Imports System.Data.OleDb

Module connect

Public sConString As String

Function Get\_Constring()

If Microsoft.VisualBasic.Right(Application.StartupPath, 1) = "\" Then

sConString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & Application.StartupPath & "SaadiT.accdb;Persist Security Info=False;"

Else

sConString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" & Application.StartupPath & "\SaadiT.accdb;Persist Security Info=False;"

End If

Return sConString

End Function

Function Check\_ToolsAvailability(ByVal itemNumber As String, ByVal qtty As String)

Dim available As Boolean

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim sql As String

Dim currentStocks As Integer

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

sql = "select tstockLevel FROM tblToolslist WHERE tPartNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "tstockLevel")

currentStocks = CInt(ds.Tables(0).Rows(0).Item("tstockLevel"))

If currentStocks < qtty Then

available = False

Else

available = True

End If

cn.Close()

Return available

End Function

Function check\_ItemAvailability(ByVal itemNumber As String, ByVal qtty As String)

Dim available As Boolean

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim sql As String

Dim currentStocks As Integer

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

sql = "select stockLevel FROM tblItemList WHERE ItemNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "stockLevel")

currentStocks = ds.Tables(0).Rows(0).Item("stockLevel")

If currentStocks < qtty Then

available = False

Else

available = True

End If

cn.Close()

Return available

End Function

End Module

Imports System.Data.OleDb

Imports System.IO

Public Class frmAddItem

Private Sub Clear\_Fields()

Me.txtItemNum.Text = ""

Me.txtItemName.Text = ""

Me.txtDescript.Text = ""

End Sub

Private Sub btnClear\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click

Clear\_Fields()

End Sub

Private Sub btnSave\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click

If (Me.txtItemNum.Text) <> "" Then

If Len(Trim(Me.txtItemNum.Text)) > 0 And IsNumeric(Me.txtItemNum.Text) Then

If (Me.txtItemName.Text) <> "" Then

Save\_Item()

Else

MsgBox("Item Name should not be empty", MsgBoxStyle.OkOnly, "Field Error")

End If

Else

MsgBox("Item Number should be Numeric only", MsgBoxStyle.OkOnly, "Field Error")

Exit Sub

End If

Else

MsgBox("Item Number should be filled up", MsgBoxStyle.OkOnly, "Field Error")

End If

End Sub

Private Sub Save\_Item()

Dim cn As New OleDbConnection

Dim cmd As New OleDbCommand

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "INSERT INTO tblItemlist (itemNum, itemName, iDescript) VALUES(@itemNum, @itemName, @iDescript)"

cmd.Parameters.Add("@itemNum", OleDb.OleDbType.Numeric).Value = IIf(Len(Trim(Me.txtItemNum.Text)) > 0, Me.txtItemNum.Text, "")

cmd.Parameters.Add("@itemName", OleDb.OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtItemName.Text)) > 0, Me.txtItemName.Text, "")

cmd.Parameters.Add("@iDescript", OleDb.OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtDescript.Text)) > 0, Me.txtDescript.Text, "")

If cmd.ExecuteNonQuery() Then

MsgBox("Item has been Save", 0, "Add Item Table")

Load\_Item()

Clear\_Fields()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub frmAddItem\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Load\_Item()

Load\_Tools()

End Sub

Private Sub Load\_Item()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = " SELECT (itemNum) as [Item #], (itemName) as [Iten Name], (iDescript) as [Description] FROM tblItemlist order by itemNum"

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.dataGVItem.DataSource = dt

Dim column As DataGridViewColumn = dataGVItem.Columns(2)

column.Width = 200

'Me.dataGVItem.Width = 250

'Me.dataGVItem.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.ColumnHeader)

If dt.Rows.Count = 0 Then

MsgBox("No record found!")

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Delete\_Item()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "DELETE FROM tblItemlist where itemNum = @itemNum"

cmd.Parameters.Add("@itemNum", OleDbType.Numeric).Value = Me.dataGVItem.Item(0, Me.dataGVItem.CurrentRow.Index).Value

If cmd.ExecuteNonQuery Then

MsgBox("Successfully Deleted", MsgBoxStyle.OkOnly, "Item Deleted")

Load\_Item()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

'Private Sub btnReload\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

' Load\_Item()

'End Sub

Private Sub tbnDeleteItem\_Click(sender As System.Object, e As System.EventArgs) Handles tbnDeleteItem.Click

Dim confirm As String = MsgBox("Are you sure you want to delete this Item?", MsgBoxStyle.YesNo, "Delete Item")

If confirm = DialogResult.Yes Then

Delete\_Item()

Else

MsgBox("Delete event was canceled", MsgBoxStyle.Critical, "Failed to Delete")

End If

End Sub

Private Sub Delete\_Tools()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "DELETE FROM tblToolslist where tPartNum = @tPartNum"

cmd.Parameters.Add("@tPartNum", OleDbType.Numeric).Value = Me.dataGVTools.Item(0, Me.dataGVTools.CurrentRow.Index).Value

If cmd.ExecuteNonQuery Then

MsgBox("Successfully Deleted", MsgBoxStyle.OkOnly, "Item Deleted")

Load\_Tools()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Save\_Tools()

Dim cn As New OleDbConnection

Dim cmd As New OleDbCommand

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "INSERT INTO tblToolslist (tPartNum, tPartName, tDescript) VALUES(@tPartNum, @tPartName, @tDescript)"

cmd.Parameters.Add("@tPartNum", OleDb.OleDbType.Numeric).Value = IIf(Len(Trim(Me.txtToolsNumber.Text)) > 0, Me.txtToolsNumber.Text, "")

cmd.Parameters.Add("@tPartName", OleDb.OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtToolsName.Text)) > 0, Me.txtToolsName.Text, "")

cmd.Parameters.Add("@tDescript", OleDb.OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtTDescript.Text)) > 0, Me.txtTDescript.Text, "")

If cmd.ExecuteNonQuery() Then

MsgBox("Tools has been Save", 0, "Add Tools")

Load\_Tools()

Clear\_FieldsTools()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub btnSaveTools\_Click(sender As System.Object, e As System.EventArgs) Handles btnSaveTools.Click

If (Me.txtToolsNumber.Text) <> "" Then

If Len(Trim(Me.txtToolsNumber.Text)) > 0 And IsNumeric(Me.txtToolsNumber.Text) Then

If (Me.txtToolsName.Text) <> "" Then

Save\_Tools()

Else

MsgBox("Tools Name should not be empty", MsgBoxStyle.OkOnly, "Field Error")

End If

Else

MsgBox("Part Number should be Numeric only", MsgBoxStyle.OkOnly, "Field Error")

Exit Sub

End If

Else

MsgBox("Part Number should be filled up", MsgBoxStyle.OkOnly, "Field Error")

End If

End Sub

Private Sub Load\_Tools()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = " SELECT (tPartNum) as [Part #], (tPartName) as [Name], (tDescript) as [Description] FROM tblToolslist order by tPartNum"

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.dataGVTools.DataSource = dt

Dim column As DataGridViewColumn = dataGVTools.Columns(2)

column.Width = 200

'Me.dataGVItem.Width = 250

'Me.dataGVItem.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.ColumnHeader)

If dt.Rows.Count = 0 Then

MsgBox("No record found!")

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Clear\_FieldsTools()

txtToolsNumber.Text = ""

txtToolsName.Text = ""

txtTDescript.Text = ""

End Sub

Private Sub btnDeleteTools\_Click(sender As System.Object, e As System.EventArgs) Handles btnDeleteTools.Click

Dim confirm As String = MsgBox("Are you sure you want to delete this Tools?", MsgBoxStyle.YesNo, "Delete Tools")

If confirm = DialogResult.Yes Then

Delete\_Tools()

Else

MsgBox("Delete event was canceled", MsgBoxStyle.Critical, "Failed to Delete")

End If

End Sub

End Class

Imports System.Data.OleDb

Imports System.IO

Public Class frmBorrow

Dim rDate As String = Format(Now, "dd/MM/yyy").ToString

Dim rTime As String = Now.ToLongTimeString.ToString

Dim requestCreated As String = rDate + " " + rTime

Dim borrowedItem(50, 11) As String

Dim x As Integer = 0

Dim num\_item As Integer

Private Sub frmBorrow\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Load\_ToolsItem()

Clear\_Toolsfields()

Generate\_BorrowCode()

End Sub

Private Sub btnAdd\_Click(sender As System.Object, e As System.EventArgs) Handles btnAdd.Click

Add\_BorrowTools()

Clear\_Toolsfields()

End Sub

Private Sub btnRemove\_Click(sender As System.Object, e As System.EventArgs) Handles btnRemove.Click

Dim confirm As String = MsgBox("Are you sure you want to delete this Item?", MsgBoxStyle.YesNo, "Delete Item")

If confirm = DialogResult.Yes Then

Remove\_BorrowTools()

Else

MsgBox("Delete event was canceled", MsgBoxStyle.Critical, "Canceled")

