



Project: Application for Supply Chain management (SCM)

Source Code

Course: Internet Application Development (IAD)

Name: Muhammad Musa

Roll number: 0-3-31-036-2022

Instructor: Mr. Irfan Hameed

Pakistan Institute of Engineering & Applied Sciences (PIEAS)

I am Adding .aspx and .vb file of all pages

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin.aspx.vb" Inherits="Admin"
%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head runat="server">
```

```
    <title>Admin Panel</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <h2>Admin Panel</h2>
```

```
        <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />
```

```
        <asp:RadioButtonList ID="rblOptions" runat="server" AutoPostBack="True"
```

```
            OnSelectedIndexChanged="rblOptions_SelectedIndexChanged">
```

```
            <asp:ListItem>See Customers</asp:ListItem>
```

```
            <asp:ListItem>See Suppliers</asp:ListItem>
```

```
            <asp:ListItem>See Products</asp:ListItem>
```

```
        </asp:RadioButtonList>
```

```
    </form>
```

```
</body>
```

```
</html>
```

Imports System

Partial Class Admin

Inherits System.Web.UI.Page

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

' Role-Based Access & Session Management

If Session("UserID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse
Session("RoleName").ToString() <> "Admin" Then

 Response.Redirect("Login.aspx")

End If

End Sub

Protected Sub rblOptions_SelectedIndexChanged(sender As Object, e As EventArgs)

' Input Validation & Navigation

Select Case rblOptions.SelectedValue

 Case "See Customers"

 Response.Redirect("Admin_Cust.aspx")

 Case "See Suppliers"

 Response.Redirect("Admin_Supp.aspx")

 Case "See Products"

 Response.Redirect("Admin_Prod.aspx")

 Case Else

 lblMessage.Text = "Invalid selection."

End Select

End Sub

End Class

.....
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin_Cust.aspx.vb"
Inherits="Admin_Cust" %>

<!DOCTYPE html>

<html>

<head runat="server">

```
<title>Manage Customers</title>

</head>

<body>

    <form id="form1" runat="server">

        <h2>Manage Customers</h2>

        <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />

        <asp:Button ID="btnBack" runat="server" Text="Back to Admin Panel"
        OnClick="btnBack_Click" /><br /><br />

        <asp:GridView ID="gvCustomers" runat="server" AutoGenerateColumns="False"
        DataKeyNames="CUSTOMER_ID">

            <Columns>

                <asp:BoundField DataField="CUSTOMER_ID" HeaderText="ID" ReadOnly="True" />

                <asp:BoundField DataField="CUSTOMER_NAME" HeaderText="Name" />

                <asp:BoundField DataField="CUSTOMER_EMAIL" HeaderText="Email" />

                <asp:TemplateField HeaderText="Actions">

                    <ItemTemplate>

                        <asp:Button ID="btnDelete" runat="server" Text="Delete"
                        CommandName="DeleteCustomer"

                            CommandArgument='<%# Eval("CUSTOMER_ID") %>'

                            OnClientClick="return confirm('Are you sure you want to delete this customer?');"
                        />

                    </ItemTemplate>

                </asp:TemplateField>

            </Columns>

            <EmptyDataTemplate>

                <asp:Label ID="lblNoData" runat="server" Text="No customers found." ForeColor="Red" />

            </EmptyDataTemplate>

        </asp:GridView>

    </form>
```

</body>

</html>

Imports System.Data

Imports System.Data.SqlClient

Partial Class Admin_Cust

 Inherits System.Web.UI.Page

 Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial Catalog=SCM;Integrated Security=True"

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

 ' Role-Based Access & Session Management

 If Session("UserID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Admin" Then

 Response.Redirect("default/login.aspx")

 End If

 If Not IsPostBack Then

 LoadCustomers()

 End If

 End Sub

 Private Sub LoadCustomers()

 lblMessage.Text = ""

 Using conn As New SqlConnection(ConnStr)

 conn.Open()

 ' SQL Injection Prevention

```

Dim sql As String = "SELECT c.CUSTOMER_ID, c.CUSTOMER_NAME, c.CUSTOMER_EMAIL " &
    "FROM CUSTOMER_T c " &
    "LEFT JOIN USER_T u ON c.CUSTOMER_ID = u.USER_ID " &
    "LEFT JOIN ROLE_T r ON u.ROLE_ID = r.ROLE_ID " &
    "WHERE r.ROLE_NAME = @role OR r.ROLE_NAME IS NULL"

Using cmd As New SqlCommand(sql, conn)

    cmd.Parameters.AddWithValue("@role", "Customer")

    Using da As New SqlDataAdapter(cmd)

        Dim dt As New DataTable()

        Try

            da.Fill(dt)

            If dt.Rows.Count = 0 Then

                lblMessage.Text = "No customers found in the database."

                lblMessage.ForeColor = Drawing.Color.Red

            Else

                lblMessage.Text = "Found " & dt.Rows.Count & " customer(s)."

                lblMessage.ForeColor = Drawing.Color.Green

            End If

            gvCustomers.DataSource = dt

            gvCustomers.DataBind()

        Catch ex As Exception

            lblMessage.Text = "Error loading customers: " & ex.Message

            lblMessage.ForeColor = Drawing.Color.Red

        End Try

    End Using

End Using

End Using

```

End Sub

Protected Sub gvCustomers_RowCommand(sender As Object, e As
GridViewCommandEventArgs) Handles gvCustomers.RowCommand

 If e.CommandName = "DeleteCustomer" Then

 Dim customerId As Integer

 If Not Integer.TryParse(e.CommandArgument.ToString(), customerId) Then

 lblMessage.Text = "Invalid customer ID."

 lblMessage.ForeColor = Drawing.Color.Red

 Return

 End If

 ' Error Handling

 Using conn As New SqlConnection(ConnStr)

 conn.Open()

 Dim trans = conn.BeginTransaction()

 Try

 ' Delete from USER_T first

 Dim sqlUser As String = "DELETE FROM USER_T WHERE USER_ID = @id AND ROLE_ID =
(SELECT ROLE_ID FROM ROLE_T WHERE ROLE_NAME = @role)"

 Using cmdUser As New SqlCommand(sqlUser, conn, trans)

 cmdUser.Parameters.AddWithValue("@id", customerId)

 cmdUser.Parameters.AddWithValue("@role", "Customer")

 cmdUser.ExecuteNonQuery()

 End Using

 ' Delete from CUSTOMER_T

 Dim sql As String = "DELETE FROM CUSTOMER_T WHERE CUSTOMER_ID = @id"

```

Using cmd As New SqlCommand(sql, conn, trans)

    cmd.Parameters.AddWithValue("@id", customerId)

    Dim rowsAffected = cmd.ExecuteNonQuery()

    If rowsAffected > 0 Then

        lblMessage.Text = "Customer deleted successfully."

        lblMessage.ForeColor = Drawing.Color.Green

    Else

        lblMessage.Text = "Customer not found."

        lblMessage.ForeColor = Drawing.Color.Red

    End If

End Using

    trans.Commit()

Catch ex As Exception

    trans.Rollback()

    lblMessage.Text = "Error deleting customer: " & ex.Message

    lblMessage.ForeColor = Drawing.Color.Red

End Try

End Using

LoadCustomers()

End If

End Sub

Protected Sub btnBack_Click(sender As Object, e As EventArgs)

    Response.Redirect("Admin.aspx")

End Sub

End Class

```

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin_d.aspx.vb"
Inherits="Admin_d" %>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
    <title>Admin Registration</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <h2>Register Admin</h2>
```

```
        <table>
```

```
            <tr>
```

```
                <td>Name:</td>
```

```
                <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
```

```
            </tr>
```

```
            <tr>
```

```
                <td>Phone:</td>
```

```
                <td><asp:TextBox ID="txtPhone" runat="server"></asp:TextBox></td>
```

```
            </tr>
```

```
            <tr>
```

```
                <td>Email:</td>
```

```
                <td><asp:TextBox ID="txtEmail" runat="server"></asp:TextBox></td>
```

```
            </tr>
```

```
            <tr>
```

```
                <td>Password:</td>
```

```
        <td><asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox></td>

    </tr>

    <tr>

        <td>Address:</td>

        <td><asp:TextBox ID="txtAddress" runat="server"
TextMode="MultiLine"></asp:TextBox></td>

    </tr>

    <tr>

        <td colspan="2">

            <asp:Button ID="btnRegister" runat="server" Text="Register Admin"
OnClick="btnRegister_Click" />

        </td>

    </tr>

</table>

<asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>

</form>

</body>

</html>
```

Imports System.Data.SqlClient

Imports System.Security.Cryptography

Imports System.Text

Partial Class Admin_d

Inherits System.Web.UI.Page

Protected Sub btnRegister_Click(sender As Object, e As EventArgs)

Dim adminID As Integer = 1

Dim name As String = txtName.Text.Trim()

Dim phone As String = txtPhone.Text.Trim()

Dim email As String = txtEmail.Text.Trim()

Dim password As String = txtPassword.Text.Trim()

Dim address As String = txtAddress.Text.Trim()

Dim salt As String = GenerateSalt()

Dim hash As String = HashPassword(password, salt)

Dim conStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

Using con As New SqlConnection(conStr)

Dim query As String = "INSERT INTO ADMIN_T (ADMIN_ID, ADMIN_NAME, ADMIN_PHONE,
ADMIN_EMAIL, ADMIN_ADDRESS, PASSWORD_HASH, SALT) VALUES (@ID, @Name, @Phone,
@Email, @Address, @Hash, @Salt)"

Using cmd As New SqlCommand(query, con)

cmd.Parameters.AddWithValue("@ID", adminID)

cmd.Parameters.AddWithValue("@Name", name)

cmd.Parameters.AddWithValue("@Phone", phone)

cmd.Parameters.AddWithValue("@Email", email)

cmd.Parameters.AddWithValue("@Address", address)

cmd.Parameters.AddWithValue("@Hash", hash)

cmd.Parameters.AddWithValue("@Salt", salt)

Try

con.Open()

cmd.ExecuteNonQuery()

```
        lblMessage.Text = "Admin registered successfully."

        lblMessage.ForeColor = Drawing.Color.Green

    Catch ex As Exception

        lblMessage.Text = "Error: " & ex.Message

    End Try

End Using

End Using

End Sub
```

```
Private Function GenerateSalt() As String

    Dim rng As New RNGCryptoServiceProvider()

    Dim buff(31) As Byte

    rng.GetBytes(buff)

    Return Convert.ToBase64String(buff)

End Function
```

```
' Hash = SHA256( salt + password ), base64-encoded

Private Function HashPassword(pw As String, salt As String) As String

    Using sha As SHA256 = SHA256.Create()

        Dim combined = Encoding.UTF8.GetBytes(salt & pw)

        Dim sum = sha.ComputeHash(combined)

        Return Convert.ToBase64String(sum)

    End Using

End Function

End Class
```

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin_Prod.aspx.vb"
Inherits="Admin_Prod" %>
```

```
<!DOCTYPE html>

<html>

<head runat="server">

    <title>Manage Products</title>

</head>

<body>

    <form id="form1" runat="server">

        <h2>Manage Products</h2>

        <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />

        <asp:Button ID="btnBack" runat="server" Text="Back to Admin Panel"
OnClick="btnBack_Click" /><br /><br />

        <asp:GridView ID="gvProducts" runat="server" AutoGenerateColumns="False"
DataKeyNames="PRODUCT_ID">

            <Columns>

                <asp:BoundField DataField="PRODUCT_ID" HeaderText="ID" ReadOnly="True" />

                <asp:BoundField DataField="PRODUCT_NAME" HeaderText="Name" />

                <asp:BoundField DataField="PRODUCT_QUANTITY" HeaderText="Quantity" />

                <asp:BoundField DataField="SELLING_COST" HeaderText="Price"
DataFormatString="{0:C}" />

                <asp:TemplateField HeaderText="Actions">

                    <ItemTemplate>

                        <asp:Button ID="btnDelete" runat="server" Text="Delete"
CommandName="DeleteProduct"

                            CommandArgument='<%# Eval("PRODUCT_ID") %>'

                            OnClientClick="return confirm('Are you sure you want to delete this product?');" />

                    </ItemTemplate>

                </asp:TemplateField>

            </Columns>

            <EmptyDataTemplate>
```

```
        <asp:Label ID="lblNoData" runat="server" Text="No products found." ForeColor="Red" />
    </EmptyDataTemplate>
</asp:GridView>
</form>
</body>
</html>
```

Imports System.Data

Imports System.Data.SqlClient

Partial Class Admin_Prod

Inherits System.Web.UI.Page

Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

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

' Role-Based Access & Session Management

If Session("UserID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse
Session("RoleName").ToString() <> "Admin" Then

Response.Redirect("default/login.aspx")

End If

If Not IsPostBack Then

LoadProducts()

End If

End Sub

```

Private Sub LoadProducts()

    lblMessage.Text = ""

    Using conn As New SqlConnection(ConnStr)

        conn.Open()

        ' SQL Injection Prevention

        Dim sql As String = "SELECT PRODUCT_ID, PRODUCT_NAME, PRODUCT_QUANTITY,
SELLING_COST FROM PRODUCT_T"

        Using cmd As New SqlCommand(sql, conn)

            Using da As New SqlDataAdapter(cmd)

                Dim dt As New DataTable()

                Try

                    da.Fill(dt)

                    If dt.Rows.Count = 0 Then

                        lblMessage.Text = "No products found in the database."

                        lblMessage.ForeColor = Drawing.Color.Red

                    Else

                        lblMessage.Text = "Found " & dt.Rows.Count & " product(s)."

                        lblMessage.ForeColor = Drawing.Color.Green

                    End If

                    gvProducts.DataSource = dt

                    gvProducts.DataBind()

                Catch ex As Exception

                    lblMessage.Text = "Error loading products: " & ex.Message

                    lblMessage.ForeColor = Drawing.Color.Red

                End Try

            End Using

        End Using

    End Using

End Using

```

End Sub

Protected Sub gvProducts_RowCommand(sender As Object, e As GridViewCommandEventArgs)
Handles gvProducts.RowCommand

```
If e.CommandName = "DeleteProduct" Then
```

Dim productId As Integer

```
If Not Integer.TryParse(e.CommandArgument.ToString(), productId) Then
```

```
lblMessage.Text = "Invalid product ID."
```

```
lblMessage.ForeColor = Drawing.Color.Red
```

Return

End If

' Error Handling

Using conn As New SqlConnection(ConnStr)

```
conn.Open()
```

```
Dim trans = conn.BeginTransaction()
```

Try

```
' Delete from PRODUCT_T
```

```
Dim sql As String = "DELETE FROM PRODUCT_T WHERE PRODUCT_ID = @id"
```

Using cmd As New SqlCommand(sql, conn, trans)

```
cmd.Parameters.AddWithValue("@id", productId)
```

```
Dim rowsAffected = cmd.ExecuteNonQuery()
```

If rowsAffected > 0 Then

```
lblMessage.Text = "Product deleted successfully."
```

```
lblMessage.ForeColor = Drawing.Color.Green
```

Else

```
lblMessage.Text = "Product not found."
```

```
lblMessage.ForeColor = Drawing.Color.Red
```

```

        End If
    End Using
    trans.Commit()
Catch ex As Exception
    trans.Rollback()
    lblMessage.Text = "Error deleting product: " & ex.Message
    lblMessage.ForeColor = Drawing.Color.Red
End Try
End Using
LoadProducts()
End If
End Sub

```

```

Protected Sub btnBack_Click(sender As Object, e As EventArgs)
    Response.Redirect("Admin.aspx")
End Sub
End Class

```

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Admin_Supp.aspx.vb"
Inherits="Admin_Supp" %>

```

```

<!DOCTYPE html>

```

```

<html>

```

```

<head runat="server">

```

```

    <title>Manage Suppliers</title>

```

```

</head>

```

```

<body>

```

```

    <form id="form1" runat="server">

```

```

<h2>Manage Suppliers</h2>

<asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />

<asp:Button ID="btnBack" runat="server" Text="Back to Admin Panel"
OnClick="btnBack_Click" /><br /><br />

<asp:GridView ID="gvSuppliers" runat="server" AutoGenerateColumns="False"
DataKeyNames="SUPPLIER_ID">

    <Columns>

        <asp:BoundField DataField="SUPPLIER_ID" HeaderText="ID" ReadOnly="True" />

        <asp:BoundField DataField="SUPPLIER_FIRST_NAME" HeaderText="First Name" />

        <asp:BoundField DataField="SUPPLIER_EMAIL" HeaderText="Email" />

        <asp:TemplateField HeaderText="Actions">

            <ItemTemplate>

                <asp:Button ID="btnDelete" runat="server" Text="Delete"
CommandName="DeleteSupplier"

                    CommandArgument='<%# Eval("SUPPLIER_ID") %>'

                    OnClientClick="return confirm('Are you sure you want to delete this supplier?');"
/>

            </ItemTemplate>

        </asp:TemplateField>

    </Columns>

    <EmptyDataTemplate>

        <asp:Label ID="lblNoData" runat="server" Text="No suppliers found." ForeColor="Red" />

    </EmptyDataTemplate>

</asp:GridView>

</form>

</body>

</html>

```

Imports System.Data

Imports System.Data.SqlClient

Partial Class Admin_Supp

Inherits System.Web.UI.Page

Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial Catalog=SCM;Integrated Security=True"

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

' Role-Based Access & Session Management

If Session("UserID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Admin" Then

Response.Redirect("default/login.aspx")

End If

If Not IsPostBack Then

LoadSuppliers()

