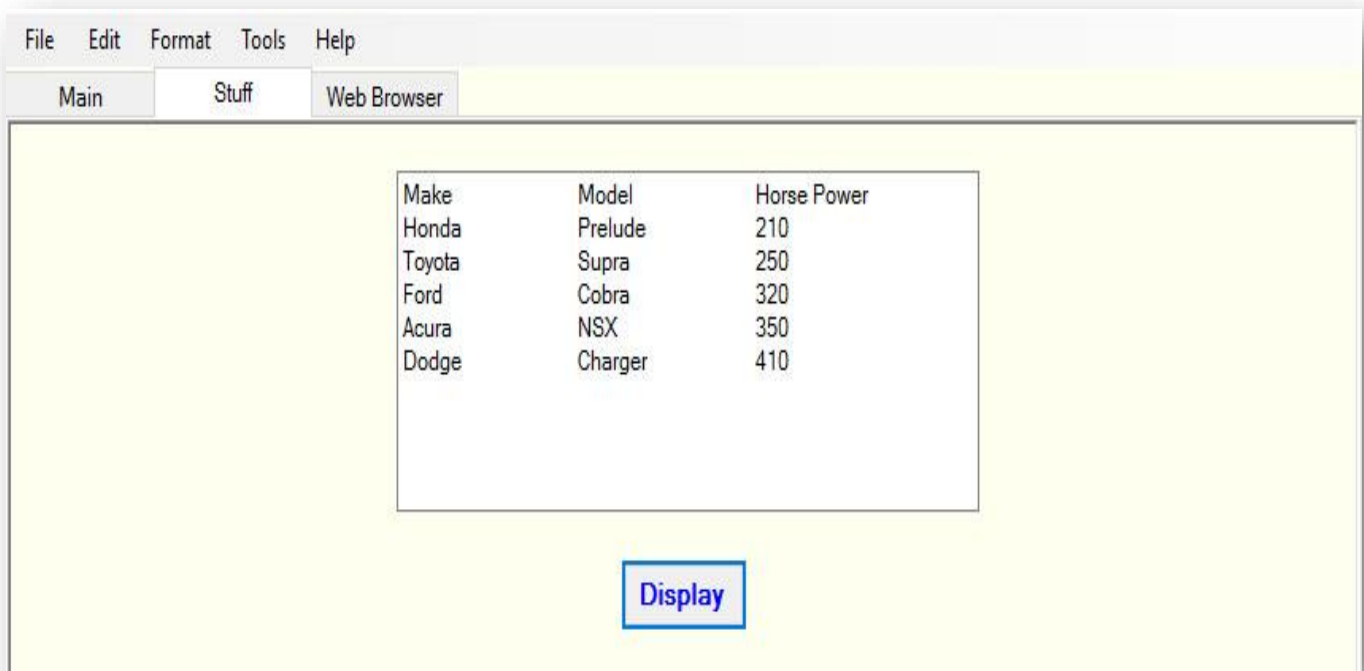
type of car you want are looking for.

Web Browser

This invoice program comes with a built-in web browser just in case you need to research a product or service. There is no need to open a third-party application. The web browser is clean and user friendly as well. Just go to the Web Browser tab to begin your search. The default search engine will be Google.
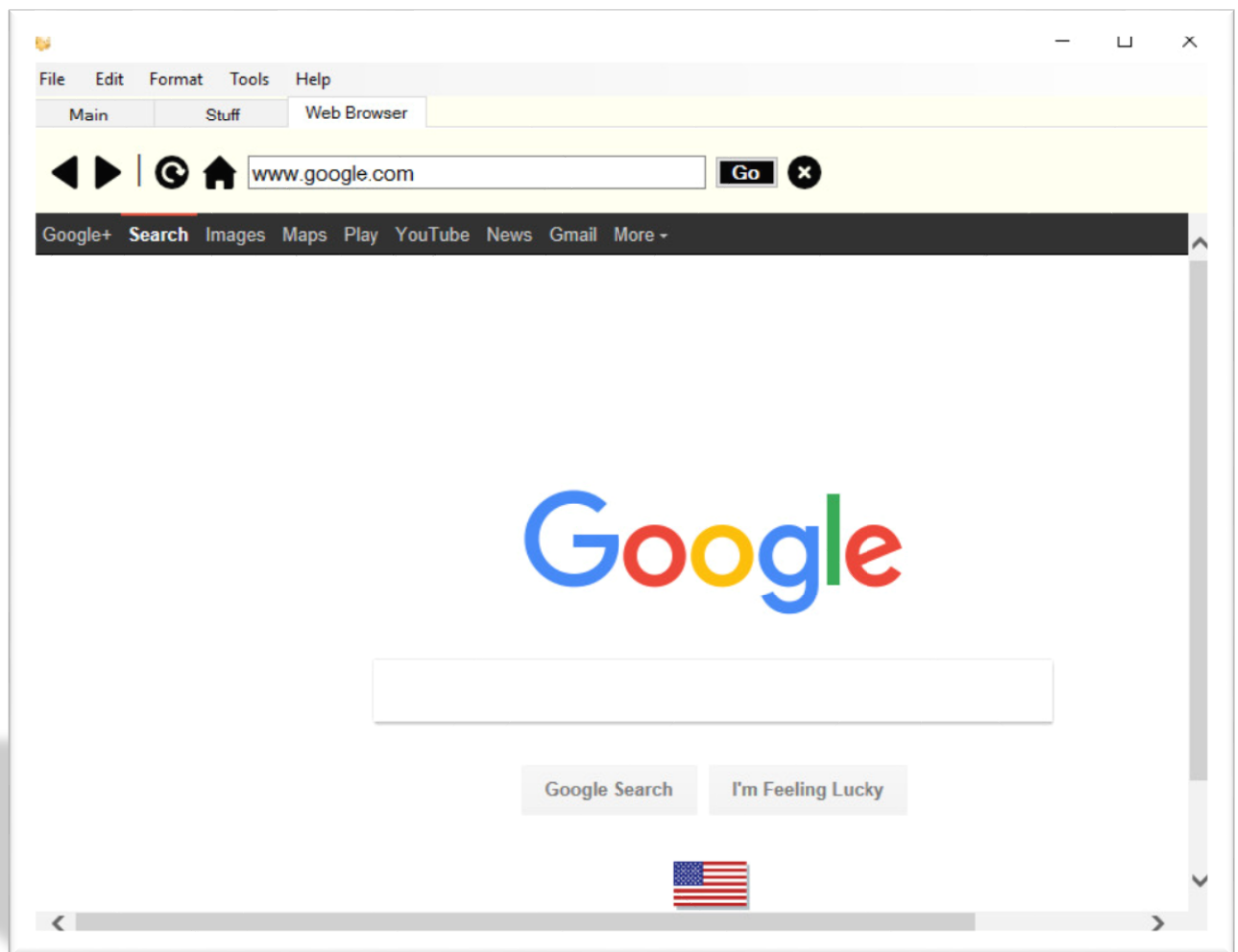
Word Processor

Along with the web browser, this program also comes with a word processor. The word processor allows you to jot down notes, write up reports, etc.

