

Lab 11: Code Implementation Report

File: Helpers\AccessControl.vb

```

' =====
' AccessControl.vb - Shared login & role-checking helpers
' =====
' PURPOSE:
' This module lives in App_Code so every page can call its functions.
' It uses SESSION variables to remember who is logged in and what role
' they have. Session values are set during Login and cleared on Logout.
'
' SESSION KEYS USED:
' Session("Username") - the logged-in user's name (Nothing = not logged in)
' Session("UserRole") - "Admin", "Customer", "Manager", "Employee", etc.
' Session("CustomerID") - the CUSTOMER_ID from the database (may be Nothing)
' =====

Imports System
Imports System.Web
Imports System.Web.UI
Imports System.Web.UI.WebControls

Public Module AccessControl

    ' —— Check if the user is logged in ——
    Public Function IsLoggedIn(ByVal page As Page) As Boolean
        Return page.Session("Username") IsNot Nothing
    End Function

    ' —— Get the user's role from session (returns "" if not set) ——
    Public Function GetCurrentRole(ByVal page As Page) As String
        If page.Session("UserRole") Is Nothing Then
            Return ""
        End If
        Return page.Session("UserRole").ToString()
    End Function

    ' —— Check if the user is an Admin ——
    Public Function IsAdminUser(ByVal page As Page) As Boolean
        Return GetCurrentRole(page) = "Admin"
    End Function

    ' —— Check if the user's role matches any of the given roles ——
    Public Function HasRole(ByVal page As Page, ParamArray allowedRoles() As String) As Boolean
        If Not IsLoggedIn(page) Then Return False

        Dim currentRole As String = GetCurrentRole(page)
        For Each allowedRole As String In allowedRoles
            If String.Equals(currentRole, allowedRole,
StringComparison.OrdinalIgnoreCase) Then
                Return True
            End If
        Next
        Return False
    End Function

    ' —— Redirect to Login if not logged in ——
    Public Sub RequireLogin(ByVal page As Page)
        If Not IsLoggedIn(page) Then
            Dim returnUrl As String = HttpUtility.UrlEncode(page.Request.RawUrl)
            page.Response.Redirect("Login.aspx?returnUrl=" & returnUrl, True)
        End Sub
    End Sub

```

```
End If
End Sub

' —— Redirect to Home if user doesn't have one of the allowed roles ——
Public Sub RequireRole(ByVal page As Page, ParamArray allowedRoles() As String)
    RequireLogin(page)
    If Not HasRole(page, allowedRoles) Then
        page.Response.Redirect("Default.aspx?error=access", True)
    End If
End Sub

' —— Get the customer ID from session. Returns -1 if not available. ——
Public Function GetCustomerId(ByVal page As Page) As Integer
    If page.Session("CustomerID") Is Nothing Then Return -1

    Dim customerId As Integer
    If Integer.TryParse(page.Session("CustomerID").ToString(), customerId) Then
        Return customerId
    End If
    Return -1
End Function

' —— Decide the customer ID for an action ——
' Admin users (or users with no CustomerID) pick from a dropdown.
' Regular users use their own session CustomerID.
' Returns the customer ID, or -1 if something is wrong.
Public Function ResolveCustomerId(ByVal page As Page, ByVal dropdown As
DropDownList) As Integer

    ' Admin or missing CustomerID → read from the dropdown.
    If IsAdminUser(page) Or GetCustomerId(page) = -1 Then
        Dim selectedId As Integer
        If Integer.TryParse(dropdown.SelectedValue, selectedId) AndAlso selectedId >
0 Then
            Return selectedId
        End If
        Return -1
    End If

    ' Regular user → use their own CustomerID from session.
    Return GetCustomerId(page)

End Function

' —— Should this user see the admin/customer-picker UI? ——
Public Function ShouldShowAdminView(ByVal page As Page) As Boolean
    Return IsAdminUser(page) Or GetCustomerId(page) = -1
End Function

End Module
```

File: Helpers\ProductHelper.vb

```

' =====
' ProductHelper.vb - Shared product utility functions
' =====
' PURPOSE:
' This module contains helper functions that are used by MULTIPLE pages.
' Keeping them here avoids copying the same code into every page.
'
' FUNCTIONS:
' GetProductName()           - look up a product's description by ID
' LoadProductsIntoDropDown() - fill a DropDownList with all products
' GetRecommendations()     - find "also bought" products using SQL
' FormatRecommendationMessages() - turn raw data into user-friendly text
' =====

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration
Imports System.Web.UI.WebControls

Public Module ProductHelper

    ' -----
    ' GetConnectionString - Reads the "PVFC" connection string
    '                      from Web.config so we don't repeat it.
    ' -----

    Private Function GetConnectionString() As String
        Return ConfigurationManager.ConnectionStrings("PVFC").ConnectionString
    End Function

    ' -----
    ' GetProductName - Given a product ID, return its description.
    '                Returns "this product" if not found.
    ' -----

    Public Function GetProductName(productId As Integer) As String

        ' Step 1: Open a connection to the database.
        Using connection As New SqlConnection(GetConnectionString())

            ' Step 2: Write a simple SELECT query.
            Dim sqlQuery As String = "SELECT TOP 1 PRODUCT_DESCRIPTION FROM Product_T
WHERE PRODUCT_ID = @ProductId"

            ' Step 3: Create the command and add the parameter.
            Dim command As New SqlCommand(sqlQuery, connection)
            command.Parameters.AddWithValue("@ProductId", productId)

            ' Step 4: Open, execute, and read the result.
            connection.Open()
            Dim result As Object = command.ExecuteScalar()

            ' Step 5: Return the name if found, or a fallback string.
            If result IsNot Nothing AndAlso result IsNot DBNull.Value Then
                Return result.ToString()
            End If

        End Using

        Return "this product"
    End Function

