

Stationery Store Management System

By

M. Umer Riaz

To

Mr. Irfan Hameed

May 22, 2025



Department of Computer and Information Sciences

PAKISTAN INSTITUTE OF ENGINEERING AND APPLIED SCIENCES

NILORE, ISLAMABAD 45650

Code Implementation

Login.aspx.vb

```
Imports System.Data.SqlClient
Imports System.Security.Cryptography
Imports System.Text

Partial Class Login
    Inherits System.Web.UI.Page
    Protected Sub btnLogin_Click(sender As Object, e As EventArgs)

        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim username As String = txtUsername.Text.Trim()
        Dim password As String = txtPassword.Text.Trim()
        Dim hashedPwd As String = GetHash(password)

        Using conn As New SqlConnection(connStr)
            Dim cmd As New SqlCommand("SELECT User_ID, Role_ID FROM
User_t WHERE Username=@uname AND Password_Hash=@pwd", conn)
            cmd.Parameters.AddWithValue("@uname", username)
            cmd.Parameters.AddWithValue("@pwd", hashedPwd)

            conn.Open()
            Dim reader As SqlDataReader = cmd.ExecuteReader()
            If reader.Read() Then
                Dim userId = reader("User_ID")
                Dim roleId = reader("Role_ID")
                Session("UserID") = userId
                Session("RoleID") = roleId
                RecordLogin(userId, "Success")
                Response.Redirect("Dashboard.aspx")
            Else
                lblError.Text = "Invalid credentials."
                lblError.Visible = True
            End If
        End Using
    End Sub

    Private Function GetHash(input As String) As String
        Using sha As SHA256 = SHA256.Create()
            Dim bytes = Encoding.UTF8.GetBytes(input)
            Dim hash = sha.ComputeHash(bytes)
            Return Convert.ToBase64String(hash)
        End Using
    End Function

    Private Sub RecordLogin(userId As Object, status As String)
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Using conn As New SqlConnection(connStr)
```

```

        Dim cmd As New SqlCommand("INSERT INTO Login_Audit
(User_ID, Login_Time, Status) VALUES (@uid, GETDATE(), @status)",
conn)
        cmd.Parameters.AddWithValue("@uid", If(userId,
DBNull.Value))
        cmd.Parameters.AddWithValue("@status", status)
        conn.Open()
        cmd.ExecuteNonQuery()
    End Using
End Sub
End Class

```

Logout.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Security.Cryptography

Partial Class Logout
    Inherits System.Web.UI.Page
    Protected Sub Page_Load(sender As Object, e As EventArgs)
        Dim userId = Session("UserID")
        If userId IsNot Nothing Then
            Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
            Using conn As New SqlConnection(connStr)
                Dim cmd As New SqlCommand("UPDATE TOP (1) Login_Audit
SET Logout_Time = GETDATE() WHERE User_ID = @uid AND Logout_Time IS
NULL ORDER BY Login_Time DESC", conn)
                cmd.Parameters.AddWithValue("@uid", userId)
                conn.Open()
                cmd.ExecuteNonQuery()
            End Using
        End If
        Session.Clear()
        Response.Redirect("Login.aspx")
    End Sub
End Class

```

Dashboard.aspx.vb

```

Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            If Session("UserID") Is Nothing OrElse Session("RoleID")
Is Nothing Then
                Response.Redirect("Login.aspx")
            Else

```

```

        Dim roleId As Integer =
Convert.ToInt32(Session("RoleID"))
        litCards.Text = GetDashboardCards(roleId)
    End If
End If
End Sub

Private Function GetDashboardCards(roleId As Integer) As String
    Select Case roleId
        Case 1 ' Admin
            Return AdminCards()
        Case 2 ' Manager
            Return ManagerCards()
        Case 3 ' Salesperson
            Return SalespersonCards()
        Case Else
            Return "<div class='card'><h3>Unauthorized
Role</h3></div>"
    End Select
End Function

Private Function AdminCards() As String
    Return "<div class='card'><h3>Manage Users</h3><a
href='Users.aspx'>Go</a></div>" &
        "<div class='card'><h3>Manage Employees</h3><a
href='Employees.aspx'>Go</a></div>" &
        "<div class='card'><h3>Audit Logs</h3><a
href='LoginAudit.aspx'>Go</a></div>"
End Function

Private Function ManagerCards() As String
    Return "<div class='card'><h3>Inventory Summary</h3><a
href='InventorySummary.aspx'>View</a></div>" &
        "<div class='card'><h3>Low Stock Report</h3><a
href='LowStockReport.aspx'>Check</a></div>" &
        "<div class='card'><h3>Supplier Info</h3><a
href='Suppliers.aspx'>Go</a></div>"
End Function