End If

End Sub

Private Sub btnSave\_Click(sender As System.Object, e As System.EventArgs) Handles btnSave.Click

Save\_BorrowTools()

Generate\_BorrowCode()

End Sub

Private Sub View\_BorrowItem()

Me.DataGridView1.Rows.Add(borrowCode.Text, cbToolsID.Text, txtToolsName.Text, txtToolsDescript.Text, txtQuantity.Text)

End Sub

Private Sub Add\_BorrowTools()

Dim status As Boolean

If Len(Trim(cbToolsID.Text)) > 0 Then

If IsNumeric(cbToolsID.Text) Then

borrowedItem(x, 0) = borrowCode.Text

borrowedItem(x, 1) = cbToolsID.Text

borrowedItem(x, 2) = txtToolsName.Text

borrowedItem(x, 3) = txtQuantity.Text

borrowedItem(x, 4) = txtToolsDescript.Text

borrowedItem(x, 5) = requestCreated

borrowedItem(x, 6) = txtEmpid.Text

borrowedItem(x, 7) = txtEmpName.Text

borrowedItem(x, 8) = txtreturnDate.Text

borrowedItem(x, 9) = logInBy

borrowedItem(x, 10) = DateAdd(DateInterval.Day, 2, Now)

status = Check\_ToolsAvailability(cbToolsID.Text, txtQuantity.Text)

If status = True Then

x = x + 1

num\_item = x

View\_BorrowItem()

Else

MsgBox("Insuficient Item # '" & txtToolsName.Text & "'")

Exit Sub

End If

End If

End If

End Sub

Private Sub Save\_BorrowTools()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim counter As Integer = 0

Dim strdb As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

While counter < x

cmd.Connection = cn

cmd.CommandType = CommandType.Text

strdb = "INSERT INTO tblBorrowTools ( tBorrowCode, tItemID, tName, tQuantity, tDescript, tDateCreated, tEmpNum, tEmpName, tReturnDate, tEncodedby, tDueDate, tBorrowNum) "

cmd.CommandText = strdb + " VALUES(@t1, @t2, @t3, @t4, @t5, @t6, @t7, @t8, @t9, @t10, @t11,@t12) "

cmd.Parameters.Add("@t1", OleDbType.VarChar).Value = borrowedItem(counter, 0)

cmd.Parameters.Add("@t2", OleDbType.VarChar).Value = borrowedItem(counter, 1)

cmd.Parameters.Add("@t3", OleDbType.VarChar).Value = borrowedItem(counter, 2)

cmd.Parameters.Add("@t4", OleDbType.VarChar).Value = borrowedItem(counter, 3)

cmd.Parameters.Add("@t5", OleDbType.VarChar).Value = borrowedItem(counter, 4)

cmd.Parameters.Add("@t6", OleDbType.VarChar).Value = requestCreated

cmd.Parameters.Add("@t7", OleDbType.VarChar).Value = borrowedItem(counter, 6)

cmd.Parameters.Add("@t8", OleDbType.VarChar).Value = borrowedItem(counter, 7)

cmd.Parameters.Add("@t9", OleDbType.VarChar).Value = borrowedItem(counter, 8)

cmd.Parameters.Add("@t10", OleDbType.VarChar).Value = borrowedItem(counter, 9)

cmd.Parameters.Add("@t11", OleDbType.VarChar).Value = borrowedItem(counter, 10)

cmd.Parameters.Add("@t11", OleDbType.VarChar).Value = counter.ToString

Dim a As String = borrowedItem(counter, 1)

Dim b As String = borrowedItem(counter, 3)

counter = counter + 1

deduct\_ToolStockLevel(a, b)

cmd.ExecuteNonQuery()

cmd.Dispose()

End While

'MsgBox("Item/s has been save to Database", MsgBoxStyle.OkOnly, "Save")

to\_Print()

Me.DataGridView1.Rows.Clear()

x = 0

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Remove\_BorrowTools()

If (Me.DataGridView1.Rows.Count) > 0 Then

For counter As Integer = Me.DataGridView1.SelectedRows.Count - 1 To 0 Step -1

DataGridView1.Rows.RemoveAt(DataGridView1.SelectedRows(counter).Index)

x = x - 1

Next

Else

MsgBox("No Selected Rows")

End If

End Sub

Private Sub Generate\_BorrowCode()

Dim a As String

Dim b As String

Dim c As String

Dim rand As New Random

a = Now.Month

b = Now.Day

c = rand.Next(100, 999)

borrowCode.Text = a + b + c

End Sub

Private Sub Load\_ToolsItem()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandText = "SELECT tPartNum from tblToolslist"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.cbToolsID.DataSource = dt

.cbToolsID.DisplayMember = "tPartNum"

End With

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub cbToolsID\_SelectedIndexChanged(sender As Object, e As System.EventArgs) Handles cbToolsID.SelectedIndexChanged

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dr As OleDbDataReader

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "SELECT \* from tblToolslist where tPartNum = '" & cbToolsID.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

txtToolsName.Text = dr("tPartName")

txtToolsDescript.Text = dr("tDescript")

End While

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Clear\_Toolsfields()

cbToolsID.Text = ""

txtToolsName.Text = ""

txtToolsDescript.Text = ""

txtQuantity.Text = ""

'txtEmpid.Text = ""

'txtEmpName.Text = ""

'requestCreated = ""

End Sub

Private Sub deduct\_ToolStockLevel(ByVal itemNumber As String, ByVal quantity As String)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim updatedStocks As Integer = 0

Dim sql As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

sql = "select tstockLevel FROM tblToolslist WHERE tPartNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "tstockLevel")

updatedStocks = ds.Tables(0).Rows(0).Item("tstockLevel")

updatedStocks = updatedStocks - CInt(quantity)

cmd.CommandText = "UPDATE tblToolslist SET tstockLevel = @tstockLevel , tstockDateUpdate = @tstockDateUpdate Where tPartNum = '" & itemNumber & "'"

cmd.Parameters.Add("@tstockLevel", OleDbType.Integer).Value = updatedStocks

cmd.Parameters.Add("@tstockDateUpdate", OleDbType.VarChar).Value = rDate + " " + rTime

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub PrintDocument1\_PrintPage(sender As Object, e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim fnt As New Font("Arial Black", 20)

Dim fnt1 As New Font("Segoe UI", 12)

Dim fnt2 As New Font("Tahoma", 16)

Dim center, Mleft, Mright, buttton, top As Integer

Dim mystring As String = "Saad Specialist Hospital"

Mleft = e.PageSettings.Margins.Left

Mright = e.PageSettings.Margins.Right

buttton = e.PageSettings.Margins.Bottom

top = e.PageSettings.Margins.Top

center = e.PageSettings.PaperSize.Width / 2

Dim temp = CInt(center - (mystring.Length / 2) \* 10.5)

Dim x, y, width, height As Integer

x = 50

y = 350

width = 150

height = 22

Dim fntH As Integer = fnt1.Height

Dim rct As New Rectangle(x, y, width, height)

Dim rct1 As New Rectangle(200, 350, 120, height)

Dim rct2 As New Rectangle(320, 350, 130, height)

Dim rct3 As New Rectangle(450, 350, 200, height)

Dim rct4 As New Rectangle(650, 350, 120, height)

Dim rct5 As New Rectangle(x, y, width, height)

Dim newImage As Image = Image.FromFile("logo.png")

Dim Pcenter As New PointF(330, 20)

e.Graphics.DrawImage(newImage, Pcenter)

e.Graphics.DrawString(mystring, fnt, Brushes.Black, 250, 150)

e.Graphics.DrawString("I.T.D.", fnt, Brushes.Black, 385, 180)

e.Graphics.DrawString("Borrow Request Form", fnt2, Brushes.Black, 325, 260)

e.Graphics.DrawString("Resquest by:" + " " + borrowedItem(0, 7), fnt1, Brushes.Black, 50, y - 44)

e.Graphics.DrawString("Project:" + " " + "N Project", fnt1, Brushes.Black, 50, y - 22)

e.Graphics.DrawString("Date:" + " " + rDate, fnt1, Brushes.Black, 650, y - 22)

e.Graphics.FillRectangle(Brushes.Gray, rct)

e.Graphics.FillRectangle(Brushes.Gray, rct1)

e.Graphics.FillRectangle(Brushes.Gray, rct2)

e.Graphics.FillRectangle(Brushes.Gray, rct3)

e.Graphics.FillRectangle(Brushes.Gray, rct4)

Dim arrRct() As Rectangle = {rct, rct1, rct2, rct3, rct4}

e.Graphics.DrawRectangles(Pens.Black, arrRct)

e.Graphics.DrawString("Request Number", fnt1, Brushes.Black, x, y)

e.Graphics.DrawString("Item Number", fnt1, Brushes.Black, 200, y)

e.Graphics.DrawString("Item Name", fnt1, Brushes.Black, 320, y)

e.Graphics.DrawString("Description", fnt1, Brushes.Black, 450, y)

e.Graphics.DrawString("Quantity", fnt1, Brushes.Black, 650, y)