```

End Function

```
' LoadProductsIntoDropDown - Fills a DropDownList with every
' product from the Product_T table. Adds a blank "select"
' option at the top.
```

```
Public Sub LoadProductsIntoDropDown(productDropDown As DropDownList)
```

```
    ' Step 1: Open a connection.
```

```
    Using connection As New SqlConnection(GetConnectionString())
```

```
        ' Step 2: Get all products.
```

```
        Dim sqlQuery As String = "SELECT PRODUCT_ID, PRODUCT_DESCRIPTION FROM
Product_T"
```

```
        Dim adapter As New SqlDataAdapter(sqlQuery, connection)
```

```
        Dim productsTable As New DataTable()
```

```
        adapter.Fill(productsTable)
```

```
        ' Step 3: Bind the data to the dropdown.
```

```
        productDropDown.DataTextField = "PRODUCT_DESCRIPTION"
```

```
        productDropDown.DataValueField = "PRODUCT_ID"
```

```
        productDropDown.DataSource = productsTable
```

```
        productDropDown.DataBind()
```

```
        ' Step 4: Insert a blank "please select" item at position 0.
```

```
        productDropDown.Items.Insert(0, New ListItem("-- Select Product --", "0"))
```

```
    End Using
```

```
End Sub
```

```
' GetRecommendations - THE CORE RECOMMENDATION QUERY
```

```
' This is the "association rule mining" part of Lab 11.
' It finds products that were bought IN THE SAME ORDER as
' the given product. The more often two products appear
' together, the higher the count (Cnt).
```

```
' HOW THE SQL WORKS (step by step):
```

```
' 1. Start with Order_Line_T rows where PRODUCT_ID = our product (o11)
```

```
' 2. JOIN to OTHER Order_Line_T rows in the SAME order (o12)
```

```
' 3. Exclude the product itself (o12.PRODUCT_ID <> o11.PRODUCT_ID)
```

```
' 4. JOIN Product_T to get the name of the co-occurring product
```

```
' 5. GROUP BY to count how many times each pairing occurred
```

```
' 6. ORDER BY count descending so the most common pair is first
```

```
' 7. TOP N limits the results
```

```
' RETURNS: A DataTable with columns:
```

```
' PRODUCT_ID | PRODUCT_DESCRIPTION | Cnt
```

```
Public Function GetRecommendations(productId As Integer, topN As Integer) As
DataTable
```

```
    Dim recommendationsTable As New DataTable()
```

```
    ' Step 1: Open a connection.
```

```
    Using connection As New SqlConnection(GetConnectionString())
```

```
        ' Step 2: Write the co-occurrence SQL.
```

```
        ' "Find products that appear in the same orders as @ProductId"
```

```
        Dim sqlQuery As String =
```

```
            "SELECT TOP " & topN & " " &
```

```

        "o12.PRODUCT_ID, p.PRODUCT_DESCRIPTION, COUNT(*) AS Cnt " &
        "FROM Order_Line_T o11 " &
        "JOIN Order_Line_T o12 ON o11.ORDER_ID = o12.ORDER_ID " &
        " AND o12.PRODUCT_ID <> o11.PRODUCT_ID " &
        "JOIN Product_T p ON o12.PRODUCT_ID = p.PRODUCT_ID " &
        "WHERE o11.PRODUCT_ID = @ProductId " &
        "GROUP BY o12.PRODUCT_ID, p.PRODUCT_DESCRIPTION " &
        "ORDER BY Cnt DESC"

        ' Step 3: Create command and add the parameter.
        Dim command As New SqlCommand(sqlQuery, connection)
        command.Parameters.AddWithValue("@ProductId", productId)

        ' Step 4: Fill the DataTable with results.
        Dim adapter As New SqlDataAdapter(command)
        adapter.Fill(recommendationsTable)

    End Using

    Return recommendationsTable

End Function

' -----
' FormatRecommendationMessages - Takes the raw recommendations
'   DataTable and builds human-readable strings like:
'   "2+ customers who bought Dining Table also bought Chair"
'
'   If no recommendations exist, it returns a friendly fallback.
'
'   RETURNS: A DataTable with one column "RecommendationText"
' -----

Public Function FormatRecommendationMessages(productId As Integer, recommendations
As DataTable) As DataTable

    Dim messagesTable As New DataTable()
    messagesTable.Columns.Add("RecommendationText")

    ' Get the name of the primary product for the message.
    Dim primaryProductName As String = GetProductName(productId)

    If recommendations.Rows.Count > 0 Then

        ' Build one message per recommended product.
        For Each row As DataRow In recommendations.Rows
            Dim secondaryProductName As String =
row("PRODUCT_DESCRIPTION").ToString()
            Dim pairCount As Integer = Convert.ToInt32(row("Cnt"))

            Dim message As String = String.Format(
                "{0}+ customers who bought {1} also bought {2}",
                pairCount, primaryProductName, secondaryProductName)

            messagesTable.Rows.Add(message)
        Next

    Else

        ' Fallback message when there is not enough order history.
        messagesTable.Rows.Add(
            "We at PVFC believe " & primaryProductName &
            " is a very graceful set with helpful matching pieces.")

    End If

```

```
Return messagesTable
```

```
End Function
```

```
End Module
```

File: Admin.aspx

```
<%@ Page Title="Admin" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Admin.aspx.vb" Inherits="Lab11.Admin" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
  <div class="content">
    <h3>Admin: Recommendation Insights</h3>
    <p>Top product pairings (how often two products were bought together):</p>
    <asp:GridView ID="gvPairs" runat="server" CssClass="pvf-grid"
AutoGenerateColumns="True"></asp:GridView>

    <h4>Inventory Helper</h4>
    <p>Select a primary product to see which secondary items should be kept ready
when demand rises:</p>

    <div class="form-group">
      <asp:DropDownList ID="ddlForecastPrimary" runat="server"></asp:DropDownList>
      <asp:Button ID="btnForecast" runat="server" CssClass="pvf-btn" Text="Run
Helper" OnClick="btnForecast_Click" />
    </div>

    <asp:Panel ID="pnlForecast" runat="server" Visible="false" CssClass="recommend">
      <h5>Display Recommendations</h5>
      <asp:Repeater ID="rptForecast" runat="server">
        <ItemTemplate>
          <div><%# Eval("RecommendationText") %></div>
        </ItemTemplate>
      </asp:Repeater>
    </asp:Panel>
  </div>
</asp:Content>
```

File: Admin.aspx.vb

```

' =====
' Admin.aspx.vb - Admin page: product pairing analytics & inventory helper
' =====
' Only Admin / Manager / Employee roles can access this page.
' =====

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Public Partial Class Admin
    Inherits System.Web.UI.Page

    Protected WithEvents gvPairs As Global.System.Web.UI.WebControls.GridView
    Protected WithEvents ddlForecastPrimary As
Global.System.Web.UI.WebControls.DropDownList
    Protected WithEvents btnForecast As Global.System.Web.UI.WebControls.Button
    Protected WithEvents pnlForecast As Global.System.Web.UI.WebControls.Panel
    Protected WithEvents rptForecast As Global.System.Web.UI.WebControls.Repeater

    Private connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        AccessControl.RequireRole(Me, "Admin", "Manager", "Employee")

        If Not IsPostBack Then
            LoadTopPairs()
            ProductHelper.LoadProductsIntoDropDown(ddlForecastPrimary)
        End If
    End Sub

    ' Show top product pairings across all orders.
    Private Sub LoadTopPairs()
        Using con As New SqlConnection(connectionString)
            Dim adapter As New SqlDataAdapter(
                "SELECT TOP 20 " &
                " p1.PRODUCT_DESCRIPTION AS PrimaryProduct, " &
                " p2.PRODUCT_DESCRIPTION AS SecondaryProduct, " &
                " COUNT(*) AS PairCount " &
                "FROM Order_Line_T ol1 " &
                "JOIN Order_Line_T ol2 ON ol1.ORDER_ID = ol2.ORDER_ID AND ol1.PRODUCT_ID
<> ol2.PRODUCT_ID " &
                "JOIN Product_T p1 ON ol1.PRODUCT_ID = p1.PRODUCT_ID " &
                "JOIN Product_T p2 ON ol2.PRODUCT_ID = p2.PRODUCT_ID " &
                "GROUP BY p1.PRODUCT_DESCRIPTION, p2.PRODUCT_DESCRIPTION " &
                "ORDER BY PairCount DESC", con)
            Dim dt As New DataTable()
            adapter.Fill(dt)
            gvPairs.DataSource = dt
            gvPairs.DataBind()
        End Using
    End Sub

    ' Forecast button - show what sells alongside the selected product.
    Protected Sub btnForecast_Click(sender As Object, e As EventArgs)
        pnlForecast.Visible = False
        If ddlForecastPrimary.SelectedValue = "0" Then Return
    End Sub