End If

End Sub

Private Sub LoadSuppliers()

lblMessage.Text = ""

Using conn As New SqlConnection(ConnStr)

conn.Open()

' SQL Injection Prevention

Dim sql As String = "SELECT s.SUPPLIER_ID, s.SUPPLIER_FIRST_NAME, s.SUPPLIER_EMAIL "

&

"FROM SUPPLIER_T s " &

"LEFT JOIN USER_T u ON s.SUPPLIER_ID = u.USER_ID " &

```
"LEFT JOIN ROLE_T r ON u.ROLE_ID = r.ROLE_ID " &  
"WHERE r.ROLE_NAME = @role OR r.ROLE_NAME IS NULL"
```

```
Using cmd As New SqlCommand(sql, conn)
```

```
cmd.Parameters.AddWithValue("@role", "Supplier")
```

```
Using da As New SqlDataAdapter(cmd)
```

```
Dim dt As New DataTable()
```

```
Try
```

```
da.Fill(dt)
```

```
If dt.Rows.Count = 0 Then
```

```
    lblMessage.Text = "No suppliers found in the database."
```

```
    lblMessage.ForeColor = Drawing.Color.Red
```

```
Else
```

```
    lblMessage.Text = "Found " & dt.Rows.Count & " supplier(s)."
```

```
    lblMessage.ForeColor = Drawing.Color.Green
```

```
End If
```

```
gvSuppliers.DataSource = dt
```

```
gvSuppliers.DataBind()
```

```
Catch ex As Exception
```

```
    lblMessage.Text = "Error loading suppliers: " & ex.Message
```

```
    lblMessage.ForeColor = Drawing.Color.Red
```

```
End Try
```

```
End Using
```

```
End Using
```

```
End Using
```

```
End Sub
```

```
Protected Sub gvSuppliers_RowCommand(sender As Object, e As GridViewCommandEventArgs)  
Handles gvSuppliers.RowCommand
```

```
If e.CommandName = "DeleteSupplier" Then
```

```
    Dim supplierId As Integer
```

```
    If Not Integer.TryParse(e.CommandArgument.ToString(), supplierId) Then
```

```
        lblMessage.Text = "Invalid supplier ID."
```

```
        lblMessage.ForeColor = Drawing.Color.Red
```

```
    Return
```

```
End If
```

```
' Error Handling
```

```
Using conn As New SqlConnection(ConnStr)
```

```
    conn.Open()
```

```
    Dim trans = conn.BeginTransaction()
```

```
    Try
```

```
        ' Delete from USER_T first
```

```
        Dim sqlUser As String = "DELETE FROM USER_T WHERE USER_ID = @id AND ROLE_ID =  
(SELECT ROLE_ID FROM ROLE_T WHERE ROLE_NAME = @role)"
```

```
        Using cmdUser As New SqlCommand(sqlUser, conn, trans)
```

```
            cmdUser.Parameters.AddWithValue("@id", supplierId)
```

```
            cmdUser.Parameters.AddWithValue("@role", "Supplier")
```

```
            cmdUser.ExecuteNonQuery()
```

```
        End Using
```

```
        ' Delete from SUPPLIER_T
```

```
        Dim sql As String = "DELETE FROM SUPPLIER_T WHERE SUPPLIER_ID = @id"
```

```
        Using cmd As New SqlCommand(sql, conn, trans)
```

```
            cmd.Parameters.AddWithValue("@id", supplierId)
```

```
            Dim rowsAffected = cmd.ExecuteNonQuery()
```

```
            If rowsAffected > 0 Then
```

```

        lblMessage.Text = "Supplier deleted successfully."

        lblMessage.ForeColor = Drawing.Color.Green

    Else

        lblMessage.Text = "Supplier not found."

        lblMessage.ForeColor = Drawing.Color.Red

    End If

End Using

    trans.Commit()

Catch ex As Exception

    trans.Rollback()

    lblMessage.Text = "Error deleting supplier: " & ex.Message

    lblMessage.ForeColor = Drawing.Color.Red

End Try

End Using

LoadSuppliers()

End If

End Sub

Protected Sub btnBack_Click(sender As Object, e As EventArgs)

    Response.Redirect("Admin.aspx")

End Sub

End Class

.....

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Customer.aspx.vb"
Inherits="Customer" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

```

```
<head runat="server">

    <title>Customer Dashboard</title>

</head>

<body>

    <form id="form1" runat="server">

        <div>

            <h2>Customer Dashboard</h2>

            <asp:Label ID="lblWelcome" runat="server" Font-Bold="True"></asp:Label><br />

            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label><br />


            <asp:Button ID="btnUpdateProfile" runat="server" Text="Update Profile"
OnClick="btnUpdateProfile_Click" />

            <asp:Button ID="btnViewProducts" runat="server" Text="View Products"
OnClick="btnViewProducts_Click" />

            <asp:Button ID="btnPlaceOrder" runat="server" Text="Place Order"
OnClick="btnPlaceOrder_Click" />

            <asp:Button ID="btnViewOrders" runat="server" Text="View Orders"
OnClick="btnViewOrders_Click" />

        </div>

    </form>

</body>
```

Imports System.Data

Imports System.Data.SqlClient

Partial Class Customer

Inherits System.Web.UI.Page

```
Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial  
Catalog=SCM;Integrated Security=True"
```

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

```
    If Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Customer" Then
```

```
        Response.Redirect("Login.aspx")
```

```
    End If
```

```
    If Not IsPostBack Then
```

```
        LoadWelcomeMessage()
```

```
    End If
```

```
End Sub
```

```
Private Sub LoadWelcomeMessage()
```

```
    Using con As New SqlConnection(connString)
```

```
        con.Open()
```

```
        Dim sql As String = "SELECT CUSTOMER_NAME FROM CUSTOMER_T WHERE CUSTOMER_ID  
= @id"
```

```
        Using cmd As New SqlCommand(sql, con)
```

```
            cmd.Parameters.AddWithValue("@id", Session("CUSTOMER_ID"))
```

```
            Using dr As SqlDataReader = cmd.ExecuteReader()
```

```
                If dr.Read() Then
```

```
                    lblWelcome.Text = "Welcome, " & dr("CUSTOMER_NAME").ToString()
```

```
                End If
```

```
            End Using
```

```
        End Using
```

```
    End Using
```

```
End Sub
```

```
Protected Sub btnUpdateProfile_Click(sender As Object, e As EventArgs) Handles  
btnUpdateProfile.Click
```

```
    Response.Redirect("UpdateProfile.aspx")
```

```
End Sub
```

```
Protected Sub btnViewProducts_Click(sender As Object, e As EventArgs) Handles  
btnViewProducts.Click
```

```
    Response.Redirect("ViewProducts.aspx")
```

```
End Sub
```

```
Protected Sub btnPlaceOrder_Click(sender As Object, e As EventArgs) Handles  
btnPlaceOrder.Click
```

```
    Response.Redirect("PlaceOrder.aspx")
```

```
End Sub
```

```
Protected Sub btnViewOrders_Click(sender As Object, e As EventArgs) Handles  
btnViewOrders.Click
```

```
    Response.Redirect("ViewOrders.aspx")
```

```
End Sub
```

```
End Class
```

.....

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="PlaceOrder.aspx.vb"  
Inherits="PlaceOrder" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>Place Order</title>

</head>

<body>

    <form id="form1" runat="server">

        <div>

            <h2>Place Order</h2>

            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label><br />

            <asp:Button ID="btnBackToDashboard" runat="server" Text="Back to Dashboard"
OnClick="btnBackToDashboard_Click" /><br />

            <p>Select Product:</p>

            <asp:DropDownList ID="ddlProducts" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlProducts_SelectedIndexChanged"></asp:DropDownList><br />

            <p>Select Quantity:</p>

            <asp:DropDownList ID="ddlQuantity" runat="server"></asp:DropDownList><br />

            <asp:Button ID="btnAddProduct" runat="server" Text="Add to Order"
OnClick="btnAddProduct_Click" /><br />

            <p>Selected Products:</p>

            <asp:ListBox ID="lstSelectedProducts" runat="server" Width="300px"></asp:ListBox><br />

            <p>Payment Method:</p>

            <asp:DropDownList ID="ddlPaymentMethod" runat="server">

                <asp:ListItem Text="Select Method" Value="0"></asp:ListItem>

                <asp:ListItem Text="Card" Value="Card"></asp:ListItem>

            </asp:DropDownList>

        </div>

    </form>

</body>

</html>
```

```

        <asp:ListItem Text="PayPal" Value="PayPal"></asp:ListItem>

        <asp:ListItem Text="Cash" Value="Cash"></asp:ListItem>

    </asp:DropDownList><br />

    <asp:Button ID="btnPlaceOrder" runat="server" Text="Place Order"
OnClick="btnPlaceOrder_Click" /><br />

    <asp:Label ID="lblTotalBill" runat="server" ForeColor="Blue"></asp:Label>

</div>

</form>

</body>

</html>

```

Imports System.Data

Imports System.Data.SqlClient

Imports System.Text.RegularExpressions

Partial Class PlaceOrder

Inherits System.Web.UI.Page

```

    Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

```

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

If Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Customer" Then

 Response.Redirect("Login.aspx")

End If

If Not IsPostBack Then

LoadProducts()

End If

End Sub

Private Sub LoadProducts()

Using con As New SqlConnection(connString)

con.Open()

Dim sql As String = "SELECT PRODUCT_ID, PRODUCT_NAME, PRODUCT_QUANTITY FROM
PRODUCT_T WHERE PRODUCT_QUANTITY > 0"

Using cmd As New SqlCommand(sql, con)

Using da As New SqlDataAdapter(cmd)

Dim dt As New DataTable()

da.Fill(dt)

ddlProducts.DataSource = dt

ddlProducts.DataTextField = "PRODUCT_NAME"

ddlProducts.DataValueField = "PRODUCT_ID"

ddlProducts.DataBind()

ddlProducts.Items.Insert(0, New ListItem("Select Product", "0"))

End Using

End Using

End Using

LoadQuantities()

End Sub

Protected Sub ddlProducts_SelectedIndexChanged(sender As Object, e As EventArgs)

LoadQuantities()

End Sub

```

Private Sub LoadQuantities()

    ddlQuantity.Items.Clear()

    If ddlProducts.SelectedValue <> "0" Then

        Using con As New SqlConnection(connString)

            con.Open()

            Dim sql As String = "SELECT PRODUCT_QUANTITY FROM PRODUCT_T WHERE
PRODUCT_ID = @pid"

            Using cmd As New SqlCommand(sql, con)

                cmd.Parameters.AddWithValue("@pid", ddlProducts.SelectedValue)

                Dim qty As Integer = Convert.ToInt32(cmd.ExecuteScalar())

                For i As Integer = 1 To qty

                    ddlQuantity.Items.Add(New ListItem(i.ToString(), i.ToString()))

                Next

            End Using

        End Using

    End If

    ddlQuantity.Items.Insert(0, New ListItem("Select Quantity", "0"))

End Sub

```

```

Protected Sub btnAddProduct_Click(sender As Object, e As EventArgs)

    If ddlProducts.SelectedValue = "0" OrElse ddlQuantity.SelectedValue = "0" Then

        lblMessage.Text = "Please select a product and quantity."

        Return

    End If

    Dim itemText As String = ddlProducts.SelectedItem.Text & " (ID: " & ddlProducts.SelectedValue
& ", Qty: " & ddlQuantity.SelectedValue & ")"

```

```
lstSelectedProducts.Items.Add(New ListItem(itemText, ddlProducts.SelectedValue & ":" &
ddlQuantity.SelectedValue))
```

```
lblMessage.Text = ""
```

```
ddlProducts.SelectedIndex = 0
```

```
LoadQuantities()
```

```
End Sub
```

```
Protected Sub btnPlaceOrder_Click(sender As Object, e As EventArgs)
```

```
lblTotalBill.Text = ""
```

```
If Not ValidateOrder() Then Exit Sub
```

```
Dim paymentMethod As String = ddlPaymentMethod.SelectedItem.Text
```

```
Dim totalBill As Decimal = 0
```

```
Dim orderList As New List(Of Tuple(Of Integer, Integer)) ' ProductID, Quantity
```

```
Using con As New SqlConnection(connString)
```

```
con.Open()
```

```
Using tran = con.BeginTransaction()
```

```
Try
```

```
' Validate selected products
```

```
For Each item As ListItem In lstSelectedProducts.Items
```

```
Dim parts() As String = item.Value.Split(":")
```

```
Dim productId As Integer = Integer.Parse(parts(0))
```

```
Dim quantity As Integer = Integer.Parse(parts(1))
```

```
Dim sqlProduct As String = "SELECT SELLING_COST, PRODUCT_QUANTITY FROM
PRODUCT_T WHERE PRODUCT_ID = @pid"
```

```
Using cmd As New SqlCommand(sqlProduct, con, tran)
```

```

cmd.Parameters.AddWithValue("@pid", productId)

Using dr As SqlDataReader = cmd.ExecuteReader()

    If dr.Read() Then

        Dim availableQty As Integer = CInt(dr("PRODUCT_QUANTITY"))

        Dim unitPrice As Decimal = CDec(dr("SELLING_COST"))

        If quantity > availableQty Then

            lblMessage.Text = "Insufficient stock for Product ID " & productId & " "

            Return

        End If

        totalBill += unitPrice * quantity

        orderList.Add(New Tuple(Of Integer, Integer)(productId, quantity))

    Else

        lblMessage.Text = "Product ID " & productId & " not found."

        Return

    End If

End Using

End Using

Next

' Create order with payment method

Dim orderId As Integer = GetNextId("ORDER_T", "ORDER_ID", con, tran)

Dim sqlOrder As String = "INSERT INTO ORDER_T (ORDER_ID, ORDER_DATE,
CUSTOMER_ID, PAYMENT_METHOD) " &

    "VALUES (@id, GETDATE(), @cid, @pm)"

Using cmd As New SqlCommand(sqlOrder, con, tran)

    cmd.Parameters.AddWithValue("@id", orderId)

    cmd.Parameters.AddWithValue("@cid", Session("CUSTOMER_ID"))

    cmd.Parameters.AddWithValue("@pm", paymentMethod)

```

```

        cmd.ExecuteNonQuery()

    End Using

    ' Add order lines and update product quantities
    For Each orderItem In orderList

        Dim productId As Integer = orderItem.Item1

        Dim quantity As Integer = orderItem.Item2

        Dim sqlLine As String = "INSERT INTO ORDER_LINE_T (ORDER_ID, PRODUCT_ID,
ORDERED_QUANTITY) " &
            "VALUES (@oid, @pid, @qty)"

        Using cmd As New SqlCommand(sqlLine, con, tran)

            cmd.Parameters.AddWithValue("@oid", orderId)

            cmd.Parameters.AddWithValue("@pid", productId)

            cmd.Parameters.AddWithValue("@qty", quantity)

            cmd.ExecuteNonQuery()

        End Using

        Dim sqlUpdateQty As String = "UPDATE PRODUCT_T SET PRODUCT_QUANTITY =
PRODUCT_QUANTITY - @qty WHERE PRODUCT_ID = @pid"

        Using cmd As New SqlCommand(sqlUpdateQty, con, tran)

            cmd.Parameters.AddWithValue("@pid", productId)

            cmd.Parameters.AddWithValue("@qty", quantity)

            cmd.ExecuteNonQuery()

        End Using

    Next

    tran.Commit()

```

```

        lblMessage.ForeColor = Drawing.Color.Green

        lblMessage.Text = "Order placed successfully."

        lblTotalBill.Text = "Total Bill: " & totalBill.ToString("C")

        lstSelectedProducts.Items.Clear()

        ddlPaymentMethod.SelectedIndex = 0

        LoadProducts()

    Catch ex As Exception

        tran.Rollback()

        lblMessage.ForeColor = Drawing.Color.Red

        lblMessage.Text = "Error: " & ex.Message

    End Try

End Using

End Using

End Sub

```

```

Protected Sub btnBackToDashboard_Click(sender As Object, e As EventArgs)

    Response.Redirect("Customer.aspx")

End Sub

```

```

Private Function GetNextId(tableName As String, idCol As String, con As SqlConnection, tran As
SqlConnection) As Integer

    Dim sql As String = String.Format("SELECT ISNULL(MAX({0}), 0) + 1 FROM {1}", idCol,
tableName)

    Using cmd As New SqlCommand(sql, con, tran)

        Return Convert.ToInt32(cmd.ExecuteScalar())

    End Using

End Function

```

```
Private Function ValidateOrder() As Boolean
```

```
    If lstSelectedProducts.Items.Count = 0 Then
```

```
        lblMessage.Text = "Please add at least one product to the order."
```

```
        Return False
```

```
    End If
```

```
    If ddlPaymentMethod.SelectedValue = "0" Then
```

```
        lblMessage.Text = "Please select a payment method."
```

```
        Return False
```

```
    End If
```

```
    Return True
```

```
End Function
```

```
End Class
```

```
.....
```

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="UpdateProfile.aspx.vb"  
Inherits="UpdateProfile" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>Update Profile</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div>
```

<h2>Update Profile</h2>

<asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label>

<asp:Button ID="btnBackToDashboard" runat="server" Text="Back to Dashboard"
OnClick="btnBackToDashboard_Click" />

<p>Customer Name:</p>

<asp:TextBox ID="txtCustomerName" runat="server" Placeholder="Customer
Name"></asp:TextBox>

<p>Contact Number:</p>

<asp:TextBox ID="txtCustomerContact" runat="server" Placeholder="Contact
Number"></asp:TextBox>

<p>Email:</p>

<asp:TextBox ID="txtCustomerEmail" runat="server" Placeholder="Email"></asp:TextBox>