Private Function SalespersonCards() As String
    Return "<div class='card'><h3>Place Orders</h3><a
href='PlaceOrder.aspx'>Go</a></div>" &
        "<div class='card'><h3>Add Products</h3><a
href='AddProduct.aspx'>Go</a></div>" &
        "<div class='card'><h3>Customer Details</h3><a
href='Customers.aspx'>View</a></div>" &
        "<div class='card'><h3>Search Products</h3><a
href='Search.aspx'>Search</a></div>"
End Function
End Class

```

Admin Role

Users.aspx.vb

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

Partial Class Users
    Inherits System.Web.UI.Page

    Private connStr As String =
        ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
        Me.Load
        If Session("UserID") Is Nothing Then
            Response.Redirect("Login.aspx")
        ElseIf Session("RoleID") <> 1 Then
            Response.Redirect("Logout.aspx")
        End If

        If Not IsPostBack Then
            LoadUsers()
        End If
    End Sub

    Private Sub LoadUsers()
        Using conn As New SqlConnection(connStr)
            Dim query As String = "SELECT U.User_ID, U.Username,
R.Role_Name AS RoleName," &
                                "E.Employee_Name AS EmployeeName " &
                                "FROM User_t U " &
                                "JOIN Role_t R ON U.Role_ID =
R.Role_ID " &
                                "JOIN Employee E ON U.Employee_ID =
E.Employee_ID"
            Dim cmd As New SqlCommand(query, conn)
            Dim adapter As New SqlDataAdapter(cmd)
            Dim dt As New DataTable()
            adapter.Fill(dt)
            gvUsers.DataSource = dt
            gvUsers.DataBind()
        End Using
    End Sub

    Protected Sub btnAddUser_Click(sender As Object, e As EventArgs)
        Response.Redirect("AddUser.aspx")
    End Sub

    Protected Sub gvUsers_RowEditing(sender As Object, e As
        GridViewEditEventArgs)
        gvUsers.EditIndex = e.NewEditIndex
        LoadUsers()
    End Sub
```

```

        Protected Sub gvUsers_RowCancelingEdit(sender As Object, e As
GridViewCancelEventArgs)
            gvUsers.EditIndex = -1
            LoadUsers()
        End Sub

        Protected Sub gvUsers_RowUpdating(sender As Object, e As
GridViewUpdateEventArgs)
            Dim userId As Integer =
Convert.ToInt32(gvUsers.DataKeys(e.RowIndex).Value)
            Dim row As GridViewRow = gvUsers.Rows(e.RowIndex)
            Dim username As String = CType(row.Cells(1).Controls(0),
TextBox).Text.Trim()

            Using conn As New SqlConnection(connStr)
                Dim cmd As New SqlCommand("UPDATE User_t SET Username =
@username WHERE User_ID = @userId", conn)
                cmd.Parameters.AddWithValue("@username", username)
                cmd.Parameters.AddWithValue("@userId", userId)
                conn.Open()
                cmd.ExecuteNonQuery()
            End Using

            gvUsers.EditIndex = -1
            LoadUsers()
        End Sub

        Protected Sub gvUsers_RowDeleting(sender As Object, e As
GridViewDeleteEventArgs)
            Dim userId As Integer =
Convert.ToInt32(gvUsers.DataKeys(e.RowIndex).Value)

            Using conn As New SqlConnection(connStr)
                Dim cmd As New SqlCommand("DELETE FROM User_t WHERE
User_ID = @userId", conn)
                cmd.Parameters.AddWithValue("@userId", userId)
                conn.Open()
                cmd.ExecuteNonQuery()
            End Using

            LoadUsers()
        End Sub
    End Class

```

AddUser.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient
Imports System.Security.Cryptography
Imports System.Text

Partial Class AddUser

```

```

Inherits System.Web.UI.Page

Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString

Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
    If Session("UserID") Is Nothing Then
        Response.Redirect("Login.aspx")
    ElseIf Session("RoleID") <> 1 Then
        Response.Redirect("Logout.aspx")
    End If

    If Not IsPostBack Then
        LoadEmployees()
        LoadRoles()
    End If
End Sub

Private Sub LoadEmployees()
    Using conn As New SqlConnection(connStr)
        Dim cmd As New SqlCommand("SELECT Employee_ID,
Employee_Name FROM Employee", conn)
        conn.Open()
        ddlEmployee.DataSource = cmd.ExecuteReader()
        ddlEmployee.DataTextField = "Employee_Name"
        ddlEmployee.DataValueField = "Employee_ID"
        ddlEmployee.DataBind()
    End Using
End Sub

Private Sub LoadRoles()
    Using conn As New SqlConnection(connStr)
        Dim cmd As New SqlCommand("SELECT Role_ID, Role_Name FROM
Role_t", conn)
        conn.Open()
        ddlRole.DataSource = cmd.ExecuteReader()
        ddlRole.DataTextField = "Role_Name"
        ddlRole.DataValueField = "Role_ID"
        ddlRole.DataBind()
    End Using
End Sub

Protected Sub btnAddUser_Click(sender As Object, e As EventArgs)
    Dim username As String = txtUsername.Text.Trim()
    Dim password As String = txtPassword.Text
    Dim empId As Integer =
Convert.ToInt32(ddlEmployee.SelectedValue)
    Dim roleId As Integer = Convert.ToInt32(ddlRole.SelectedValue)
    Dim hash As String = ComputeHash(password)

    Using conn As New SqlConnection(connStr)
        conn.Open()

```

```

        ' Check if username already exists
        Dim checkCmd As New SqlCommand("SELECT COUNT(*) FROM
User_t WHERE Username = @username", conn)
        checkCmd.Parameters.AddWithValue("@username", username)