```

```

        Dim productId As Integer = Convert.ToInt32(ddlForecastPrimary.SelectedValue)
        Dim recs As DataTable = ProductHelper.GetRecommendations(productId, 10)
        Dim msgs As DataTable = ProductHelper.FormatRecommendationMessages(productId,
recs)

        rptForecast.DataSource = msgs
        rptForecast.DataBind()
        pnlForecast.Visible = True
    End Sub

End Class

```

File: Default.aspx

```

<%@ Page Title="Home" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Default.aspx.vb" Inherits="Lab11._Default" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <div class="content">
        <h2>PVFC - Lab 11: Recommendation System</h2>
        <asp:Label ID="lblAccessMsg" runat="server" CssClass="err-label"></asp:Label>
        <p>This lab adds a simple "customers who bought this also bought" recommendation
feature based on order co-occurrence counts.</p>
        <p>Use the <a href="Products.aspx">Products</a> page to browse and see
recommendations for each product.</p>
    </div>
</asp:Content>

```

File: Default.aspx.vb

```

' =====
' Default.aspx.vb - Home page (shows access-denied message if redirected)
' =====

Imports System
Imports System.Web.UI

Public Partial Class _Default
    Inherits Page

    Protected WithEvents lblAccessMsg As Global.System.Web.UI.WebControls.Label

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If String.Equals(Request.QueryString("error"), "access",
StringComparison.OrdinalIgnoreCase) Then
            lblAccessMsg.ForeColor = Drawing.Color.Red
            lblAccessMsg.Text = "Access denied. You do not have permission for that
page."
        Else
            lblAccessMsg.Text = ""
        End If
    End Sub

End Class

```

File: Help.aspx

```
<%@ Page Title="Help" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Help.aspx.vb" Inherits="Lab11.Help" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <div class="content">
        <h3>Help</h3>
        <p>Lab 11 adds a recommendation system based on past orders.</p>
        <ul style="text-align:left; display:inline-block;">
            <li>Register or login to access protected pages.</li>
            <li>Open Products to see "Customers who bought this also bought"
suggestions.</li>
            <li>Open Orders to place an order and view add-on recommendations.</li>
            <li>Open Catalog for admin insights.</li>
        </ul>
    </div>
</asp:Content>
```

File: Help.aspx.vb

```
Imports System
Imports System.Web.UI

Public Partial Class Help
    Inherits Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) Handles
Me.Load
        ' No extra logic needed for the help page.
    End Sub
End Class
```

File: Login.aspx

```
<%@ Page Title="Login" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Login.aspx.vb" Inherits="Lab11.Login" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
  <div class="mainBody">
    <h3 class="title">Login</h3>
    <asp:Label ID="lblLoginMsg" runat="server" CssClass="err-label"></asp:Label>

    <div style="background-color: #fcebbd; padding: 15px; margin-bottom: 20px;
border-radius: 5px; border-left: 4px solid #4a3f2f;">
      <h4 style="margin-top: 0; color: #4a3f2f;">Available Test Accounts:</h4>
      <asp:GridView ID="gvUsers" runat="server" CssClass="data-table"
        AutoGenerateColumns="False" GridLines="Both"
        BorderStyle="Solid" BorderWidth="1px" BorderColor="#4a3f2f"
        RowStyle-BackColor="#fff8e7" HeaderStyle-BackColor="#f9d9a1"
        HeaderStyle-ForeColor="#4a3f2f" HeaderStyle-Font-Bold="true">
        <Columns>
          <asp:BoundField DataField="Username" HeaderText="Username" />
          <asp:BoundField DataField="User_Password" HeaderText="Password" />
          <asp:BoundField DataField="User_Role" HeaderText="Role" />
        </Columns>
      </asp:GridView>
    </div>

    <div class="form-group">
      <label>Username</label>
      <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>Password</label>
      <asp:TextBox ID="txtPassword" runat="server" TextMode="Password">
    </asp:TextBox>
    </div>

    <div class="form-group center">
      <asp:Button ID="btnLogin" runat="server" CssClass="pvf-btn" Text="Login"
        OnClick="btnLogin_Click" />
    </div>
  </div>
</asp:Content>
```

File: Login.aspx.vb

```

' =====
' Login.aspx.vb - Code-behind for the Login page
' =====
' Shows all demo users in a grid, then authenticates when login is clicked.
' =====

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Public Partial Class Login
    Inherits System.Web.UI.Page

    Protected WithEvents gvUsers As Global.System.Web.UI.WebControls.GridView
    Protected WithEvents txtUsername As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtPassword As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents lblLoginMsg As Global.System.Web.UI.WebControls.Label
    Protected WithEvents btnLogin As Global.System.Web.UI.WebControls.Button

    Private connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

    ' ----- PAGE LOAD -----
    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        If Not IsPostBack Then
            ' Show all accounts in the grid so testers can see credentials.
            Using con As New SqlConnection(connectionString)
                Dim adapter As New SqlDataAdapter("SELECT Username, User_Password,
User_Role FROM USERS ORDER BY Username", con)
                Dim dt As New DataTable()
                adapter.Fill(dt)
                gvUsers.DataSource = dt
                gvUsers.DataBind()
            End Using
        End If
    End Sub

    ' ----- LOGIN BUTTON -----
    Protected Sub btnLogin_Click(sender As Object, e As EventArgs)

        Using con As New SqlConnection(connectionString)
            Dim cmd As New SqlCommand(
                "SELECT TOP 1 UserID, User_Role, CUSTOMER_ID FROM USERS " &
                "WHERE Username = @u AND User_Password = @p", con)
            cmd.Parameters.AddWithValue("@u", txtUsername.Text.Trim())
            cmd.Parameters.AddWithValue("@p", txtPassword.Text.Trim())

            con.Open()
            Dim reader As SqlDataReader = cmd.ExecuteReader()

            If reader.Read() Then
                ' Save login info into session.
                Session("Username") = txtUsername.Text.Trim()
                Session("UserRole") = reader("User_Role").ToString()

                ' CUSTOMER_ID can be NULL for admin-only accounts.
                If IsDBNull(reader("CUSTOMER_ID")) Then
                    Session("CustomerID") = Nothing
                End If
            End If
        End Using
    End Sub
End Class