To access the word processor, click on Tools > Memo to open the window.
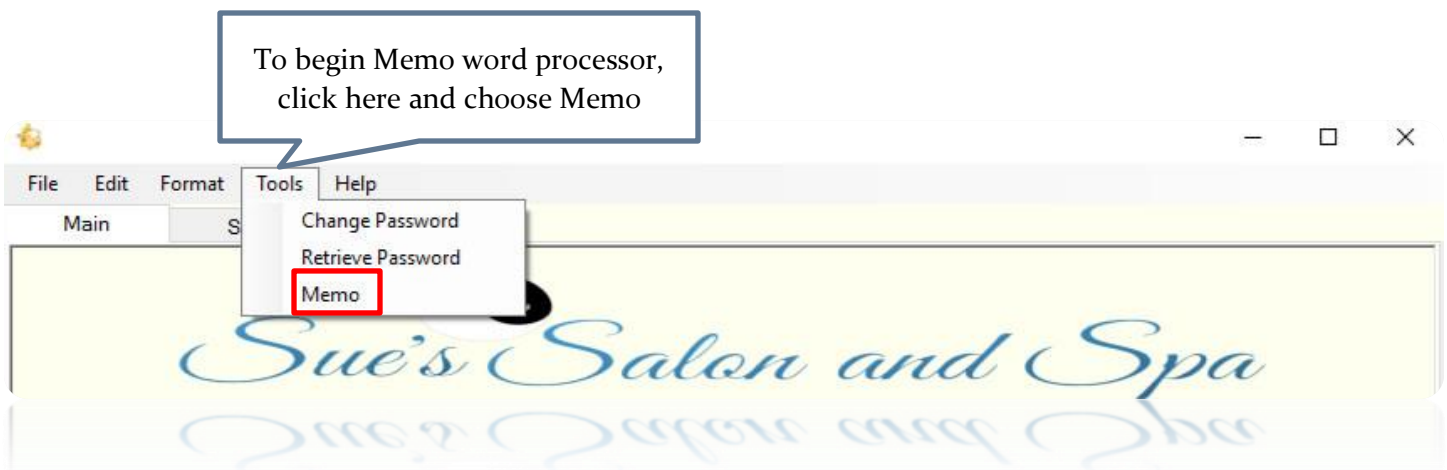
Once memo is selected, the word processor will open in a separate window. It's a simple little tool that you will find easy to use.
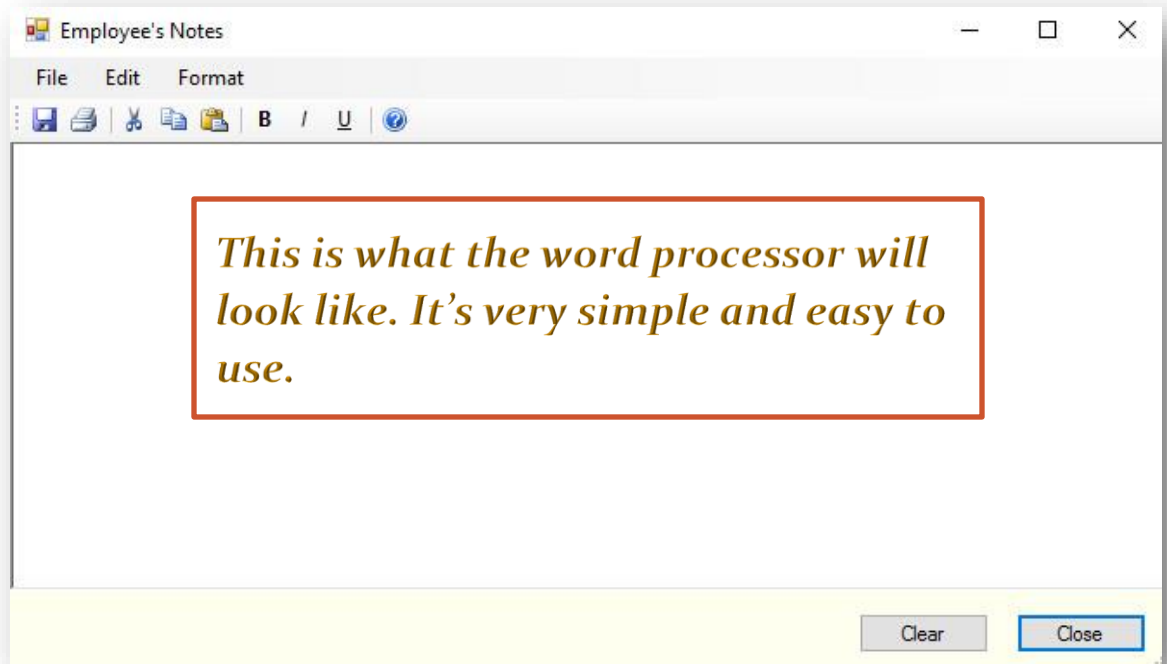
Should you ever need any help, there is also a help section that you can refer to by clicking the help button.

Menu Options

The menu bar is a great tool that allows you to choose from a few options that will enhance your document. We will go through the options below.