        Dim count As Integer =
Convert.ToInt32(checkCmd.ExecuteScalar())
        If count > 0 Then
            lblMessage.ForeColor = Drawing.Color.Red
            lblMessage.Text = "Username already exists. Please
choose another one."
            Return
        End If

        ' Insert new user
        Dim insertCmd As New SqlCommand("INSERT INTO User_t
(Employee_ID, Username, Password_Hash, Role_ID) VALUES (@empId,
@username, @passHash, @roleId)", conn)
        insertCmd.Parameters.AddWithValue("@empId", empId)
        insertCmd.Parameters.AddWithValue("@username", username)
        insertCmd.Parameters.AddWithValue("@passHash", hash)
        insertCmd.Parameters.AddWithValue("@roleId", roleId)

        Try
            insertCmd.ExecuteNonQuery()
            lblMessage.ForeColor = Drawing.Color.Green
            lblMessage.Text = "User added successfully!"
        Catch ex As SqlException
            lblMessage.ForeColor = Drawing.Color.Red
            lblMessage.Text = "Error: " & ex.Message
        End Try
    End Using
End Sub

Private Function ComputeHash(password As String) As String
    Using sha256 As SHA256 = SHA256.Create()
        Dim bytes As Byte() = Encoding.UTF8.GetBytes(password)
        Dim hashBytes As Byte() = sha256.ComputeHash(bytes)
        Return BitConverter.ToString(hashBytes).Replace("-",
"").ToLower()
    End Using
End Function
End Class

```

Employees.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Employees
    Inherits System.Web.UI.Page

```



```

Private connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString

Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
    If Session("UserID") Is Nothing Then
        Response.Redirect("Login.aspx")
    ElseIf Session("RoleID") <> 1 Then
        Response.Redirect("Logout.aspx")
    End If

    If Not IsPostBack Then
        LoadEmployees()
    End If
End Sub

Private Sub LoadEmployees()
    Using conn As New SqlConnection(connStr)
        Dim query As String = "SELECT Employee_ID, Employee_Name,
Phone, Email, Address, City, Hire_Date, Salary FROM Employee"
        Dim cmd As New SqlCommand(query, conn)
        Dim adapter As New SqlDataAdapter(cmd)
        Dim dt As New DataTable()
        adapter.Fill(dt)
        gvEmployees.DataSource = dt
        gvEmployees.DataBind()
    End Using
End Sub

Protected Sub btnAddEmployee_Click(sender As Object, e As
EventArgs)
    Response.Redirect("AddEmployee.aspx")
End Sub

Protected Sub gvEmployees_RowEditing(sender As Object, e As
GridViewEditEventArgs)
    gvEmployees.EditIndex = e.NewEditIndex
    LoadEmployees()
End Sub

Protected Sub gvEmployees_RowCancelingEdit(sender As Object, e As
GridViewCancelEventArgs)
    gvEmployees.EditIndex = -1
    LoadEmployees()
End Sub

Protected Sub gvEmployees_RowUpdating(sender As Object, e As
GridViewUpdateEventArgs)
    Dim empId As Integer =
Convert.ToInt32(gvEmployees.DataKeys(e.RowIndex).Value)
    Dim row As GridViewRow = gvEmployees.Rows(e.RowIndex)
    Dim empName As String = CType(row.Cells(1).Controls(0),
TextBox).Text.Trim()

```

```

        Dim phone As String = CType(row.Cells(2).Controls(0),
TextBox).Text.Trim()
        Dim email As String = CType(row.Cells(3).Controls(0),
TextBox).Text.Trim()
        Dim address As String = CType(row.Cells(4).Controls(0),
TextBox).Text.Trim()
        Dim city As String = CType(row.Cells(5).Controls(0),
TextBox).Text.Trim()
        Dim salary As Decimal =
Convert.ToDecimal(CType(row.Cells(7).Controls(0),
TextBox).Text.Trim())

        Using conn As New SqlConnection(connStr)
            Dim cmd As New SqlCommand("UPDATE Employee SET
Employee_Name = @empName, Phone = @phone, Email = @email, " &
"Address = @address, City =
@city, Salary = @salary WHERE Employee_ID = @empId", conn)
            cmd.Parameters.AddWithValue("@empName", empName)
            cmd.Parameters.AddWithValue("@phone", phone)
            cmd.Parameters.AddWithValue("@email", email)
            cmd.Parameters.AddWithValue("@address", address)
            cmd.Parameters.AddWithValue("@city", city)
            cmd.Parameters.AddWithValue("@salary", salary)
            cmd.Parameters.AddWithValue("@empId", empId)
            conn.Open()
            cmd.ExecuteNonQuery()
        End Using

        gvEmployees.EditIndex = -1
        LoadEmployees()
    End Sub

    Protected Sub gvEmployees_RowDeleting(sender As Object, e As
GridViewDeleteEventArgs)
        Dim empId As Integer =
Convert.ToInt32(gvEmployees.DataKeys(e.RowIndex).Value)

        ' Show a confirmation dialog with a Button to proceed with the
deletion
        Dim confirmationMessage As String = "Are you sure you want to
delete this employee?"
        lblConfirmationMessage.Text = confirmationMessage
        confirmDelete.Visible = True
        ViewState("empIdToDelete") = empId
    End Sub

    Protected Sub btnConfirmDelete_Click(sender As Object, e As
EventArgs)
        If ViewState("empIdToDelete") IsNot Nothing Then
            Dim empId As Integer =
Convert.ToInt32(ViewState("empIdToDelete"))

            Using conn As New SqlConnection(connStr)