```

```
        Else
            Session("CustomerID") = Convert.ToInt32(reader("CUSTOMER_ID"))
        End If

        ' Redirect back to where they came from, or to home.
        Dim returnUrl As String = Request.QueryString("returnUrl")
        If Not String.IsNullOrEmpty(returnUrl) Then
            Response.Redirect(returnUrl)
        Else
            Response.Redirect("Default.aspx")
        End If
    Else
        lblLoginMsg.ForeColor = Drawing.Color.Red
        lblLoginMsg.Text = "Invalid username or password."
    End If
End Using

End Sub

End Class
```

File: Orders.aspx

```

<%@ Page Title="Orders" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Orders.aspx.vb" Inherits="Lab11.Orders" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <div class="content">
        <h3>Place Order</h3>
        <asp:Label ID="lblMsg" runat="server" CssClass="msg-label"></asp:Label>

        <div class="form-group">
            <label>Select Buyer</label>
            <asp:DropDownList ID="ddlCustomers" runat="server"></asp:DropDownList>
        </div>

        <h4>Compose Order</h4>
        <asp:Repeater ID="rptOrderLines" runat="server"
OnItemDataBound="rptOrderLines_ItemDataBound" OnItemCommand="rptOrderLines_ItemCommand">
            <ItemTemplate>
                <div class="form-group" style="padding: 10px; border: 1px solid #e0d7c8;
margin-bottom: 10px; background: #fffaf0;">
                    <label>Product</label>
                    <asp:DropDownList ID="ddlLineProduct" runat="server">
</asp:DropDownList>
                    <label style="display:block; margin-top:8px;">Quantity</label>
                    <asp:TextBox ID="txtLineQty" runat="server" Text="1"
Style="width:80px" />
                    <div style="margin-top:8px;">
                        <asp:Button ID="btnRemoveLine" runat="server" Text="Remove"
CommandName="RemoveLine" CommandArgument="<%=# Container.ItemIndex %>" />
                    </div>
                </div>
            </ItemTemplate>
        </asp:Repeater>

        <div class="form-group center">
            <asp:Button ID="btnAdd" runat="server" CssClass="pvf-btn" Text="Add Product"
OnClick="btnAdd_Click" />
        </div>

        <div class="form-group center" style="margin-top:15px;">
            <asp:Button ID="btnPlace" runat="server" CssClass="pvf-btn" Text="Place
Order" OnClick="btnPlace_Click" />
        </div>

        <asp:Panel ID="pnlOrderRec" runat="server" Visible="false" CssClass="recommend">
            <h4>Recommended add-ons after this order:</h4>
            <asp:Repeater ID="rptOrderRec" runat="server">
                <ItemTemplate>
                    <div><%=# Eval("RecommendationText") %></div>
                </ItemTemplate>
            </asp:Repeater>
        </asp:Panel>

        <hr />
        <h4>Order History</h4>
        <asp:GridView ID="gvOrders" runat="server" AutoGenerateColumns="false"
EmptyDataText="No orders found.">
            <Columns>
                <asp:BoundField DataField="ORDER_ID" HeaderText="Order #" />
                <asp:BoundField DataField="ORDER_DATE" HeaderText="Date"
DateFormatString="{0:yyyy-MM-dd}" />

```

```
<asp:BoundField DataField="CUSTOMER_NAME" HeaderText="Customer" />
  <asp:BoundField DataField="PRODUCT_DESCRIPTION" HeaderText="Product" />
  <asp:BoundField DataField="ORDERED_QUANTITY" HeaderText="Qty" />
</Columns>
</asp:GridView>
</div>

</asp:Content>
```

File: Orders.aspx.vb

```

' =====
' Orders.aspx.vb - Code-behind for the Orders page
' =====
' WHAT THIS PAGE DOES:
' 1. Lets the user build a multi-line order (pick products + quantities).
' 2. Inserts the order into Order_T and Order_Line_T.
' 3. After placing an order, shows "also bought" recommendations.
' 4. Displays the user's order history in a GridView at the bottom.
' =====

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Public Partial Class Orders
    Inherits System.Web.UI.Page

    Protected WithEvents ddlCustomers As Global.System.Web.UI.WebControls.DropDownList
    Protected WithEvents gvOrders As Global.System.Web.UI.WebControls.GridView
    Protected WithEvents rptOrderLines As Global.System.Web.UI.WebControls.Repeater
    Protected WithEvents lblMsg As Global.System.Web.UI.WebControls.Label
    Protected WithEvents pnlOrderRec As Global.System.Web.UI.WebControls.Panel
    Protected WithEvents rptOrderRec As Global.System.Web.UI.WebControls.Repeater

    Private connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

    ' —— Ensure the session has an order-lines table, return it ——
    Private Function GetOrderLinesTable() As DataTable

        Dim orderLines As DataTable = TryCast(Session("OrderLines"), DataTable)

        If orderLines Is Nothing Then
            orderLines = New DataTable()
            orderLines.Columns.Add("PRODUCT_ID", GetType(Integer))
            orderLines.Columns.Add("PRODUCT_DESCRIPTION", GetType(String))
            orderLines.Columns.Add("QUANTITY", GetType(Integer))
            orderLines.Rows.Add(0, "", 1) ' start with one blank line
            Session("OrderLines")= orderLines
        End If

        Return orderLines
    End Function

    End Function

    ' —— PAGE INIT - build repeater controls early so postback values survive ——
    Protected Sub Page_Init(sender As Object, e As EventArgs) Handles Me.Init
        rptOrderLines.DataSource = GetOrderLinesTable()
        rptOrderLines.DataBind()
    End Sub

    ' —— PAGE LOAD ——
    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        AccessControl.RequireLogin(Me)

        If Not IsPostBack Then
            ' Show customer picker only for admins / users without a linked customer.
            ddlCustomers.Visible = AccessControl.ShouldShowAdminView(Me)
        End If
    End Sub
End Class