'Dim row, counter As Integer

'counter = 5

For row = 0 To num\_item Step 1 'Borrowed Code

e.Graphics.DrawRectangle(Pens.Black, x, y + 22, width, height)

e.Graphics.DrawString(borrowedItem(row, 0), fnt1, Brushes.Black, x, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1 ' Item Number

e.Graphics.DrawRectangle(Pens.Black, 200, y + 22, 120, height)

e.Graphics.DrawString(borrowedItem(row, 1), fnt1, Brushes.Black, 200, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1 'Item Name

e.Graphics.DrawRectangle(Pens.Black, 320, y + 22, 130, height)

e.Graphics.DrawString(borrowedItem(row, 2), fnt1, Brushes.Black, 320, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1 'Description

e.Graphics.DrawRectangle(Pens.Black, 450, y + 22, 200, height)

e.Graphics.DrawString(borrowedItem(row, 4), fnt1, Brushes.Black, 450, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1 'Quantity

e.Graphics.DrawRectangle(Pens.Black, 650, y + 22, 120, height)

e.Graphics.DrawString(borrowedItem(row, 3), fnt1, Brushes.Black, 650, y + 22)

y = y + 22

Next

e.Graphics.DrawLine(Pens.Black, 50, 750, 250, 750)

e.Graphics.DrawLine(Pens.Black, 300, 750, 550, 750)

e.Graphics.DrawLine(Pens.Black, 600, 750, 800, 750)

e.Graphics.DrawString("Requested By:", fnt1, Brushes.Black, 100, 760)

e.Graphics.DrawString("StoreKeeper", fnt1, Brushes.Black, 370, 760)

e.Graphics.DrawString("Manager", fnt1, Brushes.Black, 680, 760)

End Sub

Private Sub to\_Print()

Dim toPrint As String = MsgBox("Do you want to Print?", MsgBoxStyle.OkCancel, "Print Option")

If toPrint = DialogResult.OK Then

PrintPreviewDialog1.Document = PrintDocument1

PrintPreviewDialog1.WindowState = FormWindowState.Maximized

PrintPreviewDialog1.ShowDialog()

Else

MsgBox("You can Print the Item Later")

End If

End Sub

End Class

Imports System.Data.OleDb

Imports System.IO

Public Class frmChangePass

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As OleDbDataAdapter

Dim ds As DataSet

Dim formUser As New frmUsers

Private Sub frmChangePass\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Load\_MyRecord()

End Sub

Private Sub Load\_MyRecord()

txtUserName.Text = tagItem

txtUserName.Enabled = False

End Sub

Private Sub btnCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCancel.Click

Me.Close()

End Sub

Private Sub btnSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click

Save\_Password()

Me.Close()

End Sub

Private Sub Save\_Password()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim dr As OleDbDataReader

If Len(Trim(txtOldPass.Text)) = 0 Then

MsgBox("Please Enter Password")

txtOldPass.Focus()

Exit Sub

End If

If Len(Trim(txtNewPass.Text)) = 0 Or Len(Trim(txtConPass.Text)) = 0 Then

MessageBox.Show("Please Complete New Password and Conform", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

txtOldPass.Focus()

Exit Sub

End If

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.CommandType = CommandType.Text

cmd.CommandText = "Select [uName],[pasW] from tbluser where uName = '" & tagItem & "'"

cmd.Connection = cn

dr = cmd.ExecuteReader

If dr.Read Then

If dr("pasW") = txtOldPass.Text Then

If Len(Trim(txtNewPass.Text)) > 0 Then

If txtNewPass.Text = txtConPass.Text Then

Dim cmd1 As New OleDb.OleDbCommand("update tblUser set pasW = @pasW where uName = '" & tagItem & "'", cn)

Dim passWord As OleDbParameter = New OleDbParameter("@pasW", OleDbType.VarChar, 8)

passWord.Value = txtNewPass.Text.ToString

cmd1.Parameters.Add(passWord)

If cmd1.ExecuteNonQuery Then

cn.Close()

MsgBox("Password Update Succesfully", MsgBoxStyle.Information, "Updated")

End If

Else

MsgBox("Please make sure that the new password is correctly conform", MsgBoxStyle.Critical, "Failed To Update")

End If

Else

MessageBox.Show("Please Enter New Password", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

txtOldPass.Focus()

End If

Else

MsgBox("Old Password does not match." + vbCrLf + "Wrong Password.")

End If

End If

Catch ex As Exception

MsgBox(ex.ToString)

End Try

End Sub

Private Sub btnClear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClear.Click

Me.txtOldPass.Text = ""

Me.txtNewPass.Text = ""

Me.txtConPass.Text = ""

txtOldPass.Focus()

End Sub

End Class

Imports System.IO

Imports System.Data

Imports System.Data.OleDb

Imports System.Data.OleDb.OleDbConnection

Public Class frmLogin

' TODO: Insert code to perform custom authentication using the provided username and password

' (See http://go.microsoft.com/fwlink/?LinkId=35339).

' The custom principal can then be attached to the current thread's principal as follows:

' My.User.CurrentPrincipal = CustomPrincipal

' where CustomPrincipal is the IPrincipal implementation used to perform authentication.

' Subsequently, My.User will return identity information encapsulated in the CustomPrincipal object

' such as the username, display name, etc.

Private Sub OK\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OK.Click

Dim cn As New OleDbConnection

Dim cmd As New OleDbCommand

Dim dr As OleDbDataReader

If Len(Trim(UsernameTextBox.Text)) = 0 Then

MessageBox.Show("Please enter user name", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

UsernameTextBox.Focus()

Exit Sub

End If

If Len(Trim(PasswordTextBox.Text)) = 0 Then

MessageBox.Show("Please enter password", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

PasswordTextBox.Focus()

Exit Sub

End If

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandText = "Select [fName], [mName], [lName], [uName], [pasW], [myLevel] from tblUser where uName = @Uname"

Dim UName As OleDbParameter = New OleDbParameter("@Uname", OleDbType.VarWChar, 20)

UName.Value = UsernameTextBox.Text.ToString()

cmd.Parameters.Add(UName)

cmd.Connection = cn

'to read

dr = cmd.ExecuteReader

If dr.Read Then

If (dr("pasW")) = PasswordTextBox.Text Then

logInUser = dr("myLevel")

logInBy = dr("fName") + " " + dr("mName") + " " + dr("lName")

cn.Close()

Dim mainForm As New frmMain

Me.Hide()

mainForm.Show()

Else

MessageBox.Show("Incorrect Password", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)

cn.Close()

Label1.Visible = True

Label1.Text = "Forget Password"

Label1.Focus()

End If

Else

MessageBox.Show("UserName is wrong", "Input Error", MessageBoxButtons.OK, MessageBoxIcon.Warning)

cn.Close()

UsernameTextBox.Focus()

Return

End If

Catch x As Exception

MessageBox.Show(x.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

End Sub

Private Sub Cancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Cancel.Click

End

End Sub

Private Sub UsernameTextBox\_KeyPress(sender As Object, e As System.Windows.Forms.KeyPressEventArgs) Handles UsernameTextBox.KeyPress

Dim NotAllowed As String = "~`%^&+={[}]()!:,;'><?/|\-.#+"

'allowed characters only ( \_ $ \*@)

If e.KeyChar <> ControlChars.Back = True Then

If NotAllowed.IndexOf(e.KeyChar) = -1 = False Then

e.Handled = True

End If

End If

End Sub

Private Sub PasswordTextBox\_KeyPress(sender As Object, e As System.Windows.Forms.KeyPressEventArgs) Handles PasswordTextBox.KeyPress

Dim NotAllowed As String = "~`%^&+={[}]()!:,;'><?/|\-.#+"

'allowed characters only ( \_ $ \*@)

If e.KeyChar <> ControlChars.Back = True Then

If NotAllowed.IndexOf(e.KeyChar) = -1 = False Then

e.Handled = True

End If

End If

End Sub

End Class

Imports System.Data.OleDb

Imports System.IO

Public Class frmRequest

'Dim itemCounter As Integer = 0

Dim requestItem(50, 10) As String

Dim x As Integer = 0

Dim y As Integer = 0

Dim num\_item As Integer

Private Sub frmRequest\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Load\_forRequest()

Clear\_fields()

Generate\_RequestCode()

End Sub

Private Sub Clear\_fields()

cbItemNum.Text = ""

txtItemName.Text = ""

txtQuantity.Text = ""

txtDescription.Text = ""

txtrequestedby.Text = ""

txtApproved1.Text = logInBy

txtApproved2.Text = ""

txtApproved3.Text = ""

End Sub

Private Sub Generate\_RequestCode()

Dim a As String

Dim b As String

Dim c As String

Dim rand As New Random

a = Now.Month

b = Now.Day

c = rand.Next(100, 999)

Label6.Text = a + b + c

End Sub

Private Sub Load\_forRequest()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandText = "SELECT itemNum from tblItemlist"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.cbItemNum.DataSource = dt

.cbItemNum.DisplayMember = "itemNum"