Color Dialog Box

The great thing about this program is that you can customize it to how ever you feel comfortable with. With the color dialog box, you can change the color that works for you. There is an assortment of different color to choose from. Just go to the main menu, select format, and select the color option.
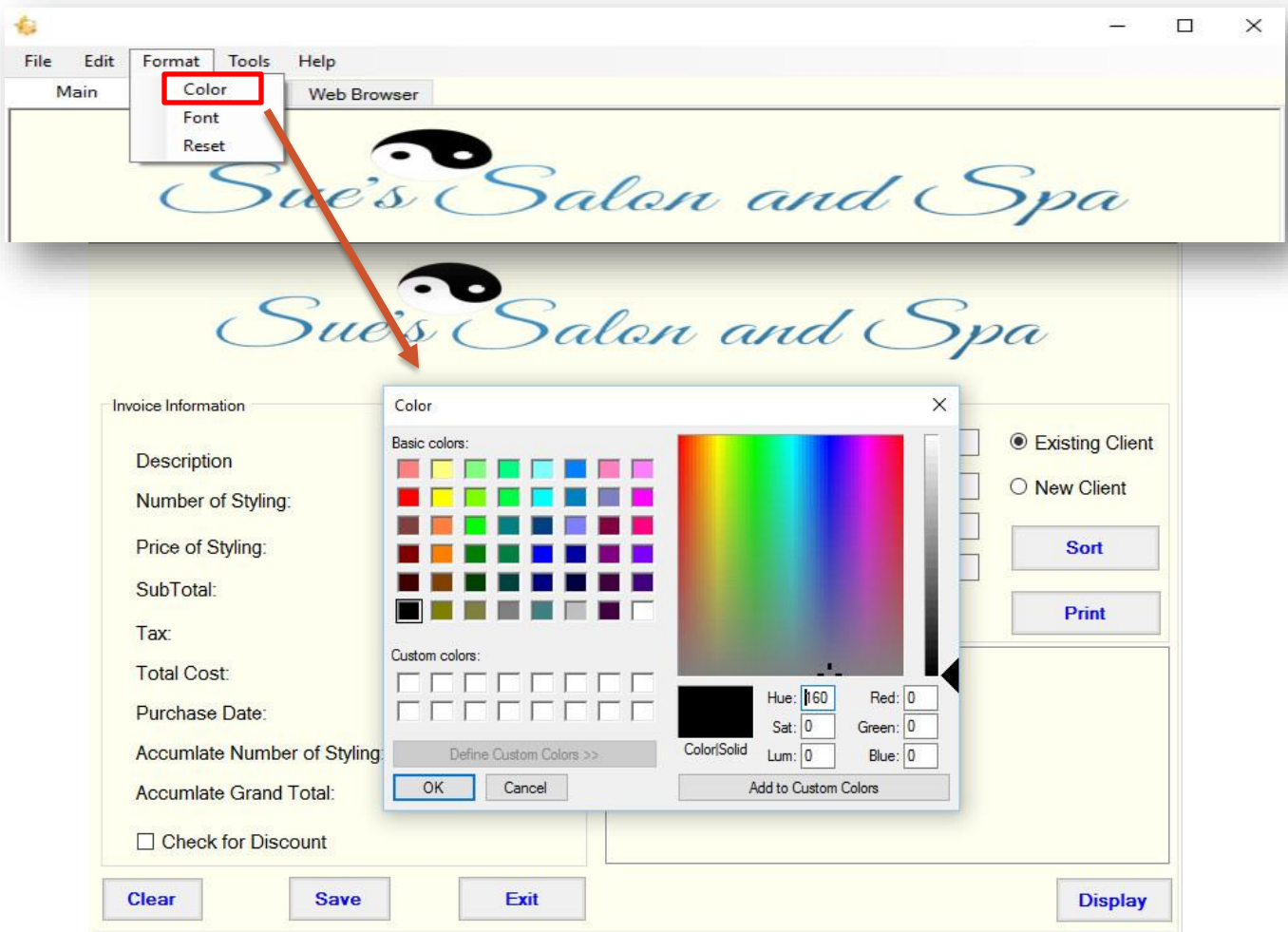
Font Dialog Box

Just like the color scheme of your data, you can also change
the font as well. There are a number of font options available.
All you have to do is go to the main program menu, select
format tab, and there is an option for font.
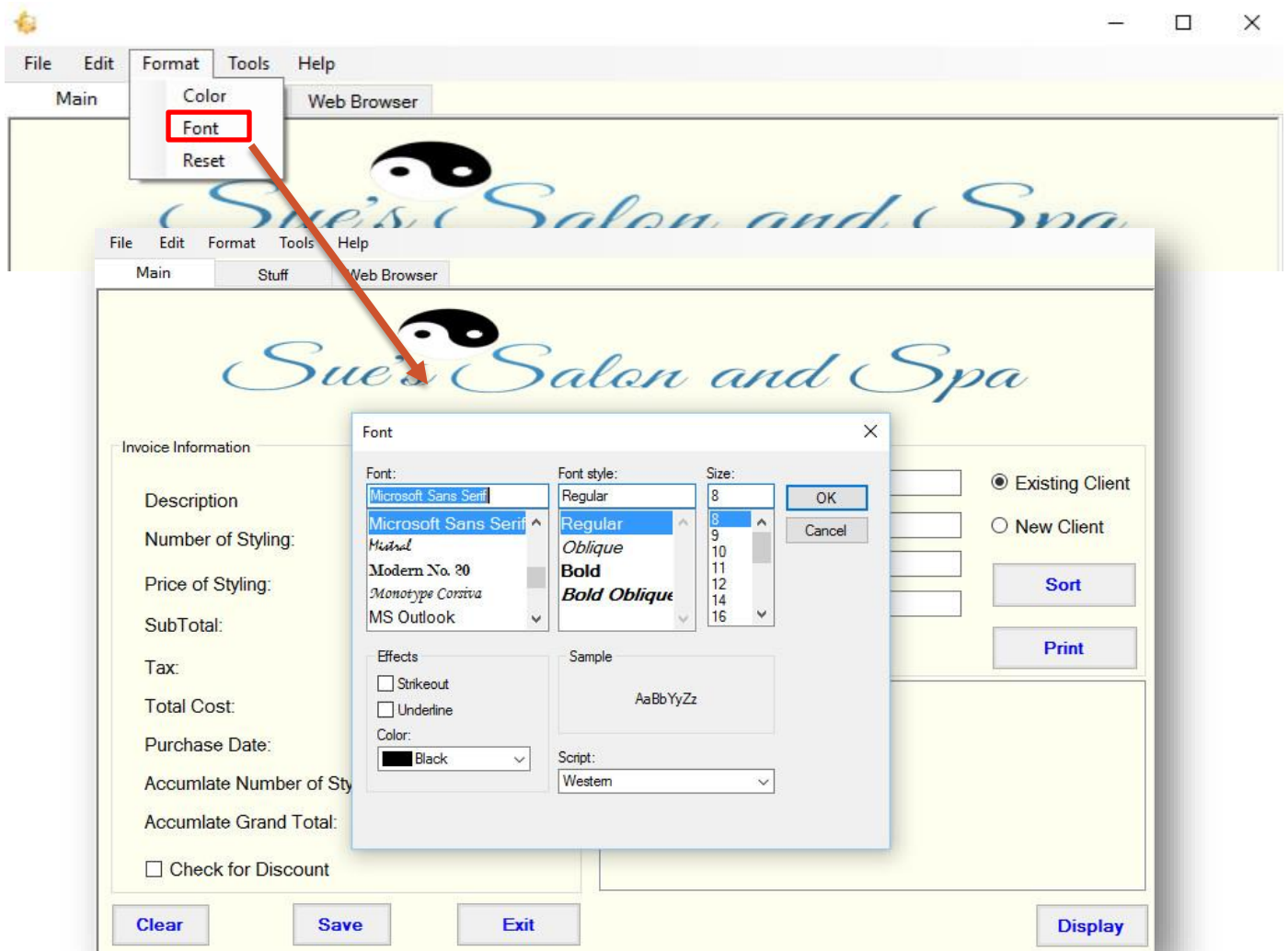
About Box