```

```

        If ddlCustomers.Visible Then LoadCustomers()
        LoadOrdersHistory()
    End If
End Sub

' —— Fill the buyer dropdown ——
Private Sub LoadCustomers()
    Using con As New SqlConnection(connectionString)
        Dim adapter As New SqlDataAdapter("SELECT CUSTOMER_ID, CUSTOMER_NAME FROM
customer_T", con)
        Dim dt As New DataTable()
        adapter.Fill(dt)

        ddlCustomers.DataTextField = "CUSTOMER_NAME"
        ddlCustomers.DataValueField = "CUSTOMER_ID"
        ddlCustomers.DataSource = dt
        ddlCustomers.DataBind()
        ddlCustomers.Items.Insert(0, New System.Web.UI.WebControls.ListItem("--
Select Buyer --", "0"))
    End Using
End Sub

' —— Load order history (admin sees all, customer sees own) ——
Private Sub LoadOrdersHistory()

    ' Build one query; add a WHERE only for regular customers.
    Dim sqlQuery As String =
        "SELECT o.ORDER_ID, o.ORDER_DATE, c.CUSTOMER_NAME, " &
        "        p.PRODUCT_DESCRIPTION, ol.ORDERED_QUANTITY " &
        "FROM Order_T o " &
        "INNER JOIN customer_T c ON o.CUSTOMER_ID = c.CUSTOMER_ID " &
        "INNER JOIN Order_Line_T ol ON o.ORDER_ID = ol.ORDER_ID " &
        "INNER JOIN Product_T p ON ol.PRODUCT_ID = p.PRODUCT_ID "

    Dim customerId As Integer = AccessControl.GetCustomerId(Me)

    If Not AccessControl.IsAdminUser(Me) AndAlso customerId > 0 Then
        sqlQuery &= "WHERE o.CUSTOMER_ID = @CustomerId "
    End If

    sqlQuery &= "ORDER BY o.ORDER_ID DESC, ol.PRODUCT_ID DESC"

    Using con As New SqlConnection(connectionString)
        Dim cmd As New SqlCommand(sqlQuery, con)
        If Not AccessControl.IsAdminUser(Me) AndAlso customerId > 0 Then
            cmd.Parameters.AddWithValue("@CustomerId", customerId)
        End If

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

        gvOrders.DataSource = dt
        gvOrders.DataBind()
    End Using

End Sub

' —— Fill dropdown & restore selection when a repeater row is created ——
Protected Sub rptOrderLines_ItemDataBound(sender As Object, e As
RepeaterItemEventArgs)
    If e.Item.ItemType <> ListItemType.Item AndAlso e.Item.ItemType <>
ListItemType.AlternatingItem Then Return

    Dim ddl As DropDownList = CType(e.Item.FindControl("ddlLineProduct"),

```

```

DropDownList)
    Dim txt As TextBox = CType(e.Item.FindControl("txtLineQty"), TextBox)
    If ddl Is Nothing OrElse txt Is Nothing Then Return

    Dim row As DataRowView = CType(e.Item.DataItem, DataRowView)
    ProductHelper.LoadProductsIntoDropDown(ddl)

    Dim pid As Integer = 0
    Integer.TryParse(row("PRODUCT_ID").ToString(), pid)
    If pid > 0 Then ddl.SelectedValue = pid.ToString()

    txt.Text = row("QUANTITY").ToString()
End Sub

' —— Remove a line from the order ——
Protected Sub rptOrderLines_ItemCommand(sender As Object, e As
RepeaterCommandEventArgs)
    If e.CommandName <> "RemoveLine" Then Return

    Dim idx As Integer
    If Not Integer.TryParse(e.CommandArgument.ToString(), idx) Then Return

    Dim dt As DataTable = GetOrderLinesTable()
    If idx >= 0 AndAlso idx < dt.Rows.Count Then dt.Rows.RemoveAt(idx)
    If dt.Rows.Count = 0 Then dt.Rows.Add(0, "", 1)

    Session("OrderLines") = dt
    rptOrderLines.DataSource = dt
    rptOrderLines.DataBind()
    lblMsg.Text = "Line removed."
End Sub

' —— Add another product line ——
Protected Sub btnAdd_Click(sender As Object, e As EventArgs)
    Dim dt As DataTable = GetOrderLinesTable()
    dt.Rows.Add(0, "", 1)
    Session("OrderLines") = dt
    rptOrderLines.DataSource = dt
    rptOrderLines.DataBind()
    lblMsg.Text = "Add the next product line, then place the order when finished."
End Sub

' —— PLACE ORDER ——
Protected Sub btnPlace_Click(sender As Object, e As EventArgs)
    lblMsg.Text = ""

    ' 1. Read product lines from the repeater.
    ' If the same product appears twice, merge quantities into one line.
    Dim lines As New DataTable()
    lines.Columns.Add("PRODUCT_ID", GetType(Integer))
    lines.Columns.Add("QUANTITY", GetType(Integer))

    For Each item As RepeaterItem In rptOrderLines.Items
        Dim ddl As DropDownList = CType(item.FindControl("ddlLineProduct"),
DropDownList)
        Dim txt As TextBox = CType(item.FindControl("txtLineQty"), TextBox)
        If ddl Is Nothing OrElse txt Is Nothing Then Continue For

        Dim pid As Integer
        If Not Integer.TryParse(ddl.SelectedValue, pid) OrElse pid = 0 Then Continue
For

        Dim qty As Integer = 1
        Integer.TryParse(txt.Text, qty)
        If qty < 1 Then qty = 1

```

```

        ' Merge: if this product already exists in our list, add quantities
together.
        Dim existing() As DataRow = lines.Select("PRODUCT_ID = " & pid)
        If existing.Length > 0 Then
            existing(0)("QUANTITY") = CInt(existing(0)("QUANTITY")) + qty
        Else
            lines.Rows.Add(pid, qty)
        End If
    Next

    If lines.Rows.Count = 0 Then
        lblMsg.Text = "Add products before placing order."
        Return
    End If

    ' 2. Figure out who the customer is (shared helper handles admin vs regular).
    Dim customerId As Integer = AccessControl.ResolveCustomerId(Me, ddlCustomers)
    If customerId = -1 Then
        lblMsg.Text = "Please select a customer before placing order."
        Return
    End If

    ' 3. Insert order + lines.
    Using con As New SqlConnection(connectionString)
        con.Open()

        Dim cmdOrder As New SqlCommand(
            "INSERT INTO Order_T (CUSTOMER_ID, ORDER_DATE) OUTPUT INSERTED.ORDER_ID
VALUES (@cid, @dt)", con)
        cmdOrder.Parameters.AddWithValue("@cid", customerId)
        cmdOrder.Parameters.AddWithValue("@dt", DateTime.Today)
        Dim newOrderId As Integer = Convert.ToInt32(cmdOrder.ExecuteScalar())

        For Each row As DataRow In lines.Rows
            Dim cmdLine As New SqlCommand(
                "INSERT INTO Order_Line_T (ORDER_ID, PRODUCT_ID, ORDERED_QUANTITY)
VALUES (@oid, @pid, @qty)", con)
            cmdLine.Parameters.AddWithValue("@oid", newOrderId)
            cmdLine.Parameters.AddWithValue("@pid", CInt(row("PRODUCT_ID")))
            cmdLine.Parameters.AddWithValue("@qty", CInt(row("QUANTITY")))
            cmdLine.ExecuteNonQuery()
        Next
    End Using

    ' 4. Done - refresh the page.
    lblMsg.Text = "Order placed successfully!"
    lblMsg.ForeColor = System.Drawing.Color.Green
    Session("OrderLines") = Nothing
    rptOrderLines.DataSource = GetOrderLinesTable()
    rptOrderLines.DataBind()
    LoadOrdersHistory()

    ' 5. Show recommendations for the first product ordered.
    ShowRecommendations(CInt(lines.Rows(0)("PRODUCT_ID")))
End Sub

' —— Show "also bought" recommendations ——
Private Sub ShowRecommendations(productId As Integer)
    Dim recs As DataTable = ProductHelper.GetRecommendations(productId, 5)
    Dim msgs As DataTable = ProductHelper.FormatRecommendationMessages(productId,
recs)
    rptOrderRec.DataSource = msgs
    rptOrderRec.DataBind()
    pnlOrderRec.Visible = True