<p>City:</p>

<asp:TextBox ID="txtCustomerCity" runat="server" Placeholder="City"></asp:TextBox>

<p>State:</p>

<asp:TextBox ID="txtCustomerState" runat="server" Placeholder="State"></asp:TextBox>

<p>Postal Code:</p>

<asp:TextBox ID="txtCustomerPostalCode" runat="server" Placeholder="Postal
Code"></asp:TextBox>

<p>Street Address:</p>

<asp:TextBox ID="txtCustomerStreet" runat="server" Placeholder="Street Address"></asp:TextBox>

<asp:Button ID="btnUpdate" runat="server" Text="Update Profile" OnClick="btnUpdate_Click" />

</div>

</form>

</body>

</html>

Imports System.Data

Imports System.Data.SqlClient

Imports System.Text.RegularExpressions

Partial Class UpdateProfile

Inherits System.Web.UI.Page

Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial Catalog=SCM;Integrated Security=True"

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

If Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Customer" Then

Response.Redirect("Login.aspx")

End If

If Not IsPostBack Then

LoadProfile()

End If

End Sub

Private Sub LoadProfile()

Using con As New SqlConnection(connString)

con.Open()

Dim sql As String = "SELECT CUSTOMER_NAME, CUSTOMER_CONTACT, CUSTOMER_EMAIL,
CUSTOMER_CITY, CUSTOMER_STATE, CUSTOMER_POSTALCODE, CUSTOMER_STREET_ADDRESS "
&

"FROM CUSTOMER_T WHERE CUSTOMER_ID = @id"

Using cmd As New SqlCommand(sql, con)

cmd.Parameters.AddWithValue("@id", Session("CUSTOMER_ID"))

Using dr As SqlDataReader = cmd.ExecuteReader()

If dr.Read() Then

txtCustomerName.Text = dr("CUSTOMER_NAME").ToString()

txtCustomerContact.Text = dr("CUSTOMER_CONTACT").ToString()

txtCustomerEmail.Text = dr("CUSTOMER_EMAIL").ToString()

txtCustomerCity.Text = dr("CUSTOMER_CITY").ToString()

txtCustomerState.Text = dr("CUSTOMER_STATE").ToString()

txtCustomerPostalCode.Text = dr("CUSTOMER_POSTALCODE").ToString()

txtCustomerStreet.Text = dr("CUSTOMER_STREET_ADDRESS").ToString()

End If

End Using

End Using

End Using

End Sub

Protected Sub btnUpdate_Click(sender As Object, e As EventArgs) Handles btnUpdate.Click

If Not ValidateCustomerUpdate() Then Exit Sub

```

Using con As New SqlConnection(connString)

con.Open()

Using tran = con.BeginTransaction()

Try

    Dim sql As String = "UPDATE CUSTOMER_T SET " &

        "CUSTOMER_NAME = COALESCE(NULLIF(@name, ''), CUSTOMER_NAME), " &

        "CUSTOMER_CONTACT = COALESCE(NULLIF(@phone, ''),

CUSTOMER_CONTACT), " &

        "CUSTOMER_EMAIL = COALESCE(NULLIF(@em, ''), CUSTOMER_EMAIL), " &

        "CUSTOMER_CITY = COALESCE(NULLIF(@city, ''), CUSTOMER_CITY), " &

        "CUSTOMER_STATE = COALESCE(NULLIF(@st, ''), CUSTOMER_STATE), " &

        "CUSTOMER_POSTALCODE = COALESCE(NULLIF(@postal, ''),

CUSTOMER_POSTALCODE), " &

        "CUSTOMER_STREET_ADDRESS = COALESCE(NULLIF(@street, ''),

CUSTOMER_STREET_ADDRESS) " &

        "WHERE CUSTOMER_ID = @id"

    Using cmd As New SqlCommand(sql, con, tran)

        cmd.Parameters.AddWithValue("@id", Session("CUSTOMER_ID"))

        cmd.Parameters.AddWithValue("@name", txtCustomerName.Text.Trim())

        cmd.Parameters.AddWithValue("@phone", txtCustomerContact.Text.Trim())

        cmd.Parameters.AddWithValue("@em", txtCustomerEmail.Text.Trim())

        cmd.Parameters.AddWithValue("@city", txtCustomerCity.Text.Trim())

        cmd.Parameters.AddWithValue("@st", txtCustomerState.Text.Trim())

        cmd.Parameters.AddWithValue("@postal", txtCustomerPostalCode.Text.Trim())

        cmd.Parameters.AddWithValue("@street", txtCustomerStreet.Text.Trim())

        cmd.ExecuteNonQuery()

    End Using

```

```
        tran.Commit()

        lblMessage.ForeColor = Drawing.Color.Green

        lblMessage.Text = "Profile updated successfully."

    Catch ex As Exception

        tran.Rollback()

        lblMessage.ForeColor = Drawing.Color.Red

        lblMessage.Text = "Error: " & ex.Message

    End Try

End Using

End Using

End Sub
```

```
Protected Sub btnBackToDashboard_Click(sender As Object, e As EventArgs) Handles  
btnBackToDashboard.Click
```

```
    Response.Redirect("Customer.aspx")

End Sub
```

```
Private Function ValidateCustomerUpdate() As Boolean

    Dim cname = txtCustomerName.Text.Trim()

    If cname <> "" AndAlso (Not Regex.IsMatch(cname, "^[A-Za-z\s]+$") OrElse cname.Length > 40)
Then
        lblMessage.Text = "Customer Name must be letters/spaces only, max 40 chars."

        Return False

    End If
```

```
    Dim contact = txtCustomerContact.Text.Trim()

    If contact <> "" AndAlso Not Regex.IsMatch(contact, "^\d{11}$") Then
```

```
lblMessage.Text = "Contact Number must be exactly 11 digits."
```

```
Return False
```

```
End If
```

```
Dim email = txtCustomerEmail.Text.Trim()
```

```
If email <> "" AndAlso Not Regex.IsMatch(email, "[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$") Then
```

```
    lblMessage.Text = "Email must be in a valid format."
```

```
    Return False
```

```
End If
```

```
Dim city = txtCustomerCity.Text.Trim()
```

```
If city <> "" AndAlso (Not Regex.IsMatch(city, "[A-Za-z\s]+$") OrElse city.Length > 20) Then
```

```
    lblMessage.Text = "City must be letters/spaces only, max 20 chars."
```

```
    Return False
```

```
End If
```

```
Dim state = txtCustomerState.Text.Trim()
```

```
If state <> "" AndAlso (Not Regex.IsMatch(state, "[A-Za-z\s]+$") OrElse state.Length > 50) Then
```

```
    lblMessage.Text = "State must be letters/spaces only, max 50 chars."
```

```
    Return False
```

```
End If
```

```
Dim cPostal = txtCustomerPostalCode.Text.Trim()
```

```
If cPostal <> "" AndAlso (Not Regex.IsMatch(cPostal, "\d+$") OrElse cPostal.Length > 10) Then
```

```
    lblMessage.Text = "Postal Code must be numeric only, max 10 digits."
```

```
    Return False
```

```
End If
```

```
Dim street = txtCustomerStreet.Text.Trim()

If street <> "" AndAlso street.Length > 30 Then

    lblMessage.Text = "Street Address max 30 chars."

    Return False

End If
```

```
Return True
```

```
End Function
```

```
End Class
```

```
.....

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ViewProducts.aspx.vb"
Inherits="ViewProducts" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>View Products</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div>
```

```
            <h2>View Products</h2>
```

```
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label><br />
```

```
            <asp:Button ID="btnBackToDashboard" runat="server" Text="Back to Dashboard"
OnClick="btnBackToDashboard_Click" /><br />
```

```

<asp:GridView ID="gvProducts" runat="server" AutoGenerateColumns="False">
    <Columns>
        <asp:BoundField DataField="PRODUCT_ID" HeaderText="Product ID" />
        <asp:BoundField DataField="PRODUCT_NAME" HeaderText="Product Name" />
        <asp:BoundField DataField="SELLING_COST" HeaderText="Unit Price"
DataFormatString="{0:C}" />
        <asp:BoundField DataField="PRODUCT_QUANTITY" HeaderText="Available Quantity" />
    </Columns>
</asp:GridView>
</div>
</form>
</body>
</html>

```

Imports System.Data

Imports System.Data.SqlClient

Partial Class ViewProducts

Inherits System.Web.UI.Page

```

Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

```

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

If Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Customer" Then

Response.Redirect("Login.aspx")

End If

If Not IsPostBack Then

 LoadProducts()

End If

End Sub

Private Sub LoadProducts()

 Using con As New SqlConnection(connString)

 con.Open()

 Dim sql As String = "SELECT PRODUCT_ID, PRODUCT_NAME, SELLING_COST,
PRODUCT_QUANTITY FROM PRODUCT_T WHERE PRODUCT_QUANTITY > 0"

 Using cmd As New SqlCommand(sql, con)

 Using da As New SqlDataAdapter(cmd)

 Dim dt As New DataTable()

 da.Fill(dt)

 gvProducts.DataSource = dt

 gvProducts.DataBind()

 End Using

 End Using

 End Using

End Sub

Protected Sub btnBackToDashboard_Click(sender As Object, e As EventArgs) Handles
btnBackToDashboard.Click

 Response.Redirect("Customer.aspx")

End Sub

End Class

.....

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ViewOrders.aspx.vb"
Inherits="ViewOrders" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title>View Orders</title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<h2>View Orders</h2>
```

```
<asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label><br />
```

```
<asp:Button ID="btnBackToDashboard" runat="server" Text="Back to Dashboard"
OnClick="btnBackToDashboard_Click" /><br />
```

```
<asp:GridView ID="gvOrders" runat="server" AutoGenerateColumns="False"
OnRowDataBound="gvOrders_RowDataBound">
```

```
<Columns>
```

```
<asp:BoundField DataField="ORDER_ID" HeaderText="Order ID" />
```

```
<asp:BoundField DataField="ORDER_DATE" HeaderText="Order Date"
DataFormatString="{0:d}" />
```

```
<asp:BoundField DataField="PAYMENT_METHOD" HeaderText="Payment Method" />
```

```
<asp:TemplateField HeaderText="Order Details">
```

```
<ItemTemplate>
```

```
<asp:GridView ID="gvOrderDetails" runat="server" AutoGenerateColumns="False">
```

```
<Columns>
```

```
<asp:BoundField DataField="PRODUCT_ID" HeaderText="Product ID" />
```

```

        <asp:BoundField DataField="ORDERED_QUANTITY" HeaderText="Quantity" />
        <asp:BoundField DataField="UnitPrice" HeaderText="Unit Price"
DataFormatString="{0:C}" />
        <asp:BoundField DataField="Total" HeaderText="Total" DataFormatString="{0:C}"
/>

    </Columns>
</asp:GridView>
</ItemTemplate>
</asp:TemplateField>
</Columns>
</asp:GridView>
</div>
</form>
</body>
</html>

```

Imports System.Data

Imports System.Data.SqlClient

Partial Class ViewOrders

Inherits System.Web.UI.Page

```

Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

```

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

If Session("RoleName") Is Nothing OrElse Session("RoleName").ToString() <> "Customer" Then

Response.Redirect("Login.aspx")

End If

If Not IsPostBack Then

 LoadOrders()

End If

End Sub

Private Sub LoadOrders()

 Using con As New SqlConnection(connString)

 con.Open()

 Dim sql As String = "SELECT ORDER_ID, ORDER_DATE, PAYMENT_METHOD FROM ORDER_T
WHERE CUSTOMER_ID = @cid"

 Using cmd As New SqlCommand(sql, con)

 cmd.Parameters.AddWithValue("@cid", Session("CUSTOMER_ID"))

 Using da As New SqlDataAdapter(cmd)

 Dim dt As New DataTable()

 da.Fill(dt)

 gvOrders.DataSource = dt

 gvOrders.DataBind()

 End Using

 End Using

End Using

End Sub

Protected Sub gvOrders_RowDataBound(sender As Object, e As GridViewRowEventArgs)

 If e.Row.RowType = DataControlRowType.DataRow Then

 Dim orderId As Integer = Convert.ToInt32(DataBinder.Eval(e.Row.DataItem, "ORDER_ID"))

 Dim gvOrderDetails As GridView = e.Row.FindControl("gvOrderDetails")

```

Using con As New SqlConnection(connString)

con.Open()

Dim sql As String = "SELECT OL.PRODUCT_ID, OL.ORDERED_QUANTITY, P.SELLING_COST
AS UnitPrice, " &

"(OL.ORDERED_QUANTITY * P.SELLING_COST) AS Total " &

"FROM ORDER_LINE_T OL " &

"JOIN PRODUCT_T P ON OL.PRODUCT_ID = P.PRODUCT_ID " &

"WHERE OL.ORDER_ID = @oid"

Using cmd As New SqlCommand(sql, con)

cmd.Parameters.AddWithValue("@oid", orderId)

Using da As New SqlDataAdapter(cmd)

Dim dt As New DataTable()

da.Fill(dt)

gvOrderDetails.DataSource = dt

gvOrderDetails.DataBind()

End Using

End Using

End Using

End If

End Sub

Protected Sub btnBackToDashboard_Click(sender As Object, e As EventArgs)

Response.Redirect("Customer.aspx")

End Sub

End Class

```

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Supplier.aspx.vb"
Inherits="Supplier" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>Supplier Hub</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div align="center">
```

```
            <h2>Supplier Hub</h2>
```

```
            <asp:Label ID="lblWelcome" runat="server" Font-Bold="true" Font-Size="Large"
ForeColor="DarkBlue" /><br />
```

```
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red"></asp:Label><br /><br />
```

```
        <div>
```

```
            <asp:Button ID="btnAddMaterial" runat="server" Text="Add New Material"
OnClick="btnAddMaterial_Click" /><br /><br />
```

```
        </div>
```

```
        <div>
```

```
            <asp:Button ID="btnEditBio" runat="server" Text="Edit Bio" OnClick="btnEditBio_Click"
/><br /><br />
```

```
        </div>
```

```
        <div>
```

```
            <asp:Button ID="btnModifyMaterial" runat="server" Text="Modify Material"
OnClick="btnModifyMaterial_Click" /><br /><br />
```

```
        </div>
```

```
        <div>
```

```
<asp:Button ID="btnListMaterials" runat="server" Text="List Materials"
OnClick="btnListMaterials_Click" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Button ID="btnRemoveMaterial" runat="server" Text="Remove Material"
OnClick="btnRemoveMaterial_Click" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Button ID="btnSignOut" runat="server" Text="Sign Out" OnClick="btnSignOut_Click"
/><br /><br />
```

```
</div>
```

```
<p>--- SCM Portal 2025 ---</p>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

Imports System.Data

Imports System.Data.SqlClient

Partial Class Supplier

Inherits System.Web.UI.Page

```
Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"
```

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

If Not IsPostBack Then

If Session("SUPPLIER_ID") IsNot Nothing Then

```

Dim supplierId As Integer = Convert.ToInt32(Session("SUPPLIER_ID"))

Using conn As New SqlConnection(connString)

    Dim cmd As New SqlCommand("SELECT SUPPLIER_FIRST_NAME,
SUPPLIER_LAST_NAME FROM SUPPLIER_T WHERE SUPPLIER_ID = @id", conn)

    cmd.Parameters.AddWithValue("@id", supplierId)

    conn.Open()

    Using reader As SqlDataReader = cmd.ExecuteReader()

        If reader.Read() Then

            Dim firstName As String = reader("SUPPLIER_FIRST_NAME").ToString()

            Dim lastName As String = reader("SUPPLIER_LAST_NAME").ToString()

            lblWelcome.Text = "Welcome, " & firstName & " " & lastName

        Else

            lblWelcome.Text = "Welcome, Supplier"

        End If

    End Using

End Using

Else

    Response.Redirect("Login.aspx")

End If

End If

End Sub

```

```

Protected Sub btnAddMaterial_Click(sender As Object, e As EventArgs)

    Response.Redirect("EnterRawMaterial.aspx")

End Sub

```

```

Protected Sub btnEditBio_Click(sender As Object, e As EventArgs)

    Response.Redirect("UpdateSupplierBio.aspx")

End Sub

```

End Sub

Protected Sub btnModifyMaterial_Click(sender As Object, e As EventArgs)

 Response.Redirect("UpdateRawMaterial.aspx")

End Sub

Protected Sub btnListMaterials_Click(sender As Object, e As EventArgs)

 Response.Redirect("ViewMyMaterials.aspx")

End Sub

Protected Sub btnRemoveMaterial_Click(sender As Object, e As EventArgs)

 Response.Redirect("DeleteRawMaterial.aspx")

End Sub

Protected Sub btnSignOut_Click(sender As Object, e As EventArgs)

 Session.Clear()

 Response.Redirect("Login.aspx")

End Sub

End Class

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="ViewMyMaterials.aspx.vb"
Inherits="ViewMyMaterials" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

 <title>View My Supplied Materials</title>

</head>

```

<body>

    <form id="form1" runat="server">

        <div align="center">

            <h2>View My Supplied Raw Materials</h2>

            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />

            <asp:GridView ID="gvMaterials" runat="server" AutoGenerateColumns="False"
EmptyDataText="No materials found.">

                <Columns>

                    <asp:BoundField DataField="RAW_MATERIAL_NAME" HeaderText="Material Name" />

                    <asp:BoundField DataField="RAW_MATERIAL_QUANTITY" HeaderText="Quantity" />

                    <asp:BoundField DataField="RAW_MATERIAL_UNIT_PRICE" HeaderText="Unit Price" />

                    <asp:BoundField DataField="RM_RECEIVED_DATE" HeaderText="Received Date"
DataFormatString="{0:yyyy-MM-dd}" />