Along with getting help, users can also view the about screen to get information on the program and see if they have the latest and greatest updates available. It is important to maintain up to date software so that key issues and bugs can be fixed.

Error

This program allows you to make simple calculations. But the information that you enter must match the criteria or else it will not accept your input. The most common error is entering a word rather than a number in the numbers field or wrong punctuation. Below are some examples of the type of errors you may encounter.

This is a number's only error

Invoice Information

Description	Hair Cuts
Number of Styling:	Money
Price of Styling:	
SubTotal:	
Tax:	
Total Cost:	
Purchase Date:	12/11/16

Accumlate Number of Styling:

Accumlate Grand Total:

☐ Check for Discount

Clear	Save	Exit

Ready

Client Information

Full Name:	
Address:	
City:	
Zip Code:	
State:	AK

○ Existing Client
○ New Client

Sort

Print

Warning!   ✕

Enter only numbers for Hours

OK

Display

You will be prompt this message and can't do anything else until you correct the issue.

User Backup

When you are done with your calculations, saving and backing up the data is pretty simple. On the menu strip, there are set of tabs. The file tab give you the option to save your work.

Click here to save

This Save As window will appear.
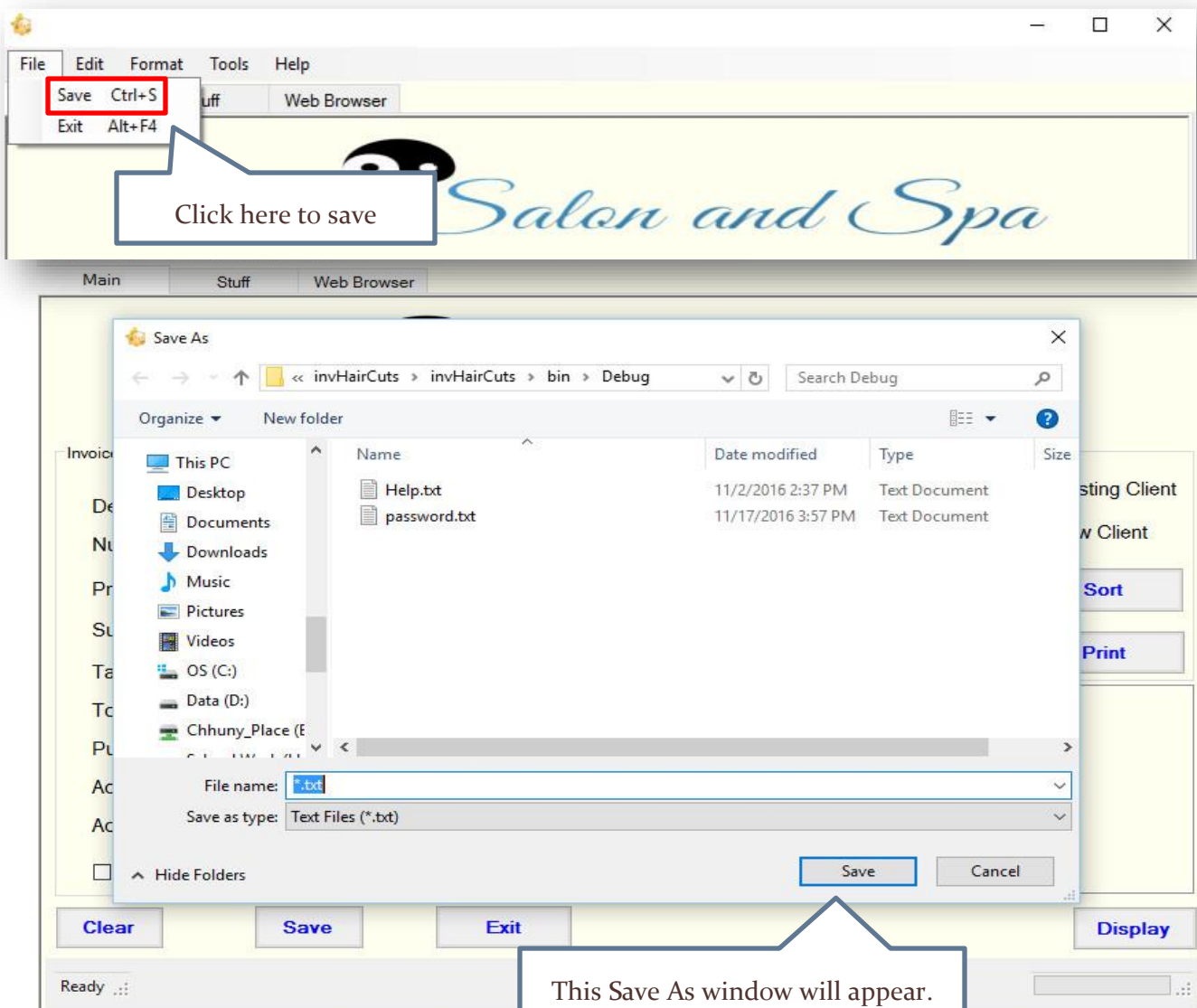
<u>Help</u>

Should you run into any trouble, the program offers 2 ways to
get the help you need. In the menu tab under "Help", there
are 2 options that allows you to open the help documents that
came with the program (figure below), and a web link to the
contact page of our website.