```

End Sub

End Class

File: Products.aspx

```
<%@ Page Title="Products" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Products.aspx.vb" Inherits="Lab11.Products" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
  <div class="content">
    <h3>Product Catalog</h3>
    <asp:GridView ID="gvProducts" runat="server" CssClass="pvf-grid"
AutoGenerateColumns="False" DataKeyNames="PRODUCT_ID"
OnSelectedIndexChanged="gvProducts_SelectedIndexChanged">
      <Columns>
        <asp:BoundField DataField="PRODUCT_ID" HeaderText="ID" />
        <asp:BoundField DataField="PRODUCT_DESCRIPTION" HeaderText="Description"
/>
        <asp:BoundField DataField="STANDARD_PRICE" HeaderText="Price"
DataFormatString="{0:C}" />
        <asp:CommandField ShowSelectButton="True" SelectText="View" />
      </Columns>
    </asp:GridView>

    <asp:Panel ID="pnlRecommend" runat="server" Visible="false"
CssClass="recommend">
      <h4>Customers who bought <asp:Label ID="lblRecProduct" runat="server" />
also bought:</h4>
      <asp:Repeater ID="rptRecommend" runat="server">
        <ItemTemplate>
          <div><%# Eval("PRODUCT_DESCRIPTION") %> - Bought together <b><%#
Eval("Cnt") %></b> times</div>
        </ItemTemplate>
      </asp:Repeater>
    </asp:Panel>
  </div>
</asp:Content>
```

File: Products.aspx.vb

```

' =====
' Products.aspx.vb - Browse products and see "also bought" recommendations
' =====

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Configuration

Public Partial Class Products
    Inherits System.Web.UI.Page

    Protected WithEvents gvProducts As Global.System.Web.UI.WebControls.GridView
    Protected WithEvents pnlRecommend As Global.System.Web.UI.WebControls.Panel
    Protected WithEvents lblRecProduct As Global.System.Web.UI.WebControls.Label
    Protected WithEvents rptRecommend As Global.System.Web.UI.WebControls.Repeater

    Private connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        ' Products page is public - no login required.
        If Not IsPostBack Then
            Using con As New SqlConnection(connectionString)
                Dim adapter As New SqlDataAdapter("SELECT PRODUCT_ID,
PRODUCT_DESCRIPTION, STANDARD_PRICE FROM Product_T", con)
                Dim dt As New DataTable()
                adapter.Fill(dt)
                gvProducts.DataSource = dt
                gvProducts.DataBind()
            End Using
        End If
    End Sub

    ' When user clicks "Select" on a product row.
    Protected Sub gvProducts_SelectedIndexChanged(sender As Object, e As EventArgs)
        Dim productId As Integer = Convert.ToInt32(gvProducts.SelectedDataKey.Value)
        Dim recs As DataTable = ProductHelper.GetRecommendations(productId, 5)

        If recs.Rows.Count > 0 Then
            lblRecProduct.Text = "<b>" & ProductHelper.GetProductName(productId) & "
</b>"

            rptRecommend.DataSource = recs
            rptRecommend.DataBind()
            pnlRecommend.Visible = True
        Else
            pnlRecommend.Visible = False
        End If
    End Sub

End Class

```

File: Register.aspx

```
<%@ Page Title="Register" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="Register.aspx.vb" Inherits="Lab11.Register" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
  <div class="mainBody">
    <h3 class="title">Register</h3>
    <asp:Label ID="lblRegMsg" runat="server" CssClass="msg-label"></asp:Label>

    <div class="form-group">
      <label>Name</label>
      <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>Address</label>
      <asp:TextBox ID="txtAddress" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>City</label>
      <asp:TextBox ID="txtCity" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>State</label>
      <asp:TextBox ID="txtState" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>Postal Code</label>
      <asp:TextBox ID="txtZip" runat="server"></asp:TextBox>
    </div>

    <hr />

    <div class="form-group">
      <label>Username</label>
      <asp:TextBox ID="txtUsername" runat="server"></asp:TextBox>
    </div>

    <div class="form-group">
      <label>Password</label>
      <asp:TextBox ID="txtPassword" runat="server" TextMode="Password">
</asp:TextBox>
    </div>

    <div class="form-group center">
      <asp:Button ID="btnRegister" runat="server" CssClass="pvf-btn" Text="Create
Account" OnClick="btnRegister_Click" />
    </div>
  </div>
</asp:Content>
```

File: Register.aspx.vb

```

' =====
' Register.aspx.vb - Code-behind for the Registration page
' =====
' Creates a customer record, then a linked user login record.
' =====

Imports System
Imports System.Data.SqlClient
Imports System.Configuration

Public Partial Class Register
    Inherits System.Web.UI.Page

    Protected WithEvents txtName As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtAddress As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtCity As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtState As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtZip As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtUsername As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtPassword As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents lblRegMsg As Global.System.Web.UI.WebControls.Label
    Protected WithEvents btnRegister As Global.System.Web.UI.WebControls.Button

    ' —— REGISTER BUTTON ——
    Protected Sub btnRegister_Click(sender As Object, e As EventArgs)

        Dim connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

        Try
            Using con As New SqlConnection(connectionString)
                con.Open()

                ' Insert customer and get back the new ID.
                Dim cmdCust As New SqlCommand(
                    "INSERT INTO customer_T (CUSTOMER_NAME, CUSTOMER_ADDRESS,
CUSTOMER_CITY, CUSTOMER_STATE, POSTAL_CODE) " &
                    "OUTPUT INSERTED.CUSTOMER_ID VALUES (@n,@a,@c,@s,@z)", con)
                cmdCust.Parameters.AddWithValue("@n", txtName.Text.Trim())
                cmdCust.Parameters.AddWithValue("@a", txtAddress.Text.Trim())
                cmdCust.Parameters.AddWithValue("@c", txtCity.Text.Trim())
                cmdCust.Parameters.AddWithValue("@s", txtState.Text.Trim())
                cmdCust.Parameters.AddWithValue("@z", txtZip.Text.Trim())
                Dim newCustId As Integer = Convert.ToInt32(cmdCust.ExecuteScalar())

                ' Insert user linked to that customer.
                Dim cmdUser As New SqlCommand(
                    "INSERT INTO USERS (Username, User_Password, User_Role, CUSTOMER_ID)
" &
                    "VALUES (@u, @p, 'Customer', @cid)", con)
                cmdUser.Parameters.AddWithValue("@u", txtUsername.Text.Trim())
                cmdUser.Parameters.AddWithValue("@p", txtPassword.Text.Trim())
                cmdUser.Parameters.AddWithValue("@cid", newCustId)
                cmdUser.ExecuteNonQuery()

                lblRegMsg.ForeColor = Drawing.Color.Green
                lblRegMsg.Text = "Account created successfully. You can login now."
            End Using
        End Try
    End Sub
End Class