End With

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub btnAdd\_Click(sender As System.Object, e As System.EventArgs) Handles btnAdd.Click

Add\_Request()

Clear\_fields()

End Sub

Private Sub Add\_Request()

Dim status As Boolean

If Len(Trim(txtQuantity.Text)) > 0 Then

If IsNumeric(txtQuantity.Text) = True Then

requestItem(x, 0) = Label6.Text

requestItem(x, 1) = cbItemNum.Text

requestItem(x, 2) = txtItemName.Text

requestItem(x, 3) = txtDescription.Text

requestItem(x, 4) = txtQuantity.Text

requestItem(x, 5) = requestCreated

requestItem(x, 6) = txtrequestedby.Text

requestItem(x, 7) = txtApproved1.Text

requestItem(x, 8) = txtApproved2.Text

requestItem(x, 9) = txtApproved3.Text

status = check\_ItemAvailability(requestItem(x, 1), requestItem(x, 4))

If status = True Then

x = x + 1

num\_item = x

View\_RequestItem()

Else

MsgBox("Insuficient Stock Item '" & txtItemName.Text & "'")

Exit Sub

End If

Else

MsgBox("Quantity is should be numeric")

End If

Else

MsgBox("Quantity mus be filled up")

End If

End Sub

Private Sub View\_RequestItem()

Me.DataGridView1.Rows.Add(Label6.Text, cbItemNum.Text, txtItemName.Text, txtDescription.Text, txtQuantity.Text)

End Sub

Private Sub cbItemNum\_SelectedIndexChanged(sender As Object, e As System.EventArgs) Handles cbItemNum.SelectedIndexChanged

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dr As OleDbDataReader

Dim dt As New DataTable

Dim sql As String = String.Empty

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "SELECT \* from tblItemlist where itemNum = '" & cbItemNum.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

txtItemName.Text = dr("itemName")

txtDescription.Text = dr("iDescript")

End While

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub btnRemove\_Click(sender As System.Object, e As System.EventArgs) Handles btnRemove.Click

Dim confirm As String = MsgBox("Are you sure you want to delete this Item?", MsgBoxStyle.YesNo, "Delete Item")

If confirm = DialogResult.Yes Then

Delete\_Request()

Else

MsgBox("Delete event was canceled", MsgBoxStyle.Critical, "Canceled")

End If

End Sub

Private Sub Delete\_Request()

If Me.DataGridView1.SelectedRows.Count > 0 Then

For counter As Integer = Me.DataGridView1.SelectedRows.Count - 1 To 0 Step -1

DataGridView1.Rows.RemoveAt(DataGridView1.SelectedRows(counter).Index)

x = x - 1

num\_item = x

Next

Else

MsgBox("No Selected Rows")

End If

End Sub

Private Sub Save()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim counter As Integer = 0

Try

If requestItem(counter, 1) = "" Then

MsgBox("No Item requested")

Exit Sub

Else

cn = New OleDbConnection(Get\_Constring)

cn.Open()

While counter < x

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "INSERT INTO tblRequestItem ( requestItemNum, itemNum, itemName, iDescript, iQuantity, iDate, requestby, storekeeper, approved2, approved3,requestNum ) VALUES(@i1, @i2, @i3, @i4, @i5, @i6, @i7, @i8, @i9, @i10, @i11) "

cmd.Parameters.Add("@i1", OleDbType.Numeric).Value = requestItem(counter, 0)

cmd.Parameters.Add("@i2", OleDbType.VarChar).Value = requestItem(counter, 1)

cmd.Parameters.Add("@i3", OleDbType.VarChar).Value = requestItem(counter, 2)

cmd.Parameters.Add("@i4", OleDbType.VarChar).Value = requestItem(counter, 3)

cmd.Parameters.Add("@i5", OleDbType.Numeric).Value = requestItem(counter, 4)

cmd.Parameters.Add("@i6", OleDbType.VarChar).Value = requestItem(counter, 5)

cmd.Parameters.Add("@i7", OleDbType.VarChar).Value = requestItem(counter, 6)

cmd.Parameters.Add("@i8", OleDbType.VarChar).Value = requestItem(counter, 7)

cmd.Parameters.Add("@i9", OleDbType.VarChar).Value = requestItem(counter, 8)

cmd.Parameters.Add("@i10", OleDbType.VarChar).Value = requestItem(counter, 9)

cmd.Parameters.Add("@i11", OleDbType.VarChar).Value = counter.ToString

Dim a As String = requestItem(counter, 1)

Dim b As String = requestItem(counter, 4)

counter = counter + 1

deduct\_StockLevel(a, b)

cmd.ExecuteNonQuery()

cmd.Dispose()

End While

'MsgBox("Item/s has been save to Database", MsgBoxStyle.OkOnly, "Save")

to\_Print()

'num\_item = x

Me.DataGridView1.Rows.Clear()

x = 0

End If

Catch ex As Exception

Finally

cn.Close()

End Try

End Sub

Private Sub btnSave\_Click(sender As System.Object, e As System.EventArgs) Handles btnSave.Click

Save()

Generate\_RequestCode()

End Sub

Public Sub deduct\_StockLevel(ByVal itemNumber As String, ByVal quantity As String)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim updatedStocks As Integer = 0

Dim sql As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

sql = "select stockLevel FROM tblItemList WHERE itemNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "stockLevel")

updatedStocks = ds.Tables(0).Rows(0).Item("StockLevel")

updatedStocks = updatedStocks - CInt(quantity)

cmd.CommandText = "UPDATE tblItemList SET stockLevel = @stockLevel , stockDateUpdate = @stockDateUpdate Where itemNum = '" & itemNumber & "'"

cmd.Parameters.Add("@stockLevel", OleDbType.Integer).Value = updatedStocks

cmd.Parameters.Add("@stockDateUpdate", OleDbType.VarChar).Value = rDate + " " + rTime

cmd.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub PrintDocument1\_PrintPage(sender As Object, e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim fnt As New Font("Arial Black", 20)

Dim fnt1 As New Font("Segoe UI", 12)

Dim fnt2 As New Font("Tahoma", 16)

Dim center, Mleft, Mright, buttton, top As Integer

Dim mystring As String = "Saad Specialist Hospital"

Mleft = e.PageSettings.Margins.Left

Mright = e.PageSettings.Margins.Right

buttton = e.PageSettings.Margins.Bottom

top = e.PageSettings.Margins.Top

center = e.PageSettings.PaperSize.Width / 2

Dim temp = CInt(center - (mystring.Length / 2) \* 10.5)

Dim x, y, width, height As Integer

x = 50

y = 350

width = 150

height = 22

Dim fntH As Integer = fnt1.Height

Dim rct As New Rectangle(x, y, width, height)

Dim rct1 As New Rectangle(200, 350, 120, height)

Dim rct2 As New Rectangle(320, 350, 130, height)

Dim rct3 As New Rectangle(450, 350, 200, height)

Dim rct4 As New Rectangle(650, 350, 120, height)

Dim rct5 As New Rectangle(x, y, width, height)

Dim newImage As Image = Image.FromFile("logo.png")

Dim Pcenter As New PointF(330, 20)

e.Graphics.DrawImage(newImage, Pcenter)

e.Graphics.DrawString(mystring, fnt, Brushes.Black, 250, 150)

e.Graphics.DrawString("I.T.D.", fnt, Brushes.Black, 385, 180)

e.Graphics.DrawString("Material Request Form", fnt2, Brushes.Black, 325, 260)

e.Graphics.DrawString("Resquest by:" + " " + "User", fnt1, Brushes.Black, 50, y - 44)

e.Graphics.DrawString("Project:" + " " + "N Project", fnt1, Brushes.Black, 50, y - 22)

e.Graphics.DrawString("Date:" + " " + rDate, fnt1, Brushes.Black, 650, y - 22)

e.Graphics.FillRectangle(Brushes.Gray, rct)

e.Graphics.FillRectangle(Brushes.Gray, rct1)

e.Graphics.FillRectangle(Brushes.Gray, rct2)

e.Graphics.FillRectangle(Brushes.Gray, rct3)

e.Graphics.FillRectangle(Brushes.Gray, rct4)

Dim arrRct() As Rectangle = {rct, rct1, rct2, rct3, rct4}

e.Graphics.DrawRectangles(Pens.Black, arrRct)

e.Graphics.DrawString("Request Number", fnt1, Brushes.Black, x, y)

e.Graphics.DrawString("Item Number", fnt1, Brushes.Black, 200, y)

e.Graphics.DrawString("Item Name", fnt1, Brushes.Black, 320, y)

e.Graphics.DrawString("Description", fnt1, Brushes.Black, 450, y)

e.Graphics.DrawString("Quantity", fnt1, Brushes.Black, 650, y)

