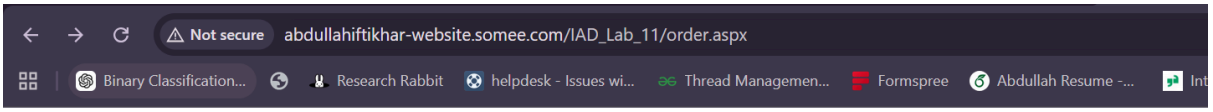

Selected Product Details

Product ID: 21
Description: Garden Table
Finish: Teak
Unit Price: $320.00

Customer ID:
19

Quantity:
2

Total: $320.00

Proceed to Payment

Related Products You Might Like

ID	Description	Finish	Price	
20	Bench	Teak	$210.00	View Product
22	Outdoor Table	Teak	$290.00	View Product
15	Footstool	Modern White	$80.00	View Product

Web hosting by Somee.com

Code :

Order.aspx:

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

<!DOCTYPE html>
<html>
<head runat="server">
   <title>Place Your Order</title>

</head>
<body>
   <form id="form1" runat="server">
      <div class="order-container">
         <h2>Selected Product Details</h2>
         <asp:Label ID="lblid" runat="server" CssClass="data-label"></asp:Label><br />
         <asp:Label ID="lblDesc" runat="server"></asp:Label><br />
         <asp:Label ID="lblFin" runat="server"></asp:Label><br />
         <asp:Label ID="lblPrice" runat="server"></asp:Label><br />
         <asp:HiddenField ID="hfPrice" runat="server" />

         <hr />

         <p>
            <asp:Label runat="server" Text="Customer ID:"></asp:Label><br />
```

```
            <asp:TextBox ID="CustID_TB" runat="server" TextMode="Number"
required="required"></asp:TextBox>
        </p>

        <p>
            <asp:Label runat="server" Text="Quantity:"></asp:Label><br />
            <asp:TextBox ID="Quantity_TB" runat="server" TextMode="Number" Min="1"
Text="1" AutoPostBack="true"></asp:TextBox>
        </p>

        <h3><asp:Label ID="lbltotal" runat="server" Text="Total: $0.00"></asp:Label></h3>

        <asp:Button ID="btnProceed" runat="server" Text="Proceed to Payment"
Height="40px" Width="16%" />

        <!-- Inside the order-container div, after the btnProceed -->
<hr />
<div class="recommendations-section">
  <h3>Related Products You Might Like</h3>
  <asp:GridView ID="gvRecommendations" runat="server" AutoGenerateColumns="False"
    CssClass="table"
OnSelectedIndexChanged="gvRecommendations_SelectedIndexChanged"
DataKeyNames="ID">
    <Columns>
      <asp:BoundField DataField="ID" HeaderText="ID" />
      <asp:BoundField DataField="Description" HeaderText="Description" />
      <asp:BoundField DataField="Finish" HeaderText="Finish" />
      <asp:BoundField DataField="Price" HeaderText="Price" DataFormatString="${0:n2}"
/>
      <asp:CommandField ShowSelectButton="True" SelectText="View Product" />
    </Columns>
  </asp:GridView>
</div>
    </div>
  </form>
</body>
</html>
```

Order.aspx.vb:

```
Imports System.Net
Imports System.Web.Script.Serialization
Imports System.Data
Imports System.Data.SqlClient

Partial Class OrderPage
```

```
Inherits System.Web.UI.Page

' Property names must match API JSON
Public Class RecommendationResult
    Public Property ID As Integer
    Public Property Description As String
    Public Property Finish As String
    Public Property Price As Decimal
End Class

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

        ' 1. Handle Customer Session
        If Session("CustomerID") IsNot Nothing Then
            CustID_TB.Text = Session("CustomerID").ToString()
            CustID_TB.Enabled = False
        End If

        ' 2. Load Selected Product Details
        If Session("ProductID") IsNot Nothing Then

            lblid.Text = "Product ID: " & Session("ProductID").ToString()
            lblDesc.Text = "Description: " & Session("ProductName").ToString()
            lblFin.Text = "Finish: " & Session("ProductFinish").ToString()
            lblPrice.Text = "Unit Price: $" & Session("ProductPrice").ToString()
            hfPrice.Value = Session("ProductPrice").ToString()

            UpdateTotal()

            '  CALL API
            FetchRecommendations()

        Else
            Response.Redirect("home.aspx")
        End If
    End If
End Sub

' ================================
'  RECOMMENDATION SYSTEM LOGIC
' ================================

Private Sub FetchRecommendations()
    Try
```

```
        '  Required for HTTPS (Render APIs)
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12

        ' Encode values safely
        Dim finish As String =
Uri.EscapeDataString(Session("ProductFinish").ToString())
        Dim name As String =
Uri.EscapeDataString(Session("ProductName").ToString())
        Dim id As String = Session("ProductID").ToString()

        ' Build API URL
        Dim apiUrl As String = String.Format(

"https://product-recomendation-api.onrender.com/recommendations/{0}/{1}/{2}",
            finish, name, id
        )

        ' Call API
        Using client As New WebClient()
            Dim json As String = client.DownloadString(apiUrl)