```

```

        Dim cmd As New SqlCommand("DELETE FROM Employee WHERE
Employee_ID = @empId", conn)
        cmd.Parameters.AddWithValue("@empId", empId)
        conn.Open()
        cmd.ExecuteNonQuery()
    End Using

    LoadEmployees()

    ' Hide the confirmation panel after deletion
    confirmDelete.Visible = False
End If
End Sub

Protected Sub btnCancelDelete_Click(sender As Object, e As
EventArgs)
    ' Hide the confirmation panel without deleting
    confirmDelete.Visible = False
End Sub
End Class

```

AddEmployee.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class AddEmployee
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Session("UserID") Is Nothing Then
            Response.Redirect("Login.aspx")
        ElseIf Session("RoleID") <> 1 Then
            Response.Redirect("Logout.aspx")
        End If

    End Sub

    Private connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString

    Protected Sub btnAdd_Click(sender As Object, e As EventArgs)
        Dim empId As Integer
        If Not Integer.TryParse(txtEmployeeID.Text.Trim(), empId) Then
            lblMessage.Text = "Invalid Employee ID."
            lblMessage.ForeColor = Drawing.Color.Red
            Return
        End If

        Dim name As String = txtName.Text.Trim()
        Dim phone As String = txtPhone.Text.Trim()
        Dim email As String = txtEmail.Text.Trim()
    End Sub

```

```

Dim address As String = txtAddress.Text.Trim()
Dim city As String = txtCity.Text.Trim()
Dim hireDate As DateTime
Dim salary As Decimal

If Not DateTime.TryParse(txtHireDate.Text, hireDate) OrElse
Not Decimal.TryParse(txtSalary.Text, salary) Then
    lblMessage.Text = "Invalid Hire Date or Salary."
    lblMessage.ForeColor = Drawing.Color.Red
    Return
End If

Using conn As New SqlConnection(connStr)
    conn.Open()

    ' Check if Employee_ID already exists
    Dim checkCmd As New SqlCommand("SELECT COUNT(*) FROM
Employee WHERE Employee_ID = @EmpID", conn)
    checkCmd.Parameters.AddWithValue("@EmpID", empId)
    Dim exists As Integer =
Convert.ToInt32(checkCmd.ExecuteScalar())

    If exists > 0 Then
        lblMessage.Text = "Employee with this ID already
exists."
        lblMessage.ForeColor = Drawing.Color.Red
        Return
    End If

    ' Insert new employee
    Dim cmd As New SqlCommand("INSERT INTO Employee
(Employee_ID, Employee_Name, Phone, Email, Address, City, Hire_Date,
Salary)" &
                                " VALUES (@EmpID, @Name,
@Phone, @Email, @Address, @City, @HireDate, @Salary)", conn)
    cmd.Parameters.AddWithValue("@EmpID", empId)
    cmd.Parameters.AddWithValue("@Name", name)
    cmd.Parameters.AddWithValue("@Phone", phone)
    cmd.Parameters.AddWithValue("@Email", email)
    cmd.Parameters.AddWithValue("@Address", address)
    cmd.Parameters.AddWithValue("@City", city)
    cmd.Parameters.AddWithValue("@HireDate", hireDate)
    cmd.Parameters.AddWithValue("@Salary", salary)

    Try
        cmd.ExecuteNonQuery()
        lblMessage.Text = "Employee added successfully!"
        lblMessage.ForeColor = Drawing.Color.Green
    Catch ex As Exception
        lblMessage.Text = "Error: " & ex.Message
        lblMessage.ForeColor = Drawing.Color.Red
    End Try
End Using
End Sub

```

End Class

LoginAudit.aspx.vb

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

Partial Class LoginAudit
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            LoadAuditLogs()
        End If
    End Sub

    Private Sub LoadAuditLogs()
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim query As String = "SELECT A.Audit_ID, ISNULL(U.Username,
'Unknown') AS Username, " &
            "A.Login_Time, A.Logout_Time, A.Status " &
            "FROM Login_Audit A " &
            "LEFT JOIN User_T U ON A.User_ID = U.User_ID " &
            "ORDER BY A.Login_Time DESC"

        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand(query, conn)
                Dim dt As New DataTable()
                Using adapter As New SqlDataAdapter(cmd)
                    adapter.Fill(dt)
                End Using
                gvAuditLogs.DataSource = dt
                gvAuditLogs.DataBind()
            End Using
        End Using
    End Sub
End Class
```

Sales Person Role

PlaceOrder.aspx.vb

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

Partial Class PlaceOrder
    Inherits System.Web.UI.Page
```

```

    Private connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            LoadCustomers()
            LoadProducts()
            InitializeCart()
        End If
    End Sub