'Dim row, counter As Integer

'counter = 5

For row = 0 To num\_item Step 1

e.Graphics.DrawRectangle(Pens.Black, x, y + 22, width, height)

e.Graphics.DrawString(requestItem(row, 0), fnt1, Brushes.Black, x, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1

e.Graphics.DrawRectangle(Pens.Black, 200, y + 22, 120, height)

e.Graphics.DrawString(requestItem(row, 1), fnt1, Brushes.Black, 200, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1

e.Graphics.DrawRectangle(Pens.Black, 320, y + 22, 130, height)

e.Graphics.DrawString(requestItem(row, 2), fnt1, Brushes.Black, 320, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1

e.Graphics.DrawRectangle(Pens.Black, 450, y + 22, 200, height)

e.Graphics.DrawString(requestItem(row, 3), fnt1, Brushes.Black, 450, y + 22)

y = y + 22

Next

y = 350

For row = 0 To num\_item Step 1

e.Graphics.DrawRectangle(Pens.Black, 650, y + 22, 120, height)

e.Graphics.DrawString(requestItem(row, 4), fnt1, Brushes.Black, 650, y + 22)

y = y + 22

Next

e.Graphics.DrawLine(Pens.Black, 50, 750, 250, 750)

e.Graphics.DrawLine(Pens.Black, 300, 750, 550, 750)

e.Graphics.DrawLine(Pens.Black, 600, 750, 800, 750)

e.Graphics.DrawString("Requested By:", fnt1, Brushes.Black, 100, 760)

e.Graphics.DrawString("StoreKeeper", fnt1, Brushes.Black, 370, 760)

e.Graphics.DrawString("Manager", fnt1, Brushes.Black, 680, 760)

End Sub

Private Sub to\_Print()

Dim toPrint As String = MsgBox("Do you want to Print?", MsgBoxStyle.YesNo, "Print Option")

If toPrint = DialogResult.Yes Then

PrintPreviewDialog1.Document = PrintDocument1

PrintPreviewDialog1.WindowState = FormWindowState.Maximized

PrintPreviewDialog1.ShowDialog()

Else

MsgBox("You can Print the Item Later")

End If

End Sub

End Class

Imports System.IO

Imports System.Data

Imports System.Data.OleDb

Imports System.Data.OleDb.OleDbConnection

Public Class frmStocks

Private Sub btnSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnSave.Click

Save\_Items()

End Sub

Private Sub btnClear\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Dim confirm As New DialogResult

confirm = MsgBox("Do you want to Reset the Password? ", MsgBoxStyle.YesNo)

If confirm = DialogResult.Yes Then

Clear\_Fields()

Else

Exit Sub

End If

End Sub

Private Sub Save\_Items()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = "INSERT INTO tblStocks ( pRequest, invoiceNum, dDeliver, delby, itemNum, itemName, idescript, iQuantity, iUnit, iCategory, iLocation, entryDateTime, entryBy )"

str = str & "VALUES ( @D1, @D2, @D3, @D4, @D5, @D6, @D7, @D8, @D9, @D10, @D11, @D12, @D13 )"

cmd.CommandText = str

cmd.Parameters.Add("@d1", OleDbType.VarChar).Value = IIf(Len(Trim(txtPR.Text)) > 0, txtPR.Text, "")

cmd.Parameters.Add("@d2", OleDbType.VarChar).Value = IIf(Len(Trim(txtInvoice.Text)) > 0, txtInvoice.Text, "")

cmd.Parameters.Add("@d3", OleDbType.VarChar).Value = dtpDeliver.Text.ToString()

cmd.Parameters.Add("@d4", OleDbType.VarChar).Value = IIf(Len(Trim(txtDeliverBy.Text)) > 0, txtDeliverBy.Text, "")

cmd.Parameters.Add("@d5", OleDbType.VarChar).Value = IIf(Len(Trim(cbItemNum.Text)) > 0, cbItemNum.Text, "")

cmd.Parameters.Add("@d6", OleDbType.VarChar).Value = IIf(Len(Trim(txtItemName.Text)) > 0, txtItemName.Text, "")

cmd.Parameters.Add("@d7", OleDbType.VarChar).Value = IIf(Len(Trim(txtDescript.Text)) > 0, txtDescript.Text, "")

cmd.Parameters.Add("@d8", OleDbType.VarChar).Value = IIf(Len(Trim(txtQuantity.Text)) > 0, txtQuantity.Text, "")

cmd.Parameters.Add("@d9", OleDbType.VarChar).Value = IIf(Len(Trim(txtUnit.Text)) > 0, txtUnit.Text, "")

cmd.Parameters.Add("@d10", OleDbType.VarChar).Value = IIf(Len(Trim(txtCategory.Text)) > 0, txtCategory.Text, "")

cmd.Parameters.Add("@d11", OleDbType.VarChar).Value = IIf(Len(Trim(txtLocation.Text)) > 0, txtLocation.Text, "")

cmd.Parameters.Add("@d12", OleDbType.VarChar).Value = Format(Now, "dd/MM/yyy") + " " + Now.ToLongTimeString.ToString()

cmd.Parameters.Add("@d13", OleDbType.VarChar).Value = "user"

If cmd.ExecuteNonQuery() Then

MsgBox("Data has been save", MsgBoxStyle.Information, "Save Data")

cmd.CommandText = "SELECT @@Identity"

txtPR.Text = cmd.ExecuteScalar()

Add\_StockLevel(cbItemNum.Text, txtQuantity.Text)

Clear\_Fields()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Clear\_Fields()

txtPR.Text = ""

txtInvoice.Text = ""

dtpDeliver.Value = Today

txtDeliverBy.Text = ""

cbItemNum.Text = ""

txtItemName.Text = ""

txtDescript.Text = ""

txtQuantity.Text = ""

txtUnit.Text = ""

txtLocation.Text = ""

End Sub

Private Sub Load\_forEntry(ByVal listFrom As String)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

If listFrom = "Materials" Then

cmd.CommandText = "SELECT itemNum from tblItemlist"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.cbItemNum.DataSource = dt

.cbItemNum.DisplayMember = "itemNum"

End With

Else

cmd.CommandText = "SELECT tPartNum from tblToolslist"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.cbItemNum.DataSource = dt

.cbItemNum.DisplayMember = "tPartNum"

End With

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub frmStocks\_Load(sender As Object, e As System.EventArgs) Handles Me.Load

Clear\_Fields()

End Sub

Private Sub cbItemNum\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles cbItemNum.SelectedIndexChanged

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim dr As OleDbDataReader

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

If txtCategory.Text = "Materials" Then

cmd.CommandText = "SELECT \* from tblItemlist where itemNum = '" & cbItemNum.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

txtItemName.Text = dr("itemName")

txtDescript.Text = dr("iDescript")

End While

Else

cmd.CommandText = "SELECT \* from tblToolslist where tPartNum = '" & cbItemNum.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

txtItemName.Text = dr("tPartName")

txtDescript.Text = dr("tDescript")

End While

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Public Sub Add\_StockLevel(ByVal itemNumber As String, ByVal quantity As String)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim updatedStocks As Integer = 0

Dim sql As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

If txtCategory.Text = "Materials" Then

sql = "select stockLevel FROM tblItemList WHERE itemNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "stockLevel")

updatedStocks = ds.Tables(0).Rows(0).Item("StockLevel")

updatedStocks = updatedStocks + CInt(quantity)

cmd.CommandText = "UPDATE tblItemList SET stockLevel = @stockLevel , stockDateUpdate = @stockDateUpdate Where itemNum = '" & itemNumber & "'"

cmd.Parameters.Add("@stockLevel", OleDbType.Integer).Value = updatedStocks

cmd.Parameters.Add("@stockDateUpdate", OleDbType.VarChar).Value = rDate + " " + rTime

cmd.ExecuteNonQuery()

Else

sql = "Select tstockLevel From tblToolslist Where tPartNum = '" & itemNumber & "'"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "tstockLevel")

updatedStocks = ds.Tables(0).Rows(0).Item("tstockLevel")

updatedStocks = updatedStocks + CInt(quantity)

cmd.CommandText = "UPDATE tblToolslist SET tstockLevel = @stockLevel , tstockDateUpdate = @stockDateUpdate Where tPartNum = '" & itemNumber & "'"

cmd.Parameters.Add("@stockLevel", OleDbType.Integer).Value = updatedStocks

cmd.Parameters.Add("@stockDateUpdate", OleDbType.VarChar).Value = rDate + " "

cmd.ExecuteNonQuery()

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub txtCategory\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles txtCategory.SelectedIndexChanged, Me.Load

Load\_forEntry(txtCategory.Text)

If txtCategory.Text = "" Then

cbItemNum.Enabled = False

txtItemName.Enabled = False

txtDescript.Enabled = False

txtQuantity.Enabled = False

txtUnit.Enabled = False

txtLocation.Enabled = False

Clear\_Fields()

Else

cbItemNum.Enabled = True

txtItemName.Enabled = True

txtDescript.Enabled = True

txtQuantity.Enabled = True

txtUnit.Enabled = True

txtLocation.Enabled = True

Clear\_Fields()

End If

End Sub

Private Sub btnClear\_Click\_1(sender As System.Object, e As System.EventArgs) Handles btnClear.Click

Clear\_Fields()

End Sub

End Class

Imports System.Data.OleDb

Public Class frmtrans

Dim toprint(20, 20) As String

' Private Sub Button1\_Click(sender As System.Object, e As System.EventArgs)

' Load\_ItemList()

' End Sub

Private Sub Load\_ItemList()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

If TabControl1.SelectedIndex = True Then

str = " SELECT (requestItemNum) as [Request Item], (itemNum) as [Iten Number], (itemName) as [Name], (iDate) as [Date] FROM tblRequestItem order by itemNum "

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.DgvRequest.DataSource = dt

Dim column As DataGridViewColumn = DgvRequest.Columns(2)

column.Width = 100

'Me.dataGVItem.Width = 250

'Me.dataGVItem.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.ColumnHeader)

Else

str = "SELECT (tBorrowCode) as [Borrow Code], (tItemID)as [Item #], (tName) as [Item Name], (tDateCreated) as [Date] from tblBorrowTools order by tBorrowCode"

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.DgvRequest.DataSource = dt

End If

If dt.Rows.Count = 0 Then

MsgBox("No record found!")