            '  DEBUG (you can comment this later)
            ' Response.Write("<pre>" & json & "</pre>")

            If Not String.IsNullOrEmpty(json) Then
                Dim js As New JavaScriptSerializer()

                Dim results As List(Of RecommendationResult) =
                    js.Deserialize(Of List(Of RecommendationResult))(json)

                ' Bind to GridView
                gvRecommendations.DataSource = results
                gvRecommendations.DataBind()
            Else
                Response.Write("⚠ API returned empty response")
            End If
        End Using

    Catch ex As Exception
        ' DON'T hide errors anymore
        Response.Write("<br/> ERROR: " & ex.Message)
    End Try
End Sub

' ================================
```

```
    '  WHEN USER SELECTS RECOMMENDATION
    ' ================================

    Protected Sub gvRecommendations_SelectedIndexChanged(sender As Object,
e As EventArgs)
        Dim row As GridViewRow = gvRecommendations.SelectedRow

        Session("ProductID") =
gvRecommendations.SelectedDataKey.Value.ToString()


Session("ProductName") = row.Cells(1).Text   ' Description
Session("ProductFinish") = row.Cells(2).Text ' Finish

Dim priceText As String = row.Cells(3).Text  ' Price
Session("ProductPrice") = priceText.Replace("$", "").Trim()



        Response.Redirect("order.aspx")
    End Sub

    ' ================================
    '  ORDER CALCULATION
    ' ================================

    Protected Sub Quantity_TB_TextChanged(sender As Object, e As EventArgs)
Handles Quantity_TB.TextChanged
        UpdateTotal()
    End Sub

    Private Sub UpdateTotal()
        Dim qty As Integer = If(IsNumeric(Quantity_TB.Text),
Convert.ToInt32(Quantity_TB.Text), 1)
        Dim price As Double = If(IsNumeric(hfPrice.Value),
Convert.ToDouble(hfPrice.Value), 0)

        lbltotal.Text = "Total: $" & (qty * price).ToString("0.00")
    End Sub

    ' ================================
    '  PROCEED BUTTON
    ' ================================

    Protected Sub btnProceed_Click(sender As Object, e As EventArgs) Handles
btnProceed.Click
```

```
        Session("OrderQuantity") = Quantity_TB.Text
        Session("CustomerID") = CustID_TB.Text

        Response.Redirect("payment.aspx")
    End Sub

End Class
```

Recommendation System API code in C#:

```
using Microsoft.Data.SqlClient;

var builder = WebApplication.CreateBuilder(args);
var connString = builder.Configuration.GetConnectionString("PVFConnection");
builder.Services.AddOpenApi();
var app = builder.Build();

app.MapGet("/recommendations/{finish}/{productName}/{currentId}", async (string
finish, string productName, int currentId) =>
{
    var recommendations = new List<object>();

    // 1. The Intelligent Dictionary (Mapping: Current -> Recommended)
    var complementaryRules = new Dictionary<string,
string>(StringComparer.OrdinalIgnoreCase)
    {
        { "Dining Table", "Dining Chair" },
        { "Sofa", "Side Table" },
        { "Study Table", "Study Chair" },
        { "Bench", "Garden Table" },
        { "Single Bed", "Bedside Table" },
        { "Arm Chair", "Footstool" },
        { "Dressing Table", "Chest of Drawers" }
    };

    // 2. Identify the target item based on the dictionary
    string targetCategory = "";
    foreach (var rule in complementaryRules)
    {
        if (productName.Contains(rule.Key, StringComparison.OrdinalIgnoreCase))
        {
            targetCategory = rule.Value;
            break;
        }
```

```
    }

    using (SqlConnection conn = new SqlConnection(connString))
    {
        // 3. The Weighted SQL Query
        // Points: +10 for Dictionary Match, +5 for Finish Match
        string query = @"
            SELECT TOP 3 Product_Id, Product_Description, Product_Finish,
Standard_Price,
            (
                (CASE WHEN Product_Description LIKE @target THEN 10 ELSE 0 END) +
                (CASE WHEN Product_Finish = @finish THEN 5 ELSE 0 END)
            ) AS MatchScore
            FROM PRODUCT_t
            WHERE Product_Id <> @id
            AND (Product_Description LIKE @target OR Product_Finish = @finish)
            ORDER BY MatchScore DESC, Standard_Price ASC";

        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@target", "%" + targetCategory + "%");
        cmd.Parameters.AddWithValue("@finish", finish);
        cmd.Parameters.AddWithValue("@id", currentId);

        await conn.OpenAsync();
        using (var reader = await cmd.ExecuteReaderAsync())
        {
            while (await reader.ReadAsync())
            {
                recommendations.Add(new
                {
                    ID = reader["Product_Id"],
                    Description = reader["Product_Description"].ToString().Trim(),
                    Finish = reader["Product_Finish"].ToString().Trim(),
                    Price = reader["Standard_Price"],
                    Score = reader["MatchScore"] // Useful for debugging the 'intelligence'
                });
            }
        }
    }

    return recommendations.Count > 0 ? Results.Ok(recommendations) :
Results.NotFound();
});

app.Run();
```