Weblink will take you to our website contact form
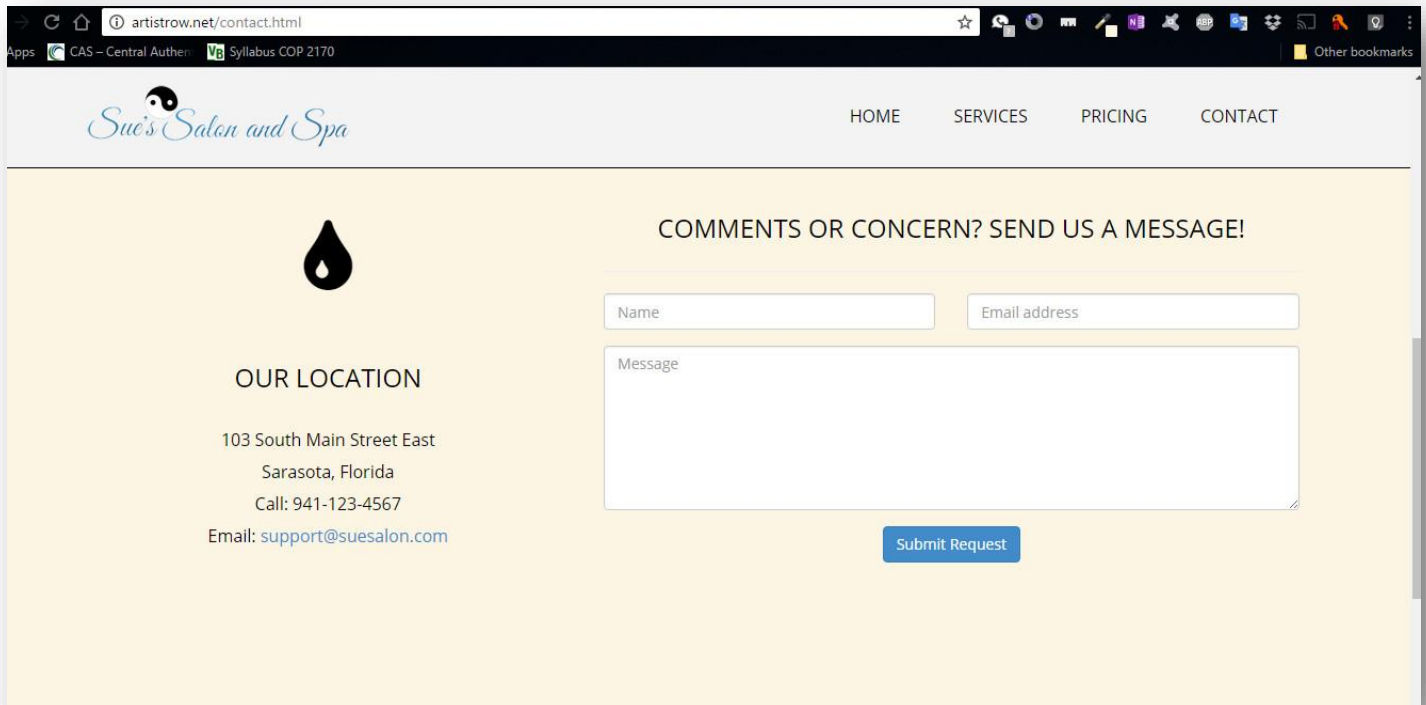
The "View Help" option will allow you to view documents that will help you trouble shoot key issues.
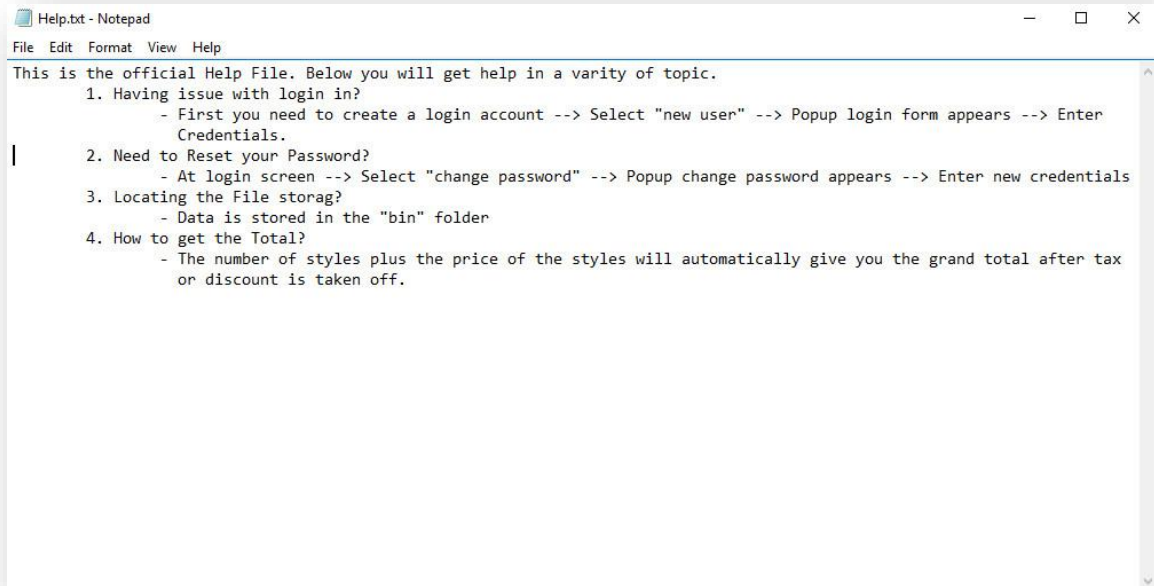
Program's Codes

```vbnet
Imports System.IO

Public Class frmHairCuts
    ' Gobal Invoice Information Area Varibles
    Dim decRate As Decimal
    Dim decHours As Decimal
    Dim decSubTotal As Decimal
    Dim decDiscountRate As Decimal
    Dim decDiscount As Decimal = 0.05D
    Dim decTotal As Decimal
    Dim decPrice As Decimal
    Dim intNumberOfStyling As Integer
    Dim GrandTotal As Decimal
    Dim Counter As Integer = 0
    Dim decCalcGrandTotal As Decimal
    Dim intCalcStyling As Integer
    Dim Response As DialogResult
    Dim DiscountRate As Decimal
    Dim dblSubTotal As Double
    Dim dblDiscountRate As Double

    ' Global Loan Calculator Varibles
    Dim dblMonthsYear As Double = 12
    Dim dblNewRate As Double
    Dim dblUsesRate As Double


    Dim dblVehicleCost As Double = 0          ' Vehicle cost
    Dim dblDownPayment As Double = 0          ' Down payment
    Dim intYears As Integer = 2               ' Number of months for the loan
    Dim intMonths As Integer = 0
    Dim decLoanAmount As Decimal = 0              ' Amount of the loan
    Dim decMonthlyPayment As Decimal = 0      ' Monthly payment
    Dim dblInterest As Double = 0             ' Interest paid for the period
    Dim dblPrincipal As Double           ' Principal paid for the period
    Dim intCount As Integer              ' Counter for the loop
    Dim strOut As String                 ' Used to hold a line of output
    Dim blnInputOk As Boolean = True
    Dim dblMonthlyInterest As Double = 0
    Dim decBalance As Decimal

    ' Global Variable to Hold the Annual Interest Rate
    Dim dblAnnualRate As Double = dblNewRate

    ' Global Definition for Accumulator Varibles
    Private CommissionTotalDecimal As Decimal

    ' Gobal Format Varibles
    Dim colFormItems As New Collection() ' Key word New is very important
    Dim ctrlFormItems As Control ' Used to iterate the FormItems collection
```

```vbnet
    ' StreamWriter Varibles
    Dim SW As StreamWriter

    ' Parallel Array Variables

    Dim Description(50) As String
    Dim Cost(50) As Decimal
    Dim Quantity(50) As Integer
    Dim Total(50) As Decimal
    Dim count As Integer
    Dim arrDescriptionSort() As String = {"Hair Cut", "Coloring", "Massage",
"Consult"}

    ' Two Dimensional Array

    Dim Car(20, 3) As String



Private Sub frmHairCuts_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        colFormItems.Add(lblTotalCost) ' FormItems could be labels, textboxes,
buttons...
        datDate.Format = DateTimePickerFormat.Custom
        datDate.CustomFormat = "MM/dd/yy"

        Car(1, 1) = "Honda"
        Car(1, 2) = "Prelude"
        Car(1, 3) = "210"
        Car(2, 1) = "Toyota"
        Car(2, 2) = "Supra"
        Car(2, 3) = "250"
        Car(3, 1) = "Ford"
        Car(3, 2) = "Cobra"
        Car(3, 3) = "320"
        Car(4, 1) = "Acura"
        Car(4, 2) = "NSX"
        Car(4, 3) = "350"
        Car(5, 1) = "Dodge"
        Car(5, 2) = "Charger"
        Car(5, 3) = "410"

        WebBrowser1.Navigate(TextBox1.Text)
    End Sub


    Private Sub Calculations()
        If txtPrice.Text = "" Or txtNumberofStyling.Text = "" Then Exit Sub

        Try
            decPrice = CDec(txtPrice.Text)
            intNumberOfStyling = CInt(txtNumberofStyling.Text)
            dblSubTotal = decPrice * intNumberOfStyling
            txtSubTotal.Text = dblSubTotal.ToString("c")
            dblDiscountRate = dblSubTotal * decDiscount
            txtDiscount.Text = dblDiscountRate.ToString("c")
            decTotal = CDec(dblDiscountRate + dblSubTotal)
            txtTotalCost.Text = decTotal.ToString("c")
```

```vbnet
                ' Counter
                Counter += 1
                lblCalcStyling.Text = CStr(Counter)

                ' Accumulator
                GrandTotal = CDec(GrandTotal + decTotal)
                lblCalcGrandTotal.Text = FormatCurrency(GrandTotal)

        Catch ex As OverflowException
            MessageBox.Show("Something seems off. Please go back and correct.",