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub TabControl1\_Click(sender As Object, e As System.EventArgs) Handles TabControl1.Click

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = " SELECT (requestItemNum) as [Request Item], (itemNum) as [Iten Number], (itemName) as [Name], (iDate) as [Date] FROM tblRequestItem order by itemNum "

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.DgvRequest.DataSource = dt

Dim column As DataGridViewColumn = DgvRequest.Columns(2)

column.Width = 150

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub TabPage2\_Click(sender As Object, e As System.EventArgs) Handles TabControl1.Click

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = "SELECT (tBorrowCode) as [Borrow Code], (tItemID) as [Item #], (tName) as [Item Name], (tDateCreated) as [Date] from tblBorrowTools order by tBorrowCode"

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.DgvBorrowed.DataSource = dt

Dim column As DataGridViewColumn = DgvBorrowed.Columns(2)

column.Width = 150

If dt.Rows.Count = 0 Then

MsgBox("No record found!")

End If

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub DgvRequest\_DoubleClick(sender As Object, e As System.EventArgs) Handles DgvRequest.DoubleClick

If DgvRequest.RowCount > 0 Then

Dim clicked As String = Me.DgvRequest.Item(0, Me.DgvRequest.CurrentRow.Index).Value

CollectReq\_sameCode(clicked)

End If

End Sub

Private Sub CollectReq\_sameCode(ByVal clicked)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = "SELECT requestNum as [#], requestItemNum as [Request Number], itemNum as [Item Number], itemName as [Name], iDescript as [Description], iQuantity as [Quantity] from tblRequestItem where requestItemNum = '" & clicked & "'"

'CStr(cmd.CommandText) = cmd.CommandText

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.Dgv.DataSource = dt

Dim column As DataGridViewColumn = Dgv.Columns(1)

column.Width = 180

cn.Close()

End Sub

Private Sub DgvBorrowed\_DoubleClick(sender As Object, e As System.EventArgs) Handles DgvBorrowed.DoubleClick

If DgvBorrowed.RowCount > 0 Then

Dim clicked As String = Me.DgvBorrowed.Item(0, Me.DgvBorrowed.CurrentRow.Index).Value

CollectBor\_sameCode(clicked)

End If

End Sub

Private Sub CollectBor\_sameCode(ByRef clicked)

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim str As String = String.Empty

cn = New OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

str = "SELECT tBorrowNum as [#], tBorrowCode as [Request Number], tItemID as [Item Number], tName as [Name], tDescript as [Description], tQuantity as [Quantity] from tblBorrowTools where tBorrowCode = '" & clicked & "'"

'CStr(cmd.CommandText) = cmd.CommandText

cmd.CommandText = str

da.SelectCommand = cmd

da.Fill(dt)

Me.Dgv.DataSource = dt

Dim column As DataGridViewColumn = Dgv.Columns(1)

column.Width = 180

cn.Close()

End Sub

Private Sub printReq\_Click(sender As System.Object, e As System.EventArgs) Handles printReq.Click

to\_Print()

End Sub

Private Sub PrintDocument1\_PrintPage(sender As Object, e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim fnt As New Font("Arial Black", 20)

Dim fnt1 As New Font("Segoe UI", 12)

Dim fnt2 As New Font("Tahoma", 16)

Dim center, Mleft, Mright, buttton, top As Integer

Dim mystring As String = "Saad Specialist Hospital"

Mleft = e.PageSettings.Margins.Left

Mright = e.PageSettings.Margins.Right

buttton = e.PageSettings.Margins.Bottom

top = e.PageSettings.Margins.Top

center = e.PageSettings.PaperSize.Width / 2

Dim temp = CInt(center - (mystring.Length / 2) \* 10.5)

Dim x, y, width, height As Integer

x = 50

y = 350

width = 150

height = 22

Dim fntH As Integer = fnt1.Height

Dim rct As New Rectangle(x, y, width, height)

Dim rct1 As New Rectangle(200, 350, 120, height)

Dim rct2 As New Rectangle(320, 350, 130, height)

Dim rct3 As New Rectangle(450, 350, 200, height)

Dim rct4 As New Rectangle(650, 350, 120, height)

Dim rct5 As New Rectangle(x, y, width, height)

Dim newImage As Image = Image.FromFile("logo.png")

Dim Pcenter As New PointF(330, 20)

e.Graphics.DrawImage(newImage, Pcenter)

e.Graphics.DrawString(mystring, fnt, Brushes.Black, 250, 150)

e.Graphics.DrawString("I.T.D.", fnt, Brushes.Black, 385, 180)

e.Graphics.DrawString("Material Request Form", fnt2, Brushes.Black, 325, 260)

e.Graphics.DrawString("Resquest by:" + " " + "User", fnt1, Brushes.Black, 50, y - 44)

e.Graphics.DrawString("Project:" + " " + "N Project", fnt1, Brushes.Black, 50, y - 22)

e.Graphics.DrawString("Date:" + " " + rDate, fnt1, Brushes.Black, 650, y - 22)

e.Graphics.FillRectangle(Brushes.Gray, rct)

e.Graphics.FillRectangle(Brushes.Gray, rct1)

e.Graphics.FillRectangle(Brushes.Gray, rct2)

e.Graphics.FillRectangle(Brushes.Gray, rct3)

e.Graphics.FillRectangle(Brushes.Gray, rct4)

Dim arrRct() As Rectangle = {rct, rct1, rct2, rct3, rct4}

e.Graphics.DrawRectangles(Pens.Black, arrRct)

e.Graphics.DrawString("Request Number", fnt1, Brushes.Black, x, y)

e.Graphics.DrawString("Item Number", fnt1, Brushes.Black, 200, y)

e.Graphics.DrawString("Item Name", fnt1, Brushes.Black, 320, y)

e.Graphics.DrawString("Description", fnt1, Brushes.Black, 450, y)

e.Graphics.DrawString("Quantity", fnt1, Brushes.Black, 650, y)

'Dim row, counter As Integer

'counter = 5

For row = 0 To Me.Dgv.RowCount - 1 Step 1

e.Graphics.DrawRectangle(Pens.Black, x, y + 22, width, height)

e.Graphics.DrawString(Me.Dgv.Item(1, row).Value, fnt1, Brushes.Black, x, y + 22)

y = y + 22

Next

y = 350

For row = 0 To Me.Dgv.RowCount - 1 Step 1

e.Graphics.DrawRectangle(Pens.Black, 200, y + 22, 120, height)

e.Graphics.DrawString(Me.Dgv.Item(2, row).Value, fnt1, Brushes.Black, 200, y + 22)

y = y + 22

Next

y = 350

For row = 0 To Me.Dgv.RowCount - 1 Step 1

e.Graphics.DrawRectangle(Pens.Black, 320, y + 22, 130, height)

e.Graphics.DrawString(Me.Dgv.Item(3, row).Value, fnt1, Brushes.Black, 320, y + 22)

y = y + 22

Next

y = 350

For row = 0 To Me.Dgv.RowCount - 1 Step 1

e.Graphics.DrawRectangle(Pens.Black, 450, y + 22, 200, height)

e.Graphics.DrawString(Me.Dgv.Item(4, row).Value, fnt1, Brushes.Black, 450, y + 22)

y = y + 22

Next

y = 350

For row = 0 To Me.Dgv.RowCount - 1 Step 1

e.Graphics.DrawRectangle(Pens.Black, 650, y + 22, 120, height)

e.Graphics.DrawString(Me.Dgv.Item(5, row).Value, fnt1, Brushes.Black, 650, y + 22)

y = y + 22

Next

e.Graphics.DrawLine(Pens.Black, 50, 750, 250, 750)

e.Graphics.DrawLine(Pens.Black, 300, 750, 550, 750)

e.Graphics.DrawLine(Pens.Black, 600, 750, 800, 750)

e.Graphics.DrawString("Requested By:", fnt1, Brushes.Black, 100, 760)

e.Graphics.DrawString("StoreKeeper", fnt1, Brushes.Black, 370, 760)

e.Graphics.DrawString("Manager", fnt1, Brushes.Black, 680, 760)