                    <asp:BoundField DataField="WAREHOUSE_NAME" HeaderText="Warehouse Name" />

                    <asp:BoundField DataField="WAREHOUSE_LOCATION" HeaderText="Location" />

                    <asp:BoundField DataField="WAREHOUSE_CAPACITY" HeaderText="Max Capacity" />

                    <asp:BoundField DataField="MANAGER_NAME" HeaderText="Manager" />

                    <asp:BoundField DataField="CONTACT_NUMBER" HeaderText="Contact" />

                    <asp:BoundField DataField="EMAIL" HeaderText="Email" />

                    <asp:BoundField DataField="OPERATIONAL_STATUS" HeaderText="Status" />

                </Columns>

            </asp:GridView>

        </div>

    </form>

</body>

</html>

```

Imports System.Data

Imports System.Data.SqlClient

Partial Class ViewMyMaterials

Inherits System.Web.UI.Page

Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial Catalog=SCM;Integrated Security=True"

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

If Not IsPostBack Then

If Session("SUPPLIER_ID") Is Nothing Then

Response.Redirect("Login.aspx")

Else

LoadMyMaterials()

End If

End If

End Sub

Private Sub LoadMyMaterials()

Dim sql As String

sql = "SELECT RM.RAW_MATERIAL_NAME, RM.RAW_MATERIAL_QUANTITY,
RM.RAW_MATERIAL_UNIT_PRICE, RM.RM_RECEIVED_DATE, "

sql &= "W.WAREHOUSE_NAME, "

sql &= "(W.WAREHOUSE_STREET_ADDRESS + ', ' + W.WAREHOUSE_CITY + ', ' +
W.WAREHOUSE_STATE) AS WAREHOUSE_LOCATION, "

sql &= "W.WAREHOUSE_CAPACITY, W.MANAGER_NAME, W.CONTACT_NUMBER, W.EMAIL,
W.OPERATIONAL_STATUS "

sql &= "FROM RAW_MATERIAL_T RM "

sql &= "INNER JOIN WAREHOUSE_T W ON RM.WAREHOUSE_ID = W.WAREHOUSE_ID "

sql &= "WHERE RM.SUPPLIER_ID = @sid "

```
sql &= "ORDER BY RM.RM_RECEIVED_DATE DESC"
```

```
Using conn As New SqlConnection(ConnStr),
```

```
    cmd As New SqlCommand(sql, conn)
```

```
    cmd.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))
```

```
    Dim dt As New DataTable()
```

```
    Try
```

```
        conn.Open()
```

```
        Using rdr As SqlDataReader = cmd.ExecuteReader()
```

```
            dt.Load(rdr)
```

```
        End Using
```

```
        gvMaterials.DataSource = dt
```

```
        gvMaterials.DataBind()
```

```
    Catch ex As Exception
```

```
        lblMessage.Text = "Error loading materials: " & ex.Message
```

```
    End Try
```

```
End Using
```

```
End Sub
```

```
End Class
```

.....

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="EnterRawMaterial.aspx.vb"  
Inherits="EnterRawMaterial" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title>Add Raw Material</title>

</head>

<body>

    <form id="form1" runat="server">

        <div align="center">

            <h2>Add New Raw Material to Inventory</h2>

            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />

            <table border="0" cellpadding="10">

                <tr>

                    <td>

                        <!-- Warehouse Selector and Details -->

                        Select Warehouse:<br />

                        <asp:DropDownList ID="ddlWarehouse" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlWarehouse_SelectedIndexChanged" /><br /><br />

                        <asp:Panel ID="pnlWhDetails" runat="server" Visible="False">

                            <b><asp:Label ID="lblWhName" runat="server" /></b><br />

                            <asp:Label ID="lblMaxCap" runat="server" /><br />

                            <asp:Label ID="lblUsedCap" runat="server" /><br />

                            <asp:Label ID="lblAvailCap" runat="server" ForeColor="Green" /><br /><br />

                        </asp:Panel>

                    </td>

                </tr>

            </table>

        </div>

    </form>

</body>

</html>
```

```

<table border="0" cellpadding="10">

  <tr>

    <td>

      <!-- Raw Material Form -->

      <asp:Panel ID="pnlRawForm" runat="server" Visible="False">

        <h4>New Raw Material Form</h4>

        Raw Material Name:<br />

        <asp:TextBox ID="txtRawName" runat="server" Width="300px" /><br /><br />

        Quantity to Add:<br />

        <asp:TextBox ID="txtRawQty" runat="server" Width="100px" /><br /><br />

        Unit Price (e.g., 25.50):<br />

        <asp:TextBox ID="txtRawPrice" runat="server" Width="100px" /><br /><br />

        Description (optional):<br />

        <asp:TextBox ID="txtRawDesc" runat="server" TextMode="MultiLine" Rows="3"
Columns="50" /><br /><br />

        <asp:Button ID="btnAddRaw" runat="server" Text="Insert Material"
OnClick="btnAddRaw_Click" />

      </asp:Panel>

    </td>

  </tr>

</table>

</div>

</form>

</body>

</html>

```

Imports System.Data.SqlClient

Partial Class EnterRawMaterial

Inherits System.Web.UI.Page

```
Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial  
Catalog=SCM;Integrated Security=True"
```

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

 ' Role-Based Access & Session Management

```
    If Session("SUPPLIER_ID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse  
Session("RoleName").ToString() <> "Supplier" Then
```

```
        Response.Redirect("Login.aspx")
```

```
    End If
```

 If Not IsPostBack Then

```
        LoadWarehouses()
```

```
    End If
```

End Sub

''' <summary>

''' Load only warehouses that are Active and not full.

''' </summary>

Private Sub LoadWarehouses()

```
    ddlWarehouse.Items.Clear()
```

```
    ddlWarehouse.Items.Add(New ListItem("-- Select --", ""))
```

 ' SQL Injection Prevention

```
    Dim sql As String = "SELECT W.WAREHOUSE_ID, W.WAREHOUSE_NAME " &
```

```
        "FROM WAREHOUSE_T W " &
```

```
        "WHERE W.OPERATIONAL_STATUS = @status " &
```

```

"AND W.WAREHOUSE_CAPACITY > ISNULL( " &
"SELECT SUM(RM.RAW_MATERIAL_QUANTITY) " &
"FROM RAW_MATERIAL_T RM " &
"WHERE RM.WAREHOUSE_ID = W.WAREHOUSE_ID " &
"), 0) " &
"ORDER BY W.WAREHOUSE_NAME"

```

```

Using conn As New SqlConnection(ConnStr)

conn.Open()

Using cmd As New SqlCommand(sql, conn)

    cmd.Parameters.AddWithValue("@status", "Active")

    Using rdr = cmd.ExecuteReader()

        While rdr.Read()

            ddlWarehouse.Items.Add(New ListItem(

                rdr("WAREHOUSE_NAME").ToString(),

                rdr("WAREHOUSE_ID").ToString()

            ))

        End While

    End Using

End Using

End Using

End Sub

Protected Sub ddlWarehouse_SelectedIndexChanged(sender As Object, e As EventArgs)

    lblMessage.Text = ""

    ' Input Validation

    If ddlWarehouse.SelectedValue = "" Then

```

pnlWhDetails.Visible = False

pnlRawForm.Visible = False

Return

End If

Dim wid As Integer

If Not Integer.TryParse(ddlWarehouse.SelectedValue, wid) Then

lblMessage.Text = "Invalid warehouse selected."

Return

End If

Using conn As New SqlConnection(ConnStr)

conn.Open()

' SQL Injection Prevention

Dim sqlWh As String = "SELECT WAREHOUSE_NAME, WAREHOUSE_CAPACITY " &
"FROM WAREHOUSE_T WHERE WAREHOUSE_ID = @id"

Using cmdWh As New SqlCommand(sqlWh, conn)

cmdWh.Parameters.AddWithValue("@id", wid)

Using rdr = cmdWh.ExecuteReader()

If rdr.Read() Then

lblWhName.Text = "Warehouse: " & rdr("WAREHOUSE_NAME").ToString()

lblMaxCap.Text = "Max Capacity: " & rdr("WAREHOUSE_CAPACITY").ToString()

Else

lblMessage.Text = "Warehouse not found."

Return

End If

End Using

End Using

' Compute used & available capacity

Dim sqlUsed As String = "SELECT ISNULL(SUM(RAW_MATERIAL_QUANTITY), 0) " &
"FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID = @id"

Using cmdUsed As New SqlCommand(sqlUsed, conn)

cmdUsed.Parameters.AddWithValue("@id", wid)

Dim used = Convert.ToDecimal(cmdUsed.ExecuteScalar())

lblUsedCap.Text = "Used Capacity: " & used.ToString()

Dim maxCap As Decimal

If Not Decimal.TryParse(lblMaxCap.Text.Split(":")(1), maxCap) Then

lblMessage.Text = "Error retrieving capacity."

Return

End If

Dim avail = maxCap - used

lblAvailCap.Text = "Available Capacity: " & avail.ToString()

ViewState("AvailableCapacity") = avail

End Using

End Using

pnlWhDetails.Visible = True

pnlRawForm.Visible = True

End Sub

Protected Sub btnAddRaw_Click(sender As Object, e As EventArgs)

lblMessage.ForeColor = Drawing.Color.Red

```
lblMessage.Text = ""
```

```
' Input Validation
```

```
If ddlWarehouse.SelectedValue = "" Then
```

```
    lblMessage.Text = "Please select a warehouse."
```

```
    Return
```

```
End If
```

```
Dim name = txtRawName.Text.Trim()
```

```
If String.IsNullOrEmpty(name) Then
```

```
    lblMessage.Text = "Enter a raw material name."
```

```
    Return
```

```
End If
```

```
Dim qty As Decimal
```

```
If Not Decimal.TryParse(txtRawQty.Text.Trim(), qty) OrElse qty <= 0 Then
```

```
    lblMessage.Text = "Enter a valid positive quantity."
```

```
    Return
```

```
End If
```

```
Dim price As Decimal
```

```
If Not Decimal.TryParse(txtRawPrice.Text.Trim(), price) OrElse price <= 0 Then
```

```
    lblMessage.Text = "Enter a valid positive unit price."
```

```
    Return
```

```
End If
```

```
' Case-insensitive uniqueness check
```

```

Using conn As New SqlConnection(ConnStr)

    conn.Open()

    Dim sqlCheck As String = "SELECT COUNT(*) FROM RAW_MATERIAL_T WHERE
LOWER(RAW_MATERIAL_NAME) = LOWER(@n)"

    Using cmdCheck As New SqlCommand(sqlCheck, conn)

        cmdCheck.Parameters.AddWithValue("@n", name)

        If Convert.ToInt32(cmdCheck.ExecuteScalar()) > 0 Then

            lblMessage.Text = "This raw material already exists."

            Return

        End If

    End Using

End Using

' Capacity check

Dim avail As Decimal

If Not Decimal.TryParse(ViewState("AvailableCapacity").ToString(), avail) OrElse qty > avail Then

    lblMessage.Text = String.Format("Quantity exceeds available capacity {0}.", avail)

    Return

End If

' Error Handling: Use transaction

Using conn As New SqlConnection(ConnStr)

    conn.Open()

    Dim trans = conn.BeginTransaction()

    Try

        ' Compute new ID

        Dim sqlNewId As String = "SELECT ISNULL(MAX(RAW_MATERIAL_ID), 0) + 1 FROM
RAW_MATERIAL_T"

```

Dim newRawMaterialId As Integer

Using cmdNewId As New SqlCommand(sqlNewId, conn, trans)

newRawMaterialId = Convert.ToInt32(cmdNewId.ExecuteScalar())

End Using

' Insert into RAW_MATERIAL_T

Dim sqlIns As String = "INSERT INTO RAW_MATERIAL_T " &

"(RAW_MATERIAL_ID, WAREHOUSE_ID, RAW_MATERIAL_NAME, " &

"RAW_MATERIAL_QUANTITY, RAW_MATERIAL_UNIT_PRICE, SUPPLIER_ID,
RM_RECEIVED_DATE) " &

"VALUES (@rmid, @wid, @n, @q, @price, @sid, @date)"

Using cmdIns As New SqlCommand(sqlIns, conn, trans)

cmdIns.Parameters.AddWithValue("@rmid", newRawMaterialId)

cmdIns.Parameters.AddWithValue("@wid", ddlWarehouse.SelectedValue)

cmdIns.Parameters.AddWithValue("@n", name)

cmdIns.Parameters.AddWithValue("@q", qty)

cmdIns.Parameters.AddWithValue("@price", price)

cmdIns.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))

cmdIns.Parameters.AddWithValue("@date", DateTime.Now)

cmdIns.ExecuteNonQuery()

End Using

' Insert into MATERIAL_SUPPLIER

Dim sqlSupp As String = "INSERT INTO MATERIAL_SUPPLIER " &

"(SUPPLIER_ID, RAW_MATERIAL_ID, SUPPLY_DATE, QUANTITY_SUPPLIED) " &

"VALUES (@sid, @rmid, @supp_date, @q)"

Using cmdSupp As New SqlCommand(sqlSupp, conn, trans)

cmdSupp.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))

```
cmdSupp.Parameters.AddWithValue("@rmid", newRawMaterialId)

cmdSupp.Parameters.AddWithValue("@supp_date", DateTime.Today)

cmdSupp.Parameters.AddWithValue("@q", qty)

cmdSupp.ExecuteNonQuery()

End Using

trans.Commit()

lblMessage.ForeColor = Drawing.Color.Green

lblMessage.Text = "Raw material added successfully!"

' Refresh UI

LoadWarehouses()

pnlWhDetails.Visible = False

pnlRawForm.Visible = False

txtRawName.Text = ""

txtRawQty.Text = ""

txtRawPrice.Text = ""

txtRawDesc.Text = ""

Catch ex As Exception

trans.Rollback()

lblMessage.Text = "Error adding raw material: " & ex.Message

End Try

End Using

End Sub

End Class
```

Imports System.Data.SqlClient

Partial Class EnterRawMaterial

Inherits System.Web.UI.Page

Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"

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

' Role-Based Access & Session Management

If Session("SUPPLIER_ID") Is Nothing OrElse Session("RoleName") Is Nothing OrElse
Session("RoleName").ToString() <> "Supplier" Then

Response.Redirect("Login.aspx")

End If

If Not IsPostBack Then

LoadWarehouses()

End If

End Sub

''' <summary>

''' Load only warehouses that are Active and not full.

''' </summary>

Private Sub LoadWarehouses()

ddlWarehouse.Items.Clear()

ddlWarehouse.Items.Add(New ListItem("-- Select --", ""))

' SQL Injection Prevention

```

Dim sql As String = "SELECT W.WAREHOUSE_ID, W.WAREHOUSE_NAME " &
    "FROM WAREHOUSE_T W " &
    "WHERE W.OPERATIONAL_STATUS = @status " &
    "AND W.WAREHOUSE_CAPACITY > ISNULL( " &
    "SELECT SUM(RM.RAW_MATERIAL_QUANTITY) " &
    "FROM RAW_MATERIAL_T RM " &
    "WHERE RM.WAREHOUSE_ID = W.WAREHOUSE_ID " &
    "), 0) " &
    "ORDER BY W.WAREHOUSE_NAME"

```

```

Using conn As New SqlConnection(ConnStr)
    conn.Open()
    Using cmd As New SqlCommand(sql, conn)
        cmd.Parameters.AddWithValue("@status", "Active")
        Using rdr = cmd.ExecuteReader()
            While rdr.Read()
                ddlWarehouse.Items.Add(New ListItem(
                    rdr("WAREHOUSE_NAME").ToString(),
                    rdr("WAREHOUSE_ID").ToString()
                ))
            End While
        End Using
    End Using
End Using
End Sub

```

```

Protected Sub ddlWarehouse_SelectedIndexChanged(sender As Object, e As EventArgs)

```

```
lblMessage.Text = ""
```

```
' Input Validation
```

```
If ddlWarehouse.SelectedValue = "" Then
```

```
    pnlWhDetails.Visible = False
```

```
    pnlRawForm.Visible = False
```

```
    Return
```

```
End If
```

```
Dim wid As Integer
```

```
If Not Integer.TryParse(ddlWarehouse.SelectedValue, wid) Then
```

```
    lblMessage.Text = "Invalid warehouse selected."
```

```
    Return
```

```
End If
```

```
Using conn As New SqlConnection(ConnStr)
```

```
    conn.Open()
```

```
' SQL Injection Prevention
```

```
Dim sqlWh As String = "SELECT WAREHOUSE_NAME, WAREHOUSE_CAPACITY " &
```

```
    "FROM WAREHOUSE_T WHERE WAREHOUSE_ID = @id"
```

```
Using cmdWh As New SqlCommand(sqlWh, conn)
```

```
    cmdWh.Parameters.AddWithValue("@id", wid)
```

```
    Using rdr = cmdWh.ExecuteReader()
```

```
        If rdr.Read() Then
```

```
            lblWhName.Text = "Warehouse: " & rdr("WAREHOUSE_NAME").ToString()
```

```
            lblMaxCap.Text = "Max Capacity: " & rdr("WAREHOUSE_CAPACITY").ToString()
```

```
        Else
```

```
            lblMessage.Text = "Warehouse not found."
```

```

        Return

    End If

    End Using

End Using

' Compute used & available capacity

Dim sqlUsed As String = "SELECT ISNULL(SUM(RAW_MATERIAL_QUANTITY), 0) " &
    "FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID = @id"

Using cmdUsed As New SqlCommand(sqlUsed, conn)

    cmdUsed.Parameters.AddWithValue("@id", wid)

    Dim used = Convert.ToDecimal(cmdUsed.ExecuteScalar())

    lblUsedCap.Text = "Used Capacity: " & used.ToString()

    Dim maxCap As Decimal

    If Not Decimal.TryParse(lblMaxCap.Text.Split(":")(1), maxCap) Then

        lblMessage.Text = "Error retrieving capacity."

        Return

    End If

    Dim avail = maxCap - used

    lblAvailCap.Text = "Available Capacity: " & avail.ToString()

    ViewState("AvailableCapacity") = avail

End Using

End Using

pnlWhDetails.Visible = True

pnlRawForm.Visible = True

End Sub

```

```
Protected Sub btnAddRaw_Click(sender As Object, e As EventArgs)
```

```
    lblMessage.ForeColor = Drawing.Color.Red
```

```
    lblMessage.Text = ""
```

```
    ' Input Validation
```

```
    If ddlWarehouse.SelectedValue = "" Then
```

```
        lblMessage.Text = "Please select a warehouse."
```

```
        Return
```

```
    End If
```

```
    Dim name = txtRawName.Text.Trim()
```

```
    If String.IsNullOrEmpty(name) Then
```

```
        lblMessage.Text = "Enter a raw material name."
```

```
        Return
```

```
    End If
```

```
    Dim qty As Decimal
```

```
    If Not Decimal.TryParse(txtRawQty.Text.Trim(), qty) OrElse qty <= 0 Then
```

```
        lblMessage.Text = "Enter a valid positive quantity."
```

```
        Return
```

```
    End If
```

```
    Dim price As Decimal
```

```
    If Not Decimal.TryParse(txtRawPrice.Text.Trim(), price) OrElse price <= 0 Then
```

```
        lblMessage.Text = "Enter a valid positive unit price."
```

```
        Return
```

End If

' Case-insensitive uniqueness check

Using conn As New SqlConnection(ConnStr)

conn.Open()

Dim sqlCheck As String = "SELECT COUNT(*) FROM RAW_MATERIAL_T WHERE
LOWER(RAW_MATERIAL_NAME) = LOWER(@n)"

Using cmdCheck As New SqlCommand(sqlCheck, conn)

cmdCheck.Parameters.AddWithValue("@n", name)

If Convert.ToInt32(cmdCheck.ExecuteScalar()) > 0 Then

lblMessage.Text = "This raw material already exists."