"Warning")
        End Try
        Debug.WriteLine("txtDiscount=" & txtDiscount.Text & "txtTotalCost=" &
txtTotalCost.Text)

    End Sub



#Region "Buttons"
    Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles
btnClear.Click
        ' To Do: Add a clear button to clear all inputs
        txtPrice.Text = ""
        txtSubTotal.Text = ""
        txtDiscount.Text = ""
        txtTotalCost.Text = ""
        txtNumberofStyling.Text = ""
        txtName.Text = ""
        txtCity.Text = ""
        txtZipCode.Text = ""
        cmbDescription.Text = ""

    End Sub

    Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles
btnExit.Click
        Dim result As MsgBoxResult
        result = CType(MessageBox.Show("Are you sure you're ready to exit this
program", "Confirmation", MessageBoxButtons.YesNo), MsgBoxResult)
        If result = MsgBoxResult.Yes Then
            Me.Close()
        ElseIf CBool(MsgBoxResult.No) Then
            Return
        End If
    End Sub

    Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles
btnSave.Click
        If cmbDescription.Text = Description(count) Then
            MessageBox.Show("Record Already Saved", "Probably Duplicate")
            Exit Sub
        End If
        count += 1
        Description(count) = cmbDescription.Text
        Cost(count) = CDec(txtPrice.Text)
        Quantity(count) = CInt(txtNumberofStyling.Text)
        Total(count) = CDec(txtSubTotal.Text)
```

```vb
            Debug.WriteLine(count & " " & Description(count) & " " & Cost(count) & " "
& Quantity(count) & " " & Total(count))
    End Sub

    Private Sub btnDisplay_Click(sender As Object, e As EventArgs) Handles
btnDisplay.Click
        Dim x As Integer

        ListBoxDisplay.Items.Clear()
        GrandTotal = 0
        ListBoxDisplay.Items.Add("Description" & vbTab & "Cost    " & vbTab & vbTab
& "Quantity" & vbTab & vbTab & "Total")
        For x = 1 To count
            ListBoxDisplay.Items.Add(Description(x) & vbTab & vbTab & "$" &
Cost(x) & vbTab & vbTab & Quantity(x) & vbTab & vbTab & "$" & Total(x))
            GrandTotal += Total(x)
        Next
        ListBoxDisplay.Items.Add(vbCrLf)
        ListBoxDisplay.Items.Add(vbTab & vbTab & vbTab & vbTab & "GrandTotal:" &
vbTab & "$" & GrandTotal)
        ToolStripStatusLabel1.Text = "Please Wait"
        Timer1.Enabled = True
    End Sub

#End Region


#Region "Inputs"
    Private Sub txtNumberofStyling_LostFocus(sender As Object, e As EventArgs)
Handles txtNumberofStyling.LostFocus
        If txtNumberofStyling.Text = "" Then Exit Sub
        If Not IsNumeric(txtNumberofStyling.Text) Then
            MessageBox.Show("Enter only numbers for Hours", "Warning!")
            txtNumberofStyling.Focus()
            Exit Sub
        Else
            Call Calcs()
        End If

        If Integer.TryParse(txtNumberofStyling.Text, intNumberOfStyling) Then
        Else
            MessageBox.Show("Error: This is not a valid Quantity", "Not Valid
Quantity", MessageBoxButtons.OK, MessageBoxIcon.Error)
            txtNumberofStyling.Focus()
        End If

        If CInt(txtNumberofStyling.Text) > 100 Then
            Beep()
            Response = MessageBox.Show("Quantity limits only 100 per stylist per
month.", "Warning!", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
            If Response = DialogResult.Yes Then
                txtNumberofStyling.Focus()
                Exit Sub
            End If
        End If
    End Sub
```

```vb
    Private Sub txtPrice_LostFocus(sender As Object, e As EventArgs) Handles
txtPrice.LostFocus
        If txtPrice.Text = "" Then Exit Sub
        txtPrice.Text = FormatCurrency(txtPrice.Text)
        If Not IsNumeric(txtPrice.Text) Then
            MessageBox.Show("Enter only numbers for Price", "Message")
            txtPrice.Focus()
            Exit Sub
        Else
            Call Calcs()
        End If

        If CDbl(txtPrice.Text) > 1000 Then
            Beep()
            Response = MessageBox.Show("Are you sure this is the right price?",
"Confirmation", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation)
            If Response = vbYes Then
                txtPrice.Focus()
                Exit Sub
            Else