End Sub

Private Sub to\_Print()

Dim toPrint As String = MsgBox("Do you want to Print?", MsgBoxStyle.YesNo, "Print Option")

If toPrint = DialogResult.Yes Then

PrintPreviewDialog1.Document = PrintDocument1

PrintPreviewDialog1.WindowState = FormWindowState.Maximized

PrintPreviewDialog1.ShowDialog()

Else

MsgBox("You can Print the Item Later")

End If

End Sub

End Class

Imports System.IO

Imports System.Data

Imports System.Data.OleDb

Imports System.Data.OleDb.OleDbConnection

Public Class frmUsers

'Dim sCurrUser As String = String.Empty

Private Sub frmUsers\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Set\_Button(0)

End Sub

Private Sub tsbSave\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbSave.Click

Dim confirm As New DialogResult

confirm = MsgBox("Do you want to save Data? ", MsgBoxStyle.YesNo, "Save")

If confirm = DialogResult.Yes Then

Save\_Record()

Set\_Button(2)

Else

Clear\_Fields()

First\_load()

End If

End Sub

Private Sub tsbNew\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbNew.Click

Set\_Button(1)

End Sub

Private Sub tsbClose\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbClose.Click

Me.Close()

End Sub

Private Sub tsbDelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbDelete.Click

Dim confirm As New DialogResult

confirm = MsgBox("Do you want to Delete is User? ", MsgBoxStyle.YesNo)

If confirm = DialogResult.Yes Then

If logInUser = "admin" Then

Delete\_Record()

Set\_Button(3)

Clear\_Fields()

Else

MsgBox("Only Administrator can delete.")

Exit Sub

End If

End If

End Sub

Private Sub btnReset\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbReset.Click

Dim confirm As New DialogResult

confirm = MsgBox("Do you want to Reset the Password? ", MsgBoxStyle.YesNo)

If confirm = DialogResult.Yes Then

Reset\_Password()

Set\_Button(2)

Else

Exit Sub

End If

End Sub

Private Sub tsbChangePass\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbChangePass.Click

Change\_Password()

End Sub

Private Sub tsbLoadUser\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbLoadUser.Click

Load\_User()

End Sub

Private Sub Change\_Password()

tagItem = txtUserName.Text

frmChangePass.ShowDialog()

End Sub

Private Sub Load\_User()

Dim cn As New OleDb.OleDbConnection

Dim da As New OleDbDataAdapter

Dim dt As New DataTable

Dim ds As New DataSet

Dim sql As String = String.Empty

Try

cn = New OleDbConnection(Get\_Constring)

cn.Open()

sql = "SELECT \* FROM tbluser order by uName"

da = New OleDbDataAdapter(sql, cn)

da.Fill(ds, "uName")

ListBox1.DataSource = ds.Tables("uName")

ListBox1.DisplayMember = "uName"

cn.Close()

Catch ex As Exception

MsgBox(ex.ToString)

End Try

Set\_Button(6)

End Sub

Private Sub Save\_Record()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Try

cn = New OleDbConnection(Get\_Constring)

'cn.ConnectionString = "Provider = Microsoft.ACE.OLEDB.12.0 ; Data Source = " & Application.StartupPath & "\SaadiT.accdb"

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

If IsNumeric(txtEmp.Text) Then

If Me.txtEmp.Tag <> 0 Then

cmd.CommandText = "INSERT INTO tbluser (empNo,fName,mName,lName,myPos,myDept,uName,pasW,myLevel,myPic,cDate\_Time) VALUES (@empNo,@fName,@mName,@lName,@myPos,@myDept,@uName,@pasW,@myLevel,@myPic,@cDate\_Time )"

Else

cmd.CommandText = "UPDATE tbluser set empNo = @empNo,fName = @fName,mName = @mName,lName = @lName,myPos = @myPos,myDept = @myDept,uName = @uName,pasW = @pasW,myLevel = @myLevel,myPic = @myPic,cDate\_Time = @cDate\_Time where empNo = @empNo"

End If

cmd.Parameters.Add("@empNo", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtEmp.Text)) > 0, Me.txtEmp.Text, "")

cmd.Parameters.Add("@fName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtFname.Text)) > 0, Me.txtFname.Text, "")

cmd.Parameters.Add("@mName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtMname.Text)) > 0, Me.txtMname.Text, "")

cmd.Parameters.Add("@lName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtLname.Text)) > 0, Me.txtLname.Text, "")

cmd.Parameters.Add("@myPos", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtPosition.Text)) > 0, Me.txtPosition.Text, "")

cmd.Parameters.Add("@myDept", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtDepartment.Text)) > 0, Me.txtDepartment.Text, "")

cmd.Parameters.Add("@uName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtUserName.Text)) > 0, Me.txtUserName.Text, "")

cmd.Parameters.Add("@pasW", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtPassword.Text)) > 0, Me.txtPassword.Text, "")

cmd.Parameters.Add("@myLevel", OleDbType.VarChar).Value = IIf((Me.opt\_User.Checked) = True, "user", "admin")

Dim memStream As New MemoryStream

Dim picbyte As Byte()

Me.userPic.Image.Save(memStream, Imaging.ImageFormat.Png)

picbyte = memStream.GetBuffer()

memStream.Read(picbyte, 0, memStream.Length)

'this is to save image

Dim userPhoto As OleDbParameter = New OleDbParameter("@myPic", SqlDbType.Image)

userPhoto.Value = picbyte

cmd.Parameters.Add(userPhoto)

'cmd.Parameters.Add("@myPic", SqlDbType.Image).Value = picbyte

cmd.Parameters.Add("@cDate\_Time", OleDbType.VarChar).Value = Format(Now, "dd/MM/yyy") + " " + Now.ToLongTimeString.ToString()

Else

MsgBox("Employee Number is invalid", MsgBoxStyle.Exclamation, "Invalid Entry")

Exit Sub

End If

cmd.ExecuteNonQuery()

MsgBox("Successfully Registered")

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Load\_Record()

'This will save into into listbox1

Label7.Text = "Double Click List" + vbCrLf + "Item to Edit"

Me.txtEmp.DataBindings.Clear() 'This will clear the binding to prevent double binding

Me.txtEmp.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "empNo", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtFname.DataBindings.Clear()

Me.txtFname.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "fName", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtMname.DataBindings.Clear()

Me.txtMname.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "mName", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtLname.DataBindings.Clear()

Me.txtLname.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "lName", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtPosition.DataBindings.Clear()

Me.txtPosition.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "myPos", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtDepartment.DataBindings.Clear()

Me.txtDepartment.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "myDept", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtUserName.DataBindings.Clear()

Me.txtUserName.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "uName", True, DataSourceUpdateMode.OnPropertyChanged))

Me.txtPassword.DataBindings.Clear()

Me.txtPassword.DataBindings.Add(New Binding("Text", ListBox1.SelectedItem, "pasW", True, DataSourceUpdateMode.OnPropertyChanged))

If (ListBox1.SelectedItem("myLevel")) = "admin" Then

opt\_Admin.Checked = True

opt\_User.Checked = False

Else

opt\_Admin.Checked = False

opt\_User.Checked = True

End If

If IsDBNull(ListBox1.SelectedItem("myPic")) Then

userPic.Image = System.Drawing.Image.FromFile(My.Application.Info.DirectoryPath & "\Image\mypic.jpg")

Else

Dim x As Byte() = ListBox1.SelectedItem("myPic")

Dim ms As MemoryStream = New MemoryStream(x)

userPic.Image = Image.FromStream(ms)

End If

Set\_Button(7)

End Sub

Private Sub Delete\_Record()

Dim con As New OleDbConnection

Dim cmd As New OleDbCommand

Try

'get connection string declared in the Module1.vb and assing it to conn variable

con = New OleDbConnection(Get\_Constring)

con.Open()

cmd.Connection = con

cmd.CommandType = CommandType.Text

cmd.CommandText = "delete from tbluser where empNo = @empNo"

cmd.Parameters.Add("@empNo", OleDbType.Numeric).Value = Me.txtEmp.Text

cmd.ExecuteNonQuery()

MsgBox("Data has been deleted.")