    Private Sub LoadCustomers()
        Using conn As New SqlConnection(connStr)
            Dim cmd As New SqlCommand("SELECT Customer_ID,
Customer_Name FROM Customer", conn)
            conn.Open()
            ddlCustomer.DataSource = cmd.ExecuteReader()
            ddlCustomer.DataTextField = "Customer_Name"
            ddlCustomer.DataValueField = "Customer_ID"
            ddlCustomer.DataBind()
        End Using
    End Sub

    Private Sub LoadProducts()
        Using conn As New SqlConnection(connStr)
            Dim cmd As New SqlCommand("SELECT Product_ID, Product_Name
FROM Product", conn)
            conn.Open()
            ddlProduct.DataSource = cmd.ExecuteReader()
            ddlProduct.DataTextField = "Product_Name"
            ddlProduct.DataValueField = "Product_ID"
            ddlProduct.DataBind()
        End Using
    End Sub

    Private Sub InitializeCart()
        Dim dt As New DataTable()
        dt.Columns.Add("ProductID", GetType(Integer))
        dt.Columns.Add("ProductName", GetType(String))
        dt.Columns.Add("Quantity", GetType(Integer))
        ViewState("Cart") = dt
    End Sub

    Protected Sub btnAddToCart_Click(sender As Object, e As EventArgs)
        Dim cart As DataTable = CType(ViewState("Cart"), DataTable)
        Dim productId As Integer =
Convert.ToInt32(ddlProduct.SelectedValue)
        Dim quantity As Integer

        If Not Integer.TryParse(txtQuantity.Text.Trim(), quantity)
OrElse quantity <= 0 Then
            lblMessage.Text = "Enter a valid quantity."
            Return

```

```

End If

Dim productName As String = ddlProduct.SelectedItem.Text
cart.Rows.Add(productId, productName, quantity)
ViewState("Cart") = cart

gvCart.DataSource = cart
gvCart.DataBind()
End Sub

Protected Sub btnConfirmOrder_Click(sender As Object, e As
EventArgs)
    Dim cart As DataTable = CType(ViewState("Cart"), DataTable)
    If cart.Rows.Count = 0 Then
        lblMessage.Text = "Cart is empty."
        Return
    End If

    Dim customerId As Integer =
Convert.ToInt32(ddlCustomer.SelectedValue)
    Dim paymentMode As String = ddlPaymentMode.SelectedValue
    Dim orderDate As Date = Date.Today
    Dim totalAmount As Decimal = 0D

    Using conn As New SqlConnection(connStr)
        conn.Open()
        Dim trans As SqlTransaction = conn.BeginTransaction()

        Try
            'Insert the Order
            Dim sqlInsertOrder As String = "DECLARE
@InsertedOrders TABLE (Order_ID INT); " &
"INSERT INTO Order_T (Customer_ID, Order_Date) " &
"OUTPUT INSERTED.Order_ID INTO @InsertedOrders " &
"VALUES (@cust, @date); " &
"SELECT Order_ID FROM @InsertedOrders;"

            Dim orderCmd As New SqlCommand(sqlInsertOrder, conn,
trans)

            orderCmd.Parameters.AddWithValue("@cust", customerId)
            orderCmd.Parameters.AddWithValue("@date", orderDate)

            Dim result As Object = orderCmd.ExecuteScalar()
            Dim orderId As Integer

            If result IsNot Nothing AndAlso Not
Convert.IsDBNull(result) Then
                orderId = Convert.ToInt32(result)
            Else
                trans.Rollback()
                lblMessage.Text = "Failed to retrieve Order ID.
Please try again."
            End If
            Exit Sub
        End Try
    End Using
End Sub

```

```

        ' Insert Order Items
        For Each row As DataRow In cart.Rows
            Dim productId As Integer =
Convert.ToInt32(row("ProductID"))
            Dim quantity As Integer =
Convert.ToInt32(row("Quantity"))

            ' Insert into Order_Item
            Dim itemCmd As New SqlCommand("INSERT INTO
Order_Item (Order_ID, Product_ID, Quantity) VALUES (@oid, @pid,
@qty)", conn, trans)
            itemCmd.Parameters.AddWithValue("@oid", orderId)
            itemCmd.Parameters.AddWithValue("@pid", productId)
            itemCmd.Parameters.AddWithValue("@qty", quantity)
            itemCmd.ExecuteNonQuery()

            ' Get unit price and calculate amount
            Dim priceCmd As New SqlCommand("SELECT Unit_Price
FROM Product WHERE Product_ID = @pid", conn, trans)
            priceCmd.Parameters.AddWithValue("@pid",
productId)
            totalAmount +=
Convert.ToDecimal(priceCmd.ExecuteScalar()) * quantity
        Next