                Call Calcs()
            End If
        End If
    End Sub
    ' Calculate Discount into SubTotal
    Private Sub chbxDiscount_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles chbxDiscount.CheckedChanged
        If chbxDiscount.Checked = True Then        ' Discount Box is checked
            DiscountRate = 0.05D
            txtDiscount.Text = CStr(FormatPercent(DiscountRate))
            txtTotalCost.Text = FormatCurrency(decPrice * intNumberOfStyling)
        Else
            Call Calcs()
        End If
    End Sub
    Private Sub txtName_LostFocus(sender As Object, e As EventArgs) Handles
txtName.LostFocus
        If txtName.Text = "" Then Exit Sub
        txtName.Text = txtName.Text.Trim.ToUpper
    End Sub
    Private Sub txtAddress_LostFocus(sender As Object, e As EventArgs) Handles
txtAddress.LostFocus
        If txtAddress.Text = "" Then Exit Sub

    End Sub

    Private Sub txtCity_LostFocus(sender As Object, e As EventArgs) Handles
txtCity.LostFocus
        If txtCity.Text = "" Then Exit Sub

    End Sub
```

```vbnet
    Private Sub txtZipCode_LostFocus(sender As Object, e As EventArgs) Handles
txtZipCode.LostFocus
        If txtZipCode.Text = "" Then Exit Sub
    End Sub

#End Region

#Region "Pictures"

    Private Sub PicLogo_Click(sender As Object, e As EventArgs) Handles
PicLogo.Click
        lblMessage2.Visible = True
        lblMessage2.Text = "Welcome to Sue's Salon and Spa"
    End Sub


#End Region

#Region "Misc"
    Private Sub btnCounter_Click(sender As Object, e As EventArgs)
        Static intCounter As Integer

        intCounter += 1
        'lblCounter.Text = intCounter.ToString("n0")
    End Sub

    Private Sub btnCalcCommission_Click(sender As Object, e As EventArgs)
        Dim decCommission, decSales As Decimal
        Static intCommissionCount As Integer

        ' count each commission paid by adding 1 to the counter
        intCommissionCount += 1

        If CDec(decSales) >= 5000 Then
            decCommission = CDec(decSales * 0.1)
        Else
            decCommission = CDec(decSales * 0.05)
        End If

        ' add the commission to the running total of commissions
        CommissionTotalDecimal = CommissionTotalDecimal + decCommission


    End Sub

    Private Sub btnCountDown_Click(sender As Object, e As EventArgs)
        Timer.Enabled = True
    End Sub

    Private Sub Timer_Tick(sender As Object, e As EventArgs) Handles Timer.Tick
        Static Race As Integer = 5

        Race -= 1
        If Race = 4 Then
        End If
        If Race <= -1 Then
            Timer.Interval = 100
        End If
```

```vb
        End Sub
#End Region



#Region "Manu Tab"
    Private Sub mnuFileExit_Click(sender As Object, e As EventArgs) Handles
mnuFileExit.Click
        Me.Close()
    End Sub

    Private Sub mnuFormatColor_Click(sender As Object, e As EventArgs) Handles
mnuFormatColor.Click
        ColorDialog1.ShowDialog()
        For Each ctrlFormItems In colFormItems
            ctrlFormItems.ForeColor = ColorDialog1.Color
        Next
    End Sub

    Private Sub mnuFormatFont_Click(sender As Object, e As EventArgs) Handles
mnuFormatFont.Click
        FontDialog1.ShowColor = True


        FontDialog1.ShowDialog()
        For Each ctrlFormItems In colFormItems
            ctrlFormItems.Font = FontDialog1.Font
        Next
    End Sub

    Private Sub mnuFormatReset_Click(sender As Object, e As EventArgs) Handles
mnuFormatReset.Click
        Dim myFont As New Font("Time New Roman", 10, FontStyle.Bold)
        lblTotalCost.Font = myFont
        ' or ctrlFormLabel.Font = myFont (for Collections)
        lblTotalCost.ForeColor = Color.Black
    End Sub

    Private Sub mnuHelpViewHelp_Click(sender As Object, e As EventArgs) Handles
mnuHelpViewHelp.Click
        Process.Start("NotePad", "Help.txt")
    End Sub
    Private Sub mnuHelpWebLink_Click(sender As Object, e As EventArgs) Handles
mnuHelpWebLink.Click
        Process.Start("http://scf.edu")

    End Sub
    Private Sub mnuHelpAbout2_Click(sender As Object, e As EventArgs) Handles
mnuHelpAbout2.Click
        Dim frmAbout As New frmAboutBox()
        frmAboutBox.ShowDialog()
    End Sub

    Private Sub ctxMenuColor_Click(sender As Object, e As EventArgs) Handles