Return

End If

End Using

End Using

' Capacity check

Dim avail As Decimal

If Not Decimal.TryParse(ViewState("AvailableCapacity").ToString(), avail) OrElse qty > avail Then

lblMessage.Text = String.Format("Quantity exceeds available capacity {0}.", avail)

Return

End If

' Error Handling: Use transaction

Using conn As New SqlConnection(ConnStr)

conn.Open()

Dim trans = conn.BeginTransaction()

Try

```

' Compute new ID

Dim sqlNewId As String = "SELECT ISNULL(MAX(RAW_MATERIAL_ID), 0) + 1 FROM
RAW_MATERIAL_T"

Dim newRawMaterialId As Integer

Using cmdNewId As New SqlCommand(sqlNewId, conn, trans)

    newRawMaterialId = Convert.ToInt32(cmdNewId.ExecuteScalar())

End Using


' Insert into RAW_MATERIAL_T

Dim sqlIns As String = "INSERT INTO RAW_MATERIAL_T " &

    "(RAW_MATERIAL_ID, WAREHOUSE_ID, RAW_MATERIAL_NAME, " &

    "RAW_MATERIAL_QUANTITY, RAW_MATERIAL_UNIT_PRICE, SUPPLIER_ID,
RM_RECEIVED_DATE) " &

    "VALUES (@rmid, @wid, @n, @q, @price, @sid, @date)"

Using cmdIns As New SqlCommand(sqlIns, conn, trans)

    cmdIns.Parameters.AddWithValue("@rmid", newRawMaterialId)

    cmdIns.Parameters.AddWithValue("@wid", ddlWarehouse.SelectedValue)

    cmdIns.Parameters.AddWithValue("@n", name)

    cmdIns.Parameters.AddWithValue("@q", qty)

    cmdIns.Parameters.AddWithValue("@price", price)

    cmdIns.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))

    cmdIns.Parameters.AddWithValue("@date", DateTime.Now)

    cmdIns.ExecuteNonQuery()

End Using


' Insert into MATERIAL_SUPPLIER

Dim sqlSupp As String = "INSERT INTO MATERIAL_SUPPLIER " &

    "(SUPPLIER_ID, RAW_MATERIAL_ID, SUPPLY_DATE, QUANTITY_SUPPLIED) " &

```

```

        "VALUES (@sid, @rmid, @supp_date, @q)"
Using cmdSupp As New SqlCommand(sqlSupp, conn, trans)

    cmdSupp.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))

    cmdSupp.Parameters.AddWithValue("@rmid", newRawMaterialId)

    cmdSupp.Parameters.AddWithValue("@supp_date", DateTime.Today)

    cmdSupp.Parameters.AddWithValue("@q", qty)

    cmdSupp.ExecuteNonQuery()

End Using

trans.Commit()

lblMessage.ForeColor = Drawing.Color.Green

lblMessage.Text = "Raw material added successfully!"

' Refresh UI

LoadWarehouses()

pnlWhDetails.Visible = False

pnlRawForm.Visible = False

txtRawName.Text = ""

txtRawQty.Text = ""

txtRawPrice.Text = ""

txtRawDesc.Text = ""

Catch ex As Exception

    trans.Rollback()

    lblMessage.Text = "Error adding raw material: " & ex.Message

End Try

End Using

End Sub

```

End Class

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="DeleteRawMaterial.aspx.vb"
Inherits="DeleteRawMaterial" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>Delete Raw Material</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div align="center">
```

```
            <h2>Remove Raw Material from Inventory</h2>
```

```
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />
```

```
            <asp:DropDownList ID="ddlRawMaterials" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlRawMaterials_SelectedIndexChanged" />
```

```
            <br /><br />
```

```
            <asp:Panel ID="pnlDetails" runat="server" Visible="False">
```

```
                <h4>Raw Material Information</h4>
```

```
                <asp:Label ID="lblDetails" runat="server" /><br /><br />
```

```
                <asp:Label ID="lblUsedIn" runat="server" ForeColor="Red" /><br /><br />
```

```
                <asp:Button ID="btnDelete" runat="server" Text="Delete Material"
OnClick="btnDelete_Click" />
```

```
</asp:Panel>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
Imports System.Data.SqlClient
```

```
Partial Class DeleteRawMaterial
```

```
Inherits System.Web.UI.Page
```

```
Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial  
Catalog=SCM;Integrated Security=True"
```

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

```
If Not IsPostBack Then
```

```
LoadRawMaterials()
```

```
End If
```

```
End Sub
```

```
Private Sub LoadRawMaterials()
```

```
ddlRawMaterials.Items.Clear()
```

```
ddlRawMaterials.Items.Add("-- Select Raw Material --")
```

```
Dim supplierId = Convert.ToInt32(Session("SUPPLIER_ID"))
```

```
Using conn As New SqlConnection(connString)
```

```
Dim cmd As New SqlCommand("SELECT RAW_MATERIAL_ID, RAW_MATERIAL_NAME FROM  
RAW_MATERIAL_T WHERE SUPPLIER_ID = @sid", conn)
```

```

cmd.Parameters.AddWithValue("@sid", supplierId)

conn.Open()

Dim rdr = cmd.ExecuteReader()

While rdr.Read()

    ddlRawMaterials.Items.Add(New ListItem(rdr("RAW_MATERIAL_NAME").ToString(),
rdr("RAW_MATERIAL_ID").ToString()))

End While

End Using

End Sub

```

```

Protected Sub ddlRawMaterials_SelectedIndexChanged(sender As Object, e As EventArgs)

```

```

    If ddlRawMaterials.SelectedIndex <= 0 Then

```

```

        pnlDetails.Visible = False

```

```

        lblUsedIn.Text = ""

```

```

        btnDelete.Visible = False

```

```

        Return

```

```

    End If

```

```

Dim rawMaterialId = Convert.ToInt32(ddlRawMaterials.SelectedValue)

```

```

Using conn As New SqlConnection(connString)

```

```

    conn.Open()

```

```

    ' 1. Get Raw Material and Warehouse Details

```

```

    Dim sql As String = "SELECT RM.RAW_MATERIAL_NAME, RM.RAW_MATERIAL_QUANTITY,
RM.RAW_MATERIAL_UNIT_PRICE, " &

```

```

    "W.WAREHOUSE_NAME, (W.WAREHOUSE_STREET_ADDRESS + ', ' + W.WAREHOUSE_CITY + ',
' + W.WAREHOUSE_STATE) AS WAREHOUSE_LOCATION " &

```

```

"FROM RAW_MATERIAL_T RM " &
"JOIN WAREHOUSE_T W ON RM.WAREHOUSE_ID = W.WAREHOUSE_ID " &
"WHERE RM.RAW_MATERIAL_ID = @rmid"

Dim cmd As New SqlCommand(sql, conn)

cmd.Parameters.AddWithValue("@rmid", rawMaterialId)

Dim rdr = cmd.ExecuteReader()

If rdr.Read() Then

    lblDetails.Text = "<b>Name:</b> " & rdr("RAW_MATERIAL_NAME").ToString() & "<br/>" &
    "<b>Quantity:</b> " & rdr("RAW_MATERIAL_QUANTITY").ToString() & "<br/>" &
    "<b>Unit Price:</b> " & rdr("RAW_MATERIAL_UNIT_PRICE").ToString() & "<br/>" &
    "<b>Warehouse:</b> " & rdr("WAREHOUSE_NAME").ToString() & "<br/>" &
    "<b>Location:</b> " & rdr("WAREHOUSE_LOCATION").ToString()

    pnlDetails.Visible = True

End If

rdr.Close()

```

' 2. Check if raw material is used in any product

```

Dim sqlCheckUse As String = "SELECT P.PRODUCT_NAME " &
"FROM QUANTITY_IN QI " &
"JOIN PRODUCT_T P ON QI.PRODUCT_ID = P.PRODUCT_ID " &
"WHERE QI.RAW_MATERIAL_ID = @rmid"

Dim cmdCheck As New SqlCommand(sqlCheckUse, conn)

cmdCheck.Parameters.AddWithValue("@rmid", rawMaterialId)

Dim usedRdr = cmdCheck.ExecuteReader()

If usedRdr.HasRows Then

```

```

        ' Raw material is used in product(s)
        Dim usedInProducts As String = ""
        While usedRdr.Read()
            usedInProducts &= usedRdr("PRODUCT_NAME").ToString() & ", "
        End While
        usedInProducts = usedInProducts.TrimEnd(", ", " ")

        lblUsedIn.Text = "<b style='color:red;'>This raw material is used in making:</b> " &
usedInProducts

        btnDelete.Visible = False
    Else
        ' Raw material is not used
        lblUsedIn.Text = ""
        btnDelete.Visible = True
    End If
    usedRdr.Close()
End Using
End Sub

```

```

Protected Sub btnDelete_Click(sender As Object, e As EventArgs)
    Dim rawMaterialId = Convert.ToInt32(ddlRawMaterials.SelectedValue)

    ' Check if the raw material is used in any products
    Dim productsUsedIn As New List(Of String)
    Using conn As New SqlConnection(connString)
        Dim sql As String
    
```

```
sql = "SELECT P.PRODUCT_NAME FROM QUANTITY_IN Q JOIN PRODUCT_T P ON  
Q.PRODUCT_ID = P.PRODUCT_ID "
```

```
sql &= "WHERE Q.RAW_MATERIAL_ID = @rmid"
```

```
Dim cmd As New SqlCommand(sql, conn)
```

```
cmd.Parameters.AddWithValue("@rmid", rawMaterialId)
```

```
conn.Open()
```

```
Dim rdr = cmd.ExecuteReader()
```

```
While rdr.Read()
```

```
    productsUsedIn.Add(rdr("PRODUCT_NAME").ToString())
```

```
End While
```

```
End Using
```

```
If productsUsedIn.Count > 0 Then
```

```
    lblMessage.Text = "This raw material is used in the following product(s):<br/>" &  
String.Join("<br/>", productsUsedIn)
```

```
    Return
```

```
End If
```

```
' Not used in products, proceed to delete
```

```
Using conn As New SqlConnection(connString)
```

```
conn.Open()
```

```
Dim tran = conn.BeginTransaction()
```

```
Try
```

```
    ' Delete from MATERIAL_SUPPLY
```

```
    Dim cmd1 As New SqlCommand("DELETE FROM MATERIAL_SUPPLIER WHERE  
RAW_MATERIAL_ID = @rmid", conn, tran)
```

```
    cmd1.Parameters.AddWithValue("@rmid", rawMaterialId)
```

```

cmd1.ExecuteNonQuery()

' Delete from RAW_MATERIAL_T

Dim cmd2 As New SqlCommand("DELETE FROM RAW_MATERIAL_T WHERE
RAW_MATERIAL_ID = @rmid", conn, tran)

cmd2.Parameters.AddWithValue("@rmid", rawMaterialId)

cmd2.ExecuteNonQuery()

tran.Commit()

lblMessage.ForeColor = System.Drawing.Color.Green

lblMessage.Text = "Raw material deleted successfully."

pnlDetails.Visible = False

LoadRawMaterials()

Catch ex As Exception

tran.Rollback()

lblMessage.Text = "Error deleting raw material: " & ex.Message

End Try

End Using

End Sub

End Class

```

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="UpdateSupplierBio.aspx.vb"
Inherits="UpdateSupplierBio" %>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head runat="server">

```

```
<title>Update Your Profile</title>

</head>

<body>

  <form id="form1" runat="server">

    <h2>Welcome, <asp:Label ID="lblUsername" runat="server" /></h2>


    <table>

      <tr>

        <td>First Name:</td>

        <td><asp:TextBox ID="txtFirstName" runat="server" MaxLength="20" /></td>

      </tr>

      <tr>

        <td>Last Name:</td>

        <td><asp:TextBox ID="txtLastName" runat="server" MaxLength="20" /></td>

      </tr>

      <tr>

        <td>Email:</td>

        <td><asp:TextBox ID="txtEmail" runat="server" MaxLength="35" /></td>

      </tr>

      <tr>

        <td>City:</td>

        <td><asp:TextBox ID="txtCity" runat="server" MaxLength="20" /></td>

      </tr>

      <tr>

        <td>State:</td>

        <td><asp:TextBox ID="txtState" runat="server" MaxLength="15" /></td>

      </tr>

    </table>

  </form>

</body>

</html>
```

```
<tr>

    <td>Postal Code:</td>

    <td><asp:TextBox ID="txtPostalCode" runat="server" MaxLength="7" /></td>

</tr>

<tr>

    <td>Street Address:</td>

    <td><asp:TextBox ID="txtStreetAddress" runat="server" MaxLength="30" /></td>

</tr>

<tr>

    <td>Contact Number:</td>

    <td><asp:TextBox ID="txtContactNumber" runat="server" MaxLength="11" /></td>

</tr>

<tr>

    <td>User Name:</td>

    <td><asp:TextBox ID="username_supplier" runat="server" MaxLength="20" /></td>

</tr>

<!-- Password change section -->

<tr>

    <td>Old Password:</td>

    <td><asp:TextBox ID="txtOldPassword" runat="server" TextMode="Password" /></td>

</tr>

<tr>

    <td>New Password:</td>

    <td><asp:TextBox ID="txtNewPassword" runat="server" TextMode="Password" /></td>

</tr>

<tr>

    <td colspan="2" style="text-align:center;">
```

```
        <asp:Button ID="btnUpdate" runat="server" Text="Save Changes"
OnClick="btnUpdate_Click" />
```

```
    </td>
```

```
</tr>
```

```
</table>
```

```
    <asp:Label ID="lblMessage" runat="server" ForeColor="Red" />
```

```
</form>
```

```
</body>
```

```
</html>
```

```
Imports System.Data.SqlClient
```

```
Imports System.Net.NetworkInformation
```

```
Imports System.Security.Cryptography
```

```
Imports System.Text
```

```
Imports System.Text.RegularExpressions
```

```
Partial Class UpdateSupplierBio
```

```
    Inherits System.Web.UI.Page
```

```
    Private ReadOnly connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"
```

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

```
    If Not IsPostBack Then
```

```
        LoadCurrentSupplier()
```

```
    End If
```

```
End Sub
```

Private Sub LoadCurrentSupplier()

Dim supplierId = Convert.ToInt32(Session("SUPPLIER_ID"))

Dim sqlsts As String

sqlsts = "SELECT S.*, U.USERNAME, U.PASSWORD_HASH, U.SALT "

sqlsts &= "FROM SUPPLIER_T S "

sqlsts &= "JOIN [USER_T] U ON S.SUPPLIER_ID = U.SUPPLIER_ID "

sqlsts &= "WHERE S.SUPPLIER_ID = @sid"

Using conn As New SqlConnection(connString)

conn.Open()

' First query: Supplier and user info

Using cmd As New SqlCommand(sqlsts, conn)

cmd.Parameters.AddWithValue("@sid", supplierId)

Using rdr = cmd.ExecuteReader()

If rdr.Read() Then

lblUsername.Text = rdr("USERNAME").ToString()

txtFirstName.Text = rdr("SUPPLIER_FIRST_NAME").ToString().Trim()

txtLastName.Text = rdr("SUPPLIER_LAST_NAME").ToString().Trim()

txtEmail.Text = rdr("SUPPLIER_EMAIL").ToString().Trim()

txtCity.Text = rdr("SUPPLIER_CITY").ToString().Trim()

txtState.Text = rdr("SUPPLIER_STATE").ToString().Trim()

txtPostalCode.Text = rdr("SUPPLIER_POSTAL_CODE").ToString().Trim()

txtStreetAddress.Text = rdr("SUPPLIER_STREET_ADDRESS").ToString().Trim()

username_supplier.Text = rdr("USERNAME").ToString().Trim()

ViewState("origHash") = rdr("PASSWORD_HASH").ToString()

```

        ViewState("origSalt") = rdr("SALT").ToString()

    End If

    End Using

End Using

' Second query: Contact number from SUPPLIER_CONTACT

Dim sql2 As String

sql2 = "SELECT CONTACT_NUMBER FROM SUPPLIER_CONTACT WHERE SUPPLIER_ID =
@sid"

Using cmd2 As New SqlCommand(sql2, conn)

    cmd2.Parameters.AddWithValue("@sid", supplierId)

    Dim phone = cmd2.ExecuteScalar()

    If phone IsNot Nothing Then

        txtContactNumber.Text = phone.ToString().Trim()

    End If

End Using

End Using

End Sub

```

```

Private Function ValidatePasswordComplexity(ByVal pw As String) As Boolean

    If pw.Trim().Length < 10 Then Return False

    Dim special = Regex.Matches(pw, "[^A-Za-z0-9]").Count

    Dim digits = Regex.Matches(pw, "\d").Count

    Dim letters = Regex.Matches(pw, "[A-Za-z]").Count

    Return special >= 4 AndAlso digits >= 4 AndAlso letters >= 2

End Function

```

```

Protected Sub btnUpdate_Click(sender As Object, e As EventArgs)

```

If Not ValidateSupplier() Then Exit Sub

Dim oldPwd = txtOldPassword.Text

Dim salt = ViewState("origSalt").ToString()

Dim hashAttempt = HashPassword(oldPwd, salt)

If hashAttempt <> ViewState("origHash").ToString() Then

 lblMessage.Text = "Old password is incorrect."