```

```
    Catch ex As Exception
        lblRegMsg.ForeColor = Drawing.Color.Red
        lblRegMsg.Text = "Error: " & ex.Message
    End Try

End Sub

End Class
```

File: Site.Master

```
<%@ Master Language="VB" AutoEventWireup="true" CodeBehind="Site.Master.vb"
Inherits="Lab11.SiteMaster" %>

<!DOCTYPE html>
<html lang="en">
<head runat="server">
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Pine Valley Furniture</title>
    <link runat="server" rel="stylesheet" type="text/css" href="~/Content/PVF.css" />
    <asp:ContentPlaceHolder ID="head" runat="server"></asp:ContentPlaceHolder>
</head>
<body>
    <form runat="server">

        <div class="pvf-header">
            <h1>Pine Valley Furniture</h1>
        </div>

        <nav class="pvf-nav">
            <ul>
                <li><a href="Default.aspx">Home</a></li>
                <li>
                    <asp:Panel ID="pnlRegisterLink" runat="server"
style="display:inline;">
                        <a href="Register.aspx">Register</a>
                    </asp:Panel>
                </li>
                <li><a href="Products.aspx">Products</a></li>
                <asp:Panel ID="pnlProtectedLinks" runat="server"
style="display:contents;">
                    <li><a href="UpdateCustomer.aspx">Update Customer</a></li>
                    <li><a href="Orders.aspx">Orders</a></li>
                </asp:Panel>
                <li><asp:HyperLink ID="pnlCatalog" runat="server"
NavigateUrl="Admin.aspx">Inventory Helper</asp:HyperLink></li>
                <li><a href="Help.aspx">Help</a></li>
                <li style="margin-left: auto; display: flex; align-items: center;">
                    <asp:Label ID="lblUserInfo" runat="server" style="padding: 0 15px;
font-weight: bold; color: #4a3f2f;"></asp:Label>
                    <asp:HyperLink ID="lnkLogin" runat="server"
NavigateUrl="Login.aspx">Login</asp:HyperLink>
                    <asp:LinkButton ID="lnkLogout"
runat="server">Logout</asp:LinkButton>
                </li>
            </ul>
        </nav>

        <div class="pvf-content">
            <asp:ContentPlaceHolder ID="MainContent" runat="server">
</asp:ContentPlaceHolder>
            <hr />
            <footer class="center">
                <p>&copy; <%: DateTime.Now.Year %> - Pine Valley Furniture</p>
            </footer>
        </div>

    </form>
```

```
</body>  
</html>
```

File: Site.Master.vb

```
' =====  
' Site.Master.vb - Master page: controls the nav bar on every page  
' =====  
  
Public Partial Class SiteMaster  
    Inherits System.Web.UI.MasterPage  
  
    Protected WithEvents pnlRegisterLink As Global.System.Web.UI.WebControls.Panel  
    Protected WithEvents pnlProtectedLinks As Global.System.Web.UI.WebControls.Panel  
    Protected WithEvents pnlCatalog As Global.System.Web.UI.WebControls.HyperLink  
    Protected WithEvents lnkLogin As Global.System.Web.UI.WebControls.HyperLink  
    Protected WithEvents lnkLogout As Global.System.Web.UI.WebControls.LinkButton  
    Protected WithEvents lblUserInfo As Global.System.Web.UI.WebControls.Label  
  
    Protected Sub Page_Load(sender As Object, e As System.EventArgs) Handles Me.Load  
  
        Dim loggedIn As Boolean = AccessControl.IsLoggedIn(Me.Page)  
        Dim isStaff As Boolean = AccessControl.HasRole(Me.Page, "Admin", "Manager",  
"Employee")  
  
        ' Show/hide nav links based on login state and role.  
        pnlRegisterLink.Visible = Not loggedIn  
        pnlProtectedLinks.Visible = loggedIn  
        pnlCatalog.Visible = isStaff  
        lnkLogin.Visible = Not loggedIn  
        lnkLogout.Visible = loggedIn  
  
        If loggedIn Then  
            lblUserInfo.Text = "Welcome, " & Session("Username").ToString()  
        Else  
            lblUserInfo.Text = ""  
        End If  
  
    End Sub  
  
    Protected Sub lnkLogout_Click(sender As Object, e As System.EventArgs) Handles  
lnkLogout.Click  
        Session.Clear()  
        Session.Abandon()  
        Response.Redirect("Login.aspx")  
    End Sub  
  
End Class
```

File: UpdateCustomer.aspx

```
<%@ Page Title="Update Customer" Language="VB" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="UpdateCustomer.aspx.vb"
Inherits="Lab11.UpdateCustomer" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">
    <div style="padding: 20px; max-width: 500px; font-family: Arial;">
        <h2>Update Customer Profile</h2>
        <hr />

        <!-- Admin dropdown: select customer to update -->
        <asp:Panel ID="pnlAdminSearch" runat="server" style="margin-bottom: 20px;
padding: 15px; background-color: #f5f5f5;">
            <label><b>Select Customer to Update:</b></label><br />
            <asp:DropDownList ID="ddlCustomers" runat="server" Style="width: 100%;
padding: 8px; margin-top: 5px;"
                AutoPostBack="true"
                OnSelectedIndexChanged="ddlCustomers_SelectedIndexChanged">
                </asp:DropDownList>
            </asp:Panel>

        <!-- Customer info form -->
        <div style="margin-bottom: 15px;">
            <label><b>Full Name:</b></label><br />
            <asp:TextBox ID="txtName" runat="server" Style="width: 100%; padding: 8px;
margin-top: 5px;" />
        </div>

        <div style="margin-bottom: 15px;">
            <label><b>Address:</b></label><br />
            <asp:TextBox ID="txtAddress" runat="server" Style="width: 100%; padding:
8px; margin-top: 5px;" />
        </div>

        <div style="margin-bottom: 15px;">
            <label><b>City:</b></label><br />
            <asp:TextBox ID="txtCity" runat="server" Style="width: 100%; padding: 8px;
margin-top: 5px;" />
        </div>

        <div style="margin-bottom: 15px;">
            <label><b>State:</b></label><br />
            <asp:TextBox ID="txtState" runat="server" MaxLength="2" Style="width: 80px;
padding: 8px; margin-top: 5px;" />
        </div>

        <div style="margin-bottom: 15px;">
            <label><b>Postal Code:</b></label><br />
            <asp:TextBox ID="txtZip" runat="server" Style="width: 100%; padding: 8px;
margin-top: 5px;" />
        </div>

        <div style="margin-top: 25px;">
            <asp:Button ID="btnUpdate" runat="server" Text="Update Profile"
                OnClick="btnUpdate_Click" Style="padding: 10px 20px; background-
color: #4a3f2f; color: white; cursor: pointer; border: none; font-size: 14px;" />
        </div>

        <br />
        <asp:Label ID="lblMsg" runat="server" Style="font-weight: bold;"></asp:Label>
    </div>
</asp:Content>
```

```
</div>  
</asp:Content>
```