        ' Insert Payment
        Dim payCmd As New SqlCommand("INSERT INTO Payment
(Order_ID, Amount_Paid, Payment_Date, Payment_Mode) VALUES (@oid,
@amount, @pdate, @pmode)", conn, trans)
        payCmd.Parameters.AddWithValue("@oid", orderId)
        payCmd.Parameters.AddWithValue("@amount", totalAmount)
        payCmd.Parameters.AddWithValue("@pdate", Date.Today)
        payCmd.Parameters.AddWithValue("@pmode", paymentMode)
        payCmd.ExecuteNonQuery()

        trans.Commit()

        lblMessage.Text = "Order placed successfully. Total:
Rs " & totalAmount.ToString("F2")
        ViewState("Cart") = Nothing
        InitializeCart()
        gvCart.DataSource = Nothing
        gvCart.DataBind()
        txtQuantity.Text = ""
        Response.Redirect("Invoice.aspx?orderid=" & orderId)
    Catch ex As Exception
        Try
            trans.Rollback()
        Catch rollex As Exception

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

```



```

        End Try

    End Using
End Sub

End Class

Invoice.aspx.vb

Imports System.Data
Imports System.Data.SqlClient

Partial Class Invoice
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            Dim orderId As String = Request.QueryString("orderid")
            If Not String.IsNullOrEmpty(orderId) Then
                LoadInvoice(CInt(orderId))
            End If
        End If
    End Sub

    Private Sub LoadInvoice(orderId As Integer)
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim totalAmount As Decimal = 0

        Using conn As New SqlConnection(connStr)
            conn.Open()

            ' Load order items
            Dim itemQuery As String = "SELECT P.Product_Name,
OI.Quantity, P.Unit_Price, (OI.Quantity * P.Unit_Price) AS Subtotal" &
" FROM Order_Item OI" &
" INNER JOIN Product P ON OI.Product_ID =
P.Product_ID" &
" WHERE OI.Order_ID = @orderId"
            Using cmd As New SqlCommand(itemQuery, conn)
                cmd.Parameters.AddWithValue("@orderId", orderId)
                Dim dt As New DataTable()
                Using reader As SqlDataReader = cmd.ExecuteReader()
                    dt.Load(reader)
                    gvOrderItems.DataSource = dt
                    gvOrderItems.DataBind()
                    For Each row As DataRow In dt.Rows
                        totalAmount +=
Convert.ToDecimal(row("Subtotal"))
                    Next
                End Using
            End Using
        End Using
    End Sub
End Sub

```

```

        End Using

        ' Load header and payment info
        Dim infoQuery As String = "SELECT O.Order_ID,
O.Order_Date, C.Customer_Name, P.Amount_Paid, P.Payment_Date,
P.Payment_Mode" &
        " FROM Order_T O" &
        " JOIN Customer C ON O.Customer_ID = C.Customer_ID" &
        " JOIN Payment P ON O.Order_ID = P.Order_ID" &
        " WHERE O.Order_ID = @orderId"
        Using cmd As New SqlCommand(infoQuery, conn)
            cmd.Parameters.AddWithValue("@orderId", orderId)
            Using reader As SqlDataReader = cmd.ExecuteReader()
                If reader.Read() Then
                    lblTotalAmount.Text = "Total Amount: Rs" &
totalAmount.ToString("N2")
                    lblPaymentInfo.Text = "<b>Payment:</b> Rs" &
Convert.ToDecimal(reader("Amount_Paid")).ToString("N2") &
                    " via " & reader("Payment_Mode") &
                    " on " &
CDate(reader("Payment_Date")).ToShortDateString()

                    End If
                End Using
            End Using
        End Using

        lblTotalAmount.Text = "Total Amount: Rs" &
totalAmount.ToString("N2")
    End Sub
End Class

```

Addproduct.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Data

Partial Class AddProduct
    Inherits System.Web.UI.Page

    Private connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
    Protected Sub Page_Load(sender As Object, e As EventArgs)
        If Not IsPostBack Then
            LoadCategories()
            LoadSuppliers()
        End If
    End Sub

    Private Sub LoadCategories()
        Dim query As String = "SELECT Category_ID, Category_Name FROM
Category"
        Try
            Using conn As New SqlConnection(connStr)

```

```

        Using cmd As New SqlCommand(query, conn)
            conn.Open()
            Using reader As SqlDataReader =
cmd.ExecuteReader()
                ddlCategory.Items.Clear()
                While reader.Read()

                    Dim item As New
ListItem(reader("Category_Name").ToString(),
reader("Category_ID").ToString())
                    ddlCategory.Items.Add(item)
                End While
            End Using
        End Using
    End Using
    Catch ex As Exception
        lblMessage.Text = "<span class='error'>Error loading
categories: " & ex.Message & "</span>"
    End Try
End Sub

Private Sub LoadSuppliers()
    Dim query As String = "SELECT Supplier_ID, Supplier_Name FROM
Supplier"
    Try
        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand(query, conn)
                conn.Open()
                Using reader As SqlDataReader =
cmd.ExecuteReader()
                    ddlSupplier.Items.Clear()
                    While reader.Read()
                        Dim item As New
ListItem(reader("Supplier_Name").ToString(),
reader("Supplier_ID").ToString())
                        ddlSupplier.Items.Add(item)
                    End While
                End Using
            End Using
        End Using
    Catch ex As Exception
        lblMessage.Text = "<span class='error'>Error loading
categories: " & ex.Message & "</span>"
    End Try
End Sub

Protected Sub btnAdd_Click(sender As Object, e As EventArgs)
    Dim prodId As Integer
    Dim price As Decimal
    Dim stock As Integer