ctxMenuColor.Click
        mnuFormatColor.PerformClick()
    End Sub
```

```vb
    Private Sub ctxMenuFont_Click(sender As Object, e As EventArgs) Handles
ctxMenuFont.Click
        mnuFormatFont.PerformClick()
    End Sub

    Private Sub ViewHelpToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ViewHelpToolStripMenuItem.Click
        mnuHelpViewHelp.PerformClick()
    End Sub

    Private Sub AboutToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles AboutToolStripMenuItem.Click
        mnuHelpAbout2.PerformClick()
    End Sub

    Private Sub cmdSave_Click(sender As Object, e As EventArgs)
        sfdWriter.ShowDialog()

    End Sub

    Private Sub sfdWriter_FileOk(sender As Object, e As
System.ComponentModel.CancelEventArgs) Handles sfdWriter.FileOk
        If sfdWriter.FileName <> "" Then
            SW = New StreamWriter(sfdWriter.FileName, False)
            SW.Close()
        End If
    End Sub

    Private Sub mnuToolsPassword_Click(sender As Object, e As EventArgs) Handles
mnuToolsPassword.Click
        frmChangePass.ShowDialog()
        frmLogin.Close()

    End Sub

    Private Sub RetrievePasswordToolStripMenuItem_Click(sender As Object, e As
EventArgs) Handles RetrievePasswordToolStripMenuItem.Click
        ofdReader.ShowDialog()
    End Sub

    Private Sub mnuFileSave_Click(sender As Object, e As EventArgs) Handles
mnuFileSave.Click
        sfdWriter.ShowDialog()
    End Sub

    Private Sub mnuFileOpen_Click(sender As Object, e As EventArgs)
        ofdReader.ShowDialog()

    End Sub

    Private Sub ofdReader_FileOk(sender As Object, e As
System.ComponentModel.CancelEventArgs) Handles ofdReader.FileOk
        If ofdReader.FileName <> "" Then
            Dim SR As New StreamReader(ofdReader.FileName)

            Do Until SR.EndOfStream
            Loop
```

```vbnet
            SR.Close()
        End If
    End Sub

    Private Sub mnuToolsMemo_Click(sender As Object, e As EventArgs) Handles
mnuToolsMemo.Click
        frmNotes.ShowDialog()

    End Sub



#Region "Stuff Tab"
    Private Sub btnSort_Click(sender As Object, e As EventArgs) Handles
btnSort.Click
        Array.Sort(arrDescriptionSort)
    End Sub

    Private Sub btnPrintArray_Click(sender As Object, e As EventArgs) Handles
btnPrintArray.Click
        Dim count As Integer = 1
        ListBoxDisplay.Items.Clear()
        While count < arrDescriptionSort.Length
            ListBoxDisplay.Items.Add(count & vbTab & arrDescriptionSort(count))
            count += 1
        End While
    End Sub

    Private Sub btnDisplay2_Click(sender As Object, e As EventArgs) Handles
btnDisplay2.Click
        ListBoxDisplayArray.Items.Clear()
        ListBoxDisplayArray.Items.Add("Make" & vbTab & vbTab & "Model" & vbTab &
vbTab & "Horse Power")
        For count = 1 To 5
            ListBoxDisplayArray.Items.Add(Car(count, 1) & vbTab & vbTab &
Car(count, 2) & vbTab & vbTab & Car(count, 3))
        Next count
    End Sub


    Private Sub picHome_Click(sender As Object, e As EventArgs) Handles
picHome.Click
        WebBrowser1.GoHome()
    End Sub

    Private Sub picRefresh_Click(sender As Object, e As EventArgs) Handles
picRefresh.Click
        WebBrowser1.Refresh()
    End Sub

    Private Sub picForward_Click(sender As Object, e As EventArgs) Handles
picForward.Click
        WebBrowser1.GoForward()
    End Sub

    Private Sub picBack_Click(sender As Object, e As EventArgs) Handles
picBack.Click
        WebBrowser1.GoBack()
```

```vbnet
    End Sub

    Private Sub btnGo_Click(sender As Object, e As EventArgs) Handles btnGo.Click
        WebBrowser1.Navigate(TextBox1.Text)
    End Sub

    Private Sub picStop_Click(sender As Object, e As EventArgs) Handles
picStop.Click
        WebBrowser1.Stop()
    End Sub

    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        ToolStripProgressBar2.Value = ToolStripProgressBar2.Value + 2
        If ToolStripProgressBar2.Value = 100 Then
            Timer1.Enabled = False
        End If
    End Sub

    Private Sub TextBox1_KeyDown(sender As Object, e As KeyEventArgs) Handles
TextBox1.KeyDown
        If e.KeyCode = Keys.Enter Then
            WebBrowser1.Navigate(TextBox1.Text)
        End If
    End Sub

#End Region
```