 Return

End If

Dim newPwd = txtNewPassword.Text

If newPwd = oldPwd Then

 lblMessage.Text = "New password cannot match old password."

 Return

End If

If newPwd = "" Or oldPwd = "" Then

 lblMessage.Text = "Provide both old / new password."

 Return

End If

If Not ValidatePasswordComplexity(newPwd) Then

 lblMessage.Text = "New password must be at least 10 characters long and include at least 4 special characters, 4 digits, and 2 letters."

 Return

End If

' Check unique email/phone

If Not IsUniqueEmail(txtEmail.Text) Then

 lblMessage.Text = "Email is already taken."

 Return

End If

If Not IsUniquePhone(txtContactNumber.Text) Then

 lblMessage.Text = "Contact number is already taken."

 Return

End If

' Everything OK => update DB

Dim newSalt = GenerateSalt()

Dim newHash = HashPassword(newPwd, newSalt)

'Dim sqlstmnt As String

'sqlstmnt = "UPDATE SUPPLIER_T SET "

'sqlstmnt &= "SUPPLIER_FIRST_NAME=@fn, SUPPLIER_LAST_NAME=@ln, "

'sqlstmnt &= "SUPPLIER_EMAIL=@email, SUPPLIER_CITY=@city, "

'sqlstmnt &= "SUPPLIER_STATE=@st, SUPPLIER_POSTAL_CODE=@pc, "

'sqlstmnt &= "SUPPLIER_STREET_ADDRESS=@addr "

'sqlstmnt &= ""

Using conn As New SqlConnection(connString)

 conn.Open()

' First update to SUPPLIER_T

Dim query1 As String = "UPDATE SUPPLIER_T SET " &

"SUPPLIER_FIRST_NAME=@fn, SUPPLIER_LAST_NAME=@ln, " &

"SUPPLIER_EMAIL=@email, SUPPLIER_CITY=@city, " &

"SUPPLIER_STATE=@st, SUPPLIER_POSTAL_CODE=@pc, " &

"SUPPLIER_STREET_ADDRESS=@addr " &

"WHERE SUPPLIER_ID=@sid;"

Dim cmd1 As New SqlCommand(query1, conn)

cmd1.Parameters.AddWithValue("@fn", txtFirstName.Text.Trim())

cmd1.Parameters.AddWithValue("@ln", txtLastName.Text.Trim())

cmd1.Parameters.AddWithValue("@email", txtEmail.Text.Trim())

cmd1.Parameters.AddWithValue("@city", txtCity.Text.Trim())

cmd1.Parameters.AddWithValue("@st", txtState.Text.Trim())

cmd1.Parameters.AddWithValue("@pc", txtPostalCode.Text.Trim())

cmd1.Parameters.AddWithValue("@addr", txtStreetAddress.Text.Trim())

cmd1.Parameters.AddWithValue("@sid", Convert.ToInt32(Session("SUPPLIER_ID")))

cmd1.ExecuteNonQuery()

' Second update to USER_T

Dim query2 As String = "UPDATE USER_T SET " &

"PASSWORD_HASH=@hash, SALT=@salt, USERNAME=@uname " &

"WHERE SUPPLIER_ID=@sid;"

Dim cmd2 As New SqlCommand(query2, conn)

cmd2.Parameters.AddWithValue("@hash", newHash)

cmd2.Parameters.AddWithValue("@salt", newSalt)

cmd2.Parameters.AddWithValue("@sid", Convert.ToInt32(Session("SUPPLIER_ID")))

cmd2.Parameters.AddWithValue("@uname", username_supplier.Text)

```
cmd2.ExecuteNonQuery()
```

```
End Using
```

```
lblMessage.ForeColor = System.Drawing.Color.Green
```

```
lblMessage.Text = "Profile updated successfully."
```

```
End Sub
```

```
' — Validation routines —————
```

```
Private Function IsUniqueEmail(email As String) As Boolean
```

```
    Dim sql = "SELECT COUNT(*) FROM [USER_T] WHERE SUPPLIER_ID <> @sid AND USERNAME  
IS NOT NULL AND USERNAME = @email"
```

```
    Using conn As New SqlConnection(connString), cmd = New SqlCommand(sql, conn)
```

```
        cmd.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))
```

```
        cmd.Parameters.AddWithValue("@email", email)
```

```
        conn.Open()
```

```
        Return CInt(cmd.ExecuteScalar()) = 0
```

```
    End Using
```

```
End Function
```

```
Private Function IsUniquePhone(phone As String) As Boolean
```

```
    Dim sql = "SELECT COUNT(*) FROM SUPPLIER_CONTACT WHERE SUPPLIER_ID <> @sid AND  
CONTACT_NUMBER = @phone"
```

```
    Using conn As New SqlConnection(connString), cmd = New SqlCommand(sql, conn)
```

```
        cmd.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))
```

```
        cmd.Parameters.AddWithValue("@phone", phone)
```

```
        conn.Open()
```

```
        Return CInt(cmd.ExecuteScalar()) = 0
```

End Using

End Function

Private Function GenerateSalt() As String

Using rng As New RNGCryptoServiceProvider()

Dim buff(31) As Byte

rng.GetBytes(buff)

Return Convert.ToBase64String(buff)

End Using

End Function

Private Function HashPassword(pw As String, salt As String) As String

Using sha As SHA256 = SHA256.Create()

Dim combined = Encoding.UTF8.GetBytes(salt & pw)

Dim sum = sha.ComputeHash(combined)

Return Convert.ToBase64String(sum)

End Using

End Function

Private Function IsValidName(name As String) As Boolean

Return Regex.IsMatch(name.Trim(), "^[A-Za-z\s]+\$")

End Function

Private Function IsValidEmail(email As String) As Boolean

Return Regex.IsMatch(email.Trim(),

"^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$")

End Function

```
Private Function IsValidPhone(phone As String) As Boolean
```

```
    Return Regex.IsMatch(phone.Trim(), "^\\d{11}$")
```

```
End Function
```

```
Private Function IsValidPostalCode(postal As String) As Boolean
```

```
    Return Regex.IsMatch(postal.Trim(), "^\\d+$")
```

```
End Function
```

```
Private Function ValidateSupplier() As Boolean
```

```
    Dim fn = txtFirstName.Text.Trim()
```

```
    If fn = "" OrElse Not IsValidName(fn) OrElse fn.Length > 20 Then
```

```
        lblMessage.Text = "First Name is required, letters/spaces only, max 20 chars."
```

```
        Return False
```

```
    End If
```

```
    Dim ln = txtLastName.Text.Trim()
```

```
    If ln = "" OrElse Not IsValidName(ln) OrElse ln.Length > 20 Then
```

```
        lblMessage.Text = "Last Name is required, letters/spaces only, max 20 chars."
```

```
        Return False
```

```
    End If
```

```
    Dim email = txtEmail.Text.Trim()
```

```
    If email = "" OrElse Not IsValidEmail(email) Then
```

```
        lblMessage.Text = "Email is required and must be in valid format."
```

```
        Return False
```

```
    End If
```

```
Dim city = txtCity.Text.Trim()
```

```
If city = "" OrElse Not IsValidName(city) OrElse city.Length > 20 Then
```

```
    lblMessage.Text = "City is required, letters/spaces only, max 20 chars."
```

```
    Return False
```

```
End If
```

```
Dim state = txtState.Text.Trim()
```

```
If state = "" OrElse Not IsValidName(state) OrElse state.Length > 15 Then
```

```
    lblMessage.Text = "State is required, letters/spaces only, max 15 chars."
```

```
    Return False
```

```
End If
```

```
Dim pCode = txtPostalCode.Text.Trim()
```

```
If pCode = "" OrElse Not IsValidPostalCode(pCode) OrElse pCode.Length > 7 Then
```

```
    lblMessage.Text = "Postal Code is required, numeric only, max 7 digits."
```

```
    Return False
```

```
End If
```

```
Dim street = txtStreetAddress.Text.Trim()
```

```
If street = "" OrElse street.Length > 30 Then
```

```
    lblMessage.Text = "Street Address is required, max 30 chars."
```

```
    Return False
```

```
End If
```

```
Dim contact = txtContactNumber.Text.Trim()
```

```
If contact = "" OrElse Not IsValidPhone(contact) Then
```

```
        lblMessage.Text = "Contact Number is required, exactly 11 digits."

        Return False
    End If

    Dim uid = username_supplier.Text.Trim()
    If uid = "" Then

        lblMessage.Text = "User Name is required."

        Return False
    End If

    Return True
End Function
End Class
```

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="UpdateRawMaterial.aspx.vb"
Inherits="UpdateRawMaterial" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
    <title>Update Raw Material</title>
```

```
</head>
```

```
<body>
```

```
    <form id="form1" runat="server">
```

```
        <div align="center">
```

```
            <h2>Modify Raw Material Inventory</h2>
```

```
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br /><br />
```

```
<asp:DropDownList ID="ddlRawMaterials" runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlRawMaterials_SelectedIndexChanged" />
```

```
<br /><br />
```

```
<table border="0" cellpadding="10">
```

```
<tr>
```

```
<!-- Current Details Panel -->
```

```
<td valign="top">
```

```
<asp:Panel ID="pnlCurrent" runat="server" Visible="False">
```

```
<h4>Current Raw Material Details</h4>
```

```
Name: <asp:Label ID="lblCurrName" runat="server" /><br /><br />
```

```
Quantity: <asp:Label ID="lblCurrQty" runat="server" /><br /><br />
```

```
Warehouse: <asp:Label ID="lblCurrWarehouse" runat="server" /><br /><br />
```

```
Unit Price: <asp:Label ID="lblCurrPrice" runat="server" /><br /><br />
```

```
</asp:Panel>
```

```
</td>
```

```
<!-- Update Inputs Panel -->
```

```
<td valign="top">
```

```
<asp:Panel ID="pnlInputs" runat="server" Visible="False">
```

```
<h4>Update Raw Material Values</h4>
```

```
Quantity Change: <asp:TextBox ID="txtQtyChange" runat="server" /><br />
```

```
(Use positive to add, negative to subtract)<br /><br />
```

```
New Warehouse: <asp:DropDownList ID="ddlWarehouses" runat="server" /><br /><br
```

```
/>
```

```
Unit Price: <asp:TextBox ID="txtNewPrice" runat="server" /><br /><br />
```

```
        <asp:Button ID="btnUpdate" runat="server" Text="Update Material"
OnClick="btnUpdate_Click" />
```

```
    </asp:Panel>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
Imports System.Data.SqlClient
```

```
Partial Class UpdateRawMaterial
```

```
    Inherits System.Web.UI.Page
```

```
    Private ReadOnly ConnStr As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
Catalog=SCM;Integrated Security=True"
```

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

```
    If Not IsPostBack Then
```

```
        If Session("SUPPLIER_ID") Is Nothing Then Response.Redirect("Login.aspx")
```

```
        LoadRawMaterials()
```

```
    End If
```

```
End Sub
```

```
Private Sub LoadRawMaterials()
```

```
    ddlRawMaterials.Items.Clear()
```

```
    ddlRawMaterials.Items.Add(New ListItem("-- Select Material --", ""))
```

```
Dim sql = "SELECT RAW_MATERIAL_ID, RAW_MATERIAL_NAME FROM RAW_MATERIAL_T  
WHERE SUPPLIER_ID=@sid"
```

```
Using conn As New SqlConnection(ConnStr), cmd As New SqlCommand(sql, conn)
```

```
cmd.Parameters.AddWithValue("@sid", Session("SUPPLIER_ID"))
```

```
conn.Open()
```

```
Using rdr = cmd.ExecuteReader()
```

```
While rdr.Read()
```

```
ddlRawMaterials.Items.Add(New ListItem(rdr("RAW_MATERIAL_NAME").ToString(),  
rdr("RAW_MATERIAL_ID").ToString()))
```

```
End While
```

```
End Using
```

```
End Using
```

```
End Sub
```

```
Protected Sub ddlRawMaterials_SelectedIndexChanged(sender As Object, e As EventArgs)
```

```
lblMessage.Text = ""
```

```
If ddlRawMaterials.SelectedValue = "" Then
```

```
pnlCurrent.Visible = False : pnlInputs.Visible = False : Return
```

```
End If
```

```
' Fetch current details
```

```
Dim sql As String = "SELECT RAW_MATERIAL_NAME, RAW_MATERIAL_QUANTITY,  
RAW_MATERIAL_UNIT_PRICE, WAREHOUSE_ID FROM RAW_MATERIAL_T WHERE  
RAW_MATERIAL_ID=@rmid"
```

```
Using conn As New SqlConnection(ConnStr), cmd As New SqlCommand(sql, conn)
```

```
cmd.Parameters.AddWithValue("@rmid", ddlRawMaterials.SelectedValue)
```

```
conn.Open()
```

```
Using rdr = cmd.ExecuteReader()
```

```
If rdr.Read() Then
```

```

        lblCurrName.Text = rdr("RAW_MATERIAL_NAME").ToString()

        lblCurrQty.Text = rdr("RAW_MATERIAL_QUANTITY").ToString()

        lblCurrPrice.Text = rdr("RAW_MATERIAL_UNIT_PRICE").ToString()

        ' Load warehouse dropdown with capacity check

        Dim currentWid = Convert.ToInt32(rdr("WAREHOUSE_ID"))

        lblCurrWarehouse.Text = GetWarehouseName(currentWid)

        LoadWarehouses(currentWid)

    End If

End Using

End Using

pnlCurrent.Visible = True

pnlInputs.Visible = True

End Sub

Private Sub LoadWarehouses(currentWid As Integer)

    ddlWarehouses.Items.Clear()

    ' 1) Get current warehouse's name and availability

    Dim currName As String = ""

    Dim currAvail As Integer = 0

    Dim sqlCurr As String = ""

    sqlCurr &= "SELECT WAREHOUSE_NAME, "

    sqlCurr &= "(WAREHOUSE_CAPACITY - ISNULL((SELECT SUM(RAW_MATERIAL_QUANTITY) "

    sqlCurr &= "FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID = W.WAREHOUSE_ID), 0)) AS "

    sqlCurr &= "Available "

    sqlCurr &= "FROM WAREHOUSE_T W WHERE WAREHOUSE_ID = @wid"

```

```

Using conn As New SqlConnection(ConnStr)

Using cmd As New SqlCommand(sqlCurr, conn)

    cmd.Parameters.AddWithValue("@wid", currentWid)

    conn.Open()

Using rdr As SqlDataReader = cmd.ExecuteReader()

    If rdr.Read() Then

        currName = rdr("WAREHOUSE_NAME").ToString()

        currAvail = Convert.ToInt32(rdr("Available"))

    End If

End Using

End Using

End Using

```

```

ddlWarehouses.Items.Add(New ListItem("-- Keep Current -- " & currName & " (Available: " &
currAvail.ToString() & ")", currentWid.ToString()))

```

' 2) Load other active warehouses

```

Dim sql As String = ""

sql &= "SELECT WAREHOUSE_ID, WAREHOUSE_NAME, "

sql &= "(WAREHOUSE_CAPACITY - ISNULL((SELECT SUM(RAW_MATERIAL_QUANTITY) "

sql &= "FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID=W.WAREHOUSE_ID), 0)) AS "
Available "

sql &= "FROM WAREHOUSE_T W "

sql &= "WHERE OPERATIONAL_STATUS = 'Active' AND "

sql &= "(WAREHOUSE_CAPACITY - ISNULL((SELECT SUM(RAW_MATERIAL_QUANTITY) "

sql &= "FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID=W.WAREHOUSE_ID), 0)) > 0"

```