```

```

        If Not Integer.TryParse(txtProductID.Text.Trim(), prodId)
OrElse
        Not Decimal.TryParse(txtPrice.Text.Trim(), price) OrElse
        Not Integer.TryParse(txtStock.Text.Trim(), stock) Then
            lblMessage.Text = "<span class='error'>Please enter valid
numeric values.</span>"
            Return
        End If

        Dim name As String = txtProductName.Text.Trim()
        If name = "" Then
            lblMessage.Text = "<span class='error'>Product name is
required.</span>"
            Return
        End If

        Dim categoryId As Integer =
Convert.ToInt32(ddlCategory.SelectedValue)
        Dim supplierId As Integer =
Convert.ToInt32(ddlSupplier.SelectedValue)

        Dim checkQuery As String = "SELECT COUNT(*) FROM Product WHERE
Product_ID = @id"
        Dim insertQuery As String = "INSERT INTO Product (Product_ID,
Product_Name, Unit_Price, Stock_Quantity, Category_ID, Supplier_ID) "
&
                                "VALUES (@id, @name, @price,
@stock, @category, @supplier)"

        Using conn As New SqlConnection(connStr)
            conn.Open()

            ' Check for duplicate Product_ID
            Using checkCmd As New SqlCommand(checkQuery, conn)
                checkCmd.Parameters.AddWithValue("@id", prodId)
                If Convert.ToInt32(checkCmd.ExecuteScalar()) > 0 Then
                    lblMessage.Text = "<span class='error'>Product ID
already exists.</span>"
                    Return
                End If
            End Using

            ' Insert the product
            Using insertCmd As New SqlCommand(insertQuery, conn)
                insertCmd.Parameters.AddWithValue("@id", prodId)
                insertCmd.Parameters.AddWithValue("@name", name)
                insertCmd.Parameters.AddWithValue("@price", price)
                insertCmd.Parameters.AddWithValue("@stock", stock)
                insertCmd.Parameters.AddWithValue("@category",
categoryId)
                insertCmd.Parameters.AddWithValue("@supplier",
supplierId)
                insertCmd.ExecuteNonQuery()
            End Using

```

```

        End Using

        lblMessage.Text = "<span class='success'>Product added
successfully.</span>"
        txtProductID.Text = ""
        txtProductName.Text = ""
        txtPrice.Text = ""
        txtStock.Text = ""
    End Sub
End Class

```

Customers.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Data

Partial Class Customers
    Inherits System.Web.UI.Page

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

    Protected Sub btnSearch_Click(sender As Object, e As EventArgs)
Handles btnSearch.Click
        LoadCustomers(txtSearch.Text.Trim())
    End Sub

    Private Sub LoadCustomers(Optional keyword As String = "")
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim query As String = "SELECT Customer_ID, Customer_Name,
Phone, Address FROM Customer"
        Dim dt As New DataTable()

        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand()
                cmd.Connection = conn

                If Not String.IsNullOrEmpty(keyword) Then
                    query &= " WHERE Customer_Name LIKE @kw OR Phone
LIKE @kw"

                    cmd.CommandText = query
                    cmd.Parameters.AddWithValue("@kw", "%" & keyword &
"%")

                Else
                    ' If no keyword, do nothing (return empty table)
                    gvCustomers.DataSource = Nothing
                    gvCustomers.DataBind()
                    lblStatus.Text = ""
                    Return
                End If
            End Using
        End Using
    End Sub

```

```

        cmd.CommandText = query
        Using adapter As New SqlDataAdapter(cmd)
            adapter.Fill(dt)
        End Using
    End Using
End Using

gvCustomers.DataSource = dt
gvCustomers.DataBind()

' Show message if no records found
If dt.Rows.Count = 0 Then
    lblStatus.Text = "No customers found."
Else
    lblStatus.Text = ""
End If
End Sub

Protected Sub btnAddCustomer_Click(sender As Object, e As
EventArgs) Handles btnAddCustomer.Click
    Dim name As String = txtName.Text.Trim()
    Dim phone As String = txtPhone.Text.Trim()
    Dim address As String = txtAddress.Text.Trim()

    If name = "" Then
        lblMessage.ForeColor = Drawing.Color.Red
        lblMessage.Text = "Name is required."
        Return
    End If

    Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
    Dim query As String = "INSERT INTO Customer (Customer_Name,
Phone, Address) VALUES (@name, @phone, @address)"

    Using conn As New SqlConnection(connStr)
        Using cmd As New SqlCommand(query, conn)
            cmd.Parameters.AddWithValue("@name", name)
            cmd.Parameters.AddWithValue("@phone", phone)
            cmd.Parameters.AddWithValue("@address", address)

            conn.Open()
            cmd.ExecuteNonQuery()
        End Using
    End Using

    lblMessage.ForeColor = Drawing.Color.Green
    lblMessage.Text = "Customer added successfully."