File: UpdateCustomer.aspx.vb

```

' =====
' UpdateCustomer.aspx.vb - Code-behind for the Update Customer page
' =====
' WHAT THIS PAGE DOES:
'   Regular users edit their own profile.
'   Admins pick any customer from a dropdown and edit their profile.
' =====

Imports System.Configuration
Imports System.Data
Imports System.Data.SqlClient

Public Class UpdateCustomer
    Inherits System.Web.UI.Page

    Protected WithEvents pnlAdminSearch As Global.System.Web.UI.WebControls.Panel
    Protected WithEvents ddlCustomers As Global.System.Web.UI.WebControls.DropDownList
    Protected WithEvents txtName As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtAddress As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtCity As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtState As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents txtZip As Global.System.Web.UI.WebControls.TextBox
    Protected WithEvents lblMsg As Global.System.Web.UI.WebControls.Label
    Protected WithEvents btnUpdate As Global.System.Web.UI.WebControls.Button

    Private ReadOnly connectionString As String =
ConfigurationManager.ConnectionStrings("PVFC").ConnectionString

    ' ----- PAGE LOAD -----
    Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
        AccessControl.RequireLogin(Me)

        If Not IsPostBack Then
            ' Admin -> show dropdown. Regular user -> load their own profile.
            pnlAdminSearch.Visible = AccessControl.ShouldShowAdminView(Me)

            If pnlAdminSearch.Visible Then
                LoadCustomerDropdown()
            Else
                LoadCustomerData(AccessControl.GetCustomerId(Me))
            End If
        End If
    End Sub

    ' ----- Fill the customer dropdown -----
    Private Sub LoadCustomerDropdown()
        Using con As New SqlConnection(connectionString)
            Dim adapter As New SqlDataAdapter("SELECT CUSTOMER_ID, CUSTOMER_NAME FROM
customer_T ORDER BY CUSTOMER_NAME", con)
            Dim dt As New DataTable()
            adapter.Fill(dt)

            ddlCustomers.DataSource = dt
            ddlCustomers.DataTextField = "CUSTOMER_NAME"
            ddlCustomers.DataValueField = "CUSTOMER_ID"
            ddlCustomers.DataBind()
            ddlCustomers.Items.Insert(0, New ListItem("-- Select Customer --", "0"))
        End Using
    End Sub

```

```

' —— Admin picks a customer from dropdown ——
Protected Sub ddlCustomers_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles ddlCustomers.SelectedIndexChanged
    If ddlCustomers.SelectedValue = "0" Then
        ClearForm()
        lblMsg.Text = "Select a customer."
    Else
        LoadCustomerData(CInt(ddlCustomers.SelectedValue))
    End If
End Sub

' —— Load a customer's data into the form ——
Private Sub LoadCustomerData(customerId As Integer)
    If customerId <= 0 Then
        lblMsg.Text = "Error: no customer ID available."
        Return
    End If

    Using con As New SqlConnection(connectionString)
        Dim cmd As New SqlCommand(
            "SELECT CUSTOMER_NAME, CUSTOMER_ADDRESS, CUSTOMER_CITY, CUSTOMER_STATE,
POSTAL_CODE " &
            "FROM customer_T WHERE CUSTOMER_ID = @id", con)
        cmd.Parameters.AddWithValue("@id", customerId)

        con.Open()
        Dim reader As SqlDataReader = cmd.ExecuteReader()

        If reader.Read() Then
            txtName.Text = reader("CUSTOMER_NAME").ToString()
            txtAddress.Text = reader("CUSTOMER_ADDRESS").ToString()
            txtCity.Text = reader("CUSTOMER_CITY").ToString()
            txtState.Text = reader("CUSTOMER_STATE").ToString()
            txtZip.Text = reader("POSTAL_CODE").ToString()
            lblMsg.Text = "Loaded. Edit and click Update."
        Else
            ClearForm()
            lblMsg.Text = "Customer not found."
        End If

        reader.Close()
    End Using
End Sub

' —— Clear all form fields ——
Private Sub ClearForm()
    txtName.Text = ""
    txtAddress.Text = ""
    txtCity.Text = ""
    txtState.Text = ""
    txtZip.Text = ""
End Sub

' —— UPDATE BUTTON ——
Protected Sub btnUpdate_Click(sender As Object, e As EventArgs)

    ' One line replaces the entire admin-vs-customer branching block.
    Dim customerId As Integer = AccessControl.ResolveCustomerId(Me, ddlCustomers)
    If customerId = -1 Then
        lblMsg.Text = "Please select a customer first."
        Return
    End If

    Using con As New SqlConnection(connectionString)

```

```
Dim cmd As New SqlCommand(
    "UPDATE customer_T SET CUSTOMER_NAME=@n, CUSTOMER_ADDRESS=@a, " &
    "CUSTOMER_CITY=@c, CUSTOMER_STATE=@s, POSTAL_CODE=@z WHERE
CUSTOMER_ID=@id", con)
cmd.Parameters.AddWithValue("@n", txtName.Text)
cmd.Parameters.AddWithValue("@a", txtAddress.Text)
cmd.Parameters.AddWithValue("@c", txtCity.Text)
cmd.Parameters.AddWithValue("@s", txtState.Text)
cmd.Parameters.AddWithValue("@z", txtZip.Text)
cmd.Parameters.AddWithValue("@id", customerId)

con.Open()
If cmd.ExecuteNonQuery() > 0 Then
    lblMsg.Text = "Profile updated successfully!"
    lblMsg.ForeColor = Drawing.Color.Green
Else
    lblMsg.Text = "Update failed."
    lblMsg.ForeColor = Drawing.Color.Red
End If
End Using
End Sub

End Class
```