```

Using conn As New SqlConnection(ConnStr)

Using cmd As New SqlCommand(sql, conn)

    conn.Open()

    Using rdr As SqlDataReader = cmd.ExecuteReader()

        While rdr.Read()

            Dim wid As Integer = Convert.ToInt32(rdr("WAREHOUSE_ID"))

            If wid <> currentWid Then

                Dim name As String = rdr("WAREHOUSE_NAME").ToString()

                Dim avail As Integer = Convert.ToInt32(rdr("Available"))

                ddlWarehouses.Items.Add(New ListItem(name & " (Available: " & avail.ToString() & ")",
wid.ToString()))

            End If

        End While

    End Using

End Using

End Using

End Sub

```

```

Private Function GetWarehouseName(wid As Integer) As String

    Dim name As String = ""

    Dim sql = "SELECT WAREHOUSE_NAME FROM WAREHOUSE_T WHERE
WAREHOUSE_ID=@wid"

    Using conn As New SqlConnection(ConnStr), cmd As New SqlCommand(sql, conn)

        cmd.Parameters.AddWithValue("@wid", wid)

        conn.Open()

        name = cmd.ExecuteScalar().ToString()

    End Using

```

Return name

End Function

Protected Sub btnUpdate_Click(sender As Object, e As EventArgs)

lblMessage.ForeColor = Drawing.Color.Red

lblMessage.Text = ""

If ddlRawMaterials.SelectedValue = "" Then Return

Dim rmid = Convert.ToInt32(ddlRawMaterials.SelectedValue)

Dim delta As Integer

If Not Integer.TryParse(txtQtyChange.Text.Trim(), delta) Then

lblMessage.Text = "Enter a valid integer for quantity change."

Return

End If

Dim newPrice As Decimal

If Not Decimal.TryParse(txtNewPrice.Text.Trim(), newPrice) Then

lblMessage.Text = "Enter a valid unit price."

Return

End If

Dim newWid = Convert.ToInt32(ddlWarehouses.SelectedValue)

Using conn As New SqlConnection(ConnStr)

conn.Open()

Dim trans = conn.BeginTransaction()

Try

' Get current quantity & warehouse

Dim currQty As Integer, currWid As Integer

```
Dim sqlGet = "SELECT RAW_MATERIAL_QUANTITY, WAREHOUSE_ID FROM  
RAW_MATERIAL_T WHERE RAW_MATERIAL_ID=@rmid"
```

```
Using cmdGet As New SqlCommand(sqlGet, conn, trans)
```

```
cmdGet.Parameters.AddWithValue("@rmid", rmid)
```

```
Using rdr = cmdGet.ExecuteReader()
```

```
rdr.Read()
```

```
currQty = Convert.ToInt32(rdr("RAW_MATERIAL_QUANTITY"))
```

```
currWid = Convert.ToInt32(rdr("WAREHOUSE_ID"))
```

```
End Using
```

```
End Using
```

```
' Compute target available capacity
```

```
If newWid <> currWid Then
```

```
Dim sqlCap As String
```

```
sqlCap = "SELECT (WAREHOUSE_CAPACITY - ISNULL((SELECT  
SUM(RAW_MATERIAL_QUANTITY) "
```

```
sqlCap &= "FROM RAW_MATERIAL_T WHERE WAREHOUSE_ID=@wid),0)) FROM  
WAREHOUSE_T WHERE WAREHOUSE_ID=@wid"
```

```
Using cmdCap As New SqlCommand(sqlCap, conn, trans)
```

```
cmdCap.Parameters.AddWithValue("@wid", newWid)
```

```
Dim avail = Convert.ToInt32(cmdCap.ExecuteScalar())
```

```
If currQty + delta > avail Then
```

```
Throw New Exception(String.Format("Insufficient capacity in selected warehouse.  
Avail: {0}", avail.ToString()))
```

```
End If
```

```
End Using
```

```
End If
```

```
' Apply update
```

```

Dim sqlUpd As String

sqlUpd = "UPDATE RAW_MATERIAL_T SET "

sqlUpd &= "RAW_MATERIAL_QUANTITY = RAW_MATERIAL_QUANTITY + @delta, "

sqlUpd &= "RAW_MATERIAL_UNIT_PRICE = @price, "

sqlUpd &= "WAREHOUSE_ID = @wid "

sqlUpd &= "WHERE RAW_MATERIAL_ID = @rmid"

Using cmdUpd As New SqlCommand(sqlUpd, conn, trans)

    cmdUpd.Parameters.AddWithValue("@delta", delta)

    cmdUpd.Parameters.AddWithValue("@price", newPrice)

    cmdUpd.Parameters.AddWithValue("@wid", newWid)

    cmdUpd.Parameters.AddWithValue("@rmid", rmid)

    cmdUpd.ExecuteNonQuery()

End Using

trans.Commit()

lblMessage.ForeColor = Drawing.Color.Green

lblMessage.Text = "Raw material updated successfully."

Catch ex As Exception

    trans.Rollback()

    lblMessage.Text = "Error: " & ex.Message

End Try

End Using

End Sub

End Class
.....

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="_Default" %>

```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title>SCM Home</title>

</head>

<body>

    <form id="form1" runat="server">

        <div align="center">

            <h2>SCM System</h2>

            <asp:Label ID="lblWelcome" runat="server" Text="Welcome to SCM" Font-Bold="true" Font-Size="Large" ForeColor="DarkBlue" /><br /><br />

            <div>

                <asp:HyperLink ID="lnkLogin" runat="server" NavigateUrl="~/login.aspx" ForeColor="Blue" Font-Underline="true">Log In</asp:HyperLink><br /><br />

            </div>

            <div>

                <asp:HyperLink ID="lnkSignup" runat="server" NavigateUrl="~/signup.aspx" ForeColor="Blue" Font-Underline="true">Sign Up</asp:HyperLink><br /><br />

            </div>

            <p>--- SCM 2025 ---</p>

        </div>

    </form>

</body>

</html>
```

Imports System.Data

Partial Class _Default

Inherits System.Web.UI.Page

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

 ' Page has loaded successfully

End Sub

End Class

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Login.aspx.vb" Inherits="Login" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

 <title>SCM Login</title>

</head>

<body>

 <form id="form1" runat="server">

 <div align="center">

 <h2>SCM Login</h2>

 <asp:Label ID="lblWelcome" runat="server" Text="Sign in to SCM System" Font-Bold="true" Font-Size="Large" ForeColor="DarkBlue" />

 <div>

 <asp:Label ID="lblUserID" runat="server" Text="User ID" Font-Bold="true" />

 <asp:TextBox ID="username" runat="server" Width="200px" />

 <asp:RequiredFieldValidator ID="rfvUsername" runat="server" ControlToValidate="username"

```
ErrorMessage="User ID is required." ForeColor="Red"
Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblPassword" runat="server" Text="Password" Font-Bold="true" /><br />
```

```
<asp:TextBox ID="Password" runat="server" TextMode="Password" Width="200px" /><br />
```

```
<asp:RequiredFieldValidator ID="rfvPassword" runat="server"
ControlToValidate="Password"
```

```
ErrorMessage="Password is required." ForeColor="Red"
Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
<div>
```

```
<asp:Button ID="btnlogin" runat="server" Text="Sign In" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="errmsg" runat="server" ForeColor="Red" /><br />
```

```
<asp:ValidationSummary ID="valSummary" runat="server" ForeColor="Red"
ShowMessageBox="True" ShowSummary="True" /><br />
```

```
</div>
```

```
<div>
```

```
<asp:HyperLink ID="lnkSignup" runat="server" NavigateUrl="~/signup.aspx"
ForeColor="Blue" Font-Underline="true">Create New Account</asp:HyperLink><br /><br />
```

```
</div>
```

```
<p>--- SCM 2025 ---</p>
```

```

        </div>

    </form>

</body>

</html>

Imports System.Data

Imports System.Data.SqlClient

Imports System.Security.Cryptography

Imports System.Text

Partial Class Login

    Inherits System.Web.UI.Page

    Dim connString As String = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial
    Catalog=SCM;Integrated Security=True"

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

        ' If already logged in, redirect based on role

        Session.Clear()

        If Session("UserID") Or Session("CUSTOMER_ID") Or Session("SUPPLIER_ID") IsNot Nothing
        Then

            Dim role As String = Session("RoleName")

            Response.Redirect(role & ".aspx")

        End If

    End Sub

    Protected Sub btnlogin_Click(sender As Object, e As EventArgs) Handles btnlogin.Click

        Dim uname As String = username.Text

        Dim pwd As String = Password.Text

```

```
If String.IsNullOrEmpty(uname) OrElse String.IsNullOrEmpty(pwd) Then  
    errormsg.Text = "Username and password are required."  
  
    Return  
End If
```

```
Dim query2 As String
```

```
If uname = "admin" Then  
    query2 = "SELECT * "  
    query2 &= "FROM ADMIN_T WHERE ADMIN_ID = '1'"
```

```
Dim con2 As New SqlConnection(connString)  
Dim cmd2 As New SqlCommand(query2, con2)
```

```
Dim dr2 As SqlDataReader  
con2.Open()  
dr2 = cmd2.ExecuteReader()  
If dr2.Read() Then
```

```
    Dim storedHash2 = dr2("PASSWORD_HASH").ToString()
```

```
    Dim salt2 = dr2("SALT").ToString()
```

```
    'Response.Write("Hi")
```

```
If VerifyPassword(pwd, salt2, storedHash2) Then
```

```
    Session("UserID") = dr2("ADMIN_ID")
```

```
    Session("RoleName") = "Admin"
```

```
    Session("username") = "Admin"
```

```
    Response.Redirect("Admin.aspx")
```

```
Else
End If
End If
End If
```

```
Dim query As String
query = "SELECT * "
query &= "FROM USER_T U JOIN ROLE_T R ON U.ROLE_ID = R.ROLE_ID "
query &= "WHERE U.USERNAME = @u"
```

```
Dim con As New SqlConnection(connString)
Dim cmd As New SqlCommand(query, con)
cmd.Parameters.AddWithValue("@u", uname)
```

```
Dim dr As SqlDataReader
con.Open()
dr = cmd.ExecuteReader()
If dr.Read() Then
    Dim storedHash = dr("PASSWORD_HASH").ToString()
    Dim salt = dr("SALT").ToString()
    Dim roleName = dr("ROLE_NAME").Trim().ToString()
    If IsDBNull(dr("SUPPLIER_ID")) Then
        Session("CUSTOMER_ID") = dr("CUSTOMER_ID")
    Else
        Session("SUPPLIER_ID") = dr("SUPPLIER_ID")
    End If
End If
```

```

End If

Dim id = CInt(dr("USER_ID"))

If VerifyPassword(pwd, salt, storedHash) Then

    Session("username") = id

    Session("RoleName") = roleName

    Response.Redirect(roleName & ".aspx")

Else

    errmsg.Text = "Invalid credentials."

End If

Return

End If

'errmsg.Text = "Invalid credentials."

End Sub

'Private Function aVerifyPassword(plain As String, salt As String, hash As String) As Boolean
' re-compute SHA256(salt + plain) and compare
'Dim sha = SHA256.Create()
'Dim bytes = sha.ComputeHash(Encoding.UTF8.GetBytes(salt & plain))
'Dim sb As New StringBuilder()
'For Each b In bytes : sb.Append(b.ToString("x2")) : Next
'Return sb.ToString() = hash
'End Function

Private Function VerifyPassword(plain As String, salt As String, hash As String) As Boolean

    Using sha As SHA256 = SHA256.Create()

        Dim combined = Encoding.UTF8.GetBytes(salt & plain)

```

```

        Dim computedHash = sha.ComputeHash(combined)

        Dim computedHashBase64 = Convert.ToBase64String(computedHash)

        Return computedHashBase64 = hash

    End Using

End Function

End Class

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Signup.aspx.vb" Inherits="Signup"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

    <title>SCM Sign Up</title>

</head>

<body>

    <form id="form1" runat="server">

        <div align="center">

            <h2>SCM Sign Up</h2>

            <asp:Label ID="lblWelcome" runat="server" Text="Register New Account" Font-Bold="true"
Font-Size="Large" ForeColor="DarkBlue" /><br /><br />

            <div>

                <p id="question1" runat="server">Choose User Type</p>

                <asp:RadioButtonList ID="UserTypeRadio" runat="server" AutoPostBack="true"
OnSelectedIndexChanged="UserTypeRadio_SelectedIndexChanged">

                    <asp:ListItem Text="Supplier" Value="Supplier" />

                    <asp:ListItem Text="Customer" Value="Customer" />


```

```
</asp:RadioButtonList><br />

</div>

<!-- Supplier Form -->

<asp:Panel ID="SupplierPanel" runat="server" Visible="False">

    <h3>Supplier Sign Up</h3>

    <div>

        <asp:Label ID="lblFirstName" runat="server" Text="First Name" Font-Bold="true" /><br />

        <asp:TextBox ID="txtFirstName" runat="server" Width="200px" Placeholder="First Name"
/><br /><br />

    </div>

    <div>

        <asp:Label ID="lblLastName" runat="server" Text="Last Name" Font-Bold="true" /><br />

        <asp:TextBox ID="txtLastName" runat="server" Width="200px" Placeholder="Last Name"
/><br /><br />

    </div>

    <div>

        <asp:Label ID="lblEmail" runat="server" Text="Email" Font-Bold="true" /><br />

        <asp:TextBox ID="txtEmail" runat="server" Width="200px" Placeholder="Email" /><br
/><br />

    </div>

    <div>

        <asp:Label ID="lblCity" runat="server" Text="City" Font-Bold="true" /><br />

        <asp:TextBox ID="txtCity" runat="server" Width="200px" Placeholder="City" /><br /><br />

    </div>

</asp:Panel>

</div>
```

</div>

<div>

<asp:Label ID="lblState" runat="server" Text="State" Font-Bold="true" />

<asp:TextBox ID="txtState" runat="server" Width="200px" Placeholder="State" />

</div>

<div>

<asp:Label ID="lblPostalCode" runat="server" Text="Postal Code" Font-Bold="true" />

<asp:TextBox ID="txtPostalCode" runat="server" Width="200px" Placeholder="Postal Code" />

</div>

<div>

<asp:Label ID="lblStreetAddress" runat="server" Text="Street Address" Font-Bold="true" />

<asp:TextBox ID="txtStreetAddress" runat="server" Width="200px" Placeholder="Street Address" />

</div>

<div>

<asp:Label ID="lblContactNumber" runat="server" Text="Contact Number" Font-Bold="true" />

<asp:TextBox ID="txtContactNumber" runat="server" Width="200px" Placeholder="Contact Number" />

</div>

<div>

```
<asp:Label ID="lblUsernameSupplier" runat="server" Text="User ID" Font-Bold="true" /><br />
```

```
<asp:TextBox ID="username_supplier" runat="server" Width="200px" Placeholder="User ID" /><br />
```

```
<asp:RequiredFieldValidator ID="rfvUsernameSupplier" runat="server" ControlToValidate="username_supplier"
```

```
ErrorMessage="User ID is required for supplier." ForeColor="Red" Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblPasswordSupplier" runat="server" Text="Password" Font-Bold="true" /><br />
```

```
<asp:TextBox ID="password_supplier" runat="server" TextMode="Password" Width="200px" Placeholder="Password" /><br />
```

```
<asp:RequiredFieldValidator ID="rfvPasswordSupplier" runat="server" ControlToValidate="password_supplier"
```

```
ErrorMessage="Password is required for supplier." ForeColor="Red" Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
</asp:Panel>
```

```
<!-- Customer Form -->
```

```
<asp:Panel ID="CustomerPanel" runat="server" Visible="False">
```

```
<h3>Customer Sign Up</h3>
```

```
<div>
```

```
<asp:Label ID="lblCustomerName" runat="server" Text="Name" Font-Bold="true" /><br />
```

```
<asp:TextBox ID="txtCustomerName" runat="server" Width="200px" Placeholder="Customer Name" /><br /><br />
```

</div>

<div>

<asp:Label ID="lblCustomerContact" runat="server" Text="Contact Number" Font-Bold="true" />

<asp:TextBox ID="txtCustomerContact" runat="server" Width="200px" Placeholder="Contact Number" />

</div>

<div>

<asp:Label ID="lblCustomerEmail" runat="server" Text="Email" Font-Bold="true" />

<asp:TextBox ID="txtCustomerEmail" runat="server" Width="200px" Placeholder="Email" />

</div>

<div>

<asp:Label ID="lblCustomerCity" runat="server" Text="City" Font-Bold="true" />

<asp:TextBox ID="txtCustomerCity" runat="server" Width="200px" Placeholder="City" />

</div>

<div>

<asp:Label ID="lblCustomerState" runat="server" Text="State" Font-Bold="true" />

<asp:TextBox ID="txtCustomerState" runat="server" Width="200px" Placeholder="State" />