    ' Clear fields
    txtName.Text = ""
    txtPhone.Text = ""
    txtAddress.Text = ""

```

```

        LoadCustomers()
    End Sub
End Class

```

Search.aspx.vb

```

Imports System.Data.SqlClient
Imports System.Text

Partial Class Search
    Inherits System.Web.UI.Page

    Protected Sub btnSearch_ServerClick(sender As Object, e As
EventArgs)
        Dim keyword As String = txtSearch.Value.Trim()

        If keyword = "" Then
            results.InnerHtml = "<span style='color:red;'>Please enter
a product name.</span>"
            Return
        End If

        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim query As String = "SELECT TOP 5 Product_Name, Unit_Price
FROM Product WHERE Product_Name LIKE @name"

        Dim found As Boolean = False
        Dim sb As New StringBuilder()
        sb.Append("<table border='1'
cellpadding='5'><tr><th>Name</th><th>Unit Price</th></tr>")

        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@name", "%" & keyword &
"%")

                conn.Open()
                Dim reader As SqlDataReader = cmd.ExecuteReader()
                While reader.Read()
                    found = True
                    sb.Append("<tr>")
                    sb.Append("<td>" &
reader("Product_Name").ToString() & "</td>")
                    sb.Append("<td>" & reader("Unit_Price").ToString()
& "</td>")
                    sb.Append("</tr>")
                End While
            End Using
        End Using

        sb.Append("</table>")

        If found Then

```

```

        results.InnerHtml = sb.ToString()
    Else
        results.InnerHtml = "<span style='color:red;'>No matching
product found.</span>"
    End If
End Sub
End Class

```

Manager Role

InventorySummary.aspx.vb

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class InventorySummary
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            LoadInventory()
        End If
    End Sub

    Private Sub LoadInventory()
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim query As String = "SELECT P.Product_ID, P.Product_Name,
P.Unit_Price, P.Stock_Quantity, " &
                                "ISNULL(S.Supplier_Name, 'Unknown') AS
Supplier_Name " &
                                "FROM Product P " &
                                "LEFT JOIN Supplier S ON P.Supplier_ID =
S.Supplier_ID"

        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand(query, conn)
                Dim dt As New DataTable()
                Using adapter As New SqlDataAdapter(cmd)
                    adapter.Fill(dt)
                End Using
                gvInventory.DataSource = dt
                gvInventory.DataBind()
            End Using
        End Using
    End Sub
End Class

```

LowStockReport

```

Imports System.Data.SqlClient

```



```

Imports System.Data

Partial Class LowStockReport
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
        If Not IsPostBack Then
            LoadLowStockProducts()
        End If
    End Sub

    Private Sub LoadLowStockProducts()
        Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
        Dim threshold As Integer = 10
        Dim query As String = "SELECT P.Product_ID, P.Product_Name,
P.Stock_Quantity, " &
                                "ISNULL(S.Supplier_Name, 'Unknown') AS
Supplier_Name " &
                                "FROM Product P " &
                                "LEFT JOIN Supplier S ON P.Supplier_ID =
S.Supplier_ID " &
                                "WHERE P.Stock_Quantity < @threshold"

        Using conn As New SqlConnection(connStr)
            Using cmd As New SqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@threshold", threshold)

                Dim dt As New DataTable()
                Using adapter As New SqlDataAdapter(cmd)
                    adapter.Fill(dt)
                End Using

                gvLowStock.DataSource = dt
                gvLowStock.DataBind()

                lblNote.Visible = (dt.Rows.Count = 0)
                If dt.Rows.Count = 0 Then
                    lblNote.Text = "All products have sufficient
stock."
                End If
            End Using
        End Using
    End Sub
End Class

```

Suppliers

```

Imports System.Data
Imports System.Data.SqlClient

Partial Class Suppliers

```

```

Inherits System.Web.UI.Page

Protected Sub Page_Load(sender As Object, e As EventArgs) Handles
Me.Load
    If Not IsPostBack Then
        LoadSuppliers()
    End If
End Sub

Protected Sub btnSearch_Click(sender As Object, e As EventArgs)
Handles btnSearch.Click
    LoadSuppliers(txtSearch.Text.Trim())
End Sub

Private Sub LoadSuppliers(Optional keyword As String = "")
    Dim connStr As String =
ConfigurationManager.ConnectionStrings("myconnstr").ConnectionString
    Dim query As String = "SELECT Supplier_ID, Supplier_Name,
Phone, Email, City, State, Postal_Code FROM Supplier"

    If Not String.IsNullOrEmpty(keyword) Then
        query &= " WHERE Supplier_Name LIKE @kw OR City LIKE @kw
OR State LIKE @kw"
    End If

    Using conn As New SqlConnection(connStr)
        Using cmd As New SqlCommand(query, conn)
            If Not String.IsNullOrEmpty(keyword) Then
                cmd.Parameters.AddWithValue("@kw", "%" & keyword &
"%")
            End If

            Dim dt As New DataTable()
            Using adapter As New SqlDataAdapter(cmd)
                adapter.Fill(dt)
            End Using

            gvSuppliers.DataSource = dt
            gvSuppliers.DataBind()

            lblNote.Visible = (dt.Rows.Count = 0)
            If dt.Rows.Count = 0 Then
                lblNote.Text = "No suppliers found matching the
search criteria."
            End If
        End Using
    End Using
End Sub
End Class

```