Catch ex As Exception

MsgBox(ErrorToString)

Finally

con.Close()

End Try

End Sub

Private Sub Set\_Button(ByVal iMode As Short)

Select Case iMode

Case 0 'First Load

tsbNew.Enabled = True

tsbSave.Enabled = False

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = True

GroupBox1.Enabled = False

GroupBox2.Enabled = False

ListBox1.Enabled = False

Case 1 'New

Me.txtEmp.Tag = 1

tsbNew.Enabled = False

tsbSave.Enabled = True

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = False

GroupBox1.Enabled = True

GroupBox2.Enabled = True

ListBox1.Enabled = False

Clear\_Fields()

Case 2 'Save

tsbNew.Enabled = True

tsbSave.Enabled = False

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = True

GroupBox1.Enabled = False

GroupBox2.Enabled = False

ListBox1.Enabled = True

Clear\_Fields()

Case 3 'Delete

tsbNew.Enabled = True

tsbSave.Enabled = False

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = True

Case 4 'Change Password

Case 5 'Reset Password

tsbNew.Enabled = False

tsbSave.Enabled = True

tsbDelete.Enabled = True

tsbChangePass.Enabled = True

tsbReset.Enabled = True

tsbLoadUser.Enabled = False

GroupBox1.Enabled = False

GroupBox2.Enabled = False

ListBox1.Enabled = True

Clear\_Fields()

Case 6 'Load User

tsbNew.Enabled = False

tsbSave.Enabled = False

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = False

GroupBox1.Enabled = False

GroupBox2.Enabled = False

ListBox1.Enabled = True

Clear\_Fields()

Case 7 ' Double Click

tsbNew.Enabled = False

tsbSave.Enabled = True

tsbDelete.Enabled = True

tsbChangePass.Enabled = True

tsbReset.Enabled = True

tsbLoadUser.Enabled = False

GroupBox1.Enabled = True

GroupBox2.Enabled = False

ListBox1.Enabled = True

End Select

End Sub

Private Sub First\_load()

tsbNew.Enabled = True

tsbSave.Enabled = False

tsbDelete.Enabled = False

tsbChangePass.Enabled = False

tsbReset.Enabled = False

tsbLoadUser.Enabled = True

GroupBox1.Enabled = False

End Sub

Private Sub Clear\_Fields()

With Me

.txtEmp.Text = ""

.txtFname.Text = ""

.txtMname.Text = ""

.txtLname.Text = ""

.txtPosition.Text = ""

.txtDepartment.Text = ""

.txtUserName.Text = ""

.txtPassword.Text = ""

.opt\_User.Checked = False

.opt\_Admin.Checked = False

.userPic.Image = System.Drawing.Image.FromFile(My.Application.Info.DirectoryPath & "\Image\mypic.jpg")

End With

End Sub

Private Sub Reset\_Password()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

If logInUser = "admin" Then

Try

cn = New OleDbConnection(Get\_Constring)

'cn.ConnectionString = "Provider = Microsoft.ACE.OLEDB.12.0 ; Data Source = " & Application.StartupPath & "\SaadiT.accdb"

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "UPDATE tbluser set empNo = @empNo,fName = @fName,mName = @mName,lName = @lName,myPos = @myPos,myDept = @myDept,uName = @uName,pasW = @pasW,myLevel = @myLevel where empNo = @empNo"

cmd.Parameters.Add("@empNo", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtEmp.Text)) > 0, Me.txtEmp.Text, "")

cmd.Parameters.Add("@fName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtFname.Text)) > 0, Me.txtFname.Text, "")

cmd.Parameters.Add("@mName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtMname.Text)) > 0, Me.txtMname.Text, "")

cmd.Parameters.Add("@lName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtLname.Text)) > 0, Me.txtLname.Text, "")

cmd.Parameters.Add("@myPos", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtPosition.Text)) > 0, Me.txtPosition.Text, "")

cmd.Parameters.Add("@myDept", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtDepartment.Text)) > 0, Me.txtDepartment.Text, "")

cmd.Parameters.Add("@uName", OleDbType.VarChar).Value = IIf(Len(Trim(Me.txtUserName.Text)) > 0, Me.txtUserName.Text, "")

cmd.Parameters.Add("@pasW", OleDbType.VarChar).Value = "admin123"

cmd.Parameters.Add("@myLevel", OleDbType.VarChar).Value = IIf((Me.opt\_User.Checked) = True, "user", "admin")

cmd.ExecuteNonQuery()

MsgBox("Password reset for this user has been made", MsgBoxStyle.OkOnly, "Reset Password ")

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

Else

MsgBox("Only Administrator can reset password", MsgBoxStyle.RetryCancel, "Protected by Administrator ")

Load\_User()

Exit Sub

End If

End Sub

Private Sub tsbCancel\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tsbCancel.Click

Clear\_Fields()

Set\_Button(0)

End Sub

Private Sub ListBox1\_DataSourceChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles tsbSave.Click

ListBox1.DataSource = Nothing

ListBox1.Items.Clear()

End Sub

Private Sub ListBox1\_DataSourceChanged1(ByVal sender As Object, ByVal e As System.EventArgs) Handles tsbCancel.Click

ListBox1.DataSource = Nothing

ListBox1.Items.Clear()

End Sub

'Private Sub ListBox1\_DataSourceChanged2(ByVal sender As Object, ByVal e As System.EventArgs) Handles tsbReset.Click

' ListBox1.DataSource = Nothing

' ListBox1.Items.Clear()

' Load\_Record()

'End Sub

Private Sub ListBox1\_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs) Handles ListBox1.DoubleClick

' Me.txtEmp.Tag = 0

Load\_Record()

End Sub

Private Sub browsePic\_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles browsePic.LinkClicked

Dim OpenFile As New OpenFileDialog()

Try

With OpenFile

.FileName = ""

.Title = "Select Photo"

.Filter = "Image files: (\*.jpg)|\*.jpg|(\*.jpeg)|\*.jpeg|(\*.png)|\*.png|(\*.Gif)|\*.Gif|(\*.bmp)|\*.bmp| All Files (\*.\*)|\*.\*"

If .ShowDialog = Windows.Forms.DialogResult.OK Then

Me.userPic.Image = System.Drawing.Image.FromFile(.FileName)

End If

End With

Catch ex As Exception

MsgBox(ex.Message(), MsgBoxStyle.Critical, "Error")

End Try

End Sub

Private Sub removeMyPic\_LinkClicked(ByVal sender As System.Object, ByVal e As System.Windows.Forms.LinkLabelLinkClickedEventArgs) Handles removeMyPic.LinkClicked

Me.userPic.Image = System.Drawing.Image.FromFile(My.Application.Info.DirectoryPath & "\Image\mypic.jpg")

End Sub

Private Sub userPic\_MouseHover(ByVal sender As Object, ByVal e As System.EventArgs) Handles userPic.MouseHover

End Sub

End Class

Imports System.Data.OleDb

Imports System.IO

Public Class frmViewer

Private Sub frmViewer\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Load\_forSearchItem()

Load\_forSearchTools()

Clear\_Fields()

End Sub

Private Sub Load\_forSearchItem()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandText = "SELECT itemNum from tblItemList"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.txtSearch.DataSource = dt

.txtSearch.DisplayMember = "itemNum"

End With

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub Button1\_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

If txtSearch.Text = "" Then

MsgBox("Item Number Is empty")

Else

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim dr As OleDbDataReader

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "SELECT \* from tblItemlist where itemNum = '" & txtSearch.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

INumber.Text = dr("itemNum")

IName.Text = dr("itemName")

IDescription.Text = dr("iDescript")

totalQuantity.Text = dr("stockLevel")

End While

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End If

End Sub

Private Sub Clear\_Fields()

txtSearch.Text = ""

INumber.Text = "--------"

IName.Text = "--------"

IDescription.Text = "--------"

totalQuantity.Text = "--------"

toolsNumber.Text = ""

toolsNum.Text = "--------"

toolsName.Text = "--------"

toolsDescript.Text = "--------"

toolsTotal.Text = "--------"

End Sub

Private Sub Load\_forSearchTools()

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDb.OleDbCommand

Dim da As New OleDb.OleDbDataAdapter

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandText = "SELECT tPartNum from tblToolslist"

da.SelectCommand = cmd

da.Fill(dt)

With Me

.toolsNumber.DataSource = dt

.toolsNumber.DisplayMember = "tPartNum"

End With

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End Sub

Private Sub btnCount\_Click(sender As System.Object, e As System.EventArgs) Handles btnCount.Click

If txtSearch.Text = "" Then

MsgBox("Item Number Is empty")

Else

Dim cn As New OleDb.OleDbConnection

Dim cmd As New OleDbCommand

Dim dr As OleDbDataReader

Dim dt As New DataTable

Try

cn = New OleDb.OleDbConnection(Get\_Constring)

cn.Open()

cmd.Connection = cn

cmd.CommandType = CommandType.Text

cmd.CommandText = "SELECT \* from tblToolslist where tPartNum = '" & toolsNumber.Text & "'"

dr = cmd.ExecuteReader

While dr.Read

toolsNum.Text = dr("tPartNum")

toolsName.Text = dr("tPartName")

toolsDescript.Text = dr("tDescript")

toolsTotal.Text = dr("tstockLevel")

End While

Catch ex As Exception

MsgBox(ex.ToString)

Finally

cn.Close()

End Try

End If

End Sub

End Class

Module tempFile

Public user() As String

'Public stockIn() As String

'Public stockOut() As String

Public logInUser As String = "user"

Public tagItem As String

Public rDate As String = Format(Now, "dd/MM/yyy").ToString

Public rTime As String = Now.ToLongTimeString.ToString()

Public requestCreated As String = rDate + " " + rTime

Public logInBy As String = "" 'LogInBy is also a logInUser same also as Storekeeper

End Module