</div>

<div>

```
<asp:Label ID="lblCustomerPostalCode" runat="server" Text="Postal Code" Font-  
Bold="true" /><br />
```

```
<asp:TextBox ID="txtCustomerPostalCode" runat="server" Width="200px"  
Placeholder="Postal Code" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblCustomerStreet" runat="server" Text="Street Address" Font-  
Bold="true" /><br />
```

```
<asp:TextBox ID="txtCustomerStreet" runat="server" Width="200px" Placeholder="Street  
Address" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblCustomerID" runat="server" Text="User ID" Font-Bold="true" /><br />
```

```
<asp:TextBox ID="txtCustomerID" runat="server" Width="200px" Placeholder="User ID"  
/><br />
```

```
<asp:RequiredFieldValidator ID="rfvCustomerID" runat="server"  
ControlToValidate="txtCustomerID"
```

```
ErrorMessage="User ID is required for customer." ForeColor="Red"  
Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblCustomerPwd" runat="server" Text="Password" Font-Bold="true" /><br  
/>
```

```
<asp:TextBox ID="txtCustomerPwd" runat="server" TextMode="Password" Width="200px"  
Placeholder="Password" /><br />
```

```
<asp:RequiredFieldValidator ID="rfvCustomerPwd" runat="server"  
ControlToValidate="txtCustomerPwd"
```

```
ErrorMessage="Password is required for customer." ForeColor="Red"
Display="Dynamic">*</asp:RequiredFieldValidator><br />
```

```
</div>
```

```
</asp:Panel>
```

```
<div>
```

```
<asp:Button ID="btnSubmit" runat="server" Text="Sign Up" Visible="False"
OnClick="btnSubmit_Click" /><br /><br />
```

```
</div>
```

```
<div>
```

```
<asp:Label ID="lblMessage" runat="server" ForeColor="Red" /><br />
```

```
<asp:ValidationSummary ID="valSummary" runat="server" ForeColor="Red"
ShowMessageBox="True" ShowSummary="True" /><br />
```

```
</div>
```

```
<div>
```

```
<asp:HyperLink ID="lnkLogin" runat="server" NavigateUrl="~/Login.aspx" ForeColor="Blue"
Font-Underline="true">Already Registered? Log In</asp:HyperLink><br /><br />
```

```
</div>
```

```
<p>--- SCM 2025 ---</p>
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

```
Imports System.Data
```

```
Imports System.Data.SqlClient
```

```
Imports System.Security.Cryptography
```

Imports System.Text

Imports System.Text.RegularExpressions

Partial Class Signup

Inherits System.Web.UI.Page

' ... (keep all the helper methods: IsValidName, IsValidEmail, IsValidPhone,
' IsValidPostalCode, ValidatePasswordComplexity) ...

Protected Sub UserTypeRadio_SelectedIndexChanged(

sender As Object, e As EventArgs) _

Handles UserTypeRadio.SelectedIndexChanged

' Show the right panel, hide the question & radio

SupplierPanel.Visible = (UserTypeRadio.SelectedValue = "Supplier")

CustomerPanel.Visible = (UserTypeRadio.SelectedValue = "Customer")

UserTypeRadio.Visible = False

question1.Visible = False

btnSubmit.Visible = True

lblMessage.Text = ""

End Sub

'– your existing validation helpers here (IsValidName, etc.) –

' Generate a 32-byte random salt, base-64-encoded

Private Function GenerateSalt() As String

```

Dim rng As New RNGCryptoServiceProvider()

Dim buff(31) As Byte

rng.GetBytes(buff)

Return Convert.ToBase64String(buff)

End Function

```

```

' Hash = SHA256( salt + password ), base-64-encoded

Private Function HashPassword(pw As String, salt As String) As String

    Using sha As SHA256 = SHA256.Create()

        Dim combined = Encoding.UTF8.GetBytes(salt & pw)

        Dim sum = sha.ComputeHash(combined)

        Return Convert.ToBase64String(sum)

    End Using

End Function

```

```

' Returns True if no row in USER_T has this username or this password_hash

Private Function IsUsernameUnique(username As String, con As SqlConnection, tran As
SqlConnection) As Boolean

    Dim cnt As Integer = 0

    Const sql As String = "SELECT COUNT(*) FROM USER_T WHERE USERNAME = @u"

    Using cmd As New SqlCommand(sql, con, tran)

        cmd.Parameters.AddWithValue("@u", username)

        cnt = CInt(cmd.ExecuteScalar())

        Response.Write(cnt & "<br/>")

        Return (cnt = 0)

    End Using

End Function

```

```
' Find next integer ID by MAX(...)
```

```
Private Function GetNextId(tableName As String, idCol As String, con As SqlConnection, tran As  
SqlTransaction) As Integer
```

```
' Use String.Format (or plain concatenation) instead of interpolation
```

```
Dim sql As String = String.Format("SELECT ISNULL(MAX({0}), 0) + 1 FROM {1}", idCol,  
tableName)
```

```
Using cmd As New SqlCommand(sql, con, tran)
```

```
Return Convert.ToInt32(cmd.ExecuteScalar())
```

```
End Using
```

```
End Function
```

```
Protected Sub btnSubmit_Click(
```

```
sender As Object,
```

```
e As EventArgs) _
```

```
Handles btnSubmit.Click
```

```
lblMessage.Text = ""
```

```
If UserTypeRadio.SelectedValue <> "Supplier" AndAlso UserTypeRadio.SelectedValue <>  
"Customer" Then
```

```
lblMessage.Text = "Please select a user type."
```

```
Return
```

```
End If
```

' Validate fields first

If UserTypeRadio.SelectedValue = "Supplier" AndAlso Not ValidateSupplier() Then Exit Sub

If UserTypeRadio.SelectedValue = "Customer" AndAlso Not ValidateCustomer() Then Exit Sub

Dim connStr = "Data Source=DESKTOP-445IJRD\SQLEXPRESS;Initial Catalog=SCM;Integrated Security=True"

Using con As New SqlConnection(connStr)

con.Open()

Using tran = con.BeginTransaction()

Try

' 1) Prepare username & password

Dim username As String

Dim plainPwd As String

Dim roleId As Integer

If UserTypeRadio.SelectedValue = "Supplier" Then

username = username_supplier.Text.Trim()

plainPwd = password_supplier.Text

roleId = 2 ' assume 2 = Supplier in ROLE_T

Else

username = txtCustomerID.Text.Trim()

plainPwd = txtCustomerPwd.Text

roleId = 3 ' assume 3 = Customer

End If

' 2) Salt & hash

Dim salt = GenerateSalt()

```
Dim pwdHash = HashPassword(plainPwd, salt)
```

```
' 3) Uniqueness check
```

```
If Not IsUsernameUnique(username, con, tran) Then
```

```
    lblMessage.Text = "Username already exists; choose another."
```

```
    Return
```

```
End If
```

```
Response.Write(999)
```

```
' 4) Insert into Supplier or Customer tables
```

```
Dim newEntityId As Integer
```

```
If UserTypeRadio.SelectedValue = "Supplier" Then
```

```
    ' — create new supplier ID
```

```
    newEntityId = GetNextId("SUPPLIER_T", "SUPPLIER_ID", con, tran)
```

```
    ' — insert SUPPLIER_T
```

```
    Dim sqlSup = "INSERT INTO SUPPLIER_T (SUPPLIER_ID, SUPPLIER_FIRST_NAME,  
SUPPLIER_LAST_NAME, SUPPLIER_EMAIL, SUPPLIER_CITY, SUPPLIER_STATE, "
```

```
    sqlSup &= " SUPPLIER_POSTAL_CODE, SUPPLIER_STREET_ADDRESS) VALUES (@id,  
@fn, @ln, @em, @city, @st, @postal, @street)"
```

```
Using cmdSup As New SqlCommand(sqlSup, con, tran)
```

```
    cmdSup.Parameters.AddWithValue("@id", newEntityId)
```

```
    cmdSup.Parameters.AddWithValue("@fn", txtFirstName.Text.Trim())
```

```
    cmdSup.Parameters.AddWithValue("@ln", txtLastName.Text.Trim())
```

```
    cmdSup.Parameters.AddWithValue("@em", txtEmail.Text.Trim())
```

```
    cmdSup.Parameters.AddWithValue("@city", txtCity.Text.Trim())
```

```
    cmdSup.Parameters.AddWithValue("@st", txtState.Text.Trim())
```

```
cmdSup.Parameters.AddWithValue("@postal", txtPostalCode.Text.Trim())  
cmdSup.Parameters.AddWithValue("@street", txtStreetAddress.Text.Trim())  
cmdSup.ExecuteNonQuery()
```

End Using

' — insert SUPPLIER_CONTACT

```
Dim sqlContact = "INSERT INTO SUPPLIER_CONTACT (SUPPLIER_ID,  
CONTACT_NUMBER) VALUES (@id, @phone)"
```

Using cmdC As New SqlCommand(sqlContact, con, tran)

```
cmdC.Parameters.AddWithValue("@id", newEntityId)  
cmdC.Parameters.AddWithValue("@phone", txtContactNumber.Text.Trim())  
cmdC.ExecuteNonQuery()
```

End Using

Else

' — create new customer ID

```
newEntityId = GetNextId("CUSTOMER_T", "CUSTOMER_ID", con, tran)
```

' — insert CUSTOMER_T

```
Dim sqlCust = "INSERT INTO CUSTOMER_T"
```

```
sqlCust &= " (CUSTOMER_ID, CUSTOMER_NAME, CUSTOMER_CONTACT,  
CUSTOMER_EMAIL,"
```

```
sqlCust &= " CUSTOMER_CITY, CUSTOMER_STATE, CUSTOMER_POSTALCODE,  
CUSTOMER_STREET_ADDRESS, CUSTOMER_STATUS)"
```

```
sqlCust &= " VALUES (@id, @name, @phone, @em, @city, @st, @postal, @street,  
@status)"
```

Using cmdCust As New SqlCommand(sqlCust, con, tran)

```

cmdCust.Parameters.AddWithValue("@id", newEntityId)
cmdCust.Parameters.AddWithValue("@name", txtCustomerName.Text.Trim())
cmdCust.Parameters.AddWithValue("@phone", txtCustomerContact.Text.Trim())
cmdCust.Parameters.AddWithValue("@em", txtCustomerEmail.Text.Trim())
cmdCust.Parameters.AddWithValue("@city", txtCustomerCity.Text.Trim())
cmdCust.Parameters.AddWithValue("@st", txtCustomerState.Text.Trim())
cmdCust.Parameters.AddWithValue("@postal", txtCustomerPostalCode.Text.Trim())
cmdCust.Parameters.AddWithValue("@street", txtCustomerStreet.Text.Trim())
cmdCust.Parameters.AddWithValue("@status", "Active")
cmdCust.ExecuteNonQuery()

```

End Using

End If

' 5) Insert into USER_T

```
Dim newUserId = GetNextId("USER_T", "USER_ID", con, tran)
```

```
Dim sqlUser = " INSERT INTO USER_T"
```

```
sqlUser &= "(USER_ID, USERNAME, PASSWORD_HASH, SALT, ROLE_ID, CUSTOMER_ID,
SUPPLIER_ID)"
```

```
sqlUser &= " VALUES (@uid, @uname, @phash, @salt, @rid, " &
If(UserTypeRadio.SelectedValue = "Customer", "@cid", "NULL") & ", " &
If(UserTypeRadio.SelectedValue = "Supplier", "@sid", "NULL") & ")"
```

```
Using cmdU As New SqlCommand(sqlUser, con, tran)
```

```

cmdU.Parameters.AddWithValue("@uid", newUserId)
cmdU.Parameters.AddWithValue("@uname", username)
cmdU.Parameters.AddWithValue("@phash", pwdHash)
cmdU.Parameters.AddWithValue("@salt", salt)

```

```

        cmdU.Parameters.AddWithValue("@rid", roleId)

        If UserTypeRadio.SelectedValue = "Supplier" Then

            cmdU.Parameters.AddWithValue("@sid", newEntityId)

        Else

            cmdU.Parameters.AddWithValue("@cid", newEntityId)

        End If

        cmdU.ExecuteNonQuery()

    End Using

    tran.Commit()

    lblMessage.ForeColor = Drawing.Color.Green

    lblMessage.Text = "Registration successful!"

Catch ex As Exception

    tran.Rollback()

    lblMessage.ForeColor = Drawing.Color.Red

    lblMessage.Text = "Error: " & ex.Message

End Try

End Using

End Using

End Sub

Private Function IsValidName(ByVal name As String) As Boolean

    Return Regex.IsMatch(name.Trim(), "^[A-Za-z\s]+$")

End Function

```

' — 3. Email Validation: strict regex —————

Private Function IsValidEmail(ByVal email As String) As Boolean

```
Return Regex.IsMatch(  
    email.Trim(),  
    "[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$"  
)
```

End Function

' — 4. Phone: exactly 11 digits —————

Private Function IsValidPhone(ByVal phone As String) As Boolean

```
Return Regex.IsMatch(phone.Trim(), "^d{11}$")
```

End Function

' — 5. Postal Code: numeric only —————

Private Function IsValidPostalCode(ByVal postal As String) As Boolean

```
Return Regex.IsMatch(postal.Trim(), "^d+$")
```

End Function

' (Optional) Reuse your existing password-complexity checker here:

Private Function ValidatePasswordComplexity(ByVal pw As String) As Boolean

```
If pw.Trim().Length < 10 Then Return False  
  
Dim special = Regex.Matches(pw, "[^A-Za-z0-9]").Count  
Dim digits = Regex.Matches(pw, "\d").Count  
Dim letters = Regex.Matches(pw, "[A-Za-z]").Count  
  
Return special >= 4 AndAlso digits >= 4 AndAlso letters >= 2
```

End Function

Private Function ValidateSupplier() As Boolean

Dim fn = txtFirstName.Text.Trim()

If fn = "" OrElse Not IsValidName(fn) OrElse fn.Length > 20 Then

lblMessage.Text = "First Name is required, letters/spaces only, max 20 chars."

Return False

End If

Dim ln = txtLastName.Text.Trim()

If ln = "" OrElse Not IsValidName(ln) OrElse ln.Length > 20 Then

lblMessage.Text = "Last Name is required, letters/spaces only, max 20 chars."

Return False

End If

Dim email = txtEmail.Text.Trim()

If email = "" OrElse Not IsValidEmail(email) Then

lblMessage.Text = "Email is required and must be in valid format."

Return False

End If

Dim city = txtCity.Text.Trim()

If city = "" OrElse Not IsValidName(city) OrElse city.Length > 20 Then

lblMessage.Text = "City is required, letters/spaces only, max 20 chars."

Return False

End If

Dim state = txtState.Text.Trim()

```
If state = "" OrElse Not IsValidName(state) OrElse state.Length > 15 Then  
    lblMessage.Text = "State is required, letters/spaces only, max 15 chars."  
    Return False  
End If
```

```
Dim pCode = txtPostalCode.Text.Trim()  
If pCode = "" OrElse Not IsValidPostalCode(pCode) OrElse pCode.Length > 7 Then  
    lblMessage.Text = "Postal Code is required, numeric only, max 7 digits."  
    Return False  
End If
```

```
Dim street = txtStreetAddress.Text.Trim()  
If street = "" OrElse street.Length > 30 Then  
    lblMessage.Text = "Street Address is required, max 30 chars."  
    Return False  
End If
```

```
Dim contact = txtContactNumber.Text.Trim()  
If contact = "" OrElse Not IsValidPhone(contact) Then  
    lblMessage.Text = "Contact Number is required, exactly 11 digits."  
    Return False  
End If
```

```
Dim uid = username_supplier.Text.Trim()  
If uid = "" Then  
    lblMessage.Text = "User ID is required."  
    Return False
```

End If

Dim pwd = password_supplier.Text

If pwd.Trim() = "" OrElse Not ValidatePasswordComplexity(pwd) Then

 lblMessage.Text = "Password is required and must be ≥10 chars with ≥4 special, ≥4 digits, ≥2 letters."

 Return False

End If

Return True

End Function

Private Function ValidateCustomer() As Boolean

 Dim cname = txtCustomerName.Text.Trim()

 If cname = "" OrElse Not IsValidName(cname) OrElse cname.Length > 40 Then

 lblMessage.Text = "Customer Name is required, letters/spaces only, max 40 chars."

 Return False

 End If

 Dim contact = txtCustomerContact.Text.Trim()

 If contact = "" OrElse Not IsValidPhone(contact) Then

 lblMessage.Text = "Customer Contact is required, exactly 11 digits."

 Return False

 End If

 Dim email = txtCustomerEmail.Text.Trim()

 If email = "" OrElse Not IsValidEmail(email) Then

 lblMessage.Text = "Customer Email is required and must be valid."

Return False

End If

Dim city = txtCustomerCity.Text.Trim()

If city = "" OrElse Not IsValidName(city) OrElse city.Length > 20 Then

lblMessage.Text = "City is required, letters/spaces only, max 20 chars."

Return False

End If

Dim state = txtCustomerState.Text.Trim()

If state = "" OrElse Not IsValidName(state) OrElse state.Length > 50 Then

lblMessage.Text = "State is required, letters/spaces only, max 50 chars."

Return False

End If

Dim cPostal = txtCustomerPostalCode.Text.Trim()

If cPostal = "" OrElse Not IsValidPostalCode(cPostal) OrElse cPostal.Length > 10 Then

lblMessage.Text = "Postal Code is required, numeric only, max 10 digits."

Return False

End If

Dim street = txtCustomerStreet.Text.Trim()

If street = "" OrElse street.Length > 30 Then

lblMessage.Text = "Street Address is required, max 30 chars."

Return False

End If

```
Dim uid = txtCustomerID.Text.Trim()
```

```
If uid = "" Then
```

```
    lblMessage.Text = "User ID is required."
```

```
    Return False
```

```
End If
```

```
Dim pwd = txtCustomerPwd.Text
```

```
If pwd.Trim() = "" OrElse Not ValidatePasswordComplexity(pwd) Then
```

```
    lblMessage.Text = "Password is required and must be ≥10 chars with ≥4 special, ≥4 digits, ≥2 letters."
```

```
    Return False
```

```
End If
```

```
Return True
```

```
End Function
```

```
End Class
```